

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE APLICACIONES DISTRIBUIDAS *ONLINE* Y *SMART CLIENT* PARA DISPOSITIVOS MÓVILES UTILIZANDO .NET, QUE PERMITAN ACCEDER AL SISTEMA DE ADMINISTRACIÓN ESTUDIANTIL

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y REDES DE INFORMACIÓN**

**ARMIJOS CURIPOMA ADRIÁN ESTUARDO
adrianarmijos@yahoo.com
PELÁEZ RAMOS VERÓNICA ELIZABETH
vepelaez@yahoo.com**

**DIRECTOR: ING. PABLO HIDALGO
phidalgo@mailfie.epn.edu.ec**

Quito, marzo 2008

DECLARACIÓN

Nosotros, Adrián Estuardo Armijos Curipoma y Verónica Elizabeth Peláez Ramos, declaramos que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Adrián Estuardo Armijos Curipoma

Verónica Elizabeth Peláez Ramos

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Adrián Estuardo Armijos Curipoma y Verónica Elizabeth Peláez Ramos, bajo mi supervisión.

ING. PABLO HIDALGO
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Mis sinceros agradecimientos al Ing. Pablo Hidalgo, ya que sin su ayuda este trabajo no sería realidad.

Mi gratitud para mis tíos Segundo Armijos y Silvia Moreira, quiénes me hicieron parte de su hogar, su constante apoyo me permitió terminar mis estudios universitarios.

Adrián.

AGRADECIMIENTOS

Mis agradecimientos a todas las personas, profesores, amigos y familiares que se vieron involucrados en mi formación profesional.

Un agradecimiento especial al Ingeniero Pablo Hidalgo por sus enseñanzas y su colaboración para la realización de este proyecto.

Vero.

DEDICATORIA

A mi esposa e hija, Verito y Catalina.

A mis padres Aidé y Eladio, por su permanente sacrificio y apoyo constante en todos los momentos.

A mis segundos padres Rosa y Luis que siempre han sido un referente en mi vida.

A mis hermanos Santiago, Paulina, Enrique, Vinicio, Galo y Medardo pues siempre creyeron en mí y porque nunca me faltará su apoyo.

También quiero dedicar este trabajo a mis amigos de universidad, en especial a Santiago y Víctor.

Adrián.

DEDICATORIA

Este trabajo va dedicado a mi mami, por darme su amor, su apoyo incondicional, y por siempre estar cuando te necesité; a mi papi por creer en mí y apoyarme siempre; a mis hermanos por su cariño y su fe en mí, a mi Adriancito por el amor, comprensión, dedicación y esfuerzo, a mi Catita el mayor de mis tesoros, quien me da la inspiración para ser cada día mejor; a mis tíos, abuelitos y amigos de quienes no faltó una palabra de aliento.

Vero.

CONTENIDO

1	INTRODUCCIÓN.....	0
1.1	Dispositivos Móviles.....	2
1.1.1	Téfonos Móviles.....	3
1.1.1.1	Téfonos inteligentes.....	3
1.1.2	Asistentes Digitales Personales (PDA).....	4
1.1.2.1	<i>Pocket PC y Pocket PC Phone</i>	4
1.1.3	Sistemas Operativos.....	5
1.1.3.1	<i>Symbian</i>	6
1.1.3.2	<i>PalmOS</i>	6
1.1.3.3	<i>Windows Mobile</i>	8
1.1.4	Selección del dispositivo móvil.....	10
1.2	Tecnología .NET.....	11
1.2.1	<i>Framework .NET</i>	11
1.2.1.1	Características.....	11
1.2.1.2	Arquitectura del <i>.NET Framework</i>	12
1.2.1.2.1	Entorno de Ejecución Common Language Runtime.....	13
1.2.1.2.2	Ejecución de una aplicación .NET.....	14
1.2.1.2.3	Biblioteca de clases del <i>.NET Framework</i>	15
1.2.1.3	Versiones del <i>.NET Framework</i>	15
1.2.2	<i>.NET Compact Framework</i>	16
1.2.2.1	Arquitectura del <i>.NET Compact Framework</i>	16
1.2.2.2	Biblioteca de clases de <i>.NET Compact Framework</i>	17
1.2.3	<i>ADO.NET</i>	19
1.2.3.1	Arquitectura <i>ADO.NET</i>	19
1.2.4	<i>ASP.NET</i>	20
1.2.4.1	Componentes de una aplicación <i>ASP.NET</i>	21
1.2.4.2	Formularios Web.....	22
1.2.4.2.1	Eventos de un Formulario Web.....	22
1.2.4.2.2	Ciclo de vida de un Formulario Web.....	23
1.2.4.3	Autenticación.....	23
1.2.4.3.1	Autenticación basada en Formularios.....	24
1.2.4.3.2	Autenticación basada en Windows.....	24
1.2.4.3.3	Autenticación basada en Passport.....	24
1.2.4.4	Autorización.....	25
1.2.5	<i>ASP.NET Mobile</i>	25
1.2.5.1	Controles <i>ASP.NET Mobile</i>	25
1.2.6	Servicios Web <i>XML</i>	27
1.2.6.1	Infraestructura de servicios Web <i>XML</i>	27
1.2.6.2	Autenticación en servicios Web.....	29
1.2.6.3	Autorización en servicios Web.....	30
2	DISEÑO E IMPLEMENTACIÓN DEL SERVICIO WEB.....	30
2.1	Requerimientos.....	31
2.1.1	Requerimientos del negocio.....	31
2.1.1.1	Lista de actores del negocio.....	31
2.1.1.2	Lista de casos de uso del negocio.....	32
2.1.1.3	Diagrama de casos de uso del negocio.....	32
2.1.1.4	Especificación de casos de uso del negocio.....	33
2.1.1.5	Diagrama de actividad de casos de uso del negocio.....	34
2.1.1.5.1	CU1: Publicar horarios.....	34
2.1.1.5.2	CU2: Consultar horarios.....	34
2.1.1.5.3	CU3: Consultar calificaciones.....	35

2.1.2	Requerimientos del Sistema	37
2.1.2.1	Lista de actores del sistema	37
2.1.2.2	Lista de casos de uso del sistema	37
2.1.2.3	Diagrama de casos de uso del sistema	38
2.1.2.4	Especificación de casos de uso del sistema	38
2.1.2.5	Diseño de Interfaces.....	42
2.2	Diseño	43
2.2.1	Base de datos.....	43
2.2.1.1	Base de Datos SAE.....	43
2.2.1.1.1	Tablas utilizadas del SAE	43
2.2.1.1.2	Campos utilizados de la tabla SAEESTUD.....	44
2.2.1.1.3	Campos utilizados de la tabla SAENOMMA.....	45
2.2.1.1.4	Campos utilizados de la tabla SAEHORAR.....	45
2.2.1.1.5	Campos utilizados de la tabla SAEPARAL.....	45
2.2.1.1.6	Campos utilizados de la tabla SAECALIF.....	46
2.2.1.1.7	Campos utilizados de la tabla SAEPROFE	46
2.2.1.1.8	Procedimientos Almacenados	46
2.2.1.2	Base de Datos SIEM.....	47
2.2.1.2.1	Tablas de la Base de Datos SIEM	47
2.2.1.2.2	Procedimientos Almacenados	48
2.2.2	Capa de Acceso a Datos.....	48
2.2.2.1	Diagrama de clases.....	48
2.2.3	Capa Reglas del Negocio.....	49
2.2.3.1	Diagrama de clases.....	49
2.2.3.2	Atributos.....	50
2.2.3.3	Métodos.....	54
2.2.4	Diagramas de Actividad.....	56
2.2.4.1	Diagrama de actividad Inicio de sesión.....	56
2.2.4.2	Diagrama de actividad Consultar información de cuenta.....	58
2.2.4.3	Diagrama de actividad Consultar horario	59
2.2.4.4	Diagrama de actividad Consultar calificaciones.....	60
2.2.4.5	Diagrama de actividad Consultar información de materia	61
2.2.5	Diagramas de Secuencia	62
2.2.5.1	Diagrama de secuencia Inicio de sesión.....	63
2.2.5.2	Diagrama de secuencia Consultar información de cuenta.....	64
2.2.5.3	Diagrama de secuencia Consultar horario	65
2.2.5.4	Diagrama de secuencia Consultar calificaciones.....	66
2.2.5.5	Diagrama de secuencia Consultar información de materia	66
2.3	Implementación.....	67

3 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN ONLINE..... 74

3.1	Requerimientos	74
3.1.1	Requerimientos del Negocio	74
3.1.2	Requerimientos del Sistema	74
3.1.2.1	Lista de actores	74
3.1.2.2	Lista de casos de uso.....	74
3.1.2.3	Diagrama de casos de uso.....	75
3.1.2.4	Especificación de casos de uso	76
3.1.2.5	Diseño de Interfaces.....	80
3.2	Diseño	83
3.2.1	Diagrama de Clases	83
3.2.2	Atributos	85
3.2.3	Métodos.....	91
3.2.4	Diagramas de Actividad.....	94
3.2.4.1	Diagrama de actividad Inicio de sesión.....	95
3.2.4.2	Diagrama de actividad Consultar información de cuenta.....	96

3.2.4.3	Diagrama de actividad Consultar horario	97
3.2.4.4	Diagrama de actividad Consultar calificaciones.....	99
3.2.4.5	Diagrama de actividad Consultar información de materia	100
3.2.4.6	Diagrama de actividad Cerrar sesión	101
3.2.5	Diagramas de Secuencia	102
3.2.5.1	Diagrama de secuencia Inicio de sesión.....	102
3.2.5.2	Diagrama de secuencia Consultar información de cuenta.....	103
3.2.5.3	Diagrama de secuencia Consultar horario	104
3.2.5.4	Diagrama de secuencia Consultar calificaciones.....	106
3.2.5.5	Diagrama de secuencia Consultar información de materia	107
3.2.5.6	Diagrama de secuencia Cerrar sesión	108
3.2.6	Implementación	108
4	DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN <i>SMART CLIENT</i>.....	115
4.1	Requerimientos	116
4.1.1	Requerimientos del Negocio	116
4.1.2	Requerimientos del Sistema	116
4.1.2.1	Lista de actores	116
4.1.2.2	Lista de casos de uso.....	116
4.1.2.3	Diagrama de casos de uso del sistema	117
4.1.2.4	Especificación de casos de uso del sistema	118
4.1.2.5	Diseño de Interfaces.....	121
4.2	Diseño	124
4.2.1	Diagrama de Clases	124
4.2.2	Atributos	125
4.2.3	Métodos.....	127
4.2.4	Diagramas de Actividad.....	129
4.2.4.1	Diagrama de actividad Inicio de sesión.....	129
4.2.4.2	Diagrama de actividad Consultar información de cuenta.....	130
4.2.4.3	Diagrama de actividad Consultar horario	131
4.2.4.4	Diagrama de actividad Consultar calificaciones.....	132
4.2.4.5	Diagrama de actividad Consultar información de materia	132
4.2.4.6	Diagrama de actividad Cerrar sesión	133
4.2.5	Diagramas de Secuencia	134
4.2.5.1	Diagrama de secuencia Inicio de sesión.....	134
4.2.5.2	Diagrama de secuencia Consultar información de cuenta.....	135
4.2.5.3	Diagrama de secuencia Consultar horario	136
4.2.5.4	Diagrama de secuencia Consultar calificaciones.....	137
4.2.5.5	Diagrama de secuencia Consultar información de materia	138
4.2.5.6	Diagrama de secuencia Cerrar sesión	139
4.3	Implementación.....	139
5	PRUEBAS Y COMPARACIÓN	148
5.1	Casos de Pruebas	148
5.2	Realización de Pruebas	149
5.2.1	Ambiente de pruebas funcionales	149
5.2.2	Resultados de la aplicación <i>Online</i>	150
5.2.3	Resultados de la aplicación <i>Smart Client</i>	154
5.3	Medidas de comparación.....	158
5.3.1	Herramientas a utilizarse.....	158
5.3.2	Ambiente de pruebas de rendimiento	160
5.3.3	Carga de trabajo de las pruebas de rendimiento	163
5.3.4	Métricas para las pruebas	163

5.3.4.1	Tiempos de respuesta	164
5.3.4.2	Eficiencia	164
5.3.4.3	Utilización de recursos	165
5.3.4.3.1	Procesador	165
5.3.4.3.2	Memoria	166
5.3.4.3.3	E/S Disco	166
5.3.4.3.4	Recursos de red	167
5.3.5	Métricas obtenidas de la aplicación <i>Online</i>	167
5.3.5.1	Tiempos de respuesta	167
5.3.5.2	Eficiencia	168
5.3.5.3	Utilización de recursos	169
5.3.5.3.1	Procesador	169
5.3.5.3.2	Memoria	170
5.3.5.3.3	E/S Disco	171
5.3.5.3.4	Recursos de Red	172
5.3.6	Métricas obtenidas de la aplicación <i>Smart Client</i>	173
5.3.6.1	Tiempos de respuesta	173
5.3.6.2	Eficiencia	173
5.3.6.3	Utilización de recursos	174
5.3.6.3.1	Procesador	174
5.3.6.3.2	Memoria	175
5.3.6.3.3	E/S Disco	176
5.3.6.3.4	Recursos de Red	177
5.4	Comparación	177
5.4.1	Tiempos de respuesta	178
5.4.2	Eficiencia	179
5.4.3	Utilización de recursos	181
5.4.3.1	Procesador	181
5.4.3.2	Memoria	185
5.4.3.3	E/S Disco	186
5.4.3.4	Recursos de red	189
5.4.4	Facilidad para el desarrollo	191
5.4.5	Facilidad para el mantenimiento	191
5.5	Costo del sistema	192
5.5.1	Actividades	192
5.5.1.1	Diagrama Gantt de actividades	193
5.5.2	Recursos	193
5.5.2.1	Recursos por actividad	193
5.5.3	Resumen de costos	195
6	CONCLUSIONES y RECOMENDACIONES	196
6.1	Conclusiones	196
6.2	Recomendaciones	198
	REFERENCIAS BIBLIOGRÁFICAS	200
ANEXOS		
Anexo A: Manual <i>Dell AXIM V51</i>		
	Anexo A-1: Generalidades del dispositivo <i>Dell AXIM</i>	A-1-1
	Anexo A-2: Como conectarse	A-2-1
	Anexo A-3: Adición y eliminación de programas	A-3-1
	Anexo A-4: Características detalladas del dispositivo <i>Dell AXIM</i>	A-4-1
Anexo B: Código de implementación		
	Anexo B-1: Procedimientos almacenados	B-1-1
	Anexo B-2: Autenticación en el servicio Web	B-2-1
Anexo C: Resumen de los resultados de pruebas		

Anexo C-1: Manual <i>Web Application Stress</i>	C-1-1
Anexo C-2: Manual <i>Management Console</i>	C-2-1
Anexo C-3: Resumen de resultados de pruebas de la aplicación <i>Online</i>	C-3-1
Anexo C-4: Resumen de resultados de pruebas de la aplicación <i>Smart Client</i>	C-3-1
Anexo D: Código completo de las aplicaciones (CD).....	D-1

ÍNDICE DE FIGURAS

Capítulo 1

Figura 1.1 Teléfonos Inteligentes (<i>Smartphone</i>) con <i>Windows Mobile</i>	4
Figura 1.2 (a) <i>Pocket PC</i> , (b) <i>Pocket PC Phone</i>	5
Figura 1.3 Arquitectura <i>Framework .NET</i>	12
Figura 1.4 Ejecución de una aplicación <i>.NET</i>	14
Figura 1.5 Arquitectura del <i>.NET Compact Framework</i>	16
Figura 1.6 Arquitectura <i>ADO.NET</i>	20
Figura 1.7 Componentes de una aplicación <i>ASP.NET</i>	22
Figura 1.8 Modelo de eventos <i>WebForm</i>	22
Figura 1.9 Infraestructura de servicios Web <i>XML</i>	28

Capítulo 2

Figura 2.1 Diagrama de casos de uso del negocio.....	32
Figura 2.2 Diagrama de actividad del negocio CU1.....	34
Figura 2.3 Diagrama de actividad del negocio CU2.....	35
Figura 2.4 Diagrama de actividad del negocio CU3.....	36
Figura 2.5 Diagrama de casos de uso del sistema.....	38
Figura 2.6 Diagrama relacional de las tablas del SAE utilizadas.....	44
Figura 2.7 Diagrama de relaciones de la Base de Datos <i>SIEM</i>	47
Figura 2.8 Clase AccesoDatos <i>SIEM</i>	49
Figura 2.9 Diagrama de clases de la capa de reglas de negocio	50
Figura 2.10 Atributos de la clase <i>Estudiante</i>	51
Figura 2.11 Atributos de la clase <i>TicketDeAutenticacion</i>	51
Figura 2.12 Atributos de la clase <i>Materia</i>	52
Figura 2.13 Atributos de la clase <i>DetalleMateria</i>	52
Figura 2.14 Atributos de la clase <i>HoraClase</i>	53
Figura 2.15 Atributos de la clase <i>Calificacion</i>	53
Figura 2.16 Diagrama de actividad Inicio de sesión	57
Figura 2.17 Diagrama de actividad Consultar información de cuenta	58
Figura 2.18 Diagrama de actividad Consultar horario.....	60
Figura 2.19 Diagrama de actividad Consultar calificaciones	61
Figura 2.20 Diagrama de actividad Consultar información de materia	62
Figura 2.21 Diagrama de secuencia Inicio de sesión	63
Figura 2.22 Diagrama de secuencia Consultar información de cuenta	64
Figura 2.23 Diagrama de secuencia Consultar horario	65
Figura 2.24 Diagrama de secuencia Consultar calificaciones	66
Figura 2.25 Diagrama de secuencia Consultar información de materia	67
Figura 2.26 Agregar procedimiento almacenado	68
Figura 2.27 Abrir el diseñador de consultas <i>SQL</i>	69
Figura 2.28 Generador de consultas <i>SQL</i>	69
Figura 2.29 Crear un servicio Web <i>ASP.NET</i>	72

Capítulo 3

Figura 3.1 Diagrama de casos de uso de la aplicación <i>OnlineSIEM</i>	75
Figura 3.2 Interfaz de usuario para inicio de sesión	80
Figura 3.3 Interfaz de usuario para información del estudiante	80
Figura 3.4 Interfaz de usuario para horario de clases.....	81
Figura 3.5 Interfaz de usuario para calificaciones.....	82
Figura 3.6 Diseño de interfaz para información de materia	83
Figura 3.7 Diagrama de clases de la aplicación <i>Online</i>	84
Figura 3.8 Atributos de la clase <i>IniciarSesion</i>	85
Figura 3.9 Atributos de la clase <i>Estudiante</i>	86
Figura 3.10 Atributos de la clase <i>Materia</i>	87
Figura 3.11 Atributos de la Clase <i>Horario</i>	88
Figura 3.12 Atributos de la Clase <i>Calificaciones</i>	90

Figura 3.13 Atributos de la clase <i>CalificacionUC</i>	90
Figura 3.14 Atributos de la clase <i>HorarioUC</i>	90
Figura 3.15 Atributos de la clase <i>HoraClaseI</i>	91
Figura 3.16 Diagrama de actividad Inicio de sesión	95
Figura 3.17 Diagrama de actividad Consultar información de cuenta	97
Figura 3.18 Diagrama de actividad Consultar horario.....	98
Figura 3.19 Diagrama de actividad Consultar calificaciones	99
Figura 3.20 Diagrama de actividad Consultar información de materia	100
Figura 3.21 Diagrama de actividad Cerrar sesión.....	101
Figura 3.22 Diagrama de secuencia Inicio de sesión	102
Figura 3.23 Diagrama de secuencia Consultar información de cuenta	104
Figura 3.24 Diagrama de secuencia Consultar horario	105
Figura 3.25 Diagrama de secuencia Consultar calificaciones	106
Figura 3.26 Diagrama de secuencia Consultar información de materia	107
Figura 3.27 Diagrama de secuencia Cerrar sesión.....	108
Figura 3.28 Crear nuevo Sitio Web <i>ASP.NET</i>	109
Figura 3.29 Asignar nombre y ubicación para el nuevo sitio Web.....	109
Figura 3.30 Crear una nueva página para dispositivos móviles	110
Figura 3.31 Asignar nombre a la nueva página para móviles.....	110
Figura 3.32 Ver cuadro de herramientas	111
Figura 3.33 Agregar un control Web a una página Web.....	112
Figura 3.34 Agregar referencia Web	113
Figura 3.35 Seleccionar servicio Web.....	114

Capítulo 4

Figura 4.1 Diagrama de casos de uso de la aplicación <i>SmartSIEM</i>	117
Figura 4.2 Interfaz de usuario para inicio de sesión	122
Figura 4.3 Interfaz de usuario para información del estudiante.....	122
Figura 4.4 Interfaz de usuario para horario de clases.....	123
Figura 4.5 Interfaz de usuario para calificaciones.....	123
Figura 4.6 Diseño de interfaz para información de materia	124
Figura 4.7 Diagrama de clases de la aplicación <i>Smart Client</i>	125
Figura 4.8 Atributos de la clase <i>FormSmartSIEM</i>	126
Figura 4.9 Diagrama de actividad Inicio de sesión	129
Figura 4.10 Diagrama de actividad Consultar información de cuenta	130
Figura 4.11 Diagrama de actividad Consultar horario.....	131
Figura 4.12 Diagrama de actividad Consultar calificaciones	132
Figura 4.13 Diagrama de actividad Consultar información de materia	133
Figura 4.14 Diagrama de actividad Cerrar sesión.....	134
Figura 4.15 Diagrama de secuencia Inicio de sesión	134
Figura 4.16 Diagrama de secuencia Consultar información de cuenta	135
Figura 4.17 Diagrama de secuencia Consultar horario	136
Figura 4.18 Diagrama de secuencia Consultar calificaciones	137
Figura 4.19 Diagrama de secuencia Consultar información de materia	138
Figura 4.20 Diagrama de secuencia Cerrar sesión.....	139
Figura 4.21 Crear proyecto cliente inteligente	140
Figura 4.22 Seleccionar plantilla adecuada de dispositivo	140
Figura 4.23 Formulario para proyecto <i>Pocket PC</i>	141
Figura 4.24 <i>Microsoft ActiveSync</i>	142
Figura 4.25 Administrador de emuladores de dispositivo	142
Figura 4.26 Conectar un emulador.....	143
Figura 4.27 Colocar en la base un emulador	143
Figura 4.28 Ver cuadro de herramientas	144
Figura 4.29 Agregar controles al formulario.....	145

Capítulo 5

Figura 5.1 Configuración de ambiente de pruebas para la aplicación <i>Online</i>	150
Figura 5.2 Prueba iniciar sesión en la aplicación <i>Online</i>	151
Figura 5.3 Prueba Consultar información de cuenta en la aplicación <i>Online</i>	151

Figura 5.4 Prueba Consultar horario en la aplicación <i>Online</i>	152
Figura 5.5 Prueba Consultar calificaciones en la aplicación <i>Online</i>	152
Figura 5.6 Prueba Consultar información de materia en la aplicación <i>Online</i>	153
Figura 5.7 Prueba cerrar sesión en la aplicación <i>Online</i>	153
Figura 5.8 Prueba iniciar sesión en la aplicación <i>Smart Client</i>	154
Figura 5.9 Prueba Consultar información de cuenta en la aplicación <i>Smart Client</i>	155
Figura 5.10 Prueba Consultar horario en la aplicación <i>Smart Client</i>	156
Figura 5.11 Prueba Consultar calificaciones en la aplicación <i>Smart Client</i>	156
Figura 5.12 Prueba Consultar información de materia en la aplicación <i>Smart Client</i>	157
Figura 5.13 Prueba cerrar sesión en la aplicación <i>Smart Client</i>	158
Figura 5.14 Web Application Stress	159
Figura 5.15 <i>Microsoft Management Console</i>	160
Figura 5.16 Configuración del ambiente de pruebas para la aplicación <i>Online</i>	162
Figura 5.17 Configuración del ambiente de pruebas para la aplicación <i>Smart Client</i>	162
Figura 5.18 Comparación de <i>Time to first byte</i> de las aplicaciones	178
Figura 5.19 Comparación de <i>Time to last byte</i> de las aplicaciones.....	179
Figura 5.20 Comparación de Solicitudes por ejecución de las aplicaciones	179
Figura 5.21 Comparación de Solicitudes por segundo de las aplicaciones	180
Figura 5.22 Comparación de Peticiones de extensión ISAPI/s de las aplicaciones.....	181
Figura 5.23 Comparación del % de tiempo de interrupción de las aplicaciones	182
Figura 5.24 Comparación del % de tiempo de procesador de las aplicaciones	182
Figura 5.25 Comparación del % de tiempo privilegiado de las aplicaciones.....	183
Figura 5.26 Comparación de Cambios de contexto/s de las aplicaciones	183
Figura 5.27 Comparación de Longitud de la cola del procesador de las aplicaciones.....	184
Figura 5.28 Comparación de Errores de caché/s de las aplicaciones.....	185
Figura 5.29 Comparación de MBytes disponibles de las aplicaciones	186
Figura 5.30 Comparación de Escrituras en segundos de las aplicaciones	186
Figura 5.31 Comparación de Longitud media de la cola de disco de las aplicaciones	187
Figura 5.32 Comparación de Long. promedio de cola de escritura de disco de las aplicaciones.	187
Figura 5.33 Comparación de Long. promedio de cola de lectura de disco de las aplicaciones....	188
Figura 5.34 Comparación de Media en segundos/lectura de las aplicaciones.....	188
Figura 5.35 Comparación de Bytes enviados/s de las aplicaciones.....	189
Figura 5.36 Comparación de Bytes recibidos/s de las aplicaciones.....	189
Figura 5.37 Comparación de Total de bytes/s de las aplicaciones.....	190
Figura 5.38 Diagrama Gantt de actividades.....	194

ÍNDICE DE TABLAS

Capítulo 1

Tabla 1.1 Versiones de <i>Windows Mobile 6</i>	9
Tabla 1.2 Características <i>Dell Axim X51v</i>	10
Tabla 1.3 Versiones del <i>.NET Framework</i>	15
Tabla 1.4 Controles permitidos en el <i>.NET Compact Framework</i>	18
Tabla 1.5 Controles <i>ASP.NET Mobile</i>	26
Tabla 1.6 Autenticación en servicios Web	29

Capítulo 2

Tabla 2.1 Especificación caso de uso del negocio CU1	33
Tabla 2.2 Especificación caso de uso del negocio CU2	33
Tabla 2.3 Especificación caso de uso del negocio CU3	33
Tabla 2.4 Especificación del caso de uso Inicio de sesión	39
Tabla 2.5 Especificación del caso de uso Consultar información de cuenta	39
Tabla 2.6 Especificación del caso de uso Consultar horario	40
Tabla 2.7 Especificación del caso de uso Consultar calificaciones	40
Tabla 2.8 Especificación del caso de uso Consultar información de materias	41
Tabla 2.9 Especificación del caso de uso Cerrar sesión	41
Tabla 2.10 Descripción de las tablas utilizadas en el sistema <i>SIEM</i>	44
Tabla 2.11 Descripción de los campos de la tabla <i>SAEESTUD</i>	45
Tabla 2.12 Descripción de los campos de la tabla <i>SAENOMMA</i>	45
Tabla 2.13 Descripción de los campos de la tabla <i>SAEHORAR</i>	45
Tabla 2.14 Descripción de los campos de la tabla <i>SAEPARAL</i>	46
Tabla 2.15 Descripción de los campos de la tabla <i>SAECALIF</i>	46
Tabla 2.16 Descripción de los campos de la tabla <i>SAEPROFE</i>	46
Tabla 2.17 Procedimientos almacenados incluidos en la base de datos <i>SAE</i>	47
Tabla 2.18 Procedimientos almacenados de la base de datos <i>SIEM</i>	48
Tabla 2.19 Descripción de los métodos de la clase <i>AccesoDatosSIEM</i>	49
Tabla 2.20 Atributos de la clase <i>Estudiante</i>	51
Tabla 2.21 Descripción de los atributos de la clase <i>TicketDeAutenticacion</i>	51
Tabla 2.22 Descripción de los atributos de la clase <i>Materia</i>	52
Tabla 2.23 Descripción de los atributos de la clase <i>DetalleMateria</i>	52
Tabla 2.24 Descripción de los atributos de la clase <i>HoraClase</i>	53
Tabla 2.25 Descripción de los atributos de la clase <i>Calificacion</i>	53
Tabla 2.26 Descripción de las propiedades de la clase <i>Estudiante</i>	54
Tabla 2.27 Descripción de las propiedades de la clase <i>Materia</i>	54
Tabla 2.28 Descripción de las propiedades de la clase <i>DetalleMateria</i>	55
Tabla 2.29 Descripción de las propiedades de la clase <i>HoraClase</i>	55
Tabla 2.30 Descripción de las propiedades de la clase <i>Calificacion</i>	56

Capítulo 3

Tabla 3.1 Especificación del caso de uso Inicio de sesión	76
Tabla 3.2 Especificación del caso de uso Consultar información de cuenta	77
Tabla 3.3 Especificación del caso de uso Consultar horario	77
Tabla 3.4 Especificación del caso de uso Consultar calificaciones	78
Tabla 3.5 Especificación del caso de uso Consultar información de materia.....	79
Tabla 3.6 Especificación del caso de uso Cerrar sesión	79
Tabla 3.7 Descripción de los atributos de la clase <i>IniciarSesion</i>	86
Tabla 3.8 Descripción de los atributos de la clase <i>Estudiante</i>	87
Tabla 3.9 Descripción de los atributos de la clase <i>Materia</i>	88
Tabla 3.10 Descripción de los atributos de la clase <i>Horario</i>	89
Tabla 3.11 Descripción de los atributos de la clase <i>Calificaciones</i>	89
Tabla 3.12 Descripción de los atributos de la clase <i>CalificacionUC</i>	90
Tabla 3.13 Descripción de los atributos de la clase <i>HorarioUC</i>	91
Tabla 3.14 Descripción de los atributos de la clase <i>HoraClaseI</i>	91

Tabla 3.15 Descripción de los métodos de la clase <i>IniciarSesion</i>	92
Tabla 3.16 Descripción de los métodos de la clase <i>Estudiante</i>	92
Tabla 3.17 Descripción de los métodos de la clase <i>Materia</i>	93
Tabla 3.18 Descripción de los métodos de la clase <i>Horario</i>	93
Tabla 3.19 Descripción de los métodos de la clase <i>Calificaciones</i>	93
Tabla 3.20 Descripción de los métodos de la clase <i>CalificacionUC</i>	94
Tabla 3.21 Descripción de los métodos de la clase <i>HorarioUC</i>	94
Tabla 3.22 Descripción de los métodos de la clase <i>HoraClaseI</i>	94

Capítulo 4

Tabla 4.1 Especificación del caso de uso Inicio de sesión.....	118
Tabla 4.2 Especificación del caso de uso Consultar información de cuenta.....	119
Tabla 4.3 Especificación del caso de uso Consultar horario.....	119
Tabla 4.4 Especificación del caso de uso Consultar calificaciones.....	120
Tabla 4.5 Especificación del caso de uso Consultar información de materia.....	120
Tabla 4.6 Especificación del caso de uso Cerrar sesión.....	121
Tabla 4.7 Descripción de los atributos de la clase <i>FormSmartSIEM</i>	126
Tabla 4.8 Descripción de los métodos de la clase <i>FormSmartSIEM</i>	128

Capítulo 5

Tabla 5.1 Características del servidor.....	149
Tabla 5.2 Características del dispositivo móvil.....	149
Tabla 5.3 Características del servidor.....	161
Tabla 5.4 Características del equipo cliente.....	161
Tabla 5.5 Parámetros de medición de tiempos de respuesta.....	164
Tabla 5.6 Parámetros de medición de eficiencia.....	165
Tabla 5.7 Parámetros de medición de utilización del procesador.....	165
Tabla 5.8 Parámetros de medición de utilización de memoria.....	166
Tabla 5.9 Parámetros de medición de utilización de E/S Disco.....	166
Tabla 5.10 Parámetros de medición de utilización de E/S Red.....	167
Tabla 5.11 Tiempos de respuesta de la aplicación <i>Online</i>	168
Tabla 5.12 Eficiencia de la aplicación <i>Online</i>	169
Tabla 5.13 Utilización del procesador de la aplicación <i>Online</i>	170
Tabla 5.14 Utilización de memoria de la aplicación <i>Online</i>	171
Tabla 5.15 Utilización de disco de la aplicación <i>Online</i>	171
Tabla 5.16 Utilización de red de la aplicación <i>Online</i>	172
Tabla 5.17 Tiempos de respuesta de la aplicación <i>Smart Client</i>	173
Tabla 5.18 Eficiencia de la aplicación <i>Smart Client</i>	174
Tabla 5.19 Utilización del procesador de la aplicación <i>Smart Client</i>	175
Tabla 5.20 Utilización de memoria de la aplicación <i>Smart Client</i>	176
Tabla 5.21 Utilización de disco de la aplicación <i>Smart Client</i>	176
Tabla 5.22 Utilización de red de la aplicación <i>Smart Client</i>	177
Tabla 5.23 Personal de Ingeniería.....	193
Tabla 5.24 Recursos por actividad.....	193
Tabla 5.25 Costo de actividades.....	195
Tabla 5.26 Gastos varios.....	195
Tabla 5.27 Costo total del proyecto.....	196

RESUMEN

Capítulo 1: Introducción

Se presenta conceptos de dispositivos móviles, los tipos de dispositivos que se encuentran en el mercado, sus sistemas operativos. Se detalla la evolución de *Microsoft* en mundo de los móviles, señalando su sistema operativo *Windows Mobile* y los dispositivos del tipo *Pocket PC*.

También se hace un resumen de la tecnología *.NET*, se resumen los más importantes componentes de esta tecnología: *Framework .NET*, *Compact Framework .NET*, *ADO.NET*, *ASP.NET*, *ASP.NET Mobile*.

Capítulo 2: Desarrollo e implementación del servicio Web

Se propone un diseño de dos capas para el acceso a base de datos del Sistema de Administración Estudiantil (SAE). La capa de acceso a datos ejecuta procedimientos almacenados para obtener los datos del SAE; la capa de reglas de negocio define un conjunto de clases que encapsula los datos obtenidos del SAE.

Capítulo 3: Desarrollo e implementación de la aplicación *Online*

Obteniendo los datos del servicio Web se diseña e implementa una capa de presentación *Online*. Para esto se utiliza *ASP.NET Mobile* e *Internet Information Server (IIS)*.

Capítulo 4: Desarrollo e implementación de la aplicación *Smart Client*

Se diseña e implementa una aplicación que consulta los datos del SAE directamente sobre el servicio Web. Para esto se desarrolla una aplicación para el *Compact Framework* del dispositivo *Pocket PC*.

Capítulo 5: Pruebas y Comparación

Las aplicaciones desarrolladas son probadas generando requerimientos mediante la herramienta *Microsoft Web Application Stress*; con esto se logra generar una carga de trabajo en el servidor. Esto a su vez hace que el servidor procese cada

solicitud para responder la misma; esta actividad en los servidores es censada para obtener datos que servirán para comparar el desempeño de las dos aplicaciones.

Capítulo 6: Conclusiones y Recomendaciones

Este proyecto finaliza haciendo conclusiones y recomendaciones acerca del trabajo realizado.

Anexos

Se creyó conveniente incluir información adicional referente a algunos temas, la cual se presenta en cuatro secciones de anexos.

El Anexo A denominado Manual *Dell AXIM V51*, contiene información sobre el dispositivo *Pocket PC* utilizado. Se trata de un manual de usuario en el cual se indican las características y funcionalidades del dispositivo.

El Anexo B nombrado Código de Implementación, contiene el código de procedimientos almacenados que fueron necesarios agregarlos a la base de datos para obtener la información y se explica la implementación en el servicio Web.

El Anexo C designado Resumen de los resultados de pruebas, contiene los manuales de las herramientas *Web Application Stress* y *Management Console*. Además se muestran los resultados tabulados de las pruebas de funcionamiento y rendimiento de las aplicaciones *Online* y *Smart Client*.

El Anexo D denominado Código completo de las aplicaciones, consta de un CD en el cual se incluye el código de las aplicaciones desarrolladas.

PRESENTACIÓN

Los dispositivos móviles y la transmisión inalámbrica de datos han marcado una nueva era en las comunicaciones. Los dispositivos móviles han evolucionado hasta convertirse en terminales inteligentes capaces de ejecutar aplicaciones empresariales; esto de la mano con las bondades de la telefonía celular, ha permitido que los usuarios finales tengan en un dispositivo móvil una herramienta indispensable en el trabajo, el hogar y la vida cotidiana.

Actualmente gran parte de estudiantes de la Carrera de Electrónica y Redes de Información, cuentan con un teléfono celular y en algunos casos *PDA*s, los cuales tiene la posibilidad de conectarse a Internet. El presente proyecto de titulación pretende implementar dos aplicaciones distribuidas que permitan a los estudiantes de la Carrera de Electrónica y Redes de Información consultar sus horarios y calificaciones por medio de un dispositivo móvil (*Pocket PC*) que tenga acceso a la red local de la Carrera. Este trabajo permitirá a los estudiantes tener una opción más de consulta de sus notas y horarios en cada bimestre.

Las aplicaciones desarrolladas son del tipo *Smart Client* y *Online*. Estas aplicaciones obtienen los datos del Sistema de Administración Estudiantil (SAE) mediante un servicio Web. En este contexto es necesario realizar un análisis comparativo a fin de establecer el rendimiento de cada aplicación detallando las ventajas y desventajas de cada una de ellas.

Capítulo 1

Introducción

1 INTRODUCCIÓN

Las tecnologías móviles se han convertido en los últimos años en uno de los motores clave en la industria de las telecomunicaciones. Los avances han permitido a teléfonos celulares conectarse a Internet a través de la red celular y a dispositivos de mano del tipo *PDA* hacer lo mismo a través de redes locales inalámbricas.

La comunicación móvil se ha hecho indispensable para la mayoría de usuarios y ha evolucionado en nuevas tecnologías que han permitido el acceso a servicio de datos. En el Ecuador la telefonía celular tiene gran acogida y las redes de área local inalámbrica están ganando popularidad. Este repunte se logra gracias a los beneficios de esta tecnología, como movilidad, portabilidad, acceso en cualquier sitio y en cualquier momento.

El gran número de usuarios que disponen de un dispositivo móvil, es una ventaja que está siendo explotada por los proveedores de servicios. Es así, que los servicios tradicionales de datos (correo electrónico, chat, navegación Web, servicios Web *XML*, etc.) ahora también están disponibles para usuarios que cuentan con un dispositivo móvil.

Con este argumento, el presente proyecto de titulación, se enfoca en el desarrollo de dos aplicaciones distribuidas para dispositivos móviles. La una del tipo *Online* y la otra del tipo *Smart Client*.

Los teléfonos celulares y *PDA*s, tienen diferentes sistemas operativos dependiendo de su hardware, fabricante, y modelo; esto dificulta el desarrollar una aplicación que se ejecute en cada una de estas plataformas propietarias pues se debería realizar una aplicación a medida para cada plataforma. La solución es el uso de una aplicación en línea o aplicación *Online*, la cual sería única y válida para la mayoría de dispositivos móviles que tengan acceso al servidor Web, mediante el navegador Web del dispositivo móvil.

Por otro lado el avance en hardware ha permitido tener dispositivos móviles cada vez más sofisticados, que pueden funcionar como terminales inteligentes; para

aprovechar estos avances se deben utilizar aplicaciones que se ejecuten en un dispositivo móvil o aplicaciones *Smart Client*.

Microsoft facilita el desarrollo de aplicaciones distribuidas para dispositivos móviles; para aplicaciones *Online* se utiliza *ASP.NET Mobile* y para aplicaciones *Smart Client* se desarrollan aplicaciones que se ejecutan sobre el *.NET Compact Framework* del dispositivo móvil.

En el presente capítulo se tratará generalidades de dispositivos móviles y las herramientas para el desarrollo de aplicaciones distribuidas para dispositivos móviles.

1.1 DISPOSITIVOS MÓVILES

Un dispositivo móvil es un equipo de tamaño pequeño, diseñado para que el usuario lo transporte de manera sencilla, además sus funcionalidades pueden ser utilizadas en movimiento. Son los dispositivos idóneos para trabajos de campo donde la información se puede requerir y/o presentar en cualquier momento y en cualquier sitio.

Entre las características de un dispositivo móvil se puede resaltar:

- Movilidad, el usuario puede desplazarse de un lugar a otro y durante este movimiento las funcionalidades del dispositivo deben ser utilizables.
- Portabilidad, estos dispositivos son fáciles de llevar, la mayoría son transportables en el bolsillo del usuario.
- Comunicación, esta capacidad permite al usuario acceso a servicios de voz, así como de datos.
- Igualmente se caracterizan por tener limitaciones en hardware y por consiguiente en software, poseen pantallas pequeñas, baja memoria, limitada capacidad de procesamiento y en algunos casos no existe un teclado para interactuar con el usuario.

Poseen algunas funcionalidades como acceso a Internet (ya sea por medio de la red celular o *Wi-Fi*), procesamiento de funciones matemáticas, agenda electrónica, permiten escuchar música, juegos, ver películas, etc.

Como ejemplo de estos dispositivos se pueden citar: sistemas de navegación de automóviles, sistemas de entretenimiento, teléfonos móviles, asistentes personales digitales.

