

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA EN SISTEMAS**

**Tesis previa a la obtención del título de: INGENIERO EN SISTEMAS**

**TEMA:**

**“ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA FINANCIERO  
PARA ADMINISTRAR RECURSOS Y SIMULAR PROCESOS  
INDUSTRIALES DE PUNTO CLASSIC TEXTILERA NACIONAL.”**

**AUTORES:**

**CHRISTIAN EDUARDO REYES ARGÜELLO  
DANIEL ADOLFO ROCHA RODRÍGUEZ**

**DIRECTOR:**

**DANIEL GIOVANNY DÍAZ ORTIZ**

**Quito, septiembre de 2013**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO  
DEL TRABAJO DE GRADO**

Nosotros Christian Eduardo Reyes Argüello y Daniel Adolfo Rocha Rodríguez autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

---

Christian Eduardo Reyes Argüello  
CC: 1720884723

---

Daniel Adolfo Rocha Rodríguez  
CC: 1722220520

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1 FUNDAMENTOS TEÓRICOS</b> .....	2
1.1 Planteamiento del problema.....	2
1.2 Objetivos .....	3
a. Objetivo general:.....	3
b. Objetivos específicos: .....	3
1.3 Justificación del proyecto.....	4
1.4 Alcance del proyecto.....	4
1.4.1 Módulo financiero.....	5
1.4.2 Módulo de simulación procesos industriales .....	6
1.5 Herramientas .....	7
1.5.1 Visual studio (c#).....	7
1.5.2 Java .....	9
1.5.3 Php .....	11
1.5.4 Cuadro comparativo de lenguajes de programación .....	13
1.5.5 Software de uso en sistemas scada.....	14
1.5.5.1 Lookout.....	14
1.5.5.2 Intouch .....	16
1.5.6 Cuadro comparativo de las características del software scada.....	17
1.5.7 Software a utilizar .....	18
1.6 Especificaciones de requisitos de software (ieee 830).....	18
1.6.1 Introducción .....	18
1.6.2 Propósito .....	18
1.6.3 Ámbito del sistema.....	18
1.6.4 Visión general .....	19
1.6.5 Descripción general.....	19

1.6.6	Requisitos específicos .....	21
1.7	Marco teórico .....	22
1.7.1	Intouch .....	22
1.7.2	Web service wcf.....	23
 <b>CAPÍTULO 2 ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>		<b>26</b>
2.1.	Análisis y documentación del modelo de negocios a implementarse .....	26
2.1.1	Clientes .....	26
2.1.2	Proveedores:.....	26
2.1.3	Registro de cotizaciones.....	27
2.1.4	Orden de compra.....	27
2.1.5	Administrador de bodegas .....	28
2.1.6	Administrador de pagos .....	28
2.1.7	Administrador de usuarios .....	28
2.1.8	Administrador de inventarios.....	29
2.1.9	Generación de índices financieros .....	29
2.1.10	Administrador de permisos .....	29
2.2	Análisis y diseño del control de procesos industriales.....	30
2.2.1	Previo al tejido .....	30
2.2.2	Tejido .....	31
2.2.3	Teñido .....	33
2.2.4	Planchado.....	33
2.3	Análisis y diseño de la base de datos .....	34
2.3.1	Clientes .....	34
2.3.2	Proveedores.....	36
2.3.3	Cotización .....	37
2.3.4	Inventarios.....	38

2.3.5	Facturación.....	40
2.4	Análisis y diseño del sistema financiero .....	41
2.4.1.	Proveedores.....	42
2.4.2	Cotización .....	42
2.4.3	Inventarios.....	43
2.4.4	Facturación.....	44
2.4.5	Cuentas por cobrar .....	44
2.4.6	Cuentas por pagar .....	45
2.5	Análisis y diseño de módulo de simulación .....	45
2.6	Índices financieros .....	47
<b>CAPÍTULO 3 DESARROLLO DEL SISTEMA .....</b>		<b>48</b>
3.1.	Introduccion .....	48
3.2.	Módulo de seguridad.....	48
3.2.1.	Proceso de ingreso al sistema.....	48
3.2.2.	Proceso de creación de usuario .....	54
3.3.	Módulo de clientes .....	59
3.3.1.	Proceso de ingreso de clientes .....	59
3.4.	Módulo de proveedores.....	70
3.4.1.	Proceso de ingreso de proveedores .....	70
3.5.	Módulo de inventarios .....	73
3.5.1.	Proceso de ingreso de productos .....	73
3.5.2.	Proceso de transferencia de productos .....	76
3.5.3.	Proceso de baja de productos .....	79
3.6.	Módulo de Facturacion .....	82
3.6.1.	Proceso de compras.....	82
3.7.	Desarrollo de la base de datos.....	88

3.7.1.	Tablas .....	88
3.7.1.1.	Descripción de las tablas.....	90
3.7.2.	Store procedures.....	93
3.7.2.1.	Descripción de los store procedures.....	95
3.8.	Desarrollo del módulo financiero .....	97
3.8.1.	Arquitectura del sistema.....	97
3.8.1.1.	Modelo de datos .....	98
3.8.1.2.	Acceso a datos.....	104
3.8.1.3.	Capa de negocio .....	106
3.8.1.4.	Publicación de los servicios .....	111
3.9.	Desarrollo del módulo de simulación industrial .....	116
3.9.1.	Inicio .....	117
3.9.2.	Secuencia en la elaboración de tela.....	117
3.10.	Desarrollo de la interfaz de comunicación entre los módulos desarrollados ...	129
 <b>CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBAS.....</b>		<b>132</b>
4.1.	Introducción .....	132
4.2.	Máquinas Virtuales .....	133
4.2.1.	Creación de las máquinas virtuales .....	133
4.3.	Implementación de los módulos, y unificación de los mismos .....	138
4.3.1.	Instalación de los servicios web (win7) .....	138
4.3.2.	Instalación de la aplicación web (win7).....	142
4.3.3.	Instalación del servicio windows (win7).....	148
4.3.4.	Instalación de la base de datos (win7).....	151
4.3.5.	Instalación del simulador (winxp).....	153
4.4.	Pruebas y resultados.....	154
 <b>CONCLUSIONES .....</b>		<b>157</b>

RECOMENDACIONES .....	159
LISTA DE REFERENCIAS .....	161
GLOSARIO .....	164
ANEXOS .....	<b>¡Error! Marcador no definido.</b>

## ÍNDICE DE FIGURAS

Figura	1 . Perspectiva del Producto	19
Figura	2 . Esquema de un Web Service	24
Figura	3 . Rollo de hilo	30
Figura	4 . Rollos de Hijo colocados en los carretes de la Tejedora	31
Figura	5 . Tejedora 1	31
Figura	6 . Tejedora 2	32
Figura	7 . Eje de agujas de la Tejedora	32
Figura	8 . Panel de Controles básicos de la Tejedora	32
Figura	9 . Carretes de hijo de la Tejedora	33
Figura	10 . Maquinaria de Teñido de Tela	33
Figura	11 . Maquinaria de Planchado de Tela	34
Figura	12 . Bodegas de Tela Planchada	34
Figura	13 . Diseño Conceptual de Clientes	35
Figura	14 . Diseño Modelo Lógico de Clientes	35
Figura	15 . Diseño Modelo Físico de Clientes	35
Figura	16 . Diseño Modelo Lógico de Proveedores	36
Figura	17 . Diseño Conceptual de Proveedores	36
Figura	18 . Diseño Modelo Físico de Proveedores	37
Figura	19 . Diseño Lógico de Cotización	37
Figura	20 . Diseño Modelo Conceptual de Cotización	38
Figura	21 . Diseño Modelo Físico de Cotización	38
Figura	22 . Diseño Lógico de Inventarios	39
Figura	23 . Diseño Conceptual de Inventarios	39

Figura	24 . Diseño Modelo Físico de Inventarios	40
Figura	25 . Diseño Lógico de Facturación	40
Figura	26 . Diseño Modelo Conceptual de Facturación	41
Figura	27 . Diseño Físico de Facturación	41
Figura	28 . Proveedores	42
Figura	29 . Cotización	42
Figura	30 . Inventarios	43
Figura	31 . Facturación	44
Figura	32 . Cuentas por Cobrar	44
Figura	33 . Cuentas por Pagar	45
Figura	34 . Secuencia de procesos en la Textilera	46
Figura	35 . Diagrama de Simulador	46
Figura	36 . Proceso de ingreso al sistema	49
Figura	37 . Pantalla de Login	50
Figura	38 . Servicio Login	50
Figura	39 . Método ConsultarExistenciaUsuario	51
Figura	40 . Clase BllLogin	51
Figura	41 . Tabla logín	52
Figura	42 . Método ObtenerIdPorLogin	52
Figura	43 . Método ObtenerMóduloPorId	53
Figura	44 . Método ObtenerSucursalPorUsuario	54
Figura	45 . Proceso de ingreso al sistema	55
Figura	46 . Pantalla de creacion de usuario	56
Figura	47 . Pantalla de permisos	56
Figura	48 . Pantalla de accesos	57

Figura	49 . Método AdministrarIngresoUsuario	58
Figura	50 . Método AsignarAcceso	59
Figura	51 . Proceso de ingreso de clientes al sistema	60
Figura	52 . Pantalla de clientes	61
Figura	53 . Pantalla de contactos	61
Figura	54 . Servicio de clientes	62
Figura	55 . Método ConsultarCliente	62
Figura	56 . Método ConsultarCiudad	63
Figura	57 . Método ConsultarPais	64
Figura	58 . Método ConsultarMoneda	65
Figura	59 . Método ConsultarTipoCliente	66
Figura	60 . Método ConsultarEstadoCliente	67
Figura	61 . Método AdministrarCliente	68
Figura	62 . Método AdministrarContacto	69
Figura	63 . Método AdministrarGrupo	70
Figura	64 . Proceso de ingreso de proveedores al sistema	71
Figura	65 . Pantalla de clientes	72
Figura	66 . Servicio de proveedores	72
Figura	67 . Método AdministrarProvedor	73
Figura	68 . Proceso de ingreso de productos al sistema	74
Figura	69 . Pantalla de ingreso de productos	74
Figura	70 . Pantalla de agregar productos	75
Figura	71 . Servicio de inventarios	75
Figura	72 . Método AdministrarIngresoProducto	76
Figura	73 . Proceso transferencia de productos	77

Figura	74 . Pantalla de ingreso de solicitud	78
Figura	75 . Pantalla de aprobacion de solicitud	78
Figura	76 . Método IngresoSolicitud	79
Figura	77 . Proceso de baja de productos	80
Figura	78 . Pantalla de baja de producto	81
Figura	79 . Método IngresoSolicitud	81
Figura	80 . Proceso de compra de productos	83
Figura	81 . Pantalla de solicitud de compra	84
Figura	82 . Pantalla de cotizacion	84
Figura	83 . Pantalla de compra	85
Figura	84 . Servicio de facturacion	85
Figura	85 . Servicio de cotizacion	86
Figura	86 . Método AdministrarIngresoSolicitudCompra	86
Figura	87 . Método AdministrarIngresoCotizacion	87
Figura	88 . Método AdministrarIngresoCompra	87
Figura	89 . Tabla Creada Datos de Contacto	97
Figura	90 . Tabla Ingreso de Registros	97
Figura	91 . Proyecto WCF's Punto Classic	98
Figura	92 . Modelo de datos "Seguridad"	99
Figura	93 . Modelo de datos "Clientes"	100
Figura	94 . Modelo de datos "Proveedores"	101
Figura	95 . Modelo de datos "Inventarios"	102
Figura	96 . Modelo de datos "Cotizacion"	103
Figura	97 . Modelo de datos "Facturacion"	103
Figura	98 . Modelo de datos "Cuentas"	104

Figura	99 . Imagen de conexión a la BDD	105
Figura	100 . Proyecto Capa de Negocio	106
Figura	101 . Clase BllCliente	107
Figura	102 . Clase BllCotización	107
Figura	103 . Clase BllProveedor	107
Figura	104 . Clase BllCuentas	108
Figura	105 . Clase BllLogin	108
Figura	106 . Clase BllUtilidad	108
Figura	107 . Clase BllCuentas	109
Figura	108 . Clase BllInventario	110
Figura	109 . Carpeta creada del WCF	111
Figura	110 . Visualización de Código WSDL	112
Figura	111 . Selección de Pool de Aplicación	112
Figura	112 . Explorador de IIS	113
Figura	113 . Publicar WCF	113
Figura	114 . Publicación de WCF	114
Figura	115 . Selección de Aplicación a ser Publicada	114
Figura	116 . Selección de Path	115
Figura	117 . Visualización de la publicación del WCF en consola de Visual	115
Figura	118 . Servicio de Cotización	116
Figura	119 . Version Intouch, Licencia	116
Figura	120 . Pantalla de Inicio del Simulador	117
Figura	121 . Consola de programación del Script de Aplicación	118
Figura	122 . Procesos en la Simulación	124
Figura	123 . Preparación del Tinte	124

Figura	124 . Porcentaje de Concentración del Ligante	125
Figura	125 . Indicadores en la generación de Tinte	126
Figura	126 . Histograma de la preparación de Tinte	126
Figura	127 . Flujo de procesos en fabricación de Tinte	127
Figura	128 . Terminación de tinturado de tela e inicio de planchado	127
Figura	129 . Finalización en la Produccion de Rollos de tela	128
Figura	130 . Archivo Excel para sincronización de Inventarios	128
Figura	131 . Estructura de la solución del Servicio Excel	129
Figura	132 . Referencia a la dll Microsoft Excel	130
Figura	133 . Diagrama de Máquinas Virtuales	132
Figura	134 . Pantalla VWWare WorkStation 8	133
Figura	135 . Creación de Nueva Máquina Virtual (paso 1)	134
Figura	136 . Creación de Nueva Máquina Virtual (paso 2)	134
Figura	137 . Creación de Nueva Máquina Virtual (paso 3)	135
Figura	138 . Tamaño de memoria de la Máquina Virtual	136
Figura	139 . Creación de Disco Duro de la Máquina Virtual	136
Figura	140 . Tamaño de Disco Duro de la Máquina Virtual	137
Figura	141 . Nombre del Disco Duro de la Máquina Virtual	137
Figura	142 . Inicio de Máquina Virtual	138
Figura	143 . Instalación de WCF's de Punto Classic (paso 1)	139
Figura	144 . Instalación de WCF's de Punto Classic (paso 2)	140
Figura	145 . Instalación de WCF's de Punto Classic (paso 3)	140
Figura	146 . Instalación de WCF's de Punto Classic (paso 4)	141
Figura	147 . Instalación de WCF's de Punto Classic (paso 5)	141
Figura	148 . Instalación de WCF's de Punto Classic (paso 6)	142

Figura	149 . Acceso a consola IIS	142
Figura	150 . Sitio Web Nuevo para Punto Classic	143
Figura	151 . Nombre del Sitio Web de Punto Classic	143
Figura	152 . Ruta del Sitio Web de Punto Classic	144
Figura	153 . Configuración Conectar Como	145
Figura	154 . Configuración Protocolo del Sitio Web de Punto Classic	145
Figura	155 . Configuración dirección IP del Sitio Web de Punto Classic	146
Figura	156 . Configuración del puerto para el Sitio Web de Punto Classic	146
Figura	157 . Configuración Nombre del Host del Sitio Web de Punto Classic	147
Figura	158 . Configuración inicio por Defecto del Sitio Web de Punto Classic	147
Figura	159 . Instalación del Servicio Windows (paso 1)	148
Figura	160 . Instalación del Servicio Windows (paso 2)	149
Figura	161 . Instalación del Servicio Windows (paso 3)	149
Figura	162 . Instalación del Servicio Windows (paso 4)	150
Figura	163 . Instalación del Servicio Windows (paso 5)	150
Figura	164 . Instalación del Servicio Windows (paso 6)	151
Figura	165 . Restauración de la base de datos de Punto Classic (paso 1)	152
Figura	166 . Restauración de la base de datos de Punto Classic (paso 2)	152
Figura	167 . Restauración de la base de datos de Punto Classic (paso 3)	153
Figura	168 . Restauración de la base de datos de Punto Classic (paso 4)	153
Figura	169 . Pruebas de Stress y Calidad (paso 1)	155
Figura	170 . Pruebas de Stress y Calidad (paso 2)	156
Figura	171 . Pruebas de Stress y Calidad (paso 3)	156
Figura	172 . Pantalla de Inicio	170
Figura	173 . Pantalla de Inicio de Sesión	171

Figura	174 . Pantalla de Menú Principal	171
Figura	175 . Pantalla de Ciudad	172
Figura	176 . Pantalla de Pais	173
Figura	177 . Pantalla de Moneda	174
Figura	178 . Pantalla de Estado Cliente	175
Figura	179 . Pantalla de Tipo Cliente	176
Figura	180 . Pantalla de Producto	177
Figura	181 . Pantalla de Tipo Producto	178
Figura	182 . Pantalla de Tipo Producto	179
Figura	183 . Pantalla de Sucursal	180
Figura	184 . Pantalla de Estado Proveedor	181
Figura	185 . Pantalla de Tipo Proveedor	182
Figura	186 . Pantalla de Forma de Pago	183
Figura	187 . Pantalla de Inscripción Factura	184
Figura	188 . Pantalla de Usuario	185
Figura	189 . Pantalla de Permisos	186
Figura	190 . Pantalla de Acceso	187
Figura	191 . Pantalla de Cliente	188
Figura	192 . Pantalla de Agregar Contacto	188
Figura	193 . Pantalla de Proveedores	189
Figura	194 . Pantalla de Agregar Contacto	190
Figura	195 . Pantalla de Ajuste Producto	191
Figura	196 . Pantalla de Ingreso Producto	192
Figura	197 . Pantalla de Solicitud Producto	193
Figura	198 . Pantalla de Agregar Producto	193

Figura	199 . Pantalla de Aprobar Solicitud	194
Figura	200 . Pantalla de Baja Producto	195
Figura	201 . Pantalla de Solicitud Compra	195
Figura	202 . Pantalla de Ingreso Compra	196
Figura	203 . Pantalla de Ingreso Venta	197
Figura	204 . Pantalla de Anulación Factura	197
Figura	205 . Pantalla de Anulación Factura Venta	198
Figura	206 . Pantalla de Cotizacion	198
Figura	207 . Pantalla de Cuentas por Cobrar	199
Figura	208 . Pantalla de Cuentas por Pagar	200
Figura	209 . Instalación de Win7 en la Máquina Virtual 1 (paso 1)	200
Figura	210 . Instalación de Win7 en la Máquina Virtual 1 (paso 2)	201
Figura	211 . Instalación de Win7 en la Máquina Virtual 1 (paso 3)	201
Figura	212 . Instalación de Win7 en la Máquina Virtual 1 (paso 4)	202
Figura	213 . Instalación de Win7 en la Máquina Virtual 1 (paso 5)	202
Figura	214 . Inicio de Win7 en la Máquina Virtual 1 (paso 1)	203
Figura	215 . Inicio de Win7 en la Máquina Virtual 1 (paso 2)	203
Figura	216 . Inicio de Instalación de WinXp en la Máquina Virtual 2 (paso 1)	204
Figura	217 . Inicio de Instalación de WinXp en la Máquina Virtual 2 (paso 2)	204
Figura	218 . Inicio de Instalación de WinXp en la Máquina Virtual 2 (paso 3)	205
Figura	219 . Aceptación de Contrato de WinXp en la Máquina Virtual 2	205
Figura	220 . Selección de Partición de WinXp en la Máquina Virtual 2	206
Figura	221 . Tamaño de partición para la Máquina Virtual 2	206
Figura	222 . Selección de la partición creada para la Máquina Virtual 2	206
Figura	223 . Formateo de la partición creada para la Máquina Virtual 2	207

Figura	224	. Formateando la partición de la Máquina Virtual 2	207
Figura	225	. Copia de archivos necesarios para la instalación de WinXp	208
Figura	226	. Finalización de la copia de archivos necesarios para la instalación	208
Figura	227	. Reinicio del programa de instalación de Win XP	208
Figura	228	. Instalación de WinXp en la Máquina Virtual 2 (paso 1)	209
Figura	229	. Instalación de WinXp en la Máquina Virtual 2 (paso 2)	209
Figura	230	. Instalación de WinXp en la Máquina Virtual 2 (paso 3)	210
Figura	231	. Instalación de WinXp en la Máquina Virtual 2 (paso 4)	210
Figura	232	. Instalación de WinXp en la Máquina Virtual 2 (paso 5)	211
Figura	233	. Instalación de WinXp en la Máquina Virtual 2 (paso 6)	211
Figura	234	. Instalación de WinXp en la Máquina Virtual 2 (paso 7)	212
Figura	235	. Instalación de WinXp en la Máquina Virtual 2 (paso 8)	212
Figura	236	. Instalación de WinXp en la Máquina Virtual 2 (paso 9)	213
Figura	237	. Instalación de WinXp en la Máquina Virtual 2 (paso 10)	213
Figura	238	. Inicio de WinXp en la Máquina Virtual 2 (paso 1)	214
Figura	239	. Inicio de WinXp en la Máquina Virtual 2 (paso 2)	214
Figura	240	. Inicio de WinXp en la Máquina Virtual 2 (paso 3)	215
Figura	241	. Inicio de WinXp en la Máquina Virtual 2 (paso 4)	215
Figura	242	. Instalación de Silverlight	216
Figura	243	. Instalación de Silverlight	216
Figura	244	. Instalación de Silverlight	216
Figura	245	. Instalación de Visual Studio 2010 (paso 1)	217
Figura	246	. Instalación de Visual Studio 2010 (paso 2)	217
Figura	247	. Instalación de Visual Studio 2010 (paso 3)	218
Figura	248	. Instalación de Visual Studio 2010 (paso 4)	218

Figura	249	. Instalación de Visual Studio 2010 (paso 5)	219
Figura	250	. Instalación de Visual Studio 2010 (paso 5)	219
Figura	251	. Instalación de Visual Studio 2012 (paso 1)	220
Figura	252	. Instalación de Visual Studio 2012 (paso 2)	220
Figura	253	. Instalación de Visual Studio 2012 (paso 3)	221
Figura	254	. Instalación de Visual Studio 2012 (paso 4)	221
Figura	255	. Instalación de Visual Studio 2012 (paso 5)	222
Figura	256	. Instalación de Visual Studio 2012 (paso 6)	222
Figura	257	. Instalación de SQL Server 2008 (paso 1)	223
Figura	258	. Instalación de SQL Server 2008 (paso 2)	223
Figura	259	. Instalación de SQL Server 2008 (paso 3)	224
Figura	260	. Instalación de SQL Server 2008 (paso 4)	224
Figura	261	. Instalación de SQL Server 2008 (paso 5)	225
Figura	262	. Instalación de SQL Server 2008 (paso 6)	225
Figura	263	. Instalación de SQL Server 2008 (paso 7)	226
Figura	264	. Instalación de SQL Server 2008 (paso 8)	226
Figura	265	. Instalación de SQL Server 2008 (paso 9)	227
Figura	266	. Instalación de SQL Server 2008 (paso 10)	227
Figura	267	. Instalación de SQL Server 2008 (paso 11)	228
Figura	268	. Instalación de SQL Server 2008 (paso 12)	228
Figura	269	. Instalación de SQL Server 2008 (paso 13)	229
Figura	270	. Instalación de SQL Server 2008 (paso 14)	229
Figura	271	. Instalación de SQL Server 2008 (paso 15)	230
Figura	272	. Instalación de SQL Server 2008 (paso 16)	230
Figura	273	. Instalación de SQL Server 2008 (paso 17)	231

Figura	274 . Instalación de SQL Server 2008 (paso 18)	231
Figura	275 . Instalación de SQL Server 2008 (paso 19)	232
Figura	276 . Instalación de SQL Server 2008 (paso 20)	232
Figura	277 . Instalación de SQL Server 2008 (paso 21)	233
Figura	278 . Instalación de SQL Server 2008 (paso 22)	233
Figura	279 . Instalación de Intouch (paso 1)	234
Figura	280 . Instalación de Intouch (paso 2)	234
Figura	281 . Instalación de Intouch (paso 3)	235
Figura	282 . Instalación de Intouch (paso 4)	235
Figura	283 . Instalación de Intouch (paso 5)	236
Figura	284 . Instalación de Intouch (paso 6)	236
Figura	285 . Instalación de Intouch (paso 7)	237

## ÍNDICE DE TABLAS

Tabla	1 .	Proceso de Producción	6
Tabla	2 .	Lenguajes de Programación	13
Tabla	3 .	Software SCADA	17
Tabla	4 .	Software a utilizar	18
Tabla	5 .	Requisitos Específicos	21
Tabla	6 .	Cuadro de Requisitos	30
Tabla	7 .	Tablas que componen la Base de Datos	89
Tabla	8 .	Store Procedures Creados para interactuar con el Sistema	94

## **RESUMEN**

La tesis describe el análisis, desarrollo e implementación de un sistema financiero en línea para la simulación de procesos industriales y administración de recursos de Punto Classic Textilera Nacional.

La necesidad de presentar un mejor y más eficiente servicio obligó a Punto Classic a mirar alternativas dentro de la evolución tecnológica, en este caso se ha optado por el desarrollo de un sitio web basado en Silverlight, que es la cara de presentación del sistema. También ofrece internamente una arquitectura orientada a servicios que brinda gran flexibilidad en el sistema para futuras funcionalidades que se necesiten, y a su vez, estos interactúan con la base de datos de Punto Classic, ejecutando funciones tales como: creación de usuarios, ingreso de insumo, creación de producto, parametrización de permisos al sistema, etc.

Además cuenta con un simulador desarrollado en Intouch que emula la producción diaria de Punto Classic; interactuando en tiempo real con la base de datos del módulo financiero, esto gracias a un servicio Windows que interconecta intouch con la arquitectura orientada a servicios que se utiliza en el sitio web.

Un mundo de posibilidades se abre sobre la integración de sistemas web (.Net – SilverLight – Wcf) y aplicaciones Windows (intouch - control industrial), para abrir nuestras mentes a la evolución de nuevas y mayores formas de solución a problemas vigentes en la mayoría de las empresas o industrias.

## **ABSTRACT**

The thesis describes the analysis, development and implementation of an online financial system for the simulation of industrial processes and resource management of National Textile Classic Point.

The need to present a better and more efficient service forced to look at alternatives Punto Classic in technological evolution, in this case it was decided to develop a website based on Silverlight, which is the system display face. It also features an internal service-oriented architecture that provides great flexibility in the system for future functionalities that are needed, and in turn, they interact with the database of Punto Classic, performing functions such as creating users, entering input, product creation, parameterization of the system permissions, etc..

It also has developed a simulator that emulates the production Intouch Daily Point Classic, interacting in real time with the database of financial module, this thanks to a Windows service that interconnects intouch with service-oriented architecture that is used in the website.

A world of possibilities opens on web systems integration (. Net - SilverLight - Wcf) and Windows applications (intouch - industrial control), to open our minds to the evolution of new and old ways of solving existing problems in most companies or industries.

## INTRODUCCIÓN

El mundo de la tecnología varía constantemente, y los medios de Información y transmisión de datos evolucionan a diario innovando las distintas formas de operar o trabajar, ofreciendo grandes ventajas a las organizaciones e industrias en cuanto a fiabilidad, operatividad y seguridad de los datos; por consiguiente las organizaciones han optado por posturas emprendedoras, confiando en el mundo tecnológico para las operaciones de sus empresas.

En Punto Classic textilera nacional, no es la excepción, ya que, al notar que el manejo de información computarizadamente eleva los estándares en la entrega de un servicio o producto al cliente, se ha optado por profundizar en el análisis y desarrollo de un sistema financiero capaz de administrar recursos que se sincronizan en tiempo real con un simulador de la producción diaria en la textilera, manteniendo un control más preciso de insumos, productos, sucursales, cuentas por cobrar y cuentas por pagar.

Con la implementación de un sistema web desarrollado en un entorno amigable para el usuario como lo es Silverlight, y usando una arquitectura orientada a servicios se ha logrado descubrir una solución muy loable para mejorar problemas como latencia en el servicio, mala documentación de productos, extravíos de información archivada, etc en Punto Classic.

Para lo que se ha realizado el siguiente trabajo de investigación formado por (4) capítulos que rezan lo siguiente:

Capítulo 1: planteamiento del problema y fundamentación teórica necesaria para el análisis y solución del mismo.

Capítulo 2: análisis y diseño de cada uno de los módulos a ser implementados.

Capítulo 3: desarrollo de los módulos que conforman el sistema.

Capítulo 4: instalación de los módulos en el ambiente en que fue a ser utilizado, y pruebas de calidad en el mismo.

# CAPÍTULO 1

## FUNDAMENTOS TEÓRICOS

### 1.1 Planteamiento del problema

En la actualidad los Sistemas Informáticos son muy importantes ya que, gracias a ellos la mayoría de las instituciones, manejan las bases de datos de una forma más cómoda y sencilla, sin la necesidad de archivar los registros en papeles u otros documentos lo que no proporciona una información rápida de los registros diarios y mensuales que se necesitan en su debido momento.

Con los sistemas informáticos esta tarea resulta más fácil de manejar para los empleados que laboran en dichas instituciones, ya sean públicas o privada, ya que, los documentos de una empresa se consideran como los recursos más importantes de la misma, por este motivo es conveniente optar por una manera más segura y eficiente para acceder a ellos, a través de la tecnología de almacenamiento y tratamiento digital de datos.

Punto Classic Textilera Nacional, no cuenta con el beneficio antes mencionado debido a que en esta industria aún se lleva a cabo el tratamiento de estos recursos de manera física lo que ha ocasionado en algunas circunstancias la pérdida o uso mal intencionado de dichos documentos, motivo por el cual la empresa considera necesario manejar sus recursos o documentos (facturas, órdenes de compra, inventarios, etc.) de manera digital, lo que le permitirá tener un control, visualización y acceso a la información sin pérdidas de tiempo, sin riesgo del contenido tan valioso que este tipo de documentos posee.

Otro gran problema de la Industria Punto Classic Textilera Nacional es el no poseer un control adecuado de la producción que se genera diariamente, ya que los procesos de elaboración son varios dependiendo de la demanda del producto; es necesario tener en cuenta que cuando se elabora un producto se necesita materia prima, la que deberá estar registrada en el sistema financiero y ligada al aumento o disminución de la producción.

Al tener automatizada la información de los procesos de producción, proporcionarían un mejor manejo del nivel de producción, una disminución considerable de tiempo muertos y la confiabilidad de la información obtenida.

Por lo anteriormente expuesto se propone la siguiente solución: realizar el análisis, diseño y desarrollo de un sistema formado por un módulo financiero y la simulación industrial como un análisis previo a la implementación y viabilidad de la automatización industrial en la producción de Punto Classic Textilera Nacional abriendo la posibilidad de que la industria textil nacional vaya creciendo día a día y modernizando los medios de producción actualmente existentes.

## **1.2 Objetivos**

### **a. Objetivo general:**

Analizar, diseñar y desarrollar un Sistema Financiero en línea para administrar recursos y simular procesos industriales para la empresa Punto Classic Textilera Nacional.

### **b. Objetivos específicos:**

- Recopilar los conocimientos necesarios sobre metodologías de desarrollo de Software, tratamiento de BDD, lenguajes de programación, control y automatización electrónica.
- Desarrollar un Módulo Financiero para la empresa Punto Classic Textilera Nacional.
- Desarrollar un Módulo de Simulación Industrial para la empresa Punto Classic Textilera Nacional.
- Unificar los Módulos Financiero y de Simulación Industrial desarrollados.

### **1.3 Justificación del proyecto**

El propósito de este Proyecto de Tesis es el de optimizar tiempo, esfuerzo y recursos, mediante el desarrollo de un Sistema conformado por dos subsistemas, uno financiero conectado a otro de simulación de procesos industriales, y viceversa para brindar así seguridad y eficiencia a través del tratamiento digital, un manejo adecuado de la recolección de datos importantes que se usan y generan en Punto Classic Textilera Nacional y al mismo tiempo generar un control completo del Sistema de Producción.

De lo anterior se puede destacar que por medio del sistema se pueden obtener beneficios, tales como:

- Controlar el nivel de producción.
- Agilidad en el acceso a datos de la industria en el momento oportuno o que fuera necesario para la toma de decisiones.
- Ofrecer un servicio eficaz y eficiente a la hora del cumplimiento o negociaciones de Punto Classic.

Para contribuir a la solución de la problemática se puso énfasis en la automatización de la información que la Industria considera como importante para el giro del negocio permitiendo que los elementos interactúan entre sí con el fin de apoyar las actividades de las instituciones y corregir los problemas de control y administración de datos en procesos industriales de manera eficiente optimizando tiempo, esfuerzos y recursos.

### **1.4 Alcance del proyecto**

El sistema se desarrollará con la finalidad de ser ejecutado en la Web. Constará de:

1. Módulo Financiero
2. Módulo de Simulación Procesos Industriales

### 1.4.1 Módulo financiero

El módulo financiero se encargará de solucionar los siguientes procesos:

**Cientes.-** administración de Clientes.

**Cotizaciones.-** se podrá realizar informes de precios de los productos según el tipo de consumidor, ya sea este consumidor final o distribuidor; además se encargará del registro adecuado de los datos pertinentes a cotizaciones de proveedores de la fábrica.

**Proveedores.-** se llevará el almacenamiento de los datos de cada uno de los proveedores y de los productos que suministran a Punto Classic Textilera Nacional, estará estrechamente relacionado con el proceso de facturación.

**Inventarios.-** se encargará del registro y relación de productos elaborados y productos necesarios, estará estrechamente ligado con el módulo de simulación de producción industrial.

**Cuentas por pagar.-** se encargará de almacenar todas las cuentas pertinentes a pagar mensualmente proporcionando alertas vía correo electrónico de acuerdo a fechas previas a cada uno de los pagos.

**Cuentas por Cobrar.-** se encargará de almacenar todas las cuentas pertinentes a ser cobradas mensualmente proporcionando alertas vía correo electrónico de acuerdo a fechas previas a cada uno de los pagos.

**Facturación.-** se encargará de llevar registros de ventas a un documento físico, estará estrechamente ligado a Inventarios.

**Índices Financieros.-** proporcionará una idea acerca del comportamiento o el desempeño de toda la organización o una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia puede estar señalando una desviación sobre la cual se podrán tomar acciones preventivas o correctivas según el caso.

### 1.4.2 Módulo de simulación procesos industriales

Éste llevará a cabo las funciones de simular de manera virtual la producción de tela y preparación de la misma de acuerdo a la gestión de procesos de Punto Classic Textilera Nacional para un seguimiento constante de la producción.

**Interfaz de Interconexión:** Esta interfaz se encargará de manejar la comunicación entre los Módulos financiero y de simulación industrial.

La simulación industrial, se realizará con una herramienta de desarrollo y control industrial, que será determinada en el transcurso de la investigación.

El simulador se encargará de mostrar gráficamente el funcionamiento de la línea de procesos de Punto Classic Textilera Nacional, dicha línea de procesos se encontrará dividida en bloques dependiendo del proceso y la funcionalidad del mismo.

En cada bloque de proceso se determinará el comportamiento de entradas y salidas por medio de un modelo matemático ya sea este lineal o no lineal, este modelo matemático se determinará en el transcurso de la investigación, y dependiendo de la funcionalidad para la que será implementado.

Este módulo replicará los procesos de producción industrial de Punto Classic Textilera Nacional los cuales son: Proceso de Tejido, Proceso de Teñido y Proceso de Planchado; serán replicados de tal manera que se pueda obtener respuestas inmediatas y sea posible la sincronización con el módulo financiero. Para esto se va a usar herramientas generadoras de aplicaciones SCADA.

El simulador se encargará de:

**Tabla 1. Proceso de Producción**

Proceso de Tejido	Proceso de Teñido	Proceso de Planchado
Velocidad de Tejido	Color	Ancho del planchado de la tela
Tipo de Tejido	Temperatura de secado	Temperatura de planchado

**Elaborado por: Christian Reyes - Daniel Rocha**

## **1.5 Herramientas**

Lenguajes de programación.

### **1.5.1 Visual studio (c#)**

#### **Sencillez**

El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio fuente.

El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina lo que facilita la portabilidad del código. (González, 2007, p.23)

#### **Modernidad**

C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal, la inclusión de una instrucción foreach, la inclusión de un tipo básico string o la distinción de un tipo bool. (González, 2007, p.23)

#### **Orientación a objetos**

C# es un lenguaje orientado a objetos, una diferencia de este enfoque orientado a objetos es que este lenguaje es más puro, en tanto que no admiten ni funciones ni variables globales, sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo. (González, 2007, p.23)

#### **Orientación a componentes**

La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

Es decir, la sintaxis de C# permite definir cómodamente, eventos o atributos. (González, 2007, p.23)

### **Gestión automática de memoria**

Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, dado que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en la fuente. (González, 2007, p.23)

### **Seguridad de tipos**

C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. Sólo se admiten conversiones entre tipos compatibles, es decir, entre tipos para los que explícitamente se haya definido un operador de conversión, y entre un tipo y un tipo hijo suyo del que un objeto del primero almacenase una referencia del segundo (downcasting). (González, 2007, p.23)

### **Extensibilidad de tipos básicos**

C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. (González, 2007, p.23)

### **Versionable**

C# incluye una política de versionado que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y extendidos con miembros de igual nombre a los recién introducidos. (González, 2007, p.23)

### **Eficiente**

En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros., lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes. (González, 2007, p.23)

### **Compatible**

Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros. (González, 2007, p.23)

## **1.5.2 Java**

### **Lenguaje Simple**

Se lo conoce como lenguaje simple porque viene de la misma estructura de c y c++; ya que c++ fue un referente para la creación de java por eso utiliza determinadas características de c++ y se han eliminado otras. (Álvarez M., 1999, p.1)

### **Orientado a Objetos**

Toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras en estructuras encapsuladas es más fácil su manipulación. (Álvarez M., 1999, p.1)

### **Robusto**

Es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. (Álvarez M., 1999, p.1)

### **Seguro**

La seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real. (Álvarez M., 1999, p.1)

### **Indiferente a la arquitectura**

Java es compatible con los más variados entornos de red, cualquiera sean estos desde Windows 95, Unix a Windows y Mac, para poder trabajar con diferentes sistemas operativos. (Álvarez M., 1999, p.1)

### **Portable**

Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de java. (Álvarez M., 1999, p.1)

### **Interpretado y compilado a la vez**

Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina. (Álvarez M., 1999, p.1)

### **Multihebra o Multihilos**

Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multahilos ya que por cada hilo que el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo. (Álvarez M., 1999, p.1)

### **Dinámico**

El lenguaje java es muy dinámico en la fase de enlazado, sus clases solamente actuaran en medida en que sean requeridas o necesitadas con esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas o desde la red. (Álvarez M., 1999, p.1)

### **Alto rendimiento**

Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código. (Álvarez M., 1999, p.1)

### **1.5.3 Php**

#### **Multiplataforma**

PHP funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra. (Chávez, 2006, p.1)

#### **Ejecución en Servidor**

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página PHP. Dado que la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. (Chávez, 2006, p.1)

#### **Licencia de software libre**

PHP es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, ni se limita en su distribución y es posible ampliarlo con nuevas funcionalidades si así se lo desea. (Chávez, 2006, p.1)

#### **Sintaxis cómoda**

PHP cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Es decir, cuando se crea una variable no tenemos que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador, aunque los errores que se cometen pueden ser muchos más graves y difíciles de corregir al reducirse mucho las posibilidades del intérprete para detectar incompatibilidades entre variables. (Chávez, 2006, p.1)

### **Soporta objetos y herencia**

PHP tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores, etc. Además soporta herencia, aunque no múltiple. (Chávez, 2006, p.1)

### **Extensa librería de funciones**

PHP cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores. (Chávez, 2006, p.1)

### **Compatibilidad con bases de datos**

Quizá la característica más fuerte de PHP sea su amplio soporte para una gran cantidad de bases de datos. Tiene acceso un gran número de gestores de bases de datos: Adabas D, dBase, Empress, Ingress, InterBase, FrontBase, DB2, Informix, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase, etc. (Chávez, 2006, p.1)

### **Expansión**

PHP está alcanzando unos niveles de uso tan elevados que hacen que su conocimiento sea algo indispensable para los profesionales del desarrollo en Internet. Se estima que PHP es usado por cientos de miles de programadores y muchos millones de sitios informan que lo tienen instalado, sumando más del 20% de los dominios en Internet. (Chávez, 2006, p.1)

## 1.5.4 Cuadro comparativo de lenguajes de programación

**Tabla 2. Lenguajes de Programación**

	<b>C#</b>	<b>JAVA</b>	<b>PHP</b>
Sencillez.	Es auto contenido y el tipo de dato es fijo e independiente.	Viene de la misma estructura de c y c++; ya que c++ fue un referente para la creación de java	Php cuenta con una sintaxis similar a la de c, c++ o perl, el tipado es muy poco estricto.
Orientación a Objetos.	C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.	Toda la programación en java en su mayoría está orientada a objetos.	Php tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores.
Robusto	C# todo el código incluye numerosas restricciones para garantizar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones.	Es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.	Los errores que se cometen pueden ser muchos más graves y difíciles de corregir al reducirse mucho las posibilidades del intérprete para detectar incompatibilidades entre variables.
Licencia	Es software Propietario.	Es software libre.	Es software libre.
Portabilidad	C# es compatible en Sistemas Operativos Windows.	Java es compatible con los más variados entornos de red, cualquiera sean estos desde Windows 95, Unix a	Php funciona tanto en sistemas unix o linux con servidor web apache como en sistemas windows con microsoft internet information

		Windows y Mac.	server.
Rendimiento	C# manipula objetos a través de punteros., lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.	Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código.	Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

**Fuente: Adrformacion**

Para este proyecto se usará el lenguaje de programación C# de Visual Studio 2010 debido a que permite el desarrollo del proyecto en menor tiempo ya que dispone de gran cantidad de librerías que permiten realizar el trabajo en menor tiempo, mayor soporte, además se trabajará en un servidor Windows Server 2008 y por consiguiente se usará la base de datos SQL SERVER 2008 debido a que tiene mejor adaptación con el lenguaje de programación seleccionado.

### **1.5.5 Software de uso en sistemas scada**

El conjunto de programas que se implementan en uno o varios ordenadores para la supervisión de procesos y control de la producción se denomina software SCADA.

El software SCADA debe tener acceso a lo que ocurre en el sistema de control, mediante mecanismos de comunicación adecuados con controladores industriales, reguladores básicos, autómatas programables o SAD (Sistemas de Adquisición de Datos).

#### **1.5.5.1 Lookout**

Éste paquete HMI-SCADA de National Instruments se caracteriza por su sencillez y fácil aprendizaje, sin que ello le reste potencia y solidez en su misión principal: control y adquisición de datos de una planta.

## **Funciones de Lookout**

Las funciones características que ofrece este software son las contempladas por la mayoría de ellos. En particular, Lookout ofrece:

- Gráficos de tendencia y para SPC (Control estadístico de procesos).
- Gestión de alarmas.
- Seguridad.
- Animación.
- Registro de históricos.
- Consulta a bases de datos mediante SQL (Lenguaje de consulta estructurado).
- Plantillas múltiples de visualización.

## **Arquitectura basada en objetos**

La tecnología que permite a Lookout la facilidad de uso implementada, es su arquitectura basada en objetos. Para crear cualquier aplicación, sin importar su grado de complejidad, simplemente se crearán objetos y se conectarán adecuadamente, sin necesidad de escribir programas o ejecutar compilaciones.

## **Configuración on-line**

El desarrollo de aplicaciones y las modificaciones de estas se pueden hacer totalmente on-line, sin necesidad de parar actividad alguna relacionada con el proceso todo se llevará a cabo sin pérdida de datos y sin esperas por procesos de compilación.

## **Ejecución basada en eventos (sucesos)**

El hecho de que Lookout desarrolle su ejecución basándose en eventos, le hace idóneo para múltiples aplicaciones de control: continuo, discreto, por lotes (batch) y telemetría. Estas aplicaciones plantean unas necesidades al SCADA que requieren de éste altas prestaciones y/o gestión de elevado número de variables o puntos de control.

La programación orientada a eventos o sucesos espera a que se produzca un determinado acontecimiento prefijado en el programa para ejecutar una acción

programada, en lugar de disponer el programa de un bucle cerrado que va chequeando sucesivamente hasta encontrar el suceso que desencadena la acción.

#### **1.5.5.2 Intouch**

##### **Sistema de Alarmas Distribuida**

El sistema distribuido soporta múltiples servidores de alarmas o suministradores de forma concurrente, proporcionando a los operadores la capacidad de visualizar o reconocer simultáneamente información de alarmas desde múltiples ubicaciones remotas.

##### **Historial Distribuido**

El sistema de tendencia histórica distribuida le permite la generación de informes via web, la conectividad open data source, con su enorme velocidad, amplia escalabilidad, almacenamiento y recuperación de datos altamente eficiente.

##### **Conversión de resolución dinámica**

Puede desarrollar aplicaciones con una resolución de pantalla y ejecutarlas en otra sin afectar la aplicación original. Las aplicaciones se ejecutan con una resolución definida por el usuario en lugar de una resolución de visualización.

##### **Dirección de referencia dinámica**

Las referencias de fuente de datos se pueden modificar dinámicamente para direccionar múltiples fuentes de datos con una única etiqueta.

##### **Desarrollo de aplicaciones de red**

Las nuevas características de desarrollo remoto dan cabida a grandes instalaciones multinodo, incluyendo la actualización de todos los nodos de una red desde una única estación de desarrollo.

##### **FactoryFocus**

Permite a los administradores y supervisores visualizar un proceso continuo de aplicación HMI en tiempo real. La seguridad del sistema se incrementa con la

capacidad de solo visualización ya que no puede cambiar ningún dato. No es necesario realizar modificaciones en las aplicaciones para utilizar FactoryFocus.

### **Intouch FactoryFocus**

Esta característica solo funciona como cliente. No se puede escribir ningún dato por medio de DDE, FastDDE o Poked en programas tales como Excel, las alarmas se pueden visualizar pero no reconocer. Las características tales como enlaces de animación, etiquetas, tendencias históricas y en tiempo real son solo de visualización.

### **1.5.6 Cuadro comparativo de las características del software scada**

**Tabla 3. Software SCADA**

	<b>Intouch</b>	<b>Lookout</b>
Configuración Online	Para realizar un cambio es necesario parar las actividades.	Tiene la capacidad de realizar cambios sin necesidad de realizar cambios.
Alarmas Distribuidas	Soporta múltiples servidores de alarmas concurrentes.	Tiene gestión de alarmas pero no soporta múltiples servidores.
Resolución Dinámica	Permite desarrollar aplicaciones en una resolución y ejecutarlas en otra.	No puede desarrollar en aplicaciones en diferentes resoluciones.
Ejecución Basada en eventos.	Espera que se active una determinada acción en el entorno para realizar una tarea programada.	Espera que se active una determinada acción en el entorno para realizar una tarea programada.

**Fuente: Scribd**

Teniendo presentes las características de cada software usado en sistemas SCADA, se ha considerado la elección de Lookout, porque este software realiza simulaciones en tiempo real, ésto en una industria textil es fundamental, debido a que se desea tener una idea clara y veraz de lo que sucede en cada instante de la producción. Otra característica para la elección del software es la ejecución basada en eventos porque facilita y estructura el desarrollo de las aplicaciones realizadas.

