

CONTENIDO

CONTENIDO.....	1
CONTENIDO DE TABLAS	5
CONTENIDO DE GRÁFICOS.....	11
RESUMEN	24
CAPITULO 1. ANTECEDENTES.....	25
1.1. INTRODUCCIÓN	25
1.2. OBJETIVO GENERAL	27
1.3. OBJETIVOS ESPECÍFICOS.....	27
1.4. DEFINICIÓN DEL PROBLEMA.....	27
1.5. DESCRIPCIÓN DEL PROYECTO	28
1.6. JUSTIFICACIÓN DEL PROYECTO	28
1.7. ALCANCE DEL PROYECTO	29
CAPITULO 2. FUNDAMENTACIÓN TEÓRICA.....	31
2.1. METODOLOGÍA RUP	31
2.1.1. HISTORIA.....	31
2.1.2. INTRODUCCIÓN.....	32
2.1.3. CARACTERÍSTICAS	33
2.1.4. CICLO DE VIDA DEL PROCESO UNIFICADO	34
2.1.4.1. INICIO.....	35
2.1.4.2. ELABORACIÓN.....	35
2.1.4.3. CONSTRUCCIÓN	35
2.1.4.4. TRANSICIÓN	36
2.1.5. FLUJOS DE TRABAJO.....	36
2.1.5.1. REQUISITOS	36
2.1.5.2. ANÁLISIS	37
2.1.5.3. DISEÑO.....	37
2.1.5.4. IMPLEMENTACIÓN	38
2.1.5.5. PRUEBAS	38

2.2. METODOLOGÍA OOADM.....	39
2.2.1. DISEÑO CONCEPTUAL.....	40
2.2.2. DISEÑO NAVEGACIONAL.....	40
2.2.3. DISEÑO DE INTERFACES ABSTRACTAS.....	41
2.2.4. IMPLEMENTACIÓN.....	41
2.3. LENGUAJE UNIFICADO DE MODELAMIENTO - UML.....	42
2.3.1. JERARQUÍA DE LOS DIAGRAMAS UML	42
2.3.1.1. DIAGRAMAS DE ESTRUCTURA.....	42
2.3.1.2. DIAGRAMAS DE COMPORTAMIENTO.....	43
2.3.1.3. DIAGRAMAS DE INTERACCIÓN	43
2.4. PATRONES DE DISEÑO.....	43
2.4.1. CLASIFICACIÓN DE LOS PATRONES DE DISEÑO	44
2.5. MODELO MVC.....	45
2.6. LENGUAJE DE PROGRAMACIÓN JAVA	46
2.6.1. HISTORIA DE JAVA	46
2.6.2. CARACTERÍSTICAS	47
2.6.3. PLATAFORMAS	48
2.6.4. TECNOLOGÍAS.....	49
2.7. BASE DE DATOS ORACLE 10G.....	50
2.7.1. CARACTERÍSTICAS	50
2.7.2. CAPAS.....	51
2.7.3. VERSIONES.....	52
2.8. SISTEMA OPERATIVO LINUX	52
2.8.1. HISTORIA.....	52
2.8.2. CARACTERÍSTICAS	52
2.8.3. DISTRIBUCIONES	53
CAPITULO 3. ANÁLISIS DEL SISTEMA	55
3.1. ANÁLISIS	55
3.2. DIAGRAMA DE PROCESOS.....	57
3.2.1. CADENA DE VALOR.....	57
3.2.2. PROCESOS DE NEGOCIO.....	57

3.3. DIAGRAMAS UML	61
3.3.1. CASOS DE USO DEL NEGOCIO.....	61
3.3.2. DIAGRAMAS DE ACTIVIDADES DEL NEGOCIO.....	69
3.3.3. DIAGRAMAS DE CASOS DE USO DEL SISTEMA.....	75
3.3.3.1. DEFINICIÓN DE ACTORES DEL SISTEMA	76
3.3.4. DIAGRAMAS DE SECUENCIA DEL SISTEMA.....	77
3.4. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE	78
3.4.1. INTRODUCCIÓN	78
3.4.1.1. PROPÓSITO	78
3.4.1.2. ALCANCE.....	79
3.4.1.3. PERSONAL INVOLUCRADO.....	79
3.4.1.4. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	80
3.4.2. DESCRIPCIÓN GENERAL.....	80
3.4.2.1. PERSPECTIVA DEL PRODUCTO	80
3.4.2.2. FUNCIONALIDAD DEL PRODUCTO	81
3.4.2.3. CARACTERÍSTICAS DE LOS USUARIOS	81
3.4.2.4. RESTRICCIONES	82
3.4.2.5. SUPOSICIONES Y DEPENDENCIAS.....	82
3.4.2.6. EVOLUCIÓN PREVISIBLE DEL SISTEMA	82
3.4.3. REQUISITOS ESPECÍFICOS.....	82
3.4.4. REQUISITOS COMUNES DE LAS INTERFACES	86
3.4.4.1. INTERFACES DE USUARIO.....	86
3.4.4.2. INTERFAZ DE HARDWARE	86
3.4.4.3. INTERFACES DE COMUNICACIÓN.....	86
3.4.5. REQUISITOS FUNCIONALES	86
3.4.6. REQUISITOS NO FUNCIONALES	87
CAPÍTULO 4. DISEÑO DEL SISTEMA	90
4.1. DISEÑO CONCEPTUAL.....	90
4.2. DISEÑO NAVEGACIONAL	91
4.2.1. ESQUEMA DE CONTEXTO NAVEGACIONAL.....	91
4.2.2. ESQUEMA DE CLASES NAVEGACIONAL.....	101
4.3. DISEÑO DE INTERFAZ.....	105

4.4. DISEÑO DE BASE DATOS.....	107
CAPÍTULO 5. IMPLEMENTACIÓN DEL SISTEMA	108
5.1. MODELO VISTA CONTROLADOR	108
5.1.1. INTERFAZ DE USUARIO	108
5.1.2. LÓGICA DE NEGOCIOS	144
5.1.3. GESTOR DE COMUNICACIÓN	148
5.1.4. CONTROLADOR	153
5.2. PRUEBAS	156
CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES.....	157
6.1. CONCLUSIONES	157
6.2. RECOMENDACIONES	158
REFERENCIAS BIBLIOGRÁFICAS	159
ANEXOS	160
ANEXO 1. RESULTADO DE ENCUESTAS	160
ANEXO 2. MANUAL TÉCNICO.....	165
ANEXO 3. MANUAL DE INSTALACIÓN.....	341
ANEXO 4. MANUAL DE USUARIO	353
ANEXO 5. CD CON LOS INTALADORES DEL SISTEMA	

CONTENIDO DE TABLAS

Tabla 2.1. “Patrones de Diseño”	45
Tabla 3.1. Definición del problema	56
Tabla 3.2. Procesos de Negocio	61
Tabla 3.3. Identificación de casos de uso de negocio	63
Tabla 3.4. Definición de actores del sistema	76
Tabla 3.5. Especificación de caso de uso: Ingresar al sistema	77
Tabla 3.6. Personal involucrado 1	79
Tabla 3.7. Personal involucrado	80
Tabla 3.8. Características del usuario Cajero.....	81
Tabla 3.9. Características del usuario Administrador	82
Tabla 3.10. Requisitos específicos - 1	83
Tabla 3.11. Requisitos específicos – 2.....	83
Tabla 3.12. Requisitos específicos – 3.....	84
Tabla 3.13. Requisitos específicos – 4.....	84
Tabla 3.14. Requisitos específicos – 5.....	85
Tabla 3.15. Requisitos específicos – 6.....	85
Tabla 7.1. Especificación de caso de uso: Ingresar empresa	169
Tabla 7.2. Especificación de caso de uso: Consultar empresa	170
Tabla 7.3. Especificación de caso de uso: Modificar empresa	170
Tabla 7.4. Especificación de caso de uso: Ingresar sucursal	172
Tabla 7.5. Especificación de caso de uso: Consultar sucursal.....	172
Tabla 7.6. Especificación de caso de uso: Modificar sucursal.....	173
Tabla 7.7. Especificación de caso de uso: Eliminar sucursal	173
Tabla 7.8. Especificación de caso de uso: Ingresar ubicación geográfica	175
Tabla 7.9. Especificación de caso de uso: Consultar ubicación geográfica	175

Tabla 7.10. Especificación de caso de uso: Modificar ubicación geográfica	176
Tabla 7.11. Especificación de caso de uso: Eliminar ubicación geográfica.....	177
Tabla 7.12. Especificación de caso de uso: Ingresar forma de pago	178
Tabla 7.13. Especificación de caso de uso: Consultar forma de pago	178
Tabla 7.14. Especificación de caso de uso: Modificar forma de pago.....	179
Tabla 7.15. Especificación de caso de uso: Eliminar forma de pago	179
Tabla 7.16. Especificación de caso de uso: Ingresar impuesto.....	180
Tabla 7.17. Especificación de caso de uso: Consultar impuesto.....	181
Tabla 7.18. Especificación de caso de uso: Modificar impuesto	181
Tabla 7.19. Especificación de caso de uso: Eliminar impuesto	182
Tabla 7.20. Especificación de caso de uso: Ingresar descuentos y recargos ..	183
Tabla 7.21. Especificación de caso de uso: Consultar descuentos y recargos	183
Tabla 7.22. Especificación de caso de uso: Modificar descuentos y recargos .	184
Tabla 7.23. Especificación de caso de uso: Eliminar descuentos y recargos ..	184
Tabla 7.24. Especificación de caso de uso: Crear perfil.....	185
Tabla 7.25. Especificación de caso de uso: Consultar perfil	186
Tabla 7.26. Especificación de caso de uso: Modificar perfil	187
Tabla 7.27. Especificación de caso de uso: Eliminar perfil.....	187
Tabla 7.28. Especificación de caso de uso: Ingresar usuario	188
Tabla 7.29. Especificación de caso de uso: Consultar usuario	189
Tabla 7.30. Especificación de caso de uso: Modificar usuario	189
Tabla 7.31. Especificación de caso de uso: Eliminar usuario.....	190
Tabla 7.32. Especificación de caso de uso: Generar respaldos.....	191
Tabla 7.33. Especificación de caso de uso: Ingresar línea de producto.....	192
Tabla 7.34. Especificación de caso de uso: Consultar línea de producto	192
Tabla 7.35. Especificación de caso de uso: Modificar línea de producto	193

Tabla 7.36. Especificación de caso de uso: Eliminar línea de producto.....	194
Tabla 7.37. Especificación de caso de uso: Ingresar grupo de producto	195
Tabla 7.38. Especificación de caso de uso: Consultar grupo de producto	195
Tabla 7.39. Especificación de caso de uso: Modificar grupo de producto.....	196
Tabla 7.40. Especificación de caso de uso: Eliminar grupo de producto	196
Tabla 7.41. Especificación de caso de uso: Ingresar producto	198
Tabla 7.42. Especificación de caso de uso: Consultar producto	198
Tabla 7.43. Especificación de caso de uso: Modificar producto	199
Tabla 7.44. Especificación de caso de uso: Eliminar producto	199
Tabla 7.45. Especificación de caso de uso: Ingresar receta	200
Tabla 7.46. Especificación de caso de uso: Consultar receta	201
Tabla 7.47. Especificación de caso de uso: Eliminar receta	201
Tabla 7.48. Especificación de caso de uso: Ingresar inventario inicial.....	203
Tabla 7.49. Especificación de caso de uso: Consultar inventario.....	203
Tabla 7.50. Especificación de caso de uso: Modificar inventario	204
Tabla 7.51. Especificación de caso de uso: Imprimir inventario	204
Tabla 7.52. Especificación de caso de uso: Ingresar toma física	206
Tabla 7.53. Especificación de caso de uso: Modificar toma física.....	206
Tabla 7.54. Especificación de caso de uso: Imprimir inventario	207
Tabla 7.55. Especificación de caso de uso: Ingresar movimiento de bodega ..	208
Tabla 7.56. Especificación de caso de uso: Eliminar movimiento de bodega ..	208
Tabla 7.57. Especificación de caso de uso: Imprimir movimiento de bodega ..	209
Tabla 7.58. Especificación de caso de uso: Ingresar preproducción.....	210
Tabla 7.59. Especificación de caso de uso: Consultar preproducción	211
Tabla 7.60. Especificación de caso de uso: Imprimir preproducción.....	211
Tabla 7.61. Especificación de caso de uso: Ingresar postproducción.....	213

Tabla 7.62. Especificación de caso de uso: Consultar postproducción.....	213
Tabla 7.63. Especificación de caso de uso: Imprimir postproducción	214
Tabla 7.64. Especificación de caso de uso: Ingresar cliente	215
Tabla 7.65. Especificación de caso de uso: Consultar cliente.....	215
Tabla 7.66. Especificación de caso de uso: Modificar cliente	216
Tabla 7.67. Especificación de caso de uso: Eliminar cliente	216
Tabla 7.68. Especificación de caso de uso: Ingresar Factura	218
Tabla 7.69. Especificación de caso de uso: Anular ítems del detalle de factura	218
Tabla 7.70. Especificación de caso de uso: Elegir Forma de Pago.....	219
Tabla 7.71. Especificación de caso de uso: Imprimir Factura	219
Tabla 7.72. Especificación de caso de uso: Guardar cuenta en espera.....	220
Tabla 7.73. Especificación de caso de uso: Ingresar descuentos y recargos ..	220
Tabla 7.74. Especificación de caso de uso: Consultar cuentas.....	221
Tabla 7.75. Especificación de caso de uso: Imprimir pre-cuenta	221
Tabla 7.76. Especificación de caso de uso: Anular Facturas.	222
Tabla 7.77. Especificación de caso de uso: Ingresar movimiento de caja.	222
Tabla 7.78. Especificación de caso de uso: Cerrar caja.....	223
Tabla 7.79. Especificación de caso de uso: Seleccionar reporte	224
Tabla 7.80. Especificación de caso de uso: Imprimir reporte	224
Tabla 7.81. Diccionario de Clase: Empresa	285
Tabla 7.82. Diccionario de Clase: Sucursal.....	285
Tabla 7.83. Diccionario de Clase: Zona	286
Tabla 7.84. Diccionario de Clase: País	286
Tabla 7.85. Diccionario de Clase: Provincia.....	287
Tabla 7.86. Diccionario de Clase: Ciudad	287

Tabla 7.87. Diccionario de Clase: Región	288
Tabla 7.88. Diccionario de Clase: Usuario	289
Tabla 7.89. Diccionario de Clase: Perfil	289
Tabla 7.90. Diccionario de Clase: Det_Acceso	290
Tabla 7.91. Diccionario de Clase: Det_Modulo_Sis	290
Tabla 7.92. Diccionario de Clase: Modulo_Sis	291
Tabla 7.93. Diccionario de Clase: Proceso	291
Tabla 7.94. Diccionario de Clase: Impuesto.....	292
Tabla 7.95. Diccionario de Clase: Descuento_Recargo	292
Tabla 7.96. Diccionario de Clase: Bodega	293
Tabla 7.97. Diccionario de Clase: Linea_Producto.....	294
Tabla 7.98. Diccionario de Clase: Grupo_Producto	294
Tabla 7.99. Diccionario de Clase: Producto	295
Tabla 7.100. Diccionario de Clase: Receta	296
Tabla 7.101. Diccionario de Clase: Det_Receta.....	296
Tabla 7.102. Diccionario de Clase: Produccion.....	297
Tabla 7.103. Diccionario de Clase: Det_Receta_Produccion.....	298
Tabla 7.104. Diccionario de Clase: Det_Produccion	299
Tabla 7.105. Diccionario de Clase: Mov_Bodega.....	301
Tabla 7.106. Diccionario de Clase: Toma_Fisica	301
Tabla 7.107. Diccionario de Clase: Det_Toma_Producto.....	302
Tabla 7.108. Diccionario de Clase: Det_Mov_Bodega	303
Tabla 7.109. Diccionario de Clase: Kardex	305
Tabla 7.110. Diccionario de Clase: Cliente	307
Tabla 7.111. Diccionario de Clase: Factura.....	308
Tabla 7.112. Diccionario de Clase: Det_Factura.....	308

Tabla 7.113. Diccionario de Clase: Det_Factura_Pago.....	309
Tabla 7.114. Diccionario de Clase: Cuadre_Caja.....	310
Tabla 7.115. Diccionario de Clase: Det_Cuadre_Caja.....	310
Tabla 7.116. Diccionario de Clase: Movimiento_Caja.....	311
Tabla 7.117. Casos de prueba – Caja Blanca.....	332
Tabla 7.118. Condiciones de Entrada – Caja Negra.....	333
Tabla 7.119. Casos de Prueba – Caja Negra.....	335
Tabla 7.120. Caso de Prueba - 1.....	336
Tabla 7.121. Caso de Prueba - 2.....	336
Tabla 7.122. Caso de Prueba - 3.....	337
Tabla 7.123. Caso de Prueba - 4.....	337
Tabla 7.124. Caso de Prueba - 5.....	338
Tabla 7.125. Caso de Prueba - 6.....	338
Tabla 7.126. Caso de Prueba - 7.....	339
Tabla 7.127. Caso de Prueba - 8.....	339
Tabla 7.128. Caso de Prueba - 9.....	340

CONTENIDO DE GRÁFICOS

Figura 2.1. Diagrama MVC.....	46
Figura 2.2. Capas de Oracle	51
Figura 3.1. Cadena de Valor	57
Figura 3.2.Caso de uso del negocio: Verificar existencias en bodega	64
Figura 3.3. Caso de uso del negocio: Recepción de mercadería del proveedor	64
Figura 3.4. Caso de uso del negocio: Guardar pedido en bodega	65
Figura 3.5. Caso de uso del negocio: Realizar toma física de los productos	65
Figura 3.6. Caso de uso del negocio: Realizar el pedido de productos a la matriz	66
Figura 3.7.Caso de uso del negocio: Despacho de pedidos a cada sucursal	66
Figura 3.8. Caso de uso del negocio: Entrega de pedidos en cada sucursal.....	67
Figura 3.9. Caso de uso del negocio: Toma de pedido al cliente.....	67
Figura 3.10. Caso de uso del negocio: Facturar el pedido al cliente (Servir para llevar)	68
Figura 3.11.Caso de uso del negocio: Facturar el pedido al cliente (Servir para la mesa)	68
Figura 3.12. Diagrama de Actividades: Verificar existencias en bodega.....	69
Figura 3.13. Diagrama de Actividades: Recepción de mercadería del proveedor	70
Figura 3.14. Diagrama de Actividades: Guardar el pedido en bodega	70
Figura 3.15. Diagrama de Actividades: Realizar toma física de los productos...	71
Figura 3.16. Diagrama de Actividades: Realizar el pedido de productos a la matriz	71
Figura 3.17. Diagrama de Actividades: Despacho de pedidos a cada sucursal.	72
Figura 3.18. Diagrama de Actividades: Entrega de pedido en cada sucursal	73
Figura 3.19. Diagrama de Actividades: Toma de pedido al cliente.....	73

Figura 3.20. Diagrama de Actividades: Facturar el pedido al cliente (Servir para llevar)	74
Figura 3.21. Diagrama de Actividades: Facturar el pedido al cliente (Servir para la mesa.....	75
Figura 3.22. Caso de Uso del Sistema: Ingreso al Sistema	76
Figura 3.23. Diagrama de secuencia: Ingreso al sistema.....	78
Figura 4.1. Diagrama: Estructura Organizacional.....	90
Figura 4.2. Esquema de Contexto Navegacional: Ingreso al Sistema.....	91
Figura 4.3. Esquema de Contexto Navegacional: Configuración de Empresa...	91
Figura 4.4. Esquema de Contexto Navegacional: Configuración de Sucursal ...	92
Figura 4.5. Esquema de Contexto Navegacional: Configuración de Ubicación Geográfica.....	92
Figura 4.6. Esquema de Contexto Navegacional: Configuración de Formas de Pago.....	93
Figura 4.7. Esquema de Contexto Navegacional: Configuración de Impuesto ..	93
Figura 4.8. Esquema de Contexto Navegacional: Configuración de Descuentos y Recargos.....	93
Figura 4.9. Esquema de Contexto Navegacional: Configuración Perfiles de Usuario.....	94
Figura 4.10. Esquema de Contexto Navegacional: Configuración de Usuario...	94
Figura 4.11. Esquema de Contexto Navegacional: Respaldo de Datos.....	94
Figura 4.12. Esquema de Contexto Navegacional: Configuración Línea de Productos	95
Figura 4.13. Esquema de Contexto Navegacional: Configuración Grupo de Productos	95
Figura 4.14. Esquema de Contexto Navegacional: Configuración Productos	95
Figura 4.15. Esquema de Contexto Navegacional: Configuración de Recetas ..	96

Figura 4.16. Esquema de Contexto Navegacional: Configuración Inventario Inicial	96
Figura 4.17. Esquema de Contexto Navegacional: Configuración Toma Física	97
Figura 4.18. Esquema de Contexto Navegacional: Configuración Movimiento de Bodega.....	97
Figura 4.19. Esquema de Contexto Navegacional: Configuración Preproducción	98
Figura 4.20. Esquema de Contexto Navegacional: Configuración Postproducción	98
Figura 4.21. Esquema de Contexto Navegacional: Configuración Cliente	99
Figura 4.22. Esquema de Contexto Navegacional: Configuración Factura	99
Figura 4.23. Esquema de Contexto Navegacional: Anular Factura.....	100
Figura 4.24. Esquema de Contexto Navegacional: Movimiento de Caja.....	100
Figura 4.25. Esquema de Contexto Navegacional: Cuadre de Caja	100
Figura 4.26. Esquema de clases navegacional 1.....	101
Figura 4.27. Esquema de clases navegacional 2.....	102
Figura 4.29. Esquema de clases navegacional 4.....	104
Figura 4.30. Interfaz: Ingreso al Sistema.....	105
Figura 4.31. Interfaz: Menú Principal.....	105
Figura 4.32. Interfaz: Menú de Administración	106
Figura 4.33. Interfaz: Menú de Inventario.....	106
Figura 4.34. Interfaz: Menú de Facturación.....	106
Figura 4.35. Módulo Administración – Estructura Organizacional	107
Figura 5.1. Interfaz: Facturación.....	108
Figura 6.1. Resultados – Pregunta 1	161
Figura 6.2. Resultados – Pregunta 2.....	162
Figura 6.3. Resultados – Pregunta 3.....	162

Figura 6.4. Resultados – Pregunta 4.....	162
Figura 6.5. Resultados – Pregunta 5.....	163
Figura 6.6. Resultados – Pregunta 6.....	163
Figura 6.7. Resultados – Pregunta 7.....	163
Figura 6.8. Resultados – Pregunta 8.....	164
Figura 6.9. Resultados – Pregunta 9.....	164
Figura 7.1. Caso de Uso del Sistema: Configuración de la Empresa.....	169
Figura 7.2. Caso de Uso del Sistema: Configuración de Sucursal.....	171
Figura 7.3. Caso de Uso del Sistema: Configuración de ubicación geográfica	174
Figura 7.4. Caso de Uso del Sistema: Configuración de forma de pago.....	177
Figura 7.5. Caso de Uso del Sistema: Configuración de impuesto	179
Figura 7.7. Caso de Uso del Sistema: Configuración perfiles de usuario.....	185
Figura 7.8. Caso de Uso del Sistema: Configurar usuario	188
Figura 7.9. Caso de Uso del Sistema: Generar respaldos	190
Figura 7.10. Caso de Uso del Sistema: Configuración Líneas de Productos ...	191
Figura 7.11. Caso de Uso del Sistema: Configuración Grupo de Productos	194
Figura 7.12. Caso de Uso del Sistema: Configuración de Productos.....	197
Figura 7.13. Caso de Uso del Sistema: Configuración de Recetas.....	199
Figura 7.14. Caso de Uso del Sistema: Configuración de inventario	202
Figura 7.15. Caso de Uso del Sistema: Configuración de toma física.....	205
Figura 7.16. Caso de Uso del Sistema: Configuración de movimiento de bodega	207
Figura 7.17. Caso de Uso del Sistema: Configuración de preproducción	209
Figura 7.18. Caso de Uso del Sistema: Configuración de postproducción.....	212
Figura 7.19. Caso de Uso del Sistema: Configuración de clientes.....	214
Figura 7.20. Caso de Uso del Sistema: Configuración de Factura.....	217

Figura 7.21. Caso de Uso del Sistema: Generar reportes.....	223
Figura 7.22. Modelo de Interfaz -1	225
Figura 7.23. Modelo de Interfaz -2	225
Figura 7.24. Modelo de Interfaz -3	226
Figura 7.25. Modelo de Dominio	227
Figura 7.26. Diagrama: Estructura Organizacional.....	228
Figura 7.27. Diagrama: Ubicación Geográfica.....	229
Figura 7.28. Diagrama: Perfil de Usuario	230
Figura 7.29. Diagrama: Recargos y Descuentos.....	231
Figura 7.30. Diagrama: Bodega	232
Figura 7.31. Diagrama: Producto	233
Figura 7.32. Diagrama: Receta	234
Figura 7.33. Diagrama: Producción.....	235
Figura 7.34. Diagrama: Inventario	236
Figura 7.36. Diagrama: Cliente.....	238
Figura 7.37. Diagrama: Factura.....	239
Figura 7.38. Diagrama: Cuadre de Caja.....	240
Figura 7.39. Diagrama: Movimiento de Caja	241
Figura 7.40. Diagrama de secuencia: Ingresar empresa.....	242
Figura 7.41. Diagrama de secuencia: Consultar empresa	242
Figura 7.42. Diagrama de secuencia: Modificar empresa	243
Figura 7.43. Diagrama de secuencia: Ingresar sucursal	243
Figura 7.44. Diagrama de secuencia: Consultar sucursal	244
Figura 7.45. Diagrama de secuencia: Modificar sucursal.....	244
Figura 7.46. Diagrama de secuencia: Eliminar sucursal	245
Figura 7.47. Diagrama de secuencia: Ingresar País	245

Figura 7.48. Diagrama de secuencia: Ingresar Región	246
Figura 7.49. Diagrama de secuencia: Ingresar Provincia.....	246
Figura 7.50. Diagrama de secuencia: Ingresar Ciudad	247
Figura 7.51. Diagrama de secuencia: Ingresar Zona	247
Figura 7.52. Diagrama de secuencia: Consultar Ubicación geográfica	248
Figura 7.53. Diagrama de secuencia: Modificar Ubicación geográfica	248
Figura 7.54. Diagrama de secuencia: Eliminar Ubicación geográfica	249
Figura 7.55. Diagrama de secuencia: Ingresar forma de pago.....	249
Figura 7.56. Diagrama de secuencia: Consultar forma de pago	250
Figura 7.57. Diagrama de secuencia: Modificar forma de pago	250
Figura 7.58. Diagrama de secuencia: Eliminar forma de pago.....	251
Figura 7.59. Diagrama de secuencia: Ingresar impuestos	251
Figura 7.60. Diagrama de secuencia: Consultar impuestos	252
Figura 7.61. Diagrama de secuencia: Modificar impuestos.....	252
Figura 7.62. Diagrama de secuencia: Eliminar impuestos	253
Figura 7.63. Diagrama de secuencia: Ingresar descuentos y recargos.....	253
Figura 7.64. Diagrama de secuencia: Consultar descuentos y recargos	254
Figura 7.65. Diagrama de secuencia: Modificar descuentos y recargos	254
Figura 7.66. Diagrama de secuencia: Eliminar descuentos y recargos.....	255
Figura 7.67. Diagrama de secuencia: Crear perfil.....	255
Figura 7.68. Diagrama de secuencia: Consultar perfil.....	256
Figura 7.69. Diagrama de secuencia: Modificar perfil	256
Figura 7.70. Diagrama de secuencia: Eliminar perfil	257
Figura 7.71. Diagrama de secuencia: Ingresar usuario.....	257
Figura 7.72. Diagrama de secuencia: Consultar usuario.....	258
Figura 7.73. Diagrama de secuencia: Modificar usuario	258

Figura 7.74. Diagrama de secuencia: Eliminar usuario	259
Figura 7.75. Diagrama de secuencia: Generar respaldo	259
Figura 7.76. Diagrama de secuencia: Ingresar línea de producto	260
Figura 7.77. Diagrama de secuencia: Consultar línea de producto	260
Figura 7.78. Diagrama de secuencia: Modificar línea de producto.....	261
Figura 7.79. Diagrama de secuencia: Eliminar línea de producto	261
Figura 7.80. Diagrama de secuencia: Ingresar grupo de producto.....	262
Figura 7.81. Diagrama de secuencia: Consultar línea de producto.....	262
Figura 7.82. Diagrama de secuencia: Modificar línea de producto.....	263
Figura 7.83. Diagrama de secuencia: Eliminar línea de producto	263
Figura 7.84. Diagrama de secuencia: Ingresar producto.....	264
Figura 7.85. Diagrama de secuencia: Consultar producto	264
Figura 7.86. Diagrama de secuencia: Modificar producto	265
Figura 7.87. Diagrama de secuencia: Eliminar producto	265
Figura 7.88. Diagrama de secuencia: Ingresar receta.....	266
Figura 7.89. Diagrama de secuencia: Consultar receta	266
Figura 7.90. Diagrama de secuencia: Eliminar receta.....	267
Figura 7.91. Diagrama de secuencia: Ingresar inventario inicial	267
Figura 7.92. Diagrama de secuencia: Consultar inventario.....	268
Figura 7.93. Diagrama de secuencia: Modificar inventario.....	268
Figura 7.94. Diagrama de secuencia: Imprimir inventario	269
Figura 7.95. Diagrama de secuencia: Ingresar toma física	269
Figura 7.96. Diagrama de secuencia: Modificar toma física	270
Figura 7.97. Diagrama de secuencia: Imprimir toma física	270
Figura 7.98. Diagrama de secuencia: Ingresar movimiento de bodega	271
Figura 7.99. Diagrama de secuencia: Eliminar movimiento de bodega.....	271

Figura 7.100. Diagrama de secuencia: Imprimir movimiento de bodega.....	272
Figura 7.101. Diagrama de secuencia: Ingresar preproducción.....	272
Figura 7.102. Diagrama de secuencia: Consultar preproducción.....	273
Figura 7.103. Diagrama de secuencia: Imprimir preproducción	273
Figura 7.104. Diagrama de secuencia: Ingresar postproducción	274
Figura 7.105. Diagrama de secuencia: Consultar postproducción	274
Figura 7.106. Diagrama de secuencia: Imprimir postproducción.....	275
Figura 7.107. Diagrama de secuencia: Ingresar cliente	275
Figura 7.108. Diagrama de secuencia: Consultar cliente	276
Figura 7.109. Diagrama de secuencia: Modificar cliente	276
Figura 7.110. Diagrama de secuencia: Eliminar cliente	277
Figura 7.111. Diagrama de secuencia: Ingresar datos de factura	277
Figura 7.112. Diagrama de secuencia: Anular ítems de detalle de factura	278
Figura 7.113. Diagrama de secuencia: Elegir forma de pago	278
Figura 7.114. Diagrama de secuencia: Imprimir factura.....	279
Figura 7.115. Diagrama de secuencia: Guardar cuenta en espera	279
Figura 7.116. Diagrama de secuencia: Ingresar descuentos y recargos.....	280
Figura 7.117. Diagrama de secuencia: Consultar cuentas.....	280
Figura 7.118. Diagrama de secuencia: Imprimir pre-cuenta.....	281
Figura 7.119. Diagrama de secuencia: Anular factura	281
Figura 7.120. Diagrama de secuencia: Ingresar movimiento de caja.....	282
Figura 7.121. Diagrama de secuencia: Cerrar caja	282
Figura 7.122. Diagrama de secuencia: Generar reporte	283
Figura 7.123. Diagrama de secuencia: Imprimir reporte	283
Figura 7.124. Objeto: Administración	312
Figura 7.125. Objeto: Inventario	313

Figura 7.126. Objeto: Facturación	313
Figura 7.127. Módulo Administración – Estructura Organizacional	314
Figura 7.128. Módulo Administración – Ubicación Geográfica	315
Figura 7.129. Módulo Administración – Usuario y perfil	316
Figura 7.130. Módulo Administración – IVA, Recargos y Descuentos	317
Figura 7.131. Módulo Inventario - Bodega	317
Figura 7.132. Módulo Inventario – Producto	318
Figura 7.133. Módulo Inventario - Receta	319
Figura 7.134. Módulo Inventario - Producción.....	320
Figura 7.135. Módulo Inventario - Inventario	321
Figura 7.136. Módulo Inventario - Kardex	322
Figura 7.137. Módulo Facturación – Cliente.....	323
Figura 7.138. Módulo Facturación – Factura.....	324
Figura 7.139. Módulo Facturación – Cuadre de Caja.....	325
Figura 7.140. Módulo Facturación – Movimiento de Caja	326
Figura 7.141. Arquitectura del sistema.....	326
Figura 7.142. Diagrama de componentes	327
Figura 7.143 .Código del método seleccionarFormaPago	330
Figura 7.144. Diagrama Caja Blanca	331
Figura 7.145. Interfaz Movimientos de caja.....	333
Figura 8.1. Interfaz: Conexión a Base de Datos.....	344
Figura 8.2. Interfaz: Menú Principal de Oracle	344
Figura 8.3. Interfaz: Menú Administración de Oracle	344
Figura 8.4. Interfaz: Gestionar Usuario de Base de Datos	345
Figura 8.5. Interfaz: Crear Usuario de Base de Datos.....	345
Figura 8.6. Interfaz: Instalación GlassFish v3.....	346

Figura 8.7. Interfaz: Configuración de la administración GlassFish v3.....	346
Figura 8.8. Interfaz: Listo para instalar GlassFish v3	347
Figura 8.9. Interfaz: Resumen de instalación GlassFish v3	347
Figura 8.10. Interfaz: Consola de Administración de Glassfish V3.....	348
Figura 8.11. Interfaz: Nuevo JDBC en Pool de Conexiones de Glassfish V3..	349
Figura 8.12. Interfaz: Pool de Conexiones de Glassfish V3	349
Figura 8.13. Interfaz: Editar Pool de Conexiones de Glassfish V3.....	350
Figura 8.14. Interfaz: Recursos de nuevo JDBC	350
Figura 8.15. Interfaz: Recursos de JDBC	351
Figura 8.16. Interfaz: Aplicaciones	351
Figura 8.17. Interfaz: Deployar aplicaciones o módulos.....	352
Figura 8.18. Interfaz: Aplicaciones	352
Figura 9.1. Interfaz: Ingreso al Sistema.....	355
Figura 9.2. Interfaz: Menú Principal.....	355
Figura 9.3. Interfaz: Menú de Administración.....	356
Figura 9.4. Interfaz: Empresa	357
Figura 9.5. Interfaz: Sucursal	357
Figura 9.6. Interfaz: Ingreso de Sucursal	358
Figura 9.7. Interfaz: Modificar Sucursal.....	358
Figura 9.8. Interfaz: Menú Datos Generales	359
Figura 9.9. Interfaz: País	359
Figura 9.10. Interfaz: Ingreso de País	360
Figura 9.11. Interfaz: Modificar País.....	360
Figura 9.12. Interfaz: Región.....	361
Figura 9.13. Interfaz: Ingreso de Región	361
Figura 9.14. Interfaz: Modificar Región	361

Figura 9.15. Interfaz: Provincia.....	362
Figura 9.16. Interfaz: Ingreso de Provincia.....	362
Figura 9.17. Interfaz: Modificar Provincia	363
Figura 9.18. Interfaz: Ciudad.....	363
Figura 9.19. Interfaz: Ingreso de Ciudad	364
Figura 9.20. Interfaz: Modificar Ciudad	364
Figura 9.21. Interfaz: Zona	365
Figura 9.22. Interfaz: Ingreso a Zona	365
Figura 9.23. Interfaz: Modificar Zona.....	365
Figura 9.24. Interfaz: Forma de Pago.....	366
Figura 9.25. Interfaz: Ingreso de Forma de Pago.....	366
Figura 9.26. Interfaz: Modificar Forma de Pago	367
Figura 9.27. Interfaz: Impuestos.....	367
Figura 9.28. Interfaz: Ingreso de Impuestos.....	368
Figura 9.29. Interfaz: Modificar de Impuestos	368
Figura 9.30. Interfaz: Descuentos y Recargos	369
Figura 9.31. Interfaz: Ingreso de Descuentos y Recargos	369
Figura 9.32. Interfaz: Modificar Descuentos y Recargos.....	369
Figura 9.33. Interfaz: Seguridad.....	370
Figura 9.34. Interfaz: Perfil	370
Figura 9.35. Interfaz: Ingreso de Perfil	371
Figura 9.36. Interfaz: Permisos de Acceso al Sistema.....	371
Figura 9.37. Interfaz: Modificar perfil	372
Figura 9.38. Interfaz: Usuario	372
Figura 9.39. Interfaz: Ingreso de Usuario	373
Figura 9.40. Interfaz: Procesos	373

Figura 9.41. Interfaz: Respaldo de Datos	374
Figura 9.42. Interfaz: Menú de Inventario.....	374
Figura 9.43. Interfaz: Menú de Inventario.....	375
Figura 9.44. Interfaz: Línea de Productos	375
Figura 9.45. Interfaz: Ingreso de Línea de Productos	376
Figura 9.46. Interfaz: Grupo de Productos	376
Figura 9.47. Interfaz: Ingreso de Grupo de Productos	377
Figura 9.48. Interfaz: Productos	378
Figura 9.49. Interfaz: Ingreso de Productos	378
Figura 9.50. Interfaz: Recetas	379
Figura 9.51. Interfaz: Ingredientes.....	380
Figura 9.52. Interfaz: Procesos Inventario.....	380
Figura 9.53. Interfaz: Inventario Inicial	381
Figura 9.54. Interfaz: Toma Física.....	382
Figura 9.55. Interfaz: Preproducción	383
Figura 9.56. Interfaz: Seleccionar Recetas	384
Figura 9.57. Interfaz: Preproducciones	384
Figura 9.58. Interfaz: Postproducción.....	385
Figura 9.59. Interfaz: Postproducciones.....	385
Figura 9.60. Interfaz: Movimiento de Bodega.....	386
Figura 9.61. Interfaz: Movimientos de Bodega.....	387
Figura 9.62. Interfaz: Menú de Facturación.....	387
Figura 9.63. Interfaz: Menú de Facturación.....	388
Figura 9.64. Interfaz: Cliente	388
Figura 9.65. Interfaz: Ingreso Cliente	389
Figura 9.66. Interfaz: Procesos de Facturación.....	389

Figura 9.67. Interfaz: Factura	390
Figura 9.68. Interfaz: Clientes	391
Figura 9.69. Interfaz: Consulta Productos	391
Figura 9.70. Interfaz: Cuentas pendientes	391
Figura 9.71. Interfaz: Formas de pago	392
Figura 9.72. Interfaz: Seleccionar Descuentos y recargos	393
Figura 9.73. Interfaz: Anular Facturas	393
Figura 9.74. Interfaz: Movimientos de Caja	394
Figura 9.75. Interfaz: Cuadre de Caja	394
Figura 9.76. Interfaz: Reporte.....	395

RESUMEN

El análisis, implementación y documentación del sistema informático para administración y facturación de un restaurante se realizaron a través del Proceso Unificado Racional y dado que es una aplicación web también se utilizó el Método de Diseño Hipermedia Objeto Orientado.

La plataforma de programación que se empleó para desarrollar y ejecutar el sistema de restaurantes es Java Enterprise Edition, utilizando una arquitectura Modelo Vista Controlador, lo que nos permitió distribuir el trabajo de creación de la aplicación en tres niveles: en la capa modelo para el manejo y control de datos se utilizó SessionBeans y EntityBeans; en la capa vista para la presentación visual del sistema se manejó la tecnología JavaServerFaces con la librería de componentes visuales RichFaces para facilitar la integración de AJAX en la aplicación, y para utilizar plantillas se utilizó el framework Facelets; finalmente en la capa controlador para procesar las órdenes del usuario se utilizó métodos de los Session Beans.

CAPITULO 1. ANTECEDENTES

1.1. INTRODUCCIÓN

El rápido crecimiento de la tecnología y la globalización han permitido que la velocidad con la que se realiza los negocios sea más rápida que antes, la información de las empresas es su mayor bien y la ventaja que tiene sobre la competencia. La administración de la información ha permitido organizarla de mejor manera, y el desarrollo de programas informáticos constituye una de las fuentes que proveen de herramientas para poder administrar esta información.

En el mercado existen una gran cantidad de paquetes informáticos con una variedad de programas sobre contabilidad, facturación, cartera, recursos humanos, entre otros. Debido a que todos los negocios no son iguales, las empresas ven la necesidad de tener un software que se acople a sus necesidades, originando el desarrollo de un paquete informático propio, o adquiriendo uno ya desarrollado para realizar ciertas modificaciones.

En la actualidad se utilizan diversas metodologías para ayudar a una correcta elaboración de un software, una muestra de ello es la metodología RUP que organiza todas las actividades de una empresa en procesos, facilitando el control de cada una de esas actividades y permitiendo que el estudio primero se fije en el negocio de la empresa. La ventaja de centrarse primero en los procesos del negocio de la empresa es identificar todos los posibles problemas y sus causas, permitiendo encontrar una solución y desarrollar un programa informático para mitigar los problemas.

Las empresas buscan centralizar su información para procesarla de la manera más eficiente y en el menor tiempo posible, con la llegada del internet se logró interconectar lugares muy distantes y originó la programación web para entornos distribuidos empresariales. Con la programación distribuida para la web se creó la metodología OOHDM, la cual proporciona una guía para el diseño del programa y los modelos orientados a objetos.

La metodología RUP y OOHDM proporciona procesos iterativos e incrementales, donde cada iteración se organiza en fases; al utilizar este tipo de metodologías en el desarrollo del programa nos permitirá tener una mayor comprensión de los requerimientos del mismo. Al ser OOHDM una metodología orientada a objetos se utilizará el lenguaje de modelado unificado UML para realizar los diferentes diagramas de los componentes del proyecto.

1.2. OBJETIVO GENERAL

Desarrollar un sistema que administre la producción, los inventarios y la facturación de un restaurante.

1.3. OBJETIVOS ESPECÍFICOS

- Analizar y desarrollar una herramienta que permita el manejo del proceso de producción, facturación y control de inventarios en un restaurante para optimizar sus recursos.
- Utilizar herramientas open source tales como el sistema operativo Linux, el lenguaje de programación Java para la construcción del sistema y el servidor de aplicaciones Glassfish.
- Centralizar la información generada por la aplicación utilizada en un restaurante con varias sucursales para obtener los reportes gerenciales.

1.4. DEFINICIÓN DEL PROBLEMA

Cada negocio tiene sus propias características, las mismas que deben ser controladas para su correcto funcionamiento, en las cadenas de restaurantes los procesos de facturación y control de producción ocasionan muchos problemas al tener varias sucursales, y al no disponer de información centralizada para conocer el estado del negocio produce una demora en el despacho y producción de los alimentos, además de inconvenientes con los clientes.

En la mayoría de los restaurantes el cuadro del dinero de la caja de cada sucursal origina que la información se demore en llegar a la matriz y que un administrador tenga que ir por cada local obteniendo el reporte financiero y de productos de la caja registradora o del sistema de facturación.

La necesidad de reportes gerenciales para la administración, con la consolidación de la información de la matriz y de las sucursales para la toma de decisiones es fundamental para tener una ventaja competitiva sobre el resto de empresas.

Las crecientes exigencias del mercado, y la necesidad de mejorar la administración de la empresa originaron la necesidad de centralizar la

información con la ayuda de la tecnología y el desarrollo de un programa informático.

1.5. DESCRIPCIÓN DEL PROYECTO

La aplicación dispondrá de una interfaz gráfica amigable para el usuario, el módulo de facturación permitirá el manejo ágil de cuentas en espera, devoluciones, anulaciones, consultas de los inventarios, diferentes formas de pago e impresiones de facturas y pre cuentas.

Para el módulo de inventarios se podrá crear los productos y catalogarlos para una mejor organización. Cada producto manejará tres tipos diferentes de precios y podrá formar parte de una receta o ser de tipo de servicio. Se podrá realizar la pre-producción y post-producción de las recetas y manejar el inventario de cada producto.

En el módulo de administración se podrá crear la empresa y las respectivas sucursales, se manejará la seguridad del sistema con la creación de usuarios y perfiles de acceso.

El sistema facilitará la obtención de informes gerenciales de los reportes obtenidos del cuadro de caja y de inventarios de la aplicación.

1.6. JUSTIFICACIÓN DEL PROYECTO

El presente trabajo ayudará a centralizar la información de un restaurante permitiendo obtener reportes gerenciales de cada una de las sucursales y lograr la optimización del recurso humano.

Se manejará la metodología de desarrollo de software RUP (Metodología Unificada de Procesos) para identificar los procesos que lleva un restaurante, de esta manera podremos obtener los requisitos necesarios para el desarrollo del sistema, la diagramación se realizará utilizando el lenguaje de modelado UML (Lenguaje Unificado de Modelado) que permite realizar los diagramas de los procesos obtenidos y facilitará la representación de objetos en el sistema; además se aplicará la metodología OOHDM (Metodología de Diseño

Hipermedial Orientada a Objetos) que posibilita desarrollar el plan de sistemas de información basados en la Web.

La aplicación se desarrollará bajo el sistema operativo Linux permitiendo abaratar costos ya que no se necesita adquirir licencias para su utilización; el lenguaje de programación será Java por sus características de multiplataforma y orientado a objetos, bajo la plataforma JEE(Plataforma Java, Edición Empresarial) porque es orientado a entornos distribuidos empresariales o de Internet ; el gestor de base de datos que se utilizará es Oracle debido al soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma que ofrece.

Se aplicará una arquitectura en tres capas o Modelo Vista Controlador porque este patrón es utilizado en aplicaciones web debido a que separa la interfaz de usuario, el sistema de gestión de datos y la lógica de control en tres componentes, de esta manera se facilitará las modificaciones o adaptaciones que el sistema requiera posteriormente y que pueda ser evolutivo y escalable.

Al ser una aplicación distribuida se ejecutará desde la web, facilitando la administración desde cualquier punto que tenga acceso a Internet, con esto se logra agilizar el proceso de facturación, inventarios y producción beneficiando la atención al cliente al tener datos exactos del producto o productos con los que cuenta la empresa.

1.7. ALCANCE DEL PROYECTO

La aplicación para el restaurante está enfocada a la administración de facturación, inventarios y producción de la empresa, para esto se han determinado tres módulos:

Administración

- Configuración de la empresa
- Manejo de sucursales
- Respaldos y restauración de la información
- Seguridad de acceso al sistema
- Reportes gerenciales

Inventarios

- Productos por líneas, grupos
- Tipos de productos: normal, recetas y servicios
- Codificación automática, 3 tipos de precios
- Pre producción y post producción de recetas
- Consultas y reportes

Facturación

- Cuentas en espera (cuentas abiertas)
- Formas de pago: efectivo, cheques y tarjeta de crédito
- Cierre y cuadro de caja
- Impresión de facturas y pre cuentas
- Consultas y reportes

Cabe destacar que el sistema no abarcará los puntos de contabilidad, cartera y recursos humanos, debido a que los puntos antes mencionados salen fuera del alcance del presente trabajo

Los tres módulos formarán parte de una aplicación que será instalada en un servidor de aplicaciones para poder aprovechar los recursos de hardware del servidor y no saturar a las estaciones de trabajo cuando realicen algún proceso; la información se almacenará en una única base de datos y se generarán backups periódicamente para respaldar la información.

CAPITULO 2. FUNDAMENTACIÓN TEÓRICA

2.1. METODOLOGÍA RUP

2.1.1. HISTORIA

El Proceso de Desarrollo Unificado ve la luz después de un largo periodo de desarrollo y pruebas, posee varias influencias de diferentes fuentes. Entre una de las principales tenemos la de Ericsson.

En 1967 Ericsson modelaba los sistemas enteros en bloques, ensamblándolos en bloques de más bajo nivel, para luego unirlos en subsistemas de más alto nivel, identificaban los diferentes bloques estudiando los casos de uso del negocio, sus responsabilidades y detallando sus especificaciones. Modelar el sistema en bloques enteros resultaba una técnica para manejar de mejor manera el sistema completo. Este método lo conocemos en la actualidad como Desarrollo basado en componentes.

En 1987 Ivar Jacobson dejó el desarrollo de los sistemas en bloques y luego de renunciar en Ericsson fundó Objectory AB en Estocolmo. Durante ocho años desarrolló el proceso Objectory ("Object Factory", Fábrica de Objetos) que se basa en una secuencia de actividades, las cuales representan una serie de modelos como son:

- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

Estos modelos son una perspectiva del sistema, los desarrolladores podían realizar el seguimiento de un modelo a otro mediante una traza desde el modelo hasta el código fuente o viceversa cuando ocurrían problemas.

A finales de 1995 Rational Software Corporation compró Objectory AB y comenzó el proceso para fusionar los procesos de desarrollo de software de las

dos empresas. Objectory 3.8 en el momento de la fusión estaba correctamente desarrollado en aspectos como el modelado de casos de uso, análisis y diseño pero en la gestión de requisitos, implementación y pruebas tenía falencias, por ello se añadió los conocimientos de Rational y se formó el Proceso Objectory de Rational 4.1. En el mismo año el Lenguaje de Modelado Unificado (UML) estaba en desarrollo y fue incluido como lenguaje de modelado al ROP (Proceso Objectory de Rational)

Luego que Rational Software Corporation se fusionara con muchas empresas y que cada una de ellas aportara con sus experiencias en el área de desarrollo de software y procesos en junio de 1998 Rational publicó una nueva versión de su producto el RUP (Proceso Unificado de Rational) en su versión 5.0.

2.1.2. INTRODUCCIÓN

En la actualidad debido a la globalización de los negocios se ve la necesidad de construir software cada vez más grandes y complejos, además el tiempo de salida al mercado es otro factor importante en el momento de construir un sistema informático, el problema surge cuando los desarrolladores tienen que coordinar diferentes flujos de trabajo, entonces surge la necesidad de un proceso que integre y coordine todo el proceso de desarrollo de software.

Se necesitaba un proceso que:

- “Proporcione una guía para ordenar las actividades de un equipo.
- Dirija las tareas de desarrollo por separado y del equipo como un todo.
- Especifique los artefactos que deben desarrollarse.
- Ofrezca criterios para el control y la medición de los productos y actividades del proyecto”¹.

Los procesos bien definidos y gestionados hacen la diferencia entre los proyectos exitosos y aquellos que fracasan. El RUP es más que un proceso de desarrollo de software, es un marco de trabajo que se puede utilizar en la

¹ El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh, Capítulo 1, Pág. 4

mayoría de sistemas informáticos para diferentes tipos de organizaciones y de aplicaciones.

2.1.3. CARACTERÍSTICAS

Dirigido por casos de uso

Un caso de uso es un requerimiento funcional que da al usuario una funcionalidad importante, entendiendo por usuario a alguien o algo que interactúa con el sistema. Los casos de uso al agruparse forman el modelo de casos de uso el cual describe la funcionalidad del sistema.

Los casos de uso además especifican los requerimientos del sistema, guían su diseño, implementación y pruebas, es decir están durante todo el proceso de desarrollo del sistema informático dando un hilo conductor e influyen en la arquitectura del sistema.

Centrado en la arquitectura

En la arquitectura se refleja los casos de uso y es donde funcionará el software, esto implica hardware, sistema operativo, bases de datos, protocolos de comunicación en la red, entre otros.

La arquitectura debe diseñarse de tal manera que el sistema pueda evolucionar a través de las futuras versiones, para que esto suceda los arquitectos de software deben trabajar primero sobre los casos de uso claves y que constituyen las funciones más importantes del sistema.

Los casos de uso deben encajar en la arquitectura y permitir que maduren a medida que la arquitectura también lo hace, todo este proceso se repite hasta que se obtiene una arquitectura estable.

Iterativo e incremental

El desarrollo de software a gran nivel supone un mayor esfuerzo, para esto es conveniente dividir todo el desarrollo en mini-proyectos, para que sea más manejable.

Cada mini-proyecto corresponde a una iteración la cual termina en un incremento del proyecto, cada iteración hace referencia a la secuencia de actividades y los incrementos al crecimiento del sistema.

Para la selección de lo que se desarrollará se debe tener en cuenta los casos de uso y una lista de los posibles riesgos durante la iteración, con la finalidad de disminuir posibles eventos negativos sobre el proyecto.

Al terminar cada iteración, si esta cumple con los objetivos planteados se continua con la siguiente iteración caso contrario se debe hacer un estudio de porque se obtuvo problemas y cambiar el enfoque en el desarrollo.

Al ser mini-proyectos los costos de riesgo de cada iteración se reducen comparado con el riesgo si se realizara todo el sistema, además ayuda a controlar los tiempos para sacar al mercado un producto, controlando el ritmo de trabajo en cada iteración.

“Estos conceptos – los de desarrollo dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental – son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las tres tareas reduciría drásticamente el valor del Proceso Unificado. Es como un taburete de tres patas, sin una de ellas el taburete se cae”².

2.1.4. CICLO DE VIDA DEL PROCESO UNIFICADO

El desarrollo del Proceso Unificado de Rational se repite con una serie de ciclos que constituyen la vida del sistema informático, donde cada ciclo concluye con una versión del sistema. Cada ciclo de vida del RUP consta de cuatro fases, cada fase termina con un hito, el cual ayuda a tomar la decisión si se prosigue o no con la siguiente fase.

² El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh, Capítulo 1, Pág. 8

2.1.4.1. INICIO

Se realiza una descripción del sistema en su totalidad y se presenta el análisis del negocio. En esta fase se responde a las preguntas:

- “¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?”
- ¿Cómo podría ser la arquitectura del sistema?
- ¿Cuál es el plan del proyecto y cuánto costará desarrollar el producto?”³

2.1.4.2. ELABORACIÓN

Se elaboran los casos de uso más importantes que se identificaron en la fase de inicio del sistema y se diseña su arquitectura. Al finalizar esta fase se puede estimar los recursos necesarios y planificar las actividades.

La pregunta fundamental en esta fase es:

“¿Son suficientemente estables los casos de uso, la arquitectura y el plan, y están los riesgos lo suficientemente controlados como para que seamos capaces de comprometernos al desarrollo entero mediante un contrato?”⁴

2.1.4.3. CONSTRUCCIÓN

Se desarrolla el producto y crece hasta convertirse en un sistema informático completo. Al final de esta fase el sistema ha implementado todos los casos de uso que los desarrolladores y los clientes han quedado de acuerdo para la versión terminada del sistema.

La pregunta clave en esta fase es:

“¿Cubre el producto las necesidades de algunos usuarios de manera suficiente como para hacer una primera entrega?”⁵

³El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh, Capítulo 1, Pág. 11

⁴ El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh, Capítulo 1, Pág. 12

2.1.4.4. TRANSICIÓN

Comprende el tiempo en el cual el sistema se convierte en una versión beta, es decir, los usuarios prueban el sistema para encontrar errores o defectos para que luego los desarrolladores los corrijan. En esta fase se realiza la capacitación al cliente y se le brinda la asistencia técnica y operacional. Se suele dividir los errores en dos grupos:

- “Los que tienen suficiente impacto en la operación para justificar una nueva versión incrementada.
- Los que pueden corregirse en la siguiente versión normal”⁶.

Cada fase se divide en iteraciones y en cada iteración se tiene una secuencia de actividades relacionadas o flujos de trabajo.

2.1.5. FLUJOS DE TRABAJO

2.1.5.1. REQUISITOS

Los desarrolladores no serán las personas que utilicen el sistema, por ello los requisitos y las restricciones deben estar de forma clara. Como se deduce los requisitos proporcionan una guía en el desarrollo del sistema. Los usuarios en la mayoría de los casos no saben cómo hacer más eficiente sus procesos y tampoco que parte de sus procesos se pueden transformar en un sistema, los analistas de sistemas son los encargados de registrar los requisitos y encontrar la misión por la cual se construirá el sistema.

Para obtener los requisitos del sistema se debe seguir una serie de actividades o flujos de trabajo, el cual incluye los siguientes pasos:

⁵ El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh, Capítulo 1, Pág. 12

⁶ El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh, Capítulo 1, Pág. 12

Enumerar los requisitos candidatos.- son todas aquellas ideas que surgen durante todo el desarrollo del sistema y pueden considerarse como ideas candidatas a implementarse.

Comprender el contexto del sistema.- para construir un sistema correcto los arquitectos y analistas del sistema requieren un amplio conocimiento del negocio donde se va a implementar el sistema.

Capturar requisitos funcionales.- gran parte de los requisitos funcionales se obtienen al desarrollar los casos de uso, si se logra comprender todos los casos de uso del sistema, entonces el analista sabrá lo que debe hacer el sistema; sin embargo se debe tomar en cuenta que algunos requisitos funcionales se definen antes del diseño de los casos de uso.

Capturar requisitos no funcionales.- especifican características del sistema como la fiabilidad, portabilidad, rendimiento, implementación, mantenimiento y dependencias de la plataforma.

2.1.5.2. ANÁLISIS

Se refina los requisitos que se obtuvieron y se los estructura para tener una mejor comprensión del sistema, el objetivo principal del análisis es resolver los problemas obtenidos en los requisitos con mayor profundidad pero utilizando un lenguaje más formal o lenguaje de los desarrolladores.

El análisis nos presenta una vista interna del sistema y está estructurado por clases y paquetes que son una entrada fundamental de información para la totalidad del sistema.

Las primeras iteraciones en la fase de elaboración se centran en el análisis, estas iteraciones nos ayudan a tener una arquitectura más estable y sólida.

2.1.5.3. DISEÑO

Partiendo del modelado de análisis podemos modelar o diseñar todos los requisitos funcionales y no funcionales incluyendo la arquitectura del sistema.

El propósito del diseño de un sistema es:

- Comprender los requisitos funcionales y no funcionales
- Crear un conjunto de actividades para realizar la implementación del sistema
- Descomponer los trabajos en partes más pequeñas y manejables para luego ser asignadas a los diferentes equipos de trabajo

El papel que cumple el diseño en el desarrollo de un software es contribuir a una arquitectura más sólida y estable, y obtener un plano del modelado de implementación. El diseño es el final de la fase de elaboración y el comienzo de las iteraciones en la fase de construcción.

2.1.5.4. IMPLEMENTACIÓN

Partimos del modelado final del diseño y lo implementamos en términos de componentes, como script de base de datos, código fuente, etc. El modelado de implementación organiza los componentes y los estructura de acuerdo al lenguaje o lenguajes utilizados y de la relación que tiene con otros componentes.

El propósito de la implementación es:

- Organizar las integraciones de los componentes en cada iteración.
- Implementar la arquitectura planteada en el diseño.
- Comprobar la funcionalidad individual de cada componente y luego integrarlos en un solo sistema.

Las iteraciones durante la implementación son el centro de la fase de construcción del sistema y juega un papel importante en el desarrollo del ciclo de vida del sistema.

2.1.5.5. PRUEBAS

Cuando se termina una construcción del sistema, esta es sometida a pruebas de integración y de sistema, las versiones finales son entregadas a terceros, es

decir, a personas que no desarrollaron el sistema para que realicen las respectivas pruebas.

Los objetivos de las pruebas del sistema son:

- Planificar pruebas del sistema al final de cada iteración del proyecto de software, estas pruebas pueden ser de software y pruebas con el usuario.
- Diseñar los casos de prueba a partir de los casos de uso e implementarlos creando procedimientos para cada caso de prueba.
- Los problemas detectados deben convertirse en nuevos flujos de trabajo para que puedan ser corregidos.

2.2. METODOLOGÍA OOHDM

Debido al crecimiento de programas aplicados a la web, se creó una metodología que facilite su desarrollo, OOHDM (Método de Diseño Hipermedia Orientado a Objetos) es una metodología para el desarrollo de programas hipermedia y para aplicaciones web.

Los programas hipermedia hacen referencia a la creación de hiperdocumentos que faciliten la búsqueda de la información a los usuarios a través de los diferentes enlaces establecidos entre los elementos de información multimedia, como son texto, imagen, sonido, etc. que forman parte del documento hipermedia.

Es fundamental comprender el modelo del negocio o dominio de la aplicación que regirá durante todo el desarrollo de la programación hipermedia. OOHDM es la unión de varios tipos de desarrollo, como el basado en prototipos, el iterativo y el incremental.

Todo el proceso de OOHDM está basado en cuatro fases o etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas y la implementación.

2.2.1. DISEÑO CONCEPTUAL

En términos de la orientación a objetos, el diseño conceptual tendrá aspectos del modelo de Análisis y de Diseño del ciclo de vida del software, representa los diferentes objetos con sus relaciones y colaboraciones, es decir, el diseño conceptual está construido a partir de clases, relaciones entre clases y subsistemas.

En el diseño navegacional las clases se utilizarán para definir nodos y las relaciones para definir los enlaces.

2.2.2. DISEÑO NAVEGACIONAL

La navegación es considerada un paso crítico en el diseño de páginas web, debido a que el mayor énfasis se ha puesto en el diseño de la interfaz de usuario. En OOHDM el modelado de diseño se lo realiza como una vista y permite realizar modelos específicos para cada usuario.

El diseño navegacional se expresa en dos esquemas:

- **Esquema de contexto navegacional.-** se define como navegará el usuario a través de las pantallas o hiperdocumentos.
- **Esquema de clases navegacionales.-** es una combinación entre el modelo conceptual y el esquema de contexto, donde las clases navegacionales se denominan nodos, las relaciones navegacionales se llaman vínculos y los atributos de los nodos que activan navegaciones son llamados anclas.

Además existen un conjunto de tipos predefinidos como clases de navegación:

- **Nodos.-** contienen o agrupan la información de una actividad específica dentro del hiperdocumento, se construyen como puntos de vista de los objetos conceptuales, se llaman objetos de navegación.
- **Enlaces.-** es un puente de conexión entre los nodos, además la unión de enlaces en una secuencia establecida sirve para dar una información coherente al usuario.

- **Estructuras de acceso.-** se definen como diccionarios o índices que permiten acceder a la información de manera rápida según lo necesite el usuario, un ejemplo de estas estructuras pueden ser los menús. “Las estructuras de acceso también se modelan como clases, compuestas por un conjunto de referencias a objetos que son accesibles desde ella y una serie de criterios de clasificación de las mismas.”⁷

2.2.3. DISEÑO DE INTERFACES ABSTRACTAS.

En esta fase se establecen los objetos que aparecerán en la pantalla del usuario, la navegación, que transformaciones de la interfaz son necesarias y cuando es necesario que aparezcan.

La separación de la navegación y de las interfaces abstractas, permite construir interfaces diferentes y tener un alto grado de independencia de la tecnología utilizada en la interfaz.

Se utilizan vistas de datos abstractas (ADV) para realizar las pantallas por cada usuario de nuestra aplicación web. Los objetos de navegación como nodos, e índices actuarán como objetos de datos abstractos, y su ADV asociados se usará para especificar su apariencia al usuario.

Se utilizan los diagramas de configuración para ver todos los posibles eventos que puede o debe hacer un usuario en cada pantalla y los diagramas de estados para cada usuario, con esto logramos ver mediante cada evento que es lo que el usuario hace y a que estado cambia el programa.

2.2.4. IMPLEMENTACIÓN

Se debe implementar todo el diseño que se desarrolló en forma separada y tener en cuenta el entorno en donde se ejecutará la aplicación.

⁷ WIKIPEDIA La Enciclopedia libre. OOHDM. [En línea]. <http://es.wikipedia.org/wiki/OOHDM>. [Citado el 20 de marzo del 2010].

Se debe definir y organizar los elementos que forman parte del dominio con el perfil de cada usuario y la tarea que se le permitirá realizar. Se debe decidir cómo se comportará la interfaz y donde se guardará la información.

La implementación debe hacerse con lenguajes de alta difusión, que operen en varias plataformas y permitan realizar programación distribuida.

2.3. LENGUAJE UNIFICADO DE MODELAMIENTO - UML

El lenguaje unificado de modelado (UML - Unified Modeling Language) es un lenguaje para el modelamiento de sistemas y no es un método. “La mayor parte de métodos consisten, al menos en principio, en un lenguaje y un proceso para modelar. El lenguaje de modelado es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño”⁸

El UML unifica los trabajos diseñados por Booch, Rumbaugh y Jacobson, incluso es soporte para las principales metodologías para el desarrollo de software como son el Proceso Unificado de Rational (RUP), Método de Diseño Hipermedia Orientado a Objetos (OOHDM), etc. En la actualidad está respaldado por el Grupo de Administración de Objetos (OMG - Object Management Group).

En la actualidad UML está en la versión 2.0 y cuenta con 13 diagramas los cuales se dividen jerárquicamente.

2.3.1. JERARQUÍA DE LOS DIAGRAMAS UML

2.3.1.1. DIAGRAMAS DE ESTRUCTURA

Los diagramas de estructura enfatizan en los elementos que deben existir en el sistema.

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos

⁸ UML Gota a Gota - Martin Fowler, Kendall Scott, Capítulo 1, Pág. 1

- Diagrama de estructura compuesta
- Diagrama de despliegue
- Diagrama de paquetes

2.3.1.2. DIAGRAMAS DE COMPORTAMIENTO

Los diagramas de comportamiento enfatizan en lo que debe suceder en el sistema modelado.

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

2.3.1.3. DIAGRAMAS DE INTERACCIÓN

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración.
- Diagrama de tiempos.
- Diagrama global de interacciones o Diagrama de vista de interacción.

2.4. PATRONES DE DISEÑO

Los patrones de diseño son soluciones probadas para problemas comunes en el desarrollo de software, se los utiliza como buenas prácticas o procedimientos durante el desarrollo de una aplicación y son completamente independientes del lenguaje de programación.

El arquitecto Christopher Alexander en el año de 1979 publicó un libro sobre patrones de diseño para la construcción de edificios, de esta manera él intentaba proporcionar una guía para solucionar los problemas más comunes durante la construcción de edificios, partiendo del mismo concepto en el año de 1994

aparece el libro “Design Patterns: Elements of Reusable Object Oriented Software” que fue escrito por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides , los cuales fueron llamados Gang of Four o la pandilla de los cuatro, a partir de la publicación del libro los patrones de diseño en la programación orientada a objetos comenzó a tomar fuerza y empezaron a ser factibles las implementaciones de soluciones a los problemas, a los que se enfrentan los ingenieros de software durante la construcción de un sistema.

2.4.1. CLASIFICACIÓN DE LOS PATRONES DE DISEÑO

A los patrones de diseño se los puede clasificar siguiendo dos criterios como se visualiza en la Tabla 2.1. “Patrones de diseño”. “El primero de ellos, denominado propósito, refleja que hace un patrón. Los patrones pueden tener un propósito de creación, estructural o de comportamiento. Los patrones de creación tienen que ver con el proceso de creación de objetos. Los patrones estructurales tratan con la composición de clases u objetos. Los de comportamiento caracterizan el modo en que las clases y los objetos interactúan y se reparten la responsabilidad.

El segundo criterio, denominado ámbito, especifica si el patrón se aplica principalmente a clases y subclasses. Estas relaciones se establecen a través de la herencia, de modo que son relaciones estáticas – fijadas en tiempo de compilación-. Los patrones de objetos tratan con las relaciones entre los objetos, que pueden cambiarse en tiempo de ejecución y son dinámicas. Casi todos los patrones usan herencia de un modo u otro, así los únicos patrones etiquetados como patrones de clase son aquellos que se centran en las relaciones de clases. Nótese que la mayoría de los patrones tiene un ámbito de objeto.

Los patrones de creación de clases delegan alguna parte del proceso de creación de objetos en las subclasses, mientras que los patrones de creación de objetos lo hacen en otro objeto. Los patrones estructurales de clases usan la herencia para componer las clases, mientras que los objetos describen formas de ensamblar objetos. Los patrones de comportamiento de clases usan la herencia para describir algoritmos y flujos de control, mientras que los objetos

describen como cooperar un grupo de objetos para realizar una tarea que ningún objeto puede llevar a cabo por sí solo”.⁹

Propósito				
		De Creación	Estructurales	De Comportamiento
Ámbito	Clase	Factory Method	Adapter	Interpreter Template Method
	Objeto	Abstrac Factory Builder Prototype Singleton	Adapter Birdge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Satrategy Visitor

Tabla 2.1. “Patrones de Diseño”¹⁰

2.5. MODELO MVC

El modelo MVC es un patrón de diseño en la arquitectura de software que propone separar la programación en tres capas, primero la interfaz de usuario, luego la lógica de negocio junto con el sistema de gestión de base de datos y finalmente la gestión de los ingresos del usuario. El modelo MVC significa Modelo Vista Controlador donde cada uno corresponde a un objeto específico.

- **Modelo:** maneja la gestión de la base de datos y la lógica de negocios del programa.
- **Vista:** representa a la interfaz de usuario o a las pantallas para interactuar con el usuario.
- **Controlador:** gestiona los eventos del usuario y es la unión entre la vista y el modelo.

⁹ Patrones de Diseño - Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, Capitulo 1, Pag. 8,9

¹⁰ Patrones de Diseño - Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, Capitulo 1, Tabla 1.1

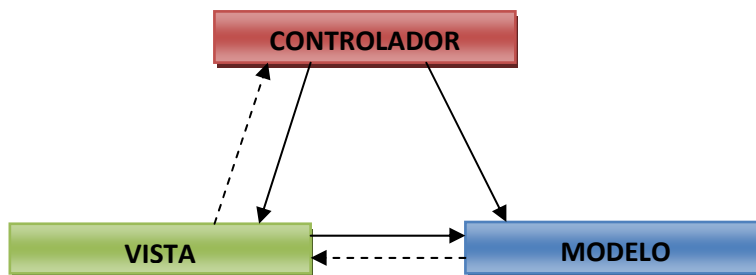


Figura 2.1. Diagrama MVC ¹¹

Al implementar el modelo MVC se logra una clara separación entre los componentes, se puede desarrollar e implementar cada componente por separado y permite un alto acoplamiento entre las vistas y los controladores.

2.6. LENGUAJE DE PROGRAMACIÓN JAVA

2.6.1. HISTORIA DE JAVA

En la década de los noventa la revolución de los microprocesadores hizo posible el desarrollo de la computadora personal y se pensó que tendrían un impacto similar en los dispositivos electrónicos para electrodomésticos, fue entonces cuando la empresa Sun Microsystems patrocinó el proyecto Green en el año 1991 para crear un lenguaje de programación que sea independiente del dispositivo donde se ejecute. El lenguaje fue desarrollado por James Gosling, se basó en el conocido C++ y fue llamado Oak debido a un roble que se veía desde la ventana de su oficina en Sun. Tiempo después se cambió el nombre a Java debido a una variedad de café que existía en una cafetería local que visitaba el equipo de desarrollo y porque ya existía un lenguaje de programación llamado Oak.

Cuando la World Wide Web despuntó en el año 1993, la gente de Sun Microsystems se dio cuenta del potencial de Java para añadir contenido dinámico y animaciones en páginas web, debido a esta característica Java llamó la atención en la comunidad de negocios y en el año 1995 fue presentada formalmente al mundo.

¹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.6.2. CARACTERÍSTICAS

Java es un lenguaje de programación orientado a objeto desde sus inicios, el cual posee las siguientes características:

- **Simple.-** es de fácil aprendizaje, para programar en Java se debe tener claro los conceptos sobre orientación a objetos.
- **Seguro.-** se impide el acceso ilegal a la memoria eliminando los punteros y casting implícito de otros lenguajes como C y C++, además posee un verificador de byte-codes que comprueba la coherencia de los fragmentos de código y no se puede tener variables globales ni acceder a recursos arbitrarios del sistema.
- **Portable.-** Java implementa estándares de portabilidad a nivel de desarrollo, sus tipos de datos básicos y los operadores aritméticos son iguales en todas las plataformas, además posee un sistema abstracto de ventanas para la interfaz gráfica de usuarios facilitando que el programa pueda ser implementado en entornos Windows, Mac y Linux, estas características son gracias a la JVM (Máquina Virtual de Java).
- **Orientado a objetos.-** Java fue desarrollado desde sus inicios con orientación a objetos, posee las tres principales características de la orientación a objetos como son herencia, encapsulamiento y polimorfismo. No posee herencia múltiple pero soluciona este inconveniente con interfaces.
- **Robusto.-** posee varios mecanismos de comprobación en compilación y en tiempo de ejecución, además posee un Garbage Colector que elimina la liberación explícita de la memoria, liberando de esta tarea al programador.
- **Multitareas.-** permite realizar varias tareas a la vez, permite un mejor rendimiento iterativo y en tiempo real, aunque lo segundo este limitado por las características del equipo y del sistema operativo donde se ejecute el programa.
- **Arquitectura neutral.-** java está diseñado para que un programa escrito en este lenguaje pueda ser ejecutado en cualquier plataforma (Windows, Mac o Linux) de aquí viene su característica escríbalo una vez y ejecútelo

donde sea (Write Once Run Anywhere). Para conseguir esto se utiliza una compilación intermedia mediante byte-codes los cuales son interpretados por la Máquina Virtual de Java (JVM)

- **Interpretado.-** los byte-codes generados pueden ser interpretados en cualquier plataforma siempre que esté instalado el intérprete (JVM) y el sistema de ejecución en tiempo real.
- **Distribuido.-** facilita la creación de programas distribuidos, al tener una gran colección de clases para la comunicación en una red de computadoras, permite manejar socket y establecer comunicaciones con servidores o clientes remotos.
- **Dinámico.-** java implementa todos los módulos que comprenden una aplicación solo en tiempo de ejecución, además implementa opciones de persistencia para que los objetos o las clases no se eliminen cuando exista una limpieza de memoria y no exista la necesidad de importarlos cada vez que se los necesite.

2.6.3. PLATAFORMAS

La plataforma de java no es un sistema operativo o hardware, es una máquina virtual que ofrece un conjunto de librerías para las aplicaciones.

Plataforma Java, Edición Estándar (Java SE)

Conocido hasta la versión 5.0 como Plataforma Java 2, Edición Estándar o J2SE, está relacionado con la creación de aplicaciones de escritorio y applets. Incluye una colección de APIs o librerías y es la base para la Edición Empresarial.

Plataforma Java, Edición Empresarial (Java EE)

Conocido como Plataforma Java 2, Edición Empresarial o J2EE hasta la versión 1.4, pensada para la creación de aplicaciones empresariales del lado del servidor, además permite el desarrollo de una arquitectura a n niveles distribuida.

Plataforma Java, Edición Micro (Java ME)

Conocido como Plataforma Java 2, Edición Micro o J2ME, posee una colección de APIs o librerías para el desarrollo de sistemas en dispositivos con recursos de hardware limitado como celulares, PDA, etc.

Plataforma Java, Edición Empresarial (JEE)

Es una plataforma para el desarrollo de aplicaciones empresariales, permite la creación de una arquitectura en n niveles y la programación de una aplicación distribuida para que se ejecute en el lado del servidor. Java EE posee varias librerías o APIs, entre las principales podemos mencionar:

- **JDBC.-** (Java Database Connectivity) es una librería que permite realizar operaciones sobre una base de datos desde el lenguaje de programación Java y utilizando el lenguaje de consulta estándar SQL.
- **RMI.-** (Java Remote Method Invocation) es un Método de Invocación Remota de Java, se lo utiliza para la comunicación con servidores en aplicaciones distribuidas, solamente utilizando el lenguaje de programación Java.
- **E-mail.-** mediante la implementación del protocolo SMTP permite el envío y recepción de e-mails desde la programación en java
- **JMS.-** (Java Message Service) es el API o librería de los Servicios de Mensajes de Java, es un estándar de aplicaciones para poder crear, enviar, recibir y leer los mensajes.
- **Servicios Web.-** (Web Service) es un conjunto de estándares y protocolos para la comunicación entre aplicaciones desarrolladas en diferentes lenguajes de programación y ejecutadas en diferentes plataformas.

2.6.4. TECNOLOGÍAS

La Edición Empresarial de Java también posee un conjunto de tecnologías para soluciones empresariales.

- **Servlets.-** son objetos que se ejecutan en un servidor web o en un servidor de aplicaciones, se los utiliza para generar páginas web de forma dinámica. Es un archivo que contiene código Java con HTML embebido.
- **JSP.-** (JavaServer Pages) es una tecnología que permite realizar páginas web dinámicas, a diferencia de un Servlet, un JSP en un archivo HTML con código Java embebido.
- **EJB.-** (Enterprise JavaBeans) proporcionan un conjunto de componentes para aplicaciones distribuidas del lado del servidor, se lo utiliza en la capa de negocios y permite al programador centrarse en la programación y no preocuparse de los problemas generales de una aplicación como son persistencia, concurrencia, seguridad, etc. Un EJB necesita un contenedor de EJB para su funcionamiento, el cual es parte de un servidor de aplicaciones.
- **JSF.-** (JavaServer Faces) es una tecnología y framework de trabajo de java utilizado para gestionar las interfaces de usuario basado en el patrón de diseño MVC.

2.7. BASE DE DATOS ORACLE 10G

Es un sistema de gestión de base de datos relacional (RDMS) actualmente considerado como el sistema de base de datos más completo a nivel mundial debido a sus características.

2.7.1. CARACTERÍSTICAS

- **Soporte de Transacciones.-** una transacción en un sistema de gestión de base de datos es un conjunto de órdenes que se efectúan conjuntamente, además se dice que posee transaccionalidad si tiene la capacidad de mantener la integridad de los datos, haciendo que estas transacciones no finalicen en un estado intermedio e inesperado.
- **Estabilidad.-** el nivel de fallos es mínimo, en producción su rendimiento es estable.
- **Escalabilidad.-** puede extender su margen de operaciones sin perder su calidad y manteniendo continuo su crecimiento de trabajo, es decir, puede

cambiar su tamaño o configuración dependiendo de las exigencias del negocio.

- **Soporte Multiplataforma.-** puede ser instalado en varias plataformas de hardware y en distintos sistemas operativos (Mac, Windows y Linux).

2.7.2. CAPAS

Oracle posee una capa física y una capa lógica como se puede observar en la Figura 2.2. Capas de Oracle, la primera posee archivos que se guardan en el disco duro de la máquina donde está instalado la base de datos, en cambio la segunda posee estructuras que mapean los datos hacia los componentes en la capa física.

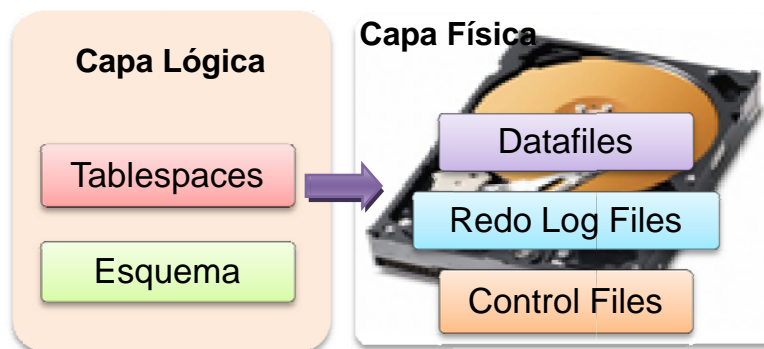


Figura 2.2. Capas de Oracle ¹²

Capa Física.- consiste de archivos físicos y pueden ser de tres tipos:

- **Datafiles.-** almacenan la información que se ingresa en la base de datos, puede existir uno o varios en el disco duro.
- **Redo Log.-** almacenan información que se utiliza para la recuperación de datos en caso de fallos y pérdidas de información.
- **Control Files.-** son archivos que se utilizan cuando se levanta una instancia de la base de datos.

Capa Lógica.- La capa lógica posee los siguientes elementos:

¹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

- **Tablespaces.-** se utiliza para separar la información en grupos y lograr simplificar su administración.
- **Esquema.-** está compuesto por tablas, índices, vistas, procedimientos almacenados, etc.

2.7.3. VERSIONES

Oracle 10g a partir de la versión Release 2, cuenta con las siguientes versiones:

- Oracle Database Enterprise Edition(EE).
- Oracle Database Standard Edition (SE).
- Oracle Database Standard Edition One (SE1).
- Oracle Database Express Edition (XE).
- Oracle Database Personal Edition (PE).
- Oracle Database Lite Edition (LE).

2.8. SISTEMA OPERATIVO LINUX

2.8.1. HISTORIA

En el año de 1991 un estudiante de la Universidad de Helsinki, Linus Torvalds comenzó el desarrollo de un núcleo para un sistema operativo similar a Unix, se basó en Minix un pequeño sistema operativo Unix que fue diseñado y desarrollado por Andrew Tanenbaum. Linus Tovalds decidió que el nuevo Kernel que estaba diseñando debería estar bajo la licencia GNU/GPL, de esta manera el 5 de octubre de 1991 se anunció la primera versión oficial de Linux Versión 0.02.

2.8.2. CARACTERÍSTICAS

GNU/Linux es el sistema operativo y Linux es su kernel o el núcleo del sistema operativo, sus características principales son las siguientes.

- **Multitarea.-** puede ejecutar varias tareas al mismo tiempo.
- **Multiusuario.-** varios usuarios pueden utilizar la misma máquina local o remotamente.

- **Multiplataforma.-** puede ser instalado y ejecutarse en diferentes plataformas de hardware.
- **Multiprocesador.-** soporta más de un procesador para la mayoría de familias de procesadores disponibles en el mercado.
- **Protección de la memoria.-** protege la memoria entre procesos de modo que no puedan colgar el sistema.
- **Carga de ejecutables por demanda.-** lectura solo de aquellas partes del disco que un programa está utilizando.
- **Gestión de Memoria.-** la memoria se gestiona como un recurso unificado para los usuarios y para la cache del disco.
- **Múltiples consolas virtuales.-** se pueden crear hasta 64 consolas para poder administrar el sistema operativo.
- **Soporte de archivos comunes.-** soporta la gran mayoría de archivos generados por otros sistemas operativos como son Windows, Mac, etc.
- **Acceso a particiones MS-DOS.-** se puede acceder a particiones FAT 32 y NTFS nativas de Windows.
- **Soporte de Protocolos de Red.-** Soporta TCP/IP, ftp, telnet y diversos protocolos de red.
- **Manejo independiente de los servicios.-** los servicios o demonios son manejados de manera independiente para que, al detener o iniciar un servicio no interfiera con el resto.

2.8.3. DISTRIBUCIONES

Una distribución de GNU/Linux comúnmente llamada distro, son paquetes de software que satisfacen las necesidades de un grupo específico de usuarios, están compuestas en su gran mayoría de software libre aunque a veces poseen algún programa propietario.

Existen distribuciones soportadas comercialmente, como por ejemplo:

- Red Hat Enterprise Linux
- OpenSuse
- Ubuntu

- Mandriva.

También existen distribuciones mantenidas por la comunidad:

- Debian
- Gentoo
- CentOS

CAPITULO 3. ANÁLISIS DEL SISTEMA

3.1. ANÁLISIS

La capacidad de innovación en una empresa es sin duda una de las mejores estrategias para alcanzar el éxito, en el caso de los restaurantes se busca satisfacer al cliente en todos los ámbitos para lograr mantenerse en el mercado; en la ciudad de Quito existen varios restaurantes que brindan un excelente servicio y ven la necesidad de expandirse para lograr abastecer a sus clientes, esto conlleva a que la información crezca y se haga más difícil su control, por este motivo se ve la necesidad de implementar un sistema informático para facilitar su control.

De acuerdo a las encuestas¹³ realizadas en las diferentes cadenas de restaurantes se identificaron los siguientes problemas:

Problemas	Causas
Falta de control del manejo de la bodega y de los inventarios.	<ul style="list-style-type: none"> • Escasa organización en el manejo de la bodega. • Falta de hojas de control para el registro de inventarios. • No realizar tomas físicas de los productos existentes en la bodega. • No registrar los productos que ingresan y salen de bodega.
Procesos de toma de pedidos y despacho de productos al cliente son inadecuados	<ul style="list-style-type: none"> • No existe organización para la toma de pedidos al cliente. • No existe un documento físico o comanda para la toma de pedidos al cliente. • No existe un documento para el despacho de los productos.
Falta de control en el	<ul style="list-style-type: none"> • La sumatoria de las facturas manualmente es extensa

¹³ Resultados de encuestas: Anexo 1, Pág. 160

cuadre de caja de la venta diaria	<p>y provoca errores.</p> <ul style="list-style-type: none"> • La inexperiencia del cajero en la utilización de una caja registradora o sistema de facturación.
Procesos para el control y procesamiento de los productos son inadecuados.	<ul style="list-style-type: none"> • No existen reportes para estimar la producción diaria. • No se maneja un control de los ingredientes que forman parte de producto. • Falta de control sobre los desperdicios de la producción. • Falta de manejo de porciones de los ingredientes que forman parte de una receta o producto. • Falta de control del despacho de productos no solicitados. • No existe un control para los productos que se dieron de baja.
Necesidades	
<ul style="list-style-type: none"> • Organizar el manejo de bodega mediante la elaboración de hojas de control. • Registrar los productos que ingresan, salen y las existencias en bodega. • Organizar la toma de pedidos al cliente. • Registrar y controlar despacho de productos al cliente. • Realizar el cuadre de caja diaria para obtener reportes de ventas. • Obtener un reporte de producción. • Controlar la producción. 	
Solución	
<p>Para tener un mejor control sobre las ventas y la elaboración de los productos de un restaurante, es necesario desarrollar un sistema que maneje producción, facturación e inventarios.</p>	

Tabla 3.1. Definición del problema ¹⁴

¹⁴ ERAZO Inés; SANDOBALÍN Julio: *Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante*; 2010

3.2. DIAGRAMA DE PROCESOS

3.2.1. CADENA DE VALOR



Figura 3.1. Cadena de Valor ¹⁵

3.2.2. PROCESOS DE NEGOCIO

Procesos	Subprocesos	Tareas	Actividades
Adquirir Materia Prima	Elegir proveedor	Solicitar propuestas de proveedores	<ul style="list-style-type: none"> • Visitar compañías • Solicitar proformas • Obtener pruebas de muestras
		Negociar con los proveedores	<ul style="list-style-type: none"> • Verificar calidad de producto • Negociar los precios de los productos
		Contratar servicio de proveedores	<ul style="list-style-type: none"> • Acordar días de entrega • Firmar el contrato para la entrega de los productos
	Adquirir pedido	Verificar existencias de bodega	<ul style="list-style-type: none"> • Toma física de los productos existentes en bodega • Tomar la decisión

¹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

			para adquirir los productos
		Solicitar productos a proveedor	<ul style="list-style-type: none"> • Pedir productos según necesidades • Recepción de la pre factura del proveedor
		Recepción de mercadería del proveedor	<ul style="list-style-type: none"> • Revisión de los productos con la pre factura del proveedor • Realizar pago • Recibir factura
Organizar	Elaborar porciones	Dividir productos en porciones	<ul style="list-style-type: none"> • Verificar estado de los productos • Realizar porciones de los productos de acuerdo a necesidades.
		Empacar porciones de productos	<ul style="list-style-type: none"> • Pesar los productos • Guardar los productos en recipientes y contenedores
	Almacenar en bodega	Clasificación de productos	<ul style="list-style-type: none"> • Dividir productos de acuerdo a características. • Verificar fechas de

			caducidad de los productos
		Guardar en bodega los productos	<ul style="list-style-type: none"> • Almacenar los productos de acuerdo a las características • Llenar hojas de control de existencias
Distribuir productos a las sucursales	Verificar existencias de productos en las sucursales	Realizar tomas físicas de los productos	<ul style="list-style-type: none"> • Verificar existencias de productos • Llenar la hoja de control de existencias
		Realizar el pedido de productos a la matriz	<ul style="list-style-type: none"> • Llenar la hoja de pedidos de la sucursal. • Realizar el pedido de productos a la matriz.
	Transportar comida	Despachar pedido a cada sucursal	<ul style="list-style-type: none"> • Entregar la hoja de pedidos a la bodega • Despacho de productos • Transporte de los producto

		Entrega de pedido en cada sucursal	<ul style="list-style-type: none"> • Recepción de los productos en cada sucursal • Revisión de los productos con la hoja de pedidos • Ingreso de los productos a la bodega de la sucursal. • Actualización de las existencias en las hojas de control.
Vender	Servir para llevar	Tomar pedido al cliente	<ul style="list-style-type: none"> • Recibir pedido • Registrar el pedido • Entregar el pedido en cada área • Preparación del producto
		Despacho de pedido al cliente	<ul style="list-style-type: none"> • Entregar el producto terminado de cada área • Entrega del producto al cliente según el pedido
		Facturar pedido al cliente	<ul style="list-style-type: none"> • Emitir la factura al cliente • Cobrar el precio de la factura al cliente
	Servir en el restaurante	Tomar pedido al cliente	<ul style="list-style-type: none"> • Entregar carta • Recibir el pedido • Registrar el pedido

			<ul style="list-style-type: none"> • Entregar el pedido en cada área • Preparación del producto
		Despachar pedido al cliente	<ul style="list-style-type: none"> • Entregar el producto terminado de cada área • Entrega del producto al cliente según el pedido
		Facturar pedido al cliente	<ul style="list-style-type: none"> • Imprimir pre cuenta • Entregar la pre cuenta al cliente • Recibir el dinero del producto consumido • Imprimir factura • Entregar la factura al cliente

Tabla 3.2. Procesos de Negocio¹⁶

3.3. DIAGRAMAS UML

3.3.1. CASOS DE USO DEL NEGOCIO

Una vez determinados los procesos del negocio, es necesario identificar los casos de uso del negocio:

¹⁶ ERAZO Inés; SANDOBALÍN Julio: *Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante*; 2010

Número	Descripción de proceso	¿Se considera Caso de Uso del Negocio?	
		Si/No	¿Por qué?
1	Solicitar propuestas de proveedores.	No	El objetivo del restaurante no es la solicitud de propuestas de productos.
2	Negociar con los proveedores.	No	El objetivo del restaurante no es negociar los productos.
3	Contratar servicio de proveedores.	No	El restaurante no tiene como fin el contratar el servicio de proveedores.
4	Verificar existencias de bodega	Si	Se verifica existencias en bodega para estimar las compras que necesita el restaurante.
5	Solicitar productos al proveedor	No	El solicitar productos es parte del negocio pero no es el objetivo del restaurante
6	Recepción de mercadería del proveedor	Si	Para tener actualizado las existencias en bodega.
7	Dividir productos en porciones	No	El dividir en porciones los productos es parte del negocio pero no es el objetivo del restaurante.
8	Empacar porciones de productos	No	El empacar las porciones de los productos es parte del negocio pero no es el objetivo del restaurante.
9	Clasificación de productos	No	Clasificar los productos es parte del negocio pero no es el objetivo del restaurante.

10	Guardar en bodega los productos	Si	Se realiza el ingreso de productos en bodega y con esto se controla las existencias para la producción.
11	Realizar tomas físicas de los productos	Si	El conteo de los productos es fundamental para el negocio saber las existencias de materia prima.
12	Realizar el pedido de productos a la matriz	Si	Al realizar los pedidos para las sucursales se generan movimientos de bodega y por ende movimientos en las existencias.
13	Despacho de pedido a cada sucursales	Si	Al despachar el pedido se generan movimientos en las existencias de los productos en las bodegas.
14	Entrega de pedido en cada sucursal	Si	En cada sucursal es necesario disponer de los productos para la preparación de las recetas y además se generan movimientos de las existencias en los productos de bodega.
15	Toma de pedido al cliente	Si	En la toma de pedido al cliente se generan descargos que se deben registrar.
17	Facturar pedido al cliente (Servir para llevar)	Si	Al facturar el pedido se lleva un control de las ventas del restaurante.
18	Facturar pedido al cliente (Servir para la mesa)	Si	Al facturar el pedido se lleva un control de las ventas del restaurante.

Tabla 3.3. Identificación de casos de uso de negocio ¹⁷

¹⁷ ERAZO Inés; SANDOBALÍN Julio: *Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante*; 2010

A continuación se presentan los diagramas de casos de uso de negocio:

Verificar existencias en bodega

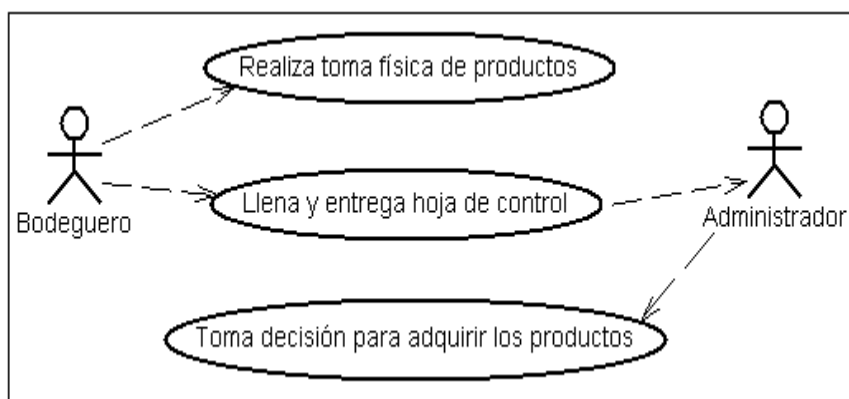


Figura 3.2. Caso de uso del negocio: Verificar existencias en bodega ¹⁸

En la figura 3.2 el Bodeguero cuenta los productos existentes en bodega es decir, realiza la toma física de los productos, luego llena la información recopilada en las hojas de control. Las hojas de control son entregadas al Administrador para que él tome la decisión de la cantidad de productos a adquirir.

Recepción de mercadería del proveedor

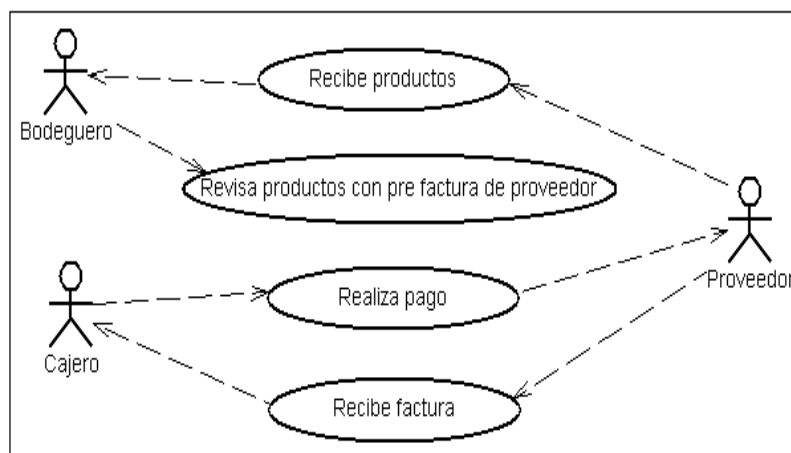


Figura 3.3. Caso de uso del negocio: Recepción de mercadería del proveedor ¹⁹

¹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

El diagrama de casos de uso de la Figura 3.3. representa la recepción de la mercadería, donde el Proveedor entrega los productos al Bodeguero, el cual revisa los productos con la pre factura que el Proveedor emitió. Una vez que el Bodeguero esté conforme con el pedido, el Cajero realiza el pago al Proveedor y este último entrega una factura.