Algunas de las cualidades que se debe tener en cuenta al adquirir un dispositivo móvil son: procesador, memoria, disco, pantalla *LCD*, micrófono, grabadora, ranuras de expansión, salida de auriculares, antena, interfaces de comunicación (infrarrojo, *Bluetooth*, *Wi-Fi*, *GSM*, *CDMA*).

1.1.1 TELÉFONOS MÓVILES

También conocidos como teléfonos celulares, son los dispositivos móviles con mayor presencia en el mercado.

Los teléfonos móviles trabajan junto con la red celular, su principal función es comunicar a los usuarios presentes en el área de cobertura de la red. En la actualidad es posible el acceso a Internet mediante estos dispositivos.

El avance tecnológico ha transformado un equipo que permite realizar llamadas telefónicas en un dispositivo inteligente con funciones tales como: reproductor multimedia, agenda electrónica, calculadora, cámara fotográfica, calendario, etc.

1.1.1.1 Teléfonos inteligentes

Teléfonos inteligentes o *Smartphones* son teléfonos celulares con características adicionales; incluyen la debida capacidad de procesamiento y memoria necesaria para brindar funcionalidades como: navegación por Internet, acceso a correo electrónico, agenda, etc. La Figura 1.1 muestra dos ejemplos de *Smartphones*.

Son dispositivos orientados a voz, mensajería, datos, permitiendo añadir nuevas aplicaciones para extender sus funcionalidades.



Figura 1.1 Teléfonos Inteligentes (*Smartphone*) con *Windows Mobile*⁽¹⁾

Se caracterizan por tener pantallas pequeñas (entre 90 x 60 y 200 x 150 píxeles) y teclado para el acceso a sus funciones.⁽²⁾

1.1.2 ASISTENTES DIGITALES PERSONALES (*PDA*)¹

“Es un computador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura”⁽³⁾

No posee teclado, en su lugar tienen una pantalla táctil que utiliza un lápiz óptico el cual realiza las funciones de un ratón.

Posee funcionalidades como el ingreso y reconocimiento de escritura a mano, acceso a Internet, sincronización con equipos de escritorio, comunicación mediante puertos *Bluetooth*, Infrarrojos, *Wi-Fi*.

1.1.2.1 *Pocket PC* y *Pocket PC Phone*

Un *Pocket PC* es un *PDA* que ejecuta el sistema operativo *Windows*, en otras palabras, *Pocket PC* es la solución *PDA* de *Microsoft*.

¹ *PDA*, *Personal Digital Assistant*

Un *Pocket PC* ofrece la versión móvil de las aplicaciones del paquete *Office*, se podrá utilizar *Word Mobile*, *Excel Mobile*, *PowerPoint Mobile* y *Outlook Mobile*. Dependiendo del modelo del *Pocket PC*, éste dispondrá de una interfaz *Wi-Fi* para el acceso a la red local cuando se encuentra en un punto de conexión inalámbrica.



Figura 1.2 (a) *Pocket PC*, (b) *Pocket PC Phone*⁽¹⁾

Un *Pocket PC Phone* tiene las mismas características de un *Pocket PC* y añade todas las funcionalidades de un teléfono móvil, incluido acceso inalámbrico a Internet haciendo uso de la red celular, ampliando su conectividad no solamente a la red local sino al área de cobertura del proveedor de telefonía móvil.

1.1.3 SISTEMAS OPERATIVOS

Un sistema operativo (SO) es una capa de software ubicada entre el hardware del dispositivo móvil y las aplicaciones. Los sistemas operativos para móviles, al igual que los SOs convencionales tienen la tarea de gestionar los recursos de hardware del dispositivo móvil; se encargan de coordinar el acceso a memoria y el uso del procesador entre los diferentes procesos. También controlan las interfaces de Entrada/Salida (*Bluetooth*, *Wi-Fi*, *GSM*, *CDMA*, interfaces seriales, infrarrojos, teclado, pantalla).

A diferencia de los SOs para computadores personales, en la que *Microsoft* ha dominado con su SO *Windows*, en dispositivos móviles no existe un SO hegemónico, esto se debe a que cada fabricante desarrolló su propio SO para sus equipos móviles.

A continuación se describen algunos de los sistemas operativos más destacados para móviles.

1.1.3.1 *Symbian*

Este SO aparece en 1998, como iniciativa de *Nokia*, *Sony Ericsson*, *Samsung*, *Siemens*, *Arima*, *Benq*, *Fujitsu*, *Lenovo*, *LG*, *Motorola*, *Mitsubishi Electric*, *Panasonic*, *Sharp*, etc., donde *Nokia* es el accionista mayoritario. ⁽⁴⁾

El SO *Symbian* tiene amplia presencia en el mercado de los SOs para *Smartphone*, debido a *Nokia*, empresa que ocupa el primer lugar en ventas y emplea *Symbian* en la mayoría de sus dispositivos móviles. ⁽⁵⁾

Sus principales características son:

- Separa la interfaz de usuario del resto de funcionalidades del sistema operativo, facilitando a cada empresa la adopción de distintas resoluciones y dimensiones de pantalla, calidad de color, teclados, etc.
- Soporte a la máquina virtual *Java*.
- El principal lenguaje para el desarrollo es *C++*.
- Proporciona un adecuado uso de energía.
- Permite reproducir contenido multimedia.

1.1.3.2 *PalmOS*

Diseñado por Jeff Hawkins para *US Robotics*, actualmente *3COM*. Es el sistema operativo para los dispositivos de la empresa *Palm*.

La primera versión del *PalmOS* aparece en 1996 con los equipos *Palm Pilot*. Luego la empresa *Palm* se dividió en *PalmOne* y *PalmSource*, la primera a cargo

del hardware y la siguiente enfocada al software. Actualmente este SO es utilizado por *PalmOne*, *Kyocera*, *Samsung* y *Sony*.

En sus principios el sistema operativo de *Palm* era el estándar para dispositivos móviles tipo *PDA*, pero en la actualidad ha perdido mercado debido a la aparición de nuevas marcas que han hecho muy competitivo este sector.

La versión original ha ido evolucionando para soportar los avances en hardware, así se tiene que el sistema operativo de *Palm* da soporte a conexiones inalámbricas mediante infrarrojos, *Wi-Fi* y *Bluetooth*, así como, permite el uso de pantallas con mayor resolución y la adopción de color, además incorporó un navegador Web.

Entre las funcionalidades incluidas en el sistema operativo de *Palm*, se puede citar: calendario, editor de notas, libreta de contactos, calculadora, además de una aplicación para sincronización de datos con un computador personal, la cual se denomina *HotSync*.

Sus características principales son:

- Es un sistema operativo de gran velocidad y ocupa poca memoria.
- Ofrece compatibilidad con documentos de *Word* y *Excel*.
- Permite conexión a Internet a través de interfaces *Wi-Fi* o *Bluetooth* y en el caso de los teléfonos que utilicen este sistema operativo la conexión también se la puede establecer a través de la red celular.
- Existe una amplia gama de aplicaciones que se ejecutan en este sistema operativo, proporcionando al usuario herramientas útiles en una variedad de escenarios.
- A partir de la versión 5 *PalmOS* aumentó la seguridad en la conexión a Internet, permitiendo navegación segura en portales Web, así como detección de *emails* mal intencionados.

1.1.3.3 *Windows Mobile*

Windows Mobile es el sistema operativo de *Microsoft* para dispositivos móviles. Este sistema operativo es utilizado en asistentes personales digitales, teléfonos inteligentes y reproductores multimedia portátiles.

Microsoft lanza en 1995 la primera versión de *Windows* para dispositivos móviles, con el nombre de *Windows CE (Compact Edition)*, este sistema operativo se ejecutaba en dispositivos móviles del tipo *Handheld PC*².

La aparición *Windows CE v2* en 1997, hizo posible que además de *Handheld PC* el sistema operativo se ejecutara en *PDA*s.

Con *Windows CE v3*, *Microsoft* entra en el mercado de los sistemas operativos para teléfonos celulares. La llegada de este sistema operativo en el 2000 incluye versiones para *Smartphone* y *Pocket PC*. El nombre *Windows CE* se mantiene hasta esta versión.

La cuarta versión sale al mercado en junio del 2003 con el nombre *Windows Mobile 2003*. Incluye mejoras como: *firewall*, soporte a VoIP³ y la posibilidad de conexión con puntos de acceso inalámbrico a través de *Wi-Fi* y *Bluetooth*, además cuenta con *Windows Media Player* versión 9. Este sistema operativo apareció en tres versiones: *Windows Mobile 2003* para *Pocket PC*, *Windows Mobile 2003* para *Pocket PC Phone Edition* y *Windows Mobile* para *Smartphone*.

En mayo del 2005 se revela la versión *Windows Mobile 5.0*, ésta incluye el *.NET Compact Framework 1.0*. Viene en tres versiones: *Pocket PC*, *Pocket PC Phone Edition* y para *Smartphone*.

Con *Windows Mobile 5.0* aparece el paquete *Office Mobile*, compuesto por *Word Mobile*, *Excel Mobile*, *PowerPoint Mobile* y *Outlook Mobile*; el paquete *Office Mobile* es compatible con la versión *Windows Mobile* para *Pocket PC* y *Pocket PC*

²*Handheld PC*, o PC de mano, es un dispositivo móvil evolución del *PDA*, dispone de teclado y de pantalla de mayor dimensión que las *PDA*; pero no han obtenido acogida en el mercado.

³ VoIP (*Voice over IP*), protocolo que permite enviar tráfico de voz en redes IP.

Phone Edition, además incluye el reproductor de *Windows Media 10*. La aplicación para intercambio de mensajes de correo electrónico *Outlook Mobile* es compatible con la versión *Pocket PC*, *Pocket PC Phone* y *Smartphone* del sistema operativo.

A la fecha, la última versión que *Microsoft* lanza al mercado es *Windows Mobile 6*. Se presenta en tres versiones que se muestran en la Tabla 1.1.

Versión	Dispositivo soportado
<i>Windows Mobile 6 Standard</i>	Teléfonos inteligentes
<i>Windows Mobile 6 Professional</i>	<i>Pocket PC Phone Edition</i>
<i>Windows Mobile 6 Classic</i>	<i>Pocket PC</i>

Tabla 1.1 Versiones de *Windows Mobile 6*

Las principales características de *Windows Mobile 6* son:

- Incluye *.NET Compact Framework v2 SP1* y *SQL Server Compact Edition* en la ROM.
- Admite las resoluciones de pantalla 800x480 y 320x320.
- El sistema operativo se actualiza automáticamente desde Internet.
- Permite un acceso mejorado al escritorio remoto.
- Soporte para VoIP.
- Incorpora *Windows Live*.
- Admite el cifrado de datos en las tarjetas de almacenamiento externo.
- Proporciona una herramienta conocida como *Smartfilter*, lo que hace que las búsquedas de *emails*, contactos, canciones, archivos, etc. sea más rápida.
- Tiene la capacidad de configurar el dispositivo como módem de una computadora portátil.
- Incorpora el soporte de *HTML* para *Outlook Mobile*.

1.1.4 SELECCIÓN DEL DISPOSITIVO MÓVIL

Las aplicaciones que se diseñarán en el presente proyecto de titulación, se implementarán en un dispositivo móvil del tipo *Pocket PC*; este dispositivo deberá contar con las siguientes características:

Características de software:

- Sistema Operativo: *Microsoft Windows Mobile 5.0* para *Pocket PC*.
- Navegador Web: *Internet Explorer Mobile*.

Características de hardware:

- Interfaz de Red: *Wi-Fi* del estándar a, b ó g.

Tomando en cuenta las características señaladas se decidió utilizar el dispositivo *Pocket PC Dell™ Axim™ X51v*.

Las características principales del dispositivo se resumen en la Tabla 1.2.

Característica	Descripción
Marca	<i>Dell</i>
Modelo	<i>Axim X51v</i>
Procesador	<i>Intel PXA270 624 MHz</i>
Memoria	RAM 64 MB
	ROM 256 MB
Conectividad Wireless	<i>Wi-Fi b</i>
	<i>Bluetooth</i>
	Infrarrojo

Tabla 1.2 Características *Dell Axim X51v*

El manual de usuario del dispositivo y las características en detalle del mismo se muestran en el Anexo A.

1.2 TECNOLOGÍA .NET

Al aparecer Internet surgieron también nuevas formas de comunicación, comercio, y educación. Estas nuevas oportunidades son explotadas por sectores públicos y privados, que ven en Internet un canal para llegar a las masas. A esto se debe sumar el creciente número de usuarios de Internet con acceso de banda ancha desde el hogar gracias a las tecnologías xDSL y el acceso móvil a Internet por redes de área local inalámbrica y redes celulares digitales.

En esta coyuntura aparece .NET, con el afán de satisfacer los requerimientos de conectividad de las aplicaciones de hoy.

1.2.1 FRAMEWORK .NET

El *Framework .NET* cuenta con el conjunto de herramientas necesarias para el desarrollo de aplicaciones, pero además provee un entorno de ejecución para las aplicaciones .NET.

“Microsoft .NET es una plataforma que permite el desarrollo y ejecución de aplicaciones”⁽⁶⁾. El entorno de trabajo o *Framework* es el componente principal de la tecnología .NET.

1.2.1.1 Características

A continuación se señalan las características más importantes del *Framework .NET*.

- Plataforma de ejecución intermedia *CLR (Common Language Runtime)*, .NET provee un componente de software intermedio que se dispone entre las aplicaciones .NET y el sistema operativo. El *CLR* toma el control de la aplicación en tiempo de ejecución.
- Lenguajes orientados a objetos, todos los lenguajes disponibles en la plataforma .NET son completamente orientados a objetos.

- Multilenguaje, los lenguajes disponibles en *.NET* pueden interactuar entre ellos, pudiéndose utilizar más de uno en el proceso de desarrollo de una misma aplicación.
- Permite la creación de aplicaciones corporativas.
- Provee un único modelo de programación para cualquier aplicación, sin importar si éstas son aplicaciones de escritorio, Web o para móviles.
- Puede interactuar con *COM*⁴ y código no administrado⁵.
- Usa protocolos estándares lo que le permite interactuar con aplicaciones desarrolladas en otras plataformas.

1.2.1.2 Arquitectura del *.NET Framework*

La Figura 1.3 muestra la arquitectura del *Framework .NET*.

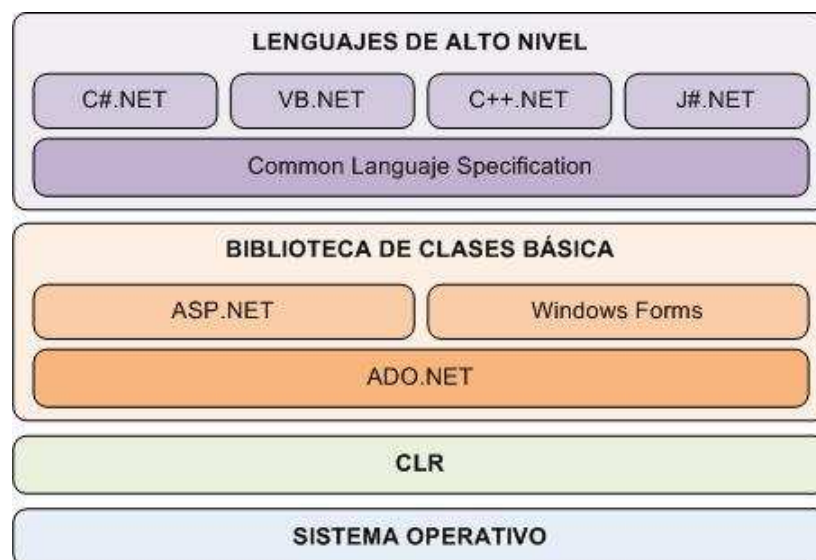


Figura 1.3 Arquitectura *Framework .NET*⁽⁶⁾

El entorno de trabajo o *Framework .NET* cuenta con los siguientes componentes:

- *Common Language Runtime (CLR)*, el entorno de ejecución de *.NET*.

⁴ *COM, Common Object Model.*

⁵ Código no administrado, código que no se ejecuta sobre el *Framework* sino directamente sobre el sistema operativo.

- Biblioteca de clases e interfaces, que pueden ser reutilizadas y extendidas por las aplicaciones.
- Lenguajes de programación de alto nivel, bajo el paradigma de la programación orientada a objetos.

1.2.1.2.1 Entorno de Ejecución Common Language Runtime

Es una capa de software, situada entre el sistema operativo y las aplicaciones .NET. El *Common Language Runtime* brinda todos los servicios requeridos a las aplicaciones en tiempo de ejecución, debido a que las aplicaciones .NET no se ejecutan directamente sobre el sistema operativo sino sobre el entorno de ejecución.

Las principales funciones del *CLR* son:

- Compilación *JIT (Just In Time)*, compila los *assemblies* o ensamblados a código de máquina compatible con la plataforma de hardware; la compilación es dinámica, no se compila todo el ensamblado sino solo lo requerido en tiempo de ejecución.
- *CLR* también se encarga del manejo de memoria, los lenguajes de programación .NET no administran memoria directamente, para esto *CLR* cuenta con un recolector de basura, el cual detecta y libera los espacios de memoria no utilizados por las aplicaciones.
- Gestión de errores consistente, si existen errores no manejados en tiempo de ejecución serán gestionados por el *CLR*; debido a esta característica un error en una aplicación no comprometerá la estabilidad y desempeño del resto de aplicaciones en ejecución.
- Ejecución basada en componentes, *CLR* ejecuta ensamblados reutilizables, los cuales son cargados en memoria en el momento en que se necesiten.
- Gestión de seguridad, “el *CLR* provee una barrera más de contención a la hora de ejecutar aplicaciones manejadas, ya que permite establecer políticas de seguridad muy detalladas, que las aplicaciones .NET que se ejecutan en una determinada computadora deberán cumplir”⁽⁶⁾

- *CLR* admite ejecución con múltiples hilos, a pesar que el sistema operativo no cuente con esta característica.

1.2.1.2.2 Ejecución de una aplicación .NET

Este proceso se ilustra en la Figura 1.4.

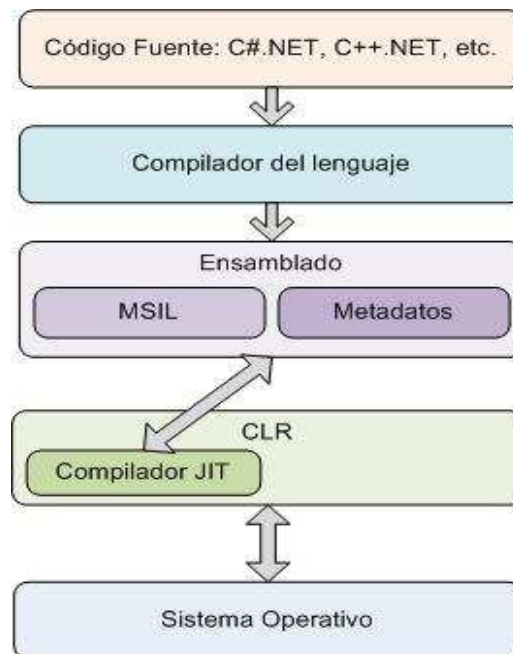


Figura 1.4 Ejecución de una aplicación .NET⁽⁶⁾

La ejecución de una aplicación .NET adopta el siguiente proceso:

1. La aplicación comienza escrita en los lenguajes de alto nivel de .NET como *C#.NET*, *C++.NET* o *VB.NET*.
2. Se compila para obtener un ensamblado o *Assembly*.
3. Este ensamblado contiene código *MSIL* (*Microsoft Intermediate Language*) que almacena la lógica de la aplicación, también contiene metadatos, los mismos que auto describen el ensamblado y los elementos externos necesarios para la ejecución.
4. El *Assembly* es compilado por el *CLR*, su compilador *JIT* se encarga de transformar dinámicamente a código de máquina; es decir, en tiempo de ejecución no se compila todo el ensamblado sino solo lo necesario.

1.2.1.2.3 Biblioteca de clases del .NET Framework

Son un conjunto de clases e interfaces, que envuelven funcionalidades básicas necesarias para el desarrollo de aplicaciones. El *Framework .NET* organiza jerárquicamente estas clases mediante “*Namespaces*”⁶, siendo la raíz el *Namespace System*.

Este conjunto de clases permite al desarrollador el trabajo con datos, formularios *Windows*, formularios Web y servicios Web *XML*. La biblioteca de clases del *.NET Framework* es el súper conjunto de *ADO.NET*, *ASP.NET*, *Windows Forms* y soporte a servicios Web *XML*.

1.2.1.3 Versiones del .NET Framework

El *.NET Framework* posee tres versiones, cada una con un propósito distinto. Las versiones del *.NET Framework* se describen en la Tabla 1.3.

Versión de <i>Framework</i>	Descripción
<i>.NET Framework Redistributable Package</i>	Unidad mínima sobre la cual se ejecutan aplicaciones <i>.NET</i> . Se compone del entorno de ejecución y el conjunto de clases reutilizables. Es utilizado en ambientes de producción.
<i>.NET Framework SDK</i>	Además de los componentes del <i>.NET Framework Redistributable Package</i> , cuenta con herramientas de desarrollo que pueden ser ejecutadas desde la línea de comandos. Es utilizado en ambientes de desarrollo.
<i>.NET Compact Framework</i>	Es una versión reducida al <i>.NET Framework Redistributable Package</i> , diseñada para adaptarse a las limitaciones de los dispositivos móviles como <i>Pocket PCs</i> y <i>Smartphone</i> . Se instala en los dispositivos móviles, sobre la versión móvil de <i>Windows</i> .

Tabla 1.3 Versiones del .NET Framework

⁶ *Namespaces*, es una forma lógica para agrupar a clases e interfaces que guardan similitud de funcionalidad.

1.2.2 .NET COMPACT FRAMEWORK

El *.NET Compact Framework* constituye un subconjunto del *.NET Framework*, “hereda la arquitectura *.NET Framework* completa de *Common Language Runtime* y la ejecución de código administrado”⁽⁷⁾

Diseñado para trabajar con dispositivos tales como *PDA*s, teléfonos móviles, *Handheld PCs*, etc., los mismos que tienen limitaciones de recursos.

Proporciona clases que se utilizan propiamente para el desarrollo de aplicaciones móviles. Permite la ejecución de programas independientes del hardware, la conexión a servicios *Web XML* y optimiza los recursos del sistema.

1.2.2.1 Arquitectura del *.NET Compact Framework*

En la Figura 1.5 se muestran los componentes del *.NET Compact Framework*.

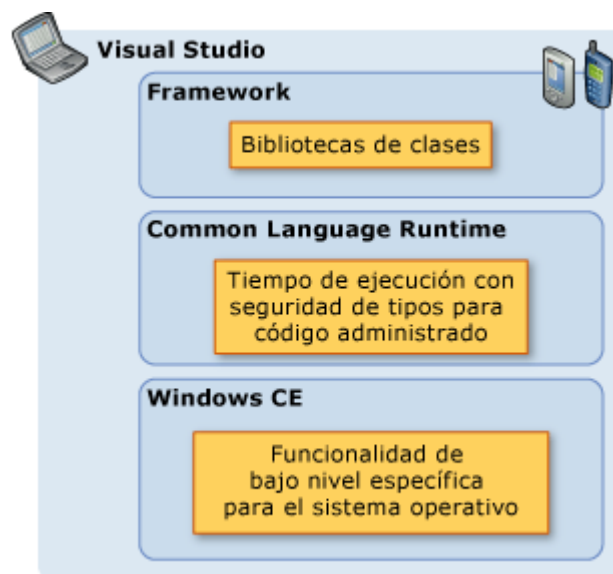


Figura 1.5 Arquitectura del *.NET Compact Framework*⁽⁷⁾

- *Visual Studio*, proporciona un conjunto de emuladores que permiten el desarrollo de aplicaciones para dispositivos móviles inteligentes como *Pocket PC*, *Smartphone* y *Windows CE*.

- *Framework*, constituye un subconjunto del *.NET Framework* y otras características diseñadas exclusivamente para el desarrollo de aplicaciones para dispositivos móviles.
- *Common Language Runtime*, permite la ejecución de aplicaciones en un ambiente con recursos limitados.
- *Windows CE*, sistema operativo que interactúa con *.NET Compact Framework* para permitir la utilización de funciones de hardware del dispositivo.

1.2.2.2 Biblioteca de clases de *.NET Compact Framework*

Es un conjunto de clases básicas, que envuelve funcionalidades habitualmente utilizadas, como: diseño de interfaces, trabajo con orígenes de datos, trabajo con ficheros, *Web services XML*.

Es posible utilizar controles para recoger datos del usuario y controles que permiten mostrar información al usuario. La Tabla 1.4 muestra los controles más importantes.

Los controles descritos no poseen todas las funcionalidades de los controles del *.NET Framework*, debido a las limitaciones de los dispositivos móviles, pero de ser necesario pueden ser implementadas; esto se logra heredando de la clase base de control, pues el *.NET Compact Framework* es flexible y permite la creación y extensión de controles.

La biblioteca de clases permite el trabajo con bases de datos y también con orígenes de datos *XML*, estas funcionalidades están contenidas en los espacios de nombres *System.Data* y *System.Xml*.

También brinda la posibilidad de crear clientes sincrónicos y asincrónicos de servicios *Web XML*. Estas funcionalidades se encuentran en el espacio de nombres *System.Web*.

El *.NET Compact Framework* no puede alojar servicios *Web XML*, es decir, no puede funcionar como servidor de aplicaciones.

Control	Descripción
Button	Botón de comando simple.
CheckBox	Casilla de verificación habitual.
ComboBox	Lista desplegable de elementos.
ContextMenu	Implementa un menú contextual.
DataGrid	Cuadrícula que se puede enlazar a un origen de datos.
DomainUpDown	Lista de elementos que se puede explorar mediante una barra de desplazamiento.
HScrollBar	Barra de desplazamiento horizontal.
ImageList	Contenedor que almacena imágenes.
InputPanel	Controla el <i>Soft Input Panel (SIP)</i> .
Label	Control simple para mostrar texto.
ListBox	Proporciona una lista de elementos.
ListView	Proporciona cuatro vistas de los datos: iconos grandes, iconos pequeños, lista y detalles.
MainMenu	Implementa un menú en un formulario.
NumericUpDown	Campo de entrada numérica que incluye una barra de desplazamiento.
OpenFileDialog	Proporciona acceso al cuadro de diálogo nativo para abrir archivo.
Panel	Contenedor utilizado para acoger otros controles.
PictureBox	Muestra imágenes.
ProgressBar	Indicador visual del progreso de una tarea.
RadioButton	Botón de opción habitual.
SaveFileDialog	Proporciona acceso al cuadro de diálogo nativo para guardar archivo.
StatusBar	Panel simple para mostrar texto.
TabControl	Proporciona una interfaz de fichas para una aplicación.
TextBox	Campo de entrada de texto estándar.
Timer	Componente de temporizador básico.
ToolBar	Implementa una barra de herramientas en un formulario.
TrackBar	Interfaz de control deslizante utilizado con datos numéricos.
TreeView	Presenta datos en formato jerárquico.
VScrollBar	Barra de desplazamiento vertical.

Tabla 1.4 Controles permitidos en el .NET Compact Framework⁽⁷⁾

1.2.3 ADO.NET

Es un conjunto de clases que forman parte de la biblioteca de clases básicas del *Framework.NET*. Estas clases envuelven las funcionalidades necesarias para el trabajo con orígenes de datos.

Permite la conexión con diversas fuentes de datos, *SQL*, *Oracle* y las bases de datos compatibles con los protocolos *Ole DB* y *ODBC*, para esto se sirve de los denominados proveedores de acceso a datos.

ADO.NET provee dos escenarios de conexión a bases de datos:

- Acceso Conectado, en el cual se mantiene una conexión permanente con la base de datos, trabajando sobre ésta todo el tiempo. Para ello *ADO.NET* utiliza la clase *DataReader*.
- Acceso Desconectado, en el cual se crea una copia de los datos localmente, no se tiene una conexión permanente. Se puede trabajar sobre esta copia y sólo se establece una conexión cuando sea necesaria una sincronización con la base de datos. La clase que brinda esta funcionalidad es *DataSet*.

1.2.3.1 Arquitectura ADO.NET

La arquitectura de *ADO.NET* (Figura 1.6) se especifica en el espacio de nombres *System.Data.Common*, donde se definen las funcionalidades que toda implementación de acceso a datos debe tener. Estas funcionalidades están envueltas por cinco clases principales, que son: *Connection*, *Command*, *DataReader*, *DataAdapter* y *DataSet*. Estas clases son derivadas para su especialización con un determinado motor de base de datos, formando lo que en *ADO.NET* se conoce como Proveedores de Datos del *.NET Framework*.

Las funciones que cumplen los objetos de *ADO.NET* son:

“*Connection* proporciona conectividad a un origen de datos.

Command permite tener acceso a comandos de base de datos para devolver datos, modificar datos, ejecutar procedimientos almacenados y enviar o recuperar información sobre parámetros.

DataReader proporciona una secuencia de datos de alto rendimiento desde el origen de datos.

DataAdapter proporciona el puente entre el objeto *DataSet* y el origen de datos. El *DataAdapter* utiliza objetos *Command* para ejecutar comandos SQL en el origen de datos tanto para cargar el *DataSet* con datos como para reconciliar en el origen de datos los cambios aplicados a los datos incluidos en el *DataSet*.⁽⁷⁾

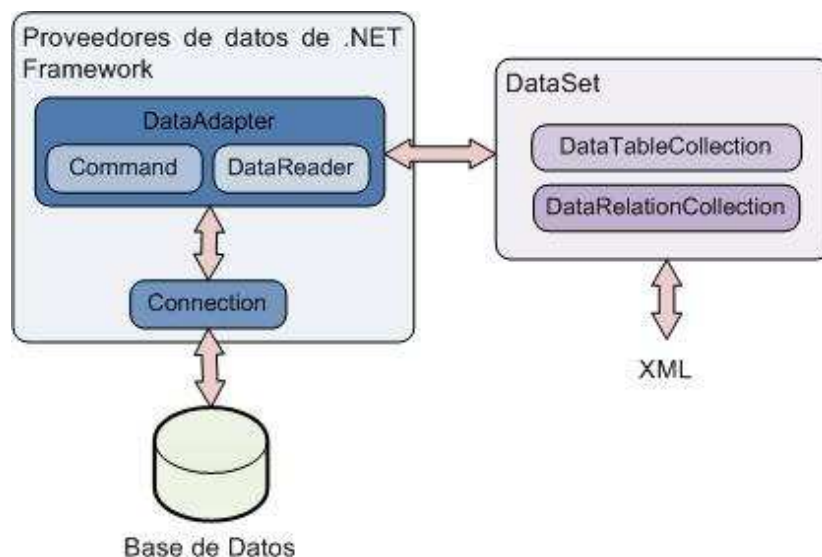


Figura 1.6 Arquitectura ADO.NET⁽⁷⁾

1.2.4 ASP.NET

Es un subconjunto de la Biblioteca de clases básicas del *.NET Framework*; consiste de un grupo de funcionalidades para el desarrollo de aplicaciones Web.

Su componente fundamental son los *WebForms*; éstos tienen la característica principal de separar la interfaz de usuario de las reglas de negocio de la aplicación. En otras palabras, se puede manejar la interfaz de usuario y la parte de la lógica de la aplicación en archivos distintos; aunque también existe la posibilidad de hacerlo en un mismo archivo.

Como características principales se pueden mencionar las siguientes:

- Se compila solo la parte necesaria o requerida por el usuario de la aplicación Web.
- Es independiente del lenguaje de programación.
- Permite la utilización de *ADO.NET* para el acceso a datos.
- Posee un mecanismo de *caching* para páginas que son frecuentemente accedidas.

1.2.4.1 Componentes de una aplicación *ASP.NET*

Una aplicación *ASP.NET* posee los siguientes componentes:

- Formularios Web (*WebForms*), formularios con la extensión *.aspx*; se encuentran escritos en lenguaje de marcas *XML* y representan la interfaz de usuario.
- Archivos *Code-Behind*, contienen código que se ejecuta en el lado del servidor.
- Archivos de configuración con formato *XML*, consiste de dos archivos, el *Web.config*, que contiene información referente a la aplicación, y un archivo *Machine.config*, que incluye información de la configuración del servidor.
- *Global.asax*, maneja los eventos a nivel de aplicación, los cuales contienen las variables de estado de la aplicación (*Application_Start*, *Application_End*, *Session_Start*, *Session_End*). Este archivo es analizado y compilado cuando se solicita una *URL* por primera vez.
- Directorio *BIN*, contiene el ensamblado de la aplicación y los ensamblados de los componentes externos.
- Enlaces a servicios Web *XML* o referencias Web, permite que la aplicación reciba y envíe datos desde y hacia servicios Web.

Estos componentes se ilustran en la Figura 1.7.

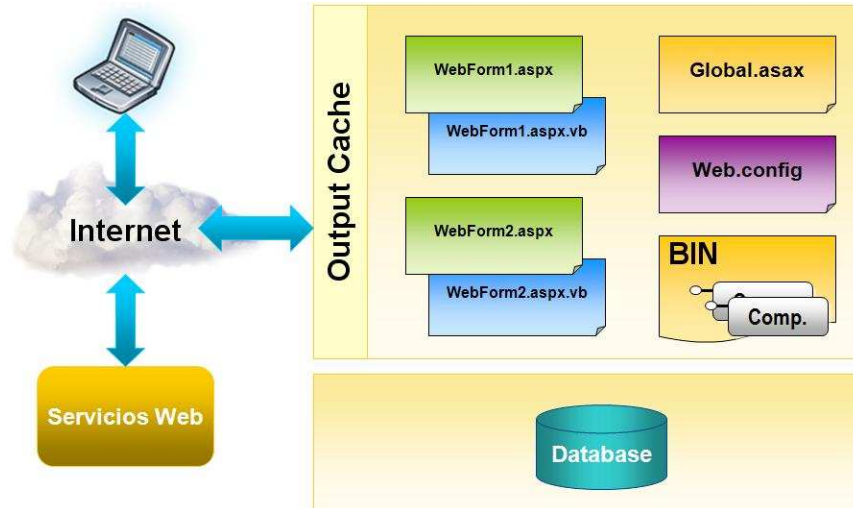


Figura 1.7 Componentes de una aplicación ASP.NET⁽⁸⁾

1.2.4.2 Formularios Web

Páginas escritas en lenguaje de marcas XML, representan la interfaz de usuario. Estas páginas son compiladas y ejecutadas en el servidor.

1.2.4.2.1 Eventos de un Formulario Web

Los eventos de un formulario Web empiezan cuando un cliente hace una petición al servidor, el cual recibe el mensaje, analiza la petición, la atiende con el manejador apropiado y envía la respuesta al cliente. Este procedimiento se ilustra en la Figura 1.8.

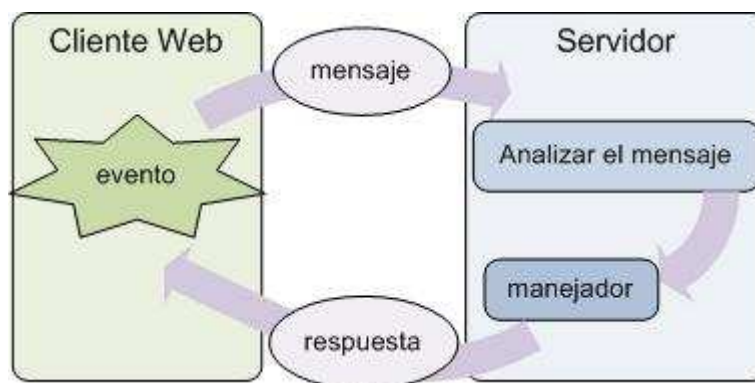


Figura 1.8 Modelo de eventos WebForm⁽⁸⁾

ASP.NET utiliza conceptos como *PostBack* y *ViewState* cuando se trata de manejar un evento. Se conoce como *PostBack* al proceso de enviar la solicitud al servidor y recibir de éste la respuesta; mientras que un *ViewState* es la característica que permite mantener el estado de los controles Web entre *PostBacks*.

1.2.4.2.2 Ciclo de vida de un Formulario Web

Corresponde al tiempo que transcurre entre el requerimiento al servidor Web y la respuesta hacia el cliente.

Se inicia cuando una página es requerida, se ejecuta el código relacionado a la página *.aspx* solicitada, y se crean instancias de los objetos *request*, *response* y los controles de la página.

Los objetos son inicializados y se configura su estado, a través del evento *Init*. Los estados son cargados del *ViewState* y se procesa el *post* inicializando los valores correspondientes en cada objeto. Luego, la página se carga y el cliente puede enviar la solicitud de algún evento, la cual es atendida por su manejador correspondiente.

A continuación el *ViewState* es guardado y se genera el lenguaje de marcas que forma la respuesta. Finalmente el evento *Unload* se ejecuta en cada control y la página es liberada.

1.2.4.3 Autenticación

La autenticación es el proceso que permite determinar si un usuario que está intentando ingresar al sistema es quien dice ser. Esto se logra validando mediante algún mecanismo la información que el usuario ingresa.

ASP.NET proporciona tres tipos de autenticación:

- Autenticación basada en Formularios.
- Autenticación basada en *Windows*.
- Autenticación basada en *Passport*.

1.2.4.3.1 Autenticación basada en Formularios

Este método de autenticación utiliza bases de datos para validar la información que el usuario ingresa. Cuando un cliente quiere acceder a una página que posee seguridades, éste es redireccionado a una página de inicio de sesión, donde debe ingresar su nombre de usuario y contraseña; si la información ingresada es correcta, se genera un *ticket* que posee la identidad del usuario y es utilizado para autenticarse en las siguientes solicitudes. Este *ticket* es almacenado en un *cookie*.

Este método es utilizado en escenarios de Internet o extranet. Permite al usuario cambiar su contraseña, utilizar una pregunta recordatoria en caso que el usuario olvide su contraseña, también es posible solicitar información adicional al usuario, como por ejemplo la cuenta de correo electrónico, nombres, apellidos y dirección.

1.2.4.3.2 Autenticación basada en Windows

Este tipo de autenticación utiliza los servicios de *IIS*⁷ como método para validar el ingreso de usuarios. Es utilizada en aplicaciones de intranet, pues es necesaria la información de las cuentas de usuario presentes en el dominio.

Consiste en verificar si el usuario es válido mediante las cuentas ya creadas en el dominio, de ser así, puede acceder a la página solicitada.

1.2.4.3.3 Autenticación basada en Passport

En este tipo de autenticación se utiliza *Microsoft Passport* como medio para validar el ingreso de usuarios. Proporciona un perfil básico y un único inicio de sesión.

Cuando un usuario intenta acceder a una página con seguridades, primero se analiza las *cookies* para determinar si existe un *ticket* de autenticación de *Passport*; si existe y es válido, el cliente será autenticado, caso contrario, será direccionado a un formulario de inicio de sesión. Una vez que los datos son

⁷ *IIS, Internet Information Server*

ingresados, se envían al servidor *Passport*, en donde el usuario es autenticado y recibe una *cookie* codificada, con la que el usuario puede ingresar a la página solicitada.

1.2.4.4 Autorización

La autorización permite establecer a qué recursos puede ingresar un usuario que ha sido previamente autenticado.

Existen dos maneras de autorización en *ASP.NET*:

- Autorización de archivos, consiste en revisar la lista de control de acceso, para determinar si un usuario tiene acceso al archivo. Debe estar habilitada la autenticación basada en *Windows*.
- Autorización de *URL*, consiste en permitir o denegar el acceso a partes específicas de la aplicación a usuarios o funciones vinculadas a una *URL*.

1.2.5 ASP.NET MOBILE

ASP.NET Mobile es un subconjunto de *ASP.NET*, que permite el desarrollo de aplicaciones Web para dispositivos móviles. Esta tecnología permite que el servidor Web interactúe adecuadamente con una amplia gama de dispositivos móviles como *PDA*s, *Smartphones*, *PocketPC*, teléfonos móviles con navegador, etc.

Una página Web para dispositivos móviles se crea a partir de la clase base *MobilePage* a la que se agrega los controles necesarios para interactuar con el usuario.

1.2.5.1 Controles ASP.NET Mobile

ASP.NET proporciona el espacio de nombres *System.Web.Mobile*, que almacena controles especiales para dispositivos móviles.

En la Tabla 1.5 se describen los controles más importantes.

Control	Descripción
AdRotator	Permite la rotación de anuncios en la página Web.
Calendar	Permite a los usuarios utilizar el calendario para ver y seleccionar fechas.
Command	Permite que los datos de usuario ingresados sean enviados al servidor.
CompareValidator	Realiza la validación por comparación de los datos ingresados.
CustomValidator	Realiza una validación personalizada de los datos ingresados.
Form	Permite la división de páginas complejas que no son móviles en páginas para dispositivos móviles.
Image	Permite la presentación de imágenes en las páginas Web.
Label	Permite la presentación de texto en la página Web.
Link	Proporciona un hipervínculo a otra página.
List	Permite presentar una lista en la página Web.
MobilePage	Clase base para todas las páginas <i>ASP.NET Mobile</i> .
Panel	Ayuda a organizar el contenido de las páginas Web. Organiza los controles en un panel y brinda información de su disposición en tiempo de ejecución.
PhoneCall	Permite que una aplicación realice llamadas.
RangeValidator	Realiza validación de datos que deben estar comprendidos entre un rango especificado.
RegularExpressionValidator	Realiza la validación mediante la comparación con una expresión regular.
RequiredFieldValidator	Verifica si el campo está vacío.
SelectionList	Permite al usuario seleccionar uno o varios elementos de una lista.
StyleSheet	Permite el uso de estilos de formato en una página <i>ASP.NET Mobile</i> .
TextBox	Permite al usuario el ingreso de texto.
TextView	Permite al usuario visualizar textos grandes.
ValidationSummary	Se utiliza para mostrar al usuario una lista de los errores que se hayan producido durante el envío de un formulario al servidor.