### 1.5.7 Software a utilizar

**Tabla 4. Software a utilizar**

<b><u>Módulo Financiero</u></b>	SQL Server
	Visual Studio (C#)
<b><u>Módulo de Simulación Industrial</u></b>	Intouch
	Visual Studio (C#)
<b><u>Web</u></b>	Expression Studio
	Visual Studio (Aspx)

Fuente: Punto Classic

## 1.6 Especificaciones de requisitos de software (iee 830)

### 1.6.1 Introducción

La Empresa PUNTO CLASSIC TEXTILERA NACIONAL tiene la necesidad de implementar un sistema para administrar tanto el área financiera como la de producción industrial, por lo tanto se presenta este documento con las características tomadas en cuenta para el desarrollo del mismo.

### 1.6.2 Propósito

En este documento se describen las necesidades de un sistema financiero en línea para administrar recursos y simular procesos industriales de la empresa.

### 1.6.3 Ámbito del sistema

El sistema se enfocará en los siguientes procesos:

- Clientes.
- Cotizaciones.
- Proveedores.
- Inventarios.
- Cuentas por pagar.
- Cuentas por Cobrar.

- Facturación.
- Índices Financieros.
- Simulación Procesos Industriales.

El sistema no tendrá integrado el módulo de contabilidad, se implementará dos módulos: Financiero y Simulación de Procesos Industriales.

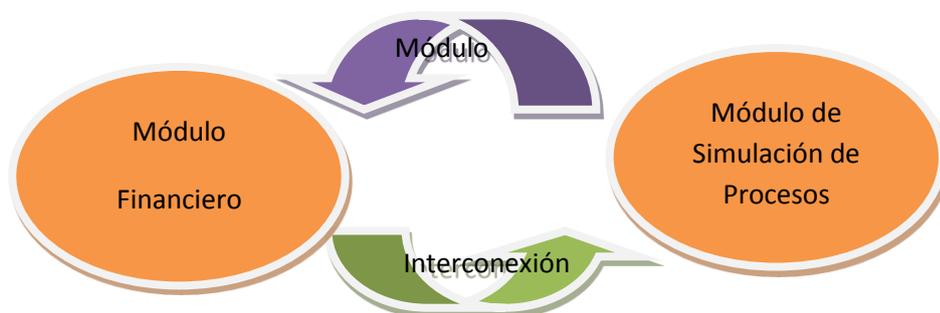
#### 1.6.4 Visión general

Punto Classic Textilera Nacional no posee un control adecuado de la producción diaria obtenida, ya que los procesos de elaboración son varios, y dependiendo de la demanda del producto; es necesario tener en cuenta que cuando se elabora un producto se necesita de variedad de materia prima, la que se deberá registrar en el sistema financiero y estar ligada al aumento o disminución de la producción.

Al respecto se propone como solución, realizar un sistema formado por un módulo financiero, y uno de simulación industrial para análisis previo a la implementación y viabilidad de la automatización industrial en la producción de Punto Classic Textilera Nacional.

#### 1.6.5 Descripción general

**Figura 1. Perspectiva del Producto**



**Elaborado por: Christian Reyes - Daniel Rocha**

## **Funciones del Producto**

El módulo Financiero se encargará de: administrar clientes, realizar diferentes tipos de cotizaciones, administrar proveedores, registrar la producción efectuada (inventarios), podrá realizar cálculos de las cuentas por pagar y por cobrar que se generen diariamente, además facturará la compra y venta de distintos productos.

La Simulación de procesos industriales de la empresa y la comunicación con el módulo financiero se verá vinculada directamente entre uno y otro, viéndose reflejada en los resultados de los reportes diarios.

## **Características de los Usuarios**

**Administrador.-** Este usuario tendrá la capacidad de administrar otros usuarios y perfiles de usuario gestores u operarios de la empresa, esto permitirá tener una mayor seguridad en el manejo del sistema.

**Gestor.-** Estos usuarios únicamente podrán realizar actividades dependiendo de los perfiles creados para cada gestor que han sido instanciados por los administradores pertinentes.

**Operarios.-** Estos usuarios son limitados completamente al cumplimiento de una sola actividad, la parametrización o perfiles que tendrá cada uno serán otorgados ya sea por los gestores o por los administradores dependiendo de la actividad a la que sea asignado.

## **Restricciones, Suposiciones y Dependencias**

Antes de empezar a desarrollar el proyecto, Punto Classic Textilera Nacional podrá pedir las modificaciones que deseen realizar al sistema, luego de esto solo se podrán realizar cambios menores o adicionales con la justificación del caso.

El Sistema Financiero se limita al uso interno de Punto Classic Textilera Nacional. Para el desarrollo de la propuesta se utilizará lenguaje .Net, la base de datos se desarrollará en SQLServer 2008 y el sistema operativo a utilizarse será Windows 7.

De acuerdo al alcance del proyecto el Sistema Financiero no tendrá el módulo de contabilidad.

### Requisitos Futuros

A futuro se pretende realizar mejoras en cuanto al funcionamiento del sistema, de tal manera que se ha considerado los siguientes aspectos:

- Versión compatible para cualquier plataforma.
- La base de datos se podrá migrar a otro motor de base de datos.
- En un futuro se planea agregar el módulo de contabilidad.

### 1.6.6 Requisitos específicos

**Tabla 5. Requisitos Específicos**

<b>Número de Requisito</b>	1
<b>Nombre Requisito</b>	Autenticación de Usuario
<b>Tipo</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Fuente</b>	Administrador
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Elaborado por: Christian Reyes - Daniel Rocha

### Requisitos comunes de las interfaces

Las interfaces de usuario deberán permitir ingresar la información de manera sencilla y mostrar de la manera solicitada por el usuario o administrador.

### Interfaces de usuario

La interfaz de usuario debe ser clara e intuitiva, para facilitar la interacción del usuario con el sistema.

La interfaz de usuario será desarrollada en Web con Aspx.

### **Interfaces de hardware**

Para el correcto funcionamiento del sistema se necesita computadoras con las siguientes características:

Dos PC con procesador Intel Core 2 DUO 2.80GHz, memoria de 2 GB, sistema operativo de 32 bits cada una.

### **Interfaces de software**

Deberá ser capaz de realizar operaciones necesarias para que los reportes solicitados tengan la información correcta. Tendrá conexión a una base de datos para registrar toda la información. Tendrá un módulo de autenticación de usuarios para realizar las distintas actividades dentro del mismo, tendrá un módulo de simulación de los procesos que ocurren en la Empresa Punto Classic Textilera Nacional.

### **Interfaces de comunicación**

Para la comunicación entre el módulo de simulación industrial y el módulo financiero se la realizará mediante Web Services para transmisión de datos de la producción los cuales serán actualizados en el sistema financiero.

## **1.7 Marco teórico**

### **1.7.1 Intouch**

Permite a ingenieros, supervisores, administradores y operadores visualizar e interactuar con el desarrollo de toda una operación a través de representaciones gráficas de procesos de producción industrial.

#### **➤ Usos**

Permite a supervisores y administradores visualizar (sin alterar) datos del área de producción de la planta en tiempo real desde una PC de escritorio ubicada en cualquier lugar de la red. También contiene una útil herramienta de software que le ayuda a los usuarios a crear sus propios “Wizards” (objetos preconfigurados).

## ➔ **Ventajas**

InTouch al ofrecer una interfaz gráfica y práctica, en especial para los ingenieros, dado que los conceptos que utiliza son los mismos aplicados en cualquier carrera de ingeniería, solo se necesita saber conocimientos básicos de programación, y el manejo del software es intuitivo.

InTouch permite realizar presentaciones muy entendibles, y las interfaces finales son de muy fácil manejo no solo para el diseñador, sino para cualquier extraño que quiera manipular el sistema.

## ➔ **Aplicaciones de INTOUCH**

InTouch se puede acoplar a módulos de SQL para ayudar a los usuarios a satisfacer una variedad de requisitos de presentación de informes de la industria.

Los campos de acción abarcan el mundo en una gran variedad de mercados verticales incluyendo la elaboración de alimentos, los semiconductores, el petróleo y el gas, automotriz, química, farmacéutica, pulpa y papel, transporte, servicios públicos y más.

### **1.7.2 Web service wcf**

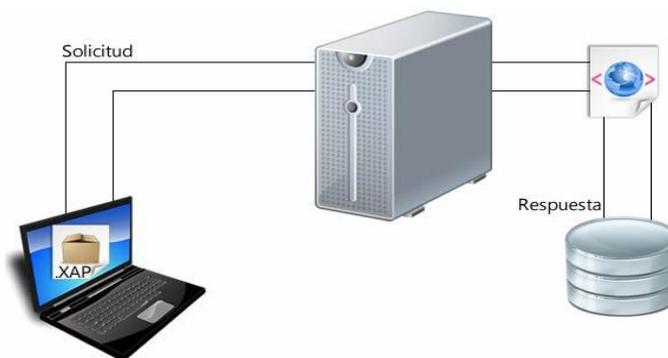
Wcf es una de las 4 mejoras circunstanciales del framework 3.0 junto con wpf, wwf y cardspace, es el primer modelo construido desde cero para el desarrollo de aplicaciones orientadas a servicios.

La mayor parte de la arquitectura se centra en la manera de utilizar componentes de una manera fácil y eficiente, wcf permite desarrollar los componentes utilizando programación orientada a objetos, combina y amplía las características que se utilizaban en los servicios web asmx, .net remoting, Enterprise service, services enhancements.

Wcf es un motor de mensajería el cual proporciona un modelo de programación simple, claro y flexible, tiene la opción de desarrollar através de clases java así como de utilizar archivos de configuración.

Los servicios web no son nada más que intermediarios entre la aplicación Silverlight y la base de datos, puesto que Silverlight se ejecuta en el “lado cliente” y no interactúa directamente con la base de datos, la cual se encuentra en el “lado servidor”, ya que, el servicio WCF se encuentra alojado del lado del servidor, puede acceder a la base de datos, solicitar la información y responder a nuestra aplicación, si se imagina en un diagrama, el resultado se lo podría visualizar como se muestra en la figura 2.

**Figura 2. Esquema de un Web Service**



**Fuente: EHU**

Existen dos tecnologías para el desarrollo de servicios web las cuales son ASP y WCF, ambos tienen características particulares que los hacen únicos, las cuales pueden ser.

### ➤ Servicios asp.net

Aparecieron, actualmente no fueron incluidos como parte del Framework 4.0 de .NET con esto dieron lugar a sus sucesores (WCF), pero aún es posible incluirlos dentro de una solución como parte de esta.

## ➤ Servicios wcf

Son la nueva herramienta para intercambio de información, abismalmente más equipados con herramientas y funcionalidades, tantas que son considerados como aplicaciones, es decir, aplicaciones orientadas a servicios, fueron creados efectivamente para reemplazar a los servicios ASP.NET y dotarlos de mejor funcionalidad.

### **WSDL**

Web Services Description Language - Lenguaje de Descripción de Servicios Web. Lenguaje basado en XML para describir servicios web. Describe la interfaz pública de los servicios web; eso significa que detalla los protocolos y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. WSDL se utiliza a menudo junto con SOAP y XML Schema.

Un programa cliente se conecta a un servicio web y puede leer el WSDL, determinando así las funciones disponibles en el servidor y por ende acceder a toda la lógica de negocio de los servicios.

## CAPÍTULO 2

### ANÁLISIS Y DISEÑO DEL SISTEMA

#### 2.1. Análisis y documentación del modelo de negocios a implementarse

Para que el módulo financiero pueda integrarse con el módulo de simulación industrial, es necesario que cumpla ciertos requisitos y restricciones que permitirán una vinculación entre estos de manera óptima, para ello se detalla los siguientes:

##### 2.1.1 Clientes

Este módulo tiene la funcionalidad de ingresar, modificar, eliminar, consultar clientes y contactos.

**Introducción.-** El sistema permite introducir información sobre los clientes como nombre, ciudad, país, contacto y las demás características de cada cliente, permitirá modificar las ya existentes y borrarlas.

**Entrada:** Nombre\_Cliente, Nombre\_Ciudad, Nombre\_Pais, Nombre\_Contacto, Apellido\_Contacto, Telefono\_Contacto, Ci\_Contacto, Tipo\_Cliente, Estado\_Cliente.

**Proceso:** Crear, actualizar o eliminar un cliente. En el caso de que se elimine un cliente se eliminarán los contactos que están asociados al mismo. Los códigos de los clientes serán autogenerados.

**Salida:** Datos del cliente creados, actualizados o eliminados y mensajes de lo que está ocurriendo.

##### 2.1.2 Proveedores:

Este módulo tiene la funcionalidad de ingresar, modificar, eliminar, consultar proveedores y contactos.

**Introducción.-** El sistema permite introducir información sobre los proveedores como nombre, ciudad, país, contacto, productos y las demás características de cada proveedor, permitirá modificar las ya existentes y borrarlas.

**Entrada:** Nombre\_Cliente, Nombre:Ciudad, Nombre\_Pais, Nombre\_Contacto, Apellido\_Contacto, Telefono\_Contacto, Ci\_Contacto, Tipo\_Cliente, Estado\_Cliente.

**Proceso:** Crear, actualizar o eliminar un proveedor. En el caso de que se elimine un proveedor se eliminarán los contactos que están asociados al mismo. Los códigos de los proveedores serán autogenerados.

**Salida:** Datos del proveedor creados, actualizados o eliminados y mensajes de lo que está ocurriendo.

### **2.1.3 Registro de cotizaciones**

Este módulo permite crear las cotizaciones enviadas por los proveedores de la empresa, para seleccionar la orden de compra y realizar los respectivos informes.

**Introducción:** El sistema permite seleccionar una solicitud de compra y con esta registrar por cada producto de la solicitud varias cotizaciones de distintos proveedores.

**Entrada:** Nombre\_Insumo, Precio, Cantidad, Proveedor.

**Proceso:** Permite registrar cotizaciones.

**Salida:** Reportes generados, mensajes de lo que está ocurriendo.

### **2.1.4 Orden de compra**

**Introducción.-** Este módulo permite el ingreso de la orden de compra así como los datos de la compra a ser realizada.

**Entrada:** Nombre\_Insumo, Cantidad, Proveedor.

**Proceso:** Dependiendo de las cotizaciones previamente ingresadas, se tendrá la oportunidad de elegir una de ellas y convertirla en orden de compra.

**Salida:** Orden de Compra Generada.

### **2.1.5 Administrador de bodegas**

Este módulo permite la administración productos que tiene la empresa, tendrá las funciones de ingresar, modificar y consultar cada uno de ellos con los respectivos reportes o informes generados según el usuario de la aplicación.

**Introducción.-** El sistema permite ingresar información sobre las bodegas como nombre, ciudad, país, dirección, teléfono, permitirá modificar las ya existentes y borrarlas.

**Entrada:** Nombre, Ciudad, País, Dirección, Teléfono, Extensión.

**Proceso:** Crear, actualizar o eliminar una bodega. En el caso de que se elimine una bodega se eliminarán los datos que están asociados a la misma. Los códigos de las bodegas serán autogenerados.

**Salida:** Datos de la bodega creados, actualizados o eliminados y mensajes de lo que está ocurriendo.

### **2.1.6 Administrador de pagos**

Este módulo se encargará de facilitar la administración de los pagos que la Textilera tiene pendientes, visualizar elementos adquiridos y los pagos correspondientes a cada uno de esos elementos, entregarnos reportes y comprobantes de los pagos realizados que han sido registrados en el sistema.

**Entrada:** Monto, detalle, número de pagos, fecha inicial de pagos, monto.

**Proceso:** Crear, actualizar, archivar y vincular pagos pendientes con facturas o pagos actuales.

**Salida:** Datos pertenecientes a los pagos pendientes, reportes individuales del proceso de pago o reportes de todos los pagos correspondientes a realizarse en rangos de fechas establecidos por el usuario.

### **2.1.7 Administrador de usuarios**

Este módulo no dará la posibilidad de administrar usuarios que podrán acceder a la aplicación.

**Entrada:** Nombre, password, identificación.

**Proceso:** Crear, actualizar y borrar usuarios dependiendo de la necesidad del administrador o gerente de la empresa.

**Salida:** Usuarios con roles dependiendo del grupo al que ha sido asignado y parametrizado previamente.

### **2.1.8 Administrador de inventarios**

En éste módulo se tiene la posibilidad de asignar máximos, mínimos a los productos que tiene la empresa, ingresar productos a la empresa, realizar solicitudes de productos de las sucursales, aprobar las solicitudes y dar de baja a productos.

**Entrada:** Código del producto, tipo de producto, sucursales, cantidad.

**Proceso:** Ingreso de los productos por medio de la pantalla, mismos que serán tomados en cuenta al momento de llamada o uso de otros módulos.

**Salida:** La entrega de reportes concernientes a los productos que se encuentran en bodega.

### **2.1.9 Generación de índices financieros**

Este módulo se encargará de generar los repostes de los índices financieros de liquides y de actividad de la empresa,.

### **2.1.10 Administrador de permisos**

Este módulo dará la posibilidad de parametrizar permisos de acceso a los módulos del sistema. Estos permisos se verificarán en el momento en el que el usuario se loguee en el sitio web, y se le mostrarán únicamente los módulos permitidos en la parametrización realizada.

**Tabla 6. Cuadro de Requisitos**

# Requisito	Nombre Requisito	Tipo	Fuente	Prioridad
1	Administración de Clientes	1	Administrador	1
2	Administración de Proveedores	1	Administrador	1
3	Registro de Cotizaciones	1	Administrador	1
4	Orden de Compra	1	Administrador	1
5	Administración de Bodegas	1	Administrador	1
6	Administración de Pagos	1	Administrador	1
7	Administración de Usuarios	1	Administrador	2
8	Administración de Inventarios	1	Administrador	1
9	Generación de Índices F.	1	Administrador	1
10	Administrador de Permisos	1	Administrador	1

**Elaborado por: Christian Reyes - Daniel Rocha**

## **2.2 Análisis y diseño del control de procesos industriales**

El análisis de los procesos industriales que se efectuaron en Punto Classic han logrado determinar una cronología en la elaboración y el procesamiento de la tela producida, o también llamada tela en “cruda”.

### **2.2.1 Previo al tejido**

Este proceso se realiza manualmente ya que, consiste en limpiar cada cono de hilo a usarse en el tejido, se lo limpia de pelusa aderida, y se lo coloca en cada uno de los carretes de la tejedora para comenzar con el tejido.

**Figura 3. Rollo de hilo**



**Imagen: Christian Reyes - Daniel Rocha**

**Figura 4. Rollos de Hijo colocados en los carretes de la Tejedora**



**Imagen: Christian Reyes - Daniel Rocha**

### **2.2.2 Tejido**

Antes de iniciar el tejido se comprueba que las cada una de las agujas que se usan en la maquinaria estén correctamente colocadas y no vayan a estar rotas, ya que si esto se diera el tejido podría hecharse a perder.

**Figura 5. Tejedora 1**



**Imagen: Christian Reyes - Daniel Rocha**

**Figura 6. Tejedora 2**



**Imagen: Christian Reyes - Daniel Rocha**

A continuación presentamos una fotografía de una tejedora más actualizada.

**Figura 7. Eje de agujas de la Tejedora**



**Fuente: ORIZIO**

**Figura 8. Panel de Controles básicos de la Tejedora**



**Fuente: ORIZIO**

**Figura 9. Carretes de hilo de la Tejedora**



**Fuente: IRIZIO**

### **2.2.3 Teñido**

Este proceso es el encargado de darle color a la tela mediante un proceso de bañado de la tela en tinturas especializadas, para evitar una despigmentación de la misma.

**Figura 10. Maquinaria de Teñido de Tela**



**Imagen: Christian Reyes - Daniel Rocha**

### **2.2.4 Planchado**

Este proceso mantiene la funcionalidad de receptor la tela teñida, y colocarla en medio de un conjunto de rodillos que se encuentran a una temperatura adecuada, a medida de la velocidad con la que estos rodillos giren, irán entregando la tela enrollada para su almacenamiento y distribución.

**Figura 11. Maquinaria de Planchado de Tela**



**Imagen: Christian Reyes - Daniel Rocha**

**Figura 12. Bodegas de Tela Planchada**



**Imagen: Christian Reyes - Daniel Rocha**

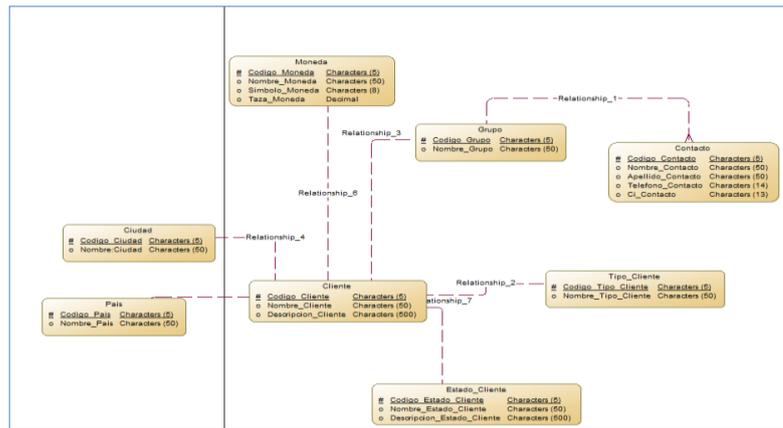
### **2.3 Análisis y diseño de la base de datos**

La Base de Datos se encuentra separada en esquemas por cada uno de los módulos que tiene el sistema Financiero y se encuentra representada por 3 diagramas que son: Conceptual, Físico y Lógico.

#### **2.3.1 Clientes**

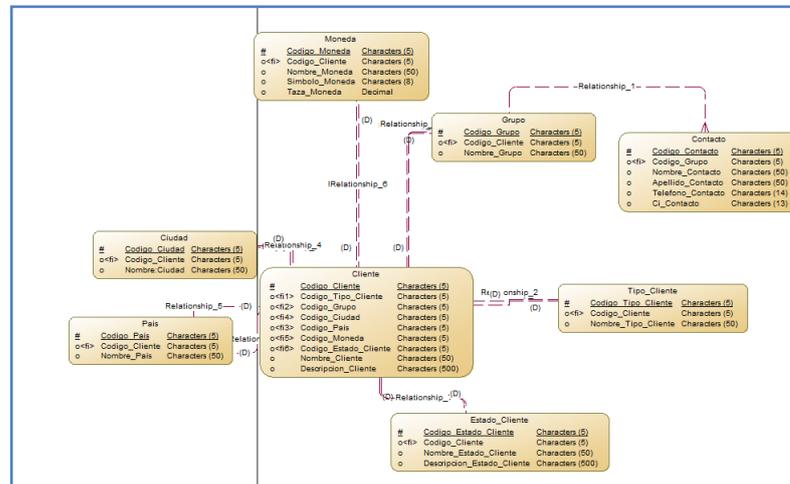
En siguientes figuras se muestran la forma en la que se almacenarán los datos de los clientes que tiene la empresa.

**Figura 13. Diseño Conceptual de Clientes**



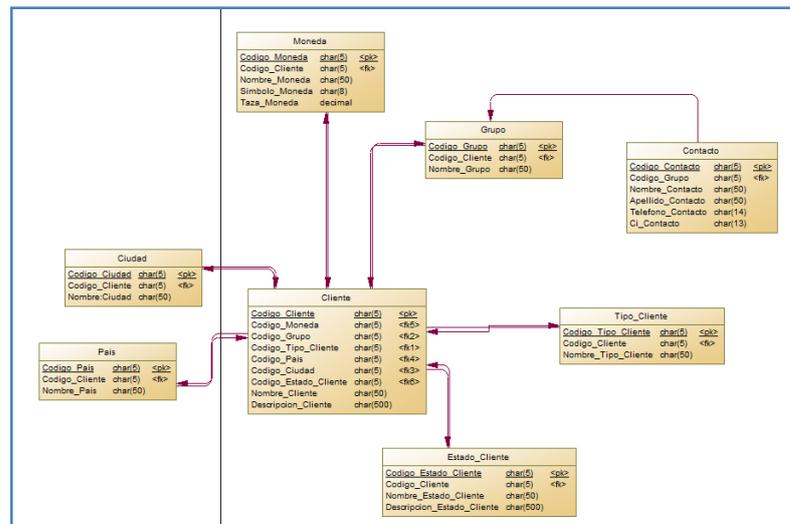
Elaborado por: Christian Reyes - Daniel Rocha

**Figura 14. Diseño Modelo Lógico de Clientes**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 15. Diseño Modelo Físico de Clientes**

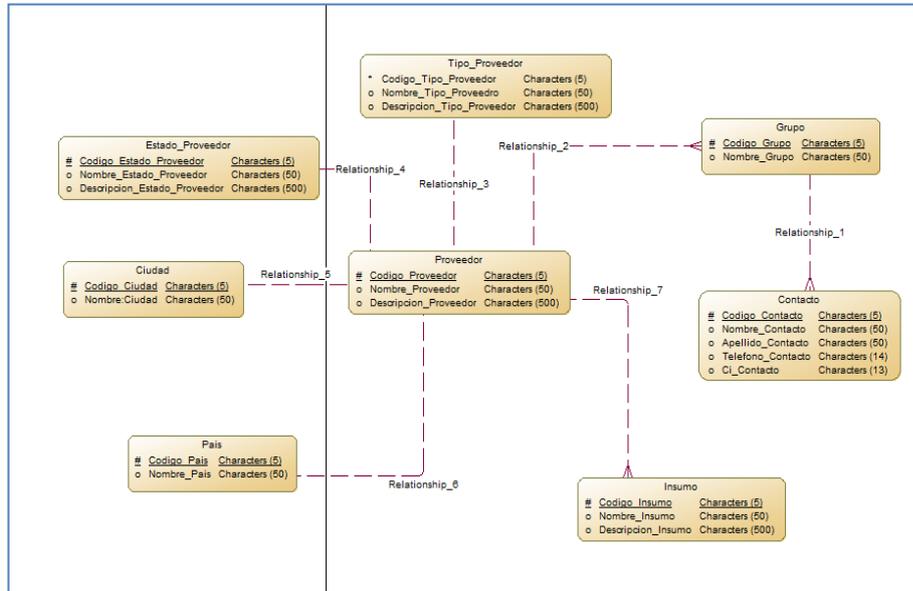


Elaborado por: Christian Reyes - Daniel Rocha

### 2.3.2 Proveedores

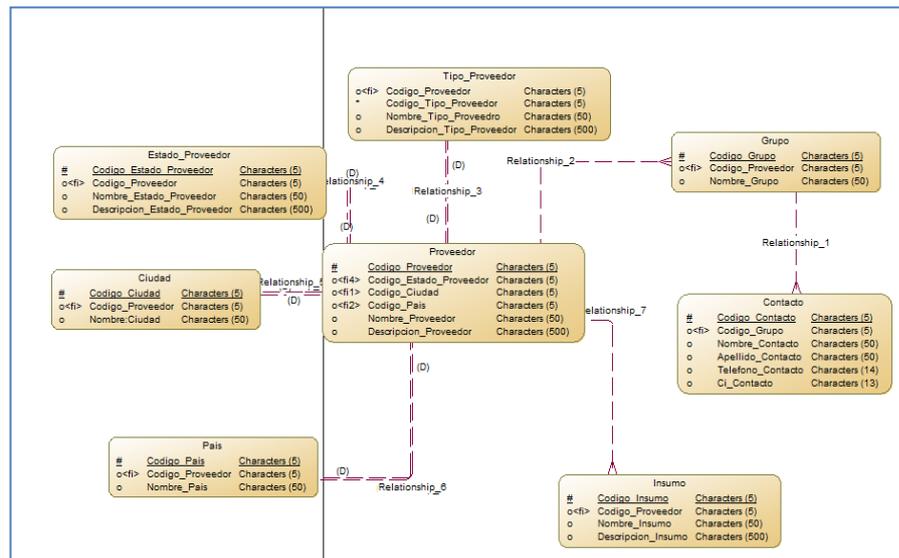
En siguientes figuras se muestran la forma en la que se almacenarán los datos de los proveedores que tiene la empresa.

**Figura 16. Diseño Modelo Lógico de Proveedores**



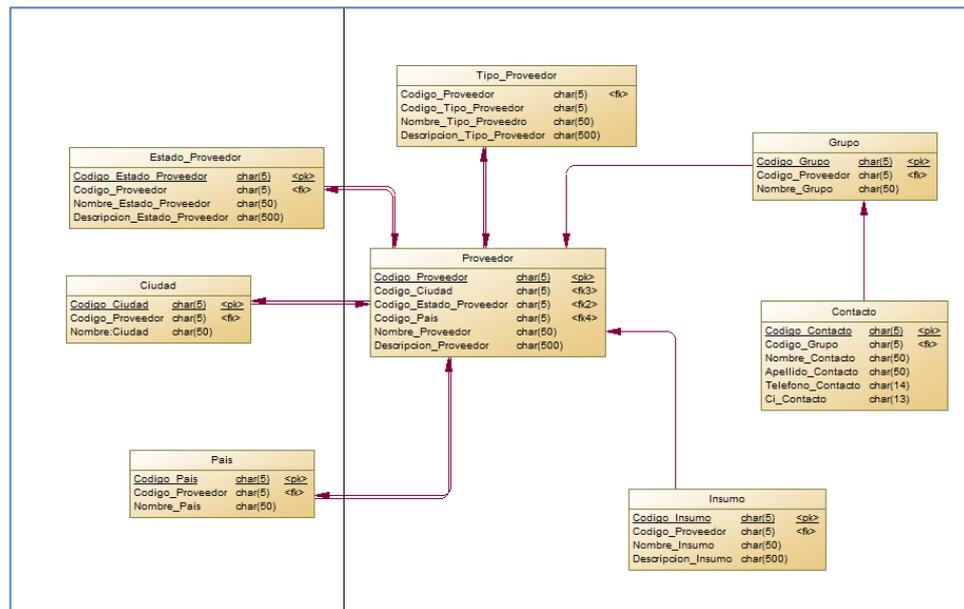
Elaborado por: Christian Reyes - Daniel Rocha

**Figura 17. Diseño Conceptual de Proveedores**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 18. Diseño Modelo Físico de Proveedores**

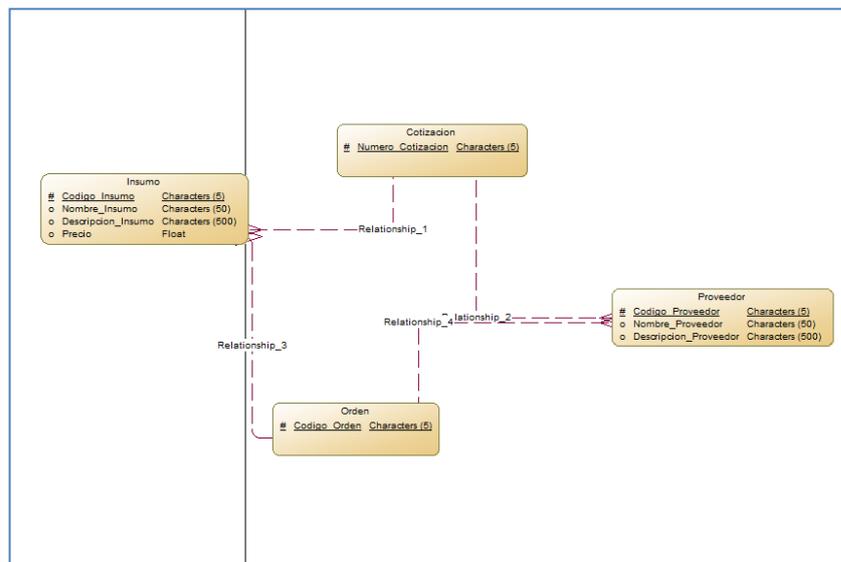


Elaborado por: Christian Reyes - Daniel Rocha

### 2.3.3 Cotización

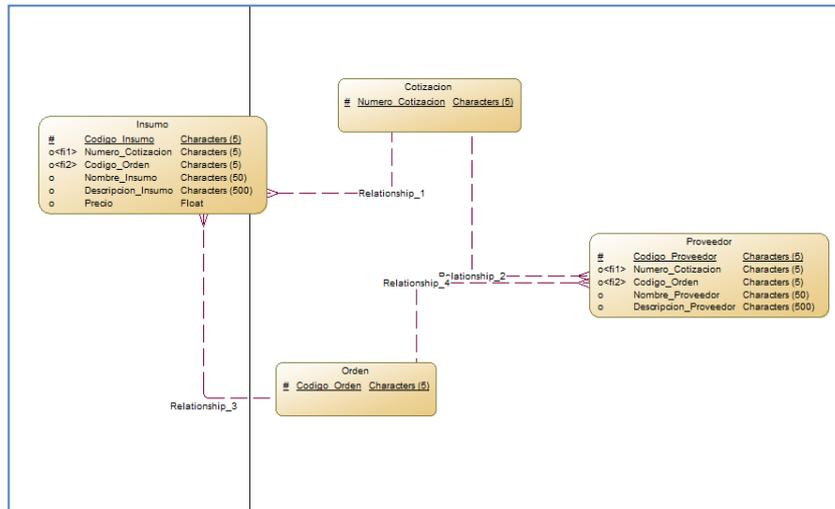
En siguientes figuras se muestran la forma en la que se almacenarán los datos de las cotizaciones que se realiza la empresa.

**Figura 19. Diseño Lógico de Cotización**



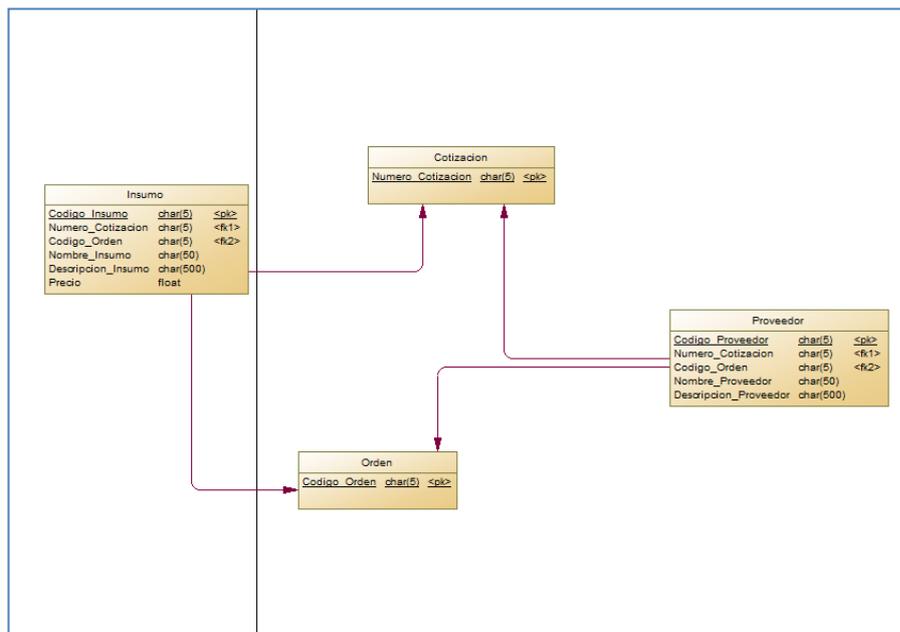
Elaborado por: Christian Reyes - Daniel Rocha

**Figura 20. Diseño Modelo Conceptual de Cotización**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 21. Diseño Modelo Físico de Cotización**

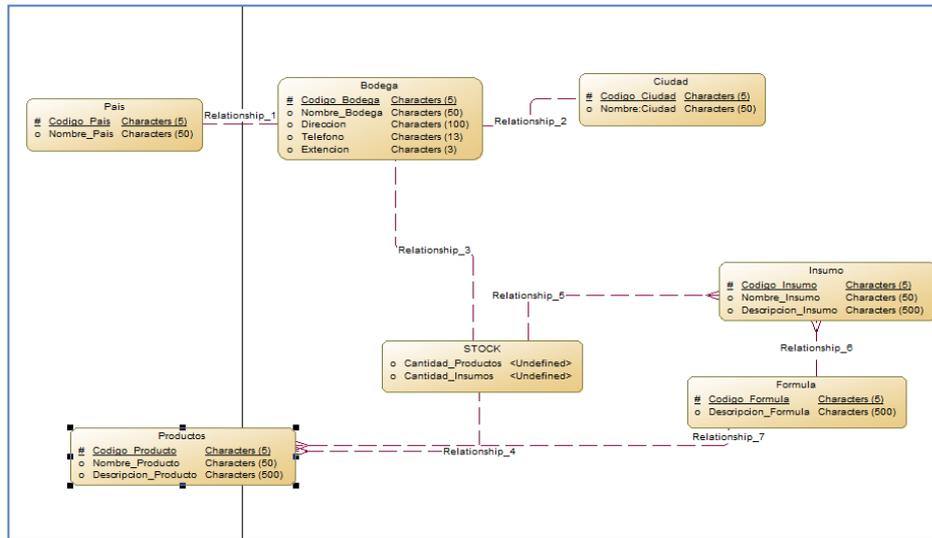


Elaborado por: Christian Reyes - Daniel Rocha

### 2.3.4 Inventarios

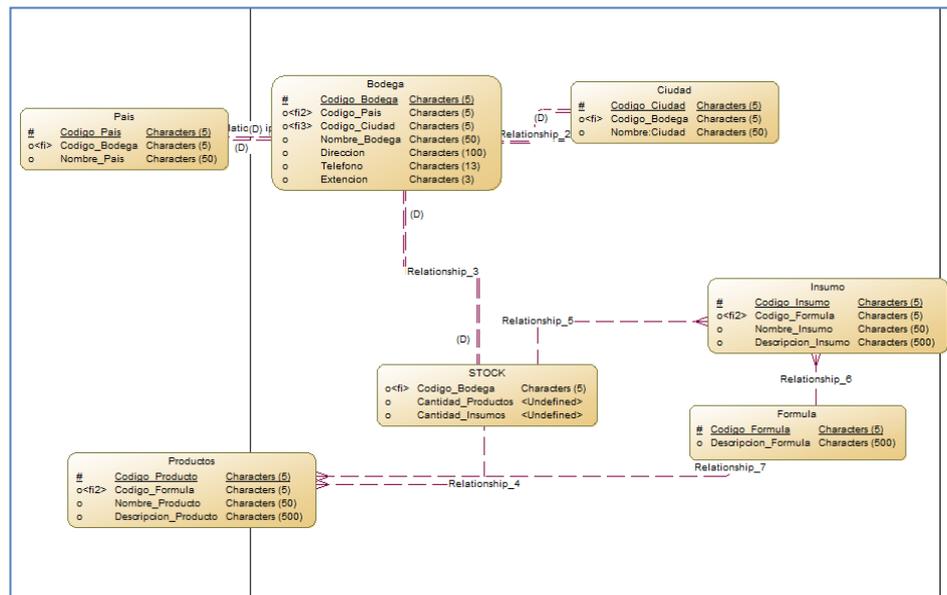
En las siguientes figuras se muestran la forma en la que se almacenarán los datos de los inventarios que realiza la empresa.

**Figura 22. Diseño Lógico de Inventarios**



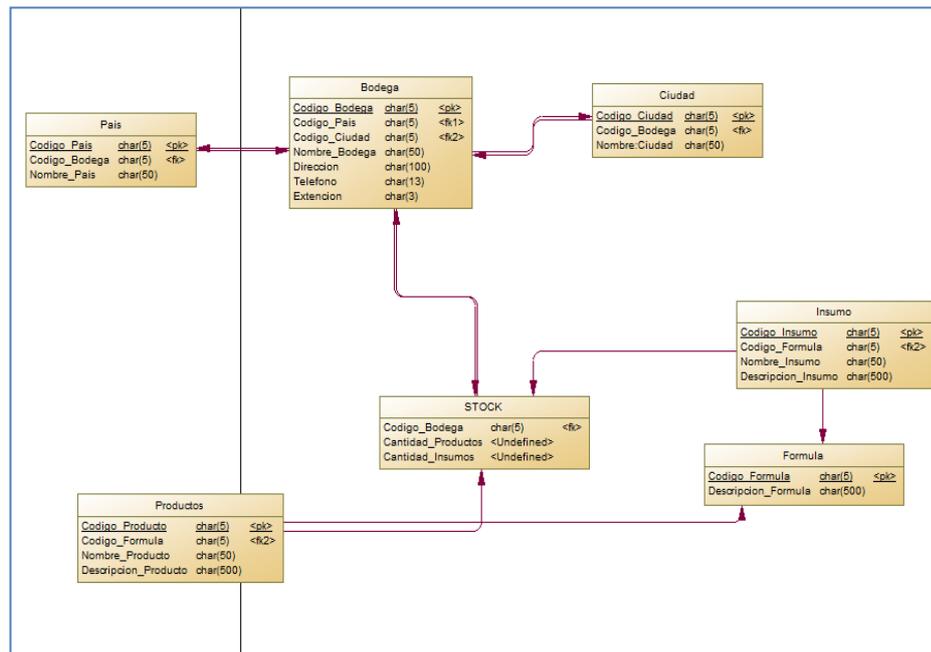
Elaborado por: Christian Reyes - Daniel Rocha

**Figura 23. Diseño Conceptual de Inventarios**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 24. Diseño Modelo Físico de Inventarios**

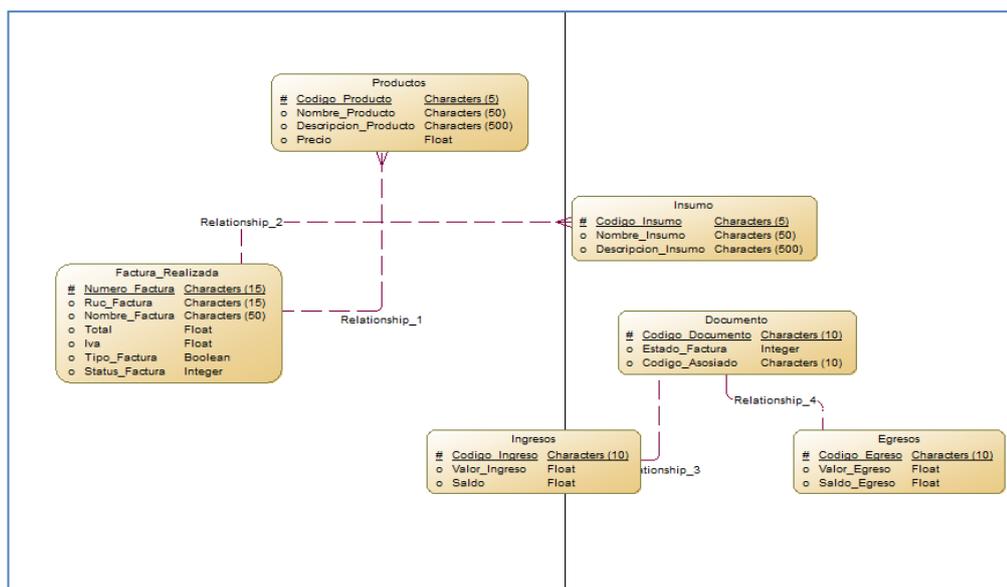


Elaborado por: Christian Reyes - Daniel Rocha

### 2.3.5 Facturación

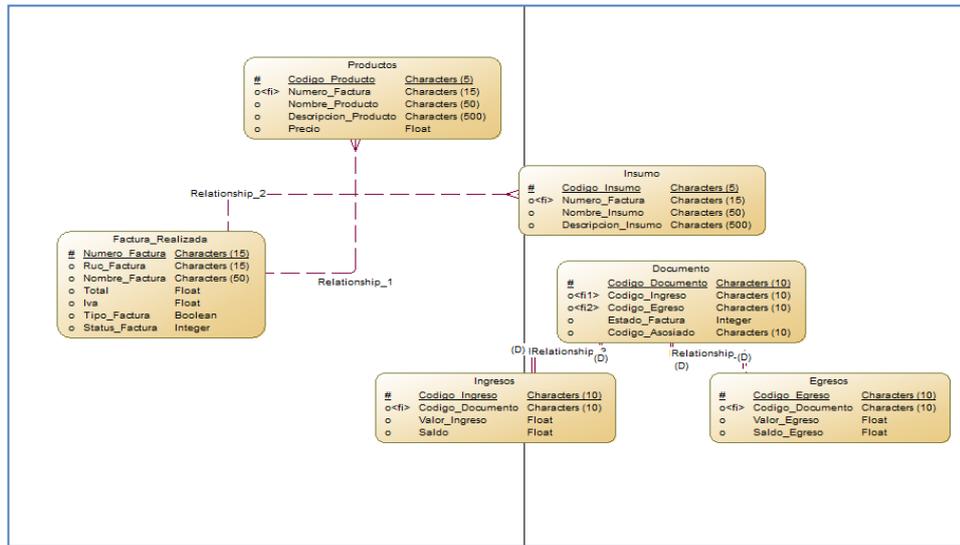
En siguientes figuras se muestran la forma en la que se almacenarán los datos de la facturación que tiene la empresa.

**Figura 25. Diseño Lógico de Facturación**



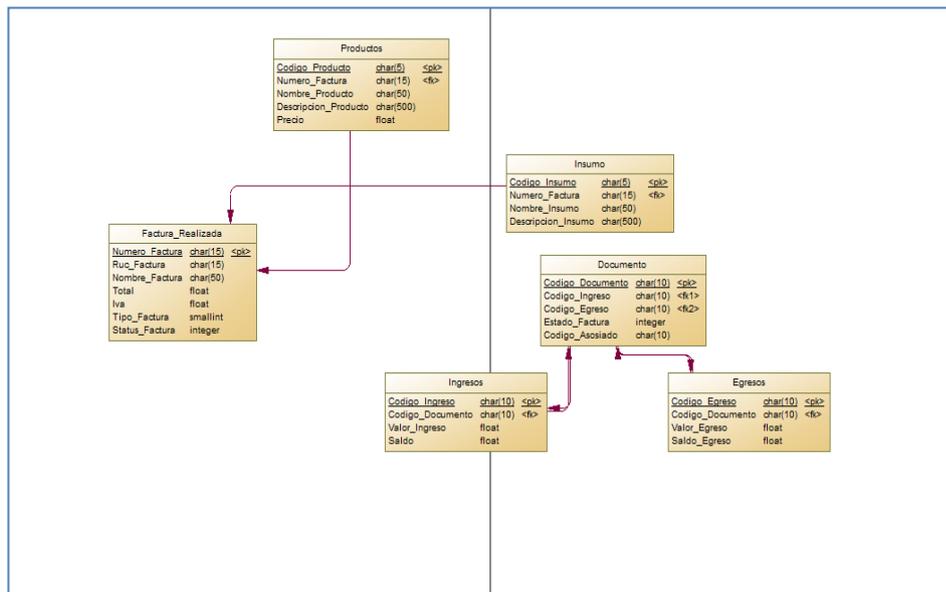
Elaborado por: Christian Reyes - Daniel Rocha

**Figura 26. Diseño Modelo Conceptual de Facturación**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 27. Diseño Físico de Facturación**



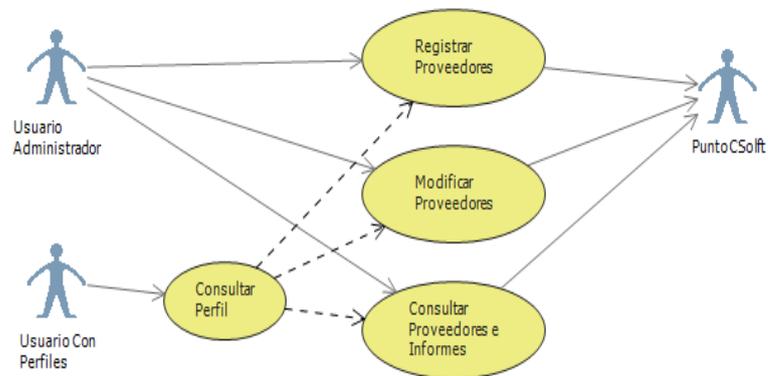
Elaborado por: Christian Reyes - Daniel Rocha

## 2.4 Análisis y diseño del sistema financiero

En Sistema Financiero contará con las siguientes funcionalidades las cuales están representadas por diagramas de caso de uso:

## 2.4.1. Proveedores

Figura 28. Proveedores



Elaborado por: Christian Reyes - Daniel Rocha

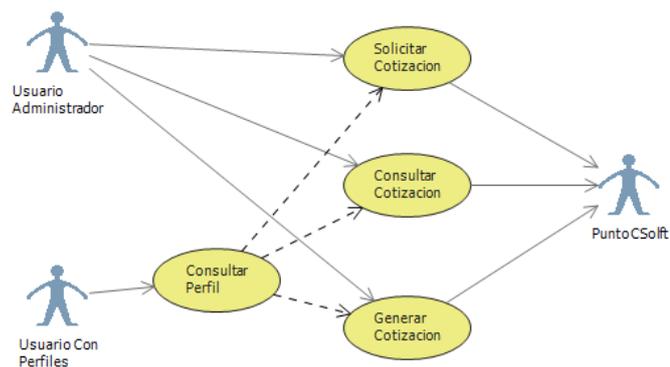
El diagrama indica las funcionalidades que se tendrán dentro del módulo de Proveedores.