Guardar pedido en bodega

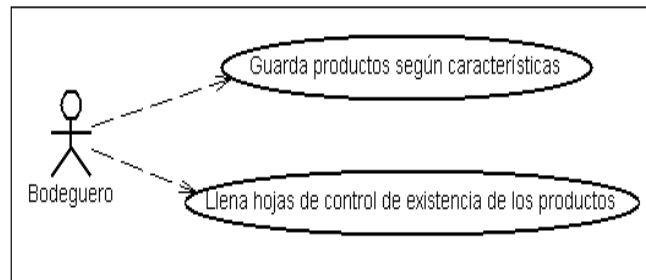


Figura 3.4. Caso de uso del negocio: Guardar pedido en bodega ²⁰

En el presente caso de uso de la Figura 3.4. el Bodeguero guarda los productos según su característica en los lugares designados, al final llena la hoja de control de existencia de los productos almacenados.

Realizar toma física de los productos

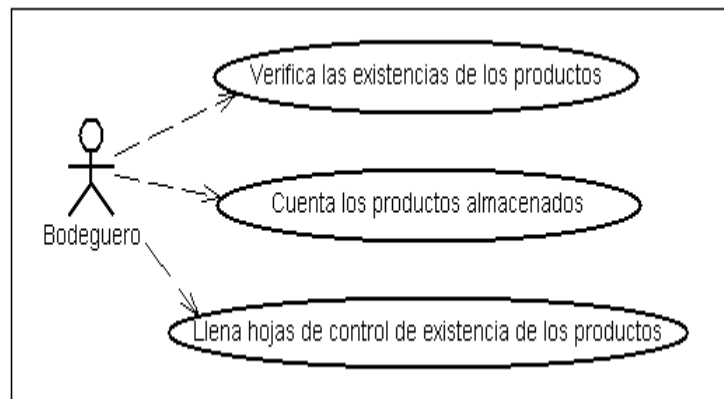


Figura 3.5. Caso de uso del negocio: Realizar toma física de los productos ²¹

En la Figura 3.5., se representa el proceso de toma física de los productos, en donde el Bodeguero primero verifica la existencia de los productos, luego

²⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

procede a contar cada producto según su característica, y finalmente llena la hoja de control de las existencias para constatar si existen un faltante o sobrante de productos.

Realizar el pedido de productos a la matriz

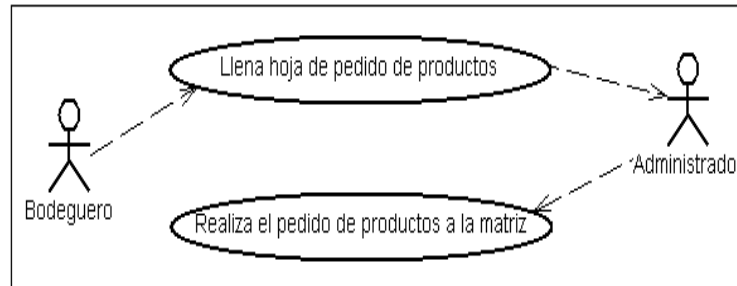


Figura 3.6. Caso de uso del negocio: Realizar el pedido de productos a la matriz ²²

En el diagrama de casos de uso de la Figura 3.6., el Bodeguero llena la hoja de pedidos de los productos según sea la necesidad de la sucursal, la misma que es entregada al Administrador. Posteriormente el Administrador realiza el pedido de los productos a la matriz.

Despacho de pedidos a cada sucursal

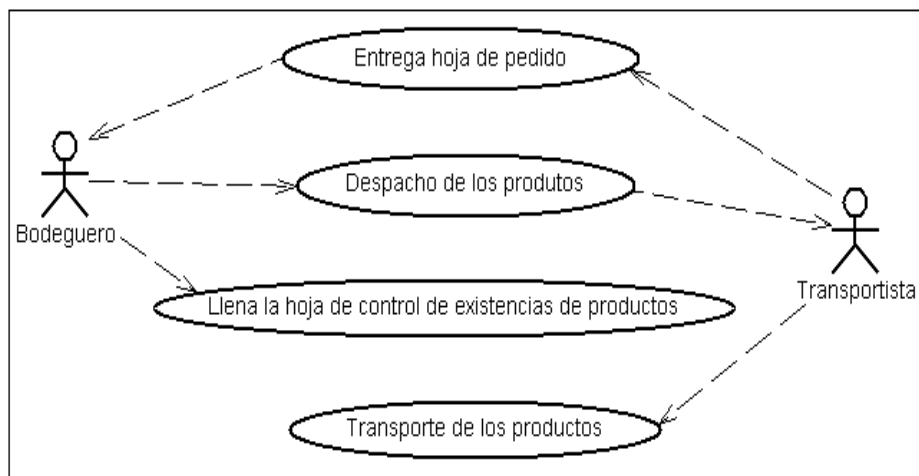


Figura 3.7. Caso de uso del negocio: Despacho de pedidos a cada sucursal ²³

En el diagrama de casos de uso de la Figura 3.7., el Transportista entrega la hoja de pedido al Bodeguero el cual separa los productos y los entrega a la

²² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Transportista. El Bodeguero llena las hojas de control de existencias para restar los productos despachados al Transportista.

Entrega de pedidos en cada sucursal

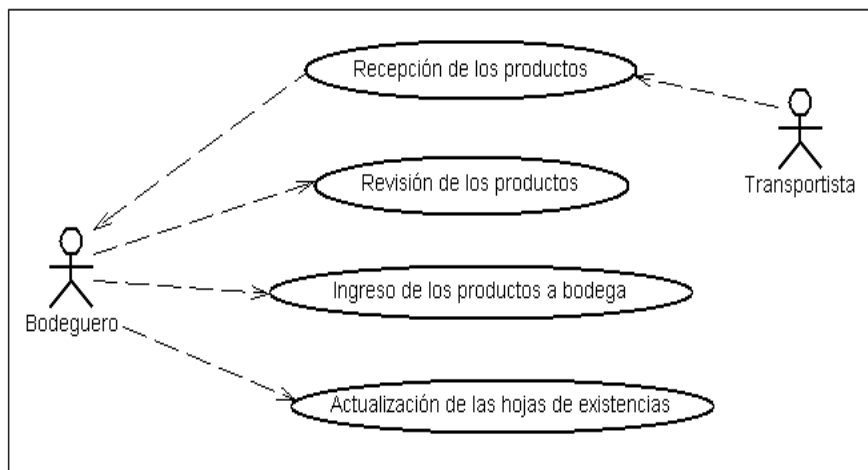


Figura 3.8. Caso de uso del negocio: Entrega de pedidos en cada sucursal ²⁴

En el diagrama de casos de uso de la Figura 3.8. el Transportista entrega el pedido al Bodeguero de cada sucursal, el cual revisa los productos del pedido y los ingresa según sus características a los respectivos contenedores en la bodega. Finalmente el Bodeguero actualiza la hoja de control de existencias con los productos recibidos.

Toma de pedido al cliente

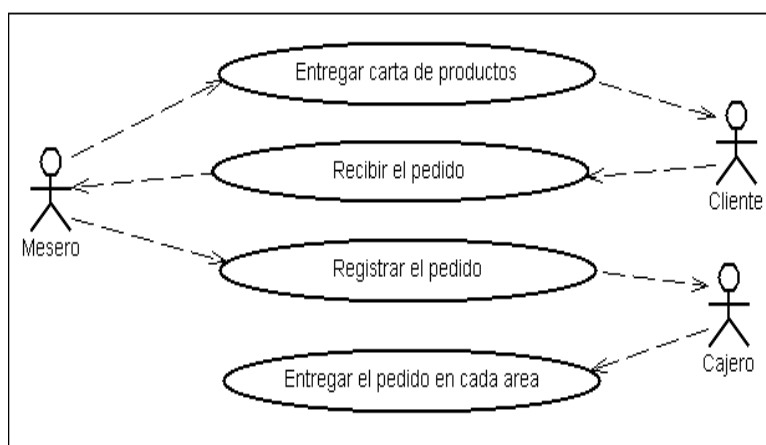


Figura 3.9. Caso de uso del negocio: Toma de pedido al cliente ²⁵

²⁴ ERAZO Inés; SANDOBALÍN Julio: *Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante*; 2010

En el diagrama de casos de uso de la Figura 3.9. el Mesero entrega la carta de productos al Cliente, el cual realiza el pedido al Mesero de los productos que va a consumir. El Mesero entrega el pedido realizado por el Cliente al Cajero y este procede a registrar el pedido, luego entrega el pedido en cada área para que se preparen los productos que solicitó el Cliente.

Facturar el pedido al cliente (Servir para llevar)

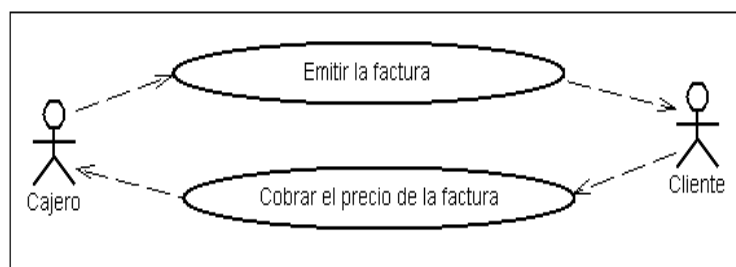


Figura 3.10. Caso de uso del negocio: Facturar el pedido al cliente (Servir para llevar) ²⁶

En el diagrama de casos de uso de la Figura 3.10. el Cajero emite la factura al Cliente, el cual entrega el dinero por el valor de los productos que consumió.

Facturar el pedido al cliente (Servir para la mesa)

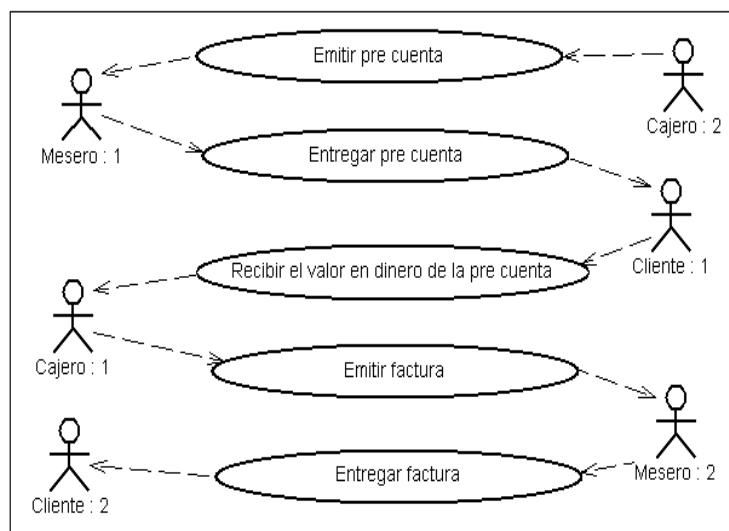


Figura 3.11. Caso de uso del negocio: Facturar el pedido al cliente (Servir para la mesa)²⁷

²⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

En el diagrama de casos de uso de la Figura 3.11., el Cajero emite la pre cuenta, la cual es entregada al mesero para que este la entregue al Cliente, el Cliente revisa su pre cuenta y entrega el valor de los productos consumidos al Mesero. Con el valor total de la pre-cuenta el Mesero entrega el dinero al Cajero, el cual emite la factura y el Mesero la entrega al Cliente.

3.3.2. DIAGRAMAS DE ACTIVIDADES DEL NEGOCIO

De acuerdo a los casos de uso de negocio identificados, por cada uno de ellos se presentan los diagramas de actividades del negocio:

Verificar existencias en bodega

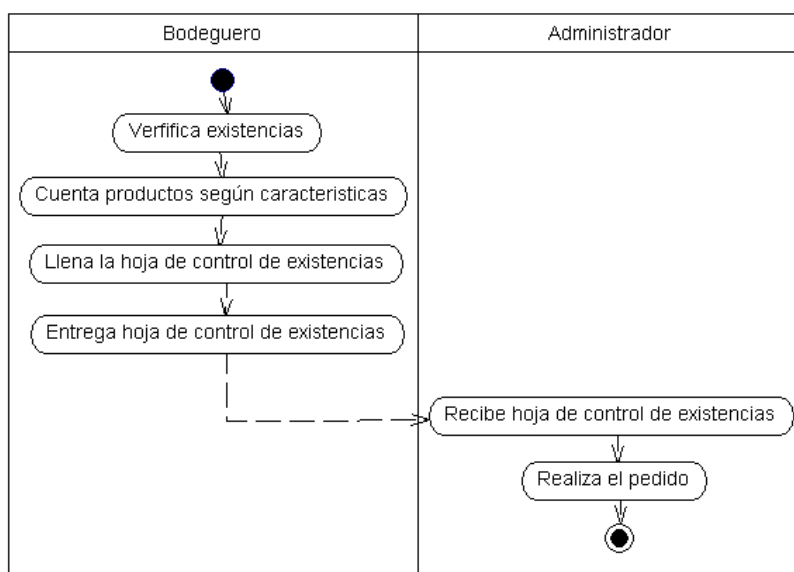


Figura 3.12. Diagrama de Actividades: Verificar existencias en bodega ²⁸

En el diagrama de actividades de la Figura 3.12. el Bodeguero para verificar las existencias en bodega primero comprueba las existencias de cada producto, luego procede a contar los productos existentes en la bodega para llenar la hoja de control, la cual será entregada al Administrador del local.

El Administrador con la hoja de existencias procederá si es necesario a realizar el pedido de los productos al proveedor para la producción.

²⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Recepción de mercadería del proveedor

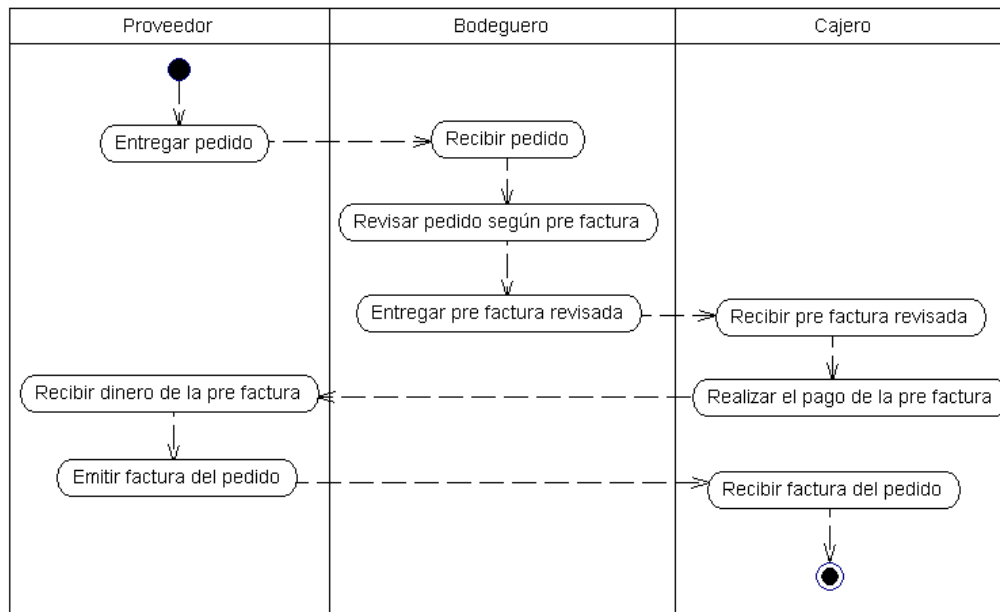


Figura 3.13. Diagrama de Actividades: Recepción de mercadería del proveedor.²⁹

En el diagrama de actividades de la Figura 3.13. al recibir la mercadería del Proveedor el Bodeguero revisa el pedido según la pre factura, una vez que la mercadería recibida esté de acuerdo con la pre factura, el Bodeguero la entrega al Cajero para que este realice el pago del pedido al Proveedor.

Finalmente el Proveedor recibe el dinero del pedido y emite la factura correspondiente, la cual es entregada al Cajero.

Guardar el pedido en bodega

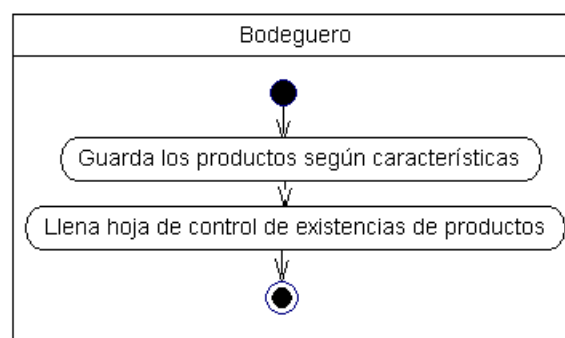


Figura 3.14. Diagrama de Actividades: Guardar el pedido en bodega³⁰

²⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

En el diagrama de actividades de la Figura 3.14., para guardar el pedido en bodega, el Bodeguero separa los productos según sus características para ser almacenados. Para finalizar el Bodeguero llena la hoja de control de existencias para registrar los productos que ingresaron.

Realizar toma física de los productos

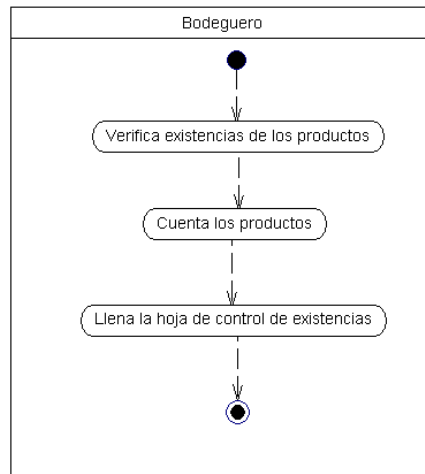


Figura 3.15. Diagrama de Actividades: Realizar toma física de los productos ³¹

En el diagrama de actividades de la Figura 3.15., para realizar la toma física de los productos el Bodeguero primero verifica las existencias de los productos en la hoja de control. El Bodeguero procede a contar los productos y finalmente llena la hoja de control de existencias para mantener actualizado el stock de los productos.

Realizar el pedido de productos a la matriz

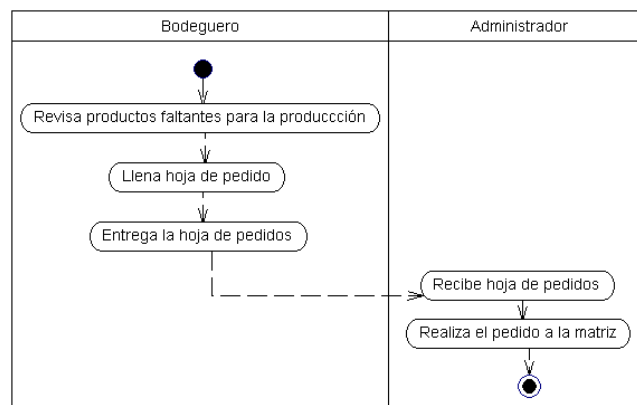


Figura 3.16. Diagrama de Actividades: Realizar el pedido de productos a la matriz ³²

³¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

En el diagrama de actividades de la figura 3.16., para realizar el pedido de los productos a la matriz, el Bodeguero revisa el stock para llenar la hoja de pedido y solicitar los productos que hacen falta para la producción, posteriormente entrega la hoja de pedido al Administrador de la sucursal.

El Administrador evalúa la hoja de pedidos y realiza el pedido de los productos necesarios para la producción a la matriz.

Despacho de pedidos a cada sucursal

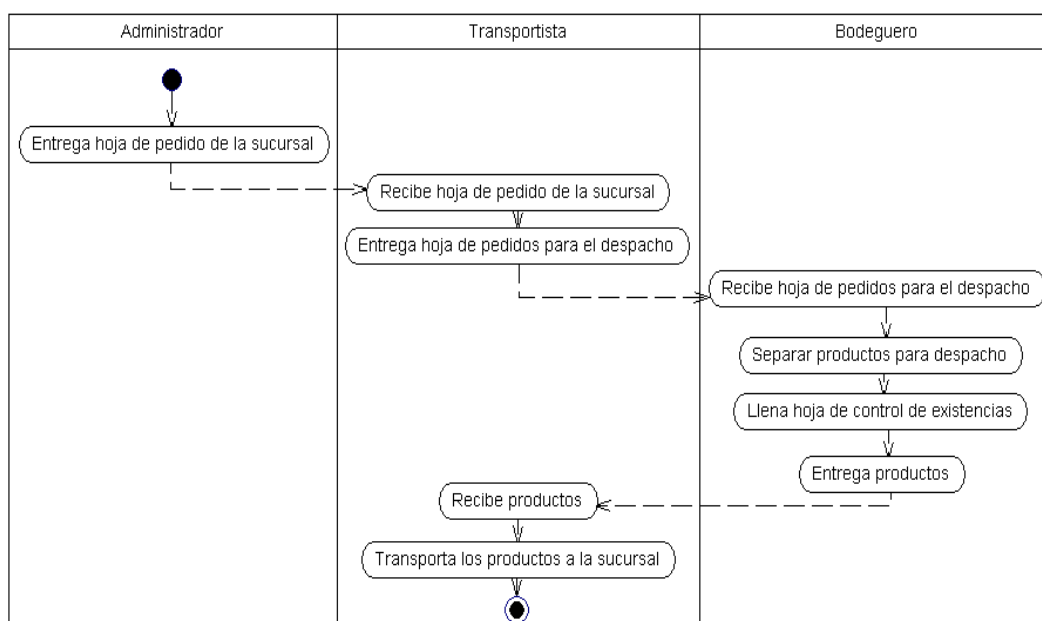


Figura 3.17. Diagrama de Actividades: Despacho de pedidos a cada sucursal ³³

En el diagrama de actividades de la Figura 3.17., para el despacho de pedidos a cada sucursal el Administrador de la matriz entrega la hoja de pedidos de cada sucursal al Transportista, el cual la entrega al Bodeguero para que separe los productos según la hoja de pedido.

Una vez que el Bodeguero terminó de separar los productos del pedido de cada sucursal, procede a llenar las hojas de control de existencias de los productos para tener actualizado el stock.

Finalmente el Bodeguero entrega los pedidos de cada sucursal al Transportista el cual se encargará de distribuirlos.

³² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Entrega de pedido en cada sucursal

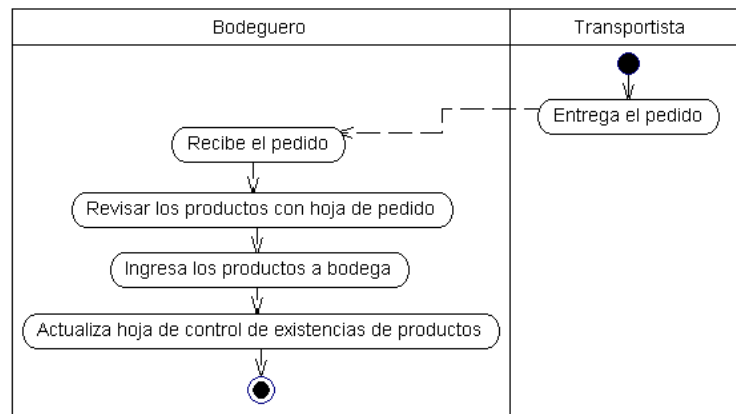


Figura 3.18. Diagrama de Actividades: Entrega de pedido en cada sucursal ³⁴

En el diagrama de actividades de la Figura 3.18. el Transportista entrega los pedidos en cada sucursal, el Bodeguero recibe el pedido en la sucursal y procede a revisar los productos según la hoja de pedidos.

Luego que el Bodeguero revisa los productos y está de acuerdo con el pedido que recibió, procede a guardarlos según sus características en los diferentes lugares de la bodega.

Finalmente el Bodeguero llena la hoja de control de existencias para mantener actualizado el stock de los productos.

Toma de pedido al cliente

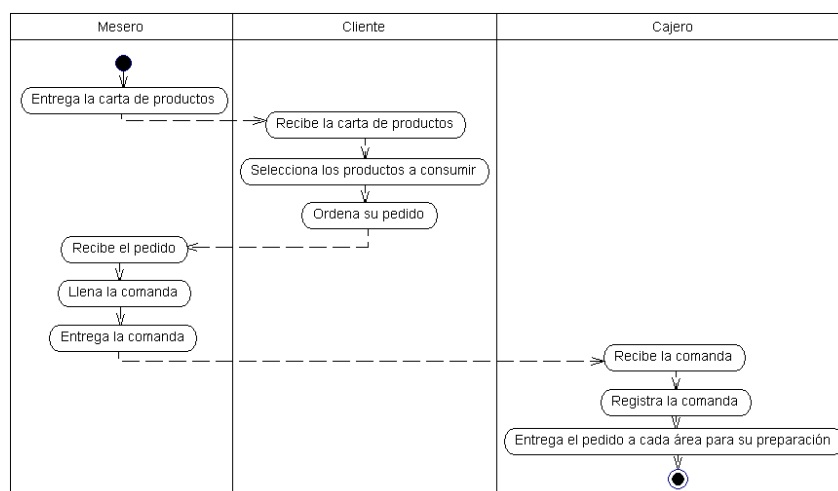


Figura 3.19. Diagrama de Actividades: Toma de pedido al cliente ³⁵

³⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

En el diagrama de actividades de la Figura 3.19. el Mesero al tomar el pedido al Cliente primero le entrega la carta de los productos, una vez que el cliente decidió los productos que va a consumir ordena su pedido al Mesero, el cual llena la comanda con los productos que el Cliente desea.

Cuando el Mesero termina de escribir el pedido del Cliente en la comanda, la entrega al cajero para que registre y distribuya el pedido en cada área y se proceda a la preparación de los productos.

Facturar el pedido al cliente (Servir para llevar)

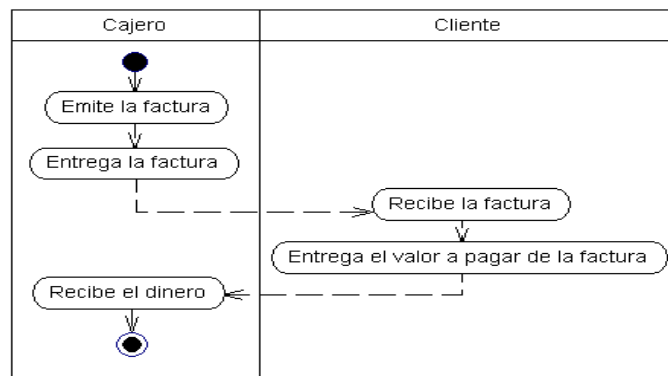


Figura 3.20. Diagrama de Actividades: Facturar el pedido al cliente (Servir para llevar)³⁶

En el diagrama de actividades de la Figura 3.20. el Cajero al facturar el pedido que el Cliente solicitó para llevar, primero emite la factura y la entrega al Cliente, el cual la recibe y revisa, para luego entregar el valor de dinero de la factura. Finalmente el Cajero recibe el dinero de la factura y termina la transacción.

³⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Facturar el pedido al cliente (Servir para la mesa)

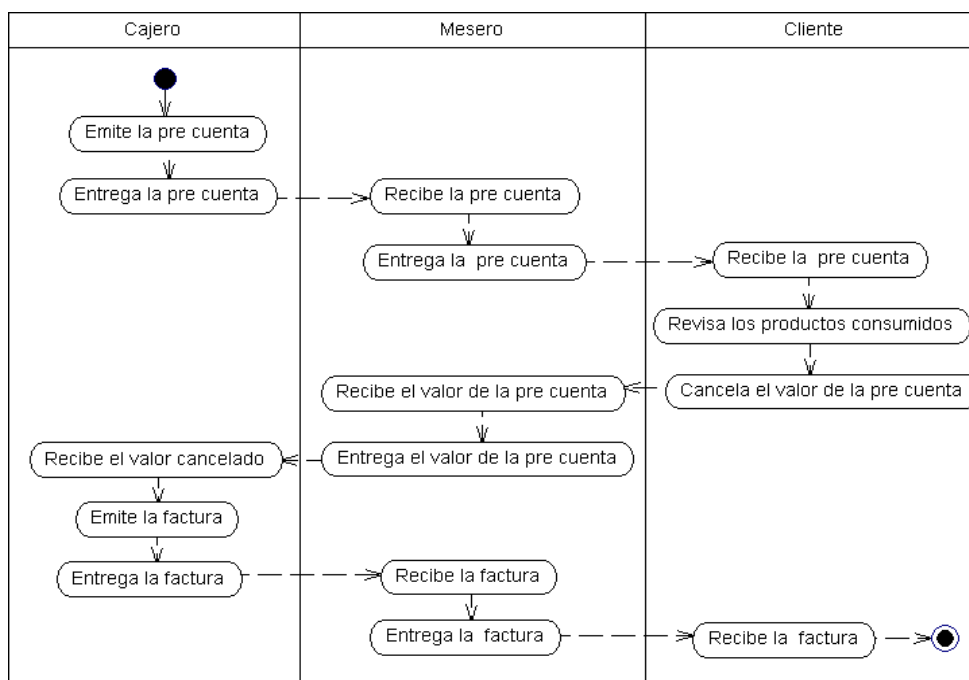


Figura 3.21. Diagrama de Actividades: Facturar el pedido al cliente (Servir para la mesa)³⁷

En el diagrama de actividades de la Figura 3.21. el Cajero para facturar el pedido que el Cliente consumió en el local, primero emite una pre cuenta la cual la entrega al Mesero quien la entregará al Cliente.

El cliente luego de revisar los productos que consumió procede a cancelar el valor de la pre-cuenta y entrega el dinero al Mesero.

El Mesero da el dinero de la pre-cuenta al Cajero y este procede a emitir la factura y la entrega al Mesero.

Para finalizar la transacción el Mesero procede a entregar la factura al Cliente.

3.3.3. DIAGRAMAS DE CASOS DE USO DEL SISTEMA

Una vez analizado el proceso que se lleva a cabo en el negocio, ahora se definen los actores y los casos de uso del sistema, para identificar los procesos que manejará el sistema de restaurantes.

³⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

3.3.3.1. DEFINICIÓN DE ACTORES DEL SISTEMA

Nº	Actor	Descripción
1	Usuario	Actor que de acuerdo a los permisos asignados en el sistema puede interactuar con el mismo, accediendo a los distintos módulos.
2	Administrador	Actor que cumple la función de administrar el sistema, tiene los permisos requeridos para acceder a todo el sistema.

Tabla 3.4. Definición de actores del sistema ³⁸

El sistema informático para administración y facturación de un restaurante consta de 3 módulos: Administración, Inventario y Facturación; para la definición de los casos de uso se tomó en cuenta el mismo esquema.

El primer caso de uso identificado, es el de ingreso al sistema:

Caso de Uso: Ingreso al Sistema

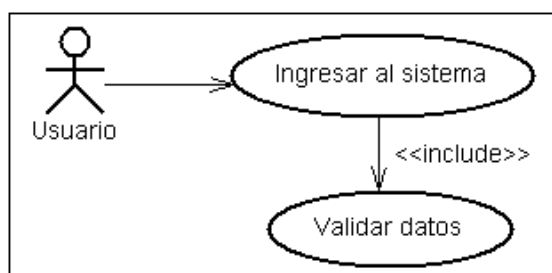


Figura 3.22. Caso de Uso del Sistema: Ingreso al Sistema ³⁹

Especificación de Caso de Uso: Ingresar al sistema

Caso de Uso	Ingresar al sistema
Objetivo	Acceder al sistema por medio de la pantalla principal del mismo.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Todos los usuarios que acceden al sistema

³⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<p>deben estar previamente registrados.</p> <ul style="list-style-type: none"> • El usuario debe conocer sus datos de autenticación: usuario y contraseña.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar al sistema. • El actor debe registrar usuario y contraseña en la primera pantalla del sistema. • El sistema valida los datos que ha ingresado el usuario. • Se le asigna los permisos al sistema. • Se muestra la pantalla del menú principal del sistema.
Pos condiciones	Si los datos ingresados por el usuario son correctos, el sistema muestra un mensaje de bienvenida.

Tabla 3.5. Especificación de caso de uso: Ingresar al sistema ⁴⁰

La definición de los casos de uso de acuerdo a los módulos del sistema, se encuentran detallados en el Manual Técnico. ⁴¹

3.3.4. DIAGRAMAS DE SECUENCIA DEL SISTEMA

Los diagramas de secuencia muestran los sucesos que se ejecutan entre objetos y se realiza un seguimiento en cada escenario, de esta manera se logra una idea general del comportamiento del sistema según los requerimientos obtenidos.

El primer diagrama de secuencia es el de ingreso al sistema:

⁴⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁴¹ Anexo 2: Manual Técnico, Modelo de Casos de Uso. Pag. 168

Diagrama de secuencia: Ingreso al sistema

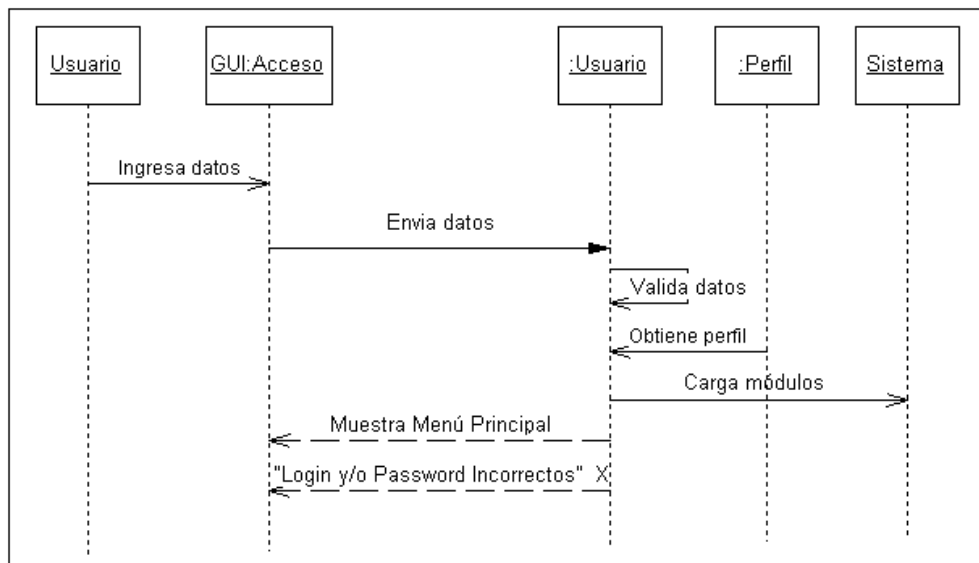


Figura 3.23. Diagrama de secuencia: Ingreso al sistema ⁴²

Los diagramas de secuencia identificados para el sistema de restaurantes, se encuentran detallados, de acuerdo a los módulos del sistema, en el Manual Técnico. ⁴³

3.4. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

3.4.1. INTRODUCCIÓN

El presente documento dará a conocer las especificaciones de requerimientos de software para el sistema de Administración y Facturación de restaurantes, el cual detallará los objetivos planteados en el proyecto, así como la planificación para el desarrollo del sistema.

3.4.1.1. PROPÓSITO

El documento da a conocer la funcionalidad, restricciones, requisitos y una perspectiva básica del sistema, con el propósito de informar al nivel gerencial sobre los temas que se incluirán en el proyecto.

⁴² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁴³ Anexo 2: Manual Técnico: Diagramas de secuencia. Pag. 242

3.4.1.2. ALCANCE

El sistema ayudará a la empresa a centralizar solo la información de ventas e inventarios de los productos; permitiendo de esta manera obtener reportes gerenciales consolidados de ventas de todas las sucursales, de igual forma se podrá llevar un control de los inventarios de cada local.

Se controlará la producción de cada local, verificando la existencia de los productos mediante una toma física de cada local y posteriormente se ingresará la información en el sistema para comparar con los productos vendidos.

Se realizará un control de producción mediante: postproducción en donde se ejecutará la producción después de la venta y preproducción en donde se ejecutará la producción antes de la venta.

3.4.1.3. PERSONAL INVOLUCRADO

Nombre	Inés Alexandra Erazo López
Rol	Análisis y Diseño de Sistemas
Categoría profesional	Tesista
Responsabilidades	<ul style="list-style-type: none"> • Recopilación de información. • Análisis del Sistema • Diseño
Información de contacto	Celular: 095620740

Tabla 3.6. Personal involucrado 1⁴⁴

Nombre	Julio Cesar Sandobalín Guamán
Rol	Análisis y Desarrollo de Sistemas
Categoría profesional	Tesista
Responsabilidades	<ul style="list-style-type: none"> • Análisis de sistema

⁴⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Desarrollo del sistema • Implementación
Información de contacto	Celular: 084088956

Tabla 3.7. Personal involucrado 2 ⁴⁵

3.4.1.4. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

Java Edición Empresarial.- se lo define como una plataforma de programación que sirve “para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones” ⁴⁶.

Oracle 10g.- es una base de datos comercial que permite el soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

Linux.- es un sistema operativo de libre distribución, tipo Unix, es portable, multitarea y multiusuario.

TCP/IP.- Protocolo de Control de Transmisión / Protocolo de Internet.

3.4.2. DESCRIPCIÓN GENERAL

3.4.2.1. PERSPECTIVA DEL PRODUCTO

El sistema que se va a desarrollar se lo hace de manera independiente, esto es sin auspicio de ninguna empresa, de acuerdo a las necesidades detectadas en el análisis de requerimientos de este tipo de negocios, llevado a efecto como parte del proyecto de tesis, en el cual se determinó que lo que se necesita sistematizar es el proceso de facturación e inventarios; cabe destacar que este producto no forma parte de otro sistema.

El sistema se desarrollará en entorno web para facilitar la centralización de la información, el control de facturación y la administración de los inventarios.

⁴⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁴⁶ WIKIPEDIA La Enciclopedia libre. JAVAEE. [En línea]. http://es.wikipedia.org/wiki/Java_EE. [Citado el 21 de mayo del 2010].

3.4.2.2. FUNCIONALIDAD DEL PRODUCTO

El sistema constará de tres módulos los cuales se detallan a continuación:

Administración: se podrá realizar el ingreso de los datos de la empresa, obtener respaldos, y generar reportes gerenciales.

Inventarios: se realizará el ingreso de los productos por líneas y grupos, a la vez que se podrá manejar tres tipos de productos, un tipo de producto normal, otro para recetas y uno para servicios. En este módulo se realizará la producción de los productos mediante recetas. Finalmente se obtendrá reportes sobre los inventarios.

Facturación: se manejará cuentas en espera para mayor comodidad del cliente. Además se obtendrá consultas y reportes de la facturación.

3.4.2.3. CARACTERÍSTICAS DE LOS USUARIOS

Tipo de usuario	Cajero
Formación	Bachiller
Habilidades	Conocimientos básicos de computación
Actividades	Facturación

Tabla 3.8. Características del usuario Cajero ⁴⁷

Tipo de usuario	Administrador
Formación	Bachilleres
Habilidades	Conocimientos básicos de computación Conocimientos sobre administración de inventarios.
Actividades	Ingreso de productos

⁴⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	Ingreso de inventarios
	Generación de producción
	Cuadre de caja
	Consultas y reportes

Tabla 3.9. Características del usuario Administrador ⁴⁸

3.4.2.4. RESTRICCIONES

La presente versión del sistema no permitirá la interoperabilidad con otros sistemas ya existentes debido a la necesidad de estabilizar el sistema para ponerlo en producción, además no se desarrollará el módulo de contabilidad.

El sistema será desarrollado en la plataforma de Java Edición Empresarial para que tenga un entorno web, utilizará una base de datos Oracle 10g y se ejecutará en un servidor con sistema operativo Linux.

3.4.2.5. SUPOSICIONES Y DEPENDENCIAS

El sistema se desarrollará en la plataforma Java Edición Empresarial y será implementado en plataforma web, por ende solo funcionará en servidores de aplicación que soporten java y sus componentes.

3.4.2.6. EVOLUCIÓN PREVISIBLE DEL SISTEMA

El sistema evolucionará y en las próximas versiones se incorporará los módulos de contabilidad, cartera y nómina.

Al existir varios sistemas contables en el mercado, se planificará la forma de integrar el sistema con un sistema contable.

3.4.3. REQUISITOS ESPECÍFICOS

Número de requisito	1
Nombre de requisito	Manejo de cuentas en espera en facturación

⁴⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente de requisito	Gerencial
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 3.10. Requisitos específicos - 1 ⁴⁹

El manejo de cuentas (Tabla 3.10.) en espera es fundamental para el sistema, porque en un restaurante y sus sucursales la forma de atender al cliente es mediante cuentas en espera, donde el cliente es atendido en su respectiva mesa con la toma del pedido, la misma que al final será cancelada en caja.

Número de requisito	2
Nombre de requisito	Producción
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente de requisito	Gerencial
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 3.11. Requisitos específicos – 2 ⁵⁰

El manejo del inventario de los productos (Tabla 3.11.) se realiza por medio de recetas las cuales serán procesadas mediante preproducción y postproducción dependiendo el caso del producto a inventariar y procesar.

⁴⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁵⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Número de requisito	3
Nombre de requisito	Cuadre de caja
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente de requisito	Gerencial
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 3.12. Requisitos específicos – 3 ⁵¹

Para cuadrar el dinero (Tabla 3.12.) de la facturación el administrador deberá realizar un cierre de caja y obtener un reporte financiero donde se detallarán totales de venta, las respectivas anulaciones y devoluciones, los ingresos y egresos de caja, los impuestos y las formas de pago. Se obtendrán otros reportes como el de productos vendidos y de inventarios.

Número de requisito	4
Nombre de requisito	Manejo de inventario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente de requisito	Gerencial
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 3.13. Requisitos específicos – 4 ⁵²

⁵¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁵² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

El sistema deberá llevar un manejo de inventarios (Tabla 3.13.) para los productos que ingresan a la bodega, el ingreso y salida de los productos se los realizará mediante movimientos de bodega.

Número de requisito	5
Nombre de requisito	Reportes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente de requisito	Gerencial
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 3.14. Requisitos específicos – 5 ⁵³

El sistema debe generar los respectivos reportes (Tabla 3.14.) de los módulos de facturación, administración e inventarios para poder llevar con control de referencias históricas ya sea de la producción o del manejo del inventario, de igual manera para llevar un control de la facturación.

Número de requisito	6
Nombre de requisito	Respaldos de información
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente de requisito	Gerencial
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 3.15. Requisitos específicos – 6 ⁵⁴

⁵³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁵⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

El manejo de respaldos de la información (Tabla 3.15.) es un proceso deseado para mantener siempre la información de la empresa de forma segura en caso de accidentes que llevaran a la pérdida de la misma.

3.4.4. REQUISITOS COMUNES DE LAS INTERFACES

3.4.4.1. INTERFACES DE USUARIO

Las interfaces de usuario llevarán un estándar para la ubicación de elementos como son botones, cajas de texto, cuadros, etc., porque de esta manera el usuario no se perderá al momento de pasar de una pantalla a otra o de pasar de un proceso a otro.

En la interfaz de usuario se manejará un estándar con los colores de los elementos que conforman las ventanas, de igual manera se manejará un estándar para el uso de texto como es tamaño, color, tipo de letra y lo más importante cada pantalla será intuitiva para el usuario permitiendo de esta manera un mejor trabajo con el sistema.

3.4.4.2. INTERFAZ DE HARDWARE

El sistema al ser desarrollado en la Plataforma Java Edición Empresarial, deberá ser instalado en un servidor de aplicaciones, y los usuarios deberán acceder al sistema a través de un browser o navegador de internet, por esto se debe tener claro que para instalar la aplicación se deberá contar con un servidor y para las estaciones de trabajo se tendrá un computador para cada usuario.

3.4.4.3. INTERFACES DE COMUNICACIÓN

El sistema se comunicará con el local principal y las sucursales mediante la utilización del internet utilizando el protocolo TCP/IP, debido a que dicho protocolo es utilizado para comunicaciones en la web.

3.4.5. REQUISITOS FUNCIONALES

El sistema tendrá los siguientes requisitos para su funcionamiento

Administración

- Configuración de la empresa
- Manejo de sucursales
- Respaldos de la información

- Seguridad de acceso al sistema
- Reportes gerenciales

Inventarios

- Productos por líneas, grupos
- Tipos de productos: normal, recetas y servicios
- Tres tipos de precios para los productos
- Pre producción y post producción de recetas
- Consultas y reportes

Facturación

- Cuentas en espera (cuentas abiertas)
- Formas de pago: efectivo, cheques y tarjeta de crédito
- Cierre y cuadro de caja
- Impresión de facturas y pre cuentas
- Consultas y reportes

3.4.6. REQUISITOS NO FUNCIONALES

Rendimiento

Debido a que el sistema será desarrollado en plataforma Java Edición Empresarial soporta una alta concurrencia de usuarios utilizando el sistema al mismo tiempo, cabe resaltar que al ser el canal de comunicación con el servidor será una línea dedicada de internet que nos proporcione el proveedor y la comunicación del 100% con el servidor siempre estará limitada a dicho canal de comunicación

Seguridad

El lenguaje de programación Java es un lenguaje seguro y orientado a la utilización de aplicaciones en la web, por esta razón el sistema será seguro respecto a la generación de código fuente y ejecutables.

El sistema constará con niveles de seguridad por perfiles de usuarios permitiendo el acceso al sistema a usuarios con sus respectivos login y password, limitándolos en el acceso dependiendo del perfil que tengan.

Fiabilidad

La fiabilidad del sistema estará únicamente restringida por el canal de comunicación que entre el servidor y la estación de trabajo, cabe destacar que el sistema será desarrollado en un entorno web, por esta razón el canal de comunicación es fundamental para que exista fiabilidad de los datos y no accedan intrusos para alterar o robar información

Disponibilidad

El sistema deberá tener una disponibilidad 24/7 lo que quiere decir que funcionará las 24 horas del día y los 7 días de la semana, es decir una disponibilidad del 100%, pero este porcentaje cambiará dependiendo de la calidad del canal de comunicación con el servidor y de que el servidor esté funcionando, para esto se deberá tener las respectivas normas de seguridad en el servidor como protecciones y respaldo de energía eléctrica mediante UPS, normas de respaldos de información, etc.

Mantenibilidad

El mantenimiento del sistema se lo realizará en la etapa de transición es decir cuando se instale el mismo y se detecten errores que no fueron detectados en la etapa de pruebas.

Partiendo de nuevos requisitos por parte del usuario, el sistema entrará nuevamente en etapa de transición y se realizarán las correcciones necesarias.

Portabilidad

El sistema al ser desarrollado para un entorno web, en el lado del usuario solo necesita un browser o navegador de internet, por esta razón en el lado de usuario el sistema operativo no influye para la ejecución del sistema, como ya se dijo solo necesita un navegador de internet que la mayoría de sistemas operativos lo tiene incluido o permite instalar uno.

En el lado del servidor el lenguaje de programación Java es portable, es decir funciona en la gran mayoría de plataformas y sistemas operativos, además la base de datos Oracle 10g también funciona en la mayoría de arquitecturas de hardware y sistemas operativos conocidos (Windows, Mac y Linux).

CAPÍTULO 4. DISEÑO DEL SISTEMA

4.1. DISEÑO CONCEPTUAL

Una parte de las clases que conforman el sistema se obtienen de la definición de los diagramas de secuencia.

En el diseño conceptual se representan las clases que componen el sistema, al ser una aplicación desarrollada con J2EE se definen a los entity beans y session beans como clases.

Los entity beans manejan la parte del negocio del sistema, uno de los esquemas identificados para el manejo de la empresa y sus sucursales es el que se muestra en la Figura 4.1.

Estructura Organizacional

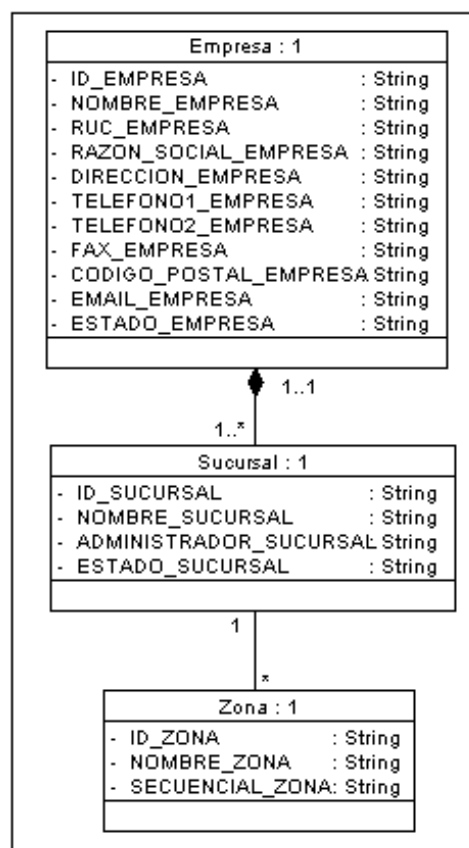


Figura 4.1. Diagrama: Estructura Organizacional ⁵⁵

⁵⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Para conocer todos los esquemas de entity beans que forman parte del sistema dirigirse al Manual Técnico.⁵⁶

4.2. DISEÑO NAVEGACIONAL

4.2.1. ESQUEMA DE CONTEXTO NAVEGACIONAL

Ingreso al Sistema.- en el esquema de la Figura 4.2., se muestra el proceso para ingresar al sistema, el usuario digita el login o nombre de usuario con su contraseña y se despliega el menú principal en donde se puede elegir las opciones: Administración, inventario, facturación, y salir.

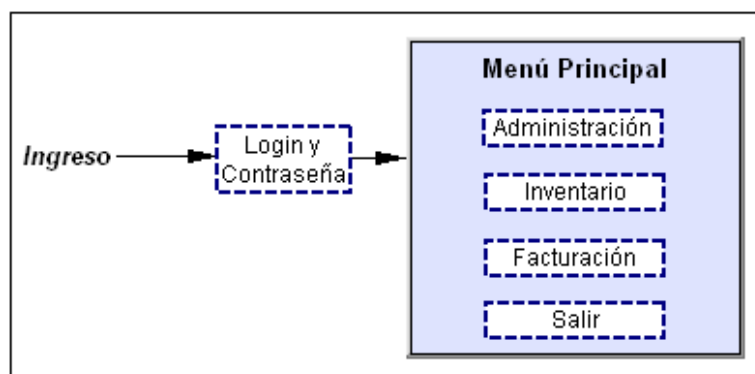


Figura 4.2. Esquema de Contexto Navegacional: Ingreso al Sistema⁵⁷

Configuración de empresa.- al ingresar a la opción empresa se podrá grabar los datos, limpiar los campos y salir, como se muestra en la Figura 4.3.

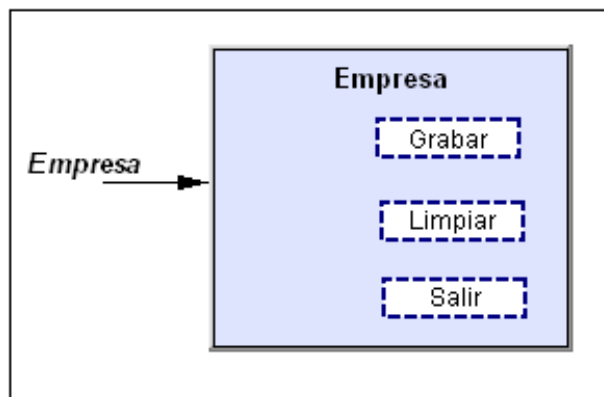


Figura 4.3. Esquema de Contexto Navegacional: Configuración de Empresa⁵⁸

⁵⁶ Anexo 2: Manual Técnico – Arquitectura de Clases. Pag. 228

⁵⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁵⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración de sucursal.- al ingresar a la configuración de sucursal se puede acceder a las opciones que se describen en la Figura 4.4. y realizar la navegación entre las dos pantallas

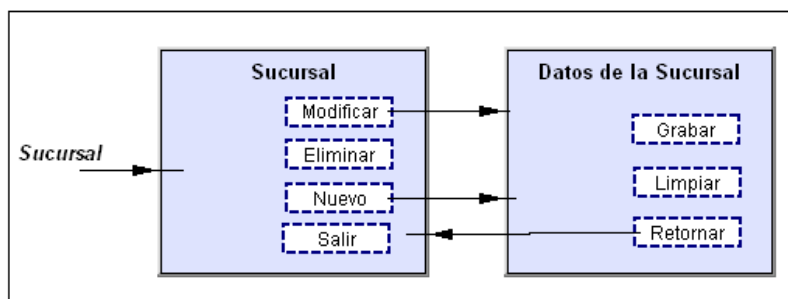


Figura 4.4. Esquema de Contexto Navegacional: Configuración de Sucursal ⁵⁹

Configuración de Ubicación Geográfica.- en configuración de la ubicación geográfica se puede acceder a las opciones que se describen en la Figura 4.5. y realizar la interacción entre las pantallas que se presentan.

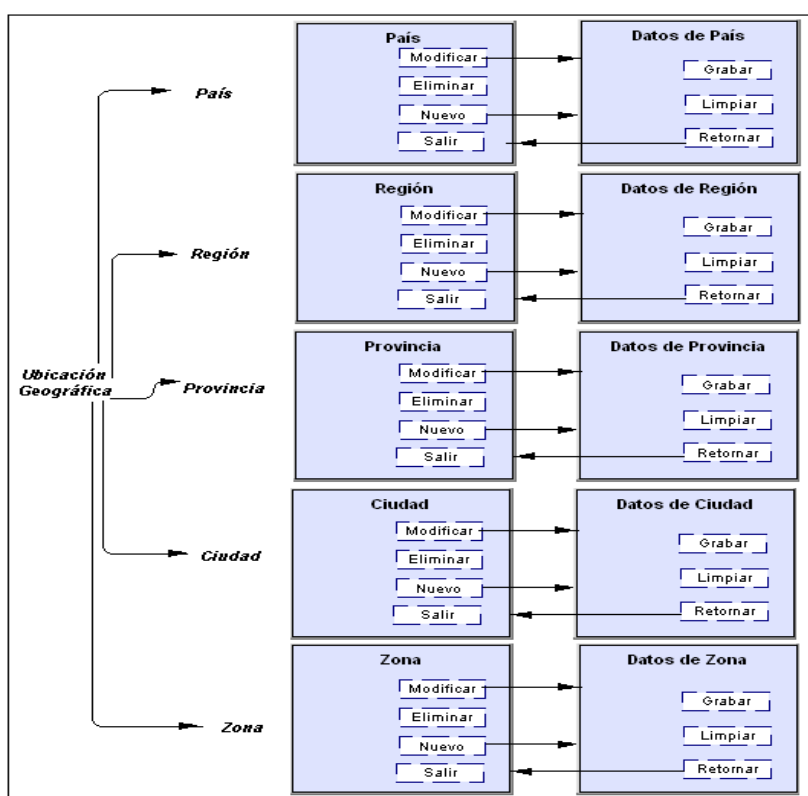


Figura 4.5. Esquema de Contexto Navegacional: Configuración de Ubicación Geográfica⁶⁰

⁵⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración de Formas de Pago.- en el esquema de la Figura 4.6., se muestra las opciones de Forma de pago y la interacción de las pantallas.

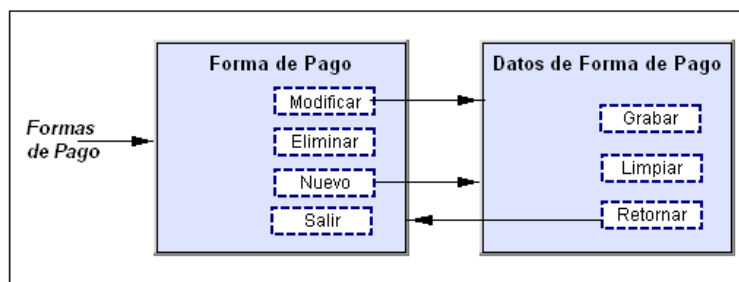


Figura 4.6. Esquema de Contexto Navegacional: Configuración de Formas de Pago⁶¹

Configuración de Impuesto.- en el esquema de la Figura 4.7, se muestra la navegacionalidad de la opción Impuestos y las distintas opciones.

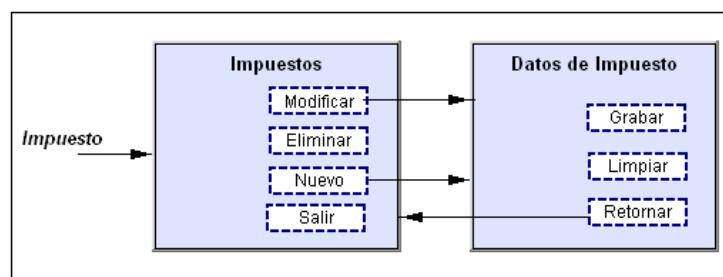


Figura 4.7. Esquema de Contexto Navegacional: Configuración de Impuesto⁶²

Configuración de Descuentos y Recargos.- en la Figura 4.8., se muestra la navegación que tendrá el usuario al ingresar a la opción descuentos y recargos.

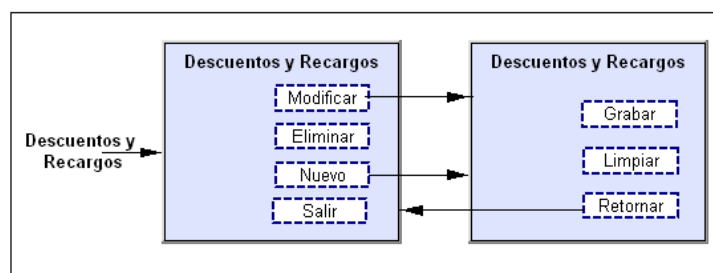


Figura 4.8. Esquema de Contexto Navegacional: Configuración de Descuentos y Recargos⁶³

⁶⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁶¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁶² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración de Perfiles de Usuario.- el usuario podrá acceder a las difentes opciones que se muestra en la Figura 4.9., cuando ingrese a Perfiles de usuario.

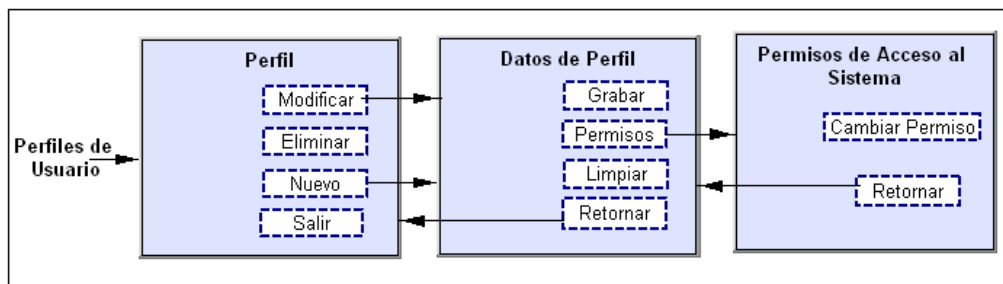


Figura 4.9. Esquema de Contexto Navegacional: Configuración Perfiles de Usuario ⁶⁴

Configuración de Usuario.- al ingresar a la opción Usuario se podrá visualizar las opciones que se muestran en la Figura 4.10., y navegar como se detalla en la misma figura.

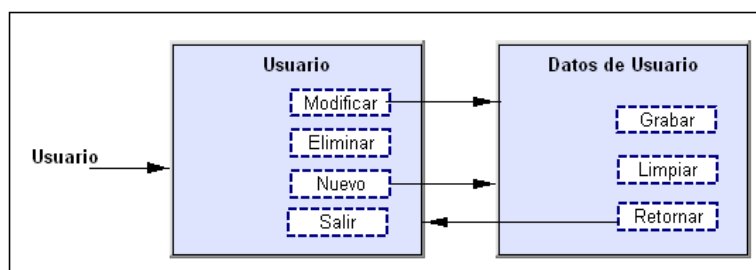


Figura 4.10. Esquema de Contexto Navegacional: Configuración de Usuario ⁶⁵

Respaldo de Datos.- en la opción respaldo de datos se tendrá una única pantalla como se muestra en la Figura 4.11.

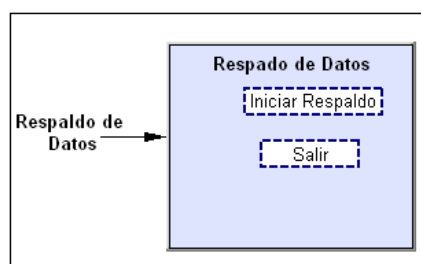


Figura 4.11. Esquema de Contexto Navegacional: Respaldo de Datos ⁶⁶

⁶³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁶⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁶⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración Línea de Productos.- al ingresar a la opción Línea de Productos el usuario podrá navegar y acceder a las distintas opciones que se muestran en la Figura 4.12.

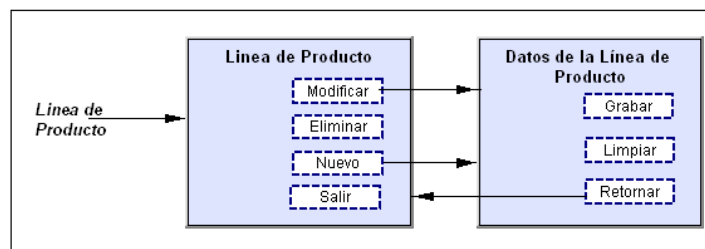


Figura 4.12. Esquema de Contexto Navegacional: Configuración Línea de Productos ⁶⁷

Configuración Grupo de Productos.- la figura 4.13., muestra en detalle las opciones a las cuales se puede acceder en Grupo de Productos y la forma en como interactúan las pantallas de esta opción.

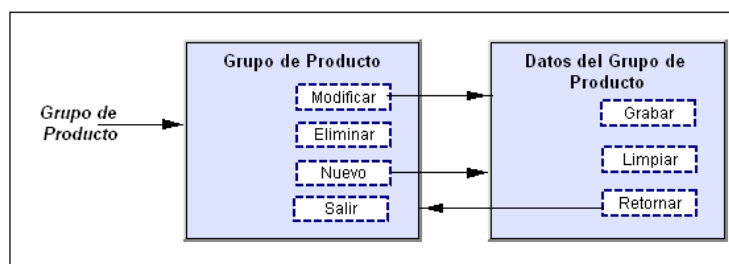


Figura 4.13. Esquema de Contexto Navegacional: Configuración Grupo de Productos ⁶⁸

Configuración Productos.- el esquema de la Figura 4.14., muestra las opciones que tienen cada pantalla y la interacción de las mismas.

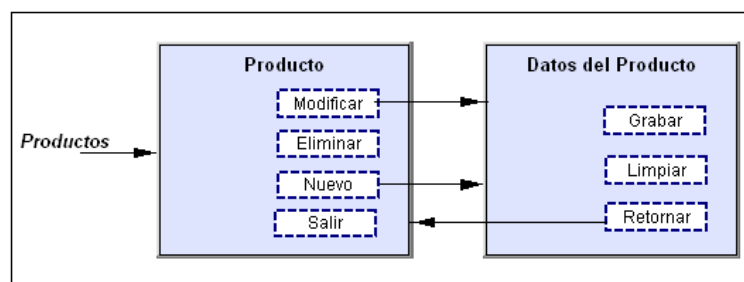


Figura 4.14. Esquema de Contexto Navegacional: Configuración Productos ⁶⁹

⁶⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁶⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁶⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración de Recetas.- si se ingresa a la opción Recetas se desplegará una pantalla con la opción Agregar ingredientes con la cual se podrá acceder a realizar cambios en los ingredientes como se visualiza en la Figura 4.15.

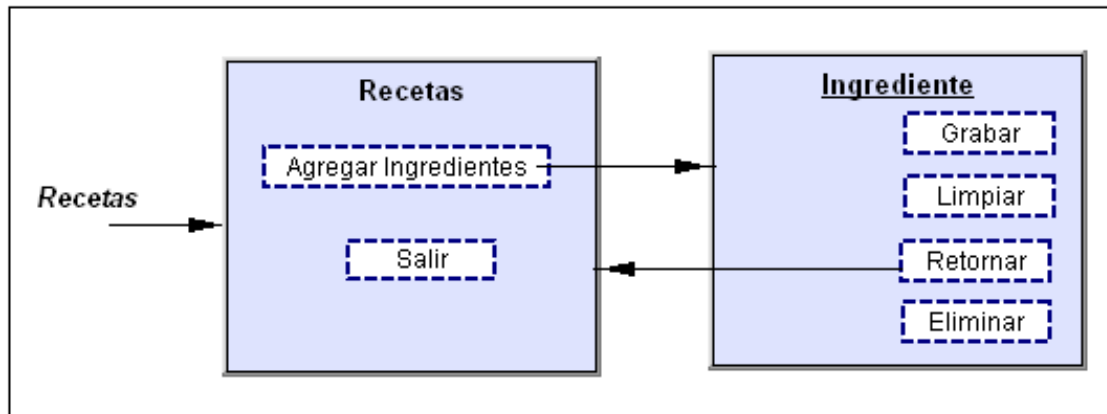


Figura 4.15. Esquema de Contexto Navegacional: Configuración de Recetas ⁷⁰

Configuración Inventario Inicial.- al acceder a la opción de Inventario se muestran las opciones de la Figura 4.16. y la navegación que existen entre las mismas.

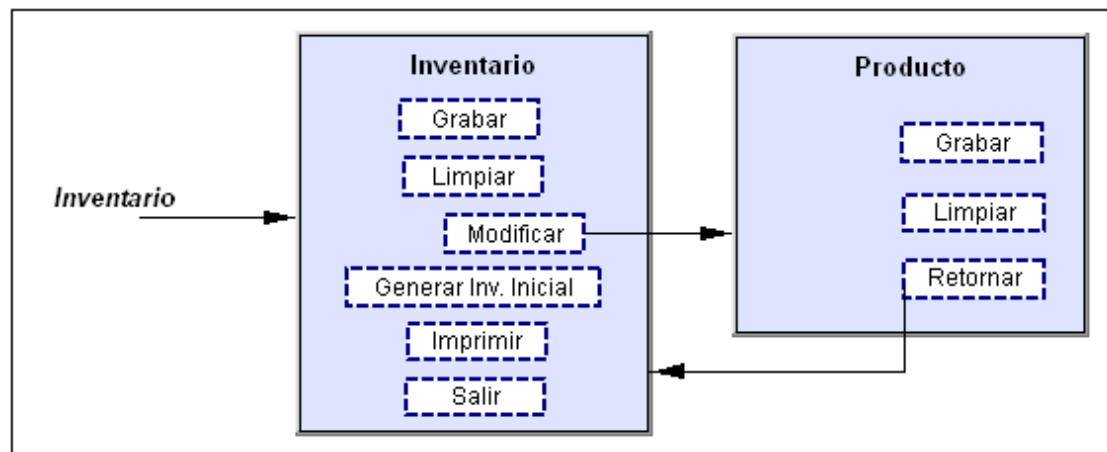


Figura 4.16. Esquema de Contexto Navegacional: Configuración Inventario Inicial ⁷¹

⁶⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁷⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁷¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración Toma Física.- el esquema de la Figura 4.17., muestra las opciones que se tiene al ingresar a Toma Física y las pantallas en las cuales se puede navegar.

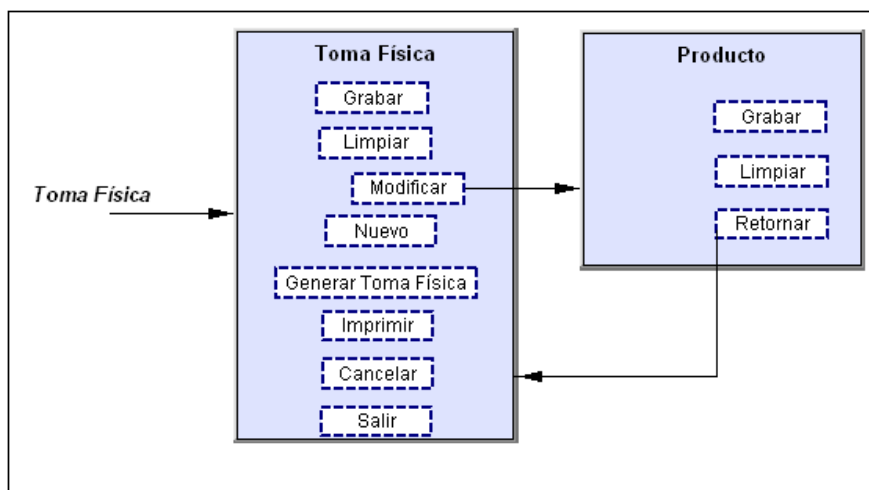


Figura 4.17. Esquema de Contexto Navegacional: Configuración Toma Física ⁷²

Configuración Movimiento de Bodega.- el esquema de la Figura 4.18., muestra las pantallas que se tiene en Movimiento de Bodega con sus respectivas opciones.

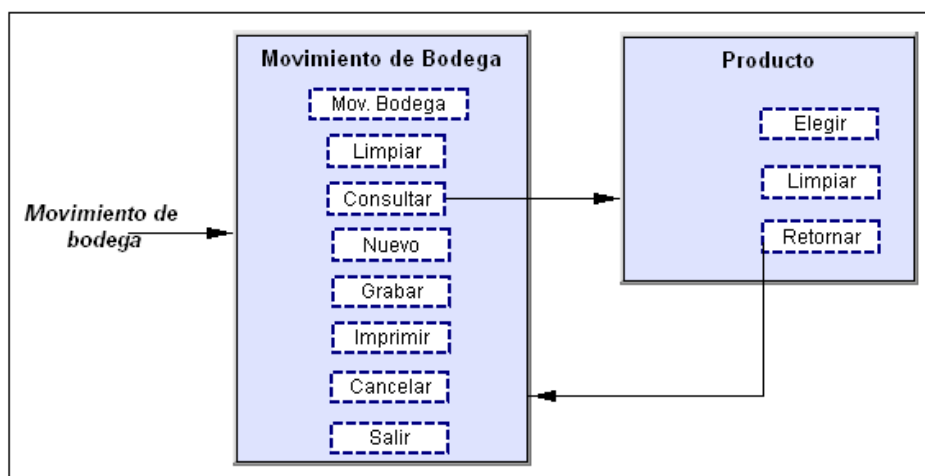


Figura 4.18. Esquema de Contexto Navegacional: Configuración Movimiento de Bodega⁷³

⁷² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁷³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración Preproducción.- al ingresar a la opción Preproducción se puede acceder a las opciones que se detallan en la Figura 4.19., y la interacción de las pantallas.

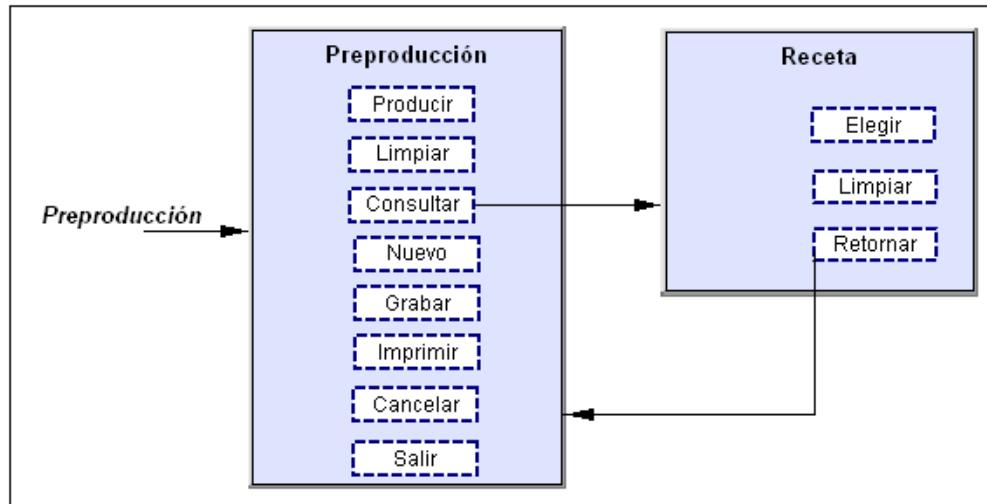


Figura 4.19. Esquema de Contexto Navegacional: Configuración Preproducción ⁷⁴

Configuración Postproducción.- las opciones a las que se puede acceder en la Figura 4.20., son las que se describen en la Figura 4.20., y las pantallas que se despliegan al ingresar a esta opción.

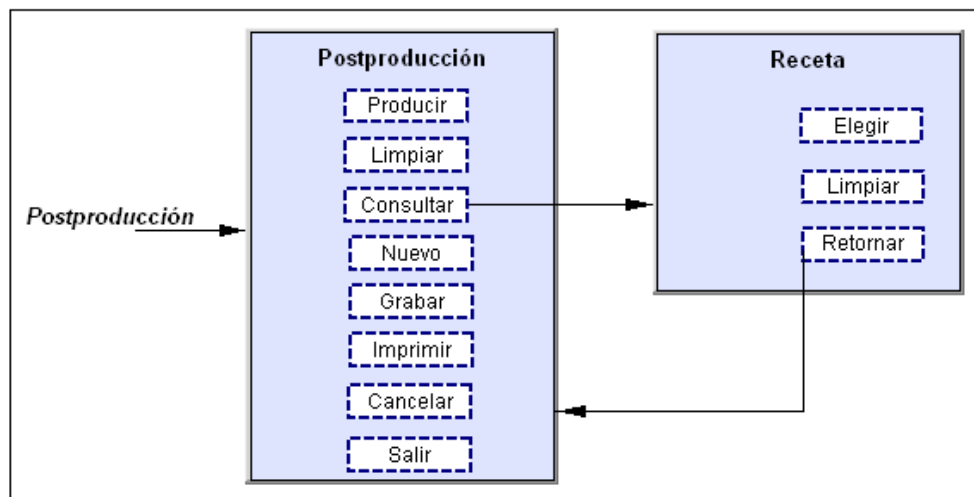


Figura 4.20. Esquema de Contexto Navegacional: Configuración Postproducción ⁷⁵

⁷⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁷⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Configuración de cliente.- al ingresar a la opción Cliente, el usuario podrá navegar en el esquema que se muestra en la Figura 4.21.

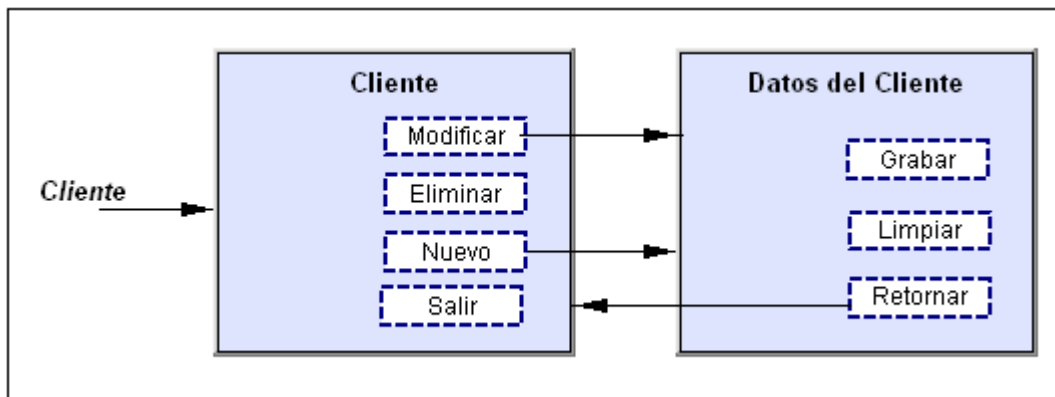


Figura 4.21. Esquema de Contexto Navegacional: Configuración Cliente ⁷⁶

Configuración de Factura.- al ingresar a la opción Factura se muestra las opciones que se describen en la Figura 4.22., y las distintas pantallas por las cuales se puede navegar.

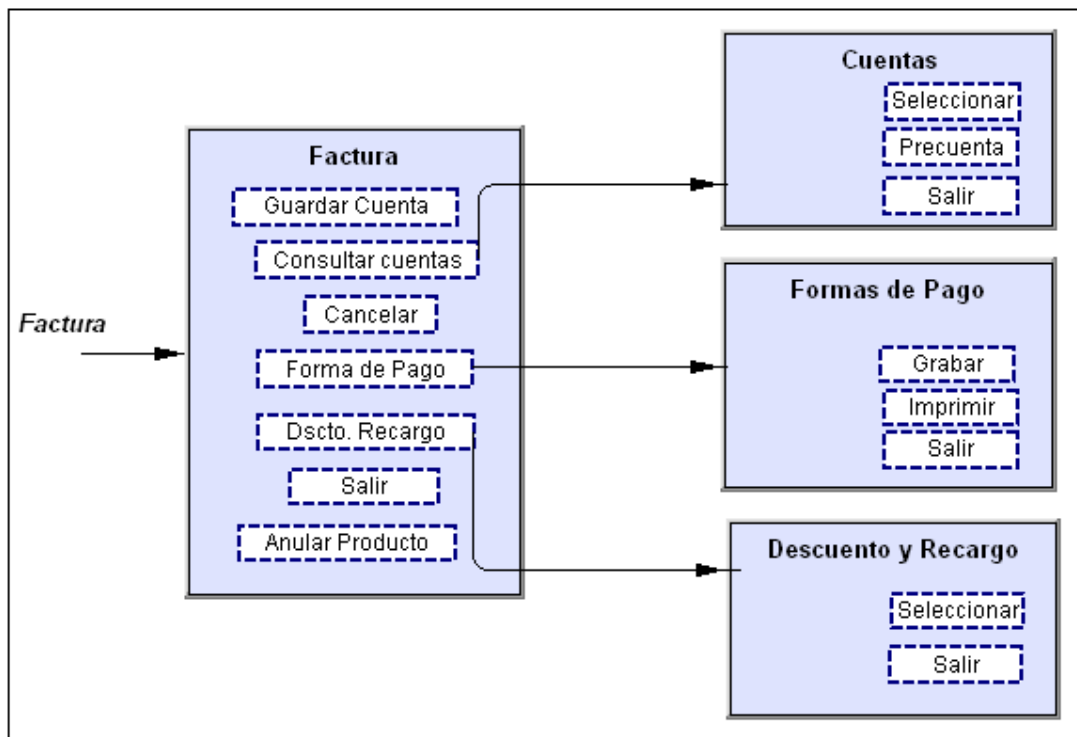


Figura 4.22. Esquema de Contexto Navegacional: Configuración Factura ⁷⁷

⁷⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Anular Factura.- al ingresar a la opción Anular factura se muestra una única pantalla con las opciones que se muestra en la Figura 4.23.

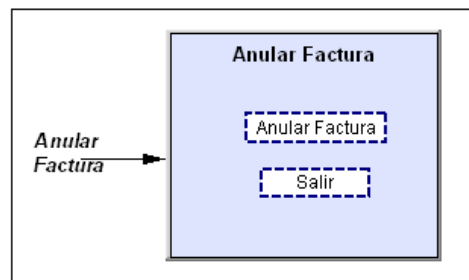


Figura 4.23. Esquema de Contexto Navegacional: Anular Factura ⁷⁸

Movimientos de Caja.- al ingresar a la opción Movimiento de caja se visualiza una pantalla con el esquema de la figura 4.24.

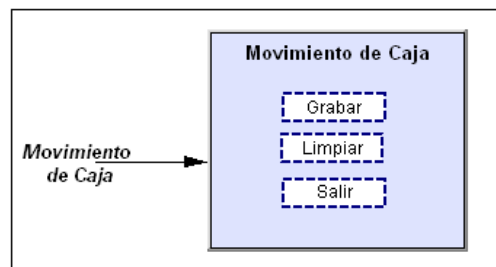


Figura 4.24. Esquema de Contexto Navegacional: Movimiento de Caja ⁷⁹

Cuadre de Caja.- se visualiza la pantalla con las opciones que se describen en la Figura 4.25.

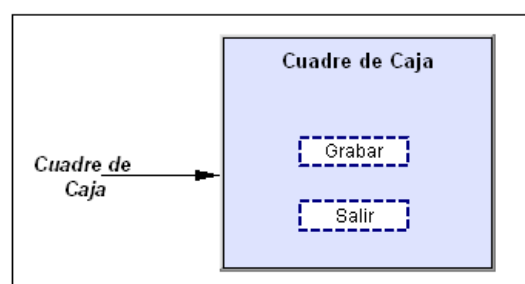


Figura 4.25. Esquema de Contexto Navegacional: Cuadre de Caja ⁸⁰

⁷⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁷⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁷⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁸⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

4.2.2. ESQUEMA DE CLASES NAVEGACIONAL

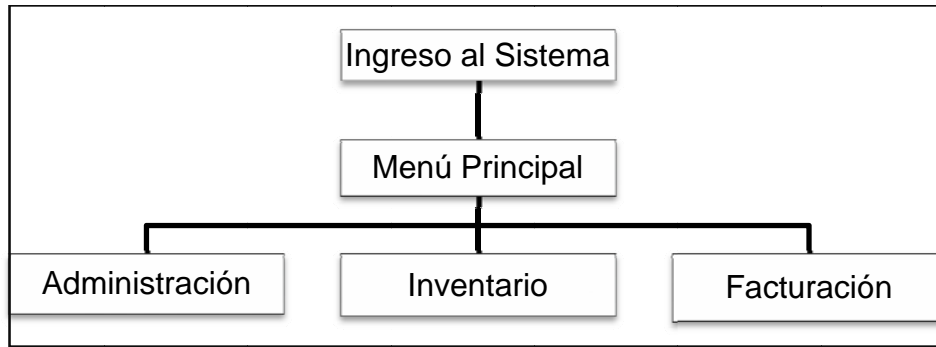


Figura 4.26. Esquema de clases navegacional 1. ⁸¹

⁸¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

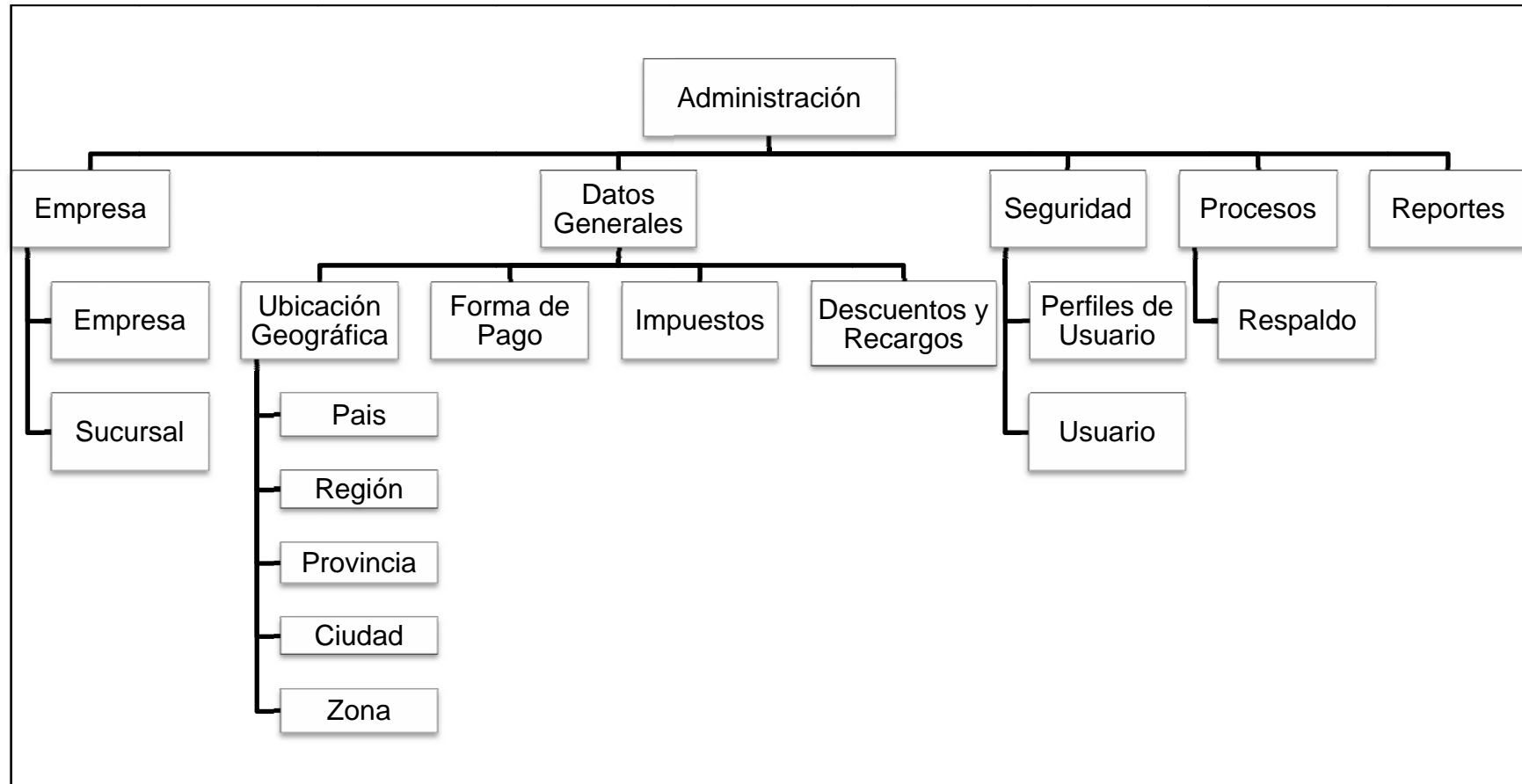


Figura 4.27. Esquema de clases navegacional 2.⁸²

⁸² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

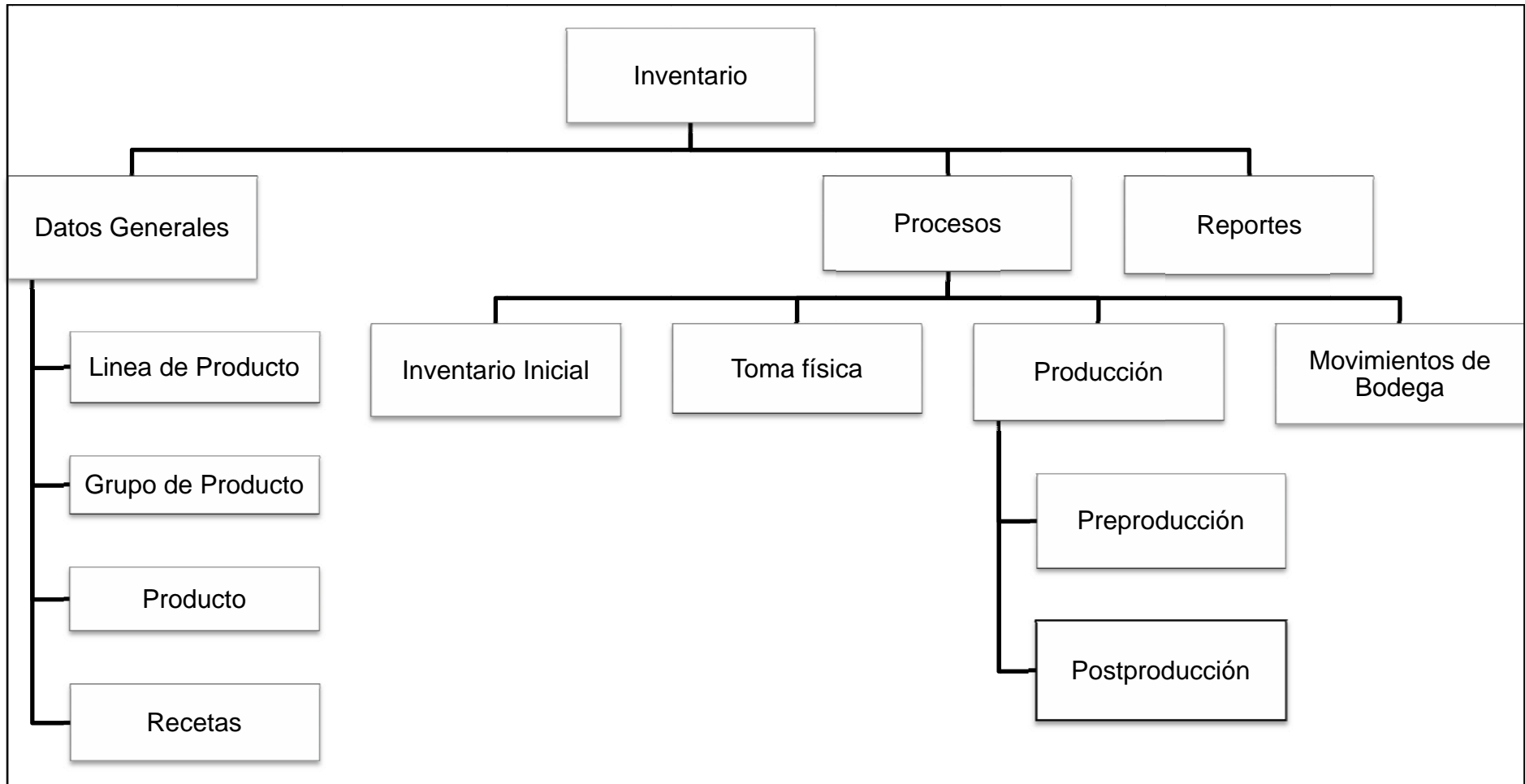


Figura 4.28. Esquema de clases navegacional 3.⁸³

⁸³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

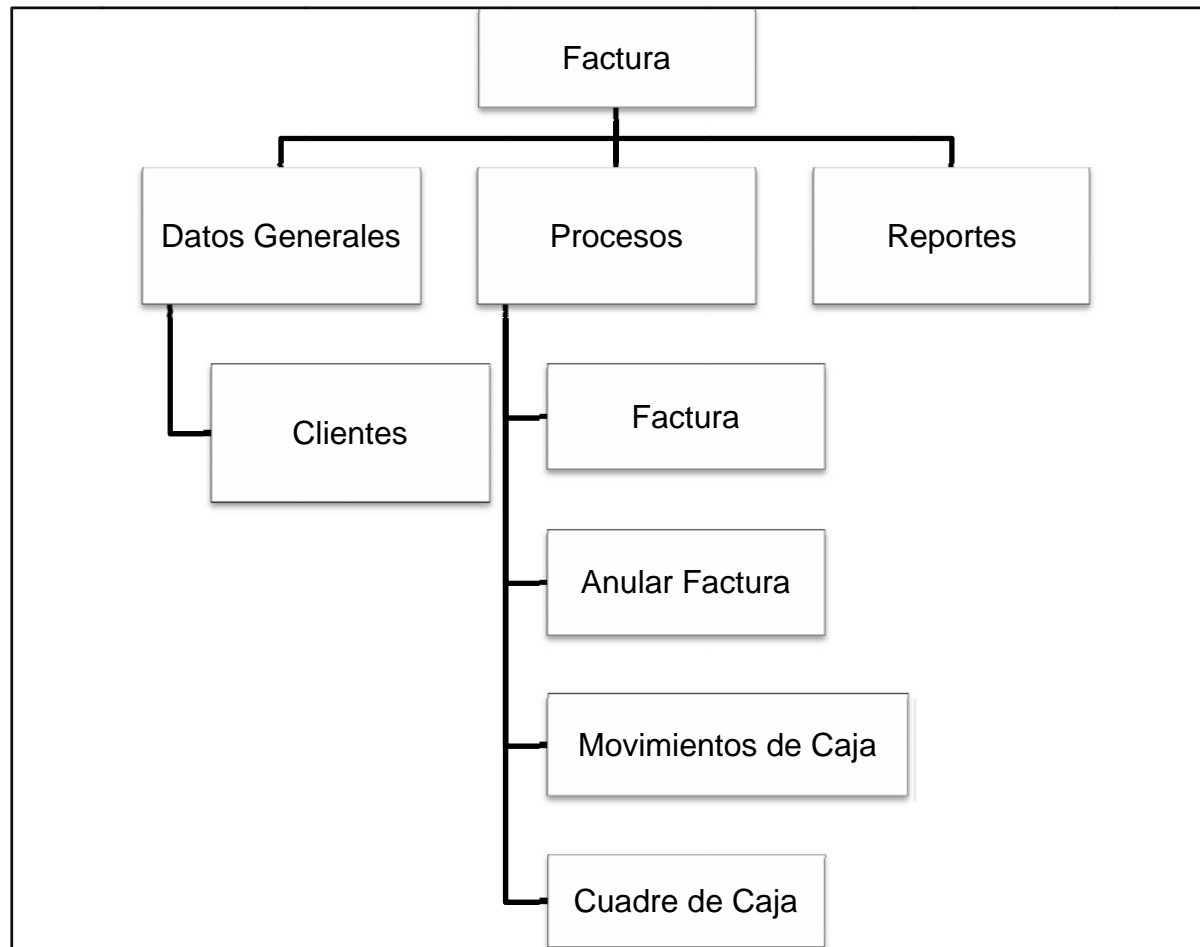


Figura 4.29. Esquema de clases navegacional 4.⁸⁴

⁸⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

4.3. DISEÑO DE INTERFAZ

Para el acceso al sistema se muestra la pantalla de la Figura 43.



Figura 4.30. Interfaz: Ingreso al Sistema ⁸⁵

El menú principal tiene las siguientes opciones: Administración, Facturación, Inventario y Salir, la pantalla de la Figura 44., muestra estas opciones.