Tabla 1.5 Controles *ASP.NET Mobile* ⁽⁷⁾

Los controles de *ASP.NET Mobile* son controles especiales que se ejecutan en el servidor Web, diseñados para responder solicitudes originadas desde navegadores de dispositivos móviles; es decir, se toma en cuenta el formato en que debe ser devuelta la información, tomando en cuenta las limitaciones propias de los dispositivos móviles, como pantallas pequeñas y menor ancho de banda.

1.2.6 SERVICIOS WEB XML

Los servicios Web XML surgieron de la necesidad del intercambio de información entre aplicaciones, pues éstas son desarrolladas con diferentes lenguajes de programación, para diferentes sistemas operativos y corren en diferente hardware.

Para entender qué es servicio Web, se comenzará por definir servicio: “Un servicio es un elemento de software con interfaz independiente de la plataforma que puede ser dinámicamente localizado e invocado” ⁽⁹⁾. Por consiguiente un servicio Web es un estándar para la comunicación entre aplicaciones de plataformas distintas.

Los servicios Web XML utilizan protocolos estándares ampliamente utilizados en Internet, como HTTP y XML, con el objetivo de lograr interoperabilidad, brindando una normalización para la comunicación entre plataformas, lenguajes y sistemas operativos.

Es posible publicar un servicio Web en Internet o en intranet; ya que un servicio Web es flexible y se adapta a distintos requerimientos.

Los dispositivos móviles pueden consumir servicios Web XML directamente, gracias al soporte integrado en el *.NET Compact Framework*.

1.2.6.1 Infraestructura de servicios Web XML

La infraestructura de los servicios Web XML soporta funcionalidades como: mecanismos de descubrimiento, descripción del servicio que define la interfaz y la utilización del servicio y formatos de conexión estándar de comunicación.

La infraestructura se ilustra en la Figura 1.9

- Directorio, es similar a un servicio de *Páginas Amarillas*, permite a los proveedores publicar sus servicios en una localización centralizada de manera que sean fáciles de encontrar. *Universal Description, Discovery, and*

Integration (UDDI) es la especificación que define un medio estándar para publicar y descubrir información de servicios Web.

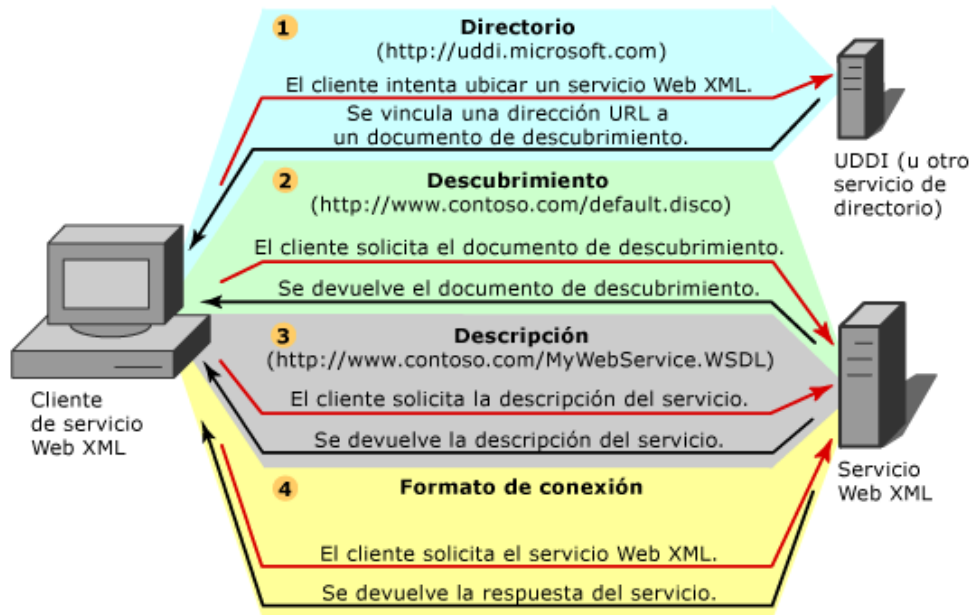


Figura 1.9 Infraestructura de servicios Web XML ⁽⁷⁾

- Descubrimiento, “es el proceso que consiste en localizar, o descubrir, uno o varios documentos relacionados que describen un servicio Web XML” ⁽⁷⁾. Se logra mediante ficheros con extensión *.disco* que guardan enlaces al fichero de descripción del servicio Web. Si la ubicación del fichero de descripción es conocida este paso no es necesario.
- Descripción, define las interacciones permitidas con el servicio. “Para que los clientes entiendan cómo interactuar con el servicio Web, debe existir una descripción de los métodos llamados, o la interfaz que el servicio Web soporta. El documento de descripción es un esquema XML llamado *Web services Description Language (WSDL)*”⁽¹⁰⁾.
- Formato de Conexión, se utilizan tres mecanismos para la comunicación hacia el servicio Web:
 - *HTTP GET*, se realiza a través de un navegador y los parámetros son enviados como cadena de consulta dentro de la URL.
 - *HTTP POST*, se utiliza con un navegador y los parámetros son enviados en el cuerpo del documento hacia el servidor.

- *SOAP*⁸, utiliza *XML* como lenguaje para la creación del mensaje. Permite establecer otras opciones como seguridades, formatos, tipos de datos, etc.

1.2.6.2 Autenticación en servicios Web

Opción de autenticación	Descripción
<i>Windows</i> : básica	Puede utilizarse para la identificación no segura de los clientes, puesto que el nombre de usuario y la contraseña se envían en cadenas de texto sin cifrar codificadas en base 64. En este tipo de autenticación, las contraseñas y los nombres de usuario se codifican pero no se cifran. Un usuario informático con intenciones maliciosas que disponga de una herramienta de supervisión de red puede interceptar los nombres de usuario y las contraseñas.
<i>Windows</i> : básica sobre <i>SSL</i> ⁹	Puede utilizarse para la identificación segura de clientes en escenarios de Internet. El nombre de usuario y la contraseña se envían a través de la red con cifrado <i>SSL</i> (Capa de <i>sockets</i> seguros), en lugar de como texto sin cifrar. Es relativamente sencillo de configurar y eficaz para escenarios de Internet. Sin embargo, el uso de <i>SSL</i> reduce el rendimiento.
<i>Windows</i> : implícita	Puede utilizarse para la identificación segura de clientes en escenarios de Internet. Utiliza <i>Hash</i> para transmitir las credenciales de los clientes de forma cifrada ya que la contraseña no se transmite en texto sin cifrar. Además, la autenticación implícita puede funcionar a través de servidores <i>proxy</i> . Sin embargo, no se admite en muchas plataformas.
<i>Windows</i> : integrada de <i>Windows</i>	Utiliza <i>NTLM</i> o <i>Kerberos</i> . Utiliza un intercambio criptográfico con el explorador Web <i>Microsoft Internet Explorer</i> del usuario.
<i>Windows</i> : certificados de cliente	Puede utilizarse para la identificación segura de clientes en escenarios de Internet y de intranet. Requiere que cada cliente obtenga un certificado de una entidad emisora de certificados de confianza mutua. Opcionalmente, los certificados se asignan a cuentas de usuario, que <i>IIS</i> utiliza para autorizar el acceso al servicio Web.
Formularios	No se admite en los servicios Web. Se trata de un sistema que redirige las solicitudes no autenticadas a un formulario <i>HTML</i> mediante redirección en el cliente <i>HTTP</i> . La mayoría de los clientes de servicios Web prefieren no proporcionar credenciales utilizando una interfaz de usuario; debe resolver este problema si desea utilizar autenticación de formularios.
Encabezados <i>SOAP</i> : personalizada	Puede utilizarse para escenarios de Internet seguros y no seguros. Las credenciales de los usuarios se pasan en el encabezado <i>SOAP</i> del mensaje <i>SOAP</i> . Con independencia de la plataforma que aloje el servicio Web, el servidor Web proporciona una implementación de autenticación personalizada.

Tabla 1.6 Autenticación en servicios Web ⁽⁷⁾

⁸ *SOAP*: Simple Object Access Protocol.

⁹ *SSL*: Secure Socket Layer.

La Tabla 1.6 muestra un resumen de los mecanismos de autenticación hábiles para servicios Web que utilizan *ASP.NET*.

ASP.NET brinda soporte a servicios Web, por tanto los mecanismos de autenticación de *ASP.NET* están disponibles para los servicios Web que utilizan *ASP.NET*.

1.2.6.3 Autorización en servicios Web

Los mecanismos de autorización de acceso a un recurso dado son dos: autorización de direcciones *URL* y autenticación de archivo. Estos mecanismos son los mismos ya señalados para *ASP.NET*.

Capítulo 2

Diseño e implementación del servicio Web

2 DISEÑO E IMPLEMENTACIÓN DEL SERVICIO WEB

2.1 REQUERIMIENTOS

Esta sección empieza indagando requerimientos; para esto se divide el análisis en dos partes: requerimientos del negocio y requerimientos del sistema.

Los requerimientos del negocio explican el proceso que existe en la actualidad. Se describen los procesos que un estudiante realiza para conocer sus calificaciones y horarios en la Carrera de Electrónica y Redes de Información.

Los requerimientos del sistema detallan los requisitos que el sistema a implementar debe cumplir para automatizar la consulta de calificaciones y horarios.

2.1.1 REQUERIMIENTOS DEL NEGOCIO

Automatizar el acceso a información del Sistema de Administración Estudiantil (SAE), que permita consultar calificaciones y horarios a los estudiantes de la Carrera de Electrónica y Redes de Información que dispongan de un dispositivo móvil del tipo *Pocket PC*. La información estará disponible en el área de cobertura de la red local inalámbrica del edificio antiguo de la Facultad del Ingeniería Eléctrica y Electrónica.

Controlar el acceso a la información mediante la autenticación de los estudiantes. Para el proceso de autenticación se utilizará el número de cédula y el número único de los estudiantes como nombre de usuario y contraseña respectivamente.

2.1.1.1 Lista de actores del negocio

Estudiante, persona matriculada en la Carrera de Electrónica y Redes de Información.

Secretaria, persona a cargo de la Secretaría de la Carrera de Electrónica y Redes de Información.

Sistema de Administración Estudiantil (SAE), base de datos que almacena la información de los estudiantes.

2.1.1.2 Lista de casos de uso del negocio

CU1: Publicar horarios, la Secretaria de la Carrera obtiene la información referente a horarios de la Carrera de Electrónica y Redes de Información del SAE, imprime esta información y la publica en la cartelera.

CU2: Consultar horarios, el estudiante se acercan a la cartelera de la Carrera y se informa de sus horarios.

CU3: Consultar calificaciones, el estudiante solicita información sobre sus calificaciones en la Secretaría de la Carrera de Electrónica y Redes de Información.

2.1.1.3 Diagrama de casos de uso del negocio

La lista de casos de uso del negocio se muestra en la Figura 2.1

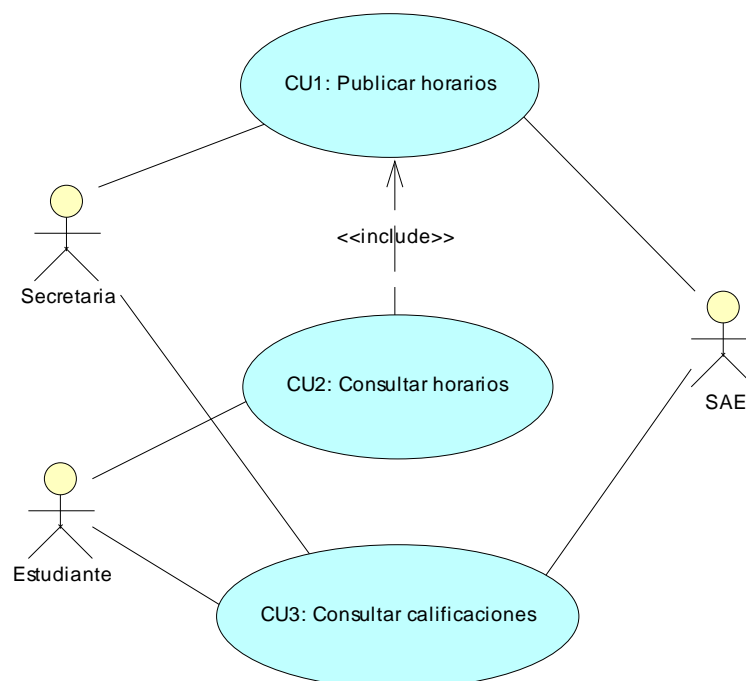


Figura 2.1 Diagrama de casos de uso del negocio

2.1.1.4 Especificación de casos de uso del negocio

La Tabla 2.1 muestra la especificación del caso de uso Publicar horarios.

Caso de Uso	CU1: Publicar horarios.
Actores	Secretaria.
Propósito	Dar a conocer horarios de clase al estudiante.
Resumen	La Secretaria se encarga de consultar la información de horarios en el SAE, esta información es impresa y publicada en la cartelera de la Carrera.
Prioridad	Alta.
Descripción	Permite a la Secretaria de la Carrera publicar información de horarios. Brinda a estudiantes información sobre horas de clase por materia e información de aulas de clase.

Tabla 2.1 Especificación caso de uso del negocio CU1

La Tabla 2.2 muestra la especificación del caso de uso Consultar horarios.

Caso de Uso	CU2: Consultar horarios.
Actores	Estudiante.
Propósito	Informarse de los horarios.
Resumen	El estudiante revisa la cartelera para informarse de horario de clases y aulas.
Prioridad	Media.
Descripción	Permite a los estudiantes conocer los horarios de clases y sus respectivas aulas.

Tabla 2.2 Especificación caso de uso del negocio CU2

La Tabla 2.3 muestra la especificación del caso de uso Consultar calificaciones.

Caso de Uso	CU3: Consultar calificaciones.
Actores	Estudiante, Secretaria, SAE
Propósito	Consultar las calificaciones de un estudiante determinado.
Resumen	El estudiante solicita información de calificaciones a la Secretaria de la Carrera; la Secretaria, consulta la información en el SAE, de haber información en el SAE ésta es comunicada al estudiante.
Prioridad	Alta.
Descripción	Permite a los estudiantes de la Carrera de Electrónica y Redes de Información conocer sus calificaciones académicas.

Tabla 2.3 Especificación caso de uso del negocio CU3

2.1.1.5 Diagrama de actividad de casos de uso del negocio

2.1.1.5.1 CU1: Publicar horarios

El diagrama de actividad del caso de uso Publicar horarios se muestra en la Figura 2.2.

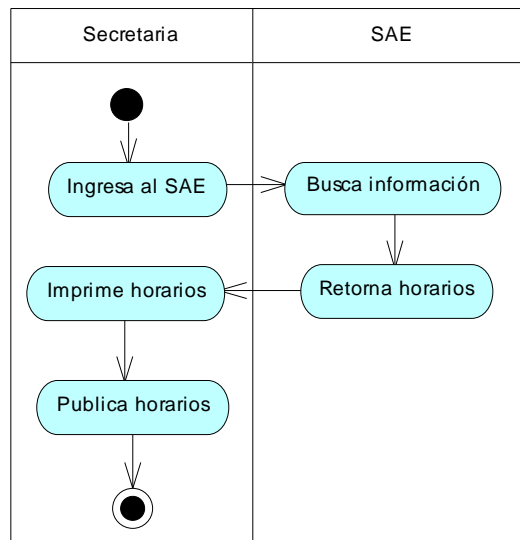


Figura 2.2 Diagrama de actividad del negocio CU1

El diccionario de actividades queda establecido de la siguiente manera:

1. Ingresa al SAE: la Secretaria ingresa al Sistema de Administración Estudiantil.
2. Busca información: el SAE busca los horarios de la Carrera de Electrónica y Redes de Información.
3. Retorna horarios: el SAE retorna los horarios de clase.
4. Imprime horarios: la Secretaria imprime la información obtenida.
5. Publica en cartelera: la Secretaria publica en cartelera los horarios de clase.

2.1.1.5.2 CU2: Consultar horarios

El diagrama de actividad del caso de uso Consultar horarios se muestra en la Figura 2.3.

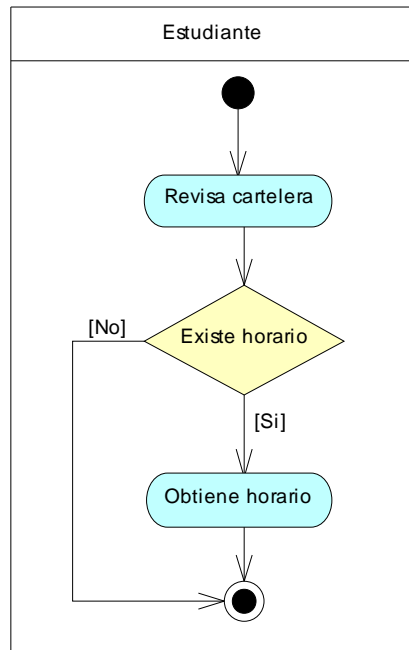


Figura 2.3 Diagrama de actividad del negocio CU2

El diccionario de actividades queda establecido de la siguiente manera:

1. Revisa cartelera: el estudiante se acerca a la cartelera para informarse de sus horarios.
2. Verifica si existe el horario en la cartelera.
3. Obtiene horario: revisa y obtiene su horario.

2.1.1.5.3 CU3: Consultar calificaciones

El diagrama de actividad del caso de uso Consultar calificaciones se muestra en la Figura 2.4.

El diccionario de actividades queda establecido de la siguiente manera:

1. Ingresa a Secretaría: el estudiante se dirige a la Secretaría de la Carrera de Electrónica y Redes de Información.
2. Solicita información de calificaciones: el estudiante solicita a la Secretaria le informe sobre sus calificaciones.
3. Solicita nombres al estudiante: la Secretaria solicita al estudiante sus nombres.

4. Ingresar al SAE: la Secretaria ingresa al sistema de administración estudiantil.
5. El sistema de administración estudiantil busca información del estudiante.
6. El SAE retorna la información obtenida.
7. Verifica existencia de notas: la Secretaria revisa la información obtenida y verifica si existe calificaciones.
8. Informa calificaciones: la Secretaria informa al estudiante sobre las calificaciones existentes.
9. Sale de Secretaría: el estudiante abandona Secretaría.

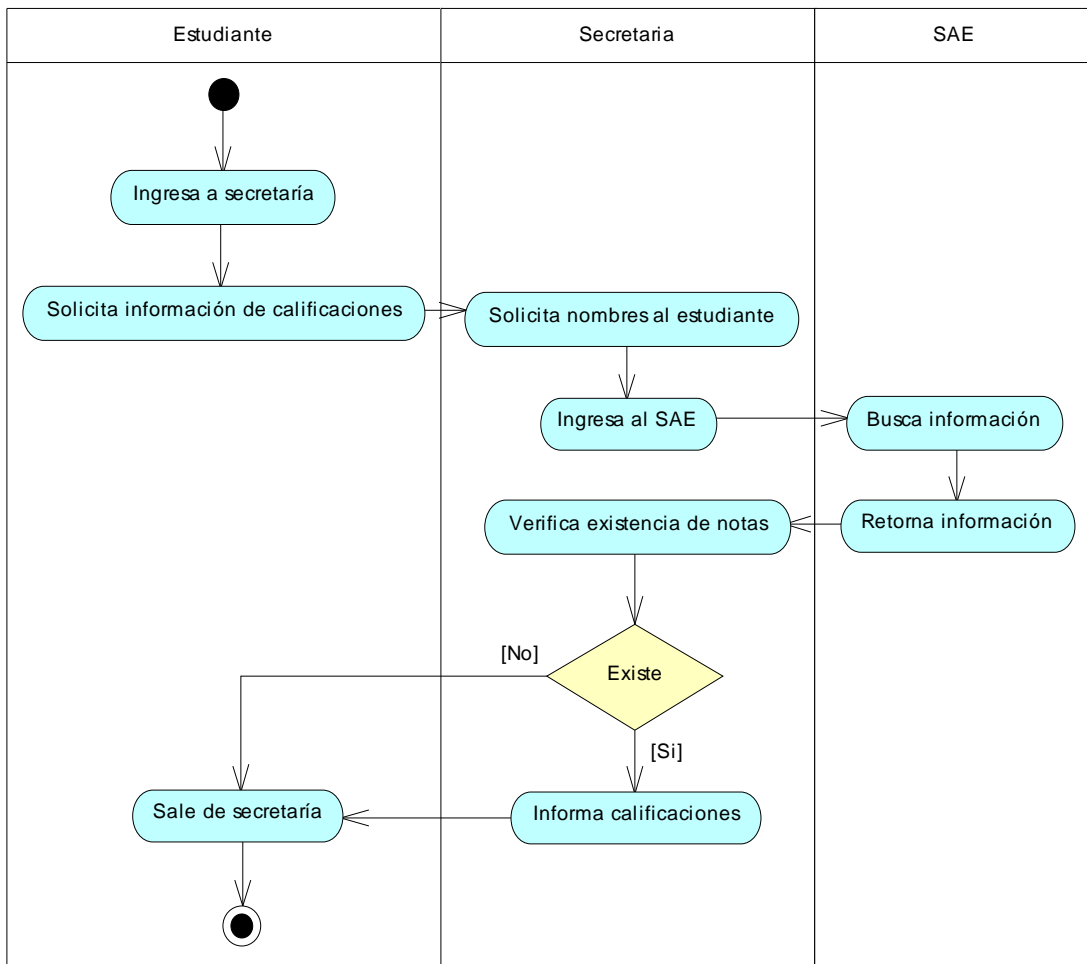


Figura 2.4 Diagrama de actividad del negocio CU3

2.1.2 REQUERIMIENTOS DEL SISTEMA

2.1.2.1 Lista de actores del sistema

Aplicación *OnlineSIEM*: aplicación en línea que se conecta al servicio Web.

Aplicación *SmartSIEM*: aplicación del tipo *Smart Client* que se conecta al servicio Web.

Sistema de Administración Estudiantil (SAE): base de datos que almacena la información de estudiantes y profesores.

Base de Datos *BDDSIEM*: base de datos que almacena el número de cédula y el número único de los estudiantes, y es utilizada para la autenticación.

2.1.2.2 Lista de casos de uso del sistema

- CU1: Iniciar sesión, el estudiante inicia una sesión en el *Sistema de Información Estudiantil Móvil (SIEM)*, para esto utiliza su número de cédula y su número único.
- CU2: Consultar información de cuenta, el estudiante puede revisar la información de su sesión.
- CU3: Consultar horario, el estudiante consulta en *SIEM* el horario de clases.
- CU4: Consultar calificaciones, el estudiante consulta en *SIEM* sus calificaciones académicas.
- CU5: Consultar información de materia, se obtiene información como nombre, número de créditos, etc. de las materias tomadas por el estudiante.
- CU6: Cerrar sesión, el estudiante termina la sesión con *SIEM*.

2.1.2.3 Diagrama de casos de uso del sistema

La Figura 2.5 muestra el diagrama de casos de uso para el servicio Web *SIEM*.

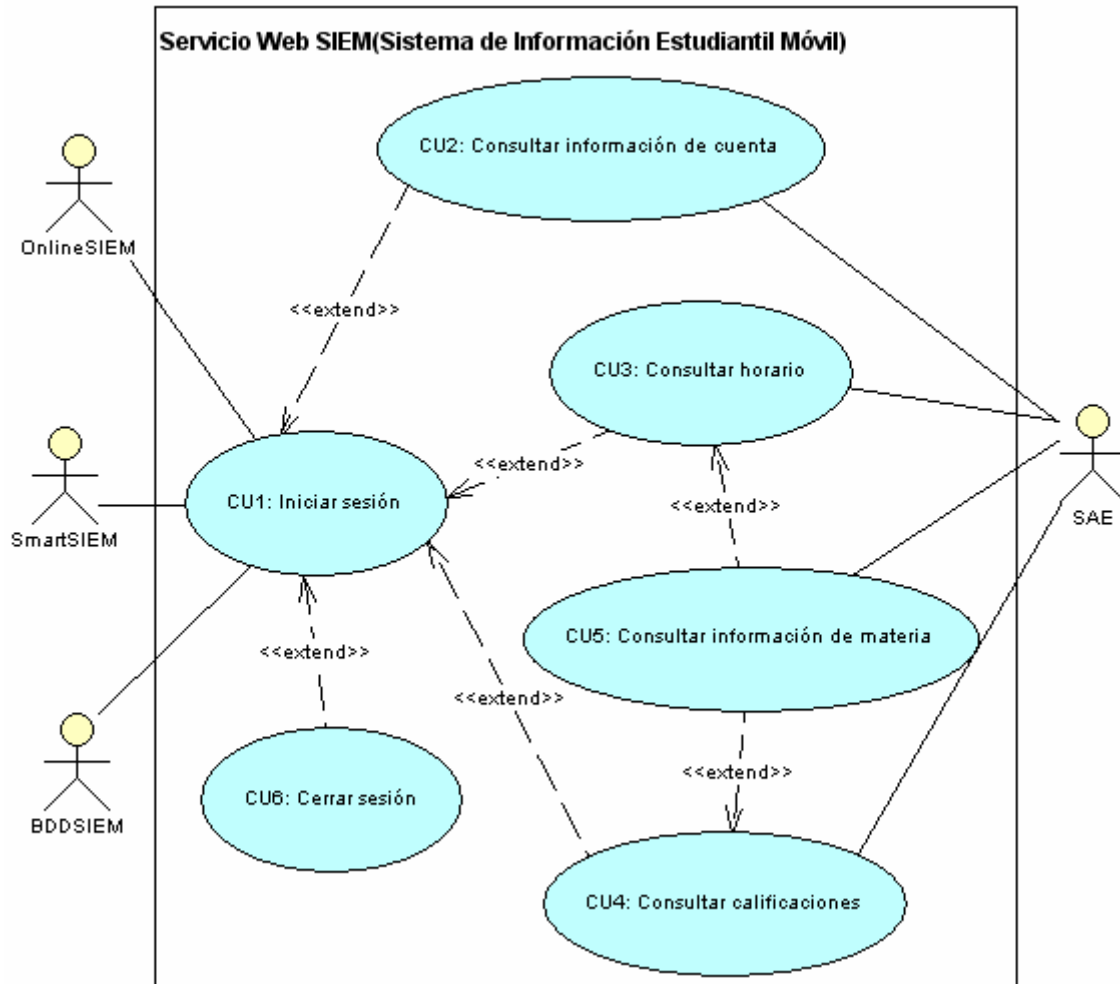


Figura 2.5 Diagrama de casos de uso del sistema

2.1.2.4 Especificación de casos de uso del sistema

La funcionalidad de cada uno de los casos del sistema es presentada desde la Tabla 2.4 hasta la Tabla 2.9.

Caso de Uso	CU1: Inicio de sesión.
Propósito	Para ingresar al sistema, se debe iniciar una sesión proporcionado la cédula de identidad y número único de un determinado estudiante.
Precondiciones	Ninguna.
Postcondiciones	Con la sesión establecida se puede solicitar información sobre calificaciones y horarios.
Actores	Aplicación <i>OnlineSIEM</i> , Petición de inicio de sesión. Aplicación <i>SmartSIEM</i> , Petición de inicio de sesión. Base de datos <i>SIEM (BDDSIEM)</i> , permite la autenticación, mediante el número de cédula y el número único de un estudiante.
Entradas	Cédula de identidad y número único.
Salidas	Identificador de sesión.
Proceso/Evento	Descripción
Las aplicaciones <i>OnlineSIEM</i> y/o <i>SmartSIEM</i> solicitan inicio de sesión.	El servicio Web, verifica con la base de datos <i>BDDSIEM</i> si el número de cédula y el número único son correctos y se asigna un <i>ticket</i> de autenticación para mantener la sesión activa.

Tabla 2.4 Especificación del caso de uso Inicio de sesión

Caso de Uso	CU2: Consultar información de cuenta.
Propósito	Una vez iniciada la sesión en el <i>SIEM</i> , las aplicaciones <i>OnlineSIEM</i> y <i>SmartSIEM</i> podrán obtener datos del estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Aplicación <i>OnlineSIEM</i> , solicita información de cuenta. Aplicación <i>SmartSIEM</i> , solicita información de cuenta. SAE, proporciona la información requerida por el <i>SIEM</i> .
Entradas	<i>Ticket</i> de autenticación.
Salidas	Los siguientes datos del estudiante: Nombre Número de cédula Número único Identificación militar Correo electrónico
Proceso/Evento	Descripción
Las aplicaciones <i>OnlineSIEM</i> y/o <i>SmartSIEM</i> solicitan información de cuenta.	El servidor <i>SWSIEM</i> verifica si el <i>ticket</i> de autenticación es válido y consulta en la base de datos del SAE los datos del estudiante.

Tabla 2.5 Especificación del caso de uso Consultar información de cuenta

Caso de Uso	CU3: Consultar horario.
Propósito	Una vez iniciada la sesión en el <i>SIEM</i> , las aplicaciones <i>OnlineSIEM</i> y <i>SmartSIEM</i> podrán obtener el horario de clases de un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Aplicación <i>OnlineSIEM</i> , solicita información de horario. Aplicación <i>SmartSIEM</i> , solicita información de horario. SAE, proporciona la información requerida por el <i>SIEM</i> .
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Horario de las materias tomadas con el siguiente detalle: Día, nombre de la materia, hora inicio, hora fin.
Proceso/Evento	Descripción
Las aplicaciones <i>OnlineSIEM</i> y/o <i>SmartSIEM</i> solicitan información horario académico.	El servidor <i>SWSIEM</i> verifica si el <i>ticket</i> de autenticación es válido y consulta en la base de datos del SAE el horario de clases del estudiante.

Tabla 2.6 Especificación del caso de uso Consultar horario

Caso de Uso	CU4: Consultar calificaciones.
Propósito	Una vez iniciada la sesión en el <i>SIEM</i> , las aplicaciones <i>OnlineSIEM</i> y <i>SmartSIEM</i> podrán obtener datos del estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Aplicación <i>OnlineSIEM</i> , solicita información de calificaciones. Aplicación <i>SmartSIEM</i> , solicita información de calificaciones. SAE, proporciona la información requerida por el <i>SIEM</i> .
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Las calificaciones del estudiante con el siguiente detalle: Nombre de la materia, calificación primer bimestre, calificación segundo bimestre, calificación supletorio, suma, aprobación.
Proceso/Evento	Descripción
Las aplicaciones <i>OnlineSIEM</i> y/o <i>SmartSIEM</i> solicitan información sobre calificaciones.	El servidor <i>SWSIEM</i> verifica si el <i>ticket</i> de autenticación es válido y consulta en la base de datos del SAE las calificaciones del estudiante.

Tabla 2.7 Especificación del caso de uso Consultar calificaciones

Caso de Uso	CU5: Consultar información de materia.
Propósito	Conocer el detalle de una materia.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Aplicación <i>OnlineSIEM</i> , solicita información de materia. Aplicación <i>SmartSIEM</i> , solicita información de materia. SAE, proporciona la información requerida por el <i>SIEM</i> .
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Las materias tomadas por el estudiante con el siguiente detalle: Código de la materia Nombre Aula Profesor de la materia Número de créditos Paralelo Categoría.
Proceso/Evento	Descripción
Las aplicaciones <i>OnlineSIEM</i> y/o <i>SmartSIEM</i> solicitan información detallada de materia.	El servidor <i>SWSIEM</i> verifica si el <i>ticket</i> de autenticación es válido y consulta en la base de datos del SAE el detalle de una materia.

Tabla 2.8 Especificación del caso de uso Consultar información de materias

Caso de Uso	CU6: Cerrar sesión.
Propósito	Terminar la sesión iniciada en el <i>SIEM</i> .
Precondiciones	La sesión debe estar iniciada.
Postcondiciones	Ninguna.
Actores	Aplicación <i>OnlineSIEM</i> , solicita finalizar la sesión. Aplicación <i>SmartSIEM</i> , solicita finalizar la sesión.
Entradas	Ninguna.
Salidas	Ninguna.
Proceso/Evento	Descripción
Las aplicaciones <i>OnlineSIEM</i> y/o <i>SmartSIEM</i> solicitan cerrar la sesión.	El <i>ticket</i> de autenticación de la sesión es borrado.

Tabla 2.9 Especificación del caso de uso Cerrar sesión

2.1.2.5 Diseño de Interfaces

En el caso de un servicio Web, la interfaz se refiere a los métodos que utilizan los clientes para obtener la información deseada.

Para cumplir con todos los requerimientos, se utiliza el siguiente grupo de métodos:

- `public string IniciarSesion(string cedulaIdentidad, string numeroUnico)`

Este método es utilizado en el proceso de inicio de sesión, se requiere que se provea el número de cédula y el número único de un estudiante, ninguno de los dos campos puede ser vacío.

- `public Estudiante ObtenerEstudiante()`

Método utilizado para obtener la información del estudiante, previamente el proceso de inicio de sesión debe ser exitoso.

- `public List<HoraClase> ObtenerHorario()`

Mediante este método se obtiene el horario de un estudiante, para ello es necesario que el inicio de sesión haya sido exitoso.

- `public List<Calificacion> ObtenerCalificaciones()`

Permite obtener las calificaciones de un estudiante, para ello también es necesario que el inicio de sesión haya sido exitoso.

- `public DetalleMateria ObtenerDetalleMateria(string codigoMateria)`

Permite obtener la información detallada de una materia, es necesario conocer el código de la materia y tener un inicio de sesión exitoso.

2.2 DISEÑO

Luego de tener conocimiento de los requerimientos del sistema, el siguiente paso es el diseño del servicio Web.

Para el diseño del servicio, se consideró necesario dividir el proceso en tres etapas.

La primera etapa trata el tema de bases de datos, la segunda etapa consiste en el diseño de la capa de acceso a datos y la tercera en el diseño de la capa de reglas del negocio.

2.2.1 BASE DE DATOS

La información requerida para el sistema *SIEM*, se obtendrá de una copia de la base de datos del SAE. A esta base de datos no se le realizarán cambios en su estructura de tablas, pero se le agregaron procedimientos almacenados para consultar la información necesaria.

Para cumplir el requerimiento de autenticación, se utiliza una nueva base de datos que almacena la información para este proceso.

2.2.1.1 Base de Datos SAE

Como se mencionó anteriormente, para obtener la información demandada por el sistema, se utilizará una copia de la base de datos del SAE. Esta base de datos guarda la información de horarios y calificaciones de los estudiantes.

No es necesario el uso de todas las tablas del SAE; en la Figura 1.1 se muestra el modelo relacional de base de datos con las tablas utilizadas.

2.2.1.1.1 Tablas utilizadas del SAE

La Tabla 2.10, indica las tablas del SAE empleadas en el *SIEM*. También se hace una breve descripción de la información que cada una de ellas proporciona.

Cada una de las tablas descritas contiene varios campos; al igual que en las tablas, no todos los campos son requeridos para el desarrollo del sistema *SIEM*.

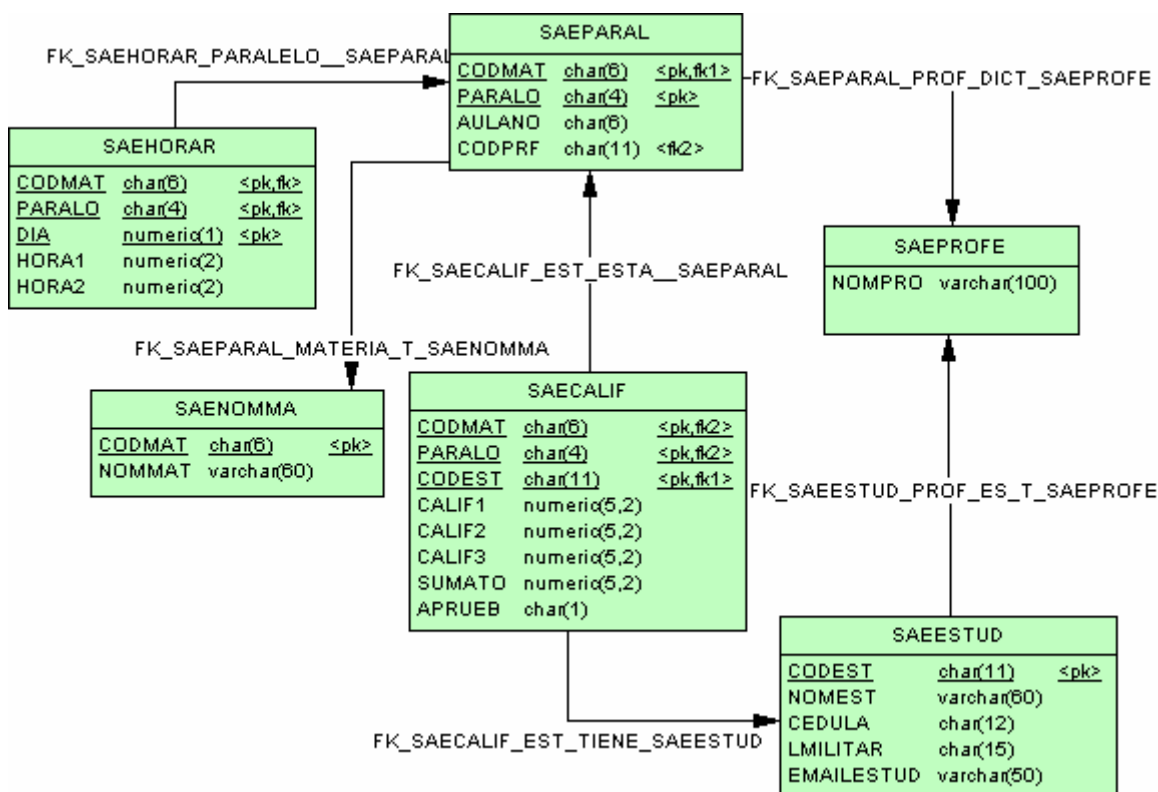


Figura 2.6 Diagrama relacional de las tablas del SAE utilizadas

Tabla	Descripción
SAEESTUD	Almacena toda la información relacionada con el estudiante.
SAENOMMA	Almacena información referente a las materias.
SAEHORAR	Almacena información acerca de los días y horas en que las materias son dictadas.
SAEPARAL	Unión lógica entre una materia, un profesor y estudiantes.
SAECALIF	Almacena las calificaciones de las materias tomadas por el estudiante.
SAEPROFE	Contiene la información correspondiente al profesor.

Tabla 2.10 Descripción de las tablas utilizadas en el sistema *SIEM*

2.2.1.1.2 Campos utilizados de la tabla SAEESTUD

En la Tabla 2.11 se da una breve descripción de los campos de la tabla SAEESTUD que proveen la información para el sistema.

Campo	Descripción
CODEST	Campo utilizado como clave primaria, se almacena el número único de cada estudiante.
NOMEST	Se almacena el nombre completo del estudiante.
CEDULA	Contiene el número de la cédula de identidad de cada estudiante.
LMILITAR	Contiene el número de la identificación militar de los estudiantes.
EMAILESTUD	Almacena el correo electrónico de los estudiantes.

Tabla 2.11 Descripción de los campos de la tabla SAEESTUD

2.2.1.1.3 Campos utilizados de la tabla SAENOMMA

En la Tabla 2.12 se da una breve descripción de los campos de la tabla SAENOMMA que proveen la información para el sistema.

Campo	Descripción
CODMAT	Campo utilizado como clave primaria, que almacena el código de la materia
NOMMAT	Almacena el nombre de la materia.

Tabla 2.12 Descripción de los campos de la tabla SAENOMMA

2.2.1.1.4 Campos utilizados de la tabla SAEHORAR

En la Tabla 2.13 se da una breve descripción de los campos de la tabla SAEHORAR que proveen la información para el sistema.

Campo	Descripción
DIA	Almacena los días de la semana.
HORA1	Campo correspondiente al inicio de un periodo de clases de una materia.
HORA2	Campo correspondiente a la terminación de un periodo de clases de una materia.

Tabla 2.13 Descripción de los campos de la tabla SAEHORAR

2.2.1.1.5 Campos utilizados de la tabla SAEPARAL

En la Tabla 2.14 se da una breve descripción del campo de la tabla SAEPARAL que provee la información para el sistema.

Campo	Descripción
AULANO	Contiene la identificación del aula.

Tabla 2.14 Descripción de los campos de la tabla SAEPARAL

2.2.1.1.6 Campos utilizados de la tabla SAECALIF

En la Tabla 2.15 se da una breve descripción de los campos de la tabla SAECALIF que proveen la información para el sistema.

Campo	Descripción
CALIF1	Se almacena la nota correspondiente al primer bimestre.
CALIF2	Se almacena la nota correspondiente al segundo bimestre.
CALIF3	Se almacena la nota correspondiente al supletorio.
SUMATO	Almacena la suma de todas las calificaciones.
APRUEB	Almacena la forma de aprobación de la materias, donde se tienen las siguientes opciones: A(Aprobado), E(Exonerado), F(Fallido), X(eXonerado certificado)

Tabla 2.15 Descripción de los campos de la tabla SAECALIF

2.2.1.1.7 Campos utilizados de la tabla SAEPROFE

En la Tabla 2.16 se da una breve descripción del campo de la tabla SAEPROFE que provee la información para el sistema.

