

**UNIVERSIDAD POLITÉCNICA  
SALESIANA**

**FACULTAD DE INGENIERÍAS**

**SEDE QUITO – CAMPUS SUR**

**CARRERA DE INGENIERÍA EN SISTEMAS**

**MENCIÓN ROBÓTICA E INTELIGENCIA ARTIFICIAL**

**Análisis, Diseño e Implementación de un Sistema de Evaluación por computadora para el Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur.**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS**

**Eleana Elizabeth Toapanta Salazar**

**DIRECTOR: Ing. Geovanny Calle**

**Quito, Diciembre de 2010.**

# DECLARACIÓN

Yo Eleana Elizabeth Toapanta Salazar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Eleana Elizabeth Toapanta Salazar

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Eleana Elizabeth Toapanta Salazar, bajo mi dirección.

---

Ing. Geovanny Calle.  
Director de Tesis.

## **AGRADECIMIENTO**

A todos a quienes con su ayuda incondicional, aportaron para la realización de este trabajo, especialmente al Departamento de Idiomas, a mis profesores, gracias por los innumerables consejos, a todos mis amigos y conocidos por su incansable aliento.

## **DEDICATORIA**

A mis padres Manuel e Hilda, mis hermanos Roberto, Silvia, Adriana, Mariela, a mi sobrino Sebas, a mis amigos los que durante todos estos años hemos compartido tanto, a los de antes, a los de siempre.

# ÍNDICE DE CONTENIDOS

## CAPÍTULO 1

### MARCO TEORICO

1.1 Antecedentes	1
1.2 Fundamentación Científica	2
1.2.1 La Evaluación	2
1.2.1.1. Evaluación	2
1.2.1.2. Características de la Evaluación	3
1.2.1.3 Tipos de Evaluación	4
1.2.1.3.1 Evaluación Diagnóstica	4
1.2.1.3.2 Evaluación Formativa	4
1.2.1.3.3 Evaluación Sumativa	4
1.2.1.4 Consideraciones de la Evaluación	5
1.2.1.5 Las Pruebas o Test	6
1.2.1.5.1 Características de las Pruebas o Test	6
1.2.1.5.2 Los elementos que deben considerar una prueba	6
1.2.1.5.3 Clasificación de las Pruebas	7
1.2.1.5.3.1 Pruebas Orales	7
1.2.1.5.3.2 Pruebas Escritas	8
1.2.1.6 Diseño y construcción de los test	11
1.2.1.6.1 Característica externa de la Prueba Objetiva	12
1.2.1.6.2 Característica interna de la prueba objetiva	12
1.2.1.6.3 Limitaciones de la prueba objetiva	13
1.2.1.6.4 Prueba Objetiva de Opción Múltiple	13
1.2.1.6.4.1 Aspectos de la Prueba de Opción Múltiple	13
1.2.1.6.4.2 Los Ítems	14
1.2.1.6.4.3 Instrumentos para Corrección de Pruebas Objetivas	15
1.2.1.6.4.4 Pautas para escribir Pruebas de Opción Múltiple	17
1.2.1.7 Exámenes Computarizados	20
1.2.1.7.1 Ventajas de los exámenes computarizados	21
1.2.1.7.2 Desventajas de los exámenes computarizados	21
1.2.1.7.3 Sistemas de Evaluación existentes en el mercado	21
1.2.2 Lenguajes de Desarrollo	23
1.2.2.1 Visual Studio.Net 2005	23
1.2.2.1.1 Generalidades del Entorno de Desarrollo Integrado (IDE)	25
1.2.2.1.2 Asp.Net	26
1.2.2.1.3 C#	28
1.2.2.1.4 XML	30
1.2.2.2 Base de Datos	31
1.2.2.2.1 Modelo de Base de Datos	32

1.2.2.2.1.1 Modelo Entidad Relación	33
1.2.2.2.1.2 Proceso de Normalización	36
1.2.2.2.1.3 Arquitectura de un Sistema de Base de Datos	41
1.2.2.2.1.4 Diseño de la Base de Datos	42
1.2.2.2.1.5 Sistema de Gestión de Base de Datos	42
1.2.2.2.1.5.1 Microsoft SQL Server 2005	43
1.2.2.2.1.6 Seguridad de Base de Datos	44
1.2.2.2.1.7 Protocolos de Seguridad	46
1.2.3 Entorno de Desarrollo Web	48
1.2.3.1 Windows Server 2003	48
1.2.3.1.1 Servidores de Windows Server 2003	49
1.2.3.1.2 Arquitectura de Servicios	53
1.2.3.1.3 Ventajas	54
1.2.3.2 Modelo Aplicación Cliente/Servidor	55
1.2.3.2.1 Cliente	55
1.2.3.2.2 Servidor	56
1.2.3.2.3 Infraestructura de Comunicaciones	57
1.2.3.2.4 Características de la Arquitectura Cliente/Servidor	58
1.2.3.2.5 Ventajas y Desventajas de este esquema	58
1.2.3.3 Arquitectura de Capas en Sistemas de Información	60
1.2.3.3.1 Arquitectura en 2 Capas	60
1.2.3.3.2 Arquitectura en 3 Capas	61
1.2.3.3.3 Arquitectura en n Capas	64
1.2.3.4 Servidores Web	65
1.2.3.4.1 Características Básicas de los Servidores Web	66
1.2.3.4.2 Servidor Internet Información Server	67
1.2.3.4.3 Instalación y Configuración de un Sitio Web	69
1.2.3.4.3.1 Configuración de Sitios Web	69
1.2.3.5 Servidor de Base de Datos	70
1.2.3.5.1 Servidor de Base de Datos Relacionales	70
1.2.3.5.1.1 Características	70
1.2.3.5.1.2 Seguridad	71
1.2.3.5.1.3 Soporte de Red	72
1.2.3.5.1.4 Herramientas de Administración	73
1.2.4 Modelo Prototipado	74
1.2.4.1 Definición de Modelo Prototipado	75
1.2.4.2 Estructura del Modelo Prototipado	76
1.2.4.3 Etapas del Método de Prototipos	76
1.2.4.3.1 Identificación de Requerimientos	76
1.2.4.3.2 Desarrollo de un modelo de trabajo	76
1.2.4.3.3 El prototipo y el usuario	77
1.2.4.3.4 Revisión del Prototipo	77
1.2.4.3.5 Repetición del Proceso	77
1.2.4.4 Construcción de Prototipos	78
1.2.4.5 Herramientas para el Prototipado	79
1.2.4.6 Ventajas del Modelo Prototipado	79
1.2.4.7 Desventajas del Modelo Prototipado	80
1.2.5 Metodología UML	82
1.2.5.1 Definición de Metodología UML	82

1.2.5.2 Características	82
1.2.5.3 Diagramas UML	83
1.2.5.3.1 Diagramas de Clase	83
1.2.5.3.2 Diagrama de Componentes	83
1.2.5.3.2.1 Usos más comunes	84
1.2.5.3.3 Diagrama de Despliegue Físico	85
1.2.5.3.4 Diagrama de Actividades	86
1.2.5.3.4.1 Contenido del Diagrama de Actividades	86
1.2.5.3.4.2 División y Unión	88
1.2.5.3.5 Diagrama de Secuencias	88
1.2.5.3.6 Diagrama de Estados	89
1.2.5.3.7 Diagrama de Casos de Uso	91
1.2.5.3.7.1 Elementos	91
1.2.5.3.8 Otros Diagramas	93
1.2.5.4 Elementos comunes a todos los diagramas	93
1.2.5.4.1 Notas	93
1.2.5.4.2 Clases	94
1.2.5.4.3 Atributos	94
1.2.5.4.3.1 Elementos Derivados	94
1.2.5.4.4 Asociaciones	95
1.2.5.4.4.1 Nombre de la Asociación y Dirección	95
1.2.5.4.4.2 Multiplicidad	95
1.2.5.4.4.3 Roles	96
1.2.5.4.4.4 Clases Asociación	96
1.2.5.4.4.5 Asociaciones n-arias	97
1.2.5.4.4.6 Ensamblados	97
1.2.5.4.4.7 Agregación	97
1.2.5.4.4.8 Composición	98
1.2.5.4.4.9 Dependencia o Instanciación	98
1.2.5.4.5 Herencia	98
1.2.5.5 Modelado con UML	99
1.2.5.5.1 Conceptos Básicos	100
1.2.5.6 Ventajas de UML	101
1.2.5.7 Desventajas del UML	102

## **CAPITULO 2**

### **ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA MODELO Y METODOLOGÍA DEL SISTEMA**

2.1 Identificación de requerimientos	103
2.2 Diseño Rápido	106
2.2.1 Diagramas del Sistema	107

2.2.1.1 Diagramas de Caso de Uso	107
2.2.1.1.1 Actores	107
2.2.1.1.1.1 Actor Administrador	109
2.2.1.1.1.2 Actor Editor	110
2.2.1.1.1.3 Actor Estudiante	110
2.2.1.1.1.4 Actor Profesor	110
2.2.1.1.2 Lista de Casos de Uso	111
2.2.1.1.2.1 Caso de Uso Agregar Sonido al Listening	112
2.2.1.1.2.2 Caso de Uso Califica Estudiantes	112
2.2.1.1.2.3 Caso de Uso Cambiar Contraseña	113
2.2.1.1.2.4 Caso de uso Crear Administrador	113
2.2.1.1.2.5 Caso de Uso Crear Estudiante	113
2.2.1.1.2.6 Caso de Uso Crear Profesor/Editor	114
2.2.1.1.2.7 Caso de Uso Enviar Notificación	114
2.2.1.1.2.8 Caso de Uso Generar Test	115
2.2.1.1.2.9 Casos de Uso Gestiona Capítulos por Nivel	115
2.2.1.1.2.10 Caso de Uso Gestiona Clases	116
2.2.1.1.2.11 Caso de Uso Gestiona Facultades	116
2.2.1.1.2.12 Caso de Uso Gestiona Grupo de Preguntas	117
2.2.1.1.2.13 Caso de Uso Gestiona Niveles	117
2.2.1.1.2.14 Caso de Uso Gestiona Preguntas	118
2.2.1.1.2.15 Caso de Uso Gestiona Tópicos	118
2.2.1.1.2.16 Caso de Uso Parámetros de las preguntas	119
2.2.1.1.2.17 Caso de Uso recuperar contraseña	119
2.2.1.1.2.18 Caso de Uso Revisar Notas	120
2.2.1.1.2.19 Caso de Uso Tomar el Examen	120
2.2.1.2 Diagrama Entidad Relación	122
2.2.1.2.1 Lista de Entidades	122
2.2.1.2.1.1 Entidad AdministratorProfiles	122
2.2.1.2.1.2 Entidad Classroom	123
2.2.1.2.1.3 Entidad Distance_Mode_Students	124
2.2.1.2.1.4 Entidad Faculties	124
2.2.1.2.1.5 Entidad Intensive_Mode_Students	124
2.2.1.2.1.6 Entidad Levels	125
2.2.1.2.1.7 Entidad Listening	125
2.2.1.2.1.8 Entidad ParametersQuestions	126
2.2.1.2.1.9 Entidad Presential_Mode_Students	126
2.2.1.2.1.10 Entidad QuestionGroups	126
2.2.1.2.1.11 Entidad Questions	127
2.2.1.2.1.12 Entidad StudentsProfiles	127
2.2.1.2.1.13 Entidad Test	128
2.2.1.2.1.14 Entidad Tópicos	128
2.2.1.2.1.15 Entidad Units	129
2.2.1.2.1.16 Entidades UserProfiles	129
2.2.1.2.2 Lista de Relaciones	130
2.2.1.3 Diagrama de Clases	131
2.2.1.3.1 Lista de Clases	131
2.2.1.3.1.1 Clase UserProfiles	131
2.2.1.3.1.2 Clase Classrooms	132

2.2.1.3.1.3 Clase StudentProfiles	133
2.2.1.3.1.4 Clase Level	133
2.2.1.3.1.5 Clase Faculties	134
2.2.1.3.1.6 Clase Units	134
2.2.1.3.1.7 Clase Topics	135
2.2.1.3.1.8 Clase Test	135
2.2.1.3.1.9 Clase Questions	136
2.2.1.3.1.10 Clase QuestionsGroups	136
2.2.1.3.1.11 Clase Presencial_Mode_Students	137
2.2.1.3.1.12 Clase Intensive_Mode_Students	138
2.2.1.3.1.13 Clase Distance_Mode_Students	139
2.2.1.3.1.14 Clase ParameterQuestions	139
2.2.1.3.1.15 Clase Listening	140
2.2.1.3.1.16 Clase AdministratorProfiles	140
2.2.1.3.1.17 Clase LevelClassroom	141
2.2.1.3.1.18 Clase QuestionsInGroupsQuestions	141
2.2.1.3.1.19 Clase Aspnet_Roles	142
2.2.1.3.1.20 Clase Aspnet_Users	142
2.2.1.3.1.21 Clase Aspnet_UserInRoles	143
2.3 Construcción del prototipo	144
2.3.1 Inicio de Sesión	144
2.3.2 Ambiente del Usuario Administrador	141
2.3.2.1 Manager User	145
2.3.2.1.1 Administrator	146
2.3.2.1.2 Teacher and Editor	147
2.3.2.1.3 Student	148
2.3.2.2 Manage Application	148
2.3.2.2.1 Faculty	149
2.3.2.2.2 Level	149
2.3.2.2.3 Classroom	150
2.3.2.2.4 Topics	150
2.3.2.2.5 Units	150
2.3.2.2.6 Parameters Questions	151
2.3.3 Ambiente del Usuario Teacher	152
2.3.3.1 Academic Options	152
2.3.3.1.1 Generate Test	152
2.3.3.1.2 Information	153
2.3.3.1.2.1 Grades	153
2.3.4 Ambiente del Usuario Student	153
2.3.4.1 Test	154
2.3.4.1.1 Take Test	154
2.3.4.2 Grades	154
2.3.5 Ambiente del Usuario Editor	155
2.3.5.1 Questions Options	155
2.3.5.1.1 Questions Group	155
2.3.5.1.2 Management Questions	156
2.3.5.1.3 Listening Audio	157

## **CAPITULO 3**

### **PRUEBAS E IMPLEMENTACIÓN DEL SISTEMA**

3.1 Pruebas del Sistema	158
3.1.1 Pruebas de Funcionalidad	158
3.1.1.1 Módulo para el Administrador en el Sistema	159
3.1.1.2 Módulo para el Profesor en el Sistema	164
3.1.1.3 Módulo para el Estudiante en el Sistema	165
3.1.1.4 Módulo para el Editor en el Sistema	165
3.1.2 Pruebas de Concurrencia	167
3.1.3 Pruebas de Seguridad	167
3.2 Detalles de Implementación del Sistema	169

### **CONCLUSIONES Y RECOMENDACIONES**

Conclusiones	170
Recomendaciones	171

### **ANEXOS**

A. Fundamentación Legal	174
B. Diseño de la Investigación	179
C. Diagramas del Sistema	188
D. Plantillas para crear Usuarios	191
E. Manual de Usuarios	194
F. Código del Sistema	214

### **BIBLIOGRAFÍA**

Bibliografía y Netgrafía	236
--------------------------	-----

## INDICE DE FIGURAS

	Pag.
Figura 1.1 Ejemplo de examen con 3 distractores	16
Figura 1.2 Ejemplo de Plantilla de corrección de la Hoja de respuestas	16
Figura 1.3 Plataforma de Ejecución	24
Figura 1.4 DataSet Faculties-capa de acceso a datos	25
Figura 1.5 Ejemplo de una clase en c#	30
Figura 1.6 Tablas de la aplicación con relación uno a uno	34
Figura 1.7. Tablas de la aplicación con relación uno a varios	35
Figura 1.8 Arquitectura de un sistema de base de datos	42
Figura 1.9 Versiones de Windows Server 2003	48
Figura 1.10 Servidor de Impresiones	50
Figura 1.11 Servidor DNS	51
Figura 1.12 Servidor Streaming de video	52
Figura 1.13 Clientes y Servidores RCP	54
Figura 1.14 Modelo Cliente/Servidor	55
Figura 1.15 Arquitectura en 2 niveles	60
Figura 1.16 Arquitectura en 3 capas	61
Figura 1.17 Arquitectura de 3 capas con Interfaz Web	63
Figura 1.18 Arquitectura de 4 capas	65
Figura 1.19. Modelo Prototipado	75
Figura 1.20. Estructura del Modelo Prototipado	76
Figura 1.21 Proceso del Modelo Prototipado	78
Figura 1.22 Diagrama de Clases reloj	83
Figura 1.23 Diagrama de Clases	86
Figura 1.24 Estado de acción	87
Figura 1.25 Estado de actividad	87
Figura 1.26 Transiciones sin disparador	87
Figura 1.27 Bifurcación	88
Figura 1.28. División y Unión	88
Figura 1.29 Diagrama de secuencia	89
Figura 1.30 Diagrama de estados	90
Figura 1.31 Actor	91
Figura 1.32 Diagrama de caso de uso	92
Figura 1.33 Nota	93
Figura 1.34 Elementos Derivados	95
Figura 1.35 Asociaciones	95
Figura 1.36 Roles	96
Figura 1.37 Clases de Asociación	96
Figura 1.38 Asociaciones n-arias	97
Figura 1.39 Agregación	98
Figura 1.40 Dependencia	98
Figura 1.41 Herencia	99
Figura 1.42 Modelo con 2 notaciones diferentes	100

Figura 1.43 Componentes del concepto reloj	101
Figura 2.1 Diagrama de las carreras que sirve Idiomas	106
Figura 2.2 Diagrama del Actor Administrador	107
Figura 2.3 Diagrama del Actor Profesor	108
Figura 2.4 Diagrama del Actor Editor	108
Figura 2.5 Diagrama del Actor Estudiante	109
Figura 2.6 Inicio de Sesión	144
Figura 2.7 Ambiente Común del sistema	144
Figura 2.8 Menú Usuario Administrador	145
Figura 2.9 Menú Manager User	145
Figura 2.10 Mensajes en caso de ingreso erróneo de datos	146
Figura 2.11 Crear Usuario Administrador	146
Figura 2.12 Aviso de usuario creado	147
Figura 2.13 Crear usuario teacher/editor	147
Figura 2.14 Crear usuario estudiante	148
Figura 2.15 Menú Manage Application	148
Figura 2.16 Agregar Facultad	149
Figura 2.17 Agregar Nivel	149
Figura 2.18 Agregar Classroom	150
Figura 2.19 Agregar Tópicos	150
Figura 2.20 Agregar Unidades a los niveles	151
Figura 2.21 Agregar parámetros del test	151
Figura 2.22 Menú Usuario Teacher/Editor	152
Figura 2.23 Generar exámenes a los estudiantes	152
Figura 2.24 Calificaciones de los estudiantes por curso	153
Figura 2.25 Menú Usuario Estudiante	153
Figura 2.26 Examen de un Estudiante	154
Figura 2.27 Calificaciones de un estudiante	155
Figura 2.28 Menú Editor	155
Figura 2.29 Configuración de Questions Group	156
Figura 2.30 Manejo de preguntas	156
Figura 2.31 Agregar audio a una pregunta	157
Figura C.1 Diagrama Entidad Relación	188
Figura C.2 Diagrama Físico	189
Figura C.3 Diagrama de Clases	190
Figura E.1 Inicio de Sesión	194
Figura E.2 Crear Usuarios	195
Figura E.3 Editar Usuarios	196
Figura E.4 Borrar Usuarios	197
Figura E.5 Manejo de Facultades	199
Figura E.6 Manejo de Niveles	200
Figura E.7 Manejo de Tópicos	201
Figura E.8 Manejo de Unidades	202
Figura E.9 Agregar Classrooms	203
Figura E.10 Manejo de Classrooms	204
Figura E.11 Parámetros de las preguntas	204
Figura E.12 Inicio de Sesión	206
Figura E.13 Take Test	207
Figura E.14 Grades	207

Figura E.15 Inicio de Sesión	208
Figura E.16 Generar Examen	209
Figura E.17 Agregar Calificaciones	209
Figura E.18 Inicio de Sesión	210
Figura E.19 Ingreso de Encabezado de preguntas.	211
Figura E.20 Ingreso de preguntas	212
Figura E.21 Relacionar audio con una pregunta de Listening	213

## INDICE DE TABLAS

	Pag.
Tabla 1.1 Ejemplo para poner en primera forma normal	36
Tabla 1.2 Ejemplo en primera forma normal	37
Tabla 1.3 Ejemplo para poner en segunda forma normal	37
Tabla 1.4 Ejemplo en segunda forma normal	38
Tabla 1.5 Ejemplo para poner en tercera forma normal	39
Tabla 1.6 Ejemplo en tercera forma normal	39
Tabla 1.7 Ejemplo para poner en cuarta forma normal	40
Tabla 1.8 Ejemplo en cuarta forma normal	40
Tabla 1.9 Clase	94
Tabla 2.1 Asociaciones del Actor Administrador	109
Tabla 2.2 Asociaciones del Actor Editor	110
Tabla 2.3 Asociaciones del Actor Estudiante	110
Tabla 2.4 Asociaciones del Actor Profesor	111
Tabla 2.5 Lista de dependencias	111
Tabla 2.6 Lista de Casos de Uso	111
Tabla 2.7 Diccionario del Caso de Uso Agregar sonido al Listening	112
Tabla 2.8 Diccionario del Caso de Uso Califica Estudiantes.	112
Tabla 2.9 Diccionario del Caso de Uso Cambiar Contraseña	113
Tabla 2.10 Diccionario del Caso de Uso Crear Administrador	113
Tabla 2.11 Diccionario del Caso de Uso Crear Estudiante	114
Tabla 2.12 Diccionario del Caso de Uso Crear Profesor/Editor	114
Tabla 2.13 Diccionario del Caso de Uso Enviar Notificación	115
Tabla 2.14 Diccionario del Caso de Uso Generar Test	115
Tabla 2.15 Diccionario del Caso de Uso Gestiona Capítulos por Nivel	116
Tabla 2.16 Diccionario del Caso de Uso Gestiona Clases	116
Tabla 2.17 Diccionario del Caso de Uso Gestiona Facultades.	117
Tabla 2.18 Diccionario del Caso de Uso Gestiona Grupo de Preguntas	117
Tabla 2.19 Diccionario del Caso de Uso Gestiona Niveles	118
Tabla 2.20 Diccionario del Caso de Uso Gestiona Preguntas	118
Tabla 2.21 Diccionario del Caso de Uso Gestiona Tópicos.	119
Tabla 2.22 Diccionario del Caso de Uso Parámetros de las Preguntas	119
Tabla 2.23 Diccionario del Caso de Uso Recuperar Contraseña	120
Tabla 2.24 Diccionario del Caso de Uso Revisar Notas	120
Tabla 2.25 Diccionario del Caso de Uso Tomar el Examen	121
Tabla 2.26 Lista de Entidades	122
Tabla 2.27 Atributos de la Entidad AdministratorProfiles.	123
Tabla 2.28 Atributos de la Entidad Classroom.	123
Tabla 2.29 Atributos de la Entidad Distance_Mode_Students.	124
Tabla 2.30 Atributos de la Entidad Faculties.	124
Tabla 2.31 Atributos de la Entidad Intensive_Mode_Student	125
Tabla 2.32 Atributos de la Entidad Levels.	125
Tabla 2.33 Atributos de la Entidad Listening	125

Tabla 2.34 Atributos de la Entidad ParameterQuestions.	126
Tabla 2.35 Atributos de la Entidad Presencial_Mode_Students.	126
Tabla 2.36 Atributos de la Entidad QuestionGroups.	127
Tabla 2.37 Atributos de la Entidad StudentProfiles.	128
Tabla 2.38 Atributos de la Entidad StudentProfiles	128
Tabla 2.39 Atributos de la Entidad Test.	128
Tabla 2.40 Atributos de la Entidad Topic.	129
Tabla 2.41 Atributos de la Entidad Units.	129
Tabla 2.42 Atributos de la Entidad UserProfiles.	130
Tabla 2.43 Lista de Relaciones	130
Tabla 2.44 Lista de Clases.	131
Tabla 2.45 Lista de Atributos de la clase UserProfiles.	132
Tabla 2.46 Lista de Operaciones de la clase UserProfiles	132
Tabla 2.47 Lista de Atributos de la clase Classrooms	132
Tabla 2.48 Lista de Operaciones de la clase Classroom.	132
Tabla 2.49 Lista de Atributos de la clase StudentProfiles	133
Tabla 2.50 Lista de Operaciones de la clase StudentProfiles	133
Tabla 2.51 Lista de Atributos de la clase Levels	134
Tabla 2.52 Lista de Operaciones de la clase Levels	134
Tabla 2.53 Lista de Atributos de la clase Faculties	134
Tabla 2.54 Lista de Operaciones de la clase Faculties	134
Tabla 2.55 Lista de Atributos de la clase Units.	134
Tabla 2.56 Lista de Operaciones de la clase Units	135
Tabla 2.57 Lista de Atributos de la clase Topics	135
Tabla 2.58 Lista de Operaciones de la clase Topics.	135
Tabla 2.59 Lista de Atributos de la clase Test	136
Tabla 2.60 Lista de Operaciones de la clase Test	136
Tabla 2.61 Lista de Atributos de la clase Questions	136
Tabla 2.62 Lista de Operaciones de la clase Questions	136
Tabla 2.63 Lista de Atributos de la clase QuestionGroups	137
Tabla 2.64 Lista de Operaciones de la clase QuestionGroups.	137
Tabla 2.65 Lista de Atributos de la clase Presential_Mode_Students	137
Tabla 2.66 Lista de Operaciones de la clase Presential_Mode_Students.	138
Tabla 2.67 Lista de Atributos de la clase Intensive_Mode_Students	138
Tabla 2.68 Lista de Operaciones de la clase Intensive_Mode_Students.	138
Tabla 2.69 Lista de Atributos de la clase Distance_Mode_Students.	139
Tabla 2.70 Lista de Operaciones de la clase Distance_Mode_Students.	139
Tabla 2.71 Lista de Atributos de la clase ParametersQuestions	139
Tabla 2.72 Lista de Operaciones de la clase ParametersQuestions.	140
Tabla 2.73 Lista de Atributos de la clase Listening	140
Tabla 2.74 Lista de Atributos de la clase AdministratorProfiles.	140
Tabla 2.75 Lista de Operaciones de la clase AdministratorProfiles	141
Tabla 2.76 Lista de Atributos de la clase LevelClassroom	141
Tabla 2.77 Lista de Operaciones de la clase LevelClassroom.	141
Tabla 2.78 Lista de Atributos de la clase Questioningroupquestions	142
Tabla 2.79 Lista de Operaciones de la clase Questioningroupquestions	142
Tabla 2.80 Lista de Atributos de la clase Aspenet_Roles	142
Tabla 2.81 Lista de Operaciones de la clase Aspnet_Roles	142
Tabla 2.82 Lista de Atributos de la clase Aspnet_Users	143

Tabla 2.83 Lista de Atributos de la clase AspNet_UserInRoles	143
Tabla 2.84 Lista de Operaciones de la clase AspNet_UserInRole	143
Tabla 2.85 Login.aspx	159
Tabla 2.86 AdministratortorAdd.aspx	159
Tabla 2.87 UserAdd.aspx	159
Tabla 2.88 StudentAdd..aspx.	160
Tabla 2.89 AdministratorControl.aspx	160
Tabla 2.90 TeacherControl.aspx	160
Tabla 2.91 StudentsControl.aspx	161
Tabla 2.92 Faculties.aspx	161
Tabla 2.93 ManagementLevels.aspx	162
Tabla 2.94 ManagemetTopics.aspx	162
Tabla 2.95 ManagemetUnits.aspx	162
Tabla 2.96 ManagemetClassroom.aspx	163
Tabla 2.97 UpdatetClassrooms.aspx	163
Tabla 2.98 ManagementParametersQuestions.aspx	163
Tabla 2.99 GenerateTest.aspx	164
Tabla 2.100 Grades.aspx	164
Tabla 2.101 TakeTest.aspx	165
Tabla 2.102 Grades.aspx	165
Tabla 2.103 QuestionGroup.aspx	166
Tabla 2.104 ManagementQuestionGroup.aspx	166
Tabla 2.105 ManagementListeningQuestions.aspx	167

## **RESUMEN**

El sistema de exámenes por computador desarrollado para el Departamento de Idiomas de la Universidad Politécnica Salesiana Campus Sur permite la automatización del proceso de rendición de exámenes de inglés basados en el mismo formato que utiliza este Departamento el cual consta de un banco de preguntas de acuerdo al nivel y subtemas y en el que de forma aleatoria se asignan las preguntas, una vez finalizado el examen el estudiante obtiene el resultado correspondiente y el profesor puede llevar el control de esta y las restantes notas necesarias para la aprobación o no de un nivel para el estudiante.

El primer capítulo presenta los antecedentes, fundamentación teórica de la evaluación, lenguajes de desarrollo, método y metodología utilizados en el proyecto de tesis.

El segundo capítulo presenta el análisis, diseño y desarrollo del sistema de evaluación con sus respectivos diagramas e interfaces.

El tercer capítulo presenta las pruebas e implementación realizadas al sistema.

El cuarto capítulo muestra las conclusiones y recomendaciones para este proyecto y los anexos.

## **PRESENTACIÓN**

**"La sabiduría popular dice que el éxito en la vida depende de dos ingredientes: inteligencia y afán de triunfar, pero que el primero fracasa sin el segundo: lo importante es la voluntad."**

El uso de la tecnología para prácticamente todas las actividades que realiza el ser humano ha conllevado a que cada día se busque aprovechar todos los beneficios que esta nos ofrece. La Educación es sin duda uno de los campos en donde resulta de suma importancia el uso y la aplicación de la tecnología en busca de favorecer el aprendizaje a los estudiantes y por supuesto facilitar esta ardua tarea a los profesores y con ello poco a poco mejorar todo el sistema educativo y por ende a la sociedad.

El desarrollo de software con fines educativos presenta un sin número de bondades para todos los actores de esta actividad ya que resulta atractivo el aprendizaje por medio de imágenes y sonidos, principalmente para los educandos, además de la facilidad en la operación, almacenamiento, portabilidad y transmisión de grandes cantidades de información en espacios físicos pequeños, la potencialidad de establecer conexiones por medio de redes de manera especial el Internet en diversos puntos geográficos distantes, entre otras, por lo tanto cambia la forma de transmitir el aprendizaje lo que ha permitido que se formen nuevos paradigmas del aprendizaje en base a la tecnología y que el aprendizaje resulte más significativo para los educandos

Este proyecto se concibió con el fin de alcanzar los objetivos para el desarrollo de un software que sea de ayuda en el proceso educativo, en este caso para la asignatura de inglés; así como también para que sirva de guía en el desarrollo de proyectos similares.

## INTRODUCCIÓN

La Educación es un proceso que a través del tiempo ha ido evolucionando de una manera drástica, la integración de la tecnología a este proceso ha llevado a que constantemente los sistemas educativos se vuelvan obsoletos sino van de la mano de la utilización de las nuevas tecnologías desarrolladas. La ingeniería de sistemas ha aportado en gran medida con estos avances ya sea con equipos o sistemas que permiten este continuo avance no solo en la educación sino en prácticamente cualquier campo. Es por esta razón que este proyecto de tesis tiene como propósito establecer el Análisis, Desarrollo e Implementación de un Sistema de Evaluación por Computador para el Departamento de Idiomas de la Universidad Politécnica Salesiana - Campus Sur, ya que, el mismo se encarga de proveer las facilidades para que los alumnos del Campus aprueben la suficiencia de inglés, necesaria para la aprobación de la carrera en todas las facultades.

Por este motivo, el desarrollo de este proyecto consiste en brindar un sistema en el que los estudiantes rindan sus exámenes de acuerdo al nivel y modalidad al que corresponda, además presentados a través de secciones como grammar, reading, listening, vocabulary, basado en preguntas de opción múltiple en un tiempo determinado. Al final del mismo se puede obtener automáticamente la calificación alcanzada por el estudiante. El profesor puede ingresar las notas restantes que son necesarias para que un alumno apruebe el nivel. Tanto el alumno como el profesor tienen la posibilidad de revisar las mismas. El sistema permite crear y mantener a todos los usuarios del sistema como administrador, editor, profesor y estudiante.

El sistema propuesto se lo realizó con una arquitectura de tres capas en Visual Studio.Net 2005 utilizando ASP.Net y C#, la base de datos utilizada es SQL Server 2005 y un Servidor Internet Information Server. Los exámenes serían realizados en los laboratorios del Campus ya que cuenta con la infraestructura necesaria para llevar a cabo esta actividad.

# CAPÍTULO 1

## MARCO TEÓRICO.

### 1.1 ANTECEDENTES

La Evaluación juega un rol muy importante en la Educación, por esta razón se han desarrollado varias investigaciones en las que se busca mejorar este proceso especialmente a través de la tecnología en donde la Ingeniería de Sistemas nos ha permitido crear sistemas que nos ayudan enormemente en el desarrollo del área educativa. Los sistemas de evaluación por computadora buscan desarrollar mejores métodos de enseñanza. Uno de los trabajos que surgieron fue el planteado por Rumbolt (1989) con el sistema Quizmaster en el que compara dos formas de administrar un examen, la primera con el sistema y la segunda con papel y lápiz, en su estudio concluye que ambas formas son equivalentes. Gibs (1991) en el proyecto TestMarker señala que con la llegada de la computadora en el proceso educativo es posible incrementar la eficiencia, tanto en la administración, la evaluación, la creación de un banco de reactivos, la personalización de exámenes, el registro y análisis. El autor sugiere que esto facilita las condiciones de exámenes estandarizados, administración individual, evaluación inmediata y diversidad de formatos, entre otros. Otras aplicaciones que han surgido son los exámenes para ingresar a los posgrados en EEUU, GRE (Graduate record Examination), además de los exámenes para aprobar la suficiencia del Idioma Ingles, entre otros. De estos y otros estudios se puede destacar que si bien ambos procesos son equivalentes, los estudiantes prefieren los exámenes computarizados, porque los podían presentar cuando realmente se encuentran preparados y por el hecho de conocer el resultado de forma inmediata; además las computadoras tienen la ventaja de ser completamente objetivas, lo cual no es posible con los humanos.

## **1.2 FUNDAMENTACIÓN CIENTÍFICA**

### **1.2.1 LA EVALUACIÓN**

#### **1.2.1.1 EVALUACIÓN**

Evaluación puede conceptualizarse como un proceso dinámico, continuo y sistemático, enfocado hacia los cambios de las conductas y rendimientos, mediante el cual verificamos los logros adquiridos en función de los objetivos propuestos. La evaluación es sin duda uno de los aspectos más importantes de la educación que conlleva dentro de sí varios objetos a ser evaluados como son las instituciones educativas, los profesores, el aprendizaje, entre otros.

El motivo de investigación de este proyecto, radica en la evaluación de los aprendizajes de los alumnos, basada en la evaluación de la asignatura de inglés, que es uno de los pilares importantes, por medio del cual un alumno puede ser considerado como egresado de la Universidad Politécnica Salesiana. Esta evaluación involucra básicamente conocer qué y para qué evaluar, por lo que es esencial recoger información, formar un juicio de valor y tomar decisiones.

La evaluación involucra en sí, la medida en que los objetivos iniciales fueron alcanzados pero no debe ser vista como una finalidad de la enseñanza sino más bien como un medio por el cual obtener perfeccionamiento y mejora constante. En la evaluación de los aprendizajes de los alumnos, si el resultado obtenido en la evaluación es menor de lo esperado o no alcanza los objetivos planteados, se debe hacer una revisión de las actividades que se están realizando, es decir un reajuste que fortalezca el proceso enseñanza-aprendizaje. La Evaluación adquiere sentido en la medida que comprueba la eficacia y posibilita el perfeccionamiento de la acción docente. La evaluación debe permitir la adaptación de los programas educativos a las características individuales del alumno, detectar sus puntos débiles para poder corregirlos y tener un conocimiento cabal de cada uno.

## **FUNCIONES DE LA EVALUACIÓN ¿PARA QUÉ EVALUAR?.**

Son los propósitos o las finalidades de la Evaluación, que permite:

- Dirigir mecanismos de selección y control social
- Medir la calidad del sistema educativo
- Promoción, acreditación, certificación.

## **INSTRUMENTOS DE EVALUACIÓN. ¿CÓMO EVALUAR?.**

Los instrumentos de evaluación se planean en base a:

- **VALIDEZ:** Grado de precisión con que se mide lo que se desea medir.
- **CONFIABILIDAD:** Exactitud con que se mide un determinado rasgo, debe ser estable y objetiva.
- **PRACTICIDAD:** Se refiere a la viabilidad de la construcción, administración y análisis de resultados.
- **UTILIDAD:** Es la medida en que una evaluación resulta útil para la orientación tanto del alumno como del docente.

### **1.2.1.2 CARACTERÍSTICAS DE LA EVALUACIÓN**

La evaluación es un proceso que implica características tanto cualitativas como cuantitativas. Sus principales características son:

- Intencionalidad en la búsqueda de información
- Información retroalimentadora tanto a alumnos, docentes y otros actores involucrados, lo que permite tomar decisiones
- Pone de manifiesto aspectos o procesos que de otra manera permanecerían ocultos.

### **1.2.1.3 TIPOS DE EVALUACIÓN**

Tomando a la evaluación como un proceso se distinguen 3 tipos<sup>1</sup>:

- Evaluación Diagnostica,
- Evaluación Formativa,
- Evaluación Sumativa.

#### **1.2.1.3.1 EVALUACIÓN DIAGNOSTICA**

Este tipo de evaluación se lleva a cabo al inicio del curso. Permite identificar si necesitamos hacer reajustes a nuestra planeación, tanto si se ha observado que la mayoría de los alumnos no cumplen con los requisitos para empezar el curso como si se observa si la mayoría ya han alcanzado dichos requisitos, ya que en este caso los alumnos podrían perder el interés en el curso. Pretende determinar:

- Si los alumnos poseen los requisitos necesarios para empezar el nivel
- En que grado los alumnos han alcanzado los conocimientos, habilidades y destrezas
- La situación personal en que se encuentran los alumnos para iniciar el curso.

#### **1.2.1.3.2 EVALUACIÓN FORMATIVA**

Se la realiza al final de un tema o unidad de cuyo buen logro depende el éxito de actividades posteriores. Orienta la forma en la que se van alcanzando los objetivos, permitiendo la búsqueda de nuevas estrategias educativas más exitosas. Aporta una retroalimentación permanente al desarrollo del programa educativo.

#### **1.2.1.3.3 EVALUACIÓN SUMATIVA**

Es aquella que tiene la estructura de un balance, realizada después de un período de aprendizaje en la finalización de un programa o curso. Sus objetivos son

---

<sup>1</sup> LAS TEORÍAS DEL APRENDIZAJE. POVEDA, Elva. Quito. 1985.

calificar en función de un rendimiento, otorgar una certificación, determinar e informar sobre el nivel alcanzado a todos los niveles (alumnos, padres, institución, docentes, etc.). Considerada la evaluación como un proceso amplio, presente a lo largo de todo el proceso de enseñanza.

#### 1.2.1.4 CONSIDERACIONES DE LA EVALUACIÓN<sup>2</sup>

La evaluación debe permitir:

**CONOCER EL RENDIMIENTO DE ÉL O LOS ALUMNOS:** Esta parece ser la finalidad tradicional de la evaluación, calificar y establecer si el nivel alcanzado permite acceder al siguiente nivel.

**DIAGNOSTICAR:** Entendiendo esto como la determinación del nivel de una situación o de un individuo con relación a determinados parámetros.

**VALORAR LA EFICACIA DEL SISTEMA DE ENSEÑANZA:** La evaluación permite al docente establecer si los objetivos han sido adecuados, si los métodos son los más convenientes y si las condiciones han sido las adecuadas. Recordemos que las deficiencias no son siempre del alumno; que también pueden ser consecuencia de la situación educativa.

**MOTIVAR E INCENTIVAR AL ALUMNO:** Para el alumno, saber de sus logros, conocer sus fallas y sus éxitos constituye un importante estímulo. El planteo de objetivos realizables y su verificación representa una importante retroalimentación del proceso educativo.

**AGRUPAR O CLASIFICAR:** La formación de grupos homogéneos, es uno de los objetivos que el docente se plantea para el logro de una tarea que respete el principio de individualización. Los criterios de agrupación son flexibles ya que la homogeneidad constante no existe. Por lo cual la reagrupación es un proceso constante. La evaluación nos aporta los datos sobre los cuales se han de basar las agrupaciones y la organización de los grupos.

**CALIFICAR:** Dar a todos los niveles información objetiva sobre el proceso de formación (alumno, padres, etc.)

---

<sup>2</sup> PEDAGOGIA DE LA EVALUACIÓN. POVEDA, Elva. Quito. 1993.

**OBTENER DATOS PARA LA INVESTIGACIÓN:** El conocimiento y estandarización de normas facilita la tarea docente, por lo cual la recolección de datos y posterior procesamiento estadístico aporta información que nos permite mejorar la enseñanza, haciéndola más científica y eficiente.

#### **1.2.1.5 LAS PRUEBAS O TEST**

Las pruebas permiten provocar una situación para observar determinados resultados de aprendizaje.

##### **1.2.1.5.1 CARACTERÍSTICAS DE LAS PRUEBAS O TEST.**

Según Tenbrink<sup>3</sup>, una prueba se caracteriza por.

- Una situación común a la que todos los estudiantes responden.
- Unas instrucciones comunes que dirigen las respuestas de los alumnos/as.
- Un conjunto de reglas comunes para corregir las respuestas de los alumnos/as.
- Una descripción numérica de la realización de cada alumno/a, realizada después de corregir las respuestas.

##### **1.2.1.5.2 LOS ELEMENTOS QUE DEBEN CONSIDERAR UNA PRUEBA O TEST**

Los elementos que requieren una buena prueba o test son:

**CONFIABILIDAD:** Trata la consistencia de los resultados.

**VALIDEZ:** Tiene que evaluar lo que se desea evaluar y no otra cosa.

**OBJETIVIDAD:** Asignar puntaje al margen del juicio que se tenga sobre el examinado.

**ADECUADO AL TIEMPO ESTIPULADO:** Seleccionar la muestra y el proceso que pueda resolverse en el tiempo previsto.

---

<sup>3</sup> EVALUACIÓN - GUÍA PRÁCTICA PARA PROFESORES. TENBRINK, Terry. Madrid. 1981.

**PRACTICIDAD:** Debe ser de fácil elaboración y corrección para el profesor (banco de Ítems). Plantillas. Fácil entendimiento para los alumnos

**OPORTUNA:** Deberá comunicarse a los alumnos con anterioridad la aplicación de la prueba.

### **1.2.1.5.3 CLASIFICACIÓN DE LAS PRUEBAS**

Existen diversas clasificaciones para las pruebas, pero aquí nos concentraremos al tipo de prueba que nos conduce a nuestros propósitos en este proyecto de Tesis. Una de las clasificaciones de las pruebas puede ser:

- Orales y
- Escritas

#### **1.2.1.5.3.1 PRUEBAS ORALES**

Son aquellas en las cuales el docente formula una pregunta oral para que sea respondido por el alumno/a en la misma forma. Utilizada mucho en nuestro medio en forma de lecciones orales. El valor de las pruebas orales debe centrarse en verificar el dominio de destrezas relacionadas con la comprensión de conceptos, solución de problemas, la expresión oral.

#### **CONSIDERACIONES DE LAS PRUEBAS ORALES.**

- Analizar los criterios de evaluación para verificar la pertinencia de la utilización de este tipo de pruebas.
- Preparar de antemano un cuestionario de preguntas de más o menos el mismo nivel de dificultad para todos los alumnos/as.
- Si la prueba se dirige a todo el grupo de alumnos/as, las preguntas deben ser estructuradas considerando el mismo grado de dificultad y profundidad.
- Para valorar la actuación de los alumnos/as, se debe construir una tabla de especificaciones de calificación que recoja los diferentes aspectos a ser considerados.

### 1.2.1.5.3.2 PRUEBAS ESCRITAS

Son las más utilizadas, ya que se consideran el instrumento más adecuado para evaluar el rendimiento, pero no deben ser sobre valoradas sino usarse como otro instrumento de la evaluación de los procesos de aprendizaje. Las pruebas escritas pueden ser clasificadas, de acuerdo con la modalidad de sus ítems en:

- Pruebas de Ensayo
- Pruebas Objetivas

#### a) PRUEBAS DE ENSAYO.

Son aquellas cuyos ítems solicitan al alumno exponer su conocimiento sobre el tema ya sea a través de una respuesta extensa o una respuesta restringida

**RESPUESTA EXTENSA:** Organización libre y personal de sus ideas, no limita la amplitud de las respuestas. Depende únicamente del conocimiento y criterio del estudiante.

**RESPUESTA RESTRINGIDA:** Adecuación de la respuesta a una serie de restricciones. Exige respuestas más precisas en cuanto a extensión y contenido.

#### VENTAJAS DE LAS PRUEBAS DE ENSAYO.

- Facilitan al alumno el estudio profundo de los temas analizados en clase.
- Concede libertad en la expresión de ideas; enseña al alumno a ser original en sus criterios, acciones y valores
- Permite al alumno elaborar su propio marco de referencia y su propia manera de resolverla.

#### DESVENTAJAS DE LAS PRUEBAS DE ENSAYO.

- Las respuestas pueden ser variadas y de diferente extensión, por lo tanto difíciles de valorar.

- Impide una evaluación global de los contenidos, ya que abarcan contenidos referidos a un solo campo.
- Resultan más difíciles de puntuar que otro tipo de ítems, dado que el contenido puede variar de una respuesta a otra.

## b) PRUEBAS OBJETIVAS

Son un instrumento que permite al docente conocer los resultados de aprendizaje de los alumnos con menor grado de intervención de opiniones personales. Existen diferentes ítems dentro de las pruebas objetivas y cada uno permite medir diferente tipo de procesos.

- **ÍTEMS COMPLEMENTARIOS:** Consiste en un enunciado verdadero en el que falta una palabra importante o central para que los alumnos cubran el espacio vacío con el término apropiado.

### **EJEMPLO**

Antonio José de Sucre nació en \_\_\_\_\_.

- **ÍTEMS DE RESPUESTA ALTERNATIVA:** Consiste en la presentación de enunciados frente a los cuales el alumno debe decidir si son verdaderos o falsos.

### **EJEMPLO.**

Escriba en el paréntesis adjunto a la oración la letra **V** si es verdadero o **F** si es falsa la aseveración.

(    ) La ciudad de Quito es llamada Luz de América

- **ÍTEMS DE EMPAREJAMIENTO:** Consiste en la presentación de dos listas a partir de las cuales se solicita al alumno que relacione cada ítem de la primera (premisas) con un ítem de la segunda (respuestas).

### **EJEMPLO**

En la columna A se presentan algunos problemas y en la columna B posibles soluciones. En los paréntesis adjuntos en la columna A escriba el número que le corresponde de acuerdo con la columna B.

**COLUMNA A**

- A. Recalentamiento de la tierra ( )  
 B. Extinción de especies ( )  
 C. Sequías ( )

**COLUMNA B**

1. Protección de fauna autóctona  
 2. Disminución de uso de aerosoles  
 3. Riego Artificial  
 4. Ingreso de especies exóticas.

- **ÍTEMS DE ORDENAMIENTO:** Consiste en presentar al alumno diversos hechos, etapas o fenómenos, para que sean ordenados de acuerdo con las disposiciones exigidas.

**EJEMPLO**

Te presentamos el nombre de algunas provincias del Ecuador, para que en los numerales adjuntos los ubiques de Norte a Sur.

IMBABURA	1. -----
EL ORO	2. -----
CARCHI	3. -----
CHIMBORAZO	4. -----
ZAMORA	5. -----

- **ÍTEMS DE ASOCIACIÓN:** Consisten en presentar expresiones relacionadas alrededor de una idea central. Las mismas ofrecen una clave para que el estudiante “asocie” la respuesta correcta.

**EJEMPLO**

Escribe el nombre del planeta que se describe a continuación:

- Es de color rojizo.
- El movimiento de traslación lo hace en un año y 321,7 días.
- Es parecido a la Tierra.
- Posee dos satélites naturales.

Respuesta: -----

La Universidad Politécnica Salesiana en su Departamento de Idiomas utiliza diversos tipos de pruebas para el desarrollo del programa de inglés en sus diferentes niveles pero nos centraremos en las pruebas escritas objetivas de

opción múltiple, que son las que se utilizan a la mitad y al final de cada nivel, con excepción de la modalidad a distancia que utiliza varios formatos.

- **ÍTEMS DE SELECCIÓN MÚLTIPLE:** Es la presentación de un enunciado problema o situación, seguido de una serie de alternativas u opciones múltiples. Entre estas, una es la correcta (la respuesta) y las otra son respuestas plausibles pero incorrectas (los distractores).

#### **EJEMPLO**

La capa atmosférica donde existe vida es:

- a) Ionosfera
- b) Troposfera
- c) Estratósfera.

#### **CARACTERÍSTICAS DE LAS PRUEBAS OBJETIVAS**

- **BREVEDAD DE RESPUESTA:** Todo ítem exige una respuesta corta que puede consistir en una señal, marca, número, letra, etc.
- **EXACTITUD DE CORRECCIÓN:** Tiene que elaborarse de tal manera que su respuesta pueda ser comprobada por medio de una clave, hace uso de una plantilla que evita la influencia subjetiva del examinador y que por lo tanto puede ser corregida por cualquier persona que desconozca la materia con solo aplicar la plantilla.
- **BREVEDAD DE CORRECCIÓN:** El uso de la plantilla produce brevedad en la corrección.

#### **1.2.1.6 DISEÑO Y CONSTRUCCIÓN DE LOS TEST**

Cabe mencionar una vez más que nos centraremos en el tipo de prueba que es utilizado por el Departamento de Idiomas de la Universidad Politécnica Salesiana, es decir, pruebas objetivas de opción múltiple.

### 1.2.1.6.1 CARACTERÍSTICA EXTERNA DE LA PRUEBA OBJETIVA

La característica externa principal en la prueba objetiva es que sus ítems constan de:

1. Una base o enunciado
2. Distractores, uno de ellos constituye la respuesta a la base.

En el aspecto externo de la prueba se debe considerar varios aspectos:

- **DATOS INFORMATIVOS DE LA PRUEBA:** Nombre del plantel, nombre del alumno, curso asignatura, fecha. En muchos casos el instrumento viene con número y se omite el nombre del alumno para que la calificación sea anónima
- **INDICACIONES DEL USO DEL INSTRUMENTO:** Pero hay que recordar que los estudiantes ya deben estar familiarizados con todos los tipos de medición.
- Al frente de cada distractor se colocará los espacios en blanco, determinados por paréntesis, cuadrados, etc., para ubicar la respuesta. Casi siempre los distractores están en el lado izquierdo de la página y el espacio para la respuesta a la derecha. Debe guarda armonía en el sentido vertical para poder aplicar la plantilla de corrección. Algunas pruebas tienen hoja de respuestas aparte.

### 1.2.1.6.2 CARACTERÍSTICA INTERNA DE LA PRUEBA OBJETIVA

- **EN LA EXTENSIÓN DE LA ASIGNATURA:** Esta prueba permite abarcar varios temas de la muestra de la asignatura, así como varios niveles de los dominios que integran la parte interna de la prueba.
- La prueba objetiva se puede aplicar en todas las asignaturas y en todos los dominios o áreas, cognoscitiva, afectiva y psicomotriz.

### **1.2.1.6.3 LIMITACIONES DE LA PRUEBA OBJETIVA**

- Los maestros señalan que estas pruebas son propensas a la comunicación ya que los alumnos, pueden decirse fácilmente las respuestas, sería de esperar que en la formación afectiva del estudiante se recalque el valor de la honradez en todo aspecto.
  
- No permite en algunos casos llegar a los procesos mentales muy altos.

### **1.2.1.6.4 PRUEBA OBJETIVA DE OPCIÓN MÚLTIPLE.**

Lafourcade, afirma que “el área de aplicación de estas pruebas abarca la mayoría de los resultados del aprendizaje que promueve la acción escolar a través de la estrategia docente”. Además expertos y docentes afirman que “estas pruebas, están libres de las agudas críticas soportadas por las otras pruebas”. La respuesta de los alumnos se hallan menos sujetas a la adivinación, su estructura es menos artificial, los puntajes son más objetivos, miden resultados de aprendizaje complejos.

#### **1.2.1.6.4.1 ASPECTOS DE LA PRUEBA DE OPCIÓN MÚLTIPLE.**

1. Los Ítems, están compuestos de: Una base y una serie de opciones, una de las cuales es la correcta y las demás son distractores. La redacción de la base es simplificada. Es casi una afirmación inconclusa.
2. Todos los ítems deben responder a la característica anterior, ya que esto da un aspecto de homogeneidad a la prueba, refuerza su objetividad.
3. Los ítems deben estructurarse de acuerdo a los procesos de los dominios cognitivo, afectivo y psicomotriz.
4. Esta prueba esta dirigida a medir los niveles del área cognitiva, sin embargo puede incluirse de acuerdo al criterio del maestro aspectos psicomotrices y afectivos en una sola prueba.
5. La prueba de opción múltiple permite medir procesos altos.

### 1.2.1.6.4.2 LOS ÍTEMS

**FORMAS DE ELABORAR LA BASE:** Generalmente se presenta en forma clara y afirmativa.

**LOS DISTRACTORES:** Elementos claves del ítem, variables destinadas a la solución de la base.

**NÚMERO:** El número de los distractores varía de acuerdo al criterio del maestro, puede constar de 3, 4 ó 5 variables.

**EXPRESIÓN DE LOS DISTRACTORES:** Están formados de variables o alternativas representadas por letras "a" "b" "c" "d", también tienen números, dígitos o romanos o cualquier elemento que el maestro utilice.

**CLASES DE DISTRACTORES:** Son alternativas que pueden ser tres o cuatro falsas y una verdadera. También hay distractores que son todos verdaderos lo que implica escoger el más acertado, asunto que permite procesos altos. Los distractores se elaboran para indicar relación de causa efecto.

### CARACTERÍSTICAS DE LOS ÍTEMS

Es importante mencionar que los ítems deben:

- Abarcar los procesos mentales, afectivos y psicomotrices.
- Redactar en lenguaje claro-directo, preciso
- Tener independencia entre un ítem y otro.
- Elaborar los distractores en armonía gramatical, es decir con equivalencia en estructura o sintaxis.

De igual manera es importante mencionar que los ítems no deben:

- Medir aspectos relacionados solo a la memorización
- Tener términos muy enfocados a la memoria o evocación ¿Qué? ¿Quién?, ¿Dónde?, ¿Cuándo?, solo se utilizarán en circunstancias cuidadosamente establecidas.
- Tener dependencia de un ítem con otro que revele la respuesta.
- Tener desarmonía gramatical en los distractores.