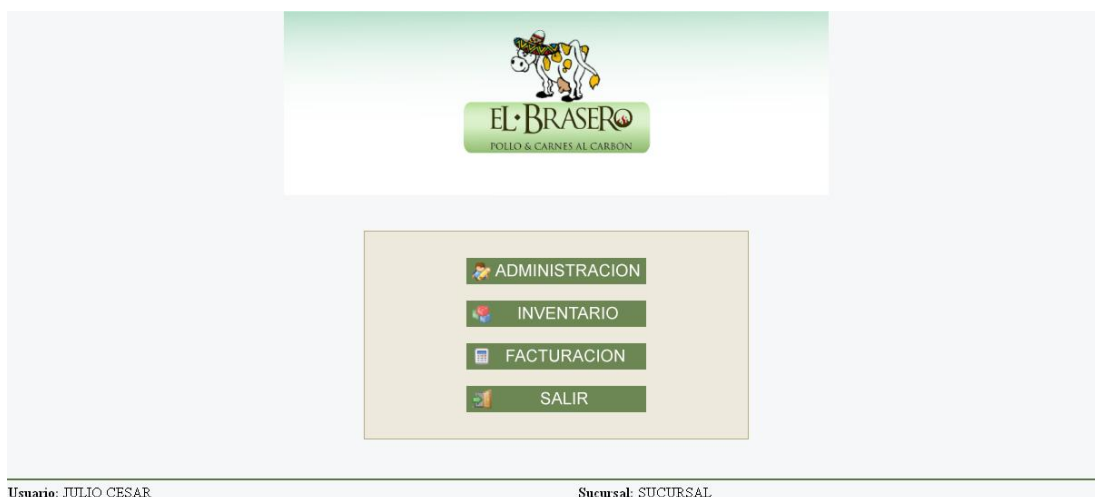


Figura 4.31. Interfaz: Menú Principal ⁸⁶

⁸⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁸⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Cuando se accede a cualquiera de las opciones del menú principal se muestra un menú secundario de acuerdo al módulo que se ingresó:

Administración:



Figura 4.32. Interfaz: Menú de Administración ⁸⁷

Inventario:

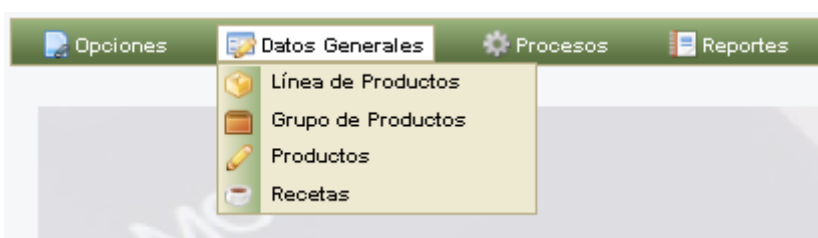


Figura 4.33. Interfaz: Menú de Inventario ⁸⁸

Facturación:



Figura 4.34. Interfaz: Menú de Facturación ⁸⁹

Para visualizar el diseño completo de la interfaz dirigirse al Manual de Usuario.⁹⁰

⁸⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁸⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁸⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁹⁰ Anexo 4: Manual de usuario. Pag. 353

4.4. DISEÑO DE BASE DATOS

De acuerdo a los módulos que posee el sistema de restaurantes, se definieron varios esquemas de base de datos como el que se muestra en la Figura 4.35.

Módulo Administración - Estructura Organizacional

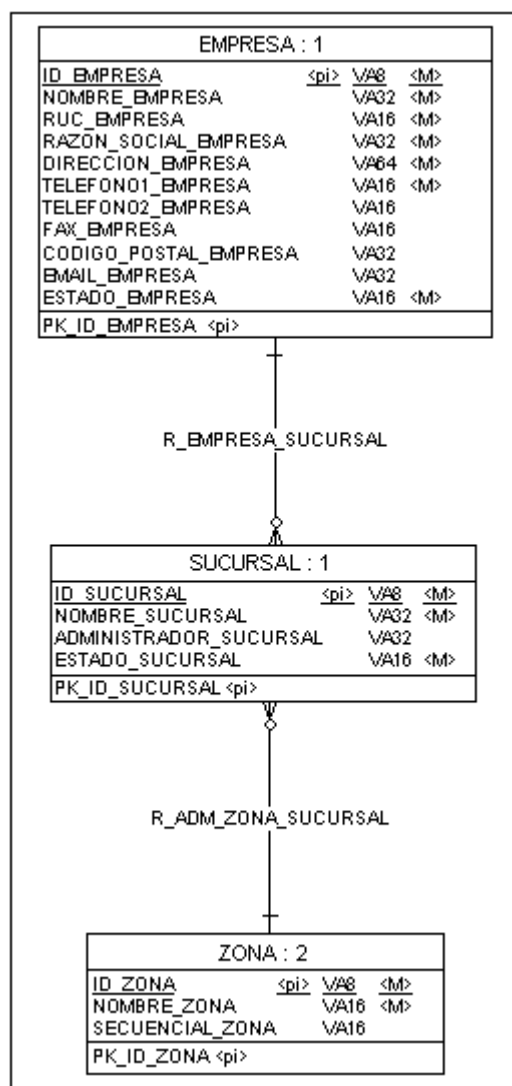


Figura 4.35. Módulo Administración – Estructura Organizacional ⁹¹

Los esquemas de la base de datos están divididos de acuerdo a los módulos del sistema, para visualizar toda la base de datos dirigirse al Manual Técnico. ⁹²

⁹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁹² Anexo 2: Manual Técnico – Diagramas Entidad Relación. Pag, 314

CAPÍTULO 5. IMPLEMENTACIÓN DEL SISTEMA

5.1. MODELO VISTA CONTROLADOR

Para el desarrollo del sistema se utilizó el patrón de diseño MVC (Modelo Vista Controlador).

5.1.1. INTERFAZ DE USUARIO

Para describir en forma rápida el funcionamiento del sistema, vamos hacer referencia al desarrollo del proceso de Facturación.

La interfaz de Facturación se muestra en la Figura 5.1.

The screenshot shows a web-based billing interface. At the top, there are input fields for 'Factura:' (001 - 001 - 261), 'Cuenta #:' (000), 'Ruc:', 'Fecha:' (2010 - 06 - 31), 'Cliente:', 'Hora:' (11 : 37), 'Teléfono:', 'Vendedor:' (JULIO CESAR), and 'Dirección:'. Below this is a section for 'Código:' and 'Cantidad:' (0), with a 'Descripción:' field. A row of buttons includes 'Guardar Cuenta', 'Consultar Cuentas', 'Cancel', 'Forma de Pago', 'Dcto - Recargo', and 'Salir'. A table titled 'PRODUCTOS' has columns for 'CANTIDAD', 'NOMRE', 'PRECIO UNIDAD', and 'PRECIO TOTAL'. The table contains one row with '0.000' in the quantity column and '0.00' in the unit and total price columns. At the bottom, a summary box shows 'Subtotal:', 'Descuento:', 'Recargo:', 'IVA:', and 'TOTAL:' all with values of '0.00'.

CANTIDAD	NOMRE	PRECIO UNIDAD	PRECIO TOTAL
0.000		0.00	0.00

Subtotal: 0.00
 Descuento: 0.00
 Recargo: 0.00
 IVA: 0.00
 TOTAL: 0.00

Figura 5.1. Interfaz: Facturación ⁹³

⁹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

JSF

“La tecnología JavaServer Faces es un marco de trabajo de interfaces de usuario del lado de servidor para aplicaciones Web basadas en tecnología Java, nos permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecida por arquitectura UI del lado del cliente.”⁹⁴

El código xhtml del JSF correspondiente al proceso de Facturación es el siguiente:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:rich="http://richfaces.org/rich"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/template/template_fac.xhtml">
<ui:define name="page"> <title>#{facturaBean.title}</title></ui:define>
<ui:define name="menu"></ui:define>
<ui:define name="title"> </ui:define>
<ui:define name="body">
<h:form id="frm_fatura">
<h:panelGrid columns="1" width="600px">
<rich:panel>
<h:panelGrid columns="1" width="570">
<h:panelGroup>
<center><h:outputText id="_mensajeDatos" value="#{facturaBean.mensajeDatos}" style="color: blue;"
/></center>
</h:panelGroup>
</h:panelGrid>
<h:panelGrid columns="2">
<h:panelGroup>
<center><h:panelGrid columns="2">
<h:panelGroup>
<h:outputText value="Factura: " />

```

⁹⁴ http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233

```

</h:panelGroup>
<h:panelGroup>
<h:inputText id="factura" value="#{facturaBean.numeroFactura}" style=" width : 200px; color: black;"
disabled="true" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="#{msg.ruc}" />
</h:panelGroup>
<h:panelGroup>
<h:panelGrid columns="2">
<h:panelGroup>
<h:inputText id="ruc" value="#{facturaBean.ruc}" style=" width : 180px;" />
<rich:suggestionbox id="suggestionCliente" for="r suggestionAction="#{facturaBean.suggestionCliente}"
var="result" fetchValue="#{result.cedulaRucCliente}"tokens="," width="400">
<f:facet name="header">
<h:outputText value="Seleccione un Cliente" />
</f:facet>
<f:facet name="nothingLabel">
<h:outputText value="No existe el Cliente" />
</f:facet>
<h:column>
<h:outputText value="#{result.cedulaRucCliente}" />
</h:column>
<h:column>
<h:outputText value="#{result.nombreCliente}" />
</h:column>
<h:column>
<h:outputText value="#{result.apellidoCliente}" />
</h:column>
<a4j:support ajaxSingle="true" event="onselect"action="#{facturaBean.seleccionarCliente}"
reRender="_cliente,_direccion,_fono,_mensajeDatos">
<f:setPropertyActionListener target="#{facturaBean.idCliente}" value="#{result.idCliente}" />
</a4j:support>
</rich:suggestionbox>
</h:panelGroup>
<h:panelGroup>
<h:commandLink action="#{facturaBean.buscarCliente}">
<h:graphicImage value=" ../imgFac/buscar.png" title="Buscar Cliente" />
</h:commandLink>
</h:panelGroup>
</h:panelGrid>
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Cliente: " />
</h:panelGroup>

```

```

<h:panelGroup>
<h:inputText id="_cliente" value="#{facturaBean.cliente}" disabled="true" style=" width : 200px; color:
black;" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="#{msg.telefono}" />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="_fono" value="#{facturaBean.telefono}" disabled="true" style=" width : 200px; color: black;"
/>
</h:panelGroup>
</h:panelGrid></center>
</h:panelGroup>
<h:panelGroup><center><h:panelGrid columns="2">
<h:panelGroup>
<h:outputText value="Cuenta # " />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="cuenta" disabled="#{facturaBean.bloqueoCuenta}"
value="#{facturaBean.numeroCuenta}" style=" width : 200px; color: black;" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Fecha: " />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="fecha" value="#{facturaBean.fecha}" disabled="true" style=" width : 200px; color: black;"
/>
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Hora: " />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="hora" value="#{facturaBean.hora}" disabled="true" style=" width : 200px; color: black;" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Vendedor: " />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="vendedor" value="#{facturaBean.vendedor}" disabled="true" style=" width : 200px; color:
black;" />
</h:panelGroup>
</h:panelGrid></center>
</h:panelGroup>
</h:panelGrid>
</h:panelGroup>

```

```

<h:outputText value="#{msg.direccion}" />
</h:panelGroup>
<h:panelGroup>
<rich:spacer width="4px" />
<h:inputText id="_direccion" value="#{facturaBean.direccion}" disabled="true" style=" width : 476px; color:
black;" />
</h:panelGroup>
</rich:panel>
</h:panelGrid>
<rich:spacer height="10px" />
<h:panelGrid columns="1" width="600px">
<rich:panel>
<h:panelGrid columns="1" width="570px">
<h:panelGroup>
<center><h:outputText id="_mensajeProductos" value="#{facturaBean.mensajeProductos}" style="color:
blue;" />
</center>
</h:panelGroup>
</h:panelGrid>
<center><h:panelGrid columns="2">
<h:panelGroup>
<h:panelGrid columns="2">
<h:panelGroup>
<h:outputText value="#{msg.codigo}" />
</h:panelGroup>
<h:panelGroup>
<h:panelGrid columns="2">
<h:panelGroup>
<h:inputText id="_idCodigo" value="#{facturaBean.codigo}" style=" width : 160px;" />
<rich:suggestionbox id="suggestionProducto" for="_idCodigo"
suggestionAction="#{facturaBean.suggestionProducto}" var="_result" fetchValue="#{_result.idProducto}"
tokens="," width="400">
<f:facet name="header">
<h:outputText value="Seleccione un Producto" />
</f:facet>
<f:facet name="nothingLabel">
<h:outputText value="No existe el Producto" />
</f:facet>
<h:column>
<h:outputText value="#{_result.idProducto}" />
</h:column>
<h:column>
<h:outputText value="#{_result.descripcionProducto}" />
</h:column>
<a4j:support ajaxSingle="true" event="onselect" action="#{facturaBean.seleccionarProducto}"

```



```

reRender="_descripcion,_mensajeDatos,_mensajeProducto">
<f:setPropertyActionListener value="#{_result.idProducto}" target="#{facturaBean.idProducto}" />
</a4j:support>
</rich:suggestionbox>
</h:panelGroup>
<h:panelGroup>
<h:commandLink action="#{facturaBean.buscarProducto}">
<h:graphicImage value=" ../imgFac/buscar.png" title="Buscar Producto" />
</h:commandLink>
</h:panelGroup>
</h:panelGrid>
</h:panelGroup>
</h:panelGrid>
</h:panelGroup>
<h:panelGroup>
<h:panelGrid columns="2">
<h:panelGroup>
<h:outputText value="#{msg.cantidad}" />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="_cantidad" value="#{facturaBean.cantidad}" style=" width : 180px;">
<a4j:support ajaxSingle="false" event="onblur" action="#{facturaBean.venderProducto}"
reRender="dataTable,_mensajeProductos,_mensajeDatos,_subtotal,_descento,_recargo,_iva,_total,_canti
dad">
<f:setPropertyActionListener value="#{facturaBean.cantidad}" target="#{facturaBean.cantidad}" />
</a4j:support>
</h:inputText>
</h:panelGroup>
</h:panelGrid>
</h:panelGroup>
</h:panelGrid></center>
<h:panelGrid columns="1">
<h:panelGroup>
<h:panelGrid columns="2">
<h:panelGroup>
<rich:spacer width="18px" />
<h:outputText value="#{msg.descripcion}" />
</h:panelGroup>
<h:panelGroup>
<rich:spacer width="4" />
<h:inputText id="_descripcion" disabled="true" value="#{facturaBean.descripcion}" style=" width : 426px;
color: black;" />
</h:panelGroup>
</h:panelGrid>
</h:panelGroup>

```

```

</h:panelGrid>
</rich:panel>
</h:panelGrid>
<rich:spacer height="10px" />
<h:panelGrid columns="1" width="600px">
<h:panelGroup>
<rich:panel>
<h:panelGrid columns="3" width="600px">
<h:panelGroup> <center><h:commandButton action="#{facturaBean.guardarCuenta}"
image="../imgFac/guardarCuenta.png">
</h:commandButton></center>
</h:panelGroup>
<h:panelGroup>
<center><h:commandButton action="#{facturaBean.buscarCuentas}"
image="../imgFac/consultarCuenta.png">
</h:commandButton></center>
</h:panelGroup>
<h:panelGroup>
<center><h:commandButton action="#{facturaBean.cancel}" image="../imgBotones/cmdcancel.png">
</h:commandButton></center>
</h:panelGroup>
<h:panelGroup>
<center><h:commandButton action="#{facturaBean.formaPago}" image="../imgFac/formaPago.png">
</h:commandButton></center>
</h:panelGroup>
<h:panelGroup>
<center><h:commandButton action="#{facturaBean.buscarPorcentaje}"
image="../imgFac/dsctoFactura.png">
</h:commandButton></center>
</h:panelGroup>
<h:panelGroup>
<center><h:commandButton action="#{facturaBean.salir}" image="../imgFac/salirFactura.png">
</h:commandButton></center>
</h:panelGroup>
</h:panelGrid>
</rich:panel>
</h:panelGroup>
</h:panelGrid>
<rich:spacer height="10px" />
<rich:dataTable id="dataTable" align="center" rows="5" value="#{facturaBean.tableDetalle}" var="data"
reRender="dataTable,scroll" onRowMouseOver="this.style.backgroundColor='#B5F3FB'"
onRowMouseOut="this.style.backgroundColor=#{a4jSkin.rowBackgroundColor}">
<f:facet name="header">
<h:outputText value="P R O D U C T O S" />
</f:facet>

```

```

<rich:column width="200px" sortBy="#{data.codigo}" filterBy="#{data.codigo}" filterEvent="onkeyup"
sortOrder="ASCENDING" visible="false">
<f:facet name="header">
<h:outputText value="CODIGO" />
</f:facet>
<center><h:outputText value="#{data.codigo}" /></center>
</rich:column>
<rich:column width="100px">
<f:facet name="header">
<h:outputText value="CANTIDAD" />
</f:facet>
<center><h:outputText value="#{data.cantidad}" /></center>
</rich:column>
<rich:column width="300px">
<f:facet name="header">
<h:outputText value="NOMRE" />
</f:facet>
<center><h:outputText value="#{data.nombre}" /></center>
</rich:column>
<rich:column width="100px">
<f:facet name="header">
<h:outputText value="PRECIO UNIDAD" />
</f:facet>
<center><h:outputText value="#{data.precioUnitario}" /></center>
</rich:column>
<rich:column width="100px">
<f:facet name="header">
<h:outputText value="PRECIO TOTAL" />
</f:facet>
<center><h:outputText value="#{data.precioTotal}" /></center>
</rich:column>
<rich:column width="50px">
<f:facet name="header">
<h:outputText value="" />
</f:facet>
<h:panelGrid columns="1" width="50px">
<h:panelGroup>
<center><a4j:commandLink reRender="dataTable,_subtotal,_descento,_recargo,_iva,_total"
action="#{facturaBean.borrar}">
<h:graphicImage value="../imgFac/eliminar.png" title="Anular Producto" />
<f:param name="#{facturaBean.codigoProducto}" value="#{data.codigo}" />
</a4j:commandLink></center>
</h:panelGroup>
</h:panelGrid>
</rich:column>

```

```

<f:facet name="footer">
<rich:datascroller id="scrool" align="center" for="dataTable" maxPages="5" ajaxSingle="false" />
</f:facet>
</rich:dataTable>
<rich:spacer height="10px" />
<h:panelGrid columns="1" width="600px">
<rich:panel>
<h:panelGrid columns="3">
<h:panelGroup>
<rich:spacer width="260px" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Subtotal: " style="FONT-SIZE: medium; FONT-WEIGHT: bold;" />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="_subtotal" value="#{facturaBean.subTotal}" disabled="true" dir="rtl"
style="FONT-WEIGHT: bold; FONT-SIZE: medium; width : 200px; color: red;" />
</h:panelGroup>
<h:panelGroup>
<rich:spacer width="260px" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Descuento: " style="FONT-SIZE: medium; FONT-WEIGHT: bold;" />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="_descuento" value="#{facturaBean.descuento}" disabled="true" dir="rtl"
style="FONT-WEIGHT: bold; FONT-SIZE: medium; width : 200px; color: red;" />
</h:panelGroup>
<h:panelGroup>
<rich:spacer width="260px" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="Recargo: " style="FONT-SIZE: medium; FONT-WEIGHT: bold;" />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="_recargo" value="#{facturaBean.recargo}" disabled="true" dir="rtl" style="FONT-WEIGHT:
bold; FONT-SIZE: medium; width : 200px; color: red;" />
</h:panelGroup>
<h:panelGroup>
<rich:spacer width="260px" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="IVA: " style="FONT-SIZE: medium; FONT-WEIGHT: bold;" />
</h:panelGroup>
<h:panelGroup>

```

```

<h:inputText id="_iva" value="#{facturaBean.iva}" disabled="true" dir="rtl" style="FONT-WEIGHT: bold;
FONT-SIZE: medium; width : 200px; color: red;" />
</h:panelGroup>
<h:panelGroup>
<rich:spacer width="260px" />
</h:panelGroup>
<h:panelGroup>
<h:outputText value="TOTAL: " style="FONT-SIZE: medium; FONT-WEIGHT: bold;" />
</h:panelGroup>
<h:panelGroup>
<h:inputText id="_total" value="#{facturaBean.total}" disabled="true" dir="rtl"
style="FONT-WEIGHT: bold; FONT-SIZE: medium; width : 200px; color: red;" />
</h:panelGroup>
</h:panelGrid>
</rich:panel>
</h:panelGrid>
</h:form>
</ui:define>
</ui:composition>
</html>

```

BACKINBEAN

“El backing bean define las propiedades de los componentes de la página y los métodos que realizan el procesamiento del componente. Este procesamiento incluye la manipulación de eventos, la validación y el procesamiento asociado a la navegación.”⁹⁵

El código correspondiente al Backinbean del JSF del proceso de Facturación es el que se muestra a continuación:

```

package com.js.backinbeans.factura;
public class FacturaBean {
    private String idFactura; private String idCuadreCaja; private String idCliente;
    private String idProducto; private String idFormaPago; private String idPorcentaje;
    private String numeroFactura; private String numeroCuenta; private String ruc;
    private String cliente;private String telefono; private String direccion;
    private String fecha;private String hora;private String vendedor;
    private String mensajeDatos;private String totalPagar;private String saldo;
    private String cambio; private String totalPagoCliente; private String cantidadPagoCliente;

```

⁹⁵ <http://docs.sun.com/app/docs/doc/819-4627/6n6p1r5me?l=es&a=view>

```

private String codigo; private String cantidad; private String descripcion;
private String mensajeProductos; private String subTotal; private String descuento;
private String recargo; private String iva; private String total;
private boolean bloqueoCuenta; private String idFacturaImprimir;
private String target; public final static String ESTADO_PENDIENTE = "PENDIENTE";
public final static String ESTADO_REGISTRADO = "REGISTRADO";
public final static String ESTADO_ANULADO = "ANULADO";
public final static String ESTADO_CANCEL = "CANCEL";
@EJB private IFacturaLocal facturaLocal;
@EJB private IDetFacturaLocal detFacturaLocal;
@EJB private IUsuarioLocal usuarioLocal;
@EJB private ISucursalLocal sucursalLocal;
@EJB private ICuadreCajaLocal cajaLocal;
@EJB private IClienteLocal clienteLocal;
@EJB private IProductoLocal productoLocal;
@EJB private IStockLocal stockLocal;
@EJB private IFormaPagoLocal formaPagoLocal;
@EJB private IDetFacPagoLocal facPagoLocal;
@EJB private IPorcentajeLocal porcentajeLocal;
@EJB private IKardexLocal kardexLocal;
@EJB private IBodegaLocal bodegaLocal;
@EJB private ITomaFisicaLocal tomaFisicaLocal;
public String buscarCliente() {
    HttpSession session = getSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);
        if(validar != null){
            return validar; }
        setMensajeDatos("");
        setMensajeProductos("");
        return "fac-cliente"; }
public String buscarProducto() {
    HttpSession session = getSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);
        if(validar != null){
            return validar; }
        setMensajeDatos("");
        setMensajeProductos("");
        return "fac-producto"; }
public String buscarPorcentaje() {
/** Validar Inventarios **/
        HttpSession session = getSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);

```

```

        if(validar != null){
            return validar; }
    /** Validar Total de la Venta */
    List<Factura> lista = facturaLocal.consultar(Long.parseLong(idFactura));
    Factura factura = (Factura) lista.get(0);
    if (factura.getTotal().compareTo(new BigDecimal("0.00")) == 0) {
        setMensajeDatos(":: Error, el TOTAL de la Venta debe ser difernete de cero ::.");
        return "/fac/facturacion.xhtml"; }
    setMensajeDatos("");
    setMensajeProductos("");
    return "fac-porcentaje"; }

    public String buscarCuentas() {
    /** Validar Inventario */
        HttpSession session = getSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);
        if(validar != null){
            return validar; }
    /** Validamos id de la factura */
        if (idFactura == null || idFactura.trim().equals("")) {
            return "/fac/facturacion.xhtml"; }
        /** Factura */
        List<Factura> lista = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) lista.get(0);
        if (factura.getTotal().compareTo(new BigDecimal("0")) == 1) {
            setMensajeDatos(":: Error, deber cancelar o Totalizar la venta para
            consultar Cuenta en Espera ::.");
            return "/fac/facturacion.xhtml"; }
        setMensajeDatos("");
        setMensajeProductos("");
        return "fac-cuentas"; }

    public String salirCliente() {
        return "cliente-fac"; }

    public String salirProducto() {
        return "producto-fac"; }

    public String salirPorcentaje() {
        setMensajeDatos("");
        return "porcentaje-fac"; }

    public String salirPago() {
        List<DetFacturaPago> _lista = facPagoLocal.consultarPorFactura(Long
            .parseLong(idFactura));
        for (int i = 0; i < _lista.size(); i++) {
            DetFacturaPago _pago = (DetFacturaPago) _lista.get(i);
            _pago.setCantidadDetPago(new BigDecimal("0.00"));
            facPagoLocal.actualizar(_pago); }
    }

```

```

        setMensajeDatos("");
        setMensajeProductos("");
        setIdFacturaImprimir(null);
        return "pago-fac"; }
public String salirCuenta() {
    return "cuentas-fac"; }
public String salir() {
if(bloqueoCuenta){
    limpiar();
        setIdFactura(null);
        setIdCuadreCaja(null);
        setIdCliente(null);
        setIdProducto(null);
        setIdFormaPago(null);
        setBloqueoCuenta(false);
        return "facturacion-fac"; }
Vector<Long> id = null;
    List<DetFactura> listaDetalle = detFacturaLocal
        .consultarPorFactura(Long.parseLong(idFactura));
    if (listaDetalle.size() > 0) {
        id = new Vector<Long>();
        for (int i = 0; i < listaDetalle.size(); i++) {
            DetFactura detFactura = (DetFactura) listaDetalle.get(i);
            id.add(detFactura.getIdDetFactura()); }
    Enumeration<Long> enumeration = id.elements();
        while (enumeration.hasMoreElements()) {
            Long codigo = (Long) enumeration.nextElement();
            try {
                detFacturaLocal.borrar(codigo);
            } catch (Exception e) {
                // TODO: handle exception
                e.printStackTrace(); } } }
    List<DetFacturaPago> listaPago = facPagoLocal.consultarPorFactura(Long
        .parseLong(idFactura));
    if (listaPago.size() > 0) {
        id = new Vector<Long>();
        for (int i = 0; i < listaPago.size(); i++) {
            DetFacturaPago pago = (DetFacturaPago) listaPago.get(i);
            id.add(pago.getIdDetFactPago()); }
        Enumeration<Long> enumeration = id.elements();
        while (enumeration.hasMoreElements()) {
            Long codigo = (Long) enumeration.nextElement();
            try {
                facPagoLocal.borrar(codigo);
            } catch (Exception e) {

```



```

// TODO: handle exception
e.printStackTrace(); } } }

try {
    facturaLocal.borrar(Long.parseLong(idFactura));
} catch (Exception e) {
    e.printStackTrace(); }

limpiar();
setIdFactura(null);
setIdCuadreCaja(null);
setIdCliente(null);
setIdProducto(null);
setIdFormaPago(null);
setBloqueoCuenta(false);
return "facturacion-fac"; }

public String formaPago() {
/** Validar Inventarios **/
    HttpSession session = getSession();
    String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
    String validar = validarInventario(idBodega);
    if(validar != null){
        return validar; }

/** Validaciones **/
    if (idFactura == null || idFactura.trim().equals("")) {
        setMensajeDatos(":: Error, no existe una Factura ::");
        return "/fac/facturacion.xhtml"; }

    if (idCliente == null || idCliente.trim().equals("")) {
        setMensajeDatos(":: Error, no existe un Cliente Seleccionado ::");
        return "/fac/facturacion.xhtml"; }

    List<DetFactura> listaDetalle =
detFacturaLocal.consultarPorFactura(Long.parseLong(idFactura));
    if (listaDetalle.size() == 0) {
        setMensajeDatos(":: Error, no existen Productos a Facturar ::");
        return "/fac/facturacion.xhtml"; }

/** Factura **/
    List<Factura> lista = facturaLocal.consultar(Long.parseLong(idFactura));
    Factura factura = (Factura) lista.get(0);

/** Operaciones **/
    List<DetFacturaPago> _lista =
facPagoLocal.consultarPorFactura(Long.parseLong(idFactura));
    for (int i = 0; i < _lista.size(); i++) {
        DetFacturaPago _pago = (DetFacturaPago) _lista.get(i);
        _pago.setCantidadDetPago(new BigDecimal("0.00"));
        facPagoLocal.actualizar(_pago); }

/** Setemos los valores **/
    setTotalPagar(factura.getTotal().setScale(2, RoundingMode.HALF_UP) + "");

```

```

        setSaldo(factura.getTotal().setScale(2, RoundingMode.HALF_UP) + "");
        setTotalPagoCliente("0.00");
        setCambio("0.00");
        setTarget("_self");
        setMensajeDatos("");
        setMensajeProductos("");
        setIdFacturalImprimir(null);
        return "fac-pago"; }

public String imprimirFactura(){
if(idFacturalImprimir == null || idFacturalImprimir.trim().equals("")){
    setTarget("_self");
    setMensajeDatos(":: Error, debe Grabar la Forma de Pago antes de Imprimir la Factura
:: ");

        return "/fac/pagoFactura.xhtml"; }
setTarget("blank");
setMensajeDatos("");
HttpSession session = getSession();
session.setAttribute("idFactura", idFacturalImprimir);
session.setAttribute("estado", ESTADO_REGISTRADO);
try {
FacesContext.getCurrentInstance().getExternalContext().redirect("/TesisWebJSF/Factura");
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace(); }
return ""; }

/** Imprimir precuentanta**/
public String imprimirPrecuenta() {
    FacesContext context = FacesContext.getCurrentInstance();
    String id = getValue(context, getCodigoCuenta());
    if (id == null || id.trim().equals("")) {
        return "/fac/pagoFactura.xhtml"; }
    HttpSession session = getSession();
    session.setAttribute("idFactura", id);
    try {
FacesContext.getCurrentInstance().getExternalContext().redirect("/TesisWebJSF/Precuenta");
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace(); }
return ""; }

public String seleccionarCuenta() {
    if(idFactura == null || idFactura.trim().equals("")){
        return null; }

/** Eliminamos**/
    Vector<Long> _id = null;
    List<DetFactura> listaDetalle = detFacturaLocal

```

```

        .consultarPorFactura(Long.parseLong(idFactura));
    if (listaDetalle.size() > 0) {
        _id = new Vector<Long>();
        for (int i = 0; i < listaDetalle.size(); i++) {
            DetFactura detFactura = (DetFactura) listaDetalle.get(i);
            _id.add(detFactura.getIdDetFactura()); }
        Enumeration<Long> enumeration = _id.elements();
        while (enumeration.hasMoreElements()) {
            Long codigo = (Long) enumeration.nextElement();
            try {
                detFacturaLocal.borrar(codigo);
            } catch (Exception e) {
                // TODO: handle exception
                e.printStackTrace(); } } }
    List<DetFacturaPago> listaPago = facPagoLocal.consultarPorFactura(Long
        .parseLong(idFactura));
    if (listaPago.size() > 0) {
        _id = new Vector<Long>();
        for (int i = 0; i < listaPago.size(); i++) {
            DetFacturaPago pago = (DetFacturaPago) listaPago.get(i);
            _id.add(pago.getIdDetFactPago()); }
        Enumeration<Long> enumeration = _id.elements();
        while (enumeration.hasMoreElements()) {
            Long codigo = (Long) enumeration.nextElement();
            try {
                facPagoLocal.borrar(codigo);
            } catch (Exception e) {
                // TODO: handle exception
                e.printStackTrace(); } } }
    try {
        facturaLocal.borrar(Long.parseLong(idFactura));
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace(); }
    /** Setemos los valores de la Facura seleccionada **/
    FacesContext context = FacesContext.getCurrentInstance();
    String id = getValue(context, getCodigoCuenta());
    if (id == null || id.trim().equals("")) {
        return null; }
    limpiar();
    setIdFactura(id);
    List<Factura> lista = facturaLocal.consultar(Long.parseLong(idFactura));
    Factura factura = (Factura) lista.get(0);
    Cliente cliente = (Cliente) factura.getCliente();
    Usuario usuario = (Usuario) factura.getUsuario());

```

```

setNumeroFactura(factura.getNumeroFactura());
setRuc(cliente.getCedulaRucCliente());
setCliente(cliente.getNombreCliente() + " "
            + cliente.getApellidoCliente());
setTelefono(cliente.getTelefono1ClienteProveedor());
setDireccion(cliente.getDireccionClienteProveedor());
setNumeroCuenta(factura.getNumeroCuentaFactura());
setVendedor(usuario.getNombreUsuario());
/** Seteamos la Hora y Fecha **/
Calendar calendar = Calendar.getInstance(TimeZone.getTimeZone("GMT-5:00"));
String fecha[] = new String[6];
fecha[0] = String.valueOf(calendar.get(Calendar.YEAR));
fecha[1] = String.valueOf(calendar.get(Calendar.MONTH));
fecha[2] = String.valueOf(calendar.get(Calendar.DAY_OF_MONTH));
fecha[3] = String.valueOf(calendar.get(Calendar.HOUR_OF_DAY));
fecha[4] = String.valueOf(calendar.get(Calendar.MINUTE));
fecha[5] = String.valueOf(calendar.get(Calendar.SECOND));
for (int i = 0; i < fecha.length; i++) {
    if (fecha[i].trim().length() == 1) {
        fecha[i] = "0" + fecha[i]; } }
setFecha(fecha[0] + "-" + fecha[1] + "-" + fecha[2]);
setHora(fecha[3] + ":" + fecha[4]);
setSubTotal(factura.getSubtotalFactura().setScale(2,
            RoundingMode.HALF_UP) + "");
setDescuento(factura.getDescuentoFactura().setScale(2,
            RoundingMode.HALF_UP) + "");
setRecargo(factura.getRecargoFactura()
            .setScale(2, RoundingMode.HALF_UP) + "");
setIva(factura.getImpuestoFactura().setScale(2, RoundingMode.HALF_UP) + "");
setTotal(factura.getTotal().setScale(2, RoundingMode.HALF_UP) + "");

setBloqueoCuenta(true);
setIdCliente(cliente.getIdCliente() + "");
return "cuentas-fac"; }

public String seleccionarPorcentaje() {
    FacesContext context = FacesContext.getCurrentInstance();
    String id = getValue(context, getCodigoPorcentaje());
    if (id == null || id.trim().equals("")) {
        setIdPorcentaje(null);
        return "porcentaje-fac"; }

/** Factura **/

List<Factura> listaFact = facturaLocal.consultar(Long
            .parseLong(idFactura));
Factura factura = (Factura) listaFact.get(0);

/** Porcentaje **/

```

```

List<DescuentoRecargo> lista = porcentajeLocal.consultar(id);
DescuentoRecargo porcentaje = (DescuentoRecargo) lista.get(0);
if (porcentaje.getTipoDescRecg().trim()
    .equals(PorcentajeBean.DESCUENTO)) {
    if (factura.getDescuentoFactura().compareTo(new BigDecimal("0")) == 1) {
        setMensajeDatos("Error, solo se permite un Descuento por Venta");
        return "/fac/porcentajeFactura.xhtml";
    }
} else if (porcentaje.getTipoDescRecg().trim().equals(
    PorcentajeBean.RECARGO)) {
    if (factura.getRecargoFactura().compareTo(new BigDecimal("0")) == 1) {
        setMensajeDatos("Error, solo se permite un Recargo por Venta");
        return "/fac/porcentajeFactura.xhtml"; } }

BigDecimal _total = factura.getTotal();
BigDecimal _subtotal = factura.getSubtotalFactura();
BigDecimal _porcentaje = porcentaje.getPorcentajeDescRecg();
BigDecimal _cantidad = (_subtotal.multiply(_porcentaje)).divide(
    new BigDecimal("100"), 10, RoundingMode.HALF_UP);
if (porcentaje.getTipoDescRecg().trim()
    .equals(PorcentajeBean.DESCUENTO)) {
    _total = _total.subtract(_cantidad);
    factura.setDescuentoFactura(_cantidad);
    setDescuento(_cantidad.setScale(2, RoundingMode.HALF_UP) + "");
} else if (porcentaje.getTipoDescRecg().trim().equals(
    PorcentajeBean.RECARGO)) {
    _total = _total.add(_cantidad);
    factura.setRecargoFactura(_cantidad);
    setRecargo(_cantidad.setScale(2, RoundingMode.HALF_UP) + ""); }
setTotal(_total.setScale(2, RoundingMode.HALF_UP) + "");
factura.setTotal(_total);
facturaLocal.actualizar(factura);
setMensajeDatos("");
setMensajeProductos("");
return "porcentaje-fac"; }

public void seleccionarFormaPago() {
    System.out.println("idFormaPago: " + idFormaPago);
    System.out.println("cantidadPagoCliente: " + cantidadPagoCliente);
    Pattern pattern = Pattern.compile("^([0-9]{1,7}(\\.[0-9]{0,2})?)?$");
    Matcher matcher = pattern.matcher((String) cantidadPagoCliente);
    if (!matcher.find()) {
        setMensajeDatos(" :: Error, ingrese un cantidad válida ::.");
        return; }

List<DetFacturaPago> lista = facPagoLocal.consultar(Long.parseLong(idFormaPago));
DetFacturaPago pago = (DetFacturaPago) lista.get(0);
pago.setCantidadDetPago(new BigDecimal(cantidadPagoCliente));

```

```

        facPagoLocal.actualizar(pago);
    /** Validar si existe un toma física pendiente en la bodega origen seleccionada **/
        List<TomaFisica> listaTomaFisicaOrigen =
tomaFisicaLocal.consultarTomaFisicaPorEstado(idBodega, TomaFisicaBean.TIPO_TOMA,
TomaFisicaBean.ESTADO_TOMA_FISICA_PENDIENTE);
        if(listaTomaFisicaOrigen.size() > 0){
            setMensajeDatos(Mensajes.ERROR_TOMA_FISICA_PEDIENTE_BODEGA);
                return "/fac/facturacion.xhtml"; }
            return null; }
public String grabarFormaPago() {
        if(idFacturalImprimir != null && idFacturalImprimir.trim().equals("")){
            setMensajeDatos(":: Error, no Existe una Factura Pendiente ::.");
                return "/fac/pagoFactura.xhtml"; }
            BigDecimal _total = new BigDecimal("0.00");

    /** Forma Pago **/
        List<DetFacturaPago> listaPago =
facPagoLocal.consultarPorFactura(Long.parseLong(idFactura));
        for (int i = 0; i < listaPago.size(); i++) {
            DetFacturaPago pago = (DetFacturaPago) listaPago.get(i);
                _total = _total.add(pago.getCantidadDetPago()); }

    /** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);
        if (factura.getTotal().compareTo(_total) == 1) {
            setMensajeDatos(":: Error, el valor a pagar no es Completado ::.");
                return "/fac/pagoFactura.xhtml"; }
            factura.setNumeroFactura(getNumeroFactura());
            factura.setFechaFactura(new Date());
            factura.setHoraFactura(new Date());
            factura.setEstadoFactura(ESTADO_REGISTRADO);
            facturaLocal.actualizar(factura);

    /** Detalle Factura **/
        String jpql = "Select d from DetFactura d "
            + "where d.estadoDetFactura = " + ESTADO_PENDIENTE + " "
            + "and d.factura.idFactura = " + idFactura + " ";
        List<DetFactura> listaDetalle = detFacturaLocal.query(jpql);
        for (int j = 0; j < listaDetalle.size(); j++) {
            DetFactura detFactura = (DetFactura) listaDetalle.get(j);
                detFactura.setEstadoDetFactura(ESTADO_REGISTRADO);
                detFacturaLocal.actualizar(detFactura); }

    /** Kardex **/
        HttpSession session = getHttpSession();
        String _id = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        List<Sucursal> listaSucursal = sucursalLocal.consultar(_id);

```

```

        Sucursal sucursal = (Sucursal) listaSucursal.get(0);
        List<Bodega> listaBodega = bodegaLocal.consultar(_id);
        Bodega bodega = (Bodega) listaBodega.get(0);
        Cliente cliente = (Cliente) factura.getCliente();
        egreso(Long.parseLong(idFactura), bodega, sucursal, cliente);
        setIdFacturaImprimir(idFactura);
        setTarget("blank");
        nuevaFactura();
        setBloqueoCuenta(false);
        setMensajeDatos("== Factura totalizada correctamente ==");
        return "/fac/pagoFactura.xhtml"; }

public void egreso(long id, Bodega bodega, Sucursal sucursal, Cliente cliente) {
    BigDecimal cantidadDetalle, costoDetalle, totalDetalle;
    BigDecimal cantidadStock, costoStock;
    BigDecimal cantidadExistenciaKardex, costoExistenciaKardex, totalExistenciaKardex;
    String jpql = "Select d from DetFactura d "
        + "where d.estadoDetFactura = " + ESTADO_REGISTRADO + " "
        + "and d.factura.idFactura = " + idFactura + " ";
    List<DetFactura> listaDetalle = detFacturaLocal.query(jpql);
    for (int i = 0; i < listaDetalle.size(); i++) {
        DetFactura detalle = (DetFactura)listaDetalle.get(i);
        /** Producto */
        Producto producto = (Producto) detalle.getProducto();
        /** Stock */
        List<StockProducto> listaStock = stockLocal.consultar(producto.getIdProducto(),
sucursal.getIdSucursal());
        StockProducto stock = (StockProducto) listaStock.get(0);
            cantidadStock = stock.getStockActualProducto();
            costoStock = stock.getCostoActualProducto();
            cantidadDetalle = detalle.getCantidadDetFactura();
            costoDetalle = stock.getCostoActualProducto();
            totalDetalle = cantidadDetalle.multiply(costoDetalle);
            cantidadExistenciaKardex = cantidadStock.subtract(cantidadDetalle);
            costoExistenciaKardex = costoStock;
            totalExistenciaKardex =
(cantidadExistenciaKardex.multiply(costoExistenciaKardex)).abs();
        /** Kardex */
        Kardex _kardex = new Kardex();
        _kardex.setProducto(producto);
        _kardex.setBodega(bodega);
        _kardex.setSucursal(sucursal);
        _kardex.setFechaKardex(new Date());
        _kardex.setHoraKardex(new Date());
        _kardex.setCodTransaccionKardex(id + "");
        _kardex.setTipoTransaccionKardex(KardexBean.TIPO_TRANSACCION_FACTURA);
    }
}

```

```

        _kardex.setCodClienteProvDocumento(cliente.getIdCliente() + "");
        _kardex.setDescripcionClienteProvDoc(cliente.getNombreCliente() + " " +
cliente.getApellidoCliente() );

        _kardex.setTipoMovimientoKardex(KardexBean.TIPO_MOVIMIENTO_EGRESO);
        _kardex.setCantidadKardex(cantidadDetalle);
        _kardex.setCostoKardex(costoDetalle);
        _kardex.setTotalKardex(totalDetalle);
        _kardex.setCantidadExistenciaKardex(cantidadExistenciaKardex);
        _kardex.setCostoExistenciaKardex(costoExistenciaKardex);
        _kardex.setTotalExistenciaKardex(totalExistenciaKardex);
        kardexLocal.crear(_kardex);
/** Actualizar el stock del Producto */
        stock.setCostoActualProducto(costoExistenciaKardex);
        stock.setStockActualProducto(cantidadExistenciaKardex);
        stockLocal.actualizar(stock); } }

    public void venderProducto() {
        System.out.println("Cantidad: " + cantidad);
/** Validamos que exista un Cliente **/
        if (idCliente == null || idCliente.trim().equals("")) {
            setMensajeDatos(":: Error, seleccione un Cliente ::.");
            return;}
/** Validamos que exista un producto para vender **/
        if (idProducto == null || idProducto.trim().equals("")) {
            setMensajeProductos(":: Error, seleccione un Producto ::.");
            return; }
/** Validamos que sea un valor válido **/
        Pattern pattern = Pattern.compile("^([0-9]{1,7}(\\.[0-9]{0,3})?)?$");
        Matcher matcher = pattern.matcher((String) cantidad);
        if (!matcher.find()) {
            setMensajeProductos(":: Error, ingrese una cantidad válida ::.");
            return; }
/** Validamos que la cantidad sea diferente de cero **/
        if (new BigDecimal(cantidad).compareTo(new BigDecimal("0.000")) == 0) {
            setMensajeProductos(":: Error, ingrese una cantidad diferente de ceo ::.");
            return; }

/** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);

/** Producto **/
        List<Producto> listaProducto = productoLocal.consultar(idProducto);
        Producto producto = (Producto) listaProducto.get(0);

/** Cliente **/
        List<Cliente> listaCliente = clienteLocal.consultar(Long .parseLong(idCliente));
        Cliente cliente = (Cliente) listaCliente.get(0);

/** Operaciones **/

```



```

        BigDecimal cantidad = new BigDecimal(getCantidad());
        BigDecimal unitario = new BigDecimal("0.00");
        BigDecimal total = new BigDecimal("0.00");
        if (cliente.getTipoPrecioCliente().equals(ClienteBean.PRECIO_A)) {
            unitario = producto.getPrecio1Producto();
        } else if (cliente.getTipoPrecioCliente().equals(ClienteBean.PRECIO_B)) {
            unitario = producto.getPrecio2Producto();
        } else if (cliente.getTipoPrecioCliente().equals(ClienteBean.PRECIO_C)) {
            unitario = producto.getPrecio3Producto(); }
        total = cantidad.multiply(unitario);
    /** Detalle Factura **/
        DetFactura detalle = new DetFactura();
        detalle.setFactura(factura);
        detalle.setProducto(producto);
        detalle.setNombreDetFactura(producto.getDescripcionProducto());
        detalle.setCantidadDetFactura(cantidad);
        detalle.setPrecioUnitarioDetFactura(unitario);
        detalle.setPrecioTotalDetFactura(total);
        detalle.setEstadoDetFactura(ESTADO_PENDIENTE);
        detFacturaLocal.crear(detalle);
    /** Operaciones para los totales **/
        BigDecimal neto = new BigDecimal("0.00");
        BigDecimal tax = new BigDecimal("0.00");
        BigDecimal ttl = new BigDecimal("0.00");
        Impuesto impuesto = (Impuesto) producto.getImpuesto();
        if (impuesto.getTipImpuesto().equals(ImpuestoBean.VAT)) {

            neto = (total.multiply(new BigDecimal("100")).divide(new BigDecimal("112"), 10,
RoundingMode.HALF_UP);
            tax = (total.multiply(new BigDecimal("12")).divide(new BigDecimal("112"), 10,
RoundingMode.HALF_UP);
            ttl = total;
        } else if (impuesto.getTipImpuesto().equals(ImpuestoBean.ADD)) {
            neto = total;
            tax = (total.multiply(new BigDecimal("12")).divide(new BigDecimal("100"), 10,
RoundingMode.HALF_UP);
            ttl = neto.add(tax); }
    /** Factura **/
        factura.setSubtotalFactura(factura.getSubtotalFactura().add(neto));
        factura.setImpuestoFactura(factura.getImpuestoFactura().add(tax));
        factura.setTotal(factura.getTotal().add(ttl));
        facturaLocal.actualizar(factura);
    /** Setemos los valores **/
        setSubTotal(factura.getSubtotalFactura().setScale(2,RoundingMode.HALF_UP) + "");
        setIva(factura.getImpuestoFactura().setScale(2, RoundingMode.HALF_UP) + "");

```

```

        setTotal(factura.getTotal().setScale(2, RoundingMode.HALF_UP) + "");
        setCantidad("0");
        setMensajeDatos("");
        setMensajeProductos("");
        setCodigo(idProducto);}

    public void seleccionarCliente() {
        HttpSession session = getSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);
        if(validar != null){
            return;}
        System.out.println("idCliente: " + idCliente);
    /** Cliente **/
        List<Cliente> lista = clienteLocal.consultar(Long.parseLong(idCliente));
        Cliente cliente = (Cliente) lista.get(0);
    /** Setemamos los datos del Cliente **/
        setCliente(cliente.getNombreCliente() + " " + cliente.getApellidoCliente());
        setTelefono(cliente.getTelefono1ClienteProveedor());
        setDireccion(cliente.getDireccionClienteProveedor());
    /** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);
        factura.setCliente(cliente);
        facturaLocal.actualizar(factura);
        setMensajeDatos("");
        setMensajeProductos(""); }

    public String seleccionarClienteTable() {
        FacesContext context = FacesContext.getCurrentInstance();
        String id = getValue(context, getCodigoCliente());
        if (id == null || id.trim().equals("")) {
            return "/fac/consultarClientes.xhtml"; }
        setIdCliente(id);
        System.out.println("idCliente: " + idCliente);
    /** Cliente **/
        List<Cliente> lista = clienteLocal.consultar(Long.parseLong(idCliente));
        Cliente cliente = (Cliente) lista.get(0);
    /** Setemamos los datos del Cliente **/
        setCliente(cliente.getNombreCliente() + " " + cliente.getApellidoCliente());
        setRuc(cliente.getCedulaRucCliente());
        setTelefono(cliente.getTelefono1ClienteProveedor());
        setDireccion(cliente.getDireccionClienteProveedor());
    /** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);
        factura.setCliente(cliente);

```

```

        facturaLocal.actualizar(factura);
        return "cliente-fac"; }
public List<Cliente> suggestionCliente(Object suggests) {
    String pref = (String) suggests;
    ArrayList<Cliente> result = new ArrayList<Cliente>();
    List<Cliente> lista = clienteLocal.consultar();
    Iterator<Cliente> iterator = lista.iterator();
    while (iterator.hasNext()) {
        Cliente cliente = (Cliente) iterator.next();
        String elem = cliente.getCedulaRucCliente();
        if ((elem != null && elem.toLowerCase().indexOf(pref.toLowerCase()) == 0)
            || "".equals(pref)) {
            result.add(cliente);}}
    return result;}
public void seleccionarProducto() {
/** Validar Inventarios **/
    HttpSession session = getSession();
    String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
    String validar = validarInventario(idBodega);
    if(validar != null){
        return;}
    System.out.println("idProducto: " + idProducto);
/** Producto **/
    List<Producto> lista = productoLocal.consultar(idProducto);
    Producto producto = (Producto) lista.get(0);
    /** Setemos los datos del Cliente **/
    setDescripcion(producto.getDescripcionProducto());
    setMensajeDatos("");
    setMensajeProductos(""); }
public String seleccionarProductoTable() {
    FacesContext context = FacesContext.getCurrentInstance();
    String id = getValue(context, getCodigoProducto());
    if (id == null || id.trim().equals("")) {
        setIdProducto(null);
        setCodigo(null);
        setDescripcion("");
        setMensajeDatos("");
        setMensajeProductos("");
        return "producto-fac"; }
    List<Producto> lista = productoLocal.consultar(id);
    Producto producto = (Producto) lista.get(0);
    setIdProducto(producto.getIdProducto());
    setCodigo(producto.getIdProducto());
    setDescripcion(producto.getDescripcionProducto());
    setMensajeDatos("");

```

```

        setMensajeProductos("");
        return "producto-fac"; }
    public List<Producto> suggestionProducto(Object suggests) {
        String pref = (String) suggests;
        ArrayList<Producto> result = new ArrayList<Producto>();
        List<Producto> lista = productoLocal.consultarParaVender();
        Iterator<Producto> iterator = lista.iterator();
        while (iterator.hasNext()) {
            Producto producto = (Producto) iterator.next();
            String elem = producto.getIdProducto();
            if ((elem != null && elem.toLowerCase().indexOf(pref.toLowerCase()) == 0)
                || "".equals(pref)) {
                result.add(producto); } }
        return result; }
    public String load() {
        nuevaFactura();
        return "fac-facturacion"; }
    public String cancel() {
/** Validar Iventario **/
        HttpSession session = getHttpSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);
        if(validar != null){
            return validar; }
/** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);
        factura.setEstadoFactura(ESTADO_CANCEL);
        facturaLocal.actualizar(factura);
/** Detalle Factura **/
        List<DetFactura> listaDetalle = detFacturaLocal
        .consultarPorFactura(Long.parseLong(idFactura));
        for (int j = 0; j < listaDetalle.size(); j++) {
            DetFactura detFactura = (DetFactura) listaDetalle.get(j);
            detFactura.setEstadoDetFactura(ESTADO_CANCEL);
            detFacturaLocal.actualizar(detFactura); }
        limpiar();
        nuevaFactura();
        setBloqueoCuenta(false);
        return "/fac/facturacion.xhtml"; }
    public String guardarCuenta() {
/** Validar Inventarios **/
        HttpSession session = getHttpSession();
        String idBodega = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String validar = validarInventario(idBodega);

```

```

        if(validar != null){
            return validar; }
/** Validamos que exista un Cliente **/
        if (idCliente == null || idCliente.trim().equals("")) {
            setMensajeDatos(":: Error, seleccione un Cliente ::");
            return "/fac/facturacion.xhtml"; }
/** Validamos el número de Cuenta **/
        if (getNumeroCuenta() == null || getNumeroCuenta().trim().equals("")) {
            setMensajeDatos(":: Error, el campo Cuenta no debe estar vacio ::");
            return "/fac/facturacion.xhtml"; }
        Pattern pattern = Pattern.compile("[^0-9]");
        Matcher matcher = pattern.matcher((String) getNumeroCuenta());
        if (matcher.find()) {
            setMensajeDatos(":: Error, ingrese solo números en la Cuenta ::");
            return "/fac/facturacion.xhtml"; }
        if (Integer.parseInt(getNumeroCuenta()) == 0) {
            setMensajeDatos(":: Error, ingrese solo números diferente de cero en la Cuenta
::");

            return "/fac/facturacion.xhtml"; }
        if (getNumeroCuenta().trim().length() > 3) {
            setMensajeDatos(":: Error, el campo Cuenta debe tener máximo 3 dígitos ::");
            return "/fac/facturacion.xhtml"; }
        if(!bloqueoCuenta){
            String id = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
            String jpql = "Select f from Factura f "
                + "where f.sucursal.idSucursal = " + id + " "
                + "and f.numeroCuentaFactura = " + getNumeroCuenta().trim() + " "
                + "and f.estadoFactura = " + ESTADO_PENDIENTE + """;
            List<Factura> lista = facturaLocal.query(jpql);
            if (lista.size() == 1) {
                setMensajeDatos(":: Error, el número de Cuenta está ocupado ::");
                return "/fac/facturacion.xhtml"; } }
        List<Factura> listaFac = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFac.get(0);
        factura.setNumeroCuentaFactura(getNumeroCuenta());
        facturaLocal.actualizar(factura);
        limpiar();
        nuevaFactura();
        setBloqueoCuenta(false);
        return "/fac/facturacion.xhtml"; }
        public void borrar() {
            FacesContext context = FacesContext.getCurrentInstance();
            String _id = getValue(context, getCodigoProducto());
            if (_id == null || _id.trim().equals("")) {
                return; }

```

```

        List<DetFactura> lista = detFacturaLocal.consultar(Long.parseLong(_id));
        DetFactura _detalle = (DetFactura) lista.get(0);
        _detalle.setEstadoDetFactura(ESTADO_ANULADO);
        detFacturaLocal.actualizar(_detalle);
    /** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);
    /** Producto **/
        Producto producto = (Producto) _detalle.getProducto();
    /** Operaciones **/
        BigDecimal total = _detalle.getPrecioTotalDetFactura();
    /** Operaciones para los totales **/
        BigDecimal neto = new BigDecimal("0.00");
        BigDecimal tax = new BigDecimal("0.00");
        BigDecimal ttl = new BigDecimal("0.00");
        Impuesto impuesto = (Impuesto) producto.getImpuesto();
        if (impuesto.getTipoImpuesto().equals(ImpuestoBean.VAT)) {
            neto = (total.multiply(new BigDecimal("100"))).divide(new BigDecimal("112"), 10,
RoundingMode.HALF_UP);
            tax = (total.multiply(new BigDecimal("12"))).divide(new BigDecimal("112"), 10,
RoundingMode.HALF_UP);
            ttl = total;
        } else if (impuesto.getTipoImpuesto().equals(ImpuestoBean.ADD)) {
            neto = total;
            tax = (total.multiply(new BigDecimal("12"))).divide(new BigDecimal("100"), 10,
RoundingMode.HALF_UP);
            ttl = neto.add(tax);}
    /** Factura **/
        factura.setSubtotalFactura(factura.getSubtotalFactura().subtract(neto));
        factura.setImpuestoFactura(factura.getImpuestoFactura().subtract(tax));
        factura.setTotal(factura.getTotal().subtract(ttl));
        facturaLocal.actualizar(factura);
    /** Setemos los valores **/
        setSubTotal(factura.getSubtotalFactura().setScale(2,RoundingMode.HALF_UP)+ "");
        setIva(factura.getImpuestoFactura().setScale(2, RoundingMode.HALF_UP) + "");
        setTotal(factura.getTotal().setScale(2, RoundingMode.HALF_UP) + "");
        setMensajeDatos("");
        setMensajeProductos("");
        setCodigo(idProducto); }
    public void nuevaFactura() {
    /** Limpiar **/
        limpiar();
    /** Usuario **/
        HttpSession session = getHttpSession();
        String idUsuario = (String) session.getAttribute(LoginBean.CODIGO);

```

```

String idSucursal = (String) session
    .getAttribute(LoginBean.ID_SUCURSAL);
List<Usuario> listaUsuarios = usuarioLocal.consultar(idUsuario);
Usuario usuario = (Usuario) listaUsuarios.get(0);
** Sucursal **/
List<Sucursal> listaSucursal = sucursalLocal.consultar(idSucursal);
Sucursal sucursal = (Sucursal) listaSucursal.get(0);
/** Validamos si existe un CC pendiente **/
CuadreCaja caja = new CuadreCaja();
String jpql = "Select c from CuadreCaja c "
    + "where c.sucursal.idSucursal = " + idSucursal + " "
    + "and c.estadoCuadreCaja = "
    + CuadreCajaBean.ESTADO_PENDIENTE + """;
List<CuadreCaja> listaCaja = cajaLocal.query(jpql);
if (listaCaja.size() == 0) {
/** Cuadre Caja **/
    caja.setUsuario(usuario);
    caja.setSucursal(sucursal);
    caja.setFechaCuadreCaja(new Date());
    caja.setHoraCuadreCaja(new Date());
    caja.setCantEstCuadreCaja(new BigDecimal("0.00"));
    caja.setTotalCuadreCaja(new BigDecimal("0.00"));
    caja.setEstadoCuadreCaja(CuadreCajaBean.ESTADO_PENDIENTE);
    caja.setEstadoUsusarioCCaja(CuadreCajaBean.ESTADO_PENDIENTE);
    cajaLocal.crear(caja);
} else {
    caja = (CuadreCaja) listaCaja.get(0); }
/** Cliente **/
List<Cliente> listaCliente = clienteLocal.consultar();
Cliente cliente = (Cliente) listaCliente.get(0);
/** Factura **/
Factura factura = new Factura();
factura.setUsuario(usuario);
factura.setSucursal(sucursal);
factura.setCuadreCaja(caja);
factura.setCliente(cliente);
factura.setSubtotalFactura(new BigDecimal("0"));
factura.setImpuestoFactura(new BigDecimal("0"));
factura.setDescuentoFactura(new BigDecimal("0"));
factura.setRecargoFactura(new BigDecimal("0"));
factura.setTotal(new BigDecimal("0"));
factura.setEstadoFactura(ESTADO_PENDIENTE);
facturaLocal.crear(factura);
/** Pago Factura **/
List<FormaPago> listaFPago = formaPagoLocal.consultar();

```

```

        for (int i = 0; i < listaFPago.size(); i++) {
            FormaPago formaPago = (FormaPago) listaFPago.get(i);
            DetFacturaPago pago = new DetFacturaPago();
            pago.setFactura(factura);
            pago.setFormaPago(formaPago);
            pago.setCantidadDetPago(new BigDecimal("0.000"));
            pago.setFechaDetPago(new Date());
            facPagoLocal.crear(pago);
        }
    /** Seteamos el id de la Factura **/
        setIdFactura(factura.getIdFactura() + "");
        setNumeroFactura("001 - 001 - " + idFactura);
        setNumeroCuenta("000");
        factura.setNumeroFactura(getNumeroFactura());
        factura.setNumeroCuentaFactura(getNumeroCuenta());
        factura.setFechaFactura(new Date());
        factura.setHoraFactura(new Date());
        facturaLocal.actualizar(factura);
    /** Seteamos la Hora y Fecha **/
        Calendar calendar = Calendar.getInstance(TimeZone
            .getTimeZone("GMT-5:00"));
        String fecha[] = new String[6];
        fecha[0] = String.valueOf(calendar.get(Calendar.YEAR));
        fecha[1] = String.valueOf(calendar.get(Calendar.MONTH));
        fecha[2] = String.valueOf(calendar.get(Calendar.DAY_OF_MONTH));
        fecha[3] = String.valueOf(calendar.get(Calendar.HOUR_OF_DAY));
        fecha[4] = String.valueOf(calendar.get(Calendar.MINUTE));
        fecha[5] = String.valueOf(calendar.get(Calendar.SECOND));
        for (int i = 0; i < fecha.length; i++) {
            if (fecha[i].trim().length() == 1) {
                fecha[i] = "0" + fecha[i]; } }
        setFecha(fecha[0] + " - " + fecha[1] + " - " + fecha[2]);
        setHora(fecha[3] + " : " + fecha[4]);
    /** Seteamos el vendedor **/
        setVendedor(usuario.getNombreUsuario()); }
    public FacClientesTable[] getTableClientes() {
        List<Cliente> lista = clienteLocal.consultar();
        if (lista.size() == 0) {
            FacClientesTable[] table = new FacClientesTable[1];
            table[0] = new FacClientesTable(null, "", "", "", "", "", "");
            return table; }
        FacClientesTable[] table = new FacClientesTable[lista.size()];
        for (int i = 0; i < lista.size(); i++) {
            Cliente cliente = (Cliente) lista.get(i);
            table[i] = new FacClientesTable(cliente.getIdCliente() + "",

```



```

        cliente.getCedulaRucCliente(), cliente.getNombreCliente(),
cliente.getApellidoCliente(), cliente .getNombreComercialClipro(), cliente
.getTelefono1ClienteProveedor(), cliente, getTipoPrecioCliente()); }
        return table; }
    public FacDetalleTable[] getTableDetalle() {
        if (idFactura == null || idFactura.trim().equals("")) {
            FacDetalleTable table[] = new FacDetalleTable[1];
            table[0] = new FacDetalleTable(null, "0.000", "", "0.00", "0.00");
            return table; }
        List<DetFactura> lista = detFacturaLocal.consultarPorEstado(Long
            .parseLong(idFactura), ESTADO_PENDIENTE);
        if (lista.size() == 0) {
            FacDetalleTable table[] = new FacDetalleTable[1];
            table[0] = new FacDetalleTable(null, "0.000", "", "0.00", "0.00");
            return table; }
        FacDetalleTable table[] = new FacDetalleTable[lista.size()];
        for (int i = 0; i < table.length; i++) {
            DetFactura detalle = (DetFactura) lista.get(i);
            table[i] = new FacDetalleTable(detalle.getIdDetFactura() + "",
                detalle.getCantidadDetFactura().setScale(3,
                    RoundingMode.HALF_UP)
                    + "", detalle.getNombreDetFactura(), detalle
                .getPrecioUnitarioDetFactura().setScale(2,
                    RoundingMode.HALF_UP) + "", detalle.getPrecioTotalDetFactura().setScale(2,
RoundingMode.HALF_UP)+ "");
            }
        return table; }
    public FacProductoTable[] getTableProducto() {
        List<Producto> lista = productoLocal.consultarParaVender();
        if (lista.size() == 0) {
            FacProductoTable table[] = new FacProductoTable[1];
            table[0] = new FacProductoTable(null, "", "", "0.00", "0.00",
                "0.00", "0.000");
            return table; }
    /** Usuario **/
        HttpSession session = getHttpSession();
        String id = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        FacProductoTable table[] = new FacProductoTable[lista.size()];
        for (int i = 0; i < table.length; i++) {
    /** Producto **/
        Producto producto = (Producto) lista.get(i);
    /** Stock **/
        List<StockProducto> listaStock = stockLocal.consultar(producto
            .getIdProducto(), id);
        StockProducto stock = (StockProducto) listaStock.get(0);
    /** Impuesto **/

```

```

Impuesto impuesto = (Impuesto) producto.getImpuesto();
table[i] = new FacProductoTable(producto.getIdProducto(), producto
.getDescripcionProducto(), impuesto.getNombreImpuesto(),
producto.getPrecio1Producto().setScale(2,
RoundingMode.HALF_UP) + "", producto.getPrecio2Producto().setScale(2,
RoundingMode.HALF_UP) + "", producto.getPrecio3Producto().setScale(2,
RoundingMode.HALF_UP) + "", stock.getStockActualProducto().setScale(3,
RoundingMode.HALF_UP) + ""); }

return table; }

public FacPagoTable[] getTablePago() {
    if (idFactura == null || idFactura.trim().equals("")) {
        FacPagoTable[] pagoTable = new FacPagoTable[1];
        pagoTable[0] = new FacPagoTable(null, "", "0.00");
        return pagoTable; }

    List<DetFacturaPago> lista = facPagoLocal.consultarPorFactura(Long
.parseLong(idFactura));
    if (lista.size() == 0) {
        FacPagoTable[] pagoTable = new FacPagoTable[1];
        pagoTable[0] = new FacPagoTable(null, "", "0.00");
        return pagoTable; }

    FacPagoTable[] pagoTable = new FacPagoTable[lista.size()];
    for (int i = 0; i < pagoTable.length; i++) {
        DetFacturaPago pago = (DetFacturaPago) lista.get(i);
        FormaPago formaPago = (FormaPago) pago.getFormaPago();
        pagoTable[i] = new FacPagoTable(pago.getIdDetFactPago() + "",
        formaPago.getDescripcionFormaPago(), pago
        .getCantidadDetPago().setScale(2,
        RoundingMode.HALF_UP) + ""); }

    return pagoTable; }

public FacPorcentajeTable[] getTablePorcentaje() {
    List<DescuentoRecargo> lista = porcentajeLocal.consultar();
    if (lista.size() == 0) {
        FacPorcentajeTable[] table = new FacPorcentajeTable[1];
        table[0] = new FacPorcentajeTable(null, "", "0.00", "");
        return table; }

    FacPorcentajeTable[] table = new FacPorcentajeTable[lista.size()];
    for (int i = 0; i < table.length; i++) {
        DescuentoRecargo porcentaje = (DescuentoRecargo) lista.get(i);
        if (porcentaje.getTipoDescRecg().trim().equals(
            PorcentajeBean.DESCUENTO)) {
            table[i] = new FacPorcentajeTable(porcentaje.getIdDesRecg(),
            porcentaje.getDescripcionDescRecg(), porcentaje
            .getPorcentajeDescRecg().setScale(2,
            RoundingMode.HALF_UP) + "", porcentaje.getTipoDescRecg()); } }

    for (int i = 0; i < table.length; i++) {

```

```

        DescuentoRecargo porcentaje = (DescuentoRecargo) lista.get(i);
        if (porcentaje.getTipoDescRecg().trim().equals(
            PorcentajeBean.RECARGO)) {
            table[i] = new FacPorcentajeTable(porcentaje.getIdDesRecg(),
                porcentaje.getDescripcionDescRecg(), porcentaje.
getPorcentajeDescRecg().setScale(2,
            RoundingMode.HALF_UP) + "", porcentaje.getTipoDescRecg()); } }
        return table; }

    public FacCuentasTable[] getTableCuentas() {
        HttpSession session = getHttpSession();
        String id = (String) session.getAttribute(LoginBean.ID_SUCURSAL);
        String jpql = "Select f from Factura f "
            + "where f.sucursal.idSucursal = " + id + " "
            + "and f.estadoFactura = " + ESTADO_PENDIENTE + " "
            + "and f.numeroCuentaFactura <> '000'";

        List<Factura> lista = facturaLocal.query(jpql);
        if (lista.size() == 0) {
            FacCuentasTable[] table = new FacCuentasTable[1];
            table[0] = new FacCuentasTable(null, "", "", "0.00");
            return table; }

        FacCuentasTable[] table = new FacCuentasTable[lista.size()];
        for (int i = 0; i < table.length; i++) {
            Factura factura = (Factura) lista.get(i);
            Usuario usuario = (Usuario) factura.getUsuario();
            table[i] = new FacCuentasTable(factura.getIdFactura() + "", factura
                .getNumeroCuentaFactura(), usuario.getNombreUsuario(),
factura.getTotal().setScale(2, RoundingMode.HALF_UP) + ""); }
        return table; }

    private boolean validateSession() {
        HttpSession s = getHttpSession();
        String codigo = (String) s.getAttribute(LoginBean.CODIGO);
        if (codigo != null && !codigo.trim().equals("")) {
            return true;
        } else {
            try {
                FacesContext.getCurrentInstance().getExternalContext()
                    .redirect(LoginBean.URL);
            } catch (IOException e) {
                e.printStackTrace(); }
            return false; } }

    private HttpSession getHttpSession() {
        return (HttpSession) FacesContext.getCurrentInstance()
            .getExternalContext().getSession(true); }

    private String getValue(FacesContext context, String name) {
        Map<String, String> map = context.getExternalContext()

```

```
        .getRequestParameterMap();
        return map.get(name); }
public String getTitle() {
    if (validateSession()) {
        return "FACTURA";
    } else {
        return null; } }
public String getTitleClientes() {
    if (validateSession()) {
        return "FACTURA - CLIENTES";
    } else {
        return null; } }
public String getTitleProductos() {
    if (validateSession()) {
        return "FACTURA - PRODUCTOS";
    } else {
        return null; } }
public String getTitlePago() {
    if (validateSession()) {
        return "FACTURA - PAGO";
    } else {
        return null; } }
public String getTitlePorcentaje() {
    if (validateSession()) {
        return "FACTURA - DSCTO/RECARGO";
    } else {
        return null; } }
public String getTitleCuentas() {
    if (validateSession()) {
        return "FACTURA - CONSULTA CUENTAS";
    } else {
        return null; } }
public String getIdFactura() {
    return idFactura; }
public void setIdFactura(String idFactura) {
    this.idFactura = idFactura; }
public String getIdCuadreCaja() {
    return idCuadreCaja; }
public void setIdCuadreCaja(String idCuadreCaja) {
    this.idCuadreCaja = idCuadreCaja; }
public String getIdCliente() {
    return idCliente; }
public void setIdCliente(String idCliente) {
    this.idCliente = idCliente;}
public String getNumeroFactura() {
```

```
        return numeroFactura; }
    public void setNumeroFactura(String numeroFactura) {
        this.numeroFactura = numeroFactura; }
    public String getNumeroCuenta() {
        return numeroCuenta; }
    public void setNumeroCuenta(String numeroCuenta) {
        this.numeroCuenta = numeroCuenta; }
    public String getRuc() {
        return ruc; }
    public void setRuc(String ruc) {
        this.ruc = ruc; }
    public String getCliente() {
        return cliente; }
    public void setCliente(String cliente) {
        this.cliente = cliente; }
    public String getTelefono() {
        return telefono; }
    public void setTelefono(String telefono) {
        this.telefono = telefono; }
    public String getDireccion() {
        return direccion; }
    public void setDireccion(String direccion) {
        this.direccion = direccion; }
    public String getFecha() {
        return fecha; }
    public void setFecha(String fecha) {
        this.fecha = fecha; }
    public String getHora() {
        return hora; }
    public void setHora(String hora) {
        this.hora = hora; }
    public String getVendedor() {
        return vendedor; }
    public void setVendedor(String vendedor) {
        this.vendedor = vendedor; }
    public String getMensajeDatos() {
        return mensajeDatos; }
    public void setMensajeDatos(String mensajeDatos) {
        this.mensajeDatos = mensajeDatos; }
    public String getCodigo() {
        return codigo; }
    public void setCodigo(String codigo) {
        this.codigo = codigo; }
    public String getCantidad() {
        return cantidad; }
```

```
public void setCantidad(String cantidad) {
    this.cantidad = cantidad; }
public String getDescripcion() {
    return descripcion; }
public void setDescripcion(String descripcion) {
    this.descripcion = descripcion; }
public String getMensajeProductos() {
    return mensajeProductos; }
public void setMensajeProductos(String mensajeProductos) {
    this.mensajeProductos = mensajeProductos; }
public String getSubTotal() {
    return subTotal; }
public void setSubTotal(String subTotal) {
    this.subTotal = subTotal; }
public String getDescuento() {
    return descuento; }
public void setDescuento(String descuento) {
    this.descuento = descuento; }
public String getRecargo() {
    return recargo; }
public void setRecargo(String recargo) {
    this.recargo = recargo; }
public String getIva() {
    return iva; }
public void setIva(String iva) {
    this.iva = iva; }
public String getTotal() {
    return total; }
public void setTotal(String total) {
    this.total = total;}
public String getCodigoCliente() {
    return "codigoCliente"; }
public void setIdProducto(String idProducto) {
    this.idProducto = idProducto;}
public String getIdProducto() {
    return idProducto; }
public String getCodigoProducto() {
    return "codigoProducto"; }
public void setTotalPagar(String totalPagar) {
    this.totalPagar = totalPagar; }
public String getTotalPagar() {
    return totalPagar; }
public void setSaldo(String saldo) {
    this.saldo = saldo; }
public String getSaldo() {
```

```
        return saldo; }
    public void setCambio(String cambio) {
        this.cambio = cambio; }
    public String getCambio() {
        return cambio; }
    public void setTotalPagoCliente(String totalPagoCliente) {
        this.totalPagoCliente = totalPagoCliente; }
    public String getTotalPagoCliente() {
        return totalPagoCliente; }
    public void setIdFormaPago(String idFormaPago) {
        this.idFormaPago = idFormaPago; }
    public String getIdFormaPago() {
        return idFormaPago; }
    public void setCantidadPagoCliente(String cantidadPagoCliente) {
        this.cantidadPagoCliente = cantidadPagoCliente; }
    public String getCantidadPagoCliente() {
        return cantidadPagoCliente; }
    public void setBloqueoCuenta(boolean bloqueoCuenta) {
        this.bloqueoCuenta = bloqueoCuenta; }
    public boolean isBloqueoCuenta() {
        return bloqueoCuenta; }
    public void setIdPorcentaje(String idPorcentaje) {
        this.idPorcentaje = idPorcentaje; }
    public String getIdPorcentaje() {
        return idPorcentaje; }
    public String getCodigoPorcentaje() {
        return "codigoPorcentaje"; }
    public String getCodigoCuenta() {
        return "codigoCuenta"; }
    public void setIdFacturalImprimir(String idFacturalImprimir) {
        this.idFacturalImprimir = idFacturalImprimir; }
    public String getIdFacturalImprimir() {
        return idFacturalImprimir; }
    public void setTarget(String target) {
        this.target = target; }
    public String getTarget() {
        return target; }
```

5.1.2. LÓGICA DE NEGOCIOS

EJB

Session Bean

“Los beans de sesión o Session Bean representan sesiones interactivas con uno o más clientes. Los bean de sesión pueden mantener un estado, pero sólo durante el tiempo que el cliente interactúa con el bean. Esto significa que los beans de sesión no almacenan sus datos en una base de datos después de que el cliente termine el proceso. Por ello se suele decir que los beans de sesión no son persistentes.”⁹⁶

Código correspondiente a los Session Beans del proceso de Facturación:

```
package com.js.sessionbeans.factura;
@Stateless
public class FacturaImpl implements IFacturaLocal, IFacturaRemote {
    @PersistenceContext(unitName = "TesisEJB")
    private EntityManager em;
    private Query query;
    @Override    public void actualizar(Factura factura) {
        em.merge(factura); }
    @Override    public void borrar(long id) throws Exception {
        Factura factura = em.find(Factura.class, id);
        if (factura != null) {
            try {
                em.remove(factura);
            } catch (Exception e) {
                throw new Exception(e.getCause()); } } }
    @SuppressWarnings("unchecked")
    @Override    public List<Factura> consultar() {
        query = em.createQuery("Select f from Factura f");
        return (List<Factura>) query.getResultList(); }
    @SuppressWarnings("unchecked")
    @Override    public List<Factura> consultar(long id) {
        query = em
            .createQuery("Select f from Factura f where f.idFactura = :id");
        query.setParameter("id", id);
        return (List<Factura>) query.getResultList(); }
```

⁹⁶ <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm>


```

@SuppressWarnings("unchecked")
@Override      public List<Factura> consultarPorCliente(String apellido) {
    query = em.createQuery("Select f from Factura f "
        + "where f.cliente.apellidoCliente LIKE %:apellido%");
    query.setParameter("apellido", apellido);
    return (List<Factura>) query.getResultList();}

@SuppressWarnings("unchecked")
@Override      public List<Factura> consultarPorEstado(String estado) {
    query = em.createQuery("Select f from Factura f "
        + "where f.estadoFactura = :estado");
    query.setParameter("estado", estado);
    return (List<Factura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<Factura> consultarPorRuc(String ruc) {
    query = em.createQuery("Select f from Factura f "
        + "where f.cliente.cedulaRucCliente = :ruc");
    query.setParameter("ruc", ruc);
    return (List<Factura>) query.getResultList(); }

@Override      public void crear(Factura factura) {
    em.persist(factura); }

@SuppressWarnings("unchecked")
@Override      public List<Factura> nativeQuery(String nativeQuery) {
    query = em.createNativeQuery(nativeQuery);
    return (List<Factura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<Factura> query(String jpql) {
    query = em.createQuery(jpql);
    return (List<Factura>) query.getResultList(); } }

```

```
package com.js.sessionbeans.factura;
```

```
@Stateless
```

```
public class DetFacturaImpl implements IDetFacturaLocal, IDetFacturaRemote {
```

```
    @PersistenceContext(unitName = "TesisEJB")
```

```
    private EntityManager em;
```

```
    private Query query;
```

```
    @Override      public void actualizar(DetFactura detFactura) {
        em.merge(detFactura); }

```

```
    @Override      public void borrar(long id) throws Exception {
        DetFactura detFactura = em.find(DetFactura.class, id);
        if (detFactura != null) {
            try {
                em.remove(detFactura);
            } catch (Exception e) {
                throw new Exception(e.getCause()); } } }

```

```
    @SuppressWarnings("unchecked")
```

```
    @Override      public List<DetFactura> consultar() {
```

```

        query = em.createQuery("Select d from DetFactura d");
        return (List<DetFactura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<DetFactura> consultar(long id) {
        query = em.createQuery("Select d from DetFactura d "
                + "where d.idDetFactura = :id");
        query.setParameter("id", id);
        return (List<DetFactura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<DetFactura> consultarPorEstado(long idFactura, String
estado) {
        query = em.createQuery("Select d from DetFactura d "
                + "where d.factura.idFactura = :idFactura "
                + "and d.estadoDetFactura = :estado");
        query.setParameter("idFactura", idFactura);
        query.setParameter("estado", estado);
        return (List<DetFactura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<DetFactura> consultarPorFactura(long idFactura) {
        query = em.createQuery("Select d from DetFactura d "
                + "where d.factura.idFactura = :idFactura");
        query.setParameter("idFactura", idFactura);
        return (List<DetFactura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<DetFactura> consultarPorProducto(String idProducto) {
        query = em.createQuery("Select d from DetFactura d "
                + "where d.producto.idProducto = :idProducto");
        query.setParameter("idProducto", idProducto);
        return (List<DetFactura>) query.getResultList(); }

@Override      public void crear(DetFactura detFactura) {
        em.persist(detFactura); }

@SuppressWarnings("unchecked")
@Override      public List<DetFactura> nativeQuery(String nativeQuery) {
        query = em.createNativeQuery(nativeQuery);
        return (List<DetFactura>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<DetFactura> query(String jpql) {
        query = em.createQuery(jpql);
        return (List<DetFactura>) query.getResultList(); }
}

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

```

```

import com.js.entitybeans.DetFacturaPago;

@Stateless
public class DetFacPagoImpl implements IDetFacPagoLocal, IDetFacPagoRemote {
    @PersistenceContext(unitName = "TesisEJB")
    private EntityManager em;
    private Query query;
    @Override public void actualizar(DetFacturaPago detFacturaPago) {
        em.merge(detFacturaPago);
    }
    @Override public void borrar(long id) throws Exception {
        DetFacturaPago detFacturaPago = em.find(DetFacturaPago.class, id);
        if (detFacturaPago != null) {
            try {
                em.remove(detFacturaPago);
            } catch (Exception e) {
                throw new Exception(e.getCause()); } } }
    @SuppressWarnings("unchecked")
    @Override public List<DetFacturaPago> consultar() {
        query = em.createQuery("Select d from DetFacturaPago d");
        return (List<DetFacturaPago>) query.getResultList(); }
    @SuppressWarnings("unchecked")
    @Override public List<DetFacturaPago> consultar(long id) {
        query = em.createQuery("Select d from DetFacturaPago d "
            + "where d.idDetFactPago = :id");
        query.setParameter("id", id);
        return (List<DetFacturaPago>) query.getResultList(); }
    @SuppressWarnings("unchecked")
    @Override
    public List<DetFacturaPago> consultar(long idFactura, String idFormaPago) {
        query = em.createQuery("Select d from DetFormaPago d "
            + "where d.factura.idFactura = :idFactura "
            + "and d.formaPago.idFormaPago = :idFormaPago");
        query.setParameter("idFactura", idFactura);
        query.setParameter("idFormaPago", idFormaPago);
        return (List<DetFacturaPago>) query.getResultList();
    }
    @SuppressWarnings("unchecked")
    @Override public List<DetFacturaPago> consultarPorFactura(long idFactura) {
        query = em.createQuery("Select d from DetFacturaPago d "
            + "where d.factura.idFactura = :idFactura");
        query.setParameter("idFactura", idFactura);
        return (List<DetFacturaPago>) query.getResultList();
    }
    @Override public void crear(DetFacturaPago detFacturaPago) {
        em.persist(detFacturaPago);
    }
    @SuppressWarnings("unchecked")

```

```

@Override      public List<DetFacturaPago> nativeQuery(String nativeQuery) {
                query = em.createNamedQuery(nativeQuery);
                return (List<DetFacturaPago>) query.getResultList(); }

@SuppressWarnings("unchecked")
@Override      public List<DetFacturaPago> query(String jpql) {
                query = em.createQuery(jpql);
                return (List<DetFacturaPago>) query.getResultList();}

```

5.1.3. GESTOR DE COMUNICACIÓN

EJB

Entity Bean

“Los beans de entidad o Entity Beans se corresponden con datos en un almacenamiento persistente (base de datos, sistema de ficheros, etc.). Las variables de instancia del bean representan los datos en las columnas de la base de datos.

El contenedor debe sincronizar las variables de instancia del bean con la base de datos. Los beans de entidad se diferencian de los beans de sesión en que las variables de instancia se almacenan de forma persistente.”⁹⁷

Código correspondiente a la comunicación mediante entitybeans del proceso de Facturación:

```

package com.js.entitybeans;

@Entity
public class Factura implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @SequenceGenerator(name = "FACTURA_IDFACTURA_GENERATOR", sequenceName =
"S_FACTURA", allocationSize = 1)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"FACTURA_IDFACTURA_GENERATOR")
    @Column(name = "ID_FACTURA")
    private long idFactura;
    @Column(name = "DESCUENTO_FACTURA")
    private BigDecimal descuentoFactura;

```

⁹⁷ <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm>

```

@Column(name = "ESTADO_FACTURA")
private String estadoFactura;
@Temporal(TemporalType.DATE)
@Column(name = "FECHA_FACTURA")
private Date fechaFactura;
@Temporal(TemporalType.DATE)
@Column(name = "HORA_FACTURA")
private Date horaFactura;
@Column(name = "IMPUESTO_FACTURA")
private BigDecimal impuestoFactura;
@Column(name = "NUMERO_FACTURA")
private String numeroFactura;
@Column(name = "NUMERO_CUENTA_FACTURA")
private String numeroCuentaFactura;
@Column(name = "RECARGO_FACTURA")
private BigDecimal recargoFactura;
@Column(name = "SUBTOTAL_FACTURA")
private BigDecimal subtotalFactura;
@Column(name = "TOTAL")
private BigDecimal total;
@OneToMany(mappedBy = "factura")
private Set<DetFactura> detFacturas;
@OneToMany(mappedBy = "factura")
private Set<DetFacturaPago> detFacturaPagos;
@ManyToOne
@JoinColumn(name = "ID_CUADRE_CAJA")
private CuadreCaja cuadroCaja;
@ManyToOne
@JoinColumn(name = "ID_SUCURSAL")
private Sucursal sucursal;
@ManyToOne
@JoinColumn(name = "ID_USUARIO")
private Usuario usuario;
@ManyToOne
@JoinColumn(name = "ID_CLIENTE")
private Cliente cliente;
public Factura() { }
public long getIdFactura() {
    return this.idFactura; }
public void setIdFactura(long idFactura) {
    this.idFactura = idFactura; }
public BigDecimal getDescuentoFactura() {
    return this.descuentoFactura; }
public void setDescuentoFactura(BigDecimal descuentoFactura) {
    this.descuentoFactura = descuentoFactura; }

```

```
public String getEstadoFactura() {
    return this.estadoFactura; }
public void setEstadoFactura(String estadoFactura) {
    this.estadoFactura = estadoFactura; }
public Date getFechaFactura() {
    return this.fechaFactura; }
public void setFechaFactura(Date fechaFactura) {
    this.fechaFactura = fechaFactura;}
public Date getHoraFactura() {
    return this.horaFactura;}
public void setHoraFactura(Date horaFactura) {
    this.horaFactura = horaFactura; }
public BigDecimal getImpuestoFactura() {
    return this.impuestoFactura; }
public void setImpuestoFactura(BigDecimal impuestoFactura) {
    this.impuestoFactura = impuestoFactura;}
public String getNumeroFactura() {
    return this.numeroFactura; }
public void setNumeroFactura(String numeroFactura) {
    this.numeroFactura = numeroFactura; }
public BigDecimal getRecargoFactura() {
    return this.recargoFactura; }
public void setRecargoFactura(BigDecimal recargoFactura) {
    this.recargoFactura = recargoFactura; }
public BigDecimal getSubtotalFactura() {
    return this.subtotalFactura; }
public void setSubtotalFactura(BigDecimal subtotalFactura) {
    this.subtotalFactura = subtotalFactura; }
public BigDecimal getTotal() {
    return this.total; }
public void setTotal(BigDecimal total) {
    this.total = total; }
public Set<DetFactura> getDetFacturas() {
    return this.detFacturas; }
public void setDetFacturas(Set<DetFactura> detFacturas) {
    this.detFacturas = detFacturas; }
public Set<DetFacturaPago> getDetFacturaPagos() {
    return this.detFacturaPagos; }
public void setDetFacturaPagos(Set<DetFacturaPago> detFacturaPagos) {
    this.detFacturaPagos = detFacturaPagos; }
public CuadreCaja getCuadreCaja() {
    return this.cuadreCaja; }
public void setCuadreCaja(CuadreCaja cuadreCaja) {
    this.cuadreCaja = cuadreCaja; }
public Sucursal getSucursal() {
```

```

        return this.sucursal; }
    public void setSucursal(Sucursal sucursal) {
        this.sucursal = sucursal; }
    public Usuario getUsuario() {
        return this.usuario; }
    public void setUsuario(Usuario usuario) {
        this.usuario = usuario; }
    public Cliente getCliente() {
        return this.cliente;}
    public void setCliente(Cliente cliente) {
        this.cliente = cliente; }
    public void setNumeroCuentaFactura(String numeroCuentaFactura) {
        this.numeroCuentaFactura = numeroCuentaFactura; }
    public String getNumeroCuentaFactura() {
        return numeroCuentaFactura; } }

```

```

package com.js.entitybeans;
@Entity
@Table(name = "DET_FACTURA")
public class DetFactura implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @SequenceGenerator(name = "DET_FACTURA_IDDETFACTURA_GENERATOR",
sequenceName = "S_DET_FACTURA", allocationSize = 1)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"DET_FACTURA_IDDETFACTURA_GENERATOR")
    @Column(name = "ID_DET_FACTURA")
    private long idDetFactura;
    @Column(name = "CANTIDAD_DET_FACTURA")
    private BigDecimal cantidadDetFactura;
    @Column(name = "ESTADO_DET_FACTURA")
    private String estadoDetFactura;
    @Column(name = "NOMBRE_DET_FACTURA")
    private String nombreDetFactura;
    @Column(name = "PRECIO_TOTAL_DET_FACTURA")
    private BigDecimal precioTotalDetFactura;
    @Column(name = "PRECIO_UNITARIO_DET_FACTURA")
    private BigDecimal precioUnitarioDetFactura;
    @ManyToOne
    @JoinColumn(name = "ID_FACTURA")
    private Factura factura;
    @ManyToOne
    @JoinColumn(name = "ID_PRODUCTO")
    private Producto producto;
    public DetFactura() { }
    public long getIdDetFactura() {

```

```

        return this.idDetFactura; }
    public void setIdDetFactura(long idDetFactura) {
        this.idDetFactura = idDetFactura; }
    public BigDecimal getCantidadDetFactura() {
        return this.cantidadDetFactura; }
    public void setCantidadDetFactura(BigDecimal cantidadDetFactura) {
        this.cantidadDetFactura = cantidadDetFactura; }
    public String getNombreDetFactura() {
        return this.nombreDetFactura; }
    public void setNombreDetFactura(String nombreDetFactura) {
        this.nombreDetFactura = nombreDetFactura; }
    public BigDecimal getPrecioTotalDetFactura() {
        return this.precioTotalDetFactura; }
    public void setPrecioTotalDetFactura(BigDecimal precioTotalDetFactura) {
        this.precioTotalDetFactura = precioTotalDetFactura; }
    public BigDecimal getPrecioUnitarioDetFactura() {
        return this.precioUnitarioDetFactura; }
    public void setPrecioUnitarioDetFactura(BigDecimal precioUnitarioDetFactura) {
        this.precioUnitarioDetFactura = precioUnitarioDetFactura; }
    public Factura getFactura() {
        return this.factura; }
    public void setFactura(Factura factura) {
        this.factura = factura; }
    public Producto getProducto() {
        return this.producto; }
    public void setProducto(Producto producto) {
        this.producto = producto;}
    public void setEstadoDetFactura(String estadoDetFactura) {
        this.estadoDetFactura = estadoDetFactura;}
    public String getEstadoDetFactura() {
        return estadoDetFactura;} }

```

```

package com.js.entitybeans;
@Entity
@Table(name = "DET_FACTURA_PAGO")
public class DetFacturaPago implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @SequenceGenerator(name =
"DET_FACTURA_PAGO_IDDETFACTPAGO_GENERATOR", sequenceName =
"S_DET_FACTURA_PAGO", allocationSize = 1)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"DET_FACTURA_PAGO_IDDETFACTPAGO_GENERATOR")
    @Column(name = "ID_DET_FACT_PAGO")
    private long idDetFactPago;
    @Column(name = "CANTIDAD_DET_PAGO")

```



```

private BigDecimal cantidadDetPago;
@Temporal(TemporalType.DATE)
@Column(name = "FECHA_DET_PAGO")
private Date fechaDetPago;
// bi-directional many-to-one association to Factura
@ManyToOne
@JoinColumn(name = "ID_FACTURA")
private Factura factura;
// bi-directional many-to-one association to FormaPago
@ManyToOne
@JoinColumn(name = "ID_FORMA_PAGO")
private FormaPago formaPago;
public DetFacturaPago() { }
public long getIdDetFactPago() {
    return this.idDetFactPago; }
public void setIdDetFactPago(long idDetFactPago) {
    this.idDetFactPago = idDetFactPago; }
public BigDecimal getCantidadDetPago() {
    return this.cantidadDetPago; }
public void setCantidadDetPago(BigDecimal cantidadDetPago) {
    this.cantidadDetPago = cantidadDetPago; }
public Date getFechaDetPago() {
    return this.fechaDetPago; }
public void setFechaDetPago(Date fechaDetPago) {
    this.fechaDetPago = fechaDetPago; }
public Factura getFactura() {
    return this.factura; }
public void setFactura(Factura factura) {
    this.factura = factura; }
public FormaPago getFormaPago() {
    return this.formaPago; }
public void setFormaPago(FormaPago formaPago) {
    this.formaPago = formaPago; } }

```

5.1.4. CONTROLADOR

El Controlador corresponde a las interfaces creadas juntamente con el Session Bean:

“Local: las interfaces remotas también son muy útiles si se desea tener el Contenedor Weben un servidor y el Contenedor EJB en otro servidor dentro de la misma intranet

Remote: las interfaces locales son ideales para servidores Java EE que albergan ambos Contenedores Web y EJB en la misma aplicación, las interfaces locales son muy rápidas de ejecutar porque no utilizan RMI ni serialización.”⁹⁸

Utilizamos el patrón de Diseño Fachada, porque para ejecutar un EJB no se instancia el Session Bean sino sus Interfaces mediante anotaciones.

Código correspondiente al controlador del proceso de facturación:

```

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Local;
import com.js.entitybeans.Factura;
@Local
public interface IFacturaLocal {
    public void actualizar(Factura factura);
    public void borrar(long id) throws Exception;
    public List<Factura> consultar();
    public List<Factura> consultar(long id);
    public List<Factura> consultarPorEstado(String estado);
    public List<Factura> consultarPorRuc(String ruc);
    public List<Factura> consultarPorCliente(String apellido);
    public List<Factura> query(String jpql);
    public List<Factura> nativeQuery(String nativeQuery);
    public void crear(Factura factura); }

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Remote;
import com.js.entitybeans.Factura;
@Remote
public interface IFacturaRemote {
    public void actualizar(Factura factura);
    public void borrar(long id) throws Exception;
    public List<Factura> consultar();
    public List<Factura> consultar(long id);
    public List<Factura> consultarPorEstado(String estado);
    public List<Factura> consultarPorRuc(String ruc);
    public List<Factura> consultarPorCliente(String apellido);

```

⁹⁸<http://www.javadabbadoo.org/cursos/infosintesis.net/javaeeav/sessionBean/stateless/convertidor/paso02/operativaCrearStatelessSessionBean.html>

```

public List<Factura> query(String jpql);
public List<Factura> nativeQuery(String nativeQuery);
public void crear(Factura factura); }

```

```

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Local;
import com.js.entitybeans.DetFactura;
@Local
public interface IDetFacturaLocal {
    public void actualizar(DetFactura detFactura);
    public void borrar(long id) throws Exception;
    public List<DetFactura> consultar();
    public List<DetFactura> consultar(long id);
    public List<DetFactura> consultarPorEstado(long idFactura, String estado);
    public List<DetFactura> consultarPorFactura(long idFactura);
    public List<DetFactura> consultarPorProducto(String idProducto);
    public List<DetFactura> query(String jpql);
    public List<DetFactura> nativeQuery(String nativeQuery);
    public void crear(DetFactura detFactura); }

```

```

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Remote;
import com.js.entitybeans.DetFactura;
@Remote
public interface IDetFacturaRemote {
    public void actualizar(DetFactura detFactura);
    public void borrar(long id) throws Exception;
    public List<DetFactura> consultar();
    public List<DetFactura> consultar(long id);
    public List<DetFactura> consultarPorEstado(long idFactura, String estado);
    public List<DetFactura> consultarPorFactura(long idFactura);
    public List<DetFactura> consultarPorProducto(String idProducto);
    public List<DetFactura> query(String jpql);
    public List<DetFactura> nativeQuery(String nativeQuery);
    public void crear(DetFactura detFactura); }

```

```

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Local;
import com.js.entitybeans.DetFacturaPago;
@Local
public interface IDetFacPagoLocal {
    public void actualizar(DetFacturaPago detFacturaPago);
    public void borrar(long id) throws Exception;
    public List<DetFacturaPago> consultar();
}

```

```

public List<DetFacturaPago> consultar(long id);
public List<DetFacturaPago> consultar(long idFactura, String idFormaPago);
public List<DetFacturaPago> consultarPorFactura(long idFactura);
public List<DetFacturaPago> nativeQuery(String nativeQuery);
public List<DetFacturaPago> query(String jpql);
public void crear(DetFacturaPago detFacturaPago); }

```