El cual podrá ser ingresado por el usuario administrador o un usuario que posea permisos a este módulo.

En el módulo de proveedores el sistema permitirá el ingreso, la modificación y la consulta de los mismos con sus respectivos reportes.

## 2.4.2 Cotización

Figura 29. Cotización



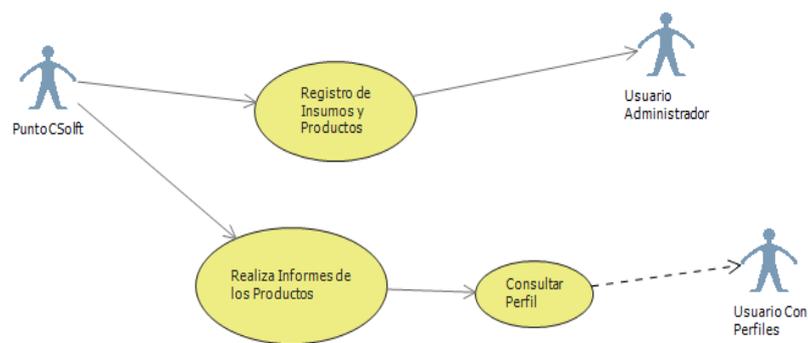
Elaborado por: Christian Reyes - Daniel Rocha

En la figura 29 se observa las funcionalidades del módulo de cotizaciones. En el cual se podrá ingresar con el usuario de la aplicación, tomando en cuenta los permisos que posea dicho usuario.

En el módulo de cotización el sistema podrá realizar el ingreso, consulta y generación de cotizaciones.

### 2.4.3 Inventarios

**Figura 30. Inventarios**



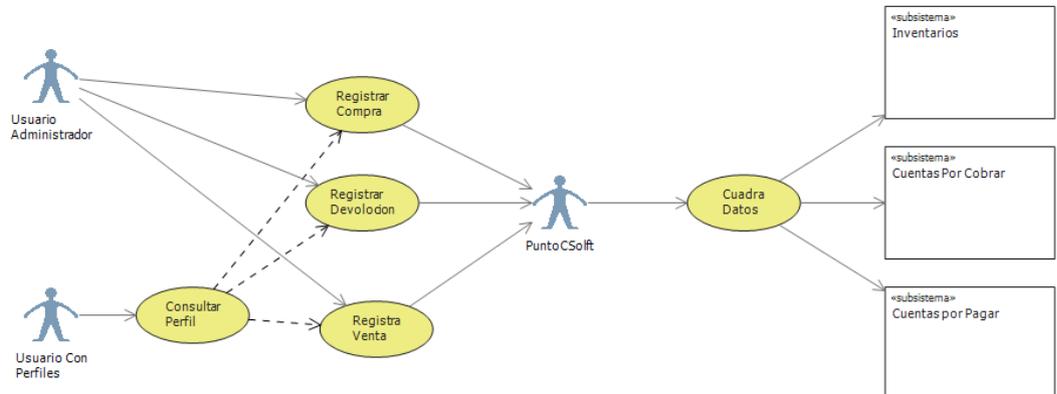
**Elaborado por: Christian Reyes - Daniel Rocha**

En el figura 30 se observa las funcionalidades que tendrá el módulo de inventarios. A este módulo tendrá acceso solo un grupo de usuarios dependiendo de la parametrización realizada.

El módulo de Inventarios permitirá el registro de productos y la generación de reportes, además de mantener una comunicación con el subsistema de simulación industrial.

## 2.4.4 Facturación

Figura 31. Facturación



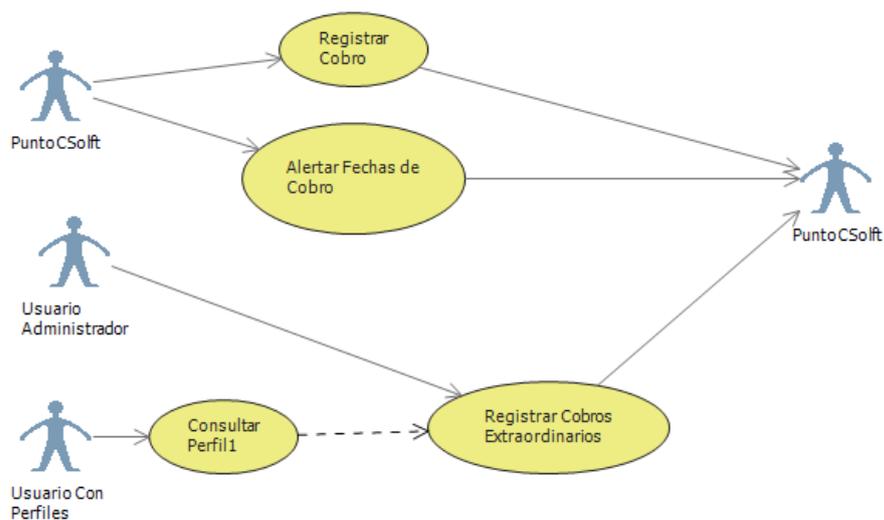
Elaborado por: Christian Reyes - Daniel Rocha

El gráfico 31 muestra funcionalidades del módulo de facturación. Al cual podrán acceder los usuarios administradores o grupo de usuarios que posean permisos.

En el módulo de facturación el sistema realizará el registro de la compra, devolución y venta de productos y la generación de informes, éste módulo se relaciona directamente con el módulo de inventario, cuentas por cobrar y cuentas por pagar.

## 2.4.5 Cuentas por cobrar

Figura 32. Cuentas por Cobrar



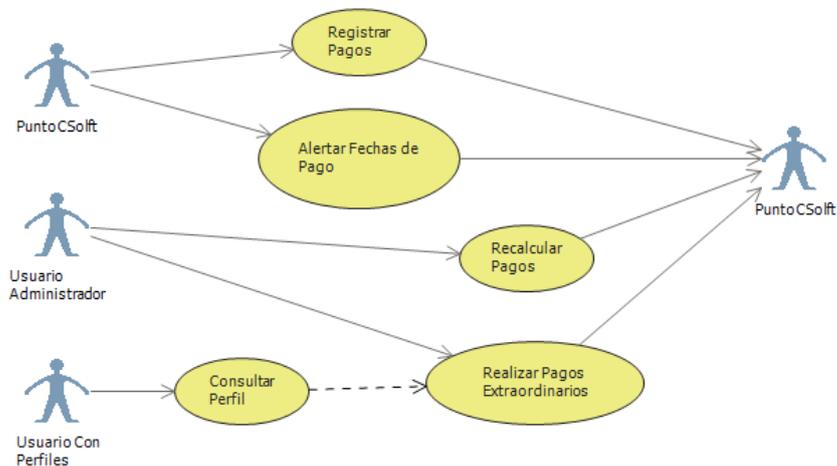
Elaborado por: Christian Reyes - Daniel Rocha

El Figura 32 muestra las funciones que posee el módulo de cuentas por cobrar. A este módulo podrán acceder los usuarios administrador o el grupo de usuarios para el que se encuentren parametrizados los permisos.

En éste módulo el sistema realizará el registro, alerta cobros y la generación de reportes, cabe recalcar que se encuentra vinculado directamente con el módulo de facturación.

## 2.4.6 Cuentas por pagar

**Figura 33. Cuentas por Pagar**



**Elaborado por: Christian Reyes - Daniel Rocha**

El módulo de cuentas por pagar detalla las funcionalidades en la imagen (*Figura 33*) que se muestra.

En el módulo el sistema permitirá el registro, alerta, calcular pagos y la generación de informes, éste módulo se encuentra vinculado directamente con el módulo de facturación.

## 2.5 Análisis y diseño de módulo de simulación

Para el análisis de éste módulo es necesario tomar muy en cuenta la cronología de procesos en la elaboración de la tela hasta dejarla como producto terminado, a continuación:

1. Tejido
2. Teñido
3. Planchado

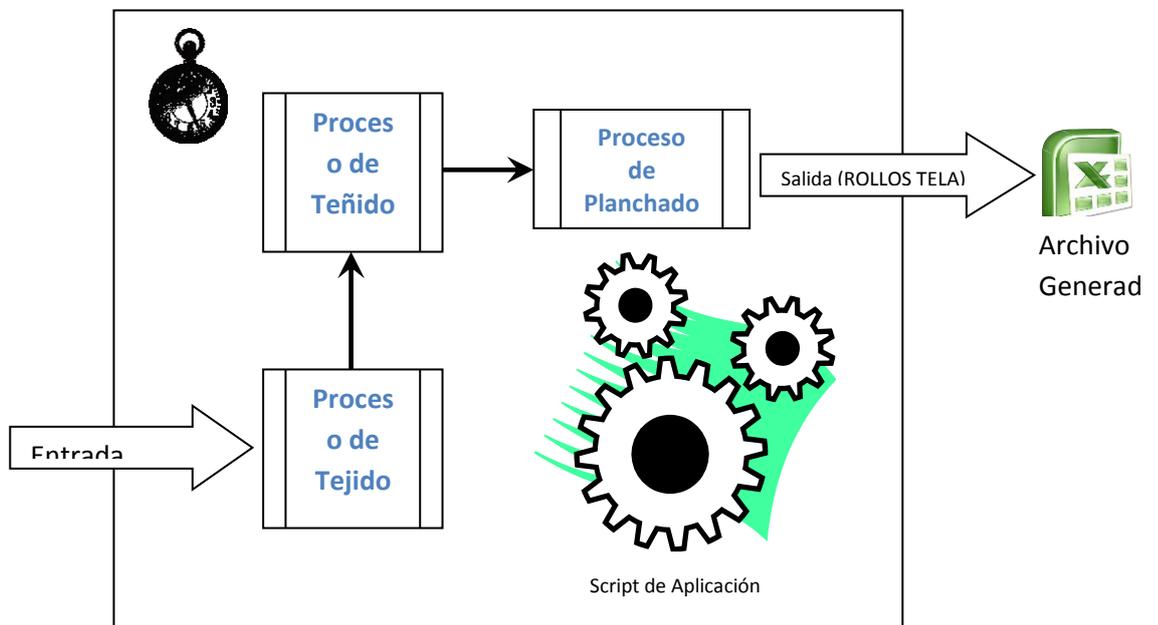
**Figura 34. Secuencia de procesos en la Textilera**



Elaborado por: Christian Reyes - Daniel Rocha

Una vez mostrada la secuencia en la elaboración de la tela, se estableció el siguiente diagrama donde se explica la simulación realizada en Intouch.

**Figura 35. Diagrama de Simulador**



Elaborado por: Christian Reyes - Daniel Rocha

En el diagrama del figura 35 se observa el funcionamiento del simulador que se encuentra formado por una entrada (hilos) y una salida (rollos de tela). Además se observa un gráfico de piñones que referencia al script de la aplicación el cual es el

motor de todo el simulador, para esto es necesario un reloj o timer el que permite controlar el proceso de simulación y generar pulsos de avance o de movimiento en cada uno de los gráficos que conforman el simulador.

## **2.6 Índices financieros**

Este módulo muestra las interacciones y dependencias entre los anteriormente detallados, y determina la idea acerca del comportamiento o desempeño de toda la organización, visualizado en una tabla de resultados, también llamado reporte.

## **CAPÍTULO 3**

### **DESARROLLO DEL SISTEMA**

#### **3.1. Introduccion**

En este capítulo se verá como fue desarrollado el sistema, sus diferentes módulos y los procesos que componen cada uno de estos.

Entre los módulos del sistema, se tiene los siguientes.

- Módulo de seguridad
- Módulo de administracion
- Módulo de clientes
- Módulo de proveedores
- Módulo de inventarios
- Módulo de facturacion
- Módulo de cotización
- Módulo de cuentas

#### **3.2. Módulo de seguridad**

Este módulo se encarga de controlar siguienter procesos, el ingreso al sistema, asi como la creación de nuevos usuarios, asignación de permisos sobre diferentes módulos y la parametrizacion de los distintas sucursales sobre las cuales pueden afectar.

##### **3.2.1. Proceso de ingreso al sistema**

Este proceso se encarga de verificar las credenciales del usuario que son nombre de usuario y contraseña, después carga los módulos asignados al usuario y las sucursales a las cuales puede acceder.

Para realizar este proceso se invocan los siguientes métodos del servicio web Login, para verificar la existencia del usuario se invoca el método ConsultarExistenciaUsuario, para obtener la identificación se utiliza el método

ObtenerIdPorLogin, para cargar los módulos que tiene activo el usuario se utiliza el método ObtenerMóduloPorUsuario y para cargar las sucursales a las cuales se tiene acceso se utiliza el método Obtener SucursalPorUsuario.

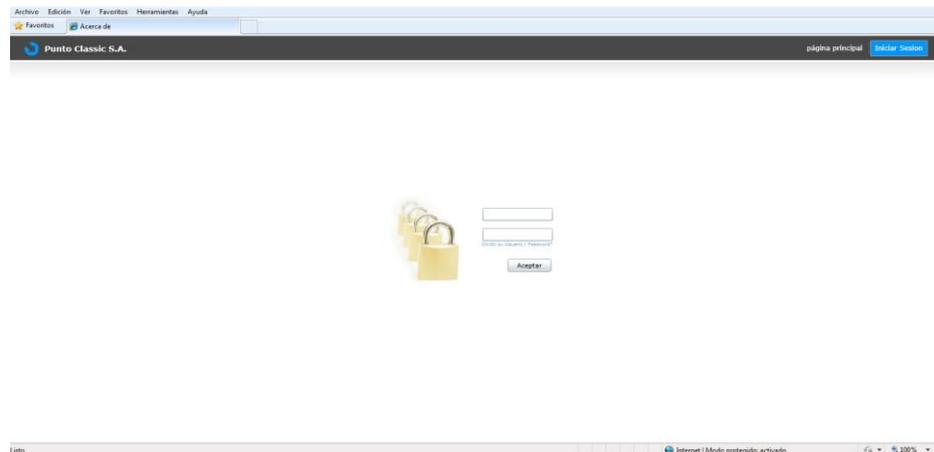
**Figura 36. Proceso de ingreso al sistema**



**Elaborado por: Christian Reyes - Daniel Rocha**

Para el desarrollo de este proceso intervienen los siguientes elementos, en la capa de presentación tenemos la pantalla Login.xaml la cual tiene las cajas de texto txtLogin, la caja de password txtPassword y el botón btnAceptar.

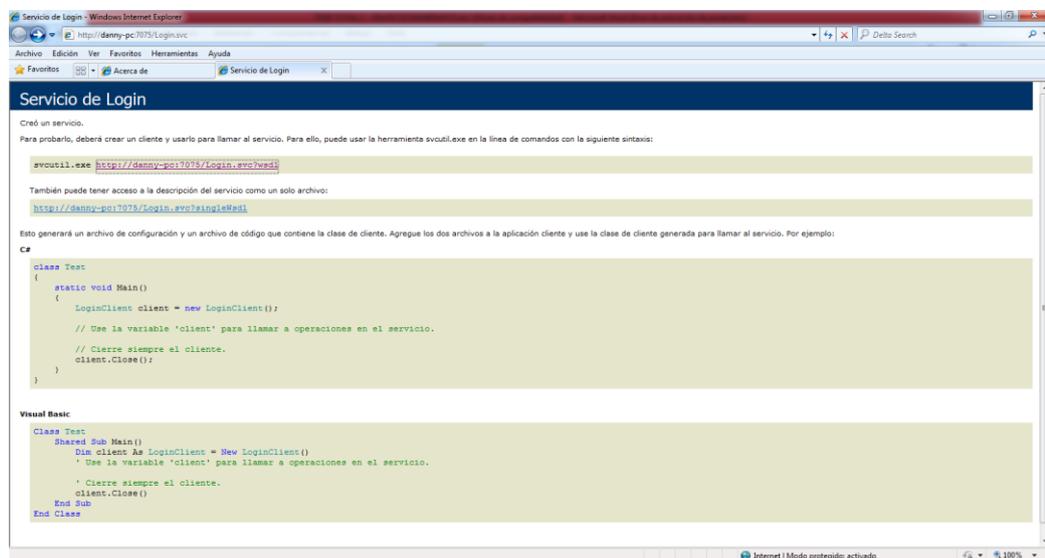
**Figura 37. Pantalla de Login**



**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la capa de servicios se utiliza el servicio <http://localhost:7075/Login.svc>, en el cual intervienen los métodos ConsultarExistenciaUsuario, ObtenerIdPorLogin, ObtenerMóduloPorUsuario, ObtenerSucursalPorUsuario.

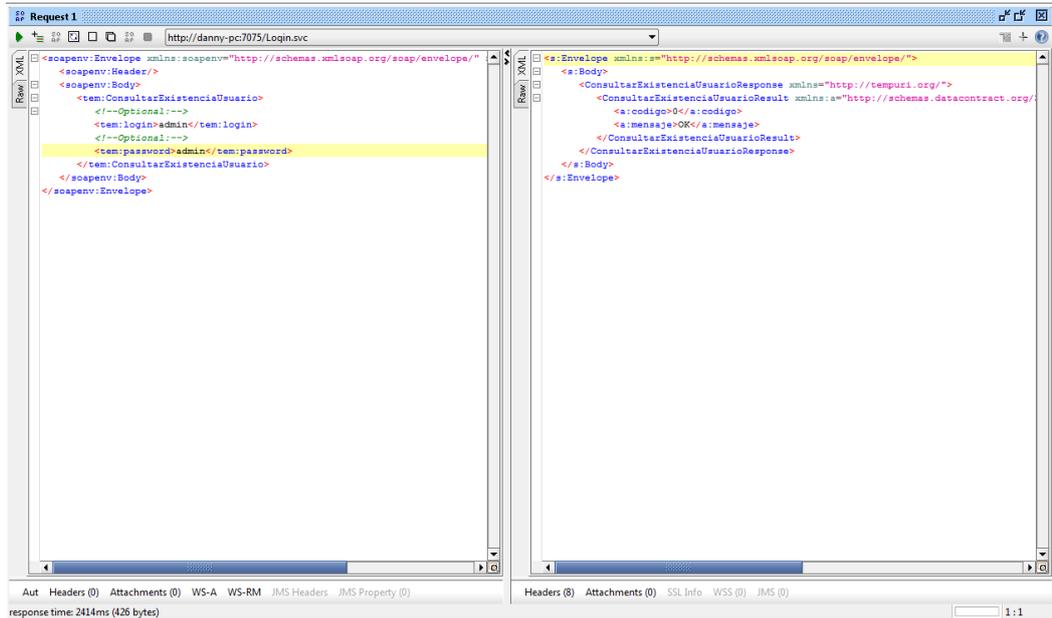
**Figura 38. Servicio Login**



**Fuente: Proyecto Capa de Servicios de Punto Classic**

El método ConsultarExistenciaUsuario se encarga de verificar si las credenciales son correctas, se tiene como entrada el login, password, como salida se tiene código, mensaje, devuelve código 0, mensaje "ok" en caso de ser correctas.

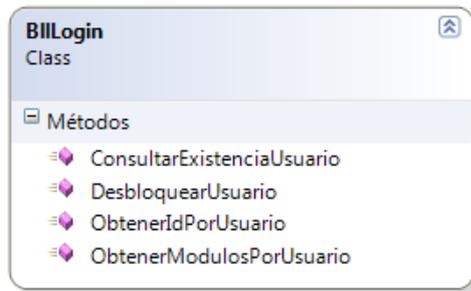
**Figura 39. Método ConsultarExistenciaUsuario**



**Fuente: Proyecto Capa de Servicios de Punto Classic**

En este método se utiliza en la capa de negocios la clase BllLogin el método ConsultarExistenciaUsuario.

**Figura 40. Clase BllLogin**



**Fuente: Proyecto Capa de Negocio de Punto Classic**

Este a su vez utiliza de la capa de acceso a datos la clase ALogin los métodos validarLogin y ActualizarIntentosFallidos en el caso de que existiese un error en las credenciales que se ingresan.

En la base de datos se invocarán los procedimientos almacenados sp\_validarLogin y sp\_ActualizarIntentosFallidos los cuales utilizan la tabla login

**Figura 41. Tabla login**

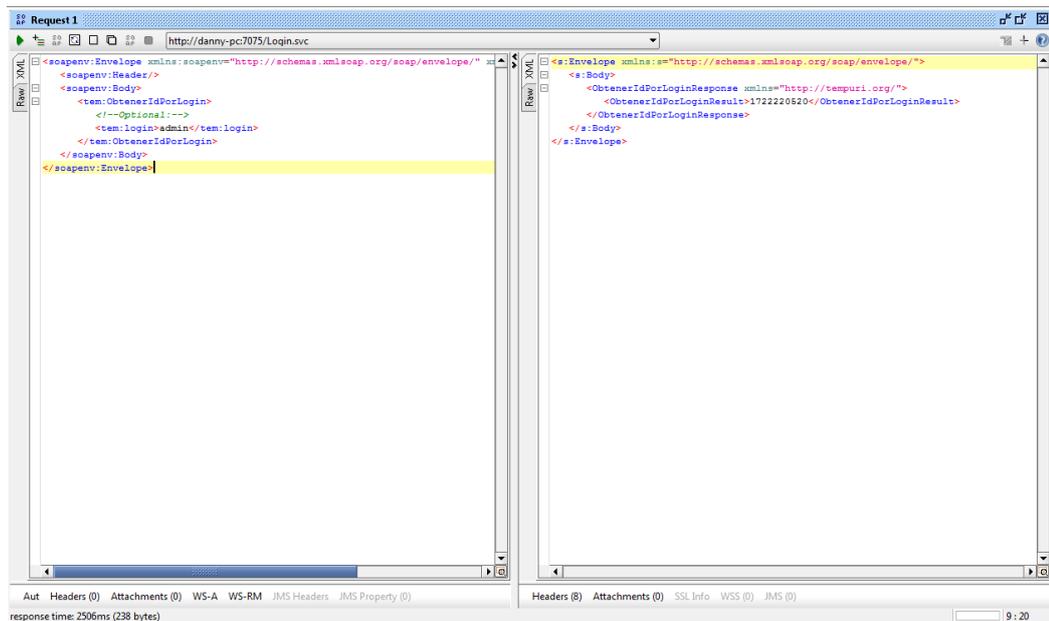


LOGIN	
ID_USUARIO	
NOMBRE_USUARIO	
PASSWORD	
INTENTOS	

**Fuente: Proyecto base de datos de Punto Classic**

El método ObtenerIdPorLogin se encarga de obtener la identificación por medio del usuario, se tiene como entrada el login, como salida se tiene ObtenerIdPorLoginResult en el cual se devuelve la identificación.

**Figura 42. Método ObtenerIdPorLogin**



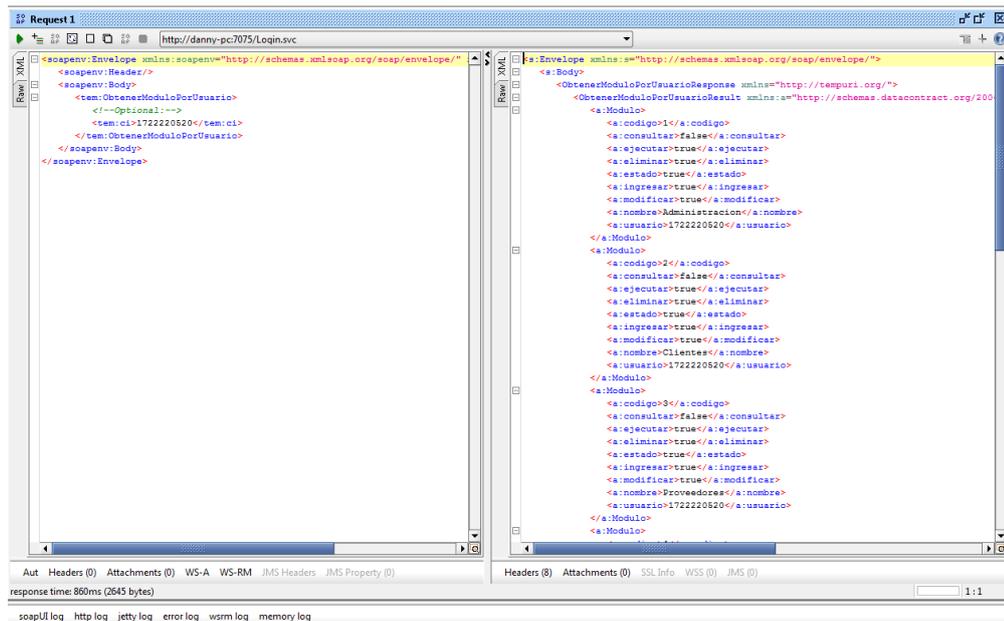
**Fuente: Proyecto capa de servicios de Punto Classic**

En este método se utiliza en la capa de negocios la clase BllLogin el método ObtenerIdPorUsuario este a su vez utiliza de la capa de acceso a datos la clase ALogin los métodos ObtenerId.

En la base de datos se usará el procedimiento almacenado sp\_ConsultarId, en este sp se consulta la tabla logín.

El método ObtenerMóduloPorUsuario se encarga de obtener los módulos por medio de la identificación, se tiene como entrada el CI, como salida se tiene una lista de Módulos los cuales se componen de código, consultar, ejecutar, eliminar, estado, ingresar, modificar, nombre, usuario.

**Figura 43. Método ObtenerMóduloPorId**



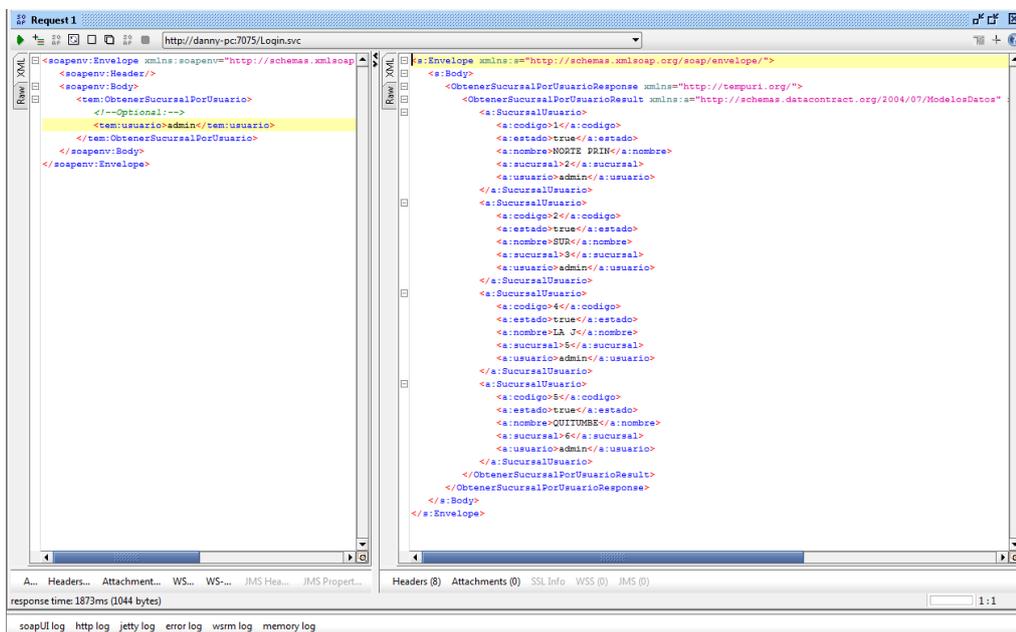
**Fuente: Proyecto capa de servicios de Punto Classic**

En este método se utiliza en la capa de negocios la clase BILLogin el método ConsultarMódulos este a su vez utiliza de la capa de acceso a datos la clase ALogin el métodos ConsultarMódulos.

En la base de datos se invocará el procedimiento almacenado sp\_AdministrarMódulo, en este sp se consulta la tabla módulo.

El método ObtenerSucursalPorUsuario se encarga de obtener las sucursales por medio de la identificación, se tiene como entrada el usuario, como salida se tiene una lista de SucursalUsuario los cuales se componen de código, nombre, estado, sucursal, usuario.

**Figura 44. Método ObtenerSucursalPorUsuario**



**Fuente: Proyecto capa de servicios de Punto Classic**

En este método se utiliza en la capa de negocios la clase BllLogin el método ObtenerSucursalPorUsuario este a su vez utiliza de la capa de acceso a datos la clase ASucursalUsuario el método ConsultarSucursalUsuario.

En la base de datos se invocará el procedimiento almacenado sp\_AdministrarSucursalUsuario, en este sp se consulta la tabla usuario\_sucursal.

### 3.2.2. Proceso de creación de usuario

Este proceso se encarga de la creación de nuevos usuario, asignación de módulos y de permitir el acceso sobre diferentes sucursales.

Para realizar este proceso se invocan los siguientes métodos del servicio web login, para la creación del usuario, asignar módulos se utiliza el método AdministrarIngresoUsuario, para asignar permisos a diferentes sucursales se utiliza el método AsignarAcceso.

**Figura 45. Proceso de ingreso al sistema**

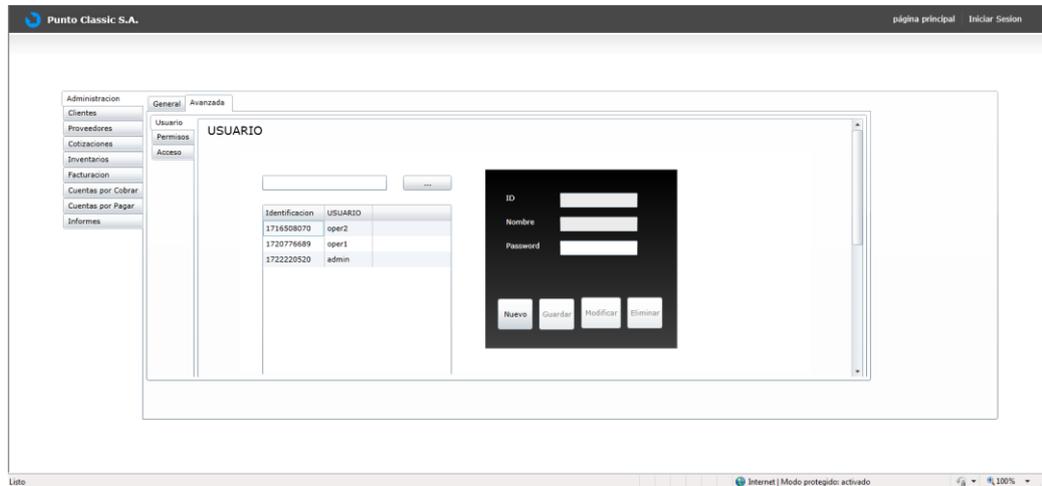


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas CreacionUsuario.xaml, permisos.xaml, acceso.xaml.

En la pantalla CreacionUsuario.xaml es donde se crea el usuario ingresando el nombre y el password como se muestra en la figura 46.

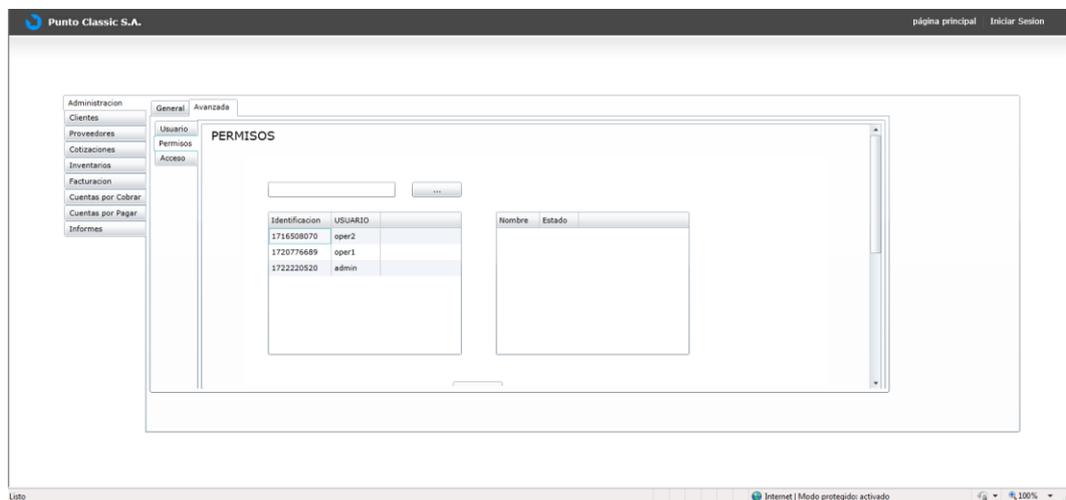
**Figura 46. Pantalla de creacion de usuario**



**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la pantalla permisos.xaml seleccionamos los módulos a los cuales tiene acceso el usuario.

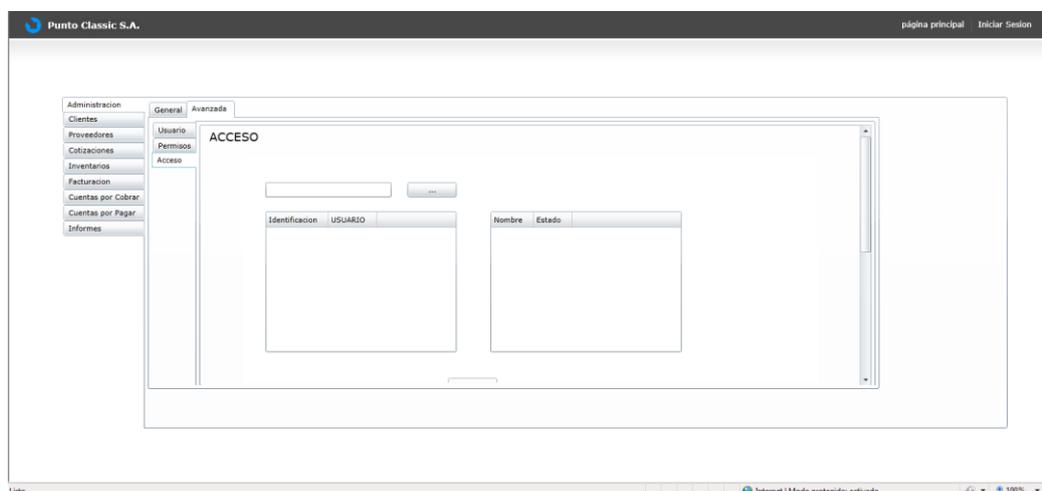
**Figura 47. Pantalla de permisos**



**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la pantalla accesos.xaml seleccionamos las sucursales a las cuales tiene acceso el usuario.

**Figura 48. Pantalla de accesos**

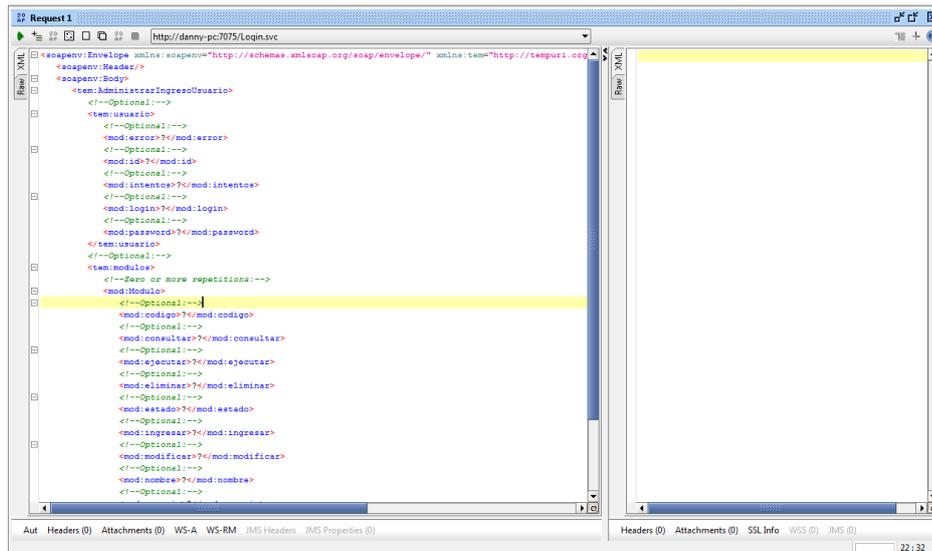


**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la capa de servicios se utiliza el servicio <http://localhost:7075/Login.svc> como muestra la figura 38, en el cual intervienen los métodos AdministrarIngresoUsuario, AsignarAcceso.

El método AdministrarIngresoUsuario se encarga ingresar usuarios y de asignar los módulos que tiene acceso, se tiene como entrada el los datos de usuario como id, login, password, intentos también se tiene los datos de los módulos como código, nombre, usuario, estado y de salida se tiene código, mensaje, devuelve código 0, mensaje “ok” en caso de ser correctas.

**Figura 49. Método AdministrarIngresoUsuario**



**Fuente: Proyecto Capa de Servicios de Punto Classic**

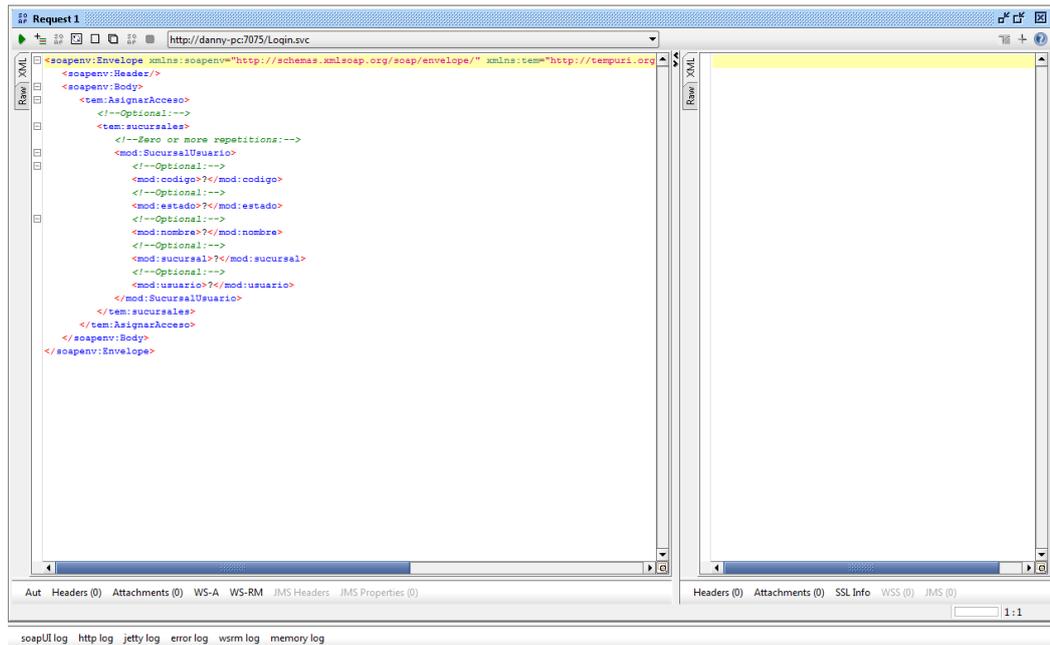
En este método se utiliza en la capa de negocios la clase `BillLogin`, el método `CrearUsuario`, `AdministrarUsuario` y `AdministrarMódulo`.

En la capa de acceso a datos utilizamos la clase `ALogin`, `ASucursalUsuario`, los métodos `AdministrarUsuario`, `AdministrarMódulo`, `AdministrarSucursalUsuario`.

En la base de datos se utilizan los procedimientos almacenados `sp_AdministrarMódulo`, `sp_AdministrarUsuario`, `sp_AdministrarSucursalUsuario` y las tablas afectadas son `login`, `usuario_sucursal`, `módulo`.

El método `AsignarAcceso` se encarga de asignar las sucursales que tiene acceso, se tiene como entrada `código`, `nombre`, `estado`, `sucursal`, `usuario` y de salida se tiene `código`, `mensaje`, devuelve `código 0`, `mensaje "ok"` en caso de ser correctas.

**Figura 50. Método AsignarAcceso**



**Fuente: Proyecto Capa de Servicios de Punto Classic**

En este método se utiliza en la capa de negocios la clase `BillLogin`, el método `AsignarAcceso`.

En la capa de acceso a datos se utiliza la clase `ALogin`, el método `ActualizarSucursalPorUsuario`.

En la base de datos se utiliza el procedimiento almacenado `sp_ActualizarSucursalPorUsuario` y las tablas afectada es `usuario_sucursal`.

### **3.3. Módulo de clientes**

Este módulo se encarga del siguiente proceso, el ingreso de los clientes al sistema incluyendo los contactos que tiene cada cliente.

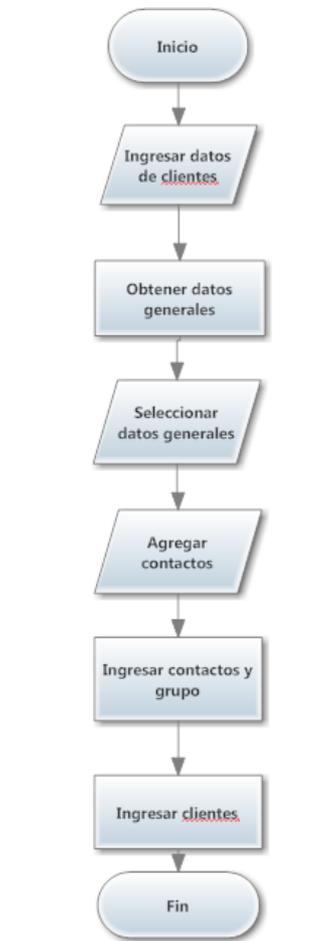
#### **3.3.1. Proceso de ingreso de clientes**

Este proceso se encarga de ingresar los datos de los cliente una vez ingresado se agregan los contactos, se cargan los datos de monedas, ciudades, países, tipos de clientes y estados de clientes.

Para realizar este proceso se invoca los siguientes métodos del servicio web `Cliente`, para consultar los datos generales se invocan los métodos `ConsultarPais`, `ConsultarMoneda`, `ConsultarCiudad`, `ConsultarTipoCliente`, `ConsultarEstadoCliente`

para realizar el ingreso de los contactos se utiliza el método AdministrarContacto, AdministrarGrupo y para ingresar el cliente se utiliza el método AdministrarCliente.

**Figura 51. Proceso de ingreso de clientes al sistema**

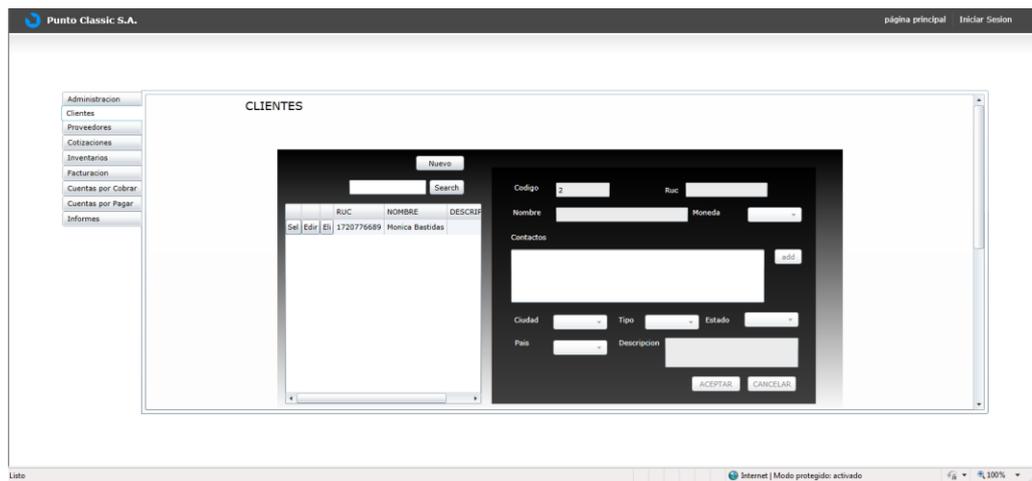


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas MenuCliente.xaml, AgregarContacto.xaml.

En la pantalla MenuCliente.xaml es donde se crean los clientes ingresando los datos del cliente como se muestra en la figura 52.

**Figura 52. Pantalla de clientes**



**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la pantalla `AgregarContacto.xaml` se crean los contactos asociados al cliente actual ingresando los datos del contacto como se muestra en la figura 53.

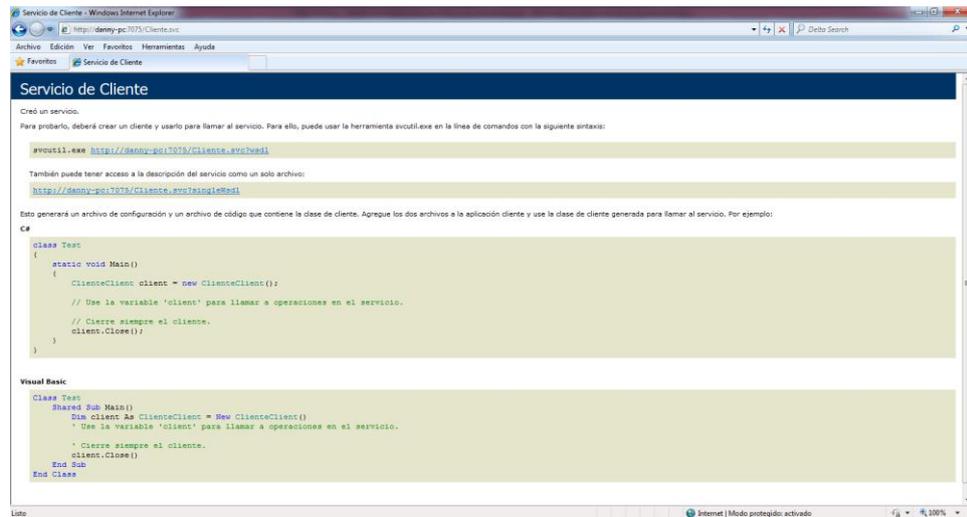
**Figura 53. Pantalla de contactos**



**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la capa de servicios se utiliza el servicio `http://localhost:7075/Ciente.svc` como muestra la figura 54, en el cual intervienen los métodos `ConsultarCliente`, `ConsultarEstadoCliente`, `ConsultarTipoCliente`, `ConsultarPais`, `ConsultarCiudad`, `ConsultarMoneda`, `AdministrarCliente`, `AdministrarGrupo`, `AdministrarContacto`.