### 1.2.1.6.4.3 INSTRUMENTOS PARA CORRECCIÓN DE PRUEBAS OBJETIVAS<sup>4</sup>

#### LAS PLANTILLAS

En la actualidad la velocidad es un requisito indispensable en las actividades, claro esta, que debe ir de la mano con la eficiencia y la eficacia. La corrección de las pruebas constituye una tarea dura, larga para el docente, en la que emplea horas extra-clase. Para aliviar un tanto esta actividad, las pruebas objetivas permiten el uso de las “**plantillas de corrección**” formada por una hoja con orificios, los que deben ser perforados, precisamente en la respuesta correcta, lo que permite que una persona que desconoce el tema pueda corregir la prueba, ya que la plantilla guía la respuesta.

#### TIPOS DE PLANTILLAS.

Existen dos tipos de plantillas: Plantilla sobre la prueba y plantilla de respuesta en hoja aparte.

- **PLANTILLA SOBRE LA PRUEBA:** Se utiliza cuando el alumno escribe la respuesta en la misma hoja de la prueba. La perforación de la plantilla será en la opción correcta.
- **PLANTILLA DE RESPUESTAS EN HOJA APARTE:** Se entrega una hoja aparte donde deben escribir los alumnos las respuestas

#### REQUISITOS PARA ESCRIBIR LAS RESPUESTAS EN UNA HOJA APARTE.

- Instrucciones precisas
- Las letras de los distractores

En las instrucciones se debe señalar:

- Tache con una X su respuesta correcta
- Escoja una sola letra

---

<sup>4</sup> PEDAGOGÍA DE LA EVALUACIÓN. POVEDA, Elva. Quito. 1993.

- Cuidar de que el número de la pregunta de la prueba corresponda con el número del formulario de la prueba.

### EJEMPLO

Se supone que la prueba tiene 20 preguntas con 3 distractores, la hoja aparte constará de: distractores y el alumno hará la cruz en la correcta o la que escoja.

1.- a b <input checked="" type="checkbox"/> c	8.- a b <input checked="" type="checkbox"/> c	15.- a <input checked="" type="checkbox"/> b c
2.- a <input checked="" type="checkbox"/> b c	9.- a <input checked="" type="checkbox"/> b c	16.- a b <input checked="" type="checkbox"/> c
3.- a b <input checked="" type="checkbox"/> c	10.- <input checked="" type="checkbox"/> a b c	17.- a <input checked="" type="checkbox"/> b c
4.- a <input checked="" type="checkbox"/> b c	11.- a <input checked="" type="checkbox"/> b c	18.- a b <input checked="" type="checkbox"/> c
5.- <input checked="" type="checkbox"/> a b c	12.- <input checked="" type="checkbox"/> a b c	19.- a <input checked="" type="checkbox"/> b c
6.- <input checked="" type="checkbox"/> a b c	13.- <input checked="" type="checkbox"/> a b c	20.- <input checked="" type="checkbox"/> a b c
7.- a <input checked="" type="checkbox"/> b c	14.- a b <input checked="" type="checkbox"/> c	

Figura 1.1 Ejemplo de examen con 3 distractores<sup>5</sup>

### PLANTILLA DE CORRECCIÓN DE LA HOJA DE RESPUESTAS PARA EL PROFESOR

a b c	a b c	a b c
1.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	8.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	15.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
2.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	9.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	16.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
3.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	10.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	17.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
4.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	11.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	18.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
5.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	12.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	19.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
6.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	13.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	20.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
7.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	14.- <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	

Figura 1.2 Ejemplo de Plantilla de corrección de la Hoja de Respuestas<sup>6</sup>

<sup>5</sup> Fuente: Poveda Elva. Pedagogía de la evaluación. 1993.

<sup>6</sup> Fuente: Poveda Elva. Pedagogía de la evaluación. 1993.

#### **1.2.1.6.4.4 PAUTAS PARA ESCRIBIR PRUEBAS DE OPCIÓN MÚLTIPLE.**

Cabe mencionar en este punto, la importancia de la serie de alternativas que tienen las Pruebas Objetivas, cada una de ellas demuestra su importancia y diversidad en el uso, no solo permiten una evaluación objetiva del conocimiento, sino que ayudan al docente en su pronta calificación.

Como ya se ha mencionado se pondrá especial énfasis en las Pruebas de Opción Múltiple, por lo tanto se presenta las pautas más importantes para elaborar este tipo de pruebas.

#### **PAUTAS PARA EL CONTENIDO**

1. Cada pregunta debe reflejar un contenido específico y un sólo comportamiento mental específico, tal como se requiere en las especificaciones del examen.
2. Basar cada pregunta en algún punto importante del contenido que aprender; evitar contenido trivial.
3. Usar material novedoso para medir aprendizaje de un nivel superior. Parafrasear el lenguaje del libro de texto o el lenguaje usado durante la instrucción en un examen para evitar la simple memorización del material.
4. Guardar el contenido de cada pregunta independiente del contenido de otras preguntas del examen.
5. Evitar un contenido demasiado específico o demasiado general al escribir las Pruebas de Opción Múltiple.
6. Evitar preguntas basadas en opiniones.
7. Evitar preguntas capciosas.
8. Mantener el vocabulario sencillo (apropiado) para cada grupo de estudiantes.

Por lo tanto es importante en las pautas sobre el contenido tener un vocabulario simplificado. Además es esencial para este proyecto de tesis para el Departamento de Idiomas mencionar que como se enseña a estudiantes otra lengua, la inclusión de vocabulario sofisticado y difícil puede resultar en calificaciones más bajas de lo que en realidad se merecen.

**PAUTAS PARA EL FORMATO.**

1. Usar las tres versiones de la Prueba de Opción Múltiple: la de la pregunta, la que se tiene que completar y la que requiere la mejor respuesta.
2. Formatear la pregunta verticalmente en lugar de horizontalmente. Se debe escoger el formato indicado y presentar la pregunta de manera legible, preferiblemente en formato vertical.

**PAUTAS PARA EL ESTILO.**

1. Editar y corregir las preguntas.
2. Usar gramática, puntuación, mayúsculas y ortografía correctas.
3. Reducir al mínimo la cantidad de palabras en cada pregunta.

No deberían existir errores ortográficos o gramaticales ya que se implica una distracción para los estudiantes, además que puede originar una mala impresión del examen y su autor.

**REDACCIÓN DEL TALLO.**

1. Asegurar que las instrucciones en el tallo estén claras.
2. Incluir la idea central en el tallo en lugar de ponerla en las opciones.
3. Evitar adornos (verbosidad innecesaria).
4. Afirmar en lugar de negar y evitar palabras como NO y EXCEPTO. Si se usan palabras negativas, usar la palabra con cuidado y siempre asegurar que aparezca en MAYÚSCULAS y en NEGRITA.

El tallo debe presentar ideas sencillas, claras, singulares y en forma de pregunta preferiblemente.

## **REDACCIÓN DE LAS OPCIONES.**

1. Incluir tantas opciones viables como pueda, pero se sugiere que tres opciones son suficientes.
2. Asegurarse que sólo una de las opciones sea la respuesta correcta.
3. Variar el lugar de la opción correcta según el número de opciones.
4. Ubicar las opciones en un orden lógico o numérico.
5. Mantener las opciones independientes; las opciones no deberían traslapar.
6. Mantener las opciones homogéneas en términos de contenido y de estructura gramatical.
7. Mantener homogeneidad en la extensión de las opciones.
8. Nunca se debería usar como última opción: **NINGUNA DE LAS OPCIONES ANTERIORES.**
9. Así mismo, evitar: **TODAS LAS OPCIONES ANTERIORES.**
10. Escribir opciones que afirman; evitar negativos como NO.
11. Evitar pistas que indican la opción correcta, como:
  - Palabras inclusivas como “siempre”, “nunca”, “completamente” y “absolutamente”.
  - Asociaciones lingüísticas, opciones con palabras idénticas o que se parecen a palabras en el tallo.
  - Inconsistencias gramaticales que dan una pista a la opción correcta.
  - Una opción notablemente correcta.
  - Opciones en pares o en tríos que identifican claramente la opción correcta.
  - Opciones patentemente absurdas o ridículas.
12. Asegurar que todas las añagazas sean opciones verosímiles.
13. Usar errores típicos de los estudiantes para sus artimañas.
14. Usar humor si es compatible con el profesor y con el medio ambiente del aula.

No se debe influir engañosamente a los estudiantes a escoger una opción incorrecta, ya que se trata de diferenciar entre los que han aprendido más y los que han aprendido menos.

### 1.2.1.7 EXÁMENES COMPUTARIZADOS.

El mundo está en constante cambio. Un gran ejemplo de esto es la tecnología, vemos como esta ha estado tomando gran control de nuestras rutinas diarias tanto en el ámbito profesional como en el educativo. Esto involucra innovación y renovación de los materiales educativos a utilizar. Dentro de todos estos avances se ha desarrollado un campo llamado Tecnología Educativa, que busca crear materiales educativos que permitan a los actores involucrados en el proceso enseñanza-aprendizaje, mejorar y facilitar este proceso, esta tiene muchos efectos positivos, uno de ellos poner mas cuidado en descubrir cuales son los verdaderos objetivos de la enseñanza y cuales son los objetivos en relación con las características individuales de los estudiantes, además de conocer y valorar el aspecto humano y humanizador de la enseñanza.

Uno de los campos en que más lento se ha introducido el uso de las computadoras es la Educación, en contraste con otros campos como las industrias y negocios en donde su uso es notable, esto sucede debido a la dificultad para precisar las capacidades de dichos equipos para apoyar el proceso de enseñanza-aprendizaje. Sin embargo una vez demostrada su capacidad en el proceso se ha utilizado principalmente en la investigación educativa, esto derivó en lo que hoy se conoce como enseñanza dirigida por la computadora que se basa en cuatro funciones básicas:

- Almacenar la información sobre cada uno de los estudiantes
- Almacenar series de objetivos y los medios de evaluación
- Establecer una relación de las características de cada estudiante y
- Registrar las posibles decisiones de orden pedagógico en relación con el material y las actividades que se puedan realizar con cada estudiante.

En la actualidad el enfoque que se le ha dado al uso de la computadora es en la administración de tareas, exámenes y asesorías, con la intención de agilizar los mismos, facilitar el aprendizaje y reducir el tiempo utilizado en la evaluación, así como la reducción de costos (Bernt & Bugbee, 1990).

#### **1.2.1.7.1 VENTAJAS DE LOS EXÁMENES COMPUTARIZADOS.**

- No existen diferencias significativas ni en actitud ni en ansiedad de un examen computarizado respecto a uno tradicional.
- Hace posible incrementar la eficiencia en la duplicación de exámenes en: la administración, la evaluación, creación de un banco de reactivos, la personalización de exámenes, el registro y análisis.
- Facilitan las condiciones de exámenes estandarizados, administración individual, evaluación inmediata y diversidad de formatos, entre otros.
- Proporciona a los educadores retroalimentación inmediata
- Ha demostrado que puede conseguir un mejor rendimiento del estudiante cuando recibe información inmediata sobre la corrección o falsedad de su respuesta.
- A los alumnos les agrada el entorno gráfico de los exámenes.
- Las computadoras tienen la ventaja de ser completamente objetivas, lo que no es posible con los humanos.

#### **1.2.1.7.2 DESVENTAJAS DE LOS EXÁMENES COMPUTARIZADOS.**

- Frustración por no poder ver preguntas previas o cambiar alguna de las respuestas ya enviadas.

#### **1.2.1.7.3 SISTEMAS DE EVALUACIÓN EXISTENTES EN EL MERCADO**

En el área Educativa destacan los sistemas asistidos por computadora, los generadores de exámenes o inclusive cursos completos de entrenamiento, por lo que se puede decir que la tecnología computarizada ha acelerado el desarrollo de las aplicaciones educativas. Algunos de los sistemas que destacan son:

**TOEFL® - TEST OF ENGLISH AS A FOREIGN LANGUAGE.**

El Test of English as a Foreign Language™ (TOEFL®), mide la habilidad de los angloparlantes no nativos para utilizar y entender el Inglés ya sea hablado, escrito y auditivamente comprendido en las instituciones educativas en el exterior. Este examen se ofrece en diferentes formatos, dependiendo de la ubicación de quien aplica. La mayor parte de las personas toman el TOEFL como un prerrequisito para la admisión en los Colleges y Universidades donde el inglés es utilizado o requerido. El examen TOEFL IBT, tiene 4 secciones, las cuales son: Comprensión Auditiva (Listening), Estructura (Structure), Lectura (Reading) y Escritura (Writing).

**USMLE® - UNITED STATES MEDICAL LICENSING EXAMINATION, STEP 1 Y STEP 2.**

Estos exámenes computarizados están dirigidos a estudiantes de medicina ó médicos graduados, aspirantes a la certificación Educational Comission for Foreign Medical Graduates (ECFMG).

**EL EXAMEN PET.**

Examen ofrecido por la Universidad de Cambridge ESOL (Inglés para Hablantes de Lenguas Extranjeras). Los candidatos que eligen la versión en computador hacen las partes de lectura, escritura y comprensión auditiva en el computador y éstas se transfieren automáticamente a Cambridge ESOL vía electrónica. La prueba de expresión oral todavía se lleva a cabo en presencia de los examinadores.

## **1.2.2 LENGUAJES DE DESARROLLO.**

### **1.2.2.1 VISUAL STUDIO.NET 2005.**

Microsoft punto Net es una plataforma de desarrollo y ejecución de aplicaciones<sup>7</sup>, ya que brinda las herramientas y servicios necesarios para desarrollar aplicaciones empresariales con mecanismos robustos, seguros y eficientes. Se compone de:

- Entorno de ejecución de aplicaciones o Runtime, que ejecuta las aplicaciones e interactúa con el sistema operativo ofreciendo sus servicios y recursos.
- Biblioteca de funcionalidades y controles reutilizables.
- Lenguajes de Programación: Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) para sistemas Windows. Este entorno soporta varios lenguajes de programación como: Visual C#, ASP.NET, Visual C++, Visual J# y Visual Basic.NET, aunque existen extensiones necesarias para muchos otros.
- Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de las aplicaciones.
- Documentación y guías de arquitectura que describen las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de aplicaciones.NET.

Visual Studio punto NET es la nueva tecnología desarrollada por Microsoft desde el diseño de Windows 3.0. Esta tecnología aumenta la productividad de los desarrolladores y crea nuevas oportunidades de negocio. Además simplifica considerablemente el ciclo de vida de desarrollo de aplicaciones. Permite a los desarrolladores crear aplicaciones, sitios, aplicaciones y servicios Web, en cualquier entorno que soporte la plataforma NET. Hace uso extensivo de numerosos estándares globales que son de uso extensivo en la industria como XML, HTTP, SOAP, WSDL, UDDI.

---

<sup>7</sup> INTRODUCCIÓN MICROSOFT.NET. Desarrollador 5 Estrellas.2005



Figura 1.3 Plataforma de Ejecución<sup>8</sup>

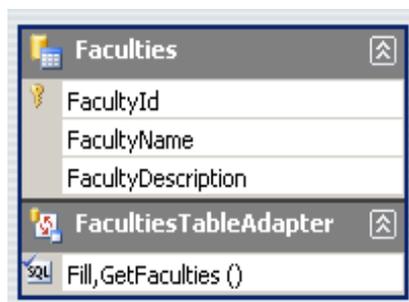
Visual Studio punto NET trae consigo novedades a nivel del Common Language Runtime (CLR), sus nuevas funcionalidades son la base para las mejoras en la tecnología, una de ellas es la definición de clases parciales que permite dividir el código de una clase en varios ficheros, que al compilarse se unen automáticamente para formar la clase completa, esto, evita problemas de sincronización que ocurría anteriormente. Las clases parciales también permiten que varios trabajadores puedan trabajar sobre diferentes partes de una clase sin los conflictos de versionado que esto podía ocasionar.

Con respecto a El Acceso de Datos, Visual Studio trae consigo avances importantes, debido a la optimización del rendimiento, nuevas funcionalidades y principalmente por el aumento de la productividad. Se puede definir Data Sets con una o varias tablas fácilmente con el asistente para configuración de acceso de datos y definir la lógica de acceso a datos básica asociada a cada tabla a través de TableAdapters. Además se puede añadir métodos de acceso a cada tabla asociando sentencias SQL a cada TableAdapter existente en el Data Set, lo que permite crear una capa de acceso a datos de forma rápida y visual.

<sup>8</sup> Fuente: Desarrollador 5 Estrellas. Introducción a Microsoft.Net. 2005.

La capa de presentación mejora mucho tanto en Web como en aplicaciones de escritorio. En Web los vínculos con los datos pueden ser bidireccionales, y nos permiten obtener datos tanto de la base de datos como escribir sus modificaciones en ella. Por otro lado, la vinculación a los datos ahora se realiza automáticamente al cargar la página sin necesidad de llamar al método `DataBind`. En cuanto a aplicaciones de escritorio, se simplifica el trabajo gracias a la incorporación de nuevos componentes y controles para trabajo con datos como el `DataGridView`, `BindingSource` y `BindingNavigator`.

### EJEMPLO.



**Figura 1.4 DataSet Faculties - capa de acceso a datos**

Este ejemplo muestra el `dsFaculties` que es parte de la capa de acceso a datos que se utiliza en este proyecto de tesis. La capa de acceso a datos contiene métodos para acceder a los datos de la base de datos, en el ejemplo `dsFaculties` tenemos el método `GetFaculties()` que devolverá información acerca de todas las facultades, cuando se invoque al método, hace la consulta respectiva y devuelve los resultados.

#### 1.2.2.1.1 GENERALIDADES DEL ENTORNO DE DESARROLLO INTEGRADO (IDE).

Muestra una página de inicio que contiene una lista de vínculos a recursos en el propio IDE o en el Internet, los vínculos están organizados en las siguientes secciones: Proyectos recientes, Introducción, Encabezados de C# Y MSDN.

La barra de menús son los comandos para administrar el IDE y para desarrollar, mantener y ejecutar programas y la barra de herramientas contiene iconos de los comandos más comunes en forma gráfica.

El IDE ofrece ventanas para acceder a los archivos de proyectos y personalizar controles. Una de ellas es la ventana de explorador de soluciones que proporciona acceso a todos los archivos de una solución.

Otro aspecto importante en el IDE es el uso de la ayuda, posee ayuda dinámica la cual es una excelente y rápida forma para obtener información acerca del IDE y sus características. También cuenta con ayuda sensible al contexto la cual muestra al instante un artículo de ayuda relevante, en vez de presentar una lista de artículos.

## **VENTAJAS DE .NET**

- Unifica los modelos de programación.
- Simplifica el desarrollo.
- Es independiente del lenguaje de programación.
- Su Entorno de Ejecución es robusto y seguro.
- Interoperabilidad con código existente.
- Simplifica la instalación y administración de las aplicaciones.

### **1.2.2.1.2 ASP.NET**

ASP.NET es un programa que puede utilizarse en un servidor para generar sitios Web, el cual está formado por páginas dinámicas que pueden acceder y/o modificar recursos del servidor. Una aplicación ASP.NET contiene:

- Formularios .aspx
- Páginas de código.
- Archivos de configuración web.config.
- Archivo global.asax
- Enlaces y/o servicios Web. asmx.
- Conectividad a base de datos.

- Sistema de cache.
- Otros recursos.

**WEB FORMS:** Es un formulario Web, es decir, una página Web, que solo posee una definición de la interfaz visual ya que no contiene código. Este Web form tiene asociado uno o varios archivos de código y su programación esta basada en eventos. Los objetos accesibles desde el Web Form son:

- **RESPONSE:** Es decir la página Web que se enviará al cliente.
- **REQUEST:** Se refiere a la solicitud del usuario.
- **SERVER:** Contiene métodos para acceder a la información en el servidor. El método típico de esta clase es el que permite obtener la ruta de un archivo en el servidor.
- **SESSION:** Los usuarios al conectarse comienzan sesiones en el servidor, que puede guardar información sobre el usuario.
- **APPLICATION:** Esta clase contiene los métodos.

**WEB SERVICE:** Consiste en una función disponible en un servidor conectado al Web. Esta función puede ser un simple cálculo con datos que se le envían como parámetro o acceder a una base de datos para recuperar un conjunto de registros, etc. A un servicio Web se puede conectar: una página asp.net, una aplicación WinForms, un dispositivo móvil, por ejemplo una PDA.

#### **VENTAJAS.**

- Mejor rendimiento y escalabilidad
- Eficacia, flexibilidad y simplicidad
- Facilidad de uso.
- Seguridad.
- Productividad de los desarrolladores.

### 1.2.2.1.3 C #

C# es un lenguaje desarrollado por Microsoft, que tiene sus raíces en C, C++ y Java; agrega nuevas características y adapta las mejores características de los programas anteriores. Es un lenguaje orientado a objetos y posee una poderosa biblioteca de clases, que consta de componentes preconstruidos que permiten a los programadores desarrollar aplicaciones con rapidez.

La plataforma.NET es la infraestructura sobre la cual pueden distribuirse aplicaciones basadas en Web a una gran variedad de dispositivos como teléfonos celulares y computadoras de escritorio. Esta plataforma ofrece un nuevo modelo de desarrollo de software, que permite crear aplicaciones en distintos lenguajes de programación, de manera que se comuniquen entre sí. La plataforma.NET permite la interoperatividad de los lenguajes e incluso empaquetar el software antiguo para que trabaje con nuevos programas en C#.

C# es un lenguaje de programación visual, controlada por eventos, en el cual se crean programas mediante el uso del IDE y en el que un programador puede crear, ejecutar, probar y depurar programas en C# de manera conveniente, con lo cual se reduce el tiempo requerido para producir un programa funcional. Las aplicaciones en C# pueden interactuar a través de Internet mediante el uso de estándares industriales como XML.

**CARACTERÍSTICAS DE C #:** Sus principales características son<sup>9</sup>:

- **SENCILLEZ:** Elimina elementos innecesarios como ficheros de cabecera, macros, etc.
- **MODERNIDAD:** Incorpora elementos útiles para el desarrollo de aplicaciones como un tipo básico decimal para operaciones de alta precisión con reales, incluye la instrucción foreach que permite recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, entre otros.

---

<sup>9</sup> EL LENGUAJE DE PROGRAMACIÓN C#, GONZALES, José Antonio. 2005

- **ORIENTACIÓN A OBJETOS:** C# soporta todas las características propias del paradigma de programación orientada a objetos como encapsulación, herencia y polimorfismo.
- **ORIENTADO A COMPONENTES:** Permite definir cómodamente propiedades, eventos o atributos.
- **GESTIÓN AUTOMÁTICA DE MEMORIA:** No es necesario incluir instrucciones de destrucción de objetos, además proporciona un mecanismo de liberación de recursos determinista a través de la instrucción **using**.
- **SEGURIDAD DE TIPOS:** provee mecanismos que permiten asegurar que los accesos a los tipos de datos siempre se realicen correctamente, evitando que se produzca errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto. Estos mecanismos incluyen por ejemplo: no se pueden usar variables no inicializadas, solo se admiten conversiones entre tipos compatibles, uso de delegados, entre otras.
- **INSTRUCCIONES SEGURAS:** Restricciones en el uso de las instrucciones de control más comunes, por ejemplo, todo **switch** debe terminar en un **break** o **goto** que indique cuál es la siguiente acción a realizar, evitando ejecución accidental de casos y facilita su reordenación.

## EJEMPLO.

Este ejemplo es una de las clases de esta aplicación, la clase se denomina *Modality*, en la que están definidos los atributos y métodos de los que van a disponer todos los objetos de la misma, en este caso tenemos definidas propiedades que permiten implementar el encapsulamiento, impidiendo el acceso directo a los atributos de una clase y proporcionando un mecanismo para poder obtener y asignar sus valores

```

1  using System.Web.UI.HtmlControls;
2  using System.Web.UI.WebControls;
3  using System.Web.UI.WebControls.WebParts;
4  )
5  /// <summary>
6  /// Summary description for Modality
7  /// </summary>
8  namespace OnlineTesting.BLL
9  {
10     public class Modality
11     {
12         #region variables
13         private string valueModality;
14         private string itemModality;
15
16         #endregion
17         Constructor
18
19         #region Properties
20         public string ValueModality
21         {
22             get { return this.valueModality; }
23             set { this.valueModality = value; }
24         }
25         public string ItemModality
26         {
27             get { return this.itemModality; }
28             set { this.itemModality = value; }
29         }
30         #endregion
31     }
32 }

```

**Figura 1.5 Ejemplo de una Clase en C#**

#### 1.2.2.1.4 XML

El lenguaje de marcado extensible XML es una tecnología no propietaria con amplio soporte para describir datos y se ha convertido en el estándar para el intercambio de datos entre aplicaciones a través de Internet.

Los documentos de XML describen datos en forma independiente de la máquina. Al integrar XML a las aplicaciones se mejora la funcionalidad e interoperabilidad de la Web. XML no se limita únicamente a las aplicaciones Web sino que cada vez se está utilizando más en las bases de datos. Utiliza el protocolo simple de acceso a objetos SOAP, una tecnología para la transmisión de objetos a través de Internet.

XML permite a los autores de documentos crear **marcado**, es decir, una notación basada en texto para describir datos para casi cualquier tipo de información, en una forma que tanto los seres humanos como las computadoras pueden comprender. Los documentos XML son altamente portables, no se necesita un software especial, cualquier editor de texto con soporte para los caracteres de código ASCII/Unicode puede verlos y editarlos. Para el procesamiento de documentos XML se utiliza un analizador de documentos XML, usualmente integrado en el Visual Studio.

XML con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzadas, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes.

#### **1.2.2.2 BASE DE DATOS.**

Una base de datos es una colección de datos organizados y estructurados según un determinado modelo de información que refleja tanto datos como relaciones existentes entre ellos. Una base de datos es diseñada con un propósito específico, sus datos podrán ser compartidos por distintos usuarios y aplicaciones pero debe conservar su integridad y seguridad.

Sus principales características son:

- Independencia lógica y física de los datos
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de datos.
- Consultas complejas optimizadas
- Seguridad de acceso y auditoria.
- Respaldo y recuperación.
- Redundancia mínima.
- Acceso a través de lenguajes de programación estándar.

### 1.2.2.2.1 MODELO DE BASE DE DATOS.

Un modelo de datos es una colección de conceptos bien definidos matemáticamente que se emplean para describir la estructura de una base de datos, es decir, esta colección ayuda a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo.

El modelado de datos es uno de los elementos más importantes a la hora de iniciar el desarrollo de cualquier proyecto, ya que es la estructura sobre la que reside la esencia de la aplicación.

Los modelos de datos pueden clasificarse en:

- **MODELOS DE DATOS DE ALTO NIVEL O CONCEPTUALES:** Disponen de conceptos cercanos a la forma en que los usuarios finales perciben una base de datos. El diseño de un modelo conceptual parte de la especificación de requisitos. Su propósito es describir el contenido de la información de la base de datos, en vez de las estructuras de almacenamiento que se requerirán para manejar esa información.
- **MODELO DE DATOS DE BAJO NIVEL O FÍSICOS:** Disponen de conceptos que describen detalles sobre el almacenamiento de los datos en la computadora. El paso del modelo lógico a uno físico requiere un profundo entendimiento del manejador de bases de datos que se desea emplear, incluyendo características como: Conocimiento a fondo de los tipos de objetos soportados, detalles acerca del indexamiento, integridad referencial, restricciones, tipos de datos, detalles y variaciones de las versiones, parámetros de configuración, Data Definition Language (DDL).
- **MODELO DE DATOS DE REPRESENTACIÓN O DE IMPLEMENTACIÓN:** Disponen de conceptos que pueden entender los usuarios finales, pero que no están alejados de la forma en que se almacenan los datos en la computadora.

Conceptualmente una aplicación puede ser caracterizada por:

- **PROPIEDADES ESTÁTICAS:** entidades u objetos, propiedades o atributos de esas entidades y relaciones entre esas entidades
- **PROPIEDADES DINÁMICAS:** operaciones sobre entidades, sobre propiedades o relaciones entre operaciones.
- **REGLAS DE INTEGRIDAD:** sobre las entidades y las operaciones.

Un modelo se diferencia de otro por el trato que se le da a estas 3 categorías. Los modelos clásicos de tratamiento de datos son:

- Jerárquico
- En red
- Relacional

En la actualidad también son muy utilizadas las bases de datos distribuidas debido al uso de arquitecturas de cliente servidor y trabajo en grupo, y los modelos de bases de datos orientada a objetos, la cual esta representada por un conjunto de clases que definen las características y el comportamiento de los objetos que conforman la base de datos. El tema propuesto en esta tesis se basa en el modelo relacional, es por esta razón que nos enfocaremos únicamente en este modelo.

#### 1.2.2.2.1.1 MODELO ENTIDAD RELACIÓN

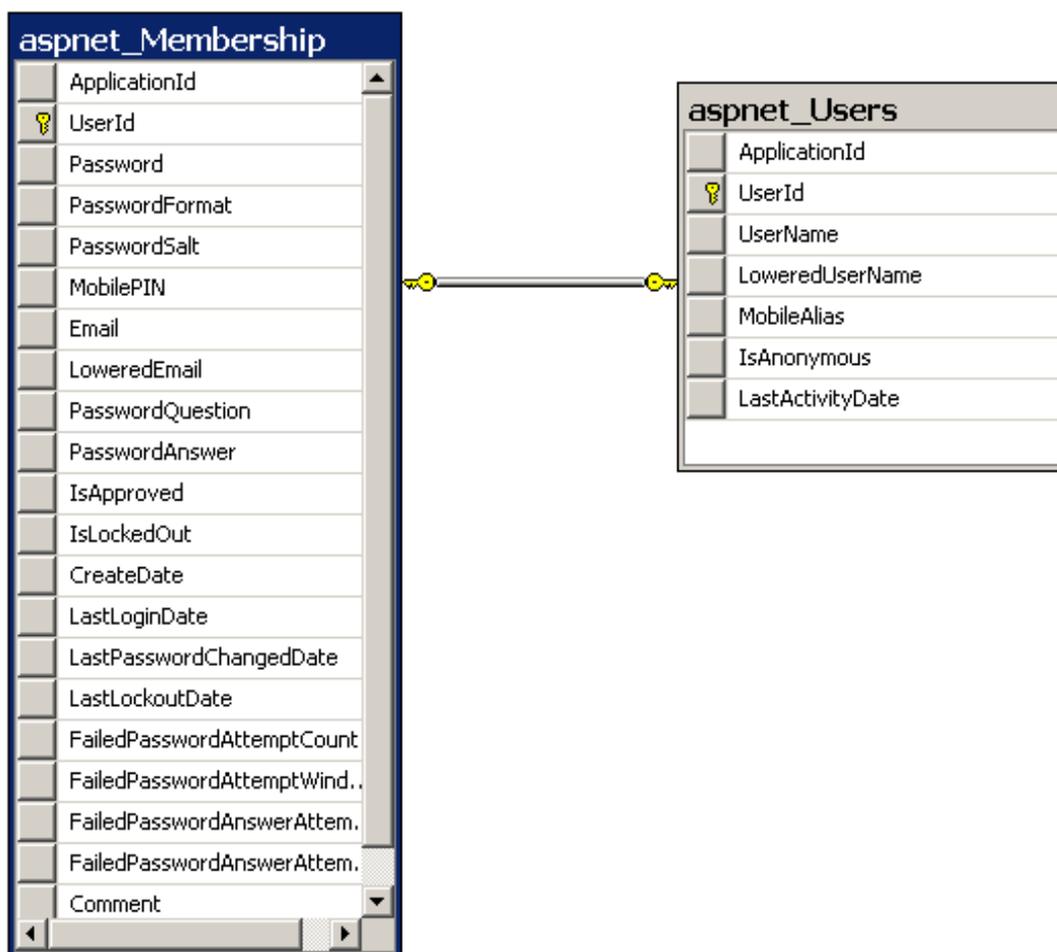
Este es el modelo de base de datos más utilizado, ya que permite una mayor eficacia, flexibilidad y confianza en el tratamiento de datos. El modelo entidad relación permite diseñar el esquema conceptual de una base de datos. Se basa en los conceptos de entidad, tipos de entidad, atributo y relación. Toda esta información se representa en los diagramas entidad/relación.....

En este modelo se representa el mundo real mediante tablas relacionadas entre si por columnas comunes. La estructura principal en el modelo de relaciones es la **relación**, es decir, una tabla bidimensional compuesta por tuplas y columnas. Las características de cada entidad están definidas por las columnas de las relaciones, que se llaman **atributos**.

Las **interrelaciones** son las que existen entre varias tablas del sistema. Existen 3 formas de interrelaciones dependiendo de la cardinalidad con la que se combinan los elementos en ambas tablas:

- **INTERRELACIONES UNO A UNO:** Una interrelación es uno a uno entre la tabla A y la tabla B cuando a cada elemento de la clave de la tabla A se le asigna un único elemento de la tabla B y para cada elemento de la clave de la tabla B contiene un único elemento de la tabla A. Se representa la relación como A 1:1 B.

### EJEMPLO



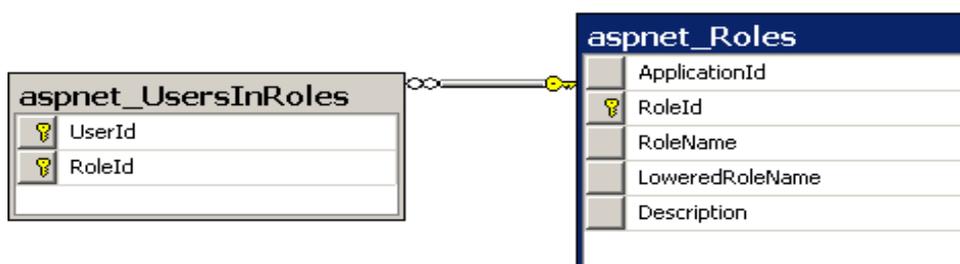
**Figura 1.6** Tablas de la aplicación con relación uno a uno

En este par de tablas encontramos el almacenamiento de información de la cuenta de usuario. La tabla aspnet\_User contiene la información esencial de la

cuenta de usuario como UserId, UserName, entre otras. La tabla aspnet\_Membership incluye información adicional de la cuenta de usuario como contraseña, dirección de correo, fecha de último acceso, etc. En este esquema el UserId, UserName y Password representan las credenciales de usuario. La relación uno a uno esta garantizada por el campo UserId de las tablas que sirve como clave principal.

- **INTERRELACIÓN UNO A VARIOS:** Una relación es de uno a varios entre las tablas A y B cuando una clave de la tabla A posee varios elementos relacionados en la tabla B y cuando una clave de la tabla B posee un único elemento relacionado en la tabla A. Se representa la relación como A 1: n B.

#### EJEMPLO



**Figura 1.7 Tablas de la aplicación con relación uno a varios**

Este par de tablas permiten especificar funciones y usuarios que pertenecen a dichas funciones. Cada registro de `aspnet_Roles` representa un rol en la aplicación, cuenta con los campos `RoleId`, `RoleName`, entre otros. La tabla `UserInRoles` cuenta con los campos `UserId` y `RoleId`, juntos constituyen una clave principal compuesta. Esta tabla sirve como una asignación entre usuarios y roles.

- **INTERRELACIÓN DE VARIOS A VARIOS:** Una interrelación es de varios a varios entre la tabla A y B cuando una clave de la tabla A posee varios elementos relacionados en la tabla B y cuando una clave de la tabla B posee varios elementos relacionados en la tabla A. Se representa como A n: n B.

### 1.2.2.2.1.2 PROCESO DE NORMALIZACIÓN.

El proceso de normalización es un estándar que consiste en un proceso de conversión de las relaciones entre las entidades, evitando:

- La redundancia de los datos
- Anomalías de actualización
- Anomalías de borrado
- Anomalías de inserción.

Antes de proceder a la normalización de la tabla primero se debe **definir una clave** que contiene un valor único para cada registro, y podrá estar formado por un único campo o por un grupo de campos. El proceso de normalización nos conduce hasta el modelo físico de datos y consta de varias fases denominadas formas normales:

#### a) PRIMERA FORMA NORMAL (1NF)

Se dice que una tabla se encuentra en 1FN si y solo si cada uno de los campos contiene un único valor para un registro determinado.

#### EJEMPLO

Considere la siguiente tabla:

<b>Código</b>	<b>Nombre</b>	<b>Cursos</b>
1	Eleana	Inglés
2	Roberto	Informática, Matemática
3	Mariela	Matemática, Física

**Tabla 1.1 Ejemplo para poner en Primera Forma Normal**

Aquí el registro de código 1 si cumple la primera forma normal, cada campo del registro contiene un único dato. Los registros 2 y 3 no cumplen con la 1FN

ya que en el campo cursos contiene más de un dato cada uno. La solución es crear dos tablas del siguiente modo:

TABLA A		TABLA B	
Código	Nombre	Código	Curso
1	Eleana	1	Inglés
2	Roberto	2	Informática
3	Mariela	2	Matemática
		3	Matemática
		3	Física

**Tabla 1.2. Ejemplo en Primera Forma Normal**

Ahora se puede observar que todos los registros de ambas tablas contienen valores únicos en sus campos, por lo tanto ambas cumplen la 1FN.

#### b) SEGUNDA FORMA NORMAL (2NF)

La segunda forma normal compara todos y cada uno de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla esta en segunda forma normal.

#### EJEMPLO

Considere la siguiente tabla:

Código Empleado	Código Departamento	Nombre	Departamento	Años
1	6	Eleana	Sistemas	6
2	3	Roberto	Contabilidad	3
3	2	Mariela	Bienes	10
4	3	Sebastian	Contabilidad	1
2	6	Roberto	Sistemas	5

**Tabla 1.3 Ejemplo para poner en Segunda Forma Normal**

Aquí la clave esta formada por los campos **Código Empleado y Código Departamento** y se encuentra en 1FN.

Análisis para la 2FN:

- El campo **nombre** no depende funcionalmente de toda la clave, solo depende del código de empleado.
- El campo **departamento** no depende funcionalmente de toda la clave, solo del código de departamento.
- El campo **años** si depende funcionalmente de la clave, ya que dependen del código de empleado y código de departamento (representa el número de años que cada empleado ha trabajado en cada departamento)

Por lo tanto al no depender todos los campos de la totalidad de la clave, la tabla no esta en 2FN, la solución sería la siguiente:

TABLA A		TABLA B	
Código Empleado	Nombre	Código Departamento	Departamento
1	Eleana	2	Bienes
2	Roberto	3	Contabilidad
3	Mariela	6	Sistemas
4	Sebastián		

TABLA C		
Código Empleado	Código Departamento	Años
1	6	6
2	3	3
3	2	1
4	3	10
2	6	5

**Tabla 1.4. Ejemplo en Segunda Forma Normal**

Se observa aquí que las 3 tablas se encuentran en 2FN, considerando que la tabla A tiene como índice el campo Código de Empleado, la tabla B Código de Departamento y la Tabla C una clave compuesta por los campos Código Empleado y Código Departamento.

### c) TERCERA FORMA NORMAL (3NF)

Una tabla esta 3FN si y solo si los campos de la tabla dependen únicamente de la clave, es decir, los campos de las tablas no dependen unos de otros.

#### EJEMPLO

Código	Nombre	Cursos	Aula
1	Eleana	Inglés	Aula A
2	Roberto	Informática	Aula B
3	Mariela	Matemática	Aula C

**Tabla 1.5 Ejemplo para poner en Tercera Forma Normal**

La dependencia de cada campo con respecto a la clave es:

- **Nombre** depende directamente del código del alumno
- **Curso** depende de igual modo del código del alumno.
- **Aula** depende en parte del alumno, pero esta más ligado al curso que el alumno esta realizando.

Por lo tanto la tabla no esta en 3FN. La solución sería la siguiente:

TABLA A			TABLA B	
Código	Nombre	Curso	Curso	Aula
1	Eleana	Informática	Informática	Aula A
2	Roberto	Inglés	Inglés	Aula B
3	Mariela	Contabilidad	Contabilidad	Aula C

**Tabla 1.6. Ejemplo en Tercera Forma Normal**

#### d) CUARTA FORMA NORMAL (4NF)

Una tabla esta en 4FN si y solo si para cualquier combinación clave-campo no existen valores duplicados.

#### EJEMPLO

GEOMETRÍA		
Figura	Color	Tamaño
Triángulo	Rojo	Grande
Triángulo	Azul	Grande
Triángulo	Azul	Mediano
Círculo	Blanco	Mediano
Círculo	Azul	Pequeño
Círculo	Azul	Mediano

**Tabla 1.7 Ejemplo para poner en Cuarta Forma Normal**

Comparando la clave que es Figura con el atributo tamaño podemos observar que cuadrado grande esta repetido, igual que círculo azul, entre otras. Estas repeticiones son las que se deben evitar para tener una tabla en 4FN. La solución en este caso sería la siguiente:

Tamaño		Color	
Figura	Tamaño	Figura	Color
Triángulo	Grande	Triángulo	Rojo
Triángulo	Pequeño	Triángulo	Azul
Círculo	Mediano	Círculo	Blanco
Círculo	Pequeño	Círculo	Azul

**Tabla 1.8 Ejemplo en Cuarta Forma Normal**

### 1.2.2.2.1.3 ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS

La arquitectura de un sistema de base de datos se basa en 3 niveles:

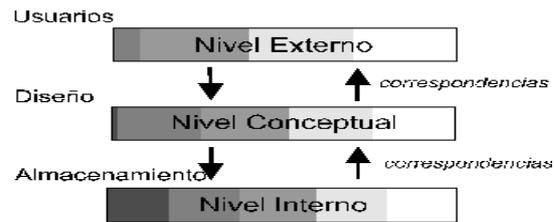
**NIVEL FÍSICO O INTERNO:** Es el nivel más bajo de abstracción y el nivel real de los datos almacenados. Define como se almacenan los datos en el soporte físico ya sea en registros o de cualquier otra forma, así como los métodos de acceso. En este nivel el Esquema Físico es la representación de datos.

**NIVEL CONCEPTUAL:** Es el nivel medio de abstracción, es la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Es una visión de la base de datos desde el punto de vista real, es decir, se trata con la entidad u objeto representado, sin importar como esta representado o almacenado. Incluye la definición de los datos y las relaciones entre ellos. Este nivel lleva asociado un Esquema Conceptual.

**NIVEL VISIÓN O EXTERNO:** Es el nivel de mayor abstracción. Son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos mientras que los usuarios por lo general solo tienen acceso a pequeñas parcelas de esta. El nivel de visión es encargado de dividir estas parcelas. El esquema asociado a este nivel es el Esquema de Visión.

Esta arquitectura establece el principio de independencia de datos, ya sea lógica o física. La independencia lógica significa que los cambios en el esquema lógico no deben afectar a los esquemas externos que no utilicen los datos modificados; la independencia física significa que el esquema lógico no se va a ver afectado por los cambios realizados en el esquema interno correspondiente a modos de acceso, etc.

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que nos permiten describir los datos del universo del discurso. Un esquema es la estructura de datos obtenida tras aplicar dicho modelo.



**Figura 1.8 Arquitectura de un sistema de base de datos<sup>10</sup>**

#### 1.2.2.2.1.4 DISEÑO DE LA BASE DE DATOS.

Los objetivos del modelado de datos son: formalizar y definir las estructuras permitidas para representar los datos, y por otro lado diseñar la base de datos.

En el diseño de base de datos hay que tener en cuenta las siguientes consideraciones:

- La velocidad de acceso.
- El tamaño de información.
- El tipo de información.
- La facilidad de acceso a la información.
- La facilidad para extraer la información requerida.
- El comportamiento del sistema de gestión de base de datos con cada tipo de información.

#### 1.2.2.2.1.5 SISTEMA DE GESTION DE BASE DE DATOS.

Es un programa o aplicación que actúa como interfaz entre los usuarios, los modelos y los sistemas físicos, es decir, es un software que se ejecuta en un ordenador anfitrión que es quien centraliza los accesos a los datos y actúa de interfaz entre los datos físicos y los usuarios. Así pues, las bases de datos pueden ser creadas, mantenidas y gestionadas por los Sistemas de Gestión de Base de Datos (SGBD). Las funciones principales de los SGBD son:

- La definición de las estructuras para almacenar datos, incorporado a través de un lenguaje denominado DDL (Data Definition Lenguaje), el cual provee los

<sup>10</sup> Fuente: <http://tramullas.com/documatica/2-2.html>

medios necesarios para definir los datos con precisión, especificando las distintas estructuras.

- La manipulación de datos, incorporado a través de un lenguaje denominado DML (Data Manipulation/Management Lenguaje), el cual facilita a los usuarios el acceso y manipulación de los datos.

Para el tema de estudio propuesto se ha escogido la utilización de Microsoft SQL Server 2005.

#### **1.2.2.2.1.5.1 MICROSOFT SQL SERVER 2005.**

SQL Server 2005 ha conseguido mejorar la fiabilidad, escalabilidad, rendimiento y manejo. Cabe mencionar que muchas empresas controlan sus redes locales con Windows Server 2003, este sistema operativo es el entorno ideal para la instalación de SQL Server 2005, convirtiéndose en la mejor pareja posible para la administración.

SQL Server 2005 se expande al Internet, ya que es capaz de generar automáticamente documentos XML, el formato estándar de datos que facilita la transmisión de datos en Internet.

SQL Server 2005 tiene varias versiones que se ajustan a las necesidades de diferentes tipos de empresas o clientes. Las características comunes de SQL Server 2005 son:

- Servidor de base de datos de gran rendimiento.
- Puede ser instalado tanto en sistemas de usuario como Windows XP, máquinas de multiprocesador de 64 bits, redes de ordenadores.
- La administración se facilita a través de la interfaz gráfica de usuario.
- Capaz de tener varias instancias del servidor en una única máquina.
- Acceso directo a datos desde página Web, gracias a la generación automática de documentos XML, consiguiendo una completa integración con Internet.
- Posibilidades de data warehousing y data mining, para almacenar y analizar datos.

- Comunicación perfecta con otras aplicaciones Microsoft, pudiendo presentar información en hojas de Excel por ejemplo.
- Integración perfecta con herramientas de desarrollo de software como Visual Studio 2005.
- Lenguaje T-SQL para ampliar las posibilidades de las tareas a realizar.
- Capacidad para interpretar funciones realizadas con CLR de plataformas. NET.

#### **VENTAJAS.**

- Fácil aprendizaje por parte de usuarios que no tienen conocimientos profundos sobre sistemas de bases de datos.
- Ofrece sistemas simples y eficaces para representar y manipular datos.
- Independencia de datos y tratamiento
- Coherencia de resultados.
- Mejora en la disponibilidad de datos.
- Cumplimiento de ciertas normas.
- Eficiente gestión de almacenamiento.

#### **1.2.2.2.1.6 SEGURIDAD DE BASES DE DATOS.**

Las bases de datos se han convertido en un componente muy importante en las aplicaciones Web y en vista de que en estas se almacena información sensible o secreta, se debe considerar seriamente la forma de protegerlas, tomando en cuenta que un buen diseño del esquema de base de datos y aplicación ayuda a contrarrestar enormemente este problema.

Los principales aspectos de la seguridad son:

- **LA CONFIDENCIALIDAD:** Se cumple solo cuando las personas autorizadas pueden conocer los datos o la información correspondiente.
- **LA INTEGRIDAD:** Consiste en que solo las personas autorizadas puedan variar (modificar o borrar) los datos.

- **LA DISPONIBILIDAD:** Se cumple si las personas autorizadas pueden acceder a tiempo a la información.

Las restricciones de seguridad están agrupadas en dos ramas. Por un lado están las que requiere la organización que posee la base de datos, mientras que por otro lado están los aspectos legales.

La seguridad como actividad puede dividirse en planificación e implementación.

**LA PLANIFICACIÓN** determina qué usuario puede hacer qué cosas sobre qué datos. Como la planificación es una política, esta a cargo del Administrador de datos.

**LA IMPLEMENTACIÓN** en cambio, le corresponde al Administrador de Base de Datos, quien tiene una función técnica.

Los dos grandes modelos de seguridad se conocen como control discrecional y control obligatorio.

**CONTROL DISCRECIONAL:** Divide a los usuarios en grupos que tienen determinados permisos y privilegios. Este es el más difundido.

**CONTROL OBLIGATORIO:** Es un sistema jerárquico en el que cada usuario tiene un nivel de clasificación.

Las técnicas disponibles para dotar de seguridad a un SGBD en la Web.

- **SERVIDOR PROXY:** Permite mejorar el rendimiento y filtrar las solicitudes.
- **CORTAFUEGOS:** Impide el acceso no autorizado desde o hacia una red privada. Estos se los implementa tanto en software y hardware.
- **ALGORITMOS DE COMPENDIO DE MENSAJES Y FIRMAS DIGITALES.**

El uso de campos cifrados de contraseñas es altamente recomendable. Es de vital importancia la necesidad de proteger totalmente la máquina completa contra todos los tipos de ataques posibles: interceptación pasiva de paquetes,

reproducción de comandos y denegación de servicios. Es aconsejable no transmitir datos sin cifrar a través del Internet, ya que es susceptible de ser interceptada. Debemos asegurarnos que el mensaje solo sea entendible para aquel usuario al que va destinado.

#### **1.2.2.2.1.7 PROTOCOLOS DE SEGURIDAD**

Los protocolos de seguridad definen las reglas para las comunicaciones, diseñadas para que los sistemas puedan soportar ataques de carácter malicioso. Es la parte visible de la aplicación que cumplen un objetivo específico y utilizan esquemas de seguridad criptográfica.

Un protocolo es una serie de pasos que involucran dos o más principales, es decir, individuos, compañías, computadores, que se comunican por diversos canales (correo, teléfono, tarjetas, etc.), diseñado para realizar una tarea particular.

Un protocolo criptográfico es un protocolo que usa funciones criptográficas en algunos o todos los pasos. Algunos de los protocolos más comunes son:

**SECURE SOCKETS LAYER (SSL):** Este protocolo actúa en la capa de comunicación y es una especie de túnel que protege a toda la información enviada y recibida. SSL puede usar diferentes algoritmos para las aplicaciones, por ejemplo RSA, MD5, SHA-1, DH, DES, TDES, RC2 Y RC4.

La comunicación que es cifrada es:

- El URL del documento requerido
- El contenido del documento requerido
- El contenido de cualquier forma requerida.
- Los cookies enviados del browser al servidor
- Los cookies enviados del servidor al browser
- El contenido de las cabeceras de los http.

**PGP:** Es un protocolo libre usado en el intercambio de correo electrónico seguro.

**IPSEC:** Proporciona seguridad en la conexión de Internet a un nivel mas bajo.

**SET:** Permite dar seguridad en las transacciones por Internet usando tarjeta de crédito, SET requiere un certificado digital en cada paso de autenticación y usa dos pares de claves, una para el cifrado del sobre digital y otra para la firma. Su proceso consiste en:

- El cliente inicializa la compra.
- El cliente usando SET envía la orden y la información de pago al comerciante.
- El comerciante pasa la información de pago al banco.
- El banco verifica la validez del requerimiento.
- El emisor de la tarjeta autoriza la transacción
- El banco del comerciante autoriza la transacción.
- El servidor del comerciante completa la transacción.
- El comerciante captura la transacción.
- El generador de la tarjeta envía el aviso de crédito al cliente.

Todos estos protocolos procuran resolver algunos de los problemas de seguridad como la integridad, confiabilidad, autenticación.

## 1.2.3 ENTORNO DE DESARROLLO WEB.

### 1.2.3.1 WINDOWS SERVER 2003

Es un servidor de la marca Microsoft, basado en la tecnología NT. Sus principales características son:

- Sistema de archivos NTFS
- Gestión de almacenamiento.
- Windows drive model, es decir, la implementación básica de los dispositivos más utilizados.
- Active Directory basado en LDAP, permite gestionar centralizadamente la seguridad de una red corporativa a nivel local
- Autenticación Kerberos5
- DNS con registros de IP's dinámicamente
- Políticas de seguridad.

Las versiones con que cuenta este servidor son:

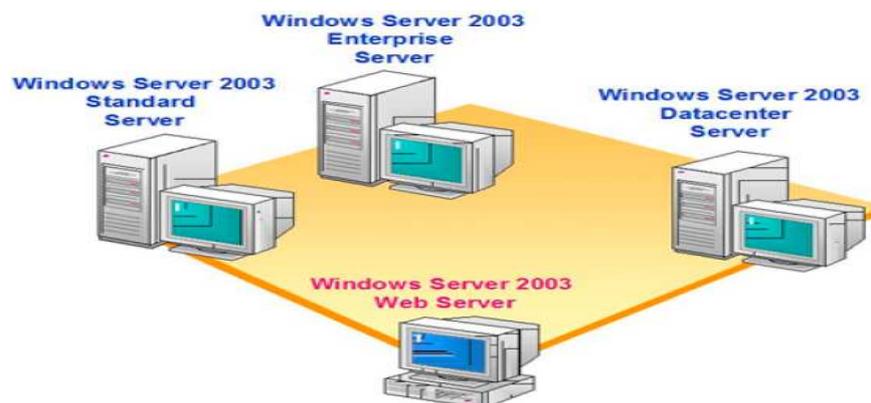


Figura 1.9 Versiones de Windows Server 2003.<sup>11</sup>

- **WEB EDITION:** Diseñado para los servicios y el hospedaje Web.

<sup>11</sup> Fuente: TechNet. Windows Server 2003. Microsoft 2003.

- **ESTÁNDAR EDITION:** El más versátil de todos, ofrece un gran número de servicios útiles para empresas de cualquier tamaño.
- **ENTERPRISE EDITION:** Para empresas de mayor tamaño que la estándar Edition.
- **DATACENTER EDITION:** Para empresas que requieran bases de datos más escalables y un proceso de transacciones de gran volumen.

#### 1.2.3.1.1 SERVIDORES DE WINDOWS SERVER 2003.

Los servidores que maneja Windows 2003 son:

- **SERVIDOR DE ARCHIVOS:** Es un tipo de servidor dentro de un grupo de computadoras conectadas entre sí a través de una red, que permite el acceso remoto a archivos que se encuentran almacenados en ella o a los que esta pueda acceder. Para el cliente, la localización de los archivos compartidos es transparente es decir, que no hay diferencias perceptibles si un archivo esta almacenado en un servidor de archivos remoto o en el disco de la propia máquina.
- **SERVIDOR DE APLICACIONES:** Windows Server 2003 ofrece una solución completa para crear rápidamente aplicaciones interconectadas. Servicios como .NET Framework, Message Queuing, COM+ y otros más permiten mejorar la productividad a lo largo de todo el ciclo de vida de las aplicaciones, desde su diseño a su puesta en marcha y operaciones. Con esta opción podremos disponer de un único equipo servidor donde se instalarán todas las aplicaciones de la organización, al que accederán los usuarios por medio del servidor de aplicaciones. Es un servicio capaz de proporcionar un equipo virtual al cliente que se conecte. Por lo tanto solo será necesario instalar las aplicaciones de la organización en un único equipo. Los clientes se conectan a este equipo y podrán obtener en su equipo una pantalla con las mismas opciones que un equipo normal, de esta forma evitamos instalar todas las aplicaciones en todos los equipos clientes que necesiten conexión a una base de datos. Cuando se tenga que actualizar una aplicación, no será necesario hacerlo en todos los equipos clientes sino solo en el servidor de aplicaciones.