```

package com.js.sessionbeans.factura;
import java.util.List;
import javax.ejb.Remote;
import com.js.entitybeans.DetFacturaPago;
@Remote
public interface IDetFacPagoRemote {
    public void actualizar(DetFacturaPago detFacturaPago);
    public void borrar(long id) throws Exception;
    public List<DetFacturaPago> consultar();
    public List<DetFacturaPago> consultar(long id);
    public List<DetFacturaPago> consultar(long idFactura, String idFormaPago);
    public List<DetFacturaPago> consultarPorFactura(long idFactura);
    public List<DetFacturaPago> nativeQuery(String nativeQuery);
    public List<DetFacturaPago> query(String jpql);
    public void crear(DetFacturaPago detFacturaPago); }

```

Para la instalación del sistema se debe revisar el Manual de Instalación.⁹⁹

5.2. PRUEBAS

Las pruebas realizadas en el sistema de restaurantes fueron las siguientes:

- Pruebas de caja blanca
- Pruebas de caja negra
- Pruebas del sistema

Para algunas de las pruebas del sistema se utilizó el programa WebServer Stress Tool 7 (Versión Trial).

El detalle de las pruebas se encuentra en el Manual Técnico.¹⁰⁰

⁹⁹ Anexo 3: Manual de Instalación- Pag: 342

¹⁰⁰ Anexo 2: Manual Técnico - Pruebas. Pag: 331

CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- Se cumplió con el objetivo de desarrollar el sistema de restaurantes que permita en manejo de facturación, producción y control de inventarios, basado en las metodologías de desarrollo RUP y OOHDM.
- El sistema fue desarrollado con herramientas open source lo que permitió el rápido desarrollo de la aplicación, ya que existe una gran cantidad de herramientas y librerías disponibles en la red, de las cuales utilizamos JSF, RichFaces y Facelets.
- Se obtuvo reportes consolidados de las sucursales de la empresa, los cuales dan a conocer las ventas, la producción y el inventario de cada sucursal para la toma de decisiones gerenciales.
- Al realizar el presente trabajo adquirimos nuevos conocimientos en cuanto al desarrollo de aplicaciones en J2EE, el manejo del contenedor EJB, la utilización del servidor de aplicaciones Glassfish y la administración de la base de datos Oracle 10g.
- Aprendimos a desarrollar un sistema en capas con MVC, para la lógica de negocios manejamos los Session Beans, para la comunicación con la base de datos ocupamos los Entity Beans, lo que vendría a ser la capa modelo de nuestra aplicación; las interfaces para llamar a la ejecución de los métodos de los Session Beans es la capa de control y finalmente en la capa de vista utilizamos el framework JSF, juntamente con RichFaces para utilizar componentes AJAX, y Facelets para utilizar templates o plantillas lo que facilitó el desarrollo con XHTML.
- El sistema está diseñado de tal forma que se puedan integrar nuevos módulos, como por ejemplo el de contabilidad, de una manera fácil sin disminuir la calidad del sistema.

6.2. RECOMENDACIONES

- Se debe tomar en cuenta que el servidor donde se ejecute la aplicación debe ser un servidor de aplicaciones y no un servidor web, porque la aplicación posee EJB y solo un servidor de aplicaciones posee un contenedor de EJB.
- Tener en cuenta que al crear un pool de conexiones debe tener el nombre de: jdbc__/_Oracle10, porque este nombre es validado para la persistencia de los datos con la Base Oracle.
- Mientras se realiza el inventario inicial o la toma física se debe tomar en cuenta que los demás procesos se bloquean hasta que se terminen de realizar los procesos mencionados, debido a que si hay alguna compra o venta las existencias o stock cambiarían y no serían datos reales.
- El framework JSF 1.2 trabaja con RichFaces y Facelets conjuntamente, si utilizamos JSF 2 tendremos problemas de compatibilidad al deployar la aplicación.
- Se recomienda utilizar Jobos Tools para eclipse para trabajar con paginas XHTML debido a que tiene las herramientas necesarias para trabajar con paginas JSF con dicha extensión.
- Se debe configurar los puertos del servidor GlassFishV3 y de la base de datos Oracle, para que no ocupen los mismos puertos y no creen conflictos para la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

- El Proceso de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh
- UML Gota a Gota - Martin Fowler, Kendall Scott
- Patrones de Diseño - Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides
- WIKIPEDIA La Enciclopedia libre. JAVAEE. [En línea]. http://es.wikipedia.org/wiki/Java_EE. [Citado el 21 de mayo del 2010].
- WIKIPEDIA La Enciclopedia libre. OOHDM. [En línea]. <http://es.wikipedia.org/wiki/OOHDM>. [Citado el 20 de marzo del 2010].
- ORACLE. Seguridades de Base de Datos. <http://www.oracle.com/global/es/products/database/db-security.html>
- http://www.programacion.com/articulo/introduccion_a_la_tecnologia_java_server_faces_233
- <http://docs.sun.com/app/docs/doc/819-4627/6n6p1r5me?l=es&a=view>
- <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm>
- <http://www.javadabbadoo.org/cursos/infosintesis.net/javaeav/sessionBean/stateless/convertidor/paso02operativaCrearStatelessSessionBean.html>

ANEXOS

ANEXO 1. RESULTADO DE ENCUESTAS

La encuesta realizada en las diferentes cadenas de comida fue la siguiente:

Nombre Comercial:.....

Fecha:.....

1. ¿Realiza controles periódicos en sus bodegas?

Regularme De vez en cuando Nunca

2. ¿Lleva registros de su inventario?

Si No

3. La toma y entrega de pedidos al cliente es:

Eficiente Defectuosa No existe

4. ¿Tiene administración de cajas en su local?

Si No

5. El control de producción es:

Eficiente Defectuosa No existe

6. ¿En la administración actual puede usted saber cuánto realmente se debe invertir para obtener un margen de ganancia x?

Si No

7. ¿Los puntos de cobro tienen respaldo para saber cuánto percibe en pérdidas y ganancias?

Si No

8. ¿En su establecimiento se realizan kardex o similares para saber los movimientos diarios que realiza la bodega?

Sí No

9. El cuadro manual de facturas de su empresa le lleva aproximadamente el tiempo de:

Menos de 15 minutos Entre 15 y 30 minutos Más de 30 minutos

Las encuestas fueron realizadas a las siguientes cadenas de comida:

- RFC Chicken
- Piko Rico
- Ceviches de la Rumiñahui
- El Lince
- Pollo Gigante
- Ceviches Pedro & Pablo
- Las Palmeras
- El palacio del Menudo
- Gus
- Pollos Sandry

Los resultados de las encuestas se muestran en las siguientes gráficas:

Pregunta N° 1:

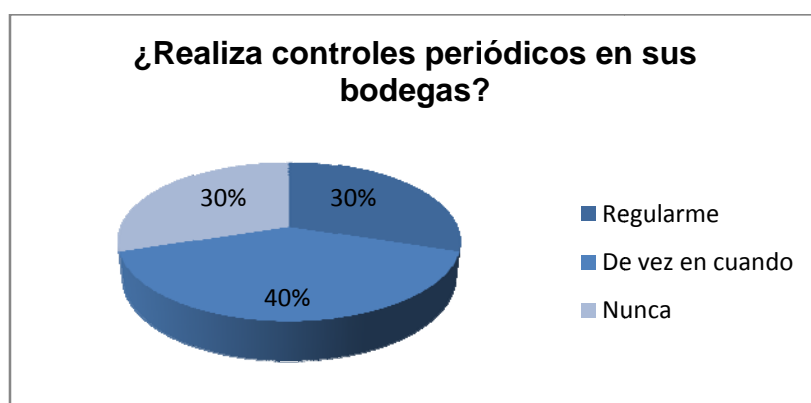


Figura 6.1. Resultados – Pregunta 1 ¹⁰¹

¹⁰¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Pregunta N° 2:

Figura 6.2. Resultados – Pregunta 2 ¹⁰²

Pregunta N° 3:

Figura 6.3. Resultados – Pregunta 3 ¹⁰³

Pregunta N° 4:

Figura 6.4. Resultados – Pregunta 4 ¹⁰⁴

¹⁰² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁰³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

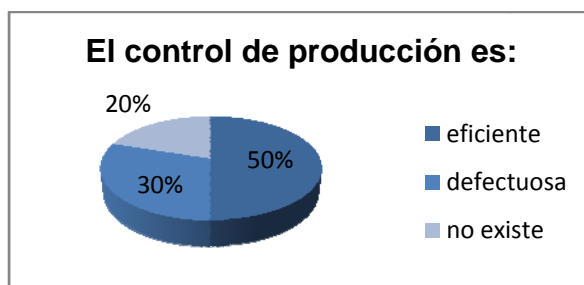
Pregunta N° 5:

Figura 6.5. Resultados – Pregunta 5 ¹⁰⁵

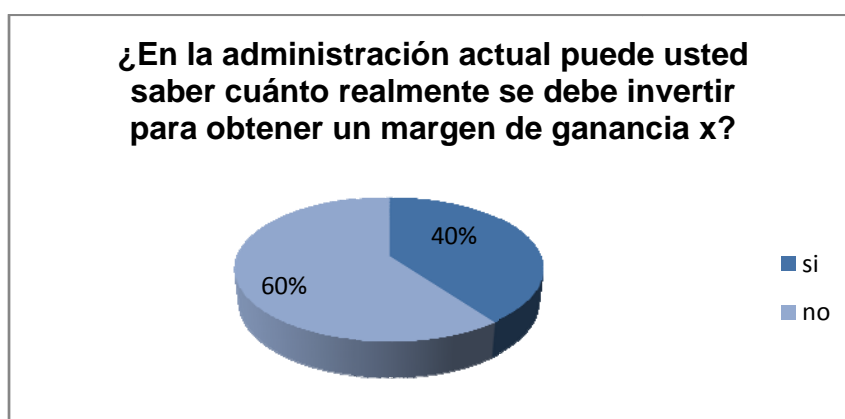
Pregunta N° 6:

Figura 6.6. Resultados – Pregunta 6 ¹⁰⁶

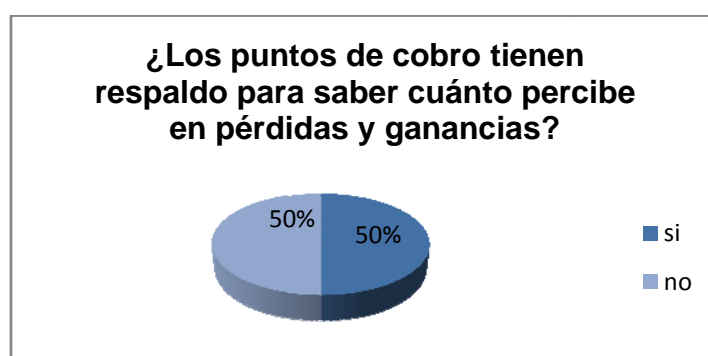
Pregunta N° 7:

Figura 6.7. Resultados – Pregunta 7 ¹⁰⁷

¹⁰⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁰⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁰⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

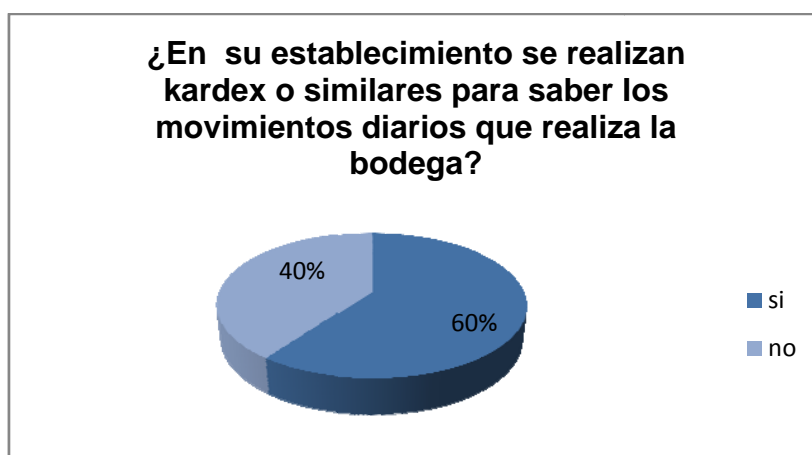
Pregunta N° 8:

Figura 6.8. Resultados – Pregunta 8 ¹⁰⁸

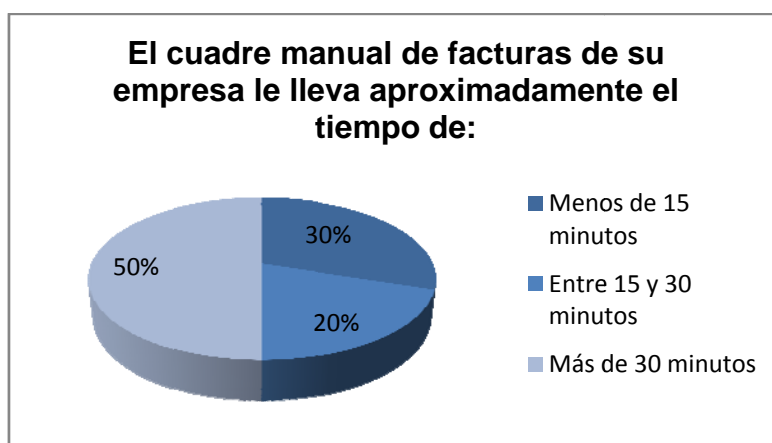
Pregunta N° 9:

Figura 6.9. Resultados – Pregunta 9 ¹⁰⁹

¹⁰⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁰⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁰⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

ANEXO 2. MANUAL TÉCNICO

CONTENIDO

CONTENIDO.....	165
1. DESCRIPCION GENERAL.....	167
1.1. ALCANCE	167
2. INGENIERIA DEL PROYECTO.....	168
2.1. MODELO DE REQUISITOS.....	168
2.1.1. MODELO DE CASOS DE USO	168
2.1.2. MODELO DE INTERFACES.....	225
2.1.3. MODELO DEL DOMINIO.....	227
2.2. MODELO DE ANÁLISIS	228
2.2.1. ARQUITECTURA DE CLASES.....	228
2.2.2. DIAGRAMAS DE SECUENCIA.....	242
2.2.3. DICCIONARIO DE CLASES SEGÚN MÓDULOS	284
2.3. MODELO DE DISEÑO	312
2.3.1. DISEÑO DE OBJETOS	312
2.3.2. DISEÑO DE INTERFASES.....	314
2.3.3. DIAGRAMA ENTIDAD RELACIÓN	314
2.3.4. DISEÑO DE ARQUITECTURA DEL SISTEMA	326
2.4. CODIFICACION Y PROGRAMACIÓN.....	327
2.5. DIAGRAMA DE COMPONENTES	327
3. INSTRUCCIONES DE INSTALACIÓN.....	327
4. REQUERIMIENTOS DE HARDWARE Y SOFTWARE.....	328
5. SEGURIDAD Y CONTROL.....	328
5.1 SEGURIDAD EN EL ACCESO DE LA INFORMACIÓN.....	328
5.2. CREACIÓN DE USUARIOS.....	328
5.3. SEGURIDAD DE ACCESO.....	329

5.4. SEGURIDAD DE USUARIO.....	329
5.5. POSIBLES USUARIOS.....	329
5.6. COPIAS DE SEGURIDAD	329
6. PRUEBAS.....	330
6.1. PRUEBA DE CAJA BLANCA.....	330
6.2. PRUEBA DE CAJA NEGRA.....	333
6.3. PRUEBAS DEL SISTEMA	335

En el manual se describe en forma general al proyecto, se especifican los recursos que fueron utilizados para la elaboración del sistema, además se detalla la instalación y configuración del proyecto con sus respectivas pruebas de funcionamiento.

1. DESCRIPCION GENERAL

La aplicación presenta una interfaz gráfica amigable para el usuario, la cual contiene un menú principal para acceder a los distintos módulos como son: Administración, Facturación e Inventario.

En el módulo de administración se puede crear la empresa con sus respectivas sucursales, a las cuales se les asigna una ubicación geográfica; además se maneja la seguridad del sistema con la creación de usuarios y perfiles de acceso.

El módulo de facturación permitirá el manejo ágil de cuentas en espera, devoluciones, anulaciones, consultas de los inventarios, diferentes formas de pago e impresiones de facturas y pre cuentas.

Para el módulo de inventarios se podrá crear los productos y catalogarlos para una mejor organización. Cada producto manejará tres tipos diferentes de precios y podrá formar parte de una receta o ser de tipo de servicio. Se podrá realizar la pre-producción y post-producción de las recetas y manejar el inventario de cada producto.

El sistema facilitará la obtención de informes gerenciales de los reportes obtenidos del cuadro de caja y de inventarios de la aplicación.

1.1. ALCANCE

La aplicación para el restaurante está enfocada a la administración de facturación, inventarios y producción de la empresa, para esto se han determinado tres módulos:

Administración

- Configuración de la empresa
- Manejo de sucursales

- Respaldos y restauración de la información
- Seguridad de acceso al sistema
- Reportes gerenciales

Inventarios

- Productos por líneas, grupos
- Tipos de productos: normal, recetas y servicios
- Codificación automática, 3 tipos de precios
- Pre producción y post producción de recetas
- Consultas y reportes

Facturación

- Cuentas en espera (cuentas abiertas)
- Formas de pago: efectivo, cheques y tarjeta de crédito
- Cierre y cuadro de caja
- Impresión de facturas y pre cuentas
- Consultas y reportes

Cabe destacar que el sistema no abarcará los puntos de contabilidad, cartera y recursos humanos, debido a que los puntos antes mencionados salen fuera del alcance del presente trabajo

Los tres módulos formarán parte de una aplicación que será instalada en un servidor de aplicaciones para poder aprovechar los recursos de hardware del servidor y no saturar a las estaciones de trabajo cuando realicen algún proceso; la información se almacenará en una única base de datos y se generarán backups periódicamente para respaldar la información.

2. INGENIERIA DEL PROYECTO

2.1. MODELO DE REQUISITOS

2.1.1. MODELO DE CASOS DE USO

Módulo Administración

Caso de Uso: Configuración de la Empresa

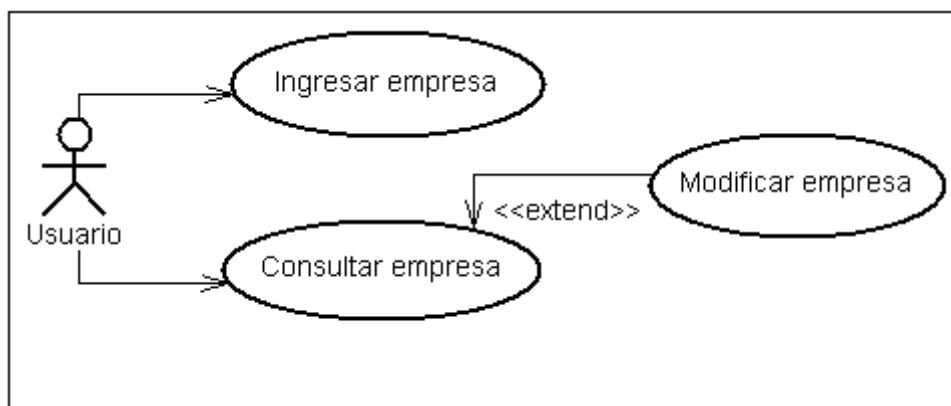


Figura 7.1. Caso de Uso del Sistema: Configuración de la Empresa ¹¹⁰

Especificación de caso de uso: Ingresar empresa

Caso de Uso	Ingresar empresa
Objetivo	Registrar los datos de la empresa
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para el ingreso de los datos de la empresa.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Empresa que se encuentra en el módulo de Administración. • Elegir la opción Nuevo • Ingresar los datos de la empresa: código, nombre, razón social, RUC, dirección, teléfono, celular, código postal, email. • Validar cada uno de los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que la empresa ha sido creada.

Tabla 7.1. Especificación de caso de uso: Ingresar empresa ¹¹¹

¹¹⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹¹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Consultar empresa

Caso de Uso	Consultar empresa
Objetivo	Visualizar los datos que la empresa posee.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar los datos de la empresa.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Empresa que se encuentra en el módulo de Administración. • Visualizar las empresas
Pos condiciones	El sistema muestra la información que la empresa posee.

Tabla 7.2. Especificación de caso de uso: Consultar empresa ¹¹²

Especificación de caso de uso: Modificar empresa

Caso de Uso	Modificar empresa
Objetivo	Realizar cambios en los datos de la empresa.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar los datos de la empresa.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Empresa que se encuentra en el módulo de Administración. • Elegir la empresa a modificar. • Elegir la opción Actualizar. • Modificar los datos de la empresa a excepción del código. • Validar los datos. • Grabar la información
Pos condiciones	El sistema muestra un mensaje de que los datos han sido actualizados correctamente.

Tabla 7.3. Especificación de caso de uso: Modificar empresa ¹¹³

¹¹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración de Sucursal

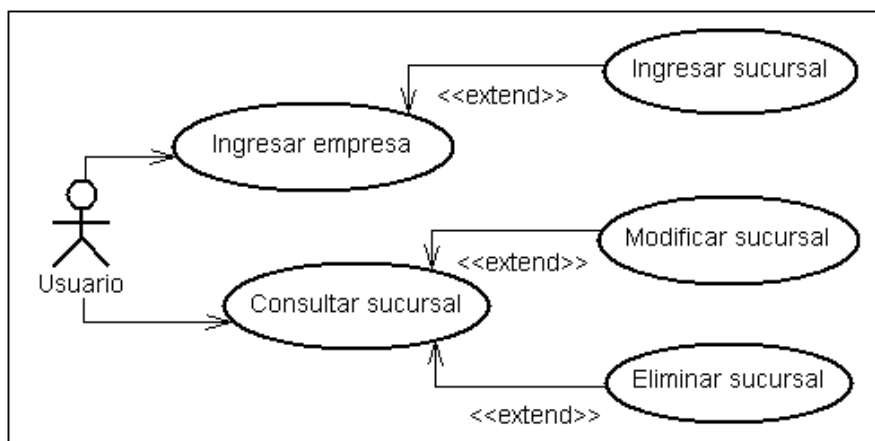


Figura 7.2. Caso de Uso del Sistema: Configuración de Sucursal ¹¹⁴

Especificación de caso de uso: Ingresar sucursal

Caso de Uso	Ingresar sucursal
Objetivo	Ingresar los datos de una sucursal que posee la empresa.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para el ingreso de los datos de la sucursal. • Se debe haber creado una ciudad y zona para asignarle a la sucursal.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Sucursal que se encuentra en la opción Empresa dentro del módulo de Administración. • Elegir la opción Nuevo. • Seleccionar una ciudad. • Seleccionar una zona. • Ingresar los datos de la sucursal: código, nombre. • Validar los datos ingresados.

¹¹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹¹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema valida si la sucursal existe o no. • Si la sucursal no existe entonces el sistema muestra un mensaje de que la sucursal ha sido creada. • El sistema crea automáticamente una bodega por sucursal.

Tabla 7.4. Especificación de caso de uso: Ingresar sucursal ¹¹⁵

Especificación de caso de uso: Consultar sucursal

Caso de Uso	Consultar sucursal
Objetivo	Visualizar la información de una sucursal
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar sucursales.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Sucursal que se encuentra en la opción Empresa dentro del módulo de Administración. • Visualizar los datos de la sucursal.
Pos condiciones	El sistema muestra la información que la sucursal posee.

Tabla 7.5. Especificación de caso de uso: Consultar sucursal ¹¹⁶

Especificación de caso de uso: Modificar sucursal

Caso de Uso	Modificar sucursal
Objetivo	Realizar cambios en los datos de la sucursal.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar sucursales.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Sucursal que se encuentra en la opción Empresa dentro del módulo de

¹¹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹¹⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	Administración. <ul style="list-style-type: none"> • Elegir la opción Actualizar de la sucursal a modificar. • Modificar los datos de la sucursal. • Validar los datos. • Grabar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.6. Especificación de caso de uso: Modificar sucursal ¹¹⁷

Especificación de caso de uso: Eliminar sucursal

Caso de Uso	Eliminar sucursal
Objetivo	Borrar los datos de una sucursal del sistema.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para eliminar sucursales.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Sucursal que se encuentra en la opción Empresa dentro del módulo de Administración. • Elegir la sucursal • Elegir la opción Borrar de la sucursal.
Pos condiciones	<ul style="list-style-type: none"> • El sistema verifica si la sucursal no posee registros de ventas y de inventarios. • Si la sucursal no posee dicha información el sistema muestra un mensaje de que la sucursal ha sido eliminada.

Tabla 7.7. Especificación de caso de uso: Eliminar sucursal ¹¹⁸

¹¹⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹¹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración de ubicación geográfica

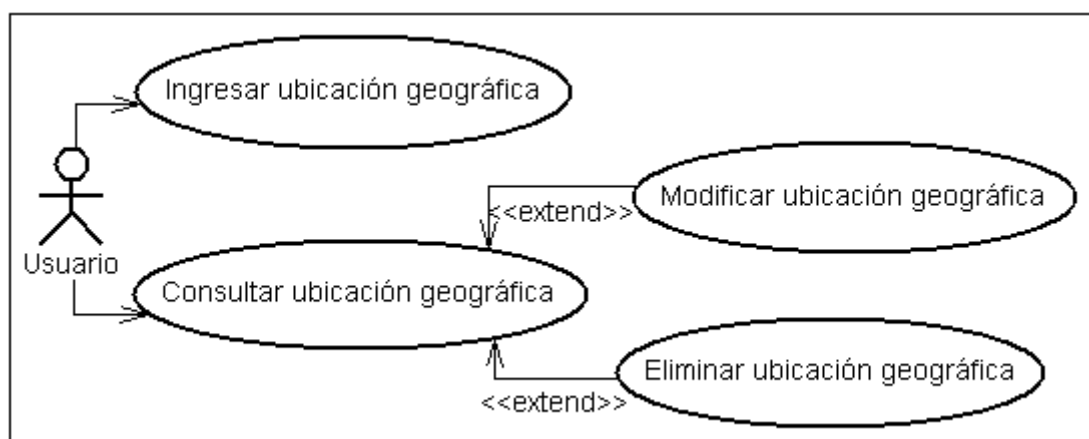


Figura 7.3. Caso de Uso del Sistema: Configuración de ubicación geográfica ¹¹⁹

Especificación de caso de uso: Ingresar ubicación geográfica

Caso de Uso	Ingresar ubicación geográfica
Objetivo	Creación de la ubicación geográfica de la empresa la cual consta de una estructura jerárquica: país, región, provincia, ciudad y zona
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para ingresar una ubicación geográfica.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Ubicación Geográfica que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Seleccionar la opción País, Región, Provincia, Ciudad o Zona en donde se podrá ingresar los datos. • Elegir Nuevo de acuerdo a la opción escogida. • Ingresar los datos. • Validar los datos en donde una provincia depende de un país, una ciudad depende de una provincia y una zona depende de una ciudad. • Grabar los datos.

¹¹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Pos condiciones	<ul style="list-style-type: none"> El sistema muestra un mensaje de que los datos han sido ingresados exitosamente.
------------------------	--

Tabla 7.8. Especificación de caso de uso: Ingresar ubicación geográfica ¹²⁰

Especificación de caso de uso: Consultar ubicación geográfica

Caso de Uso	Consultar ubicación geográfica
Objetivo	Visualizar los datos de la ubicación geográfica la cual consta de una estructura jerárquica: país, región, provincia, ciudad y zona.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar una ubicación geográfica.
Acciones Básicas	<ul style="list-style-type: none"> Ingresar a la opción Ubicación Geográfica que se encuentra en la opción Datos Generales dentro del módulo de Administración. Ingresar a la opción País, Región, Provincia, Ciudad o Zona. Visualizar los datos correspondientes a cada opción.
Pos condiciones	<ul style="list-style-type: none"> El sistema muestra los datos del País, Región, Provincia, Ciudad y Zona.

Tabla 7.9. Especificación de caso de uso: Consultar ubicación geográfica ¹²¹

Especificación de caso de uso: Modificar ubicación geográfica

Caso de Uso	Modificar ubicación geográfica
Objetivo	Cambiar los datos de la ubicación geográfica la cual consta de una estructura jerárquica: país, región, provincia, ciudad y zona.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar una ubicación geográfica.

¹²⁰ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹²¹ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Ubicación Geográfica que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Seleccionar la opción País, Región, Provincia, Ciudad o Zona para realizar la consulta. • Elegir la opción Actualizar del dato a modificar. • Modificar los datos. • Validar los datos • Grabar los cambios
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que los datos se han modificado correctamente.

Tabla 7.10. Especificación de caso de uso: Modificar ubicación geográfica ¹²²

Especificación de caso de uso: Eliminar ubicación geográfica

Caso de Uso	Eliminar ubicación geográfica
Objetivo	Borrar los datos de la ubicación geográfica la cual consta de una estructura jerárquica: país, región, provincia, ciudad y zona.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para eliminar una ubicación geográfica. • Para eliminar una opción de la ubicación geográfica esta no debe tener datos dependientes.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Ubicación Geográfica que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Seleccionar la opción País, Región, Provincia, Ciudad o Zona para realizar la consulta. • Elegir opción Borrar del dato a eliminar.
Pos condiciones	<ul style="list-style-type: none"> • Si la ubicación geográfica seleccionada no tuvo datos dependientes el sistema muestra un mensaje

¹²² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	de eliminación satisfactoria.
--	-------------------------------

Tabla 7.11. Especificación de caso de uso: Eliminar ubicación geográfica ¹²³

Caso de Uso: Configuración de Forma de Pago

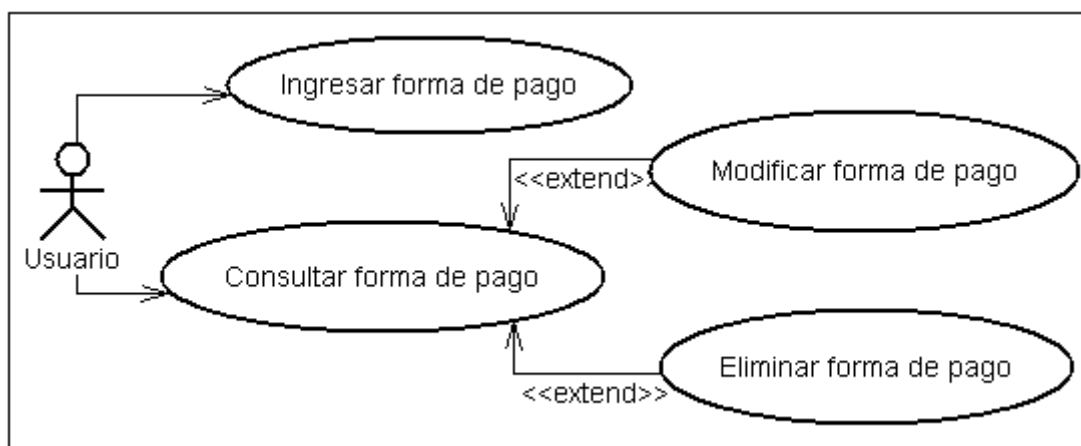


Figura 7.4. Caso de Uso del Sistema: Configuración de forma de pago ¹²⁴

Especificación de caso de uso: Ingresar forma de pago

Caso de Uso	Ingresar las formas de pago
Objetivo	Ingresar los datos de una forma de pago para la facturación.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> El usuario debe tener un perfil con los permisos necesarios para el ingreso de formas de pago.
Acciones Básicas	<ul style="list-style-type: none"> Ingresar a la opción de Forma de pago que se encuentra en la opción Datos Generales dentro del módulo de Administración. Elegir la opción Nuevo. Ingresar los datos de la forma de pago: código, nombre y observación. Validar los datos ingresados.

¹²³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹²⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que el ingreso ha sido satisfactorio.

Tabla 7.12. Especificación de caso de uso: Ingresar forma de pago ¹²⁵

Especificación de caso de uso: Consultar forma de pago

Caso de Uso	Consultar forma de pago
Objetivo	Visualizar la información de las formas de pago existentes.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar forma de pago.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Forma de pago que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Visualizar los datos de la forma de pago.
Pos condiciones	El sistema muestra la información de las formas de pago.

Tabla 7.13. Especificación de caso de uso: Consultar forma de pago ¹²⁶

Especificación de caso de uso: Modificar forma de pago

Caso de Uso	Modificar forma de pago
Objetivo	Realizar cambios en los datos de las formas de pago
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar forma de pago
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Forma de pago que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Elegir la opción Actualizar de la forma de pago a

¹²⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹²⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<p>modificar.</p> <ul style="list-style-type: none"> • Modificar los datos de la forma de pago. • Validar los datos. • Grabar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.14. Especificación de caso de uso: Modificar forma de pago ¹²⁷

Especificación de caso de uso: Eliminar forma de pago

Caso de Uso	Eliminar forma de pago
Objetivo	Borrar los datos de una forma de pago del sistema.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para eliminar formas de pago.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Forma de pago que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Elegir la forma de pago • Elegir la opción Borrar de la forma de pago.
Pos condiciones	<ul style="list-style-type: none"> • El sistema verifica que no existan registros vinculados a la forma de pago y borra.

Tabla 7.15. Especificación de caso de uso: Eliminar forma de pago ¹²⁸

Caso de Uso: Configuración de Impuesto

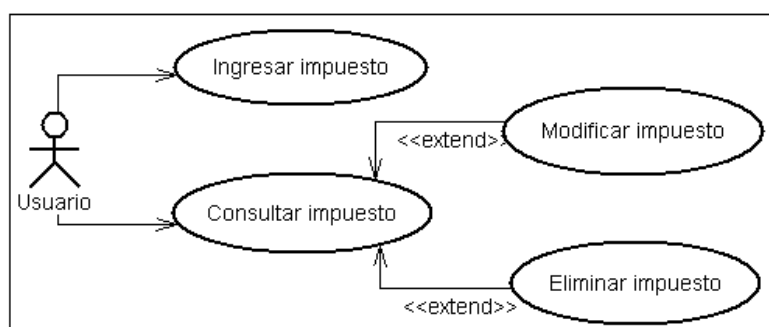


Figura 7.5. Caso de Uso del Sistema: Configuración de impuesto ¹²⁹

¹²⁷ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹²⁸ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Ingresar impuesto

Caso de Uso	Ingresar impuesto
Objetivo	Ingresar los datos de un impuesto para la facturación.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> El usuario debe tener un perfil con los permisos necesarios para el ingreso de impuestos.
Acciones Básicas	<ul style="list-style-type: none"> Ingresar a la opción de Impuesto que se encuentra en la opción Datos Generales dentro del módulo de Administración. Elegir la opción Nuevo. Ingresar los datos del impuesto: código, nombre y porcentaje. Elegir el tipo: incluido VAT, excluido ADD. Validar los datos ingresados. Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> El sistema muestra un mensaje de que el ingreso ha sido satisfactorio.

Tabla 7.16. Especificación de caso de uso: Ingresar impuesto ¹³⁰

Especificación de caso de uso: Consultar impuesto

Caso de Uso	Consultar impuesto
Objetivo	Visualizar la información de los impuestos existentes.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar impuesto.
Acciones Básicas	<ul style="list-style-type: none"> Ingresar a la opción Impuesto que se encuentra en la opción Datos Generales dentro del módulo de Administración. Visualizar los datos de los impuestos.

¹²⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹³⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Pos condiciones	El sistema muestra la información de los impuestos.
------------------------	---

Tabla 7.17. Especificación de caso de uso: Consultar impuesto ¹³¹

Especificación de caso de uso: Modificar impuesto

Caso de Uso	Modificar impuesto
Objetivo	Realizar cambios en los datos de los impuestos
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar impuestos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Impuesto que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Elegir la opción Actualizar del impuesto a modificar. • Modificar los datos del impuesto. • Validar los datos. • Grabar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.18. Especificación de caso de uso: Modificar impuesto ¹³²

Especificación de caso de uso: Eliminar impuesto

Caso de Uso	Eliminar impuesto
Objetivo	Borrar los datos de un impuesto del sistema.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para eliminar impuestos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Impuesto que se encuentra en la opción Datos Generales dentro del módulo de Administración.

¹³¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹³² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Elegir la impuesto • Elegir la opción Borrar del impuesto.
Pos condiciones	<ul style="list-style-type: none"> • El sistema verifica que no existan registros vinculados al impuesto y borra.

Tabla 7.19. Especificación de caso de uso: Eliminar impuesto ¹³³

Caso de Uso: Configuración de Descuentos y recargos

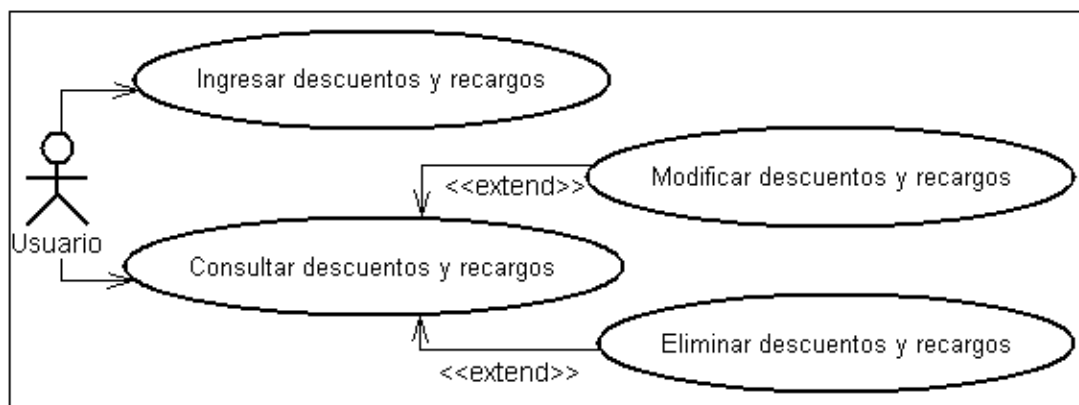


Figura 7.6. Caso de Uso del Sistema: Configuración de descuentos y recargos ¹³⁴

Especificación de caso de uso: Ingresar descuentos y recargos

Caso de Uso	Ingresar descuentos y recargos
Objetivo	Ingresar los datos de descuentos y recargos para la facturación.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para el ingreso de descuentos y recargos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Descuentos y Recargos que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Elegir la opción Nuevo. • Ingresar los datos de descuentos y recargos: código, nombre y porcentaje.

¹³³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹³⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Elegir el tipo: descuento o recargo. • Validar los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que el ingreso ha sido satisfactorio.

Tabla 7.20. Especificación de caso de uso: Ingresar descuentos y recargos ¹³⁵

Especificación de caso de uso: Consultar descuentos y recargos

Caso de Uso	Consultar descuentos y recargos
Objetivo	Visualizar la información de los descuentos y recargos existentes.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar descuentos y recargos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Descuentos y Recargos que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Visualizar los datos de los descuentos y recargos.
Pos condiciones	El sistema muestra la información de los descuentos y recargos.

Tabla 7.21. Especificación de caso de uso: Consultar descuentos y recargos ¹³⁶

Especificación de caso de uso: Modificar descuentos y recargos

Caso de Uso	Modificar descuentos y recargos
Objetivo	Realizar cambios en los datos de los descuentos y recargos.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar descuentos y recargos.
Acciones	<ul style="list-style-type: none"> • Ingresar a la opción de Descuentos y Recargos que

¹³⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹³⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Básicas	<p>se encuentra en la opción Datos Generales dentro del módulo de Administración.</p> <ul style="list-style-type: none"> • Elegir la opción Actualizar del descuento o recargo a modificar. • Modificar los datos del descuento o recargo. • Validar los datos. • Grabar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.22. Especificación de caso de uso: Modificar descuentos y recargos ¹³⁷

Especificación de caso de uso: Eliminar descuentos y recargos

Caso de Uso	Eliminar descuentos y recargos
Objetivo	Borrar los datos de un descuento o recargo del sistema.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para eliminar descuentos y recargos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Descuentos y Recargos que se encuentra en la opción Datos Generales dentro del módulo de Administración. • Elegir el descuento o recargo. • Elegir la opción Borrar del descuento o recargo.
Pos condiciones	<ul style="list-style-type: none"> • El sistema verifica que no existan registros vinculados al descuento o recargos y borra.

Tabla 7.23. Especificación de caso de uso: Eliminar descuentos y recargos ¹³⁸

¹³⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹³⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración perfiles de usuario

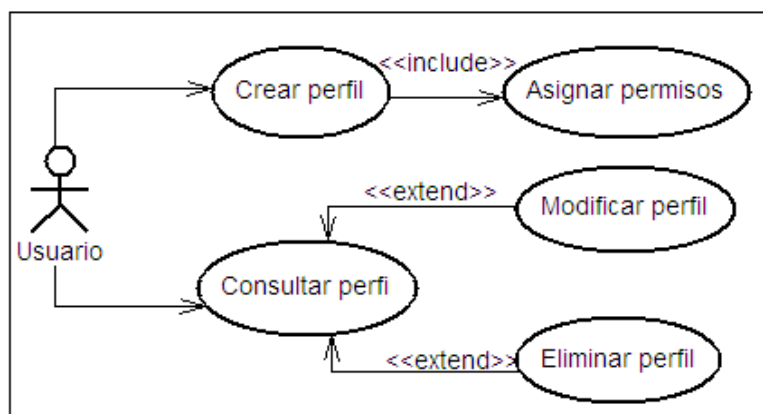


Figura 7.7. Caso de Uso del Sistema: Configuración perfiles de usuario ¹³⁹

Especificación de caso de uso: Crear perfil

Caso de Uso	Crear perfil
Objetivo	Ingresar los datos de un nuevo perfil
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para crear un perfil.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Perfil de Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Elegir la opción Nuevo. • Ingresar los datos del nuevo perfil: código, nombre. • Elegir la opción Permisos. • Asignarle los permisos al perfil de acuerdo a los módulos. • Elegir la opción Retornar. • Validar los datos. • Grabar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que los datos han sido guardados exitosamente.

Tabla 7.24. Especificación de caso de uso: Crear perfil ¹⁴⁰

¹³⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Consultar perfil

Caso de Uso	Consultar perfil
Objetivo	Visualizar la información de un perfil
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar un perfil.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Perfil de Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Visualizar los datos de los perfiles de usuario.
Pos condiciones	El sistema muestra toda la información de los perfiles.

Tabla 7.25. Especificación de caso de uso: Consultar perfil ¹⁴¹

Especificación de caso de uso: Modificar perfil

Caso de Uso	Modificar perfil
Objetivo	Cambiar los datos del perfil de usuario y/o cambiar los permisos asignados al mismo.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar un perfil.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Perfil de Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Elegir la opción Actualizar del perfil de usuario a modificar. • Modificar los datos del perfil de usuario. • Para modificar el acceso, elegir la opción Permisos, y marcar los accesos de acuerdo a cada módulo. • Elegir la opción Retornar

¹⁴⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁴¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.26. Especificación de caso de uso: Modificar perfil ¹⁴²

Especificación de caso de uso: Eliminar perfil

Caso de Uso	Eliminar perfil
Objetivo	Borrar los datos de un perfil
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario los permisos necesarios para eliminar un perfil. • No se puede eliminar un perfil si uno o más usuarios tienen asignados dicho perfil
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Perfil de Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Elegir la opción Borrar del perfil. • Eliminar la información del perfil.
Pos condiciones	El sistema muestra un mensaje de que el perfil ha sido eliminado.

Tabla 7.27. Especificación de caso de uso: Eliminar perfil ¹⁴³

¹⁴² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁴³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configurar usuario

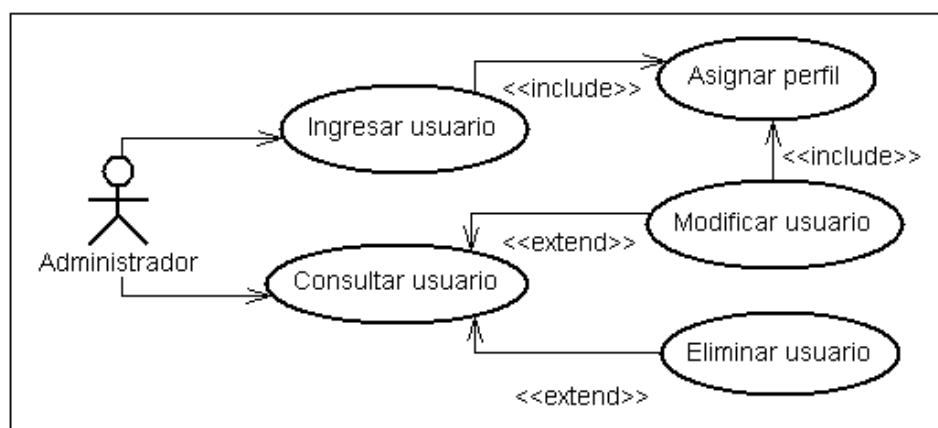


Figura 7.8. Caso de Uso del Sistema: Configurar usuario ¹⁴⁴

Especificación de caso de uso: Ingresar usuario

Caso de Uso	Ingresar usuario
Objetivo	Ingresar los datos de un nuevo usuario para que maneje el sistema.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para ingresar un usuario.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Elegir el perfil del usuario. • Ingresar los datos del usuario: código, nombre, estado, sucursal de ventas, login, password y confirmación del password. • Validar cada uno de los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema valida si el usuario existe o no. • Si el usuario no existe entonces el sistema muestra un mensaje de "Ingreso exitoso".

Tabla 7.28. Especificación de caso de uso: Ingresar usuario ¹⁴⁵

¹⁴⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Consultar usuario

Caso de Uso	Consultar usuario
Objetivo	Visualizar los datos del usuario.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar usuarios.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Usuarios que se encuentra en el módulo de administración. • Visualizar los datos de los usuarios.
Pos condiciones	El sistema muestra la información de los usuarios.

Tabla 7.29. Especificación de caso de uso: Consultar usuario ¹⁴⁶

Especificación de caso de uso: Modificar usuario

Caso de Uso	Modificar usuario
Objetivo	Realizar cambios en los datos del usuario.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar usuarios.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Elegir la opción Actualizar del usuario a modificar. • Modificar los datos del usuario. • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.30. Especificación de caso de uso: Modificar usuario ¹⁴⁷

¹⁴⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁴⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁴⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Eliminar usuario

Caso de Uso	Eliminar usuario
Objetivo	Borrar los datos de un usuario.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para eliminar usuarios. • El usuario no debe constar en el registro de ventas para poder ser eliminado.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Usuario que se encuentra en la opción Seguridad dentro del módulo de Administración. • Elegir el usuario. • Elegir la opción Borrar del usuario.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que el usuario ha sido borrado.

Tabla 7.31. Especificación de caso de uso: Eliminar usuario ¹⁴⁸

Caso de Uso: Generar respaldos

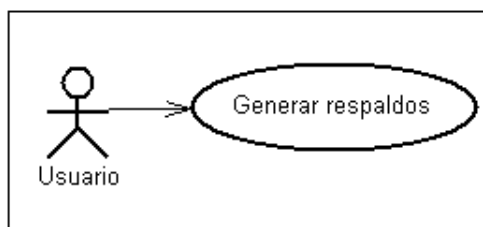


Figura 7.9. Caso de Uso del Sistema: Generar respaldos ¹⁴⁹

Especificación de caso de uso: Generar respaldos

Caso de Uso	Generar respaldos
Objetivo	Generar un respaldo de la información de la empresa.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos

¹⁴⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁴⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	necesarios generar respaldos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a opción Respaldo de Datos que se encuentra en Procesos, dentro del módulo de Administración. • Elegir la opción Iniciar respaldo.
Pos condiciones	El sistema guarda el respaldo como un archivo con la extensión .cmd

Tabla 7.32. Especificación de caso de uso: Generar respaldos ¹⁵⁰

Módulo Inventario

Caso de Uso: Configuración Líneas de Productos

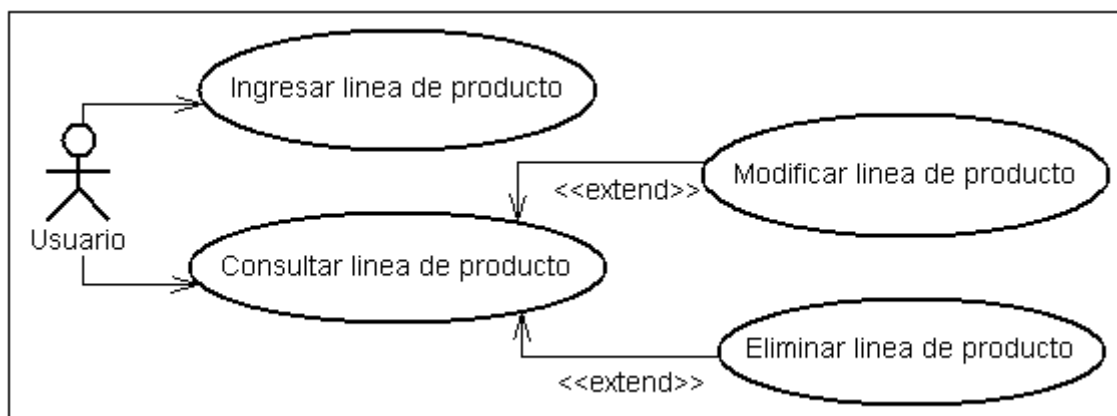


Figura 7.10. Caso de Uso del Sistema: Configuración Líneas de Productos ¹⁵¹

Especificación de caso de uso: Ingresar línea de producto

Caso de Uso	Ingresar línea de producto
Objetivo	Ingresar los datos de una nueva línea de producto
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para

¹⁵⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁵¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	crear una línea de producto.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Línea de Productos que se encuentra en la opción Datos Generales dentro del módulo de Inventario. • Ingresar los datos de la línea de producto: código y nombre • Validar cada uno de los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema valida si la línea de producto existe o no. • Si la línea de producto no existe el sistema muestra un mensaje de ingreso exitoso.

Tabla 7.33. Especificación de caso de uso: Ingresar línea de producto ¹⁵²

Especificación de caso de uso: Consultar línea de producto

Caso de Uso	Consultar línea de producto
Objetivo	Visualizar la información de línea de producto
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para crear una línea de producto.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Línea de Producto que se encuentra en la opción Datos Generales dentro del módulo de Inventario. • Visualizar los datos de las líneas de productos.
Pos condiciones	El sistema muestra la información de la línea de producto.

Tabla 7.34. Especificación de caso de uso: Consultar línea de producto ¹⁵³

¹⁵² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Modificar línea de producto

Caso de Uso	Modificar línea de producto.
Objetivo	Realizar cambios en la línea de producto.
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para crear una línea de producto.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Línea de Producto que se encuentra en la opción Datos Generales dentro del módulo de Inventario. • Elegir la opción Actualizar de la línea de producto. • Modificar los datos. • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.35. Especificación de caso de uso: Modificar línea de producto ¹⁵⁴

Especificación de caso de uso: Eliminar línea de producto

Caso de Uso	Eliminar línea de producto.
Objetivo	Borrar la información de línea de producto.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para crear una línea de producto. • La línea de producto no deberá tener información relacionada para ser eliminada.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Línea de Producto. que se encuentra en el módulo de Inventario. • Elegir la opción Borrar de la línea de producto.

¹⁵³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁵⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Pos condiciones	<ul style="list-style-type: none"> El sistema muestra un mensaje de que la línea de producto ha sido eliminada.
------------------------	--

Tabla 7.36. Especificación de caso de uso: Eliminar línea de producto ¹⁵⁵

Caso de Uso: Configuración Grupo de Productos

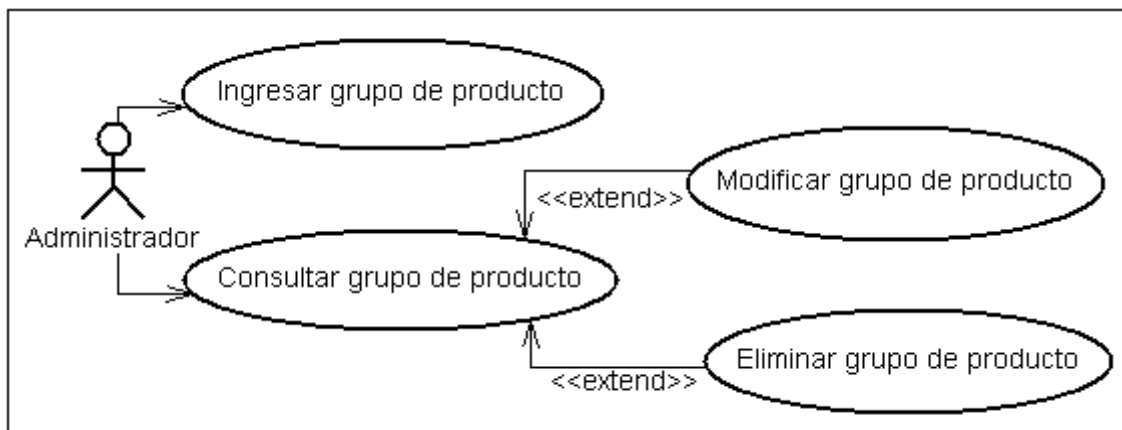


Figura 7.11. Caso de Uso del Sistema: Configuración Grupo de Productos ¹⁵⁶

Especificación de caso de uso: Ingresar grupo de producto

Caso de Uso	Ingresar grupo de producto
Objetivo	Ingresar los datos de un nuevo grupo de producto
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para crear un grupo de producto.
Acciones Básicas	<ul style="list-style-type: none"> Ingresar a Grupo de Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. Seleccionar la línea de productos a la cual corresponde el nuevo grupo de productos. Ingresar los datos del grupo de producto: código y nombre Validar cada uno de los datos ingresados.

¹⁵⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁵⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema valida si el grupo de producto existe o no. • Si el grupo de producto no existe el sistema muestra un mensaje de ingreso exitoso.

Tabla 7.37. Especificación de caso de uso: Ingresar grupo de producto ¹⁵⁷

Especificación de caso de uso: Consultar grupo de producto

Caso de Uso	Consultar grupo de producto
Objetivo	Visualizar la información del grupo de producto
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para consultar un grupo de producto.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Grupo de Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. • Visualizar los datos de los grupos de producto
Pos condiciones	El sistema muestra la información del grupo de producto.

Tabla 7.38. Especificación de caso de uso: Consultar grupo de producto ¹⁵⁸

Especificación de caso de uso: Modificar grupo de producto

Caso de Uso	Modificar grupo de producto.
Objetivo	Realizar cambios en el grupo de producto.
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para modificar un grupo de producto.
Acciones	<ul style="list-style-type: none"> • Ingresar a Grupo de Productos que se encuentra en

¹⁵⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁵⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Básicas	<p>la opción Datos Generales en el módulo de Inventario.</p> <ul style="list-style-type: none"> • Elegir la opción Actualizar del grupo de producto. • Modificar los datos. • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.39. Especificación de caso de uso: Modificar grupo de producto ¹⁵⁹

Especificación de caso de uso: Eliminar grupo de producto

Caso de Uso	Eliminar grupo de producto.
Objetivo	Borrar la información de grupo de producto.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para eliminar un grupo de producto. • El grupo de productos no deberá tener información relacionada para ser eliminada.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Grupo de Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. • Elegir la opción Borrar del grupo de productos.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que el grupo de productos ha sido eliminado.