Campo	Descripción
NOMPRO	Contiene el nombre completo de los profesores

Tabla 2.16 Descripción de los campos de la tabla SAEPROFE

2.2.1.1.8 Procedimientos Almacenados

Una vez analizadas las tablas y los campos de datos necesarios, lo siguiente es agregar los procedimientos almacenados a la base de datos del SAE, para recuperar la información de horarios y calificaciones de los estudiantes.

La Tabla 2.17 detalla los procedimientos agregados a la base de datos del SAE:

Procedimiento Almacenado	Descripción
ObtenerInformacionEstudiante	Este procedimiento permitirá obtener de la base de datos la información del estudiante como es su nombre, número único, número de cédula, correo electrónico y libreta militar.
ObtenerHorario	Por medio de este procedimiento, se obtendrá el horario de clases de un estudiante.
ObtenerCalificaciones	Este procedimiento almacenado proporcionará las calificaciones obtenidas por un estudiante.
ObtenerDetalleMateria	Mediante este procedimiento almacenado se podrá obtener la información referente a una materia.

Tabla 2.17 Procedimientos almacenados incluidos en la base de datos SAE

Una explicación detallada del código de cada uno de los procedimientos almacenados se encuentra en el anexo B-1.

2.2.1.2 Base de Datos SIEM

Para realizar el proceso de autenticación, se creó de una nueva base de datos denominada *BDDSIEM*; esta base de datos tiene la información suficiente para autenticar a los usuarios del sistema *SIEM*.

La Figura 2.7 muestra el diagrama relacional de la base de datos *BDDSIEM*.

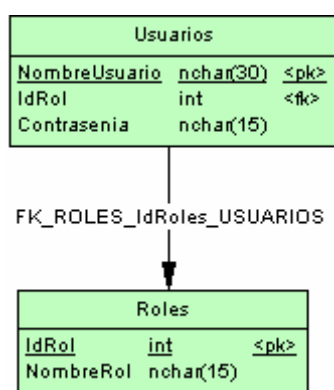


Figura 2.7 Diagrama de relaciones de la Base de Datos SIEM

2.2.1.2.1 Tablas de la Base de Datos SIEM

La base de datos cuenta con dos tablas, Usuarios y Roles:

- La tabla Usuarios permitirá almacenar la cédula y número único de los estudiantes.
- La tabla Roles permitirá almacenar roles que tienen los usuarios.

2.2.1.2.2 Procedimientos Almacenados

Para obtener los datos que ayudarán en el proceso de autenticación, se elaboraron los procedimientos almacenados que se describen en la Tabla 2.18.

Procedimiento Almacenado	Descripción
ObtenerContrasenia	Procedimiento almacenado por medio del cual se obtendrá el número único de un estudiante.
ObtenerRol	Permite determinar el rol de un determinado usuario.

Tabla 2.18 Procedimientos almacenados de la base de datos SIEM

2.2.2 CAPA DE ACCESO A DATOS

La capa de acceso a datos controla la obtención y manejo de datos. Permite acceder a información de horarios y calificaciones del SAE, y a los datos necesarios para el proceso de autenticación de la base de datos *BDDSIEM*.

La capa de acceso a datos cumple las siguientes funciones:

- Establecer conexión con las bases de datos: se encarga de crear las conexiones con las bases de datos SAE y *BDDSIEM*.
- Consultar información de las bases de datos: realiza las operaciones de consulta de información del estudiante, horarios, calificaciones, materias y datos de autenticación.
- Transformar la información consultada en objetos de negocios: la información consultada es utilizada para crear objetos de negocio.
- Cerrar la conexión con la base de datos: una vez realizadas las consultas, las conexiones con las bases de datos son cerradas.

2.2.2.1 Diagrama de clases

La Figura 2.8 muestra la clase que corresponde a la capa de accesos a datos.

AccesoDatosSIEM	
+ ObtenerEstudiante (string numeroUnico)	: Estudiante
+ ObtenerCalificaciones (string numeroUnico)	: List<Calificacion>
+ ObtenerHorario (string numeroUnico)	: List<HoraClase>
+ ObtenerDetalleMateria (string codigoMateria)	: DetalleMateria
+ VerificarCredenciales (string ceduladentidad, string numeroUnico)	: bool
+ ObtenerRoles (string ceduladentidad)	: string

Figura 2.8 Clase AccesoDatosSIEM

La capa de acceso permite recuperar información de las bases de datos SAE y *BDDSIEM*; esta capa expone los métodos descritos en la Tabla 2.19

Método	Descripción
ObtenerEstudiante(string numeroUnico):Estudiante	Permite la recuperación de datos de un estudiante mediante el ingreso de su número único.
ObtenerCalificaciones(string numeroUnico):List<Calificacion>	Se obtiene de la base de datos las calificaciones del estudiante; esta información se la recupera por medio del número único.
ObtenerHorario(string numeroUnico):List<HoraClase>	Se obtiene de la base de datos el horario de clases de un estudiante; esta información se la recupera por medio del número único.
ObtenerDetalleMateria(string codigoMateria):DetalleMateria	Se obtiene de la base de datos información referente a una materia; esta información se la recupera por medio del código de la materia.
VerificarCredenciales(string ceduladentidad, string numeroUnico):bool	Este método permite verificar que las credenciales de autenticación son correctas.
ObtenerRoles(string ceduladentidad):string	Se obtiene el rol, para cada usuario.

Tabla 2.19 Descripción de los métodos de la clase AccesoDatosSIEM

2.2.3 CAPA REGLAS DEL NEGOCIO

La capa de reglas del negocio gestiona la lógica de la aplicación. La capa de acceso a datos provee información que es utilizada para crear objetos de la capa de reglas del negocio; estos objetos agrupan de forma lógica la información obtenida de las bases de datos. También esta capa gestiona la lógica de autenticación.

2.2.3.1 Diagrama de clases

La Figura 2.9 muestra las clases de la capa de reglas del negocio.

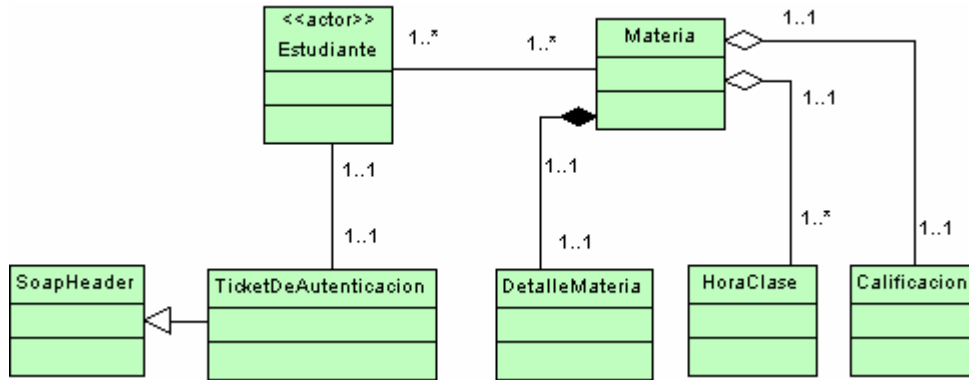


Figura 2.9 Diagrama de clases de la capa de reglas de negocio

Este diagrama expone la siguiente lógica:

Un estudiante toma varias materias, así también una materia puede ser tomada por varios estudiantes.

Una materia tiene:

- Un detalle materia que contiene información detallada de una materia.
- Una calificación que lleva información de las notas correspondientes a un semestre.
- Una o varias horas clase que representan el horario semanal de la materia.

Por otro lado se tiene que un estudiante tiene un *ticket* de autenticación y viceversa, este *ticket* es utilizado para el proceso de autenticación. Como el *ticket* es enviado en el encabezado *SOAP* de la solicitud al servicio Web, la clase *TicketDeAutenticacion* se deriva de la clase *SoapHeader*.

En el anexo B-2 se detalla la implementación de autenticación en el servicio Web.

2.2.3.2 Atributos

A continuación se da una breve descripción de los atributos de cada clase de la capa de reglas del negocio.

- Clase *Estudiante*, posee los atributos que se muestran en la Figura 2.10. En la Tabla 2.20 se da una breve descripción de cada atributo.

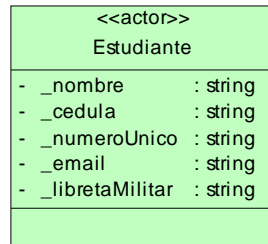


Figura 2.10 Atributos de la clase *Estudiante*

Nombre	Descripción
_nombre	Nombre completo de estudiante.
_numeroCedula	Cédula de identidad del estudiante.
_numeroUnico	Número único del estudiante.
_email	Correo electrónico del estudiante.
_libretaMilitar	Identificación militar del estudiante.

Tabla 2.20 Atributos de la clase *Estudiante*

- Clase *TicketDeAutenticacion*, posee los atributos que se describen en la Tabla 2.21.

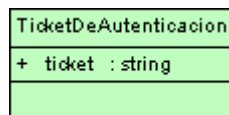


Figura 2.11 Atributos de la clase *TicketDeAutenticacion*

Nombre	Descripción
<i>ticket</i>	Guarda información encriptada de una sesión ya iniciada.

Tabla 2.21 Descripción de los atributos de la clase *TicketDeAutenticacion*

- Clase *Materia*, posee los atributos descritos en la Tabla 2.22

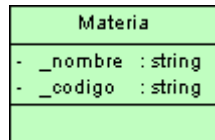


Figura 2.12 Atributos de la clase *Materia*

Nombre	Descripción
_nombre	Nombre de la materia.
_codigo	Código de la materia.

Tabla 2.22 Descripción de los atributos de la clase *Materia*

- Clase *DetalleMateria*, esta clase consta de los atributos descritos en la Tabla 2.23

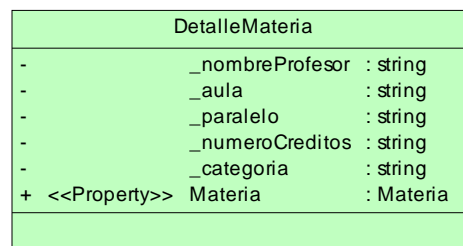


Figura 2.13 Atributos de la clase *DetalleMateria*

Nombre	Descripción
_nombreProfesor	Nombre del profesor.
_aula	Nombre y número del aula de clases.
_paralelo	Nombre del paralelo.
_numeroCreditos	Número de créditos correspondientes a la materia.
_categoria	Indica la categoría a la que pertenece la materia.
Materia	Nombre y código de la materia.

Tabla 2.23 Descripción de los atributos de la clase *DetalleMateria*

- Clase *HoraClase*, esta clase posee los atributos descritos en la Tabla 2.24.

HoraClase		
#	_dia	: string
#	_horainicio	: string
#	_horaFin	: string
+	<<Property>> Materia	: Materia

Figura 2.14 Atributos de la clase *HoraClase*

Nombre	Descripción
_dia	Día de la semana en que es dictada una materia.
_horainicio	Hora de inicio del periodo de clases de una materia.
_horaFin	Hora de finalización del periodo de clases de una materia.
Materia	Nombre y código de la materia.

Tabla 2.24 Descripción de los atributos de la clase *HoraClase*

- Clase *Calificacion*, se encuentra conformada por los atributos definidos en la Tabla 2.25.

Calificacion		
-	_primerBimestre	: string
-	_segundoBimestre	: string
-	_supletorio	: string
-	_sumatoria	: string
-	_aprobacion	: string
+	<<Property>> Materia	: Materia

Figura 2.15 Atributos de la clase *Calificacion*

Nombre	Descripción
_primerBimestre	Calificación correspondiente al primer bimestre.
_segundoBimestre	Calificación correspondiente al segundo bimestre.
_supletorio	Calificación correspondiente al examen supletorio.
_sumatoria	Promedio de las notas anteriores.
_aprobacion	Forma de aprobación de la materia.
Materia	Nombre y código de la materia.

Tabla 2.25 Descripción de los atributos de la clase *Calificacion*

2.2.3.3 Métodos

Se da una breve descripción de los métodos utilizados en las clases de la capa de reglas del negocio. Esta descripción se muestra desde la Tabla 2.26 a la Tabla 2.30.

- Clase *Estudiante*

Nombre	Descripción
get_Nombre()	Permite obtener el nombre de un estudiante.
Set_Nombre(string value)	Permite establecer el nombre de un estudiante.
Get_Cedula()	Permite obtener la cédula de identidad de un estudiante.
Set_Cedula(string value)	Permite establecer el valor de la cédula de identidad de un estudiante.
Get_NumeroUnico()	Permite obtener el número único correspondiente a un estudiante.
Set_NumeroUnico (string value)	Permite establecer número único de un estudiante.
Get_email()	Permite obtener el correo electrónico de un estudiante.
Set_email (string value)	Permite establecer el valor del correo electrónico de un estudiante.
Get_LibretaMilitar()	Permite obtener la identificación militar de un estudiante.
Set_LibretaMilitar (string value)	Permite establecer el valor del campo libreta militar.

Tabla 2.26 Descripción de las propiedades de la clase *Estudiante*

- Clase *Materia*

Nombre	Descripción
get_Nombre()	Permite obtener el nombre de una materia.
Set_Nombre(string value)	Permite establecer el valor del campo nombre de la materia.
Get_Código()	Permite obtener el código de la materia.
Set_Código(string value)	Permite establecer el valor del código de la materia.

Tabla 2.27 Descripción de las propiedades de la clase *Materia*

- Clase *DetalleMateria*

Nombre	Descripción
get_Aula()	Permite obtener el identificador del aula donde es dictada una materia.
Set_Aula(string value)	Permite establecer el aula donde es dictada una materia.
Get_Categoria()	Permite obtener la categoría a la que pertenece una materia.
Set_Categoria(string value)	Permite establecer la categoría de una determinada materia.
Get_NumeroCreditos()	Permite obtener el número de créditos correspondiente a una materia.
Set_NumeroCreditos(string value)	Permite establecer el número de créditos correspondiente a una materia.
Get_Profesor()	Permite obtener el nombre del profesor que dicta una materia.
Set_Profesor(string value)	Permite establecer el nombre del profesor que dicta una materia.
Get_Paralelo()	Permite obtener el paralelo correspondiente a una materia.
Set_Paralelo(string value)	Permite establecer el paralelo correspondiente a una materia.

Tabla 2.28 Descripción de las propiedades de la clase *DetalleMateria*

- Clase *HoraClase*

Nombre	Descripción
Get_Materia()	Permite obtener el nombre y código de una materia.
Set_Materia(string value)	Permite establecer el nombre y código de una materia.
Get_Dia()	Permite obtener el día que se dicta una materia.
Set_Dia(string value)	Permite establecer el día que se dicta una materia.
Get_HoraInicio()	Permite obtener la hora de inicio del periodo de clases de una materia.
Set_HoraInicio(string value)	Permite establecer la hora de inicio del periodo de clases de una materia.
Get_HoraFin()	Permite obtener la hora de finalización del periodo de clases de una materia.
Set_HoraFin(string value)	Permite establecer la hora de finalización del periodo de clases de una materia.

Tabla 2.29 Descripción de las propiedades de la clase *HoraClase*

- Clase *Calificacion*

Nombre	Descripción
get_Materia()	Permite obtener el nombre y código de una materia.
Set_Materia(string value)	Permite establecer el nombre y código de una materia.
Get_PrimerBimestre()	Permite obtener la calificación correspondiente al primer bimestre.
Set_PrimerBimestre(string value)	Permite establecer la calificación correspondiente al primer bimestre.
Get_SegundoBimestre()	Permite obtener la calificación correspondiente al segundo bimestre.
Set_SegundoBimestre(string value)	Permite establecer la calificación correspondiente al segundo bimestre.
Get_Supletorio()	Permite obtener la calificación correspondiente al examen supletorio.
Set_Supletorio(string value)	Permite establecer la calificación correspondiente al examen supletorio.
Get_Sumatoria()	Permite obtener el promedio de las calificaciones conseguidas en una materia por el estudiante durante el semestre.
Set_Sumatoria(string value)	Permite establecer el promedio de las calificaciones conseguidas en una materia por el estudiante durante el semestre.
Get_Aprobacion()	Permite obtener la forma de aprobación de una materia.
Set_Aprobacion(string value)	Permite establecer la forma de aprobación de una materia.

Tabla 2.30 Descripción de las propiedades de la clase *Calificacion*

2.2.4 DIAGRAMAS DE ACTIVIDAD

Los diagramas de actividad, permiten modelar el flujo interno de trabajo de los objetos de las clases.

2.2.4.1 Diagrama de actividad Inicio de sesión

La Figura 2.16 muestra el diagrama de actividad de Inicio de sesión.

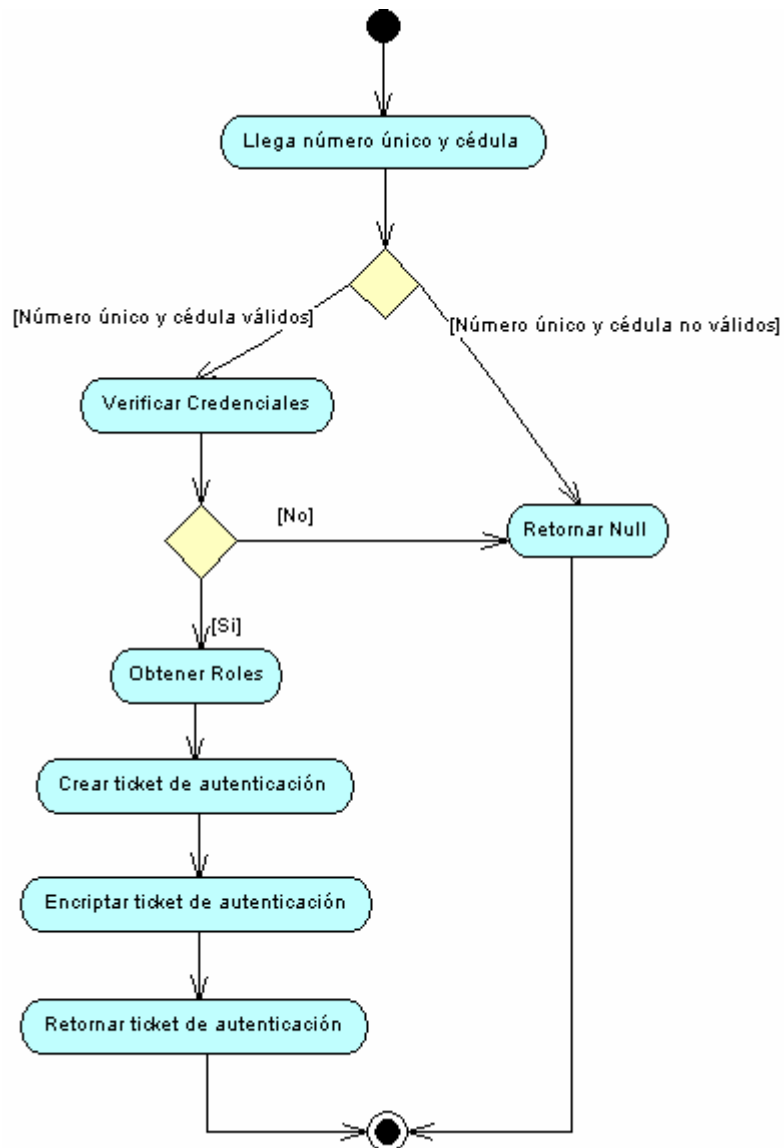


Figura 2.16 Diagrama de actividad Inicio de sesión

El diccionario para el diagrama de actividad de inicio de sesión se determinó de la siguiente manera:

1. Llega número único y cédula: el número único y la cédula no pueden ser vacíos o nulos; de suceder esto, el servicio retorna nulo.
2. Verificar credenciales: luego de comprobar que los dos campos han sido recibidos, se verifica que estos datos sean correctos; es decir que sean un usuario válido. Se consulta en la base de datos *BDDSIEM*, si el usuario no existe, el servicio retorna nulo.

3. Obtener Roles: una vez que se ha verificado que los datos son correctos, se obtienen los permisos para la sesión, para ello también esta información es consultada a la base de datos.
4. Crear *ticket* de autenticación: luego de obtener el rol correspondiente a la sesión, el servicio genera un *ticket* de autenticación; el cual tiene validez mientras la sesión permanezca activa.
5. Encriptar *ticket* de autenticación: el *ticket* de autenticación obtenido es encriptado por razones de seguridad.
6. Retornar *ticket* de autenticación: el *ticket* encriptado es devuelto por el servicio.

2.2.4.2 Diagrama de actividad Consultar información de cuenta

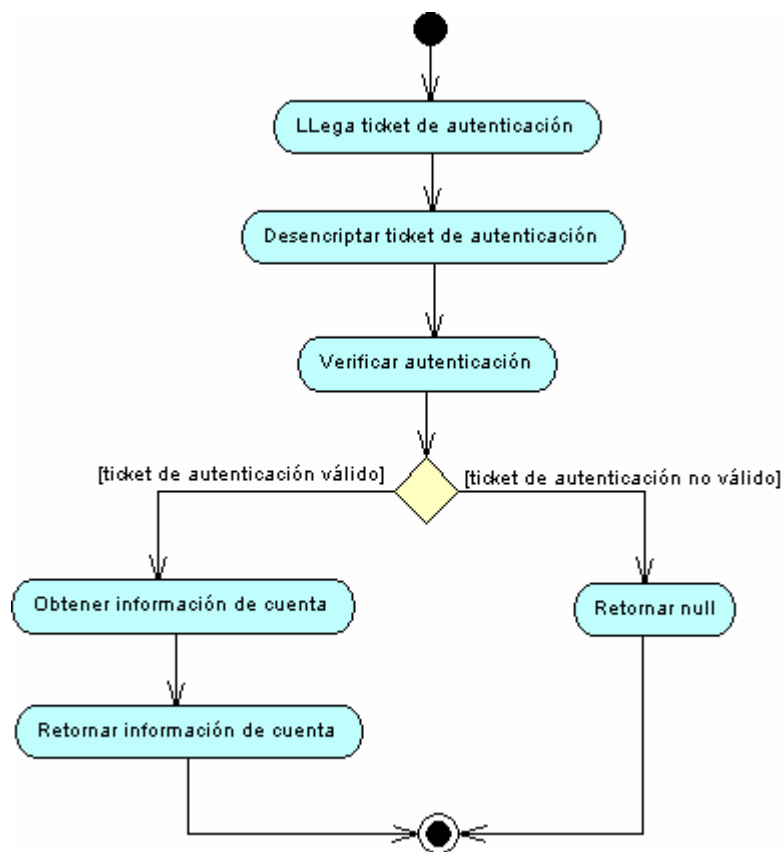


Figura 2.17 Diagrama de actividad Consultar información de cuenta

El diccionario correspondiente al diagrama de actividad (Figura 2.17), se indica a continuación:

1. Llega *ticket* de autenticación: el *ticket* de autenticación previamente obtenido es enviado al servicio Web dentro del encabezado *SOAP* de la solicitud.
2. Desencriptar *ticket* de autenticación: el servicio desencripta el *ticket* de autenticación enviado.
3. Verificar autenticación: luego de desencriptar el *ticket* de autenticación, el servicio verifica si el usuario está autenticado.
4. Obtener información de cuenta: una vez que se ha comprobado que el usuario está autenticado, el servicio obtiene la información de la cuenta de la base de datos SAE.
5. Retornar información de cuenta: el servicio devuelve la información de la cuenta.
6. Retornar nulo: sucede cuando el usuario no está autenticado.

2.2.4.3 Diagrama de actividad Consultar horario

El diccionario correspondiente al diagrama de actividad Consultar horario (Figura 2.18), se establece de la siguiente manera:

1. Llega *ticket* de autenticación: el *ticket* de autenticación previamente obtenido es enviado al servicio Web dentro del encabezado *SOAP* de la solicitud.
2. Desencriptar *ticket* de autenticación: el servicio desencripta el *ticket* de autenticación enviado.
3. Verificar autenticación, luego de desencriptar el *ticket* de autenticación, el servicio verifica si el usuario se encuentra autenticado.
4. Obtener horario de clases: una vez que se ha comprobado que el usuario está autenticado, el servicio obtiene el horario de clases de la base de datos SAE.
5. Retornar horario de clases: el servicio devuelve el horario de clases solicitado.
6. Retornar nulo: sucede cuando el usuario no está autenticado.

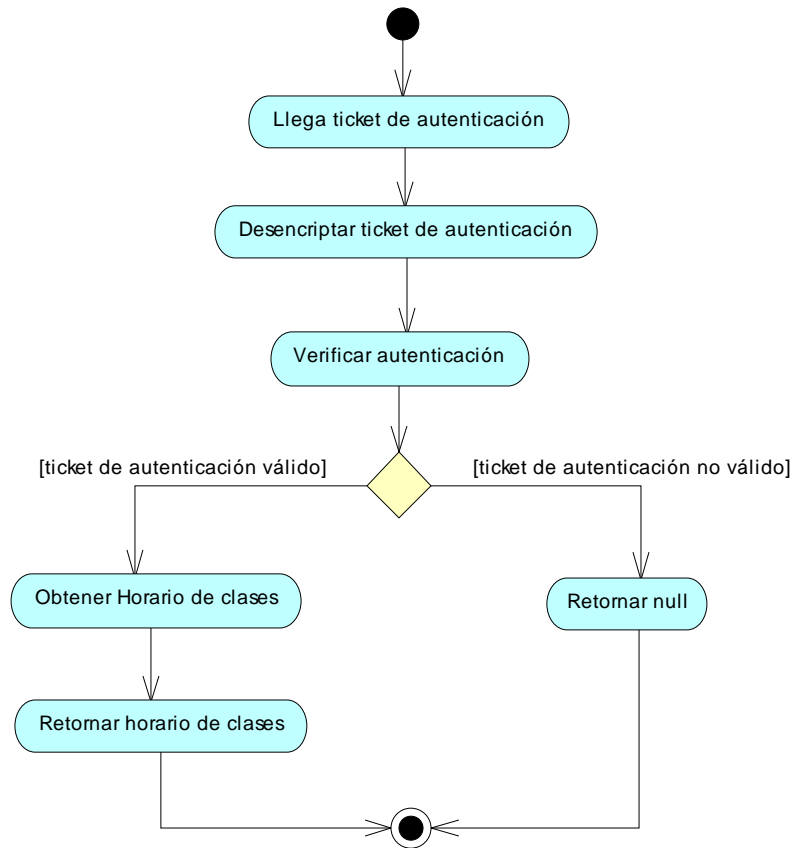


Figura 2.18 Diagrama de actividad Consultar horario

2.2.4.4 Diagrama de actividad Consultar calificaciones

El diccionario de actividades del diagrama de actividad Consultar calificaciones (Figura 2.19) queda establecido de la siguiente manera:

1. Llega *ticket* de autenticación: el *ticket* de autenticación previamente obtenido es enviado al servicio Web dentro del encabezado *SOAP* de la solicitud.
2. Desencriptar *ticket* de autenticación: el servicio desencripta el *ticket* de autenticación enviado.
3. Verificar autenticación: luego de desencriptar el *ticket* de autenticación, el servicio verifica si el usuario está autenticado.
4. Obtener calificaciones: una vez que se ha comprobado que el usuario está autenticado, el servicio obtiene las calificaciones de la base de datos SAE.
5. Retornar calificaciones: el servicio devuelve las calificaciones solicitadas.

6. Retornar nulo: sucede cuando el usuario no está autenticado.

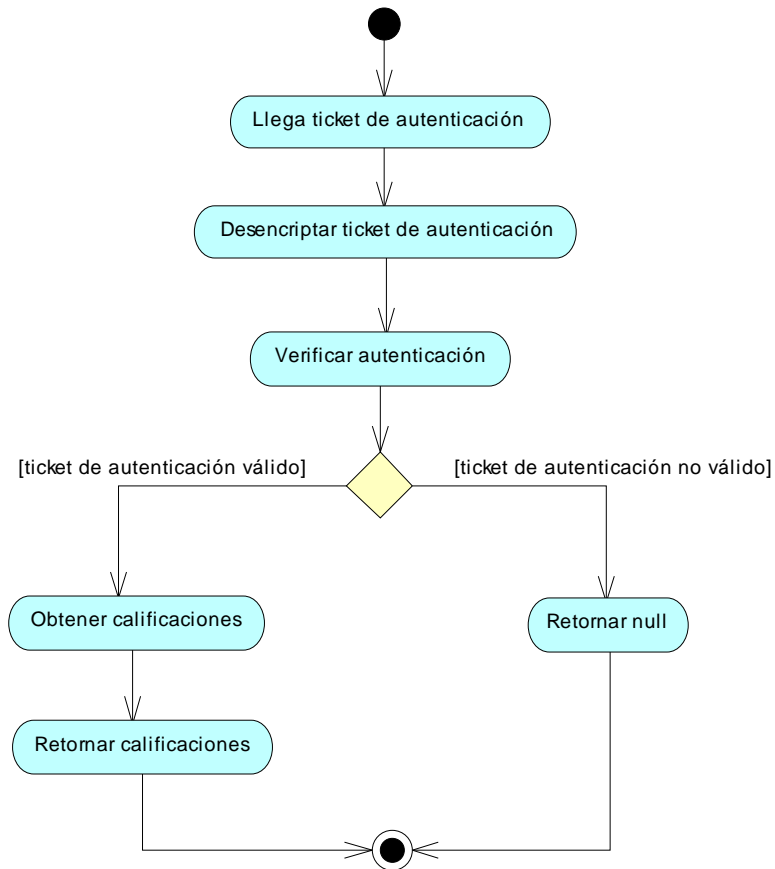


Figura 2.19 Diagrama de actividad Consultar calificaciones

2.2.4.5 Diagrama de actividad Consultar información de materia

La Figura 2.20 muestra el diagrama de actividad Consultar información de materia, cuyo diccionario de actividades queda establecido de la siguiente manera:

1. Llega *ticket* de autenticación: el *ticket* de autenticación previamente obtenido es enviado al servicio Web dentro del encabezado SOAP de la solicitud junto con el código de la materia de la cual se desea obtener su detalle.
2. Desencriptar *ticket* de autenticación: el servicio desencripta el *ticket* de autenticación enviado.

3. Verificar autenticación: luego de descryptar el *ticket* de autenticación, el servicio verifica si el usuario está autenticado.
4. Obtener detalle materia: una vez que se ha comprobado que el usuario está autenticado, el servicio obtiene la información referente a la materia de la base de datos SAE.
5. Retornar información de materia: el servicio devuelve la información de la materia correspondiente.
6. Retornar nulo: sucede cuando el usuario no está autenticado.

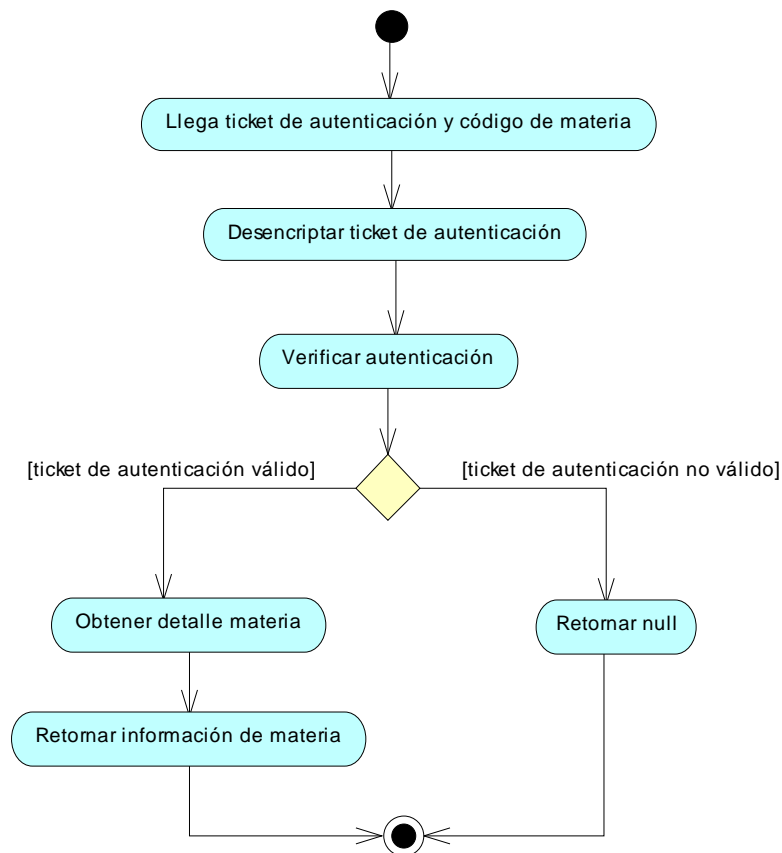


Figura 2.20 Diagrama de actividad Consultar información de materia

2.2.5 DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia, permiten indicar el flujo de acciones que ocurren entre los diferentes objetos a través del tiempo. Se presenta un diagrama de secuencia por cada caso de uso del sistema.

2.2.5.1 Diagrama de secuencia Inicio de sesión

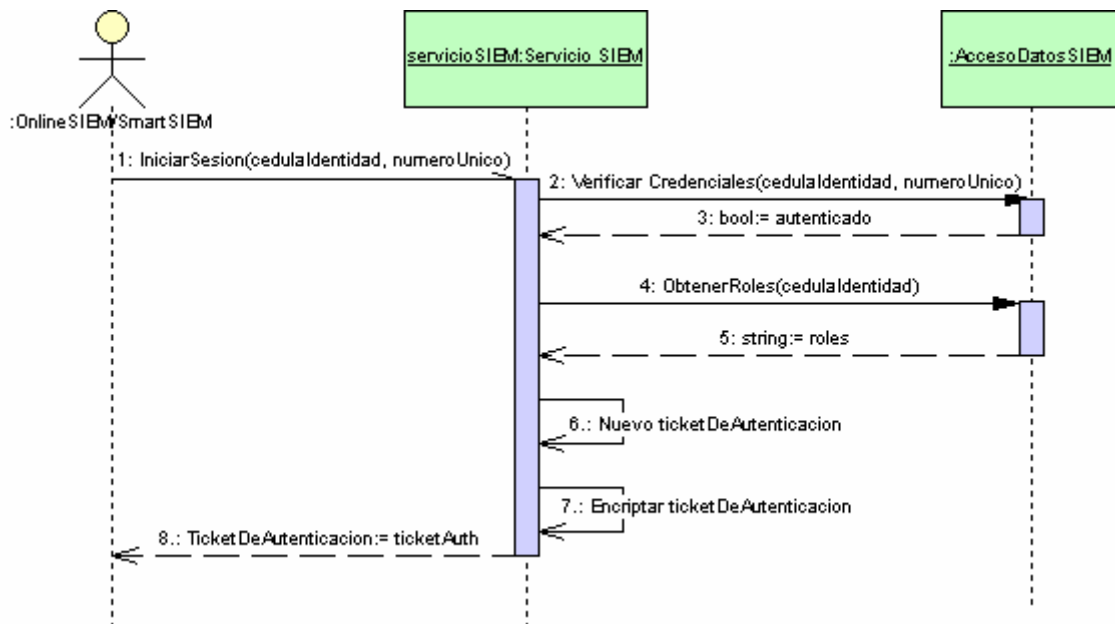


Figura 2.21 Diagrama de secuencia Inicio de sesión

La Figura 2.21 muestra la secuencia seguida en el servicio Web cuando una aplicación desea iniciar una sesión.

El proceso se inicia cuando la aplicación *OnlineSIEM* o la aplicación *SmartSIEM*, solicitan *IniciarSesion*; este método recibe la cédula de identidad y el número único de un estudiante.

El servicio Web con las credenciales recibidas, solicita *VerificarCredenciales* a una instancia de la clase *AccesoDatosSIEM*; la misma que verifica las credenciales en la base de datos *DBBSIEM* para establecer si el usuario es válido. El objeto *AccesoDatosSIEM*, retorna verdadero o falso, dependiendo si las credenciales son correctas o no.

Si el objeto de la clase *AccesoDatosSIEM*, retorna falso, el servicio retorna nulo y no se puede acceder a los servicios disponibles.

Si el objeto de la clase *AccesoDatosSIEM*, retorna verdadero, el objeto *servicioSIEM*, solicita *ObtenerRoles* donde se pasa la cédula de identidad

ingresada. El objeto de la clase *AccesoDatosSIEM* devuelve el rol correspondiente, almacenado en un *string*.

Obtenido el rol del usuario, el objeto *servicioSIEM* genera un *ticket* de autenticación y lo encripta para retornarlo a la aplicación que solicitó iniciar la sesión. El *ticket* de autenticación es almacenado en una variable de tipo *TicketDeAutenticación* y servirá para poder acceder a las funciones disponibles en el servicio.

2.2.5.2 Diagrama de secuencia Consultar información de cuenta

El diagrama mostrado en la Figura 2.22 indica la secuencia seguida para obtener información de la cuenta, luego de haber iniciado una sesión exitosa con el servicio y haber obtenido un *ticket* de autenticación válido.

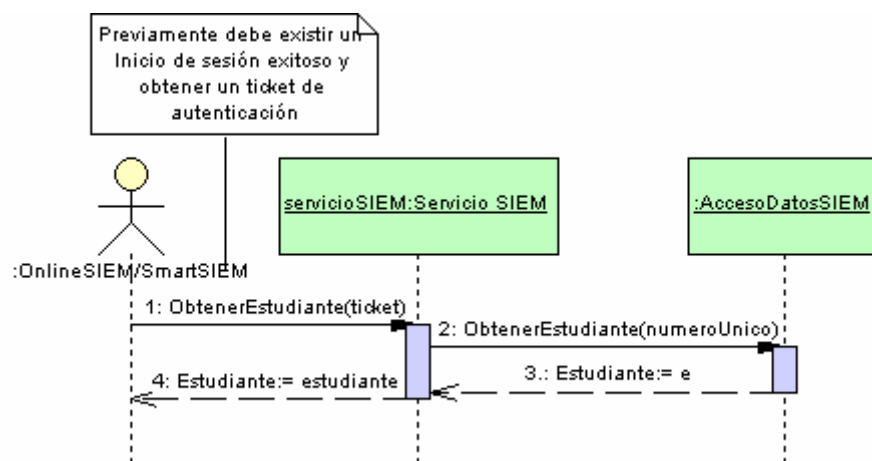


Figura 2.22 Diagrama de secuencia Consultar información de cuenta

Una de las dos aplicaciones (*OnlineSIEM* ó *SmartSIEM*) solicita *ObtenerEstudiante* al objeto *servicioSIEM*, para esto proporciona un *ticket* de autenticación al servicio Web. Se verifica si el usuario es válido, caso contrario, se retorna nulo.

Una vez que se ha verificado que el usuario es válido, el objeto *servicioSIEM* solicita el método *ObtenerEstudiante* al objeto de la clase *AccesoDatosSIEM*; en esta llamada se pasa el número único correspondiente al estudiante, el mismo

que es obtenido del *ticket* de autenticación. El objeto devuelve la información del estudiante almacenada en un objeto *Estudiante*.

Por último el objeto *servicioSIEM* retorna la información del estudiante a las aplicaciones que solicitaron la información.

2.2.5.3 Diagrama de secuencia Consultar horario

En la Figura 2.23 se muestra el diagrama de secuencia seguido para conocer el horario de clases, luego de haber iniciado una sesión exitosa con el servicio y obtenido un *ticket* de autenticación válido.

Una de las dos aplicaciones (*OnlineSIEM* ó *SmartSIEM*) solicita *ObtenerHorario* al objeto *servicioSIEM*, para esto proporciona un *ticket* de autenticación al servicio Web. Se verifica si el usuario es válido, caso contrario, se retorna nulo.

Una vez que se ha verificado que el usuario es válido, el objeto *servicioSIEM* solicita el método *ObtenerHorario* al objeto de la clase *AccesoDatosSIEM*; en esta llamada se pasa el número único correspondiente al estudiante, el mismo que es obtenido del *ticket* de autenticación. El objeto devuelve el horario de clases del estudiante almacenado en una lista de objetos *HoraClase*.

Por último el objeto *servicioSIEM* retorna el horario de clases a las aplicaciones que solicitaron la información.

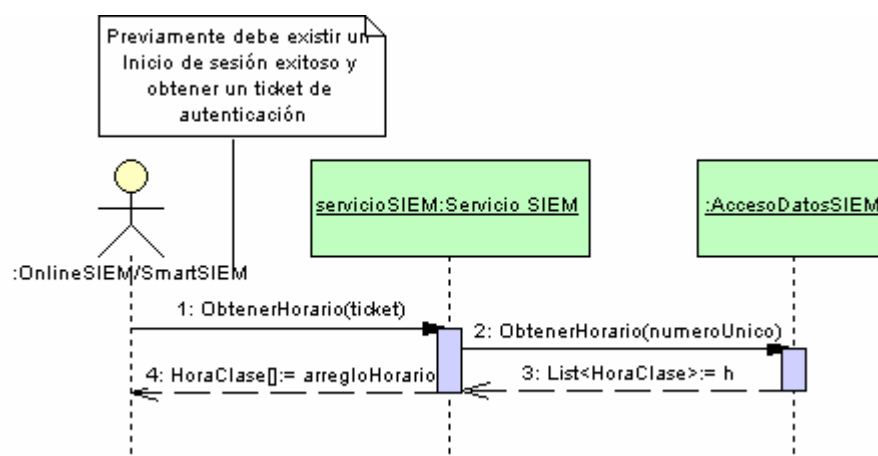


Figura 2.23 Diagrama de secuencia Consultar horario

2.2.5.4 Diagrama de secuencia Consultar calificaciones

El diagrama de la Figura 2.24 muestra la secuencia seguida para conocer las calificaciones conseguidas por el estudiante. Previamente se debe haber iniciado una sesión exitosa con el servicio y obtenido un *ticket* de autenticación válido.