- SERVIDOR DE IMPRESIONES:** Su función es administrar las impresiones, los permisos de impresión, etc. De esta manera clientes de una gran variedad de equipos y sistemas operativos podrán enviar trabajos a impresoras conectadas localmente a un solo servidor de impresión que utiliza Windows Server 2003, a través de Internet, a impresoras conectadas a la red mediante adaptadores internos o externos, o a otro servidor. Uno de los beneficios es que permite a un equipo cliente con Windows XP utilizar la impresora sin que sea necesario instalar un controlador de impresora, dado que el controlador se descarga automáticamente cuando el cliente se conecta con un servidor de impresión que ejecuta un sistema operativo de la familia de Windows Server 2003.

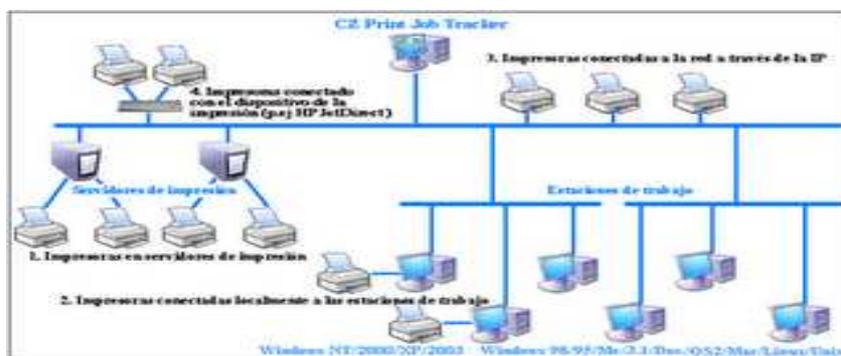
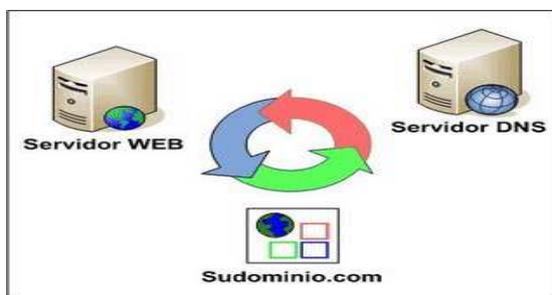


Figura 1.10 Servidor de Impresiones.<sup>12</sup>

- SERVIDOR DE CORREO:** Los servicios de correo electrónico conforman la fase para proporcionar compatibilidad con la firma digital y el cifrado de tráfico de correo electrónico dentro o fuera de una organización. El correo electrónico es una herramienta de las personas y organizaciones para enviar información confidencial, razón por la cual los sistemas de correo electrónico deben proporcionar mecanismos para impedir que los datos sean modificados, además garantizar que los mensajes no vayan a ser interceptados ni leídos por ninguna otra persona que no sea el destinatario al que van dirigidos. Los servicios de correo electrónico incluyen el servicio Protocolo de oficina de correo 3 (POP3) y el servicio Protocolo simple de transferencia de correo (SMTP) que recuperan y transfieren correo electrónico, respectivamente.

<sup>12</sup> Fuente: <http://equipo1-262m.blogspot.com/2009/03/tipos-de-servidores.html>

- **CONTROLADOR DE DOMINIOS MEDIANTE ACTIVE DIRECTORY:** Active Directory almacena información acerca de objetos en la red y hace que esta información sea fácil de encontrar por los administradores y usuarios, lo que proporciona una organización lógica y jerárquica de información en el directorio.
- **SERVIDOR DNS:** El Domain Name System es un servicio estándar de Internet y TCP/IP, permite a las computadoras cliente, colocar en su red y resolver nombres de dominio DNS. Un servidor DNS es una computadora configurada para proporcionar servicios del DNS en una red, lo que es necesario para poner en funcionamiento Active Directory.

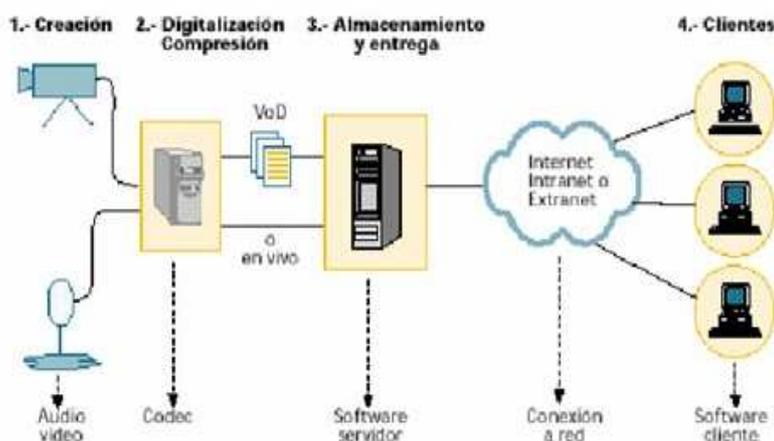


**Figura 1.11. Servidor DNS.**<sup>13</sup>

- **SERVIDOR DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL):** Es un estándar IP para simplificar la administración de la configuración del IP del cliente. El servidor DHCP reduce la complejidad y el trabajo administrativo usando configuración automática de TCP/IP y otros parámetros IP para clientes DHCP en su red.
- **SERVIDOR WINS:** Los servidores de Servicios de Nombres Internet de Windows (WINS) asignan dinámicamente direcciones IP a nombres de equipo, lo que permite a los usuarios tener acceso a los recursos a través del nombre del equipo en lugar de su dirección IP. Es el método más habitual para resolver nombres NetBIOS remotos y locales.
- **SERVIDOR DE STREAMING DE VIDEO:** Uno de los servicios de gran demanda en la actualidad es la transmisión de audio y video sobre redes. Engloba una serie de productos y técnicas con el objetivo de difundir contenidos multimedia tales como audio y video en la que el cliente puede

<sup>13</sup> Fuente: <http://equipo1-262m.blogspot.com/2009/03/tipos-de-servidores.html>

reproducir la información mientras la está recibiendo, sin necesidad de esperar la descarga completa del archivo, contempla la distribución de contenidos tanto en una Intranet corporativa como en el Internet. Los contenidos pueden estar almacenados previamente en un servidor para transmitirlos bajo demanda o crearse en el mismo momento de la transmisión en directo. Las partes de un sistema completo de streaming son: fuente de codificación, servidor y cliente.



**Figura 1.12 Servidor Streaming de Video<sup>14</sup>**

- **SERVIDOR DE REDES PRIVADAS VIRTUALES:** En Windows Server 2003 consiste en un servidor VPN, un cliente VPN, una conexión VPN y el túnel, los túneles se realizan a través de uno de los protocolos de túnel (PPTP o L2TP) que se incluyen con los servidores Windows Server 2003 y se instalan con el servicio Enrutamiento y acceso remoto, es decir, permite la construcción de conectividad entre sitios remotos mediante el uso de redes privadas virtuales e Internet.

Windows Server 2003 inicia algunos procesos durante el arranque del sistema, denominadas servicios. Un servicio es una aplicación basada en software que se ejecutan en segundo plano y que llevan a cabo una función específica del sistema

<sup>14</sup> Fuente: <http://www.rediris.es/difusion/publicaciones/boletin/58-59/ponencia10.html>.

y que proporciona una API para ser llamada por otros procesos además de las funcionalidades necesarias para otros equipos en la red.

#### 1.2.3.1.2 ARQUITECTURA DE SERVICIOS.

Los principales componentes del núcleo de la arquitectura de servicios son: service control manager, service control programs y service applications.

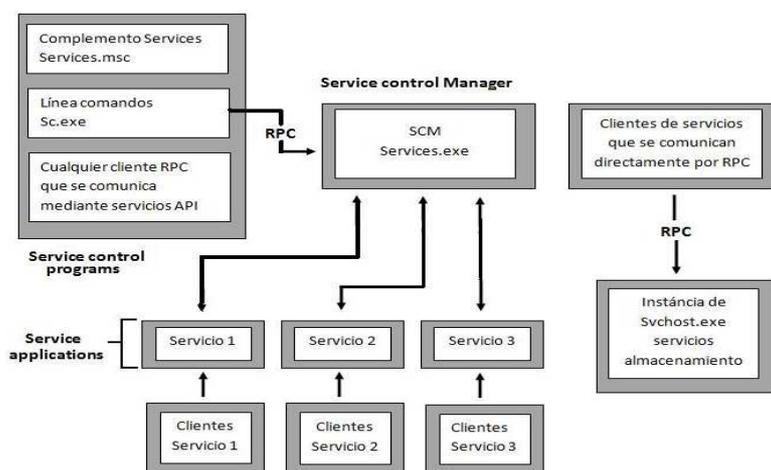
- **SERVICE CONTROL MANAGER (SCM):** Es un proceso que ejecuta service.exe, responsable del inicio, parada e interacción con los servicios. Los servicios son aplicaciones de Win32 que llaman a funciones Win32 especiales para interactuar con SCM para llevar a cabo acciones como registrar el inicio del sistema correcto, responder a solicitudes del estado, pausar o detener el servicio.
- **SERVICE CONTROL PROGRAMS:** Son aplicaciones Win32 estándar que usan las API's SCM Create Service, Open Service, Start Service, Control Service, Query Service Status y Delete Service para comunicarse o controlar servicios.

Para el uso de estas funciones Service Control Programs abre un canal de comunicaciones hacia SCM y al mismo tiempo especifica los tipos de acciones que quiere llevar a cabo. Durante su inicio, SCM crea un objeto interno que representa la base de datos de SCM y usa funciones de seguridad de Windows para proteger el objeto con un descriptor de seguridad, el cual especifica las cuentas que pueden abrir el objeto y los permisos de acceso a aquellas cuentas que pueden usarlo. SCM almacena el descriptor de seguridad en la subclave del registro del servicio como el valor de seguridad y lo lee durante el inicio de la configuración de seguridad y en el reinicio del equipo. SCM también debe decir que acceso quiere para un servicio.

- **SERVICE APPLICATIONS:** Contiene la infraestructura necesaria para la comunicación con SCM que envía comandos al servicio diciéndole que pare, pause, continúe o se apague. Un servicio llama también a funciones

especiales que comunican su estado a SCM. Los service applications como Service Web, consisten en al menos una aplicación que se ejecuta como un servicio. Un usuario que quiere iniciar, detener o configurar un servicio usa un service control program. Aunque Windows Server 2003 proporciona programas de control de servicios integrado, algunas service applications incluyen su propio programa de control de servicios que permite a los administradores especificar una configuración particular del servicio que administran. Los servicios se integran como programas de consola.

- **CLIENTES Y SERVIDORES RCP:** Para service application que usan las API's como Query Service Status y Open Service el cliente RCP comunica directamente con SCM ya que este es el servidor para éstas API's.



**Figura 1.13 Clientes y Servidores RCP.<sup>15</sup>**

### 1.2.3.1.3 VENTAJAS.

Las principales ventajas de Windows Server 2003 son:

- Facilidad de Implementar, Administrar y usar.
- Infraestructura segura.
- Confiabilidad, disponibilidad, escalabilidad y rendimiento de nivel empresarial.
- Menor costo total de propiedad gracias a la consolidación de tecnología más moderna.
- Creación fácil de sitios Web de Internet e Intranet dinámicos.

<sup>15</sup> Fuente: <http://geeks.ms/blogs/juansa/archive/2008/10/12/servicios-del-sistema-en-windows-server-2003.aspx>.

- Desarrollo rápido con el servidor de aplicaciones integrado
- Servicios Web XML fáciles de encontrar, compartir y reutilizar.
- Herramientas de administración sólidas.
- Potenciación de los empleados disminuyendo los costes de soporte técnico.
- La experiencia de una red internacional de asociados y profesionales certificados.

### 1.2.3.2 MODELO APLICACIÓN CLIENTE/SERVIDOR.

En este modelo, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.



**Figura 1.14 Modelo Cliente – Servidor.**<sup>16</sup>

Usualmente la mayoría del trabajo pesado lo hace el servidor y el o los procesos cliente sólo se ocupan de a interacción con el usuario, aunque esto puede variar.

Los principales componentes de este esquema son: los clientes, los servidores y la infraestructura de comunicaciones.

#### 1.2.3.2.1 CLIENTE.

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor.

Las características de un cliente son:

<sup>16</sup> Fuente: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf)

- Presenta el interfaz de usuario, GUI normalmente.
- Solicitudes al servidor y formularios de consulta o comandos en una estructura predefinida hacia el servidor basados en entradas del usuario.
- Se comunica con el servidor a través de RCP, para la transmisión de solicitudes o comandos hacia el servidor.
- Lleva a cabo análisis de datos sobre los resultados enviados por el servidor y muestra los resultados al usuario en un formato de aplicación definido.

Como ejemplo de clientes se puede citar interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para acceso a servidores remotos como servidores de SQL o aplicaciones que solicitan acceso a servidores para algunos servicios.

Por lo tanto, las funciones que lleva a cabo el proceso cliente son:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

#### **1.2.3.2.2 SERVIDOR.**

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Normalmente maneja todas las funciones relacionadas con la mayoría de las reglas de negocio y los recursos de datos.

Las características de un servidor son:

- Trabaja para el cliente y sus servicios están determinados por los objetivos empresariales del sistema cliente/servidor.
- Responder a consultas o comandos desde los clientes.

- Actúa principalmente como un repositorio de datos o conocimientos o como un proveedor de servicios.
- Oculta la composición del sistema cliente/servidor para el cliente y el usuario. La comunicación entre un cliente y un servidor puede ser totalmente ajeno a la plataforma de hardware/software del servidor, lo que permite al servidor cambiar o escalar según la demanda, sin afectar al cliente.

Como ejemplo de servidores pueden citarse servidores de ventana como X-Windows, servidores de archivos como NFS, servidores para el manejo de bases de datos como los servidores SQL, servidores de diseño y manufactura asistido por computador, etc.

Por lo tanto las funciones que lleva a cabo el proceso servidor son:

- Aceptar los requerimientos de base de datos que hacen los clientes.
- Procesar requerimientos de bases de datos
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de base de datos.

#### **1.2.3.2.3 INFRAESTRUCTURA DE COMUNICACIONES.**

Las características de la infraestructura de comunicaciones son:

- Sirve para que los clientes y servidores puedan comunicarse.
- Proporciona los mecanismos básicos de direccionamiento y transporte
- La mayoría de los sistemas clientes/servidor actuales se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones.
- La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

#### **1.2.3.2.4 CARACTERÍSTICAS DE LA ARQUITECTURA CLIENTE/SERVIDOR**

Sus características básicas son:

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema, mientras el proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos como velocidad del procesador, memoria, velocidad y capacidades del disco y input-output devices.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, reclusando su acceso a recursos compartidos.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- Posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

#### **1.2.3.2.5 VENTAJAS Y DESVENTAJAS DE ESTE ESQUEMA.**

Sus principales ventajas son:

- Recursos centralizados: Esto debido a que el servidor es el centro de la red, puede administrar los recursos que son comunes a todos los usuarios, por ejemplo: una base de datos centralizada se utilizaría para evitar problemas provocados por datos contradictorios o redundantes.
- Seguridad mejorada: Ya que la cantidad de puntos de entrada que permite el acceso a los datos no es importante.
- Administración a nivel del servidor: Ya que los clientes no juegan un papel importante en este modelo, requieren menos administración.
- Red escalable: Gracias a esta arquitectura, es posible quitar o agregar clientes sin afectar el funcionamiento de la red y sin la necesidad de realizar mayores modificaciones.
- Se puede usar componentes de hardware y software de varios fabricantes lo que contribuye a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes como servidores SQL o las herramientas de más bajo nivel como los sockets o el RPC.
- Reduce los costos de producción de software y disminuye los tiempos requeridos
- Contribuye a la disminución de costos de entrenamiento de personal pues favorece la construcción de interfaces gráficas interactivas, las cuales son más intuitivas y fáciles de usar por el usuario final.

Sus principales desventajas son:

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y software de distintos proveedores, lo que dificulta el diagnóstico de fallas.
- Tanto clientes como servidores debe utilizar el mismo mecanismo, por ejemplo sockets o RPC. Es decir, tener mecanismos generales que existan en diferentes plataformas.
- Hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.

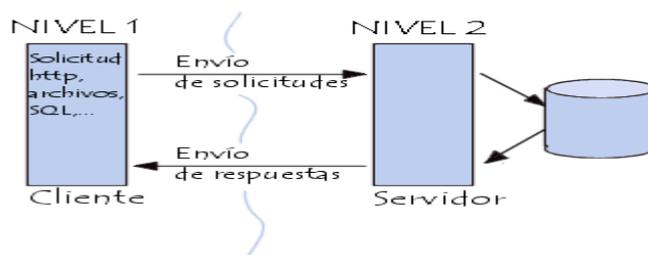
### 1.2.3.3 ARQUITECTURA DE CAPAS EN SISTEMAS DE INFORMACIÓN.

El objetivo primordial de la arquitectura en capas es la separación de la lógica de negocios de la lógica de diseño. Su desarrollo se puede llevar a cabo en varios niveles, y en caso de algún cambio, solo se ataca el nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o programación por capas, en el que a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

#### 1.2.3.3.1 ARQUITECTURA EN 2 CAPAS.

Describe los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio. Es decir se basa en Nivel de aplicación y Nivel de la base de datos.



**Figura 1.15 Arquitectura en 2 niveles.<sup>17</sup>**

Las desventajas de dos niveles son las siguientes:

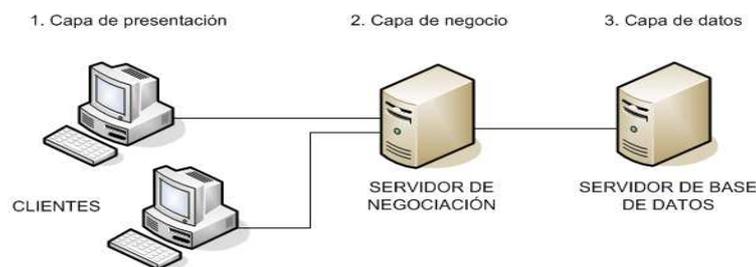
- El nivel de las aplicaciones se recargan, entremezclando aspectos típicos del manejo de la interfaz con las reglas del negocio.

<sup>17</sup> Fuente: <http://es.kioskea.net/contents/cs/cs3tier.php3>

- Las reglas del negocio quedan dispersas entre el nivel de aplicación y los stored procedures de la base de datos.
- La aplicación queda sobrecargada de información de bajo nivel si hay que extraer los datos de varias bases de datos, posiblemente con estructuras diferentes.
- El nivel de aplicación puede ser demasiado pesado para el cliente.

### 1.2.3.3.2 ARQUITECTURA EN 3 CAPAS

El diseño más utilizado actualmente es el diseño en tres capas.



**Figura 1.16 Arquitectura en 3 capas.<sup>18</sup>**

- **CAPA DE PRESENTACION:** o Capa de usuario, es la que ve el usuario, le comunica la información y captura la información del usuario en un mínimo proceso realizando un filtrado previo para comprobar que no hay errores de formato. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable, es decir entendible y fácil de usar para los usuarios.

Por lo tanto, esta capa es responsable de:

- Obtener información del usuario
- Enviar información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocios.
- Presentar estos resultados al usuario.

<sup>18</sup> Fuente: [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_tres\\_niveles](http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles)

- **CAPA DE NEGOCIO:** Es donde residen los programas que se ejecutan, se reciben las peticiones de los usuarios y se envían las respuestas tras el proceso. Se denomina capa de negocio o lógica de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

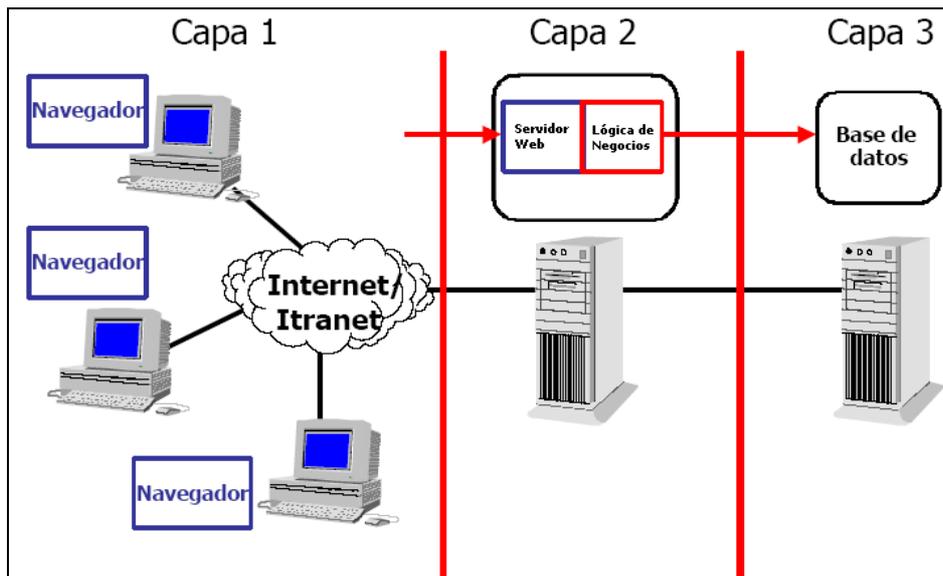
Por lo tanto, esta capa es responsable de:

- Recibir la entrada del nivel de presentación.
  - Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar.
  - Enviar el resultado procesado al nivel de presentación.
- 
- **CAPA DE DATOS:** Es donde residen los datos y es la encargada de acceder a los mismos. Esta formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Por lo tanto, esta capa es responsable de:

- Almacenar los datos.
- Recuperar los datos
- Mantener los datos
- La integridad de los datos.

La primera capa no solo podría ser de escritorio, sino también Web, pues solo bastaría con cambiar la capa de presentación en donde nuestras páginas Web estarían alojadas en un servidor Web las mismas que se conectan a la lógica de negocios y de allí a los servicios de datos.



**Figura 1.17 Arquitectura de 3 capas con interfaz Web.<sup>19</sup>**

Todas estas capas pueden residir en un único ordenador, pero lo usual es que haya una multitud de ordenadores en donde reside la capa de presentación, es decir los clientes de la arquitectura cliente/servidor. Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconsejan se pueden separar en dos o mas ordenadores.

En esta arquitectura, los términos capas y niveles no significan lo mismo ni son similares.

- **CAPA:** Hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/Lógica de Negocios/Datos.
- **NIVEL:** Corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física

Por ejemplo si:

- Una solución de tres capas (Presentación, Lógica de Negocios, Datos), reside en un solo ordenador (presentación+lógica+datos), se dice que la solución es de tres capas y un nivel.

<sup>19</sup> Fuente: <http://faustol.files.wordpress.com/2007/09/construccion-de-aplicaciones-en-capas.doc>

- Una solución de tres capas (Presentación, Lógica de Negocios, Datos), reside en dos ordenadores (presentación+lógica, lógica+datos), se dice que la solución es de tres capas y dos niveles.
- Una solución de tres capas (Presentación, Lógica de Negocios, Datos), reside en tres ordenadores (presentación, lógica, datos), se dice que la solución es de tres capas y tres niveles.

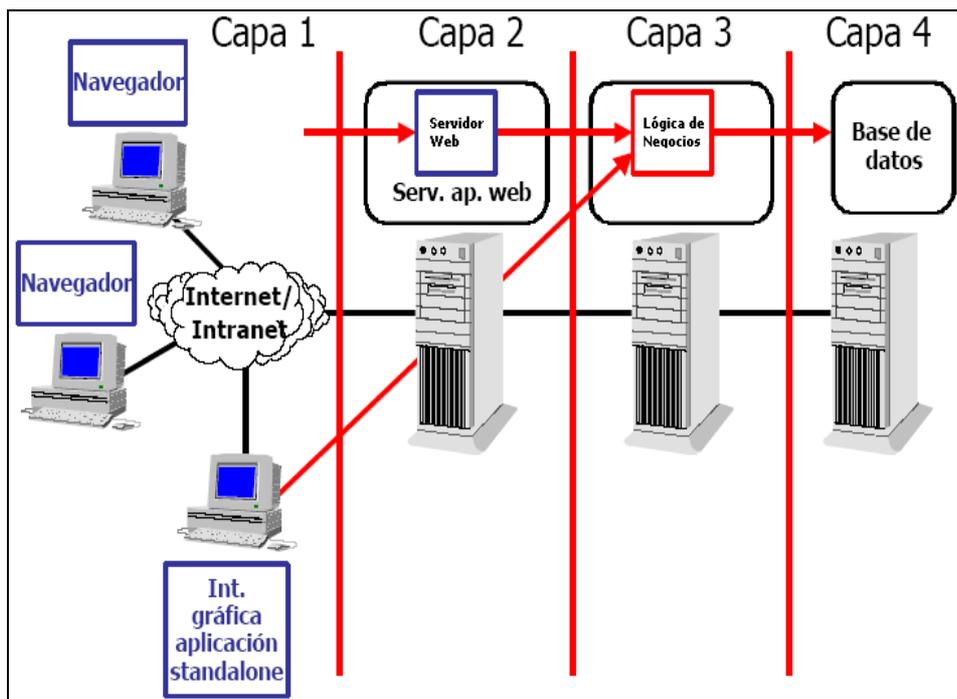
En esta arquitectura todas las peticiones de los clientes se controlan en la capa correspondiente a la lógica de negocio. Esto es muy importante, ya que quiere decir que:

- El cliente no tiene que tener drivers ODBC ni la problemática consiguiente de instalación de los drivers por lo tanto se reduce el costo de mantener las aplicaciones cliente.
- El cliente y el gestor de reglas de negocio tienen que hablar el mismo lenguaje
- El gestor de reglas de negocio y el servidor de datos tienen que hablar el mismo lenguaje.

#### **1.2.3.3.3 ARQUITECTURA EN N CAPAS.**

Se puede ir separando las aplicaciones en más niveles lógicos, por ejemplo, se podría querer que nuestra aplicación tenga múltiples interfaces, es decir interfaz gráfica (standalone o desktop) y también interfaz Web. En esta circunstancia es aconsejable separar al Servidor Web encargado de alojar las páginas Web en una capa más. Por lo tanto se tendrían 4 capas.

Mientras más servicios coloquemos a nuestra aplicación y mientras más escalable lo imaginemos, más capas lógicas van a irse añadiendo a nuestra arquitectura.



**Figura 1.18 Arquitectura de 4 capas.<sup>20</sup>**

#### 1.2.3.4 SERVIDORES WEB.

Un servidor es un tipo de software que realiza tareas en nombre del usuario, es también el ordenador físico en el cual funciona este software, que provee datos de modo que otras máquinas puedan utilizar estos datos.

Un servidor Web es un programa que sirve para atender y responder a las peticiones de los navegadores, proporcionando los recursos que soliciten utilizando el protocolo HTTP o HTTPS. Funciona de la siguiente manera:

- Espera peticiones en el puerto TCP indicado
- Recibe una petición.
- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve a recibir una petición y se repite el proceso.

<sup>20</sup> Fuente: <http://faustol.files.wordpress.com/2007/09/construccion-de-aplicaciones-en-capas.doc>

Este bucle se repite de una forma infinita y proporciona los requisitos básicos de los servidores. En base a este diseño se ha desarrollado todos los servidores http, variando solo el tipo de peticiones que pueden atender.

#### 1.2.3.4.1 CARACTERÍSTICAS BÁSICAS DE LOS SERVIDORES WEB.

- **SERVICIO DE FICHEROS ESTÁTICOS:** Todo servidor Web debe incluir al menos la capacidad para servir los ficheros estáticos que se hallen en alguna parte del disco.
- **SEGURIDAD Y AUTENTICACIÓN:** Los servidores actuales permiten controlar desde el programa servidor los aspectos relacionados con la seguridad y la autenticación de los usuarios. Permiten especificar reglas de servicio de directorios, subdirectorios y ficheros en la configuración del programa servidor Web, indicando que usuarios, máquinas, etc., tienen acceso al recurso indicado. En la autenticación los servidores permiten al menos un fichero con nombres de usuario y contraseñas mediante el cual se puede validar los datos enviados desde el cliente, cabe mencionar que en este aspecto las prestaciones ofrecidas por los diferentes servidores Web son diversas.
- **CONTENIDO DINÁMICO:** Es un aspecto fundamental ya que la mayor parte del contenido Web se genera de forma dinámica. La mayor parte de servidores Web ofrecen soporte para CGI, el método más sencillo para generar contenido dinámico. Otros ofrecen soporte para lenguajes de programación como PHP, JSP, ASP, etc.
- **SERVIDORES VIRTUALES:** Es la capacidad de algunos servidores Web de facilitar múltiples dominios con una única dirección IP, discriminando entre los diferentes dominios alojados en función del nombre de dominio enviado en la cabecera http.
- **PRESTACIÓN EXTERNA:** Son característica adicionales de servidores Web que entre otros ofrecen:
  - SPELLING:** Que define una página de error que sirve cuando el servidor no ha encontrado el recurso solicitado.

**MOD\_MP3:** Permite convertir el servidor Web en un servidor eficiente de música.

**SQL TAGS:** Añade al HTML extendido ciertos comandos para acceder a bases de datos SQL desde las páginas HTML.

**KILL FRAME:** Envía con cada página Web un código que evita que la Web quede enmarcada dentro de otra página Web.

- **ACTUACIÓN COMO REPRESENTANTES:** Algunos servidores permiten su uso como servidores intermedios o Proxy servers, que pueden ser utilizados para diferentes propósitos, como aceleradores de navegación por ejemplo.

#### 1.2.3.4.2 SERVIDOR INTERNET INFORMATION SERVER (IIS).

El IIS es un conjunto de servicios que funcionan con Windows el cual proporciona capacidades de servidor Web integrado, confiable, escalable y administrable en Internet, una Intranet o una extranet. IIS 6.0 incorpora mejoras significativas en la arquitectura para cubrir las necesidades de los clientes alrededor del mundo. Sus módulos tienen la capacidad de procesar distintos tipos de páginas tales como ASP y ASP.NET, PHP o Perl.

Al instalar IIS se crea por defecto:

- Un sitio Web
- Un sitio Web de administración.
- Un servidor virtual SMTP
- Inetpub, el directorio donde residen subdirectorios para cada servicio instalado.

Además se puede instalar o remover otros componentes como el servicio de noticias NNTP, FTP.

Al realizar una instalación completa se configuran los siguientes puertos sobre todas las IP instaladas:

- Puerto 80: Para el servicio http.
- Puerto 21: Para el servicio ftp.
- Puerto 119: Para el servicio nntp.
- Puerto 25: Para el servicio SMTP.

El sitio de administración utiliza un puerto aleatorio respondiendo solo a la IP 127.0.0.1 o localhost.

## **VENTAJAS.**

IIS 6.0 junto con Windows Server 2003 introducen muchas características nuevas, para la administración, disponibilidad, confiabilidad, seguridad, rendimiento y escalabilidad de los servidores de aplicaciones Web, también mejora el desarrollo de aplicaciones Web y la compatibilidad internacional. A continuación se detallan sus principales características:

- **Confiable y Escalable:** Ofrece un entorno más inteligente y confiable
- **Seguro y Administrable:** Las mejoras de seguridad incluyen cambios tecnológicos y de procesamiento de solicitudes, ha mejorado la autenticación y la autorización.
- **Desarrollo y compatibilidad internacional mejorados:** Un único entorno de alojamiento de aplicaciones integrado, con compatibilidad total con características avanzadas y con la caché en modo de núcleo. Ofrece compatibilidad internacional y con los estándares Web más recientes.
- **Infraestructura de gran fiabilidad.**
- **Capacidad de manejo y escalabilidad para aplicaciones Web sobre todas las versiones de Windows Server 2003.**
- **Aumenta la disponibilidad de sus sitios y aplicaciones Web a las organizaciones.**
- **Reduce costos administrativos**
- **Soporte de ASP.NET.**

#### 1.2.3.4.3 INSTALACIÓN Y CONFIGURACIÓN DE UN SITIO WEB.

IIS ofrece muchas facilidades en la instalación y configuración de un sitio Web ya que ofrece un amigable asistente de instalación y configuración. Por este motivo este tema se centra en aspectos importantes que se deben tomar en cuenta al momento de instalar y configurar un sitio Web. Esto es, tener muy en cuenta las siguientes preguntas:

- ¿Qué puertos se utilizarán para los sitios Web y cuáles para comunicaciones seguras?
- ¿Sobre que dirección IP el servicio estará escuchando?
- ¿A qué nombre responderá el servidor Web?
- ¿Funcionarán múltiples sitios Web sobre el mismo servidor?
- ¿En que directorio se ubicarán los documentos del sitio?

##### 1.2.3.4.3.1 CONFIGURACIÓN DE SITIOS WEB.

Esta configuración se basa de acuerdo al siguiente planteamiento: Windows Server 2003, IIS 6.0, Sitio Web .NET 2.0.

- **RECOMENDACIONES INICIALES:** Una de las causas más comunes de error es la instalación de componentes (IIS, Framework .NET, etc.) en el orden incorrecto. Primero se debe tener correctamente instalado IIS y después el Framework, caso contrario se tendrá que reparar/reinstalar el Framework.net.
- **REQUISITOS PREVIOS:** IIS debe tener las siguientes opciones dentro del Servidor de aplicaciones:
  - ASP.NET
  - Consola de servidor de aplicaciones
  - Habilitar el acceso de red COM+.
    - Instalar Internet Information Server (IIS)
    - Administrador de Servicios de Internet Information Services
    - Archivos comunes
    - Servicio de protocolo de transferencia de archivos (FTP)
    - Servicio World Wide Web.

- **DIRECTORIOS Y PERMISOS:** Si coloca el sitio fuera del directorio por defecto (C:\inetpub\wwwroot) tendrá que darle al directorio del sitio o mejor, a la carpeta que vaya a contener todos los sitios o proyectos, los mismos permisos que tenga C:\inetpub\wwwroot, ya que puede que al mover el directorio del sitio desde otro lugar dichos permisos no se hereden automáticamente, en este caso habría que desmarcar y marcar la opción de heredar permisos y comprobar que se actualicen.

### **1.2.3.5 SERVIDOR DE BASE DE DATOS**

Un servidor de base de datos ofrece servicios de base de datos a otros programas o computadoras. Se crea debido a la necesidad de las empresas de manejar grandes cantidades de datos y compartirlas de manera segura con diversos clientes ya sean usuarios o aplicaciones.

#### **1.2.3.5.1 SERVIDOR DE BASE DE DATOS RELACIONALES.**

Un sistema de base de datos relaciones posee una arquitectura cliente/servidor y proporciona sistemas de gestión, administración y protección de información a través de conexiones de red y protocolos definidos, en los que los usuarios acceden de manera concurrente a través de aplicaciones clientes.

##### **1.2.3.5.1.1 CARACTERÍSTICAS DEL SERVIDOR DE BASE DE DATOS RELACIONALES.**

- Estos servidores manejan grandes volúmenes de información en forma estable, fiable, coherente y segura.
- La información se almacena de modo lógico de una manera relacional, es decir tablas que se relacionan entre si a través de un conjunto definido de claves.
- El sistema y diseño de base de datos debe mostrar la información a través de conjuntos de datos planos, independizando las relaciones y la arquitectura de la necesidad de información del usuario.

- Además cuenta con mecanismos de control de transacciones basados en disparadores, reglas de definición y validación de entradas de datos.
- Cuenta con sistemas de indexación capaces de ordenar y acelerar las consultas a la información requerida.
- También proporcionan la posibilidad de automatizar operaciones de acceso, filtrado y control de los datos, a través de procedimientos almacenados.
- El estándar de diseño, desarrollo y consultas reinformación se lo realiza a través del lenguaje SQL.

#### **1.2.3.5.1.2 SEGURIDAD**

La seguridad debe garantizar que ningún intruso pueda acceder o corromper la integridad del sistema, sobretodo por el tipo de información que se maneja y que es crítica para la empresa como son: datos de contabilidad, gestión, personal o estratégicos de la cual depende para su existencia. Los niveles de seguridad básicos de un servidor de base de datos son:

##### **a) SEGURIDAD DE ACCESO AL SISTEMA**

Se lo implementa de dos maneras:

- **A NIVEL DE SISTEMA OPERATIVO:** En el cual el Sistema de Gestión de base de Datos se apoya en la seguridad de entrada al sistema operativo para comprobar la validez del acceso a los datos almacenados.
- **MODO MIXTO:** Los sistemas tiene la capacidad de incluir como cuentas de acceso o login aquellas propias del sistema operativo, lo que facilita la transición de las cuentas de seguridad. Esta forma es útil cuando los clientes que acceden al sistema provienen de sistemas operativos con poca o ninguna seguridad o de aplicaciones instaladas que necesiten acceder a los volúmenes de información del sistema.

Ambos casos cuentan con roles para el usuario al entrar al sistema para la realización de determinadas operaciones de cara al sistema.

**b) SEGURIDAD A NIVEL DE OBJETOS DE DATOS.**

Es el detalle del acceso a nivel de creación y administración de objetos de datos, es decir, tablas, vistas, índices, relaciones, reglas, etc. Hablamos de las responsabilidades y acciones que puede hacer el usuario en el esquema de base de datos, es decir el sistema definirá cómo se debe almacenar y relacionar la información. Se establece una política de delegación de responsabilidades, ya que se pueden especificar de nuevo roles a los usuarios, indicando quien podrá crear, modificar o eliminar cualquier objeto de datos.

**c) SEGURIDAD A NIVEL DE DATOS.**

Este nivel de seguridad esta en la capa de información, en la que se indica quién puede acceder a qué información, para consulta, actualización, inserción o borrado, dependiendo del motor que se utilice se determinará hasta que grado de seguridad se llega, desde la protección de las columnas de una tabla, hasta la tabla en si, creación de vistas, etc.

**d) SEGURIDAD EN CUANTO A PROTECCIÓN DE LOS ALMACENAMIENTOS FÍSICOS DE LOS DATOS.**

Este aspecto contempla la seguridad a nivel de sistema operativo de los archivos de datos del sistema y las políticas de copia de seguridad y restauración de los datos, tanto con herramientas del sistema operativo como las proporcionadas por el propio servidor de datos, además de los soportes de hardware compatibles de almacenamiento masivo empleados como destino de las copias.

**1.2.3.5.1.3 SOPORTE DE RED**

Al implementarse una solución cliente/servidor, es fundamental tener un canal apropiado para la comunicación en la conexión entre los distintos clientes y el servidor, que posibilite el intercambio de información. De la comunicación

dependerán parámetros como la velocidad de acceso a los datos. El protocolo de mayor auge es TCP/IP que garantiza que la conexión de los servidores estará al alcance de cualquier usuario desde cualquier parte del mundo.

Otro aspecto importante a tomar en cuenta es como es transmitida la información, ya que tiene que existir posibilidades de encriptación para prevenir accesos no autorizados así como mecanismos de partición de los datos, para evitar que peticiones masivas de información sobrecarguen el ancho de banda de la red. El servidor debe ser capaz de ir previendo la información requerida en el momento justo en el que es necesaria, lo que ahorra recursos de la máquina y ancho de banda.

#### **1.2.3.5.1.4 HERRAMIENTAS DE ADMINISTRACION.**

Este aspecto se refiere a las herramientas que proporciona un servidor de base de datos tanto al usuario administrador como al cliente consumidor de los datos, las mismas que deben tener un entorno amigable y sencillo para poder realizar sus tareas de manera más rápida y simplificada. Además con un nivel de automatización de tareas. También deben permitir la administración remota del servidor, lo que garantiza un alto grado de seguridad, optimización, ahorro del tiempo y esfuerzo.

## **1.2.4 MODELO PROTOTIPADO.**

### **1.2.4.1 DEFINICION DE MODELO PROTOTIPADO.**

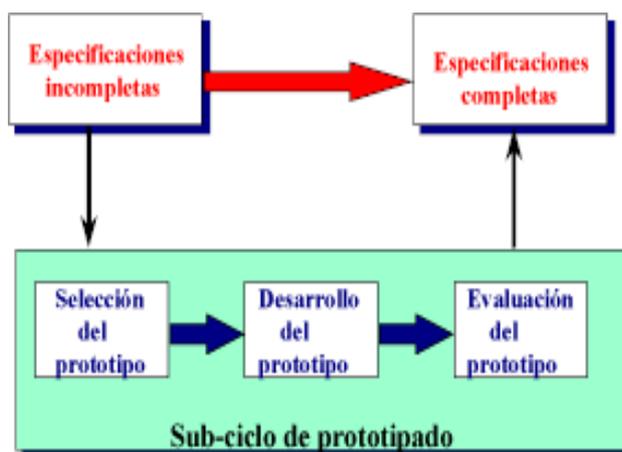
Un prototipo es una representación fácilmente ampliable y modificable de un sistema planificado, incluyendo su interfaz y funcionalidad de entradas y salidas. Esto es, una representación limitada del diseño, que permite al usuario interactuar con él y explorar sus posibilidades. Es decir consiste en modelar el producto final y testear haciendo uso del modelo.

El modelo prototipado define objetivos globales para el software, identifica requisitos conocidos y áreas del esquema en que se necesita mayor definición; Es decir se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado en forma de un diseño rápido, esto es una representación de aquellos aspectos del software que serán visibles para el usuario final, por ejemplo la configuración de la interfaz con el usuario, el mismo que sirve para la construcción de un prototipo el cual es evaluado por el usuario final para una retroalimentación, permitiendo redefinir requisitos del software que se desarrollará, lo que permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

El propósito del prototipado es establecer un conjunto de requerimientos formales que pueden ser traducidos en la producción de programas mediante el uso de métodos y técnicas de programación. Para que el modelo prototipado sea efectivo debe contar con las siguientes características:

- Debe ser un sistema con el que se pueda experimentar.
- Debe ser comparativamente barato.
- Debe desarrollarse rápidamente.
- Énfasis en la interfaz de usuario.
- Equipo de desarrollo reducido
- Herramientas y lenguajes adecuados.

Se lo lleva a cabo elaborando un prototipo del producto final: que aspecto tendrá, como funcionará. Cuanto más próximo se encuentre el prototipo al producto real, mejor será la evaluación. Esta técnica puede ser desarrollada en cualquier etapa del desarrollo. A medida que el prototipo progresa y producto se completa, el prototipo ha de abarcar más y más características del producto final.



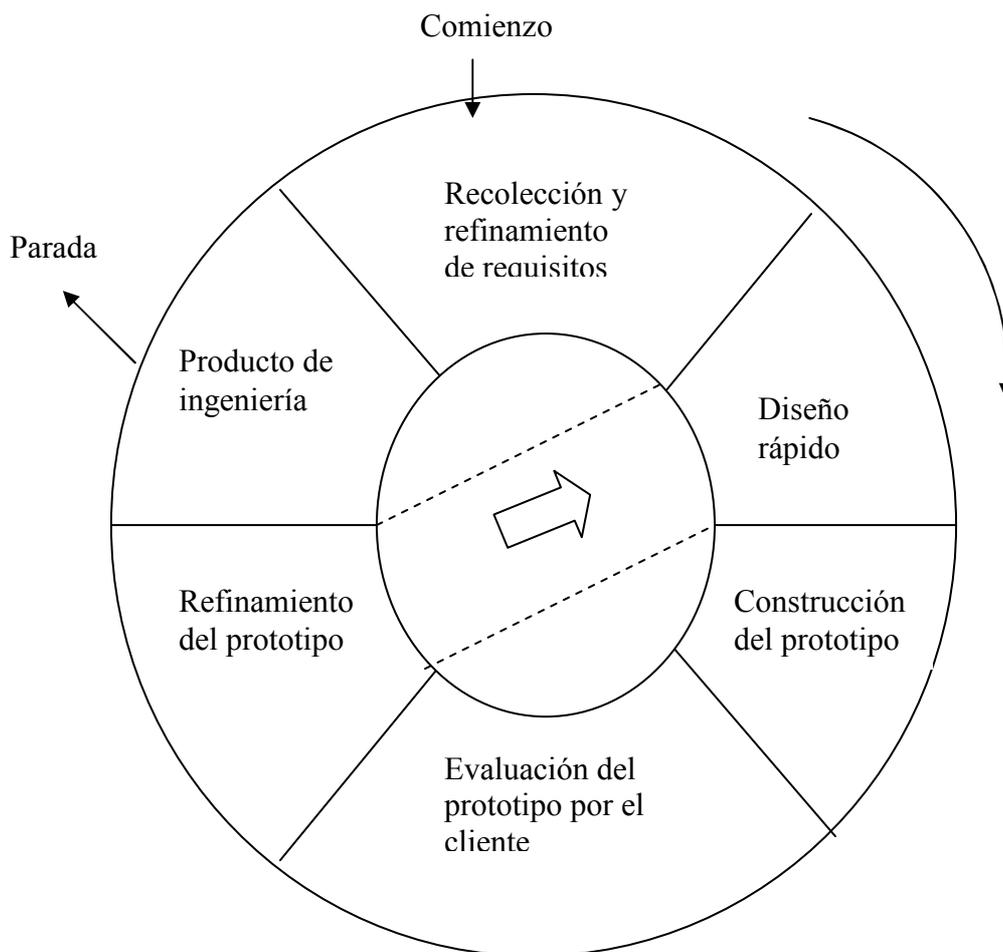
**Figura 1.19. Modelo Prototipado<sup>21</sup>**

#### 1.2.4.2 ESTRUCTURA DEL MODELO PROTOTIPADO

La estructura de un modelo prototipado muestra las siguientes fases:

- Recolección y refinamiento de requisitos
- Diseño rápido
- Construcción del prototipo
- Evaluación del prototipo por parte del cliente
- Refinamiento del prototipo
- Producto de Ingeniería.

<sup>21</sup> Fuente: <http://ingenieriadesoftwareudbciclo2008.blogspot.com/2008/08/ingenieria-de-software.html>



**Figura 1.20. Estructura del Modelo Prototipado<sup>22</sup>**

### **1.2.4.3 ETAPAS DEL MÉTODO DE PROTOTIPOS**

#### **1.2.4.3.1 IDENTIFICACIÓN DE REQUERIMIENTOS**

Antes de crear el prototipo es muy importante identificar los requerimientos conocidos que tiene que satisfacerse, estos determinan los fines para los que servirá el sistema y el alcance de sus capacidades.

#### **1.2.4.3.2 DESARROLLO DE UN MODELO DE TRABAJO.**

Consiste en el desarrollo de un plan general.

<sup>22</sup> Fuente: <http://cflores334.blogspot.es/1192848180/>

Primero los usuarios y analistas identifican de manera conjunta los datos que son necesarios para el sistema y especifican la salida que debe producir la aplicación. Presenta los siguientes componentes:

- El lenguaje para el dialogo del usuario y el sistema
- Pantallas y formato para la entrada de datos.
- Módulos esenciales de procesamiento.
- Salida del sistema.

En esta etapa es más importante la rapidez con que se construye el prototipo que la eficiencia de la operación.

#### **1.2.4.3.3 EL PROTOTIPO Y EL USUARIO.**

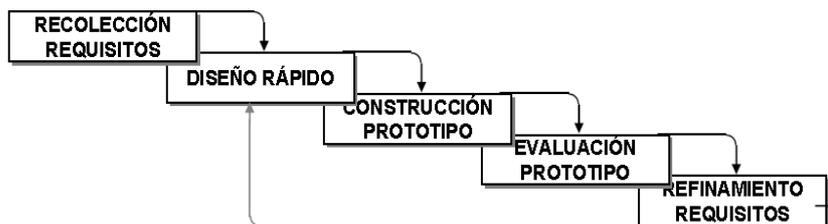
El usuario tiene la responsabilidad de trabajar con el prototipo y evaluar sus características y operación. La experiencia con el sistema bajo condiciones permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios así como la eliminación de características innecesarias.

#### **1.2.4.3.4 REVISIÓN DEL PROTOTIPO.**

Se establece lo que le agrada y lo que le desagrada al usuario. La información obtenida tendrá influencia sobre las características de la siguiente versión de la aplicación, estos cambios son planificados con el usuario antes de llevarlos a cabo por el analista.

#### **1.2.4.3.5 REPETICIÓN DEL PROCESO.**

Se repite el proceso hasta que los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias.



**Figura 1.21 Proceso del Modelo Prototipado<sup>23</sup>**

#### 1.2.4.4 CONSTRUCCIÓN DE PROTOTIPOS

Cabe recalcar que todo producto de software comienza por la petición de un cliente, ya sea en forma de descripción del problema, un informe que define un conjunto de objetivos del producto, una petición de propuesta formal, etc. De esta manera, la construcción de prototipos consta de las siguientes fases:

1. Se debe evaluar la petición del software y determinar si el programa a desarrollar es un candidato para construir el prototipo. Por lo tanto el cliente debe participar en la evaluación y refinamiento del prototipo
2. Desarrollar una representación abreviada de los requerimientos; es decir, representar los dominios funcionales y de información del programa, esto se lo puede realizar mediante métodos de análisis de requerimientos, después se crea un conjunto de especificaciones de diseño abreviadas para el prototipo. El diseño ocurre antes de la construcción del prototipo y se enfoca hacia la arquitectura a nivel superior y a aspectos de diseño de datos.
3. El software del prototipo se crea, prueba y refina.
4. Una vez probado el prototipo se presenta al cliente el cual conduce “la prueba de aplicación” y sugiere modificaciones. Este paso es muy importante ya que el cliente puede examinar una representación implementada de los requerimientos del programa y puede sugerir modificaciones que harán al programa cumplir mejor las necesidades reales

<sup>23</sup> Fuente: <http://cflores334.blogspot.es/1192848180/>

5. Se repiten los dos pasos anteriores hasta lograr que todos los requerimientos estén formalizados o hasta que el prototipo haya evolucionado a un sistema de producción.

#### **1.2.4.5 HERRAMIENTAS PARA EL PROTOTIPADO**

En el enfoque de prototipos se pretende que el modelo sea operante, es decir, una colección de programas de computadora que simulan algunas o todas las funciones que el usuario desea. Para lograrlo se utiliza las siguientes herramientas de software:

- Un diccionario de datos integrado.
- Un generador de pantallas.
- Un generador de reportes no guiado por procedimientos.
- Un lenguaje de programación de cuarta generación.
- Un lenguaje de consultas no guiado por procedimientos.
- Medios poderosos de administración de base de datos.

#### **1.2.4.6 VENTAJAS DEL MODELO PROTOTIPADO.**

- Es útil cuando el usuario final conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- El prototipado puede fortalecer la especificación de requerimientos.
- Los cambios son desplazados al prototipo
- Mejor soporte de modificaciones.
- Una vez identificados todos los requisitos mediante el prototipo, se construye el producto de ingeniería
- Reduce los costos y aumenta la probabilidad de éxito
- Reduce riesgos de construir productos que no satisfagan las necesidades de los usuarios.
- El prototipado mejora la relación entre usuarios y desarrolladores.

- Un prototipo puede servir de herramienta de marketing y como base para especificaciones operacionales.
- El prototipado puede reducir el tiempo requerido para el test si los problemas se detectan pronto, cuando son fáciles de localizar.
- El prototipado puede resultar en un ahorro inicial en costes de mantenimiento, porque los usuarios están haciendo un test de aceptación a lo largo de todo el proceso.
- Los sistemas generados por prototipado resultan más fáciles de aprender a manejar y usar.
- El entorno en el proceso de prototipado está menos sujeto a intereses creados por lo que está más abierto al cambio y a la innovación.
- El uso del prototipado durante el desarrollo reduce la cantidad de código hasta en un 60%.

#### **1.2.4.7 DESVENTAJAS DEL MODELO PROTOTIPADO.**

- El prototipado puede estimular un número de petición de cambios excesivos.
- El usuario no ve “el interior”
- Los prototipos operativos pueden inducir a pensar que el producto final está prácticamente dispuesto para su salida al mercado.
- Los clientes pueden no estar preparados para proporcionar la información con el nivel y frecuencia requeridos en un ciclo iterativo como también no estar dispuestos a participar en el ciclo de iteración durante largos periodos de tiempo.
- Los desarrolladores pueden encontrar dificultades en la ultimación de detalles que han resultado de fácil desarrollo con una herramienta de prototipado.
- Se corre el riesgo de ignorar importantes características del sistema (actuación, seguridad, robustez y fiabilidad, etc.) durante el desarrollo del prototipo.
- Puede resultar imposible implementar funciones críticas del sistema para la seguridad o la operación.

- Los prototipos de sistemas complejos pueden resultar excesivamente complicados.
- El prototipado se puede estancar si los miembros del equipo de diseño no disponen de autoridad para la toma de decisiones.

No hay garantías de que las posiciones de desarrolladores y clientes converjan durante el ciclo de iteración.

## **1.2.5 METODOLOGÍA UML.**

### **1.2.5.1 DEFINICION DE METODOLOGÍA UML.**

UML es un lenguaje elaborado por Grady Brooch, Ivar Jacobson y Jim Rumbaugh que permite modelar, construir y documentar los elementos que forman un sistema orientado a objetos. Además de convertirse en el estándar de facto de la industria, busca ser independiente del lenguaje de implementación, de tal forma que los diseños realizados con UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML. Hay que tomar en cuenta que UML no define un proceso de desarrollo específico, tan solo se trata de una notación.

Los principales factores que llevaron a la creación de UML fueron:

- La necesidad de modelar sistemas.
- Las tendencias en la industria del software.
- Unificar los distintos lenguajes y métodos existentes.
- Innovar los modelos para adaptarse a la arquitectura distribuida.

El UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clase, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

### **1.2.5.2 CARACTERÍSTICAS.**

- UML divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
- Permite describir un sistema en diferentes niveles de abstracción.
- Se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del desarrollo de una aplicación, sin definir un modelo de desarrollo.

### 1.2.5.3 DIAGRAMAS UML

Los diagramas representan las diferentes etapas del desarrollo del proyecto. Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones, es decir una vista del sistema a modelar.

#### 1.2.5.3.1 DIAGRAMA DE CLASES.

Describe la estructura del sistema. Las clases son abstracciones que especifican la estructura y el comportamiento común de un conjunto de objetos. Los objetos son instancias de las clases que se crean, modifican y destruyen durante la ejecución del sistema. Los diagramas de clase describen el sistema desde el punto de vista de objetos, clases, atributos, operaciones y sus asociaciones.

Modela la estructura estática de las clases en el sistema. Nos muestra una vista de la aplicación en un determinado momento. Muestra un conjunto de clases, interfaces y sus relaciones. Es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos.

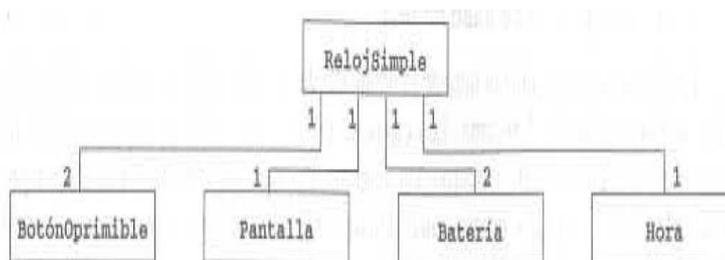


Figura 1.22 Diagrama de Clases reloj<sup>24</sup>

#### 1.2.5.3.2 DIAGRAMA DE COMPONENTES

Modela los componentes que forman una aplicación, sistema o empresa. Los diagramas de componentes contienen:

<sup>24</sup> Fuente: INGENIERÍA DE SOFTWARE ORIENTADO A OBJETOS. BRUEGGE, Bernd. DUTOIT, Allen. 2002.