Tabla 7.40. Especificación de caso de uso: Eliminar grupo de producto ¹⁶⁰

¹⁵⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁶⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración de Productos

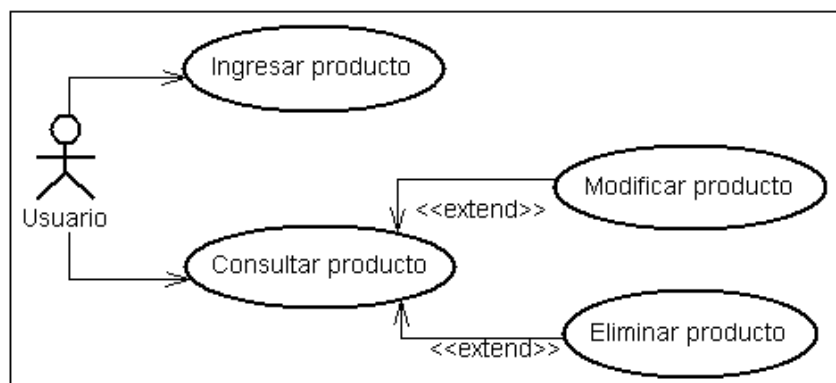


Figura 7.12. Caso de Uso del Sistema: Configuración de Productos ¹⁶¹

Especificación de caso de uso: Ingresar producto

Caso de Uso	Ingresar producto
Objetivo	Ingresar los datos de un nuevo producto
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para crear un producto. • Haber creado al menos un grupo de producto
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. • Seleccionar la línea de producto al cual pertenece el producto. • Seleccionar el grupo de producto al cual pertenece el producto. • Ingresar los datos del producto: código, nombre, impuesto, precios, tipo de producto: receta, normal o ingrediente, estado. • Si el producto es tipo receta se debe elegir el tipo de receta: postproducción o preproducción.

¹⁶¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Validar cada uno de los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • Si el producto no existe el sistema muestra un mensaje de ingreso exitoso.

Tabla 7.41. Especificación de caso de uso: Ingresar producto ¹⁶²

Especificación de caso de uso: Consultar producto

Caso de Uso	Consultar producto
Objetivo	Visualizar la información del grupo de producto
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para consultar un producto.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. • Visualizar los datos de los productos.
Pos condiciones	El sistema muestra la información del producto.

Tabla 7.42. Especificación de caso de uso: Consultar producto ¹⁶³

Especificación de caso de uso: Modificar producto

Caso de Uso	Modificar producto.
Objetivo	Realizar cambios en el producto.
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para modificar un producto.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. • Elegir la opción Actualizar del producto.

¹⁶² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁶³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Modificar los datos. • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.43. Especificación de caso de uso: Modificar producto ¹⁶⁴

Especificación de caso de uso: Eliminar producto

Caso de Uso	Eliminar producto.
Objetivo	Borrar la información de un producto.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para eliminar un producto. • El producto no deberá tener información relacionada para ser eliminada.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Productos que se encuentra en la opción Datos Generales en el módulo de Inventario. • Elegir la opción Borrar del producto.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que el producto ha sido eliminado.

Tabla 7.44. Especificación de caso de uso: Eliminar producto ¹⁶⁵

Caso de Uso: Configuración de Recetas

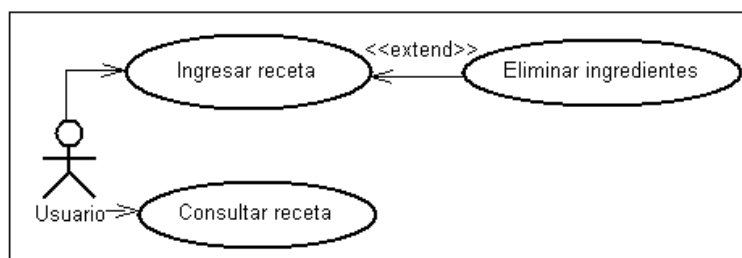


Figura 7.13. Caso de Uso del Sistema: Configuración de Recetas ¹⁶⁶

¹⁶⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁶⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Ingresar receta

Caso de Uso	Ingresar receta
Objetivo	Ingresar los datos de una nueva receta
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para crear una receta.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Recetas que se encuentra en la opción Datos Generales dentro del módulo de Inventario. • Seleccionar el producto de tipo receta a conformar. • Elegir la opción Agregar Ingredientes. • Ingresar los ingredientes y la cantidad que forman parte de la receta. • Validar cada uno de los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • Si el ingrediente no existe el sistema muestra un mensaje de ingreso exitoso.

Tabla 7.45. Especificación de caso de uso: Ingresar receta ¹⁶⁷

Especificación de caso de uso: Consultar receta

Caso de Uso	Consultar receta
Objetivo	Visualizar los datos de la receta
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para consultar una receta.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Recetas que se encuentra en la opción Datos Generales dentro del módulo de Inventario.

¹⁶⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁶⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Visualizar los datos de las recetas existentes.
Pos condiciones	El sistema muestra la información de la receta.

Tabla 7.46. Especificación de caso de uso: Consultar receta ¹⁶⁸

Especificación de caso de uso: Eliminar ingredientes

Caso de Uso	Eliminar ingredientes
Objetivo	Borrar los ingredientes que forman parte de la receta.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para eliminar los ingredientes de la receta. • La receta no deberá estar asociada a ningún registro.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Recetas que se encuentra en la opción Datos Generales dentro del módulo de Inventario. • Ingresar a la opción Agregar Ingredientes. • Elegir la opción Borrar del ingrediente que conforma la receta.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que el ingrediente ha sido eliminado.

Tabla 7.47. Especificación de caso de uso: Eliminar receta ¹⁶⁹

¹⁶⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁶⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración del Inventario

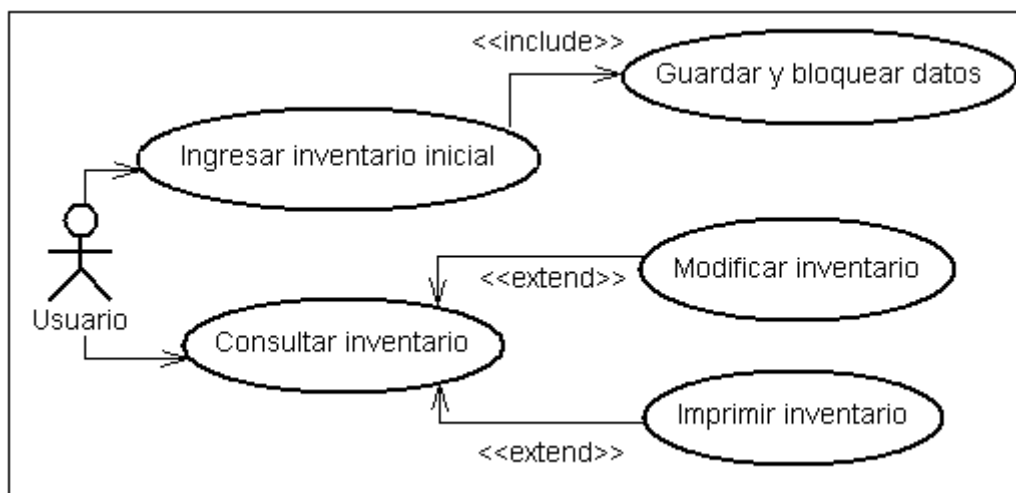


Figura 7.14. Caso de Uso del Sistema: Configuración de inventario

Especificación de caso de uso: Ingresar inventario inicial

Caso de Uso	Ingresar inventario inicial
Objetivo	Ingresar los datos del inventario inicial
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para ingresar el inventario inicial. • Haber ingresado productos y creado sucursales.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Inventario inicial que se encuentra en la opción Procesos dentro del módulo de Inventario. • Seleccionar la bodega. • Consultar el producto. • Ingresar costo y cantidad del producto. • Grabar la información. • Una vez ingresados todos los productos se debe elegir la opción Generar Inventario Inicial. • Seguir ingresando los productos

	correspondientes al inventario inicial.
Pos condiciones	<ul style="list-style-type: none"> • El sistema guarda y bloquea los datos del inventario inicial.

Tabla 7.48. Especificación de caso de uso: Ingresar inventario inicial ¹⁷⁰

Especificación de caso de uso: Consultar inventario

Caso de Uso	Consultar inventario
Objetivo	Visualizar los datos del inventario
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para consultar el inventario.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Inventario inicial que se encuentra en la opción Procesos dentro del módulo de Inventario. • Seleccionar Bodega. • Visualizar los productos del inventario inicial de acuerdo a la Bodega.
Pos condiciones	El sistema muestra la información del inventario inicial.

Tabla 7.49. Especificación de caso de uso: Consultar inventario ¹⁷¹

Especificación de caso de uso: Modificar inventario

Caso de Uso	Modificar inventario.
Objetivo	Realizar cambios en el inventario.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para modificar un inventario. • Haber consultado un inventario • No haber Generado el inventario inicial.

¹⁷⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁷¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Inventario inicial que se encuentra en la opción Procesos dentro del módulo de Inventario. • Elegir la opción Modificar del producto. • Ingresar el costo y la cantidad del producto • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de que los datos han sido modificados correctamente.

Tabla 7.50. Especificación de caso de uso: Modificar inventario ¹⁷²

Especificación de caso de uso: Imprimir inventario

Caso de Uso	Imprimir inventario.
Objetivo	Tener documentado el inventario de acuerdo a la bodega.
Actores	Usuario
Precondiciones	Haber ingresado un inventario.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Inventario inicial que se encuentra en la opción Procesos dentro del módulo de Inventario. • Elegir la Bodega para cargar los datos del Inventario. • Elegir la opción Imprimir. • Se despliega información en archivo .pdf. • Imprimir
Pos condiciones	El sistema imprime la información desplegada.

Tabla 7.51. Especificación de caso de uso: Imprimir inventario ¹⁷³

¹⁷² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁷³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración de Toma física

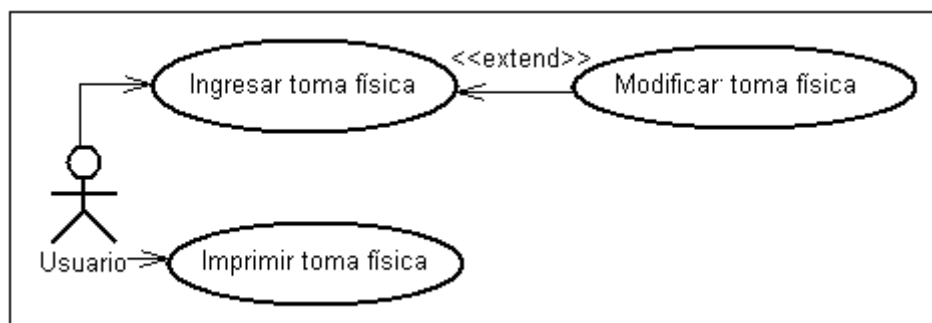


Figura 7.15. Caso de Uso del Sistema: Configuración de toma física ¹⁷⁴

Especificación de caso de uso: Ingresar toma física

Caso de Uso	Ingresar toma física
Objetivo	Ingresar al sistema los productos existentes en bodega a través de la toma física
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para ingresar a toma física. • Haber ingresado productos y creado sucursales.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Toma física que se encuentra en la opción Procesos dentro del módulo de Inventario. • Elegir la opción Nuevo • Seleccionar una bodega. • Ingresar el producto. • Ingresar la cantidad actual del producto existente. • Grabar la información. • Generar Toma física.
Pos	<ul style="list-style-type: none"> • El sistema compara las cantidades del

¹⁷⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

condiciones	producto de lo que se tiene en el sistema con lo ingresado en toma física, y muestra un mensaje de: Faltante, Sobrante o Conforme.
--------------------	--

Tabla 7.52. Especificación de caso de uso: Ingresar toma física ¹⁷⁵

Especificación de caso de uso: Modificar toma física

Caso de Uso	Modificar toma física.
Objetivo	Realizar cambios en las existencias de los productos.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para modificar la toma física. • Haber ingresado la toma física.
Acciones Básicas	<ul style="list-style-type: none"> • Elegir la opción Modificar del producto. • Ingresar la cantidad del producto. • Validar los datos. • Guardar los cambios.
Pos condiciones	<ul style="list-style-type: none"> • El sistema modificada los datos y muestra la información de faltante, sobrante o conforme.

Tabla 7.53. Especificación de caso de uso: Modificar toma física ¹⁷⁶

Especificación de caso de uso: Imprimir Toma Física

Caso de Uso	Imprimir toma física
Objetivo	Tener documentado la toma física que se realice en bodega.
Actores	Usuario
Precondiciones	Haber ingresado una toma física.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Toma Física que se encuentra en la opción Procesos dentro del módulo de Inventario.

¹⁷⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁷⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Elegir la Bodega para cargar los datos de la toma física. • Elegir la opción Imprimir. • Se despliega información en archivo .pdf. • Imprimir
Pos condiciones	El sistema imprime la información desplegada.

Tabla 7.54. Especificación de caso de uso: Imprimir inventario ¹⁷⁷

Caso de Uso: Configuración de Movimiento de bodega

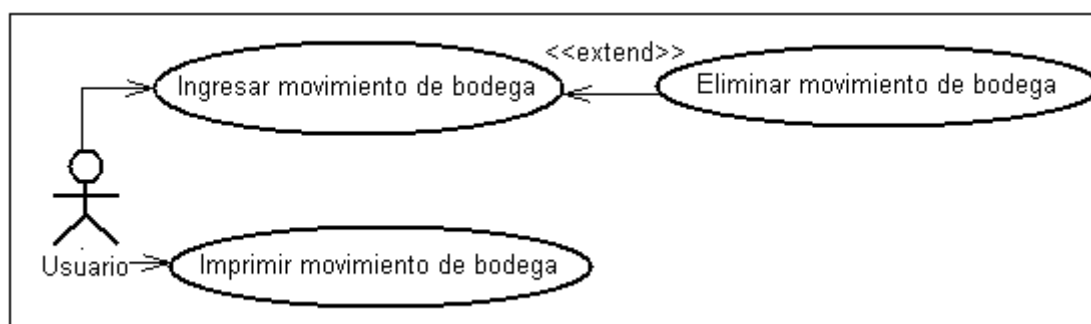


Figura 7.16. Caso de Uso del Sistema: Configuración de movimiento de bodega ¹⁷⁸

Especificación de caso de uso: Ingresar movimiento de bodega

Caso de Uso	Ingresar un movimiento de bodega
Objetivo	Realizar movimientos de bodega
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para ingresar un movimiento de bodega. • Haber ingresado productos.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Movimiento de bodega que se encuentra en la opción Procesos dentro del módulo de Inventario. • Elegir la opción Nuevo

¹⁷⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁷⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Elegir el tipo de transferencia • Ingresar la razón del movimiento. • Elegir bodega origen y bodega destino • Seleccionar el producto. • Ingresar la cantidad del producto a ser transferido. • Elegir la opción Movimiento de Bodega. • Validar datos • Una vez ingresado toda la información de los movimientos Grabar la información.
Pos condiciones	<ul style="list-style-type: none"> • Se guarda el movimiento de bodega

Tabla 7.55. Especificación de caso de uso: Ingresar movimiento de bodega ¹⁷⁹

Especificación de caso de uso: Eliminar movimiento de bodega

Caso de Uso	Eliminar movimiento de bodega
Objetivo	Borrar los datos de un movimiento de bodega que esté mal realizado.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para eliminar movimientos de bodega. • Haber realizado un movimiento de bodega.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Movimiento de bodega que se encuentra en la opción Procesos dentro del módulo de Inventario. • Elegir la opción Borrar del movimiento de bodega.
Pos condiciones	El sistema borra los datos correspondientes al movimiento de bodega elegido.

Tabla 7.56. Especificación de caso de uso: Eliminar movimiento de bodega ¹⁸⁰

¹⁷⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Imprimir Movimiento de bodega

Caso de Uso	Imprimir Movimiento de bodega
Objetivo	Tener documentado los movimientos de bodega realizados.
Actores	Usuario
Precondiciones	Haber ingresado una un movimiento de bodega.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Movimiento de bodega que se encuentra en la opción Procesos dentro del módulo de Inventario. • Elegir la opción Imprimir. • Se despliega información en archivo .pdf. • Imprimir
Pos condiciones	El sistema imprime la información desplegada.

Tabla 7.57. Especificación de caso de uso: Imprimir movimiento de bodega¹⁸¹

Caso de Uso: Configuración de Preproducción

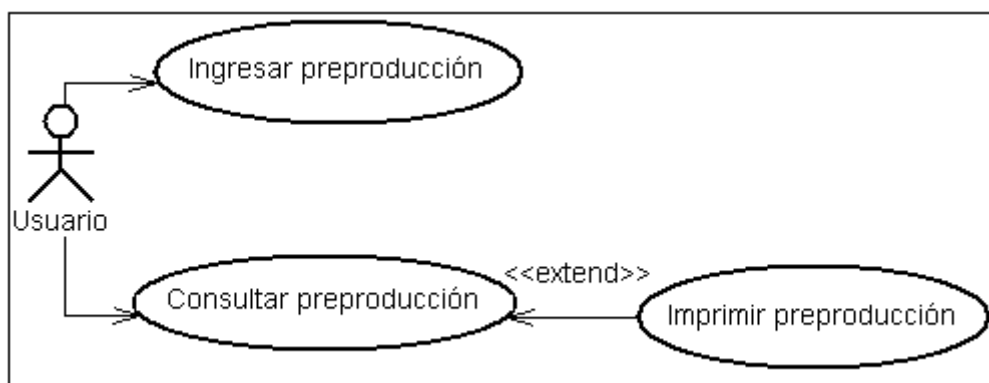


Figura 7.17. Caso de Uso del Sistema: Configuración de preproducción¹⁸²

Especificación de caso de uso: Ingresar preproducción

Caso de Uso	Ingresar preproducción
Objetivo	Realizar una preproducción

¹⁸⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁸¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁸² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para ingresar una preproducción. • Haber creado recetas.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Preproducción que se encuentra en la opción Producción dentro de la opción Procesos del módulo de Inventario. • Elegir la opción Nuevo. • Seleccionar la bodega. • Elegir Consultar. • Seleccionar la receta. • Ingresar la cantidad a producir. • Mandar a Producir • Validar la información • Grabar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de preproducción registrada.

Tabla 7.58. Especificación de caso de uso: Ingresar preproducción ¹⁸³

Especificación de caso de uso: Consultar preproducción

Caso de Uso	Consultar preproducción
Objetivo	Visualizar una preproducción realizada.
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para consultar una preproducción.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Preproducción que se encuentra en la opción Producción dentro de la opción Procesos del módulo de Inventario. • Elegir la opción Imprimir

¹⁸³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Despliega las reproducciones
Pos condiciones	El sistema muestra la información de reproducciones realizadas.

Tabla 7.59. Especificación de caso de uso: Consultar reproducción ¹⁸⁴

Especificación de caso de uso: Imprimir reproducción

Caso de Uso	Imprimir reproducción
Objetivo	Tener documentado la reproducción realizada
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para imprimir una reproducción.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Reproducción que se encuentra en la opción Producción dentro de la opción Procesos del módulo de Inventario. • Elegir la opción Imprimir • Despliega las reproducciones • Elegir la opción Imprimir de la reproducción realizada. • Se despliega información en archivo .pdf. • Imprimir
Pos condiciones	El sistema imprime la información desplegada.

Tabla 7.60. Especificación de caso de uso: Imprimir reproducción ¹⁸⁵

¹⁸⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁸⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración de Postproducción

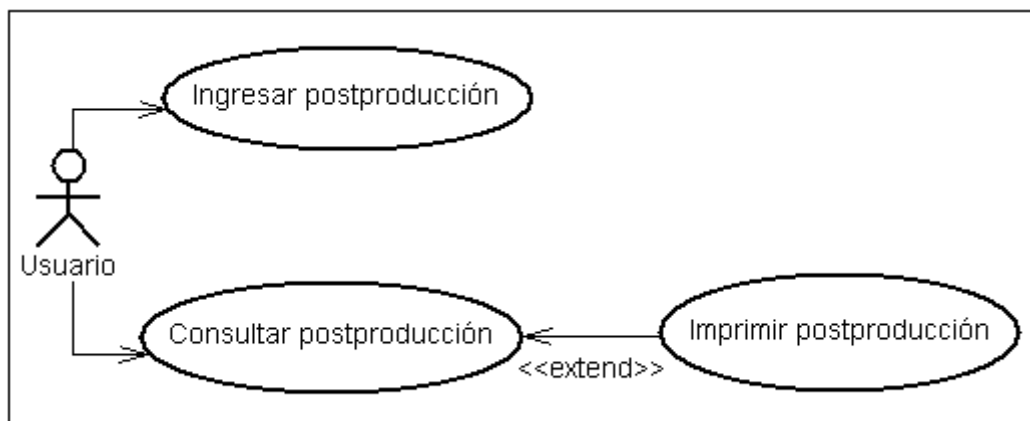


Figura 7.18. Caso de Uso del Sistema: Configuración de postproducción ¹⁸⁶

Especificación de caso de uso: Ingresar postproducción

Caso de Uso	Ingresar postproducción
Objetivo	Realizar una postproducción
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para ingresar una postproducción. • Haber creado recetas.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Postproducción que se encuentra en la opción Producción dentro de la opción Procesos del módulo de Inventario. • Elegir la opción Nuevo. • Seleccionar la bodega. • Elegir Consultar. • Seleccionar la receta. • Ingresar la cantidad a producir. • Ingresar el sobrante y el desperdicio. • Mandar a Producir • Validar la información • Grabar la información.

¹⁸⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de que la postproducción está registrada.
------------------------	---

Tabla 7.61. Especificación de caso de uso: Ingresar postproducción ¹⁸⁷

Especificación de caso de uso: Consultar postproducción

Caso de Uso	Consultar postproducción
Objetivo	Visualizar una postproducción realizada.
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para consultar una postproducción.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Postproducción que se encuentra en la opción Producción dentro de la opción Procesos del módulo de Inventario. • Elegir la opción Imprimir • Visualizar las postproducciones.
Pos condiciones	El sistema muestra la información de postproducciones realizadas.

Tabla 7.62. Especificación de caso de uso: Consultar postproducción ¹⁸⁸

Especificación de caso de uso: Imprimir postproducción

Caso de Uso	Imprimir postproducción
Objetivo	Tener documentado la postproducción.
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para imprimir una postproducción.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a Postproducción que se encuentra en la opción Producción dentro de la opción Procesos del módulo de Inventario.

¹⁸⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁸⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<ul style="list-style-type: none"> • Elegir la opción Imprimir • Despliega las postproducciones • Elegir la opción Imprimir de la postproducción realizada. • Se despliega información en archivo .pdf. • Imprimir
Pos condiciones	El sistema imprime la información desplegada.

Tabla 7.63. Especificación de caso de uso: Imprimir postproducción ¹⁸⁹

Módulo Facturación

Caso de Uso: Configuración de Clientes

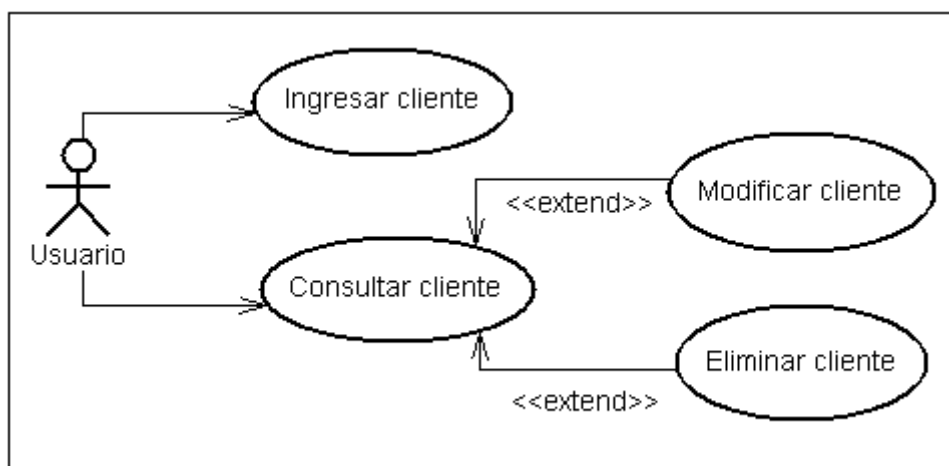


Figura 7.19. Caso de Uso del Sistema: Configuración de clientes ¹⁹⁰

Especificación de caso de uso: Ingresar cliente

Caso de Uso	Ingresar cliente
Objetivo	Crear un nuevo cliente
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para el ingreso de los datos del cliente.

¹⁸⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁹⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Clientes que se encuentra en Datos Generales dentro del módulo de Facturación. • Elegir la opción Nuevo. • Ingresar los datos del cliente: nombres, apellidos, ruc, razón social, dirección, teléfono, email, tipo de precio. • Validar cada uno de los datos ingresados. • Guardar la información.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra un mensaje de registro ingresado.

Tabla 7.64. Especificación de caso de uso: Ingresar cliente ¹⁹¹

Especificación de caso de uso: Consultar cliente

Caso de Uso	Consultar cliente
Objetivo	Visualizar los datos que el cliente posee.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para consultar los datos del cliente.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Clientes que se encuentra en Datos Generales dentro del módulo de Facturación. • Visualizar los datos de los clientes.
Pos condiciones	El sistema muestra la información de los clientes.

Tabla 7.65. Especificación de caso de uso: Consultar cliente ¹⁹²

Especificación de caso de uso: Modificar cliente

Caso de Uso	Modificar cliente
Objetivo	Realizar cambios en los datos del cliente.

¹⁹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para modificar los datos del cliente.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Clientes que se encuentra en Datos Generales dentro del módulo de Facturación. • Elegir la opción Modificar del cliente. • Modificar los datos del cliente. • Validar los datos. • Guardar los cambios.
Pos condiciones	El sistema muestra un mensaje de Registro Actualizado.

Tabla 7.66. Especificación de caso de uso: Modificar cliente ¹⁹³

Especificación de caso de uso: Eliminar cliente

Caso de Uso	Eliminar cliente
Objetivo	Eliminar un cliente del sistema.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para eliminar los datos del cliente. • El cliente no debe tener registros de ventas
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción de Clientes que se encuentra en Datos Generales dentro del módulo de Facturación. • Elegir la opción Eliminar del cliente.
Pos condiciones	<ul style="list-style-type: none"> • Si el cliente no tiene ningún registro, el sistema muestra un mensaje de que los datos del cliente han sido eliminados.

Tabla 7.67. Especificación de caso de uso: Eliminar cliente ¹⁹⁴

¹⁹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Caso de Uso: Configuración de Factura

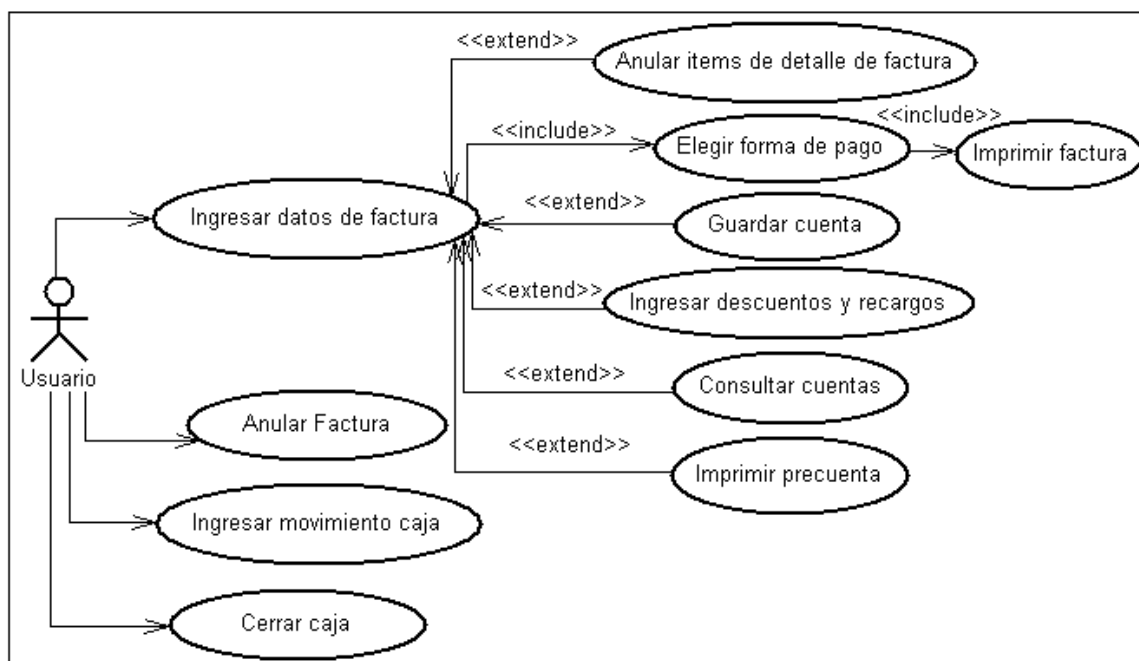


Figura 7.20. Caso de Uso del Sistema: Configuración de Factura ¹⁹⁵

Especificación de caso de uso: Ingresar datos de factura

Caso de Uso	Ingresar datos de factura
Objetivo	Realizar el ingreso de datos del cliente y de los productos solicitados.
Actores	Usuario
Precondiciones	El usuario debe tener un perfil con los permisos necesarios para el ingresar a Facturación
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Factura que se encuentra en Procesos dentro del módulo de Facturación. • Ingresar el número de cuenta. • Elegir la opción Buscar cliente. • Seleccionar el cliente. • Elegir la opción Buscar Producto • Ingresar la cantidad del producto • Seguir ingresando los productos hasta tener el

¹⁹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	listado total.
Pos condiciones	<ul style="list-style-type: none"> • El sistema muestra los productos a facturar.

Tabla 7.68. Especificación de caso de uso: Ingresar Factura ¹⁹⁶

Especificación de caso de uso: Anular ítems de detalle de factura.

Caso de Uso	Anular ítems de detalle de factura.
Objetivo	Eliminar los ítems de detalle de factura que han sido ingresados de forma incorrecta.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para eliminar los datos del detalle de la Factura. • Estar en el proceso de facturación.
Acciones Básicas	<ul style="list-style-type: none"> • Seleccionar la opción Anular Producto del ítem a anular
Pos condiciones	El sistema elimina el producto del detalle de la factura.

Tabla 7.69. Especificación de caso de uso: Anular ítems del detalle de factura ¹⁹⁷

Especificación de caso de uso: Elegir Forma de Pago

Caso de Uso	Elegir Forma de Pago
Objetivo	Realizar el pago de la factura
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Estar en el proceso de facturación • Haber ingresado todos los productos de la factura.
Acciones Básicas	<ul style="list-style-type: none"> • Seleccionar la opción Forma de pago • El sistema despliega el valor a pagar • Ingresar el valor en cada forma de pago en

¹⁹⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁹⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<p>caso de ser necesario.</p> <ul style="list-style-type: none"> • Elegir la opción Grabar
Pos condiciones	El sistema muestra un mensaje de que la factura a sido totalizada.

Tabla 7.70. Especificación de caso de uso: Elegir Forma de Pago ¹⁹⁸

Especificación de caso de uso: Imprimir Factura

Caso de Uso	Imprimir factura
Objetivo	Tener documentada la factura.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para imprimir Factura • Haber elegido Forma de Pago
Acciones Básicas	<ul style="list-style-type: none"> • Seleccionar la opción Imprimir. • Se muestra la factura en formato .pdf. • Imprimir
Pos condiciones	El sistema imprime la factura.

Tabla 7.71. Especificación de caso de uso: Imprimir Factura ¹⁹⁹

Especificación de caso de uso: Guardar cuenta

Caso de Uso	Guardar cuenta
Objetivo	Poner en espera la cuenta para luego realizar el pago.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para guardar cuentas en espera. • Estar en el proceso de facturación.
Acciones	<ul style="list-style-type: none"> • Ingresar el número de cuenta para que sea

¹⁹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

¹⁹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Básicas	guardada. <ul style="list-style-type: none"> • Seleccionar la opción Guardar cuenta.
Pos condiciones	El sistema guarda la cuenta para que luego sea recuperada.

Tabla 7.72. Especificación de caso de uso: Guardar cuenta en espera ²⁰⁰

Especificación de caso de uso: Ingresar descuentos y recargos

Caso de Uso	Ingresar descuentos y recargos
Objetivo	Ingresar a la factura un descuento o recargo
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios realizar descuentos y recargos en la factura. • Haber ingresado los datos de la factura.
Acciones Básicas	<ul style="list-style-type: none"> • Elegir la opción Descuentos y recargos. • Dar clic en la opción Seleccionar del descuento o recargo.
Pos condiciones	El sistema muestra el valor del descuento o recargo en los totales de la factura.

Tabla 7.73. Especificación de caso de uso: Ingresar descuentos y recargos ²⁰¹

Especificación de caso de uso: Consultar cuentas

Caso de Uso	Consultar cuentas
Objetivo	Realizar la recuperación de una cuenta en espera.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para consultar cuentas • Haber guardado la cuenta en espera.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Factura que se encuentra en Procesos dentro del módulo de

²⁰⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁰¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	<p>Facturación.</p> <ul style="list-style-type: none"> • Elegir la opción consultar cuentas. • Dar clic en la opción Seleccionar de la cuenta a recuperar.
Pos condiciones	El sistema devuelve todos los valores de la factura.

Tabla 7.74. Especificación de caso de uso: Consultar cuentas.²⁰²

Especificación de caso de uso: Imprimir pre-cuenta

Caso de Uso	Imprimir precuenta
Objetivo	Tener en papel la cuenta del cliente.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para imprimir una precuenta. • Haber guardado la cuenta.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Factura que se encuentra en Procesos dentro del módulo de Facturación. • Elegir la opción consultar cuentas. • Dar clic en la opción Pre-cuenta de la cuenta a imprimir
Pos condiciones	El sistema imprime la pre-cuenta.

Tabla 7.75. Especificación de caso de uso: Imprimir pre-cuenta²⁰³

Especificación de caso de uso: Anular Facturas

Caso de Uso	Anular Facturas
Objetivo	Eliminar una factura del registro
Actores	Usuario

²⁰² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁰³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para anular facturas. • Tener facturas guardadas.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Anular Facturas que se encuentra en Procesos dentro del módulo de Facturación. • Seleccionar la factura que va a ser anulada. • Elegir la opción Anular Factura.
Pos condiciones	El sistema muestra un mensaje con el número de factura anulada.

Tabla 7.76. Especificación de caso de uso: Anular Facturas.²⁰⁴

Especificación de caso de uso: Ingresar movimiento de caja

Caso de Uso	Ingresar movimiento de caja
Objetivo	Realizar un movimiento de caja
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un perfil con los permisos necesarios para realizar un movimiento de caja.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Movimientos de Caja que se encuentra en Procesos dentro del módulo de Facturación. • Seleccionar el tipo de movimiento: ingreso o egreso. • Ingresar el valor del movimiento.
Pos condiciones	El sistema muestra un mensaje de que el Registro fue ingresado.

Tabla 7.77. Especificación de caso de uso: Ingresar movimiento de caja.²⁰⁵

²⁰⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁰⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Especificación de caso de uso: Cerrar Caja

Caso de Uso	Cerrar Caja
Objetivo	Realizar el cuadro de caja.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> El usuario debe tener un perfil con los permisos necesarios para realizar el cierre de caja.
Acciones Básicas	<ul style="list-style-type: none"> Ingresar a la opción Cuadre de Caja que se encuentra en Procesos dentro del módulo de Facturación. Ingresar los valores correspondientes al dinero que se tenga en la caja.
Pos condiciones	El sistema mostrará un mensaje de que el cuadro de caja ha sido registrado

Tabla 7.78. Especificación de caso de uso: Cerrar caja²⁰⁶

Caso de Uso: Generar reportes

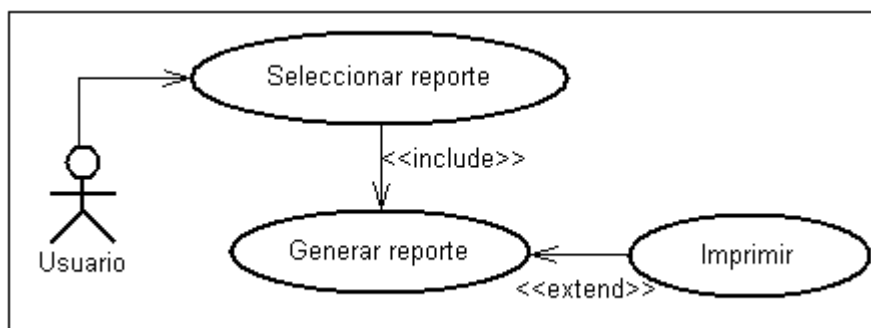


Figura 7.21. Caso de Uso del Sistema: Generar reportes²⁰⁷

Especificación de caso de uso: Generar reporte

Caso de Uso	Generar reporte
Objetivo	Generar un reporte.
Actores	Usuario

²⁰⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁰⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para generar los reportes.
Acciones Básicas	<ul style="list-style-type: none"> • Ingresar a la opción Reportes que se encuentra ya sea en el módulo de Administración, Inventario o Facturación. • Seleccionar el reporte • Validar los datos • Generar el reporte
Pos condiciones	El sistema muestra la información del reporte seleccionado.

Tabla 7.79. Especificación de caso de uso: Seleccionar reporte ²⁰⁸

Especificación de caso de uso: Imprimir reporte

Caso de Uso	Imprimir reporte
Objetivo	Imprimir el reporte generado
Actores	Usuario
Precondiciones	Ingresar al sistema con un usuario que tenga un perfil con los permisos correspondientes para generar los reportes
Acciones Básicas	<ul style="list-style-type: none"> • Haber generado el reporte • Imprimir el reporte
Pos condiciones	El sistema imprimirá el reporte seleccionado.

Tabla 7.80. Especificación de caso de uso: Imprimir reporte ²⁰⁹

²⁰⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁰⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.1.2. MODELO DE INTERFACES

Las interfaces que se presentan en el sistema poseen el mismo esquema.

La primera interfaz del sistema está compuesta por: encabezado, menú principal y pie de página como se muestra en la de la Figura 7.22.

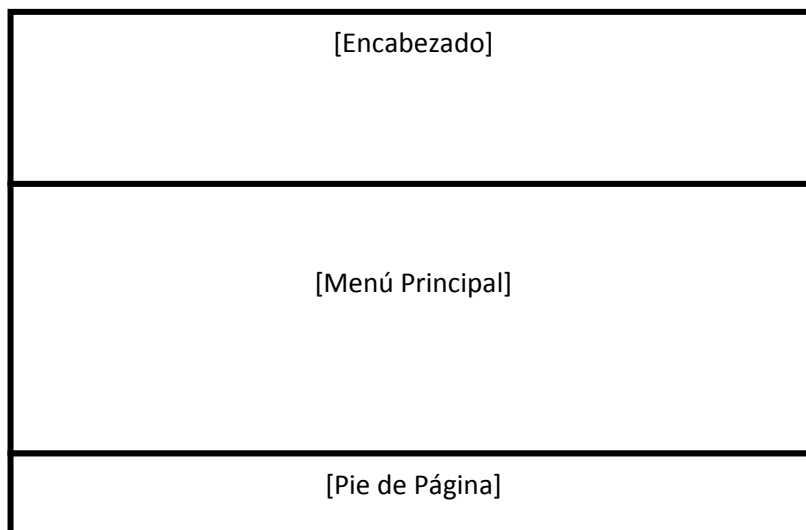


Figura 7.22. Modelo de Interfaz -1 ²¹⁰

En el menú principal se tiene el acceso a los 3 módulos del sistema, al ingresar a cada uno de ellos se muestra una interfaz con el esquema de la Figura 7.23.

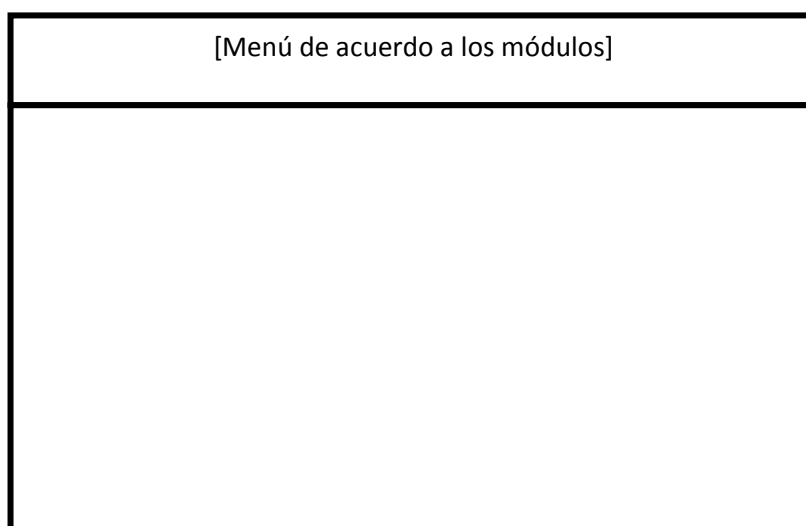


Figura 7.23. Modelo de Interfaz -2 ²¹¹

²¹⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Al ingresar a cada uno de los ítems del menú de acuerdo a los módulos, se presentan las interfaces con el esquema de la Figura 7.24.

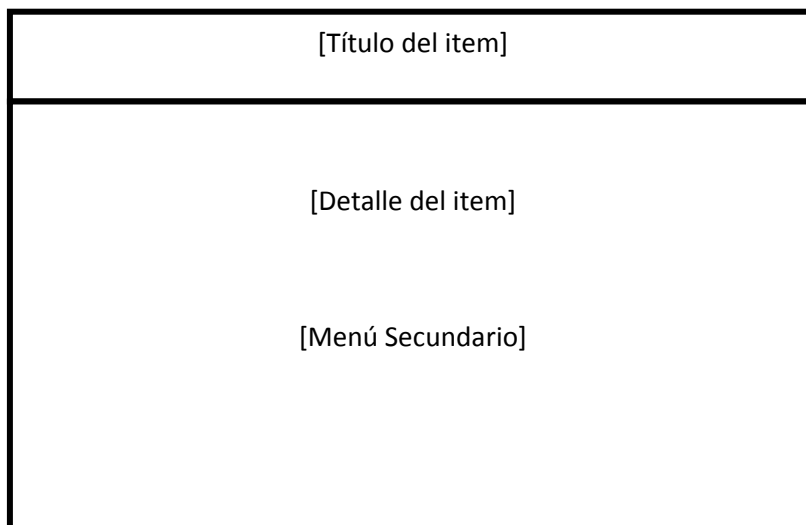
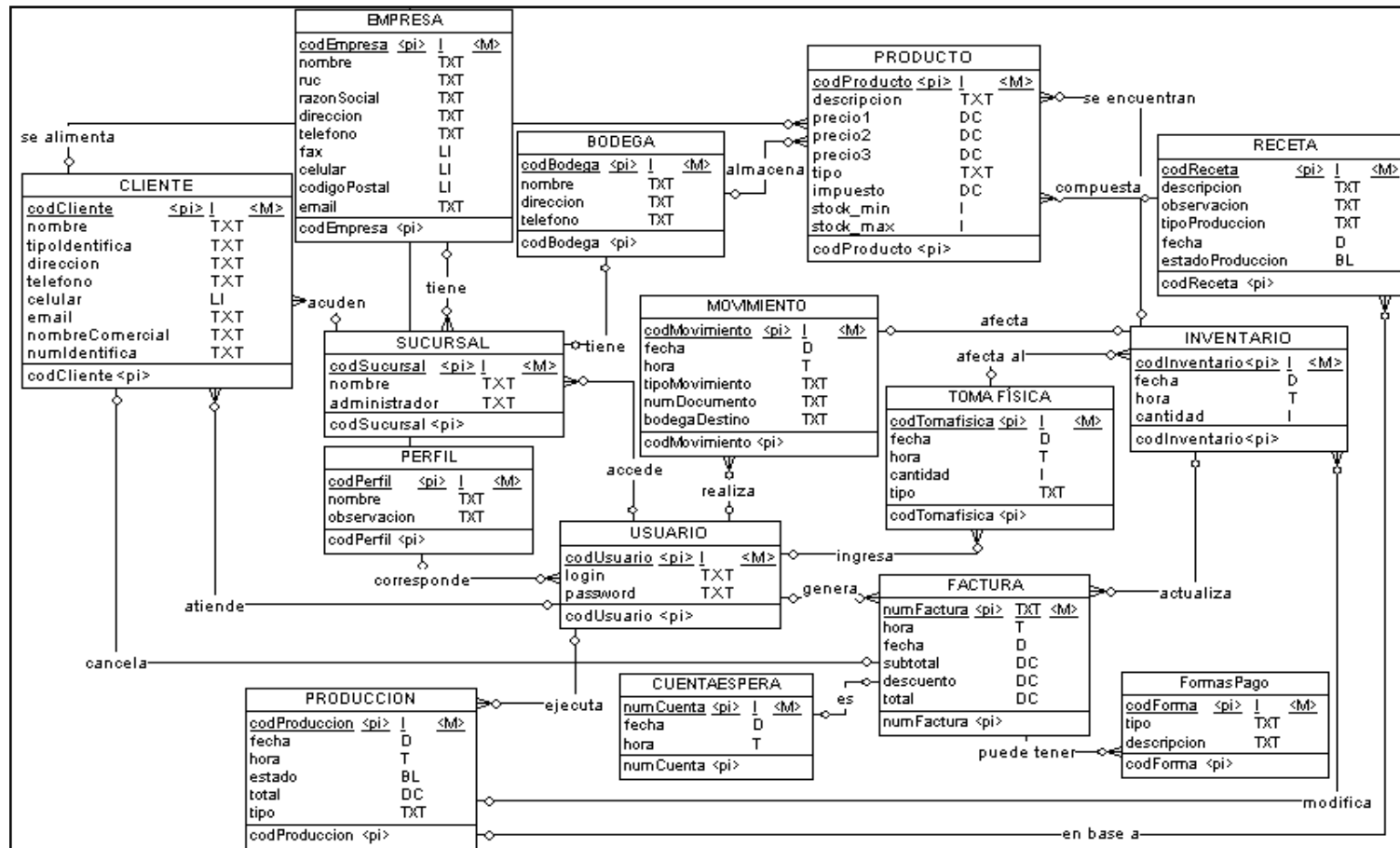


Figura 7.24. Modelo de Interfaz -3²¹²

²¹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²¹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.1.3. MODELO DEL DOMINIO

Figura 7.25. Modelo de Dominio ²¹³²¹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.2. MODELO DE ANÁLISIS

2.2.1. ARQUITECTURA DE CLASES

Módulo Administración

Estructura Organizacional

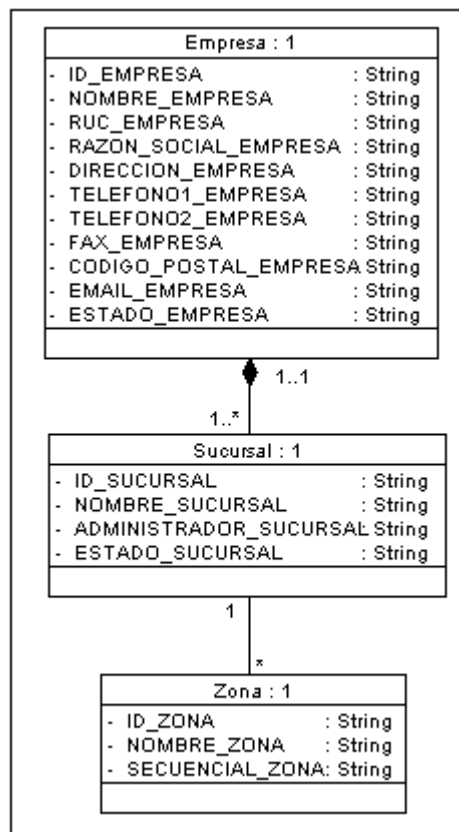


Figura 7.26. Diagrama: Estructura Organizacional ²¹⁴

²¹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Ubicación Geográfica

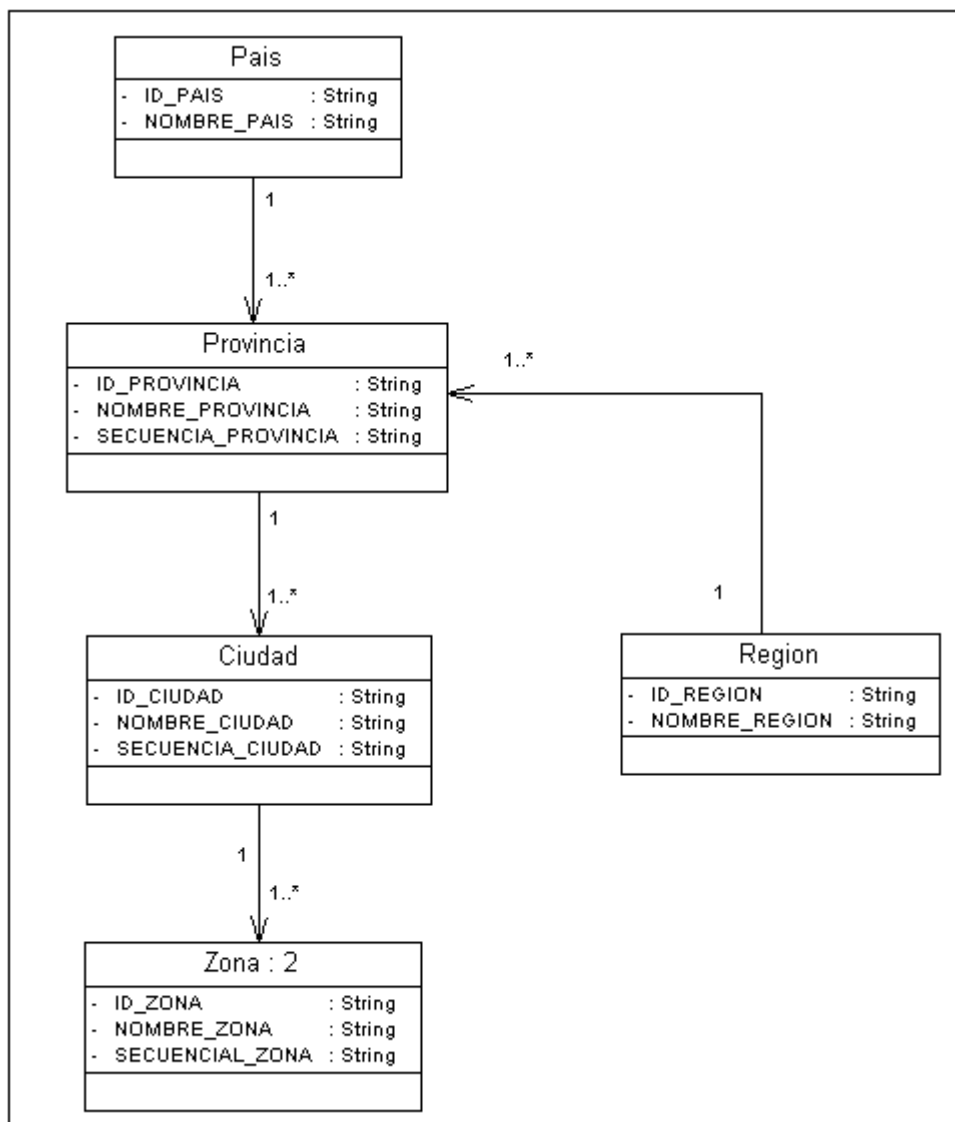


Figura 7.27. Diagrama: Ubicación Geográfica ²¹⁵

²¹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Perfil de Usuario

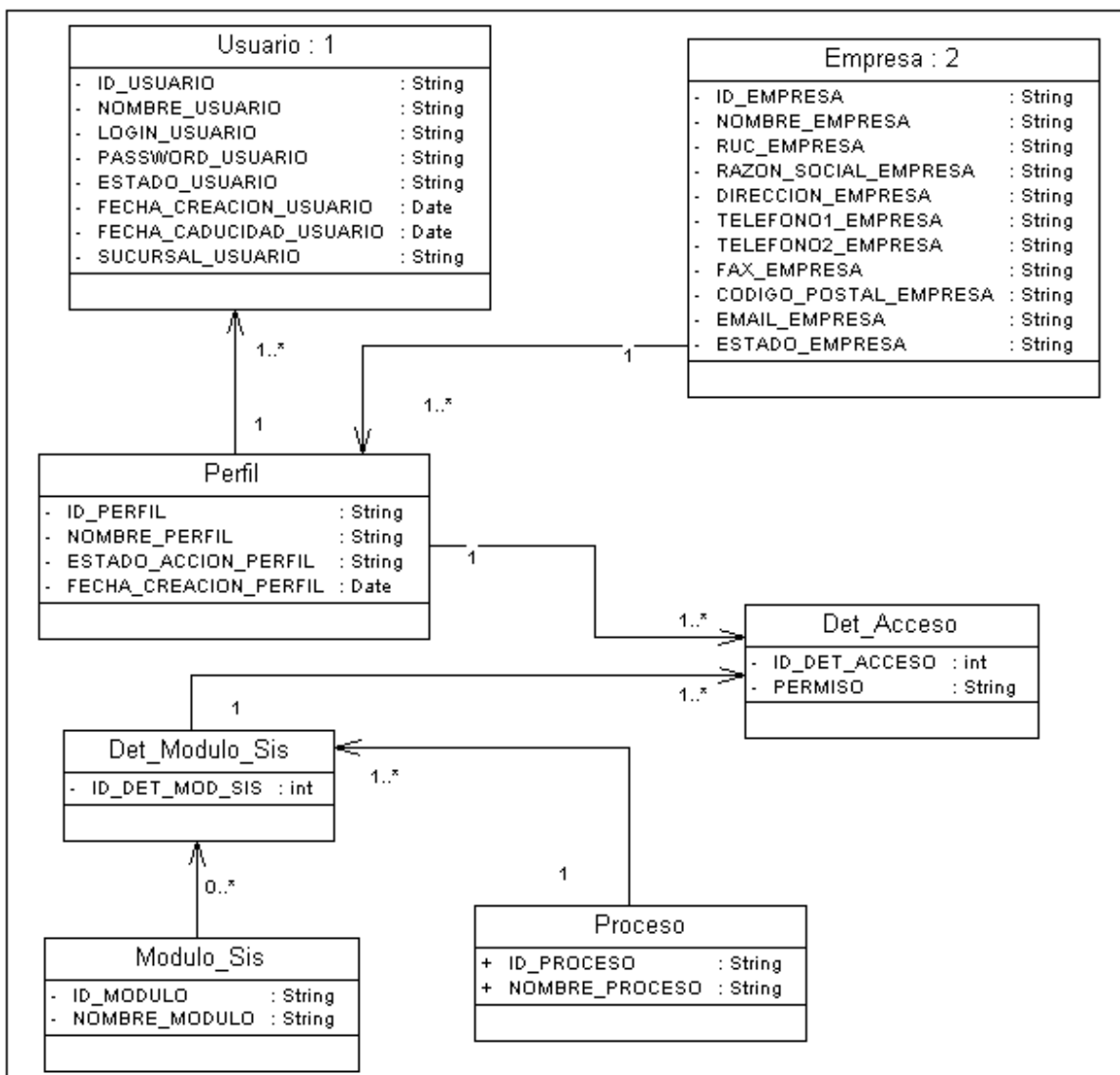


Figura 7.28. Diagrama: Perfil de Usuario ²¹⁶

²¹⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

IVA, Recargos y Descuentos

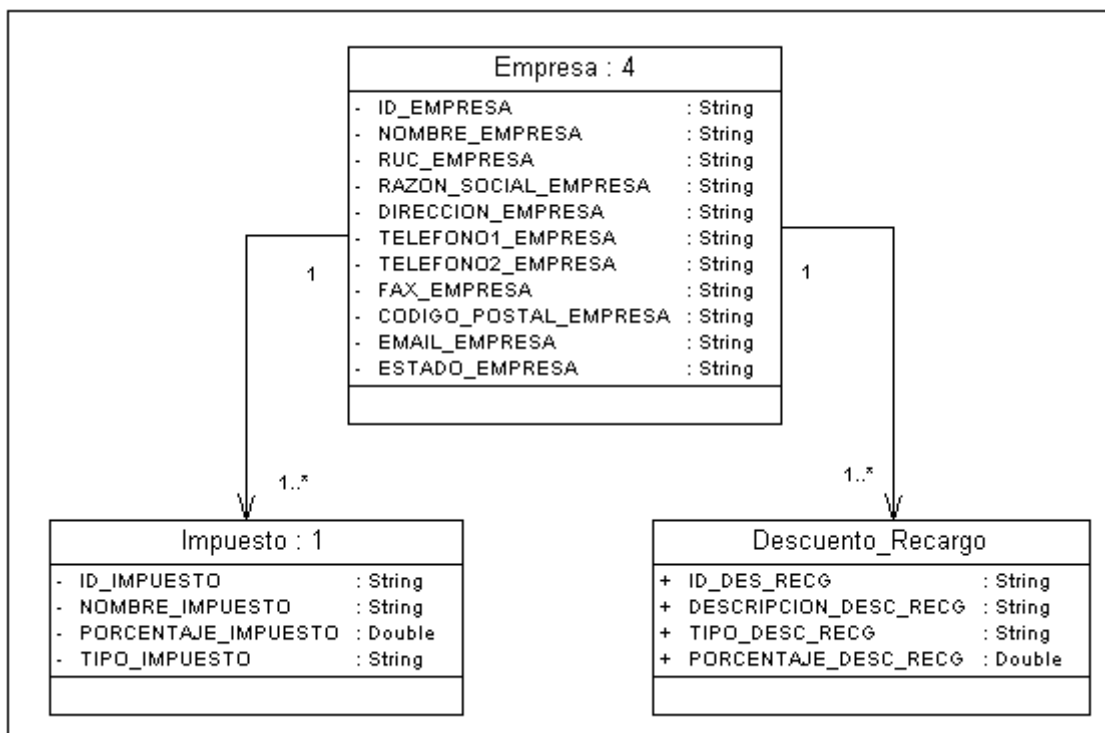


Figura 7.29. Diagrama: Recargos y Descuentos ²¹⁷

²¹⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Inventario

Bodega

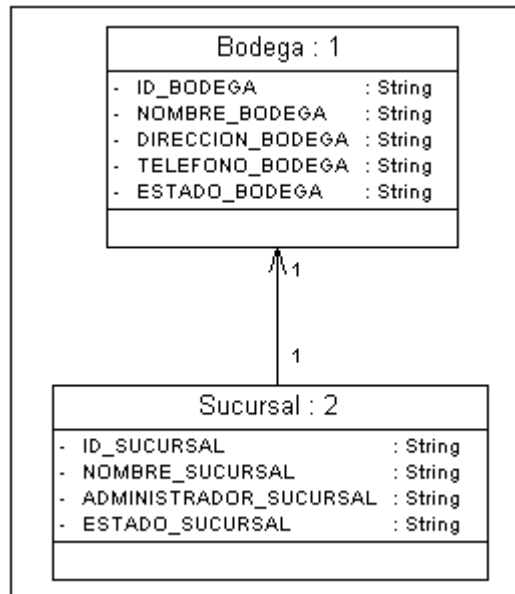


Figura 7.30. Diagrama: Bodega²¹⁸

²¹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Producto

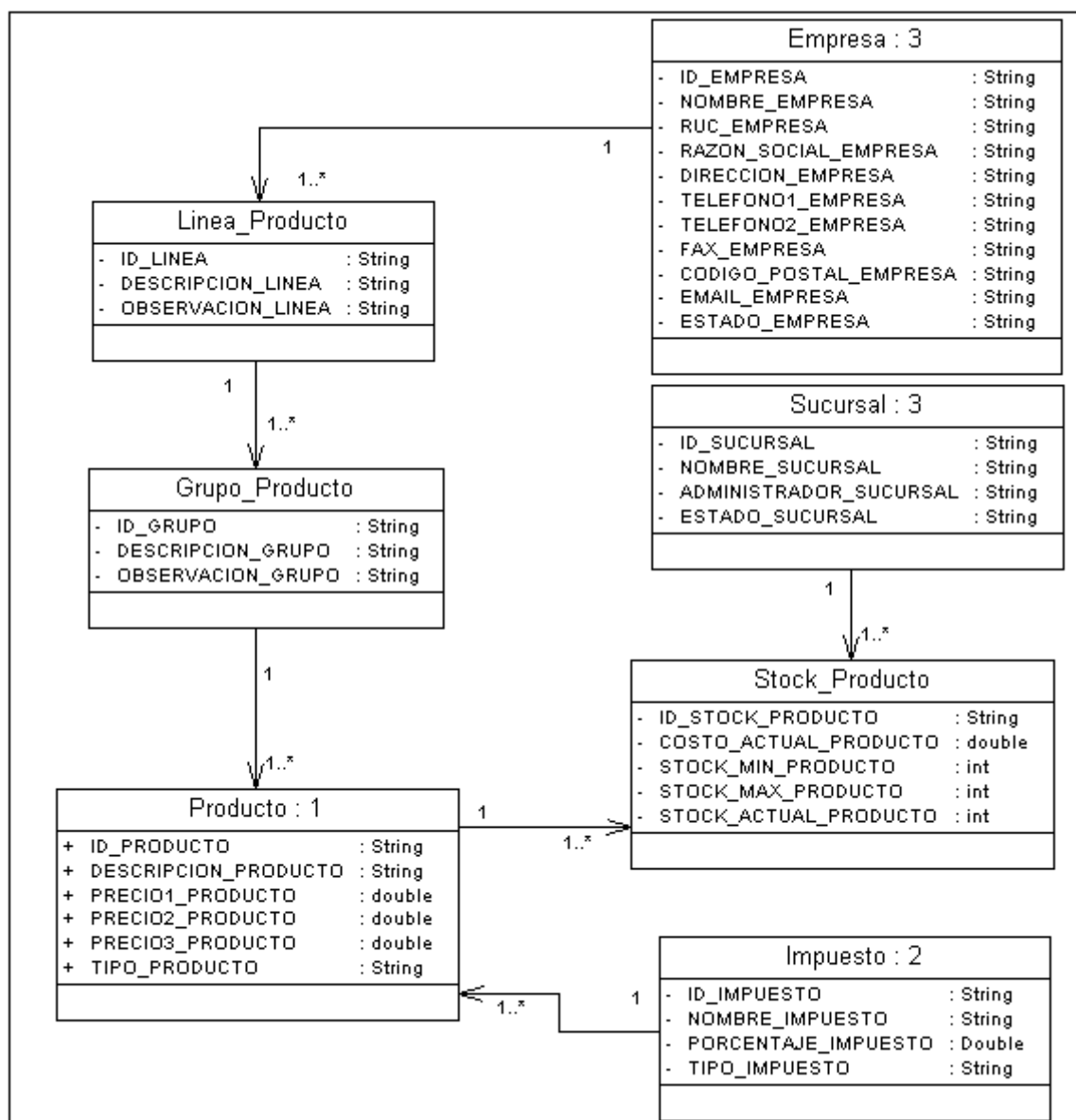


Figura 7.31. Diagrama: Producto ²¹⁹

²¹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Receta

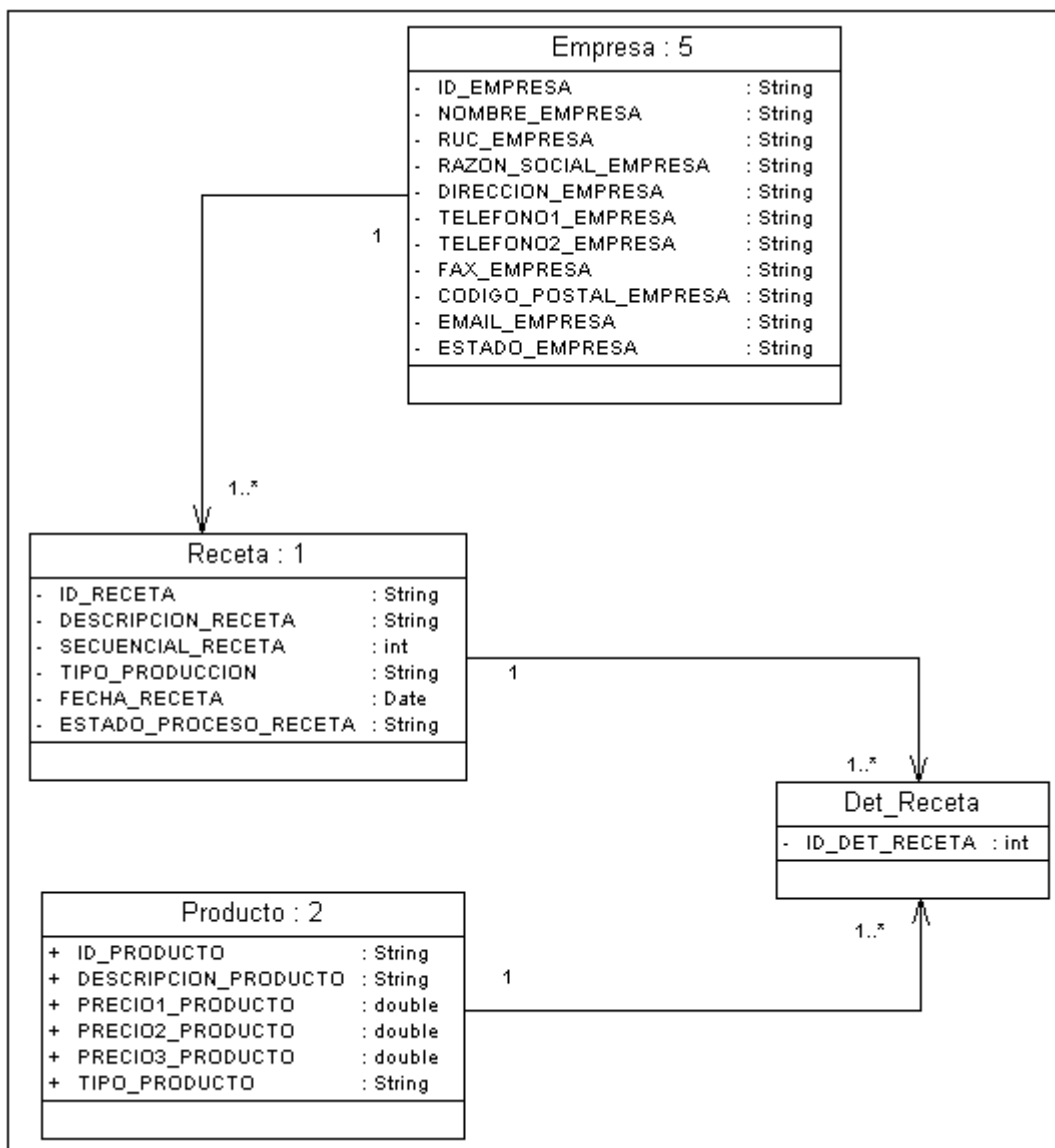


Figura 7.32. Diagrama: Receta²²⁰

²²⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Producción

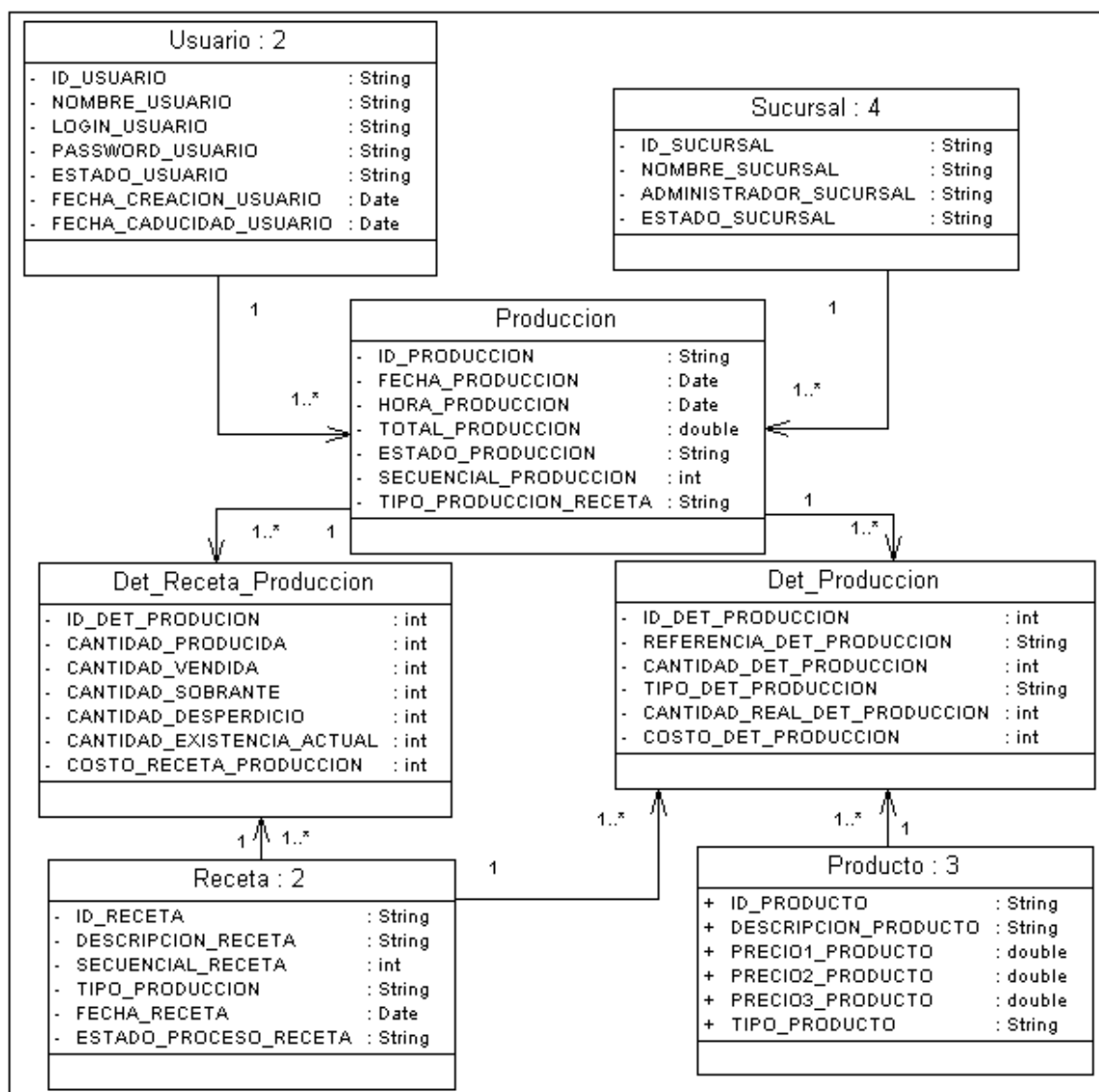


Figura 7.33. Diagrama: Producción ²²¹

²²¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Inventario

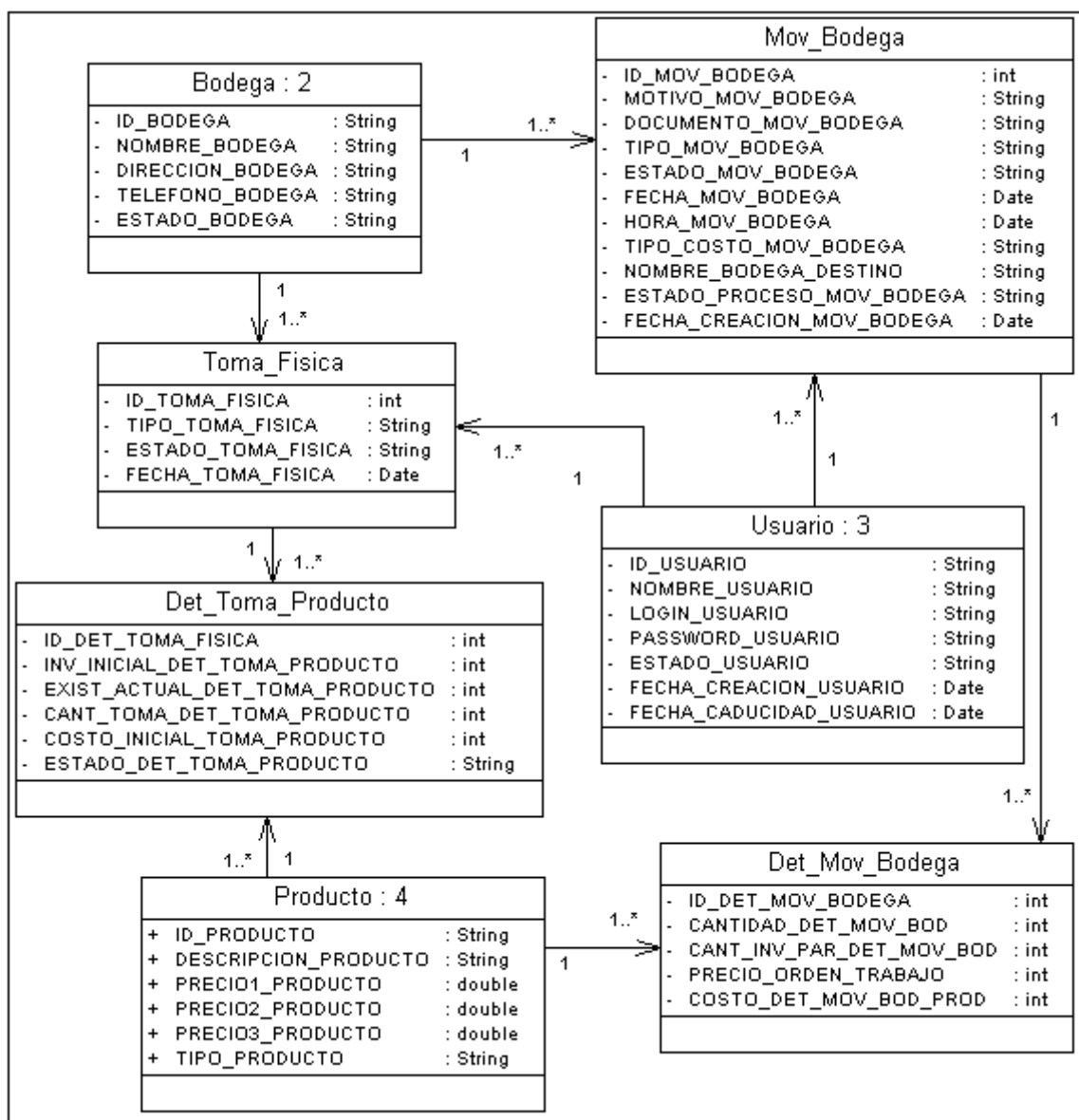


Figura 7.34. Diagrama: Inventario ²²²

²²² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Kardex

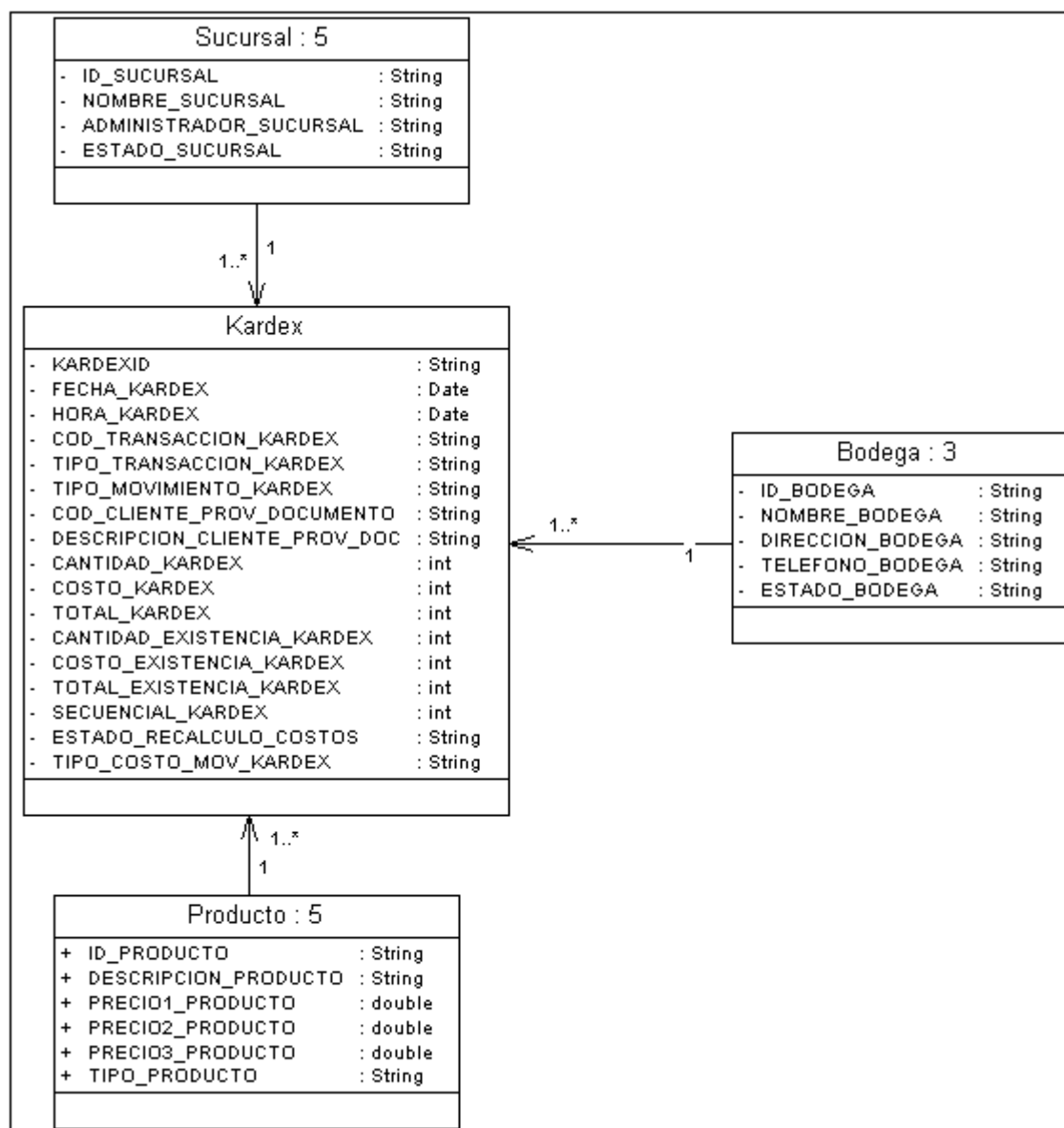
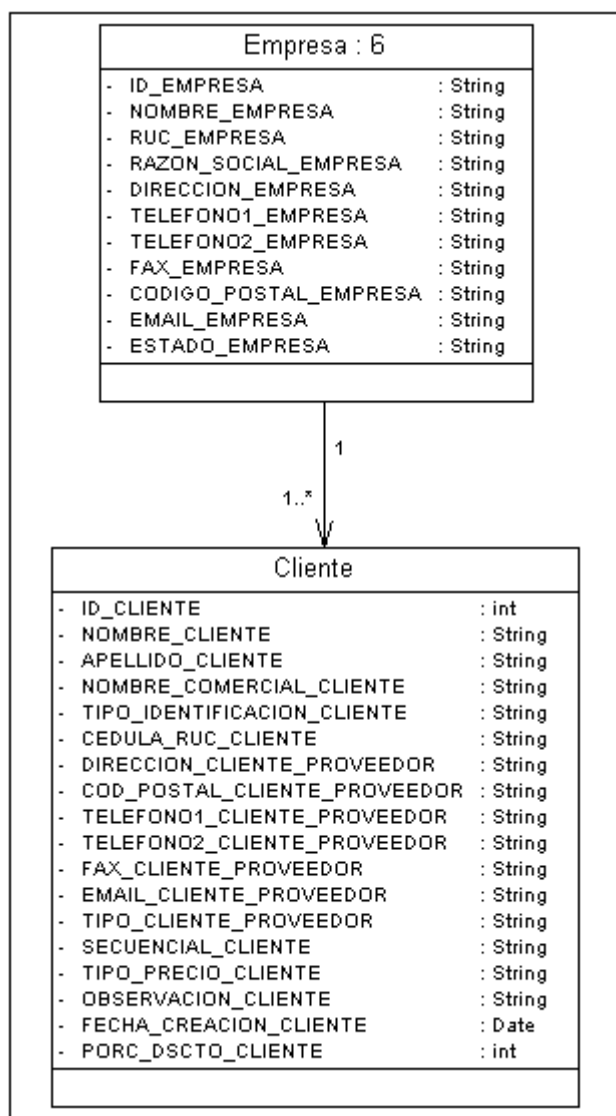


Figura 7.35. Diagrama: Kardex²²³

²²³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Facturación

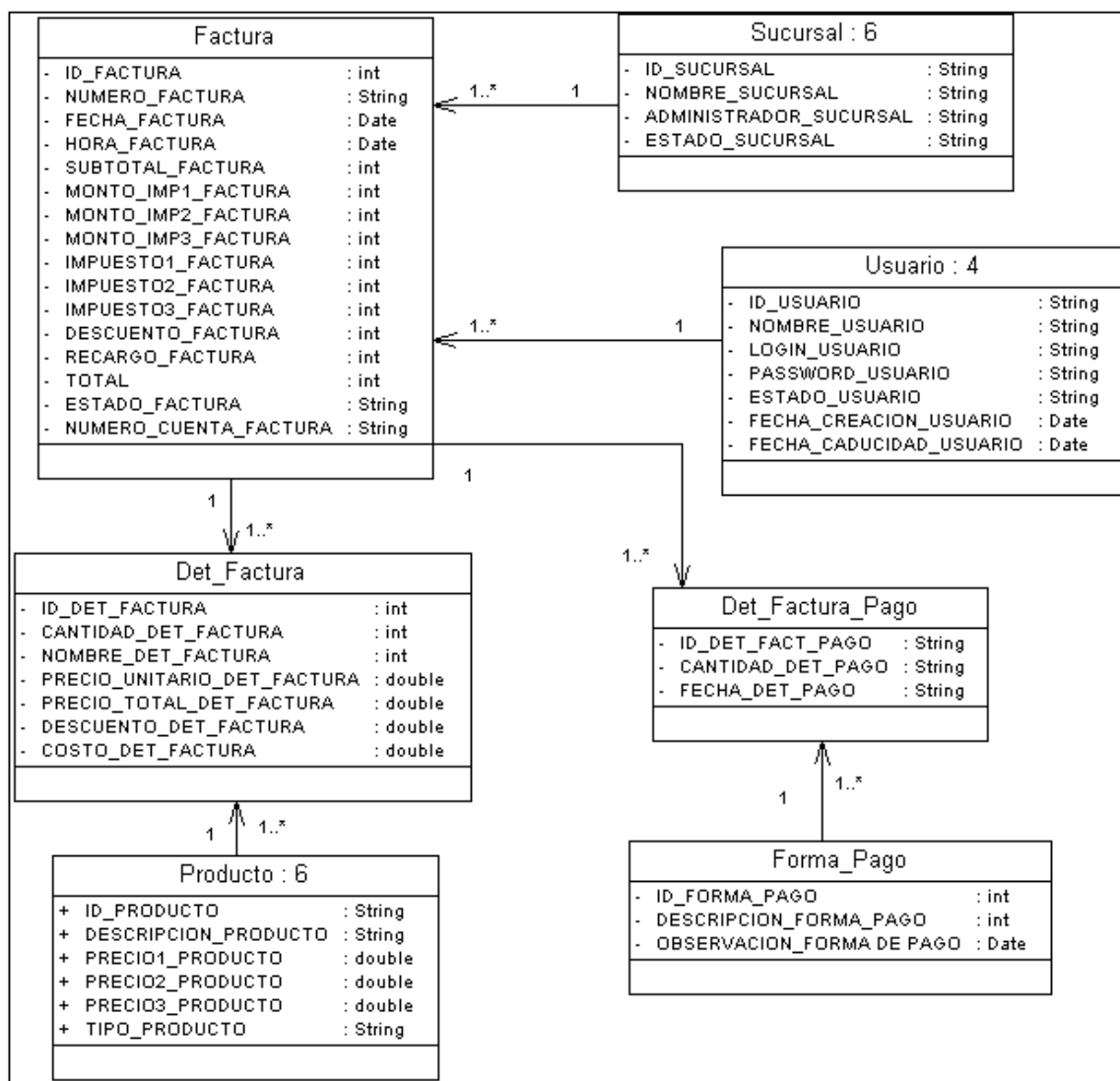
Cliente



*Figura 7.36. Diagrama: Cliente*²²⁴

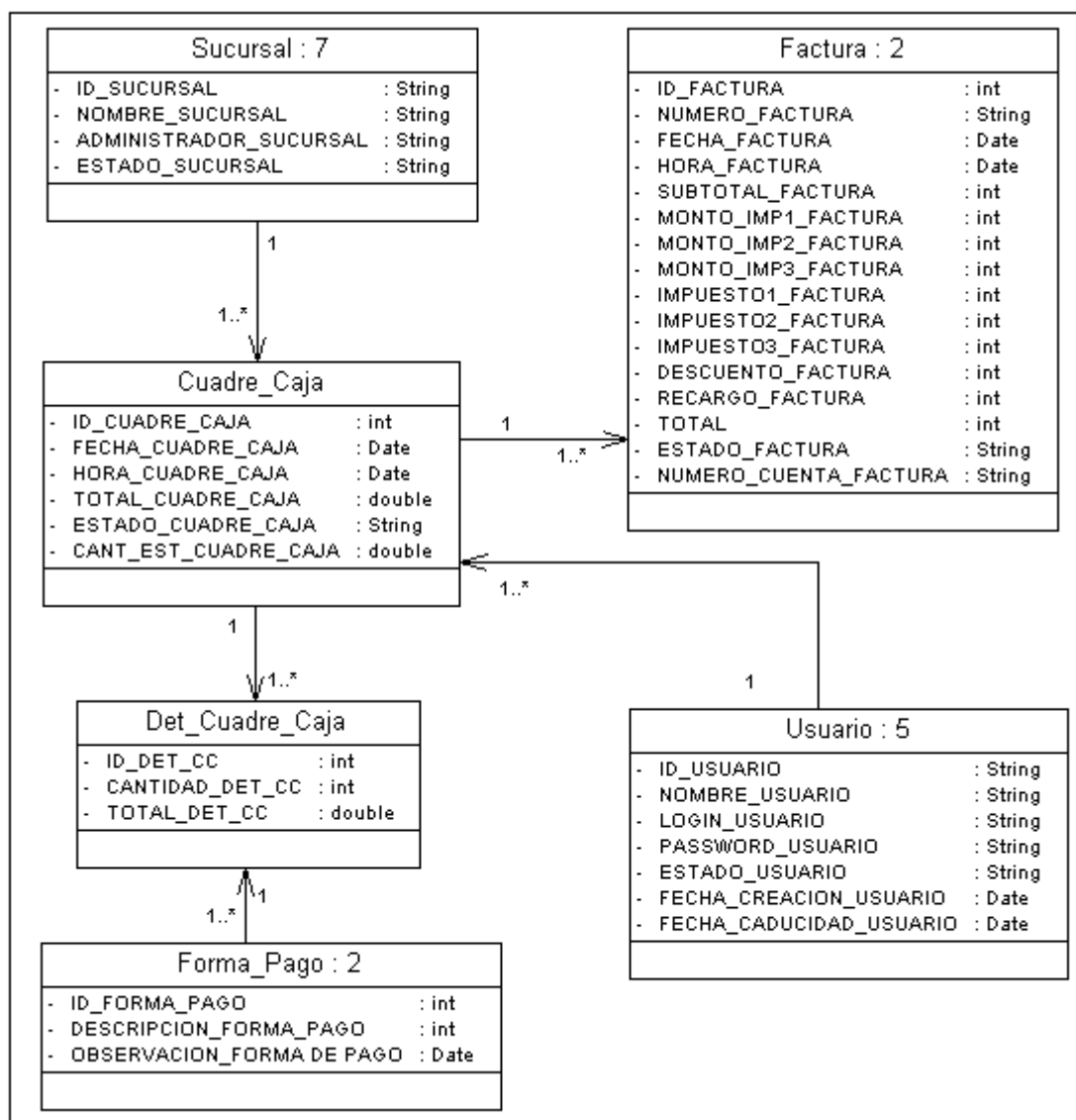
²²⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Factura

Figura 7.37. Diagrama: Factura²²⁵

²²⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Cuadre de Caja

Figura 7.38. Diagrama: Cuadre de Caja ²²⁶

²²⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Movimiento de Caja

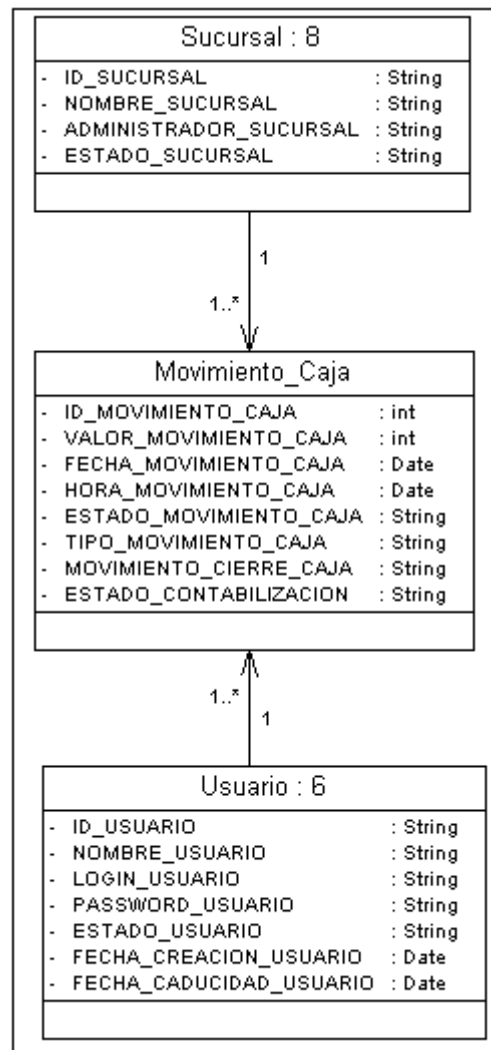


Figura 7.39. Diagrama: Movimiento de Caja ²²⁷

²²⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.2.2. DIAGRAMAS DE SECUENCIA

Módulo de Administración

Diagrama de secuencia: Ingresar empresa

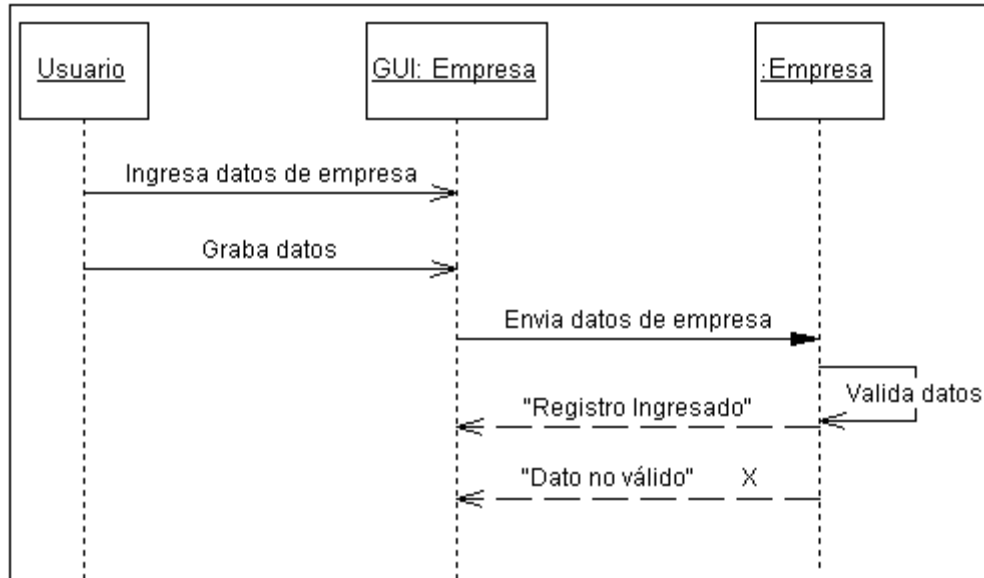


Figura 7.40. Diagrama de secuencia: Ingresar empresa ²²⁸

Diagrama de secuencia: Consultar empresa

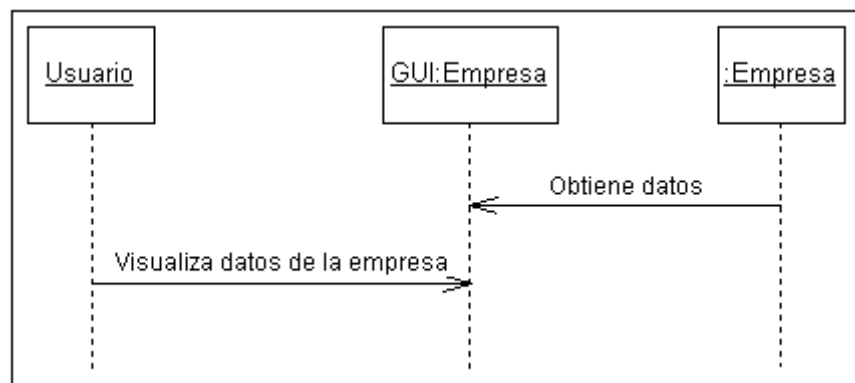


Figura 7.41. Diagrama de secuencia: Consultar empresa ²²⁹

²²⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²²⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Modificar empresa

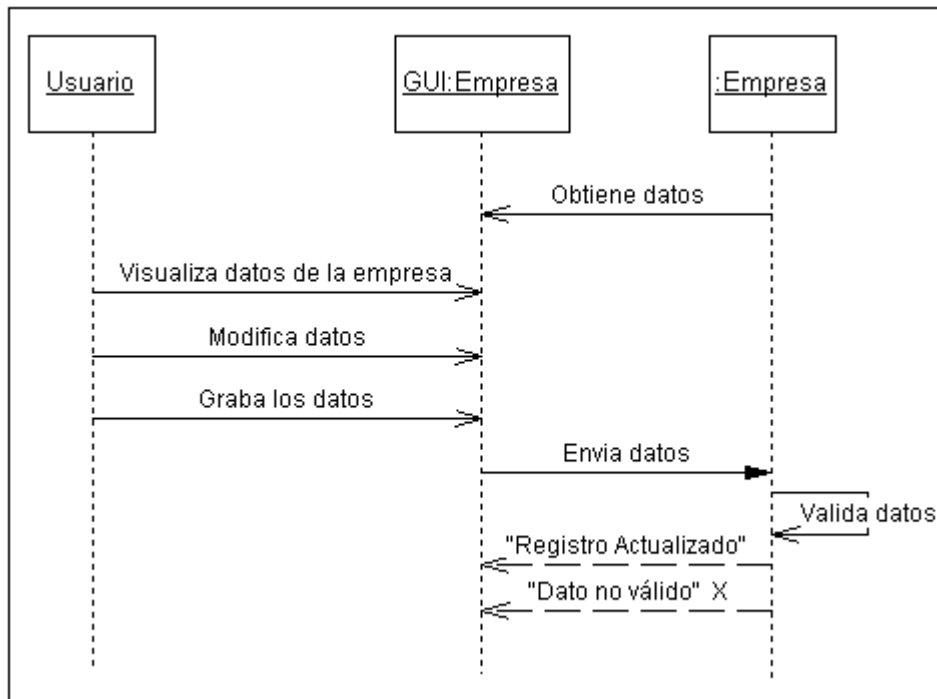


Figura 7.42. Diagrama de secuencia: Modificar empresa ²³⁰

Diagrama de secuencia: Ingresar sucursal

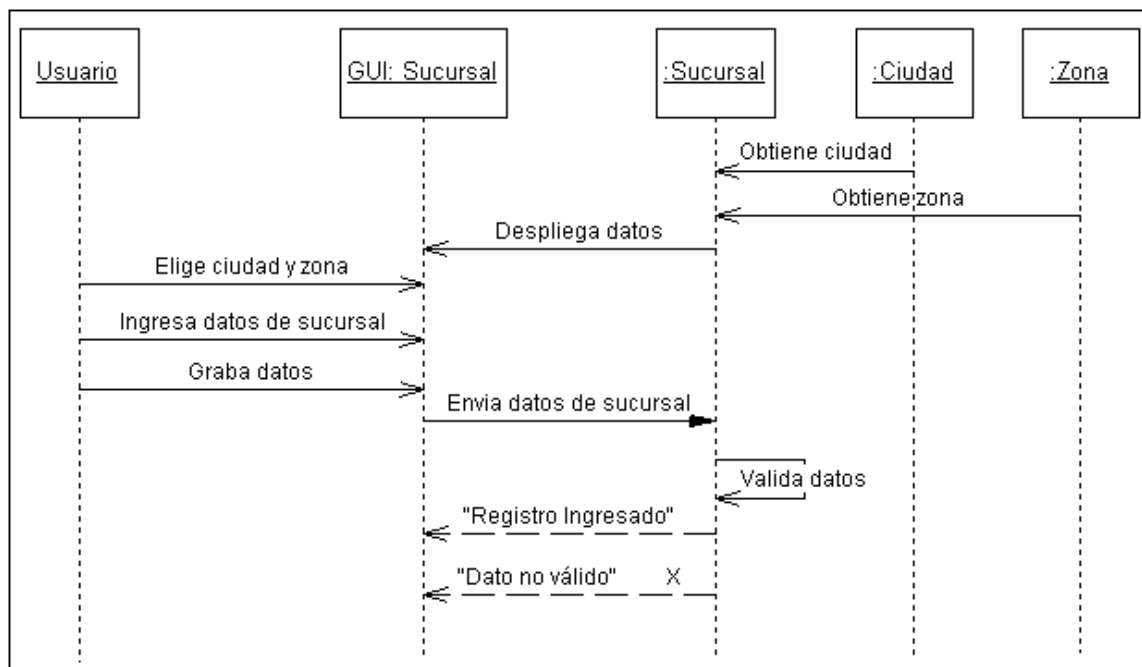


Figura 7.43. Diagrama de secuencia: Ingresar sucursal ²³¹

²³⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar sucursal

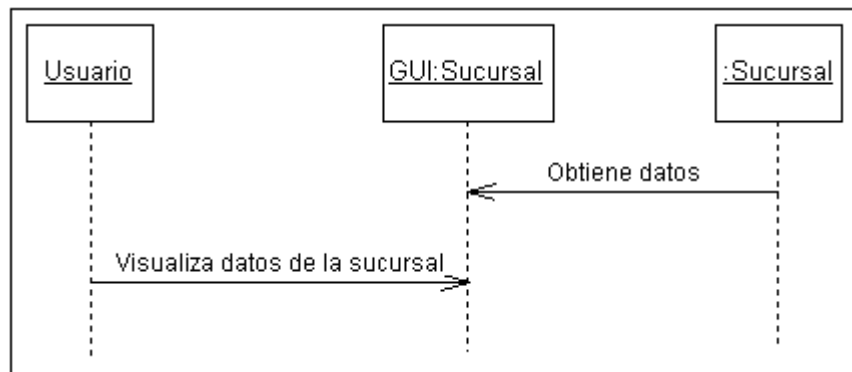


Figura 7.44. Diagrama de secuencia: Consultar sucursal²³²

Diagrama de secuencia: Modificar sucursal

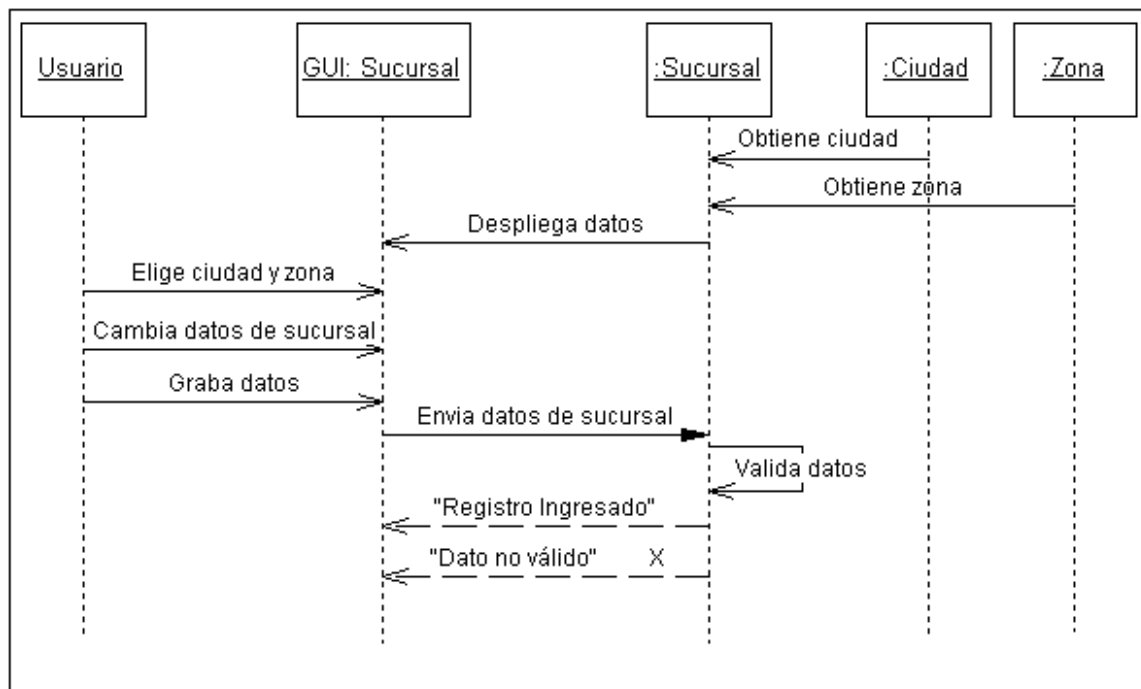


Figura 7.45. Diagrama de secuencia: Modificar sucursal²³³

²³¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²³² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²³³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar sucursal

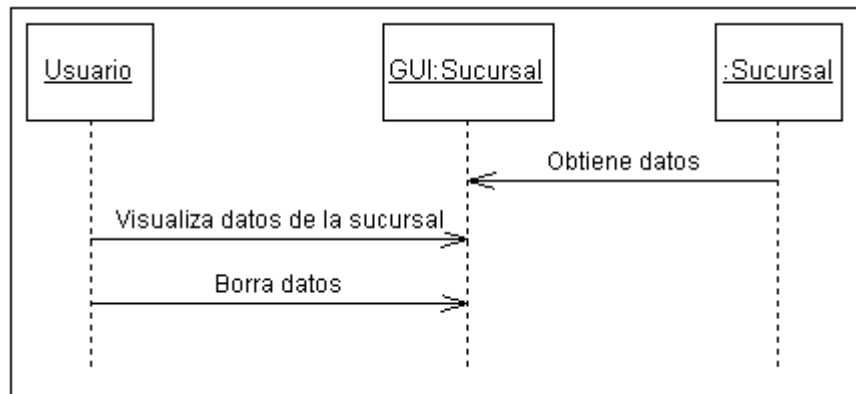


Figura 7.46. Diagrama de secuencia: Eliminar sucursal²³⁴

Diagrama de secuencia: Ingresar ubicación geográfica

Ingreso país

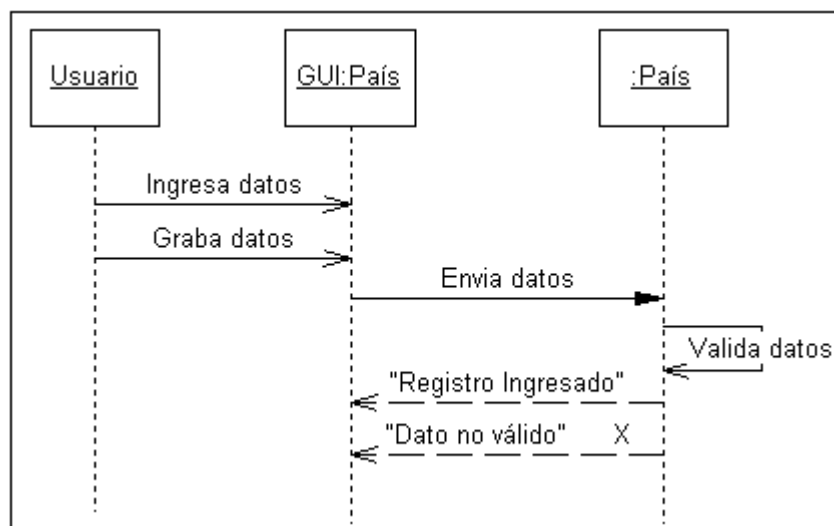


Figura 7.47. Diagrama de secuencia: Ingresar País²³⁵

²³⁴ ERAZO Inés; SAN DOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²³⁵ ERAZO Inés; SAN DOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Ingreso Región

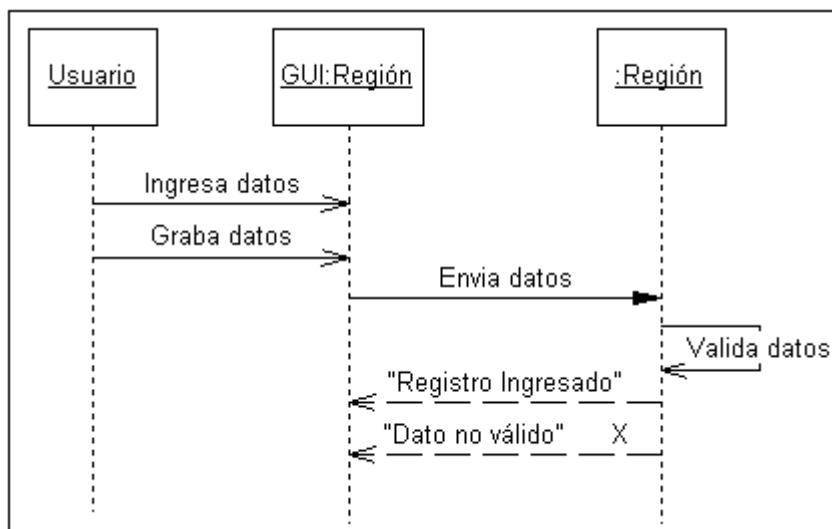


Figura 7.48. Diagrama de secuencia: Ingresar Región ²³⁶

Ingreso Provincia

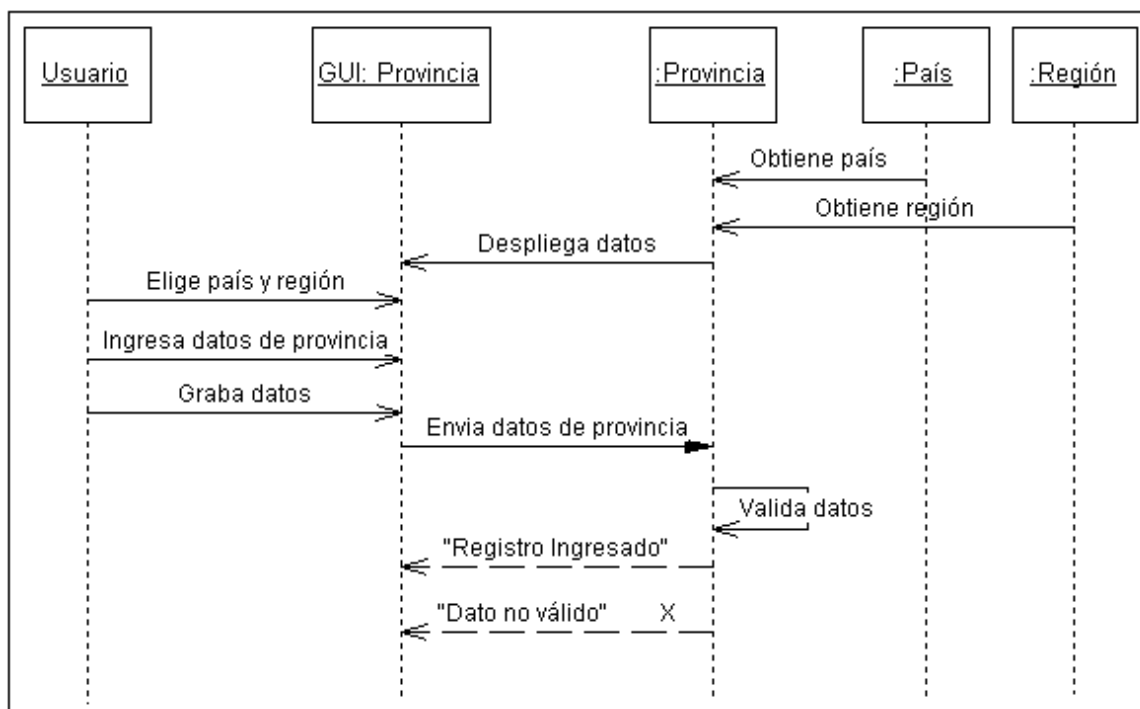


Figura 7.49. Diagrama de secuencia: Ingresar Provincia ²³⁷

²³⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²³⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Ingreso Ciudad

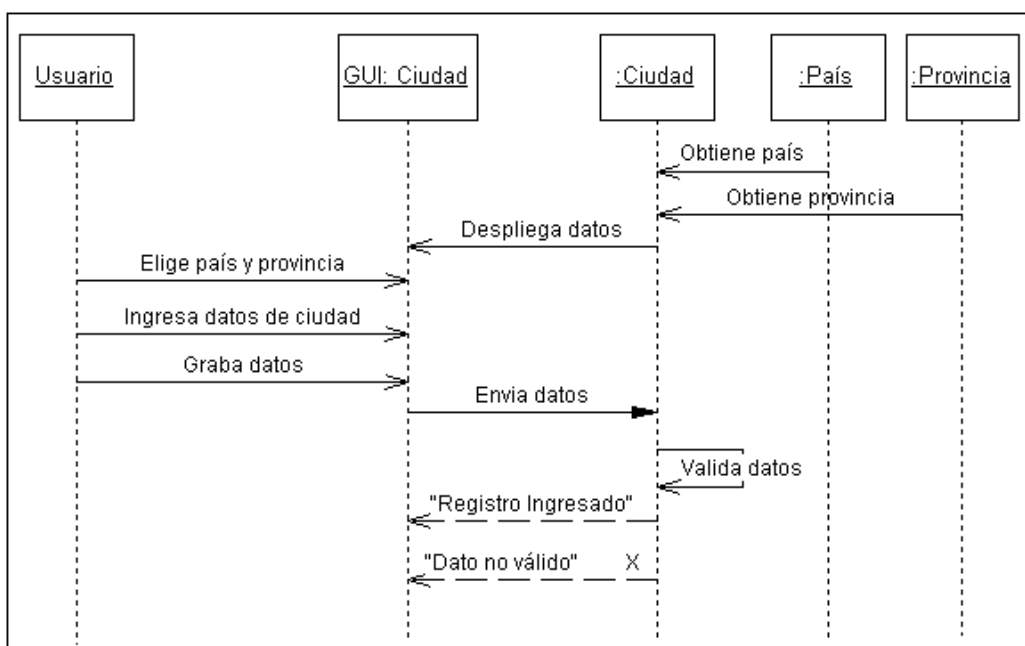


Figura 7.50. Diagrama de secuencia: Ingresar Ciudad ²³⁸

Ingresar zona

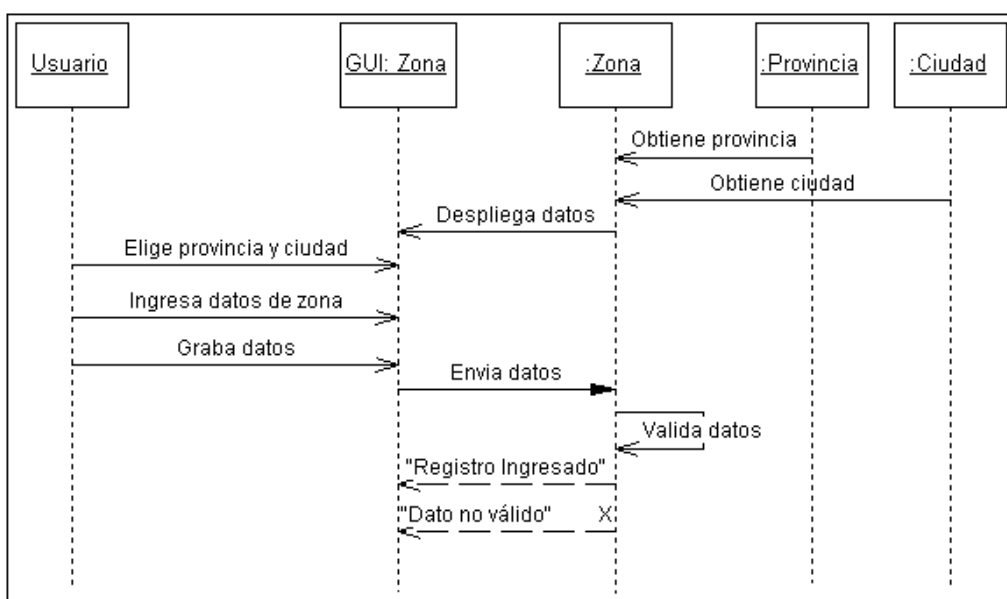


Figura 7.51. Diagrama de secuencia: Ingresar Zona ²³⁹

²³⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²³⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar ubicación geográfica

Para consultar los datos de ubicación geográfica: país, región, provincia, ciudad o zona, se sigue la misma secuencia que se muestra en la Figura 7.52.

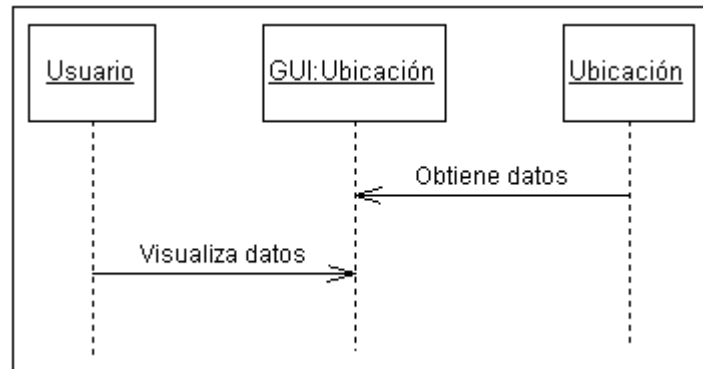


Figura 7.52. Diagrama de secuencia: Consultar Ubicación geográfica ²⁴⁰

Diagrama de secuencia: Modificar ubicación geográfica

Al modificar los datos de la ubicación geográfica: país, región, provincia, ciudad o zona, se tiene la misma secuencia que se muestra en la Figura 7.53.

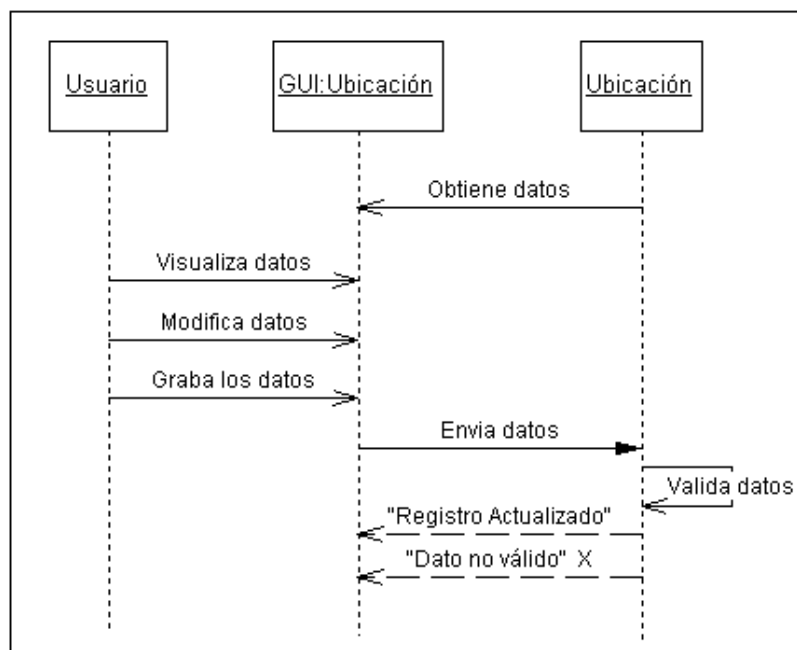


Figura 7.53. Diagrama de secuencia: Modificar Ubicación geográfica ²⁴¹

²⁴⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁴¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar ubicación geográfica

Tanto como para país, región, provincia, ciudad o zona, la secuencia para el proceso de eliminación es la que se muestra a continuación:

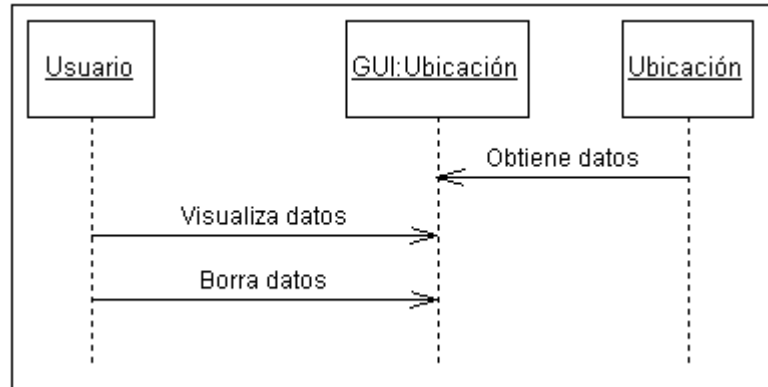


Figura 7.54. Diagrama de secuencia: Eliminar Ubicación geográfica ²⁴²

Diagrama de secuencia: Ingresar forma de pago

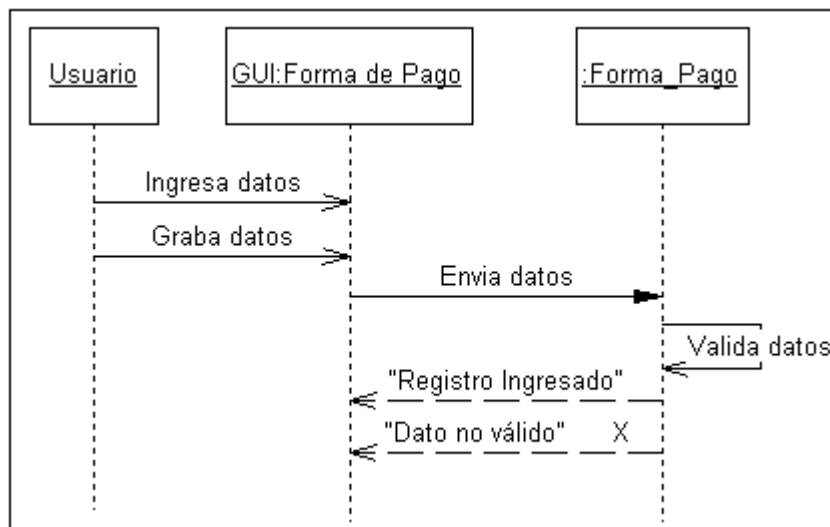


Figura 7.55. Diagrama de secuencia: Ingresar forma de pago ²⁴³

²⁴² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁴³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar forma de pago

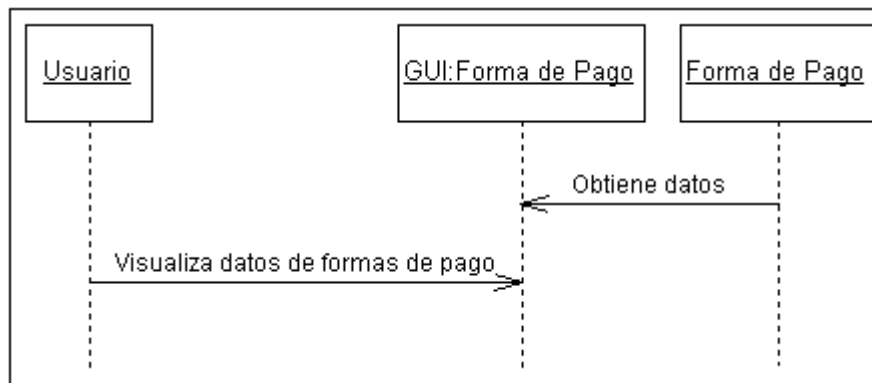


Figura 7.56. Diagrama de secuencia: Consultar forma de pago²⁴⁴

Diagrama de secuencia: Modificar forma de pago

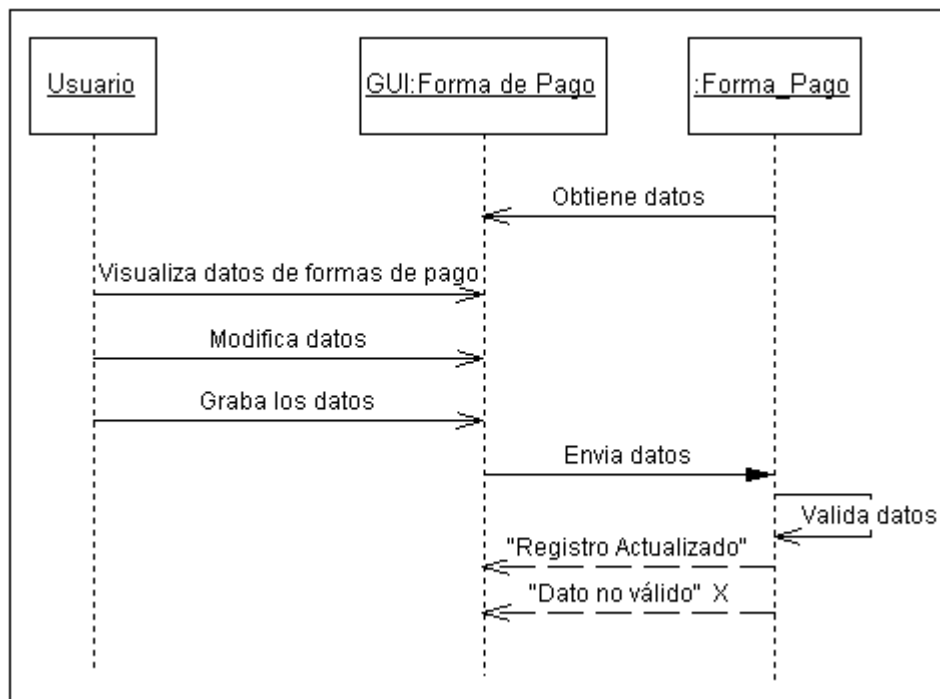


Figura 7.57. Diagrama de secuencia: Modificar forma de pago²⁴⁵

²⁴⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁴⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar forma de pago

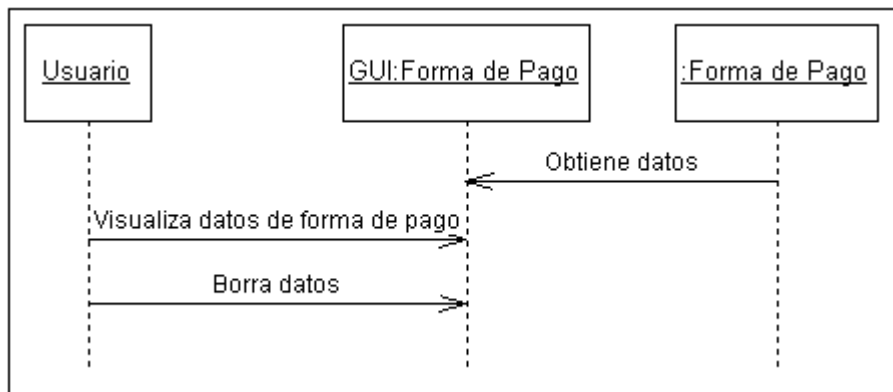


Figura 7.58. Diagrama de secuencia: Eliminar forma de pago²⁴⁶

Diagrama de secuencia: Ingresar impuestos

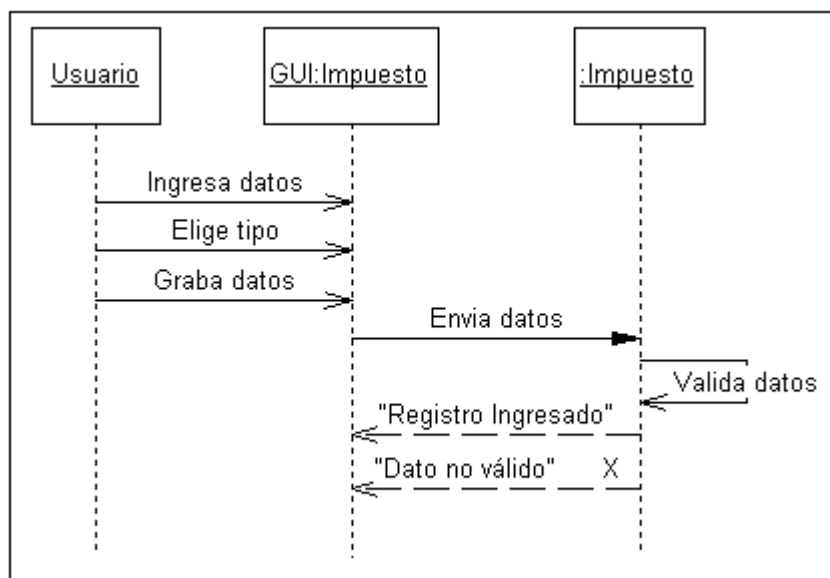


Figura 7.59. Diagrama de secuencia: Ingresar impuestos²⁴⁷

²⁴⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁴⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar impuestos

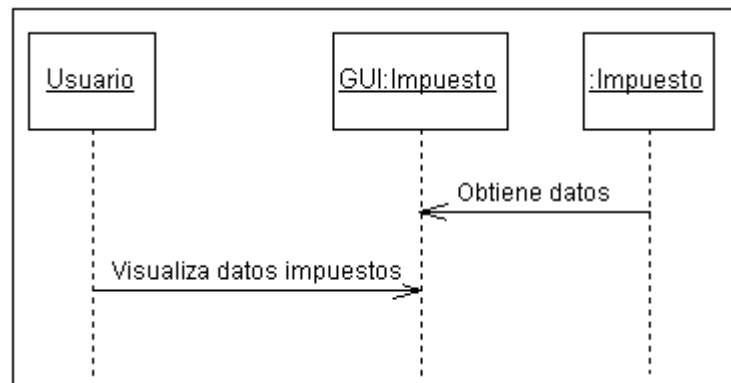


Figura 7.60. Diagrama de secuencia: Consultar impuestos²⁴⁸

Diagrama de secuencia: Modificar impuestos

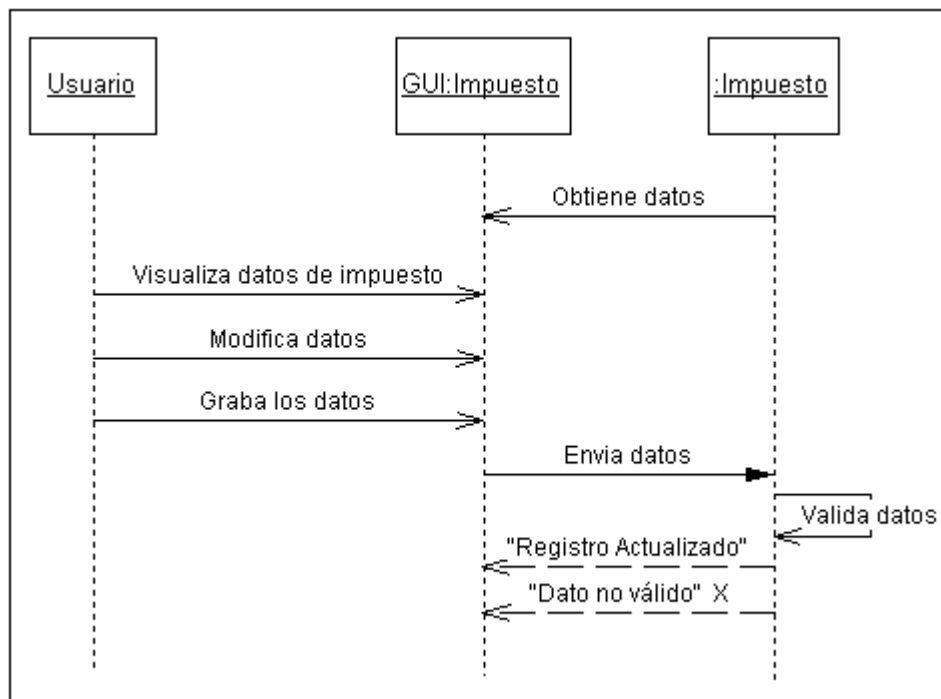


Figura 7.61. Diagrama de secuencia: Modificar impuestos²⁴⁹

²⁴⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁴⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar impuestos

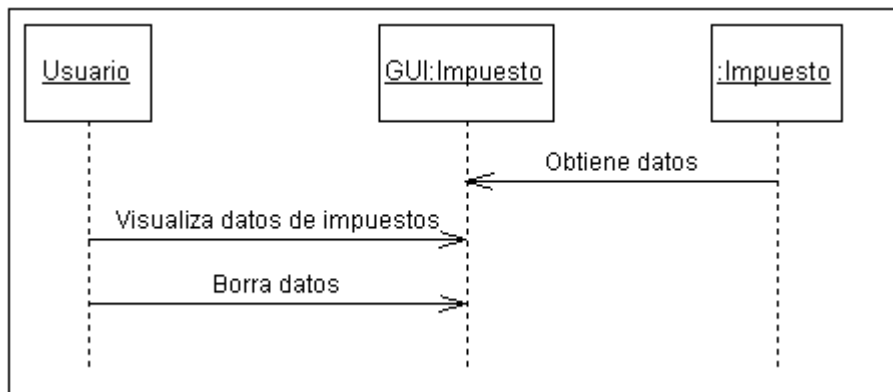


Figura 7.62. Diagrama de secuencia: Eliminar impuestos²⁵⁰

Diagrama de secuencia: Ingresar descuentos y recargos

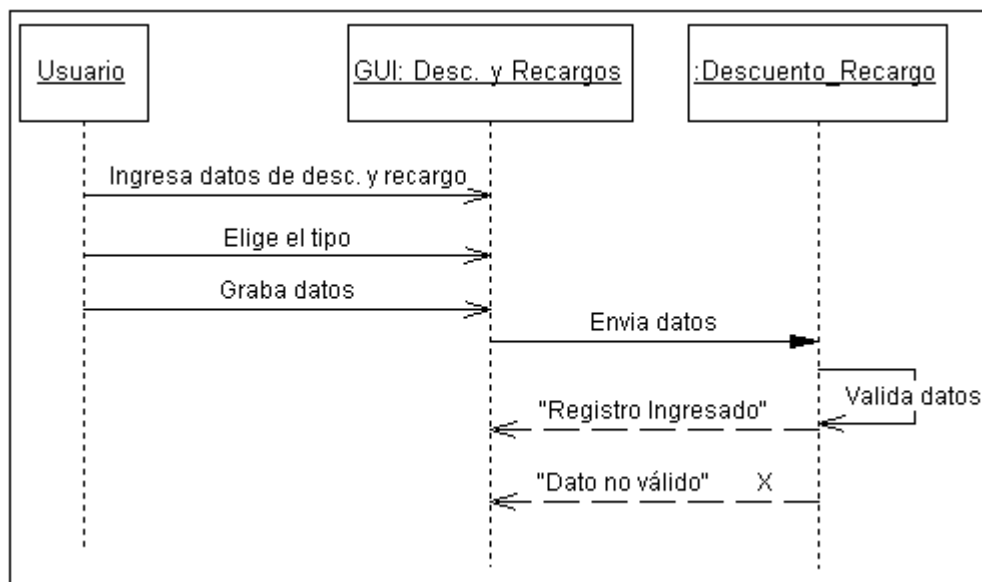


Figura 7.63. Diagrama de secuencia: Ingresar descuentos y recargos²⁵¹

²⁵⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁵¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar descuentos y recargos

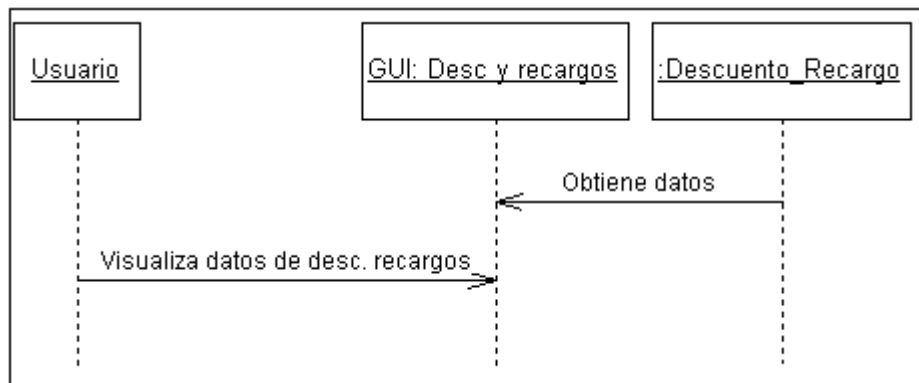


Figura 7.64. Diagrama de secuencia: Consultar descuentos y recargos²⁵²

Diagrama de secuencia: Modificar descuentos y recargos

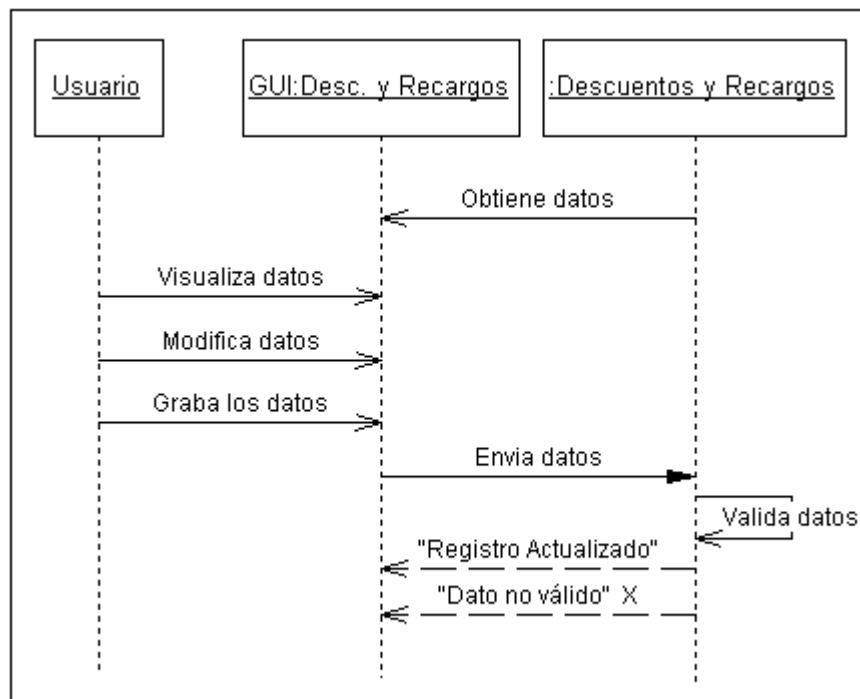


Figura 7.65. Diagrama de secuencia: Modificar descuentos y recargos²⁵³

²⁵² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁵³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar descuentos y recargos

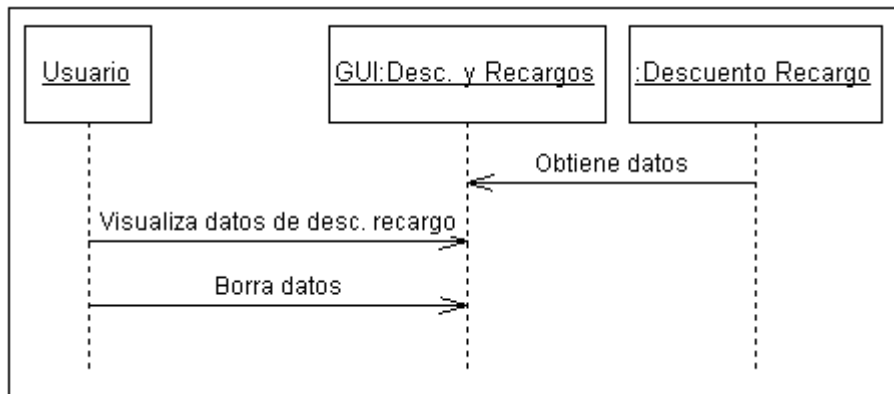


Figura 7.66. Diagrama de secuencia: Eliminar descuentos y recargos²⁵⁴

Diagrama de secuencia: Crear perfil

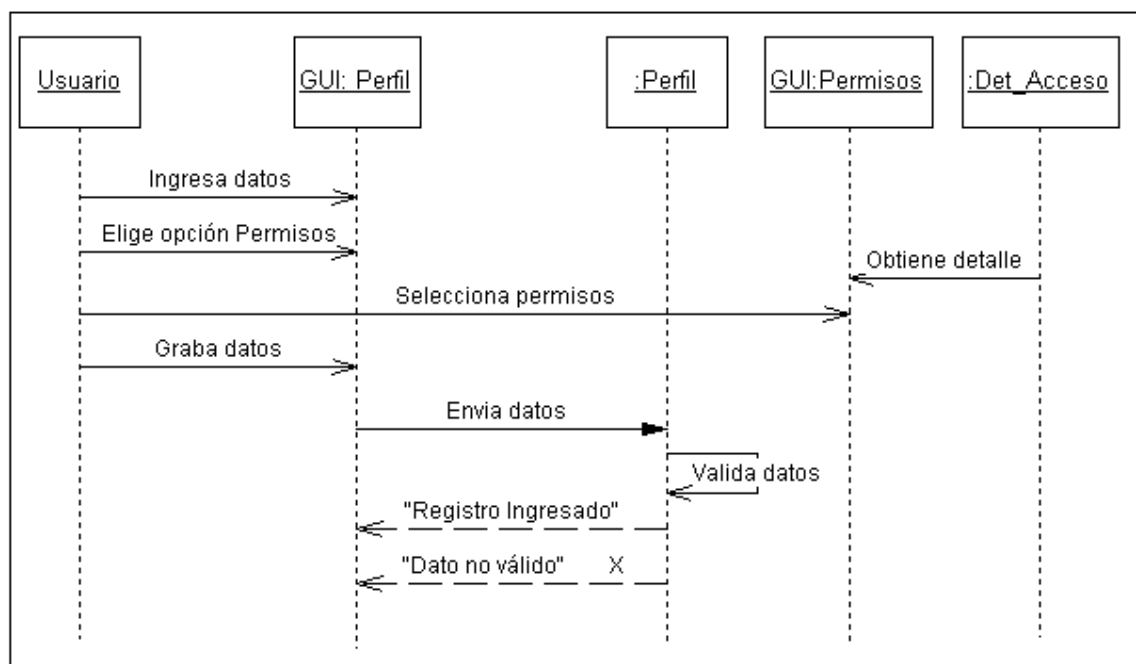


Figura 7.67. Diagrama de secuencia: Crear perfil²⁵⁵

²⁵⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁵⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar perfil

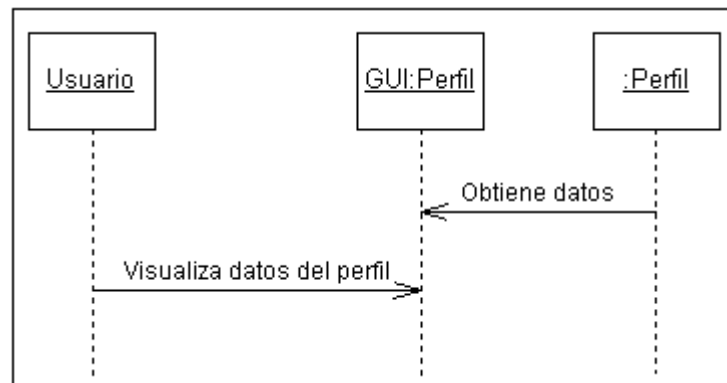


Figura 7.68. Diagrama de secuencia: Consultar perfil ²⁵⁶

Diagrama de secuencia: Modificar perfil

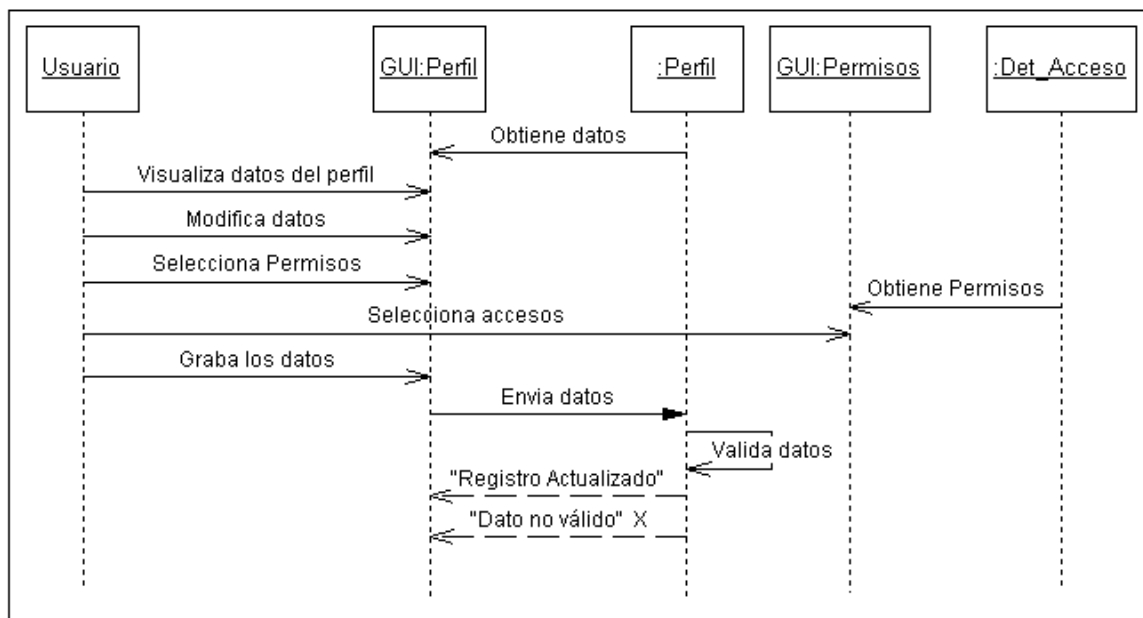


Figura 7.69. Diagrama de secuencia: Modificar perfil ²⁵⁷

²⁵⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁵⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar perfil

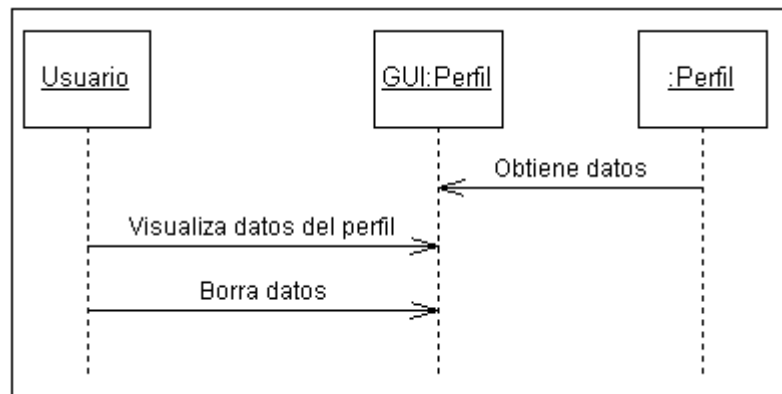


Figura 7.70. Diagrama de secuencia: Eliminar perfil²⁵⁸

Diagrama de secuencia: Ingresar usuario

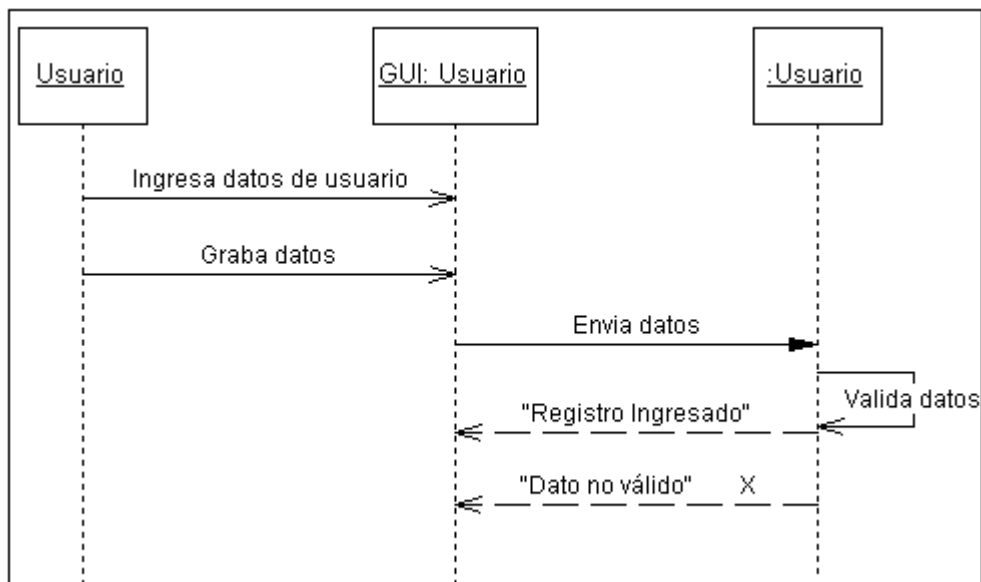


Figura 7.71. Diagrama de secuencia: Ingresar usuario²⁵⁹

²⁵⁸ ERAZO Inés; SAN DOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁵⁹ ERAZO Inés; SAN DOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar usuario

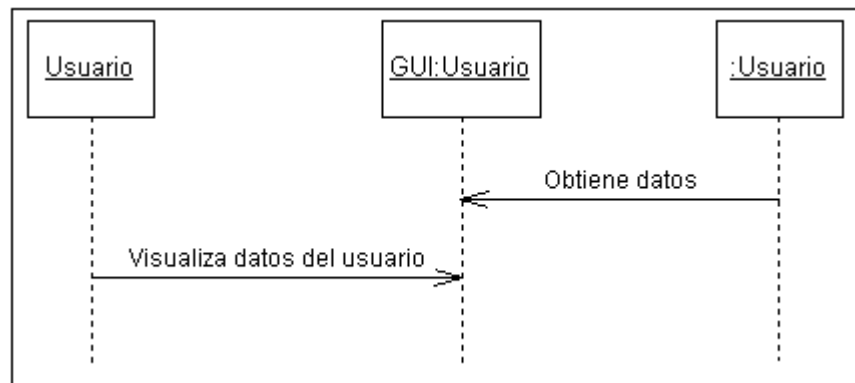


Figura 7.72. Diagrama de secuencia: Consultar usuario²⁶⁰

Diagrama de secuencia: Modificar usuario

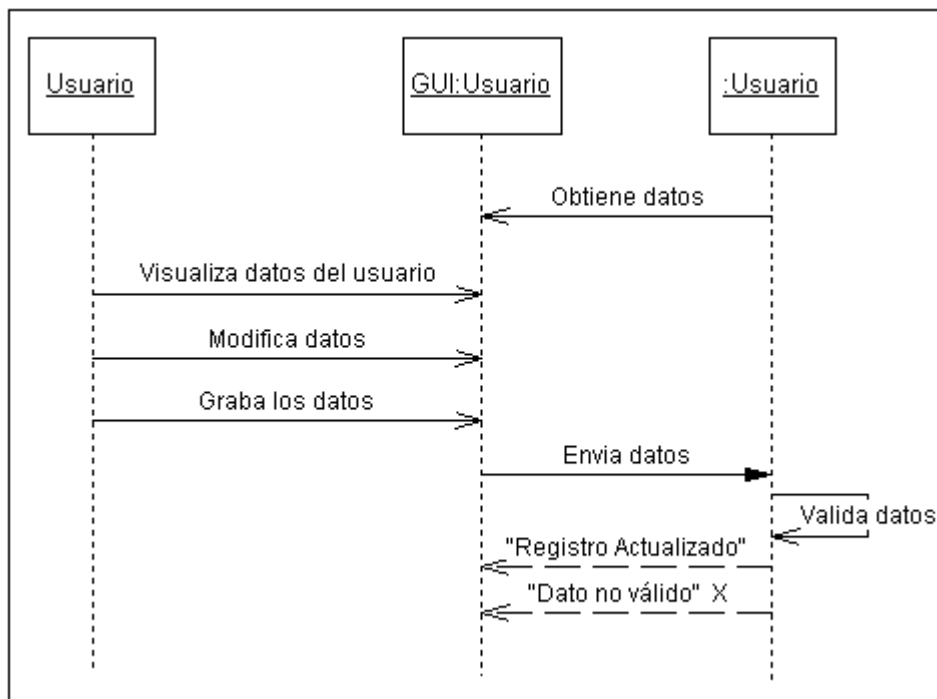


Figura 7.73. Diagrama de secuencia: Modificar usuario²⁶¹

²⁶⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁶¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar usuario

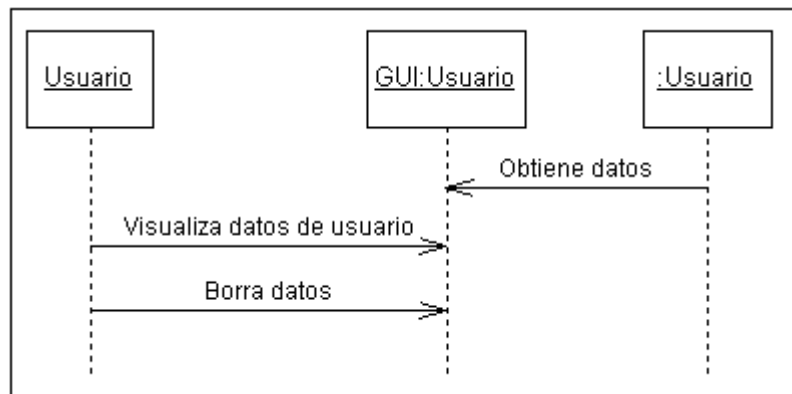


Figura 7.74. Diagrama de secuencia: Eliminar usuario²⁶²

Diagrama de secuencia: Generar respaldo

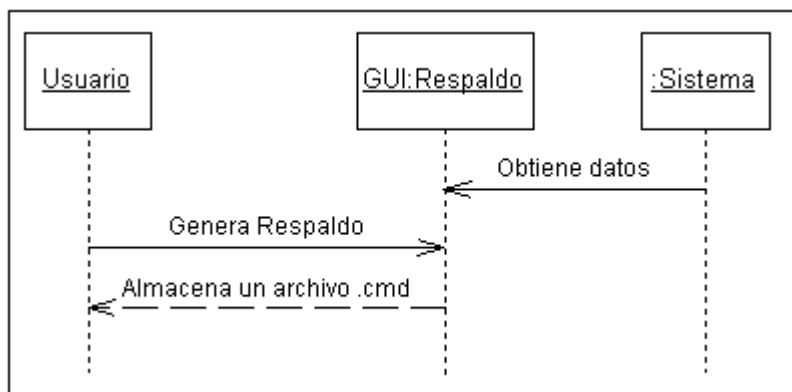


Figura 7.75. Diagrama de secuencia: Generar respaldo²⁶³

²⁶² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁶³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo de Inventario

Diagrama de secuencia: Ingresar línea de producto

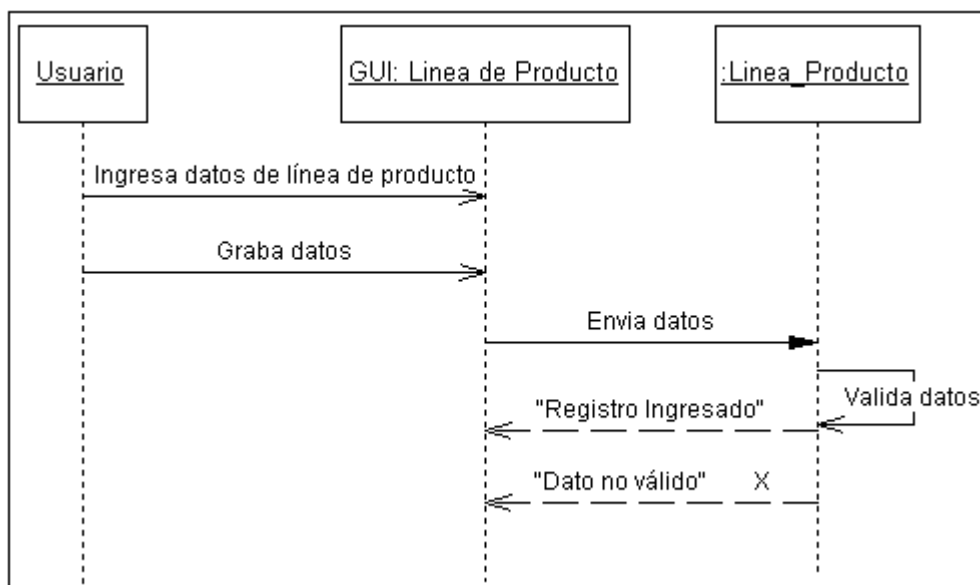


Figura 7.76. Diagrama de secuencia: Ingresar línea de producto²⁶⁴

Diagrama de secuencia: Consultar línea de producto

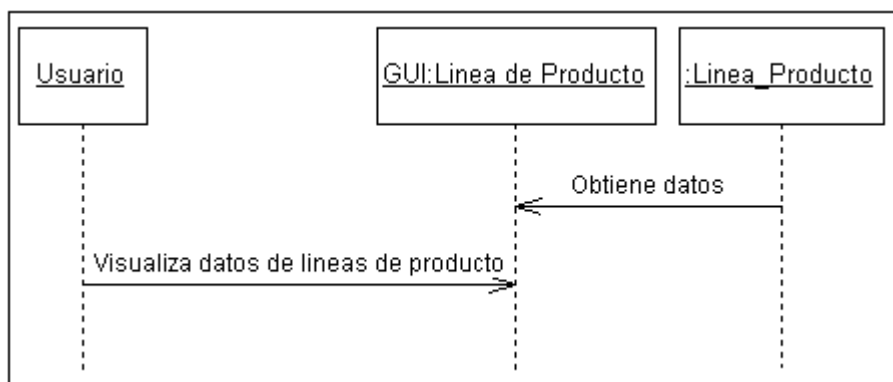


Figura 7.77. Diagrama de secuencia: Consultar línea de producto²⁶⁵

²⁶⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁶⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Modificar línea de producto

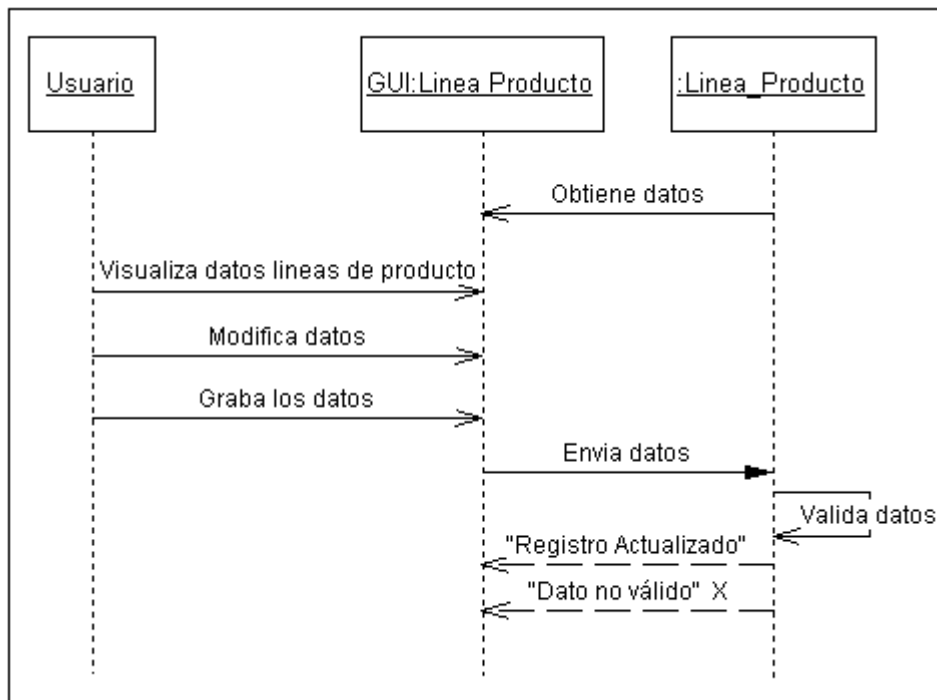


Figura 7.78. Diagrama de secuencia: Modificar línea de producto²⁶⁶

Diagrama de secuencia: Eliminar línea de producto

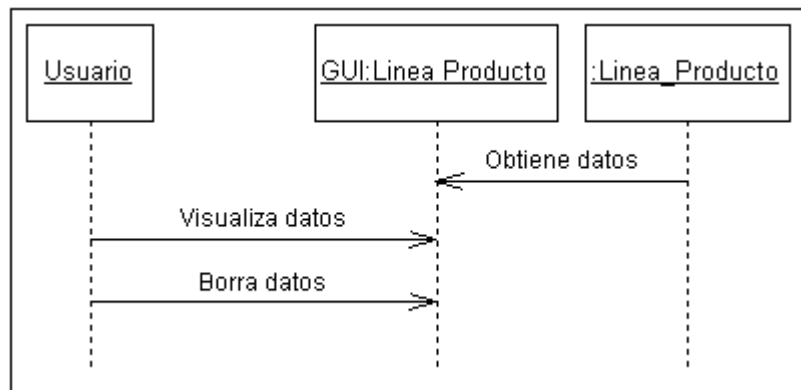


Figura 7.79. Diagrama de secuencia: Eliminar línea de producto²⁶⁷

²⁶⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁶⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar grupo de producto

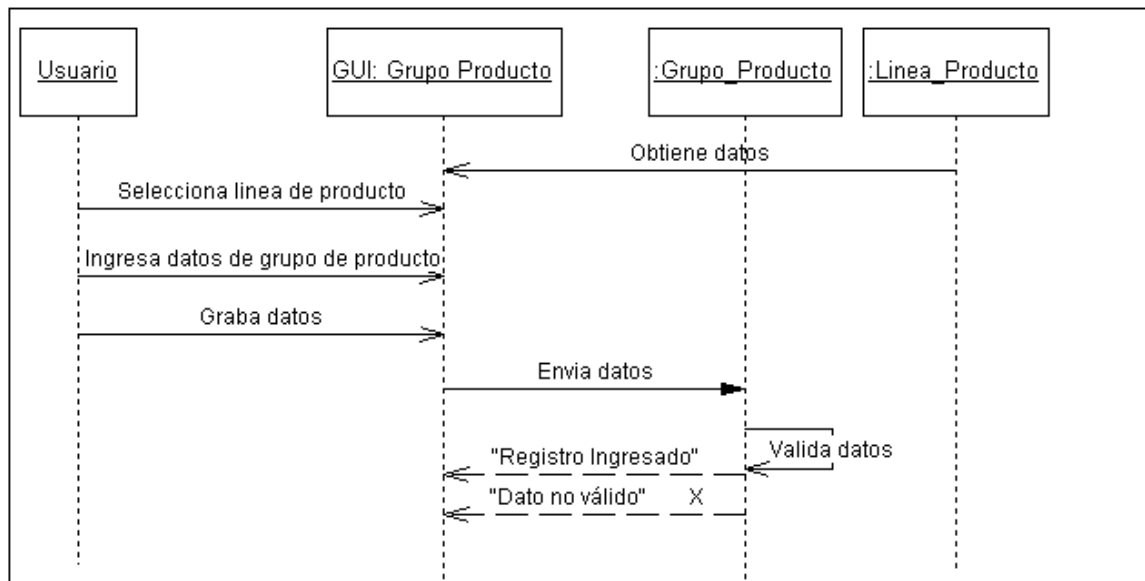


Figura 7.80. Diagrama de secuencia: Ingresar grupo de producto ²⁶⁸

Diagrama de secuencia: Consultar grupo de producto

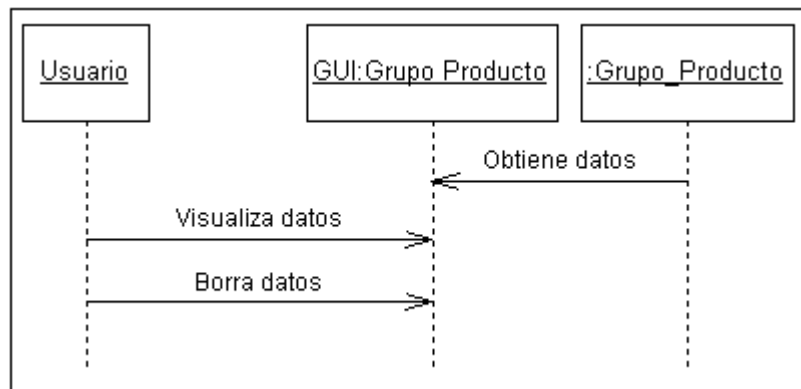


Figura 7.81. Diagrama de secuencia: Consultar línea de producto ²⁶⁹

²⁶⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁶⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Modificar grupo de producto