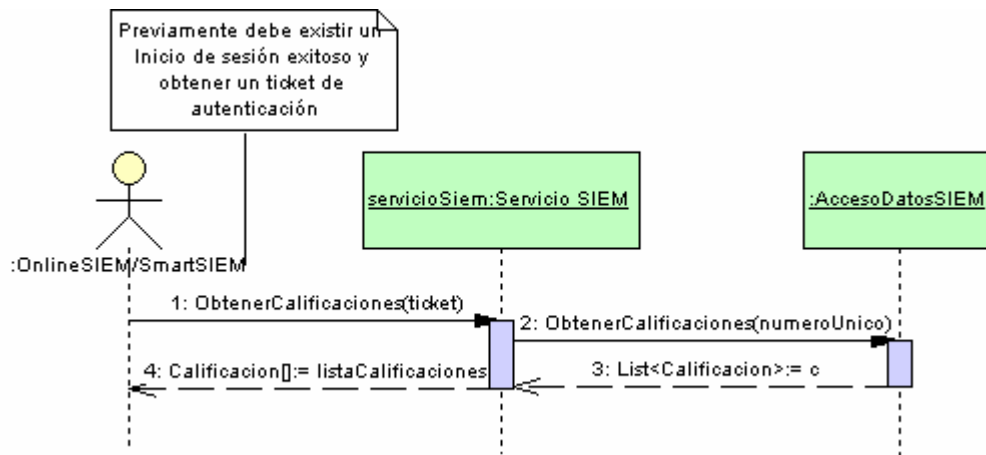


Figura 2.24 Diagrama de secuencia Consultar calificaciones

Una de las dos aplicaciones (*OnlineSIEM* ó *SmartSIEM*) solicita *ObtenerCalificaciones* al objeto *servicioSIEM*, para esto proporcionan un *ticket* de autenticación al servicio Web. Se verifica si el usuario es válido, caso contrario, se retorna nulo.

Una vez que se ha verificado que el usuario es válido, el objeto *servicioSIEM* solicita el método *ObtenerCalificaciones* al objeto de la clase *AccesoDatosSIEM*, en esta llamada se pasa el número único correspondiente al estudiante, el mismo que es obtenido del *ticket* de autenticación. El objeto devuelve las calificaciones del estudiante, almacenadas en una lista de objetos *Calificacion*.

Por último el objeto *servicioSIEM* retorna las calificaciones del estudiante.

2.2.5.5 Diagrama de secuencia Consultar información de materia

El diagrama de la Figura 2.25 muestra la secuencia seguida para conseguir información detallada de una materia luego de haber iniciado una sesión exitosa con el servicio y obtenido un *ticket* de autenticación válido.

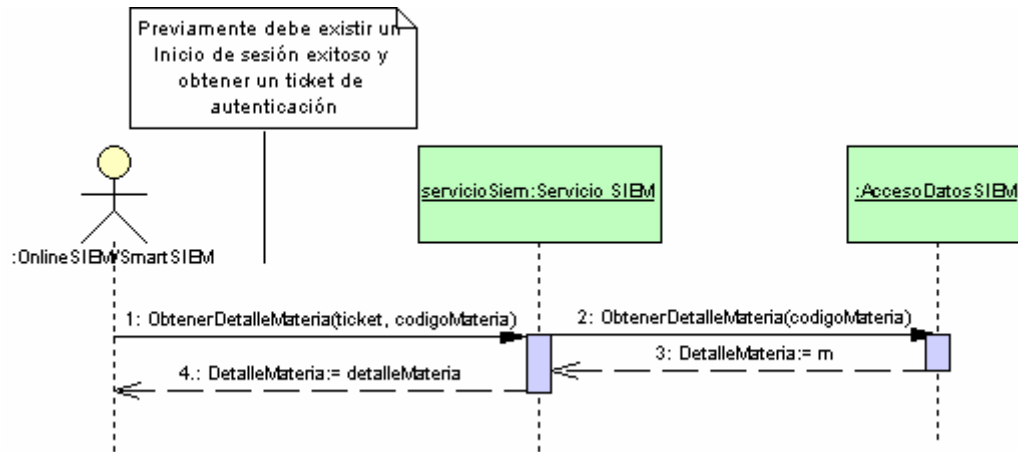


Figura 2.25 Diagrama de secuencia Consultar información de materia

Una de las dos aplicaciones (*OnlineSIEM* ó *SmartSIEM*) solicita *ObtenerDetalleMateria* al objeto *servicioSIEM*, para esto proporcionan un *ticket* de autenticación al servicio Web y el código de la materia de la que se desea obtener información. El servicio verifica si el usuario es válido, caso contrario, se retorna nulo.

Una vez que se ha verificado que el usuario es válido, el objeto *servicioSIEM* solicita el método *ObtenerDetalleMateria* al objeto de la clase *AccesoDatosSIEM*; en esta llamada se pasa el código de la materia recibido. El objeto devuelve la información detallada de la materia, almacenadas en un objeto *DetalleMateria*.

Por último el objeto *servicioSIEM* retorna la información detallada de la materia a las aplicaciones que solicitaron la información.

2.3 IMPLEMENTACIÓN

Para la implementación del servicio Web *SIEM*, se utilizó la herramienta *Visual Studio 2005* de *Microsoft*, en conjunto con el servidor de base de datos *SQL Express Edition*.

Visual Studio 2005 permite trabajar directamente con los orígenes de datos, para esto se deben agregar las bases de datos en la carpeta *App_Data* del proyecto o agregar el servidor de base de datos con la ayuda de el explorador de servidores; esto además brinda facilidades al momento de transportar el servicio Web a un

servidor de producción, pues al formar parte del proyecto la base de datos se instala en conjunto, sin necesidad de un procedimiento adicional.

La implementación comienza con la elaboración de los procedimientos almacenados para cada una de las bases de datos *BDDSIEM* y *SAE*. Como se dijo, *Visual Studio 2005* permite trabajar directamente con los orígenes de datos, y por tanto se puede escribir los procedimientos almacenados desde *Visual Studio*. Para realizar esto, se debe tener un proyecto creado y efectuar los siguientes pasos:

1. Clic en Ver, Explorador de servidores.

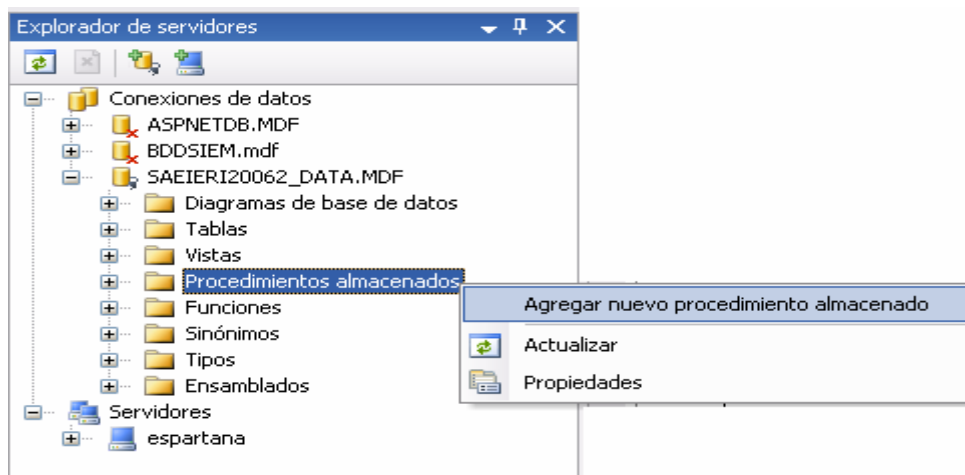


Figura 2.26 Agregar procedimiento almacenado

2. Se elige la base de datos con la que se va a trabajar, se desplaza hasta la carpeta procedimientos almacenados y dando clic derecho se selecciona Agregar nuevo procedimiento almacenado, como se muestra en la Figura 2.26.
3. Se despliega una pantalla en la que se puede escribir el procedimiento almacenado, también se puede hacer con la ayuda de un diseñador de consultas; una herramienta muy útil que es parte de *Visual Studio 2005*. Para esto se hace clic derecho sobre el código del procedimiento y se escoge Diseñar bloque SQL, como se muestra en la Figura 2.27.

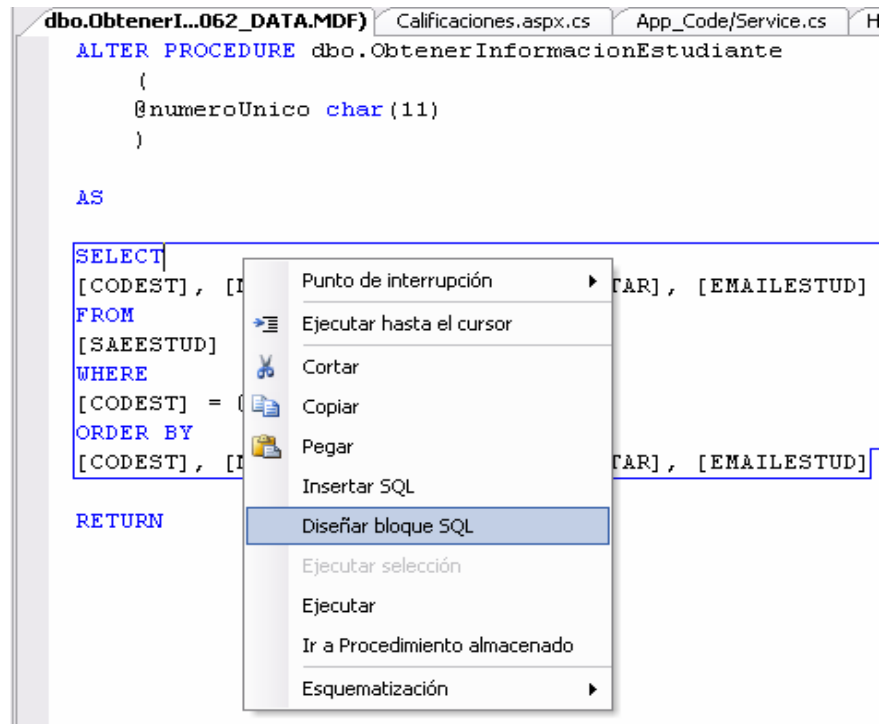


Figura 2.27 Abrir el diseñador de consultas SQL

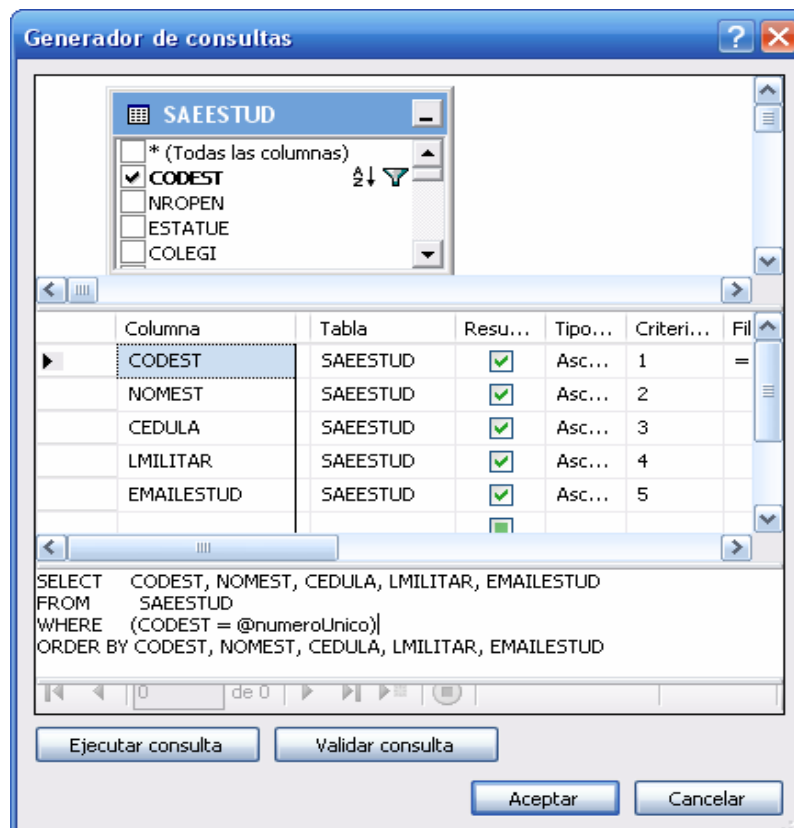


Figura 2.28 Generador de consultas SQL

- Esta acción despliega la herramienta de diseño de consultas en la que se puede agregar las tablas sobre las que se realiza las consultas y escoger los campos requeridos, como se muestra en la Figura 2.28.

El código de todos los procedimientos almacenados se presenta en el anexo B-1. El siguiente es un ejemplo de los procedimientos almacenados desarrollados.

```
CREATE PROCEDURE dbo.ObtenerInformacionEstudiante
(
    @numeroUnico char(11)
)
AS
SELECT
[CODEST], [NOMEST], [CEDULA], [LMILITAR], [EMAILESTUD]
FROM
[SAEESTUD]
WHERE
[CODEST] = @numeroUnico
ORDER BY
[CODEST], [NOMEST], [CEDULA], [LMILITAR], [EMAILESTUD]
RETURN
```

Este procedimiento almacenado recupera de la base de datos la información del estudiante como es su nombre, número único, número de cédula, correo electrónico y libreta militar. Toda esta información se encuentra en la tabla SAEESTUD y es buscada a través del número único del estudiante.

Una vez terminada la elaboración de los procedimientos almacenados, se continúa con la implementación de la capa de acceso a datos.

Los procedimientos almacenados son llamados por la capa de acceso a datos, para obtener la información requerida de las bases de datos. La capa de acceso a datos es conformada por funciones estáticas que ejecutan procedimientos almacenados.

Estos métodos se encargan de inicializar, configurar y ejecutar el procedimiento almacenado y a continuación leer los resultados y devolverlos en forma de objetos de la capa de negocio.

Se utilizan las clases:

- *SqlConnection*, para establecer la conexión con las bases de datos.
- *SqlCommand*, que representa el procedimiento almacenado.
- *SqlParameter*, que representa los parámetros de consulta del procedimiento almacenado.
- *SqlDataReader*, que permite leer los resultados de la consulta.

Se utiliza directivas *using*, que permiten liberar lo antes posible la memoria utilizada para almacenar los objetos de conexión a las bases de datos. El código de los métodos de la capa de acceso a datos se encuentra en el anexo D.

Por último se elabora el servicio Web que extrae la información necesaria de la capa de acceso a datos.

El servicio Web *SIEM* fue creado con *ASP.NET*, pues de esta manera se cuenta con las características de rendimiento, administración de estados y autenticación de *ASP.NET*. Esto permite satisfacer los requisitos planteados para el servicio Web.

Para crear el servicio Web *ASP.NET* con *Visual Studio 2005*, se realizaron los siguientes pasos:

1. Abrir el *Visual Studio 2005*.
2. Clic en Archivo, Nuevo, Sitio Web..., se presentará la pantalla que se muestra en la Figura 2.29.
3. En esta pantalla se escoge servicio Web *ASP.NET*, se asigna el nombre apropiado al servicio Web y se hace clic en Aceptar. Esto creará el proyecto de *Visual Studio* para desarrollar el servicio Web *SIEM*.

Una vez creado el servicio Web, se le agregan los métodos que se expondrán en el servicio, para esto se utiliza la palabra clave *WebMethod* seguida del método a publicar. El ejemplo de código siguiente ilustra lo expuesto:

```
[WebMethod]
public Estudiante ObtenerEstudiante()
{
    if (!User.Identity.IsAuthenticated)
        return null;
    return AccesoDatosSIEM.ObtenerEstudiante(User.Identity.Name);
}
```

Todos los procedimientos del servicio Web se encuentran en el anexo D.

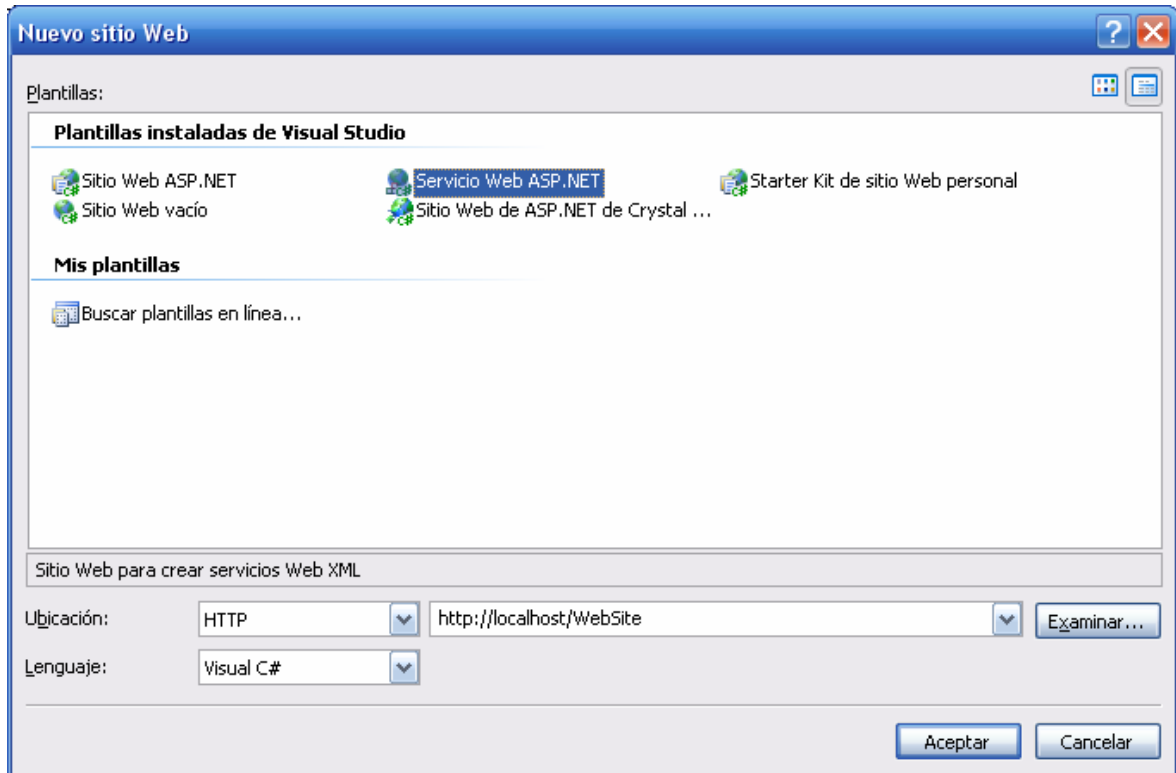


Figura 2.29 Crear un servicio Web ASP.NET

Un requerimiento fundamental para el servicio Web *SIEM* es la autenticación. Este requisito se logra mediante la adaptación de autenticación por formularios de *ASP.NET* al servicio Web *SIEM*. Lo cual se consigue enviando el *ticket* de autenticación en el encabezado *SOAP*; debe recordarse que el *ticket* de autenticación en autenticación por formularios de *ASP.NET* se envía en una *cookie*. La implementación consiste en enviar el *ticket* de autenticación en el encabezado *SOAP*, en cada requerimiento al servicio Web.

La autenticación descrita se implementa sobrescribiendo el método *Application_AuthenticateRequest* del archivo *Global.asax*, en el cual se implementa la lógica antes descrita. Además es necesario utilizar la palabra clave *SoapHeader* en los métodos del servicio Web, esto con el objetivo que el *ticket* de autenticación sea incluido en el encabezado *SOAP*. El siguiente segmento de código muestra lo expuesto.

```
[WebMethod]
[SoapHeader("ticketDeAutenticacion")]
public Estudiante ObtenerEstudiante()
{
    if (!User.Identity.IsAuthenticated)
        return null; // se retorna nulo

    return AccesoDatosSIEM.ObtenerEstudiante(User.Identity.Name);
}
```

En el anexo B-2 se enseña el código explicado del archivo *Global.asax*, donde se implementa la lógica de autenticación.

Capítulo 3

Diseño e implementación de la aplicación *Online*

3 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN *ONLINE*

3.1 REQUERIMIENTOS

3.1.1 REQUERIMIENTOS DEL NEGOCIO

Los requerimientos del negocio para la aplicación *Online* son los mismos planteados en el capítulo anterior para el servicio Web, pues los requerimientos del negocio explican cómo se informa el estudiante de sus horarios y calificaciones en la actualidad, sin el uso del sistema *SIEM*.

3.1.2 REQUERIMIENTOS DEL SISTEMA

3.1.2.1 Lista de actores

Estudiante: persona perteneciente a la Carrera de Electrónica y Redes de Información, que usando un navegador Web desde un dispositivo móvil del tipo *Pocket PC*, accede al *Sistema de Información Estudiantil Móvil Online (OnlineSIEM)*.

Servicio Web *SIEM*: aplicación Web que permite obtener la información requerida.

3.1.2.2 Lista de casos de uso

Los casos de uso son los mismos que los presentados en el capítulo anterior. Esto se debe a que tanto el servicio Web y la aplicación *Online* deben cumplir los mismos requerimientos, el informar al estudiante de horarios y calificaciones.

CU1: Iniciar sesión, el estudiante inicia una sesión en *SIEM*, para esto utiliza su número de cédula y su número único.

CU2: Consultar información de cuenta, el estudiante puede revisar la información de su sesión.

CU3: Consultar horario, el estudiante consulta en *SIEM* el horario de clases.

CU4: Consultar calificaciones, el estudiante consulta en *SIEM* sus calificaciones académicas.

CU5: Consultar información de materia, se obtiene información como nombre, número de créditos, etc. de las materias tomadas por el estudiante.

CU6: Cerrar sesión, el estudiante termina la sesión con *SIEM*.

3.1.2.3 Diagrama de casos de uso

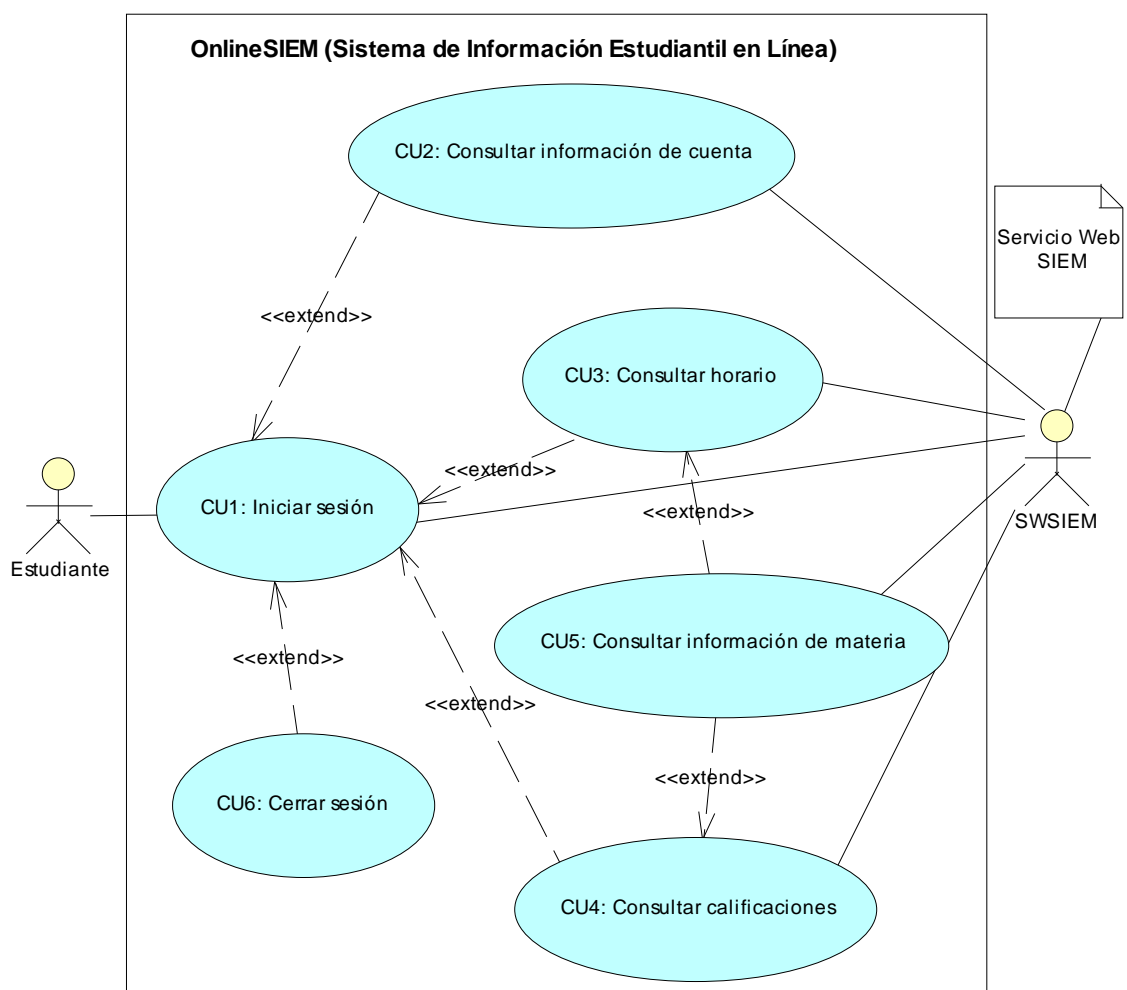


Figura 3.1 Diagrama de casos de uso de la aplicación *OnlineSIEM*

La Figura 3.1 muestra el diagrama de casos de uso para la aplicación *OnlineSIEM*. Este diagrama de casos de uso es similar al de la Figura 2.5, ya que

se tiene los mismos casos de uso; pero se diferencian en que la aplicación *Online* accede al servicio Web *SIEM* para obtener la información.

3.1.2.4 Especificación de casos de uso

La funcionalidad del caso de uso CU1 Inicio de sesión para la aplicación *Online* se detalla en la Tabla 3.1.

Caso de Uso	CU1: Inicio de sesión.
Propósito	Iniciar una sesión con el servicio Web <i>SIEM</i> , proporcionado la cédula de identidad y número único de un determinado estudiante.
Precondiciones	Ninguna.
Postcondiciones	Con la sesión establecida se puede solicitar información sobre calificaciones y horarios.
Actores	Estudiante, petición de inicio de sesión. Servicio Web <i>SIEM</i> , responde a la solicitud de inicio de sesión.
Entradas	Cédula de identidad y número único.
Salidas	Identificador de sesión.
Proceso/Evento	Descripción
P: Cargar Página.	<ol style="list-style-type: none"> 1. El estudiante solicita la página de inicio de sesión por medio de un navegador Web. 2. La página es creada y devuelta al estudiante.
E: Clic Iniciar sesión.	<ol style="list-style-type: none"> 1. Verificar que el número único y cédula, no sean nulos ni vacíos. 2. Iniciar sesión con el servicio Web. 3. Si el inicio de sesión es exitoso, se recibe un <i>ticket</i> de autenticación que permitirá interactuar con el servicio Web <i>SIEM</i>; este <i>ticket</i> es almacenado para consultas posteriores. 4. La aplicación <i>OnlineSIEM</i> crea un <i>ticket</i> de autenticación y lo agrega en una <i>cookie</i>. 5. Se redirige a la página de información de cuenta. 6. En caso de que el inicio de sesión sea fallido, se presenta un mensaje de Número de cédula o número único incorrectos.

Tabla 3.1 Especificación del caso de uso Inicio de sesión

La funcionalidad del caso de uso CU2 Consultar información de cuenta para la aplicación *Online* se especifica en la Tabla 3.2.

La funcionalidad del caso de uso CU3 Consultar horario para la aplicación *Online* se detalla en la Tabla 3.3.

Caso de Uso	CU2: Consultar información de cuenta.
Propósito	Acceder a información de cuenta de un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información de cuenta. Servicio Web <i>SIEM</i> , proporciona los datos solicitados.
Entradas	<i>Ticket</i> de autenticación.
Salidas	Los siguientes datos del estudiante: Nombre, número de cédula, número único, identificación militar, correo electrónico.
Proceso/Evento	Descripción
P: Sesión iniciada correctamente. E: Clic Ir principal.	1. El estudiante solicita la información de su cuenta o el estudiante a iniciado una sesión correctamente.
P: Autenticación.	1. Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido.
P: Cargar página.	1. Se crean la página de cuenta y sus controles. 2. Se realiza la consulta de información de cuenta al servicio Web. 3. Se carga los controles con la información recibida. 4. La página es enviada al usuario.

Tabla 3.2 Especificación del caso de uso Consultar información de cuenta

Caso de Uso	CU3: Consultar horario.
Propósito	Solicitar el horario de clases de un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información referente al horario de clases. Servicio Web <i>SIEM</i> , responde a la solicitud.
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Horario de las materias tomadas con el siguiente detalle: Día, nombre de la materia, hora inicio, hora fin.
Proceso/Evento	Descripción
E: Clic Consultar horario.	1. El estudiante solicita la información de su horario de clases.
P: Autenticación.	1. Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido.
P: Cargar página.	1. Se crean la página de horario y sus controles. 2. Se realiza la consulta de horario al servicio Web. 3. Se carga los controles con la información recibida. 4. La página es enviada al usuario.

Tabla 3.3 Especificación del caso de uso Consultar horario

La funcionalidad del caso de uso CU4 Consultar calificaciones para la aplicación *Online* se detalla en la Tabla 3.4.

Caso de Uso	CU4: Consultar calificaciones.
Propósito	Solicitar información de las calificaciones conseguidas por un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información de sus calificaciones. Servicio Web <i>SIEM</i> , responde a la solicitud.
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Las calificaciones del estudiante con el siguiente detalle: Nombre de la materia, calificación primer bimestre, calificación segundo bimestre, calificación supletorio, suma, aprobación.
Proceso/Evento	Descripción
E: Clic Consultar calificaciones.	1. El estudiante solicita la información de sus calificaciones.
P: Autenticación.	1. Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido.
P: Cargar página.	1. Se crean la página de calificaciones y sus controles. 2. Se realiza la consulta de calificaciones al servicio Web. 3. Se carga los controles con la información recibida. 4. La página es enviada al usuario.

Tabla 3.4 Especificación del caso de uso Consultar calificaciones

La funcionalidad del caso de uso CU5 Consultar información de materia para la aplicación *Online* se detalla en la Tabla 3.5.

La funcionalidad del caso de uso CU6 Cerrar sesión para la aplicación *Online* se detalla en la Tabla 3.6

Caso de Uso	CU5: Consultar información de materia.
Propósito	Conocer el detalle de una materia.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información detallada de una materia. Servicio Web <i>SIEM</i> , responde a la solicitud con la información requerida.
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Información de la materia con el siguiente detalle: Código de la materia, nombre de la materia., aula, profesor de la materia, número de créditos, paralelo, categoría.
Proceso/Evento	Descripción
E: Clic nombre materia.	1. El estudiante solicita la información detallada de una materia
P: Autenticación.	1. Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido.
P: Cargar página.	1. Se crean la página de materia y sus controles. 2. Se realiza la consulta del detalle de una materia al servicio Web. 3. Se carga los controles con la información recibida. 4. La página es enviada al usuario.

Tabla 3.5 Especificación del caso de uso Consultar información de materia

Caso de Uso	CU6: Cerrar sesión.
Propósito	Terminar la sesión.
Precondiciones	La sesión debe estar iniciada.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita finalizar la sesión.
Entradas	Ninguna.
Salidas	Ninguna.
Proceso/Evento	Descripción
E: Clic Cerrar sesión.	1. Se elimina el <i>ticket</i> de autenticación con el servicio Web. 2. Se termina la sesión. 3. Se redirige a la página de inicio de sesión.

Tabla 3.6 Especificación del caso de uso Cerrar sesión

3.1.2.5 Diseño de Interfaces

El diseño de interfaces consta de las páginas *ASP.NET* que forman parte de la aplicación *OnlineSIEM*.

- Interfaz de usuario para inicio de sesión (Figura 3.2)
Interfaz usada en el proceso de inicio de sesión, el usuario debe ingresar el número de cédula y el número único.



The screenshot shows a web interface for 'SIEM OnLine'. At the top left is the SIEM logo, and at the top right is the text 'OnLine'. Below this is the heading 'Inicio de Sesión'. There are two input fields: the first is labeled 'Número de Cedula:' and the second is labeled 'Número Único:'. Below these fields is a button labeled 'Iniciar Sesión'.

Figura 3.2 Interfaz de usuario para inicio de sesión



The screenshot shows a web interface for 'SIEM OnLine'. At the top left is the SIEM logo, and at the top right is the text 'OnLine'. Below this is the heading 'BIENVENIDO A SIEM' followed by the text 'ARMIJOS CURIPOMA' and 'ADRIAN ESTUARDO'. There is a button labeled 'Cerrar Sesión'. Below this is the heading 'CONSULTAS' followed by two links: '[Horario de Clases...](#)' and '[Calificaciones...](#)'. Below this is the heading 'DATOS PERSONALES' followed by several fields: 'Número de Cédula: 1103882302', 'Número Único: 0210075', 'Id. Militar: 198311000267', and 'Correo Electrónico: Ninguno'.

Figura 3.3 Interfaz de usuario para información del estudiante

- Interfaz de usuario para información del estudiante (Figura 3.3)
Mediante esta interfaz se muestra la información del estudiante y se proporcionan vínculos para acceder al horario y calificaciones. Además permite al estudiante cerrar la sesión.
- Interfaz de usuario para horario de clases (Figura 3.4)



HORARIO DE CLASES	
LUNES	HORA
ADMIN. Y GEST. DE REDES	9-11
PREPARACION, GESTION Y EVAL.	11-14
MARTES	HORA
ADMIN. Y GEST. DE REDES	9-11
DISEÑO Y GESTION DE SOFTWARE	14-15
SEGURIDAD EN REDES	18-20
MIERCOLES	HORA
REDES DE ÁREA LOCAL INALAMBRICAS	11-13
JUEVES	HORA
REDES E INTRANET	9-10
REDES DE ÁREA LOCAL INALAMBRICAS	11-13

Figura 3.4 Interfaz de usuario para horario de clases

Interfaz por medio de la cual se muestra al estudiante su horario de clases. La información se encuentra agrupada por días. También se presentan

vínculos para obtener más información sobre cada materia y regresar al menú principal.

- Interfaz de usuario para calificaciones (Figura 3.5)

SIEM **OnLine**

[Ir principal...](#)

CALIFICACIONES

[ADMIN. Y GEST. DE REDES](#)

1Bim	2Bim	Sup	Sum	Apr
7,70	8,20		31,80	E

[DISEÑO Y GESTION DE SOFTWARE](#)

1Bim	2Bim	Sup	Sum	Apr
7,80	7,00		29,60	E

[PREPARACION, GESTION Y EVAL.](#)

1Bim	2Bim	Sup	Sum	Apr
7,10	7,80		29,80	E

[REDES DE ÁREA LOCAL INALAMBRICAS](#)

1Bim	2Bim	Sup	Sum	Apr
8,60	7,00		31,20	E

[REDES E INTRANET](#)

1Bim	2Bim	Sup	Sum	Apr
8,70	8,10		33,60	E

Figura 3.5 Interfaz de usuario para calificaciones

Interfaz por medio de la cual se muestra al estudiante las calificaciones obtenidas por cada materia. El estudiante podrá conocer las calificaciones del primer y segundo bimestre, supletorio, así como el promedio y la forma de aprobación de cada materia.

Al igual que la interfaz de usuario para el horario de clases, se presentan vínculos para obtener más información sobre cada materia y regresar al menú principal.

- Interfaz de usuario para información de materia (Figura 3.6)

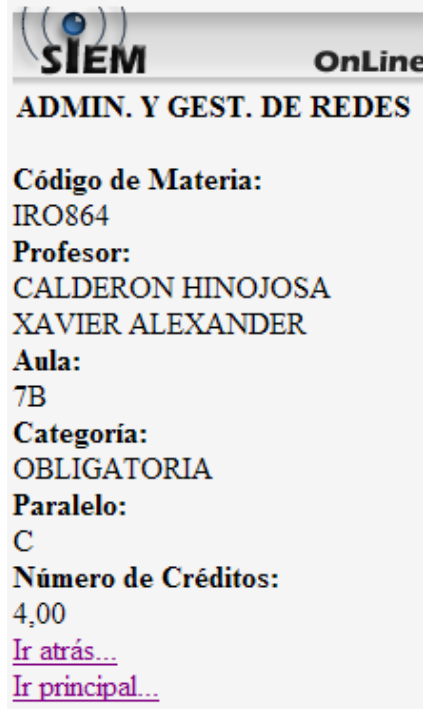


Figura 3.6 Diseño de interfaz para información de materia

Interfaz que permite mostrar al estudiante información detallada de la materia. El estudiante podrá conocer el aula, profesor, paralelo, número de créditos, la categoría y el código de la materia. Además se presentan vínculos para regresar a la página anterior o ir al menú principal.

3.2 DISEÑO

3.2.1 DIAGRAMA DE CLASES

La aplicación *Online* es básicamente una capa de interfaz, debido a que sirve para presentar la información obtenida del servicio Web (desarrollado en el capítulo anterior, el cual contiene las capas de reglas del negocio y de acceso a datos).

El diagrama de clases de la Figura 3.7, muestra las clases de interfaz del sistema *Online*.

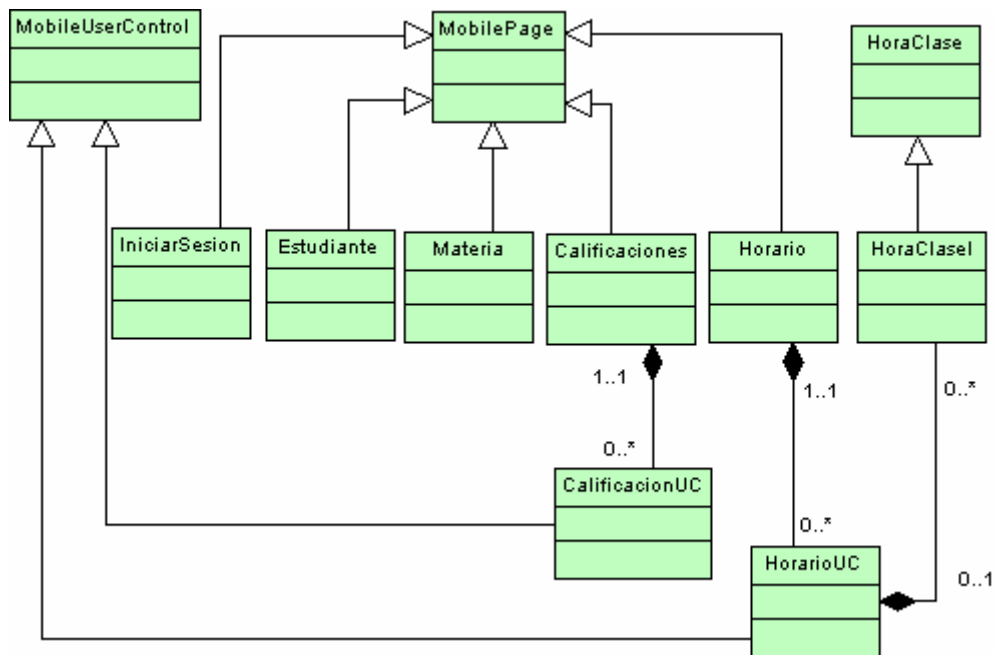


Figura 3.7 Diagrama de clases de la aplicación *Online*

Este diagrama muestra las clases que representan las páginas de contenido dinámico por las que navegará el estudiante, así como las clases que representan controles de usuario.

Como se puede ver, las clases: *IniciarSesion*, *Estudiante*, *Materia*, *Calificaciones* y *Horario* son derivadas de la clase *MobilePage*. Estas clases son las páginas utilizadas para mostrar los datos obtenidos del servicio Web.

La clase *IniciarSesion* representa la página para el inicio de sesión, esta página contiene los controles que permiten ingresar el número de cédula y el número único de un estudiante, además envía la solicitud de inicio de sesión al servicio Web *SIEM*. Cuando se inicia la sesión se tienen dos *tickets* de autenticación uno para interactuar con el servicio Web *SIEM* y el otro para interactuar con la aplicación *Online*.

La clase *Estudiante* representa la página de información de cuenta, además es la página *Home* o principal. Contiene controles que permiten mostrar la información del estudiante.

La clase *Materia* representa la página donde se muestra la información detallada de una determinada materia.

La clase *Calificaciones* es la página Web que muestra información de las calificaciones de un determinado estudiante. Contiene los controles de usuario de calificaciones (*CalificacionUC*).

La clase *CalificacionUC*, es un control de usuario por tanto se deriva de *MobileUserControl*. Es utilizada para mostrar las calificaciones del estudiante.

La clase *Horario* es la página Web que muestra el horario de clases de un determinado estudiante. Contiene los controles de usuario de horario (*HorarioUC*).

La clase *HorarioUC*, es un control de usuario por tanto se deriva de *MobileUserControl*. Es utilizada para mostrar el horario de un estudiante.

La clase *HoraClaseI* se deriva de la clase *HoraClase* y es utilizada para manejar el formato de presentación del horario de clases.

3.2.2 ATRIBUTOS

Las clases de interfaz tienen como atributos elementos para interactuar con el usuario y con el servicio Web XML.