- Componentes
- Interfaces.
- Relaciones de dependencia, generalización, asociaciones y realización.
- Paquetes o subsistemas.
- Instancias de algunas clases.

#### 1.2.5.3.2.1 USOS MÁS COMUNES.

**a) MODELADO DE CÓDIGO FUENTE:** Se puede utilizar para modelar la gestión de la configuración de los archivos de código fuente. Para modelar el código fuente de un sistema:

- Hay que identificar el conjunto de archivos de código fuente de interés y modelarlos como componentes estereotipados como archivos.
- Si el sistema es muy grande es necesario utilizar los paquetes para agrupar los archivos de código fuente.
- Es necesario identificar la versión del componente.

**b) MODELADO DE UNA VERSIÓN EJECUTABLE Y BIBLIOTECAS:** Se centra en la definición de todos los elementos que componen la versión ejecutable, es decir la documentación, los ficheros que se entregan, etc. Para modelar una versión ejecutable es preciso:

- Identificar el conjunto de componentes que se pretende modelar
- Identificar el estereotipo de cada componente del conjunto seleccionado.
- Para cada componente de este conjunto hay que considerar las relaciones con los vecinos.

**c) MODELADO DE UNA BASE DE DATOS FÍSICA:** Para modelar una base de datos física se necesita:

- Identificar las clases del modelo que representan el esquema lógico de la base de datos.

- Seleccionar una estrategia para hacer corresponder las clases con tablas. Así como la distribución física de la base de datos.
- Para poder visualizar, especificar, construir y documentar dicha correspondencia es necesario crear un diagrama de componentes que tenga componentes estereotipados como tablas.
- Donde sea posible es aconsejable utilizar herramientas que ayuden a transformar diseño lógico en físico.

### 1.2.5.3.3 DIAGRAMA DE DESPLIEGUE FÍSICO.

Muestra cómo y dónde se desplegará el sistema.

**a) MODELADO DE UN SISTEMA EMPOTRADO:** Facilita la comunicación entre los ingenieros de hardware y los de software, para lo cual es necesario:

- Identificar los dispositivos y nodos propios del sistema.
- Proporcionar señales visuales, sobre todo para los dispositivos poco usuales.
- Modelar las relaciones entre esos procesadores y dispositivos en un diagrama de despliegue.
- Si es necesario hay que detallar cualquier dispositivo inteligente, modelando su estructura en un diagrama de despliegue más pormenorizado.

**b) MODELADO DE UN SISTEMA CLIENTE SERVIDOR:** La división entre cliente y servidor en un sistema es complicada ya que implica tomar algunas decisiones sobre donde colocar físicamente sus componentes software, que cantidad de software debe residir en el cliente, etc. Para modelar un sistema cliente servidor es necesario:

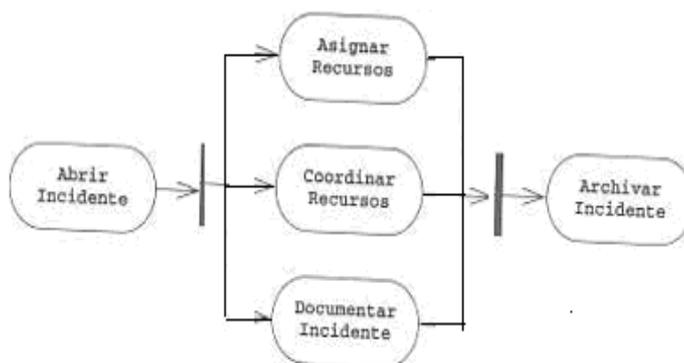
- Identificar los nodos que representan los procesadores cliente y servidor del sistema.
- Destacar los dispositivos relacionados con el comportamiento del sistema.
- Proporcionar señales visuales para esos procesadores y dispositivos a través de estereotipos.

- Modelar la tipología de esos nodos mediante un diagrama de despliegue.

#### 1.2.5.3.4 DIAGRAMA DE ACTIVIDADES.

Describe el sistema desde el punto de vista de actividades, las cuales son estados que representan la ejecución de un conjunto de operaciones. La terminación de estas operaciones dispara una transición a otra actividad....

Modela el comportamiento de los casos de uso, objetos, operaciones. Se utiliza para visualizar, especificar y construir la dinámica de un conjunto de objetos. Sirven fundamentalmente para modelar el flujo de control entre actividades. Gráficamente es un conjunto de arcos y nodos.



**Figura 1.23 Diagrama de Clases<sup>25</sup>**

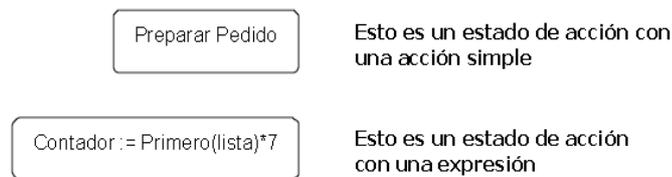
##### 1.2.5.3.4.1 CONTENIDO DEL DIAGRAMA DE ACTIVIDADES.

Básicamente contiene estados de actividad, acción, transiciones, objetos.

##### a) ESTADOS DE ACTIVIDAD Y ESTADOS DE ACCIÓN.

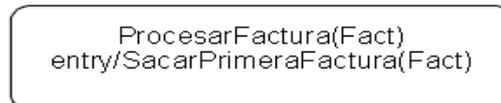
Ambos se representan en un rectángulo con las puntas redondeadas en cuyo interior se representa una actividad o una acción. Un estado que represente una acción es atómico, es decir que su ejecución se puede considerar instantánea y no puede ser interrumpida.

<sup>25</sup> Fuente: INGENIERÍA DE SOFTWARE ORIENTADO A OBJETOS. BRUEGGE, Bernd. DUTOIT, Allen. 2002.



**Figura 1.24 Estado de acción<sup>26</sup>**

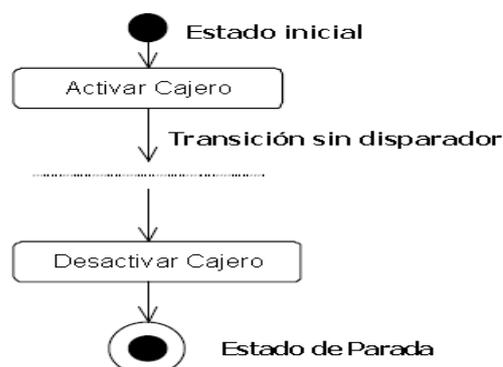
Un estado de actividad puede descomponerse en sub actividades representadas a través de otros diagramas de actividades, estos estados si pueden ser interrumpidos y tardan un cierto tiempo en completarse. Además podemos encontrar otros elementos como: acciones de entrada y de salida del estado en cuestión.



**Figura 1.25 Estado de actividad<sup>27</sup>**

## b) TRANSICIONES.

Reflejan el paso de un estado a otro de una actividad o una acción. Se produce como resultado de la finalización del estado, se lo indica utilizando dos disparadores de inicio y fin.



**Figura 1.26 Transiciones sin disparador<sup>28</sup>**

<sup>26</sup> Fuente: <http://www.clicgear.com/manuales/uml/modelodinamico.aspx>

<sup>27</sup> Fuente: <http://www.clicgear.com/manuales/uml/modelodinamico.aspx>

<sup>28</sup> Fuente: <http://www.clicgear.com/manuales/uml/modelodinamico.aspx>

### c) BIFURCACIONES.

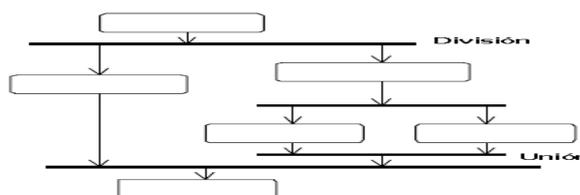
Representa caminos alternativos utilizando el símbolo de rombo. La bifurcación tendrá una transición de entrada y dos o más de salida. En cada transición de salida se colocará una expresión booleana que será evaluada una vez al llegar a la bifurcación.



**Figura 1.27 Bifurcación<sup>29</sup>**

### 1.2.5.3.4.2 DIVISIÓN Y UNIÓN.

Se utiliza en caso de tareas concurrentes. La división representa la concurrencia y al unión de nuevo al flujo de control secuencial y se representa gráficamente por una línea horizontal ancha.



**Figura 1.28 División y Unión<sup>30</sup>**

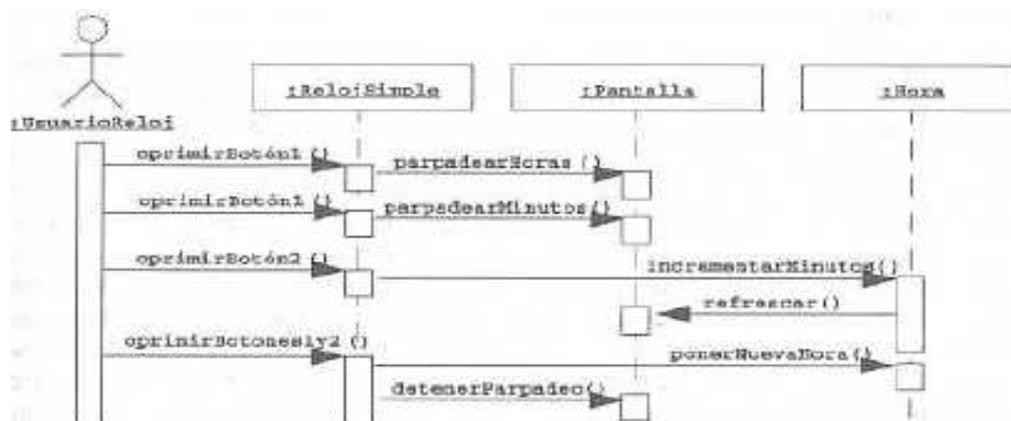
### 1.2.5.3.5 DIAGRAMA DE SECUENCIAS.

Este diagrama formaliza el comportamiento del sistema y visualiza la comunicación entre objetos. Son útiles para identificar objetos adicionales que participan en los casos de uso.

<sup>29</sup> Fuente: <http://www.clickear.com/manuales/uml/modelodinamico.aspx>

<sup>30</sup> Fuente: <http://www.clickear.com/manuales/uml/modelodinamico.aspx>

Representa una interacción, poniendo el foco en la secuencia de los mensajes que se intercambian. Muestra una iteración ordenada según la secuencia temporal de eventos. El eje vertical representa el tiempo y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin orden predeterminado, cada objeto o actor tiene un eje vertical y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas como restricciones de tiempo, descripciones de acciones, etc., bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.



**Figura 1.29 Diagrama de Secuencia<sup>31</sup>**

#### 1.2.5.3.6 DIAGRAMA DE ESTADOS.

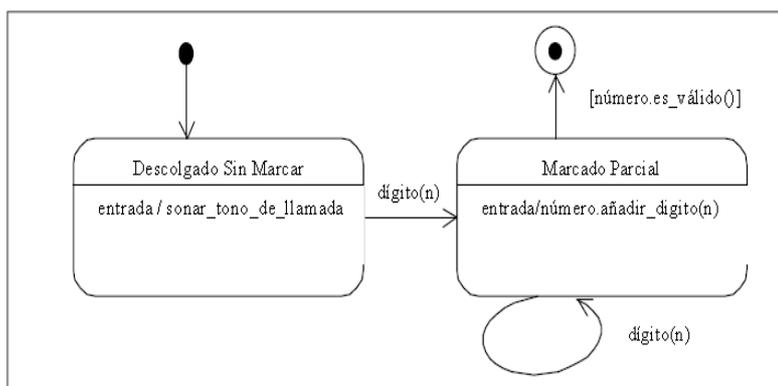
Ilustra como un elemento se puede mover entre estados que clasifican su comportamiento, es decir, describe el comportamiento de un objeto individual con varios estados y transiciones entre esos estados. Un estado representa un conjunto particular de valores para un objeto.

Muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando que eventos hacen que se pasen de un estado a otro y cuales son las respuestas y acciones que genera. Se representa con un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con

<sup>31</sup> Fuente: INGENIERÍA DE SOFTWARE ORIENTADO A OBJETOS. BRUEGGE, Bernd. DUTOIT, Allen. 2002.

los nombres de los eventos. Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa con una flecha desde el estado origen al estado destino.

La caja de un estado puede tener uno o dos compartimentos, en el primero aparece el nombre del estado, el segundo es opcional y pueden aparecer acciones de entrada, de salida y de acciones internas. Una **acción de entrada** aparece en la forma *entrada/acción \_ asociada* donde *acción \_ asociada* es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta. Una **acción de salida** aparece en la forma *salida/acción \_ asociada*. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta. Una **acción interna** es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Aparece en la forma *nombre\_de\_evento/acción \_ asociada*.



**Figura 1.30 Diagrama de estados<sup>32</sup>**

Este diagrama puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción del caso de uso o del objeto. El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo. En realidad los estados inicial y final son pseudo estados pues un objeto no puede estar en esos estados, pero nos sirve para saber cuales son las transiciones iniciales y finales.

<sup>32</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramasestado.aspx>

### 1.2.5.3.7 DIAGRAMA DE CASOS DE USO.

Son utilizados en la obtención de requerimientos y análisis para representar la funcionalidad del sistema, desde un punto de vista externo. La identificación de actores y casos de uso da como resultado la frontera del sistema, es decir, la diferencia entre las tareas realizadas por el sistema y las realizadas por el ambiente. Los actores están fuera de la frontera del sistema mientras que los casos de uso están dentro de la frontera del sistema. Muestra las relaciones entre los actores y el sujeto (sistema) y los casos de uso.

#### 1.2.5.3.7.1 ELEMENTOS.

Los elementos que pueden aparecer en los casos de uso son: actores, casos de uso y relaciones entre casos de uso.

##### a) ACTORES.

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes y se la usa para todo tipo de actores.

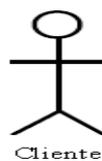


Figura 1.31 Actor

##### b) CASOS DE USO.

Un caso de uso es la descripción de una secuencia de interacciones que se producen entre un actor y un sistema, cuando un actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad y se representa en el diagrama de casos de uso mediante una elipse con el nombre del caso de uso en el interior. El nombre del caso de uso

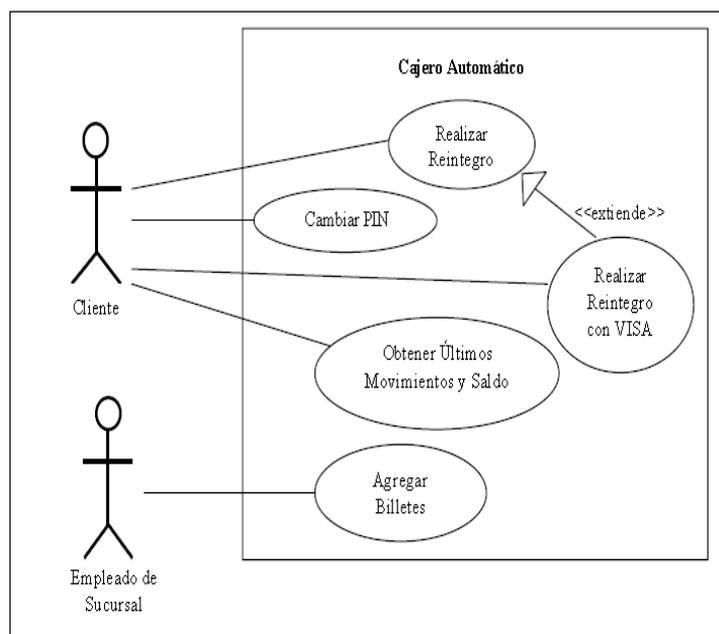
debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

### c) RELACIONES ENTRE CASOS DE USO.

Entre dos casos de uso puede haber las siguientes relaciones:

- **EXTIENDE:** Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- **USA:** Cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema y los actores fuera, cada actor está unido a los casos de uso en los que participa mediante una línea.



**Figura 1.32 Diagrama de caso de uso<sup>33</sup>**

<sup>33</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramascasouso.aspx>

### 1.2.5.3.8 OTROS DIAGRAMAS.

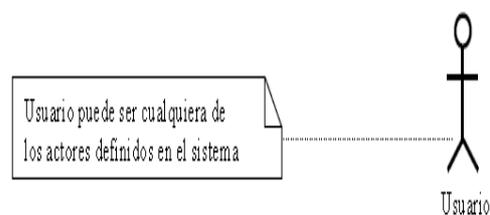
Además de los diagramas especificados anteriormente, UML posee más diagramas útiles y que serán descritos brevemente.

- **DIAGRAMA DE ESTRUCTURA DE COMPOSICIÓN:** Representa la estructura interna de un clasificador.
- **DIAGRAMA DE OBJETOS:** Modela la estructura estática de los objetos en el sistema. Muestra instancias de las clases y sus relaciones. Estos diagramas se enfocan en la perspectiva de casos reales.
- **DIAGRAMA DE COMUNICACIONES:** Modela interacciones entre objetos.
- **DIAGRAMA DE TIEMPOS:** Mostrar los cambios en el estado o la condición de una línea de vida a lo largo del tiempo lineal.

### 1.2.5.4 ELEMENTOS COMUNES A TODOS LOS DIAGRAMAS.

#### 1.2.5.4.1 NOTAS.

Añade cualquier tipo de comentario a un diagrama o a un elemento de un diagrama, indica información en un formato libre, cuando la notación del diagrama en cuestión no nos permite expresar dicha información de manera adecuada. Se representa como un rectángulo con una esquina doblada con texto en su interior. Puede aparecer en un diagrama solo o unido a un elemento por medio de una línea discontinua. Puede contener restricciones, comentarios, el cuerpo un procedimiento, etc.



**Figura 1.33 Nota<sup>34</sup>**

<sup>34</sup> Fuente: INGENIERÍA DE SOFTWARE ORIENTADO A OBJETOS. BRUEGGE, Bernd. DUTOIT, Allen. 2002.

### 1.2.5.4.2 CLASES

La clase es la unidad básica que encapsula toda la información de un objeto. A través de ella se puede modelar el entorno en estudio. Esta representada por un rectángulo con tres divisiones: superior, intermedio e inferior.

**SUPERIOR:** Contiene el nombre de la clase.

**INTERMEDIO:** Contiene atributos o variables de instancia que caracterizan a la clase. Pueden ser private, protected o public

**INFERIOR:** Contiene métodos u operaciones los cuales son la forma como interactúa el objeto con su entorno, dependiendo de la visibilidad: private, protected o public.

Nombre de la clase
Lista de atributos
Lista de métodos u operaciones

**Tabla 1.9 Clase**

### 1.2.5.4.3 ATRIBUTOS.

- Representa alguna propiedad de la clase.
- Puede representarse mostrando su nombre y su tipo incluso su valor por defecto.
- Definen la estructura de una clase y sus correspondientes objetos.
- Define el valor de un dato para todos los objetos pertenecientes a una clase.

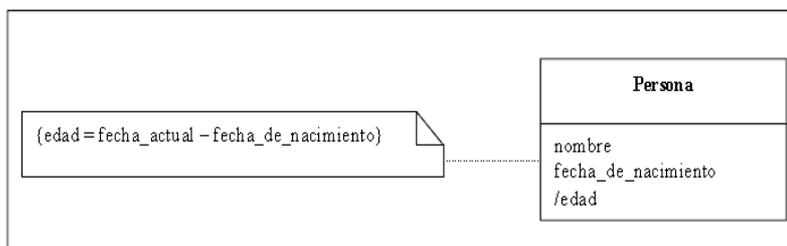
#### EJEMPLO

Nombre, cédula, edad son atributos de Persona.

#### 1.2.5.4.3.1 ELEMENTOS DERIVADOS.

Su valor se puede calcular a través de otros elementos presentes en el modelo, pero que se incluyen en el modelo por motivos de claridad o como decisión de

diseño. Se representa con una barra “/” precediendo al nombre del elemento derivado.



**Figura 1.34 Elementos derivados<sup>35</sup>**

#### 1.2.5.4.4 ASOCIACIONES.

Se representa mediante una línea que une dos clases. Puede tener una serie de elementos gráficos que expresan características particulares de la asociación. Los elementos gráficos más importantes son:

##### 1.2.5.4.4.1 NOMBRE DE LA ASOCIACIÓN Y DIRECCIÓN

Es opcional y se muestra como un texto que está próximo a la línea. Se puede añadir un pequeño triángulo sólido que indique la dirección en la cual leer el nombre de la asociación.



**Figura 1.35 Asociaciones<sup>36</sup>**

##### 1.2.5.4.4.2 MULTIPLICIDAD.

Es una restricción que se pone a la asociación. Puede expresarse con las siguientes formas:

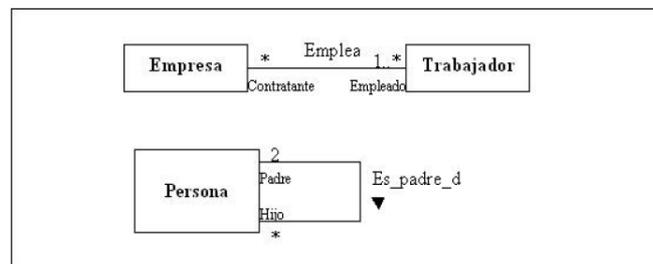
<sup>35</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramasestructuraestatica.aspx>

<sup>36</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramasestructuraestatica.aspx>

- Con un número fijo: 1.
- Con un intervalo de valores: 2.....5.
- Con un rango en el cual uno de los extremos es un asterisco. Es decir un intervalo abierto. Por ejemplo 2...\*, significa dos o más.
- Con una combinación de elementos como los anteriores separados por comas: 1, 3..5, 7. 15...\*
- Con un asterisco: \*. En este caso indica que puede tomar cualquier valor (cero o más).

#### 1.2.5.4.4.3 ROLES

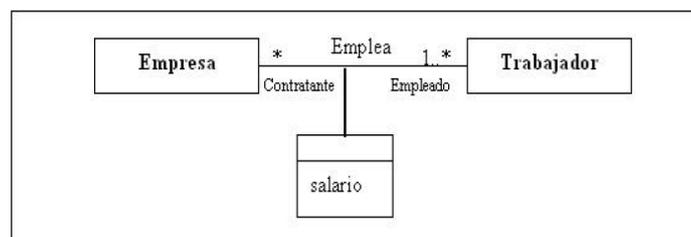
Para indicar el papel que juega una clase en una asociación se puede especificar un nombre de rol.



**Figura 1.36 Roles<sup>37</sup>**

#### 1.2.5.4.4.4 CLASES ASOCIACIÓN.

Cuando una asociación tiene propiedades propias se representa como una clase unida a la línea de la asociación por medio de una línea de trazos. Tanto la línea como el rectángulo de clase representan el mismo elemento conceptual: la asociación. Por lo tanto ambos tienen el mismo nombre el de la asociación.

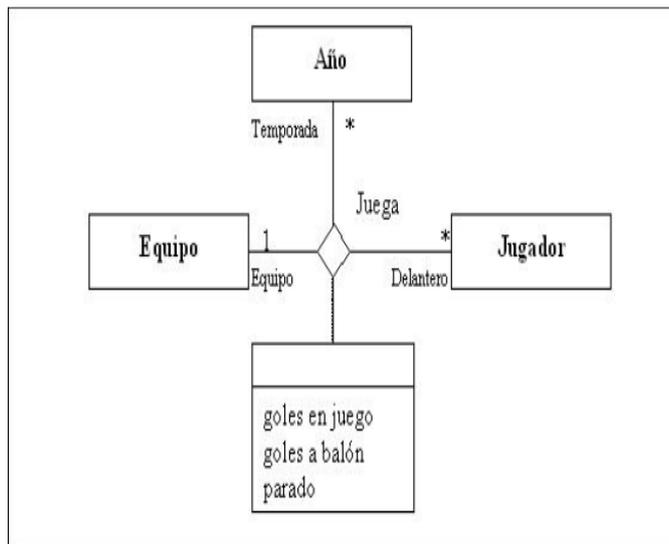


**Figura 1.37 Clases de Asociación<sup>38</sup>**

<sup>37</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramaestructuraestatica.aspx>

#### 1.2.5.4.4.5 ASOCIACIONES N-ARIAS.

En el caso de una asociación en la que participan más de dos clases, las clases se unen con una línea a un diamante.



**Figura 1.38 Asociaciones n-arias<sup>39</sup>**

#### 1.2.5.4.4.6 ENSAMBLADOS

Son formas especiales de asociación entre un todo y sus partes, en donde el ensamblado está compuesto por sus componentes.

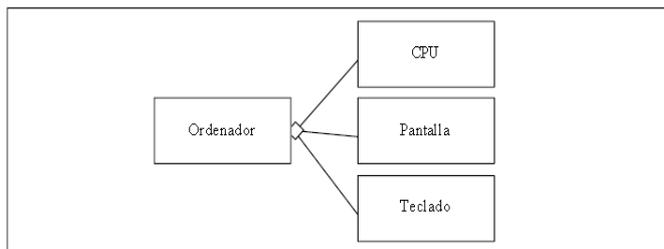
#### 1.2.5.4.4.7 AGREGACIÓN.

El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”

El objeto base utiliza el incluido para su funcionamiento. Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye.

<sup>38</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramaestructuraestatica.aspx>

<sup>39</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramaestructuraestatica.aspx>



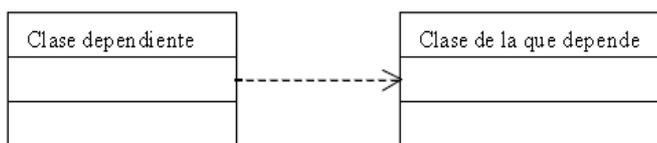
**Figura 1.39 Agregación<sup>40</sup>**

#### 1.2.5.4.4.8 COMPOSICIÓN

El objeto base se construye a partir del objeto incluido. Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye.

#### 1.2.5.4.4.9 DEPENDENCIA O INSTANCIACIÓN.

Representa un tipo de relación muy particular, en la que una clase es instanciada. El uso más particular de esta relación es para denotar la dependencia que tiene una clase de otra. Se representa por medio de una línea de trazo discontinuo entre los dos elementos con una flecha en su extremo. El elemento dependiente es el origen de la flecha y el elemento del que depende es el destino.



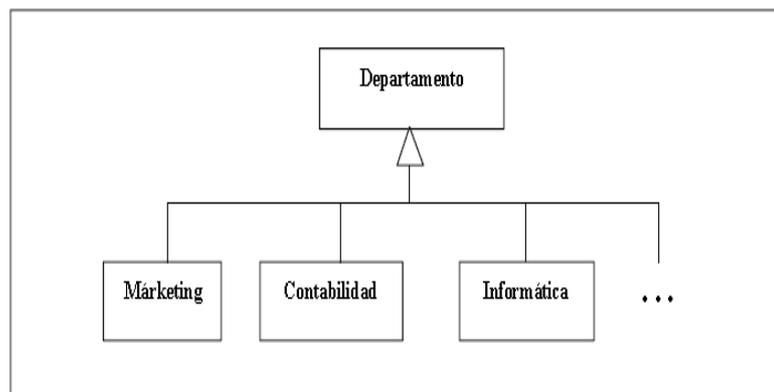
**Figura 1.40 Dependencia**

#### 1.2.5.4.5 HERENCIA

Se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o a la clase “padre”. Si se tiene una relación de herencia con varias clases subordinadas, pero en un diagrama concreto no se quieren poner todas, esto se representa mediante puntos suspensivos.

<sup>40</sup> Fuente: <http://www.clickear.com/manuales/uml/diagramasestructuraestatica.aspx>

- La herencia es una abstracción importante para compartir similitudes entre clases, donde todos los atributos y operaciones comunes a varias clases se pueden compartir por medio de la superclase.
- Las clases con atributos y operaciones comunes se pueden organizar de forma jerárquica. Es un buen vehículo para no replicar información innecesariamente.
- Es útil para el modelo conceptual al igual que para la implementación.
- La superclase generaliza a sus subclasses y las subclasses especializan a la superclase.
- La herencia indica que una subclase hereda los métodos y atributos especificados por una súper clase, por ende la sub clase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la súper clase.



**Figura 1.41 Herencia<sup>41</sup>**

### 1.2.5.5 MODELADO CON UML

El modelado con UML posee una diversidad de diagramas útiles en los diferentes procesos del desarrollo del software, pero se describirá las notaciones fundamentales del UML que usamos a lo largo de este trabajo. El UML ha sido diseñado para un amplio rango de aplicaciones. El desarrollo de sistemas se enfoca en tres modelos diferentes del sistema.

<sup>41</sup> Fuente: <http://www.clicgear.com/manuales/uml/diagramasestructuraestatica.aspx>

**MODELO FUNCIONAL:** Describe la funcionalidad del sistema desde el punto de vista del usuario, esta representado por los diagramas de casos de uso.

**MODELO DE OBJETOS:** Describe la estructura de un sistema desde el punto de vista de objetos, atributos, asociaciones y operaciones, esta representado por diagramas de clase.

**MODELO DINÁMICO:** Describe el comportamiento interno del sistema, esta representado por los diagramas de secuencia, diagramas de gráfico de estado y diagramas de actividad.

#### 1.2.5.5.1 CONCEPTOS BÁSICOS.

**SISTEMA:** Es un conjunto organizado de partes que se comunican, diseñado para un propósito específico. Puede tener subsistemas, es decir sistemas más simples.

**MODELADO:** Es la construcción de una abstracción del sistema que se enfoca en aspectos interesantes e ignora los detalles irrelevantes.

**VISTA:** Se enfoca en un subconjunto de un modelo para hacerlo comprensible.

**NOTACIONES:** Son reglas gráficas o textuales para la representación de vistas. Se puede utilizar diferentes notaciones para representar la misma vista.

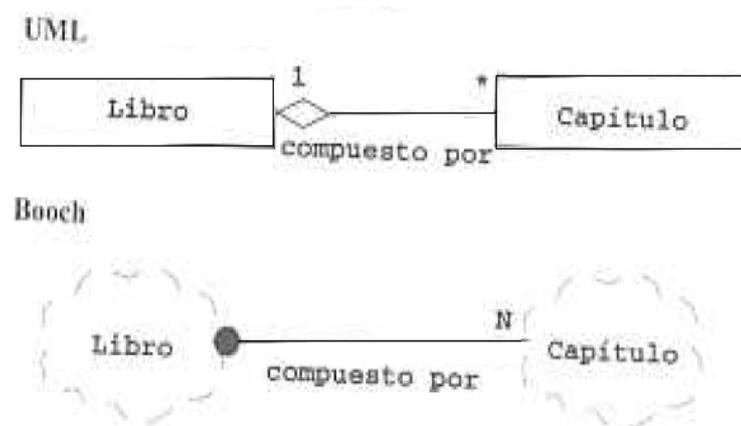
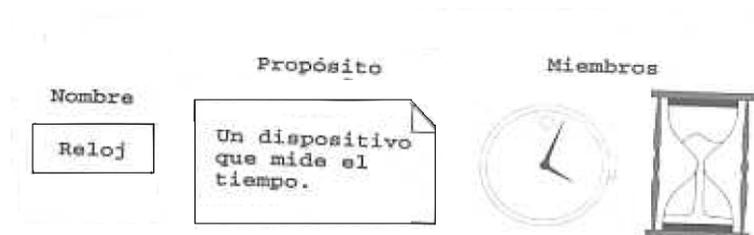


Figura 1.42 Modelo con dos notaciones diferentes<sup>42</sup>

<sup>42</sup> Fuente: INGENIERÍA DE SOFTWARE ORIENTADO A OBJETOS. BRUEGGE, Bernd. DUTOIT, Allen. 2002.

**FENÓMENO:** Es un objeto del mundo tal como es percibido.

**CONCEPTO:** Es una abstracción que describe un conjunto de fenómenos. Esta definido por su nombre, propósito y sus miembros.



**Figura 1.43 Componentes del concepto reloj<sup>43</sup>**

**ABSTRACCIÓN:** Es la clasificación de los fenómenos en conceptos.

**INSTANCIA:** Es cualquier miembro de un tipo de dato específico. Una instancia de un tipo de dato puede ser manejada con las operaciones definidas por el tipo de dato. En UML se lo representa con un rectángulo con nombre subrayado.

**OBJETO:** Es una instancia de una clase. Un objeto tiene una identidad y almacena valores de atributos.

**DOMINIO DE APLICACIÓN:** Representa todos los aspectos del problema del usuario. Esto incluye el ambiente físico, los usuarios y otras personas, sus procesos de trabajo, etc.

El análisis orientado a objetos esta interesado en el modelado del dominio de aplicación.

**DOMINIO DE SOLUCIÓN:** Es el espacio de todos los sistemas posibles. El modelado de dominio de solución representa las actividades de diseño del sistema y diseño de objetos del proceso de desarrollo.

El diseño orientado a objetos esta interesado en el modelado del dominio de la solución.

### 1.2.5.6 VENTAJAS DEL UML

- Se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real.

<sup>43</sup> Fuente: INGENIERÍA DE SOFTWARE ORIENTADO A OBJETOS. BRUEGGE, Bernd. DUTOIT, Allen. 2002.

- UML es una consolidación de muchas de las notaciones y conceptos mas usadas orientados a objetos.
- Diseño y documentación
- Código reutilizable.
- Descubrimiento de fallas
- Ahorro del tiempo en el desarrollo del software.
- Mucho más fáciles las modificaciones.
- Más fácil comunicación entre programadores.
- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se puede automatizar determinados procesos y generar código a partir de los modelos y a la inversa. Esto permite que el modelo y el código estén actualizados.

#### **1.2.5.7 DESVENTAJAS DEL UML.**

- UML no es un método de desarrollo. No muestra como pasar del análisis al diseño y de este al código.
- Al no ser un método de desarrollo es independiente del ciclo de desarrollo que se vaya a seguir.
- No se presta con facilidad al diseño de sistemas distribuidos. En tales sistemas cobran importancia factores como transmisión, serialización, persistencia, etc. UML no cuenta con maneras de describir tales factores. Por ejemplo no se puede usar UML para señalar que un objeto es persistente o remoto.
- UML es excesivamente extenso y complejo.
- Contiene diagramas y estructuras que son redundantes o muy poco usadas.
- Semántica imprecisa.
- Trata de ser todas las cosas para todas las personas.

## **CAPITULO 2**

### **ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA MODELO Y METODOLOGÍA DEL SISTEMA**

Este proyecto de tesis utiliza el modelo de desarrollo prototipado, en este capítulo se desarrolla las diferentes etapas que utiliza este modelo para conseguir el producto de ingeniería, además del uso de la metodología uml en el diseño del mismo.

#### **2.1 IDENTIFICACIÓN DE REQUERIMIENTOS**

A través de varias entrevistas realizadas tanto a la directora, maestros y alumnos del Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur, se ha podido definir diversas características concernientes al proceso que involucra este proyecto de investigación. En donde se encontró:

El Departamento de Idiomas surge para dar una solución integral al requerimiento de los estudiantes para el dominio de un segundo idioma que además de ser una necesidad del mundo actual, es un requisito incluido en el perfil de egresados de la Universidad Politécnica Salesiana, de conformidad con lo dispuesto en la Ley de Educación Superior.

La Coordinadora es la autoridad máxima y ejerce la representación administrativa y académica del mismo, en el caso del Campus Sur de la Universidad se designa una asistente quien cumple las funciones de coordinación, la cual es ejercida por la Lcda. Martha Ríos.

El Departamento esta integrado por 11 docentes a los cuales se les asigna sus respectivos cursos y horarios para que puedan dictar sus respectivas clases. Este

departamento ofrece sus servicios en 3 modalidades establecidos de la siguiente manera: modalidad presencial, modalidad intensiva y modalidad a distancia, regidas por el sistema de créditos correspondiendo a cuatro créditos cada nivel. La modalidad presencial dura todo el semestre y es parte de la malla curricular e inicia en los niveles básicos, con su horario respectivo. La modalidad a distancia pretende motivar el auto-estudio del alumno en su metodología el estudiante deberá trabajar con el material proporcionado y desarrollar trabajos mensuales y sus respectivas pruebas, apoyado por tutorías, únicamente se pueden matricular en esta modalidad los alumnos de octavo, noveno y décimo nivel además de los alumnos egresados. La modalidad intensiva es similar a la modalidad presencial, variando únicamente en el tiempo de duración del nivel, en este caso es de dos meses.

**Los estudiantes deben aprobar un nivel para poder tomar el inmediato superior, de acuerdo a la organización curricular de cada Carrera. De manera general la distribución es cuatro niveles comunes y dos aplicados a cada disciplina de la Facultad, además de dos niveles opcionales para aquellos estudiantes que desean continuar con la preparación académica del idioma extranjero, después de los cuales se otorga un certificado de suficiencia.**

**En cuanto a la rendición de exámenes escritos, se realizan a mitad de semestre y al final de semestre, a través de modelos estandarizados que son elaboradas por los docentes del Departamento. En el caso que un estudiante tenga conocimientos del idioma extranjero puede rendir un examen de ubicación antes de matricularse por primera vez, que se lo aplicará una sola vez en cada periodo académico, si el estudiante al ser ubicado no se matricula en el nivel que le corresponde y en el semestre en el que rinde el examen, este queda sin validez, pudiendo rendir un nuevo examen de ubicación, pero en otro periodo.**

**En lo referente a la puntuación para aprobar el nivel, esta comprende de trabajos, pruebas orales, actuación en clase, etc., con una valoración de 30**

puntos y el examen de 20 puntos, para un total de 50 puntos en cada trimestre. En el caso de la modalidad a distancia la calificación consta de los trabajos de 60 puntos y cuatro pruebas de 40 puntos para un total de 100 puntos.

La nota mínima de aprobación de un nivel en cualquier modalidad es de 70/100 y de acuerdo a los reglamentos de la universidad no existen exámenes supletorios.

El alumno deberá presentarse con el carné estudiantil o la cedula de identidad para poder rendir el examen.

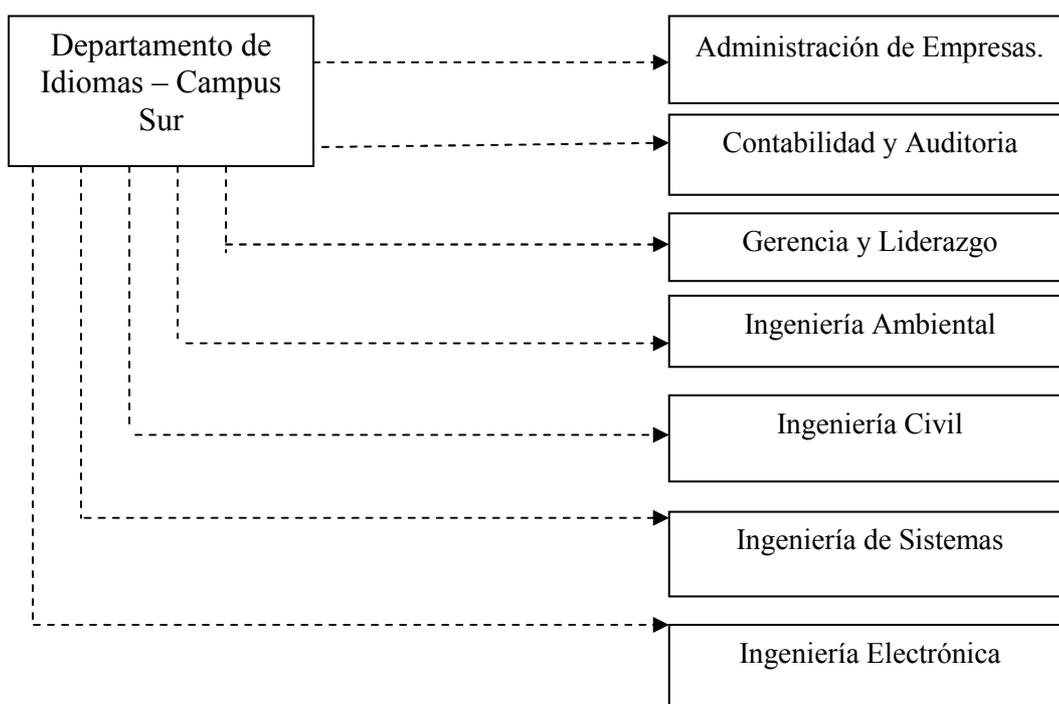
El análisis de todos los datos concernientes a este proceso nos permite determinar con exactitud el proceso que se lleva a cabo en el Departamento de Idiomas en cuanto a su estructura, roles en el departamento, rendición de exámenes y calificación de los mismos, lo que nos permite crear una solución adecuada a los requerimientos específicos de este departamento en cuanto a la automatización del proceso de Evaluación.

## 2.2 DISEÑO RÁPIDO

Esta etapa debería mostrar una representación de los aspectos básicos del sitio que serán visibles para el usuario final, pero a fin de obtener un producto de ingeniería mejor diseñado y documentado, se ha realizado un diseño más profundo.

### DEPARTAMENTO DE IDIOMAS

El siguiente diagrama representa el diseño del negocio, es decir, muestra como el departamento de idiomas planifica servir a sus alumnos.



**Figura 2.1 Diagrama para representar las distintas carreras a las que sirve el Departamento de Idiomas<sup>44</sup>**

<sup>44</sup> Fuente: [http://www.ups.edu.ec/portal/page/portal/upsportal/ofertaacademica#page\(1\)](http://www.ups.edu.ec/portal/page/portal/upsportal/ofertaacademica#page(1))

## 2.2.1 DIAGRAMAS DEL SISTEMA

### 2.2.1.1 DIAGRAMAS DE CASO DE USO.

#### 2.2.1.1.1 ACTORES

El sistema cuenta con 4 actores que son los siguientes: Administrador, Profesor, Editor, Estudiante.

#### ADMINISTRADOR

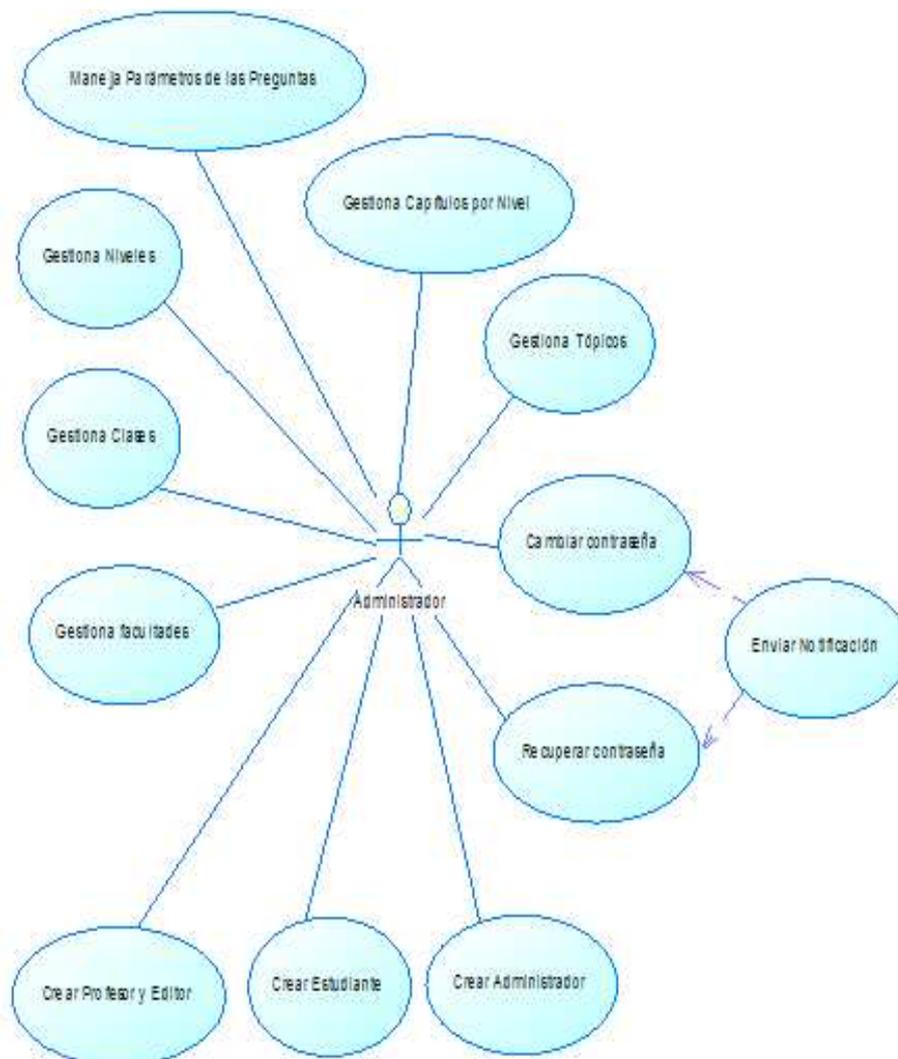


Figura 2.2 Diagrama del Actor Administrador.

PROFESOR

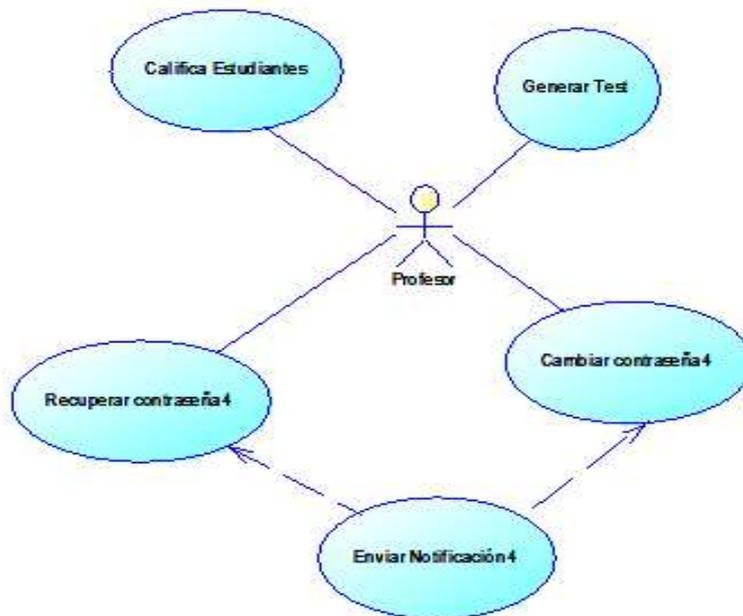


Figura 2.3 Diagrama del Actor Profesor

EDITOR

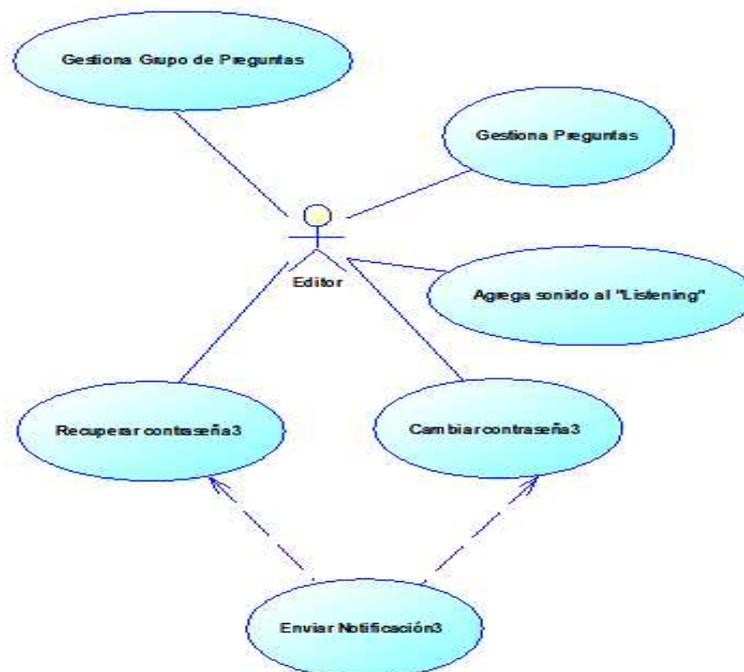
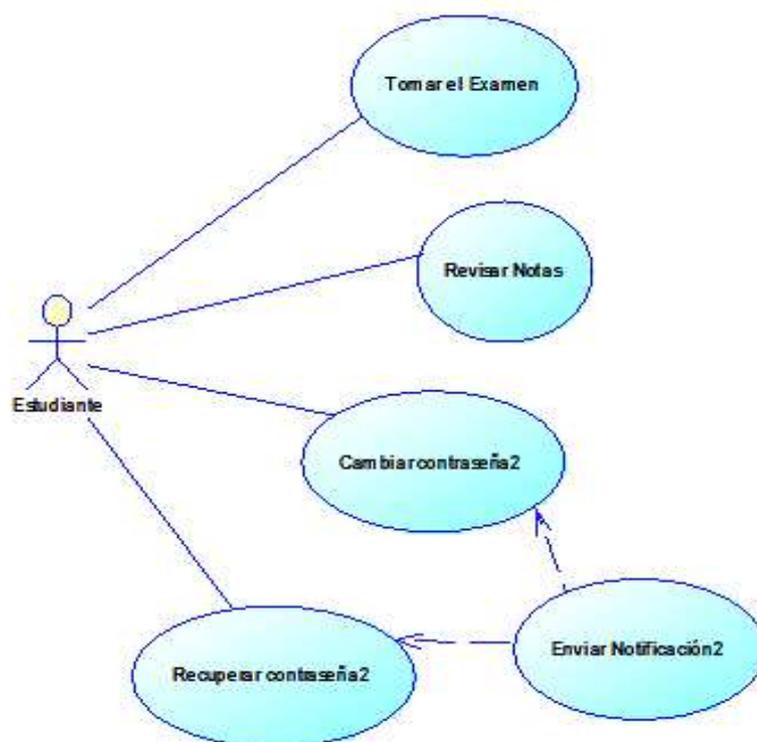


Figura 2.4 Diagrama del Actor Editor

ESTUDIANTE



**Figura 2.5 Diagrama del Actor Estudiante**

#### 2.2.1.1.1.1 ACTOR ADMINISTRADOR

Es el encargado de gestionar los usuarios del sistema, así como los parámetros necesarios para que se lleve a cabo el test online

##### a) ASOCIACIONES DEL ACTOR ADMINISTRADOR

NOMBRE	DESTINO	FUENTE
Association_3	Maneja Parámetros de las Preguntas	Administrador
Association_4	Crear Administrador	Administrador
Association_5	Gestiona Tópicos	Administrador
Association_6	Crear Estudiante	Administrador
Association_7	Gestiona Niveles	Administrador
Association_8	Gestiona Clases	Administrador
Association_9	Gestiona facultades	Administrador
Association_10	Crear Profesor y Editor	Administrador
Association_11	Gestiona Capítulos por Nivel	Administrador
Association_23	Cambiar contraseña	Administrador
Association_24	Recuperar contraseña	Administrador

**Tabla 2.1 Asociaciones del Actor Administrador**

### 2.2.1.1.1.2 ACTOR EDITOR.

Es el encargado de gestionar las preguntas asociadas por grupos de las mismas para los diferentes niveles, facultades, tópicos y unidades del que está compuesto el test. En las preguntas correspondientes al tópico "Listening" el editor puede adjuntar un archivo de sonido.

#### a) ASOCIACIONES DEL ACTOR EDITOR

<b>NOMBRE</b>	<b>DESTINO</b>	<b>FUENTE</b>
Association_12	Gestiona Grupo de Preguntas	Editor
Association_13	Gestiona Preguntas	Editor
Association_14	Agrega sonido al "Listening"	Editor
Association_19	Recuperar contraseña	Editor
Association_20	Cambiar contraseña	Editor

**Tabla 2.2 Asociaciones del Actor Editor**

### 2.2.1.1.1.3 ACTOR ESTUDIANTE.

Es quién realiza el test online y revisa las notas correspondientes al mismo, así como las restantes notas necesarias para aprobar el nivel.

#### a) LISTA DE ASOCIACIONES DEL ACTOR ESTUDIANTE.

<b>NOMBRE</b>	<b>DESTINO</b>	<b>FUENTE</b>
Association_15	Tomar el Examen	Estudiante
Association_16	Revisar Notas	Estudiante
Association_17	Recuperar contraseña	Estudiante
Association_18	Cambiar contraseña	Estudiante

**Tabla 2.3 Asociaciones del Actor Estudiante**

### 2.2.1.1.1.4 ACTOR PROFESOR.

Es el encargado de generar los exámenes para los estudiantes, además de gestionar las notas para la aprobación del nivel.

a) **LISTA DE ASOCIACIONES DEL ACTOR PROFESOR.**

Nombre	Destino	Fuente
Association_1	Generar Test	Profesor
Association_2	Califica Estudiantes	Profesor
Association_21	Cambiar contraseña	Profesor
Association_22	Recuperar contraseña	Profesor

**Tabla 2.4 Asociaciones del Actor Profesor**b) **LISTA DE DEPENDENCIAS.**

Nombre	Objeto Influyente	Objeto Dependiente
Dependency_1	Cambiar contraseña	Enviar Notificación
Dependency_2	Recuperar contraseña	Enviar Notificación

**Tabla 2.5 Lista de dependencias****2.2.1.1.2 LISTA DE CASOS DE USO**

NOMBRE
Generar Test
Califica Estudiantes
Crear Administrador
Crear Estudiante
Crear Profesor y Editor
Gestiona facultades
Gestiona Clases
Gestiona Niveles
Maneja Parámetros de las Preguntas
Gestiona Tópicos
Gestiona Capítulos por Nivel
Gestiona Grupo de Preguntas
Gestiona Preguntas
Agrega sonido al "Listening"
Tomar el Examen
Revisar Notas
Cambiar contraseña
Recuperar contraseña
Enviar Notificación

**Tabla 2.6 Lista de Casos de Uso**

**2.2.1.1.2.1 CASO DE USO AGREGA SONIDO AL LISTENING:** Enlaza un archivo de audio a la sección de preguntas Listening.

**a) DICCIONARIO DEL CASO DE USO AGREGA SONIDO AL "LISTENING"**

<b>Actores</b>	Editor
<b>Precondiciones</b>	El editor debe iniciar sesión. Se debe haber agregado anteriormente el question group correspondiente
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Se hace clic sobre el icono de edición del ListeningQuestionGroup.</li> <li>2. Se sube el archivo .mp3 y su nombre se adjunta a la base de datos para luego ser llamada</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si se sube un archivo que no sea .mp3 aparece un mensaje de error</li> </ol>
<b>Pos condiciones</b>	La pregunta de Listening esta agregada en el sistema.

**Tabla 2.7 Diccionario del Caso de Uso Agregar sonido al Listening**

**2.2.1.1.2.2 CASO DE USO CALIFICA ESTUDIANTES:** Crea un nuevo registro de calificaciones tomando en cuenta la modalidad a la que pertenece el estudiante, además puede modificarla, excepto la nota del test online.