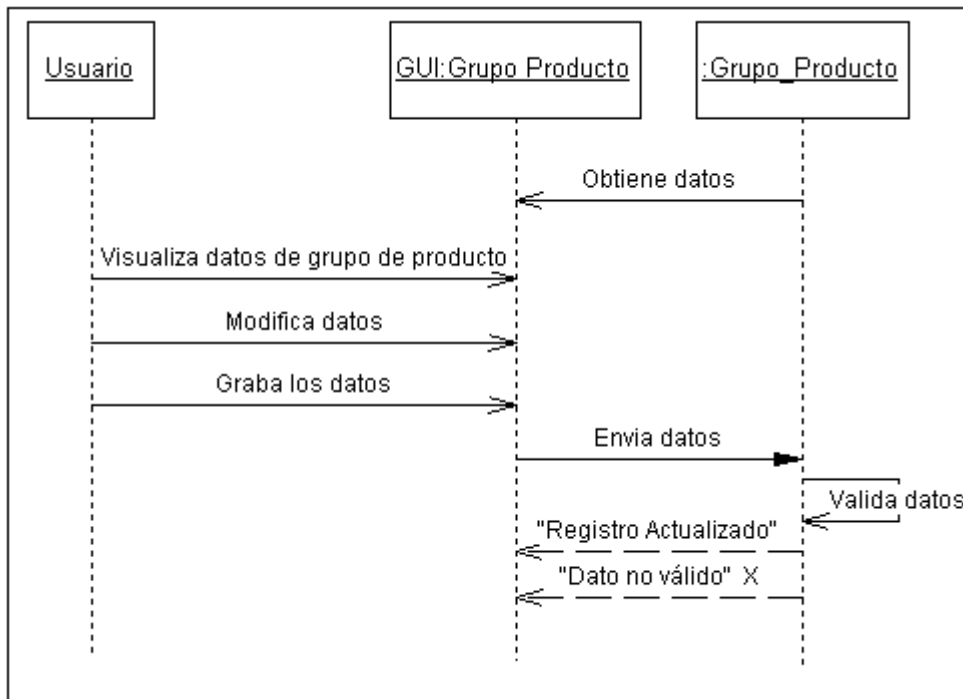


Figura 7.82. Diagrama de secuencia: Modificar línea de producto²⁷⁰

Diagrama de secuencia: Eliminar grupo de producto

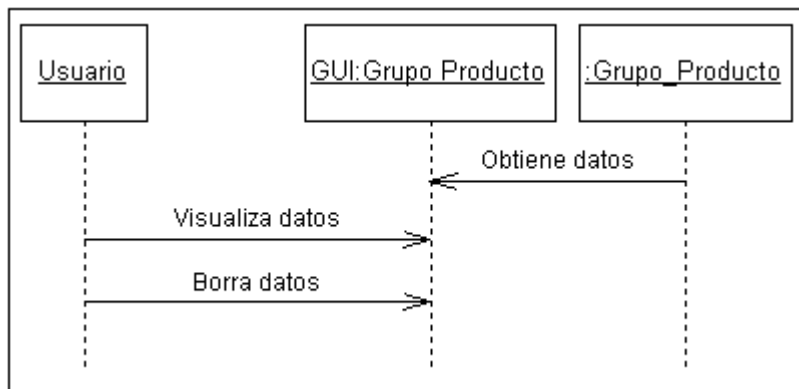


Figura 7.83. Diagrama de secuencia: Eliminar línea de producto²⁷¹

²⁷⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁷¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar producto

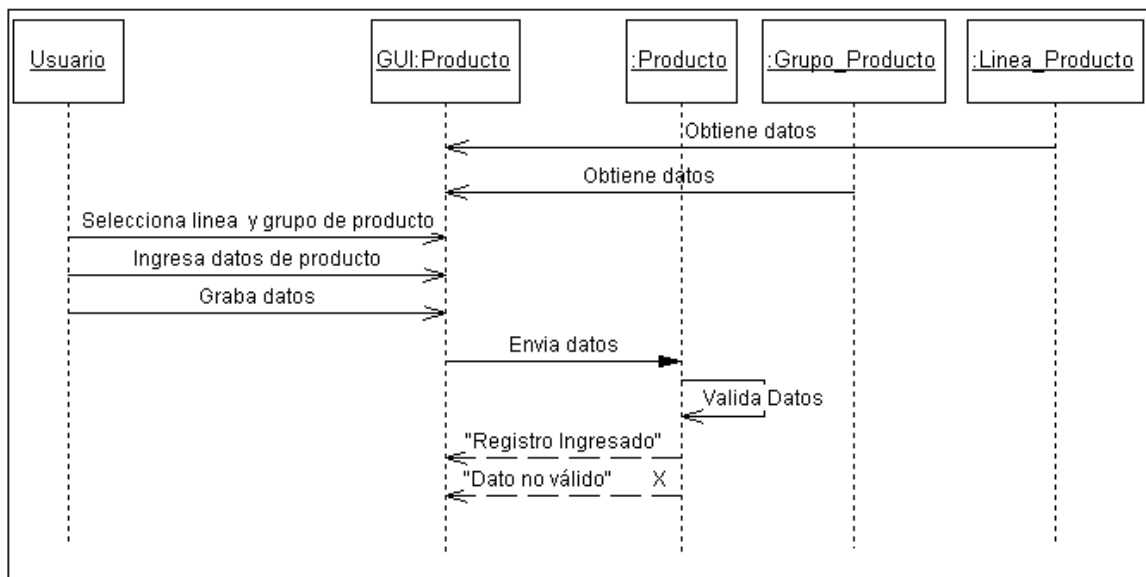


Figura 7.84. Diagrama de secuencia: Ingresar producto ²⁷²

Diagrama de secuencia: Consultar producto

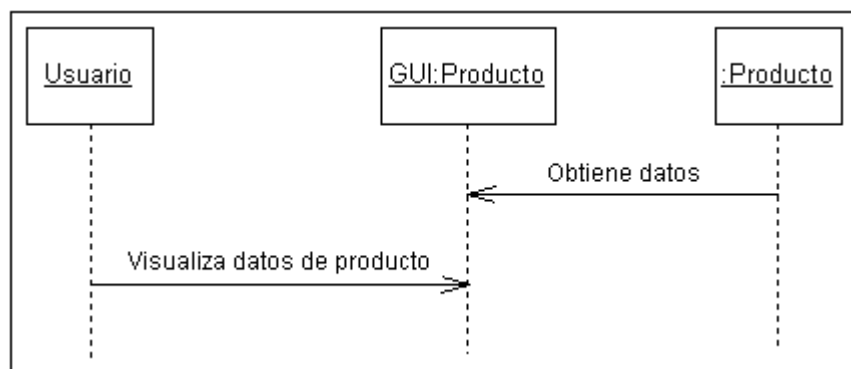


Figura 7.85. Diagrama de secuencia: Consultar producto ²⁷³

²⁷² ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁷³ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Modificar producto

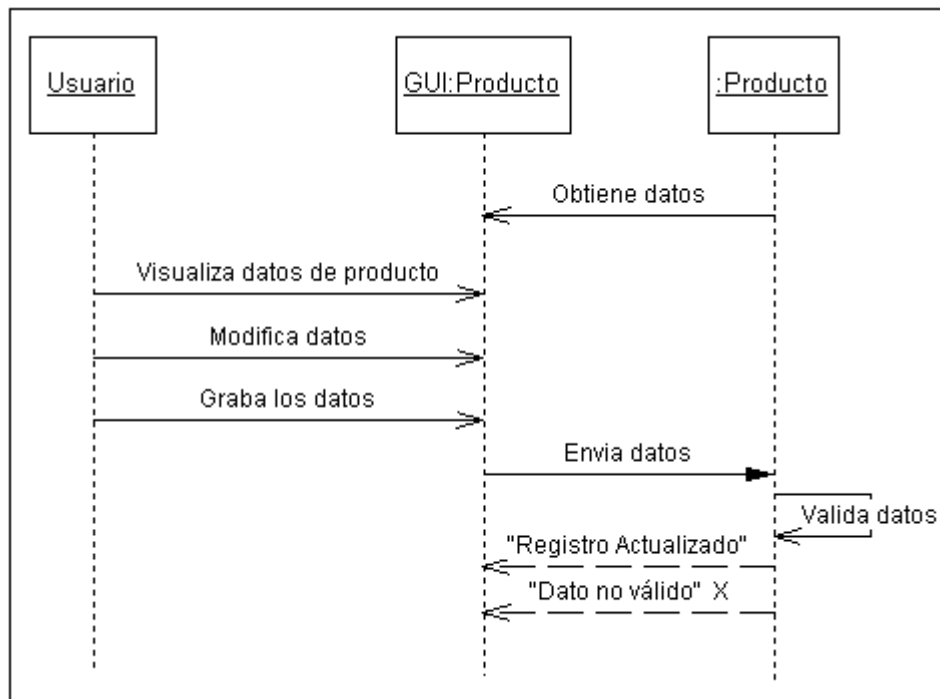


Figura 7.86. Diagrama de secuencia: Modificar producto²⁷⁴

Diagrama de secuencia: Eliminar producto

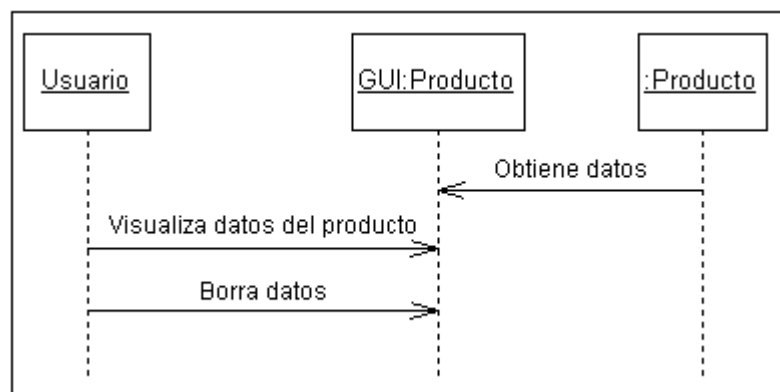


Figura 7.87. Diagrama de secuencia: Eliminar producto²⁷⁵

²⁷⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁷⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar receta

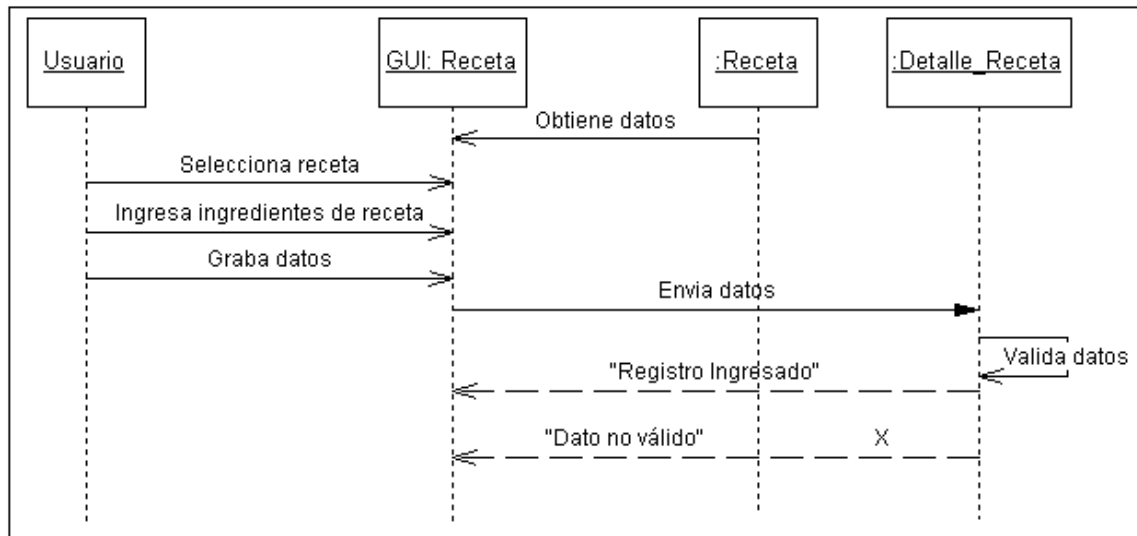


Figura 7.88. Diagrama de secuencia: Ingresar receta²⁷⁶

Diagrama de secuencia: Consultar receta

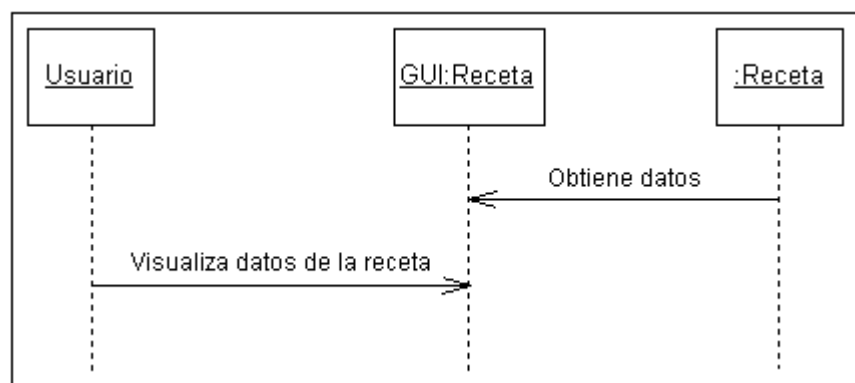


Figura 7.89. Diagrama de secuencia: Consultar receta²⁷⁷

²⁷⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁷⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar receta

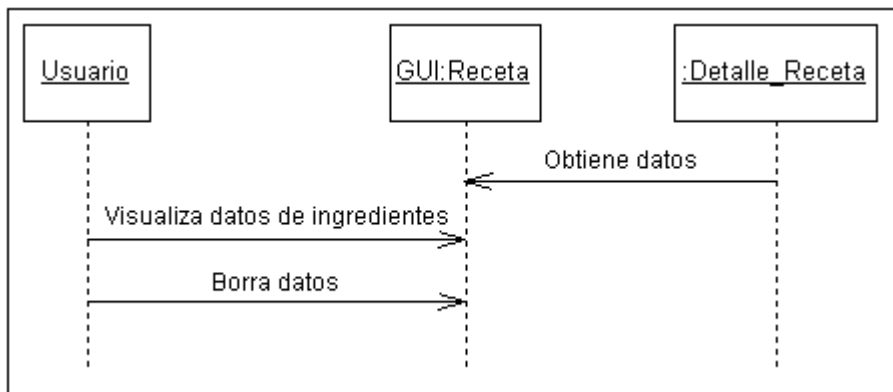


Figura 7.90. Diagrama de secuencia: Eliminar receta²⁷⁸

Diagrama de secuencia: Ingresar inventario inicial

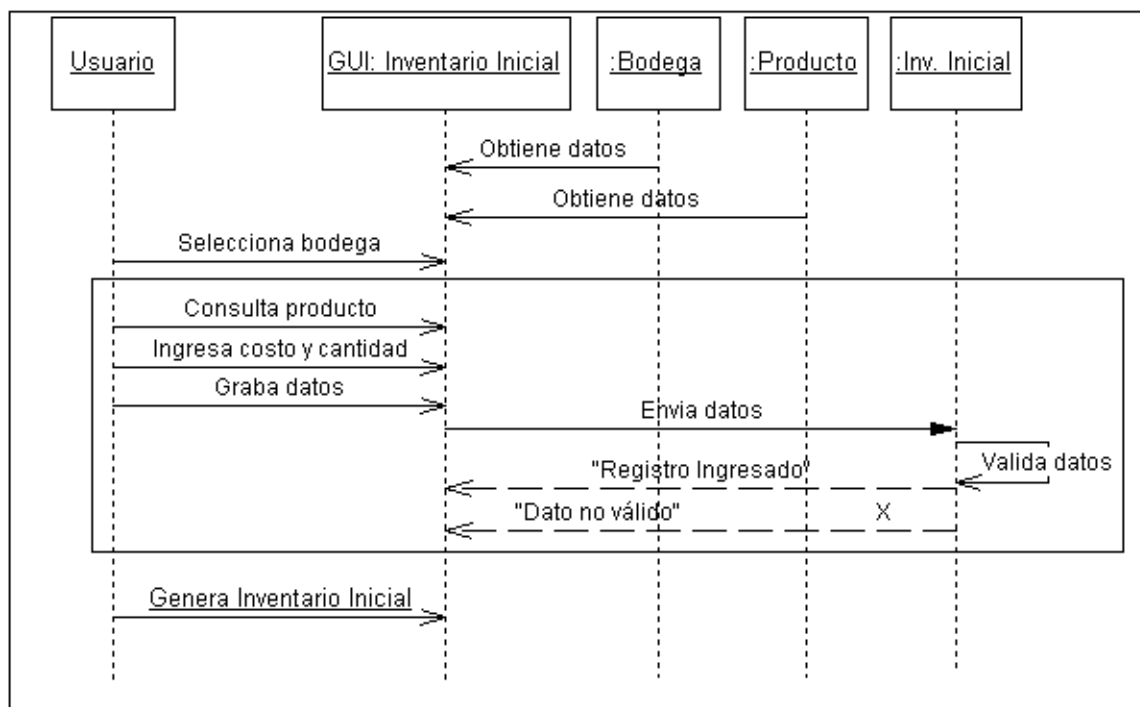


Figura 7.91. Diagrama de secuencia: Ingresar inventario inicial²⁷⁹

²⁷⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁷⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar inventario

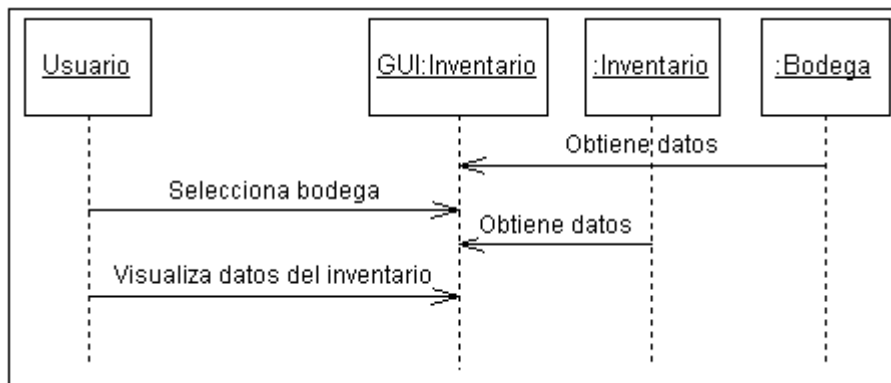


Figura 7.92. Diagrama de secuencia: Consultar inventario²⁸⁰

Diagrama de secuencia: Modificar inventario

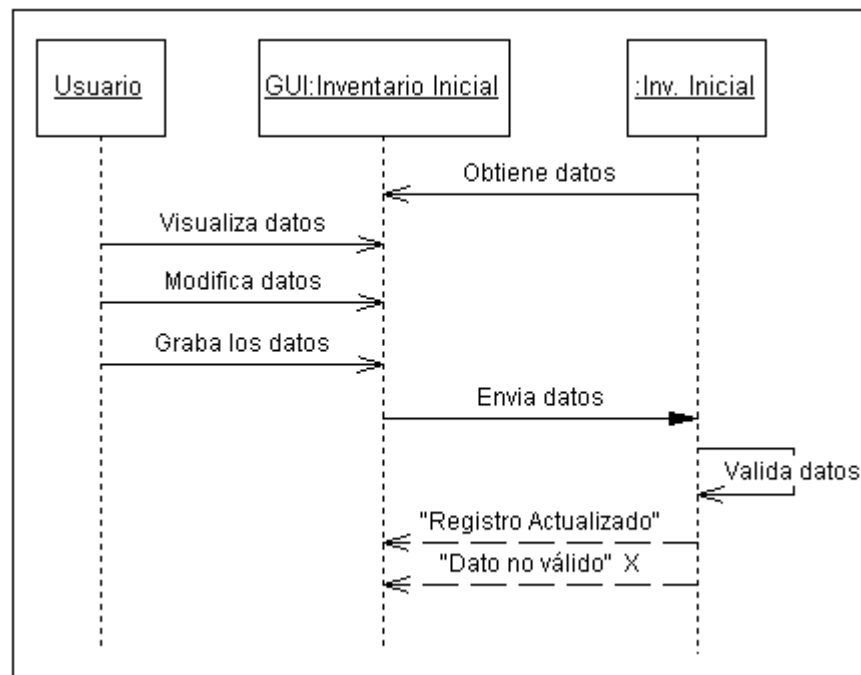


Figura 7.93. Diagrama de secuencia: Modificar inventario²⁸¹

²⁸⁰ ERAZO Inés; SAN DOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁸¹ ERAZO Inés; SAN DOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Imprimir inventario

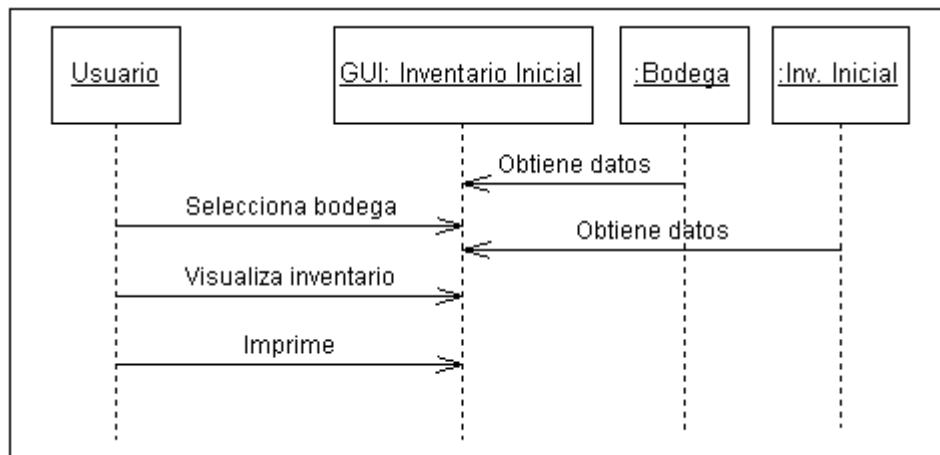


Figura 7.94. Diagrama de secuencia: Imprimir inventario ²⁸²

Diagrama de secuencia: Ingresar toma física

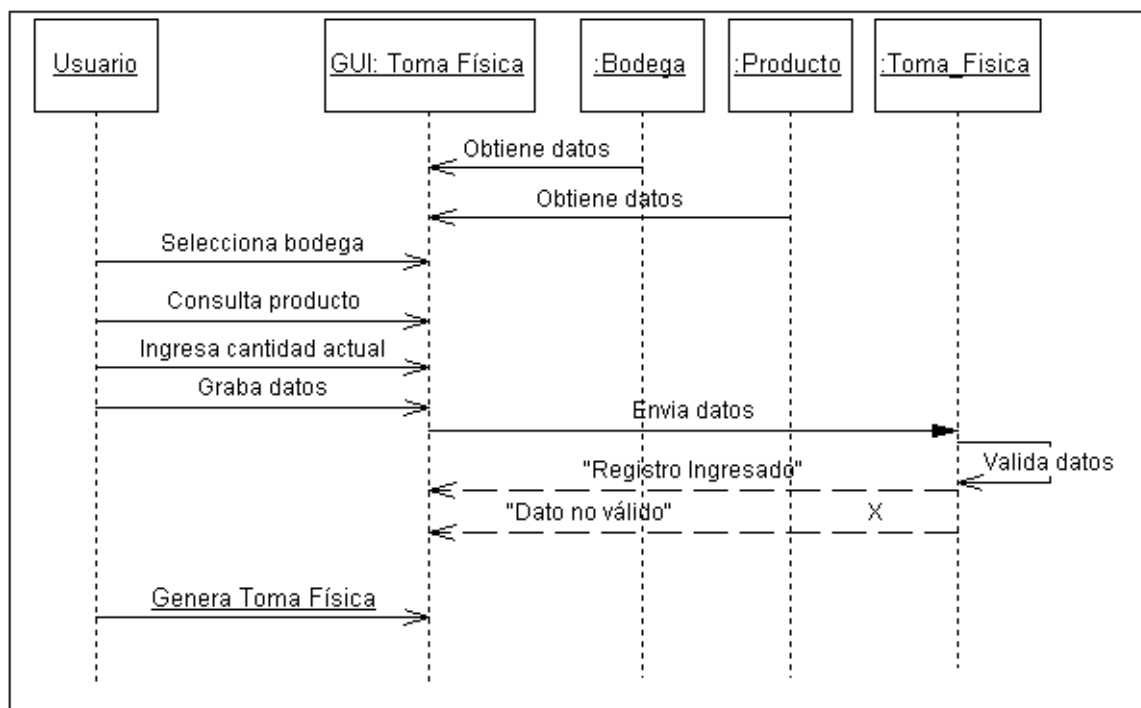


Figura 7.95. Diagrama de secuencia: Ingresar toma física ²⁸³

²⁸² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁸³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Modificar toma física

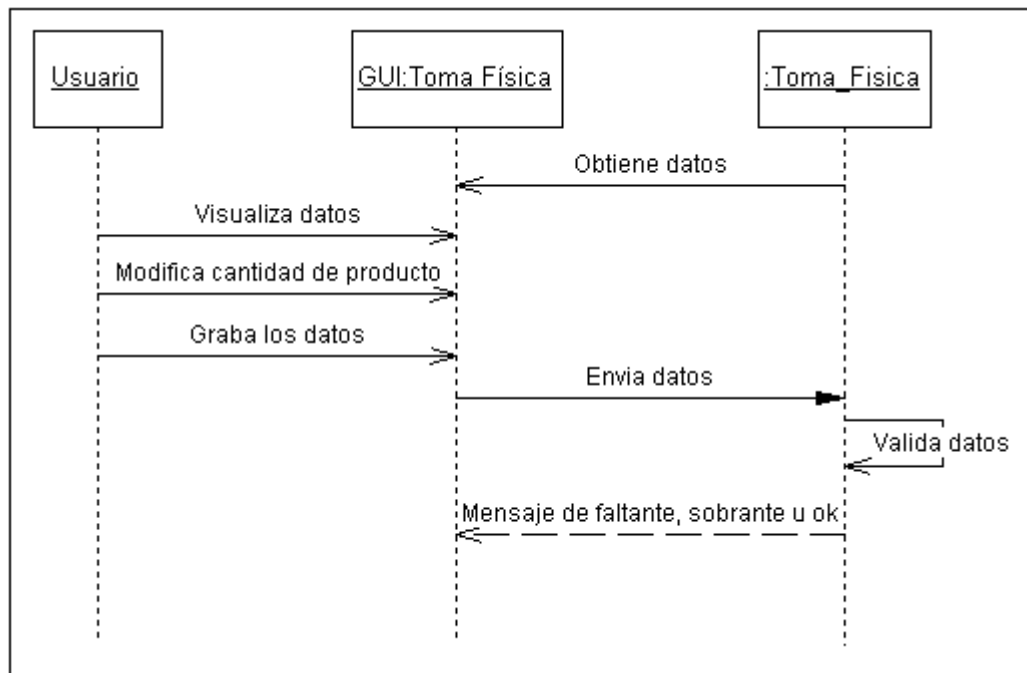


Figura 7.96. Diagrama de secuencia: Modificar toma física²⁸⁴

Diagrama de secuencia: Imprimir toma física

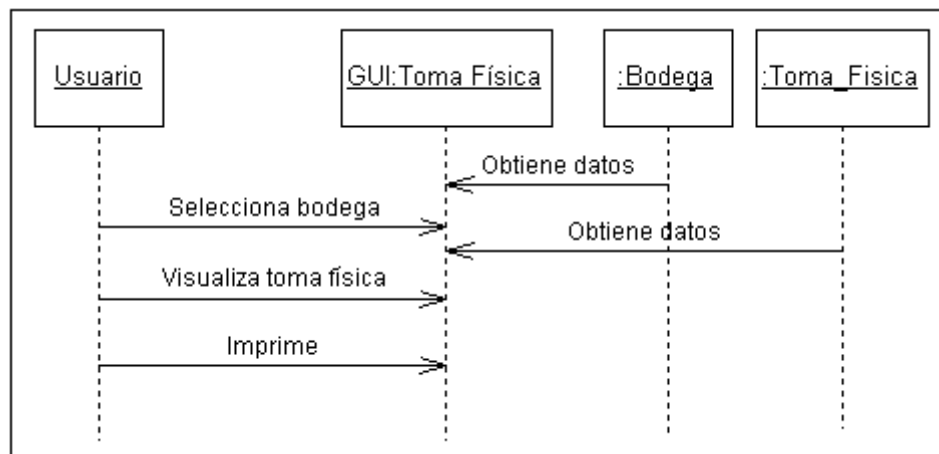


Figura 7.97. Diagrama de secuencia: Imprimir toma física²⁸⁵

²⁸⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁸⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar movimiento de bodega

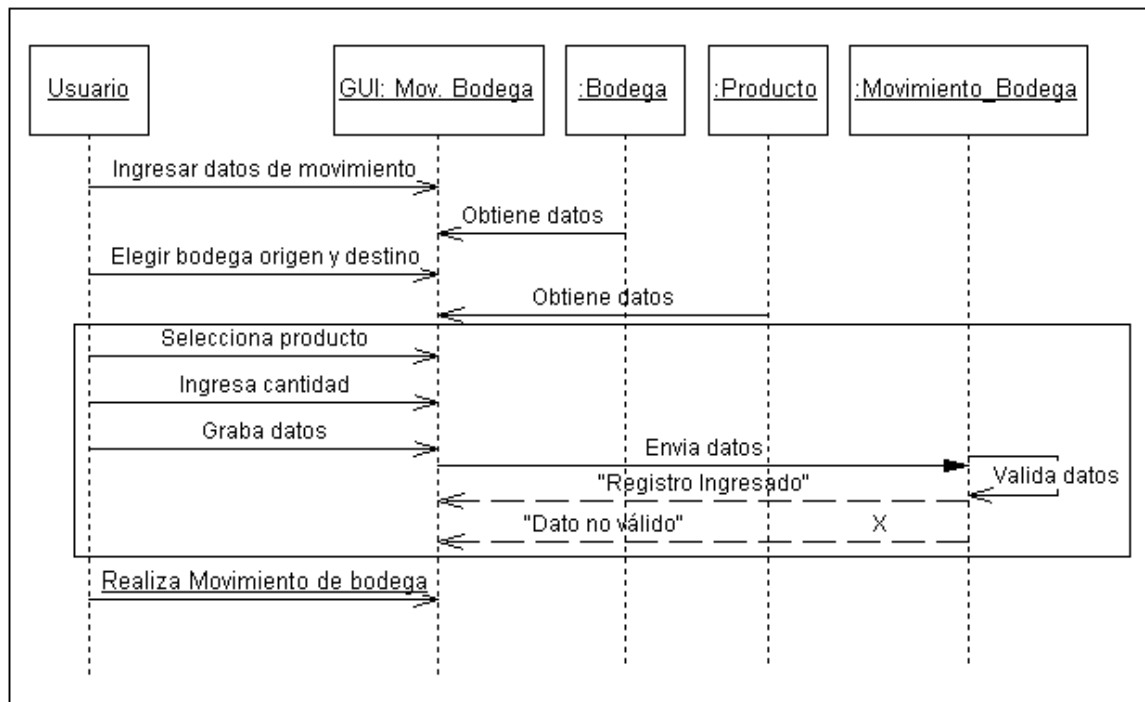


Figura 7.98. Diagrama de secuencia: Ingresar movimiento de bodega²⁸⁶

Diagrama de secuencia: Eliminar movimiento de bodega

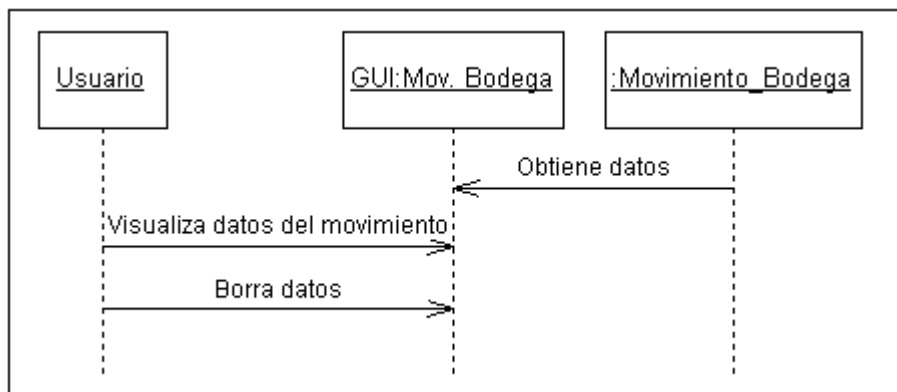


Figura 7.99. Diagrama de secuencia: Eliminar movimiento de bodega²⁸⁷

²⁸⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁸⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Imprimir movimiento de bodega

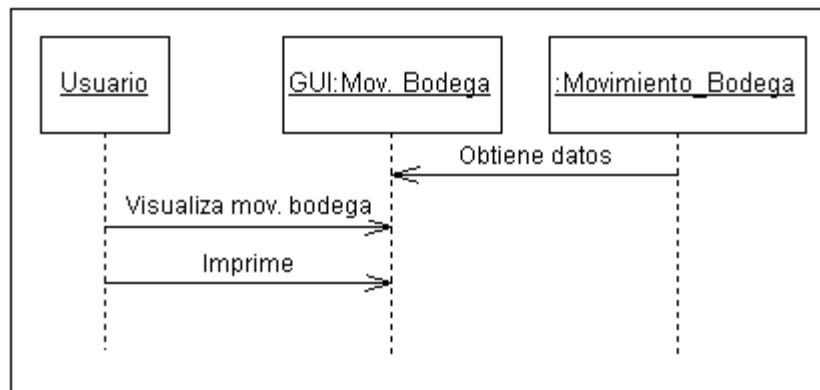


Figura 7.100. Diagrama de secuencia: Imprimir movimiento de bodega²⁸⁸

Diagrama de secuencia: Ingresar preproducción

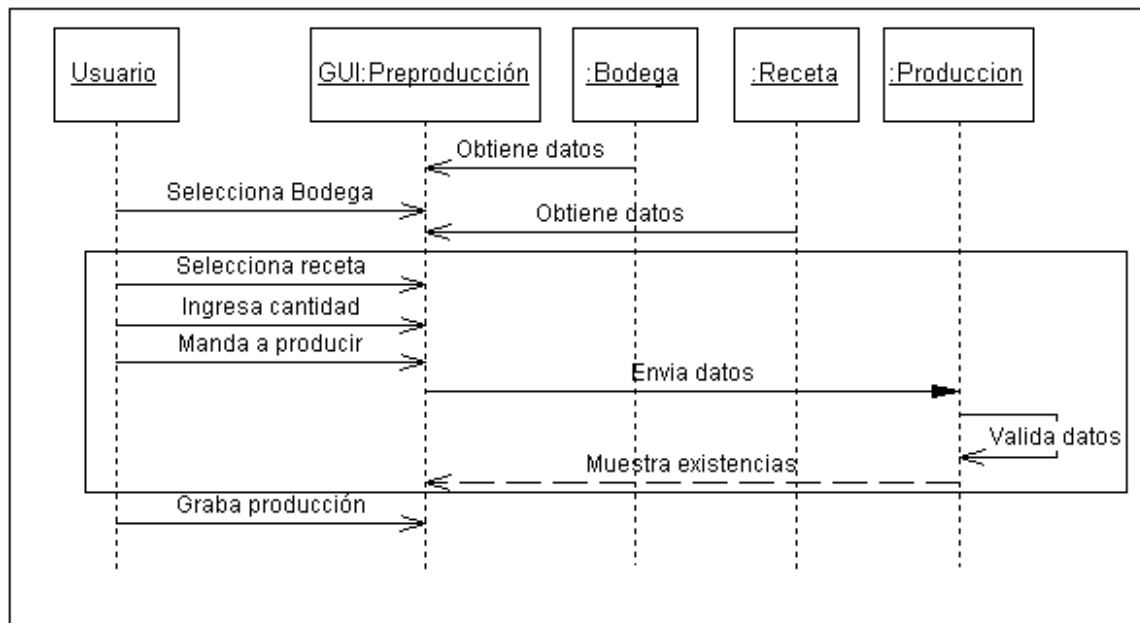


Figura 7.101. Diagrama de secuencia: Ingresar preproducción²⁸⁹

²⁸⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁸⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar preproducción

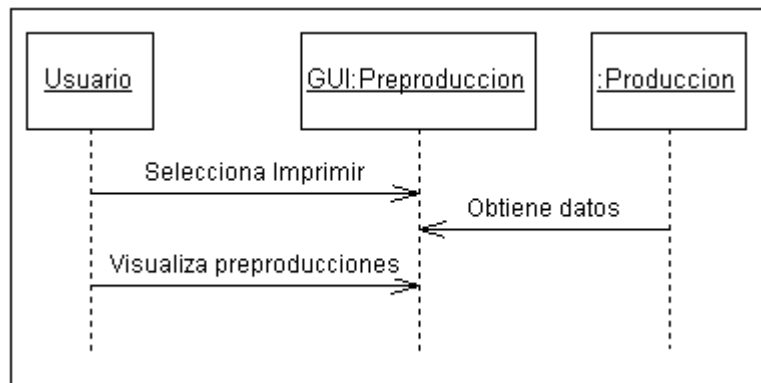


Figura 7.102. Diagrama de secuencia: Consultar preproducción²⁹⁰

Diagrama de secuencia: Imprimir preproducción

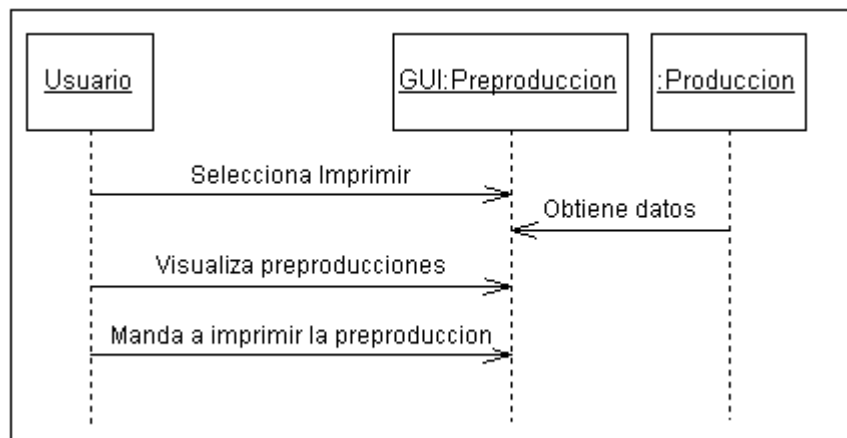


Figura 7.103. Diagrama de secuencia: Imprimir preproducción²⁹¹

²⁹⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar postproducción

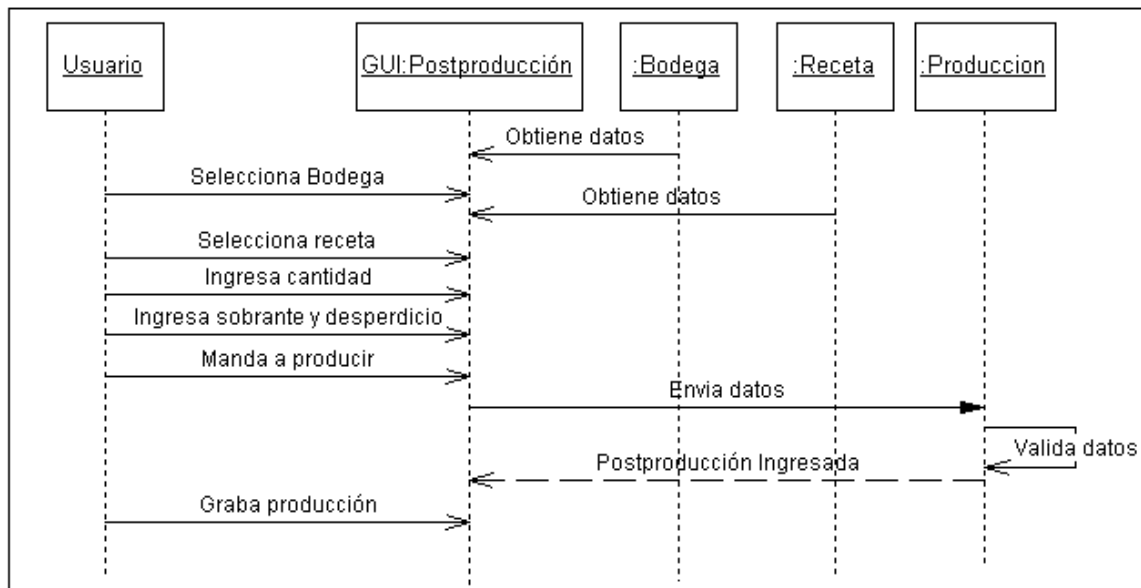


Figura 7.104. Diagrama de secuencia: Ingresar postproducción²⁹²

Diagrama de secuencia: Consultar postproducción

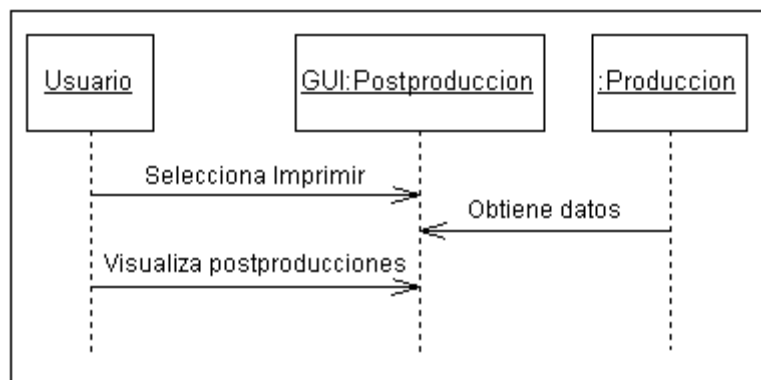


Figura 7.105. Diagrama de secuencia: Consultar postproducción²⁹³

²⁹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Imprimir postproducción

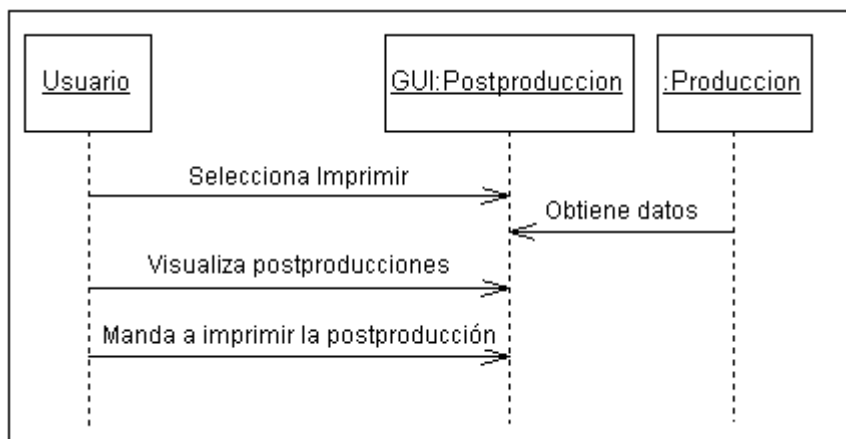


Figura 7.106. Diagrama de secuencia: Imprimir postproducción ²⁹⁴

Módulo de Facturación

Diagrama de secuencia: Ingresar cliente

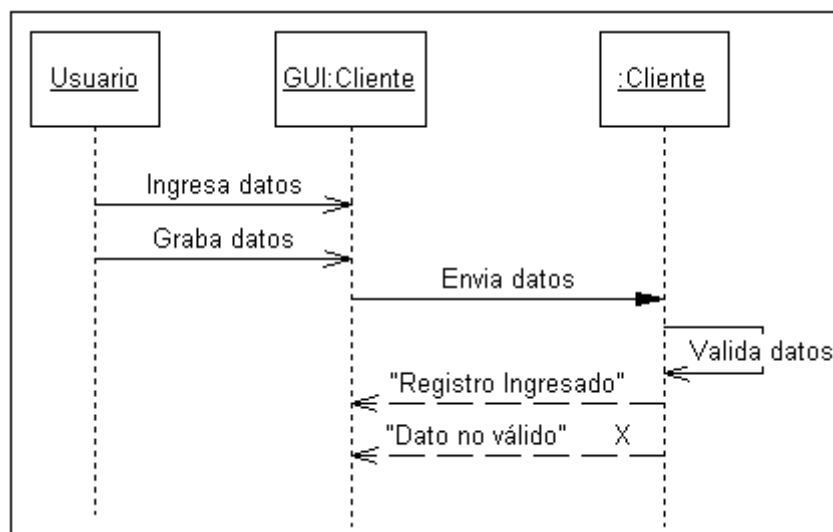


Figura 7.107. Diagrama de secuencia: Ingresar cliente ²⁹⁵

²⁹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Consultar cliente

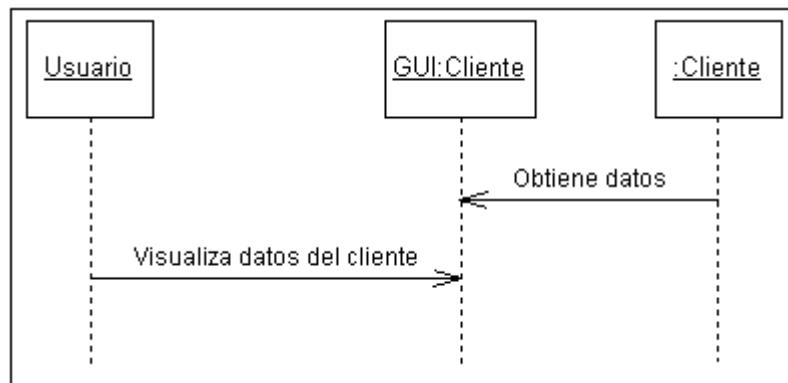


Figura 7.108. Diagrama de secuencia: Consultar cliente²⁹⁶

Diagrama de secuencia: Modificar cliente

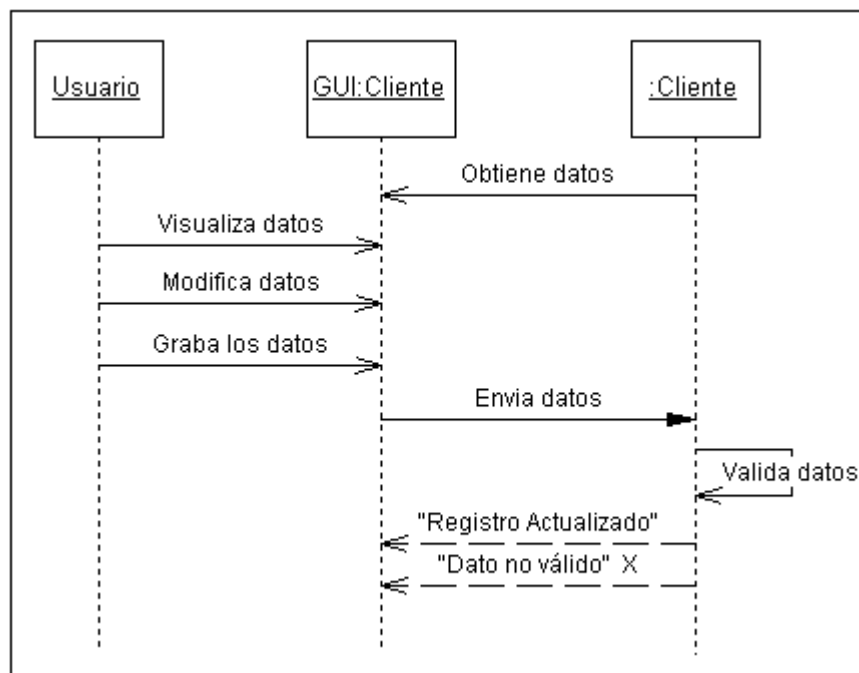


Figura 7.109. Diagrama de secuencia: Modificar cliente²⁹⁷

²⁹⁶ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁹⁷ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Eliminar cliente

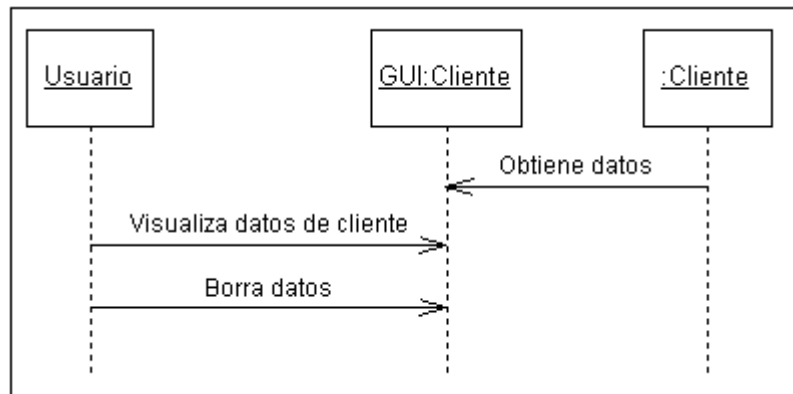


Figura 7.110. Diagrama de secuencia: Eliminar cliente ²⁹⁸

Diagrama de secuencia: Ingresar datos de factura

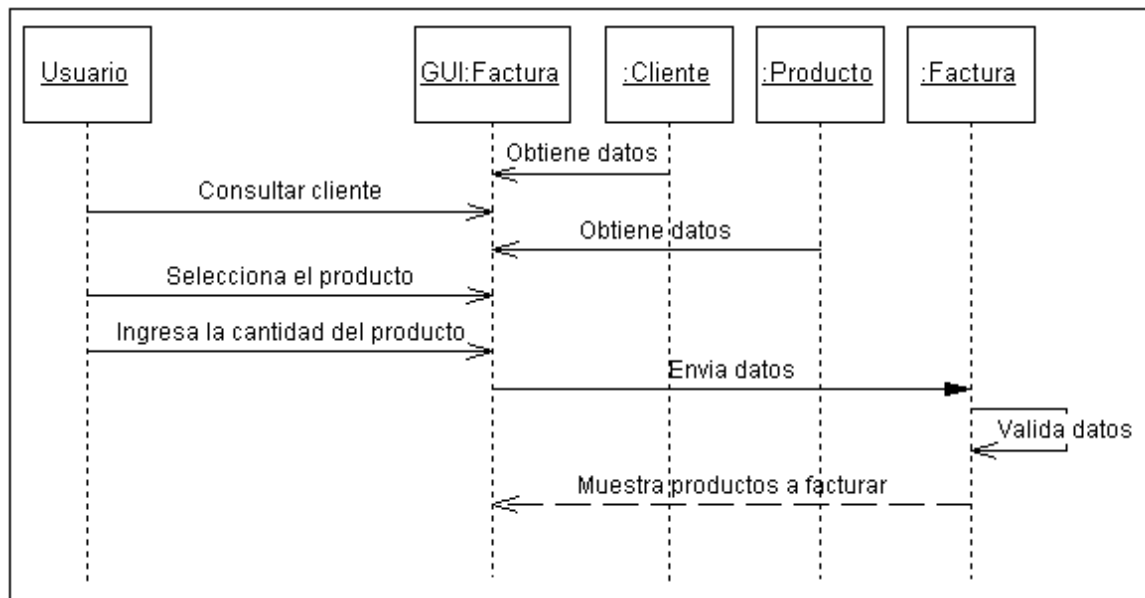


Figura 7.111. Diagrama de secuencia: Ingresar datos de factura ²⁹⁹

²⁹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

²⁹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Anular ítems de detalle de factura

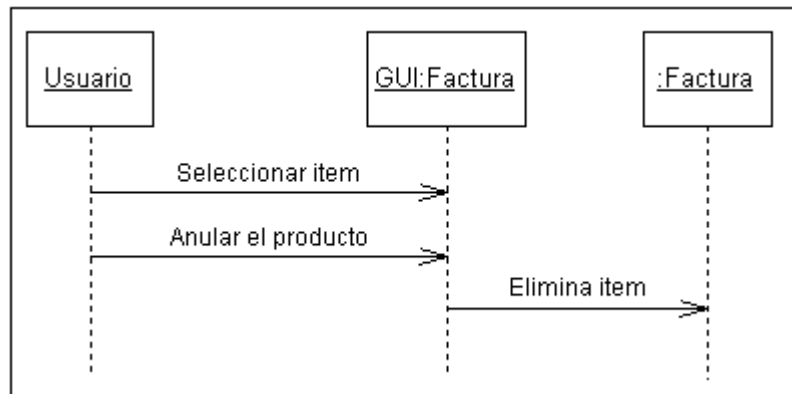


Figura 7.112. Diagrama de secuencia: Anular ítems de detalle de factura³⁰⁰

Diagrama de secuencia: Elegir forma de pago

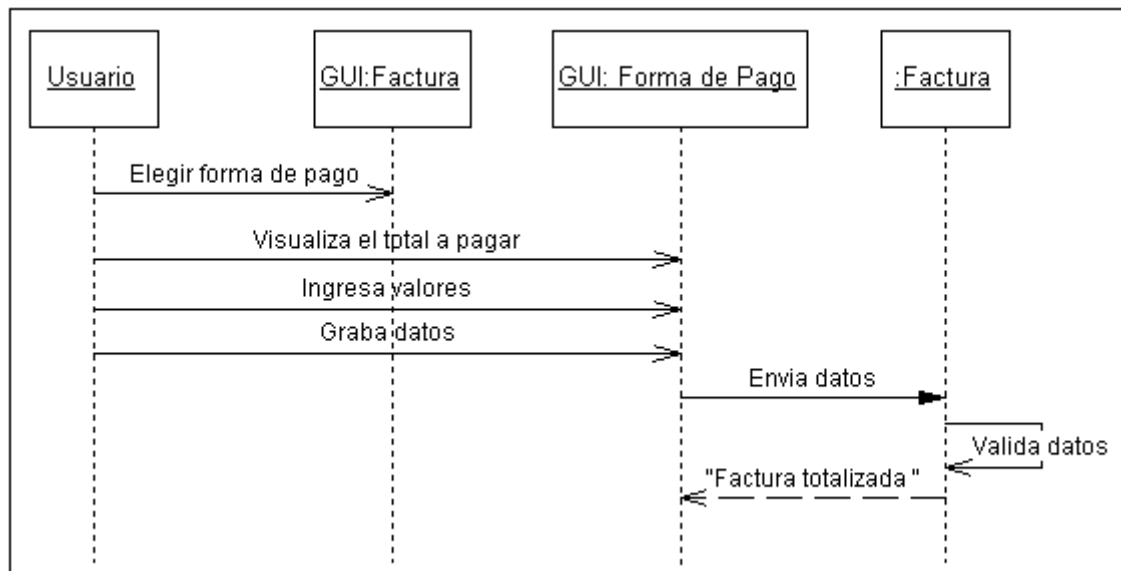


Figura 7.113. Diagrama de secuencia: Elegir forma de pago³⁰¹

³⁰⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁰¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Imprimir factura

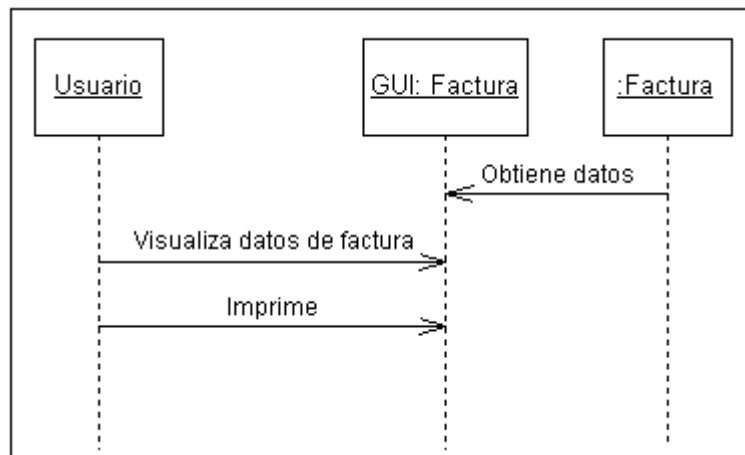


Figura 7.114. Diagrama de secuencia: Imprimir factura³⁰²

Diagrama de secuencia: Guardar cuenta en espera

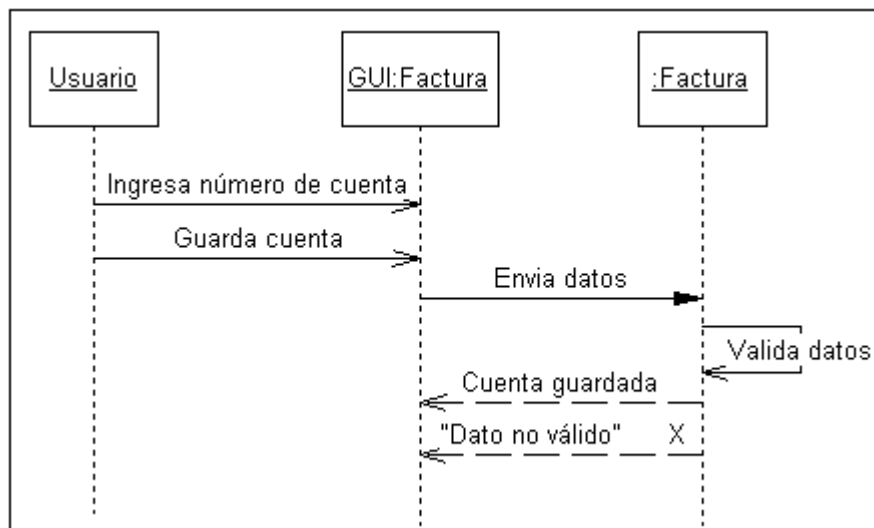


Figura 7.115. Diagrama de secuencia: Guardar cuenta en espera³⁰³

³⁰² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁰³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar descuentos y recargos

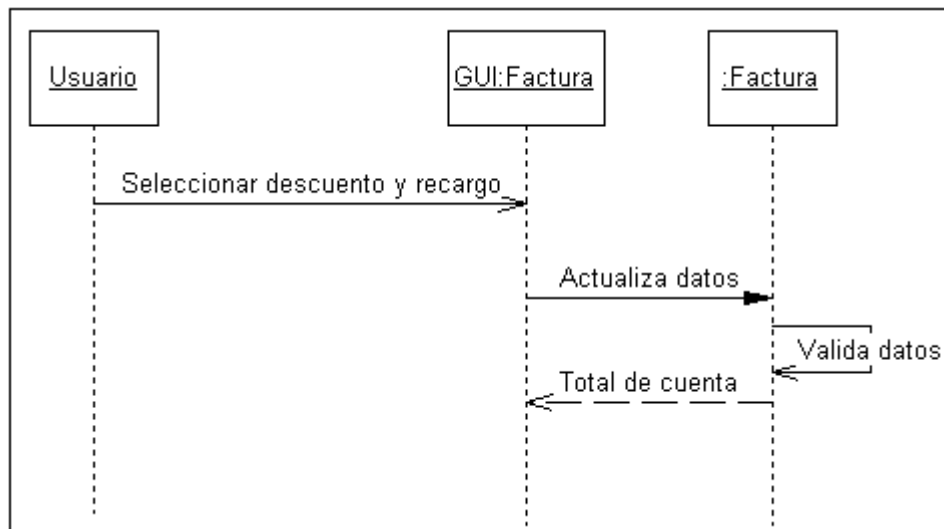


Figura 7.116. Diagrama de secuencia: Ingresar descuentos y recargos³⁰⁴

Diagrama de secuencia: Consultar cuentas

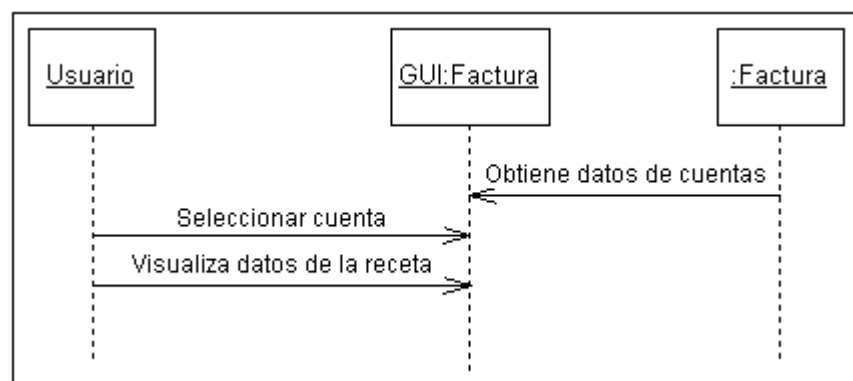


Figura 7.117. Diagrama de secuencia: Consultar cuentas³⁰⁵

³⁰⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁰⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Imprimir pre-cuenta

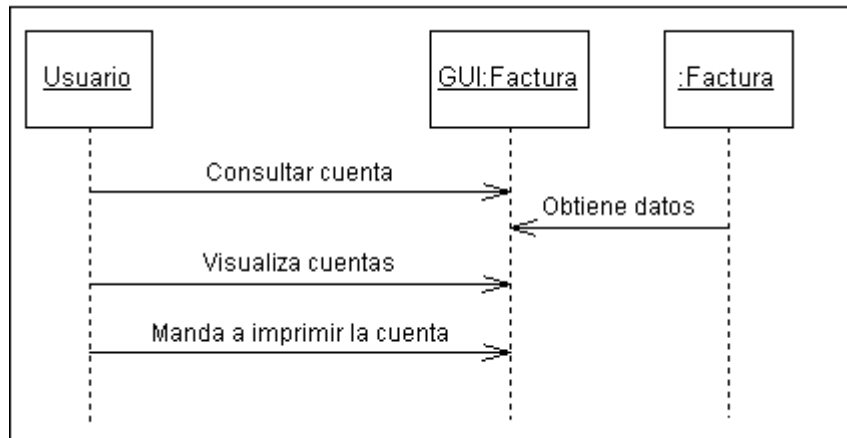


Figura 7.118. Diagrama de secuencia: Imprimir pre-cuenta ³⁰⁶

Diagrama de secuencia: Anular factura

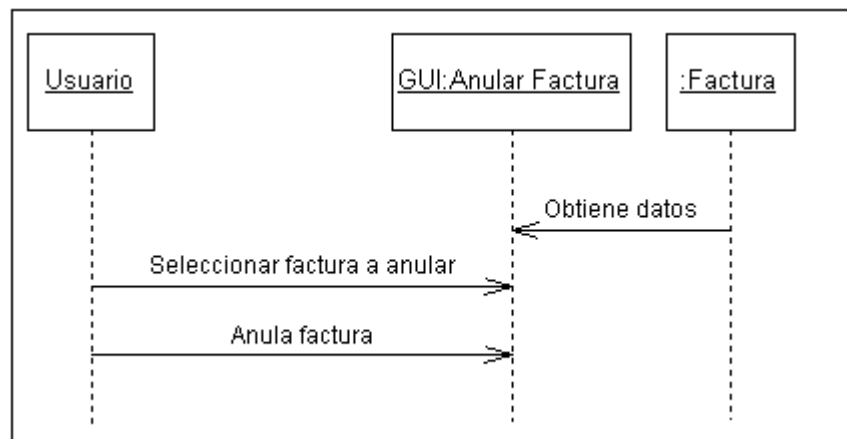


Figura 7.119. Diagrama de secuencia: Anular factura ³⁰⁷

³⁰⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁰⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Ingresar movimiento de caja

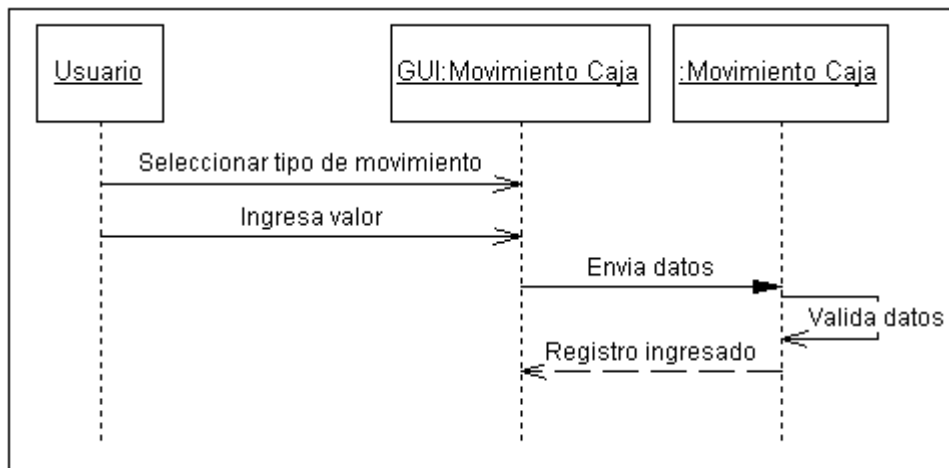


Figura 7.120. Diagrama de secuencia: Ingresar movimiento de caja³⁰⁸

Diagrama de secuencia: Cerrar caja

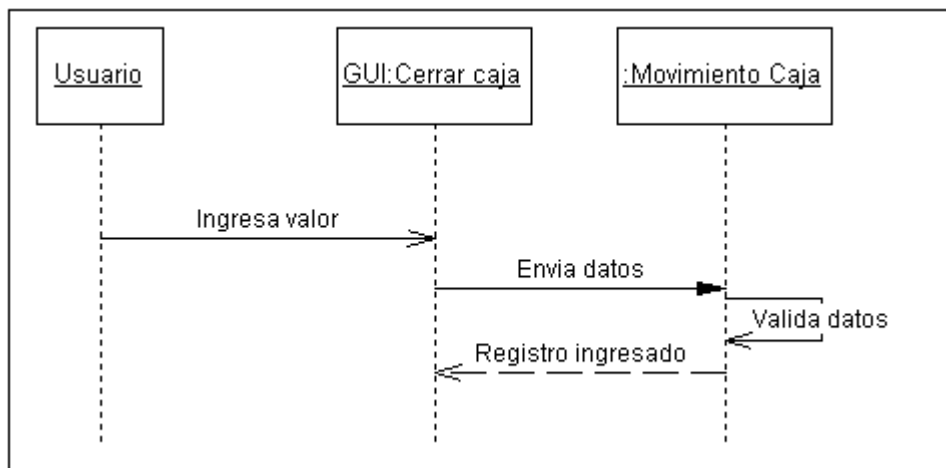


Figura 7.121. Diagrama de secuencia: Cerrar caja³⁰⁹

³⁰⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁰⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Diagrama de secuencia: Generar reporte

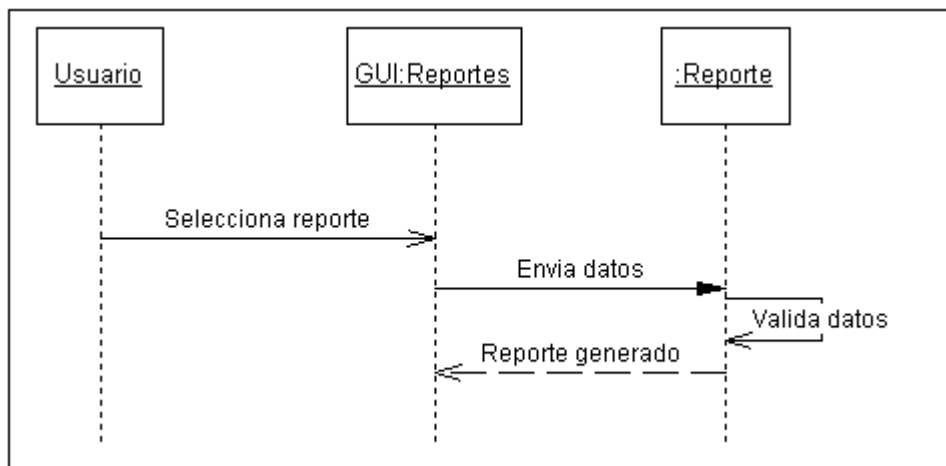


Figura 7.122. Diagrama de secuencia: Generar reporte³¹⁰

Diagrama de secuencia: Imprimir reporte

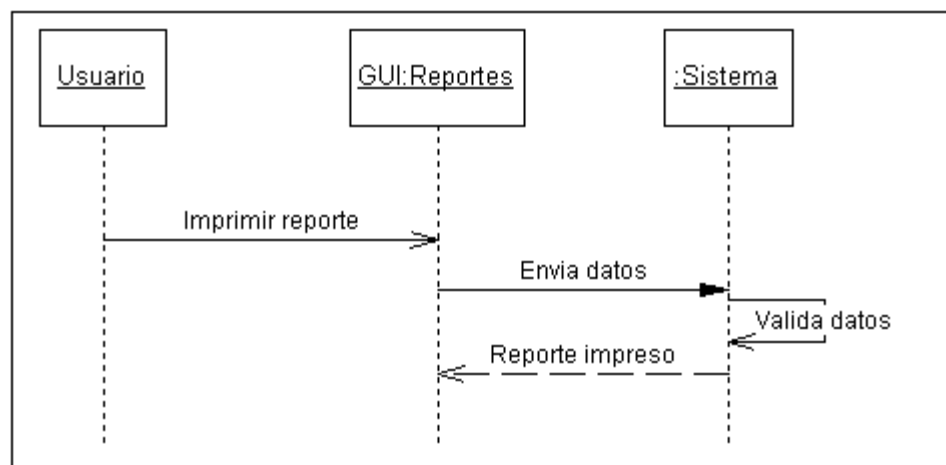


Figura 7.123. Diagrama de secuencia: Imprimir reporte³¹¹

³¹⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³¹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.2.3. DICCIONARIO DE CLASES SEGÚN MÓDULOS

Módulo de Administración

Empresa:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_EMPRESA	Contiene el código con el que se identifica a la empresa en el sistema	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
NOMBRE_EMPRESA	Contiene nombre el de la empresa.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
RUC_EMPRESA	Contiene el número de ruc de la empresa.	13 - Números	Long	No nulo	Solo números #Carac. >0 \$\$ <14
RAZON_SOCIAL_EMPRESA	Contiene la razón social de la empresa.	24 - Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
DIRECCION_EMPRESA	Contiene la dirección de la empresa.	32 - Texto	String		Todo mayúsculas #Carac. >0 \$\$ <33
TELEFONO1_EMPRESA	Contiene el número de teléfono de la empresa.	11-Números	Long	No nulo	Solo números #Carac. >0 \$\$ <12
TELEFONO2_EMPRESA	Contiene el número de celular de la empresa.	11-Números	Long		Solo números #Carac. >0 \$\$ <12
FAX_EMPRESA	Contiene el número de fax de la empresa.	11-Números	Long		Solo números #Carac. >0 \$\$ <12
CODIGO_POSTAL_EMPRESA	Contiene el código postal de	24-Números	Long	No nulo	Solo números

	la empresa.				#Carac. >0 \$\$ <25
EMAIL_EMPRESA	Contiene la dirección de correo electrónico de la empresa.	32 - Texto	String		#Carac. >0 \$\$ <33 Posea dominio, arroba.
ESTADO_EMPRESA	Contiene el estado de la empresa.	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.81. Diccionario de Clase: Empresa ³¹²

Sucursal:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_SUCURSAL	Contiene el código con el que se identifica a la sucursal en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_SUCURSAL	Contiene el nombre de la sucursal.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
ADMINISTRADOR_SUCURSAL	Contiene el nombre del administrador de la sucursal.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
ESTADO_SUCURSAL	Contiene el estado de la sucursal.	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.82. Diccionario de Clase: Sucursal ³¹³

³¹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³¹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Zona:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_SUCURSAL	Contiene el código con el que se identifica a la sucursal en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_ZONA	Contiene el nombre de la sucursal.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
SECUENCIAL_ZONA	Contiene el secuencial de la zona.	4 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <5

Tabla 7.83. Diccionario de Clase: Zona ³¹⁴

País:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_PAIS	Contiene el código con el que se identifica a un país en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_PAIS	Contiene el nombre del país.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.84. Diccionario de Clase: País ³¹⁵

³¹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³¹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Provincia:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_PROVINCIA	Contiene el código con el que se identifica a la provincia en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_PROVINCIA	Contiene el nombre de la provincia	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
SECUENCIAL_PROVINCIA	Contiene el secuencial de la provincia	4 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <5

Tabla 7.85. Diccionario de Clase: Provincia ³¹⁶

Ciudad:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_CIUADAD	Contiene el código con el que se identifica a la ciudad en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_CIUADAD	Contiene el nombre de la ciudad	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
SECUENCIAL_CIUADAD	Contiene el secuencial de la ciudad.	4 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <5

Tabla 7.86. Diccionario de Clase: Ciudad ³¹⁷

³¹⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Región:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_REGION	Contiene el código con el que se identifica a una región en el sistema.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_REGION	Contiene el nombre de la región.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.87. Diccionario de Clase: Región ³¹⁸

Usuario:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_USUARIO	Contiene el código con el que se identifica a un usuario en el sistema.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_USUARIO	Contiene el nombre del usuario.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
LOGIN_USUARIO	Contiene el nombre del usuario.	24 -Texto	String	No nulo No se puede repetir.	#Carac. >0 \$\$ <25
PASSWORD_USUARIO	Contiene la contraseña del usuario.	24-Texto	String	No nulo.	#Carac. >0 \$\$ <25 Se muestra la contraseña en

³¹⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³¹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

					caracteres especiales.
ESTADO_USUARIO	Contiene el estado del usuario.	16-Texto	String	No nulo	Todo mayúsculas
FECHA_CREACION_USUARIO	Contiene la fecha en que se creó el usuario.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
FECHA_CADUCIDAD_USUARIO	Contiene la fecha en que se caduca el usuario.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

Tabla 7.88. Diccionario de Clase: Usuario³¹⁹

Perfil:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_PERFIL	Contiene el código con el que se identifica a un perfil en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_PERFIL	Describe el nombre del perfil	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
ESTADO_ACCION_PERFIL	Contiene el estado de acción del perfil	16-Texto	String	No nulo	Todo mayúsculas
FECHA_CREACION_PERFIL	Contiene la fecha en que se creó el perfil	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

Tabla 7.89. Diccionario de Clase: Perfil³²⁰

³¹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Det_Acceso:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_ACCESO	Contiene el código con el que se identifica al detalle de acceso	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
PERMISO	Contiene el nombre del permiso al sistema.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.90. Diccionario de Clase: Det_Acceso³²¹

Det_Modulo_Sis:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_MODULO_SISTEMA	Contiene el código con el que se identifica a un módulo del sistema.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9

Tabla 7.91. Diccionario de Clase: Det_Modulo_Sis³²²

Modulo_Sis:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_MODULO	Contiene el código con el que se identifica a un	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9

³²⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³²¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³²² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	módulo del sistema.				
PERMISO	Contiene nombre del módulo del sistema.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.92. Diccionario de Clase: Modulo_Sis³²³

Proceso:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_MODULO	Contiene el código con el que se identifica a un módulo del sistema.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
PERMISO	Contiene nombre del módulo del sistema.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.93. Diccionario de Clase: Proceso³²⁴

Impuesto:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_IMPUESTO	Contiene el código con el que se identifica a un impuesto en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_IMPUESTO	Describe el nombre del impuesto.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

³²³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³²⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

PORCENTAJE_IMPUESTO	Contiene el porcentaje del impuesto.	6-Número	Integer	No nulo	Sólo números #Carac. >0 \$\$ <7
TIPO_IMPUESTO	Describe el tipo del impuesto.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.94. Diccionario de Clase: Impuesto ³²⁵

Descuento_Recargo:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DES_RECG	Contiene el código con el que se identifica a un descuento y recargo en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
DESCRIPCION_DES_RECG	Describe un descuento y recargo.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
PORCENTAJE_DES_RECG	Contiene el porcentaje del descuento y recargo.	6-Número	Integer	No nulo	Sólo números #Carac. >0 \$\$ <7
TIPO_DES_RECG	Describe el tipo de descuento y recargo.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

Tabla 7.95. Diccionario de Clase: Descuento_Recargo ³²⁶

³²⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³²⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Bodega:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_BODEGA	Contiene el código con el que se identifica a la bodega en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
NOMBRE_BODEGA	Contiene el nombre de la bodega.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
DIRECCION_BODEGA	Contiene la dirección de la bodega	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
ESTADO_BODEGA	Contiene el estado de la bodega.	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.96. Diccionario de Clase: Bodega ³²⁷

Linea_Producto:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_LINEA	Contiene el código con el que se identifica a una línea de producto en el sistema.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
DESCRIPCION_LINEA	Contiene el nombre de la línea del producto	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
OBSERVACION_LINEA	Contiene algún detalle de	32 -Texto	String	No nulo	Todo mayúsculas

³²⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	la línea del producto				#Carac. >0 \$\$ <33
--	-----------------------	--	--	--	---------------------

Tabla 7.97. Diccionario de Clase: Linea_Producto³²⁸

Grupo_Producto:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_GRUPO	Contiene el código con el que se identifica a un grupo de producto en el sistema.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
DESCRIPCION_GRUPO	Contiene el nombre del grupo del producto	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
OBSERVACION_GRUPO	Contiene algún detalle del grupo del producto	32 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <33

Tabla 7.98. Diccionario de Clase: Grupo_Producto³²⁹

Producto:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_PRODUCTO	Contiene el código con el que se identifica a un producto en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
DESCRIPCION_PRODUCTO	Contiene el detalle o nombre del producto	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

³²⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³²⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

PRECIO1_PRODUCTO	Contiene el valor de venta del producto según la temporada.	8 -Decimal	Double	No nulo	2 decimales
PRECIO2_PRODUCTO	Contiene el valor de venta del producto según la temporada.	8 -Decimal	Double		2 decimales
PRECIO3_PRODUCTO	Contiene el valor de venta del producto según la temporada.	8 -Decimal	Double		2 decimales
TIPO_PRODUCTO	Contiene el tipo de producto	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
ESTADO_PRODUCTO	Contiene el estado del producto.	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.99. Diccionario de Clase: Producto³³⁰

Receta:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_RECETA	Contiene el código con el que se identifica a una receta en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
DESCRIPCION_RECETA	Contiene el detalle o	24 -Texto	String	No nulo	Todo mayúsculas

³³⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	nombre de la receta				#Carac. >0 \$\$ <25
SECUENCIAL_RECETA	Se guarda el secuencial según la receta ingresada.	8 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <9
TIPO_PRODUCION	Es el tipo de producción	24 -Texto	String	Preproducción- postproducción	Todo mayúsculas #Carac. >0 \$\$ <25
FECHA_RECETA	Contiene la fecha de creación de la receta	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
ESTADO_PROCESO_RECETA	Contiene el estado de proceso de la receta	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.100. Diccionario de Clase: Receta³³¹

Det_Receta:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_RECETA	Contiene el código con el que se identifica a un detalle de receta en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
CANTIDAD	Contiene la cantidad del ingrediente de la receta.	6-Número	Integer	No nulo	Solo números

Tabla 7.101. Diccionario de Clase: Det_Receta³³²

³³¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Produccion:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_PRODUCION	Contiene el código con el que se identifica a una producción en el sistema	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
FECHA_PRODUCION	Contiene la fecha de creación de la producción.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
HORA_PRODUCION	Hora de creación de la producción	6-Hora	Date	No nulo	#Carac. >0 \$\$ <11
TOTAL_PRODUCION	Contiene el total de la producción.	6-Número	Integer	No nulo	Solo números
ESTADO_PRODUCION	Contiene el estado de producción	16-Texto	String	No nulo	Todo mayúsculas
SECUENCIAL_PRODUCION	Se almacena un secuencial de acuerdo a la producción realizada	8 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <9
TIPO_PRODUCION_RECE TA	Es el tipo de producción receta.	24 -Texto	String	Preproducción- postproducción	Todo mayúsculas #Carac. >0 \$\$ <25

*Tabla 7.102. Diccionario de Clase: Produccion*³³³

³³² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³³³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Det_Receta_Produccion:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_PRODUCION	Contiene el código con el que se identifica al detalle de receta de una producción	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
CANTIDAD_PRODUCIDA	Contiene la cantidad del ingrediente producido.	6-Número	Double	No nulo	Solo números 3 decimales
CANTIDAD_VENDIDA	Contiene la cantidad del ingrediente vendido.	6-Número	Double	No nulo	Solo números 3 decimales
CANTIDAD_DESPERDICIO	Contiene la cantidad del ingrediente desperdiciado.	6-Número	Double	No nulo	Solo números 3 decimales
CANTIDAD_EXISTENCIA_ACTUAL	Contiene la cantidad actual del ingrediente.	6-Número	Double	No nulo	Solo números 3 decimales
COSTO_RECETA_PRODUCION	Es el valor de la receta producida	6-Número	Double	No nulo	Solo números 2 decimales

*Tabla 7.103. Diccionario de Clase: Det_Receta_Produccion*³³⁴

Det_Produccion:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_PRODUCION	Contiene el código con el	8-Texto	String	Clave primaria.	#Carac. > 0 && <9

³³⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	que se identifica al detalle de una producción			No nulo.	
REFERENCIA_DET_PRODUCION	Contiene una referencia a la producción	8 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <9
CANTIDAD_DET_PRODUCION	Contiene la cantidad de la producción.	6-Número	Double	No nulo	Solo números 3 decimales
TIPO_DET_PRODUCION	Contiene el tipo de producción.	6-Número	Double	No nulo	Solo números 3 decimales
CANTIDAD_REAL_DET_PRODUCION	Contiene la cantidad real de la producción.	6-Número	Double	No nulo	Solo números 3 decimales
COSTO_DET_PRODUCION	Es el valor total de la producción.	6-Número	Double	No nulo	Solo números 2 decimales

Tabla 7.104. Diccionario de Clase: Det_Produccion³³⁵

Mov_Bodega:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_MOV_BODEGA	Contiene el código con el que se identifica a un movimiento de bodega.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
MOTIVO_MOV_BODEGA	Es el motivo por el cual	24 -Texto	String	No nulo	Todo mayúsculas

³³⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	se realiza el movimiento de bodega				#Carac. >0 \$\$ <25
DOCUMENTO_MOV_BODEGA	Es la identificación del documento de movimiento de bodega	8-Texto	String	No nulo.	#Carac. > 0 && <9
TIPO_MOV_BODEGA	Es el tipo de movimiento de bodega que se realiza.	8-Texto	String	No nulo.	#Carac. > 0 && <9
ESTADO_MOV_BODEGA	Es el estado del movimiento de bodega	16-Texto	String	No nulo	Todo mayúsculas
FECHA_MOV_BODEGA	Contiene la fecha en que se realizó el movimiento de bodega.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
HORA_MOV_BODEGA	Contiene la hora en que se realizó el movimiento de bodega.	6-Hora	Date	No nulo	#Carac. >0 \$\$ <11
TIPO_COSTO_MOV_BODEGA	Es el tipo del costo del movimiento de bodega.	8-Texto	String	No nulo.	#Carac. > 0 && <9
NOMBRE_BODEGA_DESTINO	Se define el nombre de la bodega destino	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
ESTADO_PROCESO_MOV_BODEGA	Es el estado del proceso del movimiento de	16-Texto	String	No nulo	Todo mayúsculas

	bodega				
FECHA_CREACION_MOV_BOD EGA	Contiene la fecha de creación del movimiento de bodega.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

Tabla 7.105. Diccionario de Clase: Mov_Bodega³³⁶

Toma_Fisica:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_TOMA_FISICA	Contiene el código con el que se identifica a una toma física.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
TIPO_TOMA_FISICA	Es el tipo de la toma física que se realiza.	8-Texto	String	No nulo.	#Carac. > 0 && <9
ESTADO_TOMA_FISICA	Es el estado de la toma física.	16-Texto	String	No nulo	Todo mayúsculas
FECHA_TOMA_FISICA	Contiene la fecha en que se realizó la toma física.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

Tabla 7.106. Diccionario de Clase: Toma_Fisica³³⁷

³³⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³³⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Det_Toma_Producto:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_TOMA_FISICA	Contiene el código con el que se identifica a una toma física.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
INV_INICIAL_DET_TOMA_PRODUCTO	Es la cantidad inicial tomada del producto	6-Número	Integer	No nulo	Solo números
EXIST_ACTUAL_DET_TOMA_PRODUCTO	Es la cantidad de las existencias actuales del producto.	6-Número	Integer	No nulo	Solo números
CANT_TOMA_DET_TOMA_PRODUCTO	Contiene la cantidad de la toma del producto	6-Número	Integer	No nulo	Solo números
COSTO_INICIAL_TOMA_PRODUCTO	El el valor inicial del producto.	6-Número	Double	No nulo	Solo números 2 decimales
ESTADO_DET_TOMA_PRODUCTO	Contiene el estado del detalle de la toma del producto.	16-Texto	String	No nulo	Todo mayúsculas

*Tabla 7.107. Diccionario de Clase: Det_Toma_Producto*³³⁸

³³⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Det_Mov_Bodega:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
ID_DET_MOV_BODEGA	Contiene el código con el que se identifica el detalle del movimiento de bodega.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
CANTIDAD_DET_MOV_BODEGA	Es la cantidad del movimiento de bodega.	6-Número	Integer	No nulo	Solo números
COSTO_DET_MOV_BODEGA	Es el costo del movimiento de bodega realizado.	6-Número	Double	No nulo	Solo números 2 decimales

Tabla 7.108. Diccionario de Clase: Det_Mov_Bodega³³⁹

Kardex:

Campo	Descripción	Long. – Tipo	Formato	Tipo de Restricción	Validación
KARDEX_ID	Contiene el código con el que se identifica el kardex.	8-Texto	String	Clave primaria. No nulo.	#Carac. > 0 && <9
FECHA_KARDEX	Es la fecha en que se realiza el kardex.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

³³⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

HORA_KARDEX	Contiene la hora en que se realizó el kardex.	6-Hora	Date	No nulo	#Carac. >0 \$\$ <11
COD_TRANSACCION_KARDEX	Se define el código de transacción realizada.	8-Texto	String	No nulo.	#Carac. > 0 && <9
TIPO_TRANSACCION_KARDEX	Es el tipo de transacción realizada.	8-Texto	String	No nulo.	#Carac. > 0 && <9
TIPO_MOVIMIENTO_KARDEX	Es el tipo de movimiento del kardex.	8-Texto	String	No nulo.	#Carac. > 0 && <9
COD_CLIENTE_PROV_DOCUMENTO	Es el código del cliente provisional del documento.	8-Texto	String	No nulo.	#Carac. > 0 && <9
DESCRIPCION_CLIENTE_PROV_DOC	Es la descripción del cliente	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
CANTIDAD_KARDEX	Contiene la cantidad del kardex	6-Número	Integer	No nulo	Solo números
COSTO_KARDEX	Es el costo del kardex	6-Número	Double	No nulo	Solo números 2 decimales
TOTAL_KARDEX	Contiene el total del kardex.	6-Número	Double	No nulo	Solo números 2 decimales
CANTIDAD_EXISTENCIA_KARDEX	Es la cantidad de existencias en el kardex.	6-Número	Integer	No nulo	Solo números
COSTO_EXISTENCIA_KARDEX	Es el costo de la	6-Número	Double	No nulo	Solo números

	existencia del kardex.				2 decimales
TOTAL_EXISTENCIA_KARDEX	Es el total de la existencia del kardex.	6-Número	Integer	No nulo	Solo números
SECUENCIAL_KARDEX	Se define un secuencial según el kardex ingresado.	6-Número	Integer	No nulo	Solo números
ESTADO_RECALCULO_COSTOS	Es el estado del recálculo de costos.	16-Texto	String	No nulo	Todo mayúsculas
TIPO_COSTO_MOV_KARDEX	Es el tipo de costo del movimiento del kardex.	8-Texto	String	No nulo.	#Carac. > 0 && <9

Tabla 7.109. Diccionario de Clase: Kardex³⁴⁰

Ciente:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_CLIENTE	Contiene el código con el que se identifica al cliente en el sistema	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
NOMBRE_CLIENTE	Contiene el nombre del cliente.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
APELLIDO_CLIENTE	Contiene el apellido del cliente.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25

³⁴⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

NOMBRE_COMERCIAL_CLIENTE	Contiene el nombre comercial del cliente.	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
CEDULA_RUC_CLIENTE	Contiene el número de ruc o cédula del cliente.	13 - Números	Long	No nulo	Solo números #Carac. >0 \$\$ <14
DIRECCION_CLIENTE_PROVEEDOR	Contiene la dirección del proveedor o cliente	24 - Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
CODIGO_POSTAL_CLIENTE	Contiene el código postal del cliente o proveedor	24-Números	Long	No nulo	Solo números #Carac. >0 \$\$ <25
TELEFONO1_CLIENTE_PROVEEDOR	Contiene el número de teléfono del cliente o proveedor.	11-Números	Long	No nulo	Solo números #Carac. >0 \$\$ <12
TELEFONO2_CLIENTE_PROVEEDOR	Contiene el número de celular del cliente o proveedor	11-Números	Long		Solo números #Carac. >0 \$\$ <12
FAX_CLIENTE_PROVEEDOR	Contiene el número de fax del cliente.	11-Números	Long		Solo números #Carac. >0 \$\$ <12
EMAIL_CLIENTE_PROVEEDOR	Contiene la dirección de correo electrónico del cliente.	32 - Texto	String		#Carac. >0 \$\$ <33 Posea dominio, arroba.
TIPO_PRECIO_CLIENTE	Contiene el tipo de precio que corresponde al cliente	8-Texto	String	No nulo.	#Carac. > 0 && <9
OBSERVACION_CLIENTE	Es algún detalle que se desee ingresar en el cliente	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$ <25
FECHA_CREACION_CLIENTE	Es la fecha de creación del	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

	cliente				
PORC_DSCTO_CLIENTE	Contiene el porcentaje de descuento del cliente	6-Número	Integer	No nulo	Sólo números #Carac. >0 \$\$ <7

Tabla 7.110. Diccionario de Clase: Cliente ³⁴¹

Factura:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_FACTURA	Contiene el código con el que se identifica a la factura en el sistema	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
NUMERO_FACTURA	Es el número de factura	8-Texto	String	No nulo.	#Carac. > 0 && <9
FECHA_FACTURA	Contiene la fecha de creación de la factura	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
HORA_FACTURA	Contiene la hora en que se emitió la factura	6-Hora	Date	No nulo	#Carac. >0 \$\$ <11
SUBTOTAL_FACTURA	Es el subtotal de la factura	6-Número	Double	No nulo	Solo números 2 decimales
DESCUENTO_FACTURA	Contiene el descuento total de la factura	6-Número	Double		Solo números 2 decimales
RECARGO_FACTURA	Es el recargo de la factura	6-Número	Double		Solo números 2 decimales
IMPUESTO_FACTURA	Contiene el impuesto al total	6-Número	Double		Solo números

³⁴¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

	de la factura				2 decimales
TOTAL_FACTURA	Es la cantidad total de la factura	6-Número	Double	No nulo	Solo números 2 decimales
ESTADO_FACTURA	Contiene el estado en el que se encuentra la factura	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.111. Diccionario de Clase: Factura ³⁴²

Det_Factura:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_DET_FACTURA	Contiene el código con el que se identifica al detalle de factura	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
CANTIDAD_DET_FACTURA	Contiene la cantidad de los productos de la factura	6-Número	Integer	No nulo	Solo números
NOMBRE_DET_FACTURA	Contiene el nombre del producto	24 -Texto	String	No nulo	Todo mayúsculas #Carac. >0 \$\$\$ <25
PRECIO_UNITARIO_DET_FACTURA	Es el precio unitario del producto	6-Número	Double	No nulo	Solo números 2 decimales
PRECIO_TOTAL_DET_FACTURA	Es el precio total del detalle de factura.	6-Número	Double	No nulo	Solo números 2 decimales
ESTADO_DET_FACTURA	Contiene el estado actual del detalle de factura	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.112. Diccionario de Clase: Det_Factura ³⁴³

³⁴² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Det_Factura_Pago:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_DET_FACT_PAGO	Contiene el código con el que se identifica al detalle de pago factura	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
CANTIDAD_DET_PAGO	Es la cantidad total de pago de factura.	6-Número	Integer	No nulo	Solo números
FECHA_DET_PAGO	Contiene la fecha en que se realizó el pago de la factura.	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11

Tabla 7.113. Diccionario de Clase: Det_Factura_Pago³⁴⁴

Cuadre_Caja:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_CUADRE_CAJA	Contiene el código con el que se identifica al cuadro de caja.	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
FECHA_CUADRE_CAJA	Es la fecha en que se realizó el cuadro de caja	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
HORA_CUADRE_CAJA	Es la hora en que se realizó el cuadro de caja	6-Hora	Date	No nulo	#Carac. >0 \$\$ <11
TOTAL_CUADRE_CAJA	Es el total del cuadro de caja	6-Número	Double	No nulo	Solo números

³⁴³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁴⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

					2 decimales
CANT_EST_CUADRE_CAJA	Es la cantidad del cuadro de caja	6-Número	Integer	No nulo	Solo números
ESTADO_CUADRE_CAJA	Contiene el estado en el que se encuentra el cuadro de caja	16-Texto	String	No nulo	Todo mayúsculas
ESTADO_USUARIO_C_CAJA	Contiene el estado en el que se encuentra el usuario que cuadra la caja	16-Texto	String	No nulo	Todo mayúsculas

Tabla 7.114. Diccionario de Clase: Cuadre_Caja ³⁴⁵

Det_Cuadre_Caja:

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_DET_CC	Contiene el código con el que se identifica al detalle de cuadro de caja	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
CANTIDAD_DET_CC	Es la cantidad en detalle del cuadro de caja	6-Número	Integer	No nulo	Solo números
TOTAL_DET_CC	Es el total en detalle del cuadro de caja	6-Número	Double	No nulo	Solo números 2 decimales

Tabla 7.115. Diccionario de Clase: Det_Cuadre_Caja ³⁴⁶

³⁴⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁴⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Movimiento_Caja

Campo	Descripción	Longitud -Tipo	Formato	Tipo de Restricción	Validación
ID_MOVIMIENTO_CAJA	Contiene el código con el que se identifica a un movimiento de caja	4 - Caracter	String	Clave primaria. No nulo.	#Carac. > 0 && <5
VALOR_MOVIMIENTO_CAJA	Es el valor del movimiento de caja	6-Número	Double	No nulo	Solo números 2 decimales
FECHA_MOVIMIENTO_CAJA	Es la fecha en que se realizó el movimiento de caja	10 – Fecha	Date	No nulo	#Carac. >0 \$\$ <11
HORA_MOVIMIENTO_CAJA	Es la hora en que se realizó el movimiento de caja	6-Hora	Date	No nulo	#Carac. >0 \$\$ <11
ESTADO_MOVIMIENTO_CAJA	Es el estado en el que se encuentra el movimiento de caja.	16-Texto	String	No nulo	Todo mayúsculas
TIPO_MOVIMIENTO_CAJA	Es el tipo de movimiento de caja	8-Texto	String	No nulo.	#Carac. > 0 && <9
MOVIMIENTO_CIERRE_CAJA	Es el movimiento que se realiza al cerrar la caja	8-Texto	String	No nulo.	#Carac. > 0 && <9

Tabla 7.116. Diccionario de Clase: *Movimiento_Caja*³⁴⁷

³⁴⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.3. MODELO DE DISEÑO

2.3.1. DISEÑO DE OBJETOS

En el diseño del sistema se puede identificar la siguiente estructura de objetos:

Administración:

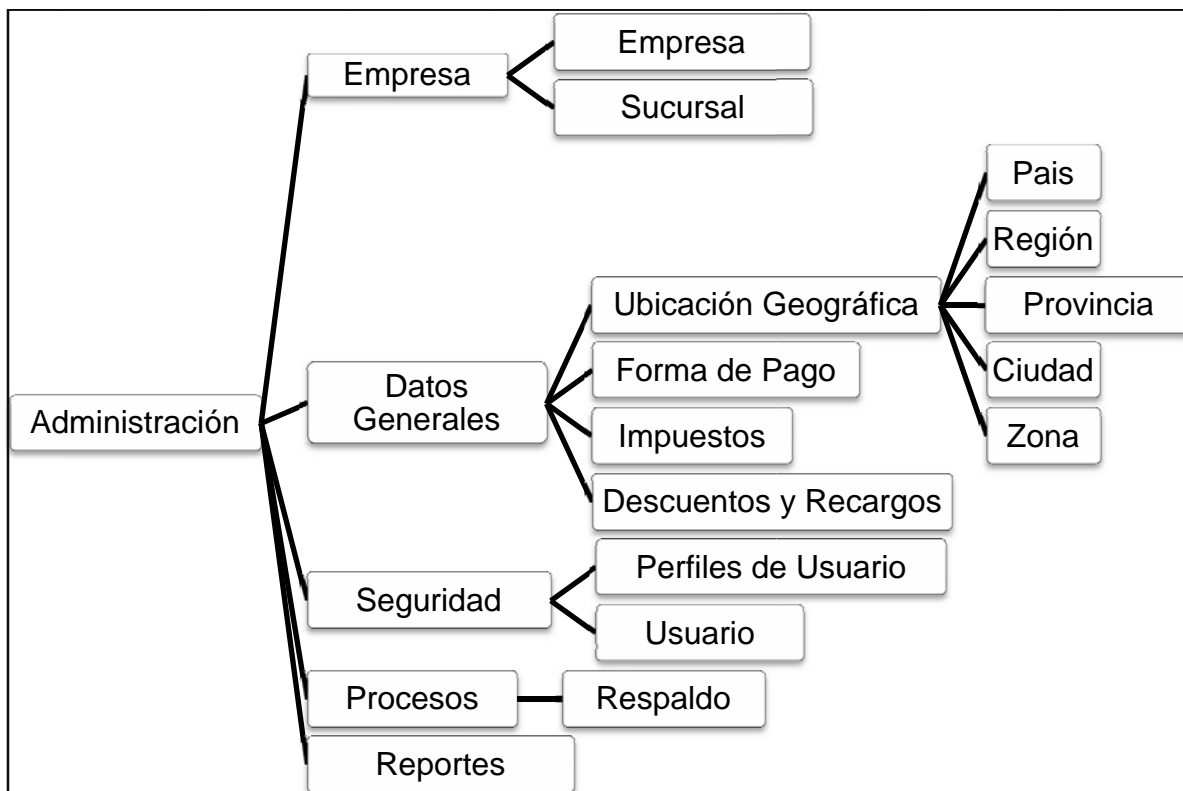


Figura 7.124. Objeto: Administración ³⁴⁸

³⁴⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Inventario:

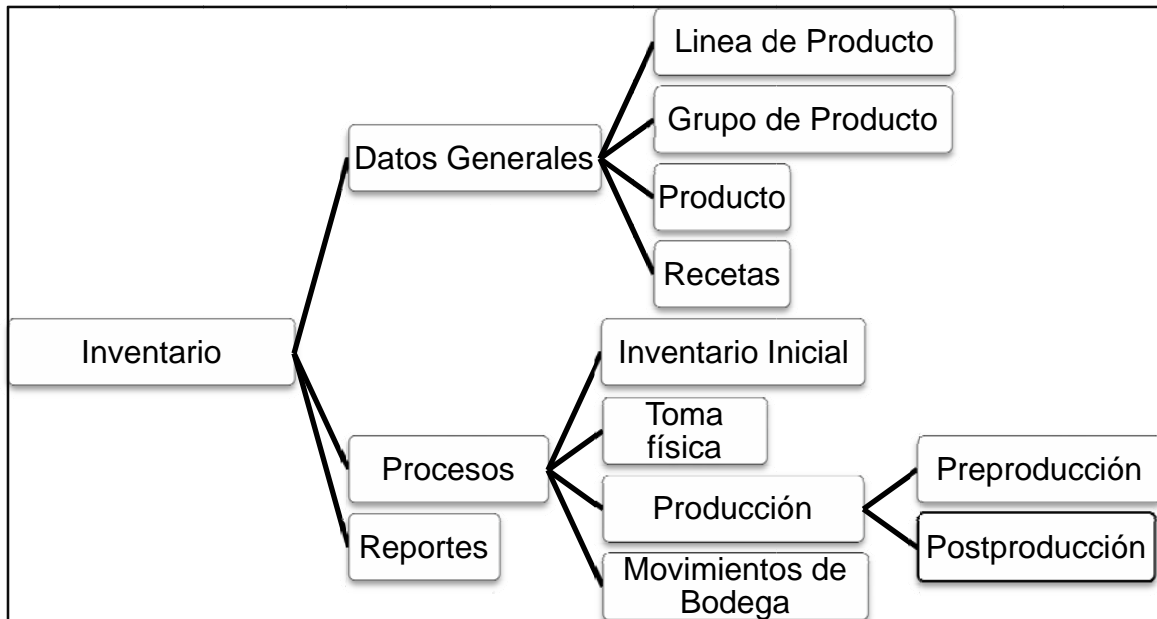


Figura 7.125. Objeto: Inventario ³⁴⁹

Facturación:

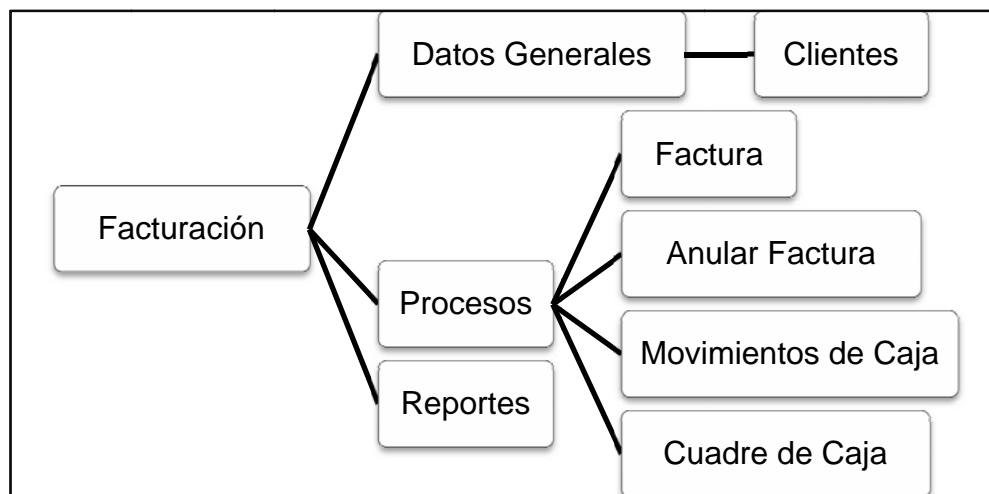


Figura 7.126. Objeto: Facturación ³⁵⁰

³⁴⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁵⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.3.2. DISEÑO DE INTERFASES

En el diseño de interfaces se describe las pantallas que forman parte del sistema; para revisar la interfaz del sistema de restaurantes dirigirse al Manual de usuario.³⁵¹

2.3.3. DIAGRAMA ENTIDAD RELACIÓN

Módulo Administración - Estructura Organizacional

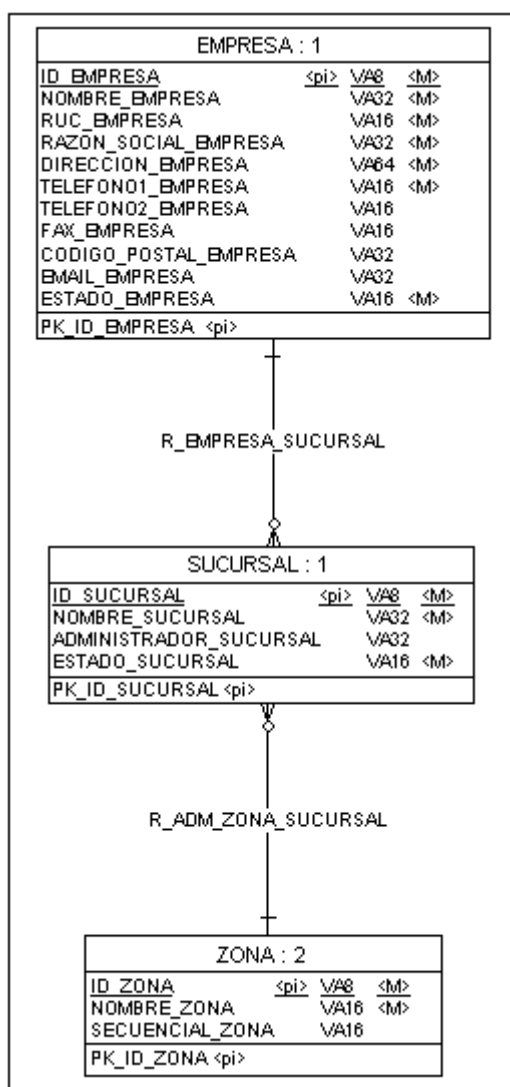


Figura 7.127. Módulo Administración – Estructura Organizacional³⁵²

³⁵¹ Manual de Usuario

Módulo Administración – Ubicación Geográfica

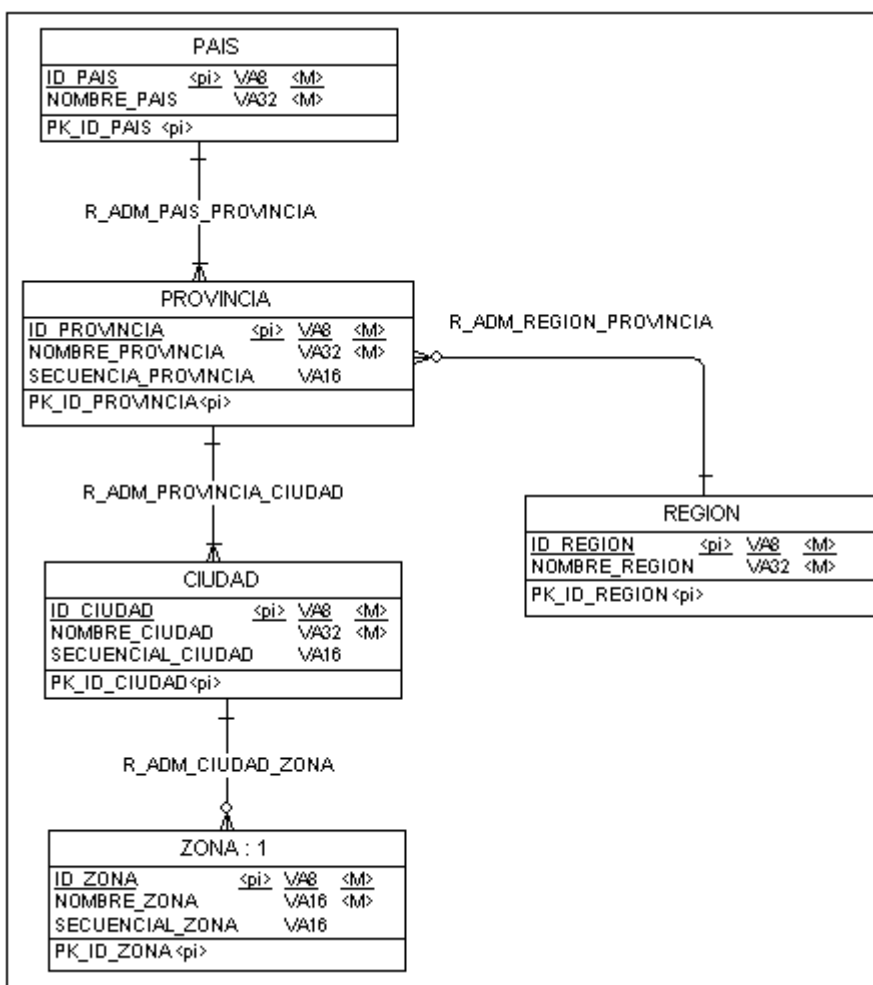


Figura 7.128. Módulo Administración – Ubicación Geográfica ³⁵³

³⁵² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁵³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Administración – Usuario y Perfil

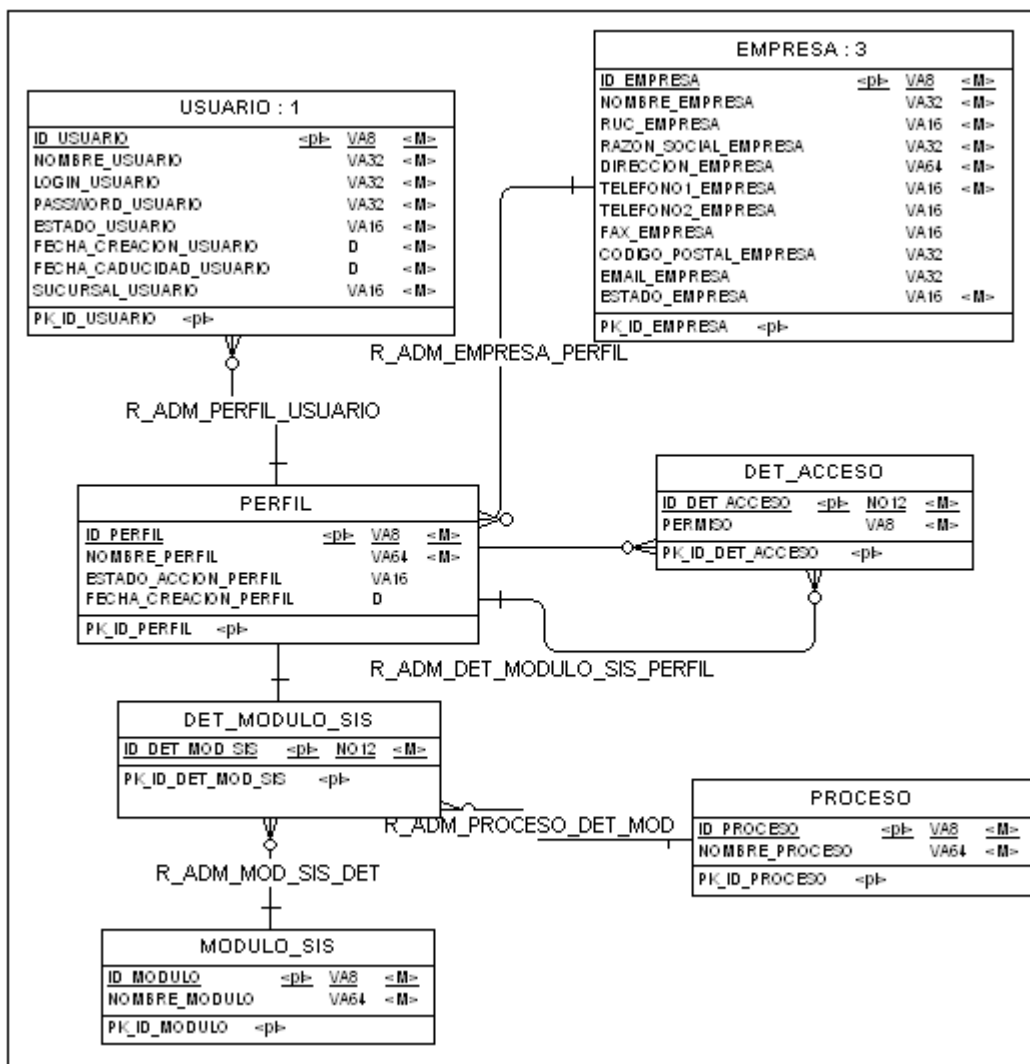


Figura 7.129. Módulo Administración – Usuario y perfil ³⁵⁴

³⁵⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Administración – IVA, Recargos y Descuentos

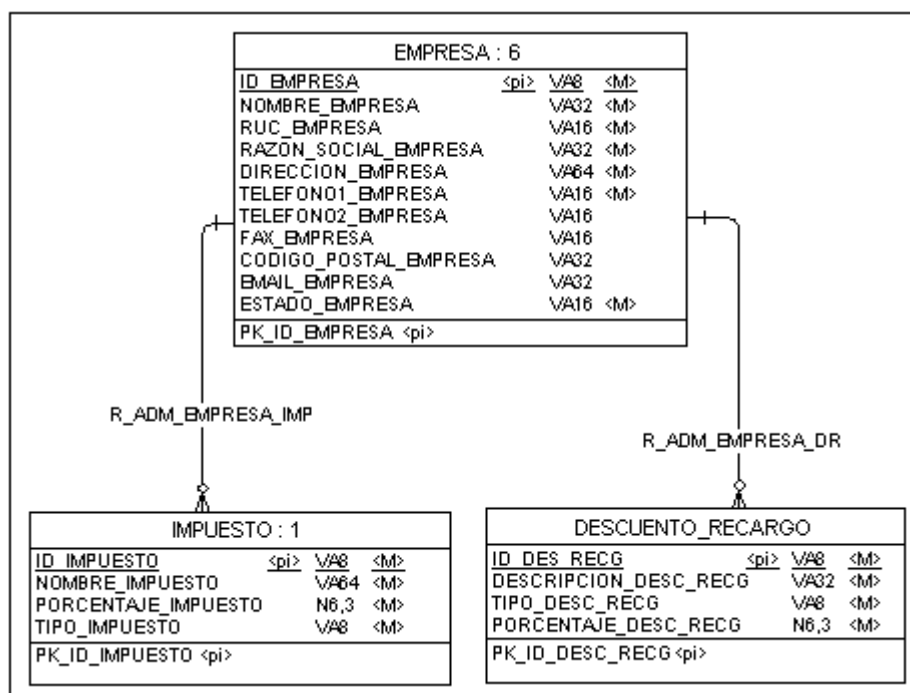


Figura 7.130. Módulo Administración – IVA, Recargos y Descuentos ³⁵⁵

Módulo Inventario – Bodega

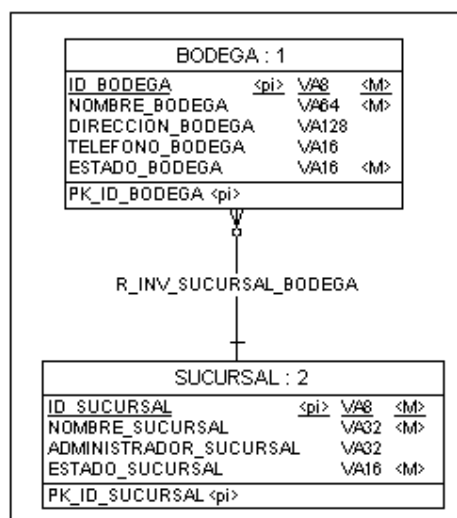


Figura 7.131. Módulo Inventario - Bodega ³⁵⁶

³⁵⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁵⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Inventario - Producto

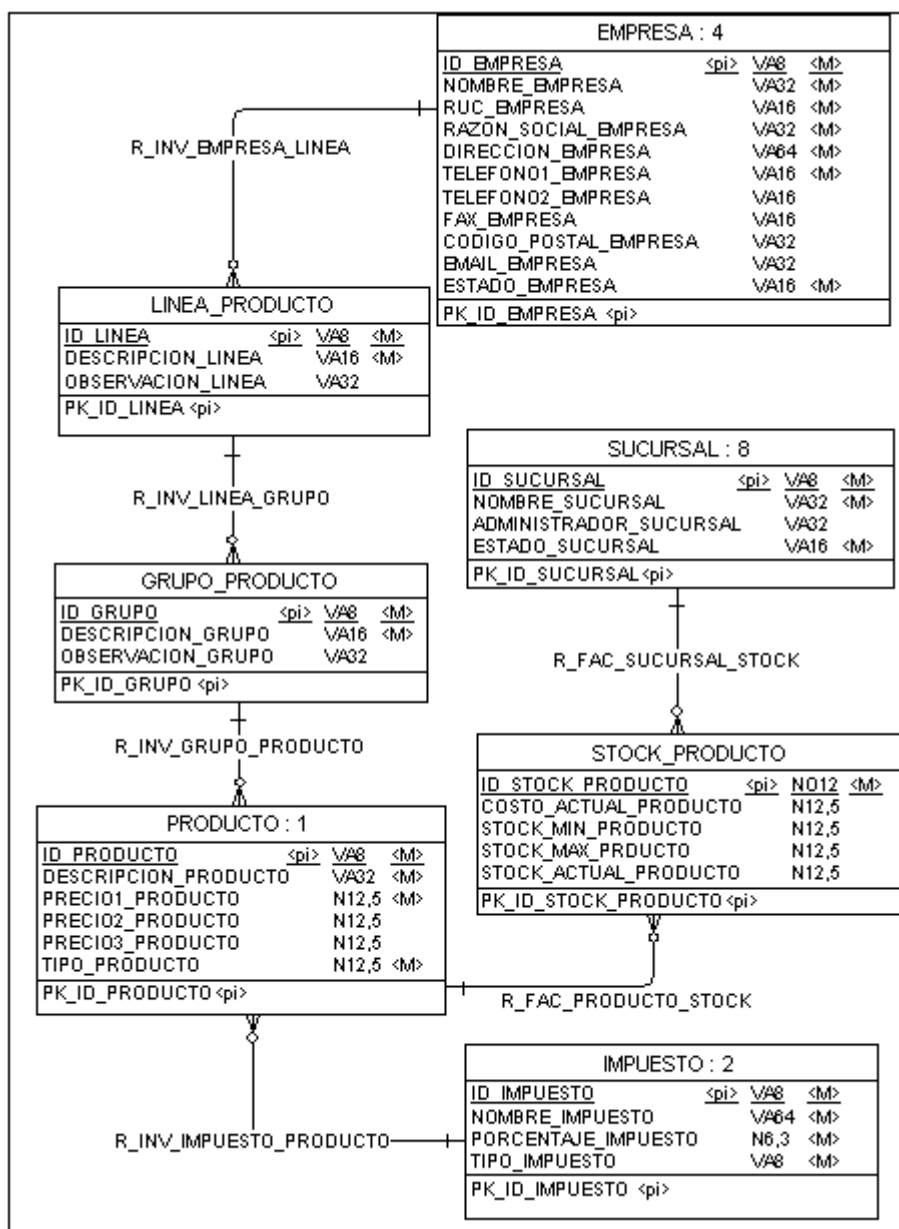


Figura 7.132. Módulo Inventario – Producto ³⁵⁷

³⁵⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Inventario – Receta

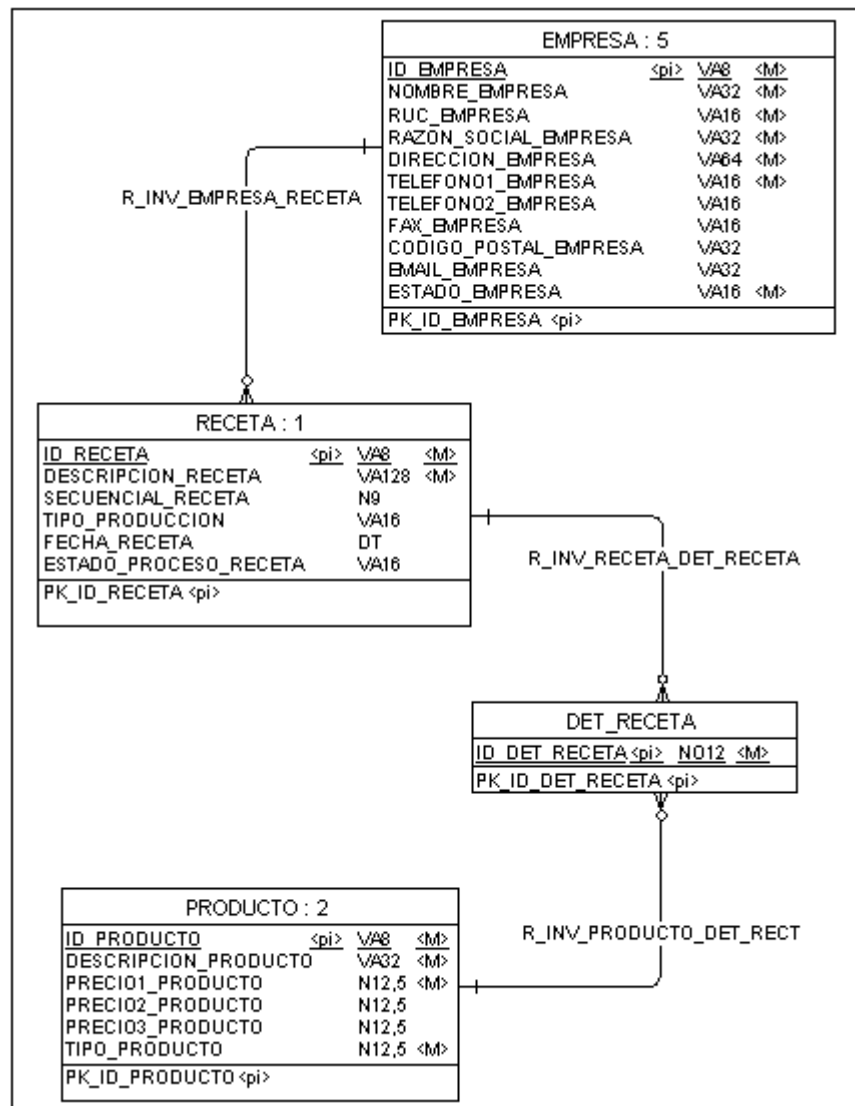


Figura 7.133. Módulo Inventario - Receta³⁵⁸

³⁵⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Inventario – Producción

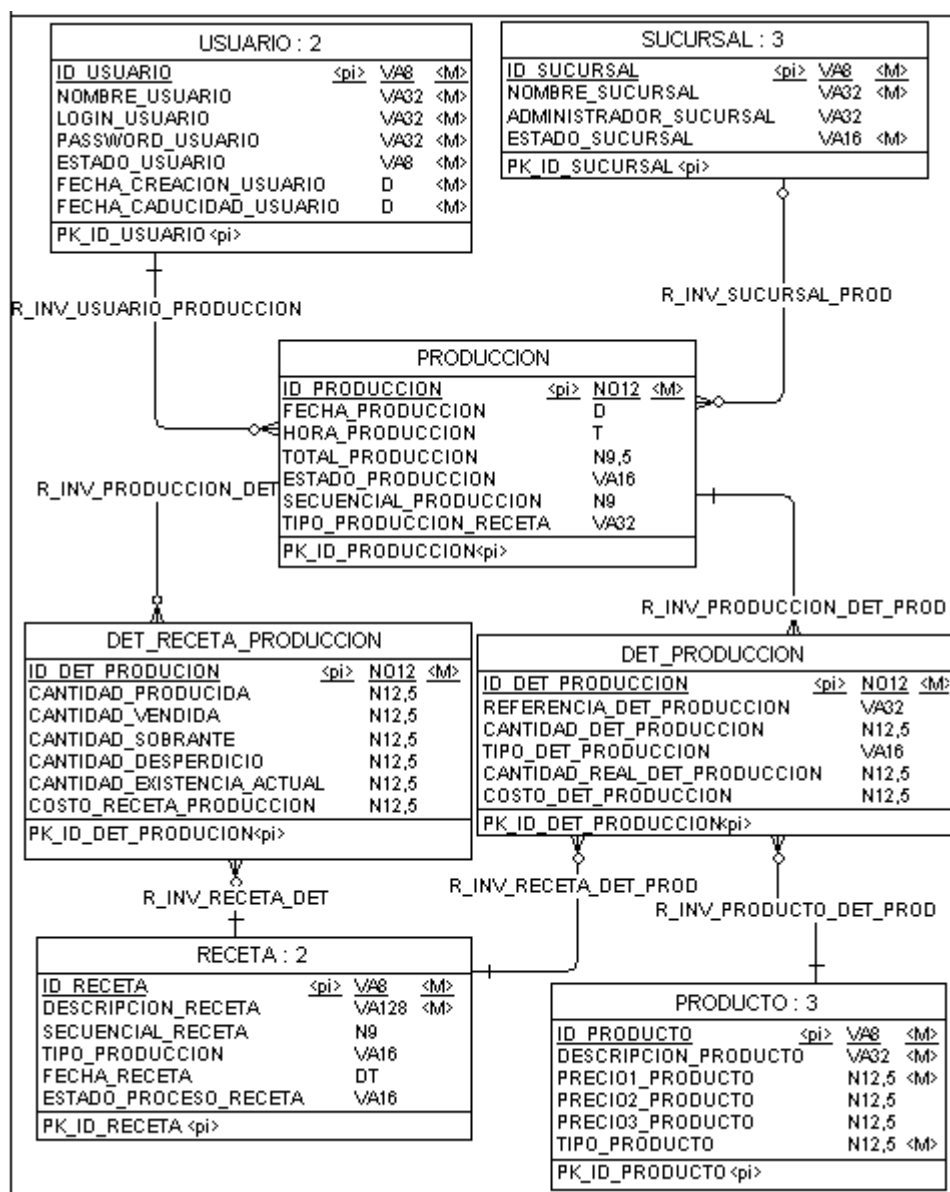


Figura 7.134. Módulo Inventario - Producción³⁵⁹

³⁵⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Inventario – Inventario

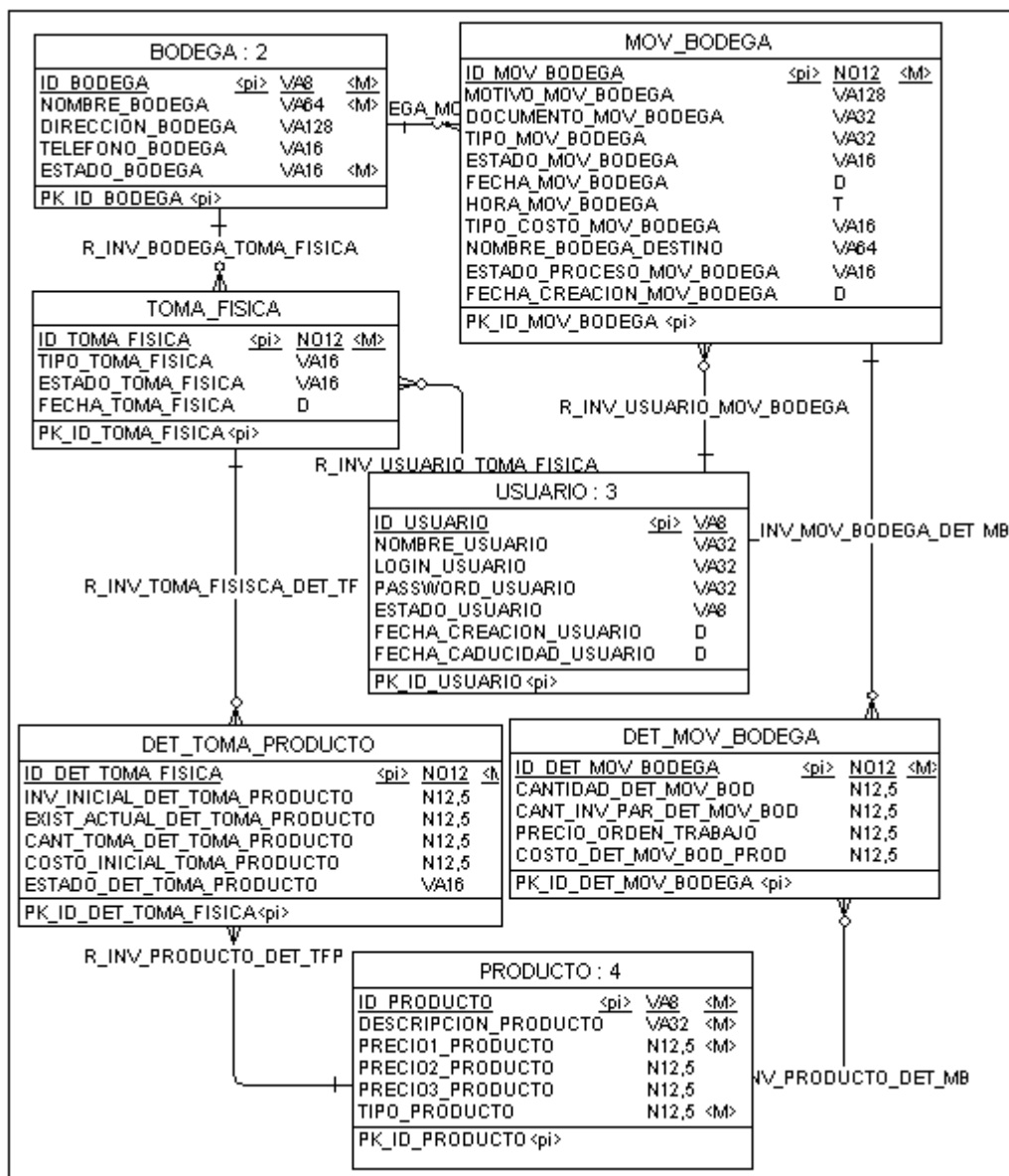


Figura 7.135. Módulo Inventario - Inventario³⁶⁰

³⁶⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Inventario – Kardex

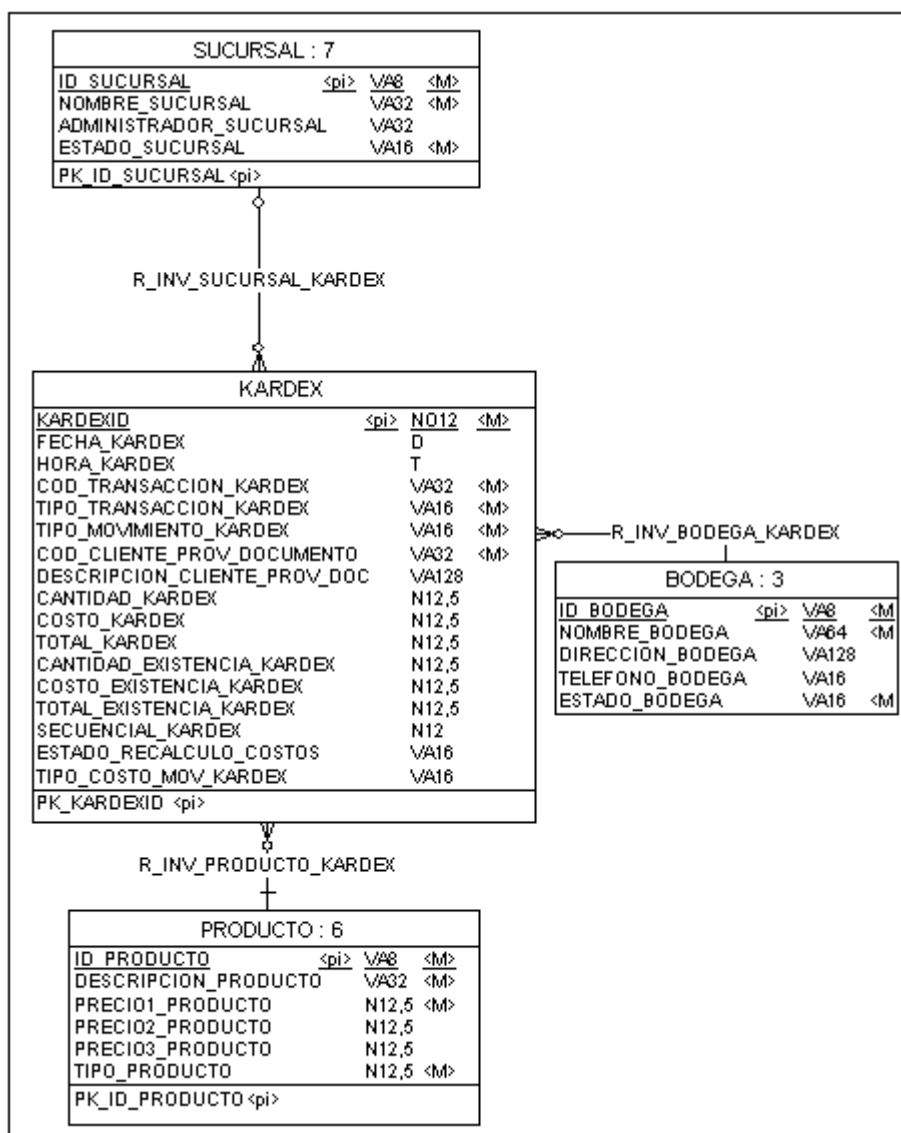


Figura 7.136. Módulo Inventario - Kardex³⁶¹

³⁶¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Facturación – Cliente

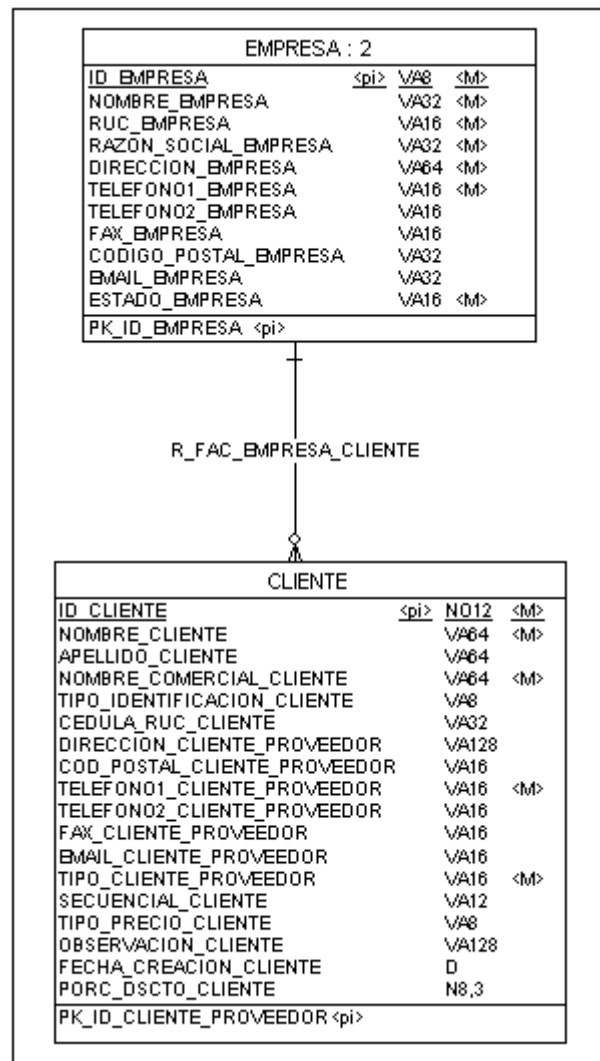


Figura 7.137. Módulo Facturación – Cliente ³⁶²

³⁶² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Facturación - Factura

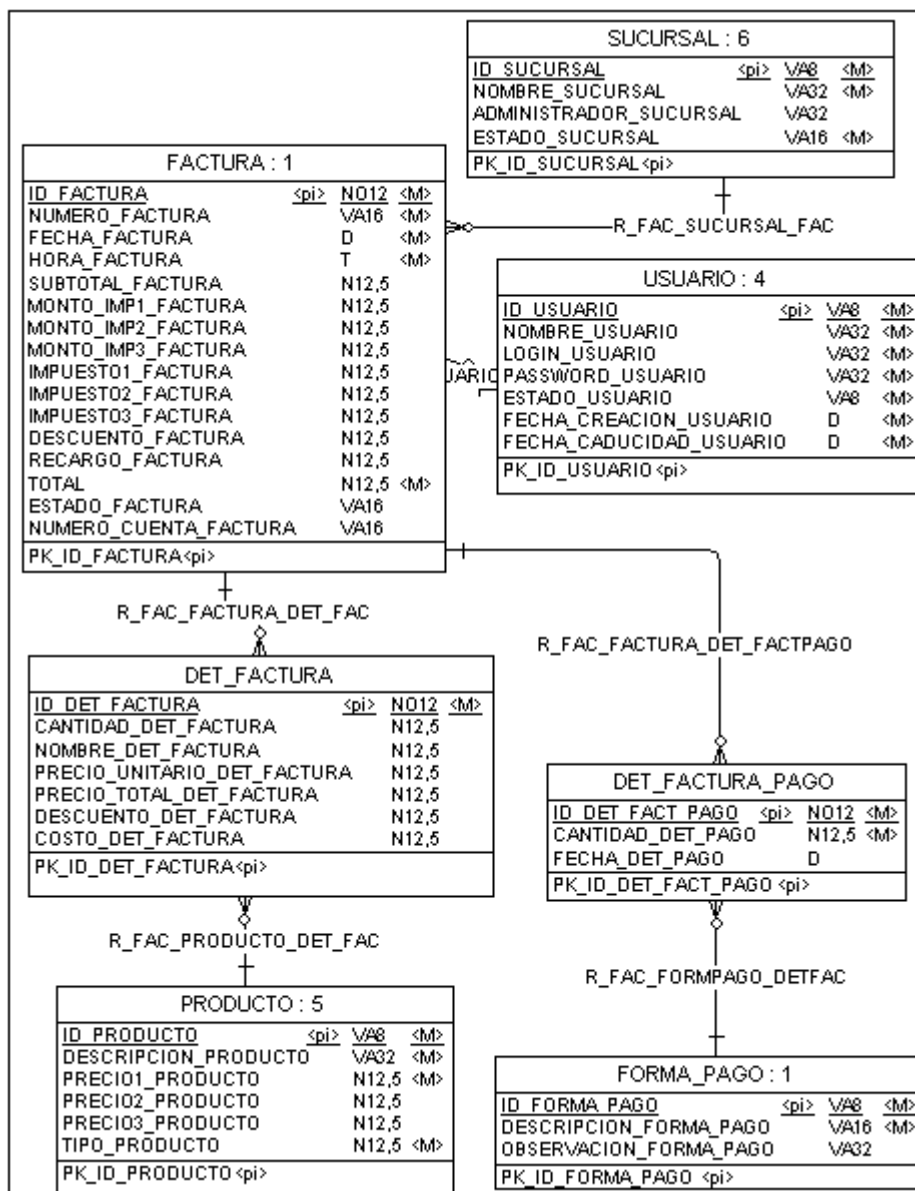


Figura 7.138. Módulo Facturación – Factura ³⁶³

³⁶³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Facturación – Cuadre de Caja

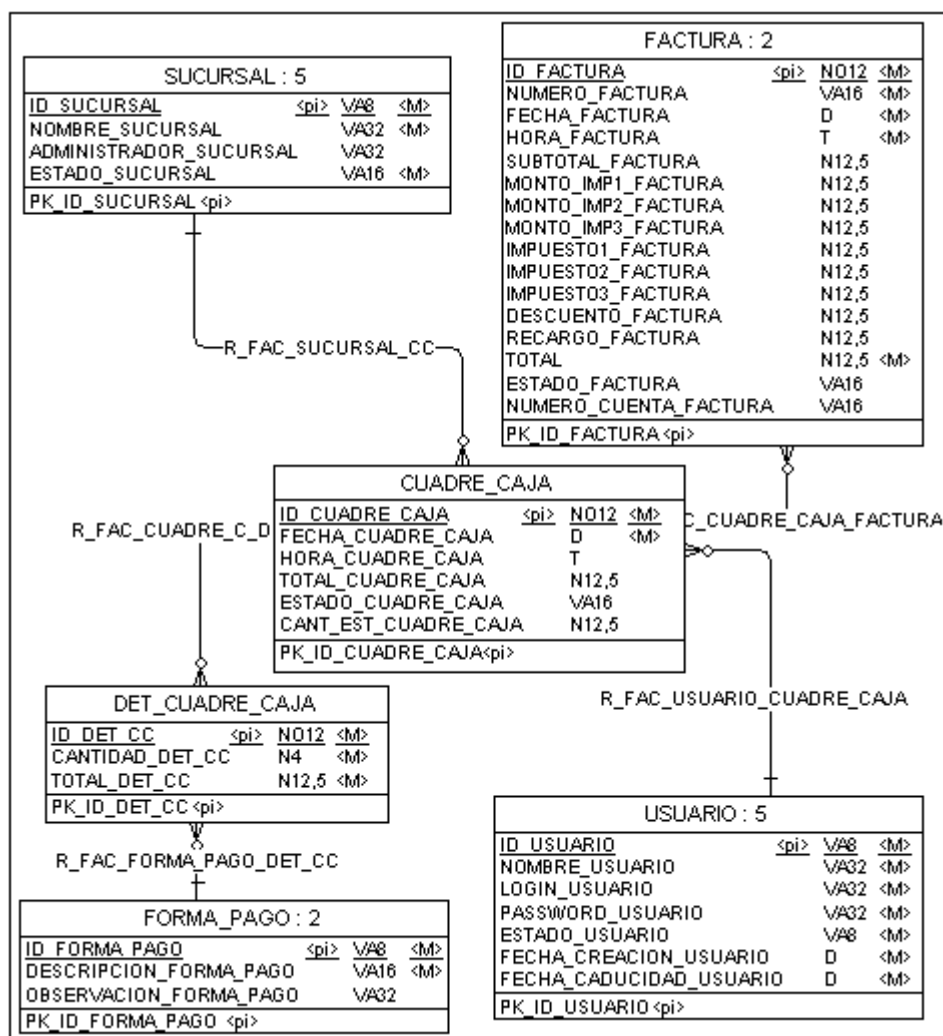


Figura 7.139. Módulo Facturación – Cuadre de Caja ³⁶⁴

³⁶⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Módulo Facturación - Movimiento de Caja

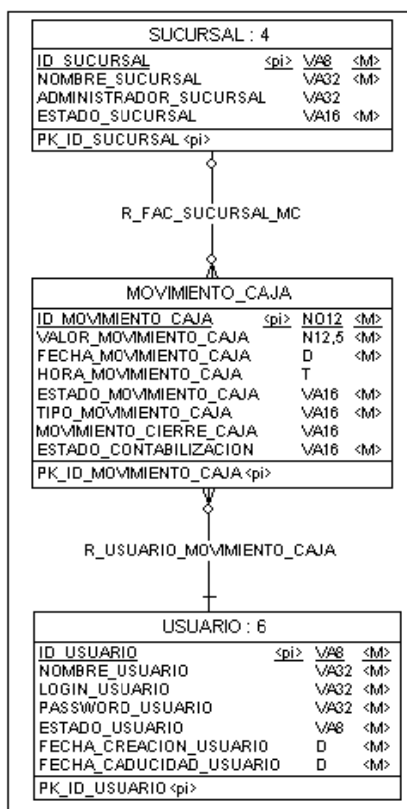


Figura 7.140. Módulo Facturación – Movimiento de Caja ³⁶⁵

2.3.4. DISEÑO DE ARQUITECTURA DEL SISTEMA

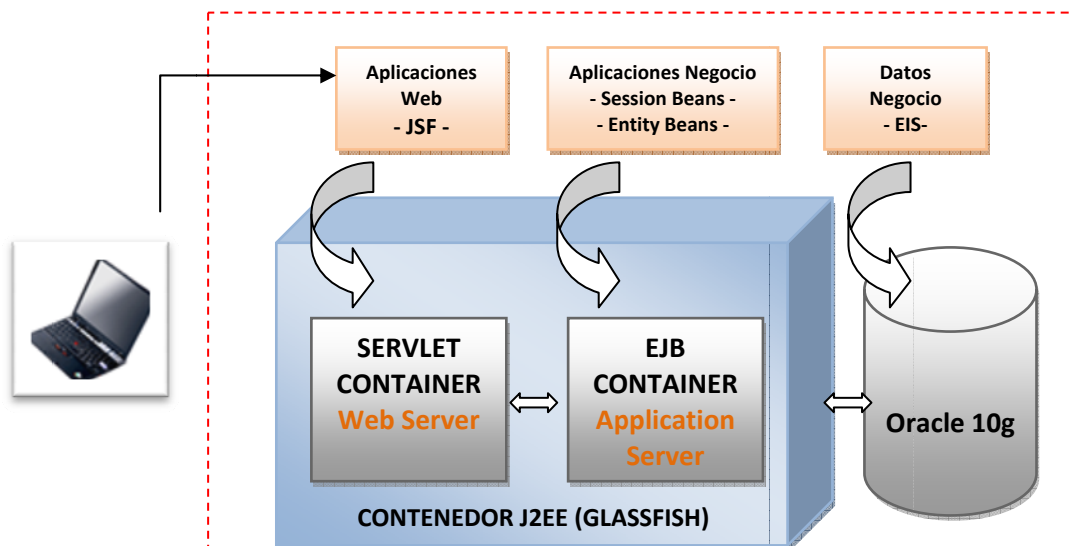


Figura 7.141. Arquitectura del sistema ³⁶⁶

³⁶⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

2.4. CODIFICACION Y PROGRAMACIÓN

Para revisar el código de uno de los procesos del sistema (Facturación), dirigirse a la página 136 a la 152 del presente trabajo.

2.5. DIAGRAMA DE COMPONENTES

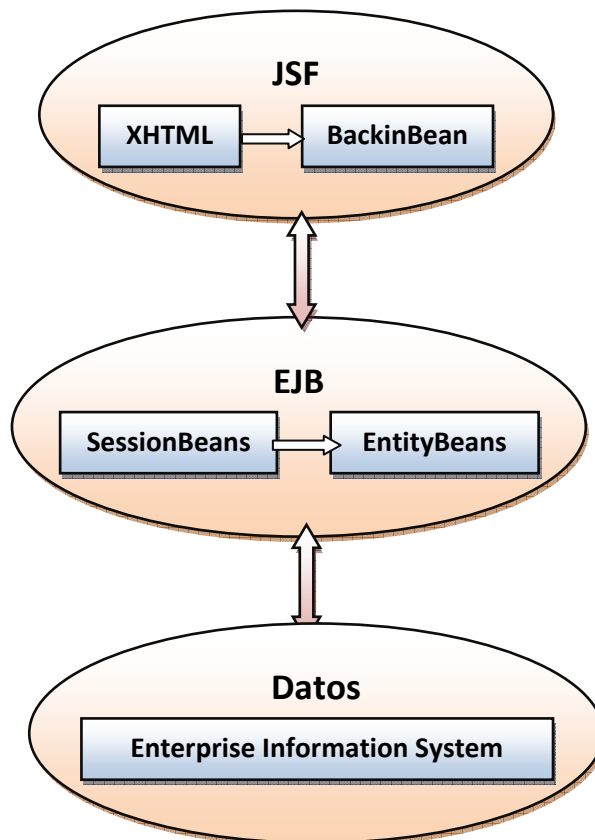


Figura 7.142. Diagrama de componentes³⁶⁷

3. INSTRUCCIONES DE INSTALACIÓN.

Las instrucciones de instalación se encuentran descritas en el Manual de instalación.³⁶⁸

³⁶⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁶⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁶⁸ Anexo 3: Manual de Instalación. Pag: 342

4. REQUERIMIENTOS DE HARDWARE Y SOFTWARE

Hardware:

- Espacio en disco duro:
 - Glassfish: 527 Mb
 - Oracle 10g Express Edition: 1,34Gb
 - Java, JDK-6: 277 Mb
- Memoria RAM:
 - 1024 Mb
- Procesador: 3.2 Ghz mínimo Pentium IV

Software:

- Servidor de aplicaciones Glassfish
- Base de Datos Oracle 10 g Express Edition
- Navegador de internet: Mozilla Firefox 3.x
- Java jdk-6

5. SEGURIDAD Y CONTROL

5.1 SEGURIDAD EN EL ACCESO DE LA INFORMACIÓN

- La aplicación orientada a la web está diseñada en capas.
- La plataforma J2EE proporciona una robusta infraestructura de seguridad, permitiendo a los administradores un mayor control sobre el código que se ejecuta en el sistema.
- Oracle garantiza la privacidad de la información a través de los niveles de seguridad a la bases de datos, además ayuda a la protección de amenazas internas.
- Seguridades que da Linux para el acceso a la máquina.

5.2. CREACIÓN DE USUARIOS

Solamente el usuario administrador podrá crear a otros usuarios, a los cuales se les asignará un perfil de acuerdo a las características que posean y a las restricciones que se les quiera dar.

5.3. SEGURIDAD DE ACCESO

- Para iniciar la aplicación se debe ingresar el login y contraseña, lo que permite que solo los usuarios registrados en el sistema puedan ingresar.
- Se han creado perfiles para cada usuario lo que permite que exista una restricción en el acceso de acuerdo a los módulos y a los procesos.

5.4. SEGURIDAD DE USUARIO

Cada usuario posee un login y contraseña, que le sirve para acceder al sistema, a su vez este usuario tiene asignado un perfil de acuerdo a los módulos que desee manejar.

En la creación de los usuarios se pide el ingreso de un login y contraseña, además se pide la confirmación de la contraseña para tener una protección de datos del usuario.

5.5. POSIBLES USUARIOS

- Administrador
- Cajero

5.6. COPIAS DE SEGURIDAD

El Administrador de Base de Datos (DBA) será el encargado de realizar un respaldo de la base de datos semanalmente.

6. PRUEBAS

6.1. PRUEBA DE CAJA BLANCA