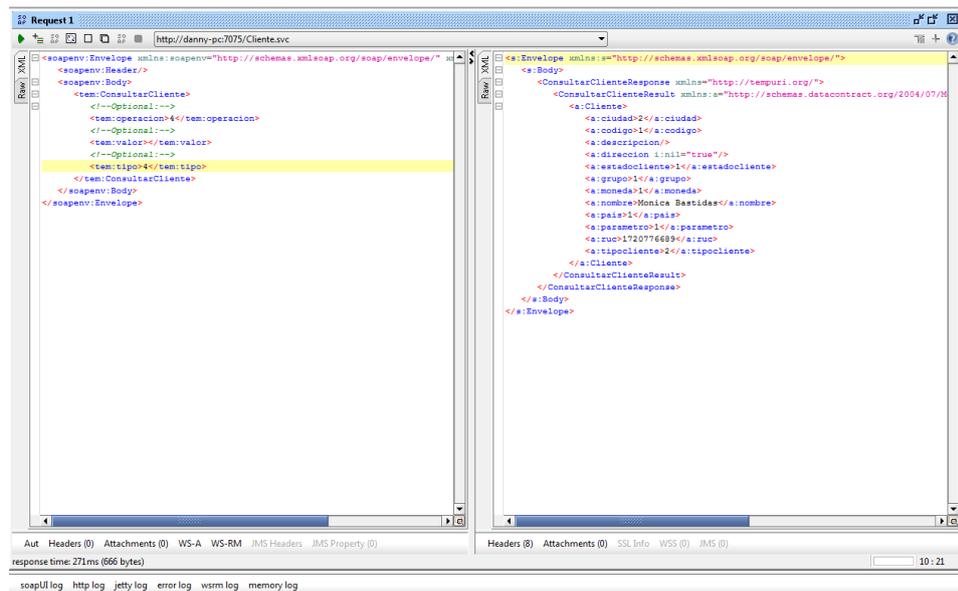
**Figura 54. Servicio de clientes**



**Fuente: Proyecto Capa de Servicios de Punto Classic**

El método ConsultarCliente se encarga consultar los datos del cliente.

**Figura 55. Método ConsultarCliente**



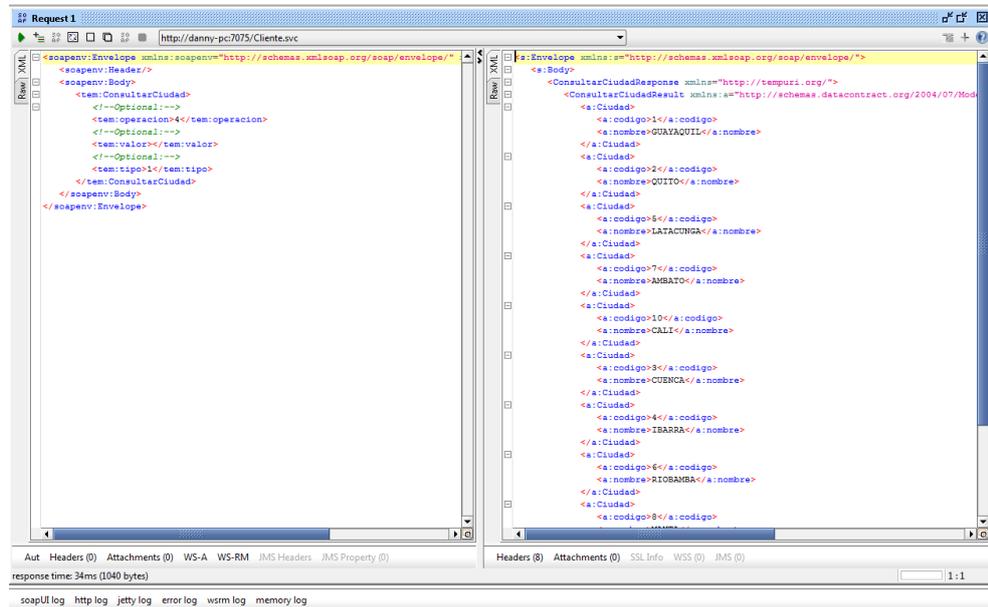
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllCliente, el método AdministrarCliente, que a su vez utiliza en la capa de acceso a datos la clase ACliente, el método CrearCliente.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarCliente que a su vez utiliza la tabla CLIENTE

El método ConsultarCiudad se encarga de consultar las ciudades que están creadas para el sistema.

**Figura 56. Método ConsultarCiudad**



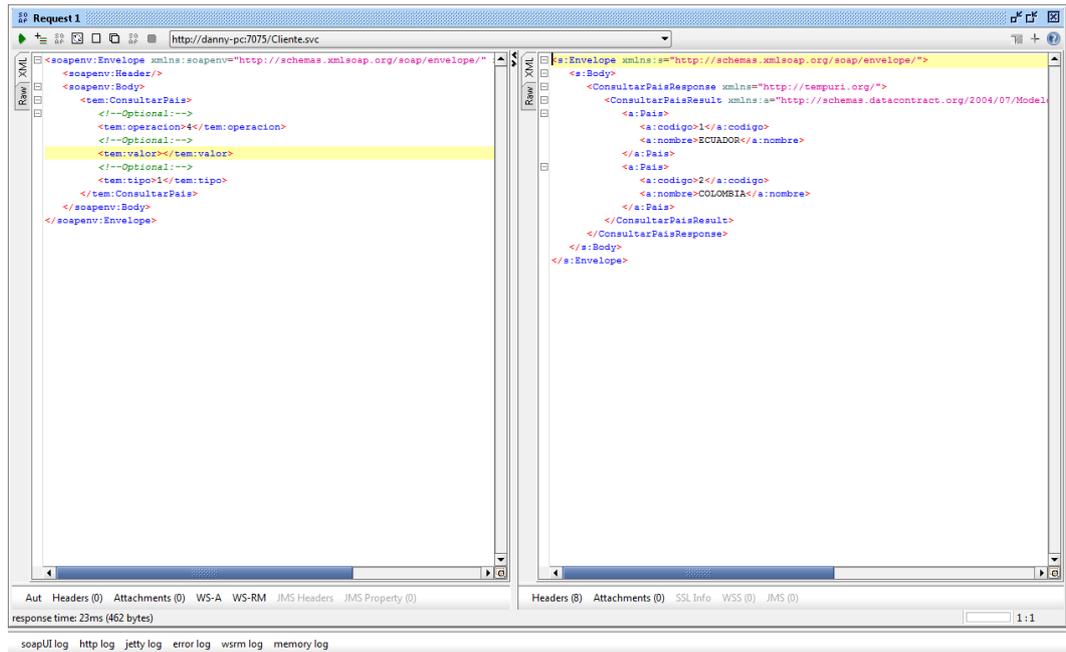
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BIICliente, el método AdministrarCiudadD que a su vez utiliza en la capa de acceso a datos la clase ACiudad, el método CrearCliente.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarCiudad que a su vez utiliza la tabla ciudad.

El método ConsultarPais se encarga de consultar los países que están creados para el sistema.

**Figura 57. Método ConsultarPais**



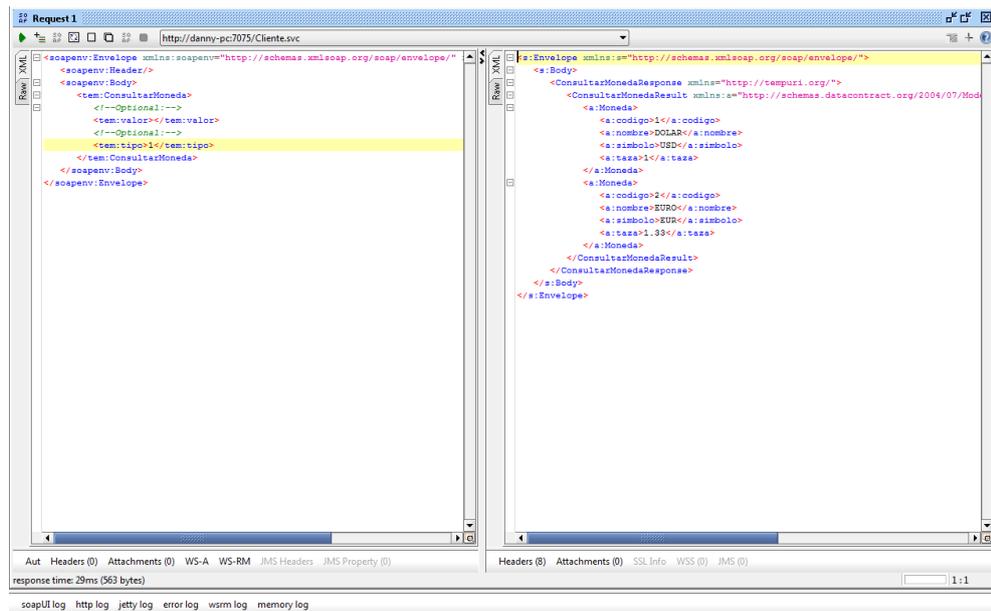
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllCliente, el método AdministrarPais que a su vez utiliza en la capa de acceso a datos la clase APais, el método CrearPais.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarPais que a su vez utiliza la tabla PAIS.

El método ConsultarMoneda se encarga de consultar las monedas creadas para manejar el sistema.

**Figura 58. Método ConsultarMoneda**



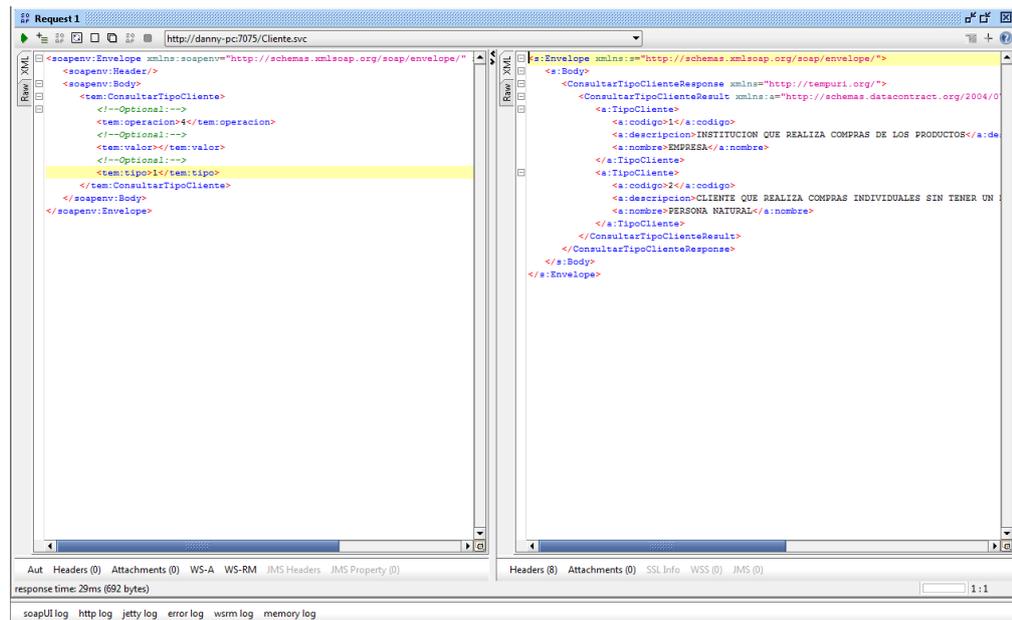
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllCliente, el método AdministrarPais que a su vez utiliza en la capa de acceso a datos la clase APais, el método CrearPais.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarPais que a su vez utiliza la tabla país.

El método ConsultarTipoCliente se encarga de consultar los tipos de clientes que están parametrizados en el sistema.

**Figura 59. Método ConsultarTipoCliente**



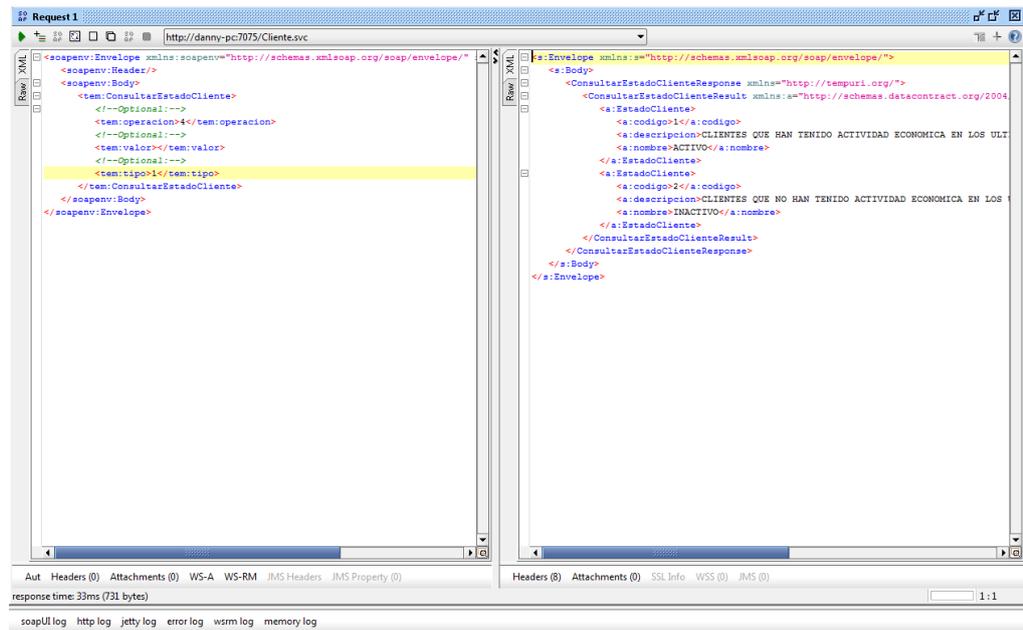
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllCliente, el método AdministrarTipoCliente que a su vez utiliza en la capa de acceso a datos la clase ATipoCliente, el método CrearTipoCliente.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarTipoCliente que a su vez utiliza la tabla TIPO\_CLIENTE.

El método ConsultarEstadoCliente se encarga de consultar los estados de clientes que se encuentran parametrizados en el sistema.

**Figura 60. Método ConsultarEstadoCliente**



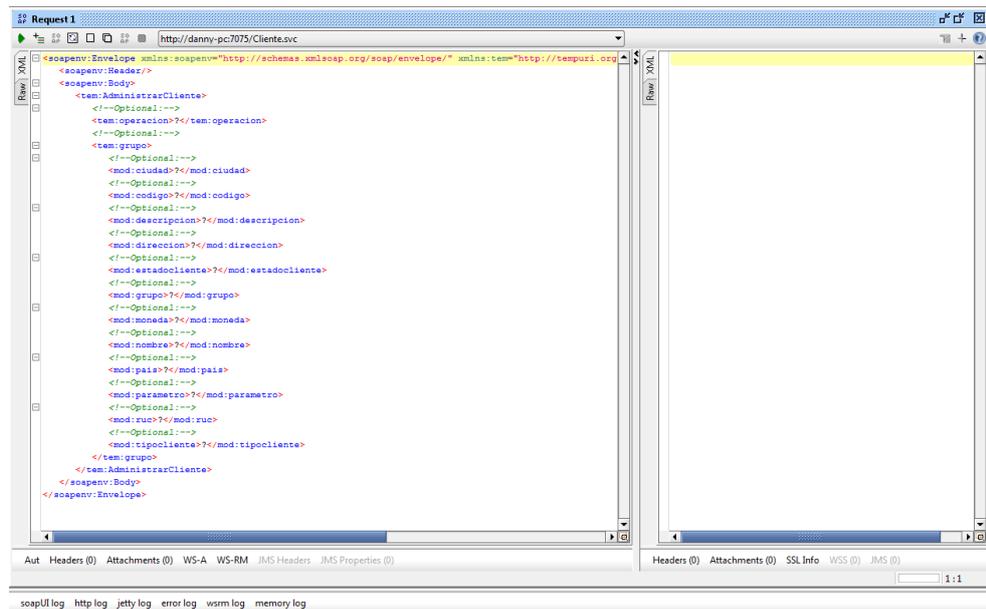
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllCliente, el método AdministrarEstadoCliente que a su vez utiliza en la capa de acceso a datos la clase AEstadoCliente, el método CrearEstadoCliente.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarEstadoCliente que a su vez utiliza la tabla estado\_cliente.

El método AdministrarCliente se encarga de crear los clientes en el sistema.

**Figura 61. Método AdministrarCliente**



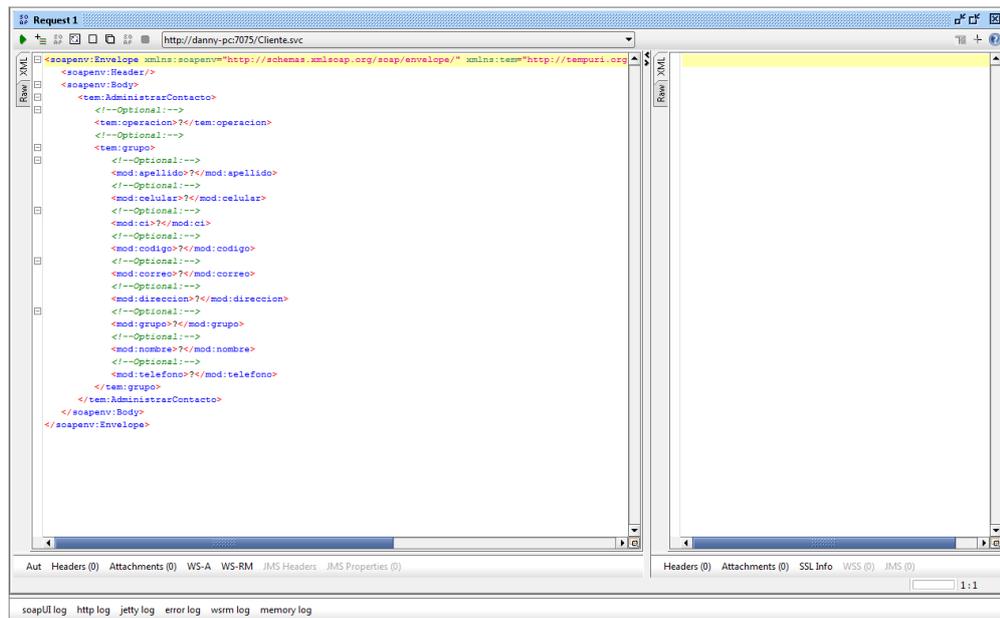
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase `BIICliente`, el método `AdministrarCliente` que a su vez utiliza en la capa de acceso a datos la clase `ACliente`, el método `CrearCliente`.

En la base de datos se utiliza el procedimiento almacenado `sp_AdministrarCliente` que a su vez utiliza la tabla `cliente`.

El método `AdministrarContacto` se encarga de crear los contactos que se encuentran asociados a los clientes.

**Figura 62. Método AdministrarContacto**



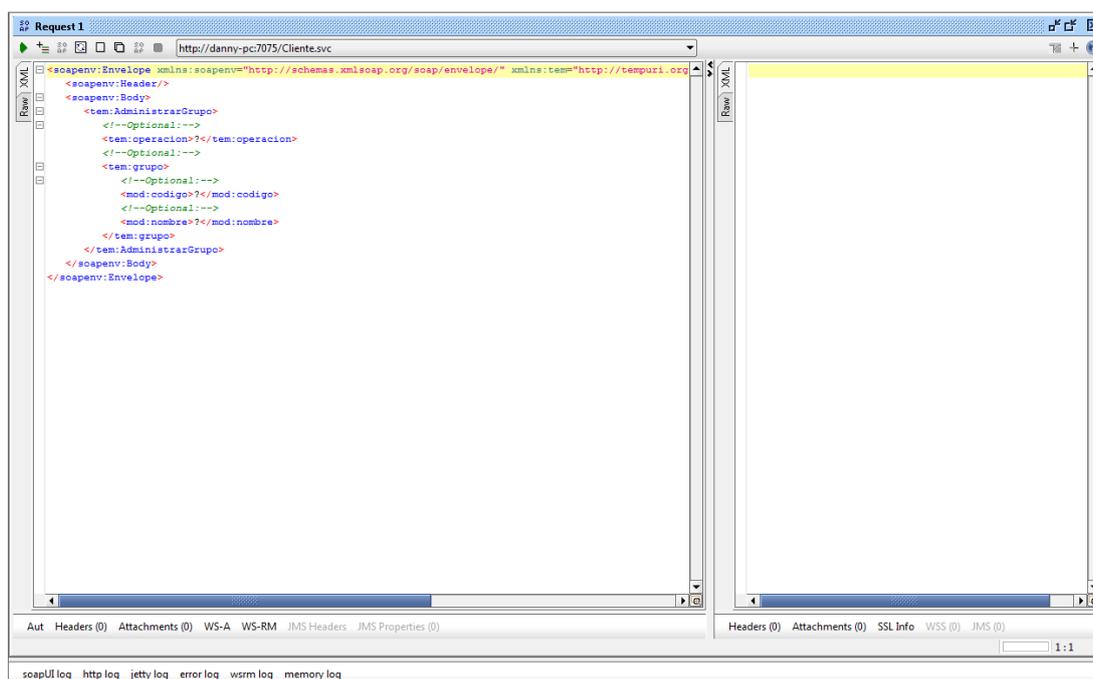
**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BIICliente, el método AdministrarContacto que a su vez utiliza en la capa de acceso a datos la clase AContacto, el método CrearContacto.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarContacto que a su vez utiliza la tabla contacto.

El método AdministrarGrupo se encarga de crear los grupos de contactos que tiene cada cliente.

**Figura 63. Método AdministrarGrupo**



**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase `BIICliente`, el método `AdministrarGrupo` que a su vez utiliza en la capa de acceso a datos la clase `AGrupo`, el método `CrearGrupo`.

En la base de datos se utiliza el procedimiento almacenado `sp_AdministrarGrupo` que a su vez utiliza la tabla `grupo`.

### 3.4. Módulo de proveedores

Este módulo se encarga del siguiente proceso, el ingreso de los proveedores al sistema incluyendo los contactos que tiene cada proveedor.

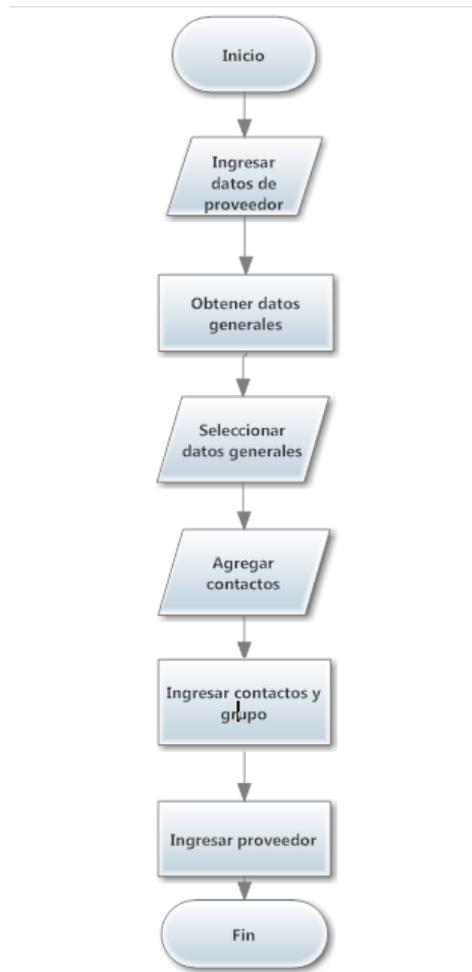
#### 3.4.1. Proceso de ingreso de proveedores

Este proceso se encarga de ingresar los datos de los proveedores una vez ingresado se agregan los contactos, se cargan los datos de monedas, ciudades, países, tipos de proveedores y estados de proveedores.

Para realizar este proceso se invoca los siguientes métodos del servicio web `Proveedor`, `Cliente`, para consultar los datos generales se invocan los métodos `ConsultarPais`, `ConsultarCiudad`, `ConsultarTipoCliente`, `ConsultarEstadoCliente` para realizar el ingreso de los contactos se utiliza el método `AdministrarContacto`,

AdministrarGrupo y para ingresar el proveedor se utiliza el método AdministrarProveedor.

**Figura 64. Proceso de ingreso de proveedores al sistema**

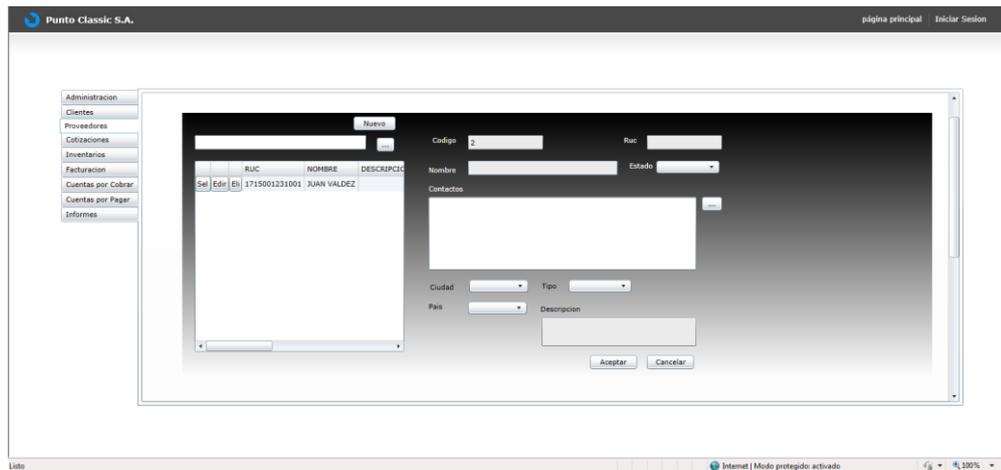


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas MenuCliente.xaml, AgregarContacto.xaml.

En la pantalla MenuProveedor.xaml es donde se crean los clientes ingresando los datos del proveedor como se muestra en la figura 65.

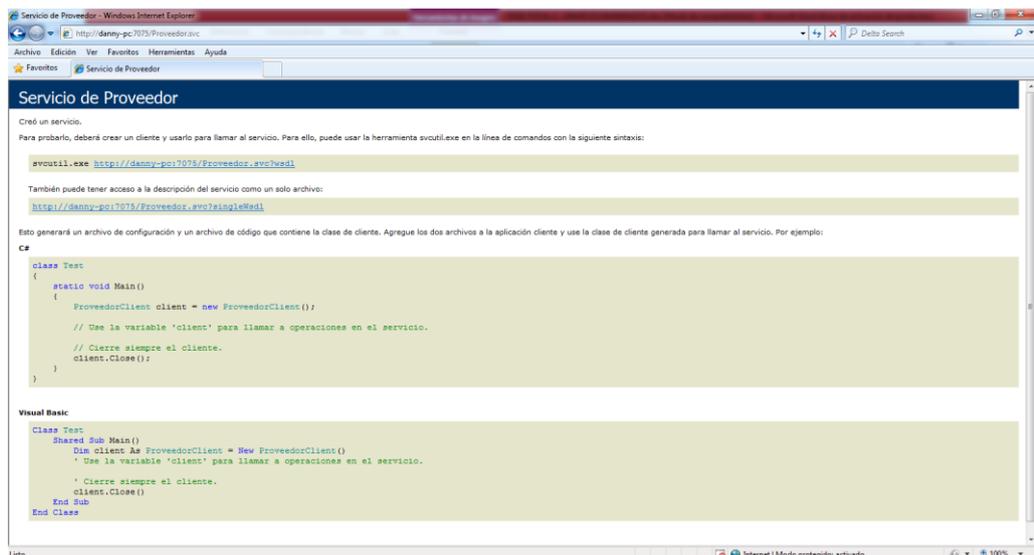
**Figura 65. Pantalla de clientes**



**Fuente: Proyecto Capa de Presentacion de Punto Classic**

En la capa de servicios se utiliza el servicio <http://localhost:7075/Cliente.svc> como muestra la figura 66, en el cual intervienen los métodos ConsultarCliente, ConsultarEstadoCliente, ConsultarTipoCliente, ConsultarPais, ConsultarCiudad, ConsultarMoneda, AdministrarGrupo, AdministrarContacto y el servicio <http://localhost:7075/Proveedor.svc> en el cual interviene el método AdministrarProveedor como muestra la figura 66.

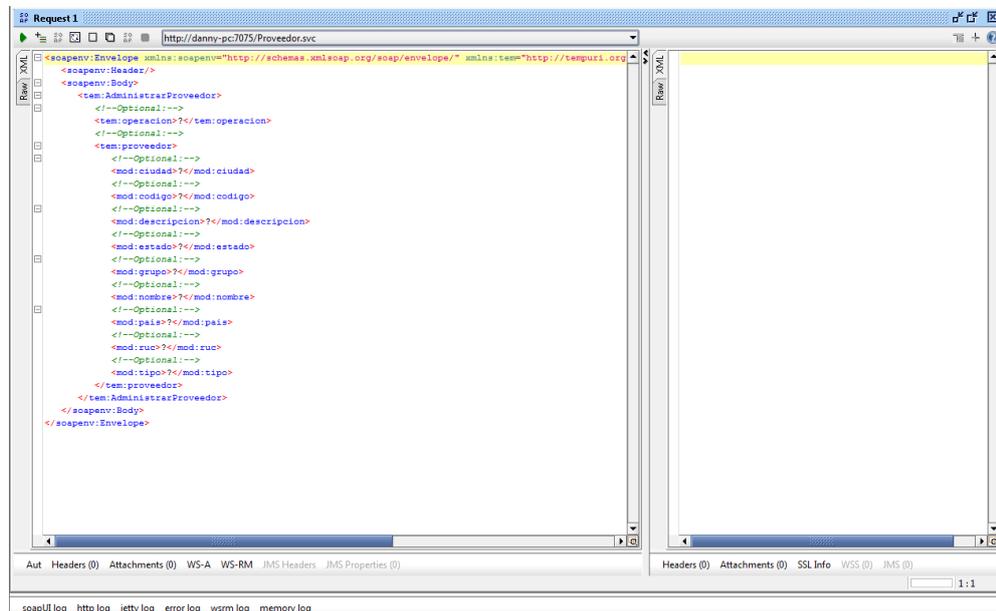
**Figura 66. Servicio de proveedores**



**Fuente: Proyecto Capa de Servicios de Punto Classic**

El método AdministrarProveedor se encarga de crear los proveedores que tiene la empresa.

**Figura 67. Método AdministrarProveedor**



**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BilCliente, el método AdministrarProveedor que a su vez utiliza en la capa de acceso a datos la clase AProveedor, el método AdministrarProveedor.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarProveedor que a su vez utiliza la tabla PROVEDOR.

### **3.5. Módulo de inventarios**

Este módulo se encarga del siguiente proceso, el ingreso de productos al sistema, la transferencia de productos entre sucursales, la baja de productos.

#### **3.5.1. Proceso de ingreso de productos**

En este proceso se agregan los productos una vez ingresado se validan contra los valores máximos que pueden tener un producto.

Para realizar este proceso se invoca los siguientes métodos del servicio web Inventario, Login, para consultar las sucursales se invocan los métodos ObtenerSucursalPorUsuario, y para registrar el ingreso de productos se utiliza el método AdministrarIngresoProducto

**Figura 68. Proceso de ingreso de productos al sistema**

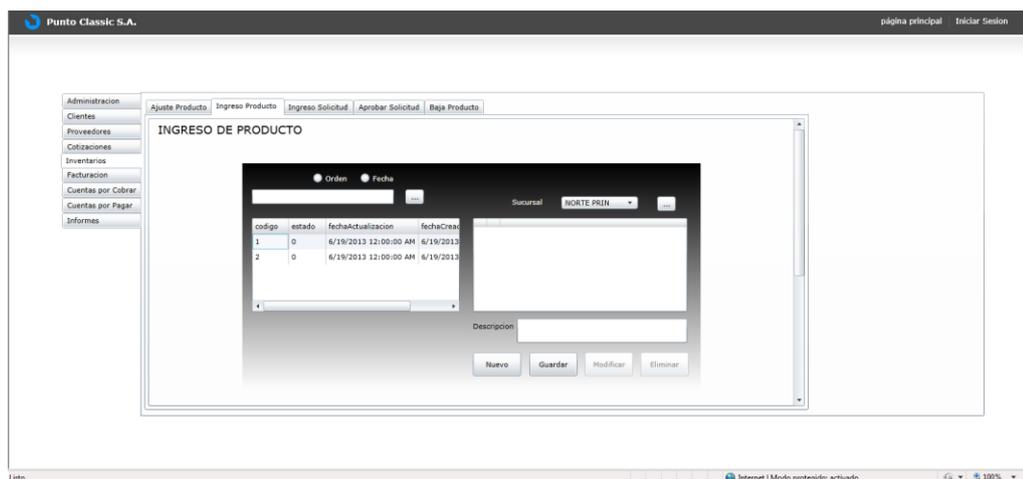


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas IngresoProducto.xaml, AgregarProducto.xaml.

En la pantalla IngresoProducto.xaml es donde se ingresan los productos a una sucursal como se muestra en la figura 69.

**Figura 69. Pantalla de ingreso de productos**



**Fuente: Proyecto capa de presentacion de Punto Classic**

En la pantalla AgregarProducto.xaml es donde se agregan productos al ingreso como se muestra en la figura 70.

**Figura 70. Pantalla de agregar productos**



**Fuente: Proyecto capa de presentacion de Punto Classic**

En la capa de servicios se utiliza el servicio <http://localhost:7075/Inventario.svc> como muestra la figura 71, en el cual intervienen los métodos AdministrarIngresoProducto y el servicio <http://localhost:7075/Login.svc> en el cual interviene el método ObtenerSucursalPorUsuario.

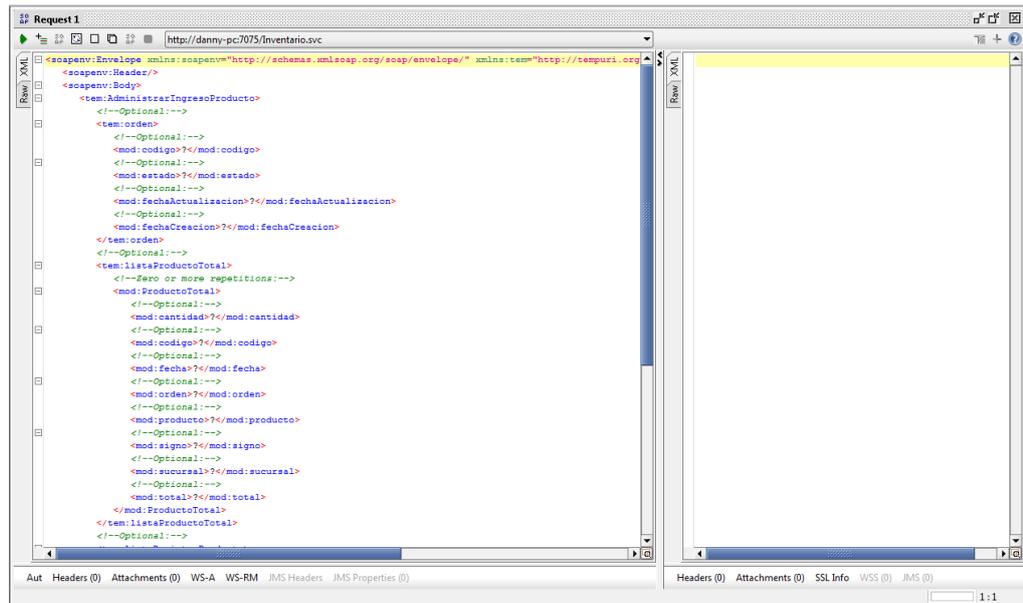
**Figura 71. Servicio de inventarios**



**Fuente: Proyecto capa de servicios de Punto Classic**

El método AdministrarIngresoProducto se encarga de ingresar los productos a una sucursal.

**Figura 72. Método AdministrarIngresoProducto**



**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllInventario, el método AdministrarIngresoProducto, AdministrarOrden, AdministrarRegistroProducto, AdministrarProductoTotal que a su vez utiliza en la capa de acceso a datos la clase AProductoTotal, AOrden, ARegistroProducto, los métodos AdministrarOrden, AdministrarProductoTotal, AdministrarOrden.

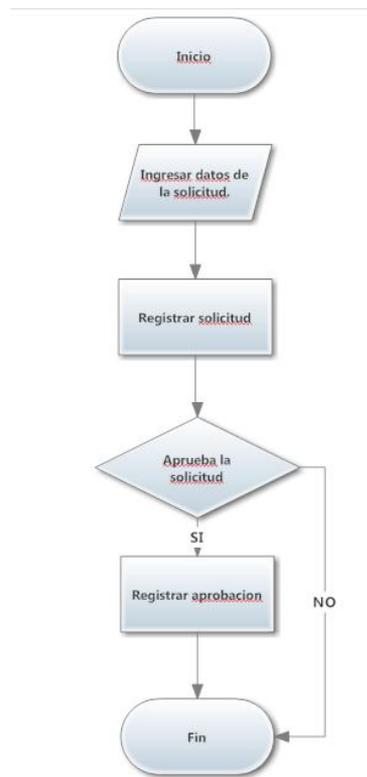
En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarProductoTotal, sp\_AdministrarOrden, sp\_AdministrarRegistroProducto que a su vez utiliza las tablas orden, registro\_producto, producto\_total.

### 3.5.2. Proceso de transferencia de productos

En este proceso se realiza una solicitud de productos de una sucursal a otra y esta solicitud puede ser aprobada o rechazada.

Para realizar este proceso se invoca los siguientes métodos del servicio web Inventario, Login, para consultar las sucursales se invocan los métodos ObtenerSucursalPorUsuario, para realizar la solicitud de productos utiliza el método AdministrarIngresarSolicitud, para realizar la aprobación de la solicitud se utiliza el método AdministrarAprobarSolicitud.

**Figura 73. Proceso transferencia de productos**

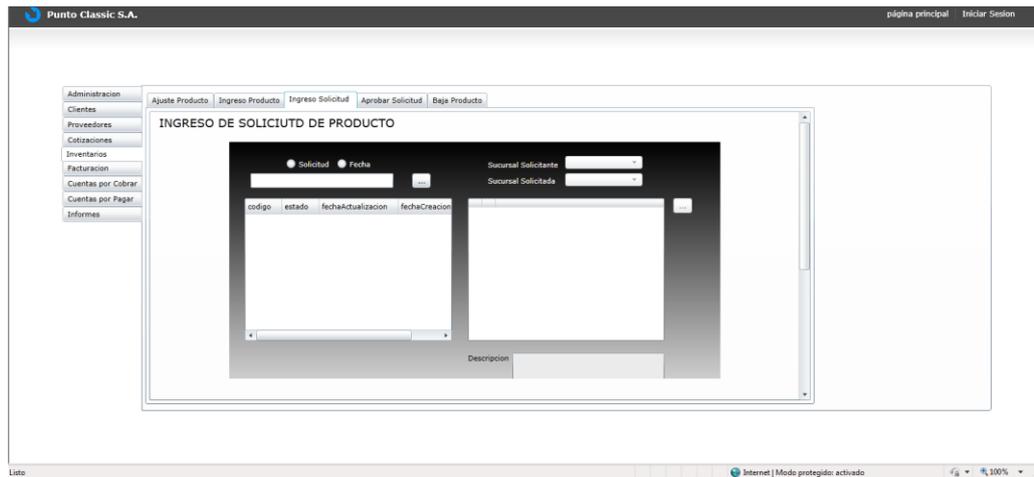


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas IngresoSolicitud.xaml, AprobarSolicitud.xaml, AgregarProducto.xaml.

En la pantalla IngresoSolicitud.xaml se solicitan los productos de una sucursal a otra.

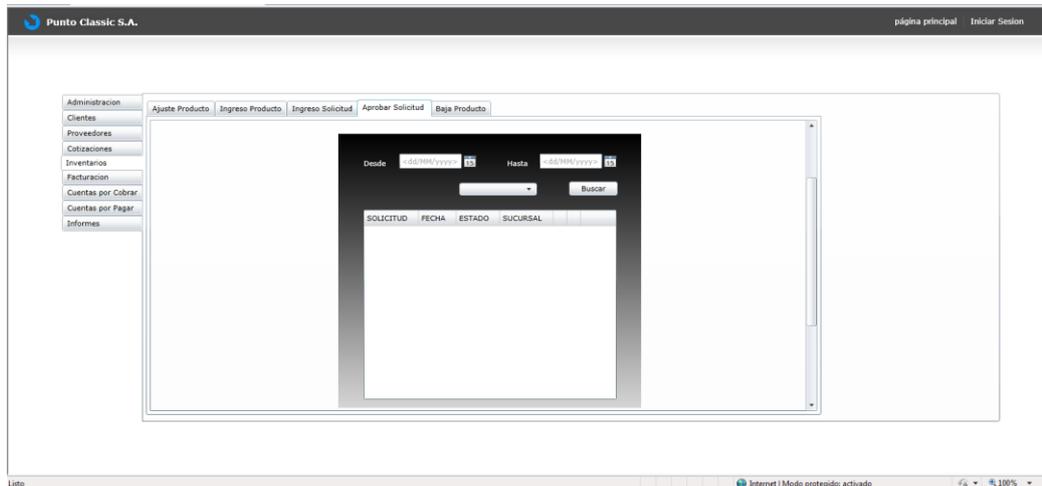
**Figura 74. Pantalla de ingreso de solicitud**



**Fuente: Proyecto capa de presentacion de Punto Classic**

En la pantalla AprobarSolicitud.xaml se aprueban las solicitudes ingresadas para realizar la transferencia.

**Figura 75. Pantalla de aprobacion de solicitud**

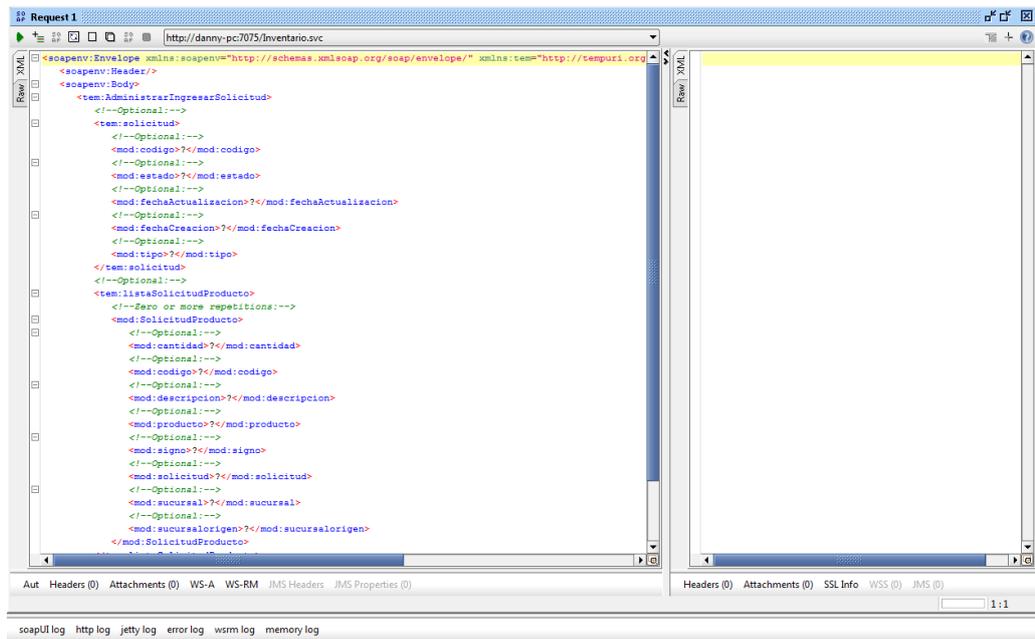


**Fuente: Proyecto capa de presentacion de Punto Classic**

En la capa de servicios se utiliza el servicio <http://localhost:7075/Inventario.svc> como muestra la figura 71, en el cual intervienen los métodos AdministrarIngresoSolicitud y AdministrarAprobarSolicitud y el servicio <http://localhost:7075/Login.svc> en el cual interviene el método ObtenerSucursalPorUsuario.

El método AdministrarIngresarSolicitud se encarga de ingresar las solicitudes de productos.

**Figura 76. Método IngresoSolicitud**



**Fuente: Proyecto capa de servicios de Punto Classic**

El método `AdministrarAprobarSolicitud` se encarga de aprobar las solicitudes de productos ingresadas.

En la capa de negocios se utiliza la clase `BllInventario`, el método `AdministrarIngresoSolilcitud`, `AdministrarIngresoSolicitudProducto.AdministrarIngresoProducto`, que a su vez utiliza en la capa de acceso a datos la clase `ASolicitud`, `ASolicitudProducto`, `ARegistroProducto`, `AProductoTotal`, `AOrden`, los métodos `AdministrarOrden`, `AdministrarProductoTotal`, `AdministrarOrden`, `AdministrarSolicitud`, `AdministrarSolicitudProducto`.

En la base de datos se utiliza el procedimiento almacenado, `sp_AdministrarProductoTotal`, `sp_AdministrarOrden`, `sp_AdministrarRegistroProducto`, `sp_AdministrarSolicitud`, `sp_AdministrarSolicitudProducto` que a su vez utiliza las tablas `orden`, `registro_producto`, `producto_total.`, `solicitud`, `solicitud_producto`.

### 3.5.3. Proceso de baja de productos

En este proceso se realiza para dar de baja productos que en las sucursales que por perdida o robo no se encuentran.

Para realizar este proceso se invoca los siguientes métodos del servicio web Inventario, Login, para consultar las sucursales se invocan los métodos ObtenerSucursalPorUsuario, para realizar la baja de productos utiliza el método AdministrarProcesoBajaProductos.

**Figura 77. Proceso de baja de productos**

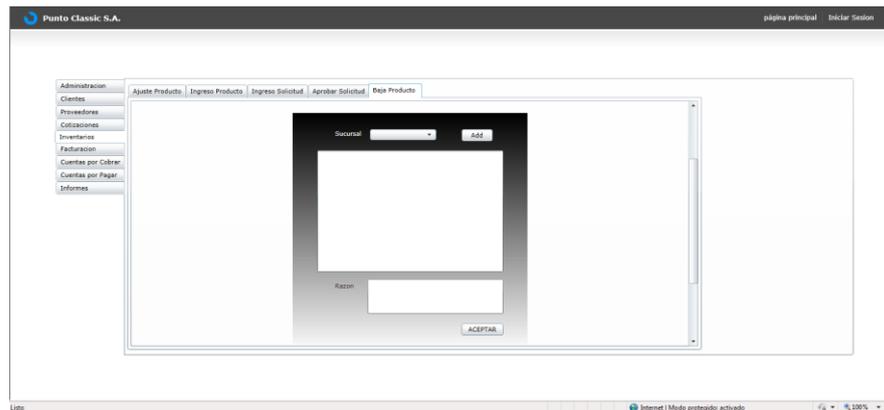


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas BajaProducto.xaml, AgregarProducto.xaml.

En la pantalla BajaProducto.xaml se ingresa la baja de los productos a una sucursal.