IniciarSesion	
- servicioSIEM	: Service
- strTicket	: string
# ImageSIEM	: Image
# LabelIniciarSesion	: Label
# LabelNumeroCedula	: Label
# TextBoxNumeroCedula	: TextBox
# LabelNumeroUnico	: Label
# TextBoxNumeroUnico	: TextBox
# CustomValidatorInicioSesion	: CustomValidator
# CommandIniciarSesion	: Command
# FormLogin	: Form

Figura 3.8 Atributos de la clase *IniciarSesion*

- Clase *IniciarSesion*, los atributos se muestran en la Figura 3.8, éstos son descritos en la Tabla 3.7.

Nombre	Descripción
servicioSIEM	Permite comunicarse con el servicio Web.
strTicket	<i>Ticket</i> de autenticación para acceder al servicio Web.
ImageSIEM	Imagen que se despliega en la parte superior de la página.
LabelIniciarSesion	Muestra el texto "Iniciar Sesión".
LabelNumeroCedula	Muestra el texto "Número de Cédula".
TextBoxNumeroCedula	Recoge el número de cedula del estudiante.
LabelNumeroUnico	Muestra el texto "Número Único".
TextBoxNumeroUnico	Recoge el número único del estudiante.
CustomValidatorIniciarSesion	Valida los datos ingresados en los campos cédula y número único.
CommandIniciarSesion	Botón que permite comenzar el inicio de sesión.
FormLogin	Contenedor de los controles anteriores.

Tabla 3.7 Descripción de los atributos de la clase *IniciarSesion*

- Clase *Estudiante*, los atributos de esta clase son mostrados en la Figura 3.9, y son descritos en la Tabla 3.8.

Estudiante	
- servicioSIEM	: Service
- ticket	: TicketDeAutenticacion
- estudiante	: Estudiante
- ImageSIEM	: Image
- LabelNombreEstudiante	: Label
- TextViewNombreEstudiante	: TextView
- LabelConsultas	: Label
- LinkHorario	: Link
- LinkCalificaciones	: Link
- LabelDatosPersonales	: Label
- TextViewNumeroCedulaEstudiante	: TextView
- LabelNumeroCedulaEstudiante	: Label
- LabelNumeroUnicoEstudiante	: Label
- TextViewNumeroUnicoEstudiante	: TextView
- LabelIdMilitarEstudiante	: Label
- TextViewIdMilitarEstudiante	: TextView
- LableCorreoEstudiante	: Label
- TextViewCorreoEstudiante	: TextView
- FormEstudiante	: Form

Figura 3.9 Atributos de la clase *Estudiante*

Nombre	Descripción
servicioSIEM	Permite comunicarse con el servicio Web.
<i>ticket</i>	Almacena el <i>ticket</i> proporcionado por el servicio Web.
estudiante	Utilizado para almacenar la información del estudiante proporcionada por el servicio Web.
LabelNombreEstudiante	Muestra el texto "BIENVENIDO A SIEM".
TextViewNombreEstudiante	Muestra el nombre del estudiante.
LabelConsultas	Muestra el texto "CONSULTAS".
LinkHorario	Proporciona un vínculo que dirige a la página Web de horario.
LinkCalificaciones	Proporciona un vínculo que dirige a la página Web de calificaciones.
LabelDatosPersonales	Muestra el texto "DATOS PERSONALES".
TextViewNumeroCedulaEstudiante	Muestra el número de cédula del estudiante.
LabelNumeroCedulaEstudiante	Muestra el texto "Número de Cédula".
LabelNumeroUnicoEstudiante	Muestra el texto "Número Único".
TextViewNumeroUnicoEstudiante	Muestra el número único del estudiante.
LabelIdMilitarEstudiante	Muestra el texto "Id. Militar".
TextViewIdMilitarEstudiante	Muestra el número de la identificación militar del estudiante.
LabelCorreoEstudiante	Muestra el texto "Correo Electrónico".
TextViewCorreoEstudiante	Muestra el correo electrónico del estudiante.
FormEstudiante	Contenedor de todos los controles descritos.

Tabla 3.8 Descripción de los atributos de la clase *Estudiante*

- Clase *Materia*, los atributos se muestran en la Figura 3.10.

Materia	
- servicioSIEM	: Service
- ticket	: TicketDeAutenticacion
- ImageSIEM	: Image
- detalleMateria	: DetalleMateria
- TextViewMateria	: TextView
- LabelCodigoMateria	: Label
- TextViewCodigoMateria	: TextView
- LabelProfesor	: Label
- TextViewProfesor	: TextView
- LabelAula	: Label
- TextViewAula	: TextView
- LabelCategoria	: Label
- TextViewCategoria	: TextView
- LabelParalelo	: Label
- TextViewParalelo	: TextView
- LabelNumeroCreditos	: Label
- TextViewNumeroCreditos	: TextView
- LinkAtras	: Link
- LinkPrincipal	: Link

Figura 3.10 Atributos de la clase *Materia*

Nombre	Descripción
servicioSIEM	Permite acceso al servicio Web.
<i>ticket</i>	Almacena el <i>ticket</i> proporcionado por el servicio Web.
ImageSIEM	Imagen que se despliega en la parte superior de la página.
detalleMateria	Utilizado para almacenar la información detallada de una materia.
TextViewMateria	Muestra el nombre de la materia.
LabelCodigoMateria	Muestra el texto "Código de Materia:".
LabelProfesor	Muestra el texto "Profesor:".
TextViewProfesor	Muestra el nombre del profesor de la materia.
LabelAula	Muestra el texto "Aula:".
TextViewAula	Muestra la identificación del aula.
LabelCategoria	Muestra el texto "Categoría:".
TextViewCategoria	Muestra la categoría a la que pertenece la materia.
LabelParalelo	Muestra el texto "Paralelo:".
TextViewParalelo	Muestra el paralelo.
LabelNumeroCreditos	Muestra el texto "Número de Créditos".
TextViewNumeroCreditos	Muestra el número de créditos correspondientes a la materia.
LinkAtras	Proporciona un vínculo que dirige a la página previa.
LinkPrincipal	Proporciona un vínculo que dirige a la página principal.

Tabla 3.9 Descripción de los atributos de la clase *Materia*

- Clase *Horario*, la Figura 3.11 muestra los atributos de la clase *Horario*. Se da una breve descripción de los atributos en la Tabla 3.10.

Horario	
- servicioSIEM	: Service
- ticket	: TicketDeAutenticacion
- listaHorario	: List<HoraClase>
- horarioPorDias	: List<List<HoraClase>>
- listaControlesHorario	: List<HorarioUC>
- ImageSIEM	: Image
- LinkPrincipal	: Link
- LabelHorario	: Label
- PanelHorario	: Panel
- LinkPrincipal2	: Link

Figura 3.11 Atributos de la Clase *Horario*

Nombre	Descripción
servicioSIEM	Permite comunicarse con el servicio Web.
<i>ticket</i>	Almacena el <i>ticket</i> proporcionado por el servicio Web.
listaHorario	Utilizado para almacenar el horario de clases consultado.
horarioPorDias	Almacena el horario de clases agrupado por días de la semana.
ImageSIEM	Imagen que se despliega en la parte superior de la página de horario.
LinkPrincipal	Proporciona un vínculo que dirige a la página Web principal.
LabelHorario	Muestra el texto "HORARIO DE CLASES".
listaControlesHorario	Controles que permiten mostrar el horario organizado por días de la semana.
LinkPrincipal2	Proporciona un vínculo que dirige a la página Web principal o <i>Home</i> .

Tabla 3.10 Descripción de los atributos de la clase *Horario*

- Clase *Calificaciones*, la clase *Calificaciones* contiene los atributos mostrados en la Figura 3.12.

Nombre	Descripción
servicioSIEM	Sirve para establecer comunicación con el servicio Web.
<i>ticket</i>	Almacena el <i>ticket</i> de autenticación proporcionado por el servicio Web.
listaCalificaciones	Utilizado para almacenar las calificaciones del estudiante obtenidas del servicio Web <i>SIEM</i> .
LinkPrincipal	Proporciona un vínculo que dirige a la página principal del sitio Web.
labelCalificaciones	Muestra el texto "CALIFICACIONES".
PanelCalificaciones	Contenedor de los controles de usuario <i>CalificacionUC</i> que permiten mostrar las calificaciones del Estudiante.
LinkPrincipal	Proporciona un vínculo que dirige a la página principal del sitio Web.

Tabla 3.11 Descripción de los atributos de la clase *Calificaciones*

Se hace una breve descripción de los atributos de la clase *Calificaciones* en la Tabla 3.11.

Calificaciones	
- servicioSIEM	: Service
- ticket	: TicketDeAutenticacion
- ImageSIEM	: Image
- listaCalificaciones	: Calificacion[]
- LinkPrincipal	: Link
- LabelCalificaciones	: Label
- PanelCalificaciones	: Panel
- LinkPrincipal2	: Link

Figura 3.12 Atributos de la Clase *Calificaciones*

- Clase *CalificacionUC*, para esta clase se tiene los atributos mostrados en la Figura 3.13. Se hace una descripción de los atributos en la Tabla 3.12.

CalificacionUC	
- _calificacion	: Calificacion = null
- _objectListCalificacion	: ObjectList = null
- _linkCalificacion	: Link = null

Figura 3.13 Atributos de la clase *CalificacionUC*

Nombre	Descripción
_calificacion	Almacena las calificaciones de una materia correspondiente a un estudiante.
_objectListCalificacion	Permite mostrar las calificaciones del estudiante.
_linkCalificacion	Vinculo que permite ir a la página de información detallada de una materia.

Tabla 3.12 Descripción de los atributos de la clase *CalificacionUC*

- Clase *HorarioUC*, los atributos de esta clase se describen en la Tabla 3.13

HorarioUC	
- _listHorario	: List<HoraClase>
- _dia	: string
- _objectListHorario	: ObjectList

Figura 3.14 Atributos de la clase *HorarioUC*

Nombre	Descripción
_listHorario	Almacena el horario de clases correspondiente a un determinado día de la semana.
_dia	Almacena el día de la semana.
_objectListHorario	Permite mostrar el horario de clases del estudiante.

Tabla 3.13 Descripción de los atributos de la clase *HorarioUC*

- Clase *HoraClaseI*, se hace una descripción de los atributos en la Tabla 3.14.

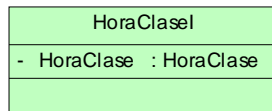


Figura 3.15 Atributos de la clase *HoraClaseI*

Nombre	Descripción
HoraClase	Contiene la información correspondiente a una hora de clase.

Tabla 3.14 Descripción de los atributos de la clase *HoraClaseI*

3.2.3 MÉTODOS

De la Tabla 3.15 a la Tabla 3.21 se realiza una descripción de los métodos utilizados por las clases de la aplicación *OnlineSIEM*.

- Clase *IniciarSesion*

Nombre	Descripción
CommandIniciarSesion_Click (object sender, EventArgs e)	<p>En este método se realizan las siguientes acciones:</p> <ul style="list-style-type: none"> • Se crea un <i>ticket</i> de autenticación para la sesión. • Se encripta el <i>ticket</i> de autenticación. • El <i>ticket</i> es añadido en una <i>cookie</i>. • Se redirige a la página de información de cuenta.
CustomValidatorInicioSesion_ServerValidate(object source, ServerValidateEventArgs args)	<p>Mediante este método se verifica si el usuario ha ingresado el número de cédula y el número único correctos.</p> <p>Si los campos se encuentran vacíos, se envía un mensaje de error (El número de cédula o número único son campos requeridos).</p> <p>Si los campos no están vacíos se hace la petición de inicio de sesión al servicio Web <i>SIEM</i>. Si el servicio devuelve un <i>ticket</i> nulo, quiere decir, que uno de los campos ingresados no es correcto y se envía un mensaje de error (Número de cédula o número único incorrectos).</p> <p>Si los campos ingresados son válidos, se almacena el <i>ticket</i> de autenticación proporcionado por el servicio Web <i>SIEM</i> para futuras peticiones.</p>

Tabla 3.15 Descripción de los métodos de la clase *IniciarSesion*

- Clase *Estudiante*

Nombre	Descripción
Page_Load(object sender, EventArgs e)	Se llama a los métodos <i>ConsultarEstudiante</i> y <i>CargarEstudiante</i> .
ConsultarEstudiante()	Este método solicita al servicio Web <i>SIEM</i> información sobre la cuenta de la sesión iniciada
CargarEstudiante()	La información obtenida desde el servicio Web <i>SIEM</i> es cargada en los controles para ser mostrada.
CommandCerrarSesion_Click(object sender, EventArgs e)	<p>Se realiza lo siguiente:</p> <ul style="list-style-type: none"> • El <i>ticket</i> de autenticación del servicio Web <i>SIEM</i> es borrado. • Se cierra la sesión. • Se redirige a la página de inicio de sesión.

Tabla 3.16 Descripción de los métodos de la clase *Estudiante*

- Clase *Materia*

Nombre	Descripción
Page_Load(object sender, EventArgs e)	Se llama a los métodos <i>ConsultarMateria</i> y <i>CargarMateria</i> .
ConsultarMateria()	Este método solicita al servicio Web <i>SIEM</i> información detallada sobre una materia.
CargarMateria()	La información obtenida desde el servicio Web <i>SIEM</i> es cargada en los controles para ser mostrada.

Tabla 3.17 Descripción de los métodos de la clase *Materia*

- Clase *Horario*

Nombre	Descripción
Page_Load(object sender, EventArgs e)	Se llama a los métodos <i>CrearControlesHorario</i> , <i>ConsultarHorario</i> y <i>ConstruirHorarioPorDias</i> .
CrearControlesHorario()	Se crean los controles necesarios para mostrar la información.
ConsultarHorario()	Este método solicita al servicio Web <i>SIEM</i> el horario de clases de un estudiante.
ConstruirHorarioPorDias()	Permite organizar el horario de clases por días.

Tabla 3.18 Descripción de los métodos de la clase *Horario*

- Clase *Calificaciones*

Nombre	Descripción
Page_Load(object sender, EventArgs e)	Se llama a los métodos <i>ConsultarCalificaciones</i> y <i>ConstruirPaginaCalificaciones</i> .
ConsultarCalificaciones()	Este método solicita al servicio Web <i>SIEM</i> las calificaciones de un estudiante.
ConstruirPaginaCalificaciones()	Permite cargar y mostrar las calificaciones de un estudiante.

Tabla 3.19 Descripción de los métodos de la clase *Calificaciones*

- Clase *CalificacionUC*

Nombre	Descripción
Page_Load(object sender, EventArgs e)	Se llama al método <i>ConstruirControles</i> .
ConstruirControles()	Crea los controles que se necesitan para presentar la información de una calificación de un estudiante.
DarFormatoObjetcListCalificacion()	Determina el formato de presentación del control.

Tabla 3.20 Descripción de los métodos de la clase *CalificacionUC*

- Clase *HorarioUC*

Nombre	Descripción
Page_Load(object sender, EventArgs e)	Se llama al método <i>ConstruirControl</i> .
ConstruirControl()	Se llama <i>FijarDia</i> y <i>DarFormatoObjetcListHorario</i> .
FijarDia()	Se fija el día de la semana del horario de clases.
DarFormatoObjetcListHorario()	Determina el formato de presentación del control.
ObjectListHorario_ItemSelect(object sender, ObjectListSelectEventArgs e)	Es el evento que permite redirigir una solicitud de detalle de materia.

Tabla 3.21 Descripción de los métodos de la clase *HorarioUC*

- Clase *HoraClaseI*.

Nombre	Descripción
DarValores(HoraClase horaClase)	Permite copiar los valores de una clase <i>HoraClase</i> .

Tabla 3.22 Descripción de los métodos de la clase *HoraClaseI*

3.2.4 DIAGRAMAS DE ACTIVIDAD

Los diagramas de actividad, permiten modelar el flujo interno de trabajo de los objetos de las clases de interfaz de la aplicación *Online*.

3.2.4.1 Diagrama de actividad Inicio de sesión

El diccionario para el diagrama de actividad de inicio de sesión (Figura 3.16) se determinó de la siguiente manera:

1. Validar número único y cédula: se comprueba que el número único y la cédula no sean vacíos o nulos.

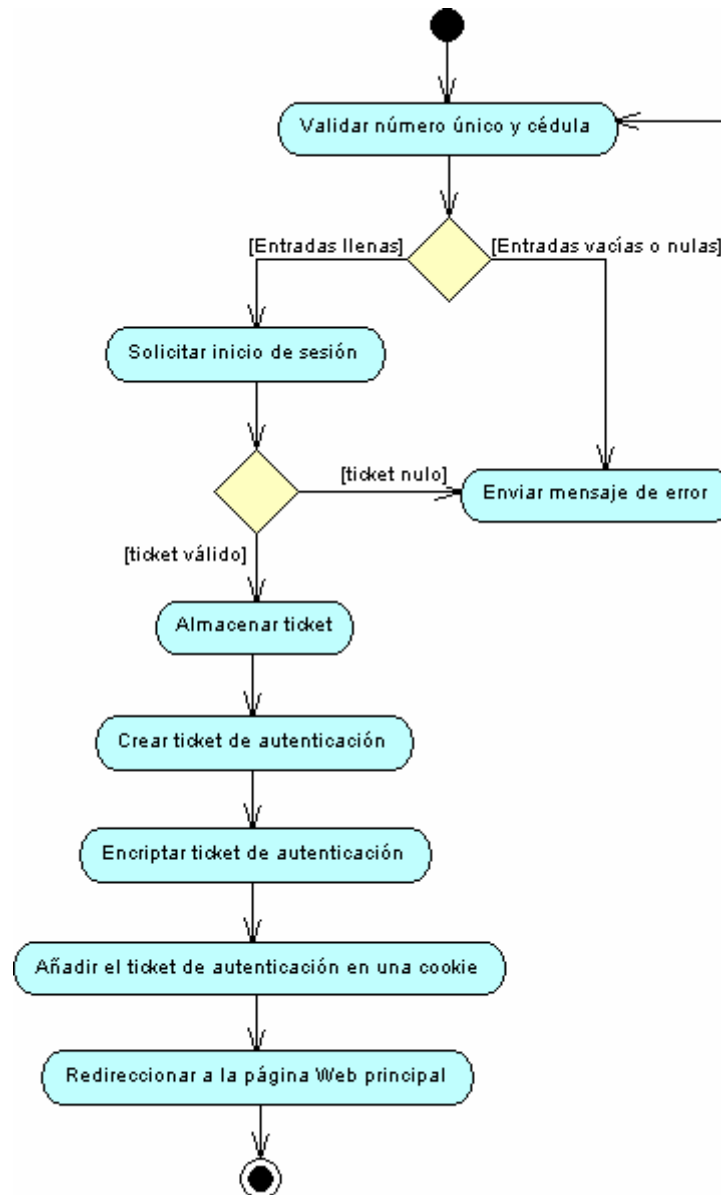


Figura 3.16 Diagrama de actividad Inicio de sesión

2. Solicitar inicio de sesión: luego de comprobar que los dos campos están llenos, se hace la solicitud de inicio de sesión al servicio Web *SIEM*.
3. Almacenar *ticket*: sucede cuando los datos ingresados corresponden a un usuario válido del sistema, entonces el servicio Web *SIEM* responde al pedido de inicio de sesión enviando un *ticket* de autenticación, el cual es almacenado para poder realizar consultas posteriores al servicio.
4. Crear *ticket* de autenticación: se genera un *ticket* de autenticación para la aplicación, que dura mientras la sesión permanezca activa.
5. Encriptar *ticket* de autenticación: el *ticket* de autenticación generado es encriptado por razones de seguridad.
6. Añadir *ticket* de autenticación: en una *cookie*, el *ticket* es añadido de manera que es enviado a través del explorador en una *cookie*.
7. Redireccionar a la página Web principal: sucede cuando el inicio de sesión ha sido exitoso.
8. Enviar mensaje de error: sucede cuando los datos de entrada son vacíos o nulos y cuando se recibe del servicio Web *SIEM* un *ticket* nulo.

3.2.4.2 Diagrama de actividad Consultar información de cuenta

El diccionario correspondiente al diagrama de actividad Consultar información de cuenta (Figura 3.17) se indica a continuación:

1. Llega *ticket* de autenticación: el *ticket* de autenticación de la sesión llega dentro de una *cookie*.
2. Desencriptar *ticket* de autenticación: la aplicación desencripta el *ticket* de autenticación.
3. Crear página Web y controles: si el *ticket* que llega en la *cookie* es de un usuario ya autenticado, la página Web de información es creada junto con los controles necesarios para presentar la información.
4. Consultar información de cuenta: se solicita al servicio Web *SIEM* información sobre la cuenta activa.
5. Cargar la información en los controles de la página: la información recibida se llena en los controles de la página Web.
6. Enviar la página Web: la página con la información se muestra al usuario.

7. Redireccionar a la página de inicio de sesión: sucede cuando el *ticket* que llega en la *cookie* no corresponde a un usuario ya autenticado.

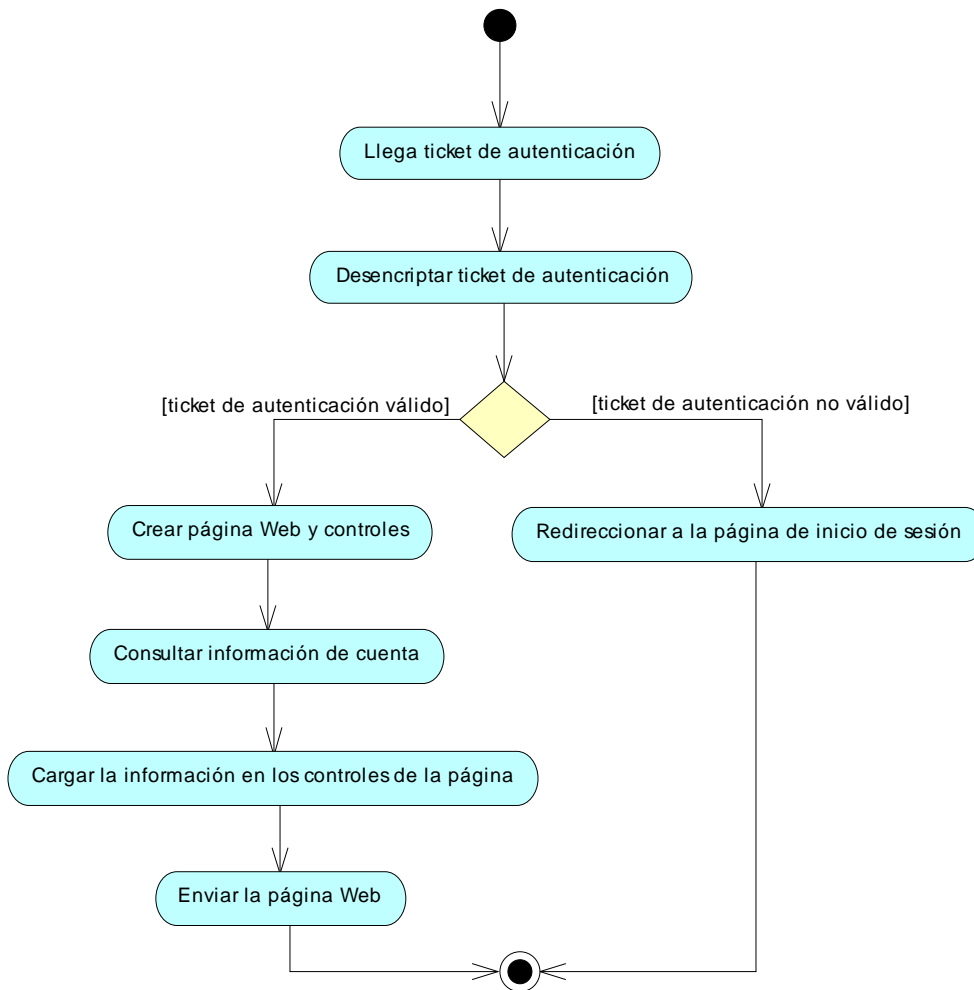


Figura 3.17 Diagrama de actividad Consultar información de cuenta

3.2.4.3 Diagrama de actividad Consultar horario

El diagrama de actividad Consultar horario se muestra en la Figura 3.18, y su diccionario de actividades se establece de la siguiente manera:

1. Llega *ticket* de autenticación: el *ticket* de autenticación de la sesión llega dentro de una *cookie*.
2. Desencriptar *ticket* de autenticación: la aplicación desencripta el *ticket* de autenticación.

3. Crear página Web: si el *ticket* que llega en la *cookie* es de un usuario ya autenticado, la página Web de información es creada.
4. Crear controles de horario: se crean los controles necesarios para presentar la información.

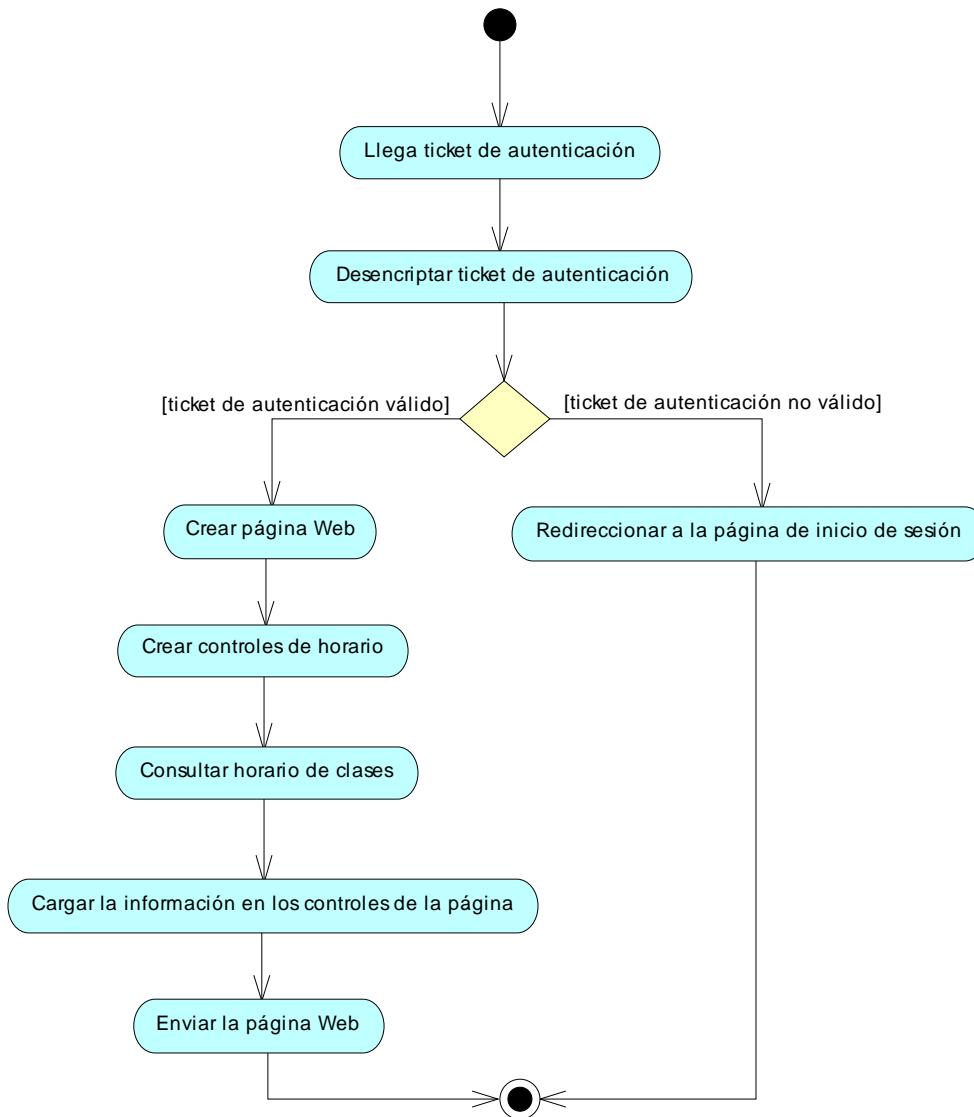


Figura 3.18 Diagrama de actividad Consultar horario

5. Consultar horario de clases: se solicita al servicio Web *SIEM* información sobre el horario de clases de un determinado estudiante.
6. Cargar la información en los controles de la página: la información recibida se llena en los controles de la página Web.
7. Enviar la página Web: la página con la información se muestra al usuario.

8. Redireccionar a la página de inicio de sesión: sucede cuando el *ticket* que llega en la *cookie* no corresponde a un usuario ya autenticado.

3.2.4.4 Diagrama de actividad Consultar calificaciones

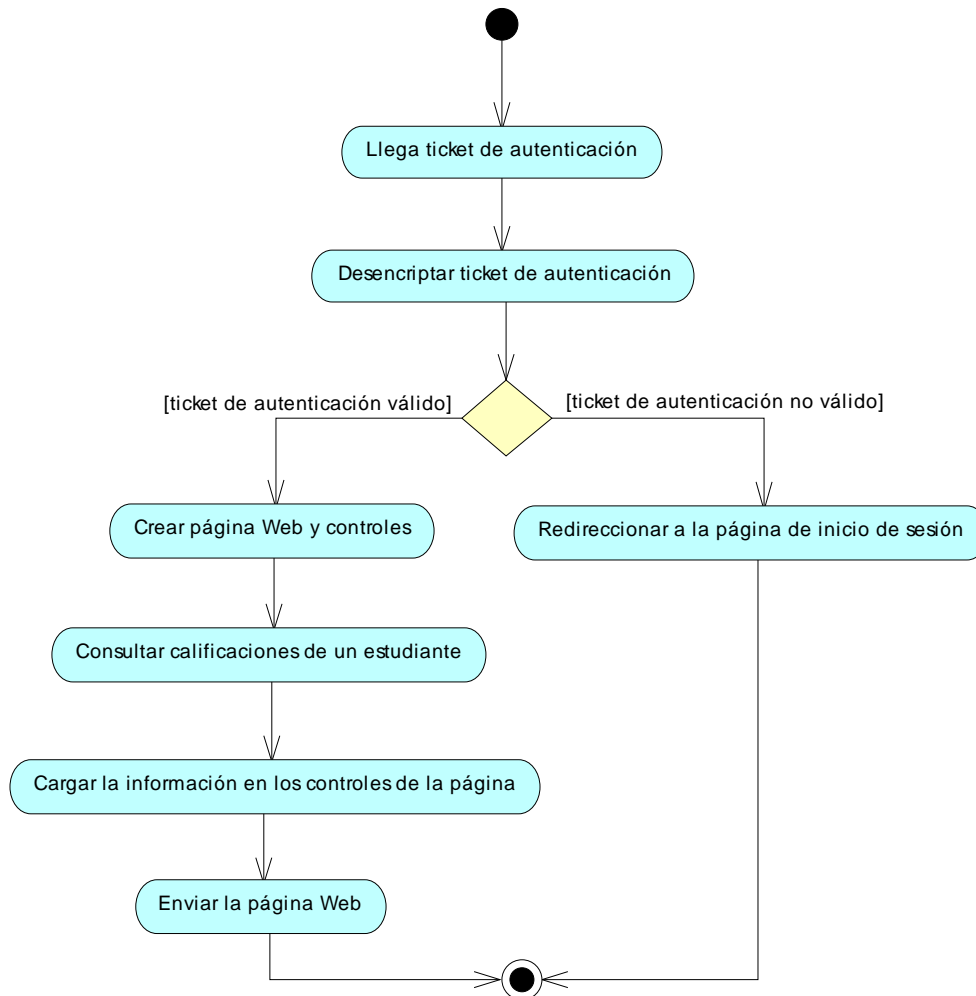


Figura 3.19 Diagrama de actividad Consultar calificaciones

El diccionario de actividades del diagrama de actividad Consultar calificaciones (Figura 3.19), queda establecido de la siguiente manera:

1. Llega *ticket* de autenticación: el *ticket* de autenticación de la sesión llega dentro de una *cookie*.
2. Desencriptar *ticket* de autenticación: la aplicación desencripta el *ticket* de autenticación.

3. Crear página Web y controles: si el *ticket* que llega en la *cookie* es de un usuario ya autenticado, la página Web de información es creada junto con los controles necesarios para presentar la información.
4. Consultar calificaciones de un estudiante: se solicita al servicio Web *SIEM* información sobre las calificaciones de un determinado estudiante.
5. Cargar la información en los controles de la página: la información recibida se llena en los controles de la página Web.
6. Enviar la página Web: la página con la información se muestra al usuario.
7. Redireccionar a la página de inicio de sesión: sucede cuando el *ticket* que llega en la *cookie* no corresponde a un usuario ya autenticado.

3.2.4.5 Diagrama de actividad Consultar información de materia

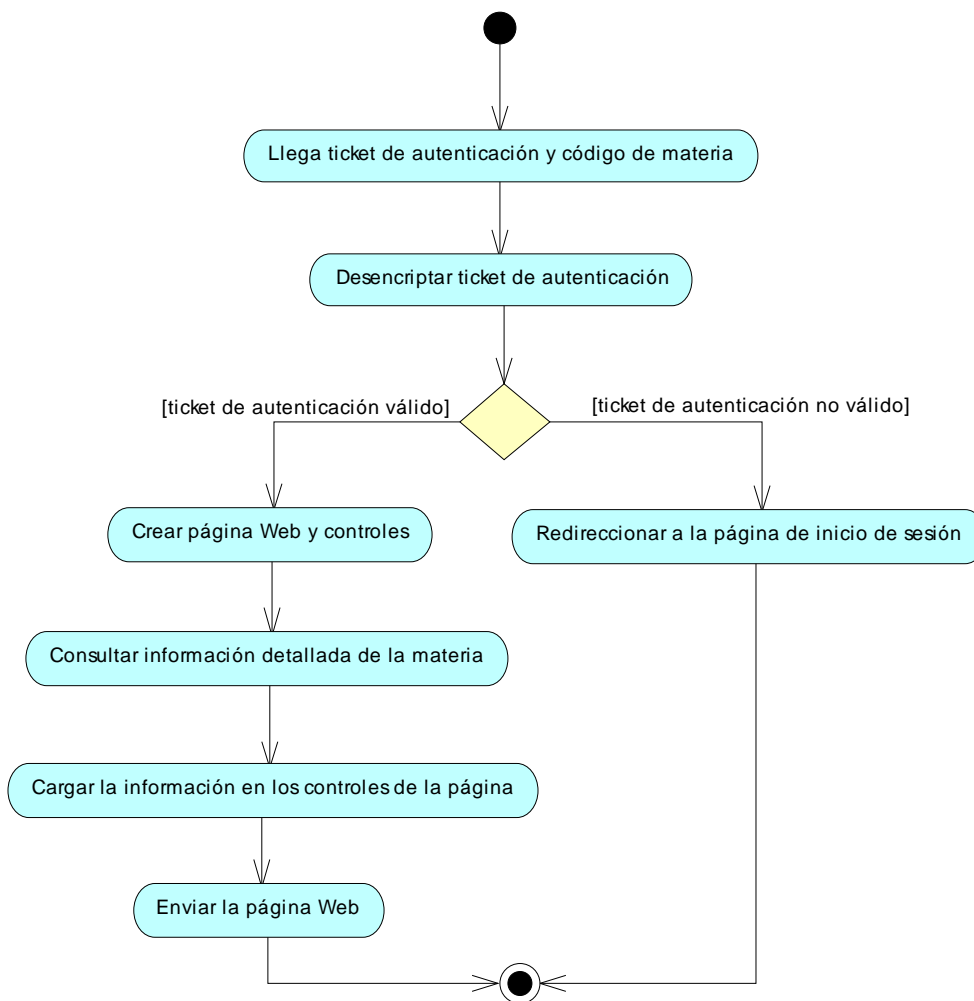


Figura 3.20 Diagrama de actividad Consultar información de materia

El diccionario de actividades del diagrama de actividad Consultar información de materia (Figura 3.20) queda establecido de la siguiente manera:

1. Llega *ticket* de autenticación: el *ticket* de autenticación de la sesión llega dentro de una *cookie*.
2. Desencriptar *ticket* de autenticación: la aplicación desencripta el *ticket* de autenticación.
3. Crear página Web y controles: si el *ticket* que llega en la *cookie* es de un usuario ya autenticado, la página Web de información es creada junto con los controles necesarios para presentar la información.
4. Consultar información detallada de la materia: se solicita al servicio Web *SIEM* información detallada de la materia, para esto, es necesario enviar el código de la materia que llega en la cadena de consulta de la página.
5. Cargar la información en los controles de la página: la información recibida se llena en los controles de la página Web.
6. Enviar la página Web: la página con la información se muestra al usuario.
7. Redireccionar a la página de inicio de sesión: sucede cuando el *ticket* que llega en la *cookie* no corresponde a un usuario ya autenticado.

3.2.4.6 Diagrama de actividad Cerrar sesión

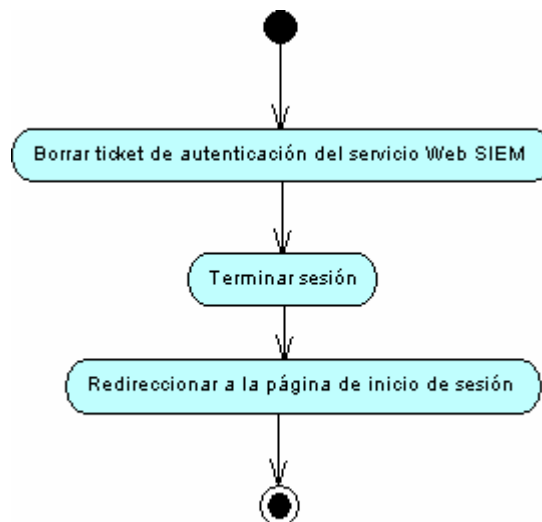


Figura 3.21 Diagrama de actividad Cerrar sesión

El diccionario de actividades del diagrama de actividad Cerrar sesión (Figura 3.21) queda establecido de la siguiente manera:

1. Borrar *ticket* de autenticación del servicio Web *SIEM*, el *ticket* de autenticación proporcionado por el servicio Web *SIEM* es borrado.
2. Terminar sesión, se termina la sesión iniciada con la aplicación Web.
3. Redireccionar a la página de inicio de sesión, se muestra la página de inicio de sesión.

3.2.5 DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia, permiten indicar el flujo de acciones que ocurren entre los diferentes objetos de las clases de interfaz de la aplicación *OnlineSIEM* a través del tiempo. Se presenta un diagrama de secuencia por cada caso de uso del sistema.

3.2.5.1 Diagrama de secuencia Inicio de sesión

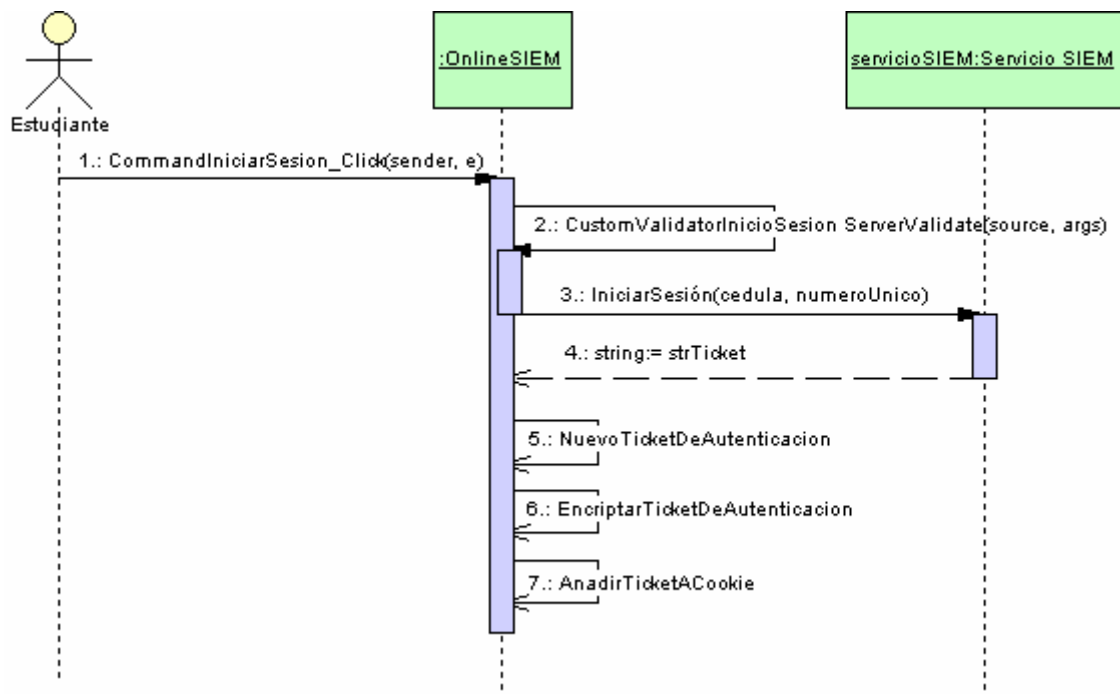


Figura 3.22 Diagrama de secuencia Inicio de sesión

El diagrama de la Figura 3.22 muestra la secuencia que se sigue cuando se inicia una sesión con el *servicio Web SIEM*, utilizando la aplicación *OnlineSIEM*.

El proceso comienza cuando un estudiante inicia la aplicación, ingresa su cédula y número único y da clic en el botón iniciar sesión de la aplicación *OnlineSIEM*

La aplicación *OnlineSIEM* controla el evento del botón, verificando primero si el usuario ha ingresado los campos requeridos (cédula y número único). Si uno o los dos campos son vacíos o nulos, se envía al usuario un mensaje informándole que los campos son requeridos.