**a) DICCIONARIO DEL CASO DE USO CALIFICA ESTUDIANTES**

<b>Actores</b>	Profesor
<b>Precondiciones</b>	El profesor debe iniciar sesión
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ingresar a Calificaciones</li> <li>2. Escoger los parámetros de filtrado (classroom, modality, Faculty) para obtener el/los estudiantes respectivos.</li> <li>3. Localizar el estudiante respectivo</li> <li>4. Ingresar o modificar la calificación</li> <li>5. Envía los datos a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>4. Si alguno de los campos no cumple con las reglas de validación de integridad de datos el sistema muestra mensajes para corregir. Una vez corregidos los datos se proceden a guardar</li> </ol>
<b>Pos condiciones</b>	Los datos de la calificación quedan registrados en la DB.

**Tabla 2.8 Diccionario del Caso de Uso Califica Estudiantes.**

**2.2.1.1.2.3 CASO DE USO CAMBIAR CONTRASEÑA:** Permite cambiar la contraseña de usuario aprovechando la funcionalidad del framework. Net

**a) DICCIONARIO DEL CASO DE USO CAMBIAR CONTRASEÑA**

<b>Actores</b>	Administrador, Estudiante, Profesor, Editor
<b>Precondiciones</b>	Tener usuario y contraseña en el sistema.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la opción cambio de contraseña</li> <li>2. Digital la clave anterior</li> <li>3. Ingresar la nueva clave</li> <li>4. Confirmar la nueva clave</li> </ol>
<b>Flujo alternativo</b>	2. No recordar la clave anterior por lo tanto ir a olvido su clave
<b>Pos condiciones</b>	El usuario cambio su contraseña.

**Tabla 2.9 Diccionario del Caso de Uso Cambiar Contraseña**

**2.2.1.1.2.4 CASO DE USO CREAR ADMINISTRADOR:** Registra los datos de un nuevo administrador.

**a) DICCIONARIO DEL CASO DE USO CREAR ADMINISTRADOR**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	El administrador debe iniciar sesión. El nuevo usuario entrega la ficha de inscripción
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Se presiona Añadir administrador</li> <li>2. Se llenan los campos del formulario</li> <li>3. Se envía los datos a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir</li> <li>3. Una vez corregidos los datos se proceden a guardar</li> </ol>
<b>Pos condiciones</b>	El administrador queda registrado en el sistema

**Tabla 2.10 Diccionario del Caso de Uso Crear Administrador**

**2.2.1.1.2.5 CASO DE USO CREAR ESTUDIANTE:** Registra los datos de un nuevo Estudiante.

**a) DICCIONARIO DEL CASO DE USO CREAR ESTUDIANTE**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. Se presiona Añadir administrador</li> <li>2. Se llenan los campos del formulario</li> <li>3. Se envía los datos a la DB</li> </ol>
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Se presiona Añadir Estudiante</li> <li>2. Se llenan los campos del formulario</li> <li>3. Se envía los datos a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir</li> <li>3. Una vez corregidos los datos se proceden a guardar</li> </ol>
<b>Pos condiciones</b>	El estudiante queda registrado en el sistema.

**Tabla 2.11 Diccionario del Caso de Uso Crear Estudiante**

**2.2.1.1.2.6 CASO DE USO CREAR PROFESOR Y EDITOR:** Registra los datos de un nuevo Profesor y/o Editor.

**a) DICCIONARIO DEL CASO DE USO CREAR PROFESOR Y EDITOR**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	<p>El administrador debe iniciar sesión. El nuevo usuario entrega la ficha de inscripción</p>
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Se presiona Añadir Teacher/Editor</li> <li>2. Se llenan los campos del formulario</li> <li>3. Se envía los datos a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir</li> <li>3. Una vez corregidos los datos se proceden a guardar</li> </ol>
<b>Pos condiciones</b>	El usuario Teacher/Editor queda registrado en el sistema.

**Tabla 2.12 Diccionario del Caso de Uso Crear Profesor/Editor**

**2.2.1.1.2.7 CASO DE USO ENVIAR NOTIFICACIÓN:** Usa el Simple Mail Transfer Protocol del framework.net que se encuentra en el servidor para notificar al usuario el cambio de la contraseña o su contraseña en caso de ser usuario nuevo.

a) **DICCIONARIO DEL CASO DE USO ENVIAR NOTIFICACIÓN**

<b>Actores</b>	Administrador, Editor, Profesor, Estudiante
<b>Precondiciones</b>	El administrador debe haber creado el usuario El usuario debe haber olvidado su contraseña o querer cambiar su contraseña
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a cambiar su contraseña o ingresa a recuperar su contraseña y hace uso de la pregunta de seguridad</li> <li>2. El sistema guarda el cambio</li> <li>3. Se envía una notificación al correo con el cambio de contraseña.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario no cambia su contraseña o el usuario no hace uso de la pregunta de seguridad.</li> </ol>
<b>Pos condiciones</b>	El usuario recibe la notificación a su correo.

**Tabla 2.13 Diccionario del Caso de Uso Enviar Notificación**

**2.2.1.1.2.8 CASO DE USO GENERAR TEST:** Se encarga de activar un examen para que pueda rendir un estudiante.

a) **DICCIONARIO DEL CASO DE USO GENERAR TEST**

<b>Actores</b>	Profesor
<b>Precondiciones</b>	El profesor debe iniciar sesión Debe existir un examen
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Localiza un estudiante de la lista</li> <li>2. Presiona el botón de activación</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si el estudiante no esta presente para el examen su examen sigue desactivado</li> </ol>
<b>Pos condiciones</b>	El examen queda activo para que el estudiante pueda rendirlo.

**Tabla 2.14 Diccionario del Caso de Uso Generar Test**

**2.2.1.1.2.9 CASO DE USO GESTIONA CAPÍTULOS POR NIVEL:** Configura los capítulos que pertenecen a cada nivel.

**a) DICCIONARIO DEL CASO DE USO GESTIONA CAPÍTULOS POR NIVEL**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	El administrador debe iniciar sesión.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ir a la configuración de capítulos</li> <li>2. Agregar los capítulos de acuerdo al nivel</li> <li>3. Enviar a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si el número de capítulos por nivel es impar se muestra un mensaje de error para corregir</li> </ol>
<b>Pos condiciones</b>	Los capítulos para poder crear los exámenes adecuadamente quedan configurados

**Tabla 2.15 Diccionario del Caso de Uso Gestiona Capítulos por Nivel**

**2.2.1.1.2.10 CASO DE USO GESTIONA CLASES:** Permite asignar a los profesores a sus respectivos niveles.

**a) DICCIONARIO DEL CASO DE USO GESTIONA CLASES.**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	El administrador debe iniciar sesión. El usuario profesor ya debe estar creado
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ingresar al manejo de classrooms</li> <li>2. Configurar los parámetros necesarios para asignar la clase a un determinado profesor</li> <li>3. Enviar a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2 Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir</li> <li>3. Una vez corregidos los datos se proceden a guardar</li> </ol>
<b>Pos condiciones</b>	Los profesores tienen asignados sus niveles respectivos.

**Tabla 2.16 Diccionario del Caso de Uso Gestiona Clases**

**2.2.1.1.2.11 CASO DE USO GESTIONA FACULTADES:** Configura las facultades que tiene la universidad y que son necesarias para determinar los niveles de inglés que un estudiante debe aprobar.

**a) DICCIONARIO DEL CASO DE USO GESTIONA FACULTADES.**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	El administrador debe iniciar sesión.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ir a la configuración de facultades</li> <li>2. Agregar las facultades que posee la Universidad</li> <li>3. Enviar a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir.</li> <li>3. Una vez corregidos los datos se proceden a guardar</li> </ol>
<b>Pos condiciones</b>	Quedan asignadas las facultades de la universidad para el uso del sistema.

**Tabla 2.17 Diccionario del Caso de Uso Gestiona Facultades.**

**2.2.1.1.2.12 CASO DE USO GESTIONA GRUPO DE PREGUNTAS:** Permite configurar las preguntas de acuerdo a un grupo determinado por el nivel, la facultad, la unidad, el tópico y modalidad para que estén disponibles para un determinado test.

**a) DICCIONARIO DEL CASO DE USO GESTIONA GRUPO DE PREGUNTAS**

<b>Actores</b>	Editor
<b>Precondiciones</b>	El editor debe iniciar sesión
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El editor presiona insertar</li> <li>2. El editor configura el nivel, la facultad, la unidad, el tópico, descripción, modalidad a la que pertenece la pregunta.</li> <li>3. Activa o desactiva una pregunta para que este disponible para el test.</li> <li>4. Guarda la pregunta</li> </ol>
<b>Flujo alternativo</b>	
<b>Pos condiciones</b>	La pregunta pertenece a un grupo y se le puede ingresar las opciones de respuesta respectiva.

**Tabla 2.18 Diccionario del Caso de Uso Gestiona Grupo de Preguntas**

**2.2.1.1.2.13 CASO DE USO GESTIONA NIVELES:** Permite asignar los niveles con que cuenta el Departamento de Idiomas.

**a) DICCIONARIO DEL CASO DE USO GESTIONA NIVELES**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	El administrador debe iniciar sesión
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ingresar al manejo de niveles</li> <li>2. Configurar los parámetros necesarios para asignar los niveles con los que cuenta el Departamento de Idiomas</li> <li>3. Enviar a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir</li> <li>3. Una vez corregidos los datos se procede a guardar</li> </ol>
<b>Pos condiciones</b>	Quedan asignados los niveles de aprobación en el sistema.

**Tabla 2.19 Diccionario del Caso de Uso Gestiona Niveles**

**2.2.1.1.2.14 CASO DE USO GESTIONA PREGUNTAS:** Permite relacionar las opciones de respuesta a su respectiva pregunta, además permite activar o desactivar las preguntas para que estén disponibles para un test.

**a) DICCIONARIO DEL CASO DE USO GESTIONA PREGUNTAS**

<b>Actores</b>	Editor
<b>Precondiciones</b>	<p>El editor debe iniciar sesión</p> <p>La pregunta debe estar relacionada a un grupo de preguntas.</p>
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El editor presiona insertar</li> <li>2. Relaciona la pregunta al grupo al que pertenece</li> <li>3. Coloca las opciones de respuestas para las preguntas</li> <li>4. Activa o desactiva la pregunta para que este disponible para el test</li> <li>5. Guarda la pregunta</li> </ol>
<b>Flujo alternativo</b>	
<b>Pos condiciones</b>	La pregunta esta relacionada al su correspondiente grupo y con sus respectivas opciones.

**Tabla 2.20 Diccionario del Caso de Uso Gestiona Preguntas**

**2.2.1.1.2.15 CASO DE USO GESTIONA TÓPICOS:** Configura los tópicos o secciones que posee el test.

a) **DICCIONARIO DEL CASO DE USO GESTIONA TÓPICOS.**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	El administrador debe iniciar sesión
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ir a la configuración de tópicos</li> <li>2. Configurar los parámetros necesarios para agregar los tópicos</li> <li>3. Enviar a la DB</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si los datos no cumplen con las reglas de validación, el sistema muestra mensajes para corregir</li> <li>3. Una vez corregidos los datos se procede a guardar.</li> </ol>
<b>Pos condiciones</b>	Quedan configuradas las secciones de que constara el examen.

**Tabla 2.21 Diccionario del Caso de Uso Gestiona Tópicos.**

**2.2.1.1.2.16 CASO DE USO PARÁMETROS DE LAS PREGUNTAS:** Gestiona el nivel y el número de preguntas de que consta el examen.

a) **DICCIONARIO DEL CASO DE USO PARÁMETROS DE LAS PREGUNTAS.**

<b>Actores</b>	Administrador
<b>Precondiciones</b>	<p>El administrador debe iniciar sesión</p> <p>Se deben haber configurado los niveles</p> <p>Se deben haber con figurado los tópicos</p>
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ingresar al manejo de parámetros</li> <li>2. Configurar los campos necesarios para cada nivel</li> <li>3. Guardar</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si no cumple las reglas de validación ,se muestra mensajes para corregir</li> <li>3.Una vez corregidos los datos se procede a guardar</li> </ol>
<b>Pos condiciones</b>	Queda configurado los exámenes con un determinado número de preguntas

**Tabla 2.22 Diccionario del Caso de Uso Parámetros de las Preguntas**

**2.2.1.1.2.17 CASO DE USO RECUPERAR CONTRASEÑA:** Permite recuperar la contraseña del usuario a través de una pregunta de seguridad.

a) **DICCIONARIO DEL CASO DE USO RECUPERAR CONTRASEÑA.**

<b>Actores</b>	Administrador, Profesor, Editor, Estudiante
<b>Precondiciones</b>	El usuario no recuerda su clave de acceso
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a recuperar contraseña</li> <li>2. Contesta la pregunta de seguridad</li> <li>3. Si contesta correctamente se envía una notificación a su correo</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>3. Si no es correcta la información no se envía ninguna notificación</li> </ol>
<b>Pos condiciones</b>	El usuario recupera su contraseña.

**Tabla 2.23 Diccionario del Caso de Uso Recuperar Contraseña.**

**2.2.1.1.2.18 CASO DE USO REVISAR NOTAS:** Permite consultar las notas tanto del examen como de los demás aportes necesarios para aprobar el nivel.

a) **DICCIONARIO DEL CASO DE USO REVISAR NOTAS.**

<b>Actores</b>	Estudiantes
<b>Precondiciones</b>	El estudiante debe iniciar sesión El profesor debe haber ingresado notas o el estudiante debe haber rendido el examen para que posea una nota.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ingresar a Grades</li> <li>2. Revisar las notas</li> <li>3. Salir</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>2. Si no posee ninguna nota, se muestra un cuadro vacío</li> </ol>
<b>Pos condiciones</b>	Se observa las notas correspondientes al usuario.

**Tabla 2.24 Diccionario del Caso de Uso Revisar Notas.**

**2.2.1.1.2.19 CASO DE USO TOMAR EL EXAMEN:** El examen con sus respectivas secciones es presentado al estudiante para que pueda rendirlo.

**a) DICCIONARIO DEL CASO DE USO TOMAR EL EXAMEN**

<b>Actores</b>	Estudiantes
<b>Precondiciones</b>	El estudiante debe iniciar sesión Se debe haber generado un examen por parte del profesor
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. Ir a take test</li> <li>2. Responder a las preguntas de cada sección</li> <li>3. Presionar Finish Test para terminar el examen y conocer su nota.</li> </ol>
<b>Flujo alternativo</b>	3. Si se termina el tiempo se califica automáticamente el examen
<b>Pos condiciones</b>	El estudiante realizo su examen y conoce su nota.

**Tabla 2.25 Diccionario del Caso de Uso Tomar el Examen.**

## 2.2.1.2 DIAGRAMA ENTIDAD RELACIÓN

### MODELO DE DATOS.

Muestra los datos relevantes para el sistema, agrupados en entidades y sus relaciones.

#### 2.2.1.2.1 LISTA DE ENTIDADES.

Las entidades que forman parte de la estructura de la base de datos son:

Name
UserProfiles
Classrooms
StudentProfiles
Levels
Faculties
Units
Topics
Tests
Questions
QuestionGroups
Presential_Mode_Students
Intensive_Mode_Students
Distance_Mode_Students
ParametersQuestions
Listening
AdministratorProfiles

**Tabla 2.26 Lista de Entidades.**

**Ver diagrama Entidad – Relación en el anexo C.**

**2.2.1.2.1.1 ENTIDAD ADMINISTRATORPROFILES:** Esta tabla contiene los datos de los usuarios Administradores del sistema.

**a) ATRIBUTOS DE LA ENTIDAD ADMINISTRATORPROFILES.**

<b>Nombre</b>	<b>Tipo de Dato</b>	<b>Obligatorio</b>
UserName	Variable characters (256)	X
Names	Characters (64)	X
LastNames	Characters (64)	X
Photo	Characters (32)	X
IdentityCard	Characters (16)	X
BloodType	Characters (16)	X
Gender	Characters (8)	X
Birthday	Date & Time	
FatherName	Characters (128)	
MotherName	Characters (128)	
TransversalStreet	Characters (64)	
PrincipalStreet	Characters (64)	
Number	Characters (8)	
Country	Characters (16)	
City	Characters (16)	
Phone	Characters (16)	
CellPhone	Characters (16)	
MaritalStatus	Characters (16)	

**Tabla 2.27 Atributos de la Entidad AdministratorProfiles.**

**2.2.1.2.1.2 ENTIDAD CLASSROOMS:** Esta tabla contiene la información de la asignación del usuario Profesor a las diferentes clases que le corresponde dictar. Classroom permite crear un curso al que se le asigna un profesor aunque en su mayoría tiene un atributo para ser un curso general puede establecerse un curso asociado a una facultad a fin de presentar clases en Inglés técnico. Una de las propiedades importantes de Classroom es la de asignar la modalidad en que se va a dictar el mismo así que tanto profesores, estudiantes y editores dispondrán de opciones distintas de acuerdo a los requerimientos de la modalidad.

**a) ATRIBUTOS DE LA ENTIDAD CLASSROOMS.**

<b>Nombre</b>	<b>Tipo de dato</b>	<b>Obligatorio</b>
ClassroomId	Integer	X
ClassNumber	Characters (2)	X
ClassModality	Characters (16)	X
UserNameTeacher	Variable characters (256)	X

**Tabla 2.28 Atributos de la Entidad Classroom.**

**2.2.1.2.1.3 ENTIDAD DISTANCE\_MODE\_STUDENTS:** Esta tabla almacena los datos de las calificaciones necesarias para aprobar un nivel de inglés de los estudiantes de la modalidad distancia.

**a) ATRIBUTOS DE LA ENTIDAD DISTANCE\_MODE\_STUDENTS.**

Nombre	Tipo de Dato	Obligatorio
DistanceGradeId	Integer	X
DPartial11	Decimal (4,2)	
DExam1	Decimal (4,2)	
DPartial21	Decimal (4,2)	
DExam2	Decimal (4,2)	
DPartial31	Decimal (4,2)	
DExam3	Decimal (4,2)	
DPartial41	Decimal (4,2)	
DExam4	Decimal (4,2)	
CompleteName	Variable characters (128)	X

**Tabla 2.29 Atributos de la Entidad Distance\_Mode\_Students.**

**2.2.1.2.1.4 ENTIDAD FACULTIES:** Esta tabla almacena los datos de las facultades que pertenecen a la Universidad y determinan que niveles y que modalidades deben tomar los estudiantes.

**a) ATRIBUTOS DE LA ENTIDAD FACULTIES.**

Nombre	Tipo de Dato	Obligatorio
FacultyId	Integer	X
FacultyName	Characters (128)	X
FacultyDescription	Characters (256)	

**Tabla 2.30 Atributos de la Entidad Faculties.**

**2.2.1.2.1.5 ENTIDAD INTENSIVE\_MODE\_STUDENTS:** Esta tabla almacena los datos de las calificaciones necesarias para aprobar un nivel de inglés de los estudiantes de la modalidad intensiva.

**a) ATRIBUTOS DE LA ENTIDAD INTENSIVE\_MODE\_STUDENTS**

Nombre	Tipo de Dato	Obligatorio
IntensiveGradeId	Integer	X
IPartial11	Decimal (4,2)	
IPartial12	Decimal (4,2)	
IPartial13	Decimal (4,2)	
IExam1	Decimal (4,2)	
IWriting1	Decimal (4,2)	
IPartial21	Decimal (4,2)	
IPartial22	Decimal (4,2)	
IPartial23	Decimal (4,2)	
IExam2	Decimal (4,2)	
IWriting2	Decimal (4,2)	
CompleteName	Variable characters (128)	X

**Tabla 2.31 Atributos de la Entidad Intensive\_Mode\_Student**

**2.2.1.2.1.6 ENTIDAD LEVELS:** Esta tabla contiene los datos de los niveles que el usuario Estudiante debe tomar de acuerdo a la Facultad a la que pertenece.

**a) ATRIBUTOS DE LA ENTIDAD LEVELS**

Nombre	Tipo de Dato	Obligatorio
LevelId	Integer	X
LevelNumber	Integer	X
LevelName	Characters (64)	X
LevelLastChange	Date & Time	X
LevelDescription	Text	

**Tabla 2.32 Atributos de la Entidad Levels.**

**2.2.1.2.1.7 ENTIDAD LISTENING:** Esta tabla almacena los archivos de audio asociados a sus QuestionGroups que se utilizaran en las preguntas de la sección Listening.

**a) ATRIBUTOS DE LA ENTIDAD LISTENING.**

Nombre	Tipo de Dato	Obligatorio
ListeningId	Int	X
ListeningURL	Characters (128)	X

**Tabla 2.33 Atributos de la Entidad Listening**

**2.2.1.2.1.8 ENTIDAD PARAMETERSQUESTIONS:** Esta tabla almacena parámetros necesarios para configurar el test a ser rendido por los estudiantes, sobre todo en la configuración de Question Groups para el número de preguntas en ese grupo. Se debe tener en cuenta que Question Group debe tener como mínimo el número de preguntas que resulta de dividir el número de QuestionGroup para el de Questions.

**a) ATRIBUTOS DE LA ENTIDAD PARAMETERSQUESTIONS**

Nombre	Tipo de Dato	Obligatorio
ParameterId	Integer	X
NumberQuestionGroups	Integer	X
NumberQuestions	Integer	X
LevelTopic	Characters (4)	X

**Tabla 2.34 Atributos de la Entidad ParameterQuestions.**

**2.2.1.2.1.9 ENTIDAD PRESENTIAL\_MODE\_STUDENTS:** Esta tabla almacena los datos de las calificaciones necesarias para aprobar un nivel de inglés de los estudiantes de la modalidad presencial.

**a) ATRIBUTOS DE LA ENTIDAD PRESENTIAL\_MODE\_STUDENTS**

Nombre	Tipo de Dato	Obligatorio
PresentialGradeId	Integer	X
PPartial11	Decimal (4,2)	
PPartial12	Decimal (4,2)	
PPartial13	Decimal (4,2)	
PExam1	Decimal (4,2)	
PWriting1	Decimal (4,2)	
PPartial21	Decimal (4,2)	
PPartial22	Decimal (4,2)	
PPartial23	Decimal (4,2)	
PExam2	Decimal (4,2)	
PWriting2	Decimal (4,2)	
CompleteName	Variable characters (128)	X

**Tabla 2.35 Atributos de la Entidad Presential\_Mode\_Students.**

**2.2.1.2.1.10 ENTIDAD QUESTIONGROUPS:** Esta tabla contiene los datos que permiten agrupar a una pregunta de acuerdo a parámetros como la modalidad, level, faculty, unit, topic y description. Esta tabla tiene un campo a fin de permitir

que una pregunta esté o no activa para ser tomada en cuenta por el sistema de selección de preguntas para el test.

**a) ATRIBUTOS DE LA ENTIDAD QUESTIONGROUPS**

Nombre	Tipo de Dato	Obligatorio
QuestionGroupId	Integer	X
QuestionGroupDescription	Text	X
IsPresentialMode	Bolean	X
IsIntensiveMode	Bolean	X
IsDistanceMode	Bolean	X
Activated	Bolean	X

**Tabla 2.36 Atributos de la Entidad QuestionGroups.**

**2.2.1.2.1.11 ENTIDAD QUESTIONS:** Esta tabla almacena las preguntas y opciones de respuesta de las mismas, acorde con el grupo a que pertenecen. Y la opción de que se encuentre o no activa para ser seleccionada por el sistema a fin de presentarse como opción válida para el test.

**a) ATRIBUTOS DE LA ENTIDAD QUESTIONS**

Nombre	Tipo de Dato	Obligatorio
QuestionId	Integer	X
QuestionDescription	Text	X
Options	Text	X
QuestionActivated	Bolean	X

**Tabla 2.37 Atributos de la Entidad Questions.**

**2.2.1.2.1.12 ENTIDAD STUDENTPROFILES:** Esta tabla contiene los datos de los usuarios Estudiantes del sistema.

**a) ATRIBUTOS DE LA ENTIDAD STUDENTPROFILES**

Nombre	Tipo de Dato	Obligatorio
UserName	Variable characters (256)	X
Semestre	Integer	X
Names	Characters (64)	X
LastNames	Characters (64)	X
Photo	Characters (32)	X
IdentityCard	Characters (16)	X
BloodType	Characters (16)	X
Gender	Characters (8)	X

BirthDay	Date & Time	
FatherName	Characters (128)	
MotherName	Characters (128)	
TransversalStreet	Characters (64)	
PrincipalStreet	Characters (64)	
Number	Characters (8)	
Country	Characters (16)	
City	Characters (16)	
RegistrationNumber	Characters (16)	
Phone	Characters (16)	
CellPhone	Characters (16)	
MaritalStatus	Characters (16)	
InTest	Bolean	

**Tabla 2.38 Atributos de la Entidad StudentProfiles.**

**2.2.1.2.1.13 ENTIDAD TESTS:** Esta tabla contiene la información sobre el test que se crea en base al grupo de preguntas, el tiempo, día de inicio, etc.

**a) ATRIBUTOS DE LA ENTIDAD TESTS**

Nombre	Tipo de Dato	Obligatorio
TestId	Integer	X
StartDate	Date & Time	X
Partial	Short integer	X
Time	Integer	X
Unblocked	Bolean	X
GrammarQuestionGroupArray	Text	
ListeningQuestionGroupArray	Text	
VocabularyQuestionGroupArray	Text	
ReadingQuestionGroupArray	Text	
GrammarQuestionArray	Text	
ListeningQuestionArray	Text	
VocabularyQuestionArray	Text	
ReadingQuestionArray	Text	
GrammarOptionArray	Text	
ListeningOptionArray	Text	
VocabularyOptionArray	Text	
ReadingOptionArray	Text	

**Tabla 2.39 Atributos de la Entidad Test.**

**2.2.1.2.1.14 ENTIDAD TOPICS:** Esta tabla almacena las secciones que se compone el test que se genera para los estudiantes. Es una tabla que almacena

los temarios al que se deben someter a prueba los estudiantes: Grammar, Listening, Reading, Vocabulary y Writing.

**a) ATRIBUTOS DE LA ENTIDAD TOPICS**

Nombre	Tipo de Dato	Obligatorio
TopicId	Integer	X
TopicName	Characters (128)	X
TopicDescription	Short integer	

**Tabla 2.40 Atributos de la Entidad Topic.**

**2.2.1.2.1.15 ENTIDAD UNITS:** Esta tabla almacena las unidades que corresponden a cada nivel y determinan los exámenes parciales que se toman en el sistema.

**a) ATRIBUTOS DE LA ENTIDAD UNITS**

Nombre	Tipo de Dato	Obligatorio
UnitId	Integer	X
UnitChapter	Integer	X
UnitName	Characters (128)	X
UnitDescription	Text	

**Tabla 2.41 Atributos de la Entidad Units.**

**2.2.1.2.1.16 ENTIDAD USERPROFILES:** Esta tabla contiene los datos de los usuarios Profesores/Editores del sistema.

**a) ATRIBUTOS DE LA ENTIDAD USERPROFILES**

Nombre	Tipo de Datos	Obligatorio
UserName	Variable characters (256)	X
Names	Characters (64)	X
LastNames	Characters (64)	X
Photo	Characters (32)	X
IdentityCard	Characters (16)	X
BloodType	Characters (16)	X
Gender	Characters (8)	X
Birthday	Date & Time	
FatherName	Characters (128)	
MotherName	Characters (128)	
TransversalStreet	Characters (64)	
PrincipalStreet	Characters (64)	
Number	Characters (8)	

Country	Characters (16)	
City	Characters (16)	
Phone	Characters (16)	
CellPhone	Characters (16)	
MaritalStatus	Characters (16)	

**Tabla 2.42 Atributos de la Entidad UserProfiles.**

### 2.2.1.2.2 LISTA DE RELACIONES.

Nombre	Entidad 2	Entidad 1	Entidad 1 -> Entidad 2 Cardinalidad	Entidad 2 -> Entidad 1 Cardinalidad
inscribe to	StudentProfiles	Classrooms	1,n	1,1
Has class	Classrooms	Levels	1,n	1,1
teach technical	Classrooms	Faculties	1,n	1,1
Teach	Classrooms	UserProfiles	1,n	1,1
is composed of	Units	Levels	1,n	1,1
Test	Tests	StudentProfiles	1,1	1,1
Has questions	Questions	QuestionGroups	1,n	1,1
Contains	QuestionGroups	Levels	1,n	1,1
faculties has	QuestionGroups	Faculties	1,n	1,1
unit has	QuestionGroups	Units	1,n	1,1
topic has	QuestionGroups	Topics	1,n	1,1
Presential student takes	Presential_Mode_Students	StudentProfiles	1,n	1,1
intensive student takes	Intensive_Mode_Students	StudentProfiles	1,n	1,1
distance student takes	Distance_Mode_Students	StudentProfiles	1,n	1,1
Formed by	ParametersQuestions	Topics	1,n	1,1
level has parameters	ParametersQuestions	Levels	1,n	1,1
Sounds	Listening	QuestionGroups	0,n	1,1

**Tabla 2.43 Lista de Relaciones.**

### 2.2.1.3 DIAGRAMA DE CLASES

Muestra el conjunto de clases que controlan la capa del modelo y permiten una interfaz transparente para el programador.

#### 2.2.1.3.1 LISTA DE CLASES

Nombre	Visibilidad
UserProfiles	Public
Classrooms	Public
StudentProfiles	Public
Levels	Public
Faculties	Public
Units	Public
Topics	Public
Tests	Public
Questions	Public
QuestionGroups	Public
Presential_Mode_Students	Public
Intensive_Mode_Students	Public
Distance_Mode_Students	Public
Chapters	Public
ParametersQuestions	Public
Listening	Public
AdministratorProfiles	Public
LevelClassroom	Public
QuestionInGroupQuestions	Public
Aspnet_Roles	Public
Aspnet_Users	Public
Aspnet_UserInRoles	Public

**Tabla 2.44 Lista de Clases.**

**Ver Diagrama de Clases en el Anexo E.**

**2.2.1.3.1.1 CLASE USERPROFILES:** Controla la tabla UserProfiles en la DB.

#### a) LISTA DE ATRIBUTOS DE LA CLASE USERPROFILES

Nombre	Tipo de Dato	Visibilidad
UserName	String	Public
Names	String	Public

LastNames	String	Public
Photo	String	Public
IdentityCard	String	Public
BloodType	String	Public
Gender	String	Public
Birthday	DateTime	Public
FatherName	String	Public
MotherName	String	Public
TransversalStreet	String	Public
PrincipalStreet	String	Public
Number	String	Public
Country	String	Public
City	String	Public
Phone	String	Public
CellPhone	String	Public
MaritalStatus	String	Public

**Tabla 2.45 Lista de Atributos de la clase UserProfiles.**

**b) LISTA DE OPERACIONES DE LA CLASE USERPROFILES**

Name	Tipo de retorno	Visibilidad
GetUserProfiles	Void	Public
GetUserprofileByUserName	Void	Public

**Tabla 2.46 Lista de Operaciones de la clase UserProfiles**

**2.2.1.3.1.2 CLASE CLASSROOMS: Controla la tabla Classrooms en la DB.**

**a) LISTA DE ATRIBUTOS DE LA CLASE CLASSROOMS**

Nombre	Tipo de Dato	Visibilidad
ClassroomId	Int	Public
ClassNumber	String	Public
ClassModality	String	Public
UserNameTeacher	String	Public

**Tabla 2.47 Lista de Atributos de la clase Classrooms**

**b) LISTA DE OPERACIONES DE LA CLASE CLASSROOMS**

Name	Tipo de retorno	Visibilidad
GetClassrooms	Object	Public
GetClassroomByClassroomId	Object	Public
GetClassroomIdByTeacherUserName	Object	Public

**Tabla 2.48 Lista de Operaciones de la clase Classroom.**

**2.2.1.3.1.3 CLASE STUDENTPROFILES:** Controla la tabla StudentProfiles en la DB.

a) **LISTA DE ATRIBUTOS DE LA CLASE STUDENTPROFILES.**

<b>Nombre</b>	<b>Tipo de Dato</b>	<b>Visibilidad</b>
UserName	String	Public
Semestre	Int	Public
Names	String	Public
LastNames	String	Public
Photo	String	Public
IdentityCard	String	Public
BloodType	String	Public
Gender	String	Public
BirthDay	DateTime	Public
FatherName	String	Public
MotherName	String	Public
TransversalStreet	String	Public
PrincipalStreet	String	Public
Number	String	Public
Country	String	Public
City	String	Public
RegistrationNumber	String	Public
Phone	String	Public
CellPhone	String	Public
MaritalStatus	String	Public
InTest	Bool	Public

**Tabla 2.49 Lista de Atributos de la clase StudentProfiles**

b) **LISTA DE OPERACIONES DE LA CLASE STUDENTPROFILES.**

<b>Nombre</b>	<b>Tipo de retorno</b>	<b>Visibilidad</b>
GetStudentProfiles	Object	Public
GetStudentProfilesByClassroomId	Object	Public
GetStudentProfilesByUserName	Object	Public
StudentInTest	Bool	Public
UpdateInTest	Bool	Public

**Tabla 2.50 Lista de Operaciones de la clase StudentProfiles**

**2.2.1.3.1.4 CLASE LEVEL:** Controla la tabla Level en la DB.

## a) LISTA DE ATRIBUTOS DE LA CLASE LEVELS

Nombre	Tipo de Dato	Visibilidad
LevelId	Int	Public
LevelNumber	Int	Public
LevelName	String	Public
LevelLastChange	DateTime	Public
LevelDescription	String	Public

Tabla 2.51 Lista de Atributos de la clase Levels

## b) LISTA DE OPERACIONES DE LA CLASE LEVELS

Nombre	Tipo de retorno	Visibilidad
GetLevels	Void	Public
GetMaxLevelID	Int	Public

Tabla 2.52 Lista de Operaciones de la clase Levels

## 2.2.1.3.1.5 CLASE FACULTIES: Controla la tabla Faculties en la DB

## a) LISTA DE ATRIBUTOS DE LA CLASE FACULTIES

Nombre	Tipo de Dato	Visibilidad
FacultyId	Int	Public
FacultyName	String	Public
FacultyDescription	String	Public

Tabla 2.53 Lista de Atributos de la clase Faculties

## b) LISTA DE OPERACIONES DE LA CLASE FACULTIES

Nombre	Tipo de retorno	Visibilidad
GetFaculties	Object	Public

Tabla 2.54 Lista de Operaciones de la clase Faculties

## 2.2.1.3.1.6 CLASE UNITS: Controla la tabla Units en la DB.

## a) LISTA DE ATRIBUTOS DE LA CLASE UNITS

Nombre	Tipo de Dato	Visibilidad
UnitId	Int	Public
UnitChapter	Int	Public
UnitName	String	Public
UnitDescription	String	Public

Tabla 2.55 Lista de Atributos de la clase Units.

**b) LISTA DE OPERACIONES DE LA CLASE UNITS**

Nombre	Tipo de retorno	Visibilidad
GetUnits	Object	Public
GetUnitsByLevelChapter	Object	Public
GetUnitsByLevelID	Object	Public
GetCountByLevelID	Int	Public
GetMaxUnitChapterByLevel	Int	Public
GetMaxUnitId	Int	Public
GetMinUnitChapterByLevel	Int	Public

**Tabla 2.56 Lista de Operaciones de la clase Units**

**2.2.1.3.1.7 CLASE TOPICS: Controla la tabla Topics en la DB.**

**a) LISTA DE ATRIBUTOS DE LA CLASE TOPICS**

Nombre	Tipo de Dato	Visibilidad
TopicId	Int	Public
TopicName	String	Public
TopicDescription	Short	Public

**Tabla 2.57 Lista de Atributos de la clase Topics**

**b) LISTA DE OPERACIONES DE LA CLASE TOPICS**

Nombre	Tipo de retorno	Visibilidad
GetTopics	Object	Public
GetMaxTopicId	Int	Public

**Tabla 2.58 Lista de Operaciones de la clase Topics.**

**2.2.1.3.1.8 CLASE TESTS: Controla la tabla Test en la DB.**

**a) LISTA DE ATRIBUTOS DE LA CLASE TESTS**

Nombre	Tipo de Dato	Visibilidad
TestId	Int	Public
StartDate	DateTime	Public
Partial	Short	Public
Time	Int	Public
Unbloqued	Bool	Public
GrammarQuestionGroupArray	String	Public
ListeningQuestionGroupArray	String	Public
VocabularyQuestionGroupArray	String	Public
ReadingQuestionGroupArray	String	Public
GrammarQuestionArray	String	Public
ListeningQuestionArray	String	Public
VocabularyQuestionArray	String	Public

ReadingQuestionArray	String	Public
GrammarOptionArray	String	Public
ListeningOptionArray	String	Public
VocabularyOptionArray	String	Public
ReadingOptionArray	String	Public

**Tabla 2.59 Lista de Atributos de la clase Test**

**b) LISTA DE OPERACIONES DE LA CLASE TESTS**

Nombre	Tipo de retorno	Visibilidad
GetTest	Object	Public
DeleteTestByUserName	Void	Public
GetTestByStudentUserName	Object	Public
UpdateGrammarOptions	Void	Public
UpdateListeningOptions	Void	Public
UpdateReadingOptions	Void	Public
UpdateVocabularyOptions	Void	Public
UpdateTimeByUnblockedUserName	Void	Public

**Tabla 2.60 Lista de Operaciones de la clase Test**

**2.2.1.3.1.9 CLASE QUESTIONS:** Controla la tabla Qestions en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE QUESTIONS**

Nombre	Tipo de Dato	Visibilidad
QuestionId	Int	Public
QuestionDescription	String	Public
Options	String	Public
QuestionActivated	Bool	Public

**Tabla 2.61 Lista de Atributos de la clase Questions**

**b) LISTA DE OPERACIONES DE LA CLASE QUESTIONS**

Nombre	Tipo de retorno	Visibilidad
GetQuestions	Void	Public
GetQuestionsByQuestionGroupId	Void	Public
GetQuestionsByQuestionGroupIdQuestionId	Void	Public
GetQuestionsByQuestionId	Void	Public

**Tabla 2.62 Lista de Operaciones de la clase Questions**

**2.2.1.3.1.10 CLASE QUESTIONGROUPS:** Controla la tabla QuestionGroups en la DB.

a) **LISTA DE ATRIBUTOS DE LA CLASE QUESTIONGROUPS.**

Nombre	Tipo de Dato	Visibilidad
QuestionGroupId	Int	Public
QuestionGroupDescription	String	Public
IsPresentialMode	Bool	Public
IsIntensiveMode	Bool	Public
IsDistanceMode	Bool	Public
Activated	Bool	Public

**Tabla 2.63 Lista de Atributos de la clase QuestionGroups**

b) **LISTA DE OPERACIONES DE LA CLASE QUESTIONGROUPS.**

Nombre	Tipo de retorno	Visibilidad
GetQuestionGroups	Object	Public
GetQuestionGroupsByQuestionGropupId	Object	public
GetQuestionGroupsByFacultyLevelTopicUnit	Object	public
GetDistanceQuestionGroupsByFacultyLevelTopicUnit	Object	public
GetIntensiveQuestionGroupsByFacultyLevelTopicUnit	Object	public
GetPresentialQuestionGroupsByFacultyLevelTopicUnit	Object	public

**Tabla 2.64 Lista de Operaciones de la clase QuestionGroups.**

**2.2.1.3.1.11 CLASE PRESENTIAL\_MODE\_STUDENTS:** Controla la tabla Presential\_Mode\_Students en la DB.

a) **LISTA DE ATRIBUTOS DE LA CLASE PRESENTIAL\_MODE\_STUDENTS.**

Nombre	Tipo de Dato	Visibilidad
PresentialGradeId	Int	public
PPartial11	Decimal	public
PPartial12	Decimal	public
PPartial13	Decimal	public
PExam1	Decimal	public
PWriting1	Decimal	public
PPartial21	Decimal	public
PPartial22	Decimal	public
PPartial23	Decimal	public
PExam2	Decimal	public
PWriting2	Decimal	public
CompleteName	String	public

**Tabla 2.65 Lista de Atributos de la clase Presential\_Mode\_Students..**

**b) LISTA DE OPERACIONES DE LA CLASE PRESENTIAL\_MODE\_STUDENTS**

Nombre	Tipo de retorno	Visibilidad
GetPresentialGrades	Object	Public
GetPresentialGradesByClassroomId	Object	Public
GetPresentialGradesByUserName	Object	Public
GetPresentialGradesByUserNameClassroomId	Object	Public
UpdatePExam1ByUserNameClassroomId	Void	Public
UpdatePExam2ByUserNameClassroomId	Void	Public
UpdatePPartialGrades	Void	Public

**Tabla 2.66 Lista de Operaciones de la clase Presential\_Mode\_Students.**

**2.2.1.3.1.12 CLASE INTENSIVE\_MODE\_STUDENTS:** Controla la tabla Intensive\_Mode\_Students en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE INTENSIVE\_MODE\_STUDENTS**

Nombre	Tipo de Dato	Visibilidad
IntensiveGradeId	Int	Public
IPartial11	Decimal	Public
IPartial12	Decimal	Public
IPartial13	Decimal	Public
IEexam1	Decimal	Public
IWriting1	Decimal	Public
IPartial21	Decimal	Public
IPartial22	Decimal	Public
IPartial23	Decimal	Public
IEexam2	Decimal	Public
IWriting2	Decimal	Public
CompleteName	String	Public

**Tabla 2.67 Lista de Atributos de la clase Intensive\_Mode\_Students**

**b) LISTA DE OPERACIONES DE LA CLASE INTENSIVE\_MODE\_STUDENTS**

Nombre	Tipo de retorno	Visibilidad
GetIntensiveGrades	Object	public
GetIntensiveGradesByClassroomId	Object	public
GetIntensiveGradesByUserName	Object	public
GetIntensiveGradesByUserNameClassroomId	Object	public
UpdateIExam1ByUserNameClassroomId	Void	public
UpdateIExam2ByUserNameClassroomId	Void	public

**Tabla 2.68 Lista de Operaciones de la clase Intensive\_Mode\_Students.**

**2.2.1.3.1.13 CLASE DISTANCE\_MODE\_STUDENTS:** Controla la tabla Distance\_Mode\_Students en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE DISTANCE\_MODE\_STUDENTS**

Nombre	Tipo de Dato	Visibilidad
DistanceGradeId	Int	Public
DPartial11	Decimal	Public
DExam1	Decimal	Public
DPartial21	Decimal	Public
DExam2	Decimal	Public
DPartial31	Decimal	Public
DExam3	Decimal	Public
DPartial41	Decimal	Public
DExam4	Decimal	Public
CompleteName	String	Public

**Tabla 2.69 Lista de Atributos de la clase Distance\_Mode\_Students.**

**b) LISTA DE OPERACIONES DE LA CLASE DISTANCE\_MODE\_STUDENTS**

Nombre	Tipo de retorno	Visibilidad
GetDistanceGrades	Object	Public
GetDistanceGradesByClassroomId	Object	Public
GetDistanceGradesByUserNameClassroomId	Object	Public
UpdateDExam1ByUserNameClassroomId	Void	Public
UpdateDExam2ByUserNameClassroomId	Void	Public
UpdateDExam3ByUserNameClassroomId	Void	Public
UpdateDExam4ByUserNameClassroomId	Void	Public
UpdateDPartialGrades	Void	Public

**Tabla 2.70 Lista de Operaciones de la clase Distance\_Mode\_Students.**

**2.2.1.3.1.14 CLASE PARAMETERSQUESTIONS:** Controla la tabla ParametersQuestions en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE PARAMETERSQUESTIONS**

Nombre	Tipo de Dato	Visibilidad
ParameterId	Int	Public
NumberQuestionGroups	Int	Public
NumberQuestions	Int	Public
LevelTopic	String	Public

**Tabla 2.71 Lista de Atributos de la clase ParametersQuestions**

**b) LISTA DE OPERACIONES DE LA CLASE PARAMETERSQUESTIONS**

Nombre	Tipo de retorno	Visibilidad
GetParametersQuestions	Object	Public
GetParametersQuestionsByLevel	Object	Public
GetParametersQuestionsByLevelTopic	Object	Public
GetMaxParameterId	Int	Public

**Tabla 2.72 Lista de Operaciones de la clase ParametersQuestions.**

**2.2.1.3.1.15 CLASE LISTENING:** Controla la tabla Listening en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE LISTENING**

Nombre	Tipo de Dato	Visibilidad
ListeningId	Int	Public
ListeningURL	String	Public

**Tabla 2.73 Lista de Atributos de la clase Listening**

**2.2.1.3.1.16 CLASE ADMINISTRATORPROFILES:** Controla la tabla AdministratorProfiles en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE ADMINISTRATORPROFILES**

Nombre	Tipo de Dato	Visibilidad
UserName	String	Public
Names	String	Public
LastNames	String	Public
Photo	String	Public
IdentityCard	String	Public
BloodType	String	Public
Gender	String	Public
Birthday	DateTime	Public
FatherName	String	Public
MotherName	String	Public
TransversalStreet	String	Public
PrincipalStreet	String	Public
Number	String	Public
Country	String	Public
City	String	Public
Phone	String	Public
CellPhone	String	Public
MaritalStatus	String	Public

**Tabla 2.74 Lista de Atributos de la clase AdministratorProfiles.**

**b) LISTA DE OPERACIONES DE LA CLASE ADMINISTRATORPROFILES**

Nombre	Tipo de retorno	Visibilidad
GetAdministrators	Object	public
GetAdministratorByUserName	Object	public

**Tabla 2.75 Lista de Operaciones de la clase AdministratorProfiles**

**2.2.1.3.1.17 CLASE LEVELCLASSROOM:** Controla la tabla LevelClassroom en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE LEVELCLASSROOM**

Nombre	Tipo de Dato	Visibilidad
ClassroomId	Int	Private
LevelID	Int	Private
UserName	Int	Private
Class	Int	Private

**Tabla 2.76 Lista de Atributos de la clase LevelClassroom**

**b) LISTA DE OPERACIONES DE LA CLASE LEVELCLASSROOM**

Nombre	Tipo de retorno	Visibilidad
GetLevelClassrooms	Object	Public
GetLevelClassroomsByTeacher	Object	Public

**Tabla 2.77 Lista de Operaciones de la clase LevelClassroom.**

**2.2.1.3.1.18 CLASE QUESTIONINGROUPQUESTIONS:** Controla la tabla QuestionInGroupQuestion en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE QUESTIONINGROUPQUESTIONS**

Nombre	Tipo de Dato	Visibilidad
QuestionGroupId	Int	private
QuestionGroupIdInQuestion	Int	private
QuestionId	Int	private
LevelID	Int	private
FacultyId	Int	private
UnitId	Int	private
TopicId	Int	private
QuestionGroupDescription	String	Private
QuestionDescription	String	Private
Options	String	Private
IsPresentialMode	Bool	Private
IsIntensiveMode	Bool	Private
Activated	Bool	Private
QuestionActivated	Bool	Private

LevelName	String	Private
FacultyName	String	Private
UnitName	String	Private
TopicName	String	Private

**Tabla 2.78 Lista de Atributos de la clase Questioningroupquestions**

**b) LISTA DE OPERACIONES DE LA CLASE QUESTIONINGROUPQUESTIONS.**

Nombre	Tipo de retorno	Visibilidad
GetQuestionInGroupQuestions	Int	Public

**Tabla 2.79 Lista de Operaciones de la clase Questioningroupquestions**

Las siguientes clases usan las tablas del Esquema SqlMembershipProvider y SqlRoleProvider usadas para el almacenamiento de información de la cuenta de usuario, roles y asignación de las funciones a los usuarios y permiten fusionar esta herramienta con la aplicación que se desarrolló.

**2.2.1.3.1.19 CLASE ASPNET\_ROLES: Controla la tabla AspNet\_Roles en la DB.**

**a) LISTA DE ATRIBUTOS DE LA CLASE ASPNET\_ROLES**

Nombre	Tipo de Dato	Visibilidad
ApplicationId	String	Private
RoleId	Int	Private
RoleName	String	Private
LoweredRoleName	String	Private
Description	String	Private

**Tabla 2.80 Lista de Atributos de la clase AspNet\_Roles**

**b) LISTA DE OPERACIONES DE LA CLASE ASPNET\_ROLES**

Nombre	Tipo de retorno	Visibilidad
GetRoles	Object	Public
GetStudentRoles	Object	Public

**Tabla 2.81 Lista de Operaciones de la clase AspNet\_Roles**

**2.2.1.3.1.20 CLASE ASPNET\_USERS: Controla la tabla AspNet\_Users en la DB**

**a) LISTA DE ATRIBUTOS DE LA CLASE ASPNET\_USERS**

Nombre	Tipo de Dato	Visibilidad
ApplicationId	String	Private
UserId	String	Private
UserName	String	Private
LoweredUserName	String	Private
MobileAlias	String	Private
IsAnonymous	Bool	Private
LastActivityDate	DateTime	Private

**Tabla 2.82 Lista de Atributos de la clase AspNet\_Users**

**2.2.1.3.1.21 CLASE ASPNET\_USERINROLES:** Controla la tabla AspNet\_UserInRoles en la DB.

**a) LISTA DE ATRIBUTOS DE LA CLASE ASPNET\_USERINROLES**

Nombre	Tipo de Dato	Visibilidad
UserId	Int	private
RoleId	Int	private
UserName	Int	private
RoleName	Int	private

**Tabla 2.83 Lista de Atributos de la clase AspNet\_UserInRoles**

**b) LISTA DE OPERACIONES DE LA CLASE ASPNET\_USERINROLES**

Nombre	Tipo de retorno	Visibilidad
GetUserInRoles	Object	Public

**Tabla 2.84 Lista de Operaciones de la clase AspNet\_UserInRole.**

## 2.3 CONSTRUCCIÓN DEL PROTOTIPO

Esta sección detalla como se desarrollo el sistema, cabe mencionar que parte del código correspondiente al sistema, se presenta en los anexos.

### 2.3.1 INICIO DE SESIÓN.

Esta página es la primera del sistema y nos permite ingresar al mismo una vez ingresados el nombre de usuario y contraseña correctos. Es común para todos los usuarios.



Login

Iniciar sesión

Nombre de usuario: 1715832075

Contraseña: ●●●●●●●●

Recordármelo la próxima vez.

Inicio de sesión

**Figura 2.6. Inicio de Sesión.**

Una vez que se ingresa a la sistema, aparece un ambiente común para los usuarios, el cual consiste en un encabezado en el que aparece los datos del usuario, su foto, el logo de la universidad, y el logout. Un pie de página en el cual aparece el copyright del sistema, y en la parte izquierda las opciones o menús propios de cada usuario.



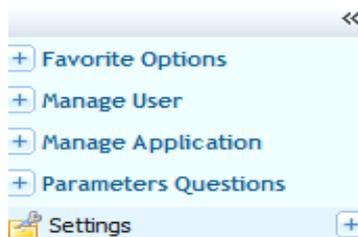
**Figura 2.7. Ambiente Común del Sistema**

### 2.3.2 AMBIENTE DEL USUARIO ADMINISTRADOR.

Esta parte describe las opciones o menús propios del usuario Administrador.

Este usuario posee las siguientes opciones:

- Manager User
- Manager Application.
- Parameter Questions.



**Figura 2.8. Menú usuario Administrador**

A continuación se describe cada uno de ellos.

#### 2.3.2.1 MANAGER USER

Esta opción permite a un Administrador crear usuarios para el sistema, ya sean alumnos, profesores, editores o administradores.



**Figura 2.9. Menú Manager User**

Cabe mencionar que cada usuario cuenta con campos obligatorios que deben completarse, los mismos se muestran con un asterisco (\*) ubicado al lado derecho de la opción y sin los cuales no se permite la creación del nuevo usuario, además deben cumplir con el formato y la longitud indicada, caso contrario aparece el mensajes de error correspondiente.

The screenshot shows the 'Create User' form with the following fields and error messages:

- DETAILS:**
  - User Name: \* (sdasdsdsda) - Error: Debe ingresar 10 dígitos
  - Password: \* (masked) - Error: Debe ingresar únicamente letras
  - Email: \* (hjsadh) - Error: Ingresar una dirección de correo electrónico valido
  - Re-Type Password: \* (masked) - Error: Debe ingresar únicamente letras
- PERSONAL INFORMATION:**
  - Names: \* (13221) - Error: Debe ingresar únicamente letras
  - Last Names: \* (12121) - Error: Debe ingresar únicamente letras
  - Photo 92x70: \* (empty) - Error: Examinar... \*
  - Identity Card: \* (empty) - Error: \*
  - Blood Type: \* (empty) - Error: \*
  - Gender: \* ([Select One]) - Error: \*
  - Birthday: (empty)

Figura 2.10 Mensajes en caso de ingreso erróneo de datos

### 2.3.2.1.1 ADMINISTRATOR

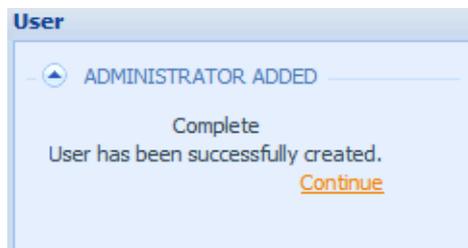
Esta página permite crear un usuario administrador, y consta de la información necesaria para crear el nuevo usuario tal como details, personal information, control information, additional information, el cual aparece a través de ítems que muestran u ocultan la información según sea necesaria.

The screenshot shows the 'Create User' form for an administrator user with the following fields:

- DETAILS:**
  - User Name: \* (1716151413)
  - Password: \* (masked)
  - Email: \* (1716151413@tesis.com)
  - Re-Type Password: \* (masked)
- PERSONAL INFORMATION:**
  - Names: \* (Carla Elizabeth)
  - Last Names: \* (Criollo Suarez)
  - Photo 92x70: \* (C:\Documents and Settings\Administrador\Mis documentos\Mis imagenes\probar\1716151413) - Error: Examinar...
  - Identity Card: \* (1716151413)
  - Blood Type: \* (o+)
  - Gender: \* (female)
  - Birthday: (3/4/1984)
- CONTROL INFORMATION:**
  - Security Question: \* (Favorite Pet)
  - Answer: \* (Dog)
  - Father Name: (Cristian Criollo)
  - Mother Name: (Carmen Suarez)
- ADDITIONAL INFORMATION:**
  - Principal Street: (Mariscal Sucre)
  - Transversal Street: (Jose Perez)
  - Street Number: (1021)
  - Country: (Ecuador)
  - City: (Quito)
  - Marital Status: (single)
  - Phone: (3031323)
  - Cellphone: (084256363)

Figura 2.11 Crear usuario Administrador

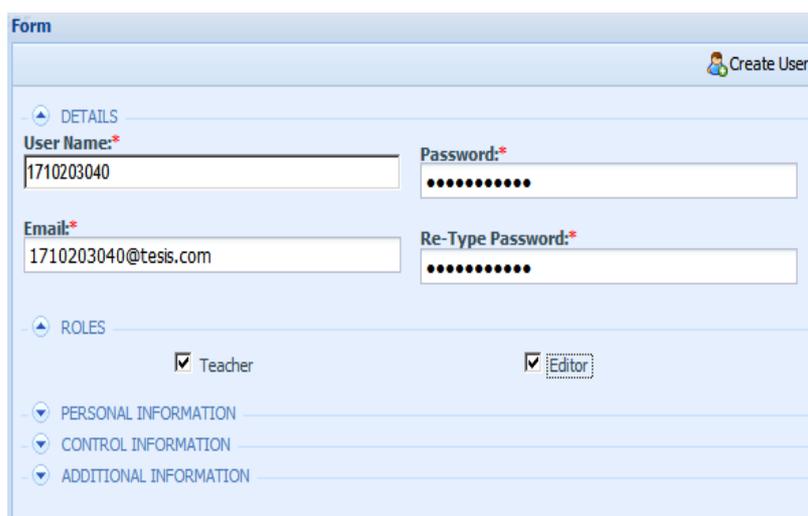
Una vez completados los campos requeridos y pulsando el botón create user, aparece la confirmación de que se ha añadido un nuevo administrador.