**Figura 78. Pantalla de baja de producto**

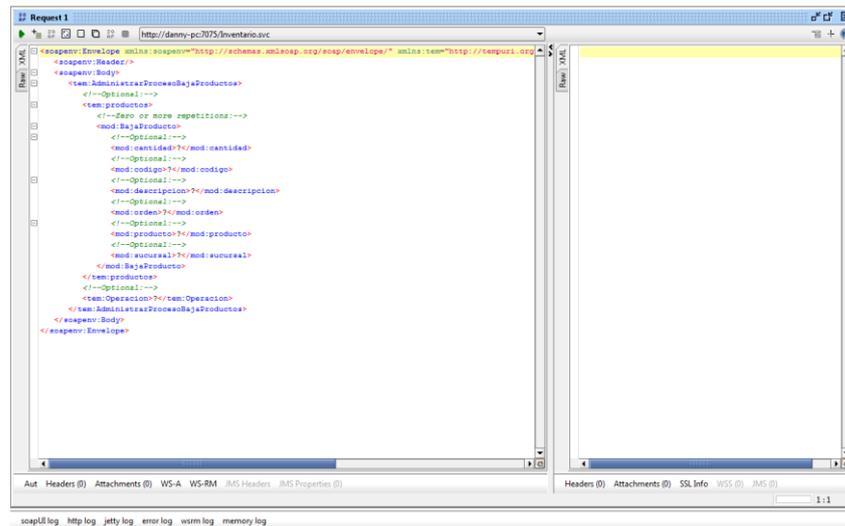


**Elaborado por: Christian Reyes - Daniel Rocha**

En la capa de servicios se utiliza el servicio <http://localhost:7075/Inventario.svc> como muestra la figura 71, en el cual intervienen los métodos AdministrarProcesoBajaProducto y el servicio <http://localhost:7075/Login.svc> en el cual interviene el método ObtenerSucursalPorUsuario.

El método AdministrarProcesoBajaProducto se encarga de ingresar la baja de los productos.

**Figura 79. Método IngresoSolicitud**



**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase BllInventario, el método AdministrarBajaProducto, AdministrarProcesoBajaProducto, AdministrarOrden, AdministrarProductoTotal, AdministrarRegistroProducto, que a su vez utiliza en la

capa de acceso a datos la clase ABajaProducto, AOrden, AProductoTotal, ARegistroProducto el método AdministrarBajaProducto, AdministrarOrden, AdministrarProductoTotal, AdministrarRegistroProducto.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarProductoTotal, sp\_AdministrarOrden, sp\_AdministrarRegistroProducto, sp\_AdministrarBajaProducto que a su vez utiliza las tablas orden, registro\_producto, producto\_total, baja\_producto.

### **3.6. Módulo de Facturacion**

Este módulo se encarga del siguiente proceso, compra de producto, venta de productos.

#### **3.6.1. Proceso de compras**

En este proceso se realiza primero una solicitud de compra, a continuación se realiza una cotización asociada a la solicitud y con la cotización generada se realiza la compra.

Para realizar este proceso se invoca los siguientes métodos del servicio web Facturacion, Cotizacion, para registrar la solicitud de productos se utiliza el método AdministrarIngresoSolicitudCompra, para realizar la cotizacion se utiliza el método AdministrarCotizacion, para realizar la compra se utiliza el método AdministrarIngresarCompra

**Figura 80. Proceso de compra de productos**

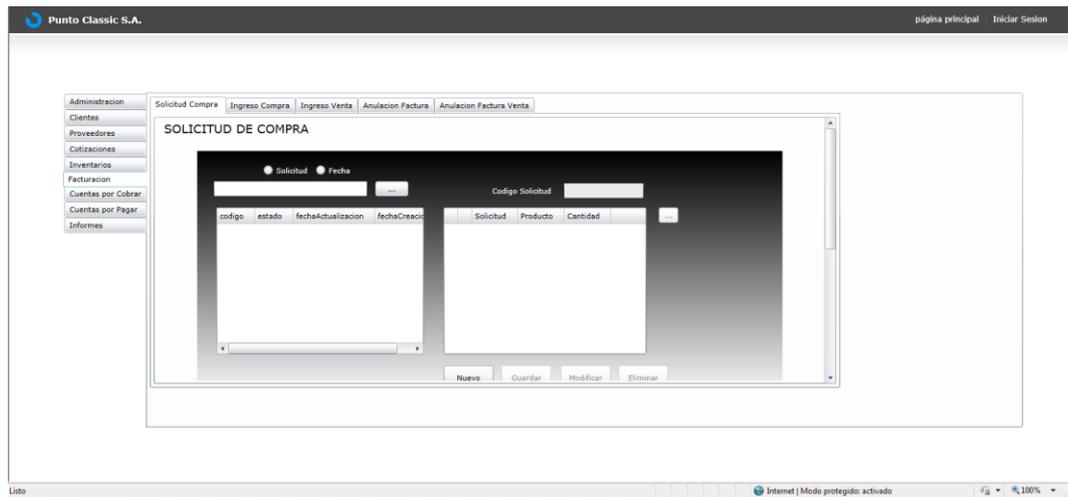


**Elaborado por: Christian Reyes - Daniel Rocha**

En el desarrollo de este proceso intervienen los siguientes elementos en la capa de presentación las pantallas `SolicitudCompra.xaml`, `CotizacionIngreso.xaml`, `Compra.xaml`.

En la pantalla `SolicitudCompra.xaml` se ingresa la solicitud de los productos a comprar.

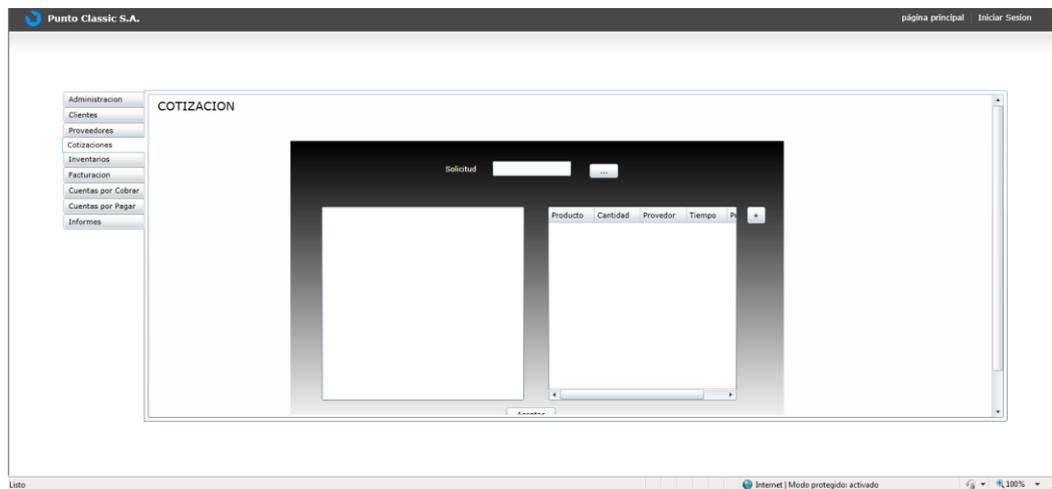
**Figura 81. Pantalla de solicitud de compra**



**Elaborado por: Christian Reyes - Daniel Rocha**

En la pantalla CotizacionIngreso.xaml es donde se realiza la cotizacion de los productos a comprar como se muestra en la figura 82.

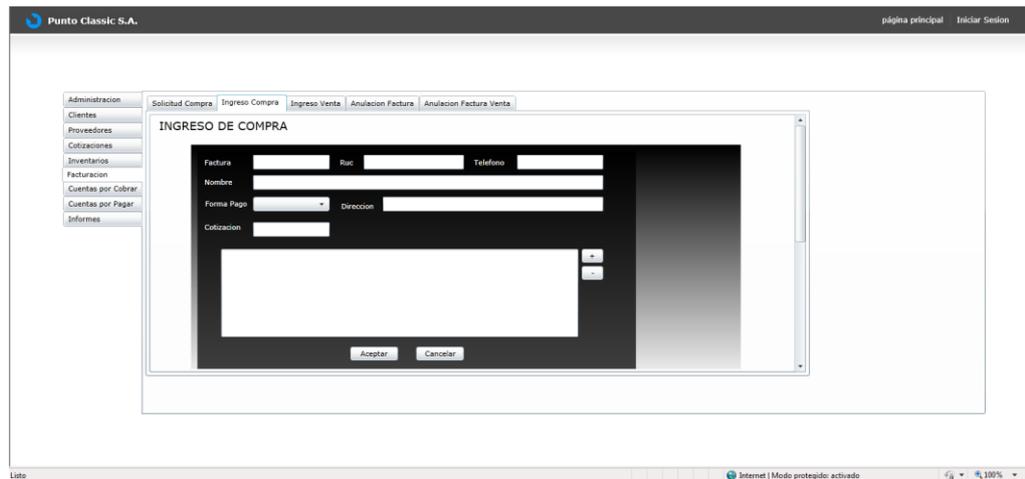
**Figura 82. Pantalla de cotizacion**



**Elaborado por: Christian Reyes - Daniel Rocha**

En la pantalla IngresoCompra.xaml se realiza la cotizacion de los productos a comprar.

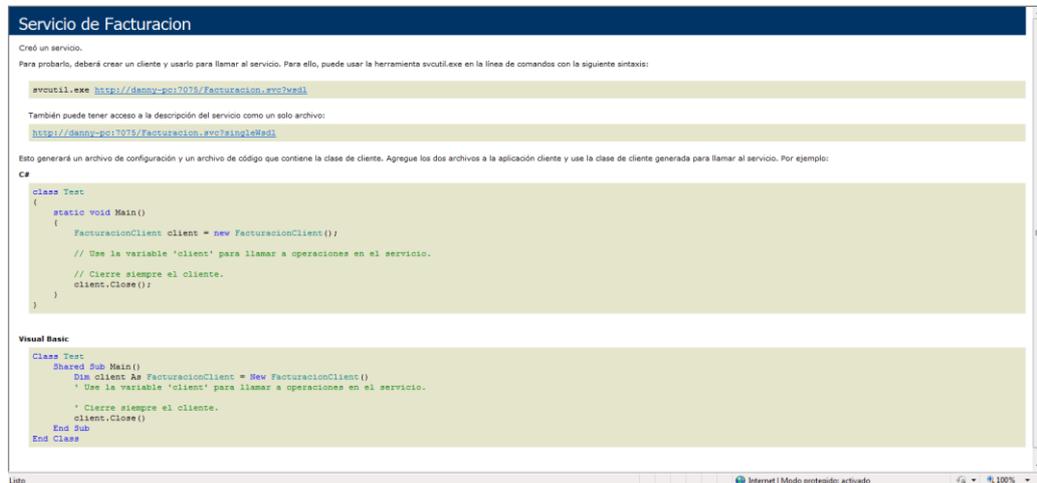
**Figura 83. Pantalla de compra**



**Elaborado por: Christian Reyes - Daniel Rocha**

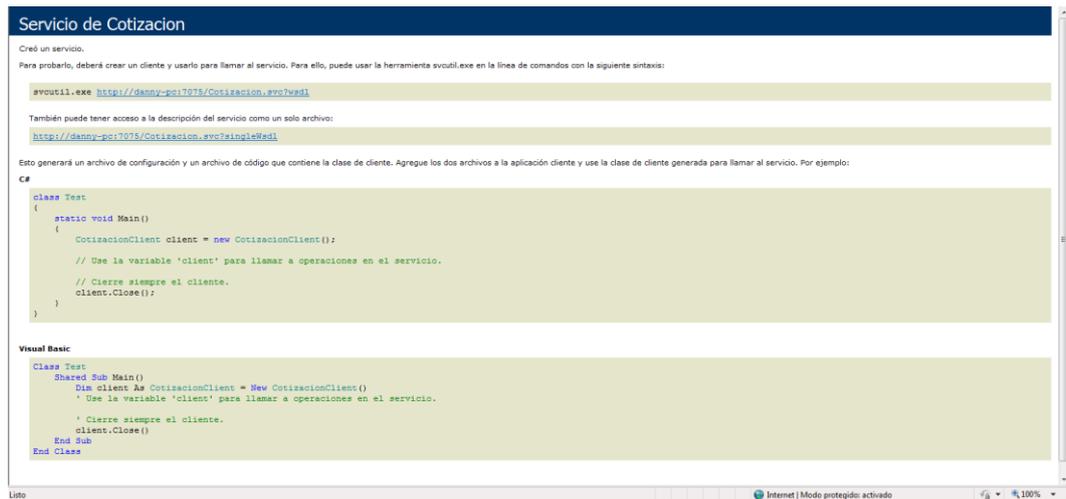
En la capa de servicios se utiliza los servicios <http://localhost:7075/Facturacion.svc>, <http://localhost:7075/Cotizacion.svc> como muestra las figuras 84 y 85, en el cual intervienen los métodos AdministrarIngresoSolicitudCompra, AdministrarIngresoCotizacion, AdministrarIngresoCompra.

**Figura 84. Servicio de facturacion**



**Fuente: Proyecto capa de servicios de Punto Classic**

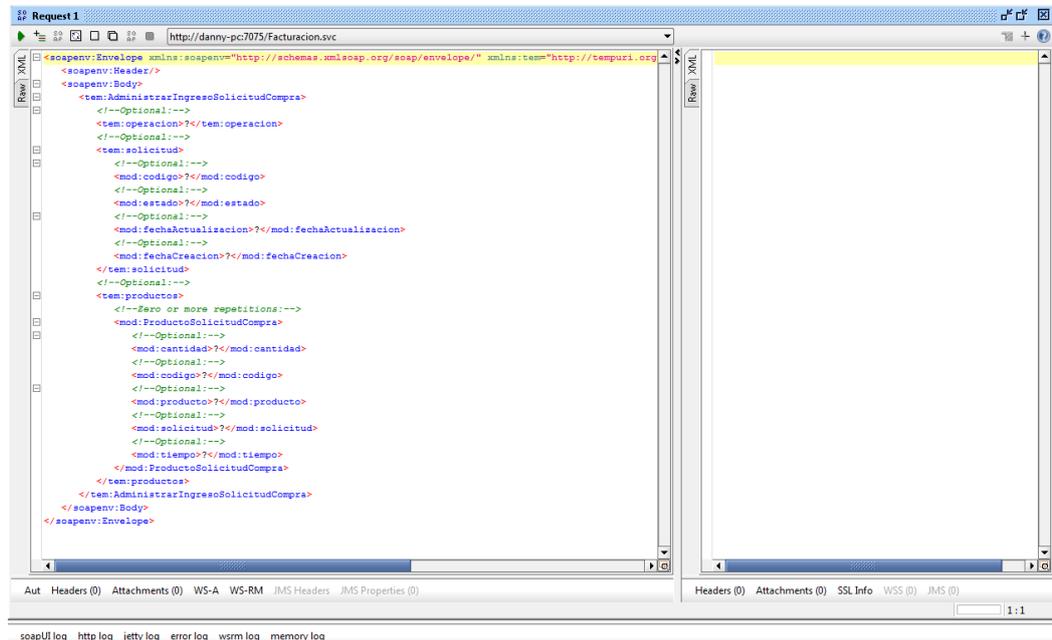
**Figura 85. Servicio de cotizacion**



**Fuente: Proyecto capa de servicios de Punto Classic**

El método `AdministrarIngresoSolicitudCompra` se encarga de ingresar la solicitud de la compra de productos.

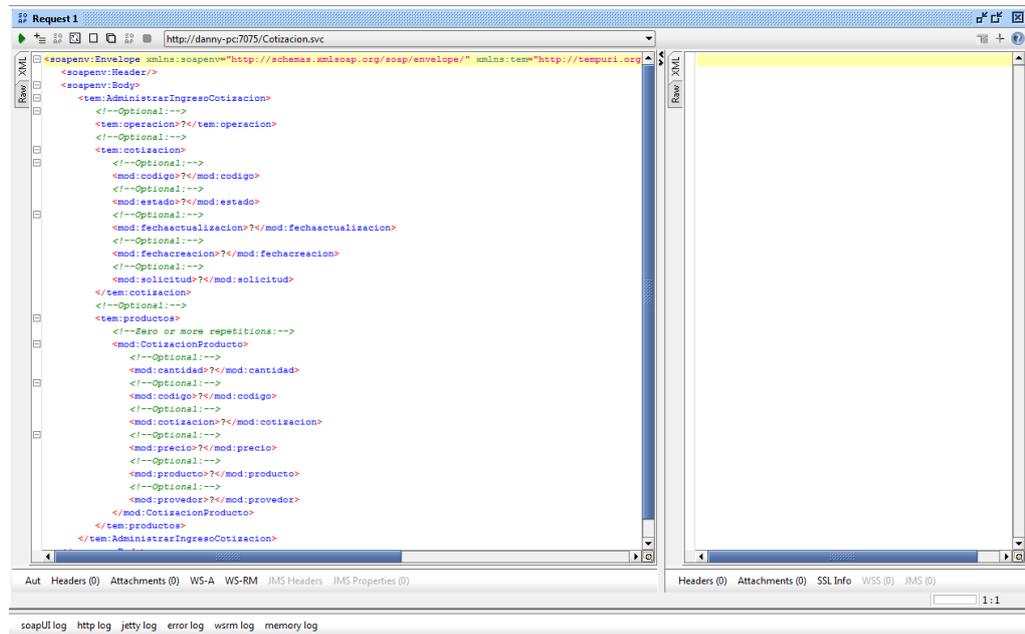
**Figura 86. Método AdministrarIngresoSolicitudCompra**



**Fuente: Proyecto capa de servicios de Punto Classic**

El método `AdministrarIngresoCotizacion` se encarga de realizar la cotización de productos.

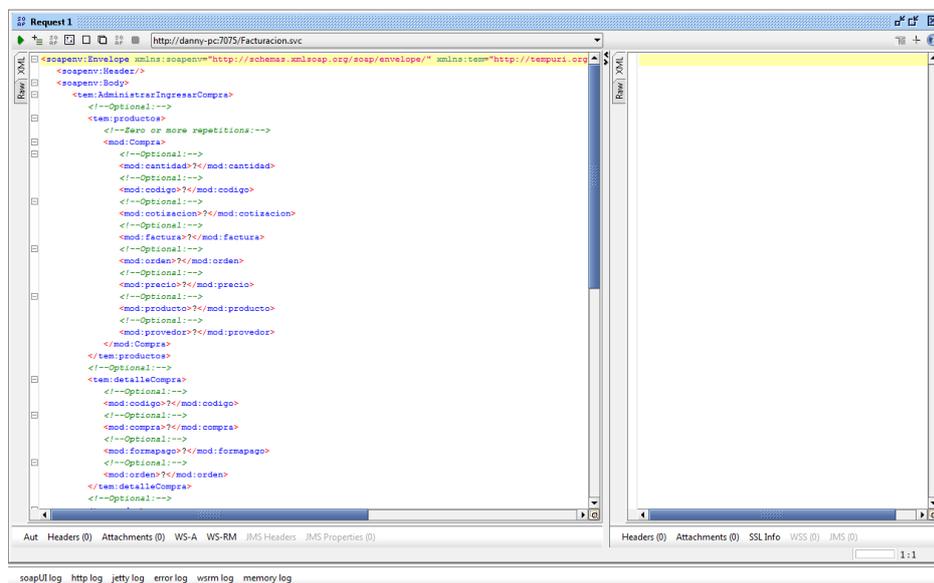
**Figura 87. Método AdministrarIngresoCotizacion**



**Fuente: Proyecto capa de servicios de Punto Classic**

El método AdministrarIngresoCompra se encarga de realizar la compra de productos.

**Figura 88. Método AdministrarIngresoCompra**



**Fuente: Proyecto capa de servicios de Punto Classic**

En la capa de negocios se utiliza la clase `BIFacturacion`, `BIICotizacion`, el método `AdministrarIngresoSolicitudProducto`, `AdministrarIngresoCotizacion`, `AdministrarOrden`, `AdministrarProductoTotal`, `AdministrarRegistroProducto`,

AdministrarEgreso, AdministrarCuentasPorPagar, que a su vez utiliza en la capa de acceso a datos la clase AEgreso, AOrden, AProductoTotal, ARegistroProducto, ACuentaPorPagar, AFactura método AdministrarEgreso, AdministrarOrden, AdministrarProductoTotal, AdministrarRegistroProducto, AdministrarCuentaPorPagar, AdministrarFactura.

En la base de datos se utiliza el procedimiento almacenado sp\_AdministrarProductoTotal, sp\_AdministrarOrden, sp\_AdministrarRegistroProducto, sp\_AdministrarEgreso, sp\_AdministrarFactura, sp\_AdministrarCuentasPorPagar, sp\_AdministrarCotizacion, sp\_AdministrarSolicitudCompra que a su vez utiliza las tablas orden, registro\_producto, producto\_total, egreso, cuentas\_por\_pagar, factura, cotizacion, solicitud\_compra.

### **3.7. Desarrollo de la base de datos**

La base de datos del sistema se encuentra formada, por un conjunto de tablas representativas a cada una de las funciones o métodos originales del sistema.

#### **3.7.1. Tablas**

Se mantiene una estructura de entidad relación, en el sistema de Bdd, debido a sus beneficios al momento de acceder a los mismos, beneficios tales como: rapidez en las consultas y subconsultas, mantener un control mayor en las tablas y tratamiento de datos, consultas mucho más estructuradas y fáciles de realizar, entre otros.

Para el almacenamiento de la información se han definido las siguientes tablas:

**Tabla 7. Tablas que componen la Base de Datos**

<i>BAJA_PRODUCTO</i>	<i>Dbo</i>	<i>user table</i>
<i>BODEGA</i>	<i>Dbo</i>	<i>user table</i>
<i>CIUDAD</i>	<i>Dbo</i>	<i>user table</i>
<i>CLIENTE</i>	<i>Dbo</i>	<i>user table</i>
<i>COMPRA</i>	<i>Dbo</i>	<i>user table</i>
<i>CONTACTO</i>	<i>Dbo</i>	<i>user table</i>
<i>COTIZACION</i>	<i>Dbo</i>	<i>user table</i>
<i>COTIZACION_PRODUCTO</i>	<i>Dbo</i>	<i>user table</i>
<i>DETALLE_FORMA_PAGO</i>	<i>Dbo</i>	<i>user table</i>
<i>DETALLECOMPRA</i>	<i>Dbo</i>	<i>user table</i>
<i>DETALLEVENTA</i>	<i>Dbo</i>	<i>user table</i>
<i>DOCUMENTO</i>	<i>Dbo</i>	<i>user table</i>
<i>EGRESOS</i>	<i>Dbo</i>	<i>user table</i>
<i>ESTADO_CLIENTE</i>	<i>Dbo</i>	<i>user table</i>
<i>ESTADO_PROVEEDOR</i>	<i>Dbo</i>	<i>user table</i>
<i>FACTURA</i>	<i>Dbo</i>	<i>user table</i>
<i>FACTURA_REALIZADA</i>	<i>Dbo</i>	<i>user table</i>
<i>FACTURA_VENTA</i>	<i>Dbo</i>	<i>user table</i>
<i>FACTURAINSCRIPCION</i>	<i>Dbo</i>	<i>user table</i>
<i>FORMAPAGO</i>	<i>Dbo</i>	<i>user table</i>
<i>FORMULA</i>	<i>Dbo</i>	<i>user table</i>
<i>GRUPO</i>	<i>Dbo</i>	<i>user table</i>
<i>GRUPO_PARAMETRO_CLIENTE</i>	<i>Dbo</i>	<i>user table</i>
<i>INGRESOS</i>	<i>Dbo</i>	<i>user table</i>
<i>LOGIN</i>	<i>Dbo</i>	<i>user table</i>
<i>MENSAJE</i>	<i>Dbo</i>	<i>user table</i>
<i>MÓDULO</i>	<i>Dbo</i>	<i>user table</i>
<i>MONEDA</i>	<i>Dbo</i>	<i>user table</i>
<i>ORDEN</i>	<i>Dbo</i>	<i>user table</i>
<i>PAIS</i>	<i>Dbo</i>	<i>user table</i>
<i>PARAMETRO_CLIENTE</i>	<i>Dbo</i>	<i>user table</i>
<i>PRODUCTO_PRECIO</i>	<i>Dbo</i>	<i>user table</i>
<i>PRODUCTO_SOLICITUD_COMPRA</i>	<i>Dbo</i>	<i>user table</i>
<i>PRODUCTO_TOTAL</i>	<i>Dbo</i>	<i>user table</i>

<i>PRODUCTOS</i>	<i>Dbo</i>	<i>user table</i>
<i>PROVEEDOR</i>	<i>Dbo</i>	<i>user table</i>
<i>RECTIFICACION</i>	<i>Dbo</i>	<i>user table</i>
<i>REGISTRO_PRODUCION</i>	<i>Dbo</i>	<i>user table</i>
<i>SOLICITUD</i>	<i>Dbo</i>	<i>user table</i>
<i>SOLICITUD_COMPRA</i>	<i>Dbo</i>	<i>user table</i>
<i>SOLICITUD_ORDEN</i>	<i>Dbo</i>	<i>user table</i>
<i>SOLICITUD_PRODUCTO</i>	<i>Dbo</i>	<i>user table</i>
<i>STOCK</i>	<i>Dbo</i>	<i>user table</i>
<i>SUCURSAL</i>	<i>Dbo</i>	<i>user table</i>
<i>SUCURSAL_PRODUCTO</i>	<i>Dbo</i>	<i>user table</i>
<i>TIPO_CLIENTE</i>	<i>Dbo</i>	<i>user table</i>
<i>TIPO_PRODUCTO</i>	<i>Dbo</i>	<i>user table</i>
<i>TIPO_PROVEEDOR</i>	<i>Dbo</i>	<i>user table</i>
<i>TIPO_SUCURSAL</i>	<i>Dbo</i>	<i>user table</i>
<i>VENTA</i>	<i>Dbo</i>	<i>user table</i>

**Fuente:** Base de Datos PuntoClassic

### 3.7.1.1.Descripción de las tablas

A continuación la descripción de las tablas detalladas en la Tabla 3.1.

**BAJA\_PRODUCTO.**- En esta tabla se almacenan los productos que se dan de baja en las sucursales los cuales pueden ser por robo, pérdida, daño etc.

**CIUDAD.**- En esta tabla se almacenan las diferentes ciudades que se registran en el aplicativo.

**CLIENTE.**- En esta tabla se almacenan los clientes que tiene la empresa.

**COMPRA.**-En esta tabla se almacenan cada uno de los ítems que tiene la compra.

**CONTACTO.**- En esta tabla se almacenan los contactos que tiene cada cliente o empresa.

**COTIZACION.**-En esta tabla se almacenan las cotizaciones que son creadas.

**COTIZACION\_PRODUCTO.**- En esta tabla se almacenan los productos que son asociados a las cotizaciones.

**DETALLE\_FORMA\_PAGO.**- En esta tabla se almacena las características de una forma de pago como si es directo o si es a plazos, el interés si es a plazos y el numero de cuotas.

**DETALLECOMPRA.**- En esta tabla se almacena la orden y la forma de pago en la que se va a realizar la compra de insumos de la Textilera.

**DETALLEVENTA.**- En esta tabla se almacena la orden y forma de pago en la que se va a realizar la venta de productos terminados de la Textilera.

**EGRESOS.**- En esta tabla se almacenan el registro de egreso que se producen.

**ESTADO\_CLIENTE.**- En esta tabla se almacenan los estados de los clientes como activo, inactivo.

**ESTADO\_PROVEEDOR.**- En esta tabla se almacenan los estados de los clientes como activo, inactivo.

**FACTURA.**- En esta tabla se almacenan las facturas que son registradas en el proceso de compras.

**FACTURA\_VENTA.**- En esta tabla se almacenan las facturas que son registradas en el proceso de compras.

**FACTURAINSCRIPCION.**- En esta tabla se almacenan el rango de facturas que tiene para emitir.

**FORMAPAGO.**- En esta tabla se almacenan todas las formas de pago que se tienen.

**GRUPO.**- En esta tabla se tiene la agrupación de los contactos.

**INGRESOS.**- En esta tabla se registran los ingresos que se tiene en la empresa.

**LOGIN.**-En esta tabla se encuentran los usuarios del sistema.

**MENSAJE.**- En esta tabla se encuentran la lista de los mensajes que se despliegan en el aplicativo.

**MÓDULO.**- En esta tabla se encuentran los módulos que tiene el aplicativo.

**MONEDA.**- En esta tabla se encuentra las diferentes monedas del aplicativo

**ORDEN.**- Esta tabla almacena las órdenes las cuales se generan por cada acción que afecta a facturación o inventarios.

**PAIS.**- En esta tabla se almacenan los paices

**PARAMETRO\_CLIENTE.**- En esta tabla se tiene parámetros como la sucursal que se define para las compras.

**PRODUCTO\_PRECIO.**- En esta tabla se almacenan los precios de los productos.

**PRODUCTO\_SOLICITUD\_COMPRA.**-En esta tabla se encuentran los productos asociados a las solicitudes de compras.

**PRODUCTO\_TOTAL.**- En esta tabla se encuentra los productos entrantes y salientes de cada sucursal.

**PRODUCTOS.**- En esta tabla se encuentra los productos.

**PROVEEDOR.**-En esta tabla se encuentran registrados los proveedores

**REGISTRO\_PRODUCION.**- En esta tabla se tiene el ingreso y salidas de productos por solicitudes.

**SOLICITUD.**- En esta tabla se encuentran las solicitudes .

**SOLICITUD\_COMPRA.**- En esta tabla se encuentra las solicitudes de compra.

**SOLICITUD\_ORDEN.**- En esta tabla se almacenan la solitud de una orden.

**SOLICITUD\_PRODUCTO** En esta tabla se almacenan la solitud de una producto.

**SUCURSAL.**-En esta tabla se almacenan las sucursales.

**SUCURSAL\_PRODUCTO.**-En esta tabla se almacenan los productos asignados a una sucursal.

**TIPO\_CLIENTE.**-En esta tabla se almacenan los tipos de clientes.

**TIPO\_PRODUCTO.**- En esta tabla se almacenan los tipos de productos.

**TIPO\_PROVEEDOR.**- En esta tabla se almacenan los tipos de proveedores.

**TIPO\_SUCURSAL.**- En esta tabla se almacenan los tipos de sucursales.

**VENTA.**- En esta tabla se almacenan los tipos de clientes.

### **3.7.2. Store procedures**

Producto de la definición de las tablas necesarias para el almacenamiento de la información se han definido los siguientes Store Procedures, que serán invocados dependiendo de la actividad realizada en el sistema.

**Tabla 8. Store Procedures Creados para interactuar con el Sistema**

<i>sp_ActualizarIntentosFallidos</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarBajaProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarCiudad</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarCompra</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarContacto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarCotizacion</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarCotizacionProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarDetalleCompra</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarDetalleFormaPago</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarDetalleVenta</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarEgreso</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarEstadoCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarEstadoProveedor</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarFactura</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarfacturaInscripcion</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarFacturaVenta</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarFormaPago</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarGrupo</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarIngreso</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarOrden</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarPais</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarProductoPrecio</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarProductoSolicitudCompra</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarProductoTotal</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarProveedor</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarRegistroProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarSolicitud</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarSolicitudCompra</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarSolicitudOrden</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarSolicitudProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarSucursal</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarSucursalProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarTipoCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarTipoProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarTipoProveedor</i>	<i>dbo</i>	<i>stored procedure</i>

<i>sp_AdministrarTipoSucursal</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_AdministrarVenta</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarContacto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarCuentasPorCobrar</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarCuentasPorPagar</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarGrupoParametroCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarId</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarLimiteFactura</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarMaximaFactura</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarMoneda</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarParametroCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarProductoDiponible</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarProductoSolicitud</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarSolicitudPorCotizacion</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarSolicitudPorFechas</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ConsultarTotalProducto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_CrearContacto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_CrearEstadoCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_CrearGrupo</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_CrearMoneda</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_creatediagram</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_DesbloquearUsuario</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_EliminarContacto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_EliminarEstadoCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_EliminarMoneda</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_EliminarParametroCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ModificarContacto</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ModificarEstadoCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ModificarMoneda</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ModificarParametroCliente</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_ObtenerMódulosPorUsuario</i>	<i>dbo</i>	<i>stored procedure</i>
<i>sp_Validarlogin</i>	<i>dbo</i>	<i>stored procedure</i>

Fuente: Base de Datos PuntoClassic

### 3.7.2.1.Descripcion de los store procedures

- **sp\_ActualizarIntentosFallidos**

Este procedimiento actualiza los intentos fallidos en el logueo del usuario.

- **sp\_AdministrarBajaProducto**

En este sp se ingresa, actualiza, elimina, consulta las bajas de productos.

- **sp\_AdministrarCiudad**

En este sp se ingresa, actualiza, elimina, consulta las ciudades..

- **sp\_AdministrarCliente**

En este sp se ingresa, actualiza, elimina, consulta los Clientes.

- **sp\_AdministrarCompra**

En este sp se ingresa, actualiza, elimina, consulta las compras realizadas.

- **sp\_AdministrarContacto**

En este sp se ingresa, actualiza, elimina, consulta los contactos..

- **sp\_AdministrarCotizacion**

En este sp se ingresa, modifica, elimina y consulta las Cotizaciones..

- **sp\_AdministrarCotizacionProducto**

Este sp ingresa, modifica, elimina, consulta los productos asociadas con las cotizaciones.

- **sp\_AdministrarDetalleCompra**

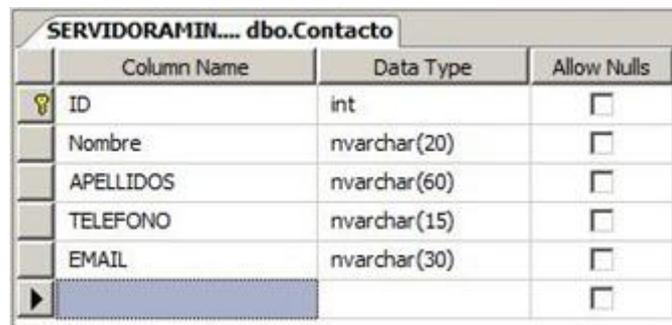
En este sp se ingresa, modifica, elimina y consulta los detalles de las compras.

### 3.8. Desarrollo del módulo financiero

Sobre el desarrollo de la BDD se ha determinado el diseño de un conjunto de web services, y clases pertinentes que se han desarrollado como se describe a continuación:

Ya creada la BDD usamos la tabla mostrada en el figura 88.

**Figura 89. Tabla Creada Datos de Contacto**

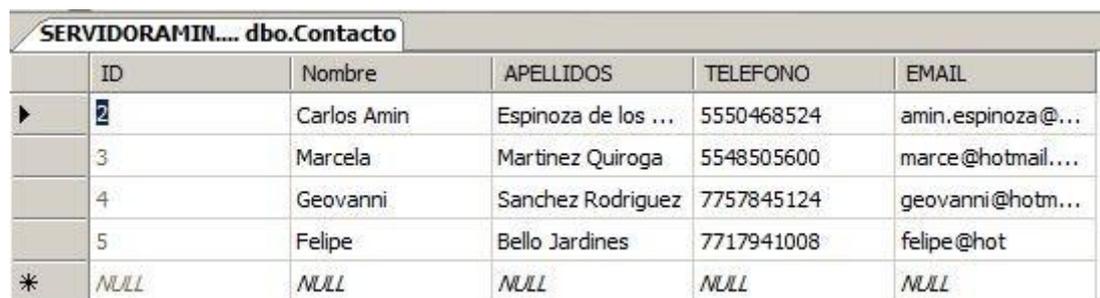


Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Nombre	nvarchar(20)	<input type="checkbox"/>
APELLIDOS	nvarchar(60)	<input type="checkbox"/>
TELEFONO	nvarchar(15)	<input type="checkbox"/>
EMAIL	nvarchar(30)	<input type="checkbox"/>
		<input type="checkbox"/>

Elaborado por: Christian Reyes - Daniel Rocha

El campo ID tiene la propiedad de Identidad. Ya que tengamos la tabla completa, ahora se la llena con unos cuantos registros.

**Figura 90. Tabla Ingreso de Registros**



ID	Nombre	APELLIDOS	TELEFONO	EMAIL
2	Carlos Amin	Espinoza de los ...	5550468524	amin.espinoza@...
3	Marcela	Martinez Quiroga	5548505600	marce@hotmail....
4	Geovanni	Sanchez Rodriguez	7757845124	geovanni@hotm...
5	Felipe	Bello Jardines	7717941008	felipe@hot
*	NULL	NULL	NULL	NULL

Elaborado por: Christian Reyes - Daniel Rocha

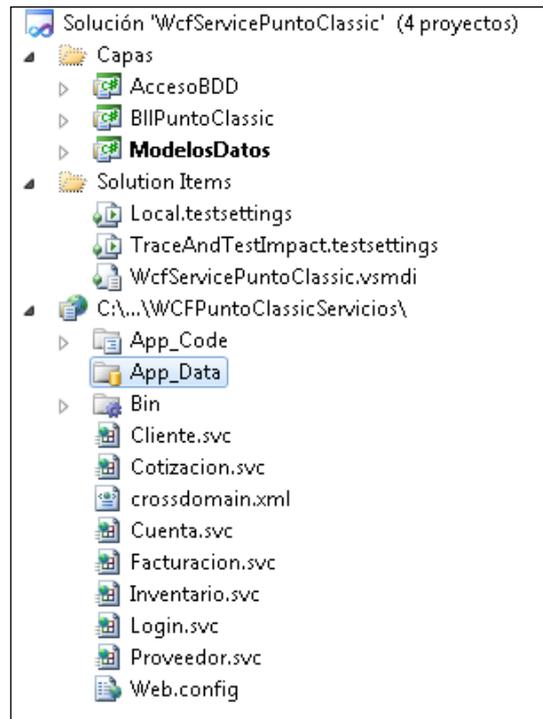
#### 3.8.1. Arquitectura del sistema

Se ha definido una arquitectura en capas para el desarrollo del Sistema, las que se mencionan a continuación:

- FrontEnd o Capa de Presentación
- Servicios WCF
  - BIIPuntoClassic o Capa de Negocio
  - AccesoBDD o Capa de Acceso a Datos

En la figura se observa la estructura del proyecto de Servicios WEB.

**Figura 91. Proyecto WCF's Punto Classic**



**Elaborado por: Christian Reyes - Daniel Rocha**

### 3.8.1.1. Modelo de datos

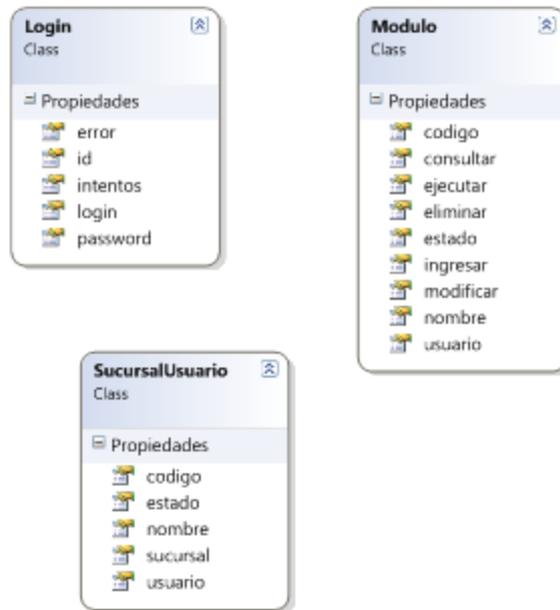
Refiere a la estructura datos que maneja el sistema y que son instanciados cuando existe una interacción entre la capa de negocio y la de acceso a datos.

Se usa como una estructura abstracta que documenta y organiza la información, en la cual se centra en el planeamiento del desarrollo de nuestra aplicación.

Las estructuras de datos que se han definido en el desarrollo de este proyecto de titulación, dan una mayor seguridad y flexibilidad en el envío y recepción de la información desde el cliente al servidor o viceversa.

Dentro de los modelos de datos definidos en nuestra aplicación se tienen los siguientes:

**Figura 92. Modelo de datos “Seguridad”**



**Fuente: Proyecto WCF's de Punto Classic**

Este modelo de datos se encuentra compuesto por tres clases que son Login, Modulo, SucursalUsuario en las cuales se encargan de representar la información de los usuarios así como de los módulos que tiene asignado cada usuario y las sucursales a las que tiene acceso.

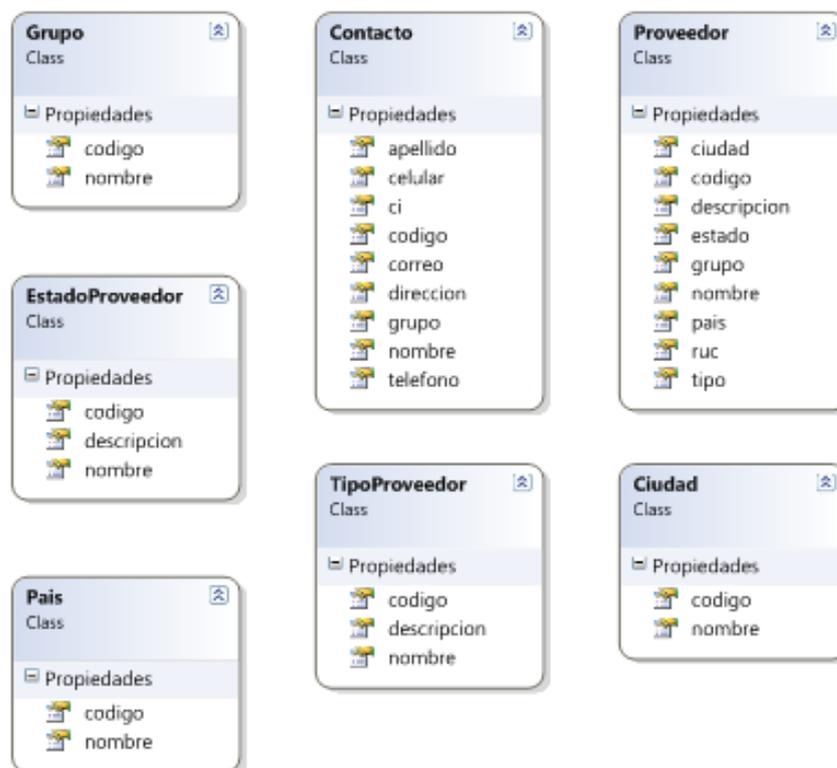
**Figura 93. Modelo de datos “Clientes”**



**Fuente: Proyecto WCF's de Punto Classic**

Este modelo de datos se encuentra compuesto por ocho clases que son Cliente, Contacto, Moneda, Pais, Grupo, Ciudad, TipoCliente, EstadoCliente en las cuales se encargan de representrar la información de los clientes así como el país de origen, ciudad de origen los tipos de clientes que se tiene parametrizado, los estados de los clientes, los datos de los clientes con su respectivo grupo de contactos.

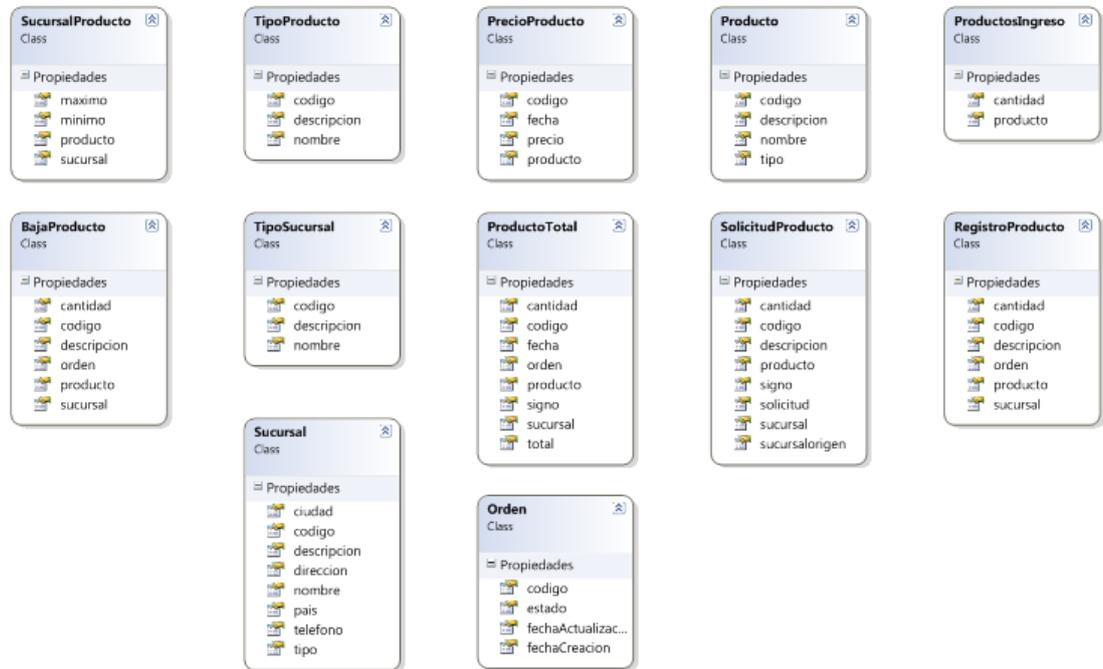
**Figura 94. Modelo de datos “Proveedores”**



**Fuente: Proyecto WCF's de Punto Classic**

Este modelo de datos se encuentra compuesto por siete clases que son Cliente, Contacto, Pais, Grupo, Ciudad, TipoProveedor, EstadoProveedor en las cuales se encargan de representar la información de los proveedores así como el país de origen, ciudad de origen los tipos de proveedores que se tiene parametrizado, los estados de los proveedores, los datos de los proveedores con su respectivo grupo de contactos.

**Figura 95. Modelo de datos “Inventarios”**



**Fuente: Proyecto WCF's de Punto Classic**

Este modelo de datos se encuentra compuesto por 12 clases que son SucursalProducto, TipoProducto, PrecioProducto, Producto, ProductoIngreso, TipoSucursal, Sucursal, BajaProducto, ProductoTotal, SolicitudProducto y Orden en las cuales se encargan de representar la información de los tipos de productos, los productos, los tipos de sucursales, las sucursales, las cantidades máximas y mínimas que tiene cada producto por sucursal, los productos ingresados en una sucursal, las solicitudes de productos que se realizan las sucursales, las ordenes de productos realizadas, el total de productos por sucursal.