Si ambos campos están llenos, la aplicación *OnlineSIEM* solicita *IniciarSesión* al *servicio Web SIEM*, donde se pasan el número único y la cédula del estudiante.

La aplicación espera por la respuesta del servicio. Si el servicio retorna nulo, se envía un mensaje de error, indicando al usuario que la cédula o número único ingresados son incorrectos.

Si la respuesta desde el servicio es un *ticket* de autenticación, éste es almacenado en una variable de sesión para realizar peticiones posteriores al *servicio Web*. Una vez, que se tiene una sesión exitosa con el *servicio Web SIEM*, la aplicación genera un *ticket* de autenticación para la sesión activa y luego este *ticket* es encriptado y añadido en una *cookie*.

Finalmente el usuario ha iniciado una sesión correctamente y puede realizar consultas al *servicio Web*.

3.2.5.2 Diagrama de secuencia Consultar información de cuenta

El diagrama de la Figura 3.23 muestra la secuencia seguida para obtener información referente a la cuenta de un estudiante, desde la aplicación *OnlineSIEM*, luego de haber iniciado una sesión correctamente con el *servicio Web SIEM*.

Luego de haber iniciado la sesión correctamente, la aplicación *OnlineSIEM* solicita *ObtenerEstudiante* al objeto *servicioSIEM*. El *servicio Web SIEM* devuelve a la aplicación la información sobre la cuenta.

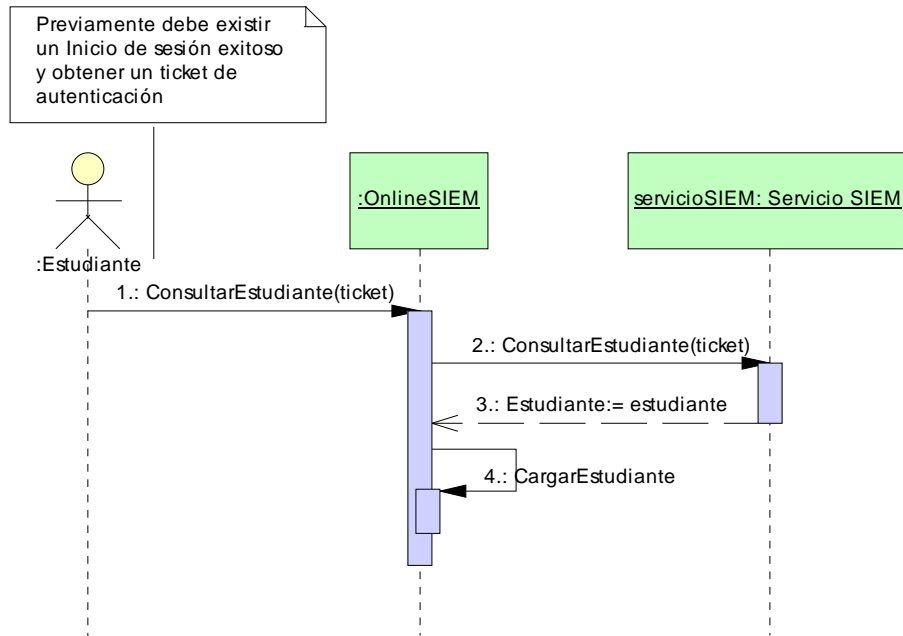


Figura 3.23 Diagrama de secuencia Consultar información de cuenta

Luego de obtener la información la aplicación llama a *CargarEstudiante*, para colocar la información en los controles de la página Web.

Por último la aplicación muestra al usuario la página con la información de la cuenta del estudiante.

3.2.5.3 Diagrama de secuencia Consultar horario

El diagrama de la Figura 3.24 muestra la secuencia seguida para conocer el horario de clases, luego de haber iniciado una sesión exitosa y obtenido un *ticket* de autenticación válido.

El estudiante da clic sobre el vínculo *Horario de clases...*, el cual conduce a la página Web del horario de clases del estudiante.

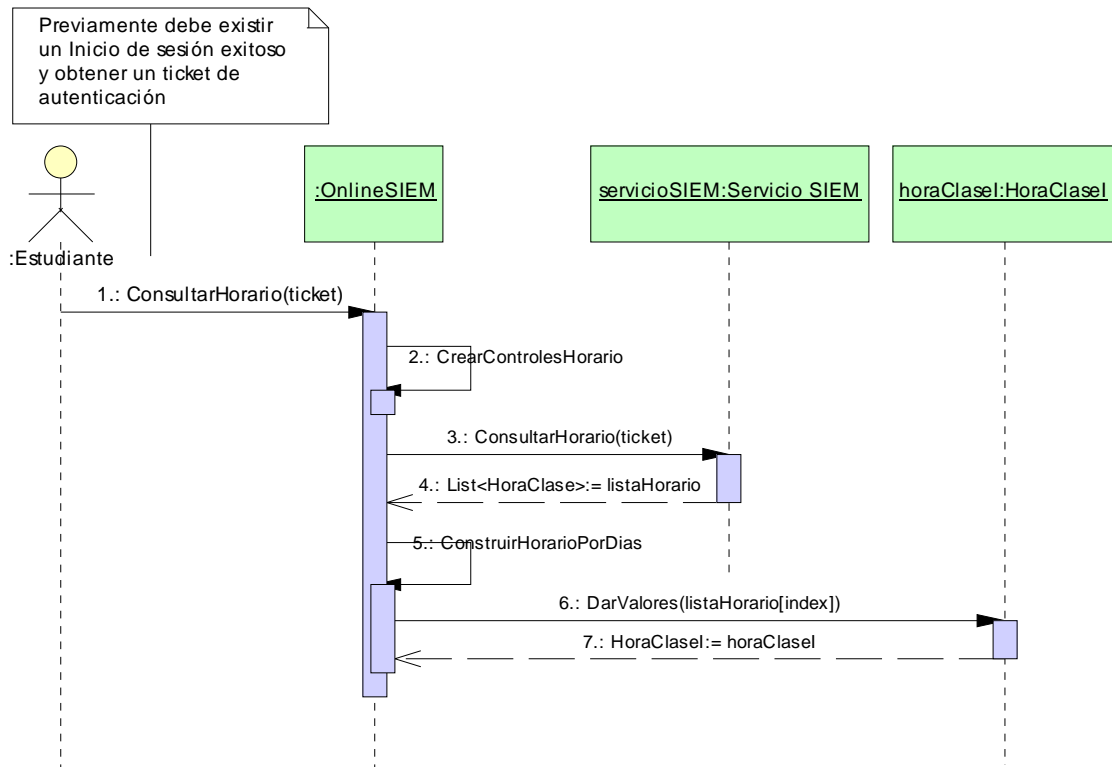


Figura 3.24 Diagrama de secuencia Consultar horario

Lo primero que hace la aplicación es crear la página Web y los controles, luego solicita *ConsultarHorario* al objeto *servicioSIEM*, enviándole el *ticket* de autenticación proporcionado en el inicio de sesión. El *servicio Web SIEM* devuelve el horario de clases del estudiante y es almacenado en una lista de objetos *HoraClase*.

Luego de obtener la información la aplicación llama a *ConstruirHorarioPorDias*, donde la información es organizada por días. Seguidamente se solicita *DarValores* a la clase *HoraClaseI*, la cual asigna la información en los objetos correspondientes.

Finalmente la aplicación muestra al usuario la página con el horario de clases.

3.2.5.4 Diagrama de secuencia Consultar calificaciones

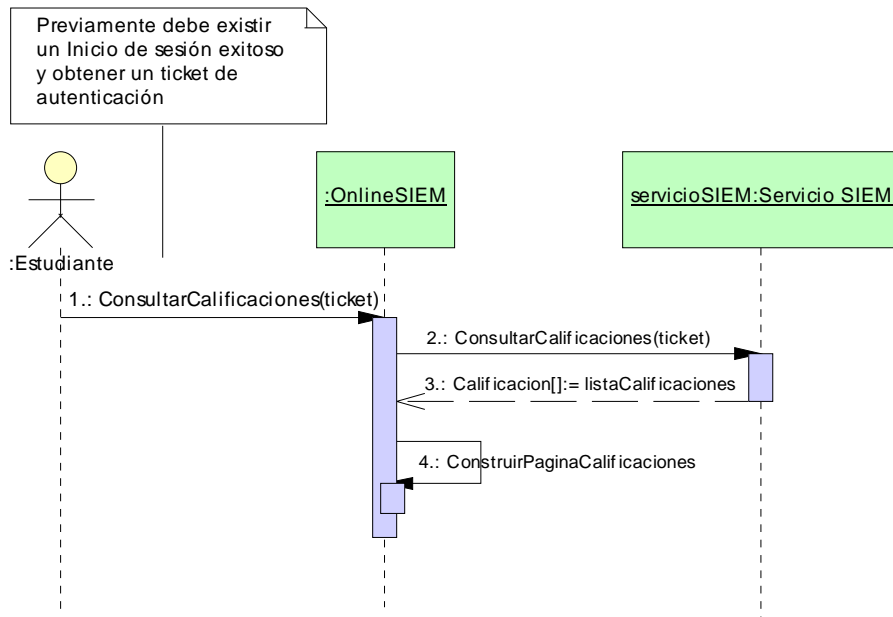


Figura 3.25 Diagrama de secuencia Consultar calificaciones

El diagrama de la Figura 3.25 muestra la secuencia seguida para conocer las calificaciones de un estudiante, luego de haber iniciado una sesión exitosa y obtenido un *ticket* de autenticación válido.

El estudiante da clic sobre el vínculo *Calificaciones...*, el cual conduce a la página Web de calificaciones del estudiante.

Lo primero que hace la aplicación es crear la página Web y los controles, luego solicita *ConsultarCalificaciones* al objeto *servicioSIEM*, enviándole el *ticket* de autenticación ya proporcionado con el inicio de sesión. El *servicio Web SIEM* devuelve las calificaciones del estudiante y es almacenado en un arreglo de objetos *Calificacion*.

Luego de obtener la información, la aplicación llama a *ConstruirPaginaCalificaciones*, donde los controles se llenan con la información de las calificaciones del estudiante.

Finalmente la aplicación muestra al usuario la página con las calificaciones obtenidas.

3.2.5.5 Diagrama de secuencia Consultar información de materia

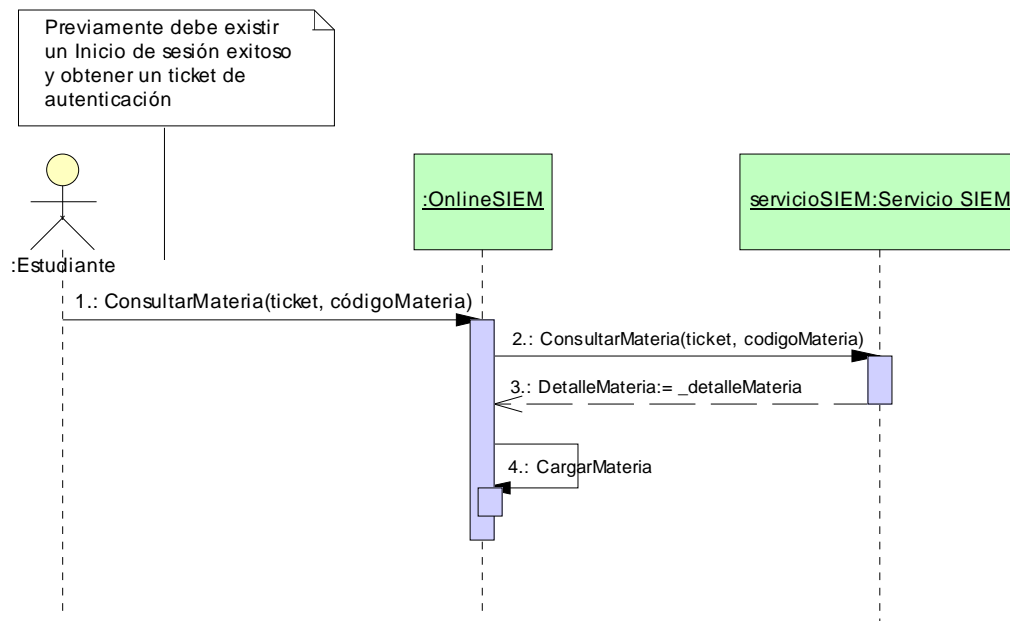


Figura 3.26 Diagrama de secuencia Consultar información de materia

El diagrama de la Figura 3.26 muestra la secuencia seguida para conocer información detallada de una materia, luego de haber iniciado una sesión exitosa y obtenido un *ticket* de autenticación válido.

El estudiante da clic sobre el vínculo del nombre de la materia, el cual conduce a la página Web de información detallada de la materia.

Lo primero que hace la aplicación es crear la página Web y los controles, luego solicita *ConsultarMateria* al objeto *servicioSIEM*, enviándole el *ticket* de autenticación proporcionado con el inicio de sesión. El *servicio Web SIEM* devuelve el detalle de la materia y es almacenado en un objeto *_detalleMateria* de la clase *DetalleMateria*.

Luego de obtener la información la aplicación llama a *CargarMateria*, donde los controles se llenan con la información de la materia.

Finalmente la aplicación muestra al usuario la página con la información de la materia.

3.2.5.6 Diagrama de secuencia Cerrar sesión

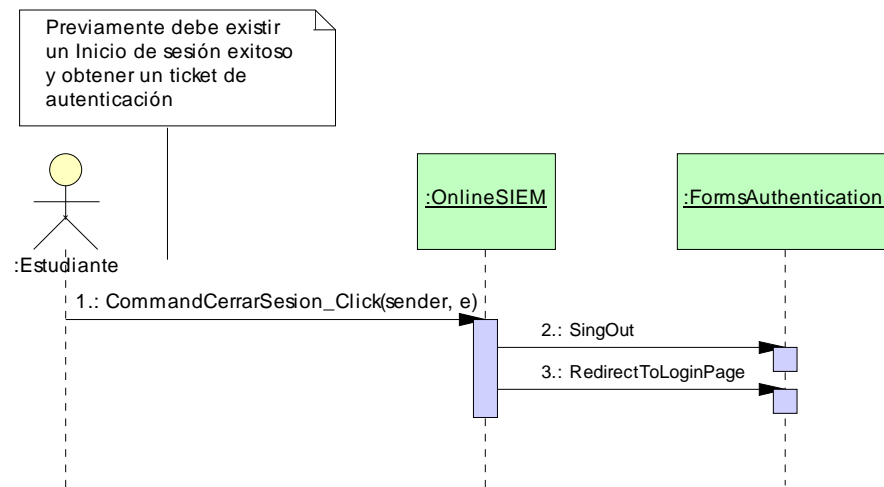


Figura 3.27 Diagrama de secuencia Cerrar sesión

El diagrama de la Figura 3.27 muestra la secuencia para cerrar la sesión con la aplicación y con el servicio Web.

El estudiante da clic sobre el botón *Cerrar Sesión*, que se encuentra en la página Web de información de cuenta. La aplicación *OnlineSIEM* controla el evento, en donde, se elimina el *ticket* de autenticación del servicio Web *SIEM*, seguidamente, se llama a *SigOut* y *RedirectToLoginPage* de la clase *FormsAuthentication*, para terminar la sesión con la aplicación y abrir la página de inicio de sesión, la cual se muestra al usuario.

3.3 IMPLEMENTACIÓN

Para la implementación de la aplicación *Online* se utilizó *ASP.NET Mobile*, esta aplicación es básicamente un interfaz para el acceso a datos del servicio Web desarrollado en el capítulo anterior.

Para empezar se debe seguir los siguientes pasos.

1. Crear un sitio Web.
2. Agregar páginas *ASP.NET* para dispositivos móviles al sitio.

Para crear un sitio Web se sigue los siguientes pasos:

1. Iniciar el *Visual Studio 2005*, clic en Archivo, Nuevo, Sitio Web (Figura 3.28).

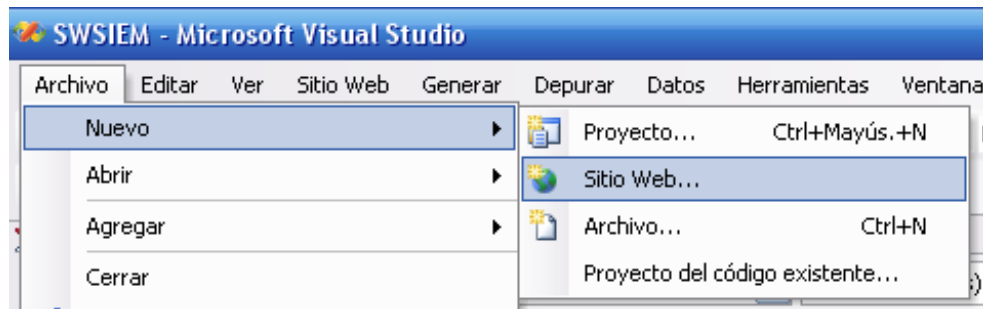


Figura 3.28 Crear nuevo Sitio Web ASP.NET

2. Seleccionar Sitio Web *ASP.NET* y dar el nombre correspondiente al sitio. Al dar clic en aceptar se creará el nuevo sitio (Figura 3.29).

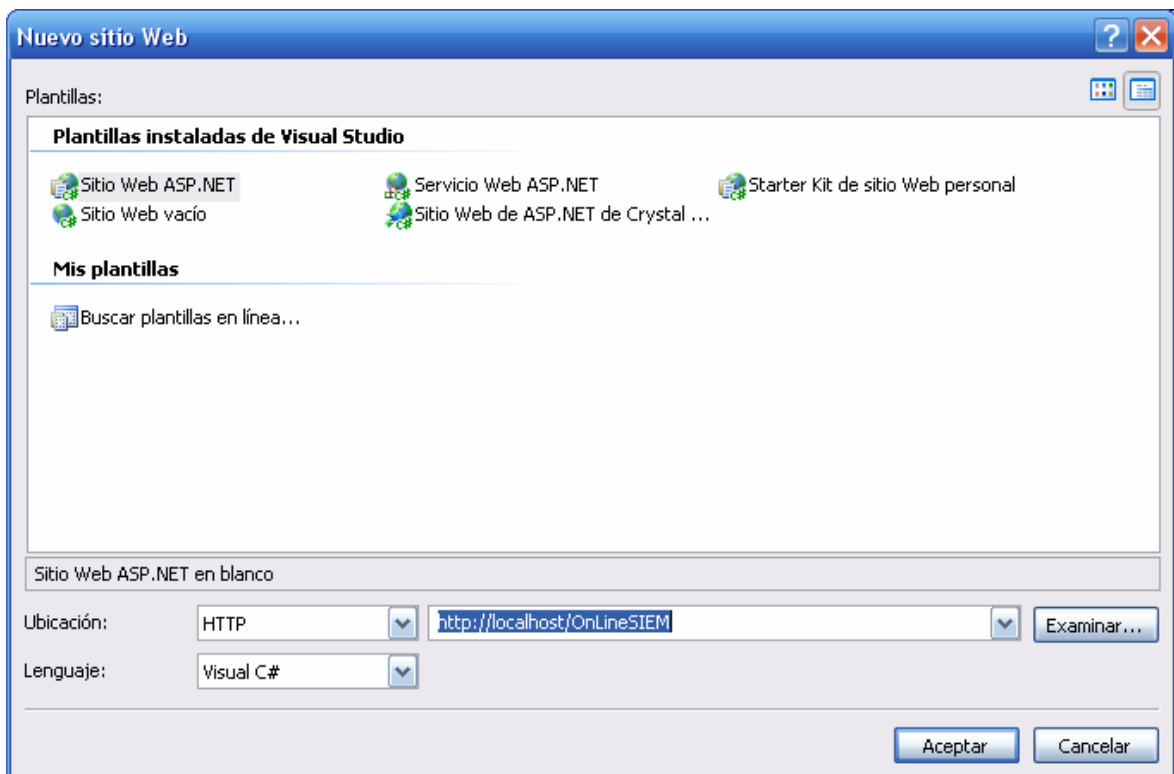


Figura 3.29 Asignar nombre y ubicación para el nuevo sitio Web

Una vez creado el sitio Web se pueden agregar páginas para dispositivos móviles, para lo cual se debe realizar lo siguiente:

1. Clic en Sitio Web, Agregar nuevo elemento o bien Ctrl+Mayús.+A (Figura 3.30)



Figura 3.30 Crear una nueva página para dispositivos móviles

2. Seleccionar Formulario *Mobile Web Forms*, dar el nombre deseado a la página Web para dispositivos móviles y clic en Aceptar (Figura 3.31).

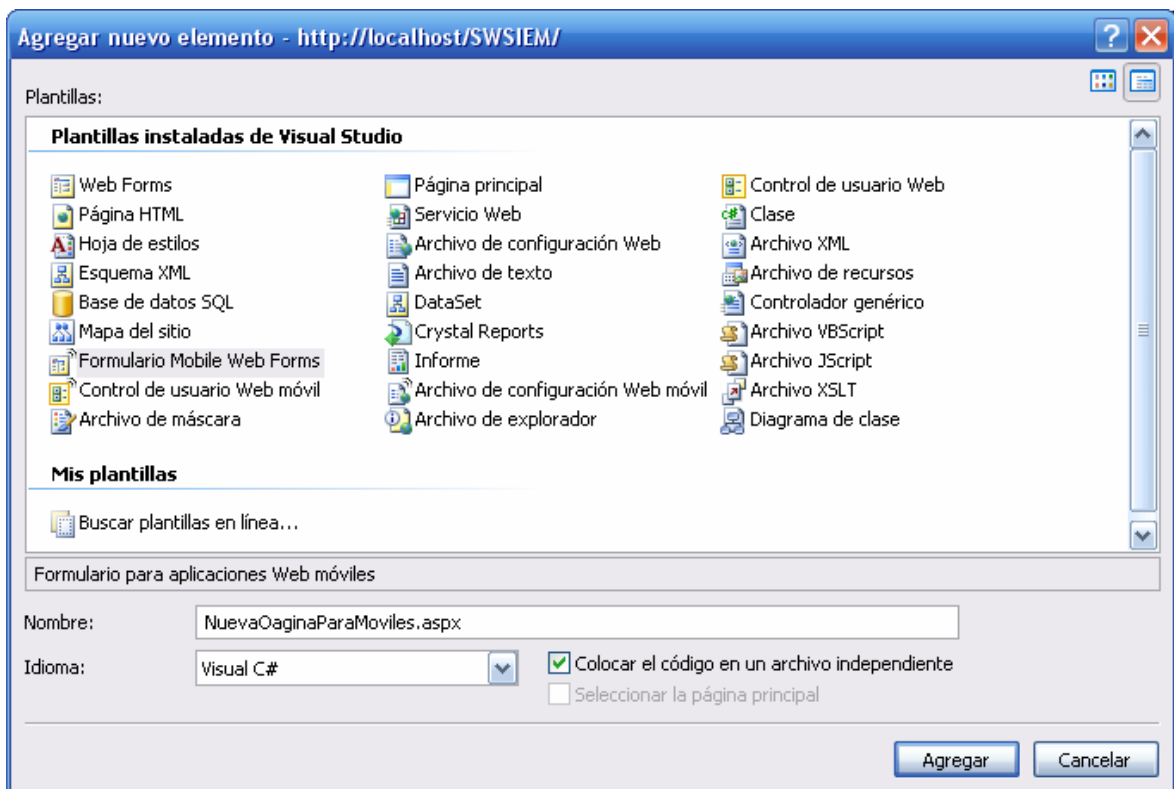


Figura 3.31 Asignar nombre a la nueva página para móviles

Este proceso se repite para crear las páginas Web móviles siguientes:

- *IniciarSesion.aspx*
- *Estudiante.aspx*

- *Horario.aspx*
- *Calificaciones.aspx*
- *Materia.aspx*

Con las páginas creadas se comienza el siguiente paso que es agregar los controles o *Mobile Controls* a cada página. Esta tarea se realiza de la siguiente forma:

1. Clic en Ver, Cuadro de herramientas o bien Ctrl + W, X, esta acción permite visualizar el conjunto de *Mobile Controls* (Figura 3.32).

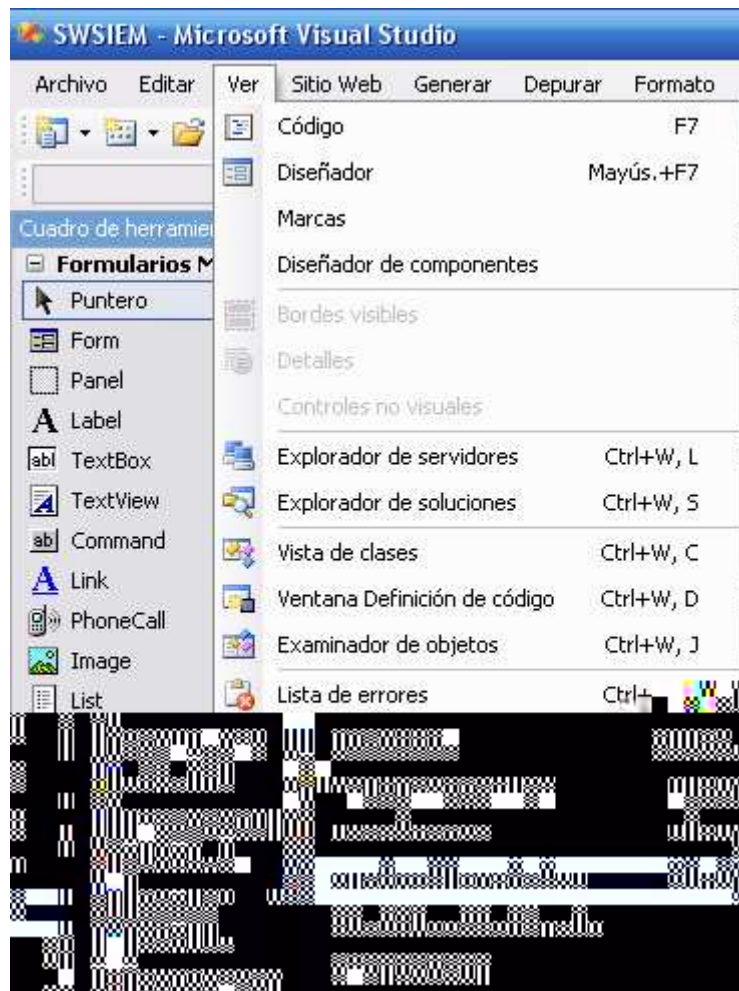


Figura 3.32 Ver cuadro de herramientas

2. Ahora se puede arrastrar los controles Web necesarios desde el cuadro de herramientas a la página Web para móviles (Figura 3.33). Según las

necesidades se podrían agregar: *Label*, *TextBox*, *Image*, *Command*, *ObjectList*, etc. cada uno con una funcionalidad diferente.

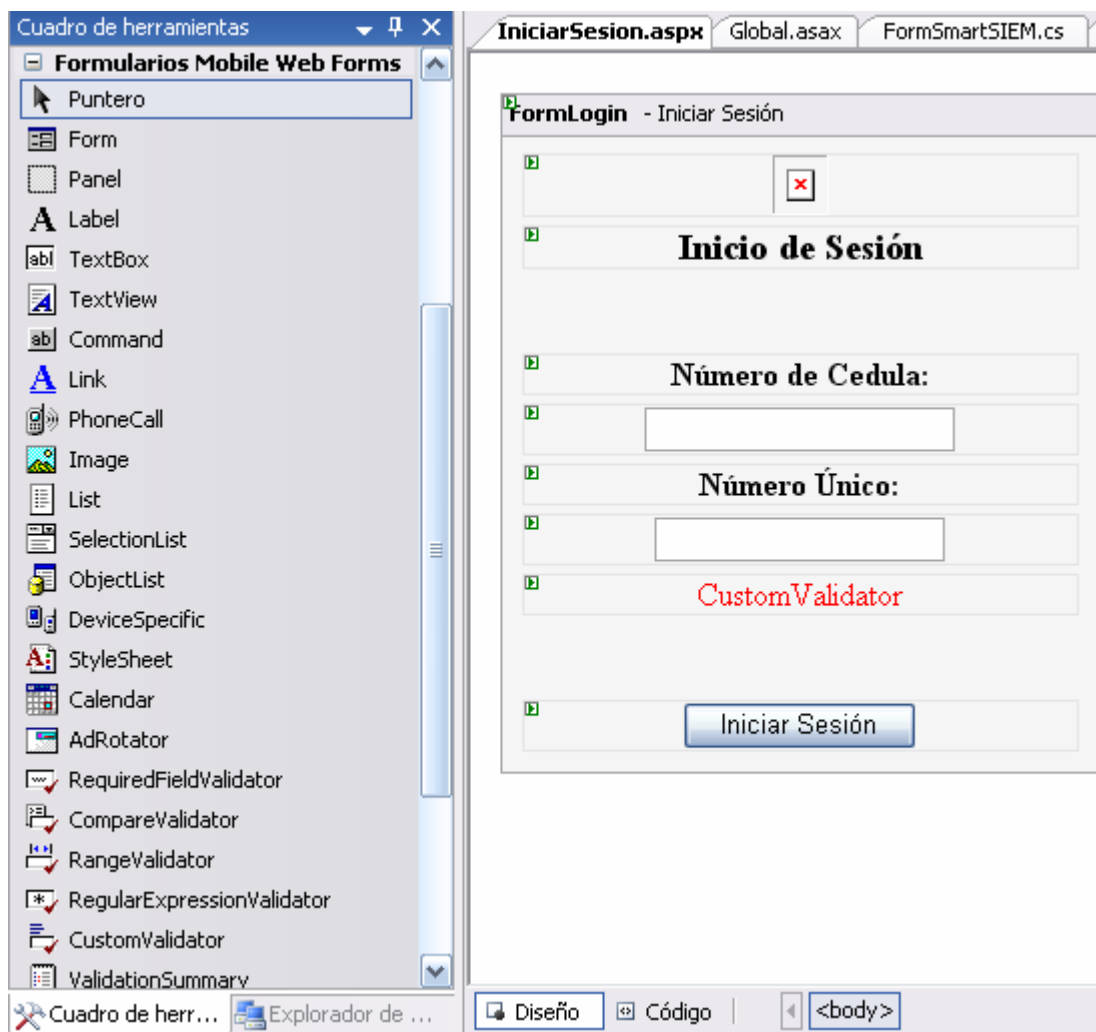


Figura 3.33 Agregar un control Web a una página Web

Este proceso se repite para cada una de las páginas del sitio, en este punto es necesario programar las acciones que cada uno de los controles tomará, por ejemplo es necesario escribir código para obtener información desde el servicio Web y mostrar estos datos en cada una de las páginas a través de sus respectivos controles Web. Este código se presenta como anexo al presente proyecto en un CD que contiene el código de las aplicaciones (Anexo D).

Como se mencionó anteriormente la aplicación toma los datos del servicio Web, esto se logra agregando una referencia Web al proyecto, lo cual se consigue de la siguiente forma:

1. Clic derecho sobre el sitio Web en el explorador de soluciones y clic Agregar referencia Web... (Figura 3.34).
2. Seleccionar el servicio Web al que se desea acceder, asignar el nombre a la referencia Web y clic en Agregar Referencia.

Con el proceso concluido es posible acceder a las funcionalidades del servicio Web desde la aplicación *Online*.

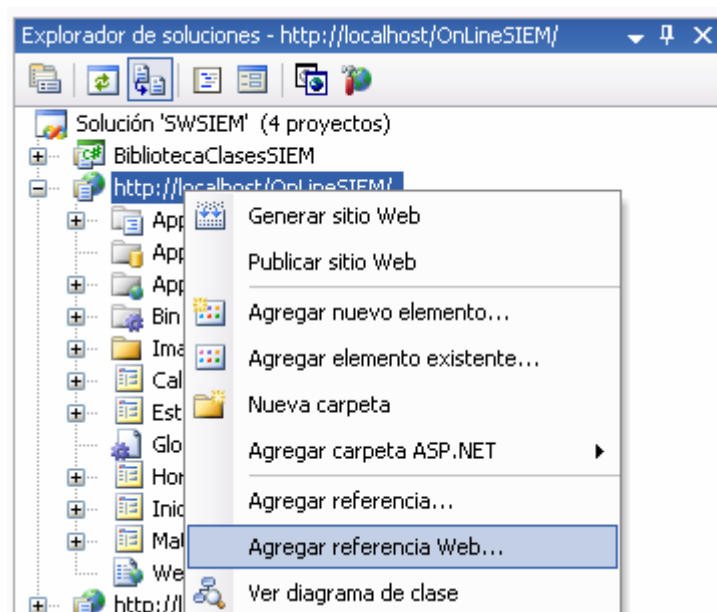


Figura 3.34 Agregar referencia Web

Con el acceso a los datos asegurado es necesario autenticar a los usuarios de la aplicación *Online*. El proceso de autenticación de la aplicación *Online* consta de los siguientes pasos:

1. El usuario ingresa su número de cédula y su número único en la página de inicio de sesión.
2. Esta información es enviada al servicio Web, para un proceso de autenticación; si el proceso es exitoso se obtiene un *ticket* de autenticación,

el mismo se guarda en una variable de sesión para que sea utilizado en futuras consultas al servicio Web.

3. Se crea un *ticket* para autenticar al usuario de la aplicación *OnlineSIEM* y se lo agrega a una *cookie*.
4. Se redirecciona a la página *Estudiante.aspx*, la cual contiene datos del estudiante.

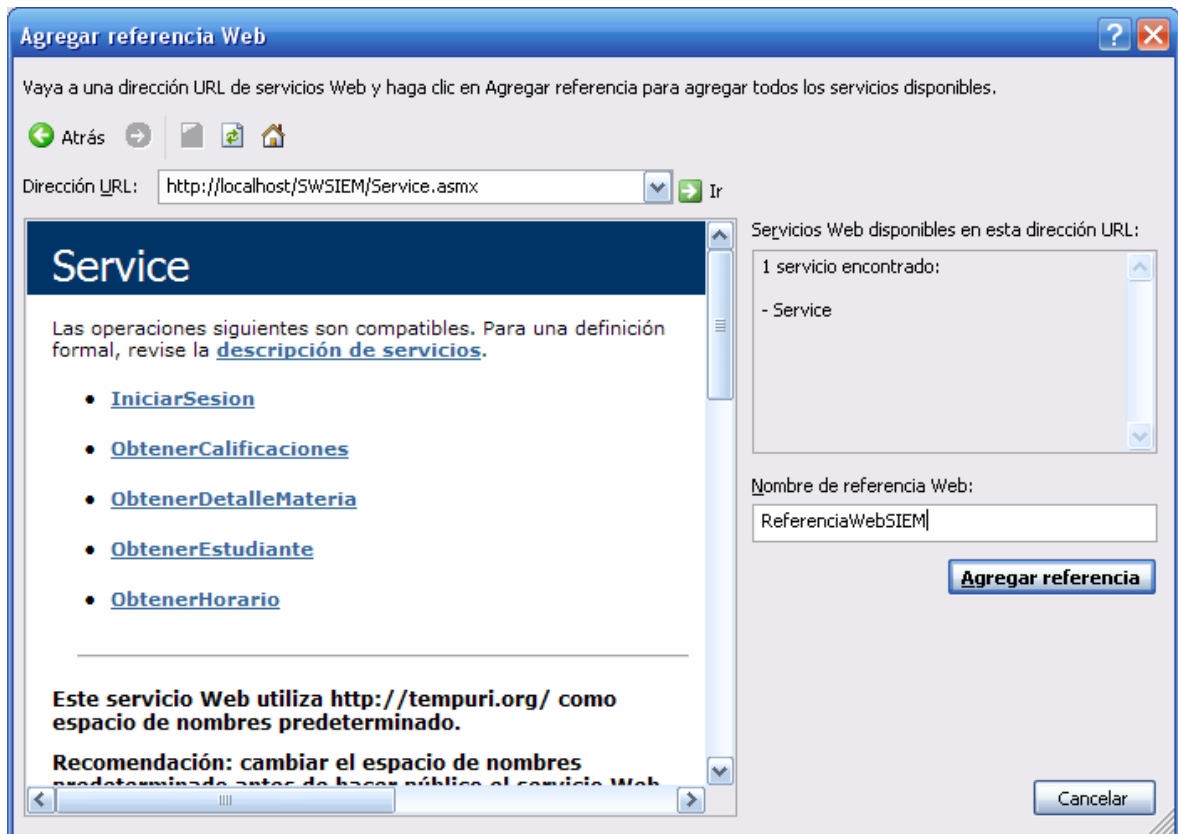


Figura 3.35 Seleccionar servicio Web

El siguiente código implementa los pasos de autenticación señalados.

```
// Servicio Web proxy
Service servicioSIEM;
// Ticket
string strTicket;

// Se obtiene el ticket desde el servicio Web
strTicket =
servicioSIEM.IniciarSesion(TextBoxNumeroCedula.Text,
TextBoxNumeroUnico.Text);

// Se guarda el ticket para futuras consultas
TicketDeAutenticacion ticketAuth = new TicketDeAutenticacion();
```

```
ticketAuth.ticket = strTicket;
Session["TicketDeAutenticacionSW"] = ticketAuth;
servicioSIEM.TicketDeAutenticacionValue = ticketAuth;

// Se crea un ticket para la aplicación OnlineSIEM
FormsAuthenticationTicket authTicket =
new FormsAuthenticationTicket(//Informacion del usuario);

// Se encripta el ticket
string encryptedTicket = FormsAuthentication.Encrypt(authTicket);

// Se agrega el ticket a una cookie
HttpCookie authCookie =
new HttpCookie(FormsAuthentication.FormsCookieName,
                encryptedTicket);
Response.Cookies.Add(authCookie);

// Se redirecciona a la página Estudiante
Response.Redirect("Estudiante.aspx");
```

Capítulo 4

Diseño e implementación de la aplicación *Smart Client*

4 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN *SMART CLIENT*

4.1 REQUERIMIENTOS

4.1.1 REQUERIMIENTOS DEL NEGOCIO

Los requerimientos del negocio para la aplicación *Smart Client* son los mismos planteados en el capítulo 2 para el servicio Web, pues los requerimientos del negocio explican cómo se informa el estudiante de sus horarios y calificaciones en la actualidad, sin el uso del sistema *SIEM*.

4.1.2 REQUERIMIENTOS DEL SISTEMA

4.1.2.1 Lista de actores

Estudiante, persona perteneciente a la Carrera de Electrónica y Redes de Información, que usando un dispositivo móvil del tipo *Pocket PC*, accede al *Sistema de Información Estudiantil Móvil Smart (SmartSIEM)*.

Servicio Web *SIEM*, aplicación Web que permite obtener la información requerida.

4.1.2.2 Lista de casos de uso

Los casos de uso son los mismos que los presentados en el capítulo anterior. Esto se debe a que tanto el servicio Web, la aplicación *Online* y la aplicación *Smart Client* deben cumplir los mismos requerimientos, el informar al estudiante de horarios y calificaciones.

CU1: Iniciar sesión, el estudiante inicia una sesión en *SIEM*, para esto utiliza su número de cédula y su número único.

CU2: Consultar información de cuenta, el estudiante puede revisar la información de su sesión.

CU3: Consultar horario, el estudiante consulta en *SIEM* el horario de clases.

CU4: Consultar calificaciones, el estudiante consulta en *SIEM* sus calificaciones académicas.

CU5: Consultar información de materia, se obtiene información como nombre, número de créditos, etc. de las materias tomadas por el estudiante

CU6: Cerrar sesión, el estudiante termina la sesión con *SIEM*.

4.1.2.3 Diagrama de casos de uso del sistema

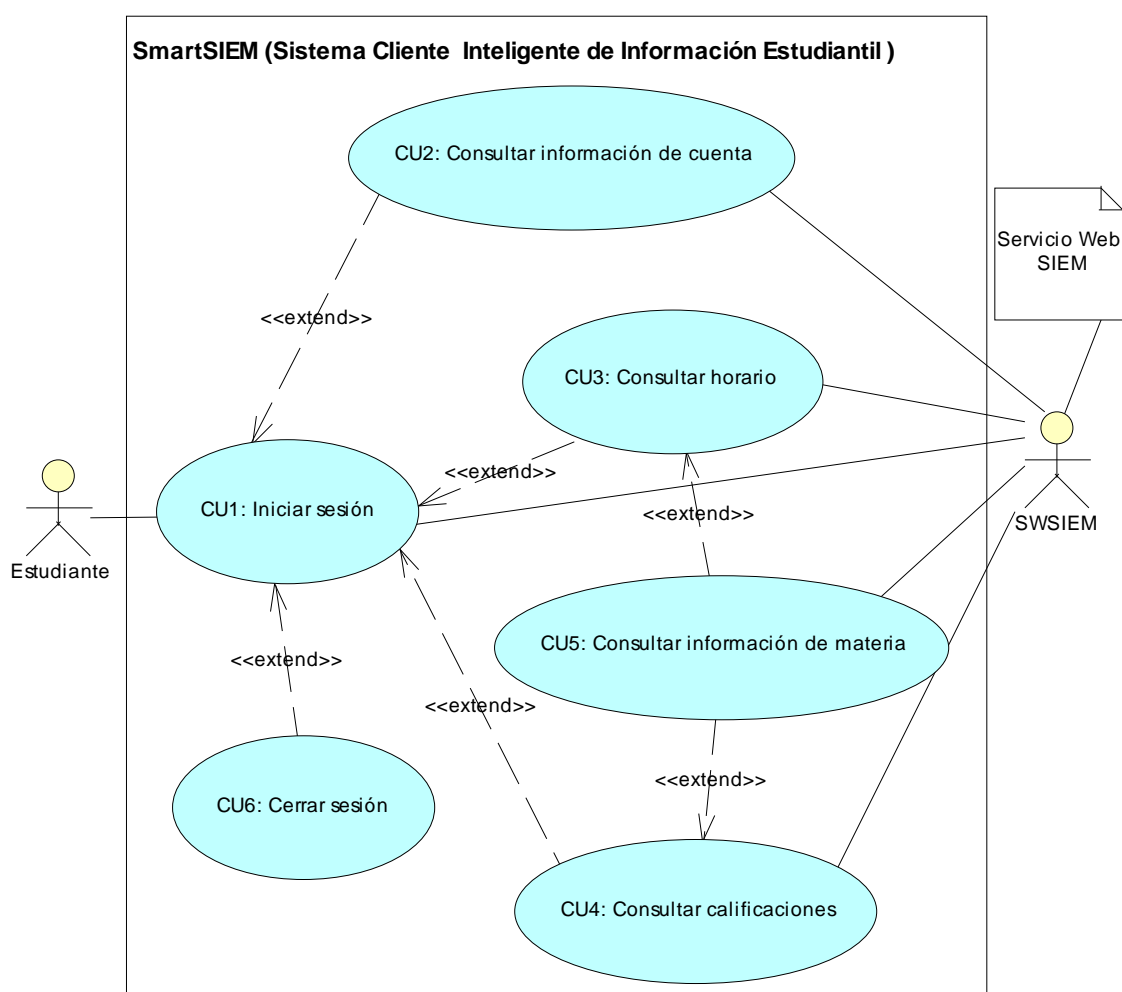


Figura 4.1 Diagrama de casos de uso de la aplicación *SmartSIEM*

La Figura 4.1 muestra los casos de uso para la aplicación *SmartSIEM*. Este diagrama de casos de uso es similar al de la Figura 2.5, ya que se tiene los

mismos casos de uso; pero se diferencian en que la aplicación *Smart* accede al servicio Web *SIEM* para obtener la información.