**Figura 2.12. Aviso de usuario creado.**

#### 2.3.2.1.2 TEACHER AND EDITOR.

Esta página permite crear un usuario teacher/editor, y consta de la información necesaria para crear el nuevo usuario tal como details, personal information, control information, additional information, el cual aparece a través de ítems que muestran u ocultan la información según sea necesaria, se diferencia del usuario administrador porque tiene un ítem mas denominado roles en el cual se puede escoger que tipo de usuario se creará, ya sea teacher y/o editor, por defecto se crea como teacher.

A screenshot of a web form titled "Form" for creating a user. The form is divided into sections: "DETAILS" (expanded), "PERSONAL INFORMATION", "CONTROL INFORMATION", and "ADDITIONAL INFORMATION". In the "DETAILS" section, there are four input fields: "User Name:" with the value "1710203040", "Password:" (masked with dots), "Email:" with the value "1710203040@tesis.com", and "Re-Type Password:" (masked with dots). Below these fields is a "ROLES" section with two checkboxes: "Teacher" (checked) and "Editor" (checked). A "Create User" button is located in the top right corner of the form.

**Figura 2.13 Crear usuario teacher/Editor**

De igual manera aparece la confirmación de que un nuevo usuario se ha creado.

### 2.3.2.1.3 STUDENT.

Esta página permite crear un usuario student, y consta de la información necesaria para crear el nuevo usuario tal como details, personal information, control information, addittional information, el cual aparece a través de ítems que muestran u ocultan la información según sea necesaria, se diferencia del usuario administrador porque tiene un ítem mas denominado academic information, en el cual se especifica la información sobre nivel, modalidad, classroom de ingles y semestre que cursan.

**Figura 2.14 Crear usuario estudiante**

De igual manera aparece la confirmación de que un nuevo usuario se ha creado.

### 2.3.2.2 MANAGE APPLICATION

Esta opción permite a un administrador configurar los parámetros necesarios para el funcionamiento del sistema.



**Figura 2.15 Menú Manage Application**

### 2.3.2.2.1 FACULTY

Esta opción configurar las Facultades que tiene la Universidad y determinan los niveles de inglés que un estudiante debe tomar de acuerdo a su carrera.

**Description**

Faculties - Administration (insert/delete/update).  
 Since this "Grid" you can manage all faculties.  
 Faculty "All" means that a question is valid for all faculties. You can use TAB key to move to another cell or use double clic.

Faculty Name	Faculty Description ^
All	All faculties, this option allows General Online Test.
Engineering	Engineering applying technical, scientific, and mathematical knowledge to materials, structures, machines, devices, syst...
Human and Education Sciences	Its purpose is the transformation of the national reality in the field of Humanities, Social and Education with Salesian styl...
Economic and Administrative Sciences	The Faculty of Administrative Sciences and Economics through their careers, looking to fill the need to direct the formati...
Agricultural and Environmental Sciences	We are forced to produce in quantity and quality transcending the individual benefit to collective solidarity.

Page 1 of 1

Save Delete selected records Insert Refresh

**Figura 2.16 Agregar Facultad**

### 2.3.2.2.2 LEVEL

Permite configurar los niveles con que cuenta el Departamento de Idiomas de la Universidad y que los estudiantes deben aprobar de acuerdo a su carrera.

**Levels**

Add Delete Save

Number of Level: 1

Name of Level: Level One

Last Change: 3/15/2010

Description: it's the first level

Page 1 of 6

**Figura 2.17. Agregar Nivel**

### 2.3.2.2.3 CLASSROOM

Esta opción permite asignar los profesores a sus respectivas clases, en base a la facultad el nivel de ingles y la modalidad.

The screenshot shows a window titled "Classroom" with a sub-section "Add Class". It contains several form fields:

- Class Number:\***: Text input field containing "1".
- Modality:\***: Dropdown menu with "Presental Mode" selected.
- Level:\***: Dropdown menu with "1 - Level One" selected.
- Faculty:\***: Dropdown menu with "All" selected.
- Teacher:\***: Dropdown menu with "Yanez Velastegui Patricio Jose" selected.

At the bottom of the form is a button labeled "Add Classroom". Below the button, a red status message reads "Classroom 1 Added".

**Figura 2.18. Agregar Classroom**

### 2.3.2.2.4 TOPICS

Permite configurar los tópicos que son parte del test online.

The screenshot shows a window titled "Topics of Evaluation" with a "Save" button in the top right corner. It contains two main fields:

- Name of Topic:** Text input field containing "Grammar".
- Description:** Text area containing the following text: "Grammar may be separated into two common broad categories: descriptive and prescriptive. Both views of grammar are in wide use, although in general, linguists tend towards a descriptive approach to grammar, while people teaching a specific language — such as English — might tend towards a more prescriptive approach. Usually, there is a bit of give and take in any approach, with a prescriptivist being at least somewhat descriptive, and a descriptivist having some prescriptivist tendencies."

At the bottom of the window, there is a page indicator showing "Page 1 of 5" and navigation icons.

**Figura 2.19. Agregar Tópicos**

### 2.3.2.2.5 UNITS

Esta opción configura los capítulos que corresponden a cada nivel, es importante tomar en cuenta que estos deben ser un número par.

**Figura 2.20 Agregar unidades a los Niveles**

### 2.3.2.2.6 PARAMETERS QUESTIONS

Esta opción permite configurar el número de preguntas que tendrá un examen de acuerdo a los tópicos, que tienen un porcentaje de 60 % grammar, 10% reading, 10% listening, 10% vocabulary y 10% writting.

**Figura 2.21. Agregar parámetros del test.**

Cabe mencionar que la mayoría de estas configuraciones se las realiza una única vez, pero son muy necesarias para el correcto funcionamiento del sistema, la excepción se da en classroom donde se van asignando profesores a los cursos y en parameters questions donde se configura el número de preguntas que tendrá un test.

### 2.3.3 AMBIENTE DEL USUARIO TEACHER.

Esta parte describe las opciones o menús propios del usuario Teacher.

Este usuario posee las siguientes opciones:

- Academic Options
- Information

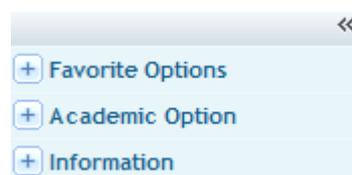


Figura 2.22. Menú usuario Teacher

A continuación se describe cada uno de ellos:

#### 2.3.3.1 ACADEMIC OPTIONS

Esta opción permite a un usuario Teacher generar los test de sus alumnos para que puedan ser rendidos.

##### 2.3.3.1.1 GENERATE TEST

Esta página permite al usuario teacher generar los test a sus estudiantes de acuerdo a la fecha, tiempo, la clase a la que enseña y si es el primer o segundo parcial, únicamente se debe escoger el alumno al cual se le va a generar el test.

 A screenshot of the 'Generate Test' web interface. On the left is a 'Parameters of Test' form with fields for Date of Test (5/5/2010), Test Time (00 hours: 05 minutes), Classroom (Level One - 15), Partial (1 - first partial), Modality (1 - PresentalMode), and Faculty (All). On the right is a 'Student's List' table with columns for Generate Test, Complete Name, and Identify. The table contains three rows of student data.
 

Generate Test	Complete Name	Identify
<input checked="" type="checkbox"/>	Carrasco Rojas Maria Mercedes	1711111111
<input checked="" type="checkbox"/>	Palacios Gomez Luis Eduardo	1733333333
<input checked="" type="checkbox"/>	Pardo Rosales Rocio Elizabeth	1744444444
<input type="checkbox"/>	Ruales Perez Jorge Esteban	1722222222

Figura 2.23. Generar examen a los estudiantes

### 2.3.3.1.2 INFORMATION

Esta página permite al usuario Teacher revisar y proporcionar las notas de sus alumnos.

#### 2.3.3.1.2.1 GRADES

Esta página permite revisar y proporcionar las calificaciones de los estudiantes de acuerdo a cada clase que el profesor imparte.



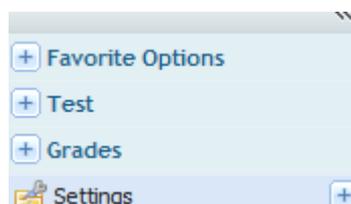
**Figura 2.24 Calificaciones de los estudiantes por curso.**

## 2.3.4 AMBIENTE DEL USUARIO STUDENT

Esta parte describe las opciones o menús propios del usuario Student.

Este usuario posee las siguientes opciones:

- Test
- Grades.



**Figura 2.25. Menú usuario Estudiante**

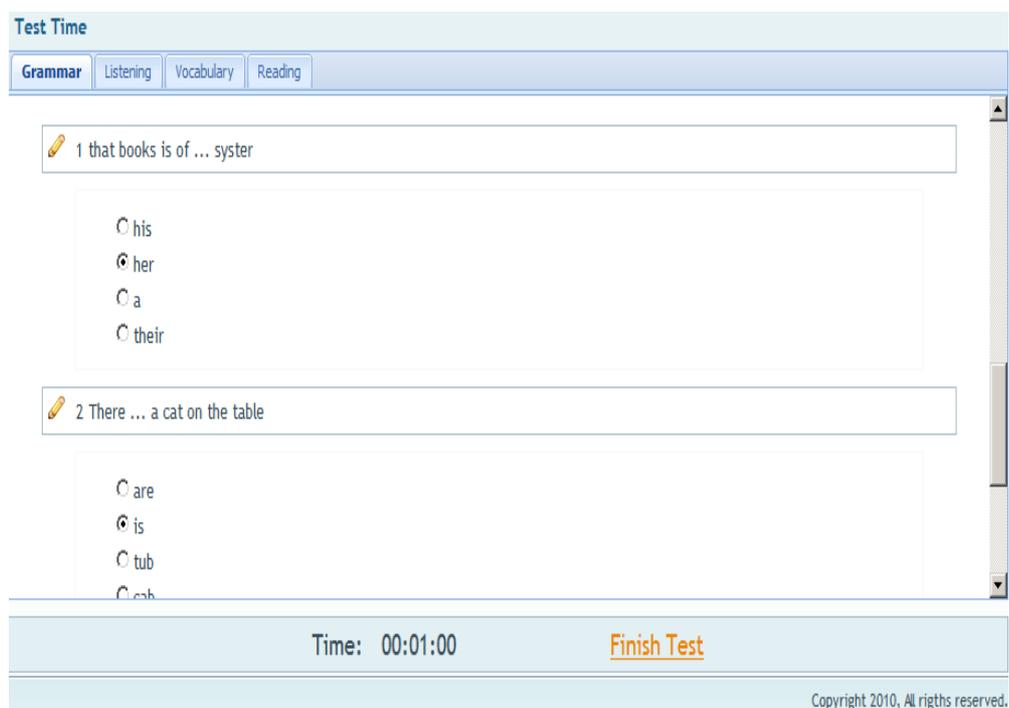
A continuación se describe cada uno de ellos:

### 2.3.4.1 TEST

Esta opción permite al usuario Student rendir el test.

#### 2.3.4.1.1 TAKE TEST

Esta página permite al usuario Student rendir el test generado por el profesor, únicamente puede acceder a esta opción si el teacher a generado el test. El test consta de las secciones de grammar, listening, reading y vocabulary, en un porcentaje de 60, 10,10, 10 respectivamente, lo que nos brinda un 90% de la nota, el 10% restante, procede del writting que no es parte del sistema.



The screenshot shows a web interface titled "Test Time". At the top, there are four tabs: "Grammar", "Listening", "Vocabulary", and "Reading". The "Grammar" tab is selected. Below the tabs, there are two question boxes. The first question is "1 that books is of ... syster" with a pencil icon. The options are:  his,  her,  a, and  their. The second question is "2 There ... a cat on the table" with a pencil icon. The options are:  are,  is,  tub, and  cab. At the bottom of the interface, there is a timer showing "Time: 00:01:00" and a button labeled "Finish Test". A copyright notice "Copyright 2010, All rigths reserved." is visible in the bottom right corner.

**Figura 2.26 Examen de un estudiante.**

### 2.3.4.2 GRADES

Esta opción permite revisar las calificaciones obtenidas por el estudiante para saber si aprobó el nivel correspondiente.



	First Partial:	Second Partial:
Partial 1:	8.00	0.00
Partial 2:	9.00	0.00
Partial 3:	6.00	0.00
Exam:	13.84	0.00
Writing:	2.00	0.00

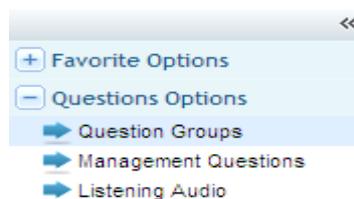
**Figura 2.27 Calificaciones de un estudiante**

### 2.3.5 AMBIENTE USUARIO EDITOR

Esta página describe las opciones o menús propios del usuario Editor.

Este usuario posee la siguiente opción:

- Questions Options.



**Figura 2.28 Menú Editor.**

A continuación se describe cada uno de ellos:

#### 2.3.5.1 QUESTIONS OPTIONS

Esta opción permite al usuario editor gestionar las preguntas asociadas por grupos, para los diferentes niveles, unidades y tópicos que están compuestos los test.

##### 2.3.5.1.1 QUESTIONS GROUP.

Esta opción permite configurar el grupo de preguntas que se determina a través del nivel, la facultad, la unidad y el tópico, además en la descripción, permite mostrar un encabezado para las preguntas de grammar, listening y vocabulary y

el texto completo en el caso del reading. Configura también la modalidad a la que pertenece y si la pregunta esta o no activada, para ser utilizada.

**Description**

Question groups (insert/delete/update).  
For this purpose you should select "LevelID", "UnitId", "TopicId" first. This section could be used by putting the principal instructions for different questions.  
You can insert different types of questions in this section, other cells are only for information purposes.

**Groups of Questions**

	Level Name	Faculty Name	Unit Name	Topic Name	Question Group Description	Presential	Intensive	Distance	Active
1	Level One	1 All	1 chapter 1	1 Grammar	Verb "to be"	true	true	true	true
1	Level One	1 All	1 chapter 1	1 Grammar	2 Verb "To Be"	true	true	true	true
1	Level One	1 All	5 chapter 4	1 Grammar	Information Questions	true	true	true	true
3	Level Three	1 All	33 chapter 31	1 Grammar	functionality test	true	true	true	true

Page 5 of 5

Save Delete selected records Insert Refresh

**Figura 2.29. Configuración de QuestionsGroup.**

### 2.3.5.1.2 MANAGEMENT QUESTIONS.

Permite relacionar las opciones de respuesta con sus respectivas preguntas, además permite activarlas o desactivarlas para poder utilizarlas en el test.

**Description**

Question (insert/delete/update).  
You can insert different questions in this section, other cells are for information purposes only.

**Questions**

Question Group Identify	Group Description	Question Description	Options	Active
10	Simple present tense	they ... their parents	meet/meet/meets	true
10	Simple present tense	I ... my lesson	learn/learns/learned	true
10	Simple present tense	You ... around the wall	walk/walks/walked	true
10	Simple present tense	After Fred ... his holiday in Italy he (to want) to learn Italian	spends/spend/waste	true
7	Choose true or false	Do you hear anybody happy	true/false	true
7	Choose true or false	Do you hear melissa is singer	true/false	true
16	Hear the conversation and ...	Joseph is a bad actor	true/false	true
16	Hear the conversation and ...	What did she find?	a car/motorcycle/dog	true
8	vocabulary of chapter 1	What is the father of my father	my grandfather/my father in law/my f...	true
17	vocabulary of chapter 2	What is a change in the lucky	lucky break/lucky OK/lucky up	true

Page 2 of 6

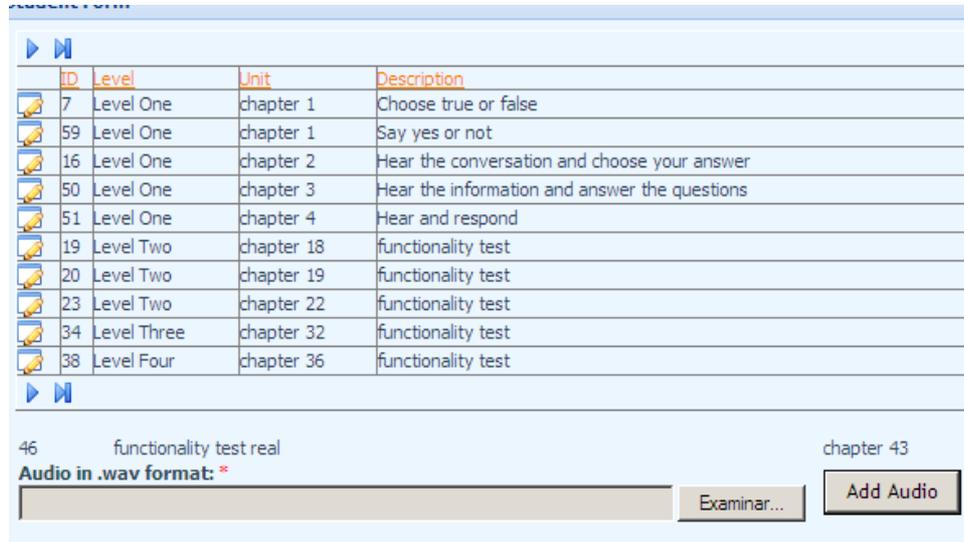
Save Delete selected records Insert Refresh

Copyright 2010, All rights reserved.

**Figura 2.30 Manejo de Preguntas**

### 2.3.5.1.3 LISTENING AUDIO.

Permite relacionar el audio con las preguntas de listening.



ID	Level	Unit	Description
7	Level One	chapter 1	Choose true or false
59	Level One	chapter 1	Say yes or not
16	Level One	chapter 2	Hear the conversation and choose your answer
50	Level One	chapter 3	Hear the information and answer the questions
51	Level One	chapter 4	Hear and respond
19	Level Two	chapter 18	functionality test
20	Level Two	chapter 19	functionality test
23	Level Two	chapter 22	functionality test
34	Level Three	chapter 32	functionality test
38	Level Four	chapter 36	functionality test

46      functionality test real      chapter 43

Audio in .wav format: \*

Examinar...      Add Audio

Figura 2.31. Agregar audio a una pregunta.

## **CAPITULO 3**

### **PRUEBAS E IMPLEMENTACIÓN DEL SISTEMA**

En este capítulo se explica y describe las pruebas realizadas al sistema y los detalles de la implementación del mismo en el departamento de Idiomas

Cabe mencionar que este departamento utilizará la infraestructura de los laboratorios de la Universidad.

#### **3.1 PRUEBAS DEL SISTEMA**

El presente capítulo trata de las pruebas realizadas al Sistema de Evaluación por Computadora diseñado para el Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur.

El objetivo de las pruebas es encontrar un error para determinar situaciones inesperadas para tomar los correctivos necesarios.

##### **3.1.1 PRUEBAS DE FUNCIONALIDAD.**

Estas pruebas fueron muy importantes ya que por medio de ellas se pudo verificar que el sitio desarrollado contará con la funcionalidad deseada, es decir, que cuando damos clic en algún botón o algún link, realiza las acciones correspondientes y nos envía a las páginas correspondientes.

Debido a que el método de desarrollo de este proyecto es el Prototipado, durante el desarrollo de cada módulo se fue probando cada función que se desarrollaba con el fin de evitar que a la conclusión del mismo se presentarán inconvenientes

mayores, además que es la base de este método ir probando los diferentes módulos en todas las etapas de desarrollo.

Las siguientes tablas muestran el desarrollo de esta prueba.

#### Login.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web	Se despliega la página solicitada	
Presionar en el Botón <i>Inicio de Sesión</i>	Se despliega la página solicitada	Mensaje de error si no ingresa el usuario y password correcto.

**Tabla 2.85 Login.aspx.**

### 3.1.1.1 MÓDULO PARA EL ADMINISTRADOR EN EL SISTEMA

- MANAGE USER

#### AdministartorAdd.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Manager User</i> y presionar en <i>Administrador</i>	Se despliega la página solicitada <i>Add User Administrador</i>	
Ingresar los datos requeridos y pulsar <i>Create User</i>	Se despliega la confirmación de usuario creado	Posibles mensajes de error si no se ingresa los campos obligatorios o datos que no corresponden a cada campo.

**Tabla 2.86 AdministartorAdd.aspx**

#### UserAdd.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Manager User</i> y presionar en <i>Teacher and Editor</i>	Se despliega la página solicitada <i>Add User</i>	
Ingresar los datos requeridos y pulsar <i>Create User</i>	Se despliega la confirmación de usuario creado	Posibles mensajes de error si no se ingresa los campos obligatorios o datos que no corresponden a cada campo.

**Tabla 2.87 UserAdd.aspx**

#### StudentAdd.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manager User</i> y presionar en <i>Students</i>	Se despliega la página solicitada <i>Add User</i>	
Ingresar los datos requeridos y pulsar <i>Create User</i>	Se despliega la confirmación de usuario creado	Posibles mensajes de error si no se ingresa los campos obligatorios o datos que no corresponden a cada campo.

**Tabla 2.88 StudentAdd..aspx**

### AdministratorControl.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manager User</i> y presionar en <i>AdministratorControl</i>	Se despliega la página solicitada <i>Add User</i>	
Escoger el usuario que se quiere editar en <i>Parameters of User</i> .	Se despliega los datos del usuario escogido.	
Se realiza los cambios al usuario y se presiona <i>Update Data</i>	Se realizan los cambios en el usuario.	
Para borrar un usuario se escoge el usuario correspondiente en <i>Parameters of user</i> y se presiona <i>Delete User</i>	El usuario es borrado del sistema.	Se presenta un mensaje indicando si esta seguro de borrar el usuario.

**Tabla 2.89 AdministratorControl.aspx**

### TeacherControl.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manager User</i> y presionar en <i>TeacherControl</i>	Se despliega la página solicitada <i>Add User</i>	
Escoger el usuario que se quiere editar en <i>Parameters of User</i> .	Se despliega los datos del usuario escogido.	
Se realiza los cambios al usuario y se presiona <i>Update Data</i>	Se realizan los cambios en el usuario.	
Para borrar un usuario se escoge el usuario correspondiente en <i>Parameters of user</i> y se presiona <i>Delete User</i>	El usuario es borrado del sistema.	Se presenta un mensaje indicando si esta seguro de borrar el usuario.

**Tabla 2.90 TeacherControl.aspx**

### StudentsControl.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manager User</i> y presionar en <i>StudentsControl</i>	Se despliega la página solicitada <i>Add User</i>	
Escoger el usuario que se quiere editar en <i>Parameters of User</i> .	Se despliega los datos del usuario escogido.	
Se realiza los cambios al usuario y se presiona <i>Update Data</i>	Se realizan los cambios en el usuario.	
Para borrar un usuario se escoge el usuario correspondiente en <i>Parameters of user</i> y se presiona <i>Delete User</i>	El usuario es borrado del sistema.	Se presenta un mensaje indicando si esta seguro de borrar el usuario.

**Tabla 2.91 StudentsControl.aspx**

- **Manage Application**

#### **Faculties.aspx**

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manage Application</i> y presionar en <i>Faculty</i>	Se despliega la página solicitada <i>Faculties</i>	
Pulsar Insert, Ingresar los datos requeridos y pulsar <i>Save</i>	Se despliega la confirmación de que los datos han sido guardados exitosamente.	Mensajes de aviso si se ingresa datos y no se presiona Save.
Para borrar un dato, seleccionarlo, presionar <i>Delete selected records</i> y presionar <i>Save</i>	El dato es borrado del sistema, Se despliega la confirmación de operación exitosa.	Mensajes de aviso si no se presiona Save.
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.92 Faculties.aspx**

#### **ManagementLevels.aspx**

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manage Application</i> y presionar en <i>Level</i>	Se despliega la página solicitada <i>Management Level</i>	
Pulsar <i>Add</i> , Ingresar los datos requeridos y pulsar	Se despliega la confirmación de que los	Mensajes de aviso si se ingresa datos y no se

<i>Save</i>	datos han sido guardados exitosamente.	presiona <i>Save</i> .
Para borrar un dato, seleccionarlo, presionar <i>Delete</i> presionar	El dato es borrado del sistema	Mensajes de aviso de Level borrado.
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.93 ManagementLevels.aspx**

### ManagemetTopics.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manage Application</i> y presionar en <i>Topic</i>	Se despliega la página solicitada <i>Management Topic</i>	
Realizar cambios en los campos y pulsar <i>Save</i>	Se despliega la confirmación de que los datos han sido guardados exitosamente.	
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.94 ManagemetTopics.aspx**

### ManagemetUnits.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manage Application</i> y presionar en <i>Unit</i>	Se despliega la página solicitada <i>Management Unit</i>	
Pulsar <i>Add</i> , Ingresar los datos requeridos y pulsar <i>Save</i>	Se despliega la confirmación de que los datos han sido guardados exitosamente.	
Para borrar un dato, seleccionarlo, presionar <i>Delete</i> presionar	El dato es borrado del sistema	Mensajes de aviso de Level borrado.
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.95 ManagemetUnits.aspx**

### ManagemetClassroom.aspx

<b>Acción</b>	<b>Respuesta</b>	<b>Mensaje</b>
Ingresar al sitio Web, en la sección <i>Manage Application</i> y presionar	Se despliega la página solicitada Management Classroom	

en <i>Classroom</i> .		
Escoger <i>Nivel</i> y <i>Modalidad</i> en <i>Available Courses</i>	Se despliega la lista de clases y sus respectivos teachers asignados	
Para asignar una nueva clase, Ingresar los datos requeridos y pulsar <i>Add Classroom</i> .	Se agrega el nuevo curso con su respectivo profesor.	

**Tabla 2.96 ManagemetClassroom.aspx**

#### UpdatetClassrooms.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Manage Application</i> y presionar en <i>Manage Classes</i> .	Se despliega la página solicitada <i>Management Classroom</i>	
Escoger <i>Nivel</i> y <i>Modalidad</i> en <i>Available Courses</i>	Se despliega la lista de clases y sus respectivos teachers asignados	
Escoger un classroom y presionar sobre el.	Aparece los datos de descripción del curso y profesor asignado al mismo	
Presionar <i>Delete</i>	Borra la clase que se escogió.	

**Tabla 2.97 UpdatetClassrooms.aspx**

- **Parameters Questions**

#### ManagementParametersQuestions.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Parameters Questions</i> y presionar en <i>Parameters</i>	Se despliega la página solicitada <i>Management Parameters Questions</i>	
Pulsar <i>Add</i> , Ingresar los datos requeridos y pulsar <i>Save</i>	Se agrega un nuevo ítem en los parámetros	
Para borrar un dato, seleccionarlo, presionar <i>Delete</i> y presionar <i>Save</i>	El ítem es borrado del sistema	
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.98 ManagementParametersQuestions.aspx**

### 3.1.1.2 MODULO PARA EL PROFESOR EN EL SISTEMA.

- **Academic Options**

#### GenerateTest.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Academic Option</i> y presionar en <i>Generate Test</i> .	Se despliega la página solicitada Genrate Test.	
Escoger la <i>fecha</i> , el <i>tiempo</i> , el <i>classroom</i> y el <i>partial</i>	Se muestra la lista de alumnos para generar el test	
Seleccionar los alumnos de la lista	Genera el Test, a los alumnos según los parámetros establecidos	
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.99 GenerateTest.aspx**

- **Information**

#### Grades.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>information</i> y presionar en <i>Grades</i> .	Se despliega la página solicitada <i>Grade Box</i> , con la lista de estudiantes.	
Escoger un estudiante de la lista	Muestra los campos de las notas que ingresa el profesor de cada alumno	
Una vez ingresadas las notas, presionar <i>Save</i>	Se guardan las notas agregadas	
Presionar <i>Salir</i>	Muestra <i>Grade Box</i> con la lista de estudiantes.	

**Tabla 2.100 Grades.aspx**

### 3.1.1.3 MODULO PARA EL ESTUDIANTE EN EL SISTEMA.

- **Test**

#### TakeTest.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Test</i> y presionar en <i>Take Test</i> .	Se despliega la página solicitada <i>Test Time</i> .	
Escoger la ficha de la sección del examen con que desea empezar (Grammar, Listening, Reading, Vocabulary)	Se muestra la lista de preguntas de acuerdo a la sección seleccionada	
De la lista de preguntas seleccionar una de las opciones	Se muestra la lista con las opciones seleccionadas	
Presionar <i>Finish</i>	Muestra un aviso de que acabo el examen	
Si el tiempo determinado concluye	Muestra un aviso de que acabo el examen	

**Tabla 2.101 TakeTest.aspx**

- **Grades**

#### Grades.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Grades</i> y presionar en <i>Grades</i> .	Se despliega la página solicitada <i>Grade Box</i> , con la lista calificaciones del estudiante.	

**Tabla 2.102 Grades.aspx**

### 3.1.1.4 MODULO PARA EL EDITOR EN EL SISTEMA.

- **Question Options**

#### QuestionGroup.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Question Options</i> y presionar en	Se despliega la página solicitada <i>Question Group</i> .	

<i>QuestionGroup.</i>		
Pulsar <i>Insert</i> , Ingresar los datos requeridos y pulsar <i>Save</i>	Se despliega la confirmación de que los datos han sido guardados exitosamente.	Mensajes de aviso si se ingresa datos y no se presiona <i>Save</i> .
Para borrar un dato, seleccionarlo, presionar <i>Delete selected records</i> y presionar <i>Save</i>	El dato es borrado del sistema, Se despliega la confirmación de operación exitosa.	Mensajes de aviso si no se presiona <i>Save</i> .
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.103 QuestionGroup.aspx**

### ManagementQuestionGroup.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Question Options</i> y presionar en <i>ManagementQuestion</i> .	Se despliega la página solicitada <i>Questions</i> .	
Pulsar <i>Insert</i> , Ingresar los datos requeridos relacionados a un <i>QuestionGroup</i> y pulsar <i>Save</i>	Se despliega la confirmación de que los datos han sido guardados exitosamente.	Mensajes de aviso si se ingresa datos y no se presiona <i>Save</i> .
Para borrar un dato, seleccionarlo, presionar <i>Delete selected records</i> y presionar <i>Save</i>	El dato es borrado del sistema, Se despliega la confirmación de operación exitosa.	Mensajes de aviso si no se presiona <i>Save</i> .
Presionar <i>Refresh</i>	Muestra la página con los cambios que se hayan presentado.	

**Tabla 2.104 ManagementQuestionGroup.aspx**

### ManagementListeningQuestions.aspx

Acción	Respuesta	Mensaje
Ingresar al sitio Web, en la sección <i>Question Options</i> y presionar en <i>ListeningAudio</i> .	Se despliega la página solicitada <i>Add Listening Audio</i> .	
Pulsar <i>Insert</i> , Ingresar los datos requeridos relacionados a un <i>QuestionGroup</i> y pulsar <i>Save</i>	Se despliega la confirmación de que los datos han sido guardados exitosamente.	Mensajes de aviso si se ingresa datos y no se presiona <i>Save</i> .
Seleccionar el id de la pregunta de audio añadida	Aparece un botón examinar para insertar el archivo de audio	

Escoger el audio correspondiente y pulsar <i>Add Audio</i>	Se agrega el audio a la pregunta correspondiente	
--	--	--

**Tabla 2.105 ManagementListeningQuestions.aspx**

Durante el desarrollo del programa se presentaron problemas como que algún link no conducía a la página adecuada o algún error de programación, pero el error fue corregido y ahora funciona correctamente.

El sistema presenta los mensajes respectivos para corregir problemas en caso de presentarse, normalmente estos se deben a ingreso de datos incorrectos o formatos inadecuados.

### **3.1.2 PRUEBAS DE CONCURRENCIA**

Estas pruebas se realizaron conectándose desde 8 computadoras al sitio web, esto debido a que el departamento de idiomas contaba con este pequeño laboratorio para realizar estas pruebas.

En esta prueba, se accedió al sitio en los distintos perfiles del sistema, es decir como administrador, profesor/editor y estudiante, realizándose y comprobando la funcionalidad de cada uno de ellos. Con lo que se aseguro que el sistema responde oportunamente a las peticiones de los usuarios y no se pierde la conexión cliente servidor.

### **3.1.3 PRUEBAS DE SEGURIDAD**

Esta prueba se basa en los roles de los usuarios existentes.

## **USUARIOS NO REGISTRADOS**

Esta prueba se realizó con usuarios no registrados, que intentan ingresar al sitio Web, en la misma no se les permite ingreso a ninguna parte del sistema. Cada usuario necesita sus credenciales (nombre de usuario y contraseña) para su uso.

## **USUARIOS REGISTRADOS**

Esta prueba se la realizó a usuarios que si poseen credenciales en el sistema, pero que tratan de ingresar a sitios que estos usuarios no pueden acceder.

- **USUARIOS NO REGISTRADOS:** No tienen ingreso al sistema.
- **USUARIOS ESTUDIANTES:** Tienen acceso únicamente al módulo de Estudiantes para rendir su examen y revisar sus notas. Acceso restringido al módulo de Profesores, Editores y Administradores.
- **USUARIOS PROFESORES:** Tienen acceso únicamente al módulo de Profesores, para activar exámenes y poner las notas de los estudiantes. Acceso restringido al módulo de Estudiantes, Editores y Administradores.
- **USUARIOS EDITOR:** Tienen acceso únicamente al módulo de Editores, para ingresar y activar preguntas para los exámenes. Acceso restringido al módulo de Estudiantes, Profesores y Administradores.
- **USUARIO ADMINISTRADOR:** Tienen acceso únicamente al módulo de Administración, para administración de usuarios y configuración de parámetros necesarios para los exámenes. Acceso restringido al módulo de Estudiantes, Profesores y Editor.

Actualmente el sistema funciona sin ningún error de programación, se podrían presentar problemas posteriores en el caso de agregar más módulos sin la correcta configuración de los mismos o debido a la falta de experiencia en el uso mismo, lo cual es solucionable con el correcto entendimiento del manual de usuario.

### **3.2 DETALLES DE IMPLEMENTACION DEL SISTEMA.**

Cabe mencionar que para la ejecución de este proyecto se utilizará la infraestructura de los laboratorios de computación de la Universidad. El Departamento de Idiomas del campus sur, cuenta con un pequeño laboratorio de 10 pcs en el que se realizó la implementación y pruebas del 6 al 8 de octubre del año en curso, es importante mencionar que durante el transcurso de la realización del proyecto se iban presentando avances del programa, con lo que los errores que se presentaban se iban corrigiendo en todo momento, para evitar mayores inconvenientes al final del mismo .

El sistema desarrollado funciona en red.

Se adecuo una pc como servidor, las características de la misma son:

Intel Dual Core 2.8Ghz, 2GB de memoria ram, Sistema Operativo: Microsoft Windows Server 2003 Enterprise Edition Service Pack 2. El nombre del servidor se estableció en Virtual2.

Las pc clientes son: Intel core 2 1..8 GHz, 1 GB de ram, Sistema Operativo: Windows Xp profesional service pack 2.

El sistema fue desarrollado en Microsoft Visual Studio 2005 con C#, además la base de datos utilizada se desarrollo en SQL Server 2005, ambos se instalaron y configuraron correctamente en el servidor.

Como el desarrollo del programa esta bajo la línea de Windows, el programa servidor para el sitio web es el Internet Information Server el cual se configuro correctamente. Se estableció a Internet Explorer como navegador predeterminado.

## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

- La educación es una parte fundamental en la sociedad, y la tecnología informática se ha convertido en una pieza primordial para su desarrollo, generando nuevas alternativas que cambian los esquemas tradicionales del sistema educativo, en particular en aplicaciones que complementan el proceso de aprendizaje a través de Internet.
- El Sistema de Evaluación por Computadora se diseñó para poder contar con una solución integral al requerimiento de los estudiantes para la aprobación de la suficiencia de Inglés. Busca facilitar el proceso tanto para estudiantes como para profesores con el uso de la tecnología, ya que el proceso actualmente se lo realiza de forma manual.
- Este sistema permite generar test aleatorios para los estudiantes, de acuerdo al nivel, modalidad y tiempo correspondiente, acorde al formato utilizado por el departamento de idiomas.
- Este sistema se basa en perfiles de usuarios los cuales determinan las funciones que a cada usuario le corresponde. Así, el administrador del sistema cuenta con una herramienta adecuada para el mantenimiento tanto de los usuarios del sistema como de los parámetros generales del test. El Editor se encarga de la administración y manejo de la base de preguntas que se utilizarán en el test. El profesor genera los test para los estudiantes y asigna las calificaciones para aprobar el nivel. El alumno puede rendir el test y revisar los aportes obtenidos a lo largo del nivel.
- El sistema desarrollado se lo realizó con una arquitectura de tres capas, La Capa de Presentación interactúa con el usuario a través de un Navegador Web. La Capa de Negocios, recibe las peticiones del usuario y envía la respuesta tras el proceso. Se comunica con la capa de presentación y la capa de datos. La capa de datos está formada por el gestor de base de datos.

- Las herramientas y metodología utilizadas en el proyecto, facilitan el desarrollo del mismo, ya que permiten realizar la aplicación de una manera fácil y rápida.
- En el desarrollo de las pruebas que se realizó al sistema una vez implementado, se pudo observar que el sistema funciona correctamente, acorde a los objetivos establecidos.
- El proyecto cumple con el método prototipado, ya que durante el desarrollo del mismo se fueron corrigiendo errores suscitados, conforme se iban creando los módulos, lo que permitió que no se presentarán mayores inconvenientes al final del mismo.
- Todo sistema es susceptible a mejorarse, por tal motivo, se espera que este proyecto de tesis sirva de base para otros proyectos relacionados.

## **RECOMENDACIONES.**

- El sistema posee varios perfiles de usuario, por lo tanto, se debe asignar con mucha responsabilidad los roles respectivos, ya que cada uno accede a procesos importantes del sistema y si no son asignados adecuadamente podrían presentarse problemas por mal funcionamiento.
- Debería haber comunicación directa entre este Sistema y el Sistema de calificaciones que usa la Universidad, ya que, a pesar de que este sistema registra las calificaciones, se necesita la verificación del sistema oficial para la inscripción del alumno en el siguiente nivel, esto ayudaría a automatizar aun más el proceso.
- La universidad cuenta con varios laboratorios computacionales a los cuales se podría proporcionar el equipo necesario para el desarrollo del test como audífonos y micrófonos.
- Los conocimientos que se adquieren durante los años de universidad son solo la base para nuestra carrera, por lo que es necesario seguirse preparando día a día, a través de cursos, el uso del Internet y personas afines a la carrera.



## **ANEXOS**

- A. FUNDAMENTACION LEGAL**
- B. DISEÑO DE LA INVESTIGACIÓN.**
- C. DIAGRAMAS DEL SISTEMA**
- D. PLANTILLAS PARA CREAR USUARIOS.**
- E. MANUAL DE USUARIOS**
- F. CÓDIGO DEL SISTEMA**

## **A. FUNDAMENTACIÓN LEGAL**

### **A.1 DEL CONESUP**

#### **DE LOS TITULOS Y GRADOS ACADEMICOS**

**ART. 23.-** Únicamente las Universidades y Escuelas Politécnicas podrán otorgar cualquiera de los títulos y grados académicos a los que se refiere el artículo 44 de la ley. Los institutos superiores técnicos y tecnológicos otorgarán solamente los títulos a los que se refiere el literal a) de la misma norma.

**ART. 25.-** Los títulos conferidos por una universidad o escuela politécnica que no hayan sido registrados en el CONESUP, no habilitan para el ejercicio profesional. Cada interesado podrá registrar su título profesional en el colegio correspondiente siempre y cuando cuente con la certificación del registro en el CONESUP.

**ART. 26.-** Para que los estudiantes que tramiten su graduación puedan cumplir con la obligación establecida en el Art. 64 de la ley, de realizar servicios a la comunidad, prácticas o pasantías, se requerirá que el CONESUP emita los lineamientos generales necesarios. Mientras no se garanticen recursos para financiar los servicios comunitarios, prácticas o pasantías, los alumnos no tienen obligación de realizarlas; en este caso, una vez que el título ha sido expedido y refrendado, podrán presentarlo para registro en el CONESUP sin ningún trámite o requisito adicional.

### **A.2 DE LA UNIVERSIDAD POLITÉCNICA SALESIANA**

#### **INSTRUCTIVO DE GRADUACION**

##### **I OBJETIVO**

El presente instructivo regula el procedimiento que deben seguir los estudiantes de la Universidad Politécnica Salesiana en fase de graduación para optar por la

titulación intermedia (Tecnólogos) o Terminal (Ingenieros, licenciados y otros de tercer nivel), con sus modalidades:

- a) Trabajo de Grado para titulación intermedia, y
- b) Tesis, Proyectos o Productos para Titulación Terminal.

## **II DE LA DENUNCIA**

1. Una vez que el estudiante ha cumplido al menos el 80% del total de materias correspondientes a cada titulación, solicita en la Secretaría de Campus el certificado habilitante que le permite matricularse en denuncia de trabajo de grado o tesis.
2. El estudiante se matricula en denuncia de Trabajo de Grado o Tesis.
3. El estudiante presenta por escrito en Secretaría de Campus la denuncia del tema y del plan de Trabajo de Grado o Tesis, con la recomendación de un profesor de la Facultad, dirigida al Director de Carrera y en los formatos establecidos por cada Facultad. Secretaría de Campus entregará al estudiante la constancia de la entrega-recepción.
4. Secretaría de Campus envía, dentro de las siguientes 8 horas laborables, la solicitud al Director de Carrera, previo ingreso en los registros respectivos.
5. El Director de Carrera deberá presentar la solicitud de denuncia del tema y plan en el próximo Consejo de Carrera.
6. El Consejo de Carrera designa al profesor Director del Trabajo de Grado o Tesis. Esta designación será comunicada por escrito a través de la Secretaría.
7. El designado Director de Trabajo de Grado o Tesis deberá entregar su informe en el siguiente Consejo de Carrera ordinario, a través del Director de Carrera, para su aprobación. En este mismo Consejo se ratificaría al Director de Trabajo de Grado o Tesis.
8. Si el informe no es favorable, el estudiante podrá presentar nuevamente el Plan, con las recomendaciones sugeridas. Esta resolución deberá ser comunicada a través de Secretaría en un plazo de 8 horas laborables.
9. El estudiante cuyo plan no sea aprobado, deberá asesorarse con el Director de Trabajo de Grado o Tesis, sobre las causas y los correctivos a incluir antes de presentar nuevamente la denuncia de Trabajo de Grado o Tesis.

### **III DEL DESARROLLO.**

10. El estudiante, una vez aprobado el Plan de Trabajo de Grado o Tesis, iniciará el desarrollo del mismo, con la asesoría del Director de Trabajo de Grado o Tesis. Esta resolución será notificada al estudiante a través de Secretaría.
11. El Director de Trabajo de Grado o Tesis informará por escrito mensualmente al Director de Carrera, los avances del Trabajo de Grado o Tesis.

### **IV DE LA PRESENTACION DEL TRABAJO ESCRITO.**

12. Una vez concluido el Trabajo de Grado o Tesis, el Director del mismo entregará en Secretaria de Campus un informe dirigido al Director de Carrera, el que incluye la calificación del trabajo escrito sobre 100 puntos presentado en el formato correspondiente
13. La Secretaria de Campus adjuntará al informe, la siguiente documentación presentada por el estudiante:
  - a. Tres ejemplares del Trabajo de Grado o Tesis empastados,
  - b. Un respaldo en formato digital (CD),
  - c. El abstract (resumen o síntesis) con una extensión no mayor a 3 páginas,
  - d. Comprobantes de pago de los derechos de grado.
14. Se deberá llenar el acta de entrega-recepción, entre el estudiante y la Secretaría de Campus.

### **V. DE LA DECLARATORIA DE APTITUD PARA SUSTENTACIÓN.**

15. Una vez receptado dicho informe, secretaria de campus remitirá en un plazo de diez días laborables, al Director de Carrera toda la documentación requerida para la declaración de aptitud para la sustentación:
  - a. Certificado de matrícula de cada uno de los períodos de estudio de la carrera;
  - b. Certificado de aprobación de los créditos por materias, y demás requisitos paracadémicos vigentes en el pensum de estudios cursado;

- c. Informe del Director de Trabajo de Grado o Tesis con la calificación total del trabajo escrito de Tesis (referido en el numeral 12).
16. El Director de Carrera, deberá elaborar un informe que será presentado en el siguiente Consejo de Carrera, respecto del cumplimiento de los requisitos legales y reglamentarios para la sustentación.
17. El Consejo de Carrera, previo informe favorable del Director de Carrera, declarará la aptitud del estudiante y determinará la fecha de sustentación para optar por la titulación respectiva, y entregará a los miembros del tribunal de sustentación, a través de secretaria, un ejemplar empastado del trabajo escrito, el cual se devolverá el día de la sustentación.

## VI DE LA SUSTENTACIÓN

18. El Tribunal conformado por el Director de Carrera quien preside, el Director de Trabajo de Grado o Tesis y un docente designado por el Consejo de Carrera, calificará la sustentación del Trabajo de Grado o Tesis sobre cien puntos. La nota de sustentación final que el estudiante obtenga será el promedio de las tres notas. Si el estudiante no obtuviere como mínimo setenta puntos sobre cien en la nota de sustentación, se presentará a una nueva sustentación en un plazo no menor a treinta días laborables, y de no aprobar en esta oportunidad el Trabajo de Grado o Tesis, se anulará y podrá denunciar otro tema por una sola vez. En caso de ser más de un estudiante, la nota deberá ser individual.
19. Para la sustentación se tomarán en cuenta los siguientes criterios:
- a) Duración: el tiempo estimado para la sustentación es de 60 minutos repartidos de la siguiente manera: 20 para la exposición del estudiante; 30 para la formulación y contestación a las preguntas de los miembros del tribunal; 10 para la calificación. En caso de ser más de un estudiante, se añadirá 20 minutos para la exposición de cada estudiante adicional.
  - b) Exposición: se deben considerar el uso de recursos o ayudas didácticas, el manejo de la temática, la capacidad de síntesis y la claridad. Se debe considerar también el manejo del tiempo.
  - c) Respuestas: pertinencia y solvencia para responder a las preguntas formuladas por el tribunal.

20. La nota final de graduación será el resultado de las notas parciales correspondientes en los siguientes porcentajes:
- a. Promedio académico 50%
  - b. Nota del trabajo escrito 30%
  - c. Nota de sustentación 20%
21. Terminada la sustentación, Secretaria distribuirá los ejemplares empastados de la siguiente manera: Uno al Director del Trabajo de Grado o Tesis, otro para la biblioteca, y el restante para el o los estudiantes.
22. El Director del Trabajo de Grado o Tesis, sugerirá al Director de Carrera, la difusión a través de una conferencia y/o publicación del citado trabajo si el caso amerita.

## **B. DISEÑO DE LA INVESTIGACIÓN**

El diseño de Investigación que se utilizara en el proyecto de Tesis es el experimental, por lo tanto se detallaran las características propias de este diseño.. El diseño es la manera de obtener la información deseada, es decir el plan que el investigador se traza para conseguir los objetivos planteados, surge cuando se ha definido el problema de investigación, la perspectiva teórica, el alcance y las hipótesis si existieran.

El diseño de la Investigación se clasifica en:

- Investigación Experimental
- Investigación no Experimental

### **B.1 INVESTIGACIÓN EXPERIMENTAL**

Empezamos definiendo que es un Experimento, el cual se refiere a “un estudio en el que se manipulan intencionalmente una o más variables independientes, por analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes, dentro de una situación de control para el investigador<sup>45</sup>”.

#### **B.1.1 REQUISITOS DE UN EXPERIMENTO**

Un experimento necesita varios requisitos para poderse desarrollar

##### **B.1.1.1 MANIPULACIÓN INTENCIONAL DE UNA O MÁS VARIABLES INDEPENDIENTES**

En donde el investigador manipula la variable independiente y observa si la dependiente varía o no....

---

<sup>45</sup> FUNDAMENTOS DE METODOLOGÍA DE LA INVESTIGACIÓN, HERNANDEZ SAMPIERI, Roberto; FERNANDEZ COLLADO, Carlos; BAPTISTA LUCIO, Pilar, Quinta Edición, 2007.

La variable dependiente es el efecto provocado por la causa o antecedente, la cual se mide para ver el efecto que la manipulación de la variable independiente tiene en ella. También se la denomina consecuente.

La variable independiente es la causa o condición antecedente, la cual manipula la variable dependiente en dos o más grados. Cada nivel o grado implica un grupo en el experimento. El grado mínimo es dos y corresponde a presencia-ausencia que implica que un grupo es expuesto a la variable independiente o estímulo experimental y el otro grupo no, al final se compara los dos grupos para ver en que difieren, al primer grupo se lo denomina grupo experimental y al segundo se le denomina grupo de control. Ambos participan en el experimento, con las mismas actividades excepto someterse al estímulo experimental.

El Experimento de más de dos grados determina si distintos niveles de la variable independiente producen diferentes efectos, mientras más niveles mayor información pero también mayor complicación y cada nivel adicional implica un grupo más.

También se puede manipular la variable independiente a través de diferentes modalidades, sin que esto implique cantidad. Por ejemplo, si se utiliza 3 métodos de enseñanza, se expone a un grupo al método 1, al otro grupo al método 2 y al otro grupo al método 3, y luego se compara el aprendizaje de los grupos.

Para definir como se va a manipular una variable es necesario:

- a) Consultar experimentos antecedentes para ver si en estos resultado exitosa la forma de manipular la variable y como adaptarlo a nuestra situación experimental.
- b) Evaluar la manipulación antes que se conduzca el experimento, es decir responder a preguntas como ¿Las operaciones experimentales representan la variable conceptual que se tiene en mente?, ¿Los diferentes niveles de variación de la variable independiente harán que los sujetos se comporten de forma distinta? (Christensen, 2000) ¿Qué otras maneras existen para

manipular la variable?, ¿Esta es la mejor? , ya que la manipulación errónea provocaría: Que el experimento no sirva para nada, Vivamos en el error, creyendo que “algo es” cuando verdaderamente no lo “es” o que Obtengamos resultados que no nos interesen

- c) Incluir verificaciones para la manipulación, ya sea entrevistando a los sujetos o incluyendo mediciones relativas a la manipulación durante el experimento.

Hay que tomar en cuenta también que si la presencia de la variable independiente es débil, probablemente no se encontraran efectos, pero no porque no pueda haberlos.

#### **B.1.1.2 MEDIR EL EFECTO QUE LA VARIABLE INDEPENDIENTE TIENE EN LA VARIABLE DEPENDIENTE.**

Es decir que sino podemos asegurar que se midió de manera adecuada, los resultados no servirán y el experimento será una pérdida de tiempo. El número de variables dependientes e independientes depende de cómo haya sido planteado el problema de investigación y de las limitaciones que existan, tomando en cuenta que al aumentar la variable independiente aumenta los grupos y al aumentar las variables dependientes aumenta el tamaño de la medición por que hay más variables que medir.

#### **B.1.1.3 CONTROL O VALIDEZ INTERNA DE LA SITUACIÓN EXPERIMENTAL**

Es decir, saber qué está ocurriendo realmente con la relación entre las variables independientes y las dependientes, esto es, controlar la influencia de otras variables extrañas sobre las variables dependientes, para así saber en realidad si las variables independientes que nos interesan tienen o no efecto en las dependientes. Se relaciona con la calidad del experimento y se logra cuando hay control. Existen diferentes factores que provocan invalidación interna:

- a) Acontecimientos que ocurren durante el desarrollo del experimento, que pueden afectar a la variable dependiente y llegan a confundir los resultados experimentales.

- b) Procesos internos de los participantes que operan como consecuencia del tiempo y que afectan a los resultados del experimento, tales como cansancio, hambre, aburrimiento, aumento de la edad, etc.
- c) Instrumentos de medición no equivalentes entre los grupos, las cuales son capaces de producir variaciones en los resultados que se obtengan.
- d) Grupos no equivalentes o no comparables.
- e) Pérdida de participantes desigual entre los grupos que se comparan.
- f) El experimentador, ya que no es un sujeto pasivo en el experimento y puede influir en los resultados del estudio, ya que a él le interesa probar lo que hipotetiza, ya sea consciente o inconscientemente, para solucionar esto quien trate con los sujetos debe ser alguien que no conozca las hipótesis, las condiciones experimentales, ni los propósitos del estudio, sino solo reciba instrucciones precisas sobre lo que debe hacer y como hacerlo.

Por lo tanto es necesario eliminar estas fuentes de invalidación interna mediante el control para conocer el efecto de la variable independiente sobre la dependiente.

#### **B.1.1.3.1 COMO LOGRAR LA VALIDEZ INTERNA**

1. Varios grupos de comparación en el experimento, mínimo dos, uno al que le administra el estímulo y otro al que no, ya que esto nos ayuda a averiguar el efecto de distintos niveles o modalidades de la variable independiente.
2. Equivalencia de los grupos en todo, excepto en la manipulación de la(s) variable(s) independiente(s), ya que la diferencia entre los grupos puede atribuirse a ella y no a otros factores. De igual forma los instrumentos de medición deben ser iguales y aplicados de la misma manera. Aquí se debe cumplir dos condiciones:
  - Equivalencia inicial: implica que los grupos son similares entre si al momento de iniciarse el experimento, una de las maneras de lograrlo es la asignación al azar de los sujetos a los grupos del experimento, esta

funciona mejor cuanto mayor sea el número de sujetos con el que se cuenta para el experimento

- Equivalencia durante el experimento: durante el experimento los grupos deben mantenerse similares en todo lo concerniente al desarrollo experimental, excepto la manipulación de la variable independiente.

## **B.1.2 DISEÑOS ESPECÍFICOS DE EXPERIMENTOS**

### **B.1.2.1 DISEÑO CON POSPRUEBA ÚNICAMENTE Y GRUPO DE CONTROL**

Incluye dos grupos, uno recibe el tratamiento experimental y el otro no, es decir presencia-ausencia de la variable independiente, los sujetos se asignan de forma aleatoria, al final del periodo experimental a ambos grupos se les administra una medición sobre la variable dependiente en estudio.

Wiersma (1999) comenta que de preferencia la posprueba debe administrarse inmediatamente después de que concluya el experimento, en especial si la variable dependiente tiende a cambiar con el paso del tiempo, además se aplica en forma simultánea a ambos grupos. La comparación entre las pospruebas de ambos grupos nos indica si hubo o no efecto en la manipulación.