```

public void seleccionarFormaPago() {
    System.out.println("idFormaPago: " + idFormaPago);
    System.out.println("cantidadPagoCliente: " + cantidadPagoCliente);
    Pattern pattern = Pattern.compile("^([0-9]{1,7}(\\.[0-9]{0,2})?)?$");
    Matcher matcher = pattern.matcher((String) cantidadPagoCliente);
    if (!matcher.find())
        setMensajeDatos("::: Error, ingrese un cantidad válida :::");
    else{
        List<DetFacturaPago> lista = facPagoLocal.consultar(Long.parseLong(idFormaPago));
        DetFacturaPago pago = (DetFacturaPago) lista.get(0);
        pago.setCantidadDetPago(new BigDecimal(cantidadPagoCliente));
        facPagoLocal.actualizar(pago);
        /** Factura **/
        List<Factura> listaFactura = facturaLocal.consultar(Long.parseLong(idFactura));
        Factura factura = (Factura) listaFactura.get(0);
        /** Operaciones **/
        BigDecimal _total = factura.getTotal();
        BigDecimal _sumatoria = new BigDecimal("0");
        List<DetFacturaPago> _lista = facPagoLocal.consultarPorFactura(Long.parseLong(idFactura));
        for (int i = 0; i < _lista.size(); i++) {
            DetFacturaPago _pago = (DetFacturaPago) _lista.get(i);
            _sumatoria = _sumatoria.add(_pago.getCantidadDetPago());
        }
        BigDecimal _saldo = _total.subtract(_sumatoria);
        BigDecimal _cambio = new BigDecimal("0.00");
        if (_sumatoria.compareTo(_total) == 1) {
            _saldo = new BigDecimal("0.00");
            _cambio = _sumatoria.subtract(_total);
        }
        /** Setemos las variables **/
        setTotalPagar(_total.setScale(2, RoundingMode.HALF_UP) + "");
        setSaldo(_saldo.setScale(2, RoundingMode.HALF_UP) + "");
        setTotalPagoCliente(_sumatoria.setScale(2, RoundingMode.HALF_UP) + "");
        setCambio(_cambio.setScale(2, RoundingMode.HALF_UP) + "");
        setMensajeDatos("");
    }
}

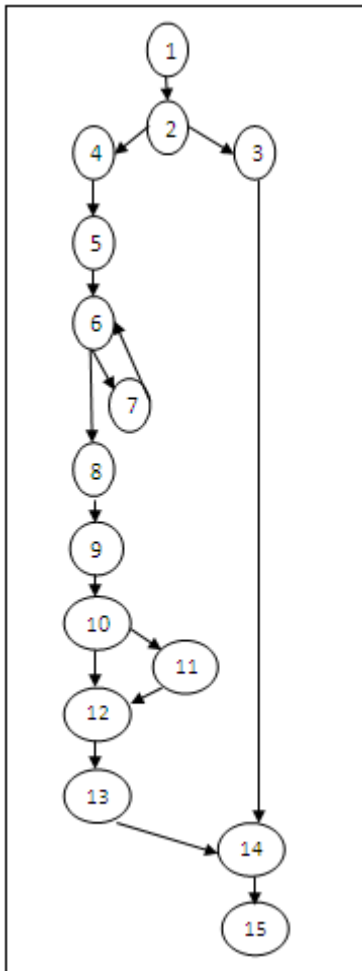
```

Diagrama de anotaciones de prueba de caja blanca:

- 1: Línea de impresión de idFormaPago.
- 2: Línea de impresión de cantidadPagoCliente.
- 3: Línea de configuración de patrón y matcher.
- 4: Línea de verificación de la existencia del pago.
- 5: Bloque de procesamiento de la factura y operaciones.
- 6: Línea de iteración sobre la lista de pagos.
- 7: Línea de actualización del pago.
- 8: Línea de cálculo del saldo.
- 9: Línea de cálculo del cambio.
- 10: Línea de verificación de la comparación de totales.
- 11: Línea de cálculo del saldo y cambio.
- 12: Línea de configuración de variables.
- 13: Línea de configuración de variables de salida.
- 14: Línea de cierre del método.
- 15: Línea de cierre de la clase.

Figura 7.143 .Código del método seleccionarFormaPago ³⁶⁹

³⁶⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010



$$V(G) = \text{N}^{\circ} \text{ Regiones} = 4$$

$$V(G) = \text{Aristas} - \text{Nodos} + 2 = 17 - 15 + 2 = 4$$

$$V(G) = \text{Nodos Predicados} + 1 = 3 + 1 = 4$$

Caminos independientes

- 1.- 1-2-3-14-15
2. 1-2-4-5-6-7-6-8-9-10-11-12-13-14-15
3. 1-2-4-5-6-7-6-8-9-10-12-13-14-15
4. 1-2-4-5-6-8-9-10-11-12-13-14-15

Figura 7.144. Diagrama Caja Blanca ³⁷⁰

Caso Prueba	Resultado esperado	Resultado obtenido	Conclusiones
cantidadPagoCliente = "abc"	Mensaje de error	"Error ingrese una cantidad válida"	La primera condición if, se encuentra funcionando
<ul style="list-style-type: none"> • cantidadPagoCliente=29 • _lista=3 	<ul style="list-style-type: none"> • Total a pagar = total de factura. • Saldo=0 	<ul style="list-style-type: none"> • Total a pagar=29 • Saldo=0 	Si la sumatoria de ingresos es mayor al total a

³⁷⁰ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

<ul style="list-style-type: none"> • $_sumatoria=30$ (Sumatoria de ingresos) • $_total=29$ (Total a pagar) 	<ul style="list-style-type: none"> • Total= sumatoria de ingresos • Cambio= sumatoria de ingresos menos el total de la factura. 	<ul style="list-style-type: none"> • Total=30 • Cambio=1 	<p>pagar de la factura queda un saldo de cero.</p>
<ul style="list-style-type: none"> • $cantidadPagoCliente=10$ • $_lista=3$ • $_sumatoria=10$ (Sumatoria de ingresos) • $_total=29$ (Total a pagar) 	<ul style="list-style-type: none"> • Total a pagar = total de factura. • Saldo=Total a pagar menos la sumatoria. • Total= sumatoria de ingresos • Cambio= cero 	<ul style="list-style-type: none"> • Total a pagar=29 • Saldo=19 • Total=10 • Cambio=0 	<p>Si la sumatoria de ingresos es menor al total a pagar de la factura queda el cambio de cero.</p>
<ul style="list-style-type: none"> • $cantidadPagoCliente=0$ • $_lista=0$ • $_sumatoria=0$ (Sumatoria de ingresos) • $_total= 0$ (Total a pagar) 	<p>No se despliegan las formas de pago y por ende no se puede facturar porque no se puede realizar la sumatoria de ingresos.</p>	<p>No se despliegan las formas de pago y por ende no se puede facturar porque no se puede realizar la sumatoria de ingresos.</p>	<p>Antes de facturar se debe realizar el ingreso de las formas de pago.</p>

Tabla 7.117. Casos de prueba – Caja Blanca ³⁷¹

³⁷¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

6.2. PRUEBA DE CAJA NEGRA

Para la prueba de caja negra se va a utilizar la interfaz de movimientos de caja

Figura 7.145. Interfaz Movimientos de caja³⁷²

Condición de Entrada	Tipo	Clase Equivalente Válida	Clase Equivalente no válida
Nº Documento	Valor específico	1. Solo números	2. Cadena de caracteres
Tipo	Miembro de un conjunto	3. Ingreso caja 4. Egreso caja	5. Cadenas distintas a las válidas
Valor	Valor específico	6. Solo Números de hasta 2 decimales.	7. Número con más de 2 decimales. 8. Cadena de caracteres 9. Caracteres especiales 10. Números negativos

Tabla 7.118. Condiciones de Entrada – Caja Negra³⁷³

³⁷² ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁷³ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Equivalencia	Caso Prueba	Respuesta Esperada	Resultado Obtenido	Conclusiones
1	520	Si realiza el ingreso.	El campo está bloqueado ya que es un número autogenerated.	El número de documento del movimiento es generado.
2	"abc"	No se realiza el ingreso.	El campo está bloqueado ya que es un número autogenerated.	No se permite el ingreso de caracteres en este campo.
3	Ingreso Caja	Si realiza el ingreso.	Si se selecciona la opción de ingreso caja.	Si se permite la selección del dato: ingreso caja
4	Egreso Caja	Si realiza el ingreso.	En el combo se queda seleccionada la opción de egreso caja.	Si se permite la selección del dato: egreso caja.
5	"Caja"	No se realiza el ingreso.	En las opciones no se puede elegir otra que no esté en el combo.	No se pueden seleccionar otro tipo de opciones que no estén en el combo.
6	7.2	Si se graban los datos	Al momento de grabar la transacción es registrada.	Si se permite el ingreso de números de hasta 2 decimales.
7	4.562	No se permita grabar la	Se despliega el mensaje de error:	No se permite en ingreso en

		información.	“Ingrese un valor válido. Ej: 1.50”	valor que tenga 3 decimales
8	“valor”	No se permita grabar la información.	Se despliega el mensaje de error: “Ingrese un valor válido. Ej: 1.50”	En el campo valor no se permite el ingreso de caracteres
9	“*.-”	No se permita grabar la información.	Se despliega el mensaje de error: “Ingrese un valor válido. Ej: 1.50”	En el campo valor no se permite el ingreso de caracteres especiales.
10	-5.2	No se permita grabar la información.	Se despliega el mensaje de error: “Ingrese un valor válido. Ej: 1.50”	En el campo valor no se permite el ingreso de números negativos.

Tabla 7.119. Casos de Prueba – Caja Negra ³⁷⁴

6.3. PRUEBAS DEL SISTEMA

1.

Caso prueba:	Ingresar login y password no existentes en la base de datos.
Resultado esperado:	Se despliega un mensaje de error.
Resultado obtenido:	Se muestra el mensaje: “Login y/o Password incorrectos.”

³⁷⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

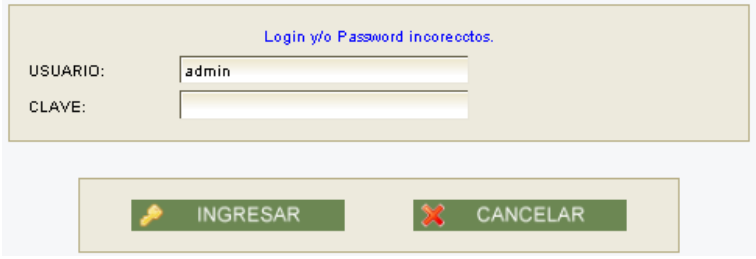
	
Conclusiones:	Si un usuario no se encuentra registrado en la base de datos no podrá ingresar al sistema.

Tabla 7.120. Caso de Prueba - 1 ³⁷⁵

2.

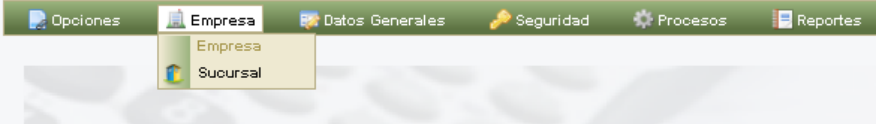
Caso prueba:	Ingresar al sistema con un usuario que no tenga acceso a la opción Empresa.
Resultado esperado:	El usuario no pueda acceder
Resultado obtenido:	<p>En el menú se encuentra desactivada la opción Empresa.</p> 
Conclusiones:	La restricción de los usuarios de acuerdo al perfil asignado funciona correctamente en el sistema.

Tabla 7.121. Caso de Prueba - 2 ³⁷⁶

3.

Caso prueba:	Ingresar espacio en blanco en el campo de login.
Resultado esperado:	Se despliega un mensaje de error
Resultado obtenido:	Se muestra el mensaje: "El campo no debe estar vacío."

³⁷⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁷⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

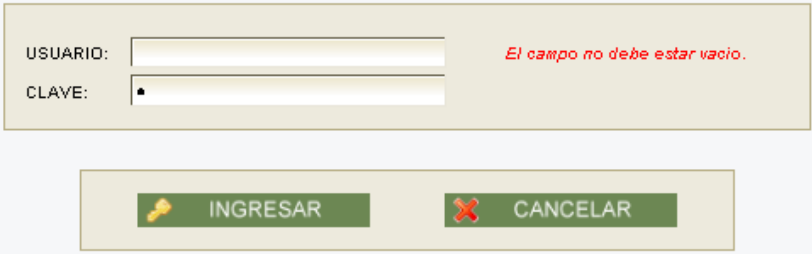
	 <p>USUARIO: <input type="text"/> El campo no debe estar vacío.</p> <p>CLAVE: <input type="password"/></p> <p style="text-align: center;"> <input type="button" value="INGRESAR"/> <input type="button" value="CANCELAR"/> </p>
Conclusiones:	Si no se ingresan datos en los campos solicitados, se desplegará un mensaje de error.

Tabla 7.122. Caso de Prueba - 3³⁷⁷

4.

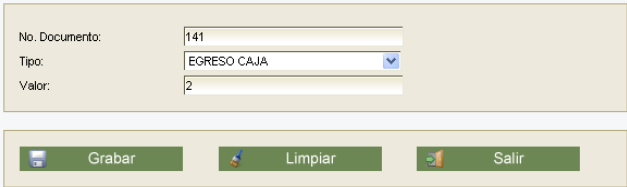


Caso prueba:	Verificar que los datos que se ingresaron se encuentren registrados en la base de datos. (Tabla Movimiento_Caja).																																																								
Resultado esperado:	Que los datos se encuentren en la tabla.																																																								
Resultado obtenido:	<p>Se ingresó los datos en movimiento de caja:</p>  <p>No. Documento: <input type="text" value="141"/> Tipo: <input type="text" value="EGRESO CAJA"/> Valor: <input type="text" value="2"/></p> <p style="text-align: center;"> <input type="button" value="Grabar"/> <input type="button" value="Limpiar"/> <input type="button" value="Salir"/> </p> <p>Los datos se muestran en la base:</p>  <table border="1"> <thead> <tr> <th>EDITAR</th> <th>ID_MOVIMIENTO_CAJA</th> <th>ID_SUCURSAL</th> <th>ID_USUARIO</th> <th>VALOR_MOVIMIENTO_CAJA</th> <th>FECHA_MOVIMIENTO_CAJA</th> <th>HORA_MOVIMIENTO_CAJA</th> <th>ESTADO_MOVIMIEN</th> </tr> </thead> <tbody> <tr> <td></td> <td>5</td> <td>01</td> <td>01</td> <td>15</td> <td>21/07/10</td> <td>21:07:10</td> <td>REGISTRADO</td> </tr> <tr> <td></td> <td>4</td> <td>01</td> <td>01</td> <td>50</td> <td>21/07/10</td> <td>21:07:10</td> <td>REGISTRADO</td> </tr> <tr> <td></td> <td>6</td> <td>01</td> <td>01</td> <td>24.35</td> <td>21/07/10</td> <td>21:07:10</td> <td>REGISTRADO</td> </tr> <tr> <td></td> <td>142</td> <td>01</td> <td>01</td> <td>0</td> <td>04/09/10</td> <td>04:09:10</td> <td>PENDIENTE</td> </tr> <tr> <td></td> <td>102</td> <td>01</td> <td>01</td> <td>12</td> <td>26/09/10</td> <td>26:09:10</td> <td>REGISTRADO</td> </tr> <tr> <td></td> <td>141</td> <td>01</td> <td>01</td> <td>2</td> <td>04/09/10</td> <td>04:09:10</td> <td>REGISTRADO</td> </tr> </tbody> </table>	EDITAR	ID_MOVIMIENTO_CAJA	ID_SUCURSAL	ID_USUARIO	VALOR_MOVIMIENTO_CAJA	FECHA_MOVIMIENTO_CAJA	HORA_MOVIMIENTO_CAJA	ESTADO_MOVIMIEN		5	01	01	15	21/07/10	21:07:10	REGISTRADO		4	01	01	50	21/07/10	21:07:10	REGISTRADO		6	01	01	24.35	21/07/10	21:07:10	REGISTRADO		142	01	01	0	04/09/10	04:09:10	PENDIENTE		102	01	01	12	26/09/10	26:09:10	REGISTRADO		141	01	01	2	04/09/10	04:09:10	REGISTRADO
EDITAR	ID_MOVIMIENTO_CAJA	ID_SUCURSAL	ID_USUARIO	VALOR_MOVIMIENTO_CAJA	FECHA_MOVIMIENTO_CAJA	HORA_MOVIMIENTO_CAJA	ESTADO_MOVIMIEN																																																		
	5	01	01	15	21/07/10	21:07:10	REGISTRADO																																																		
	4	01	01	50	21/07/10	21:07:10	REGISTRADO																																																		
	6	01	01	24.35	21/07/10	21:07:10	REGISTRADO																																																		
	142	01	01	0	04/09/10	04:09:10	PENDIENTE																																																		
	102	01	01	12	26/09/10	26:09:10	REGISTRADO																																																		
	141	01	01	2	04/09/10	04:09:10	REGISTRADO																																																		
Conclusiones:	En la base de datos si se registran los ingresos realizados.																																																								

Tabla 7.123. Caso de Prueba - 4³⁷⁸

³⁷⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁷⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

5.

Caso prueba:	Verificar que los reportes coincidan con los registros de la base de datos																												
Resultado esperado:	Que en los reportes se visualice los datos almacenados.																												
Resultado obtenido:	<p>Se tiene los datos de los clientes en la base:</p>  <p>El reporte de clientes muestra la siguiente información:</p> <p style="text-align: center;">REPORTE DE CLIENTES</p> <table border="1" data-bbox="486 896 1372 985"> <thead> <tr> <th>CODIGO</th> <th>CLIENTE</th> <th>R.U.C.</th> <th>DIRECCION</th> <th>TELEFONO</th> <th>E-MAIL</th> <th>TIPO PRECIO</th> </tr> </thead> <tbody> <tr> <td>22</td> <td>MEJIA VANESA</td> <td>123445</td> <td>QUITO</td> <td>2333</td> <td>info@empresa.com</td> <td>Precio C</td> </tr> <tr> <td>21</td> <td>SANDBALIN GUAMAN JULIO CESAR</td> <td>1715460372</td> <td>QUITO</td> <td>2</td> <td>mail@empresa.com</td> <td>Precio A</td> </tr> <tr> <td>2</td> <td>VILLACID CRISTINA</td> <td>1715460372</td> <td>2</td> <td>2</td> <td></td> <td>Precio B</td> </tr> </tbody> </table>	CODIGO	CLIENTE	R.U.C.	DIRECCION	TELEFONO	E-MAIL	TIPO PRECIO	22	MEJIA VANESA	123445	QUITO	2333	info@empresa.com	Precio C	21	SANDBALIN GUAMAN JULIO CESAR	1715460372	QUITO	2	mail@empresa.com	Precio A	2	VILLACID CRISTINA	1715460372	2	2		Precio B
CODIGO	CLIENTE	R.U.C.	DIRECCION	TELEFONO	E-MAIL	TIPO PRECIO																							
22	MEJIA VANESA	123445	QUITO	2333	info@empresa.com	Precio C																							
21	SANDBALIN GUAMAN JULIO CESAR	1715460372	QUITO	2	mail@empresa.com	Precio A																							
2	VILLACID CRISTINA	1715460372	2	2		Precio B																							
Conclusiones:	Los datos que se despliegan en los reportes son los que están almacenados en la base de datos.																												

*Tabla 7.124. Caso de Prueba - 5*³⁷⁹

6.

Caso prueba:	Se pierde comunicación con la base de datos (Detener servidor de base de datos)
Resultado esperado:	El sistema no realice transacciones.
Resultado obtenido:	El sistema sigue funcionando, quedando todas las transacciones como pendientes hasta que se inicie la base de datos.
Conclusiones:	El sistema se encuentra preparado en caso de que la base de datos se detenga.

*Tabla 7.125. Caso de Prueba - 6*³⁸⁰

³⁷⁹ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

7.

Caso prueba:	Corte inesperado de energía eléctrica en el servidor.
Resultado esperado:	Cuando se restablezca la energía eléctrica funcione correctamente.
Resultado obtenido:	Las transacciones que no se realizaron correctamente se pierden.
Conclusiones:	Para estos casos es necesario colocar un ups en el servidor y en las estaciones.

Tabla 7.126. Caso de Prueba - 7³⁸¹

8.

Caso prueba:	20 usuarios ejecutan el sistema.
Resultado esperado:	Todos los usuarios trabajen normalmente
Resultado obtenido:	El programa utilizado para realizar las pruebas de carga fue Webserver Stress, el resultado fue que todos los usuarios realizan las transacciones normalmente.
	<p>Open Requests & Transferred Data</p> <p>Active Users</p> <p>Open Requests</p> <p>Sent Requests per second</p> <p>Received Requests per second</p> <p>Network Traffic [bit/s]</p> <p>Time since start of test [s]</p> <p>Test Type: RUM (run test for 5 minutes)</p> <p>User Simulation: ramp test with up to 10 simultaneous users - 5 seconds between clicks</p>
Conclusiones	El sistema resiste a las transacciones simultáneas.

Tabla 7.127. Caso de Prueba - 8³⁸²

³⁸⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁸¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁸² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

9.

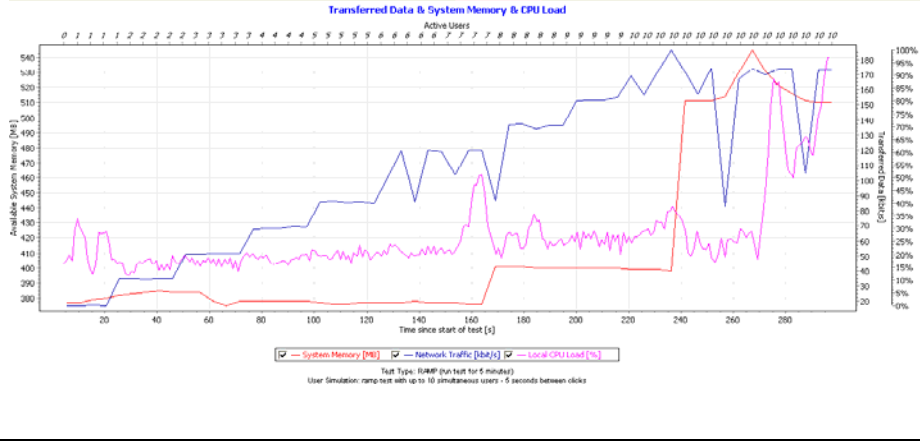
Caso prueba:	Comprobar consumo de memoria al tener transacciones simultáneas (20 usuarios).
Resultado esperado:	El consumo de memoria es mínimo y no afecta al normal funcionamiento del PC.
Resultado obtenido:	<p>El caso de prueba fue ejecutado en el programa WebServer Stress el cual muestra una gráfica.</p> <ul style="list-style-type: none"> • Línea negra es el tráfico de la red • Línea rosada es la carga del procesador • Línea roja es el consumo de memoria. 
Conclusiones :	El consumo de memoria es el adecuado de acuerdo a la transferencia de datos realizada.

Tabla 7.128. Caso de Prueba - 9³⁸³

³⁸³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

ANEXO 3. MANUAL DE INSTALACIÓN

CONTENIDO

CONTENIDO.....	341
1. REQUISITOS:	342
2. INSTALACIÓN JAVA JDK-6.....	342
3. INSTALACIÓN ORACLE 10G XE	342
4. CONFIGURACIÓN DE LA BASE DE DATOS.....	344
5. INSTALACIÓN SERVIDOR DE APLICACIONES GLASSFISHV3.....	346
6. CONFIGURACIÓN DEL SERVIDOR DE APLICACIONES GLASSFISHV3.	348

1. REQUISITOS:

- Sistema operativo Linux Ubuntu 10.04
- Java jdk-6
- Base de Datos Oracle 10g XE
- Servidor de Aplicaciones GlassFish V3

2. INSTALACIÓN JAVA JDK-6

1. Descargar el archivo `jdk-6u21-linux-i586.bin` desde la web de Oracle
2. Cambiamos los permisos de ejecución del archivo:

```
chmod 755 jdk-6u21-linux-i586.bin
```

3. Ejecutamos el archivo para instalar Java:

```
sudo ./jdk-6u21-linux-i586.bin
```

4. Creamos un enlace simbólico para el usuario donde estamos actualmente trabajando:

```
sudo mkdir /usr/java; sudo mv jdk1.6.0_21 /usr/java/jdk1.6.0_21
```

5. Ingresamos al directorio `/home` y editamos el fichero `.bashrc` para configurar las variables de entorno:

```
export JAVA_HOME=/usr/java/java-current  
export PATH=$PATH:$JAVA_HOME/bin/
```

3. INSTALACIÓN ORACLE 10G XE

1. Descargar el fichero `oracle-xe-universal_10.2.0.1-1.0_i386.deb` desde la web de oracle.
2. Damos doble clic sobre el fichero y a continuación damos clic en instalar paquete.
3. Una vez instalada la base de datos tenemos que configurarla con la siguiente sentencia:

```
sudo /etc/init.d/oracle-xe configure
```

4. Con eso arranca un configurador en modo texto para configurar el puerto de la aplicación web que nos permite administrar gráficamente la base de datos, el puerto de la base de datos, el password para el usuario SYS y SYSTEM de Oracle, y la opción para arrancar la base de datos cada vez que iniciemos la maquina.