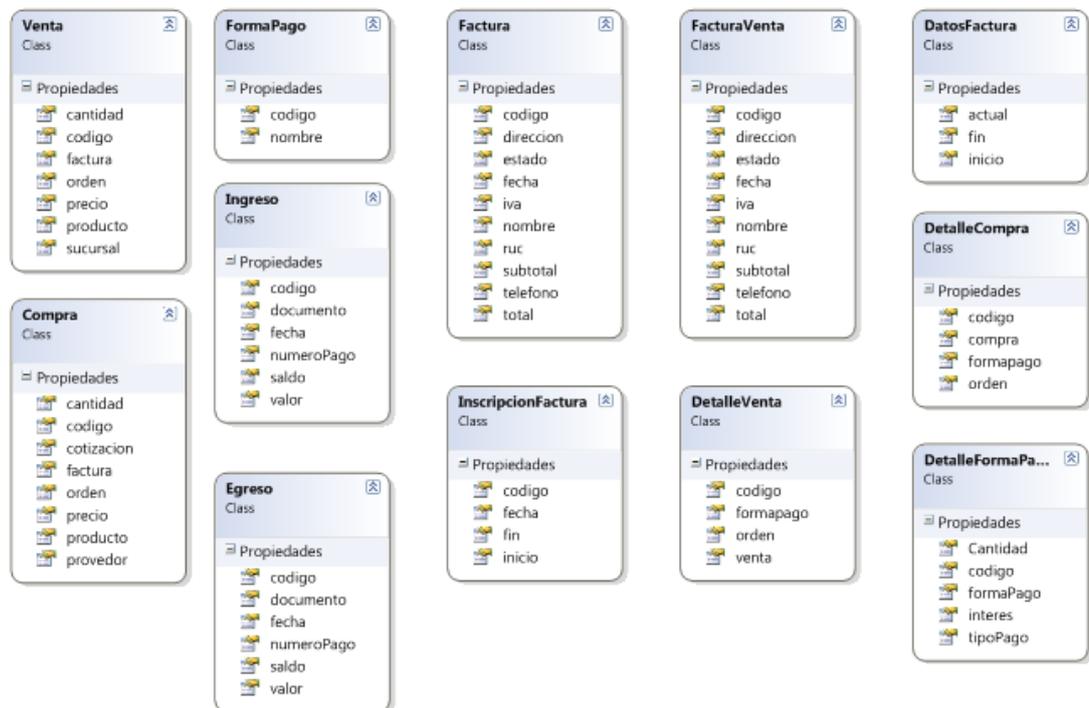
**Figura 96. Modelo de datos “Cotizacion”**



**Fuente: Proyecto WCF’s de Punto Classic**

Este modelo de datos se encuentra compuesto por 3 clases que son Cotizacion, SolicitudCompra, CotizacionProducto, las cuales se encargan de representar la información de las cotizaciones que se realizan en base a una solicitud de compra.

**Figura 97. Modelo de datos “Facturacion”**

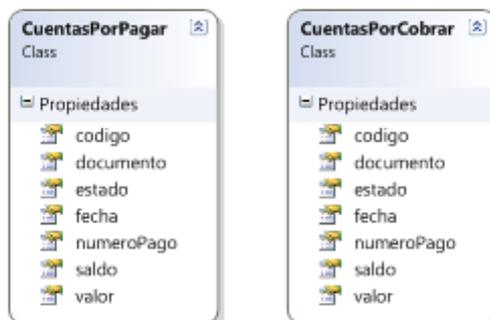


**Fuente: Proyecto WCF’s de Punto Classic**

Este modelo de datos se encuentra compuesto por 12 clases que son Venta, Compra, FormaPago, Ingreso, Egreso, Factura, InscpcionFactura, FacturaVenta, DetalleVenta, DatosFactura, DetalleCompra, and DetalleFormaPa...

DetalleVenta, DatosFactura, DetalleCompra, DetalleFormaPago las cuales se encargan de representar la información de las compras y las ventas que se realizan en la empresa.

**Figura 98. Modelo de datos “Cuentas”**



**Fuente: Proyecto WCF's de Punto Classic**

Este modelo de datos se encuentra compuesto por 2 clases que son CuentasPorPagar y CuentasPorCobrar las cuales se encargan de representar la información de las cuentas por pagar y cobrar que tiene la empresa.

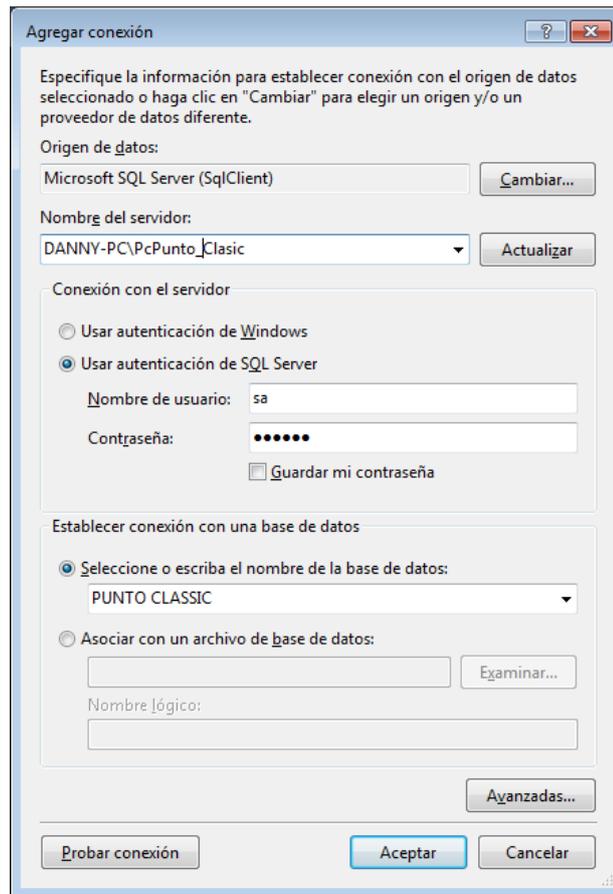
### **3.8.1.2. Acceso a datos**

Una vez definidos los modelos de datos a usarse en el aplicativo, se procede a generar un proyecto para acceder a la base de datos.

En éste caso se dejará las cadenas de conexión ordinarias a un lado para trabajar con LINQ en el acceso a la base de datos de Punto Classic.

En la figura 99 se puede visualizar la manera en la que se crea la conexión con la BDD, donde se establece el Origen, el nombre del Servidor, el tipo de autenticación que se usará para con el servidor de BDD y la base de datos que en nuestro caso es PUNTO CLASSIC.

**Figura 99. Imagen de conexión a la BDD**



**Fuente: Proyecto WCF's de Punto Classic**

En el proyecto de Acceso a datos se realiza la siguiente declaración en cada una de las clases, para obtener acceso a la conexión que se creó en la figura 98,

`using System.Linq;`

Al importar esta librería “System.Linq” se está accediendo a todos los métodos relacionados con el manejo de bases de datos por medio de LINQ.

Cuando se desee acceder a los Sp o tablas que forman parte de la base de datos, es necesario declarar un objeto de la siguiente manera:

`AccesoDataContext adc = new AccesoDataContext();`

Con esto se crea un objeto con el que se accede a la BDD de una manera muy sencilla, a continuación se observa una sentencia de ejemplo:

**adc.sp\_AdministrarEgreso(operacion, egreso.codigo,  
egreso.documento,egreso.valor,egreso.saldo,egreso.fecha,1, null, null);**

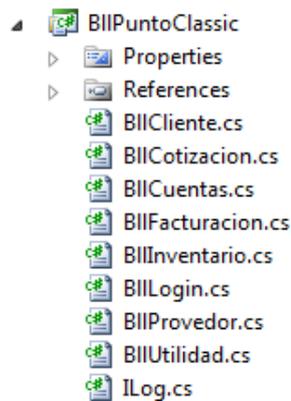
De esta manera se mantiene el acceso a la base de datos de una forma mas eficiente y rápida.

### **3.8.1.3.Capa de negocio**

Capa de negocio, denominada así ya que posee toda la lógica más significativa del sistema, y es la capa que interactúa con la interfaz de usuario o también llamada FrontEnd.

La capa de negocios que se ha definido para nuestro proyecto de titulación es la que se observa en la figura 99, se detallan las clases que conforman este proyecto.

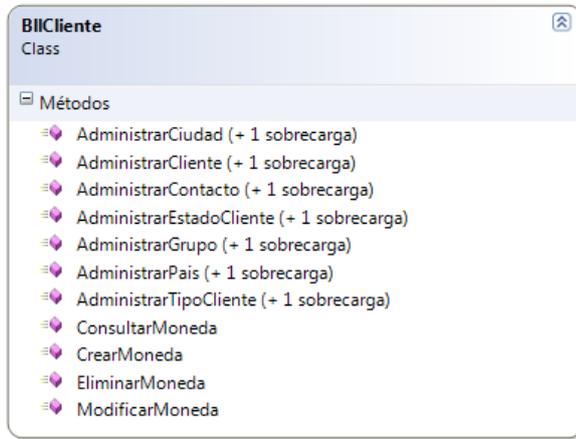
**Figura 100. Proyecto Capa de Negocio**



**Fuente: Proyecto WCF's de Punto Classic**

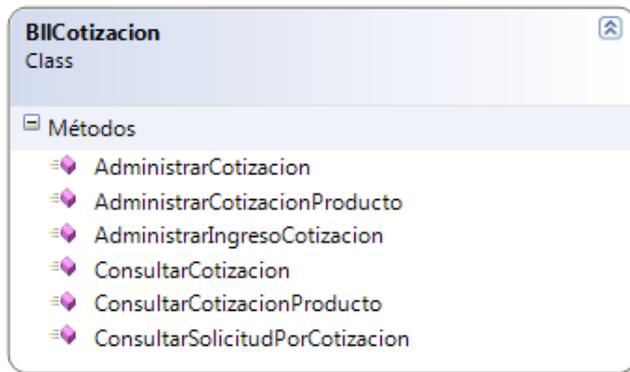
Cada clase de la BLL se encuentra formada por diversos métodos los cuales se observan en las figuras de la 100 a la 107.

**Figura 101. Clase BllCliente**



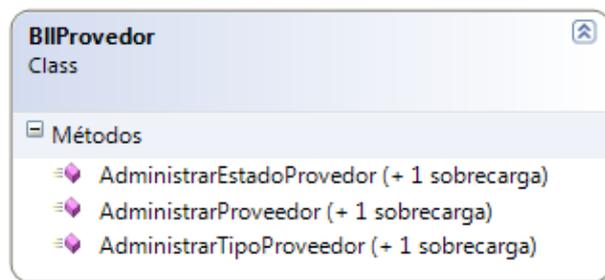
**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 102. Clase BllCotización**



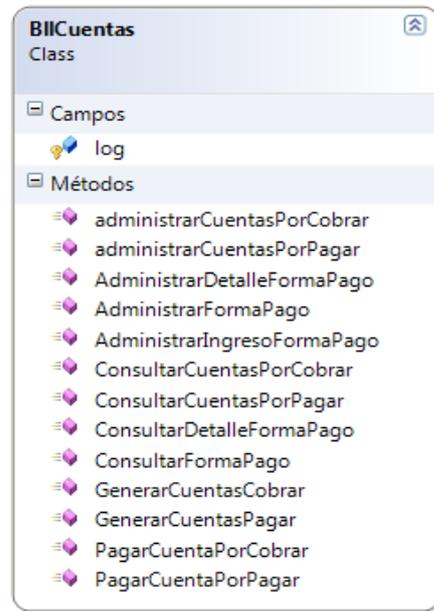
**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 103. Clase BllProveedor**



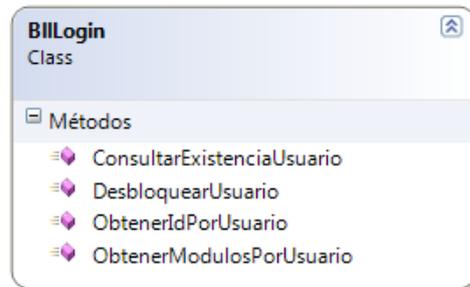
**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 104. Clase BIICuentas**



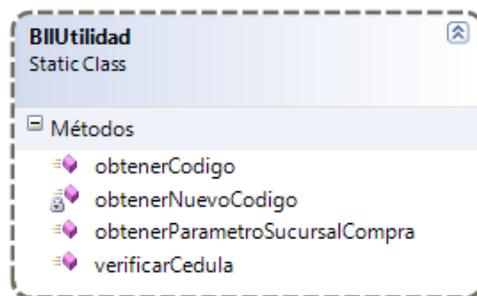
**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 105. Clase BIILogin**



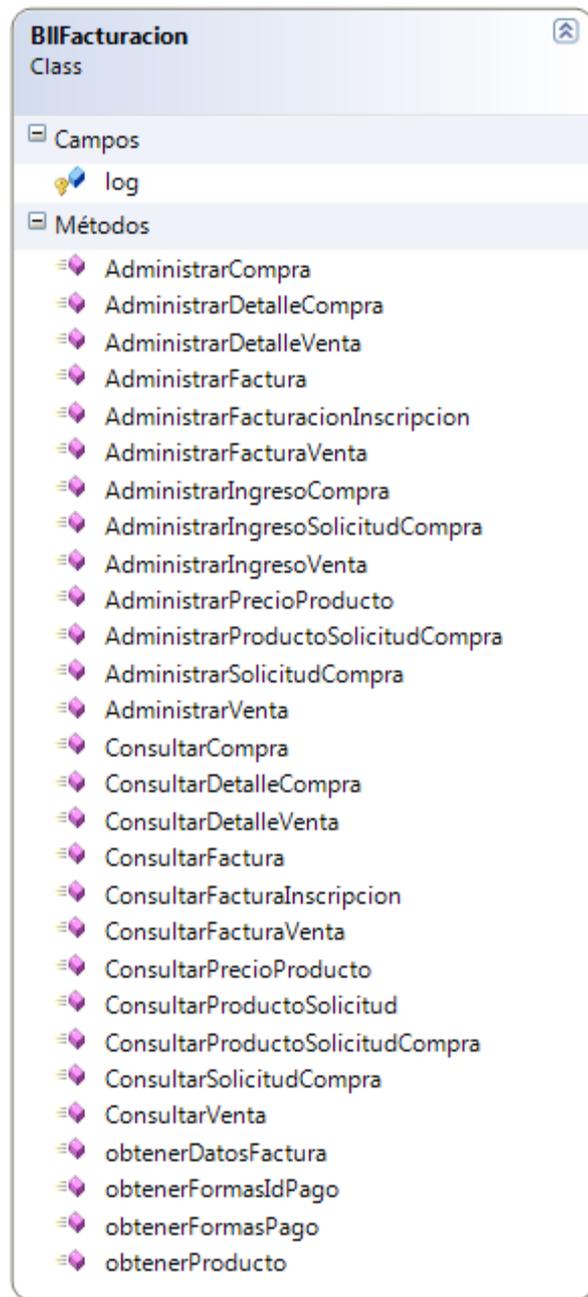
**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 106. Clase BIUtilidad**



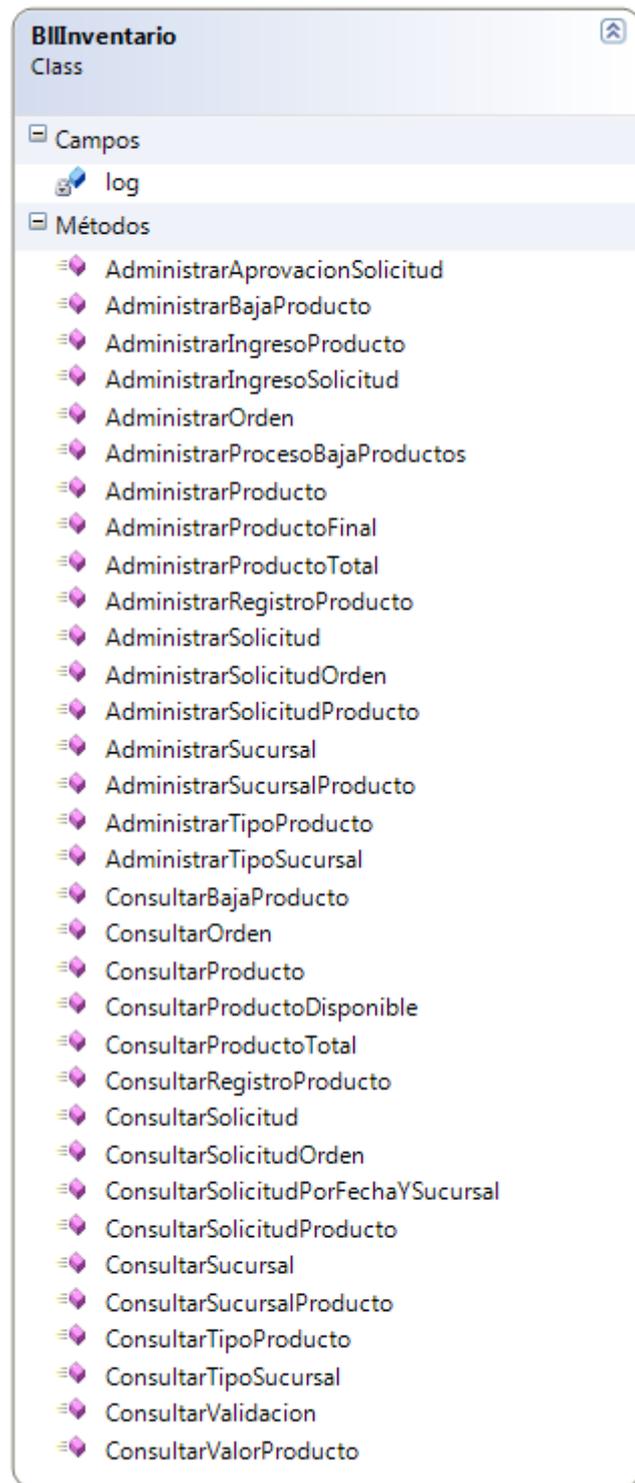
**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 107. Clase BllCuentas**



**Fuente: Proyecto Capa de Negocio de Punto Classic**

**Figura 108. Clase BllInventario**



**Fuente: Proyecto Capa de Negocio de Punto Classic**

Se puede mencionar que cada uno de los métodos descritos en las figuras (100 a 107) de las clases correspondientes a la capa de negocio, poseen un nombre descriptivo que relaciona muy bien la función para lo que fueron creados, por ejemplo:

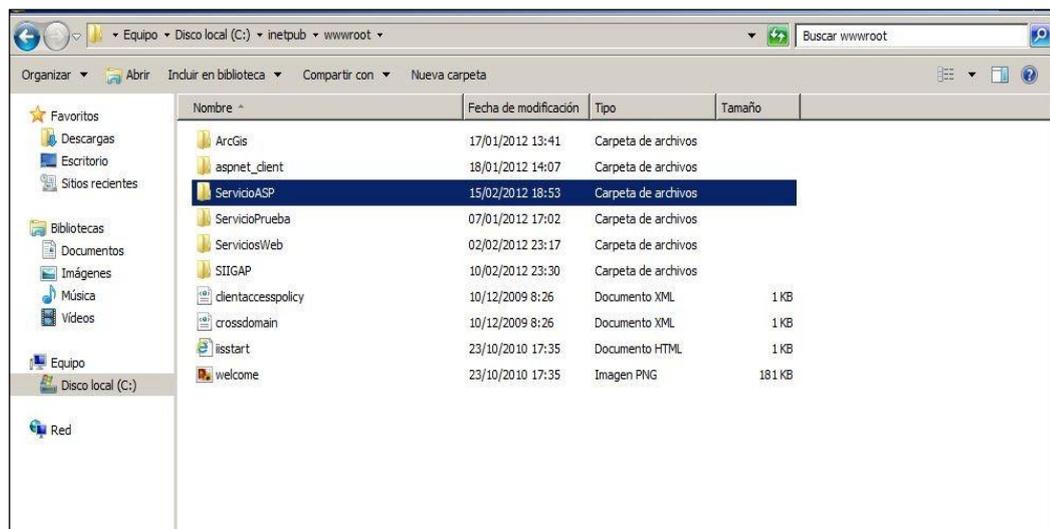
- BllCliente > AdministrarCiudad.- Administración de las ciudades parametrizadas en el sistema.
- BllCotizacion > ConsultarCotizacion.- Consultar las cotizaciones que fueron generadas en un período de tiempo.
- BllLogin > DesbloquearUsuario.- Desbloquea el usuario seleccionado.

### 3.8.1.4.Publicación de los servicios

Una vez terminado el desarrollo de los servicios, ya se puede optar por publicarlos de la siguiente manera:

En la ruta C:\inetpub\wwwroot (se debe tener instalado previamente el IIS), crea una nueva carpeta.

**Figura 109. Carpeta creada del WCF**

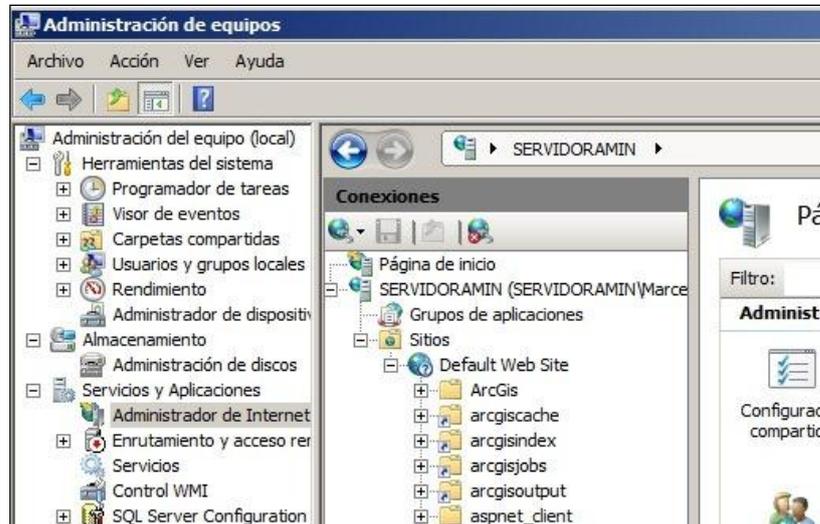


**Elaborado por: Christian Reyes - Daniel Rocha**

Ya con la carpeta creada, se accede al administrador del equipo. (Presionar la tecla de Windows + R y escribir el comando compmgmt.msc), seleccionar al

administrador de IIS, y se observa los nodos de la derecha hasta llegar a la posición de la siguiente imagen:

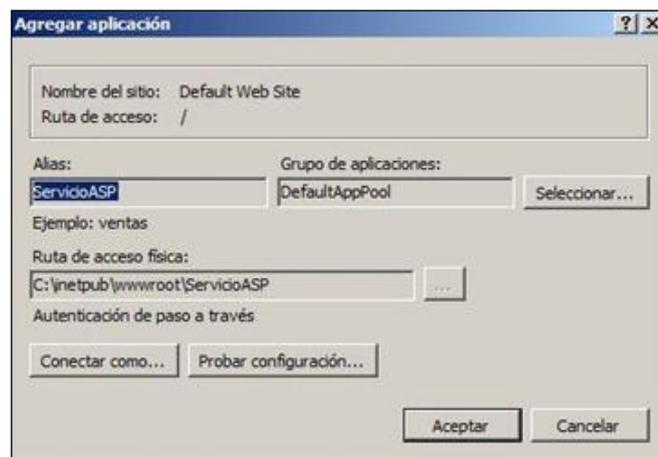
**Figura 110. Visualización de Código WSDL**



Elaborado por: Christian Reyes - Daniel Rocha

Al seleccionar la opción marcada aparecerá una ventana como la de abajo, dar clic en Aceptar.

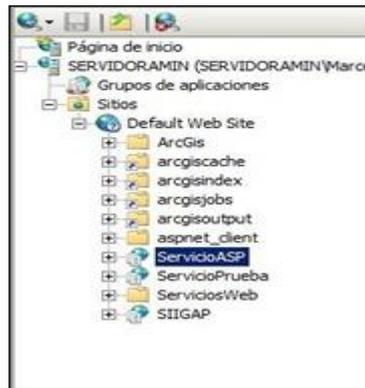
**Figura 111. Selección de Pool de Aplicación**



Elaborado por: Christian Reyes - Daniel Rocha

Y ahora se verá que la carpeta creada se muestra de la siguiente forma:

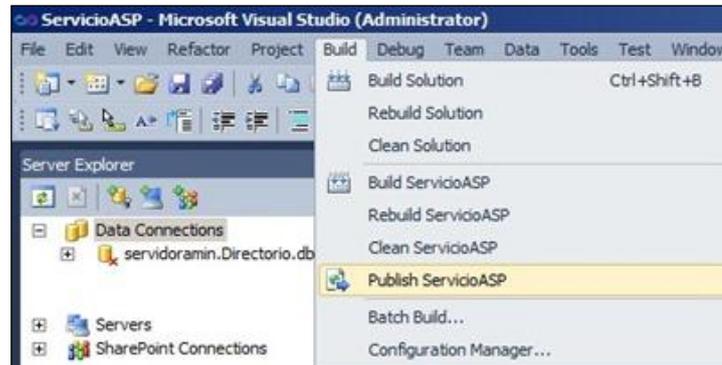
**Figura 112. Explorador de IIS**



**Elaborado por: Christian Reyes - Daniel Rocha**

Al regresar a Visual Studio y seleccionar en el menú de “Build”, se opta por la opción de “Publish Service”.

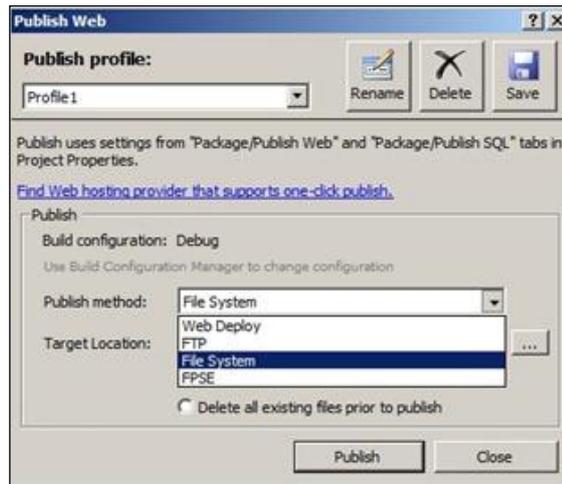
**Figura 113. Publicar WCF**



**Elaborado por: Christian Reyes - Daniel Rocha**

En el cuadro de diálogo escoger la opción de File System, en la opción de Publish method y en la de Target Location presionar el botón de “Examinar” como se muestra en la figura 112.

**Figura 114. Publicación de WCF**



**Elaborado por: Christian Reyes - Daniel Rocha**

Al presionar el botón de examinar, se tiene en pantalla ahora la siguiente ventana, ahí simplemente seleccionar la carpeta que se creó previamente.

**Figura 115. Selección de Aplicación a ser Publicada**



**Elaborado por: Christian Reyes - Daniel Rocha**

El cuadro de diálogo inicial quedará de la siguiente forma.

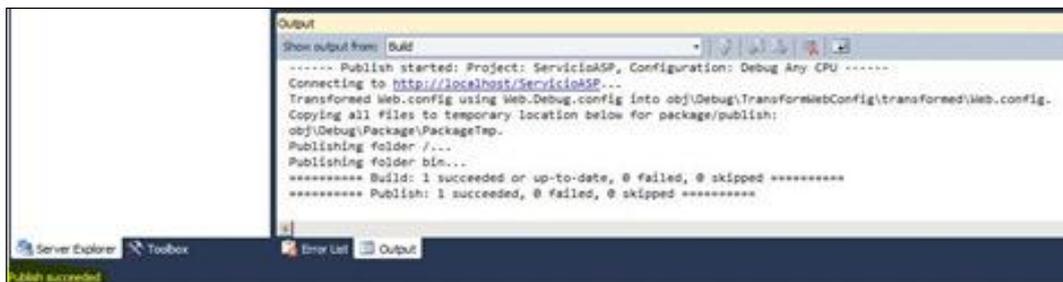
**Figura 116. Selección de Path**



Elaborado por: Christian Reyes - Daniel Rocha

Ahora presionar “Publish” y se verá en la esquina inferior izquierda y en la ventana de Output los siguientes mensajes.

**Figura 117. Visualización de la publicación del WCF en consola de Visual**



Elaborado por: Christian Reyes - Daniel Rocha

Ahora que ya se publicó el WCF, se podrá escribir la url del sitio con la siguiente secuencia `http://NombreDeTuServidor:Puerto/NombreDelServicio`, en éste caso quedará así <http://localhost:7075/Cotizacion.svc>, e irá variando únicamente el nombre del servicio entre cada uno de los creados, como se visualiza a continuación:

- <http://localhost:7075/Cotizacion.svc>

**Figura 118. Servicio de Cotización**



Elaborado por: Christian Reyes - Daniel Rocha

### 3.9. Desarrollo del módulo de simulación industrial

Para el desarrollo del simulador en la fabricación de rollos de tela, se ha seleccionado como herramienta a Intouch Version: 10.0.0,

**Figura 119. Version Intouch, Licencia**



Fuente: Intouch 10.0.0

### 3.9.1. Inicio

Consta de una pantalla inicial presentativa que contiene el mismo logotipo del módulo financiero, un mensaje indicativo y un botón que dará inicio a la simulación de la fabricación de tela.

**Figura 120. Pantalla de Inicio del Simulador**



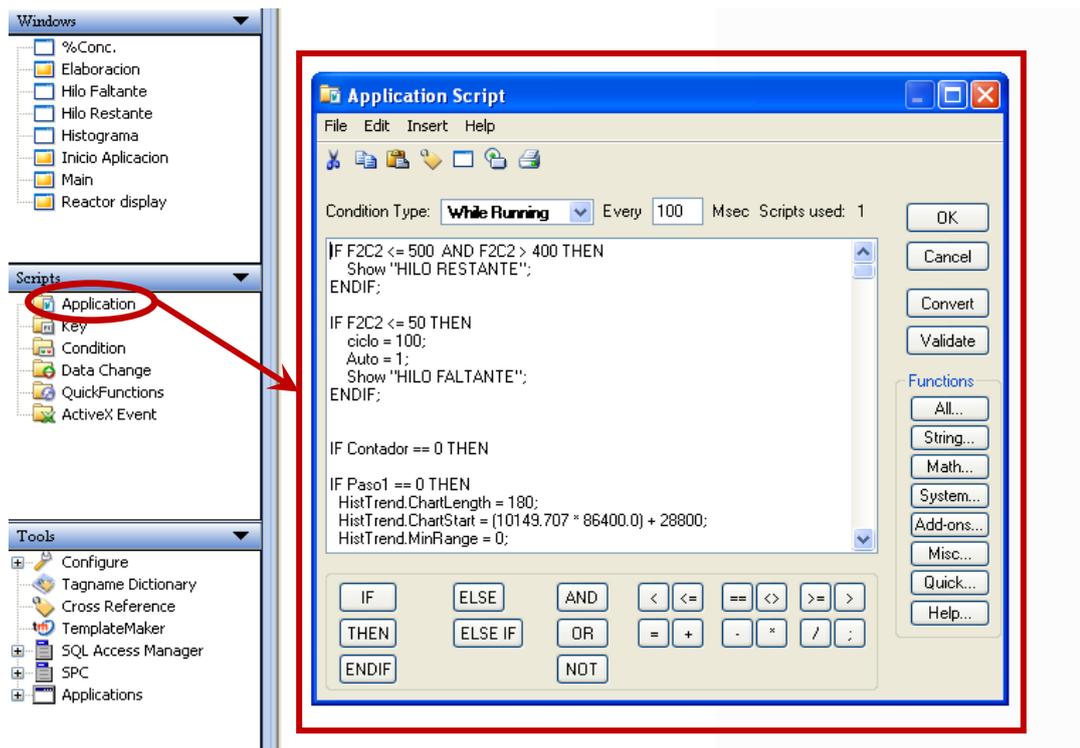
**Elaborado por: Christian Reyes - Daniel Rocha**

### 3.9.2. Secuencia en la elaboración de tela

Una vez que ya se dá clic en el botón “OK” de la pantalla presentativa, arranca el bucle del proceso que es prácticamente el motor de toda la simulación, éste proceso se plasma en un script de aplicación que es programado en la consola “Application” dentro del módulo de “Scripts” en Intouch.

En la figura 120 se muestra la pantalla donde se programa el script de la aplicación.

**Figura 121. Consola de programación del Script de Aplicación**



**Elaborado por: Christian Reyes - Daniel Rocha**

### **Script de Aplicación.**

```

IF F2C2 <= 500 AND F2C2 > 400 THEN
  Show "HILO RESTANTE";
ENDIF;

IF F2C2 <= 50 THEN
  ciclo = 100;
  Auto = 1;
  Show "HILO FALTANTE";
ENDIF;

IF Contador == 0 THEN

IF Paso1 == 0 THEN
  HistTrend.ChartLength = 180;
  HistTrend.ChartStart = (10149.707 * 86400.0) + 28800;
  HistTrend.MinRange = 0;
  HistTrend.MaxRange = 100;
  Cursor2 = 0.5;
  HistTrend.Pen4 = SetPoint.TagID;
  Pen04 = SetPoint.TagID;
  Paso1 = 1;
  
```

```

ciclo = 100;
ENDIF;

IF Auto THEN

IF Paso1 == 11 THEN
Paso1 = 1;
ENDIF;

IF Paso1 == 1 THEN
AguaValvula = 1;
ConcBomba = 0;
ConcValvula = 0;
SteamValvula = 0;
Mezclador = 0;
TransferBomba = 0;
TransferValvula = 0;
SalidaValvula = 0;
Paso1 = 2;
ENDIF;

IF Paso1 == 2 THEN
IF NiveldeMezclador >= (2000 - (Batch%Conc * 20)) THEN
Paso1 = 3;
ENDIF;
ENDIF;

IF Paso1 == 3 THEN
AguaValvula = 0;
ConcBomba = 1;
ConcValvula = 1;
Paso1 = 4;
ENDIF;

IF Paso1 == 4 THEN
IF (NiveldeMezclador >= 2000) THEN
Paso1 = 5;
ENDIF;
ENDIF;

IF Paso1 == 5 THEN
ConcBomba = 0;
ConcValvula = 0;
Mezclador = 1;
SteamValvula = 1;
Paso1 = 6;
ENDIF;

IF Paso1 == 6 THEN
IF TemperaturadeMezclador >= 195 THEN

```

```

    Paso1 = 7;
ENDIF;
ENDIF;

IF Paso1 == 7 THEN
    Mezclador = 0;
    SteamValvula = 0;
    TransferBomba = 1;
    TransferValvula = 1;
    Paso1 = 8;
ENDIF;

IF Paso1 == 8 THEN
    IF NiveldeProduccion > 9000 THEN
        Paso1 = 9;
    ELSE IF NiveldeMezclador <= 0 THEN
        Paso1 = 10;
    ENDIF;
ENDIF;
ENDIF;

IF Paso1 == 9 THEN
    SalidaValvula = 1;
    IF NiveldeProduccion < 5000 THEN
        SalidaValvula = 0;
        Paso1 = 8;
    ENDIF;
ENDIF;

IF Paso1 == 10 THEN
    TransferBomba = 0;
    TransferValvula = 0;
    BatchNumber = BatchNumber + 1;
    TemperatureMezclador = 60;
    Paso1 = 1;
ENDIF;
ELSE {AUTO}
    IF Paso1 < 11 THEN
        Paso1 = 11;
    ENDIF;

    IF TemperatureMezclador > 400 THEN
        TemperatureMezclador = 391;
    ENDIF;

    IF NiveldeProduccion > 9000 THEN
        NiveldeProduccion = 8771;
    ENDIF;

ENDIF; {AUTO}

```

```

IF AguaValvula THEN
    NiveldeMezclador = NiveldeMezclador + 50;
ENDIF;

IF ConcBomba AND ConcValvula THEN
    NiveldeMezclador = NiveldeMezclador + 50;
ENDIF;

IF SteamValvula THEN
    TemperaturadeMezclador = TemperaturadeMezclador + 5;
    IF TemperaturadeMezclador >= 200 THEN
        TemperaturadeMezclador = TemperaturadeMezclador - 8;
    ENDIF;
ELSE
    TemperaturadeMezclador = TemperaturadeMezclador - 2.3;
    IF TemperaturadeMezclador < 15 THEN
        TemperaturadeMezclador = 21;
    ENDIF;
ENDIF;

IF TransferBomba AND TransferValvula THEN
    NiveldeMezclador = NiveldeMezclador - 45;
    IF NiveldeMezclador < 0 THEN
        NiveldeMezclador = 0;
    ENDIF;
    IF NiveldeMezclador > 0 THEN
        NiveldeProduccion = NiveldeProduccion + 13;
    ENDIF;
ENDIF;

IF SalidaValvula THEN
    NiveldeProduccion = NiveldeProduccion - 91;
    IF NiveldeProduccion < 0 THEN
        NiveldeProduccion = 0;
    ENDIF;
ENDIF;

IF NiveldeMezclador >= 2000 THEN
    NiveldeMezclador = 2000;
ENDIF;
ENDIF;
Contador = Contador + 1;
IF ciclo < 100 THEN

IF ciclo == 0 THEN
    MovimientoVertical = 0;
    MovimientoHorizontal = 0;
    IF Auto THEN
        ciclo = 1;

```

```

ENDIF;
ENDIF;

IF ciclo == 1 THEN
MovimientoVertical = MovimientoVertical + Velocidad;
IF MovimientoVertical >= 41 THEN
MovimientoVertical = 41;
IF Auto THEN
ciclo = 2;
ENDIF;
ENDIF;
ENDIF;

IF ciclo == 2 THEN
MovimientoHorizontal = MovimientoHorizontal + Velocidad * 3;
IF MovimientoHorizontal >= 134 THEN
ciclo = 3;
ENDIF;
ENDIF;

IF ciclo == 3 THEN
MovimientoHorizontal = MovimientoHorizontal + Velocidad * 3;

IF Auto THEN
IF MovimientoHorizontal >= 275 THEN
Eyector = 1;
IF MovimientoHorizontal >= 289 THEN
MovimientoHorizontal = 289;
ciclo = 4;
ENDIF;
ENDIF;
ELSE {AUTO}

IF Eyector THEN

IF (MovimientoHorizontal > 250) AND (MovimientoHorizontal < 289) THEN
ciclo = 7;
ENDIF;

IF (MovimientoHorizontal > 272) AND (MovimientoHorizontal < 310) THEN
MovimientoHorizontal = 289;
ciclo = 4;
ENDIF;

ELSE
IF MovimientoHorizontal > 310 THEN
ciclo = 6;
ENDIF;
ENDIF;

```

```

ENDIF; {Auto}

ENDIF; {ciclo == 3}

IF ciclo == 4 THEN

    IF ciclo == 4 AND MovimientoVertical == 41 THEN
        F2C1 = F2C1 + 1;
    ENDIF;
    MovimientoVertical = MovimientoVertical - Velocidad;
    MovimientoHorizontal = MovimientoHorizontal - Velocidad;
    IF MovimientoVertical <= 30 THEN
        Eyector = 0;
    ENDIF;
    IF MovimientoVertical <= 0 THEN
        IF Auto THEN
            ciclo = 1;
        ELSE
            ciclo = 0;
        ENDIF;
        MovimientoHorizontal = 0;
        MovimientoVertical = 0;
    ENDIF;
ENDIF;

IF ciclo == 5 THEN
    MovimientoVertical = MovimientoVertical + 1;
    Show "Message from the Manager";

ENDIF;

IF ciclo == 6 THEN
    Show "Message from the Manager";

ENDIF;

IF ciclo == 7 THEN
    Eyector = 1;
    IF MovimientoHorizontal < 272 THEN
        MovimientoHorizontal = MovimientoHorizontal + Velocidad;
    ELSE
        MovimientoHorizontal = 272;
    ENDIF;
    ciclo = 5;
ENDIF;
ENDIF;
ENDIF;

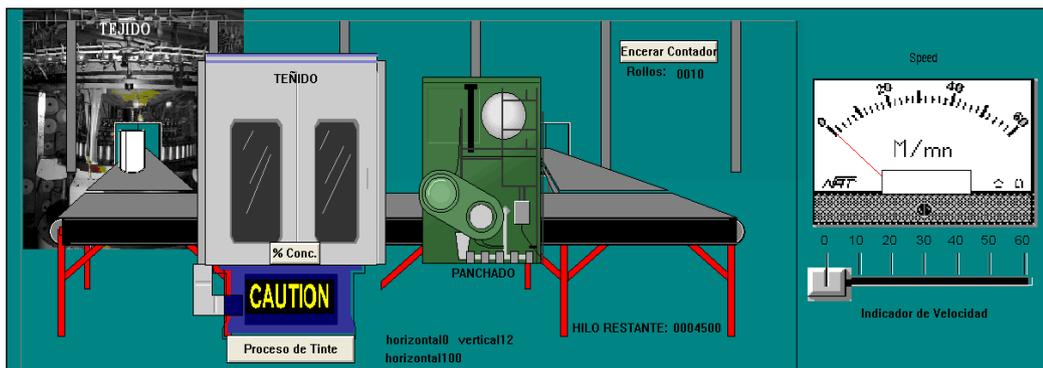
IF Contador >= 5 THEN
    Contador = 0;
ENDIF;

```

El script del simulador, describe cada una de las validaciones que son efectuadas en el bucle “While Running”, mientras el aplicativo está siendo ejecutado.

En la figura 121, se visualiza los 3 procesos que se efectúan en la fabricación de tela: tejido, teñido y planchado. Además se tiene un velocímetro a la derecha de la banda de procesos que permite controlar la velocidad de producción.

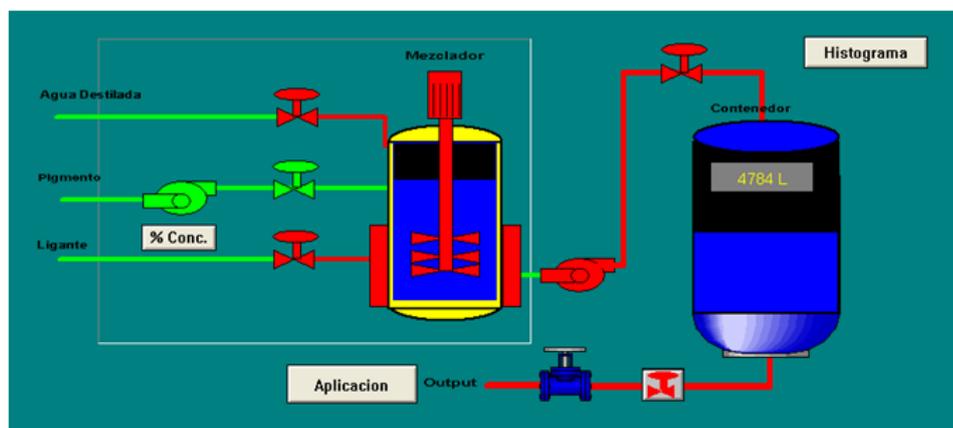
**Figura 122. Procesos en la Simulación**



Elaborado por: Christian Reyes - Daniel Rocha

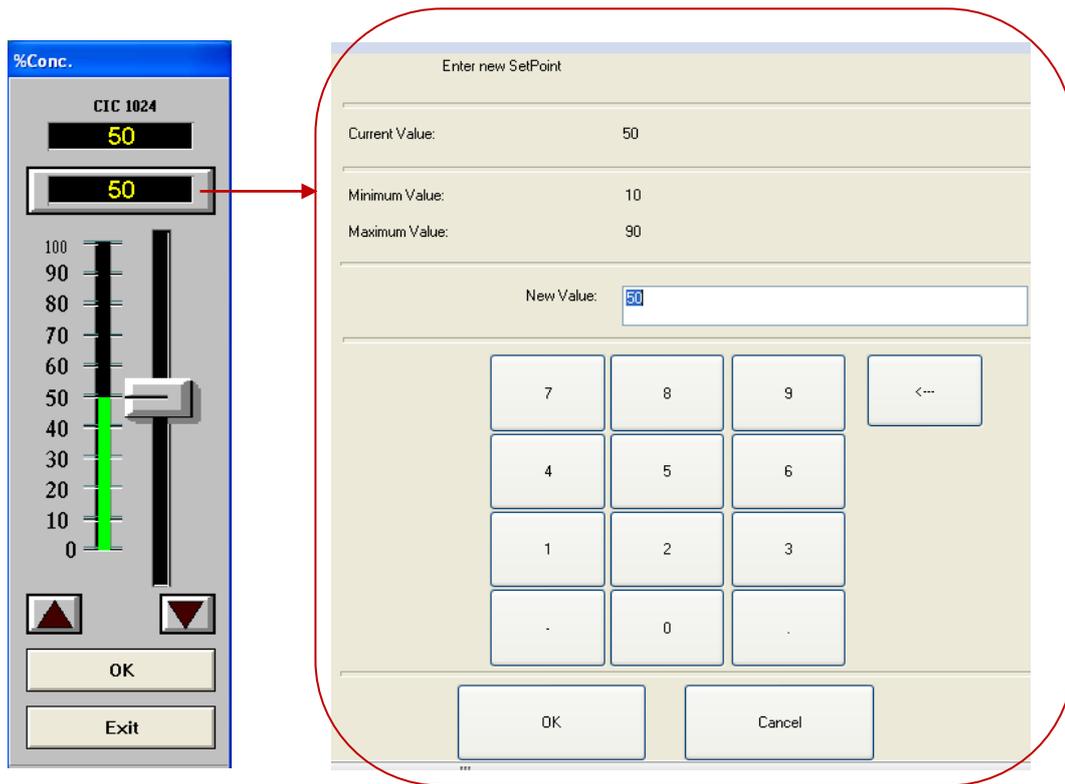
En la preparación del tinte (Figura 123) se observa que mantiene 3 ductos de alimentación al mezclador, donde uno de ellos es agua destilada, otro el pigmento o el color que se le dará al tinte, y el ligante, que es una sustancia con la propiedad de adherir el color a la tela de una manera mas optima, para ello se varia el porcentaje de concentración del ligante en el tinte a ser preparado, esto se observa en la figura 123.

**Figura 123. Preparación del Tinte**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 124. Porcentaje de Concentración del Ligante**



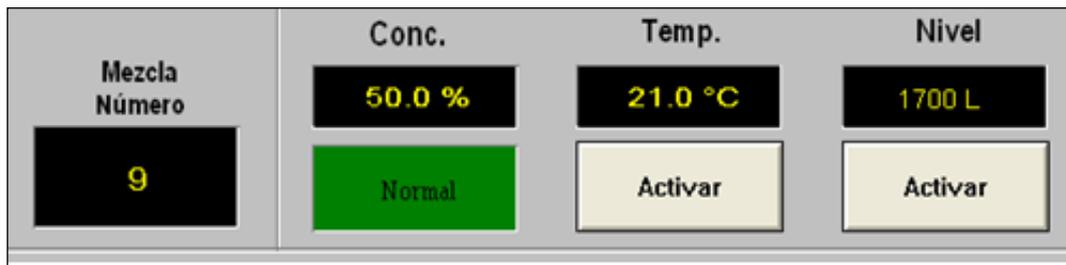
**Elaborado por: Christian Reyes - Daniel Rocha**

Una vez lleno el mezclador con la cantidad justa de cada elemento a componer el tinte, se procede a mezclarlo con aspas a gran velocidad y herméticamente sellado, cabe recalcar que la temperatura a la que son mezclados estos componentes es como máxima de 195 grados Celsius y un límite de almacenamiento del mezclador de 2000 litros.

La temperatura incrementa paulatinamente mientras se produce la mezcla de los elementos, y decrementa cuando se está vaciando el tinte creado en el contenedor y mientras se llena nuevamente el mezclador para elaborar mas tinte.