### **B.1.2.2 DISEÑO CON PREPRUEBA-POSPRUEBA Y GRUPO DE CONTROL**

Incorpora la administración de pre-pruebas a los grupos que componen el experimento simultáneamente. Los sujetos son asignados al azar a los grupos, uno de ellos recibe el tratamiento experimental y otro no. Al final se le administra una posprueba simultáneamente.

Una ventaja de la pre-prueba es que sus puntuaciones sirven para fines de control en el experimento, ya que la comparación de la pre-prueba de los grupos sirve para evaluar que tan adecuada fue la aleatorización. Otra ventaja radica en que es posible analizar el puntaje-ganancia de cada grupo.

### **B.1.2.3 DISEÑOS EXPERIMENTALES DE SERIES CRONOLÓGICAS MÚLTIPLES.**

Es un diseño de serie cronológica ya que se efectúa a través del tiempo varias mediciones sobre una variable experimental, es decir este diseño permite analizar efectos en el mediano y largo plazo, ya que se basa en suponer que la influencia de la variable independiente sobre la dependiente tarda en manifestarse.

De igual forma se tiene dos o más grupos y los sujetos son asignados al azar a dichos grupos. Una ventaja es que es posible evaluar la evolución comparativa de los grupos.

### **B.1.2.4 DISEÑOS DE SERIES CRONOLÓGICAS CON REPETICIÓN DEL ESTÍMULO.**

Surge debido a que en ocasiones el investigador anticipa que el tratamiento o estímulo experimental no tiene efecto o es mínimo si se aplica una sola vez. Así por ejemplo no se vería resultados si se realiza ejercicio un solo día o si se suministra vitaminas una sola vez. Los sujetos se asignan al azar a los distintos grupos y a cada grupo se le administra varias veces el tratamiento experimental que le corresponde.

### **B.1.3 VALIDACIÓN EXTERNA.**

Se refiere a qué tan generalizables son los resultados de un experimento a situaciones no experimentables y a otros sujetos o poblaciones. Por ejemplo, un experimento con métodos de aprendizaje, en donde los resultados pueden generalizarse a la enseñanza cotidiana en las escuelas de educación básica, el experimento tendrá validez externa.

#### **B.1.3.1 FUENTES DE INVALIDACIÓN EXTERNA.**

##### **a) EFECTO REACTIVO O DE INTERACCIÓN DE LAS PRUEBAS:**

Es cuando la prepueba aumenta o disminuye la sensibilidad o calidad de la reacción de los sujetos a la variable experimental, haciendo que los resultados obtenidos para una población con prepueba no puedan generalizarse a quienes forman parte de esa población aunque sin preprueba (Campbell y Stanley, 1966).

**b) EFECTO DE SELECCIONAR PERSONAL ATÍPICAS**

Es elegir personas con una o varias características que hagan que el tratamiento experimental produzca un efecto, que no se daría si las personas no tuvieran esas características.

**c) ARTIFICIALIDAD DE LA SITUACIÓN EXPERIMENTAL**

La artificialidad de las condiciones puede hacer que el contexto experimental resulte atípico, respecto a la manera en que se aplica regularmente el tratamiento (Campbel, 1975), por ello, el experimentador tiene que ingeniárselas para hacer que los sujetos se olviden que están en un experimento y sentirse lo menos observados posible.

**d) IMPOSIBILIDAD DE REPLICAR LOS TRATAMIENTOS.**

Cuando los tratamientos son tan complejos que no pueden replicarse en situaciones no experimentales, es difícil generalizar a estas.

Para lograr mayor validez externa, es conveniente tener grupos lo más parecidos posible a la mayoría de personas a quienes se desea generalizar y repetir el experimento varias veces con diferentes grupos, además de que el contexto experimental sea lo más similar posible al contexto que se pretende generalizar.

### **B.1.3.2 CONTEXTOS DE EXPERIMENTOS**

Los contextos de un experimento pueden ser de laboratorio o de campo.

**EXPERIMENTO DE LABORATORIO:** Es un estudio de investigación en el que el efecto de todas o casi todas las variables independientes influyentes posibles no pertinentes al problema de investigación se mantiene reducido en un mínimo (Kerlinger, 2002). Generalmente se logra un control más riguroso.

**EXPERIMENTO DE CAMPO:** Es un estudio de investigación en una situación más realista en la que una o más variables independientes son manipuladas por el experimentador en condiciones tan cuidadosamente controladas como lo permite la situación. (Kerlinger, 2002). Este experimento suele tener mayor validez externa.

#### **B.1.4 CUASIEXPERIMENTOS.**

Son experimentos pero con menor control, rigor y validez. Manipulan al menos una variable independiente para observar su efecto y relación con una o más variables dependientes. Aquí los sujetos no se asignan al azar a los grupos sino que ya están formados antes del experimento, lo que puede provocar posibles problemas de validez interna y externa.

##### **B.1.4.1 TIPOS DE DISEÑOS CUASIEXPERIMENTALES**

Los tipos de los cuasi experimentos son parecidos a los experimentos con la diferencia que no hay asignación al azar.

###### **a) DISEÑO CON POSPRUEBA ÚNICAMENTE Y GRUPOS INTACTOS**

Utiliza dos grupos, uno recibe el tratamiento experimental y el otro no, los grupos son comparados en la posprueba para analizar si el tratamiento experimental tuvo un efecto sobre la variable dependiente. Hay problemas de validez interna.

###### **b) DISEÑO CON PREPUEBA-POSPRUEBA Y GRUPOS INTACTOS**

Es similar al anterior solo que a los grupos se les administra una preprueba, la cual puede servir para verificar la equivalencia inicial de los grupos. En la interpretación de los resultados debemos tomar en cuenta que los grupos son intactos.

#### **B.1.5 PASOS DE UN EXPERIMENTO O CUASIEXPERIMENTOS<sup>46</sup>**

---

<sup>46</sup> FUNDAMENTOS DE METODOLOGÍA DE LA INVESTIGACIÓN, HERNANDEZ SAMPIERI, Roberto; FERNANDEZ COLLADO, Carlos; BAPTISTA LUCIO, Pilar, Quinta Edición, 2007.

1. Decir cuantas variables independientes y dependientes deberán incluirse en el experimento o cuasi experimento.
2. Elegir los niveles de manipulación de la(s) variable(s) independiente(s) y traducirlos en tratamientos experimentales.
3. Desarrollar el instrumento o instrumentos para medir la(s) variable(s) dependiente(s).
4. Seleccionar una muestra de personas para el experimento, idealmente representativa de la población.
5. Reclutar a los sujetos del experimento o cuasi experimento.
6. Seleccionar el diseño experimental o cuasiexperimental apropiado para nuestras hipótesis, objetivos y preguntas de investigación.
7. Planear cómo vamos a manejar a los sujetos que participen en el experimento
8. En el caso de experimentos dividirlos al azar y en caso de cuasi experimentos, analizar cuidadosamente las propiedades de los grupos intactos.
9. Aplicar las prepuebas cuando las haya, los tratamientos respectivos, cuando no se trate de grupos de control y las pospruebas.
10. Tomar nota del desarrollo del experimento, llevar una bitácora minuciosa de todo lo ocurrido a lo largo de este.

# ANEXO C

## C. DIAGRAMAS DEL SISTEMA

### C1. DIAGRAMA ENTIDAD RELACIÓN

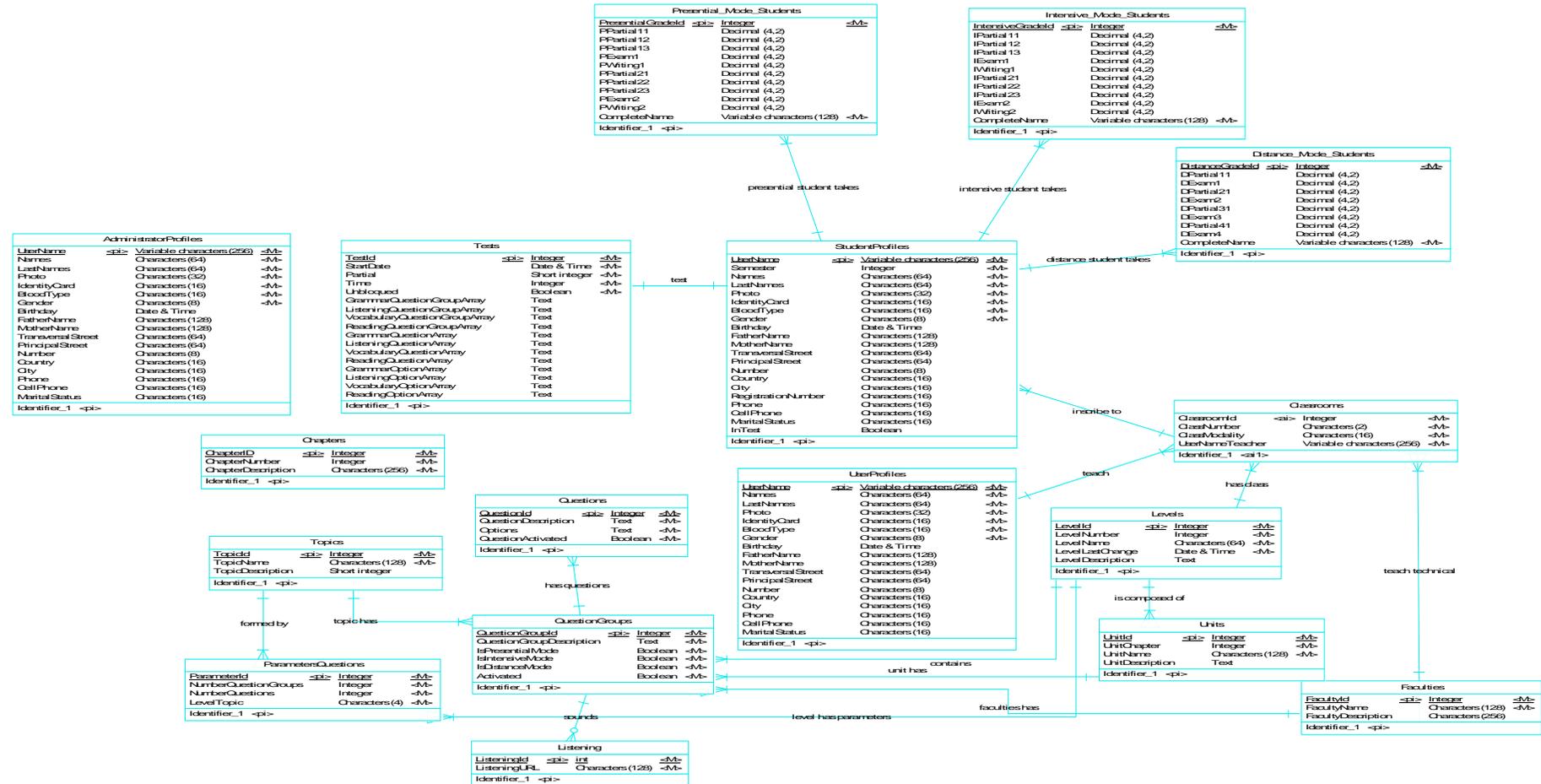


Figura C.1. Diagrama Entidad-Relación.

## C2. DIAGRAMA FÍSICO

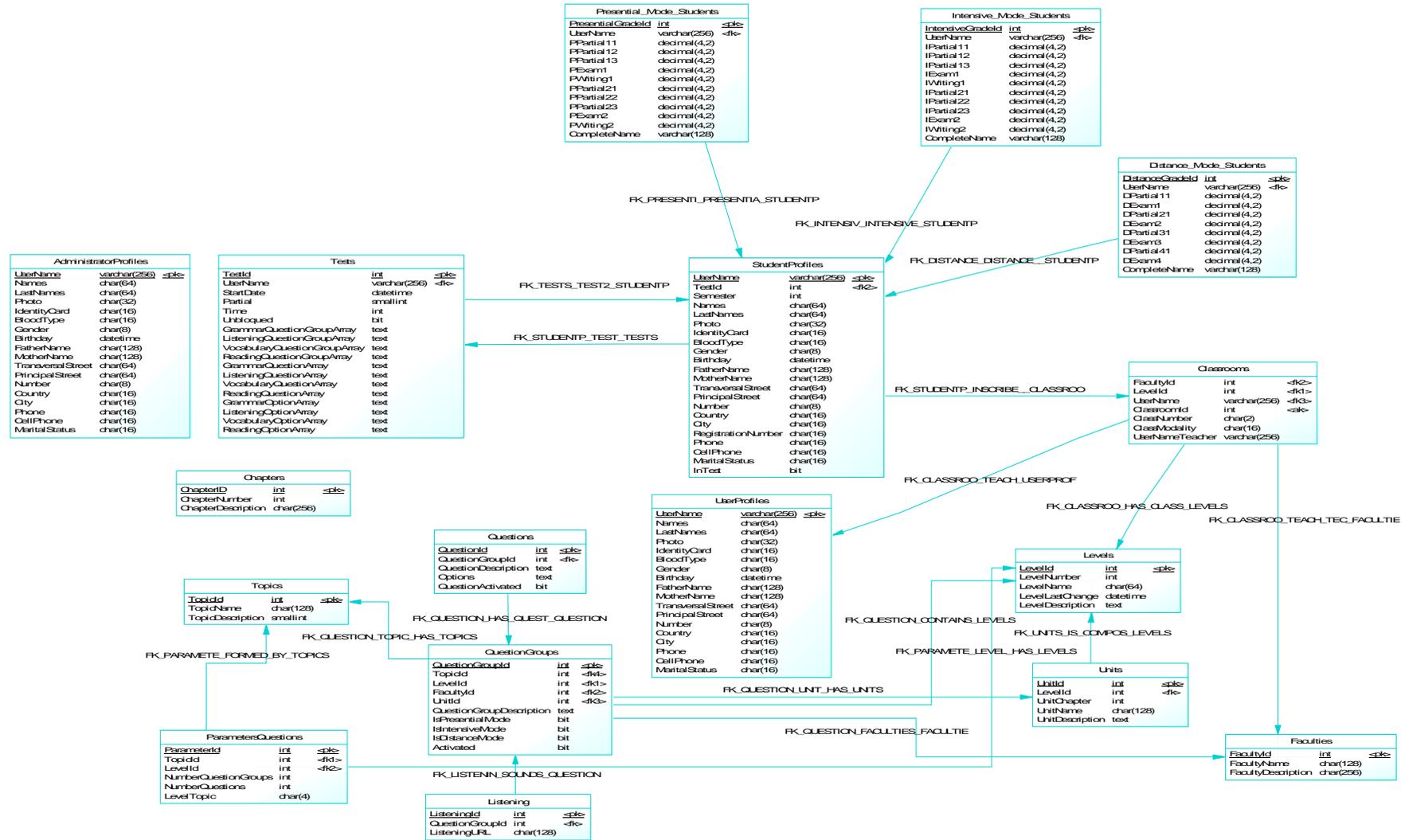


Figura C.2. Diagrama Físico.

### C3. DIAGRAMA DE CLASES.

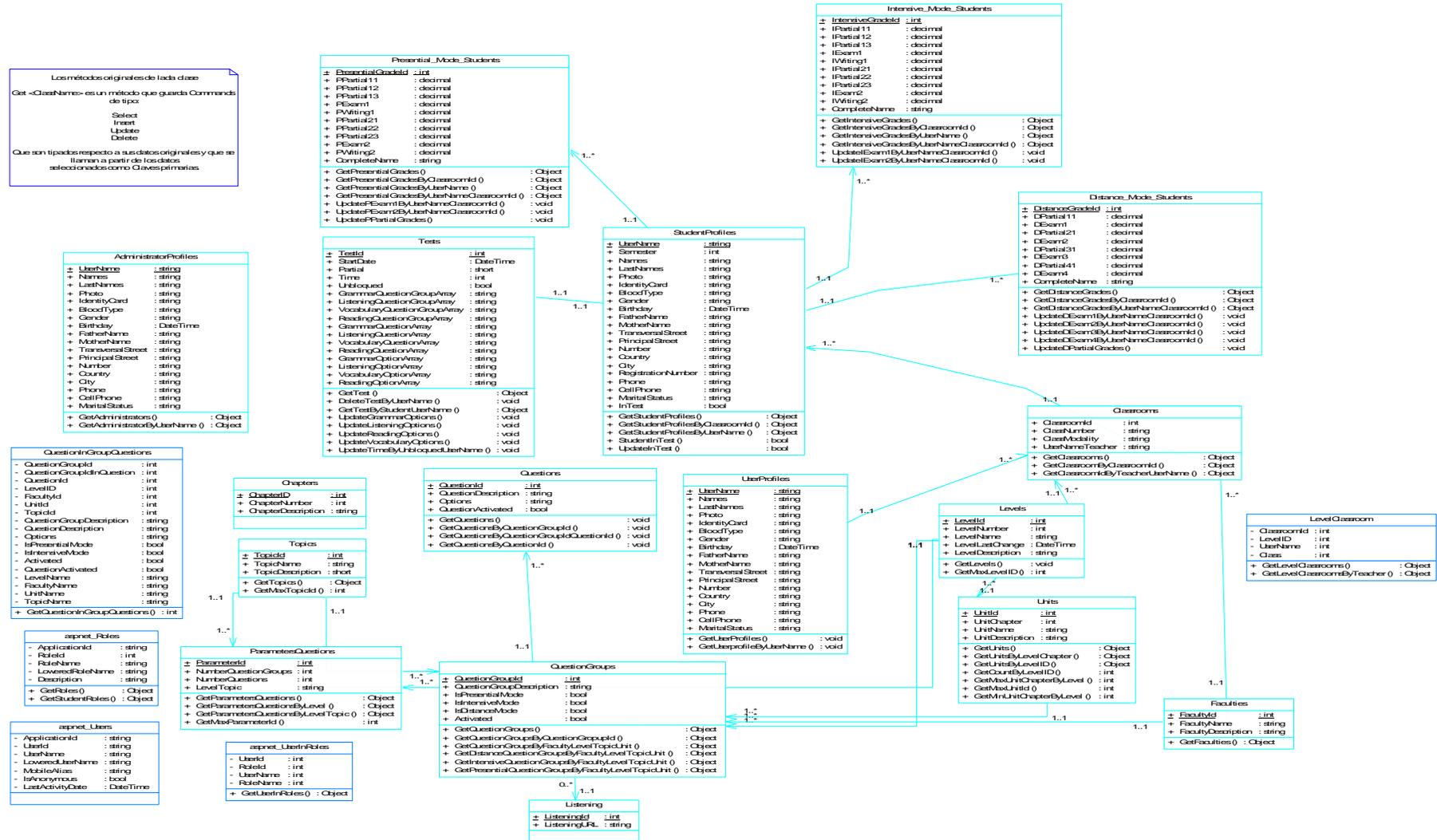


Figura C.3 Diagrama de Clases.

## D. PLANTILLAS PARA LA CREACIÓN DE USUARIOS DEL SISTEMA

### LANGUAGE DEPARTMENT

#### D.1 TEMPLATE TO CREATE *ADMINISTRATOR* USER

- **PERSONAL INFORMATION**

Names: ..... Last Names: .....

Identity Card: ..... Blood Type: ..... Birthday: .....

Gender: Male ..... Female .....

- **DETAILS**

Email: .....

- **CONTROL INFORMATION**

**Security Questions:**

**Answer**

- Mother's main name:.....

- Favorite Pet: .....

- Who was your childhood hero? : .....

- Your favorite hobby? : .....

Father's Name: ..... Mother's Name: .....

- **ADDITIONAL INFORMATION**

Principal Street: ..... Transversal Street: .....

Street Number: .....

Country: ..... City: .....

Marital Status: single..... married ..... divorced..... cohabitation.....

Phone..... Cell phone.....

## LANGUAGE DEPARTMENT

### D.2 TEMPLATE TO CREATE *TEACHER Y/O EDITOR* USER

- **PERSONAL INFORMATION**

Names: ..... Last Names: .....  
 Identity Card: ..... Blood Type: ..... Birthday: .....  
 Gender: Male ..... Female .....

- **DETAILS**

Email: .....

- **ROLES**

Teacher: ..... Editor: .....

- **CONTROL INFORMATION**

**Security Questions:**

**Answer**

- Mother's main name: .....
- Favorite Pet: .....
- Who was your childhood hero? : .....
- Your favorite hobby? : .....

Father's Name: ..... Mother's Name: .....

- **ADDITIONAL INFORMATION**

Principal Street: ..... Transversal Street: .....

Street Number: .....

Country: ..... City: .....

Marital Status: single..... married ..... divorced..... cohabitation.....

Phone..... Cell phone.....

## LANGUAGE DEPARTMENT

### D.3 TEMPLATE TO CREATE *STUDENT* USER

- **PERSONAL INFORMATION**

Names: ..... Last Names: .....  
 Identity Card: ..... Blood Type: ..... Birthday: .....  
 Gender: Male ..... Female .....

- **DETAILS**

Email: .....

- **ACADEMIC INFORMATION**

**Language Level:**

One: ..... Two: ..... Three: ..... Four: .....

Technical One: ..... Technical Two: .....

Classroom: ..... Semester: .....

**Modality:**

Presential Mode ..... Intensive Mode ..... Distance Mode .....

- **CONTROL INFORMATION**

**Security Questions:**

**Answer**

- Mother's main name: .....
- Favorite Pet: .....
- Who was your childhood hero? : .....
- Your favorite hobby? : .....

Father's Name: ..... Mother's Name: .....

- **ADDITIONAL INFORMATION**

Principal Street: ..... Transversal Street: .....

Street Number: ..... Country: ..... City: .....

Marital Status: single..... married ..... divorced..... cohabitation.....

Phone..... Cell phone.....

## ANEXO E

### MANUAL DE USUARIO ADMINISTRADOR

#### 1. INTRODUCCIÓN

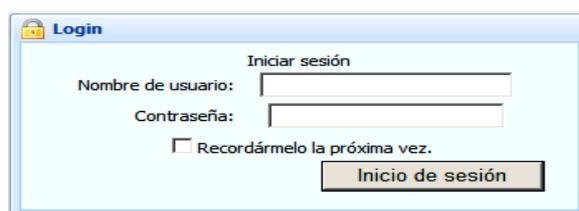
Este documento es una guía para el Administrador del Sistema de Evaluación de Inglés desarrollado para el Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur, que explica los aspectos necesarios para la creación y administración de usuarios así como la organización de cursos, profesores e ítems para las evaluaciones respectivas.

Se sugiere que previo a la creación de usuarios el Administrador de usuarios obtenga toda la información requerida de cada usuario de acuerdo a las plantillas propuestas para dicha información.

#### 2. ACCESO AL SISTEMA

- Para ingresar al sistema es necesario un browser, se sugiere Microsoft Internet Explorer 6.0 o superior.
- Cada usuario del sistema cuenta con diferentes accesos según su perfil.
- Al ingresar a la URL respectiva, se debe ingresar el username y password respectivos.

En la siguiente figura se muestra la información que es requerida para el ingreso al sistema (username y password), una vez ingresado se presiona el botón de inicio de sesión.



The image shows a web browser window with a title bar that says "Login". The main content area has a light blue background and contains the following elements: the text "Iniciar sesión" at the top center; a label "Nombre de usuario:" followed by a text input field; a label "Contraseña:" followed by a text input field; a checkbox with the text "Recordármelo la próxima vez." below it; and a button with the text "Inicio de sesión" at the bottom right.

**Figura E.1 Inicio de Sesión**

### 3. MENU DEL SISTEMA ADMINISTRADOR

#### 3.1 MANAGER USER (ADMINISTRACIÓN DE USUARIOS)

Nos permite la administración de los usuarios, es decir, crear, borrar, editar y actualizar

##### 3.1.1 ADMINISTRADOR. Permite crear el usuario administrador

Una vez en el sistema.

3.1.1.1 Seleccionar la opción Manage User –Administrator en menú del lado izquierdo

3.1.1.2 Completar todos los campos requeridos.

3.1.1.3 Presionar el botón Create User.

**Figura E.2 Crear Usuarios**

3.1.1.4 USUARIO TEACHER Y/O EDITOR. Permite crear el usuario Teacher y/oEditor

Una vez en el sistema

3.1.1.5 Seleccionar la opción Manage User –Teachers and Editors en menú del lado izquierdo

3.1.1.6 Completar todos los campos requeridos.

3.1.1.7 Presionar el botón Create User.

### 3.1.2 USUARIO STUDENTS. Permite crear el usuario Student

Una vez en el sistema

3.1.2.1 Seleccionar la opción Manage User –Students en menú del lado izquierdo.

3.1.2.2 Completar todos los campos requeridos.

3.1.2.3 Presionar el botón Create User.

Las siguientes opciones nos permiten Borrar, Editar y Actualizar los usuarios.

### 3.1.3 ADMINISTRATOR CONTROL

Una vez en el sistema

#### EDITAR

The image shows two overlapping windows from a software application. The left window, titled 'Add User', has a sidebar menu with 'Administrator Control' circled in red and labeled '1'. The main area of this window is titled 'Parameters of Test' and contains a 'Name:' dropdown menu with 'Condor Caiza' selected, circled in red and labeled '2'. Below it is a 'Complete Student Name:' list box containing 'Condor Caiza' and 'Julio David'. There are also fields for 'Phone:', 'Cellphone:', and a 'Delete User' button. The right window, titled 'Form', has a 'DETAILS' section with a 'User Name:' field containing '1700000001' and an 'Update Data' button circled in red and labeled '3'. Below this is a 'PERSONAL INFORMATION' section with 'Names:' (Julio David) and 'Last Names:' (Condor Caiza) fields. There is also a 'Photo 92x70:' field with an 'Examinar...' button. At the bottom, there are fields for 'Identity Card:' (1700000001), 'Blood Type:' (O+), 'Gender:' (Male), and 'Birthday:' (10/13/1979).

Figura E.3 Editar Usuarios

3.1.3.1 Seleccionar Manager User – Administrador Control.

3.1.3.2 En Parameter of users, escoger el usuario a Editar, cambiar los cambios que sean necesarios

3.1.3.3 Pulsar el botón Upadote Data

## BORRAR

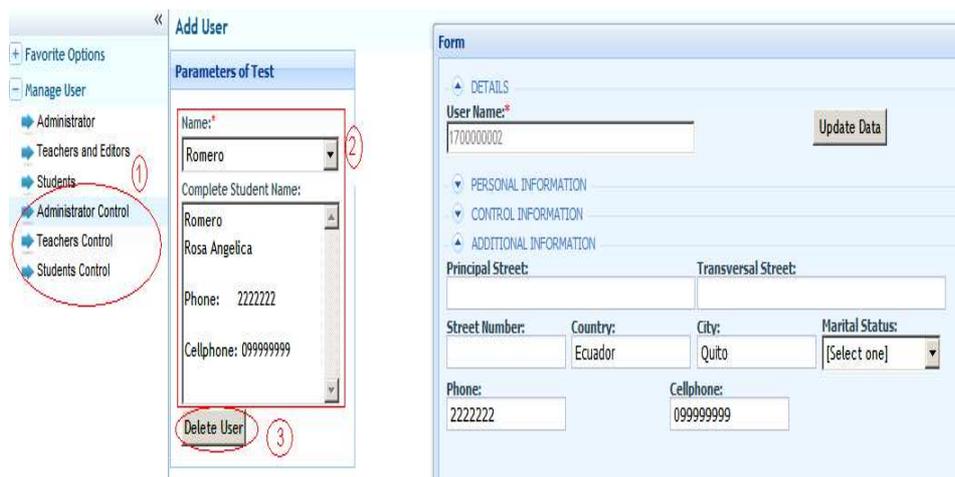


Figura E.4 Borrar Usuarios.

3.1.3.4 Seleccionar Manager User – Administrador Control

3.1.3.5 En Parameter of users, escoger el usuario a Borrar.

3.1.3.6 Pulsar Delete User

## 3.1.4 TEACHER CONTROL

Una vez en el sistema

## EDITAR

3.1.4.1 Seleccionar Manager User – Teacher Control.

3.1.4.2 En Parameter of users, escoger el usuario a Editar, cambiar los cambios que sean necesarios

3.1.4.3 Pulsar el botón Upadote Data

**BORRAR**

3.1.4.4 Seleccionar Manager User – Teacher Control

3.1.4.5 En Parameter of users, escoger el usuario a Borrar.

3.1.4.6 Pulsar Delete User

**3.1.5 STUDENT CONTROL**

Una vez en el sistema

**EDITAR**

3.1.5.1 Seleccionar Manager User – Student Control.

3.1.5.2 En Parameter of users, escoger el usuario a Editar, cambiar los cambios que sean necesarios

3.1.5.3 Pulsar el botón Upadote Data

**BORRAR**

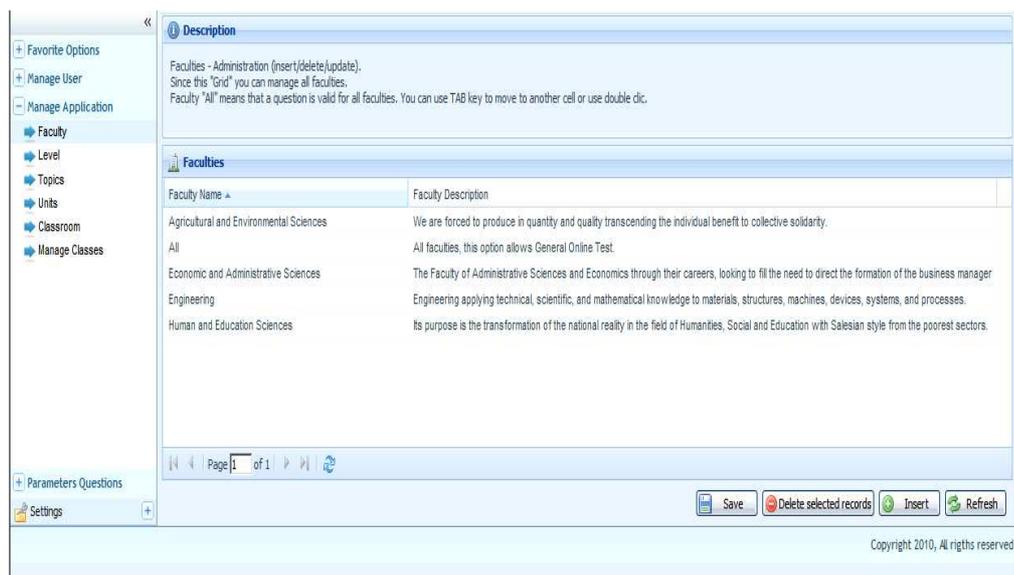
3.1.5.4 Seleccionar Manager User – Student Control

3.1.5.5 En Parameter of users, escoger el usuario a Borrar.

3.1.5.6 Pulsar Delete User.

**3.2 MANAGER APPLICATION.**

**3.2.1 FACULTIES.** Define las Facultades de la Universidad necesarias para determinar los niveles que un alumno debe tomar de acuerdo a su facultad



**Figura E.5 Manejo de Facultades**

### 3.2.1.1 INSERT.

3.2.1.1.1 Clic en Insert

3.2.1.1.2 Agregar Faculty Name y Faculty Description

3.2.1.1.3 Clic en Save para guardar los cambios.

### 3.2.1.2 DELETE SELECT RECORDS.

3.2.1.2.1 Seccionar un ítem

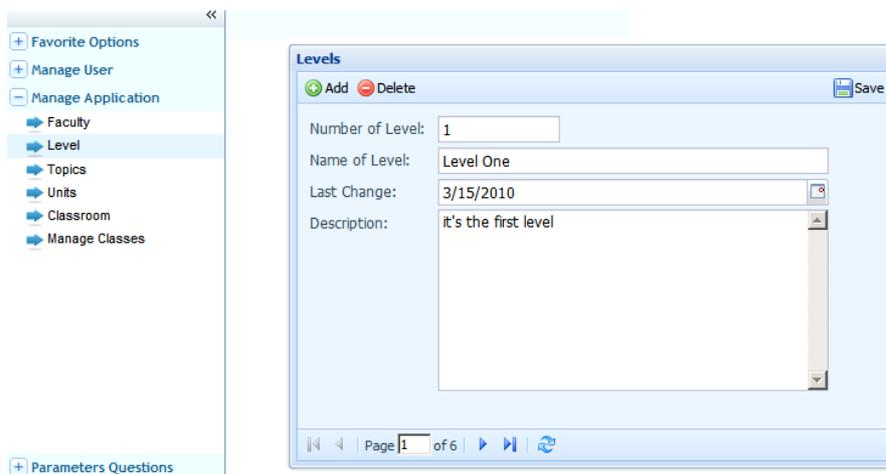
3.2.1.2.2 Clic en Delete selected records

3.2.1.2.3 Clic en Save para guardar los cambios.

### 3.2.1.3 REFRESH

Sirve para actualizar la página.

**3.2.2 LEVELS.** Define los niveles de inglés con que cuenta este departamento



**Figura E.6 Manejo de Niveles**

### **3.2.2.1 ADD**

- 3.2.2.1.1 Clic en Add
- 3.2.2.1.2 Agregar los campos requeridos
- 3.2.2.1.3 Clic en Save para guardar los cambios.

### **3.2.2.2 DELETE SELECT RECORDS.**

- 3.2.2.2.1 Seccionar un ítem
- 3.2.2.2.2 Click en Delete
- 3.2.2.2.3 Clic en OK.

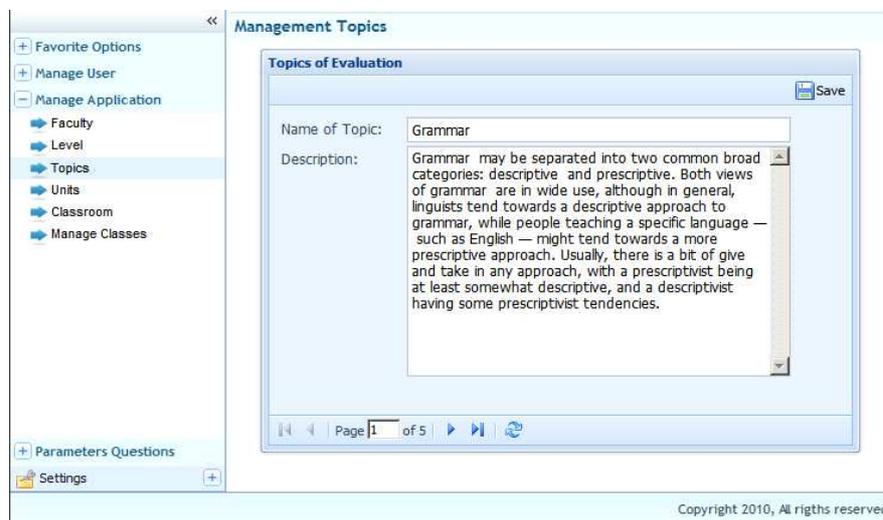
### **3.2.2.3 REFRESH**

Sirve para actualizar la página

### **3.2.2.4 BOTONES DE NAVEGACIÓN**

Sirve para desplazarse entre ítems.

**3.2.3 TOPIC.** Define los tópicos con que cuenta el examen online una descripción de cada uno de ellos, además del tópico writing que no es parte del examen online pero es parte de la nota global de que consta el examen.



**Figura E.7 Manejo de Tópicos**

#### **3.2.3.1 SAVE**

Se puede modificar la descripción de cada uno de los tópicos y guardar con el botón Save.

#### **3.2.3.2 REFRESH**

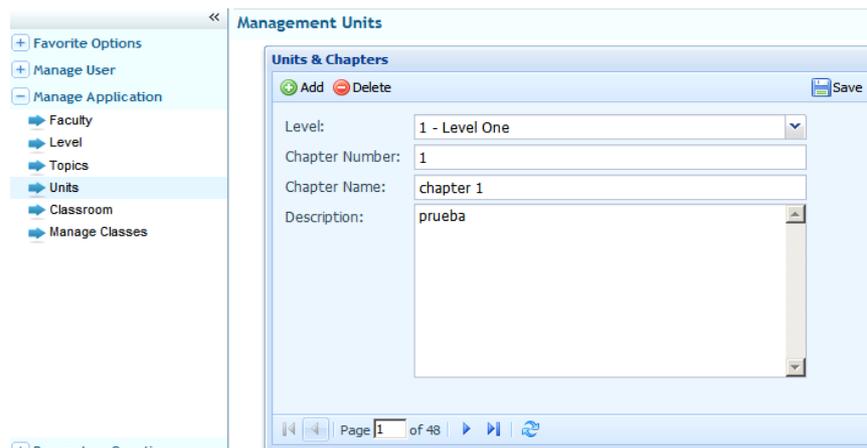
Sirve para actualizar la página

#### **3.2.3.3 BOTONES DE NAVEGACIÓN**

Sirve para desplazarse entre ítems.

### **3.2.4 UNITS**

Define los capítulos que corresponden a cada nivel. Cada nivel tiene 8 capítulos



**Figura E.8 Manejo de Unidades**

### **3.2.4.1 ADD**

3.2.4.1.1 Clic en Add

3.2.4.1.2 Agregar los campos requeridos

3.2.4.1.3 Clic en Save para guardar los cambios

### **3.2.4.2 DELETE**

3.2.4.2.1 Seccionar un ítem

3.2.4.2.2 Clic en Delete

3.2.4.2.3 Clic en OK.

### **3.2.4.3 REFRESH**

Sirve para actualizar la página

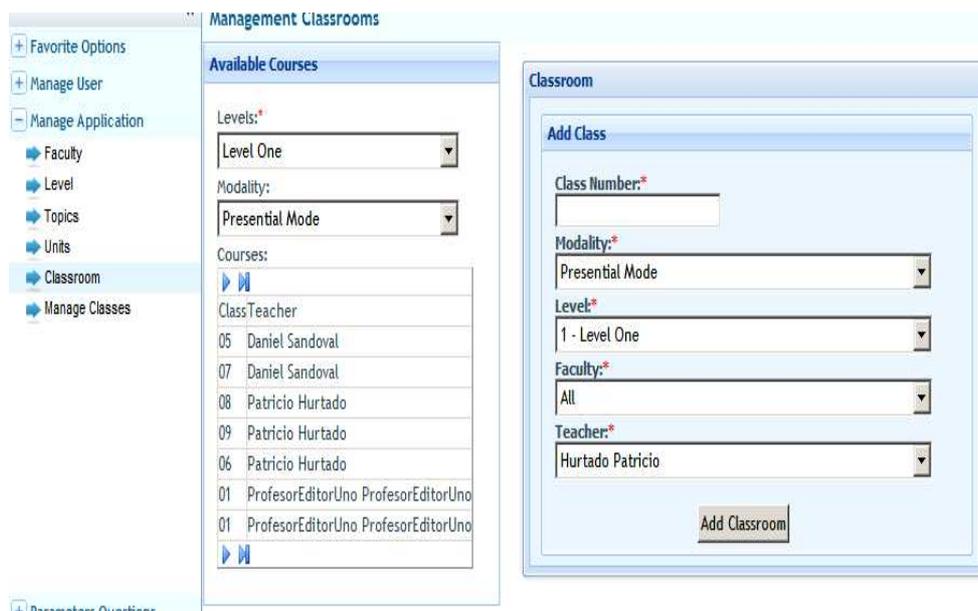
### **3.2.4.4 BOTONES DE NAVEGACIÓN**

Sirve para desplazarse entre ítems.

**3.2.5 CLASSROOMS.** Define las clases que le corresponde a cada profesor acorde a un nivel y modalidad.

En la sección Available Course se puede verificar los cursos ya asignados a los respectivos profesores de acuerdo al nivel y modalidad.

En la sección Add Classroom se sitúa el número de clase, modalidad, nivel, facultad y teacher para crear la nueva clase.



**Figura E.9 Agregar Classrooms**

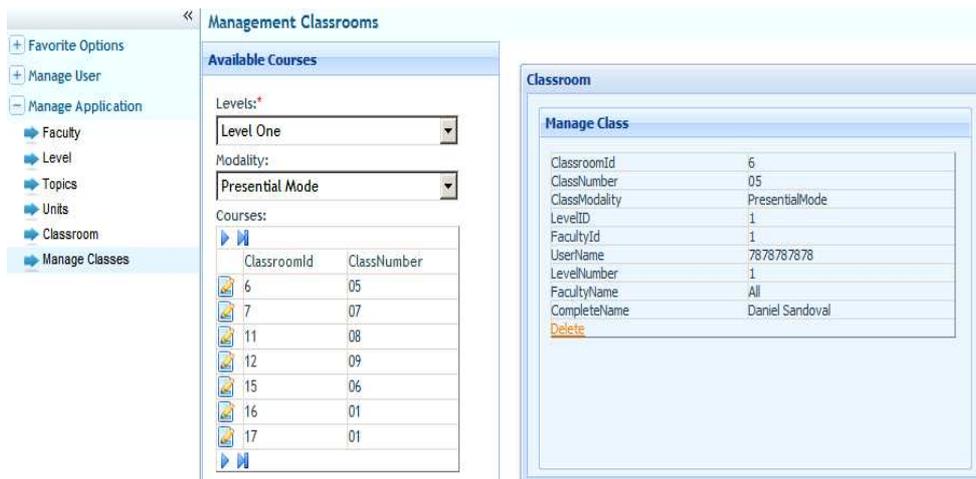
### 3.2.5.1 ADD

3.2.5.1.1 Agregar los campos requeridos

3.2.5.1.2 Clic en Add Classroom para guardar los cambios

### 3.2.6 MANAGE CLASSES

Permite la administración de los cursos asignados a los profesores obteniendo datos relevantes de los mismos al seleccionar cada curso.

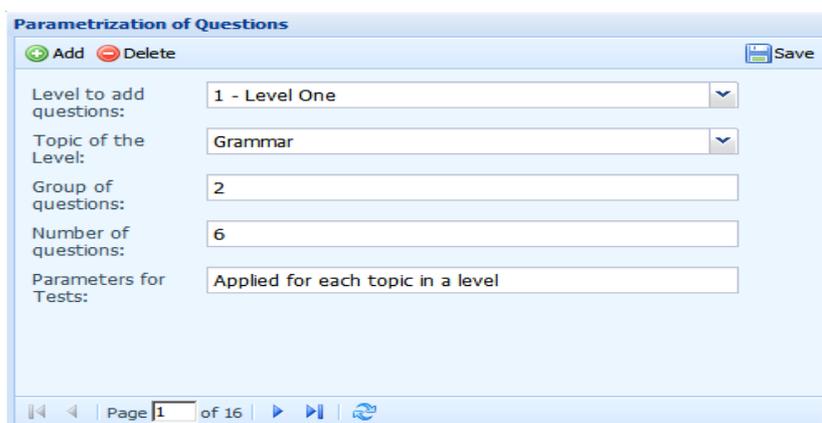


**Figura E.10 Manejo de Classrooms**

### 3.3 PARAMETER QUESTIONS.

#### 3.3.1 PARAMETERS

Esta opción permite parametrizar el examen que se generara para los estudiantes de los diferentes niveles.



**Figura E.11 Parámetros de las preguntas**

##### 3.3.1.1 ADD

3.3.1.1.1 Clic en Add

3.3.1.1.2 Agregar los campos requeridos

3.3.1.1.3 Clic en Save para guardar los cambios

### **3.3.1.2 DELETE**

3.3.1.2.1 Seccionar un ítem

3.3.1.2.2 Click en Delete

3.3.1.2.3 Clic en OK

### **3.3.1.3 REFRESH**

Sirve para actualizar la página

### **3.3.1.4 BOTONES DE NAVEGACIÓN**

Sirve para desplazarse entre ítems.

## **4. CONSIDERACIONES IMPORTANTE**

- El Username debe ser de 10 dígitos que corresponden al número de cédula del usuario con lo que se evita tener usuarios duplicados.
- Se sugiere tener un solo administrador de usuarios para mantener un mejor control sobre la seguridad del sistema.
- Es importante que se registre la mayor parte de la información de cada persona, por motivos de comunicación y localización.

## MANUAL DE USUARIO ESTUDIANTE

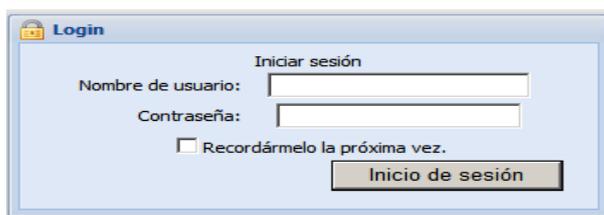
### 1. INTRODUCCIÓN

Este documento es una guía para el usuario Estudiante del Sistema de Evaluación de Inglés desarrollado para el Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur, que explica los aspectos necesarios para que este usuario pueda rendir el examen y revisar sus calificaciones.

### 2. ACCESO AL SISTEMA

- Para ingresar al sistema es necesario un browser, se sugiere Microsoft Internet Explorer 6.0 o superior.
- Cada usuario del sistema cuenta con diferentes accesos según su perfil.
- Al ingresar a la URL respectiva, se debe ingresar el username y password respectivos.

En la siguiente figura se muestra la información que es requerida para el ingreso al sistema (username y password), una vez ingresado se presiona el botón de inicio de sesión.

The image shows a web browser window with a title bar that says "Login". Inside the window, there is a form titled "Iniciar sesión". The form contains two text input fields: "Nombre de usuario:" and "Contraseña:". Below these fields is a checkbox labeled "Recordármelo la próxima vez.". At the bottom right of the form is a button labeled "Inicio de sesión".

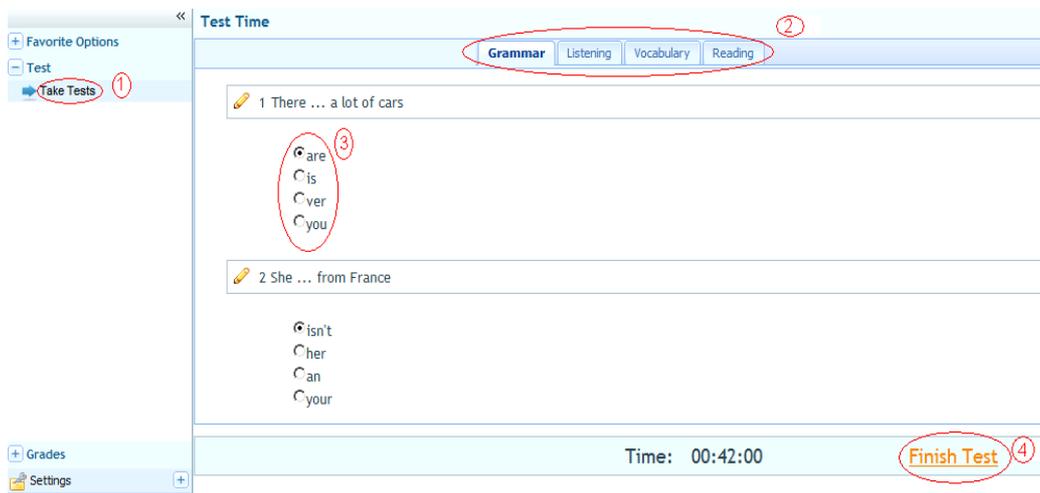
**Figura E.12 Inicio de Sesión**

### 3. MENU DEL SISTEMA STUDENT

#### 3.1 TEST

##### 3.1.1 TAKE TEST

1. Clic en Take test
2. Desplazarse a través de las pestañas que conforman el test
3. Escoger la respuesta apropiada
4. Clic en Finish Test para terminar el examen.

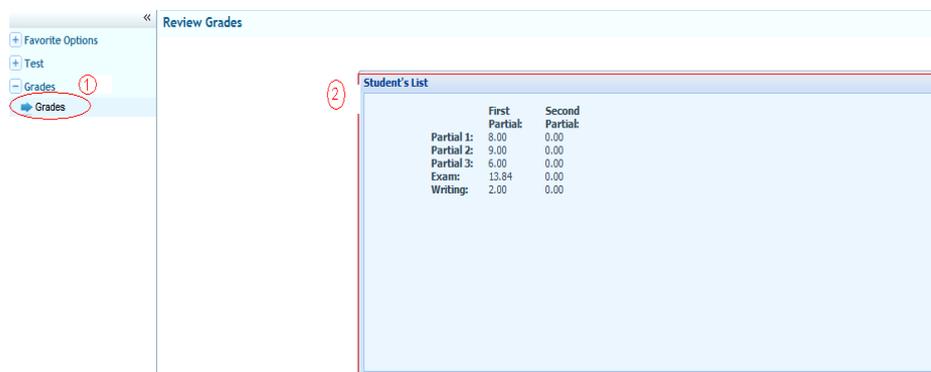


**Figura E.13 Take Test**

## 3.2 GRADES

### 3.2.1 GRADES

1. Clic en Grades
2. Revisar las notas



**Figura E.14 Grades**

## MANUAL DE USUARIO PROFESOR

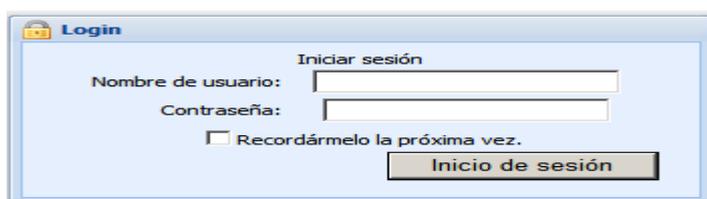
### 1. INTRODUCCIÓN

Este documento es una guía para el usuario Profesor del Sistema de Evaluación de Inglés desarrollado para el Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur, que explica los aspectos necesarios para que este usuario pueda generar el examen y asignar las calificaciones.

### 2. ACCESO AL SISTEMA

- Para ingresar al sistema es necesario un browser, se sugiere Microsoft Internet Explorer 6.0 o superior.
- Cada usuario del sistema cuenta con diferentes accesos según su perfil.
- Al ingresar a la URL respectiva, se debe ingresar el username y password respectivos.

En la siguiente figura se muestra la información que es requerida para el ingreso al sistema (username y password), una vez ingresado se presiona el botón de inicio de sesión.



The image shows a web browser window with a light blue background. At the top left, there is a small icon of a padlock and the word "Login". In the center, the text "Iniciar sesión" is displayed. Below this, there are two input fields: "Nombre de usuario:" followed by a text box, and "Contraseña:" followed by a text box. Underneath the password field is a checkbox with the text "Recordármelo la próxima vez.". At the bottom right, there is a button with the text "Inicio de sesión".

**Figura E.15 Inicio de Sesión**

### 3. MENU DEL SISTEMA PROFESOR

#### 3.1 ACADEMIC OPTIONS

##### 3.1.1 GENERATE TEST

1. En el panel Parameters of Test Seleccionamos la fecha, el tiempo, la clase y el parcial que se va a generar.

2. En Student List seleccionamos los estudiantes a los que se les generará el test.

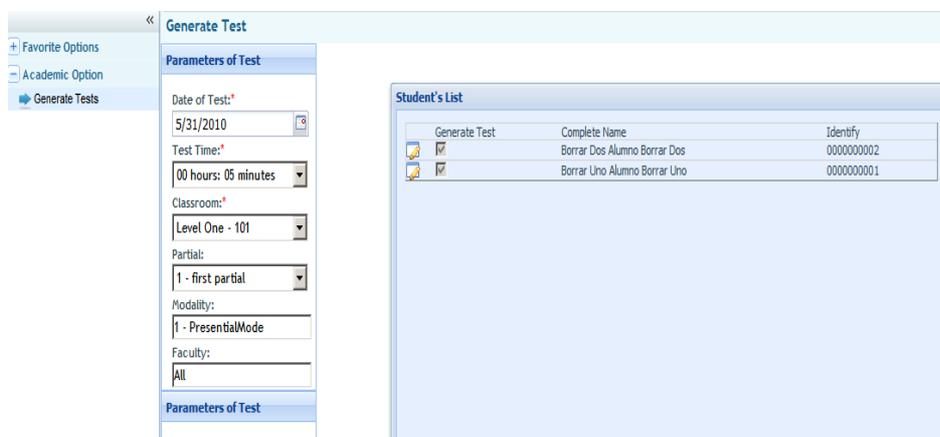


Figura E.16 Generar Examen

## 3.2 INFORMATION

### 3.2.1 GRADES

1. Clic en grades
2. Escoger un classroom
3. Clic en Check Grades
4. Seleccionar un alumno
5. Colocar las notas correspondientes
6. Guardar las notas agregadas.

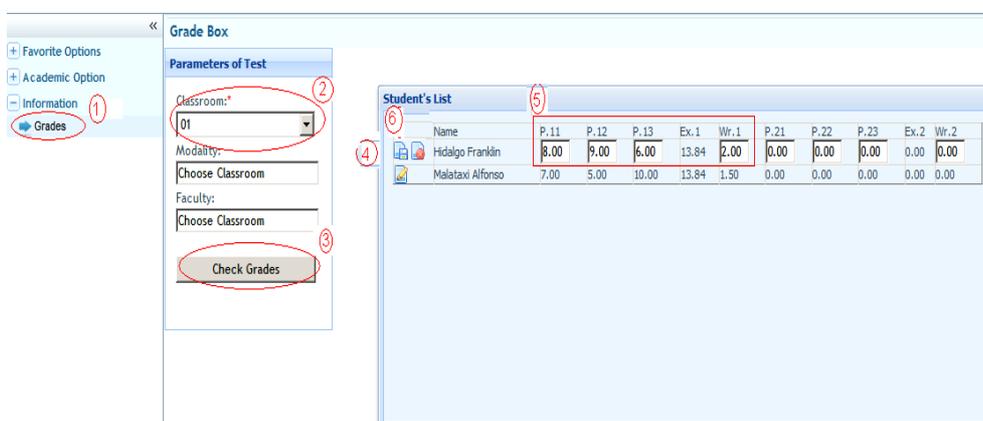


Figura E.17 Agregar Calificaciones

## MANUAL DE USUARIO EDITOR

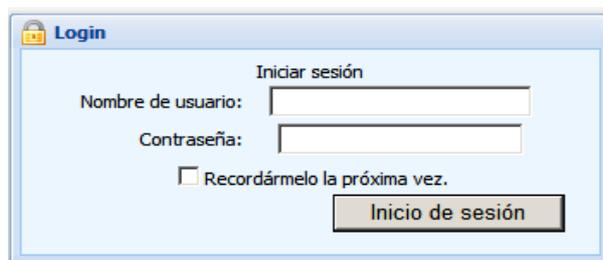
### 1. INTRODUCCIÓN

Este documento es una guía para el usuario Editor del Sistema de Evaluación de Inglés desarrollado para el Departamento de Idiomas de la Universidad Politécnica Salesiana – Campus Sur, que explica los aspectos necesarios para que este usuario pueda ingresar las preguntas para los diferentes niveles.

### 2. ACCESO AL SISTEMA

- Para ingresar al sistema es necesario un browser, se sugiere Microsoft Internet Explorer 6.0 o superior.
- Cada usuario del sistema cuenta con diferentes accesos según su perfil.
- Al ingresar a la URL respectiva, se debe ingresar el username y password respectivos.

En la siguiente figura se muestra la información que es requerida para el ingreso al sistema (username y password), una vez ingresado se presiona el botón de inicio de sesión.