```
Oracle Database 10g Express Edition Configuration
```

```
-----
```

```
This will configure on-boot properties of Oracle Database 10g Express
Edition. The following questions will determine whether the database should
be starting upon system boot, the ports it will use, and the passwords that
will be used for database accounts. Press <Enter> to accept the defaults.
Ctrl-C will abort.
```

```
Specify the HTTP port that will be used for Oracle Application Express [8080]:8080
```

```
Specify a port that will be used for the database listener [1521]:1521
```

```
Specify a password to be used for database accounts. Note that the same
password will be used for SYS and SYSTEM. Oracle recommends the use of
different passwords for each database account. This can be done after
initial configuration:
```

```
Confirm the password:
```

```
Do you want Oracle Database 10g Express Edition to be started on boot (y/n) [y]:y
```

```
Starting Oracle Net Listener...Done
```

```
Configuring Database...Done
```

```
Starting Oracle Database 10g Express Edition Instance...Done
```

```
Installation Completed Successfully.
```

```
To access the Database Home Page go to "http://127.0.0.1:8080/apex"
```

5. Ingresamos al directorio /home y editamos el fichero .bashrc para configurar las variables de entorno:

```
# variables para ORACLE
ORACLE_HOME=/usr/lib/oracle/xe/app/oracle/product/10.2.0/server
PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_HOME
export ORACLE_SID=XE
export PATH
```

4. CONFIGURACIÓN DE LA BASE DE DATOS.

1. Ingresamos a la página para administrar Oracle con la url <http://127.0.0.1:8080/apex/> luego digitamos el usuario SYSTEM y su contraseña en la pantalla de la Figura 8.1.



Figura 8.1. Interfaz: Conexión a Base de Datos ³⁸⁴

2. Ingresamos al módulo de Administración de la Base de Datos en la pantalla de la Figura 8.2.:



Figura 8.2. Interfaz: Menú Principal de Oracle ³⁸⁵

3. Ingresamos a la opción de Usuarios de Bases de Datos en la pantalla de la Figura 8.3.:



Figura 8.3. Interfaz: Menú Administración de Oracle ³⁸⁶

³⁸⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁸⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

4. Creamos un nuevo usuario de la base de datos con el nombre ADMIN y el password ADMIN que tenga los privilegios necesarios para ser DBA, para esto damos clic en la opción Crear en la pantalla de la Figura 8.4., e ingresamos la información del usuario en la pantalla de la Figura 8.5.



Figura 8.4. Interfaz: Gestionar Usuario de Base de Datos ³⁸⁷

Figura 8.5. Interfaz: Crear Usuario de Base de Datos ³⁸⁸

³⁸⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁸⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁸⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

5. INSTALACIÓN SERVIDOR DE APLICACIONES GLASSFISHV3

1. Descargar el fichero glassfish-3.0.1-unix.sh desde la web de oracle
2. Damos permiso de ejecución al fichero y lo instalamos con los comandos:

```
$sudo chmod +x glassfish-v3-prelude-unix.sh
$./glassfish-v3-prelude-unix.sh
```

3. A continuación aparece la pantalla de Bienvenida para la instalación de Glassfishv3, damos clic en siguiente (Figura 8.6.)

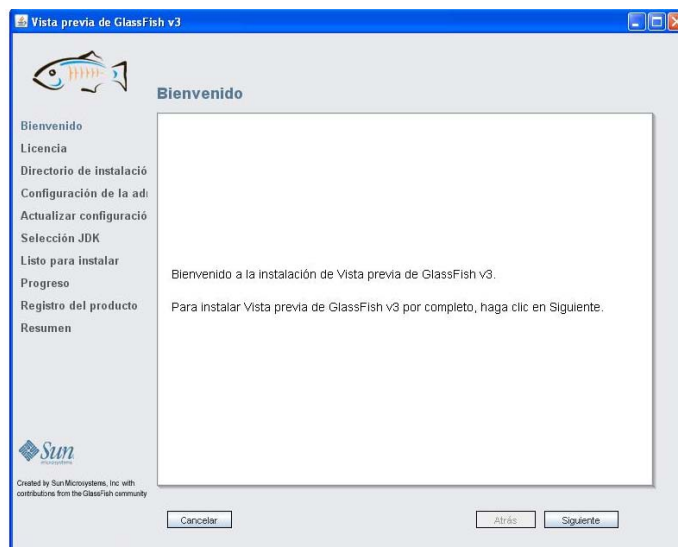


Figura 8.6. Interfaz: Instalación GlassFish v3³⁸⁹

4. Configuramos los puertos de comunicación, el nombre y contraseña del servidor de aplicaciones como se muestra en la pantalla de la Figura 8.7.

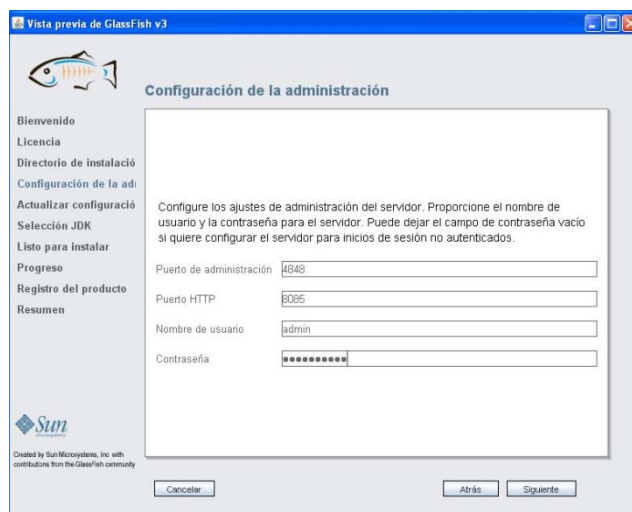


Figura 8.7. Interfaz: Configuración de la administración GlassFish v3³⁹⁰

³⁸⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

5. Una vez configuradas las opciones, instalamos el servidor glassfishv3, damos clic en instalar en la pantalla de la Figura 8.8.

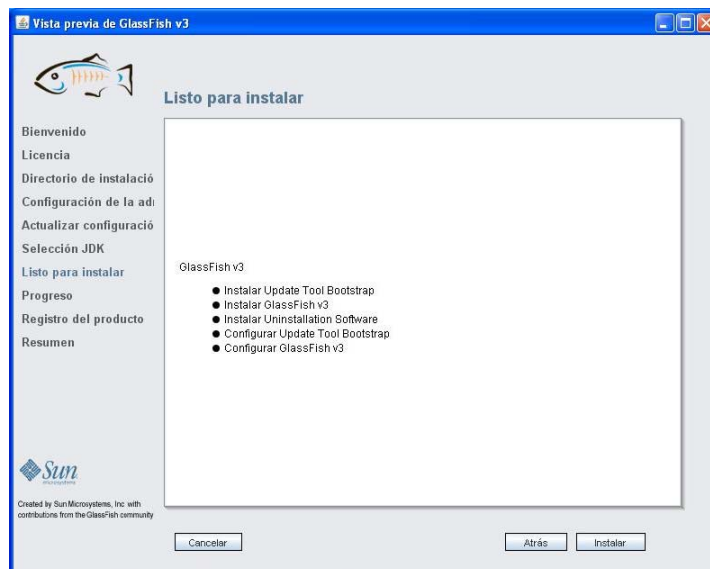


Figura 8.8. Interfaz: Listo para instalar GlassFish v3 ³⁹¹

6. Finalizada la instalación damos clic en salir en la pantalla de la Figura 8.9.

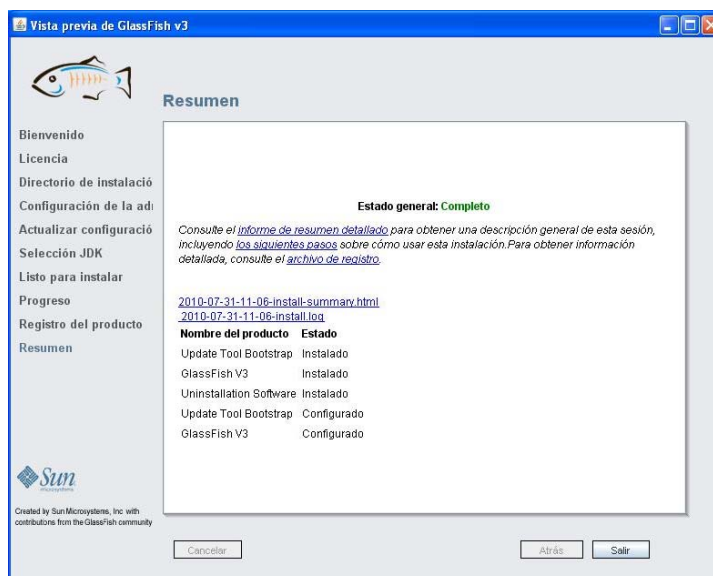


Figura 8.9. Interfaz: Resumen de instalación GlassFish v3 ³⁹²

³⁹⁰ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁹¹ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁹² ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

6. CONFIGURACIÓN DEL SERVIDOR DE APLICACIONES GLASSFISHV3

1. Copiamos el jdbc de Oracle ojdbc14.jar en la ruta:

/glassfishv3/glassfish/domains/domain1/lib

2. Ingresamos a la Consola de Administración de GlassFish con la url <http://localhost:4848/login.jsf>, y se despliega la pantalla de la Figura 8.10.

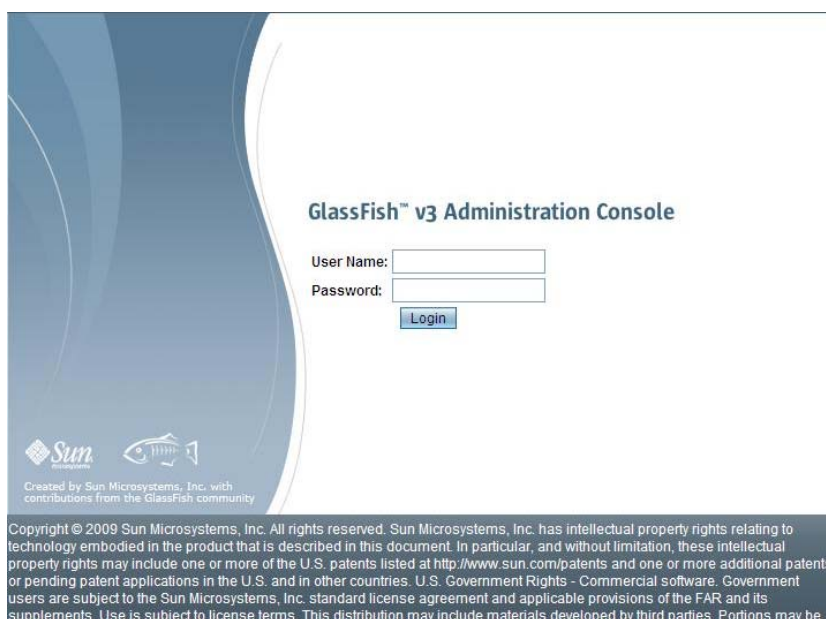


Figura 8.10. Interfaz: Consola de Administración de Glassfish V3³⁹³

3. Ingresamos al módulo Fuente de Datos (Resources) en la opción Fuente de Conexiones (Connection Pool), se despliega la pantalla de la Figura 8.11., e ingresamos los datos:

- i. Name: Pool_Oracle_10G
- ii. Resource Type: javax.sql.DataSource
- iii. Database Vendedor: Oracle

³⁹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010



Figura 8.11. Interfaz: Nuevo JDBC en Pool de Conexiones de Glassfish V3 ³⁹⁴

4. Modificamos las propiedades de la Conexión (Figura 8.12.) con lo siguiente:

- i. User = admin
- ii. DatabaseName = XE
- iii. Password = adminadmin
- iv. PortNumber = 1521
- v. ServerName = localhost
- vi. URL = jdbc:oracle:thin:@localhost:1521:XE

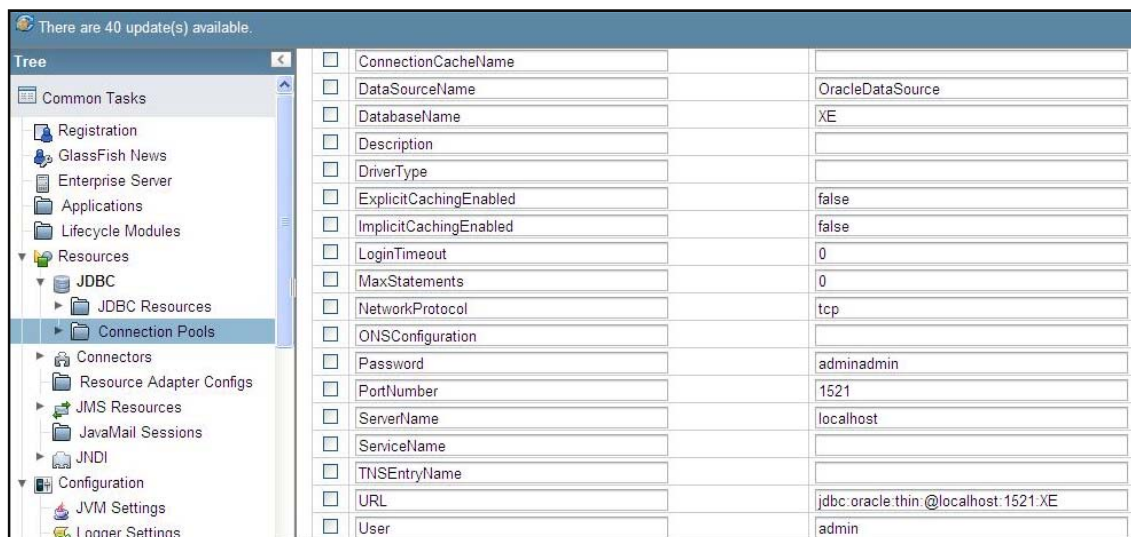


Figura 8.12. Interfaz: Pool de Conexiones de Glassfish V3 ³⁹⁵

³⁹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

5. Para verificar la comunicación con la Base de Datos hacemos un ping de pruebas dando clic en la opción Ping en la pantalla de la Figura 8.13.

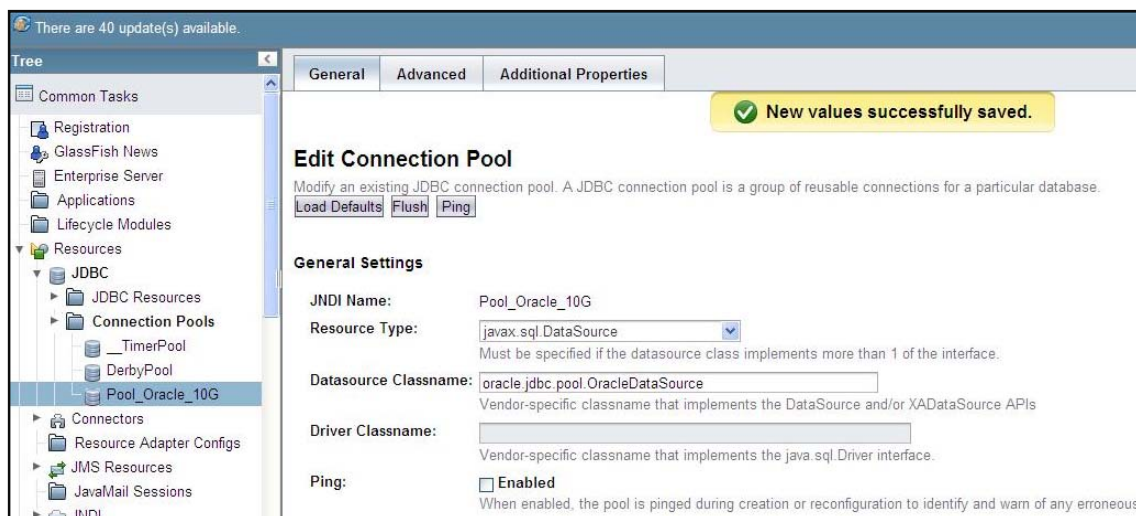


Figura 8.13. Interfaz: Editar Pool de Conexiones de Glassfish V3 ³⁹⁶

6. En el módulo JDBC en la opción JDBC Resources (Fuentes de comunicación con la Base de Datos de Java) en la pantalla de la Figura 8.14., creamos un JNDI con las siguientes propiedades:

- i. JNDI Name: jdbc_/_/Oracle10
- ii. Pool Name: Pool_Oracle_10G



Figura 8.14. Interfaz: Recursos de nuevo JDBC ³⁹⁷

³⁹⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁹⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

7. Verificamos que la fuente de datos se creó correctamente (Figura 8.15.).

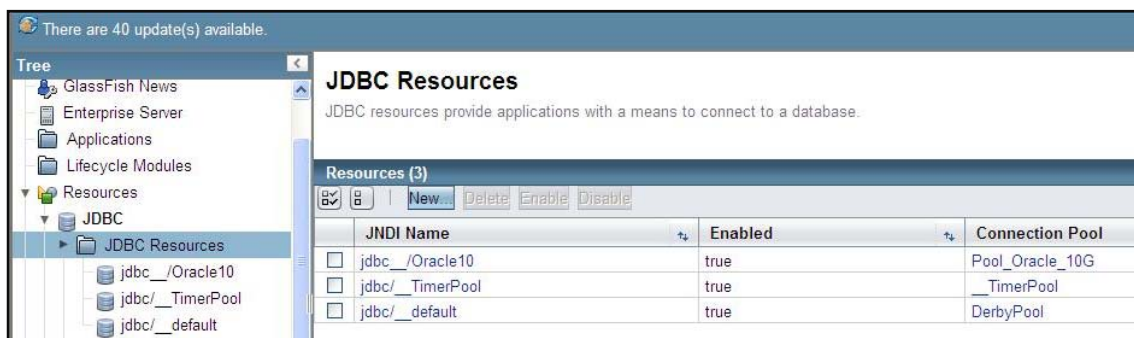


Figura 8.15. Interfaz: Recursos de JDBC ³⁹⁸

8. Ingresamos al módulo de Aplicaciones (Applications) para cargar la aplicación en el Servidor Glassfishv3, damos clic en Deploy en la pantalla de la Figura 8.16.

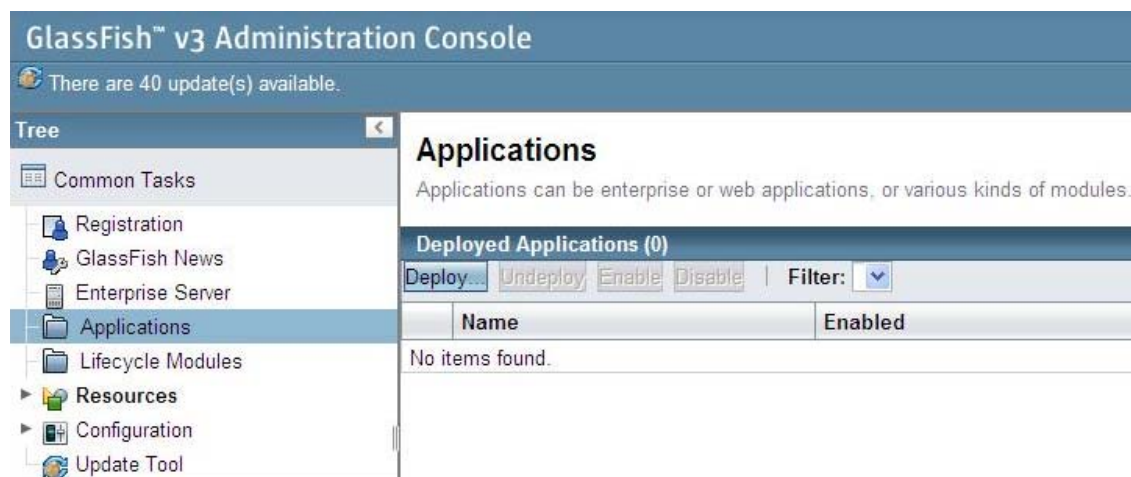


Figura 8.16. Interfaz: Aplicaciones ³⁹⁹

9. Con el botón Examinar en la pantalla de la Figura 8.17., buscamos el archivo **TesisWebJSF.war** (Anexo 5) y lo cargamos con las siguientes propiedades:

- i. Types: Web Applications
- ii. Location: path + TesisWebJSF.war

³⁹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

³⁹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

- iii. Context Root: TesisWebJSF
- iv. Application Name: TesisWebJSF

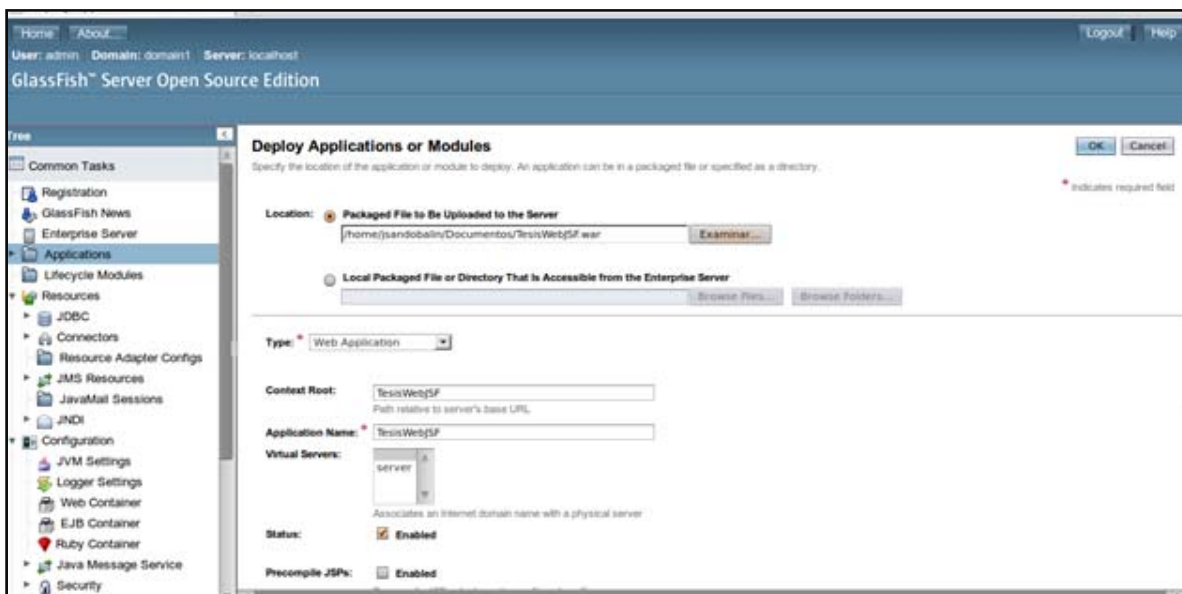


Figura 8.17. Interfaz: Deployar aplicaciones o módulos ⁴⁰⁰

10. Verificamos que la aplicación se cargo correctamente (Figura 8.18.).

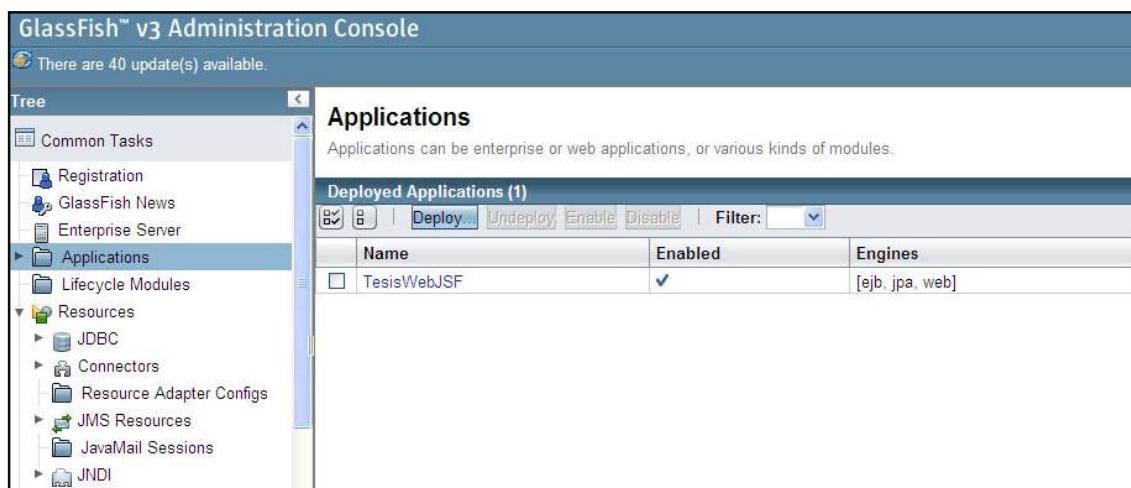


Figura 8.18. Interfaz: Aplicaciones ⁴⁰¹

⁴⁰⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁴⁰¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

ANEXO 4. MANUAL DE USUARIO

CONTENIDO

CONTENIDO.....	353
ACCESO.....	355
1. ADMINISTRACIÓN.....	356
1.1. OPCIONES.....	356
1.2. EMPRESA.....	356
1.2.1. EMPRESA.....	356
1.2.2. SUCURSAL.....	357
1.3. DATOS GENERALES.....	359
1.3.1. UBICACIÓN GEOGRÁFICA.....	359
1.3.1. 1. PAÍS.....	359
1.3.1.2. REGIÓN.....	360
1.3.1.3. PROVINCIA.....	362
1.3.1.4. CIUDAD.....	363
1.3.1.5. ZONA.....	364
1.3.2. FORMA DE PAGO.....	366
1.3.3. IMPUESTO.....	367
1.3.4. DESCUENTOS Y RECARGOS.....	368
1.4. SEGURIDAD.....	370
1.4.1. PERFIL DE USUARIO.....	370
1.4.2. USUARIO.....	372
1.5. PROCESOS.....	373
1.5.1. RESPALDO DE DATOS.....	374
2. INVENTARIO.....	374
2.1. OPCIONES.....	374
2.2. DATOS GENERALES.....	375
2.2.1. LÍNEA DE PRODUCTOS.....	375
2.2.2. GRUPO DE PRODUCTOS.....	376

2.2.3. PRODUCTOS	377
2.2.4. RECETAS	379
2.3. PROCESOS	380
2.3.1. INVENTARIO INICIAL	380
2.3.2. TOMA FÍSICA	381
2.3.3. PRODUCCIÓN	382
2.3.3.1. PREPRODUCCIÓN	382
2.3.3.2. POSTPRODUCCIÓN	384
2.3.4. MOVIMIENTOS DE BODEGA	386
3. FACTURACIÓN	387
3.1. OPCIONES	387
3.2. DATOS GENERALES	388
3.2.1. CLIENTES	388
3.3. PROCESOS	389
3.3.1. FACTURA	390
3.3.2. ANULAR FACTURAS	393
3.3.3. MOVIMIENTOS DE CAJA	394
3.3.4. CUADRE DE CAJA	394
4. REPORTE	395

ACCESO

Para acceder al sistema es necesario ingresar a un navegador web, y digitar la url:

<http://localhost:8085/TesisWebJSF/principal/login.xhtml>

La pantalla principal del sistema (Figura 9.1.) corresponde al acceso, en donde se debe digitar el usuario y la clave, y dar clic en Ingresar.

Si se desea salir del sistema, dar clic en Cancelar.



Figura 9.1. Interfaz: Ingreso al Sistema ⁴⁰²

La siguiente pantalla (Figura 9.2.), muestra el menú principal del sistema, en donde se puede ingresar a las opciones: Administración, Inventario, Facturación y Salir.



Figura 9.2. Interfaz: Menú Principal ⁴⁰³

⁴⁰² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

1. ADMINISTRACIÓN

Al ingresar a la opción Administración del menú principal, se muestra la pantalla de la Figura 9.3.; en este menú se puede ingresar a: Opciones, Empresa, Datos Generales, Seguridad, Procesos y Reportes



Figura 9.3. Interfaz: Menú de Administración ⁴⁰⁴

1.1. OPCIONES

Menú Principal

Al dar clic en la pestaña Opciones – Menú Principal, se regresa al menú principal del sistema (Figura 9.2.).

Salir

Si se desea salir del sistema se elige la pestaña Opciones – Salir, y se mostrará la pantalla de acceso (Figura 9.1.).

1.2. EMPRESA

1.2.1. EMPRESA

En la opción Empresa – Empresa, se muestra la pantalla de la Figura 9.4., en donde se deben ingresar los datos de la empresa: código, nombre, razón social, ruc, dirección, teléfono, celular, código postal, email; una vez ingresados los datos dar clic en **Grabar**.

Si se desea actualizar los datos se ingresa los campos a modificar y luego dar clic en Grabar.

⁴⁰³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁰⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

La opción **Limpiar** deja en blanco los campos; para regresar al menú principal de Administración dar clic en **Salir**.

E M P R E S A

Código:	<input type="text" value="01"/>
Nombre:	<input type="text" value="EMPRESA"/>
Razón Social:	<input type="text" value="EMPRESA"/>
Ruc:	<input type="text" value="1715460372"/>
Dirección:	<input type="text" value="QUITO"/>
Teléfono:	<input type="text" value="022334034"/>
Tel. /Celular:	<input type="text"/>
Cód. Postal:	<input type="text"/>
Email:	<input type="text"/>

Grabar
Limpiar
Salir

Figura 9.4. Interfaz: Empresa⁴⁰⁵

1.2.2. SUCURSAL

En la opción Empresa – Sucursal del menú principal de Administración, se muestra la pantalla de la Figura 9.5., en donde se pueden visualizar las sucursales existentes, un menú con Nuevo y Salir y un menú para Modificar y Eliminar

S U C U R S A L

SUCURSAL					
C O D I G O ▲	S U C U R S A L ▼	C I U D A D ▼	Z O N A ▼		
01	SUCURSAL	QUITO	ZONA NORTE		
11	BODEGA CENTRAL	QUITO	ZONA NORTE		

Nuevo
Salir

Figura 9.5. Interfaz: Sucursal⁴⁰⁶

⁴⁰⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁴⁰⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

Al ingresar a la opción **Nuevo** de sucursal se muestra la pantalla de la Figura 9.6., para el ingreso de una nueva sucursal se debe elegir la Ciudad y Zona a la que pertenece la sucursal, se debe ingresar el código y el nombre de la sucursal, una vez finalizado el ingreso, dar clic en Grabar.

Si se desea limpiar los campos ingresados dar clic en Limpiar.

Para retornar al menú principal de Administración dar clic en Retornar.



Figura 9.6. Interfaz: Ingreso de Sucursal⁴⁰⁷



Elegir el ícono:  para **Modificar** los datos de una sucursal, y se muestra la pantalla de la Figura 9.7., donde se puede modificar la ciudad, el sector de la ciudad a la pertenece y el nombre otorgado para dicha sucursal.



Figura 9.7. Interfaz: Modificar Sucursal⁴⁰⁸

⁴⁰⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010

⁴⁰⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea **Eliminar** una sucursal se debe elegir el ícono:  en la pantalla de la Figura 9.5.

1.3. DATOS GENERALES

1.3.1. UBICACIÓN GEOGRÁFICA

En el menú principal de Administración tenemos la opción de Datos Generales- Ubicación Geográfica como se muestra en la pantalla de la Figura 9.8., que nos permite acceder a la información de la localidad de cada sucursal.



Figura 9.8. Interfaz: Menú Datos Generales ⁴⁰⁹

1.3.1.1. PAÍS

Ingresamos en la opción País que se encuentra dentro de Ubicación Geográfica, y se mostrará la pantalla de la Figura 9.9. En esta pantalla tenemos un listado de los países ingresados, los íconos de Modificar y Eliminar, y los botones Nuevo y Salir.



Figura 9.9. Interfaz: País ⁴¹⁰

⁴⁰⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴¹⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Cuando se ingresa a la opción **Nuevo** se presenta la pantalla de la Figura 9.10., que nos permite ingresar los datos del país como código y nombre.

Para almacenar los datos elegir **Grabar**; si se desea borrar los datos de las cajas de texto elegir **Limpiar**, y finalmente para regresar a las opciones de país elegir **Retornar**.

Figura 9.10. Interfaz: Ingreso de País ⁴¹¹



Para **Modificar** los datos de un País se debe elegir el ícono de Modificar  en la pantalla de Figura 9.9., y se muestra la pantalla de la Figura 9.11., en donde solo se puede cambiar el nombre del país.

Figura 9.11. Interfaz: Modificar País ⁴¹²

Si se desea borrar la información del país elegir el ícono:  en la pantalla de la Figura 9.9.

1.3.1.2. REGIÓN

Al ingresar a la opción Región que se encuentra en Ubicación Geográfica, se muestra la pantalla de la Figura 9.12., en donde se visualiza un listado de regiones; los íconos de Modificar y Eliminar, y los botones Nuevo y Salir.

⁴¹¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴¹² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

REGION	
CODIGO ▲	DESCRIPCION ↓
01	SIERRA

Nuevo Salir

Figura 9.12. Interfaz: Región ⁴¹³

Para ingresar una nueva región elegir el botón **Nuevo**; se muestra la pantalla de la Figura 9.13. Una vez ingresados los datos elegir la opción **Grabar**, si se desea borrar la información de las cajas de texto elegir **Limpiar**, y para salir dar clic en **Retornar**.


REGION

Código:

Nombre:

Grabar Limpiar Retornar

Figura 9.13. Interfaz: Ingreso de Región ⁴¹⁴

Si se desea Modificar la información de la Región dar clic en el ícono:  en la pantalla de la Figura 9.12., y se mostrará la pantalla de la Figura 9.14.; actualizar los datos y **Grabar** los cambios. Para limpiar los campos elegir **Limpiar**, y para salir de esta pantalla **Retornar**.

REGION

Código:

Nombre:


Grabar Limpiar Retornar Modificar

Figura 9.14. Interfaz: Modificar Región ⁴¹⁵

⁴¹³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴¹⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴¹⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea borrar la información de la región elegir el ícono:  en la pantalla de la Figura 9.12.

1.3.1.3. PROVINCIA

Para especificar la provincia en donde se encuentra la sucursal, ingresar a la opción Provincia que se encuentra en Ubicación Geográfica; se mostrará la pantalla de la Figura 9.15., en donde se despliegan todas las provincias ingresadas y se puede acceder a las opciones Modificar, Eliminar, Nuevo y Salir.



PROVINCIA			
CODIGO ▲	PROVINCIA ▼	REGION ▼	PAIS ▼
01	PICHINCHA	SIERRA	ECUADOR
02	GUAYAS	SIERRA	ECUADOR

Figura 9.15. Interfaz: Provincia ⁴¹⁶

Al ingresar a la opción **Nuevo** se despliega la pantalla de la Figura 9.16., en donde se debe seleccionar el País y la Región, e ingresar el código y nombre de la provincia, luego dar clic en **Grabar**. Para borrar los datos de las cajas de texto elegir **Limpiar** y para retornar elegir **Salir**.



PROVINCIA

País:

Región:


Código:

Nombre:

Figura 9.16. Interfaz: Ingreso de Provincia ⁴¹⁷

⁴¹⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴¹⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea actualizar los datos de la provincia dar clic en el ícono:  de la Provincia en la pantalla de la Figura 9.15., y se desplegará la pantalla de la Figura 9.17. en donde se puede elegir entre las opciones Grabar, Limpiar o Salir.



PROVINCIA


País:

Región:

Código:

Nombre:

Figura 9.17. Interfaz: Modificar Provincia ⁴¹⁸

Para borrar la información de la provincia elegir el ícono:  en la pantalla de la Figura 9.15.

1.3.1.4. CIUDAD

Cuando se accede a la opción Ciudad dentro de Ubicación Geográfica, se visualiza la pantalla de la Figura 9.18., aquí se puede observar los datos de las ciudades ingresadas y las opciones para modificar, eliminar o crear una ciudad; además del botón salir.



CIUDAD

CIUDAD				
CODIGO ▲	CIUDAD ⇅	PROVINCIA ⇅		
01	QUITO	PICHINCHA		

Figura 9.18. Interfaz: Ciudad ⁴¹⁹

⁴¹⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴¹⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea ingresar una nueva ciudad elegir la opción **Nuevo**, se desplegará la pantalla de la Figura 9.19., en donde se debe elegir el país y la provincia e ingresar los datos del código y nombre de la ciudad.

Para guardar la información elegir **Grabar**, para limpiar los campos de texto elegir **Limpiar** y finalmente para retornar a las opciones de ciudad elegir **Retornar**.



Figura 9.19. Interfaz: Ingreso de Ciudad ⁴²⁰



Elegir el ícono:  de la ciudad para modificar sus datos en la pantalla de la Figura 9.18., se muestra la pantalla de la Figura 9.20, actualizar los datos y grabar los cambios; elegir **Limpiar** para borrar los textos y **Retornar** para salir.



Figura 9.20. Interfaz: Modificar Ciudad ⁴²¹

Elegir el ícono:  para borrar toda la información de la ciudad en la pantalla de la Figura 9.18.

1.3.1.5. ZONA

En zona se ingresa el sector específico en el cual está ubicada la sucursal, al ingresar a la opción zona dentro de ubicación geográfica se muestra la pantalla de la Figura 9.21. Se puede acceder a modificar, eliminar, nuevo y salir.

⁴²⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴²¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Z O N A

Z O N A				
C O D I G O ▲	Z O N A ↓	C I U D A D ↓		
01	ZONA NORTE	QUITO		
02	ZONA CENTRO	QUITO		


Figura 9.21. Interfaz: Zona ⁴²²

Para crear una nueva zona elegir la opción **Nuevo**, se visualizará la pantalla de la Figura 9.22., en donde se debe elegir la provincia y ciudad e ingresar el código y nombre de la zona. Una vez ingresados los datos **Grabar**, o se puede elegir las opciones **Limpiar** para borrar los datos del texto y si se desea salir dar clic en **Retornar**.

Z O N A

Prov.:	PICHINCHA
Ciudad:	QUITO
Cód.Zona:	
Nombre:	

Figura 9.22. Interfaz: Ingreso a Zona ⁴²³

Para actualizar los datos elegir el ícono:  de la zona en la pantalla de la Figura 9.21., y se accede a la pantalla de la Figura 9.23., en donde se puede elegir grabar, limpiar y retornar.


Z O N A

Prov.:	PICHINCHA
Ciudad:	QUITO
Cód.Zona:	01
Nombre:	ZONA NORTE

Figura 9.23. Interfaz: Modificar Zona ⁴²⁴

⁴²² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴²³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea eliminar los datos de zona elegir el ícono:  en la pantalla de la Figura 9.21.

1.3.2. FORMA DE PAGO

Al ingresar a la opción Forma de Pago dentro de Datos Generales se visualiza la pantalla de la Figura 9.24., en donde se despliegan todas las formas de pago que se aceptará en el momento de la facturación; además se muestran las opciones modificar, eliminar, nuevo y salir.



FORMA DE PAGO		
CODIGO ▲	FORMA DE PAGO ▼	OBSERVACION ▼
01	EFECTIVO	
02	CHEQUE	
03	TARJETA	

Figura 9.24. Interfaz: Forma de Pago ⁴²⁵

Para ingresar una forma de pago elegir la opción **Nuevo**. La pantalla que se muestra es la de la Figura 9.25, se debe ingresar el código, nombre y observación, y para guardar los datos elegir **Grabar**.

Para borrar los datos de las cajas de texto elegir **Limpiar** y para salir **Retornar**.



FORMA DE PAGO

Código:


Nombre:

Observación:

Figura 9.25. Interfaz: Ingreso de Forma de Pago ⁴²⁶

⁴²⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴²⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Los datos de forma de pago se pueden modificar a excepción del código, para esto elegir el ícono:  en la pantalla de la Figura 9.24., y se mostrará la pantalla de la Figura 9.26.

FORMA DE PAGO

Código:

Nombre:

Observación:

Grabar
Limpiar
Retornar

Figura 9.26. Interfaz: Modificar Forma de Pago ⁴²⁷

Para borrar la forma de pago elegir el ícono:  en la pantalla de la Figura 9.24.

1.3.3. IMPUESTO

Al ingresar a la opción Impuesto dentro de Datos Generales, se despliega la pantalla de la Figura 9.27., en donde se muestran los datos de todos los impuestos y las opciones: modificar, eliminar, nuevo y salir.

IMPUESTOS

IMPUESTOS				
CÓDIGO ▲	NOMBRE ▲	PORCENTAJE	TIPO ▲	
01	IVA 12%	12.00	Incluido(VAT)	 
02	IVA 0%	0.00	Incluido(VAT)	 

Nuevo
Salir

Figura 9.27. Interfaz: Impuestos ⁴²⁸

Para crear un impuesto elegir la opción **Nuevo**, aquí se pueden ingresar los datos del impuesto como código, nombre, porcentaje y elegir el tipo; una vez ingresados los datos elegir **Grabar**.

⁴²⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴²⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴²⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea borrar la información de los campos dar clic en **Limpiar**, y para salir elegir **Retornar**.

IMPUESTOS

Código:


Nombre:


Porcentaje:

Tipo:

Grabar Limpiar Retornar

Figura 9.28. Interfaz: Ingreso de Impuestos ⁴²⁹

Para actualizar los datos del impuesto, dar clic en el ícono:  en la pantalla de la Figura 9.27., y se muestra la pantalla de la Figura 9.29., en donde se puede grabar, limpiar y retornar.

Elegir el ícono:  para borrar todos los datos de impuestos en la pantalla de la Figura 9.27.

IMPUESTOS

Código:

Nombre:

Porcentaje:

Tipo:

Grabar Limpiar Retornar

Figura 9.29. Interfaz: Modificar de Impuestos ⁴³⁰

1.3.4. DESCUENTOS Y RECARGOS

En la opción Descuentos y Recargos, dentro del Datos Generales, se visualiza la pantalla de la Figura 9.30., en donde están los descuentos o recargos posibles para facturación. Se puede acceder a modificar, eliminar, nuevo y salir.

⁴²⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴³⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.


DESCUENTOS Y RECARGOS				
CODIGO ▲	NOMBRE ↓	PORCENTAJE	TIPO ↓	
01	DESCUENTO	5.00	Descuento	 
02	RECARGO	10.00	Recargo	 

Figura 9.30. Interfaz: Descuentos y Recargos ⁴³¹

Para ingresar los datos de un descuento o recargo elegir la opción **Nuevo**, y se muestra la pantalla de la Figura 9.31., los datos que se deben ingresar son: código, nombre, porcentaje y tipo.

DESCUENTOS Y RECARGOS	
Código:	<input type="text"/>
Nombre:	<input type="text"/>
Porcentaje:	<input type="text"/>
Tipo:	Descuento <input type="button" value="v"/>
<input type="button" value="Grabar"/> <input type="button" value="Limpiar"/> <input type="button" value="Retornar"/>	

Figura 9.31. Interfaz: Ingreso de Descuentos y Recargos ⁴³²


Elegir el ícono de **modificar**:  en la pantalla de la Figura 9.30., para actualizar los campos de descuentos o recargos, se muestra la pantalla de la Figura 9.32., en donde se puede elegir entre grabar, limpiar o retornar.

DESCUENTOS Y RECARGOS	
Código:	<input type="text" value="01"/>
Nombre:	<input type="text" value="DESCUENTO"/>
Porcentaje:	<input type="text" value="5"/>
Tipo:	Descuento <input type="button" value="v"/>
<input type="button" value="Grabar"/> <input type="button" value="Limpiar"/> <input type="button" value="Retornar"/>	

Figura 9.32. Interfaz: Modificar Descuentos y Recargos ⁴³³

⁴³¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴³² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea borrar todos los datos de un descuento o recargo elegir el ícono:  en la pantalla de la Figura 9.30.

1.4. SEGURIDAD

En el menú de administración se puede acceder a la opción Seguridad como se muestra en la pantalla de Figura 9.33.



Figura 9.33. Interfaz: Seguridad ⁴³⁴

1.4.1. PERFIL DE USUARIO

Es necesario crear un perfil para mantener la seguridad en el sistema; al acceder a Perfil de Usuario dentro de Seguridad se muestra la pantalla de la Figura 9.34. Se despliegan todos los perfiles ingresados y las opciones: modificar, eliminar, nuevo y salir



Figura 9.34. Interfaz: Perfil ⁴³⁵

Para ingresar los datos de un perfil elegir la opción **Nuevo**, aquí se despliega la pantalla de la Figura 9.35., en donde se debe ingresar el código y nombre, y elegir **Grabar**.

⁴³³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴³⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴³⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

PERFIL

Código:

Nombre:

Figura 9.35. Interfaz: Ingreso de Perfil ⁴³⁶

Luego de haber grabado los datos se debe asignar los accesos, para ellos elegimos la opción **Permisos** y se muestra la pantalla de la Figura 9.36., en donde se despliegan todas las opciones del sistema según los diferentes módulos.

Para restringir el acceso a una de las opciones dar clic en la casilla de verificación, una vez asignados los permisos **Retornar**.

PERMISOS DE ACCESO AL SISTEMA


Módulo:

PERMISOS		
C O D I G O ▲	D E S C R I P C I O N ●	
01	EMPRESA	<input checked="" type="checkbox"/>
02	SUCURSAL	<input checked="" type="checkbox"/>
03	PAIS	<input checked="" type="checkbox"/>
04	REGION	<input checked="" type="checkbox"/>
05	PROVINCIA	<input checked="" type="checkbox"/>
06	CIUDAD	<input checked="" type="checkbox"/>
07	ZONA	<input checked="" type="checkbox"/>
08	FORMA DE PAGO	<input checked="" type="checkbox"/>
09	IMPUESTO	<input checked="" type="checkbox"/>
10	DESCUENTOS/RECARGOS	<input checked="" type="checkbox"/>
11	PERFIL DE USUARIO	<input checked="" type="checkbox"/>
12	USUARIO	<input checked="" type="checkbox"/>
13	RESPALDO DE DATOS	<input checked="" type="checkbox"/>

Figura 9.36. Interfaz: Permisos de Acceso al Sistema ⁴³⁷

⁴³⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

En la pantalla de la Figura 9.35. se puede elegir también las opciones: limpiar y retornar.

Para **modificar** los datos de un perfil elegir el ícono:  en la pantalla de la Figura 9.34., y se muestra la pantalla de la Figura 9.37., en esta pantalla podemos actualizar el nombre del perfil y accediendo a la opción Permisos (Figura 9.36) se puede modificar los mismos.



P E R F I L


Código:

Nombre:

Grabar Permisos

Limpiar Retornar

Figura 9.37. Interfaz: Modificar perfil ⁴³⁸

Si se desea borrar todos los datos de un perfil elegir el ícono:  en la pantalla de la Figura 9.34.

1.4.2. USUARIO

Al ingresar en la opción Usuario dentro Seguridad se muestra la pantalla de la Figura 9.38., en donde se despliegan los datos de los usuarios y se puede acceder a las opciones: modificar, eliminar, nuevo y salir.



U S U A R I O

USUARIO		
CODIGO ▲	USUARIO ↓	PERFIL ↓
01	JULIO CESAR	ADMINISTRADOR

Figura 9.38. Interfaz: Usuario ⁴³⁹


⁴³⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴³⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Para crear un usuario elegir **Nuevo**, se desplegará la pantalla de la Figura 9.39., aquí se deben ingresar los datos del usuario, primero se le asigna un perfil de los que ya se encuentren creados, luego ingresar los datos: código, nombre, estado, sucursal de ventas, login, contraseña y confirmación de la contraseña. Para almacenar los datos elegir **Grabar**, también se puede limpiar los campos o retornar.



Figura 9.39. Interfaz: Ingreso de Usuario ⁴⁴⁰

Para modificar los datos del usuario elegir el ícono:  en la pantalla de la Figura 9.38.

Si se desea borrar un usuario elegir el ícono:  en la pantalla de la Figura 9.38.

1.5. PROCESOS

En el menú de Administración se puede acceder a la opción Procesos como se muestra en la pantalla de Figura 9.40.

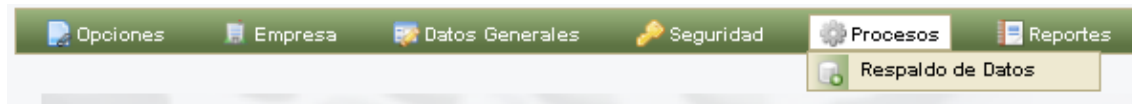


Figura 9.40. Interfaz: Procesos ⁴⁴¹

⁴³⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁴⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁴¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

1.5.1. RESPALDO DE DATOS

Al ingresar a respaldo de datos se despliega la pantalla de la Figura 9.41., solo basta con dar clic en **Iniciar respaldo** para que se almacene la información del sistema.

Si se desea retornar dar clic en Salir.



Figura 9.41. Interfaz: Respaldo de Datos ⁴⁴²

2. INVENTARIO

En el menú principal se puede acceder a la opción Inventario, aquí se muestra la pantalla de la Figura 9.42.; en este menú se puede ingresar a: Opciones, Datos Generales, Procesos y Reportes.

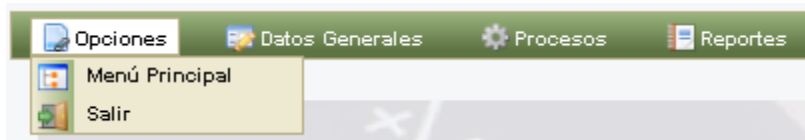


Figura 9.42. Interfaz: Menú de Inventario ⁴⁴³

2.1. OPCIONES

Menú Principal

Al dar clic en la pestaña Opciones – Menú Principal, se regresa al menú principal del sistema (Figura 9.2.).

Salir

⁴⁴² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁴³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Si se desea salir del sistema se elige la pestaña Opciones – Salir, y se mostrará la pantalla de acceso (Figura 9.1.).

2.2. DATOS GENERALES

El menú de Datos Generales es el que se despliega en la pantalla de la Figura 9.43.



Figura 9.43. Interfaz: Menú de Inventario ⁴⁴⁴

2.2.1. LÍNEA DE PRODUCTOS

Al ingresar a la opción Línea de productos se despliega la pantalla de la Figura 9.44., en donde se visualizan las líneas de los productos y se puede acceder a las opciones: modificar, eliminar, nuevo y salir.



Figura 9.44. Interfaz: Línea de Productos ⁴⁴⁵

⁴⁴⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.


⁴⁴⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.


Para crear una línea de productos elegir la opción **Nuevo**, y se desplegará la pantalla de la Figura 9.45., en donde se debe ingresar el código y nombre y dar clic en **Grabar**.

Para borrar los datos de las cajas de texto elegir **Limpiar** y para salir **Retornar**.



Figura 9.45. Interfaz: Ingreso de Línea de Productos ⁴⁴⁶

Si se desea actualizar los datos de una línea de productos elegir el ícono:  en la pantalla de la Figura 9.44.

Para borrar una línea de productos elegir el ícono:  en la pantalla de la Figura 9.44.

2.2.2. GRUPO DE PRODUCTOS

Si se elige la opción Grupo de Productos dentro de Datos Generales se despliega la pantalla de la Figura 9.46., aquí se visualiza los datos de los grupos de productos existentes, y se puede acceder a las opciones: modificar, eliminar, nuevo y salir.



GRUPO DE PRODUCTOS		
CODIGO ▲	GRUPO ⇅	LINEA ⇅
01	GRUPO	LINEA
02	ACEITES	ABASTOS

Figura 9.46. Interfaz: Grupo de Productos ⁴⁴⁷


⁴⁴⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.


Al ingresar a la opción **Nuevo**, se despliega la pantalla de la Figura 9.47., se debe elegir la línea de producto a la cual pertenece este nuevo grupo de productos, el código y nombre; luego dar clic en **Grabar**.

Se puede elegir **Limpiar** para borrar los datos de los campos de texto y **Retornar** para salir.



Figura 9.47. Interfaz: Ingreso de Grupo de Productos ⁴⁴⁸

Para modificar los datos del grupo de productos elegir el ícono:  en la pantalla de la Figura 9.46.

Para borrar un grupo de productos elegir el ícono:  en la pantalla de la Figura 9.46.

2.2.3. PRODUCTOS

Si se ingresa a la opción Productos dentro de Datos Generales, se despliega la pantalla de la Figura 9.48., en donde se puede visualizar todos los productos existentes, si es una lista muy grande de productos se puede consultar a través de los filtros que se muestran en la parte superior.

Además se puede acceder a las opciones: modificar, eliminar, nuevo y salir.

⁴⁴⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁴⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

PRODUCTO

PRODUCTO						
CODIGO	NOMBRE	PRECIO A	IMPUESTO	TIPO	GRUPO	LINEA
01	ARROZ	0.00	IVA 0%	INGREDIENTE	GRUPO	LINEA
02	PAPAS	0.00	IVA 0%	INGREDIENTE	GRUPO	LINEA
03	LOMO FINO	0.00	IVA 0%	INGREDIENTE	GRUPO	LINEA
04	LOMO SUAVE	0.00	IVA 0%	INGREDIENTE	GRUPO	LINEA
11	COCA 1/2 LITR	1.00	IVA 12%	NORMAL	GRUPO	LINEA
12	JUGO SUNNY	0.60	IVA 12%	NORMAL	GRUPO	LINEA
13	CERVEZA	1.10	IVA 12%	NORMAL	GRUPO	LINEA
21	ARROZ CON CARNE	3.50	IVA 12%	RECETA	GRUPO	LINEA
22	ARROZ MIXTO	4.50	IVA 12%	RECETA	GRUPO	LINEA

Figura 9.48. Interfaz: Productos ⁴⁴⁹

Para ingresar un producto se debe elegir la opción **Nuevo**, y se despliega la pantalla de la Figura 9.49., en donde se debe elegir la línea y el grupo al cual pertenece el producto, e ingresar el código, nombre, impuesto, precio A, precio B, precio C, tipo (si es una receta se activa el combo de tipo de receta) y estado.

Una vez ingresados los datos dar clic en **Grabar**, también se tienen las opciones Limpiar y Retornar.


PRODUCTO


Línea Prod:	LINEA
Grupo Prod:	GRUPO
Código:	
Nombre:	
Impuesto:	IVA 0%
Precio A:	0.00
Precio B:	0.00
Precio C:	0.00
Tipo:	NORMAL
Tipo Receta:	POSTPRODUCCION
Estado:	ACTIVO

Figura 9.49. Interfaz: Ingreso de Productos ⁴⁵⁰

⁴⁴⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁵⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Para modificar los datos de un producto elegir el ícono:  en la pantalla de la Figura 9.48.

Para borrar productos elegir el ícono:  en la pantalla de la Figura 9.48.


2.2.4. RECETAS

Al ingresar a la opción Recetas dentro de Datos Generales, se visualiza la pantalla de la Figura 9.50., en donde hay listado con las recetas existentes y las opciones: agregar ingredientes y salir.

RECETAS			
CODIGO ▲	DESCRIPCION ↓	TIPO ↓	
21	ARROZ CON CARNE	PREPRODUCCION	
22	ARROZ MIXTO	POSTPRODUCCION	

Figura 9.50. Interfaz: Recetas ⁴⁵¹

Si se elige la opción **Agregar Ingredientes** se despliega la pantalla de de Figura 9.51., en donde se debe ir ingresando el ingrediente y la cantidad que conforma la receta. Una vez que se ha ingresado todos los ingredientes de la receta se debe **Grabar** los cambios. Para limpiar las cajas de texto elegir **Limpiar** y para salir elegir **Retornar**.

Si se desea borrar algún ingrediente que conforme la receta elegir el ícono:  del ingrediente.

⁴⁵¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

ARROZ CON CARNE

Ingrediente:
=== Escriba el nombre del Ingrediente ===

Cantidad:

Grabar
 Limpiar
 Retornar

INGREDIENTES			
CODIGO ▲	NOMBRE ↕	CANTIDAD ↕	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
2	LOMO FINO	1.000	
<< < > >>			

Figura 9.51. Interfaz: Ingredientes ⁴⁵²

2.3. PROCESOS

En el menú de Inventario se tiene la opción Procesos como se muestra en la pantalla de la Figura 9.52.



Figura 9.52. Interfaz: Procesos Inventario ⁴⁵³

2.3.1. INVENTARIO INICIAL

Al ingresar a la opción Inventario Inicial se despliega la pantalla de la Figura 9.53., en donde se debe elegir la bodega de la cual se quiere ingresar el inventario inicial.

⁴⁵² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁵³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Luego se debe ingresar el producto, su costo y cantidad y elegir la opción **Grabar**, los productos que se sigan ingresando se desplegarán en la parte de abajo con la opción de poder modificar los valores de los mismos.

Si se desea limpiar las cajas de texto elegir **Limpiar**. Cuando se ingresen todos los productos existentes en la sucursal se debe elegir la opción **Generar Inv. Inicial**.

Para imprimir el inventario elegir **Imprimir**, y para retornar al menú **Salir**.

INVENTARIO INICIAL				
CODIGO ▲	DESCRIPCION ↓	COSTO	CANTIDAD	
01	ARROZ	0.80000	100.000	
02	PAPAS	2.35000	100.000	
03	LOMO FINO	1.55000	200.000	
04	LOMO SUAVE	1.45000	159.000	
11	COCA 1/2 LITR	0.90000	60.000	


Figura 9.53. Interfaz: Inventario Inicial ⁴⁵⁴

2.3.2. TOMA FÍSICA

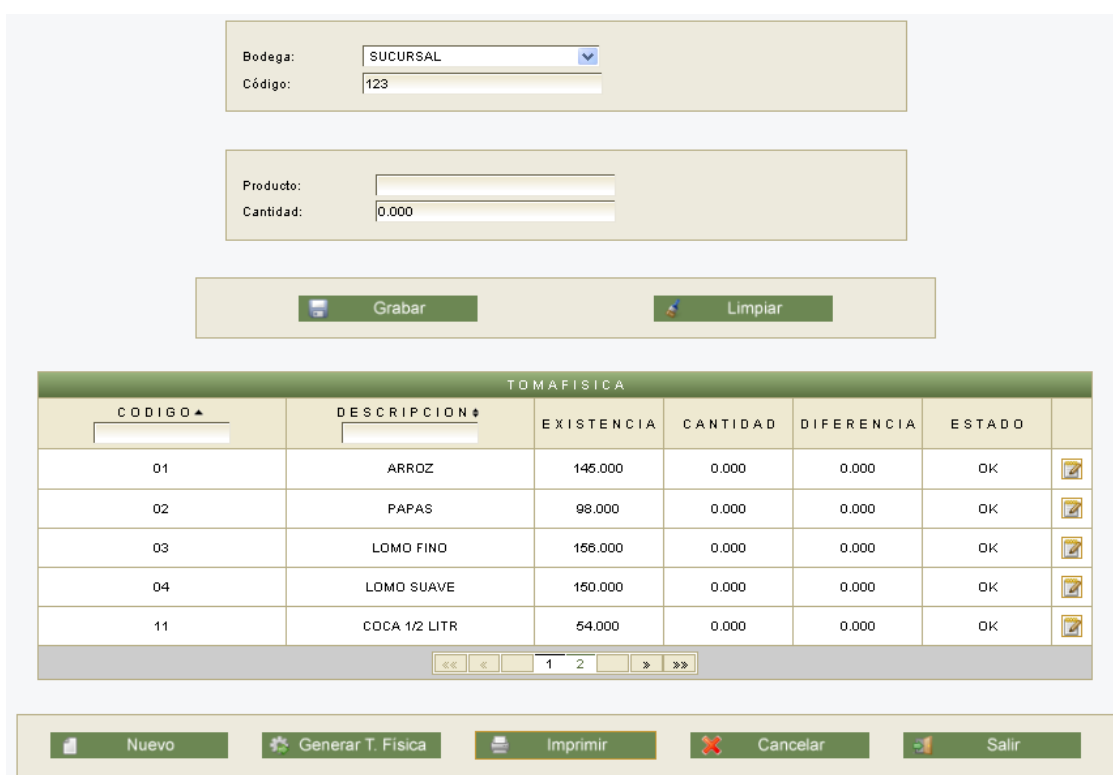
Para controlar las existencias en bodega existe la opción Toma Física (Figura 9.54.), en donde se debe elegir la opción **Nuevo**, después elegir la bodega en la cual se va a realizar la toma física de productos, luego se debe ingresar el producto y la cantidad existente, y elegir la opción **Grabar**.

⁴⁵⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Se desplegarán los datos del producto: código, descripción, existencia, cantidad, diferencia y estado; de esta manera se va controlando si la existencia del producto en bodega es la correcta.

Para modificar los datos del producto elegir el ícono: , luego de cambiar los datos se debe grabar.

Una vez ingresados todos los productos se debe elegir **Generar T. Física**; para imprimir los datos de la toma física seleccionar **Imprimir**, si ya no se desea realizar la toma **Cancelar** y para retornar al menú **Salir**.



The interface is titled 'TOMAFISICA'. It contains the following elements:

- Form Fields:**
 - Bodega: SUCURSAL (dropdown)
 - Código: 123 (text input)
 - Producto: (empty text input)
 - Cantidad: 0.000 (text input)
- Buttons:** Grabar, Limpiar
- Table:**






CODIGO ▲	DESCRIPCION *	EXISTENCIA	CANTIDAD	DIFERENCIA	ESTADO	
01	ARRDZ	145.000	0.000	0.000	OK	
02	PAPAS	98.000	0.000	0.000	OK	
03	LOMO FINO	156.000	0.000	0.000	OK	
04	LOMO SUAVE	150.000	0.000	0.000	OK	
11	COCA 1/2 LITR	54.000	0.000	0.000	OK	
- Page Navigation:** << < 1 2 > >>
- Bottom Bar Buttons:** Nuevo, Generar T. Física, Imprimir, Cancelar, Salir

Figura 9.54. Interfaz: Toma Física ⁴⁵⁵

2.3.3. PRODUCCIÓN

2.3.3.1. PREPRODUCCIÓN

Al ingresar a la opción Preproducción dentro de Producción se despliega la pantalla de la Figura 9.55. Para generar una producción se debe seleccionar la bodega, elegir la opción **Nuevo**, dar clic en el botón **Consultar** y se despliega la

⁴⁵⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

pantalla de la Figura 9.56., en donde se debe Seleccionar la receta, luego se debe ingresar la cantidad a producir, y dar clic en **Producir**.

Las recetas que se van produciendo se muestran en un listado, una vez ingresada toda la producción **Grabar** los datos. Para limpiar las cajas de texto elegir **Limpiar**, además existen las opciones: imprimir, cancelar y salir.

Al elegir la opción **Imprimir** se despliega la pantalla de la Figura 9.57., en donde se visualiza todas las preproducciones.

Cod. Prod: 141
 Bodega: BODEGA CENTRAL
 Cód. Receta:
 Receta:
 Costo: 0.00000
 Cantidad: 0.000

Producir Limpiar Consultar

INGREDIENTES			
CODIGO ▲	PRODUCTO ▼	CANTIDAD	CANTIDAD UTILIZADA
		0.000	0.000

BODEGA					
CODIGO ▲	PRODUCTO ▼	CANTIDAD BODEGA	COSTO	CANTIDAD UTILIZADA	CANTIDAD SOBRANTE
		0.000	0.00000	0.000	0.000

Nuevo Grabar Imprimir Cancelar Salir

Figura 9.55. Interfaz: Preproducción ⁴⁵⁶

⁴⁵⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.



Figura 9.56. Interfaz: Seleccionar Recetas ⁴⁵⁷

PREPRODUCCION						
CODIGO #	FECHA #	CANTIDAD #	RECETA #	COSTO #	ESTADO #	USUARIO #
145	02/09/2010	1.000	ARROZ CON CARNE	2.35	REGISTRADO	JULIO CESAR
83	17/08/2010	3.000	ARROZ CON CARNE	7.05	REGISTRADO	JULIO CESAR
82	17/08/2010	1.000	ARROZ CON CARNE	2.35	REGISTRADO	JULIO CESAR
62	15/08/2010	3.000	ARROZ CON CARNE	7.05	REGISTRADO	JULIO CESAR
61	15/08/2010	2.000	ARROZ CON CARNE	4.70	REGISTRADO	JULIO CESAR
41	31/07/2010	1.000	ARROZ CON CARNE	2.35	REGISTRADO	JULIO CESAR
23	28/07/2010	1.000	ARROZ CON CARNE	2.35	REGISTRADO	JULIO CESAR
4	28/07/2010	7.000	ARROZ CON CARNE	16.45	REGISTRADO	JULIO CESAR
3	28/07/2010	5.000	ARROZ CON CARNE	11.75	REGISTRADO	JULIO CESAR

Figura 9.57. Interfaz: Preproducciones ⁴⁵⁸

2.3.3.2. POSTPRODUCCIÓN

En la opción postproducción se despliega la pantalla de la Figura 9.58., en donde se debe seleccionar la bodega, elegir la opción **Nuevo** para crear la postproducción, dar clic en **Consultar** y se despliega la pantalla de la Figura 9.56., luego **Seleccionar** la receta e ingresar la cantidad, sobrante y desperdicio, finalmente dar clic en **Producir** y se despliegan los datos ingresados en la parte de ingredientes y bodega.

Una vez ingresada la postproducción **Grabar** los datos.

⁴⁵⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁵⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Para borrar la información de las cajas de texto elegir la opción Limpiar.
Además se puede elegir entre las opciones: imprimir, cancelar y salir.
Cuando se elige la opción **Imprimir** se despliegan todas las postproducciones realizadas como se muestra en la pantalla de la Figura 9.59.

The interface is divided into several sections:

- DATOS:** Fields for 'Cod. Prod.', 'Bodega' (dropdown menu showing 'SUCURSAL'), 'Cód. Receta', and 'Receta'.
- CANTIDADES:** Fields for 'Costo', 'Cantidad', 'Sobranie', and 'Desperdicio', all with a value of '0.0000'.
- Buttons:** 'Producir', 'Limpiar', and 'Consultar'.
- INGREDIENTES Table:**

CODIGO	PRODUCTO	CANTIDAD	CANTIDAD UTILIZADA
		0.000	0.000
- BODEGA Table:**

CODIGO	PRODUCTO	CANTIDAD BODEGA	COSTO	CANTIDAD UTILIZADA	CANTIDAD SOBRAANTE
		0.000	0.00000	0.000	0.000
- Bottom Buttons:** 'Nuevo', 'Grabar', 'Imprimir', 'Cancelar', and 'Salir'.

Figura 9.58. Interfaz: Postproducción ⁴⁵⁹

POSTPRODUCCION

PRODUCCION						
CODIGO	FECHA	CANTIDAD	RECETA	COSTO	ESTADO	USUARIO
144	02/09/2010	1.000	ARROZ MIXTO	3.80	REGISTRADO	JULIO CESAR
84	17/08/2010	3.000	ARROZ MIXTO	11.40	REGISTRADO	JULIO CESAR
9	28/07/2010	6.000	ARROZ MIXTO	22.80	REGISTRADO	JULIO CESAR

Retomar

Figura 9.59. Interfaz: Postproducciones ⁴⁶⁰

⁴⁵⁹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁶⁰ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

2.3.4. MOVIMIENTOS DE BODEGA

Al ingresar a movimientos de bodega se despliega la pantalla de la Figura 9.60., en donde se debe elegir el tipo de movimiento: ingreso, egreso o transferencia, luego se debe ingresar la razón por la que se realiza el movimiento y elegir la bodega origen (en caso de ser una transferencia se elige bodega origen y bodega destino). Dar clic en la opción **Nuevo**, luego **Consultar** el producto, **Seleccionar** el producto de una lista e ingresar la cantidad por la que se realiza el movimiento.

Una vez ingresados los datos elegir **Mov. Producto**, y se van desplegando los movimientos de bodega realizados, cuando finalice el ingreso dar clic en **Grabar**. También se puede acceder a las opciones: limpiar, imprimir, cancelar y salir.

Cuando se elige la opción **Imprimir** se despliega la pantalla de la Figura 9.61., en donde se elige el movimiento de bodega que se quiere imprimir.

CODIGO	DESCRIPCION	CANTIDAD	COSTO	TOTAL
		0.000	0.00000	0.00000

Figura 9.60. Interfaz: Movimiento de Bodega ⁴⁶¹

⁴⁶¹ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

MOVIMIENTOS DE BODEGA						
DETALLE DE MOVIMIENTOS						
CODIGO *	FECHA *	BODEGA *	TIPO *	ESTADO *	USUARIO *	
42	02/09/2010	BODEGA CENTRAL	INGRESO	REGISTRADO	JULIO CESAR	
27	29/07/2010	SUCURSAL	TRANSFERENCIA	REGISTRADO	JULIO CESAR	
26	29/07/2010	BODEGA CENTRAL	INGRESO	REGISTRADO	JULIO CESAR	
25	29/07/2010	SUCURSAL	TRANSFERENCIA	REGISTRADO	JULIO CESAR	
24	29/07/2010	BODEGA CENTRAL	INGRESO	REGISTRADO	JULIO CESAR	
23	29/07/2010	SUCURSAL	TRANSFERENCIA	REGISTRADO	JULIO CESAR	
22	29/07/2010	BODEGA CENTRAL	INGRESO	REGISTRADO	JULIO CESAR	
21	29/07/2010	SUCURSAL	EGRESO	REGISTRADO	JULIO CESAR	
2	29/07/2010	SUCURSAL	INGRESO	REGISTRADO	JULIO CESAR	

Figura 9.61. Interfaz: Movimientos de Bodega ⁴⁶²

3. FACTURACIÓN

En el menú principal se puede acceder a la opción Facturación, aquí se muestra la pantalla de la Figura 9.62.; en este menú se puede ingresar a: Opciones, Datos Generales, Procesos y Reportes.

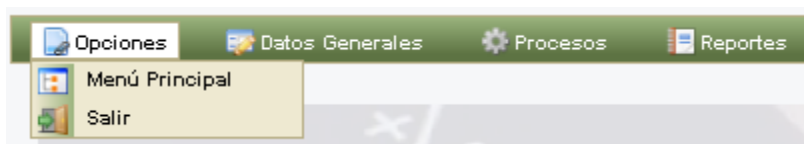


Figura 9.62. Interfaz: Menú de Facturación ⁴⁶³

3.1. OPCIONES

Menú Principal

Al dar clic en la pestaña Opciones – Menú Principal, se regresa al menú principal del sistema (Figura 9.2.).

⁴⁶² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁶³ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Salir

Si se desea salir del sistema se elige la pestaña Opciones – Salir, y se mostrará la pantalla de acceso (Figura 9.1.).

3.2. DATOS GENERALES

En el menú de Inventario se puede acceder a Datos Generales como se muestra en la pantalla de la Figura 9.63., en donde se tiene la opción Clientes.



Figura 9.63. Interfaz: Menú de Facturación ⁴⁶⁴

3.2.1. CLIENTES

Al ingresar a la opción Clientes se despliega la pantalla de la Figura 9.64., en donde se tiene un listado de los clientes y se puede acceder a las opciones: modificar, eliminar, nuevo y salir.

CLIENTE						
CLIENTE						
CODIGO	NOMBRE	APELLIDO	R. U. C.	TELEFONO	TIPO	
2	CRISTINA	VILLACIS	1715460372	2	Precio B	 
21	JULIO CESAR	SANDBALIN GUAMAN	1715460372	2	Precio A	 
22	VANESA	MEJIA	123446	2333	Precio C	 

Figura 9.64. Interfaz: Cliente ⁴⁶⁵


⁴⁶⁴ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.


⁴⁶⁵ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Cuando se ingresa a la opción **Nuevo** se despliega la pantalla de la Figura 9.65., en donde se deben ingresar los datos del cliente: nombre, apellido, ruc, razón social, dirección, teléfono, email y el tipo de precio que va a tener ese cliente.

Una vez ingresados los datos dar clic en **Grabar**. Para borrar los datos de las cajas de texto elegir **Limpiar** y para salir elegir **Retornar**.

Figura 9.65. Interfaz: Ingreso Cliente ⁴⁶⁶

Para modificar los datos del cliente elegir el ícono:  en la pantalla de la Figura 9.64.

Para borrar los datos de un cliente elegir el ícono:  en la pantalla de la Figura 9.64.

3.3. PROCESOS

En la opción Procesos del menú de Facturación se puede ingresar a las opciones: factura, anular facturas, movimientos de caja y cuadro de caja.






Figura 9.66. Interfaz: Procesos de Facturación ⁴⁶⁷

⁴⁶⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

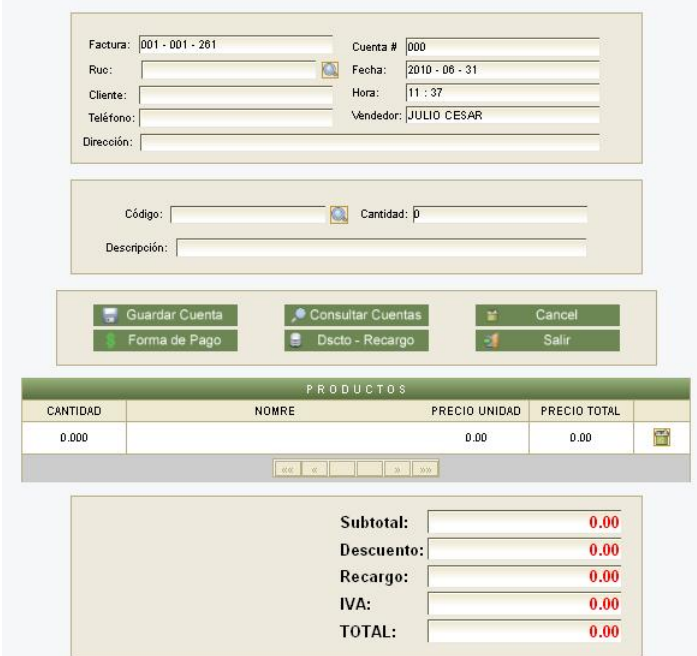
⁴⁶⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

3.3.1. FACTURA

Al ingresar a la opción Factura se despliega la pantalla de la Figura 9.67., en donde se debe dar clic en el ícono:  para consultar los datos del cliente, luego se despliega la pantalla de la Figura 9.68., y se debe **Seleccionar** el cliente; los datos del cliente se despliegan en las cajas de texto.

Se debe **Buscar el Producto** para esto dar clic en el ícono:  y se despliega la pantalla de la Figura 9.69. en donde se debe **Seleccionar** el producto. Luego ingresar la cantidad que se va a facturar y presionar la tecla **Tab**, de esta manera se van agregando los productos en el detalle de la factura. Para eliminar algún producto del detalle de la factura elegir el ícono: .

Para almacenar la cuenta se debe ingresar el número de cuenta y elegir **Guardar cuenta**. Si se desea recuperar una cuenta en espera elegir la opción **Consultar cuentas** y se desplegará la pantalla de la Figura 9.70., en donde se puede imprimir la **Pre-cuenta** o **Seleccionar** para recuperar los datos de la cuentas.



The screenshot shows a software interface for creating an invoice. It includes several input fields for customer information, a product selection area, a table of products, and a summary section.

Customer Information:

- Factura: 001 - 001 - 261
- Cuenta #: 000
- Ruc: [Empty]
- Fecha: 2010 - 06 - 31
- Cliente: [Empty]
- Hora: 11 : 37
- Teléfono: [Empty]
- Vendedor: JULIO CESAR
- Dirección: [Empty]

Product Selection:

- Código: [Empty]
- Cantidad: 0
- Descripción: [Empty]

Buttons:

- Guardar Cuenta
- Consultar Cuentas
- Cancel
- Forma de Pago
- Dcto - Recargo
- Salir

PRODUCTOS Table:

CANTIDAD	NOMBRE	PRECIO UNIDAD	PRECIO TOTAL
0.000		0.00	0.00

Summary:

- Subtotal: 0.00
- Descuento: 0.00
- Recargo: 0.00
- IVA: 0.00
- TOTAL: 0.00

Figura 9.67. Interfaz: Factura ⁴⁶⁸

⁴⁶⁸ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

CLIENTES

CLIENTES							
CODIGO *	RUC *	NOMBRE *	APELLIDO *	COMERCIAL *	TELEFONO *	TIPO *	
2	1715460372	CRISTINA	VILLACIS	2	2	Precio B	 Seleccionar
21	1715460372	JULIO CESAR	SANDBALIN GUAMAN	EMPRESA	2	Precio A	 Seleccionar
22	123456	VANESA	MEJIA	VANE	2333	Precio C	 Seleccionar



Figura 9.68. Interfaz: Clientes ⁴⁶⁹

PRODUCTOS

PRODUCTOS							
CODIGO *	DESCRIPCION *	IMPUESTO *	PRECIO A	PRECIO B	PRECIO C	STOCK	
11	COCA 1/2 LITR	IVA 12%	1.00	0.00	0.00	54.000	 Seleccionar
12	JUGO SUNNY	IVA 12%	0.60	0.00	0.00	37.000	 Seleccionar
13	CERVEZA	IVA 12%	1.10	0.00	0.00	56.000	 Seleccionar
21	ARROZ CON CARNE	IVA 12%	3.50	0.00	0.00	109.000	 Seleccionar
22	ARROZ MIXTO	IVA 12%	4.50	0.00	0.00	6.000	 Seleccionar



Figura 9.69. Interfaz: Consulta Productos ⁴⁷⁰

CUENTAS PENDIENTES

CUENTAS EN ESPERA			
CUENTA *	VENDEDOR *	TOTAL	
002	JULIO CESAR	0.00	  Seleccionar Pre Cuenta
009	JULIO CESAR	29.00	  Seleccionar Pre Cuenta
001	JULIO CESAR	17.50	  Seleccionar Pre Cuenta



Figura 9.70. Interfaz: Cuentas pendientes ⁴⁷¹

⁴⁶⁹ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁷⁰ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁷¹ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

Para facturar la cuenta se debe elegir la opción **Forma de pago** en la pantalla de la Figura 9.67., de ahí se despliega la pantalla de Figura 9.71., en donde se ingresa el monto que cancela el cliente sea en cheque, efectivo o tarjeta.

Se debe almacenar la información para esto dar clic en **Grabar**, y también se puede acceder a las opciones: imprimir y salir.

Si no se desea facturar elegir la opción **Cancel** en la pantalla de la Figura 9.65.

FORMAS DE PAGO

Total a Pagar:	<input type="text" value="13.50"/>	13.50
Saldo:	<input type="text" value="13.50"/>	13.50

FORMAS DE PAGO	
DETALLE ▲	CANTIDAD
CHEQUE	<input type="text" value="0.00"/>
EFECTIVO	<input type="text" value="0.00"/>
TARJETA	<input type="text" value="0.00"/>

Total:	<input type="text" value="0.00"/>	0.00
Cambio:	<input type="text" value="0.00"/>	0.00

Figura 9.71. Interfaz: Formas de pago ⁴⁷²

Para ingresar algún descuento o recargo en la factura se debe elegir la opción **Dcto – Recargo** en la pantalla de la Figura 9.67. (antes de facturar), y se despliega la pantalla de la Figura 9.72., en donde se debe **Seleccionar** el recargo o descuento, o sino dar clic en la opción **Salir**.

⁴⁷² ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

DESCUENTOS Y RECARGO

DESCUENTOS Y RECARGOS				
CODIGO	DESCRIPCION	PORCENTAJE	TIPO	
02	RECARGO	10.00	Recargo	✔ Seleccionar
01	DESCUENTO	5.00	Descuento	✔ Seleccionar

<< < > >>

Salir

Figura 9.72. Interfaz: Seleccionar Descuentos y recargos⁴⁷³

3.3.2. ANULAR FACTURAS

Cuando se ingresa a la opción Anular Facturas se despliega la pantalla de la Figura 9.73., en donde se debe elegir la cuenta a anular y dar clic en **Anular Factura**.

Si no se desea anular ninguna factura elegir la opción **Salir**.

ANULAR FACTURAS

No. FACTURA *	RUC *	NOMBRE *	APELLIDO *	TOTAL
001 - 001 - 381	1715460372	JULIO CESAR	SANDBALIN GUAMAN	17.50

CANTIDAD	DESCRIPCION	PRECIO UNITARI	PRECIO TOTAL
5.00	ARROZ CON CARNE	3.50	17.50

Anular Factura
Salir

Figura 9.73. Interfaz: Anular Facturas⁴⁷⁴

⁴⁷³ ERAZO Inés; SANDBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

3.3.3. MOVIMIENTOS DE CAJA

En la opción Movimientos de Caja se muestra la pantalla de la Figura 9.74., en donde se debe elegir el tipo de movimiento: ingreso o egreso, e ingresar el valor del movimiento, luego elegir la opción **Grabar**.

Para borrar el valor elegir **Limpiar** y para retornar al menú de Facturación elegir **Salir**.

MOVIMIENTOS DE CAJA

No. Documento:	<input type="text" value="41"/>
Tipo:	<input type="text" value="INGRESO CAJA"/>
Valor:	<input type="text" value="0.00"/>

Figura 9.74. Interfaz: Movimientos de Caja ⁴⁷⁵

3.3.4. CUADRE DE CAJA

En la opción Cuadre de Caja se despliega la pantalla de la Figura 9.75., en donde se deben ingresar los valores de cheque, efectivo o tarjeta que se haya recibido durante el día de venta, luego seleccionar **Grabar**. Para regresar al menú dar clic en **Salir**.

CUADRE DE CAJA

FORMAS DE PAGO	
DETALLE ▲	CANTIDAD
CHEQUE	<input type="text" value="0.00"/>
EFECTIVO	<input type="text" value="0.00"/>
TARJETA	<input type="text" value="0.00"/>

Figura 9.75. Interfaz: Cuadre de Caja ⁴⁷⁶

⁴⁷⁴ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁷⁵ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

4. REPORTEES

Finalmente se puede acceder a la opción **Reportes** en cada uno de los módulos del sistema, como por ejemplo el reporte de clientes que se muestra en la pantalla de la Figura 9.76., en donde se consulta los datos y se debe dar clic en **Imprimir** para desplegar la información en formato .pdf. Para retornar al menú dar clic en **Salir**.

REPORTE CLIENTES

Ordenar por: Código Nombre

Cód. Cliente: 

Cód. Cliente: 

 Imprimir  Salir

Figura 9.76. Interfaz: Reporte ⁴⁷⁷

⁴⁷⁶ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

⁴⁷⁷ ERAZO Inés; SANDOBALÍN Julio: Análisis y Desarrollo del Sistema Informático para Administración y Facturación de un restaurante; 2010.

**UNIVERSIDAD POLITÉCNICA
SALESIANA**

FACULTAD DE INGENIERÍAS

SEDE QUITO – CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN INFORMÁTICA PARA LA GESTIÓN

**ANÁLISIS Y DESARROLLO DEL SISTEMA INFORMÁTICO PARA
ADMINISTRACIÓN Y FACTURACIÓN DE UN RESTAURANTE**

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

INÉS ALEXANDRA ERAZO LÓPEZ, JULIO CÉSAR SANDOBALÍN GUAMÁN

DIRECTOR: ING. RENÉ ARÉVALO

Quito, Noviembre 2010

DECLARACIÓN

Nosotros, Inés Alexandra Erazo López y Julio César Sandobalín Guamán, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Inés Alexandra Erazo López

Julio César Sandobalín Guamán

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Inés Alexandra Erazo López y Julio César Sandobalín Guamán bajo mi dirección.

Ing. René Arévalo

Director de Tesis

AGRADECIMIENTO

El presente trabajo se concluyó con la ayuda de mi compañero Julio Cesar Sandobalín y por la guía del Ing. René Arévalo, gracias por su apoyo.

Además agradezco a mis amigos de la universidad por haber demostrado un gran espíritu de unidad y compañerismo durante este largo camino.

Inés Alexandra Erazo López

AGRADECIMIENTO

Este proyecto es el resultado del esfuerzo conjunto de todos los que formamos el grupo de trabajo. Por esto agradezco a nuestro director de tesis, Ing. René Arévalo, a mi compañera Inés Erazo y a todos aquellos que participaron en la realización de esta tesis.

Julio Cesar Sandobalín Guamán

DEDICATORIA

El presente trabajo se lo dedico de manera muy especial a Dios por brindarme sabiduría para poder concluir este proyecto.

A mis padres quienes han sido un pilar fundamental en todo este trayecto y que aunque estén lejos siempre he recibido su apoyo y su amor incondicional.

A mi esposo Byron por estar siempre a mi lado, por ser mi fortaleza y por darme todo su amor.

A mi hermana Marilyn por ser mi guía, gracias por estar conmigo en las buenas y malas, te quiero mucho.

A mi hermano Gabriel por ser mi consejero y por apoyarme en todo momento.

Y finalmente dedico este proyecto al gran amor de mi vida, mi hija Camila.

Inés Alexandra Erazo López

DEDICATORIA

Dedico este proyecto de tesis a Dios y a mi madre. A Dios porque me da la sabiduría y el entendimiento en cada paso que doy.

A mi madre por ser mi inspiración de lucha y tenacidad para conseguir mis sueños, con mucho amor y cariño le dedico todo mi esfuerzo y trabajo puesto para la realización de esta tesis.

Julio Cesar Sandobalín Guamán