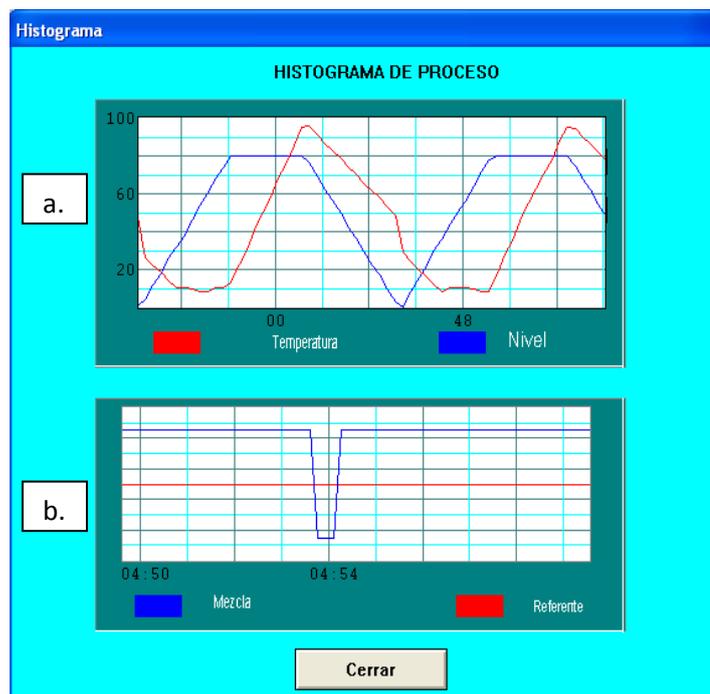
En la figura 125 se muestra los indicadores para la preparación del tinte.

**Figura 125. Indicadores en la generación de Tinte**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 126. Histograma de la preparación de Tinte**



Elaborado por: Christian Reyes - Daniel Rocha

La temperatura incrementa paulatinamente mientras se produce la mezcla de los elementos, y decrementa cuando se está vaciando el tinte creado en el contenedor, esto se visualiza en la figura 126, en una línea de color rojo que representa a la temperatura y la línea de color azul que determina el nivel de llenado en litros del tanque de mezcla.

Mientras que en el histograma de la figura 126, se muestran los saltos que se producen por cada mezcla realizada en color azul y de color rojo una línea referencial al llenado de medio tanque de mezcla.

**Figura 127. Flujo de procesos en fabricación de Tinte**

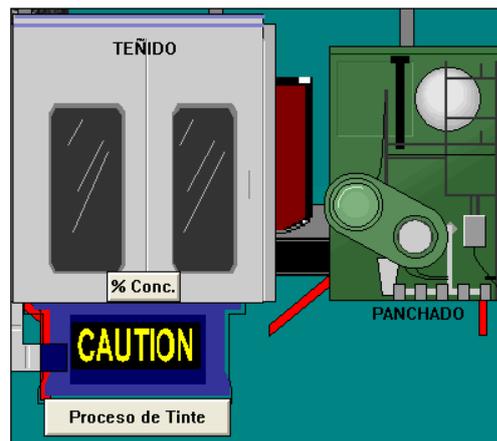


Elaborado por: Christian Reyes - Daniel Rocha

En la figura 127 se puede observar un resumen de los procesos y cronología de los mismos en la fabricación del tinte, iniciando con el llenado y la combinación de los elementos necesarios para su fabricación, luego mezcla de los elementos, y finalizando con el almacenamiento sincronizado del tinte fabricado en el contenedor general.

Una vez finalizado el teñido, sale el rollo de tela con color, y se dispone a entrar en un conjunto de rodillos donde será planchada para su posterior almacenamiento en bodegas. Se observa en la figura 128.

**Figura 128. Terminación de tinturado de tela e inicio de planchado**

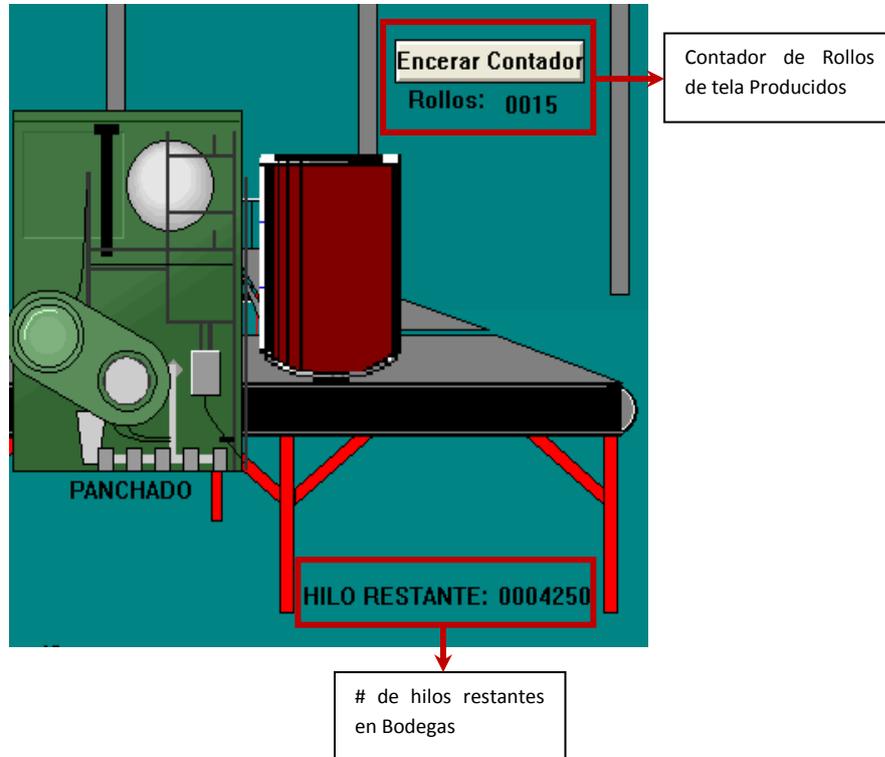


Elaborado por: Christian Reyes - Daniel Rocha

Una vez planchada la tela sale y eleva el contador de rollos de tela producidos en 1, sincronizando con la base de datos de inventarios y el módulo financiero para mantener un control en tiempo real, tanto de insumos (hilos) como de productos (rollos de tela), se puede visualizar en la figura 129, mientras que en la figura 130 se

observa como se sincronizan los datos con el archivo Excel creado para monitorizar la producción generada desde el simulador en tiempo real.

**Figura 129. Finalización en la Producción de Rollos de tela**



Elaborado por: Christian Reyes - Daniel Rocha

**Figura 130. Archivo Excel para sincronización de Inventarios**



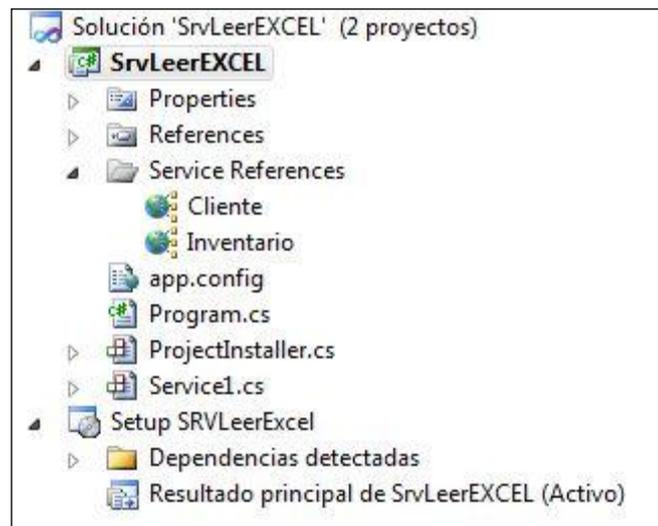
Elaborado por: Christian Reyes - Daniel Rocha

### 3.10. Desarrollo de la interfaz de comunicación entre los módulos desarrollados

El módulo financiero y el módulo de simulación industrial deben comunicarse entre sí, por éste motivo se desarrollo un servicio Windows que se encarga de leer el archivo Excel (archivo donde se almacena y actualizan datos de producción “datos.xls” ) generado por el simulador, para obtener los datos que arroja la simulación en tiempo real e interactuar con el servicio de Inventarios actualizando información como: insumos y productos de la fabrica.

La estructura de la solución para el servicio Windows se puede observar en la figura 131, donde tenemos un proyecto que representa al servicio y la lógica del mismo, y otro proyecto de instalación el cual será el generador del instalador del servicio para su posterior implementación.

**Figura 131. Estructura de la solución del Servicio Excel**

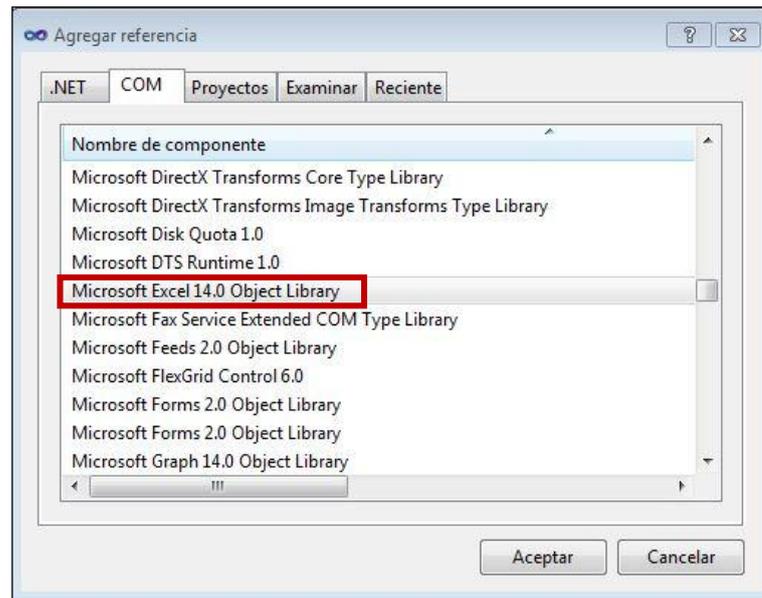


**Elaborado por: Christian Reyes - Daniel Rocha**

El nombre del proyecto del servicio Windows que se creó es “SrvLeerExcel”, al mismo que se añade la referencia “Microsoft Excel 14.0 Object Library” para poder interactuar con archivos con extensión xls.

En la figura 132 se observa como se debe añadir la referencia indicada en el párrafo anterior

**Figura 132. Referencia a la dll Microsoft Excel**



**Elaborado por: Christian Reyes - Daniel Rocha**

Una vez importada la librería necesaria para interactuar con archivos Excel se debe definir la lógica del servicio de tal modo que se mantenga un control en el mismo para una fácil detección de errores o fallos; para esto en el constructor se definió que loguee el visor de eventos de Windows, específicamente en el visor de aplicación. A continuación las sentencias de mayor importancia para la creación de la fuente que se logueará en el visor de eventos de Windows:

```
if (!System.Diagnostics.EventLog.SourceExists("SrvLeerExcel"))
{
    System.Diagnostics.EventLog.CreateEventSource("SrvLeerExcel", "SrvLeerExcel");
}
eventLogSRV.Source = "SrvLeerExcel";
eventLogSRV.Log = "SrvLeerExcel";
```

Con éstas sentencias se valida si existe alguna fuente u origen desde donde se enviara a loguear al visor de eventos de Windows con el nombre de "SrvLeerExcel", si no existe se crea y si existe únicamente iguala los valores Fuente y Log a SrvLeerExcel para comenzar a ser logueados en caso de fallo o eventos como iniciar servicio o detener servicio.

Para acceder al archivo Excel se utiliza las sentencias siguientes:

```

Microsoft.Office.Interop.Excel.Application      Aplic      =      new
Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.Workbook Libro = Aplic.Workbooks.Open(@"d:\datos.xls",
Type.Missing, true, Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Type.Missing);
Microsoft.Office.Interop.Excel.Worksheet      Hoja1      =
(Microsoft.Office.Interop.Excel.Worksheet)Libro.Worksheets.get_Item(1);

Microsoft.Office.Interop.Excel.Range rangoRolloTela = Hoja1.get_Range("A2", "A2");
string rollos = rangoRolloTela.Text.ToString();

```

**Azul.-** Declaración de objeto “Aplic” perteneciente a la librería de Microsoft.Office

**Vino.-** Instancia del Objeto Libro, que representa al archivo Excel físico, donde se coloca el path para poder leerlo.

**Naranja.-** Instancia de la Hoja de Excel a ser leída en el archivo abierto.

**Verde.-** Instancia de datos leídos en el rango de celdas seleccionado y asignación a la variable string Rollos.

De este modo se accede a la información del archivo Excel ubicado en “d:\datos.xls”, desde el servicio Windows, manteniendo una constante actualización proveniente del simulador, dicha actualización constante que es reflejada en las lecturas desde el servicio Windows hacia el archivo Excel. Las lecturas obtenidas son actualizadas en la base de datos de inventarios de Punto classic por medio del WCF de inventarios. Cabe recalcar que se realiza una referencia adicional a un servicio web, el de Clientes, que es el que proporciona un método necesario para la generación de códigos en el ingreso de productos.

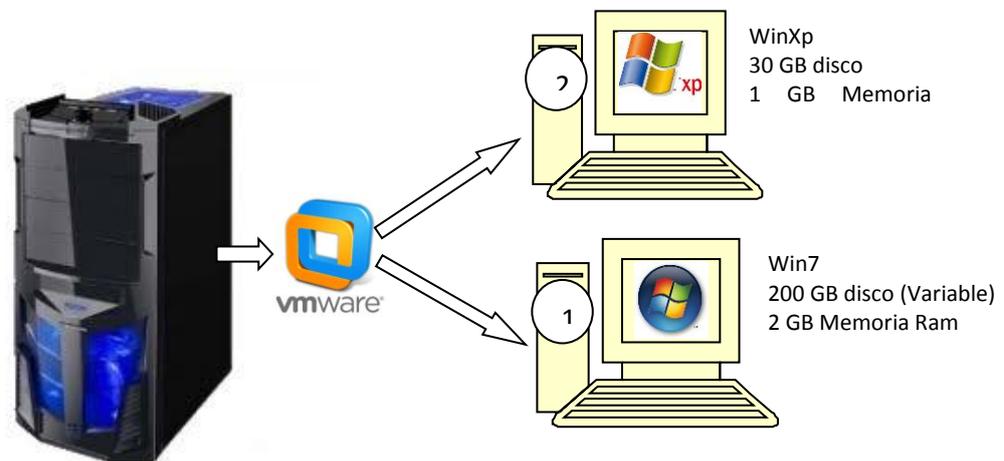
## CAPÍTULO 4

### IMPLEMENTACIÓN Y PRUEBAS

#### 4.1. Introducción

La implementación de los módulos desarrollados e instalación de herramientas necesarias en el sistema serán montados en 2 máquinas virtuales para la facilidad en la utilización de los mismos. Para lo cual se usará el gestor de máquinas virtuales VMWare Workstation 8.0, esto se observa en la figura 133.

**Figura 133. Diagrama de Máquinas Virtuales**



**Elaborado por: Christian Reyes - Daniel Rocha**

La primera máquina virtual contiene un sistema Win7, en esta se instalarán los servicios Web, la base de datos y sitio Web del módulo financiero.

Mientras que la segunda máquina virtual contiene WinXP, en esta se instalará el software Intouch y el simulador.

Las pruebas realizadas en el sistema implantado, serán efectuadas por medio de la herramienta JMeter, con la que se probarán los puntos más críticos en el funcionamiento del sistema, que son los niveles de respuesta y accesibilidad a los WCF desde el aplicativo web.

## 4.2. Máquinas Virtuales

Una máquina virtual es una emulación de un computador virtual, que permite ejecutar programas como si fuese una computadora real.

### 4.2.1. Creación de las máquinas virtuales

Teniendo el gestor de máquinas virtuales VMWare WorkStation 8, se procede a crear la máquina virtual donde se instalará el sistema operativo Win7, y basándose en el mismo método de creación, se creará la máquina virtual para WinXp.

En la figura 134 se observa la estación VMWare, se debe crear una nueva máquina virtual, dar clic en el recuadro marcado.

**Figura 134. Pantalla VWWare WorkStation 8**



**Elaborado por: Christian Reyes - Daniel Rocha**

Al dar clic en nueva máquina virtual, se despliega la figura 135.

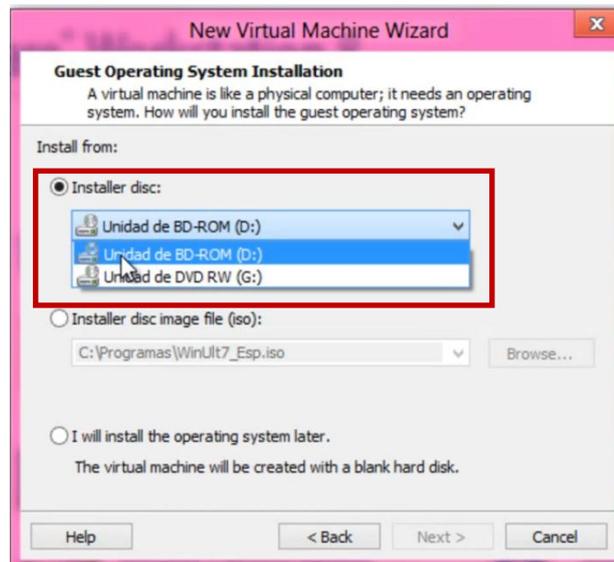
**Figura 135. Creación de Nueva Máquina Virtual (paso 1)**



Elaborado por: Christian Reyes - Daniel Rocha

Al dar clic en “Next” se presenta la figura 136, en el que se puede elegir la opción instalar desde una unidad de disco CD o DVD.

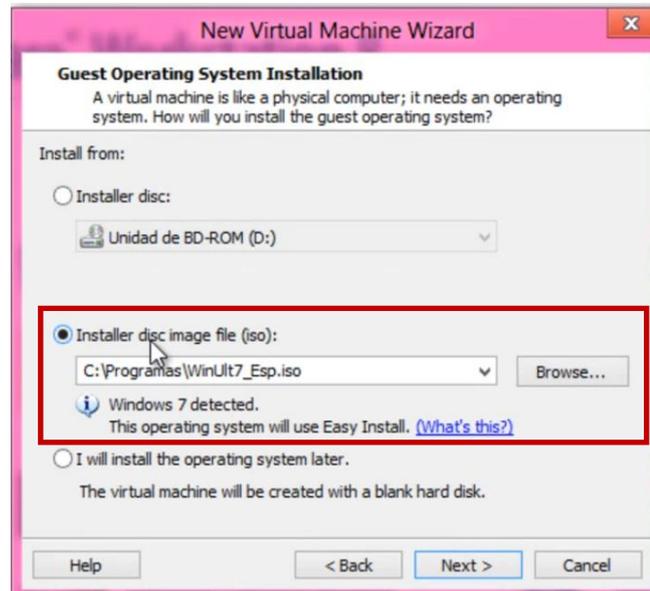
**Figura 136. Creación de Nueva Máquina Virtual (paso 2)**



Elaborado por: Christian Reyes - Daniel Rocha

Pero también es posible elegir la opción de “instalar desde una imagen virtual ISO”, por facilidad se elegirá ésta opción, se puede observar en la figura 137, en el que se establece el path físico donde se encuentra la imagen Virtual.

**Figura 137. Creación de Nueva Máquina Virtual (paso 3)**

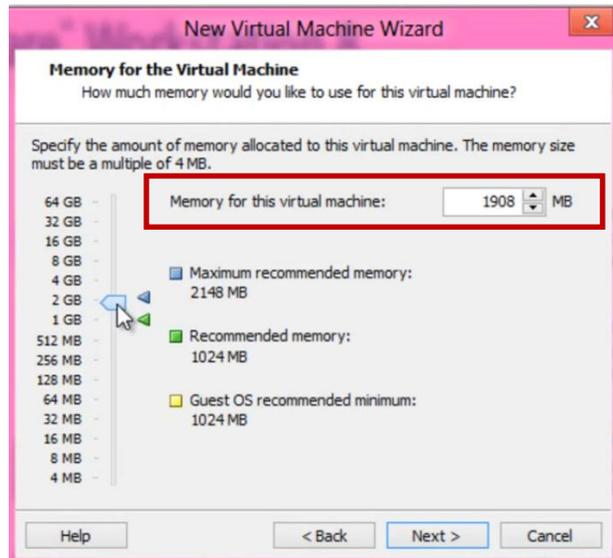


**Elaborado por: Christian Reyes - Daniel Rocha**

Después de haber seteado la imagen virtual de Win7 que se instalará, dar clic en “Next” y ahora se debe establecer el tamaño de la memoria que se asignara a ésta maquina virtual cuando sea instanciada. Esto se puede observar en la figura 138.

Éste procedimiento se repetirá cuando se instale el sistema operativo en la otra máquina virtual, con la única diferencia de que ahora se seleccionará la imagen virtual ISO de Windows XP.

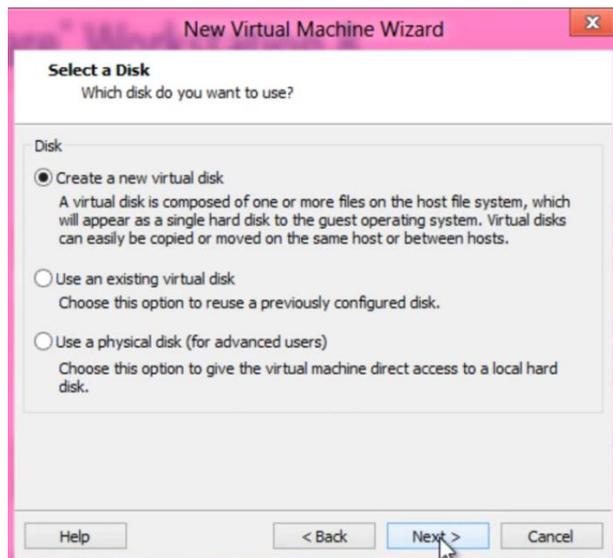
**Figura 138. Tamaño de memoria de la Máquina Virtual**



**Elaborado por: Christian Reyes - Daniel Rocha**

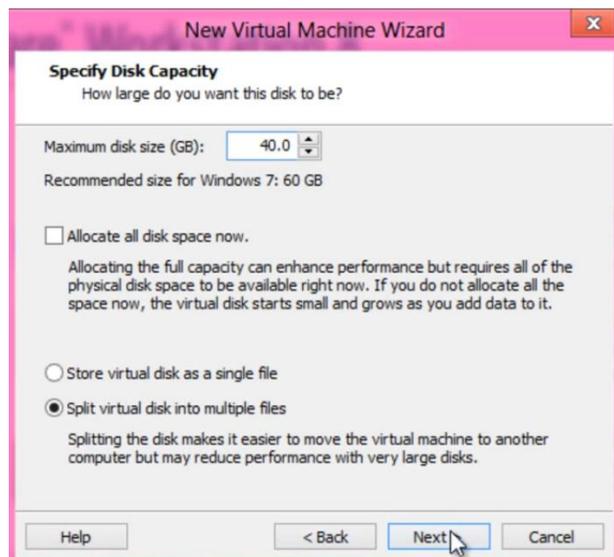
Al haber establecido el tamaño de la memoria a usarse en la máquina virtual, dar clic en “Next” y ahora se debe crear un disco duro en el que se instalará el sistema operativo seleccionado en la Imagen ISO de la figura 139. En la figura 140 y figura 141 se puede observar como se crea el disco duro y el tamaño que se establece al crearlo.

**Figura 139. Creación de Disco Duro de la Máquina Virtual**



**Elaborado por: Christian Reyes - Daniel Rocha**

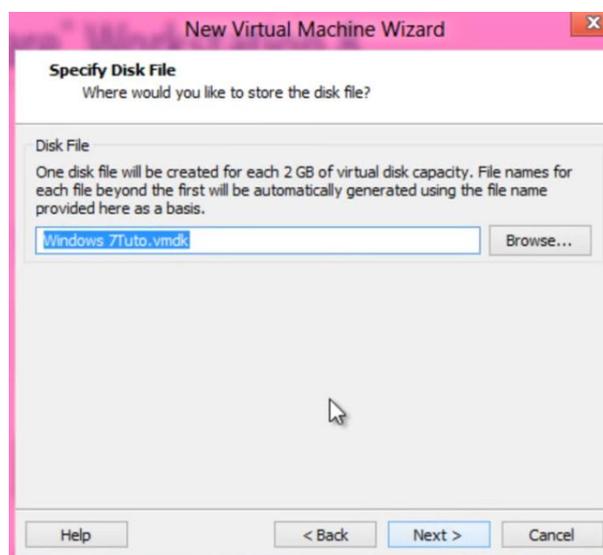
**Figura 140. Tamaño de Disco Duro de la Máquina Virtual**



**Elaborado por: Christian Reyes - Daniel Rocha**

Para finalizar con la creación del disco duro, se debe establecer un nombre al volumen de disco que se va a crear, esto se puede mirar en la figura 141.

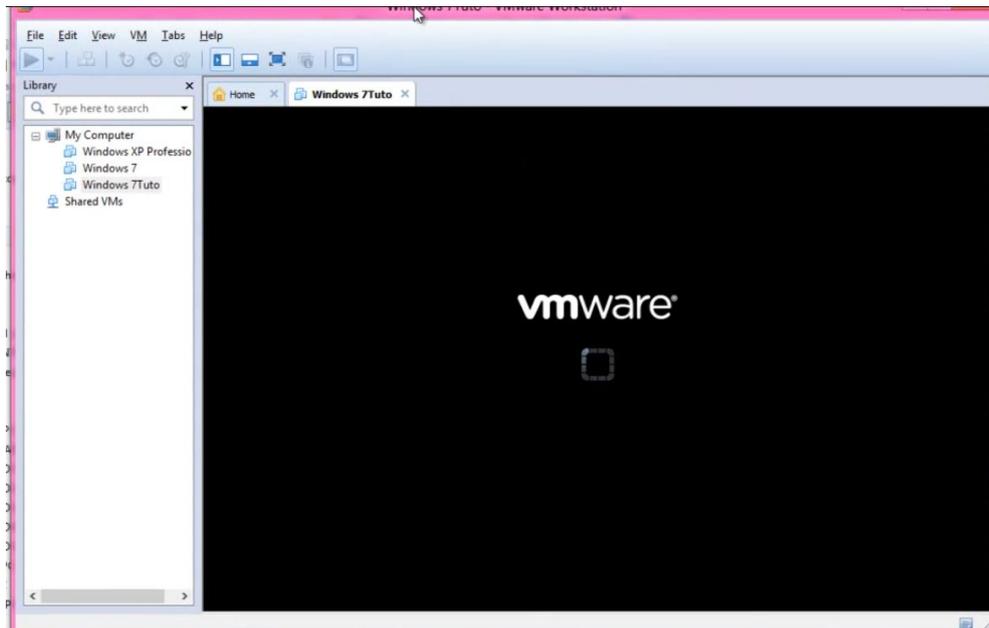
**Figura 141. Nombre del Disco Duro de la Máquina Virtual**



**Elaborado por: Christian Reyes - Daniel Rocha**

Al finalizar éste proceso, dar clic en Iniciar máquina virtual y se presentará una imagen igual a la la figura 142, y es la imagen resultante después de crear la máquina virtual XP o Win7.

**Figura 142. Inicio de Máquina Virtual**



Elaborado por: Christian Reyes - Daniel Rocha

### **4.3. Implementación de los módulos, y unificación de los mismos**

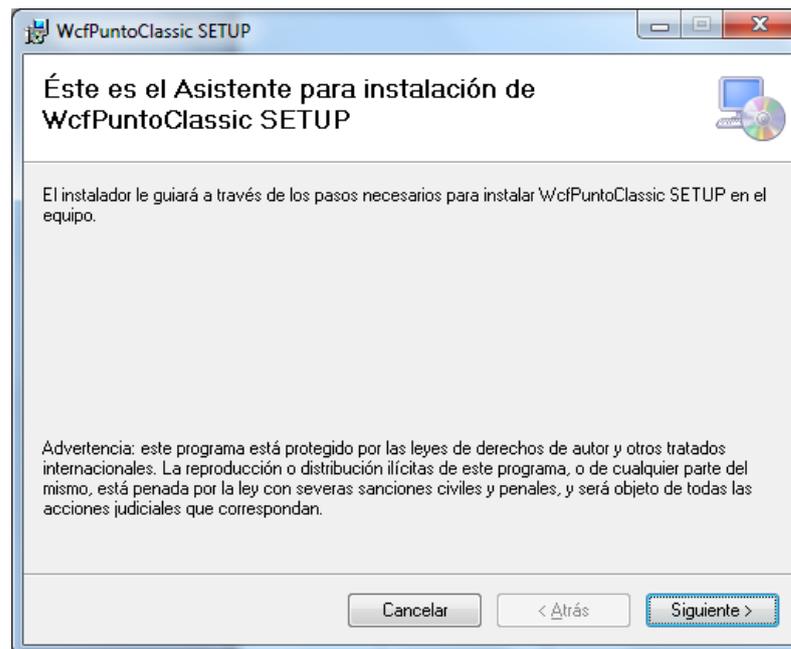
Para la implementación de los módulos del sistema en las máquinas virtuales es necesario dividir las tareas de instalación para su mejor comprensión.

#### **4.3.1. Instalación de los servicios web (win7)**

Una vez terminada la preparación de la máquina virtual 1, proceder a instalar los servicios web de Punto Classic dando doble clic en el archivo ejecutable del proyecto de WCF's  WcfPuntoClassic SETUP.msi

En la figura 143 se observa el inicio de la instalación de los servicios Web.

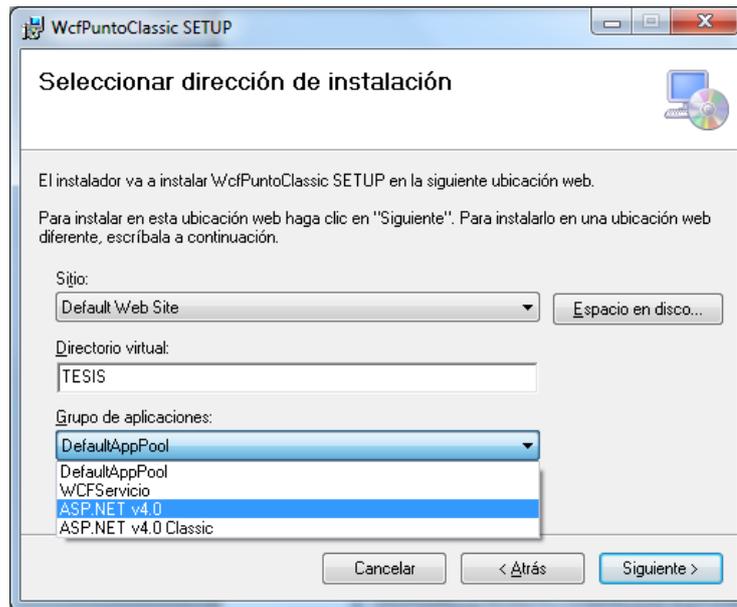
**Figura 143. Instalación de WCF's de Punto Classic (paso 1)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Después de haber dado clic en “Siguiente” del gráfico anterior, es necesario definir parámetros de instalación necesarios como son: el grupo de aplicaciones al que pertenecerá el sitio de los WCF's, en éste caso se debe seleccionar ASP NET v4.0 ya que fueron desarrollados en éste framework; el nombre del directorio Virtual con el que se lo reconocerá en el IIS; y el sitio al que pertenecerá el proyecto publicado, se lo puede dejar en el Sitio por defecto.

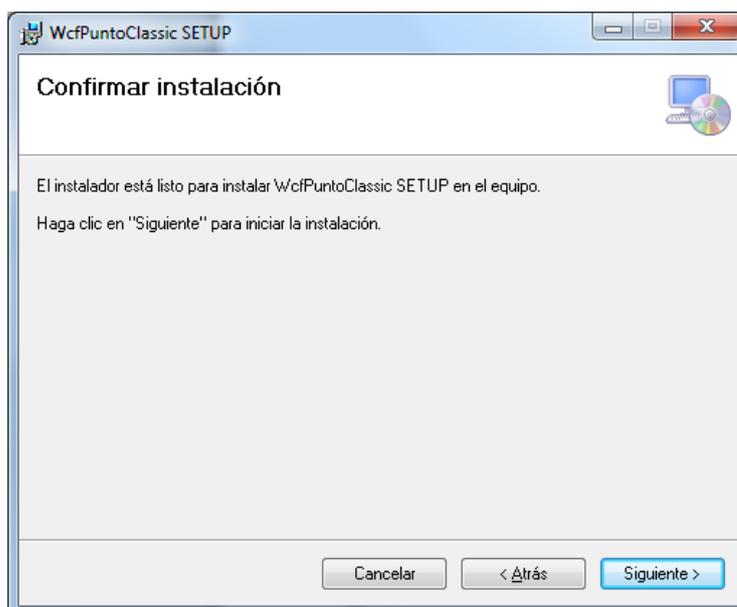
**Figura 144. Instalación de WCF's de Punto Classic (paso 2)**



**Elaborado por: Christian Reyes - Daniel Rocha**

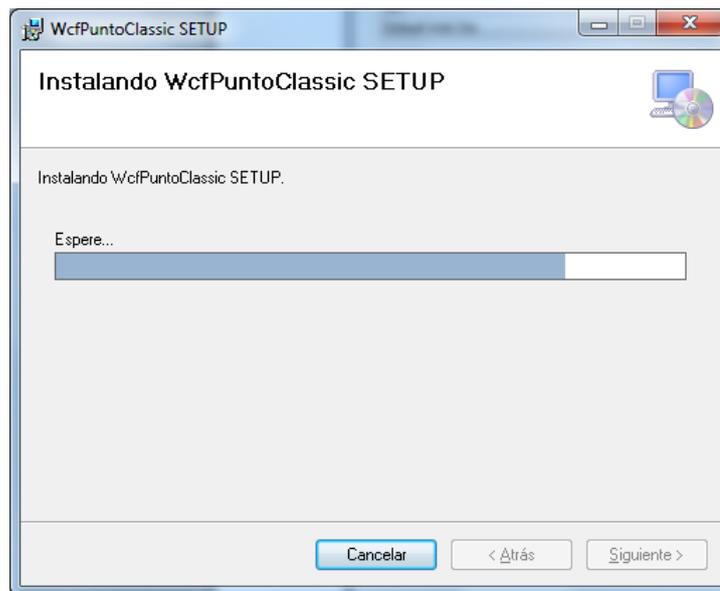
Al dar clic en “Siguiente” se presenta la imagen de la figura 145, donde se debe continuar para confirmar la instalación, después de esto comenzará a instalar y se presentará una imagen igual a la figura 146, y la figura 147 representa a la culminación de la instalación.

**Figura 145. Instalación de WCF's de Punto Classic (paso 3)**



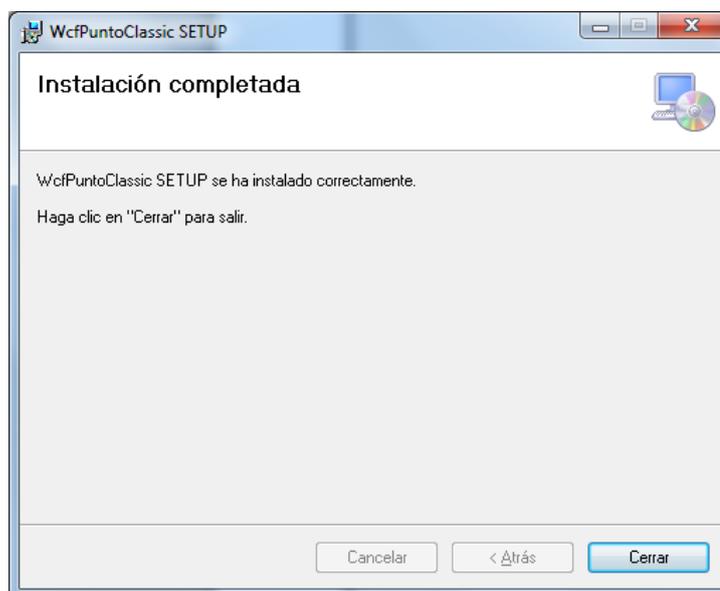
**Elaborado por: Christian Reyes - Daniel Rocha**

**Figura 146. Instalación de WCF's de Punto Classic (paso 4)**



**Elaborado por: Christian Reyes - Daniel Rocha**

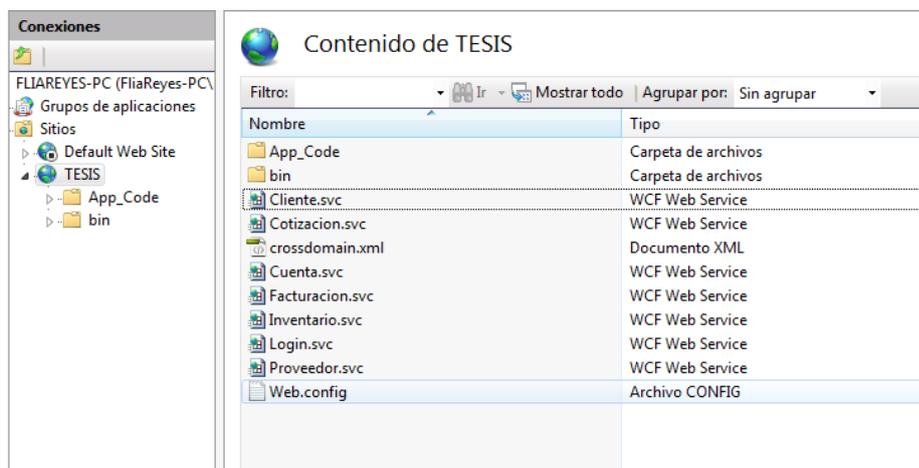
**Figura 147. Instalación de WCF's de Punto Classic (paso 5)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Una vez terminada la instalación se debe ingresar al “Administrador de Internet Information Services (IIS)” dentro de Herramientas Administrativas del Panel de Control, para verificar si fue correcta, y se obtendrá una imagen como la de la figura 148.

**Figura 148. Instalación de WCF's de Punto Classic (paso 6)**



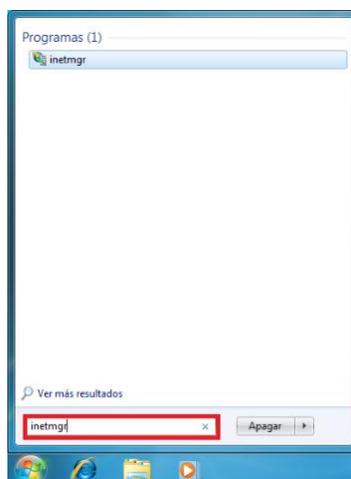
**Elaborado por: Christian Reyes - Daniel Rocha**

Para validar si los servicios fueron instalados correctamente se puede dar clic izquierdo en cualquier servicio de los instalados y luego clic en “Explorar” para ejecutar el servicio.

#### **4.3.2. Instalación de la aplicación web (win7)**

Para agregar el sitio web es necesario abrir el Administrador de IIS. Para ello, hacer clic en Inicio y, en el cuadro Iniciar búsqueda, escribe “inetmgr”. A continuación, pulsa Intro. Como se visualiza en el gráfico

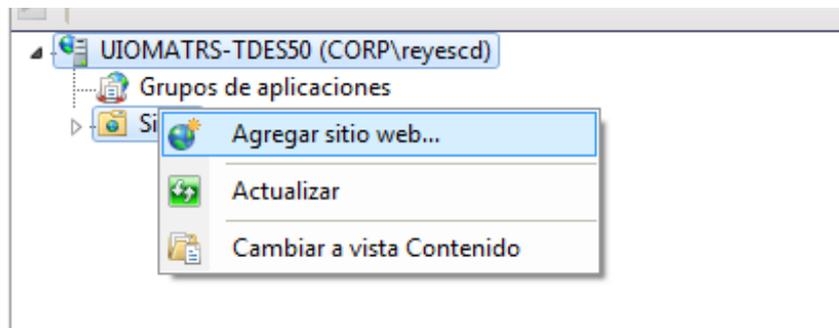
**Figura 149. Acceso a consola IIS**



**Elaborado por: Christian Reyes - Daniel Rocha**

En el panel Conexiones, hacer clic derecho en el nodo Sitios del árbol y dar clic en Agregar sitio web se observa a continuación:

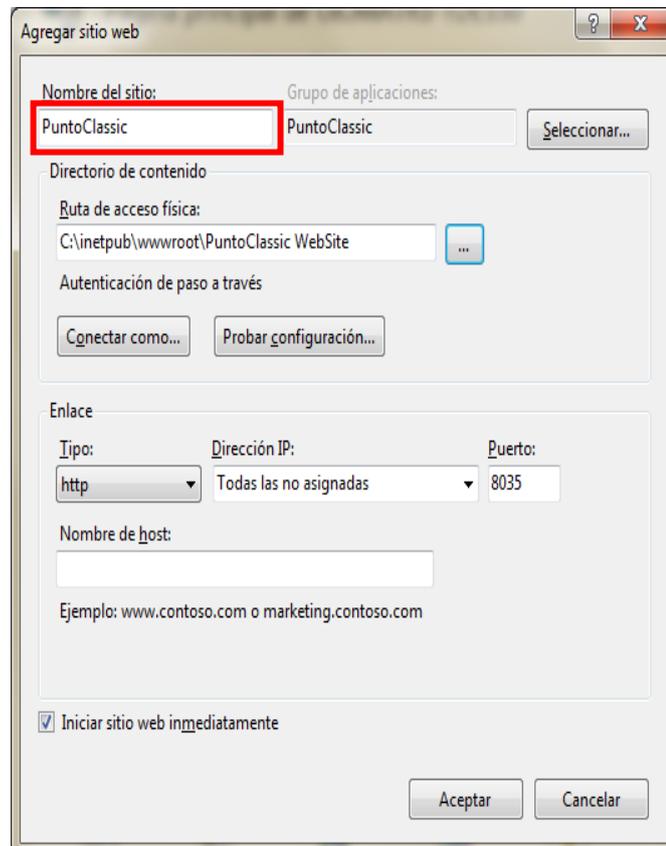
**Figura 150. Sitio Web Nuevo para Punto Classic**



Elaborado por: Christian Reyes - Daniel Rocha

En la ventana Agregar sitio web, escribir el nombre descriptivo para el sitio web en el cuadro Nombre del sitio, en éste caso “PuntoClassic”

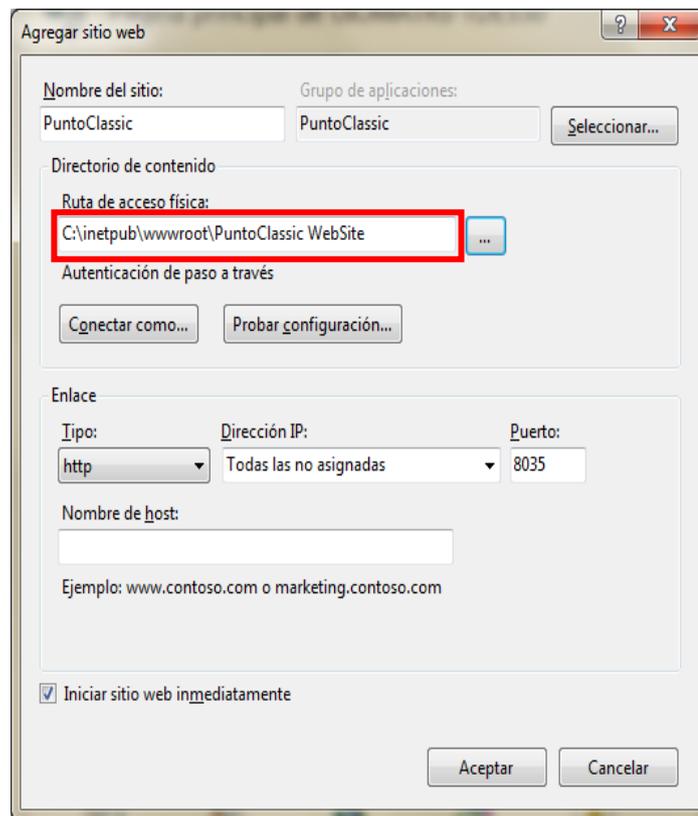
**Figura 151. Nombre del Sitio Web de Punto Classic**



Elaborado por: Christian Reyes - Daniel Rocha

En el cuadro “Ruta de acceso física”, de debe escribir la ruta de acceso de la carpeta del sitio web que contiene los archivos del mismo, en nuestro caso se creará una carpeta con el nombre “PuntoClassic WebSite” en la ruta “C:\inetpub\wwwroot\” donde se ha pegado las pantallas y archivos de configuración del sitio web.

**Figura 152. Ruta del Sitio Web de Punto Classic**

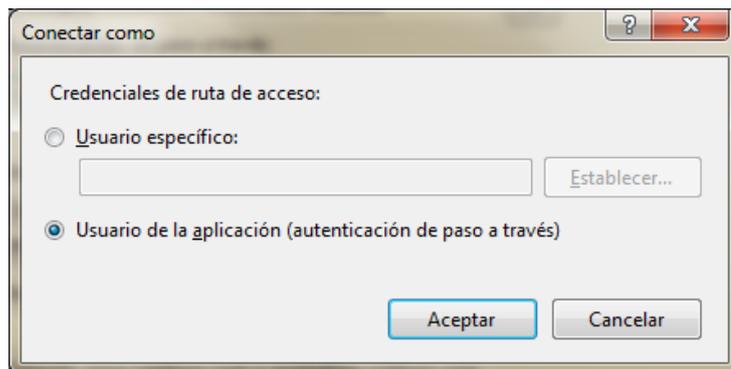


**Elaborado por: Christian Reyes - Daniel Rocha**

Si se desea, se puedes hacer clic en el botón de exploración (...), y seleccionar la carpeta de la ruta indicada.

Al dar clic en “Conectar como...” se presenta la figura 153, se selecciona la opción “Usuario de la aplicación”, dar clic en Aceptar.

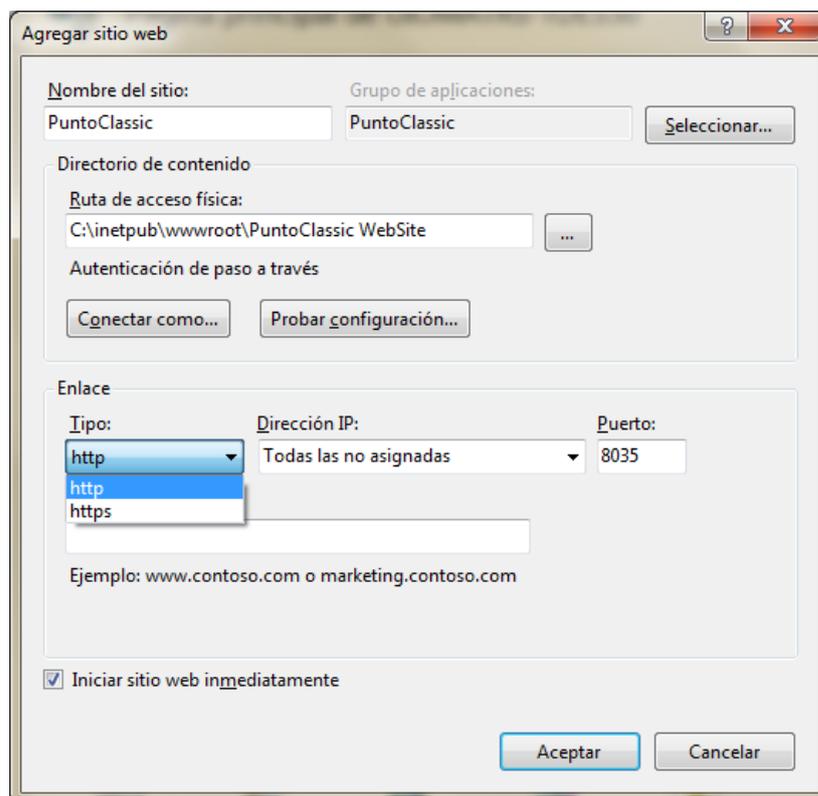
**Figura 153. Configuración Conectar Como**



**Elaborado por: Christian Reyes - Daniel Rocha**

Selecciona el protocolo web en la lista "Tipo", se muestra en la figura 154.