The image shows a web browser window with a light blue border and title bar. The title bar contains a small lock icon and the word "Login" in blue. The main content area has a white background. At the top right of the content area, the text "Iniciar sesión" is displayed. Below this, there are two input fields: "Nombre de usuario:" followed by a text box, and "Contraseña:" followed by a text box. Underneath the password field is a checkbox with the text "Recordármelo la próxima vez." To the right of these fields is a rectangular button with a grey gradient and the text "Inicio de sesión" in black.

**Figura E.18 Inicio de Sesión**

### 3. MENU DEL SISTEMA EDITOR

#### 3.1 QUESTIONS GROUP

##### 3.1.1 QUESTION GROUP

**Description**

Question groups (insert/delete/update).  
For this purpose you should select "LevelID", "UnitID", "TopicID" first. This section could be used by putting the principal instructions for different questions.  
You can insert different types of questions in this section, other cells are only for information purposes.

**Groups of Questions**

	Level Name	Faculty Name	Unit Name	Topic Name	Question Group Description	Presential	Intensive	Distance	Active
1	Level One	1 All	1 chapter 1	1 Grammar	Verb "To be"	true	true	true	true
1	Level One	1 All	1 chapter 1	1 Grammar	2 Verb "To Be"	true	true	true	true
1	Level One	1 All	1 chapter 1	2 Listening	Say yes or not	true	true	true	true
1	Level One	1 All	1 chapter 1	2 Listening	Choose true or false	true	true	true	true
1	Level One	1 All	1 chapter 1	3 Vocabulary	vocabulary of chapter 1	true	true	true	true
1	Level One	1 All	1 chapter 1	3 Vocabulary	Vocabulary	true	true	true	true
1	Level One	1 All	1 chapter 1	4 Reading	Mary has a lot of hobbies and interests. She usually gets up early...	true	true	true	true
1	Level One	1 All	2 chapter 2	1 Grammar	Simple present tense	true	true	true	true
1	Level One	1 All	2 chapter 2	2 Listening	Hear the conversation and choose your answer	true	true	true	true
1	Level One	1 All	2 chapter 2	3 Vocabulary	vocabulary of chapter 2	true	true	true	true

Page 1 of 5

Save Delete selected records Insert Refresh

**Figura E.19 Ingreso de Encabezado de preguntas.**

##### 3.1.1.1 INSERT

3.1.1.1.1 Clic en Insert

3.1.1.1.2 Escoger los parámetros Level Name, Faculty Name, Unit Name, Topic Name correspondientes.

3.1.1.1.3 Colocar en Question Group Description el grupo al que pertenece la pregunta o en el caso de una pregunta de reading el texto relacionado a la misma.

3.1.1.1.4 En los ítems Presential, Intensive, Distance y Active escoger true or false para activar la pregunta en las mismas.

3.1.1.1.5 Clic en Save para guardar los cambios

##### 3.1.1.2 DELETE SELECT RECORDS

3.1.1.2.1 Seccionar un ítem

3.1.1.2.2 Click en Delete selected records

3.1.1.2.3 Clic en Save para guardar los cambios

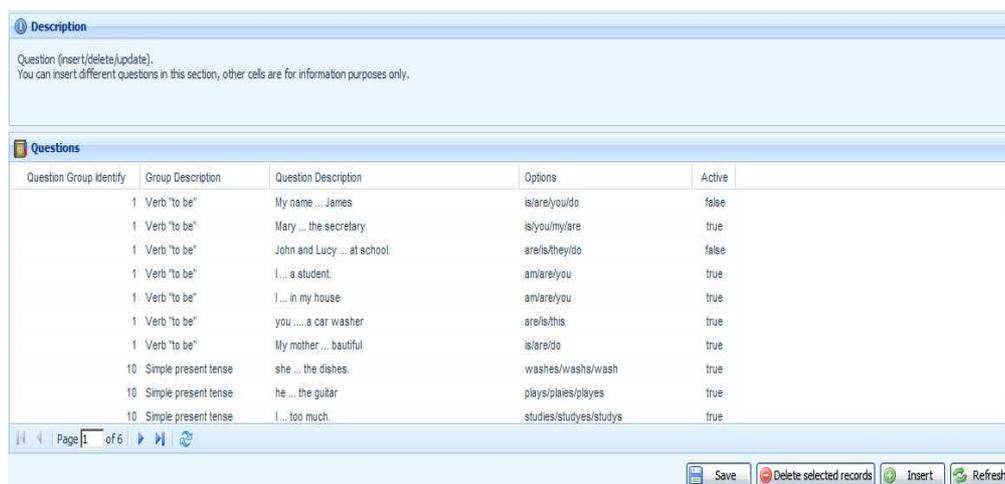
### 3.1.1.3 REFRESH

Sirve para actualizar la página

### 3.1.1.4 BOTONES DE NAVEGACIÓN

Sirve para desplazarse entre ítems.

## 3.1.2 MANAGEMENT QUESTIONS



Question Group Identity	Group Description	Question Description	Options	Active
1	Verb 'to be'	My name ... James	is/are/you/do	false
1	Verb 'to be'	Mary ... the secretary.	is/you/my/are	true
1	Verb 'to be'	John and Lucy ... at school.	are/is/they/do	false
1	Verb 'to be'	I ... a student.	am/are/you	true
1	Verb 'to be'	I ... in my house	am/are/you	true
1	Verb 'to be'	you ... a car washer	are/is/this	true
1	Verb 'to be'	My mother ... beautiful	is/are/do	true
10	Simple present tense	she ... the dishes.	washes/washes/wash	true
10	Simple present tense	he ... the guitar	plays/plays/plays	true
10	Simple present tense	I ... too much.	studies/studies/studys	true

**Figura E.20 Ingreso de preguntas**

### 3.1.2.1 INSERT

3.1.2.1.1 Clic en Insert

3.1.2.1.2 Escoger los parámetros Question Group Identity, Group Description, correspondientes.

3.1.2.1.3 Colocar en Question Description la pregunta correspondiente al grupo

3.1.2.1.4 En el ítem Option colocar las opciones correspondientes a esa pregunta

- 3.1.2.1.5 En el Ítem Active, escoger true or false para que la pregunta se active o no.
- 3.1.2.1.6 Clic en Save para guardar los cambios

### 3.1.2.2 DELETE SELECT RECORDS

- 3.1.2.2.1 Seccionar un ítem
- 3.1.2.2.2 Click en Delete selected records
- 3.1.2.2.3 Clic en Save para guardar los cambios

### 3.1.2.3 REFRESH

Sirve para actualizar la página

### 3.1.2.4 BOTONES DE NAVEGACIÓN

Sirve para desplazarse entre ítems.

## 3.1.3 LISTENING AUDIO

- 3.1.3.1 Añadir el Question Group para la pregunta
- 3.1.3.2 Añadir la pregunta en el Management question
- 3.1.3.3 Relacionar la pregunta de los ítems anteriores con el audio correspondiente, a través del botón examinar.
- 3.1.3.4 Clic en el botón Add Audio.

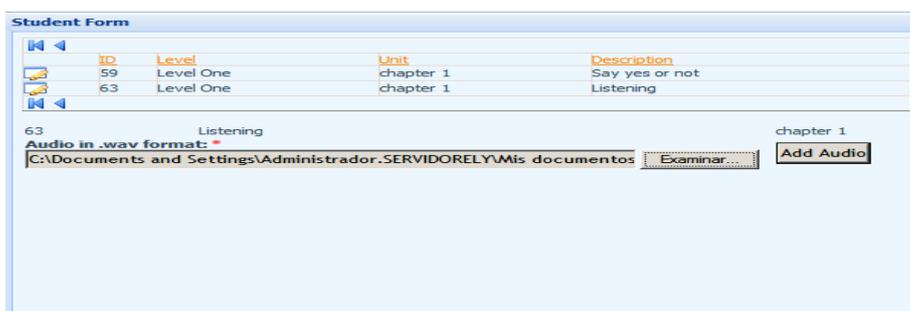


Figura E.21 Relacionar audio con una pregunta de Listening

## ANEXO F

### CÓDIGO DEL SISTEMA

En este anexo se presenta parte del código del sistema desarrollado, que se ha detallado en la sección de construcción del prototipo. Muestra tanto código aspx como código c#.

#### Web.sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="Default.aspx" title="Principal">
    <siteMapNode url="All.aspx" title="Principal">
      <siteMapNode url="Login.aspx" title="Login" />
      <siteMapNode url="~/Authentication/Public/PasswordRecovery.aspx"
        title="Recovery Password" />
      <siteMapNode url="~/Authentication/Private/ChangePassword.aspx"
        title="Change Password"
        roles="Administrator,Teacher,Editor,PresentialMode,IntensiveMode,Di
        stanceMode"/>
    </siteMapNode>
    <siteMapNode url="~/View/Administrator/Administrator.aspx"
      title="Administrator" roles="Administrator">
      <siteMapNode url="~/View/Administrator/AdministratorAdd.aspx"
        title="Add Administrator" roles="Administrator" />
      <siteMapNode url="~/View/Administrator/UserAdd.aspx" title="Add
        Teacher" roles="Administrator" />
      <siteMapNode url="~/View/Administrator/StudentAdd.aspx" title="Add
        Student" roles="Administrator" />
      <siteMapNode url="~/View/Administrator/Faculties.aspx"
        title="Manage Faculties" roles="Administrator" />
      <siteMapNode url="~/View/Administrator/ManagementLevels.aspx"
        title="Manage Levels" roles="Administrator" />
      <siteMapNode url="~/View/Administrator/ManagementClassrooms.aspx"
        title="Manage Classrooms" roles="Administrator" />
      <siteMapNode url="~/View/Administrator/ManagementTopics.aspx"
        title="Manage Topics " roles="Administrator" />
      <siteMapNode url="~/View/Administrator/ManagementUnits.aspx"
        title="Manage Units " roles="Administrator" />
      <siteMapNode
        url="~/View/Administrator/ManagementParametersQuestions.aspx"
        title="Parameters Questions" roles="Administrator" />
    </siteMapNode>
    <siteMapNode url="~/View/Teacher/Teacher.aspx" title="Teacher"
      roles="Teacher">
      <siteMapNode url="~/View/Teacher/GenerateTests.aspx"
        title="Generate Test" roles="Teacher" />
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

```

    <siteMapNode url="~/View/Editors/Editors.aspx" title="Editor"
    roles="Editor">
    <siteMapNode url="~/View/Editors/ManagementQuestionGroups.aspx"
    title="Question Groups" roles="Editor" />
    <siteMapNode url="~/View/Editors/ManagementQuestions.aspx"
    title="Questions" roles="Editor" />
  </siteMapNode>
  <siteMapNode url="~/View/Students/PresentialMode.aspx"
  title="Presential Mode"
  roles="PresentialMode, IntensiveMode, DistanceMode">
  <siteMapNode url="~/View/Students/TakeTests.aspx" title="Take
  Test" roles="PresentialMode" />
</siteMapNode>
<siteMapNode url="~/View/Students/IntensiveMode.aspx"
  title="Intensive Mode" roles="IntensiveMode">
  <siteMapNode url="~/View/Students/1.aspx" title="1"
  roles="IntensiveMode" />
</siteMapNode>
<siteMapNode url="~/View/Students/DistanceMode.aspx"
  title="Distance Mode" roles="DistanceMode">
  <siteMapNode url="~/View/Students/2.aspx" title="2"
  roles="DistanceMode" />
</siteMapNode>
</siteMapNode>
</siteMap>

```

## Login

```

<%@ Page Language="C#" AutoEventWireup="true" Title="Log In"
CodeFile="Login.aspx.cs" Inherits="Login" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Login Form</title>
</head>
<body>
  <form id="form1" runat="server">
    <ext:ScriptManager ID="ScriptManager1" runat="server" />
    <ext:Window ID="Window1" runat="server" Closable="false"
      Resizable="false" Height="160" Icon="Lock"
      Title="Login" Draggable="false" Width="350"
      Modal="true" BodyStyle="padding:5px;" >
      <Body>
        <center>
          <asp:Login ID="LoginOnlineTesting" runat="server"
            Height="100px" Width="300px"
            onloggedin="LoginOnlineTesting_LoggedIn" >
            </asp:Login>
          </center>
        </Body>
      </ext:Window>
    </form>
  </body>
</html>

```

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Web;

```

```

using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using dsUserInRolesTextTableAdapters;

public partial class Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string url = Request.Url.ToString();
        string[] urlElements = url.Split('?');
        try
        {
            if (!string.IsNullOrEmpty(urlElements[1]))
                Response.Redirect("Login.aspx");
        }
        catch
        {
        }
    }

    protected void LoginOnlineTesting_LoggedIn(object sender, EventArgs e)
    {
        ArrayList roles = new ArrayList();
        string username =
            ((TextBox)LoginOnlineTesting.FindControl("UserName")).Text;
        vw_UserInRolesTableAdapter vw_UserInRolesAdapter = new
            vw_UserInRolesTableAdapter();
        dsUserInRolesText.vw_UserInRolesDataTable userInRoles;
        userInRoles = vw_UserInRolesAdapter.GetUserRolByUserName(username);
        foreach (dsUserInRolesText.vw_UserInRolesRow userInRolesRow in
            userInRoles)
        {
            roles.Add(userInRolesRow.RoleName.ToString());
        }
        LoginOnlineTesting.DestinationPageUrl = "Login.aspx";
        if (roles.Contains("Administrator"))
            LoginOnlineTesting.DestinationPageUrl =
                "~/View/Administrator/Default.aspx";
        else if (roles.Contains("Teacher"))
            LoginOnlineTesting.DestinationPageUrl =
                "~/View/Teacher/Default.aspx";
        else if (roles.Contains("Editor"))
            LoginOnlineTesting.DestinationPageUrl =
                "~/View/Editors/Default.aspx";
        else if (roles.Contains("PresentialMode") ||
            roles.Contains("IntensiveMode") ||
            roles.Contains("DistanceMode"))
            LoginOnlineTesting.DestinationPageUrl =
                "~/View/Students/Default.aspx";
    }
}

```

### AdministratorAdd

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="AdministratorAdd.aspx.cs" Inherits="View_Administrator_UserAdd"
%>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="PageHeader" runat="server">
  <title></title>
  <style type="text/css">
    .x-form-group .x-form-group-header-text {
      background-color: #dfe8f6;
    }
    .x-label-text {
      font-weight: bold;
      font-size: 11px;
    }
  </style>
</head>
<body class="page">
  <form id="form1" runat="server">
    <ext:ScriptManager ID="ScriptManagerCoolite" runat="server"
RemoveViewState="False" />
    <asp:ScriptManager ID="ScriptManagerAjax"
runat="server"></asp:ScriptManager>
    <!-- Formulario -->
    <asp:Panel ID="pnlPageHeader" CssClass="page-header" runat="server">
      <!-- Título -->
      <asp:Panel ID="pnlTitle" CssClass="page-header" runat="server">
        <h1><asp:Label ID="lblTitle" runat="server"></asp:Label></h1>
      </asp:Panel>
    </asp:Panel>
    <asp:Panel ID="pnlFormContainer" CssClass="form-container"
runat="server">
      <!-- Panel de Favoritos -->
      <asp:Panel ID="pnlFormSection" CssClass="form-section"
runat="server">
        <asp:Panel ID="pnlContainer" CssClass="container"
runat="server">
          <!-- escritorio de trabajo -->
          <asp:CreateUserWizard ID="CreateUserWizardUser"
runat="server"
ContinueDestinationPageUrl="~/View/Administrator/AdministratorAdd.aspx"
DisableCreatedUser="false" CompleteSuccessText="The
account has been successfully created."
UnknownErrorMessage="The account was not created. Please try again."
CreateUserButtonText="Create Administrator Now"
CreateUserButtonType="Image"
CreateUserButtonImageUrl="~/Images/en-US/white.png"
OnCreatedUser="CreateUserWizardUser_CreatedUser"
          >
          <WizardSteps>
          <asp:CreateUserWizardStep ID="CreateUserWizardStep1" runat="server"
Title="Step 1: Basic User Information"
OnActivate="AssignUserToRoles_Activate"
OnDeactivate="AssignUserToRoles_Deactivate">
          <ContentTemplate>
          <ext:Window
ID="FormWindow" runat="server" Width="600" Height="405"
Title="Form" ShowOnLoad="true" CenterOnLoad="true" Resizable="false"
Closable="false" BodyStyle="padding:6px;">
          <Body>
          <ext:ContainerLayout ID="ContainerLayout1" runat="server">

```

```

<ext:Panel ID="pnlAdministratorDetails" runat="server" Title="Details"
AutoHeight="true" FormGroup="true">
<body>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text" >User Name:</span><span
style="color:red">*</span></p>
<asp:TextBox ID="UserName" runat="server" Width="275" MaxLength="10" />
<asp:RequiredFieldValidator ID="UserNameRequired" runat="server"
ControlToValidate="UserName" ErrorMessage="User Name is required."
ToolTip="User Name is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<%--regula que solo acepte 10 numeros--%>
<asp:RegularExpressionValidator ID="UserNameRegularExpressionValidator1"
runat="server"
ControlToValidate="UserName" ErrorMessage="Debe ingresar 10 d&#237;gitos"
ValidationExpression="[0-9]{10}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Password:</span><span
style="color:red">*</span></p>
<ext:TextField ID="Password" runat="server" Width="275"
AllowBlank="false" InputType="Password"/>
<asp:RequiredFieldValidator ID="PasswordRequired" runat="server"
ControlToValidate="Password" ErrorMessage="Password is required."
ToolTip="Password is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
</tr>
<tr>
<td colspan="2">
<p><span class="x-label-text">Email:</span><span
style="color:red">*</span></p>
<ext:TextField ID="Email" runat="server" Width="275" AllowBlank="false"
/>
<asp:RequiredFieldValidator ID="EmailRequired" runat="server"
ControlToValidate="Email" ErrorMessage="E-mail is required."
ToolTip="E-mail is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<%--regula que solo acepte direcciones de correo validas--%>
<asp:RegularExpressionValidator ID="EmailRegularExpressionValidator"
runat="server" ControlToValidate="Email" ErrorMessage="Ingresar una
dirección de correo electrónico valido" SetFocusOnError="True"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.] \w+)*"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Re-Type Password:</span><span
style="color:red">*</span></p>
<ext:TextField ID="ConfirmPassword" runat="server" Width="275"
AllowBlank="false" InputType="Password"/>
<asp:RequiredFieldValidator ID="ConfirmPasswordRequired" runat="server"
ControlToValidate="ConfirmPassword"
ErrorMessage="Confirm Password is required."
ToolTip="Confirm Password is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
</tr>

```

```

</table>
</body>
</ext:Panel>
<ext:Panel ID="pnlPersonalInformation" runat="server" Title="Personal
Information" AutoHeight="true" FormGroup="true" Collapsed="false" >
<body>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text">Names:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbNames" runat="server" Width="275"
AllowBlank="false" MaxLength="50" />
<asp:RequiredFieldValidator ID="txbNamesRequired" runat="server"
ControlToValidate="txbNames" ErrorMessage="Names is required."
ToolTip="Names is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<!--regula que solo acepte letras-->
<asp:RegularExpressionValidator ID="NamesRegularExpressionValidator"
runat="server"
ControlToValidate="txbNames" ErrorMessage="Debe ingresar &#250;nicamente
letras"
ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"*
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Last Names:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbLastNames" runat="server" Width="275"
AllowBlank="false" MaxLength="50" />
<asp:RequiredFieldValidator ID="txbLastNamesRequired" runat="server"
ControlToValidate="txbLastNames" ErrorMessage="Last Names is required."
ToolTip="Last Names is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<!--regula que solo acepte letras-->
<asp:RegularExpressionValidator ID="LastNamesRegularExpressionValidator"
runat="server"
ControlToValidate="txbLastNames" ErrorMessage="Debe ingresar
&#250;nicamente letras"
ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"*
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
<tr>
<td colspan="4">
<p><span class="x-label-text">Photo 92x70:</span><span
style="color:red">*</span></p>
<asp:FileUpload ID="ucPhoto" runat="server" width="560" />
<asp:RequiredFieldValidator ID="ucPhotoRequired" runat="server"
ControlToValidate="ucPhoto" ErrorMessage="Photo is required."
ToolTip="Photo is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<!--regula que solo acepte formato jpg/jpeg-->
<asp:RegularExpressionValidator ID="PhotoRegularExpressionValidator"
runat="server"
ControlToValidate="ucPhoto" ErrorMessage="Image must be a jpg/jpeg
format."
ValidationExpression="(.*\.( [jJ][pP][gG] | [jJ][pP][eE][gG] )$)"

```

```

ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
<tr>
<td >
<p><span class="x-label-text">Identity Card:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbIdentityCard" runat="server" Width="132"
AllowBlank="false" MaxLength="10" />
<asp:RequiredFieldValidator ID="txbIdentityCardRequired" runat="server"
ControlToValidate="txbIdentityCard" ErrorMessage="Identity Card is
required."
ToolTip="Identity Card is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<!--regula que solo acepte 10 numeros-->
<asp:RegularExpressionValidator
ID="IdentityCardRegularExpressionValidator" runat="server"
ControlToValidate="txbIdentityCard" ErrorMessage="Debe ingresar 10
d&#237;gitos"
ValidationExpression="[0-9]{10}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td >
<p><span class="x-label-text">Blood Type:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbBloodType" runat="server" Width="130"
AllowBlank="false" MaxLength="5" />
<asp:RequiredFieldValidator ID="txbBloodTypeRequired" runat="server"
ControlToValidate="txbBloodType" ErrorMessage="Blood Type is required."
ToolTip="Blood Type is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
<td >
<p><span class="x-label-text">Gender:</span><span
style="color:red">*</span></p>
<asp:DropDownList ID="ucGender" runat="server" Width="132" >
<asp:ListItem Text="[Select One]" />
<asp:ListItem Text="male" />
<asp:ListItem Text="female" />
</asp:DropDownList>
<asp:RequiredFieldValidator ID="ucGenderRequired" runat="server"
ControlToValidate="ucGender" ErrorMessage="Gender is required."
ToolTip="Gender is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
<td >
<p><span class="x-label-text">Birthday: (m/d/a)</span><span
style="color:red">*</span></p>
<ext:DateField ID="ucBirthday" runat="server" Width="132"
AllowBlank="false" />
<asp:RequiredFieldValidator ID="ucBirthdayRequired" runat="server"
ControlToValidate="ucBirthday" ErrorMessage="Birthday is required."
ToolTip="Birthday is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
</tr>
</table>
</body>
</ext:Panel>

```

```

<ext:Panel ID="pnlSecurityQuestion" runat="server" Title="Control
Information" AutoHeight="true" FormGroup="true" Collapsed="true" >
<body>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text">Security Question:</span><span
style="color:red">*</span></p>
<asp:DropDownList ID="Question" runat="server" Width="275">
<asp:ListItem Text="Mother's Main Name" Selected="True"/>
<asp:ListItem Text="Favorite Pet" />
<asp:ListItem Text="Who was your childhood hero?" />
<asp:ListItem Text="Your favorite hobby?" />
</asp:DropDownList>

<asp:RequiredFieldValidator ID="QuestionRequired" runat="server"
ControlToValidate="Question" ErrorMessage="Question is required."
ToolTip="Question is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Answer:</span><span
style="color:red">*</span></p>
<ext:TextField ID="Answer" runat="server" Width="275" AllowBlank="false"
/>
<asp:RequiredFieldValidator ID="AnswerRequired" runat="server"
ControlToValidate="Answer" ErrorMessage="Answer is required."
ToolTip="Answer is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
</tr>
<tr>
<td colspan="2">
<p><span class="x-label-text">Father Name:</span></p>
<ext:TextField ID="txbFatherName" runat="server" Width="275"
AllowBlank="true" MaxLength="50"/>
<!--regula que solo acepte letras-->
<asp:RegularExpressionValidator ID="FatherNameRegularExpressionValidator"
runat="server"
ControlToValidate="txbFatherName" ErrorMessage=" Debe ingresar
&#250;nicamente letras"
ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"*
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Mother Name:</span></p>
<ext:TextField ID="txbMotherName" runat="server" Width="275"
AllowBlank="true" MaxLength="50" />
<!--regula que solo acepte letras-->
<asp:RegularExpressionValidator ID="MotherNameRegularExpressionValidator"
runat="server"
ControlToValidate="txbMotherName" ErrorMessage="Debe ingresar
&#250;nicamente letras"
ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"*
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
</table>

```

```

</body>
</ext:Panel>
<ext:Panel ID="pnlAdditionalInformation" runat="server" Title="Additional
Information" AutoHeight="true" FormGroup="true" Collapsed="true" >
<body>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text">Principal Street:</span></p>
<ext:TextField ID="txbPrincipalStreet" runat="server" Width="275"
AllowBlank="true" />
</td>
<td colspan="2">
<p><span class="x-label-text">Transversal Street:</span></p>
<ext:TextField ID="txbTransversalStreet" runat="server" Width="275"
AllowBlank="true" />
</td>
</tr>
<tr>
<td >
<p><span class="x-label-text">Street Number:</span></p>
<ext:TextField ID="txbNumber" runat="server" Width="132"
AllowBlank="true" />
</td>
<td >
<p><span class="x-label-text">Country:</span></p>
<ext:TextField ID="ucCountry" runat="server" Width="132"
AllowBlank="true" Text="Ecuador" />
</td>
<td >
<p><span class="x-label-text">City:</span></p>
<ext:TextField ID="txbCity" runat="server" Width="132" AllowBlank="true"
Text="Quito" />
</td>
<td >
<p><span class="x-label-text">Marital Status:</span></p>
<asp:DropDownList ID="ucMaritalStatus" runat="server" Width="132">
<asp:ListItem Text="[Select one]" />
<asp:ListItem Text="single" />
<asp:ListItem Text="married" />
<asp:ListItem Text="divorced" />
<asp:ListItem Text="separated" />
<asp:ListItem Text="cohabitation" />
</asp:DropDownList>
</td>
</tr>
</table>
<table>
<tr>
<td >
<p><span class="x-label-text">Phone:</span></p>
<ext:TextField ID="txbPhone" runat="server" Width="132" AllowBlank="true"
MaxLength="7" />
<!--regula que solo acepte 7 numeros-->
<asp:RegularExpressionValidator ID="PhoneRegularExpressionValidator"
runat="server"
ControlToValidate="txbPhone" ErrorMessage="Debe ingresar 7 d&#237;gitos"
ValidationExpression="[0-9]{7}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>

```

```

<td >
<p><span class="x-label-text">Cellphone:</span></p>
<ext:TextField ID="txbCellphone" runat="server" Width="132"
AllowBlank="true" MaxLength="9" />
<!--regula que solo acepte 9 numeros-->
<asp:RegularExpressionValidator ID="CellphoneRegularExpressionValidator"
runat="server"
ControlToValidate="txbCellphone" ErrorMessage="Debe ingresar 9
d&#237;gitos"
ValidationExpression="[0-9]{9}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
<tr>
<td align="center" colspan="4">
<asp:CompareValidator ID="PasswordCompare" runat="server"
ErrorMessage="The Password and Confirmation Password must match."
ValidationGroup="CreateUserWizard1" ControlToValidate="ConfirmPassword"
ControlToCompare="Password"
Display="Dynamic">
</asp:CompareValidator>
</td>
</tr>
<tr>
<td style="color: red" align="center" colspan="4">
<asp:Literal ID="ErrorMessage" runat="server"
EnableViewState="False"></asp:Literal>
</td>
</tr>
</table>
</body>
</ext:Panel>
</ext:ContainerLayout>
</Body>
<TopBar>
<ext:ToolBar runat="server">
<Items>
<ext:ToolBarFill ID="ToolBarFill1" runat="server" />
<ext:Button ID="Createuser" Icon="UserAdd" AutoPostBack="true"
runat="server" CommandName="MoveNext" Text="Create User"
ValidationGroup="CreateUserWizard1" />
</Items>
</ext:ToolBar>
</TopBar>
</ext:Window>
</ContentTemplate>
</asp:CreateUserWizardStep>
<asp:CompleteWizardStep ID="CompleteWizardStep1" runat="server">
<ContentTemplate>
<ext:Window
ID="Window3" runat="server" Width="250" Height="140" Title="User"
ShowOnLoad="true" CenterOnLoad="true" Resizable="false" Closable="false"
BodyStyle="padding:6px;">
<Body>
<ext:ContainerLayout ID="ContainerLayout3" runat="server">
<ext:Panel ID="Panel6" runat="server" Title="Administrator Added"
AutoHeight="true" FormGroup="true" BodyStyle="padding:10px;" >
<Body>
<table border="0">

```

```

<tr>
<td align="center" colspan="2">Complete</td>
</tr>
<tr>
<td>User has been successfully created.</td>
</tr>
<tr>
<td align="right" colspan="2">
<asp:LinkButton ID="ContinueButton" runat="server"
CausesValidation="False"
CommandName="Continue" Text="Continue"
ValidationGroup="CreateUserWizard1" />
</td>
</tr>
</table>
</Body>
</ext:Panel>
</ext:ContainerLayout>
</Body>
</ext:Window>
</ContentTemplate>
</asp:CompleteWizardStep>
</WizardSteps>
<TitleTextStyle Font-Bold="True" BackColor="#6B696B"
ForeColor="White"></TitleTextStyle>
</asp:CreateUserWizard>
</asp:Panel>
</asp:Panel>
</asp:Panel>
</form>
</body>
</html>

```

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using dsAdministratorProfilesTableAdapters;
using System.Collections.Generic;
using Coolite.Ext.Web;
using System.IO;

public partial class View_Administrator_UserAdd : System.Web.UI.Page
{
    #region InitializeCulture - Page_PreInit
    protected override void InitializeCulture()
    {
        PageManager.InitializeCulture(this);
    }
    protected void Page_PreInit(object sender, EventArgs e)
    {
        PageManager.ChangeTheme(this);
    }
    #endregion
}

```

```

#region Page_Load
protected void Page_Load(object sender, EventArgs e)
{
    loadResources();
}
#endregion
#region loadResources
protected void loadResources()
{
    lblTitle.Text = his.GetGlobalResourceObject("GlobalResources",
        "USER_ADD").ToString();
}
#endregion
#region CreateUserWizardUser_CreatedUser
// CreatedUser event is called when a new user is successfully
// created
protected void CreateUserWizardUser_CreatedUser(object sender,
EventArgs e)
{
    // Create an empty Profile for the newly created user
    AdministratorProfilesTableAdapter administratorProfilesAdapter =
new AdministratorProfilesTableAdapter();
    // Populate some Profile properties off of the create user wizard
    string UserName = CreateUserWizardUser.UserName.ToString();
    string Names =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbNames")).Text;
    string LastNames =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbLastNames")).Text;
    string Photo = CreateUserWizardUser.UserName.ToString() + ".jpg";
    string IdentityCard =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbIdentityCard")).Text;
    string BloodType =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbBloodType")).Text;
    string Gender =
((DropDownList)CreateUserWizardUser.CreateUserStep.ContentTemplateContain
er.FindControl("ucGender")).SelectedValue;
    DateTime Birthday =
Convert.ToDateTime(((DateField)CreateUserWizardUser.CreateUserStep.Conten
tTemplateContainer.FindControl("ucBirthday")).SelectedDate);
    string FatherName =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbFatherName")).Text;
    string MotherName =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbMotherName")).Text;
    string TransversalStreet =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbTransversalStreet")).Text;
    string PrincipalStreet =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbPrincipalStreet")).Text;
    string Number =
((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
FindControl("txbNumber")).Text;
}
}

```

```

        string Country =
        ((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
        FindControl("ucCountry")).Text;
        string City =
        ((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
        FindControl("txbCity")).Text;
        string Phone =
        ((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
        FindControl("txbPhone")).Text;
        string Cellphone =
        ((TextField)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer.
        FindControl("txbCellphone")).Text;
        string MaritalStatus =
        ((DropDownList)CreateUserWizardUser.CreateUserStep.ContentTemplateContain
        er.FindControl("ucMaritalStatus")).SelectedValue;
        if (ucPhoto.HasFile)
        {
            string directorio = Session["ssPhoto"].ToString();
            if (Directory.Exists(directorio))
            {
                string archivo = directorio + ucPhoto.FileName;
                if (File.Exists(archivo))
                {
                    // ya existe un archivo con el mismo nombre en el directorio,
                    así que hay hacer algo al respecto (p.ej. renombrar el que
                    // está en el servidor o asignarle otro nombre al que se está
                    subiendo), de lo contrario el archivo en el servidor será
                    sobrescrito
                }
                else
                {
                    Photo =
                    ((FileUpload)CreateUserWizardUser.CreateUserStep.ContentTemplateContainer
                    .FindControl("ucPhoto")).FileName;
                    ucPhoto.SaveAs(archivo);
                    Photo = Session["ssPhotoDB"].ToString() + Photo;
                }
            }
            else
            {
                throw new DirectoryNotFoundException("directory don't
                exist");
            }
        }
        administratorProfilesAdapter.Insert(UserName, Names, LastNames,
        Photo, IdentityCard, BloodType, Gender, Birthday, FatherName, MotherName,
        TransversalStreet, PrincipalStreet, Number, Country, City, Phone,
        Cellphone, MaritalStatus);
    }
    #endregion
    // Activate event fires when the user hits "next" in the
    CreateUserWizard
    public void AssignUserToRoles_Activate(object sender, EventArgs e)
    {
        // Databind list of roles in the role manager system to a listbox
        in the wizard
        //AvailableRoles.DataSource = Roles.GetAllRoles(); ;
        //AvailableRoles.DataBind();
    }
}

```

```

    // Deactivate event fires when user hits "next" in the
    CreateUserWizard
    public void AssignUserToRoles_Deactivate(object sender, EventArgs e)
    {
        Roles.AddUserToRole(CreateUserWizardUser.UserName,
"Administrator");
    }
}

```

## UserControl

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UserControl.aspx.cs" Inherits="View_Administrator_UserAdd" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="PageHeader" runat="server">
    <title></title>
    <style type="text/css">
        .x-form-group .x-form-group-header-text {
            background-color: #dfe8f6;
        }
        .x-label-text {
            font-weight: bold;
            font-size: 11px;
        }
    </style>
</head>
<body class="page">
    <form id="form1" runat="server">
        <ext:ScriptManager ID="ScriptManagerCoolite" runat="server"
RemoveViewState="False" />
        <asp:ScriptManager ID="ScriptManagerAjax"
runat="server"></asp:ScriptManager>
        <%-- Formulario --%>
        <asp:Panel ID="pnlPageHeader" CssClass="page-header" runat="server">
            <%--Título--%>
            <asp:Panel ID="pnlTitle" CssClass="page-header" runat="server">
                <h1><asp:Label ID="lblTitle" runat="server"></asp:Label></h1>
            </asp:Panel>
        </asp:Panel>
        <asp:Panel ID="pnlFormContainer" CssClass="form-container"
runat="server">
            <%--Panel de Favoritos--%>
            <asp:Panel ID="pnlFormSection" CssClass="form-section"
runat="server">
                <asp:Panel ID="pnlContainer" CssClass="container"
runat="server">
                    <%-- escritorio de trabajo--%>
                    <ext:Panel ID="pnlParameters" runat="server"
Title="Parameters of User" CenterOnLoad="false"
Width="205px" Height="270px"
BodyStyle="padding:10px;" Resizable="false"
Closable="false">
                        <Body>
                            <asp:UpdatePanel ID="UpdatePanel2" runat="server"
UpdateMode="Conditional">
                                <ContentTemplate>
                                    <table>
                                        <tr>
                                        <td>

```

```

<p><span class="x-label-text">Name:</span><span
style="color:red">*</span></p>
<asp:DropDownList ID="ddlManageUsers" runat="server" Width="175px"
DataTextField="LastNames" DataValueField="UserName"
AutoPostBack="True" DataSourceID="odsUsers"
onselectedindexchanged="ddlManageUsers_SelectedIndexChanged">
</asp:DropDownList>
<asp:ObjectDataSource ID="odsUsers" runat="server"
OldValuesParameterFormatString="original_{0}"
SelectMethod="GetUserProfiles"
TypeName="dsUserProfilesTableAdapters.UserProfilesTableAdapter">
</asp:ObjectDataSource>
</td>
</tr>
<tr>
<td>
<p><span class="x-label-text">Complete Student Name:</span></p>
<asp:TextBox ID="txtCompleteName" runat="server" TextMode="MultiLine"
Width="170px" Rows="7" ></asp:TextBox>
</td>
</tr>
<tr>
<td>
<asp:Button ID="btnDelete" runat="server" Text="Delete User"
onclick="btnDelete_Click" Enabled="false"></asp:Button>
</td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</Body>
</ext:Panel>
<ext:Window
ID="FormWindow" runat="server" Width="600" Height="405" Title="Form"
ShowOnLoad="true" CenterOnLoad="true" Resizable="false" Closable="false"
BodyStyle="padding:6px;">
<Body>
<ext:ContainerLayout ID="ContainerLayout1" runat="server">
<ext:Panel ID="pnlStudentDetails" runat="server" Title="Details"
AutoHeight="true" FormGroup="true">
<body>
<asp:UpdatePanel ID="UpdatePanelUserName" runat="server" >
<ContentTemplate>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text" >User Name:</span><span
style="color:red">*</span></p>
<asp:TextBox ID="UserName" runat="server" Width="271" MaxLength="10"
Enabled="false" />
<asp:RequiredFieldValidator ID="UserNameRequired" runat="server"
ControlToValidate="UserName" ErrorMessage="User Name is required."
ToolTip="User Name is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<%--regula que solo acepte 10 numeros--%>
<asp:RegularExpressionValidator ID="UserNameRegularExpressionValidator1"
runat="server"
ControlToValidate="UserName" ErrorMessage="Debe ingresar 10 d&#237;gitos"
ValidationExpression="[0-9]{10}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>

```

```

</td>
<td>
<asp:Button ID="btnUpdate" runat="server" Text=" Update Data "
onclick="CreateUserWizardUser_CreatedUser" Enabled="false"></asp:Button>
</td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</body>
</ext:Panel>
<ext:Panel ID="pnlPersonalInformation" runat="server" Title="Personal
Information" AutoHeight="true" FormGroup="true" Collapsed="false" >
<body>
<asp:UpdatePanel ID="UpdatePanelPersonalInformation" runat="server" >
<ContentTemplate>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text">Names:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbNames" runat="server" Width="275"
AllowBlank="false" MaxLength="50"/>
<asp:RequiredFieldValidator ID="txbNamesRequired" runat="server"
ControlToValidate="txbNames" ErrorMessage="Names is required."
ToolTip="Names is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<!--regula que solo acepte letras-->
<asp:RegularExpressionValidator ID="NamesRegularExpressionValidator"
runat="server" ControlToValidate="txbNames" ErrorMessage="Debe ingresar
&#250;nicamente letras" ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"*
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Last Names:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbLastNames" runat="server" Width="275"
AllowBlank="false" MaxLength="50" />
<asp:RequiredFieldValidator ID="txbLastNamesRequired" runat="server"
ControlToValidate="txbLastNames" ErrorMessage="Last Names is required."
ToolTip="Last Names is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<!--regula que solo acepte letras-->
<asp:RegularExpressionValidator ID="LastNamesRegularExpressionValidator"
runat="server"
ControlToValidate="txbLastNames" ErrorMessage="Debe ingresar
&#250;nicamente letras"
ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"*
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
<tr>
<td colspan="4">
<p><span class="x-label-text">Photo 92x70:</span><span
style="color:red">*</span></p>
<asp:FileUpload ID="ucPhoto" runat="server" width="560" />
<!--regula que solo acepte formato jpg/jpeg-->

```

```

<asp:RegularExpressionValidator ID="PhotoRegularExpressionValidator"
runat="server" ControlToValidate="ucPhoto" ErrorMessage="Image must be a
jpg/jpeg format."
ValidationExpression="(.*\.([jJ][pP][gG]|[jJ][pP][eE][gG])$)"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
<tr>
<td >
<p><span class="x-label-text">Identity Card:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbIdentityCard" runat="server" Width="132"
AllowBlank="false" MaxLength="10" />
<asp:RequiredFieldValidator ID="txbIdentityCardRequired" runat="server"
ControlToValidate="txbIdentityCard" ErrorMessage="Identity Card is
required." ToolTip="Identity Card is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
<%--regula que solo acepte 10 numeros--%>
<asp:RegularExpressionValidator
ID="IdentityCardRegularExpressionValidator" runat="server"
ControlToValidate="txbIdentityCard" ErrorMessage="Debe ingresar 10
d&#237;gitos"
ValidationExpression="[0-9]{10}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td >
<p><span class="x-label-text">Blood Type:</span><span
style="color:red">*</span></p>
<ext:TextField ID="txbBloodType" runat="server" Width="132"
AllowBlank="false" />
<asp:RequiredFieldValidator ID="txbBloodTypeRequired" runat="server"
ControlToValidate="txbBloodType" ErrorMessage="Blood Type is required."
ToolTip="Blood Type is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
<td >
<p><span class="x-label-text">Gender:</span><span
style="color:red">*</span></p>
<asp:DropDownList ID="ucGender" runat="server" Width="132" >
<asp:ListItem Value="-1">[Select One]</asp:ListItem>
<asp:ListItem Value="male">Male</asp:ListItem>
<asp:ListItem Value="female">Female</asp:ListItem>
</asp:DropDownList>
<asp:RequiredFieldValidator ID="ucGenderRequired" runat="server"
ControlToValidate="ucGender" ErrorMessage="Gender is required."
ToolTip="Gender is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
</td>
<td >
<p><span class="x-label-text">Birthday:</span></p>
<ext:DateField ID="ucBirthday" runat="server" Width="132"
AllowBlank="false" />
</td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</body>
</ext:Panel>

```

```

<ext:Panel ID="pnlSecurityQuestion" runat="server" Title="Control
Information" AutoHeight="true" FormGroup="true" Collapsed="true" >
<body>
<asp:UpdatePanel ID="UpdatePanelSecurityQuestion" runat="server" >
<ContentTemplate>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text">Father Name:</span></p>
<ext:TextField ID="txbFatherName" runat="server" Width="275"
AllowBlank="true" MaxLength="50"/>
<%--regula que solo acepte letras--%>
<asp:RegularExpressionValidator ID="FatherNameRegularExpressionValidator"
runat="server"
ControlToValidate="txbFatherName" ErrorMessage=" Debe ingresar
&#250;nicamente letras"
ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td colspan="2">
<p><span class="x-label-text">Mother Name:</span></p>
<ext:TextField ID="txbMotherName" runat="server" Width="275"
AllowBlank="true" MaxLength="50" />
<%--regula que solo acepte letras--%>
<asp:RegularExpressionValidator ID="MotherNameRegularExpressionValidator"
runat="server"
ControlToValidate="txbMotherName" ErrorMessage="Debe ingresar
&#250;nicamente letras" ValidationExpression="[a-zA-z
|&#225;|&#233;|&#237;|&#243;|&#250;|&#241;|&#209;|]"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</body>
</ext:Panel>
<ext:Panel ID="pnlAdditionalInformation" runat="server" Title="Additional
Information" AutoHeight="true" FormGroup="true" Collapsed="true" >
<body>
<asp:UpdatePanel ID="UpdatePanelAdditionalInformation" runat="server" >
<ContentTemplate>
<table>
<tr>
<td colspan="2">
<p><span class="x-label-text">Principal Street:</span></p>
<ext:TextField ID="txbPrincipalStreet" runat="server" Width="275"
AllowBlank="true" />
</td>
<td colspan="2">
<p><span class="x-label-text">Transversal Street:</span></p>
<ext:TextField ID="txbTransversalStreet" runat="server" Width="275"
AllowBlank="true" />
</td>
</tr>
<tr>
<td >
<p><span class="x-label-text">Street Number:</span></p>

```

```

<ext:TextField ID="txbNumber" runat="server" Width="132"
AllowBlank="true" />
</td>
<td >
<p><span class="x-label-text">Country:</span></p>
<ext:TextField ID="ucCountry" runat="server" Width="132"
AllowBlank="true" Text="Ecuador" />
</td>
<td >
<p><span class="x-label-text">City:</span></p>
<ext:TextField ID="txbCity" runat="server" Width="132" AllowBlank="true"
Text="Quito" />
</td>
<td >
<p><span class="x-label-text">Marital Status:</span></p>
<asp:DropDownList ID="ucMaritalStatus" runat="server" Width="132">
<asp:ListItem Text="[Select one]" />
<asp:ListItem Text="single" />
<asp:ListItem Text="married" />
<asp:ListItem Text="divorced" />
<asp:ListItem Text="separated" />
<asp:ListItem Text="cohabitation" />
</asp:DropDownList>
</td>
</tr>
</table>
<table>
<tr>
<td >
<p><span class="x-label-text">Phone:</span></p>
<ext:TextField ID="txbPhone" runat="server" Width="132" AllowBlank="true"
MaxLength="7" />
<asp:RegularExpressionValidator ID="PhoneRegularExpressionValidator"
runat="server" ControlToValidate="txbPhone" ErrorMessage="Debe ingresar 7
d&#237;gitos" ValidationExpression="[0-9]{7}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
<td >
<p><span class="x-label-text">Cellphone:</span></p>
<ext:TextField ID="txbCellphone" runat="server" Width="132"
AllowBlank="true" MaxLength="9"/>
<%--regula que solo acepte 9 numeros--%>
<asp:RegularExpressionValidator ID="CellphoneRegularExpressionValidator"
runat="server"
ControlToValidate="txbCellphone" ErrorMessage="Debe ingresar 9
d&#237;gitos" ValidationExpression="[0-9]{9}"
ValidationGroup="CreateUserWizard1"></asp:RegularExpressionValidator>
</td>
</tr>
<tr>
<td style="color: red" align="center" colspan="4">
<asp:Literal ID="ErrorMessage" runat="server"
EnableViewState="False"></asp:Literal>
</td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</body>
</ext:Panel>

```

```

</ext:ContainerLayout>
</Body>
</ext:Window>
</asp:Panel>
</asp:Panel>
</asp:Panel>
</form>
</body>
</html>

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;

using dsUserProfilesTableAdapters;
using System.Collections.Generic;
using Coolite.Ext.Web;
using System.IO;
using OnlineTesting.BLL;

public partial class View_Administrator_UserAdd : System.Web.UI.Page
{
    #region InitializeCulture - Page_PreInit
    protected override void InitializeCulture()
    {
        PageManager.InitializeCulture(this);
    }

    protected void Page_PreInit(object sender, EventArgs e)
    {
        PageManager.ChangeTheme(this);
    }
    #endregion

    #region Page_Load
    protected void Page_Load(object sender, EventArgs e)
    {
        loadResources();
    }
    #endregion

    #region loadResources
    protected void loadResources()
    {
        lblTitle.Text = this.GetGlobalResourceObject("GlobalResources",
"USER_ADD").ToString();
    }
    #endregion

    #region CreateUserWizardUser_CreatedUser
    // CreatedUser event is called when a new user is successfully
    created

```

```

protected void CreateUserWizardUser_CreatedUser(object sender,
EventArgs e)
{
    // Create an empty Profile for the newly created user
    UserProfilesTableAdapter userProfilesAdapter = new
UserProfilesTableAdapter();

    // Populate some Profile properties off of the create user wizard
    string UserName = this.UserName.Text.Trim();
    string Names = this.txbNames.Text.Trim();
    string LastNames = this.txbLastNames.Text.Trim();
    string Photo = UserName.ToString() + ".jpg";
    string IdentityCard = txbIdentityCard.Text.Trim();
    string BloodType = txbBloodType.Text.Trim();
    string Gender = ucGender.SelectedValue;
    DateTime Birthday = ucBirthday.SelectedDate;
    string FatherName = txbFatherName.Text.Trim();
    string MotherName = txbMotherName.Text.Trim();
    string TransversalStreet = txbTransversalStreet.Text.Trim();
    string PrincipalStreet = txbPrincipalStreet.Text.Trim();
    string Number = txbNumber.Text.Trim();
    string Country = ucCountry.Text.Trim();
    string City = txbCity.Text.Trim();
    string Phone = txbPhone.Text.Trim();
    string Cellphone = txbCellphone.Text.Trim();
    string MaritalStatus = ucMaritalStatus.SelectedValue;
    if (ucPhoto.HasFile)
    {
        string directorio = Session["ssPhoto"].ToString();
        if (Directory.Exists(directorio))
        {
            string archivo = directorio + ucPhoto.FileName;
            if (File.Exists(archivo))
            {
                // ya existe un archivo con el mismo nombre en el
                directorio, así que hay hacer algo al respecto (p.ej.
                renombrar el que
                // está en el servidor o asignarle otro nombre al que
                se está subiendo), de lo contrario el archivo en el
                servidor será sobrescrito
            }
            else
            {
                Photo =
((FileUpload)FindControl("ucPhoto")).FileName;
                ucPhoto.SaveAs(archivo);
                Photo = Session["ssPhotoDB"].ToString() + Photo;
            }
        }
        else
        {
            throw new DirectoryNotFoundException("directory don't
exist");
        }
    }
    userProfilesAdapter.Update(Names, LastNames, Photo, IdentityCard,
BloodType, Gender, Birthday, FatherName, MotherName, TransversalStreet,
PrincipalStreet, Number, Country, City, Phone, Cellphone, MaritalStatus,
UserName);
}

```

```

#endregion

//Opciones de eliminación y actualización
protected void ddlManageUsers_SelectedIndexChanged(object sender,
EventArgs e)
{
    string userName = ddlManageUsers.SelectedValue;
    UserProfilesTableAdapter userProfilesAdapter = new
UserProfilesTableAdapter();
    dsUserProfiles.UserProfilesDataTable users;
    users = userProfilesAdapter.GetUserprofileByUserName(userName);
    foreach (dsUserProfiles.UserProfilesRow user in users)
    {
        txtCompleteName.Text = user.LastNames.Trim() + " \n" +
user.Names.Trim() + " \n\nPhone:      " + user.Phone +
" \n\nCellphone: " + user.Cellphone;

        UserName.Text = user.UserName.Trim();
        txbNames.Text = user.Names.Trim();
        txbLastNames.Text = user.LastNames.Trim();
        //ucPhoto
        txbIdentityCard.Text = user.IdentityCard.Trim();
        txbBloodType.Text = user.BloodType.Trim();
        ucGender.SelectedValue = user.Gender.Trim();
        ucBirthday.SelectedDate = user.Birthday;
        txbFatherName.Text = user.FatherName.Trim();
        txbMotherName.Text = user.MotherName.Trim();
        txbPrincipalStreet.Text = user.PrincipalStreet.Trim();
        txbTransversalStreet.Text = user.TransversalStreet.Trim();
        txbNumber.Text = user.Number.Trim();
        ucCountry.Text = user.Country.Trim();
        txbCity.Text = user.City.Trim();
        txbPhone.Text = user.Phone.Trim();
        txbCellphone.Text = user.Cellphone.Trim();
        ucMaritalStatus.SelectedValue = user.MaritalStatus.Trim();
    }
    btnDelete.Enabled = true;
    btnUpdate.Enabled = true;
}
protected void btnDelete_Click(object sender, EventArgs e)
{
    UserProfilesTableAdapter userProfilesAdapter = new
UserProfilesTableAdapter();
    string UserName = this.UserName.Text.Trim();
    if (UserName == "")
    {
    }
    else
    {
        userProfilesAdapter.Delete(UserName);
        Users user = new Users(UserName);
        user.DeleteUser();
    }
}
}

```

## BIBLIOGRAFÍA

- Evaluación de los aprendizajes. Ministerio de Educación y Cultura, Programa de Mejoramiento y Capacitación docente, 2004. Quito- Ecuador
- Evaluación de los aprendizajes. Ministerio de Educación y Cultura, Programa de Mejoramiento y Capacitación docente, 2000. Quito- Ecuador
- Pedagogía de la Evaluación. Elva Poveda, 1993. Quito-Ecuador
- Writing multiple-choice test items. Evanston, Computer-Adaptive Technologies. HALADYNA, T. M.; DOWNING, S. M., y RODRIGUEZ, M.: "A review of multiple-choice item writing. 2001
- Las Teorías del Aprendizaje. Poveda, Elva. Quito. 1985
- Pedagogía de la Evaluación. Poveda, Elva. Quito. 1993
- Evaluación - Guía Práctica para Profesores. Tenbrink, Terry. Madrid. 1981.
- TechNet. Windows Server 2003. Microsoft 2003
- <http://tramullas.com/documatica/2-2.html>
- <http://equipo1-262m.blogspot.com/2009/03/tipos-de-servidores.html>
- Introducción Microsoft.Net. Desarrollador 5 Estrellas.2005
- [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_tres\\_niveles](http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles)
- <http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/technologies/iis.aspx>
- <http://www.el-mundo.es/navegante/99/marzo/15/encryptar77.pdf>
- [http://www.hospedajeydominios.com/mambo/documentacion-manual\\_php-pagina-security\\_database.html](http://www.hospedajeydominios.com/mambo/documentacion-manual_php-pagina-security_database.html)
- [http://informatica.uv.es/iiguia/MC/Teoria/mc\\_capitulo12.pdf](http://informatica.uv.es/iiguia/MC/Teoria/mc_capitulo12.pdf)
- <http://fbielma.org/conferencia/itesco/DB-Security.pdf>
- <http://www.ldc.usb.ve/~poc/Seguridad/protocolos.pdf>
- <http://www.toefl.org>
- <http://cflares334.blogspot.es/1192848180/>
- [http://es.wikipedia.org/wiki/Modelo\\_de\\_prototipos](http://es.wikipedia.org/wiki/Modelo_de_prototipos)
- <http://www.sidar.org/recur/desdi/traduc/es/visitable/tecnicas/Prototyping.htm>
- [http://es.wikipedia.org/wiki/Windows\\_Server\\_2003](http://es.wikipedia.org/wiki/Windows_Server_2003)

- <http://geeks.ms/blogs/juansa/archive/2008/10/12/servicios-del-sistema-en-windows-server-2003.aspx>
- <http://www.microsoft.com/latam/technet/recursos/howto/windows2003/paso1.msp>
- <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/apps/erver.msp>
- <http://www.rediris.es/difusion/publicaciones/boletin/58-59/ponencia10.html>
- <http://www.microsoft.com/latam/windowsserver2003/evaluation/whyupgrade/to/p10best.msp>
- <http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=258>
- <http://www.microsoft.com/latam/technet/productos/windows/windowsserver2003/ncrypte.msp>
- <http://geeks.ms/blogs/jelperu/archive/2008/06/25/como-configurar-un-servidor-de-correo-en-server-2003-sin-usar-exchange-server.aspx>
- <http://technet2.microsoft.com/windowsserver/es/library/702d3b5d-91be-40ed-a7e2-cf35ac978e193082.msp?mfr=true>
- <http://sigma.poligran.edu.co/politecnico/apoyo/cuadernos/windows.pdf>
- <http://blogs.clavedigital.com/ncastillo/category/video-y-audio-stream-linux/>
- <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-httpd.html>
- <http://www.desarrolloweb.com/articulos/1112.php>
- <http://www.alegsa.com.ar/Dic/servidor%20de%20base%20de%20datos.php>
- <http://ingenieriadesoftwareudbciclo2008.blogspot.com/2008/08/ingenieria-de-software.html>