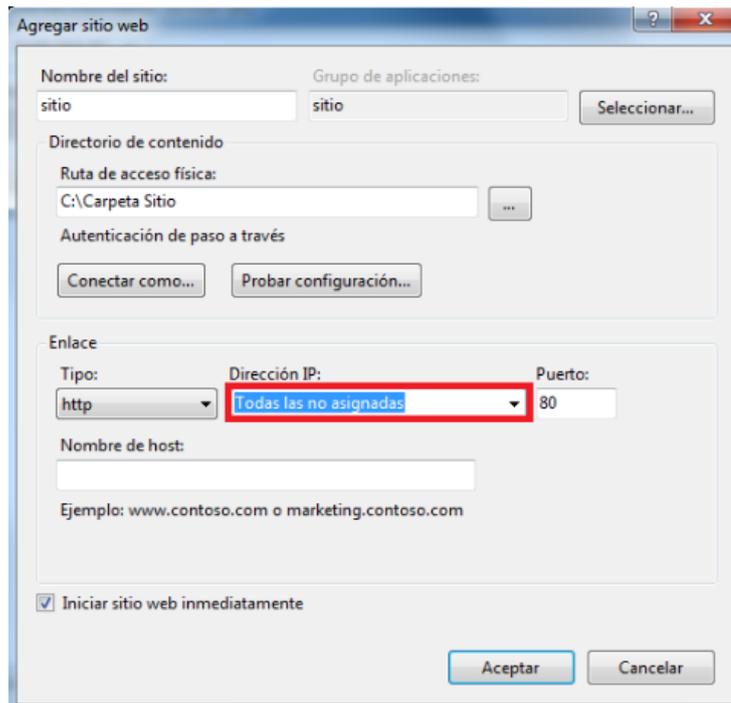
**Figura 154. Configuración Protocolo del Sitio Web de Punto Classic**



**Elaborado por: Christian Reyes - Daniel Rocha**

Si se desea especificar una dirección IP estática, escríbala en el cuadro Dirección IP.

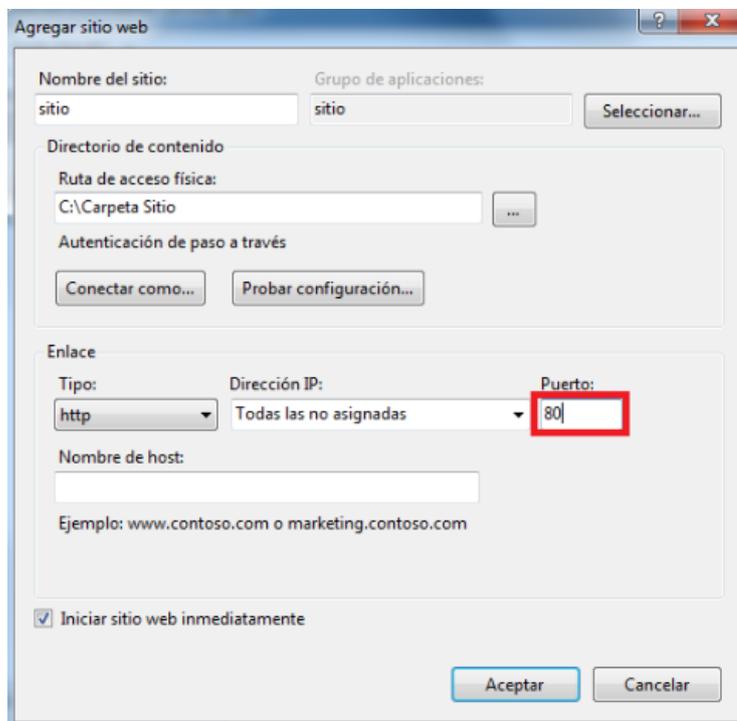
**Figura 155. Configuración dirección IP del Sitio Web de Punto Classic**



Elaborado por: Christian Reyes - Daniel Rocha

Ahora escribir un número de puerto en el cuadro Puerto.

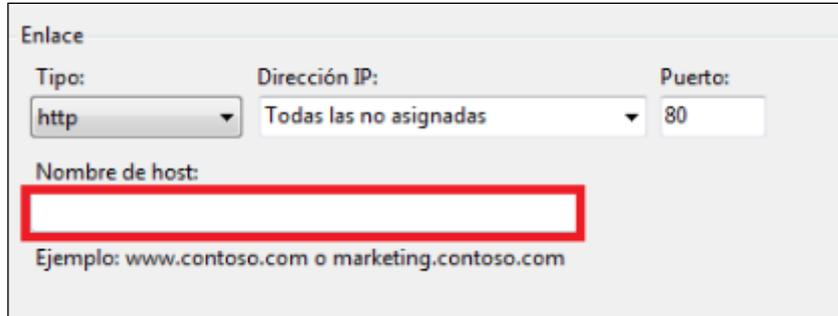
**Figura 156. Configuración del puerto para el Sitio Web de Punto Classic**



Elaborado por: Christian Reyes - Daniel Rocha

Si se desea, se puede escribir un nombre de encabezado del host para el sitio web en el cuadro “Nombre de host”.

**Figura 157. Configuración Nombre del Host del Sitio Web de Punto Classic**



Enlace

Tipo: Dirección IP: Puerto:

http Todas las no asignadas 80

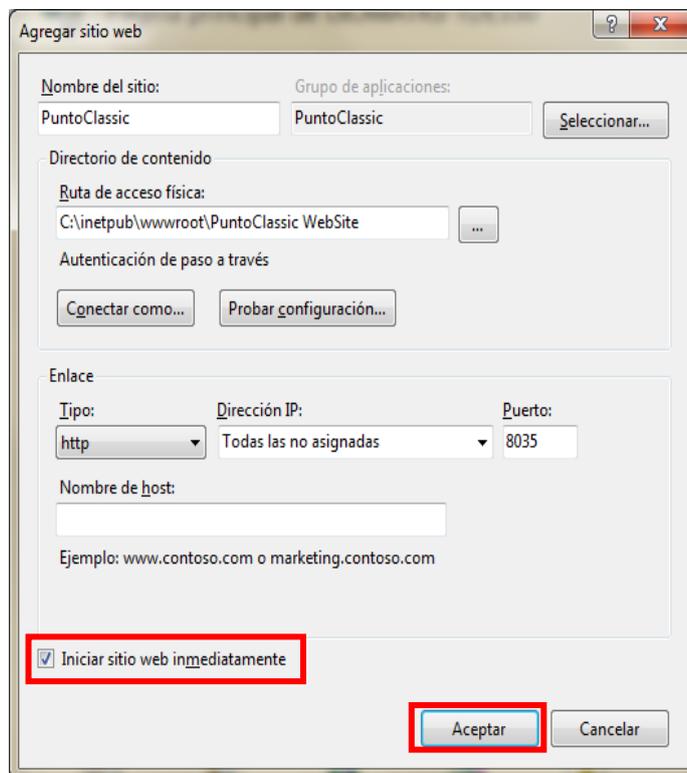
Nombre de host:

Ejemplo: www.contoso.com o marketing.contoso.com

Elaborado por: Christian Reyes - Daniel Rocha

Activar la casilla “Iniciar sitio web inmediatamente”, para dejar el sitio disponible inmediatamente al finalizar la publicación. Finalmente, dar clic en Aceptar.

**Figura 158. Configuración inicio por Defecto del Sitio Web de Punto Classic**



Agregar sitio web

Nombre del sitio: PuntoClassic Grupo de aplicaciones: PuntoClassic Selecionar...

Directorio de contenido

Ruta de acceso física: C:\inetpub\wwwroot\PuntoClassic WebSite ...

Autenticación de paso a través

Conectar como... Probar configuración...

Enlace

Tipo: Dirección IP: Puerto:

http Todas las no asignadas 8035

Nombre de host:

Ejemplo: www.contoso.com o marketing.contoso.com

Iniciar sitio web inmediatamente

Aceptar Cancelar

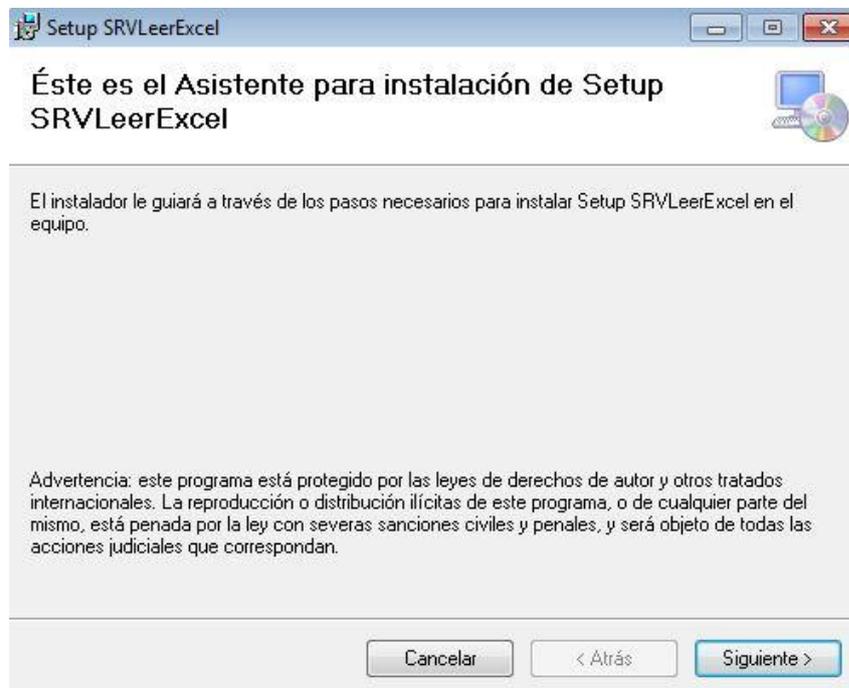
Elaborado por: Christian Reyes - Daniel Rocha

### 4.3.3. Instalación del servicio windows (win7)

Para instalar el servicio Windows que se encargará de comunicar al módulo financiero y al módulo de simulación industrial por medio de la lectura de un archivo Excel generado en el simulador, es necesario dar doble clic en el instalador del servicio Windows .

Se tendrá una imagen igual a la figura 159, se debe dar clic en “Siguiente” y se mostrará la imagen de la figura 159.

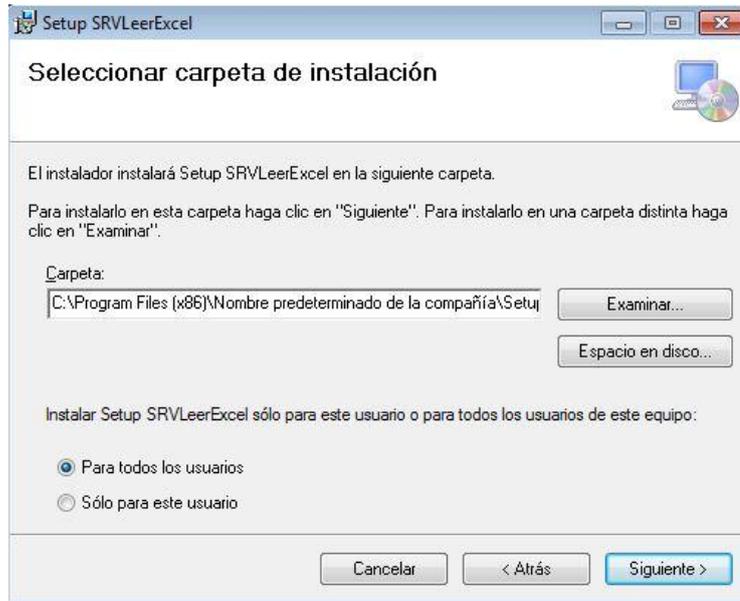
**Figura 159. Instalación del Servicio Windows (paso 1)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Dar clic en “Siguiente”

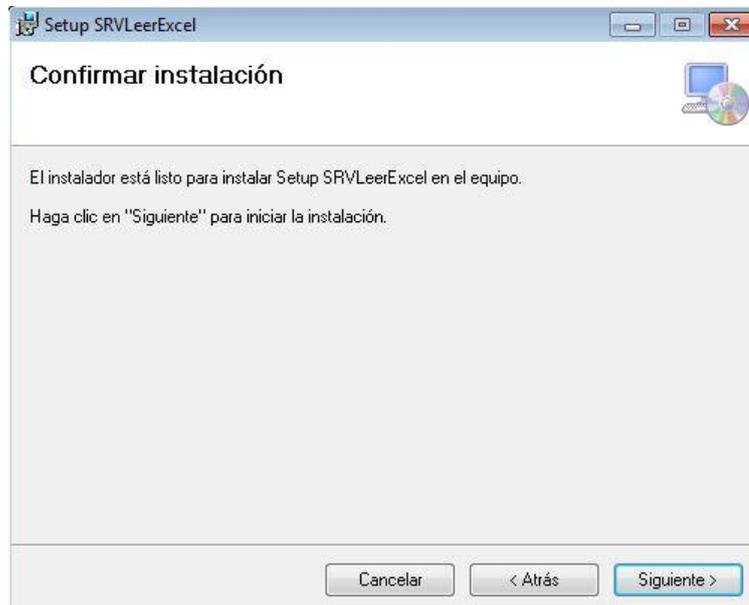
**Figura 160. Instalación del Servicio Windows (paso 2)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Nuevamente dar clic en “Siguiente”

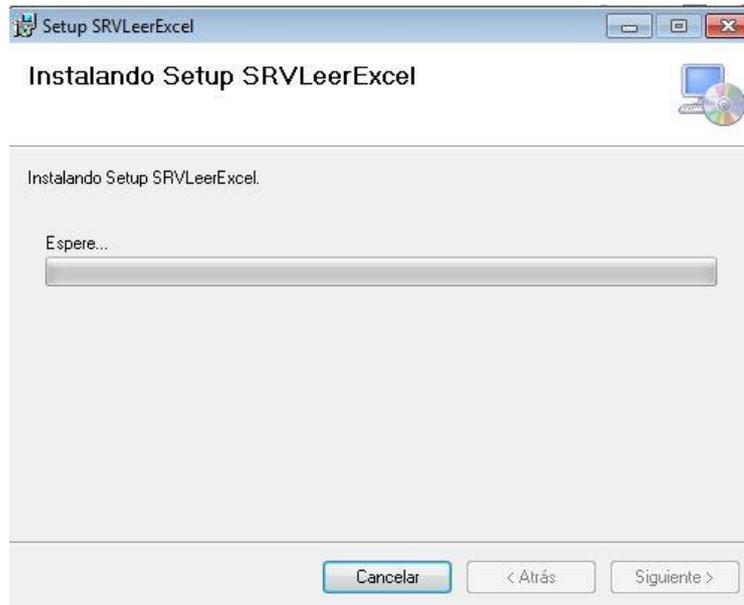
**Figura 161. Instalación del Servicio Windows (paso 3)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Se observa el progreso de la instalación

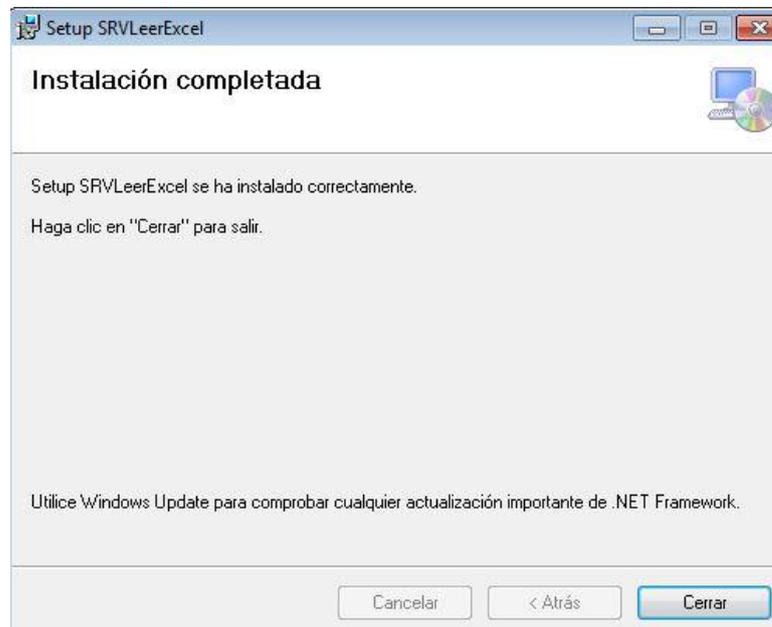
**Figura 162. Instalación del Servicio Windows (paso 4)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Dar clic en “Cerrar” al finalizar la instalación de manera correcta.

**Figura 163. Instalación del Servicio Windows (paso 5)**



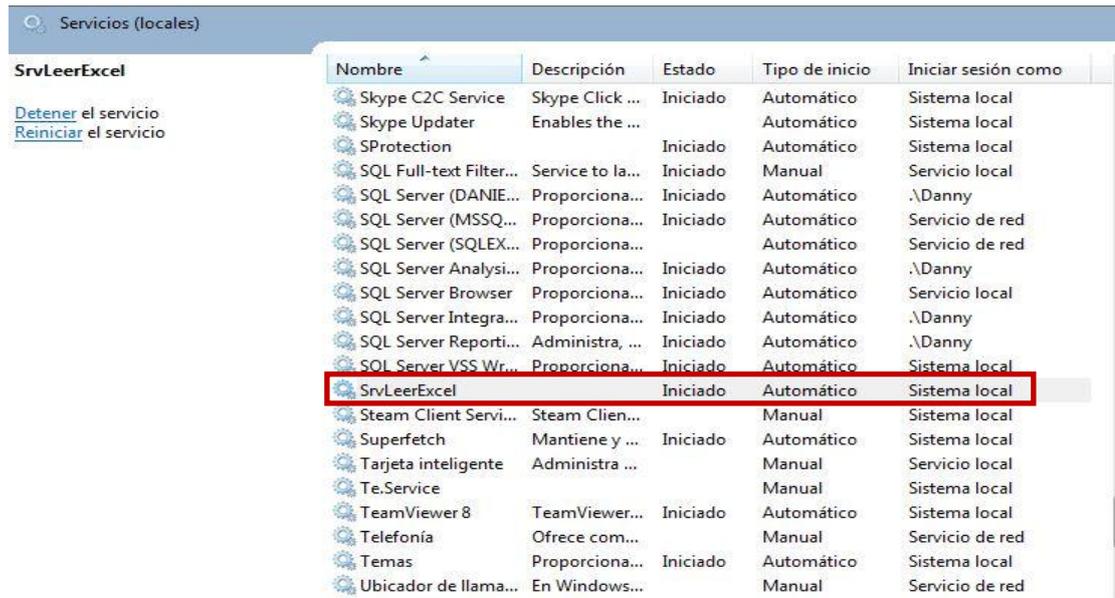
**Elaborado por: Christian Reyes - Daniel Rocha**

Al finalizar la instalación del servicio Windows acceder a la consola de administración de servicios locales dentro de “Herramientas Administrativas” en el

panel de control de Windows, y buscar el servicio “SrvLeerExcel”, dar clic derecho y luego clic en “Iniciar”.

De éste modo el servicio que se ha instalado arrancará, esto se vizualiza en el siguiente gráfico.

**Figura 164. Instalación del Servicio Windows (paso 6)**



**Elaborado por: Christian Reyes - Daniel Rocha**

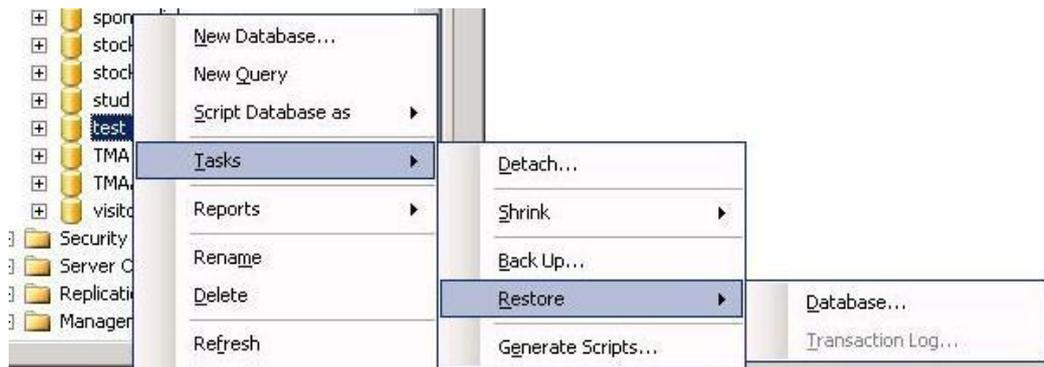
#### 4.3.4. Instalación de la base de datos (win7)

Para restaurar la base de datos desde el archivo de copia de seguridad de la Base de Datos de Punto Classic, es necesario seguir los pasos que se indican a continuación:

Abrir el Microsoft SQL Server Management Studio y conectarse a la base de datos creada en la que se desea restaurar la copia de seguridad.

Seleccionar la base de datos >> Hacer clic derecho >> tareas >> Restauración >> Base de datos como se muestra en la figura 165.

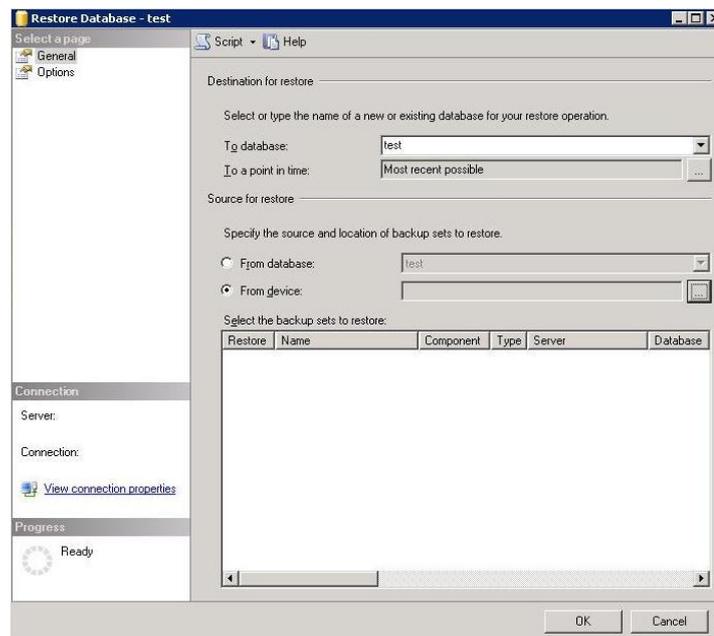
**Figura 165. Restauración de la base de datos de Punto Classic (paso 1)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Aparecerá la pantalla "Restaurar base de datos". Seleccionar "Desde dispositivo" en la opción "Origen de la restauración" y dar clic en el botón contiguo para seleccionar el dispositivo en el que se encuentra el backup de la base de datos (Figura 166):

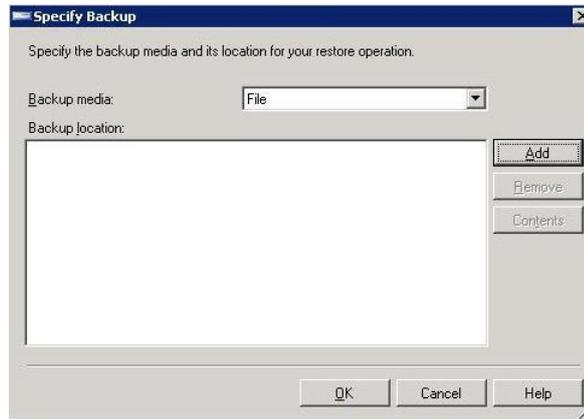
**Figura 166. Restauración de la base de datos de Punto Classic (paso 2)**



**Elaborado por: Christian Reyes - Daniel Rocha**

Seleccionar la opción "Hacer copia de seguridad como archivo" y hacer clic en el botón "Add" para añadir la ubicación del archivo de copia de seguridad (Figura 167)

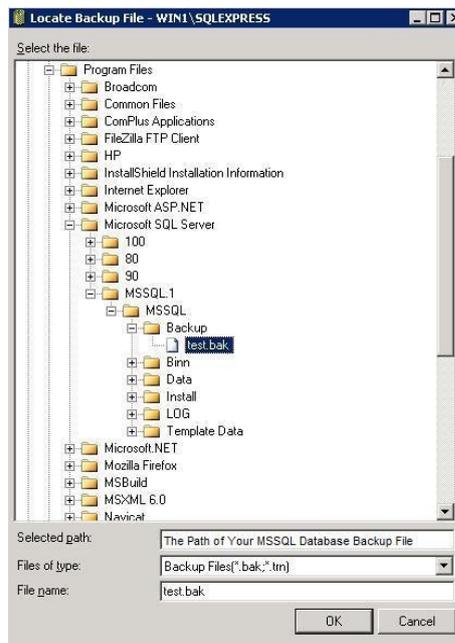
**Figura 167. Restauración de la base de datos de Punto Classic (paso 3)**



Elaborado por: Christian Reyes - Daniel Rocha

Seleccionar el archivo de copia de seguridad que se desea restaurar y pulse el botón OK (Figura 168).

**Figura 168. Restauración de la base de datos de Punto Classic (paso 4)**



Elaborado por: Christian Reyes - Daniel Rocha

#### 4.3.5. Instalación del simulador (winxp)

Para la instalación del simulador desarrollado, únicamente es necesario copiar la carpeta contenedora del proyecto intouch en la carpeta de proyectos locales de la

máquina virtual XP y seguido de esto, dar doble clic para abrir intouch e importar el proyecto creado.

#### **4.4. Pruebas y resultados**

Para la implementación de las pruebas y obtención de resultados dentro del sistema, específicamente en los servicios web ya que son el motor fundamental de todo el aplicativo se usará la herramienta JMETER de Apache.



JMeter es una herramienta de carga para llevar a cabo simulaciones sobre cualquier recurso de Software.

Inicialmente diseñada para pruebas de estrés en aplicaciones web, hoy en día, su arquitectura ha evolucionado no sólo para llevar a cabo pruebas en componentes habilitados en Internet (HTTP), sino además en Bases de Datos, programas en Perl , requisiciones FTP y prácticamente cualquier otro medio.

Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de requisiciones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción.

En este sentido, simula todas las funcionalidades de un Navegador ("Browser"), o de cualquier otro cliente.

En base a esta herramienta se obtuvo los siguientes resultados en la simulación con 20.000 peticiones concurrentes a los servicios web, los mismos que se muestran en los gráficos 169, 170 y 171.

**Figura 169. Pruebas de Stress y Calidad (paso 1)**

Reporte resumen

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo   Log/Mostrar sólo:  Escribir en Log  Sólo Errores  Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
Petición Web...	20000	9	2	205	19,29	0,00%	453,9/sec	460,96	1040,0
Total	20000	9	2	205	19,29	0,00%	453,9/sec	460,96	1040,0

¿Incluir el nombre del grupo en la etiqueta?   Guardar la cabecera de la tabla

**Elaborado por: Christian Reyes - Daniel Rocha**

En la figura 169 se observa el número de muestras enviadas como solicitud a los servicios web con la cantidad de 20.000, estas son enviadas en un paquete de 200 hilos y 100 peticiones por cada hilo, es decir 100 peticiones por cada cliente usando el sistema web.

Mientras que en la figura 170 se observan las salidas de las peticiones realizadas, todas sin excepción son correctas, indicando que las solicitudes realizadas tienen una respuesta relativamente favorable.

Pero en la figura 171 se observa la curva generada en las peticiones realizadas, el máximo de la curva en milisegundos (ms), es poco mas de 205 indicando que los servicios web acaban de ser levantados y posterior a esto se tiene respuestas en el sistema con un mínimo de 2ms.

De esta manera se comprueba la efectividad del aplicativo en funcionamiento con carga referencial un sistema de alta transaccionalidad.

**Figura 170. Pruebas de Stress y Calidad (paso 2)**

Ver Resultados en Árbol

Nombre: Ver Resultados en Árbol

Comentarios

Escribir todos los datos a Archivo

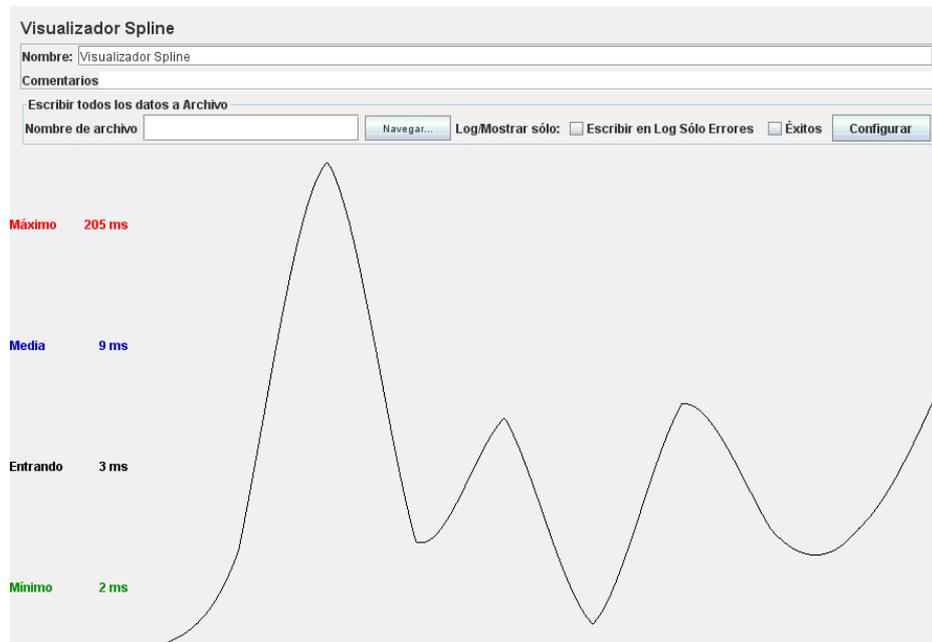
Nombre de archivo   Log/Mostrar sólo:  Escribir en Log Sólo Errores  Éxitos

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Mues...	Estado	Bytes	Latency
1	02:49:47.931	Grupo de Hilos 1-1	Petición WebServ...	11		1040	0
2	02:49:47.937	Grupo de Hilos 1-2	Petición WebServ...	7		1040	0
3	02:49:47.941	Grupo de Hilos 1-3	Petición WebServ...	5		1040	0
4	02:49:47.948	Grupo de Hilos 1-4	Petición WebServ...	7		1040	0
5	02:49:47.953	Grupo de Hilos 1-5	Petición WebServ...	7		1040	0
6	02:49:47.958	Grupo de Hilos 1-6	Petición WebServ...	5		1040	0
7	02:49:47.964	Grupo de Hilos 1-7	Petición WebServ...	7		1040	0
8	02:49:47.971	Grupo de Hilos 1-8	Petición WebServ...	5		1040	0
9	02:49:47.977	Grupo de Hilos 1-9	Petición WebServ...	7		1040	0
10	02:49:47.981	Grupo de Hilos 1-...	Petición WebServ...	5		1040	0
11	02:49:47.986	Grupo de Hilos 1-...	Petición WebServ...	4		1040	0
12	02:49:47.992	Grupo de Hilos 1-...	Petición WebServ...	5		1040	0
13	02:49:47.999	Grupo de Hilos 1-...	Petición WebServ...	13		1040	0
14	02:49:48.014	Grupo de Hilos 1-...	Petición WebServ...	8		1040	0
15	02:49:48.016	Grupo de Hilos 1-...	Petición WebServ...	15		1040	0
16	02:49:48.013	Grupo de Hilos 1-...	Petición WebServ...	20		1040	0
17	02:49:48.033	Grupo de Hilos 1-...	Petición WebServ...	34		1040	0
18	02:49:48.020	Grupo de Hilos 1-...	Petición WebServ...	47		1040	0
19	02:49:48.038	Grupo de Hilos 1-...	Petición WebServ...	34		1040	0
20	02:49:48.045	Grupo de Hilos 1-...	Petición WebServ...	27		1040	0
21	02:49:48.055	Grupo de Hilos 1-...	Petición WebServ...	26		1040	0
22	02:49:48.065	Grupo de Hilos 1-8	Petición WebServ...	17		1040	0
23	02:49:48.060	Grupo de Hilos 1-...	Petición WebServ...	23		1040	0
24	02:49:48.067	Grupo de Hilos 1-4	Petición WebServ...	18		1040	0
25	02:49:48.049	Grupo de Hilos 1-...	Petición WebServ...	36		1040	0

Scroll automatically?  Child samples? No. de Muestras 20000 Última Muestra 3 Media 9 Desviación 19

Elaborado por: Christian Reyes - Daniel Rocha

**Figura 171. Pruebas de Stress y Calidad (paso 3)**



Elaborado por: Christian Reyes - Daniel Rocha

## CONCLUSIONES

- La comunicación con las personas de Punto Classic representó una clave fundamental para poder validar los pre-requisitos y cumplir con las necesidades o requerimientos.
- Diseñar la aplicación, utilizando la herramienta de modelado de sistemas, permitió tener una visión detallada del mismo, en función de los diferentes diagramas realizados.
- El análisis del sistema actual a través de los diagramas de casos de uso y de objetos, facilitó la determinación de los requerimientos ya que a través de ellos se realizó la identificación, definición y recolección de los requisitos necesarios para el desarrollo de la aplicación que se centran en el usuario y sus necesidades.
- El desarrollo de un sistema de información, no hace referencia exclusivamente a la tarea de codificación, se enfatiza en una serie de pasos o procedimientos para la creación de un producto, incluyendo también aspectos como el modelado del negocio, las tareas de análisis y diseño.
- El uso de Windows Communication Foundation ayuda a la comunicación entre aplicativos que exponen diferentes interfaces y permiten implementar una arquitectura SOA con herramientas Microsoft.
- Realizar servicios que contengan una funcionalidad compleja, pero que se ejecuten una sola vez es más eficiente que tener servicios con funcionalidad simple que se ejecuten varias veces ya existe demasiada transferencia de información.
- El uso de Linq da mejoras en la capa de acceso a datos ya que permite un desarrollo más rápido, tener un código más entendible ya que las acciones se

encuentran en código C# y protege contra problemas de seguridad como SQL Injection en la base de datos.

- El uso de Silverlight en la capa de presentación provee interfaces de usuario más amigables, permiten el desarrollo en XAML que es muy entendible, junto con el lenguaje C# que brinda un mejor control sobre el aplicativo.
- Para desarrollar y realizar procesos de actualización de la información de inventarios según la producción realizada el utilizar servicios Windows ayuda a efectuar acciones en tiempos definidos.
- Para crear un proyecto de simulación en Intouch, se debe tener muy claro las entradas y salidas visuales que se desea crear en el mismo, ya que, dependiendo del tipo de licencia que se haya adquirido para Intouch, se tendrá el número de TAG's disponibles en el desarrollo de la aplicación.
- Intouch es una excelente herramienta de simulación Industrial, por la facilidad de uso y el gran conjunto de librerías que posee para simular procesos, además de que mantiene un modelo de diseño de las aplicaciones muy fácil de entender, dispone de un lenguaje de programación sencillo y extenso para la realización de cálculos en segundo plano, simulaciones, etc, que se usa de acuerdo a la necesidad del programador, haciéndolo muy intuitivo y amigable.

## RECOMENDACIONES

- Es necesario realizar con detalle las preguntas sobre la lógica de negocios para así tener en claro las especificaciones funcionales y realizar la menor cantidad de cambios en transcurso del desarrollo.
- Se tiene que realizar los diagramas especificando cada acción que se encuentra en la lógica de negocios, para de esta manera poder tener un mejor análisis en la etapa de desarrollo.
- En el uso de Windows Communication Foundation toca definir bien la estructura de los servicios que se van a exponer asociando por funcionalidad de uso, teniendo en cuenta que los servicios tienen que ser reutilizables.
- Al realizar servicios con lógica de negocio compleja, tener en cuenta los tiempos de respuesta, realizar el servicio de manera transaccional por si se producen errores en procesos intermedios del servicio.
- Se debe tener una correcta definición de las tablas en la base de datos para que de esta forma cuando se especifiquen en clases se tenga una definición clara y esto permita manejar de mejor manera la capa de acceso a datos.
- Es necesario tener cuidado en la capa de presentación ya que los WCF son invocados de manera asíncrona, por lo cual se pueden invocar varios en conjunto sin que finalice el primero, en operaciones que se tenga que ejecutar uno después de la respuesta de otro es necesario realizar este control.
- Es necesario tener bien parametrizados los tiempos en los que se ejecutan las invocaciones desde un servicio Windows según las capacidades del servidor para no afectar los demás procesos.

- Se debe tener muy en cuenta que para diseñar un gran proyecto de simulación en Intouch, es necesario adquirir una licencia acorde al número de TAG's que aproximadamente se usarán en el transcurso del desarrollo.
- Se recomienda guardar continuamente los proyectos diseñados en Intouch, y mucho más si la licencia que se mantiene en el software es una licencia de prueba, ya que, cada cierto tiempo se dispara una alerta de la licencia, haciendo que el programa se cierre y podría provocar que el trabajo realizado se eche a perder.

## LISTA DE REFERENCIAS

- **Libros**

al, K. D. (2009). *Microsoft SQL Server 2008 Internals*. Microsoft Press.

Barrett, F. J. (2012). *Visual Studio 2012: SQL Server Data Tools*. SkillSoft Ireland Limited.

Berzal, F. y. (2005). *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. iKor Consulting.

Bhattacharya, K. G. (2008). *Service Oriented Architecture (SOA) Field Guide for Executives*. John Wiley & Sons.

Dockx, G. C. (2010). *Microsoft Silverlight 4 Data and Services Cookbook: Over 85 Practical Recipes for Creating Rich, Data-driven Business Applications in Silverlight*. Packt Publishing.

Gaudioso, V. (2010). *Foundation Expression Blend 4 with Silverlight*. Friends of Ed.

Klein, S. (2007). *Professional WCF Programming: .NET Development with the Windows Communication Foundation*. Wrox Press.

Lair, R. (2012). *Beginning Silverlight 5 in C#, Fourth Edition*. Apress.

Leonard Lobel, A. J. (2009). *Programming Microsoft SQL Server 2008*. Microsoft Press.

Liu, M. (2008). *WCF Multi-tier Services Development with LINQ: Build SOA Applications on the Microsoft Platform in This Hands-on Guide*. Packt Publishing.

Liu, M. (2010). *WCF 4.0 Multi-tier Services Development with LINQ to Entities: Build SOA Applications on the Microsoft Platform With This Hands-on Guide Updated for VS2010*. Packt Publishing.

Louden, K. (2002). *Lenguajes de Programación: Principios y Practicas*. Cengage Learning Editores.

M., B. W. (2002). *UML with Rational Rose 2002*. SYBEX.

Mateu, C. (2004). *Desarrollo de Aplicaciones Web*. Barcelona: Eureka Media.

Moroney, L. (2009). *Introducing Microsoft Silverlight 3*. Microsoft Press.

Pablo Cibraro, K. C. (2010). *Professional WCF 4: Windows Communication Foundation with .NET 4*. Wrox Press.

Pathak, N. (2011). *Pro WCF 4: Practical Microsoft SOA Implementation, Second Edition*. Apress.

Roger., P. (2002). *Ingeniería del Software: Un enfoque práctico*. Mc Graw Hill.

- **NetGrafía**

Anónimo. (15 de 10 de 2010). *Definición de Modelo de Datos*. Recuperado el 17 de 02 de 2012, de Definición de Modelo de Datos: <http://definicion.de/modelo-de-datos>

Anónimo. (18 de 02 de 2010). *El Modelo de Datos Entidad - Relación*. Recuperado el 24 de 07 de 2012, de El Modelo de Datos Entidad - Relación: <http://www.cs.us.es/cursos/bd-2005/HTML/modeloER.htm>

Barzanallana, R. (1 de 06 de 2005). *lagb06.pdf*. Recuperado el 24 de 04 de 2012, de lagb06.pdf: <http://www.um.es/docencia/barzana/IAGP/lagp06.pdf>

Chávez, J. M. (4 de septiembre de 2006). *programandoideas*. Recuperado el 23 de noviembre de 2011, de programandoideas: <http://programandoideas.com/el-lenguaje-de-programacion-php/>

J.C. (17 de 01 de 2008). *OPC Scada*. Recuperado el 20 de 12 de 2012, de OPC Scada: [http://www.lawebdelprogramador.com/foros/SCADA/1025323-intouch\\_reporte.html](http://www.lawebdelprogramador.com/foros/SCADA/1025323-intouch_reporte.html)

Jiménez, R. (09 de 05 de 2003). *Análisis y diseño orientado a objetos de un framework para el modelado estadístico con MLG*. Recuperado el 02 de 05 de 2012, de Análisis y diseño orientado a objetos de un framework para el modelado estadístico con MLG: <http://www.tdx.cat/TDX-0713104-112553/index.html>

Luque, J. M. (14 de 01 de 2013). *Cominucación entre Intouch y Excel*. Recuperado el 18 de 02 de 2013, de Cominucación entre Intouch y Excel: <http://www.infopl.net/descargas/49-wonderware/1698-comunicacion-entre-intouch-y-excel>

Marañón, G. Á. (1 de enero de 1999). Recuperado el 22 de noviembre de 2011, de <http://www.iec.csic.es/criptonomicon/java/qesjava.html>

MDSN, M. (11 de 08 de 2010). *Usar WSDL*. Recuperado el 23 de 05 de 2012, de Usar WSDL: <http://msdn.microsoft.com/es-es/library/ms175476%28v=sql.105%29.aspx>

Perez, M. (17 de 06 de 2009). *Curso Intouch 8.0*. Recuperado el 13 de 12 de 2012, de Curso Intouch 8.0: <http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0CFYQFjAG&url=http%3A%2F%2Fwww.etitudela.com%2Felectricidad%2Fdownloads%2Fcursointouch8.doc&ei=U12xUYrWKJHG4AOPtYDgBQ&usg=AFQjCNHdLhX8kvnqC2Xal7iNshbPb8B93g&sig2=GygZwi7IM7lq2ZYAoZQdkw&>

- Pillalaza, E. (17 de 07 de 2006). *Tesis : Construcción de un módulo didáctico para el aprendizaje y operación de controladores lógicos programables, monitoreado mediante el software INTOUCH*. Recuperado el 15 de 01 de 2013, de Tesis : Construcción de un módulo didáctico para el aprendizaje y operación de controladores lógicos programables, monitoreado mediante el software INTOUCH: <http://bibdigital.epn.edu.ec/bitstream/15000/2458/1/CD-0169.pdf>
- Seco, J. A. (15 de marzo de 2007). enpalermo. Recuperado el 21 de noviembre de 2011, de enpalermo: <http://www.enpalermo.com/descargas/manuales/el-lenguaje-de-programacion-c-espanol.html>
- Sol., M. R. (7 de 08 de 2010). *Diseño e implantación de un sistema SCADA para una planta de producción y envasado de líquidos*. Recuperado el 15 de 01 de 2013, de Diseño e implantación de un sistema SCADA para una planta de producción y envasado de líquidos: <http://www.recercat.cat/bitstream/handle/2072/13784/PFC%20Manel%20Redondo%20Sol.pdf?sequence=1>

## GLOSARIO

- **Apache**

APACHE es el servidor de páginas web.

- **API**

Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

- **Backup**

Un backup es en realidad una copia de seguridad, y nada es más previsor que realizarlo para obtener réplicas adicionales de nuestra información y recurrir a ellas mediante restauración en caso de que se produzca una eventual pérdida de datos.

- **Batch**

Un proceso batch es un ciclo donde procesar muchos registros uno tras otro sin que intervenga interactivamente el usuario.

- **Browser**

También conocido como navegador Web.

- **CardSpace**

Tarjeta de presentación o tarjeta de trabajo

- **C#**

Lenguaje de programación orientado a objeto.

- **Dll**  
Un archivo DLL (Dynamic Library Link) es un módulo componente de un programa que ejecuta alguna función..
  
- **Framework**  
En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.
  
- **FTP**  
Protocolo de Transferencia de Archivos
  
- **Fuente RSS**  
RSS son las siglas de Really Simple Syndication, un formato XML para syndicar o compartir contenido en la webHMI
  
- **IIS**  
Internet information server, servidor web de Windows.
  
- **JAVA**  
Lenguaje de programación orientado a objetos
  
- **JMeter**  
Aplicación Java para medición de estándares de calidad y prueba de aplicaciones Web.
  
- **Lote**  
Grupo de procesos

- **MAC**  
Sistema Operativo de Macintosh, es el nombre del sistema operativo creado por Apple.
- **MSMQ**  
MSMQ es una herramienta de Microsoft para realizar el manejo de colas de mensajes.
- **.NET**  
Para el desarrollo y ejecución de aplicaciones en este nuevo entorno tecnológico Microsoft proporciona el conjunto de herramientas conocido .NET Framework SDK.
- **Perl**  
Es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C..
- **PHP**  
Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
- **Remoting**  
.Net Remoting es una tecnología propietaria de Microsoft que permite crear aplicaciones distribuidas. Una de las principales características es la capacidad para poder trabajar desde una máquina con los objetos en memoria de la máquina Remota.
- **RUP**  
Rational Unified Process que es una metodología de desarrollo de software.

- **SAD**  
Sistemas de Adquisición de Datos
  
- **SCADA**  
SCADA, acrónimo de Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos) es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia. Facilita retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores) y controlando el proceso automáticamente. Provee de toda la información que se genera en el proceso productivo (supervisión, control calidad, control de producción, almacenamiento de datos, etc.) y permite su gestión e intervención.
  
- **SilverLight**  
Microsoft Silverlight es una estructura para aplicaciones web que agrega nuevas funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace Adobe Flash.
  
- **SOAP**  
SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.
  
- **Unix**  
Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969, por un grupo de empleados de los laboratorios Bell de AT&T

- **WCF**

Windows Communication Foundation o WCF (también conocido como Indigo), es la plataforma de mensajería que forma parte de la API de la Plataforma .NET 3.0 (antes conocida como WinFX, y que no son más que extensiones para la versión 2.0). Se encuentra basada en la Plataforma .NET 2.0.

Fue creada con el fin de permitir una programación rápida de sistemas distribuidos y el desarrollo de aplicaciones basadas en arquitecturas orientadas a servicios (también conocido como SOA), con una API simple; y que puede ejecutarse en una máquina local, una LAN, o sobre Internet en una forma segura.

- **WPF**

Windows Presentation Foundation (WPF) es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web.

- **WSDL**

WSDL (en ocasiones leído como wisdel) son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web.

- **WSE**

Web Services Enhancements (WSE) es un add-on para el .NET Framework Microsoft que incluye un conjunto de clases que implementan especificaciones de los servicios web, principalmente en áreas como la seguridad, la mensajería fiable y el envío de archivos adjuntos. Web servicios son los componentes de lógica empresarial que proporcionan funcionalidad a través de Internet utilizando protocolos estándar como HTTP. Los servicios Web se comunican a través de cualquiera de los mensajes SOAP o REST. WSE provee extensiones para el protocolo SOAP y permite la definición de

seguridad personalizado, mensajería confiable, la política, etc Los desarrolladores pueden añadir estas capacidades en tiempo de diseño mediante código o en tiempo de despliegue mediante el uso de un archivo de política.

- **XML**

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.