4.1.2.4 Especificación de casos de uso del sistema

En la Tabla 4.1 Especificación del caso de uso Inicio de sesión, se detalla el proceso de inicio de sesión.

Caso de Uso	CU1: Inicio de sesión.
Propósito	Iniciar una sesión con el servicio Web <i>SIEM</i> , proporcionado la cédula de identidad y número único de un determinado estudiante.
Precondiciones	Ninguna.
Postcondiciones	Con la sesión establecida se puede solicitar información sobre calificaciones y horarios.
Actores	Estudiante, petición de inicio de sesión. Servicio Web <i>SIEM</i> , responde a la solicitud de inicio de sesión.
Entradas	Cédula de identidad y número único.
Salidas	Identificador de sesión.
Proceso/Evento	Descripción
P: Abrir Aplicación.	<ol style="list-style-type: none"> 1. El estudiante abre la aplicación. 2. Se presenta la página de inicio de sesión.
E: Clic Iniciar sesión.	<ol style="list-style-type: none"> 1. Verificar que el número único y cédula, no sean nulos ni vacíos. 2. Iniciar sesión con el servicio Web. 3. Si el inicio de sesión es exitoso, se recibe un <i>ticket</i> de autenticación que permitirá interactuar con el servicio Web <i>SIEM</i>; este <i>ticket</i> es almacenado para consultas posteriores. 4. Se presenta la información de cuenta. 5. En caso de que el inicio de sesión sea fallido, se presenta un mensaje de error (Número único o número de cédula incorrectos).

Tabla 4.1 Especificación del caso de uso Inicio de sesión

En la Tabla 4.2 Especificación del caso de uso Consultar información de cuenta se detalla el proceso de obtención de información personal del estudiante.

En la Tabla 4.3 Especificación del caso de uso Consultar horario, se detalla el proceso que siguen un estudiante para consultar su horario de clases.

Caso de Uso	CU2: Consultar información de cuenta.
Propósito	Acceder a la información de cuenta de un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información de cuenta. Servicio Web <i>SIEM</i> , proporciona los datos solicitados.
Entradas	<i>Ticket</i> de autenticación.
Salidas	Los siguientes datos del estudiante: nombre, número de cédula, número único, identificación militar, correo electrónico.
Proceso/Evento	Descripción
P: Sesión iniciada correctamente. E: Clic Sesión.	<ol style="list-style-type: none"> 1. El estudiante solicita la información de su cuenta o el estudiante ha iniciado una sesión correctamente. 2. P: Autenticación: Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido. 3. Los controles son creados. 4. Se realiza la consulta de información de cuenta al servicio Web de manera asincrónica. 5. Se carga los controles con la información recibida. 6. Se muestra la información al usuario.

Tabla 4.2 Especificación del caso de uso Consultar información de cuenta

Caso de Uso	CU3: Consultar horario.
Propósito	Solicitar el horario de clases de un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información referente al horario de clases. Servicio Web <i>SIEM</i> , responde a la solicitud.
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Horario de las materias tomadas con el siguiente detalle: día, nombre de la materia, hora inicio, hora fin.
Proceso/Evento	Descripción
E: Clic Horario.	<ol style="list-style-type: none"> 1. P: Autenticación: Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido. 2. Los controles son creados. 3. Se realiza la consulta de horario al servicio Web de manera asincrónica. 4. Se carga los controles con la información recibida. 5. La información se muestra al usuario.

Tabla 4.3 Especificación del caso de uso Consultar horario

La Tabla 4.4 especifica el caso de uso Consultar calificaciones.

Caso de Uso	CU4: Consultar calificaciones.
Propósito	Solicitar información de las calificaciones conseguidas por un estudiante.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información de sus calificaciones. Servicio Web <i>SIEM</i> , responde a la solicitud.
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Las calificaciones del estudiante con el siguiente detalle: nombre de la materia, calificación primer bimestre, calificación segundo bimestre, calificación supletorio, suma, aprobación.
Proceso/Evento	Descripción
E: Seleccionar Ítem Calificaciones.	<ol style="list-style-type: none"> 1. P: Autenticación: Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido. 2. Se realiza la consulta de las calificaciones del estudiante al servicio Web de manera asincrónica. 3. Se crean y se cargan los controles con la información recibida. 4. La información se muestra al usuario.

Tabla 4.4 Especificación del caso de uso Consultar calificaciones

Caso de Uso	CU5: Consultar información de materia.
Propósito	Conocer el detalle de una materia.
Precondiciones	El inicio de sesión debe ser exitoso.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita información detallada de una materia. Servicio Web <i>SIEM</i> , responde a la solicitud con la información requerida.
Entradas	<i>Ticket</i> de Autenticación.
Salidas	Información de la materia con el siguiente detalle: código de la materia, nombre de la materia, aula, profesor de la materia, número de créditos, paralelo, categoría.
Proceso/Evento	Descripción
E: Clic nombre materia.	1. El estudiante solicita la información detallada de una materia
P: Autenticación.	1. Se verifica que el <i>ticket</i> de autenticación sea de un usuario válido.
P: Mostrar Panel Materia.	<ol style="list-style-type: none"> 1. Los controles son creados. 2. Se realiza la consulta del detalle de una materia al servicio Web de manera asincrónica. 3. Se carga los controles con la información recibida. 4. La información se muestra al usuario.

Tabla 4.5 Especificación del caso de uso Consultar información de materia

La Tabla 4.5 especifica el caso que permite consultar información de determinada materia.

Caso de Uso	CU6: Cerrar sesión.
Propósito	Terminar la sesión.
Precondiciones	La sesión debe estar iniciada.
Postcondiciones	Ninguna.
Actores	Estudiante, solicita finalizar la sesión.
Entradas	Ninguna.
Salidas	Ninguna.
Proceso/Evento	Descripción
E: Clic Menú Ítem Cerrar sesión.	<ol style="list-style-type: none"> 1. Se elimina el <i>ticket</i> de autenticación con el servicio Web. 2. Se limpian los controles. 3. Se muestra controles de inicio de sesión al usuario.

Tabla 4.6 Especificación del caso de uso Cerrar sesión

Tabla 4.6 detalla el caso de uso Cerrar sesión.

4.1.2.5 Diseño de Interfaces

A continuación se muestra la interfaz diseñada para la aplicación *Smart Client*.

- Interfaz de usuario para inicio de sesión, interfaz usada en el proceso de inicio de sesión, el usuario debe ingresar el número de cédula y el número único. (Figura 4.2).
- Interfaz de usuario para información del estudiante, la interfaz contiene los controles que permiten mostrar al estudiante la información de cuenta. (Figura 4.3).
- Interfaz de usuario para horario de clases, interfaz por medio de la cual se muestra al estudiante su horario de clases. La información se encuentra agrupada por días. (Figura 4.4)
- Interfaz de usuario para calificaciones, interfaz por medio de la cual se muestra al estudiante las calificaciones obtenidas por cada materia. El estudiante podrá conocer las calificaciones del primer y segundo bimestre,

supletorio, así como el promedio y la forma de aprobación de cada materia. (Figura 4.5).



Figura 4.2 Interfaz de usuario para inicio de sesión



Figura 4.3 Interfaz de usuario para información del estudiante



Figura 4.4 Interfaz de usuario para horario de clases

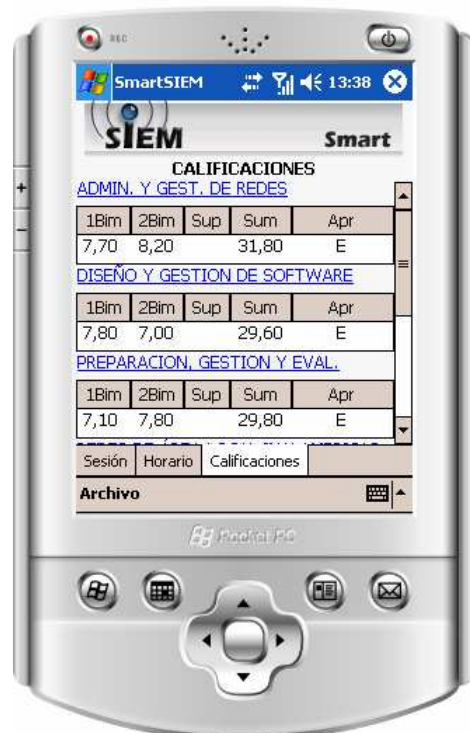


Figura 4.5 Interfaz de usuario para calificaciones

- Interfaz de usuario para información de materia, interfaz que permite mostrar al estudiante información detallada de la materia. El estudiante podrá conocer el aula, profesor, paralelo, número de créditos, la categoría y el código de la materia. Al dar clic en el botón *Aceptar* el estudiante podrá regresar a la pantalla anterior. (Figura 4.6)



Figura 4.6 Diseño de interfaz para información de materia

4.2 DISEÑO

4.2.1 DIAGRAMA DE CLASES

La aplicación *Smart Client* es una capa de interfaz, ya que sirve para presentar la información obtenida del servicio Web *SIEM* a los estudiantes.

El diagrama de clases de la Figura 4.7, muestra las clases de interfaz de la aplicación *Smart Client*.

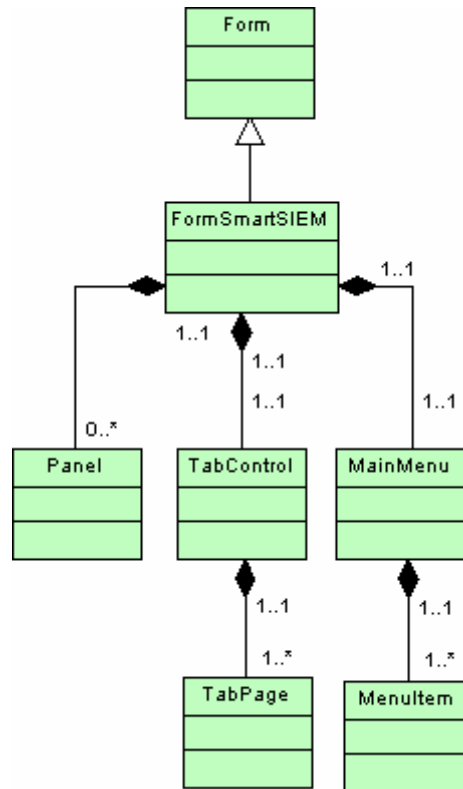


Figura 4.7 Diagrama de clases de la aplicación *Smart Client*

La clase *FormSmartSIEM* es el formulario principal de la aplicación *Smart Client*. Se deriva de la clase *Form*, y se ayuda de otras clases para mostrar la información.

Como se puede ver en la Figura 4.7, la clase *FormSmartSIEM*, contiene paneles (contenedores de controles), un *TabControl*, que a su vez contiene fichas (*TabPage*), y un *MainMenu*, todos éstos permiten mostrar la información ayudados de controles como cuadros de texto y etiquetas.

4.2.2 ATRIBUTOS

La clase de interfaz principal *FormSmartSIEM*, contiene todos los controles necesarios para presentar la información a los usuarios. En el Tabla 4.7 se da una descripción de los atributos de la clase.

FormSmartSIEM	
- estudiante	: Estudiante
- listaHorario	: List<HoraClase>
- listaCalificaciones	: List<Calificacion>
- listaDetalleMaterias	: List<DetalleMateria>
- estaConsultadoHorario	: bool
- estaConsultadoCalificaciones	: bool
- tabControlSIEM	: TabControl
- mainMenuSIEM	: MainMenu
- pictureBoxLogo	: PictureBox
- tabPageEstudiante	: tabPage
- tabPageHorario	: tabPage
- tabPageCalificaciones	: tabPage
- panelInicioDeSesión	: Panel
- panelHorario	: Panel
- panelCalificaciones	: Panel
- panelMateria	: Panel

Figura 4.8 Atributos de la clase *FormSmartSIEM*

Nombre	Descripción
estudiante	Almacena la información de un estudiante la cual es obtenida desde el servicio Web <i>SIEM</i> .
listaHorario	Almacena el horario de clases de un estudiante, obtenido desde el servicio Web <i>SIEM</i> .
listaCalificaciones	Almacena calificaciones de un estudiante que son obtenidas desde el servicio Web <i>SIEM</i> .
listaDetalleMaterias	Almacena la información detallada de una materia devuelto por el servicio Web <i>SIEM</i> .
estaConsultadoHorario	Permite establecer si el horario de clases ya se ha consultado.
estaConsultadoCalificaciones	Permite establecer si las calificaciones ya fueron consultadas.
tabControlSIEM	Contiene páginas que permiten mostrar la información.
mainMenuSIEM	Menú opciones de la aplicación que se muestra al usuario.
pictureBoxLogo	Imagen que se despliega en la parte superior del control.
tabPageEstudiante	Página que contiene los controles para mostrar la información de una cuenta.
tabPageHorario	Página que contiene los controles para mostrar al usuario el horario de clases.
tabPageCalificaciones	Página que contiene los controles para mostrar al usuario las calificaciones.
panelInicioDeSesión	Contiene los controles de inicio de sesión.
panelHorario	Contiene los controles para mostrar un horario de clases.
panelCalificaciones	Contiene los controles para mostrar las calificaciones de las materias tomadas por un estudiante.
panelMateria	Contiene los controles para mostrar información detallada de una materia.

Tabla 4.7 Descripción de los atributos de la clase *FormSmartSIEM*

4.2.3 MÉTODOS

Nombre	Descripción
InicializarFormularioSIEM()	Se inicializan los controles para permitir el inicio de sesión.
IniciarSesion(string cedulaIdentidad, string numeroUnico): bool	Se solicita un inicio de sesión al servicio Web <i>SIEM</i> .
ConsultarEstudianteAsincronicamente(): IAsyncResult	Realiza la solicitud de información sobre la cuenta al servicio Web <i>SIEM</i> de manera asincrónica, permitiendo que en la aplicación se puedan realizar otras tareas mientras se espera por la respuesta del servicio.
ConsultarEstudianteCallBack(IAsyncResult ar)	Almacena la información de cuenta del estudiante, cuando el servicio responde a la solicitud.
ConsultarCalificacionesAsincronicamente(): IAsyncResult	Realiza la solicitud de información de calificaciones al servicio Web <i>SIEM</i> de manera asincrónica, permitiendo que en la aplicación se puedan realizar otras tareas mientras se espera por la respuesta del servicio.
ConsultarCalificacionesCallBack(IAsyncResult ar)	Almacena la información de calificaciones del estudiante, cuando el servicio responde a la solicitud.
ConsultarHorarioAsincronicamente():IAsyncResult	Realiza la solicitud de información sobre el horario de clases al servicio Web <i>SIEM</i> de manera asincrónica, permitiendo que en la aplicación se puedan realizar otras tareas mientras se espera por la respuesta del servicio.
ConsultarHorarioCallBack(IAsyncResult ar)	Almacena el horario de clases del estudiante, cuando el servicio responde a la solicitud.
ConsultarDetalleMateriaAsincronicamente(string codigoMateria): IAsyncResult	Realiza la solicitud de información detallada de una materia al servicio Web <i>SIEM</i> de manera asincrónica, permitiendo que en la aplicación se puedan realizar otras tareas mientras se espera por la respuesta del servicio.
ConsultarDetalleMateriaCallBack(IAsyncResult ar)	Almacena la información detallada de una materia, cuando el servicio responde a la solicitud.
ConstruirListViewHorario(string dia): ListView	Devuelve un control ListView con el número de columnas igual a los días de la semana del horario de clases del estudiante.
listViewHorario_ItemActivate(object sender, System.EventArgs e)	Controla el evento clic en el control ListView.
ConsultarYCargarDetalleMateria(string nombreMateria)	Llama al método <i>BuscarDetalleMateria</i> . Si se obtiene información, llama a los métodos <i>CargarPanelMateria</i> y <i>MostrarPanelMateria</i> . Si se obtiene nulo, se llama a <i>BuscarMateria</i> , <i>ConsultarDetalleMateriaAsincronicamente</i> y a <i>MostrarPanelMateria</i> .
BuscarMateria(string nombreMateria): Materia	Se busca una materia por el nombre en la información de horario y calificaciones.
BuscarDetalleMateria(string nombreMateria): DetalleMateria	Se busca el detalle de una materia con el nombre de la materia.

MostrarPanelMateria()	Se muestran los controles que permiten exponer la información detallada de una materia.
CargarPanelMateria(DetalleMateria detalleMateria)	La información detallada de una materia es colocada en los controles.
CargarEstudiante()	La información de cuenta es colocada en los controles.
CargarHorario()	La información de horario de clases de un estudiante es colocada en el control ListView.
ConstruirListViewCalificacion():ListView	Se crea un control ListViex con las columnas necesarias para presentar la información de calificaciones al usuario.
ContruirLinkLabelCalificacion(string nombreMateria): LinkLabel	Se crea hipervínculos con los nombres de las materias y se llama al método <i>LinkLabel_Click</i> .
LinkLabel_Click(object sender, EventArgs e)	Controla el evento clic en los hipervínculos con los nombres de las materias. Llama al método <i>ConsultarY CargarDetalleMateria</i> .
CargarCalificaciones()	Se coloca en los controles la información de calificaciones del estudiante.
InicializarListView(ListView list, string[] columnas, int[] medidas)	Inicializa un ListView con el formato adecuado.
AgregarItemListView(string[] item, ListView list)	Agrega un fila a un ListView determinado.
tabControlSIEM_SelectedIndexChanged(object sender, EventArgs e)	Se llama al método <i>ConsultarHorarioAsincronicamente</i> o <i>ConsultarCalificacionesAsincronicamente</i> dependiendo de la opción que el usuario haya seleccionando.
buttonIniciarSesion_Click(object sender, EventArgs e)	Controla el evento clic en el botón <i>IniciarSesion</i> . Se verifica si el usuario ha ingresado el número único y cédula y se llama a los métodos <i>ConsultarEstudianteAsincronicamente</i> y <i>MostrarControlesDeSesionIniciada</i> . De lo contrario se envía un mensaje de error.
BloquearControlesParalnicioDeSesion()	Los controles de inicio de sesión son deshabilitados.
DesbloquearControlesParalnicioDeSesion()	Los controles de inicio de sesión son habilitados.
MostrarControlesDeSesionIniciada()	Se muestran los controles de información de cuenta.
buttonMateriaAceptar_Click(object sender, EventArgs e)	Controla el evento clic en el botón <i>Aceptar</i> , cuando se muestra la información detallada de una materia. Los controles se ocultan.
menuItemCerrarSesion_Click(object sender, EventArgs e)	Se elimina el <i>ticket</i> de autenticación con el servicio Web <i>SIEM</i> , se elimina la información de los controles y se inicializa el formulario.
LimpiarControles()	Se elimina la información de los controles.

Tabla 4.8 Descripción de los métodos de la clase *FormSmartSIEM*

4.2.4 DIAGRAMAS DE ACTIVIDAD

En esta sección se exponen los diagramas de actividad correspondientes a cada caso de uso de la aplicación *Smart Client*.

4.2.4.1 Diagrama de actividad Inicio de sesión

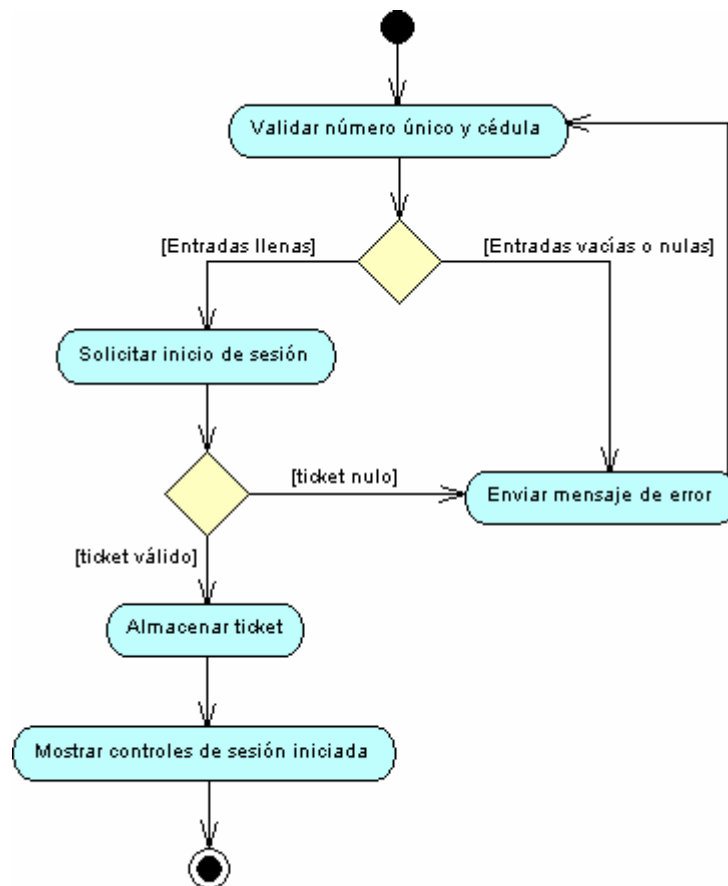


Figura 4.9 Diagrama de actividad Inicio de sesión

El diccionario para el diagrama de actividad de inicio de sesión se determinó de la siguiente manera:

1. Validar número único y cédula: se comprueba que el número único y la cédula no sean vacíos o nulos
2. Solicitar inicio de sesión: luego de comprobar que los dos campos están llenos, se hace la solicitud de inicio de sesión al servicio Web *SIEM*.

3. Almacenar *ticket*: sucede cuando los datos ingresados corresponden a un usuario válido del sistema, entonces el servicio Web *SIEM* responde al pedido de inicio de sesión enviando un *ticket* de autenticación, el cual es almacenado para poder realizar consultas posteriores al servicio.
4. Mostrar controles de sesión iniciada: sucede cuando el inicio de sesión ha sido exitoso.
5. Enviar mensaje de error: sucede cuando los datos de entrada son vacíos o nulos y cuando se recibe del servicio Web *SIEM* un *ticket* nulo.

4.2.4.2 Diagrama de actividad Consultar información de cuenta

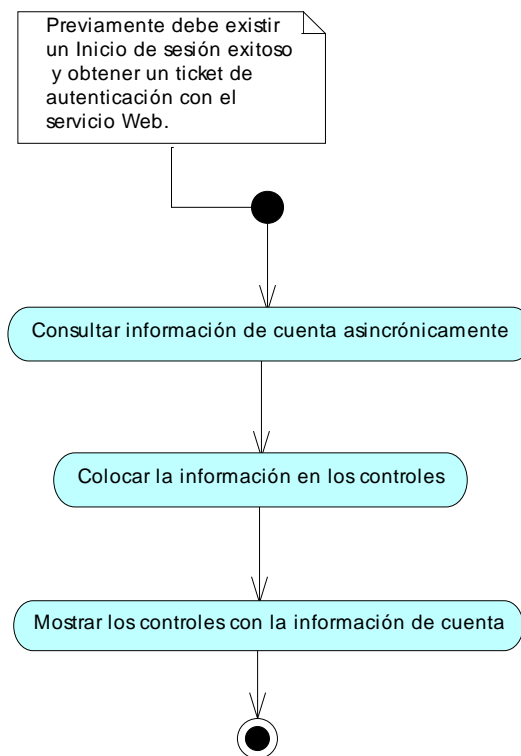


Figura 4.10 Diagrama de actividad Consultar información de cuenta

El diccionario correspondiente se indica a continuación:

1. Consultar información de cuenta asincrónicamente: se solicita al servicio Web *SIEM* información sobre la cuenta activa de forma asincrónica, de manera que se pueda realizar otras tareas en la aplicación, mientras se obtiene la respuesta desde el servicio Web.

2. Colocar la información en los controles: la información recibida se llena en los controles de información de cuenta.
3. Mostrar los controles con la información de cuenta: la información se muestra al usuario.

4.2.4.3 Diagrama de actividad Consultar horario

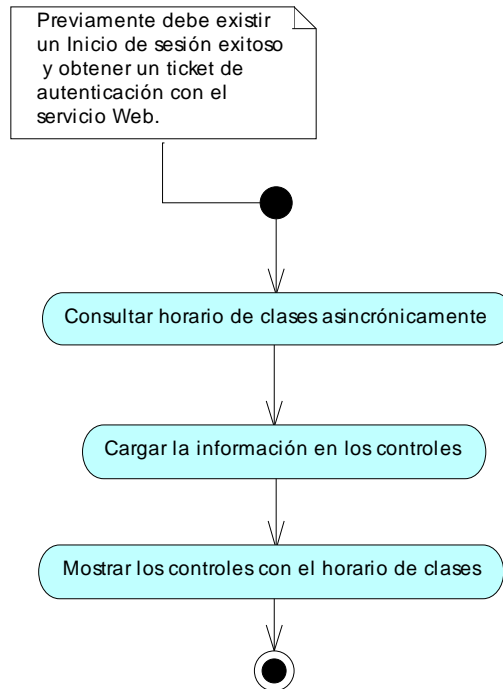


Figura 4.11 Diagrama de actividad Consultar horario

El diccionario de actividades se establece de la siguiente manera:

1. Consultar horario de clases asincrónicamente: se solicita al servicio Web *SIEM* información sobre el horario de clases de manera asincrónica, permitiendo a la aplicación realizar otras tareas mientras espera la respuesta desde el servicio Web.
2. Cargar la información en los controles: los controles son llenados con la información recibida.
3. Mostrar los controles con el horario de clases: la información se muestra al usuario.

4.2.4.4 Diagrama de actividad Consultar calificaciones

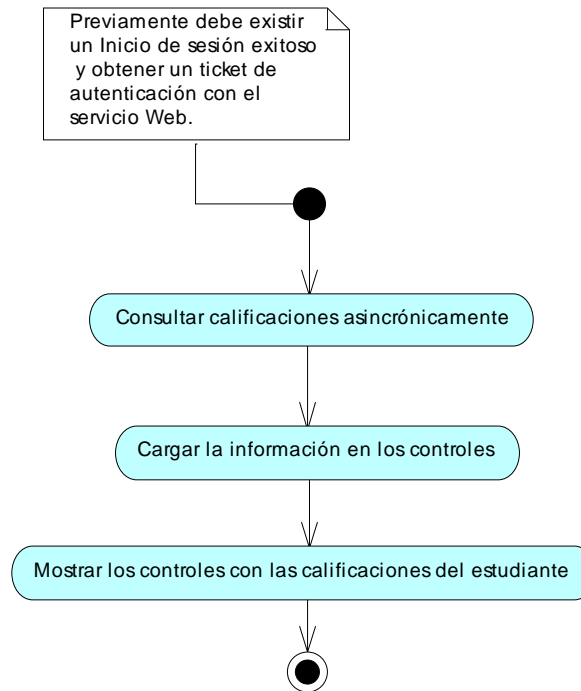


Figura 4.12 Diagrama de actividad Consultar calificaciones

El diccionario de actividades se establece de la siguiente manera:

1. Consultar calificaciones asincrónicamente: se solicita al servicio Web *SIEM* información sobre las calificaciones de un estudiante de manera asincrónica, permitiendo a la aplicación realizar otras tareas mientras espera la respuesta desde el servicio Web.
2. Cargar la información en los controles: los controles son llenados con la información recibida.
3. Mostrar los controles con las calificaciones del estudiante: la información se muestra al usuario.

4.2.4.5 Diagrama de actividad Consultar información de materia

El diccionario de actividades para la Figura 4.13 se establece de la siguiente manera:

1. Consultar información detallada de una materia asincrónicamente: se solicita al servicio Web *SIEM* información detallada de una materia de manera asincrónica, permitiendo a la aplicación realizar otras tareas mientras espera la respuesta desde el servicio Web.
2. Cargar la información en los controles: los controles son llenados con la información recibida.
3. Mostrar los controles con la información de una materia: la información se muestra al usuario.

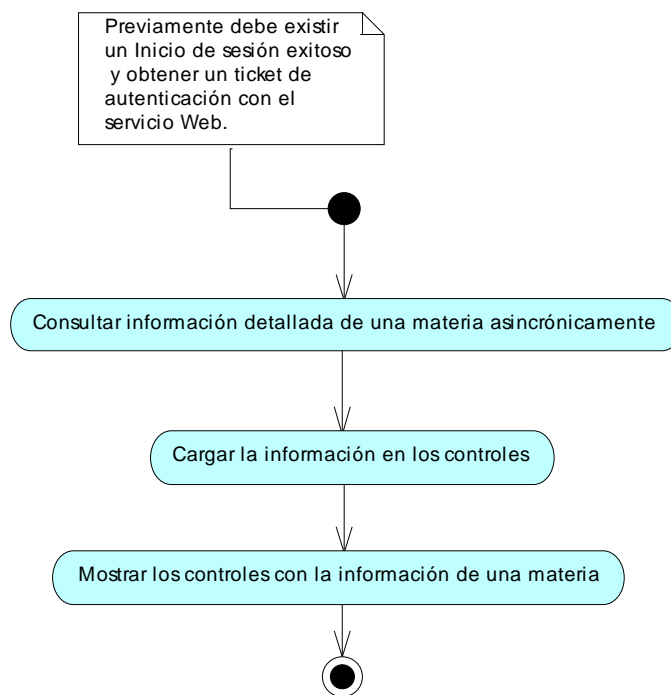


Figura 4.13 Diagrama de actividad Consultar información de materia

4.2.4.6 Diagrama de actividad Cerrar sesión

El diccionario de actividades para la Figura 4.14 se establece de la siguiente manera:

1. Borrar *ticket* de autenticación del servicio Web *SIEM*: el *ticket* de autenticación obtenido desde el servicio Web *SIEM* es borrado.
2. Borrar la información de los controles: la información recibida es borrada de todos los controles.

- Mostrar controles de inicio de sesión: se muestra los controles de inicio de sesión al usuario.

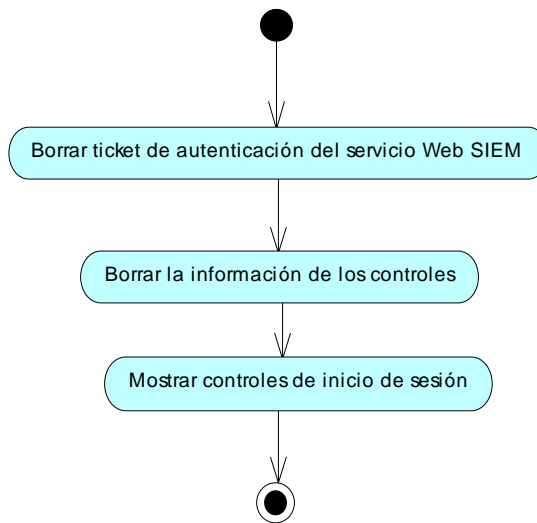


Figura 4.14 Diagrama de actividad Cerrar sesión

4.2.5 DIAGRAMAS DE SECUENCIA

4.2.5.1 Diagrama de secuencia Inicio de sesión

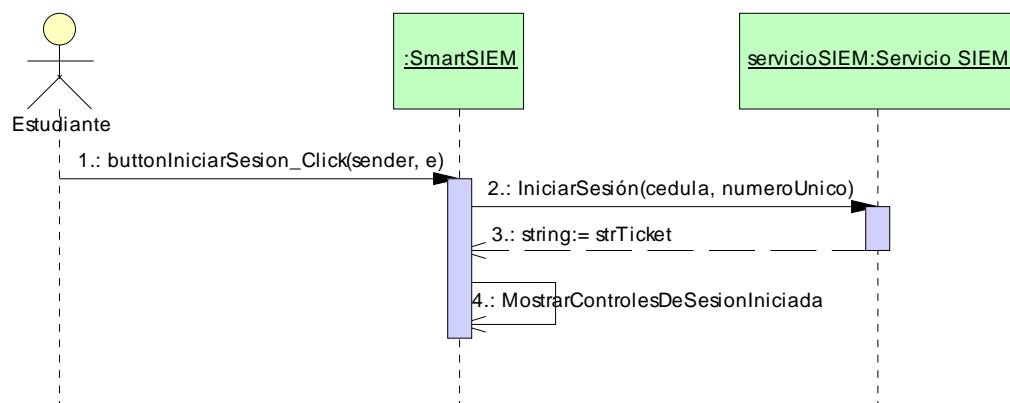


Figura 4.15 Diagrama de secuencia Inicio de sesión

La Figura 4.15 muestra la secuencia seguida al iniciar una sesión con el *servicio Web SIEM*, utilizando la aplicación *SmartSIEM*.

El proceso comienza cuando un estudiante inicia la aplicación, ingresa su cédula y número único y da clic en el botón iniciar sesión de la aplicación.

La aplicación *SmartSIEM* controla el evento del botón, verificando primero si el usuario ha ingresado los campos requeridos (cédula y número único). Si uno o los dos campos son vacíos o nulos, se envía al usuario un mensaje informándole que los campos son requeridos.

Si ambos campos están llenos, la aplicación *SmartSIEM* solicita *IniciarSesión* al servicio Web *SIEM*, donde se pasan el número único y la cédula del estudiante.

La aplicación espera por la respuesta del servicio. Si el servicio retorna nulo, se envía un mensaje de error, indicando al usuario que la cédula o número único ingresados son incorrectos.

Si la respuesta desde el servicio es un *ticket* de autenticación, éste es almacenado en una variable de sesión para realizar peticiones al servicio Web.

4.2.5.2 Diagrama de secuencia Consultar información de cuenta

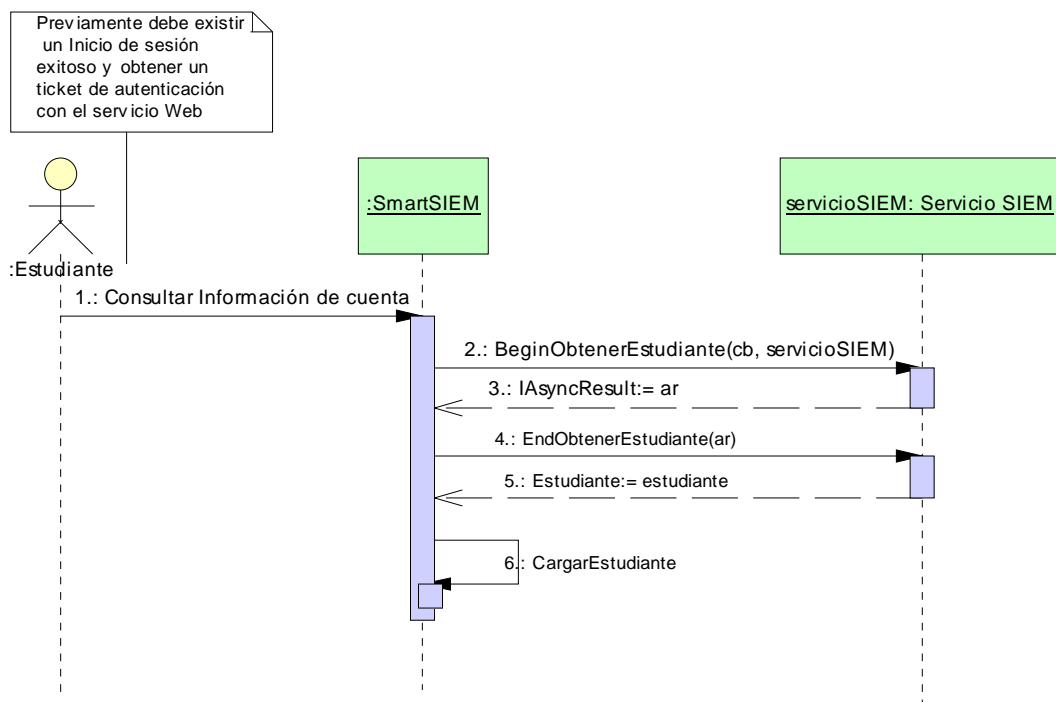


Figura 4.16 Diagrama de secuencia Consultar información de cuenta

Luego de haber iniciado la sesión correctamente con el *servicio Web SIEM*, la aplicación *SmartSIEM* solicita *BeginObtenerEstudiante* al objeto *servicioSIEM*; de esta manera se hace una solicitud asíncrona, con lo que la aplicación puede realizar otras tareas, mientras espera por la respuesta. El *servicio Web SIEM* le informa a la aplicación que la respuesta está lista y la envía; la información de cuenta es almacenada en el objeto *estudiante*.

Luego de obtener la información la aplicación llama a *CargarEstudiante*, para colocar la información en los controles. Por último la aplicación muestra al usuario la información de la cuenta del estudiante.

4.2.5.3 Diagrama de secuencia Consultar horario

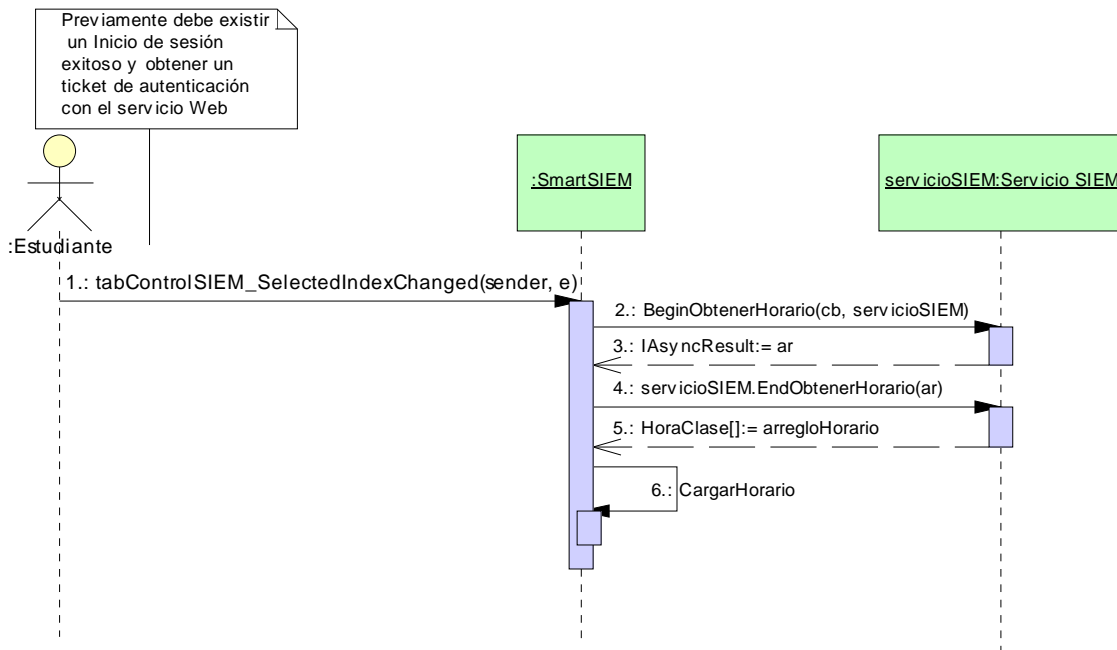


Figura 4.17 Diagrama de secuencia Consultar horario

El diagrama muestra la secuencia seguida para conocer el horario de clases, luego de haber iniciado una sesión exitosa y obtenido un *ticket* de autenticación válido.

El estudiante da clic sobre *Horario*, entonces la aplicación solicita *BeginObtenerHorario* al objeto *servicioSIEM* y puede realizar otras tareas mientras espera la respuesta. El *servicio Web SIEM* le informa a la aplicación que

la respuesta está lista y la envía. La información recibida es almacenada en un arreglo de objetos *HoraClase*. Luego la aplicación llama a *CargarHorario*, para colocar la información en los controles respectivos.

Finalmente la aplicación muestra al estudiante su horario de clases.

4.2.5.4 Diagrama de secuencia Consultar calificaciones

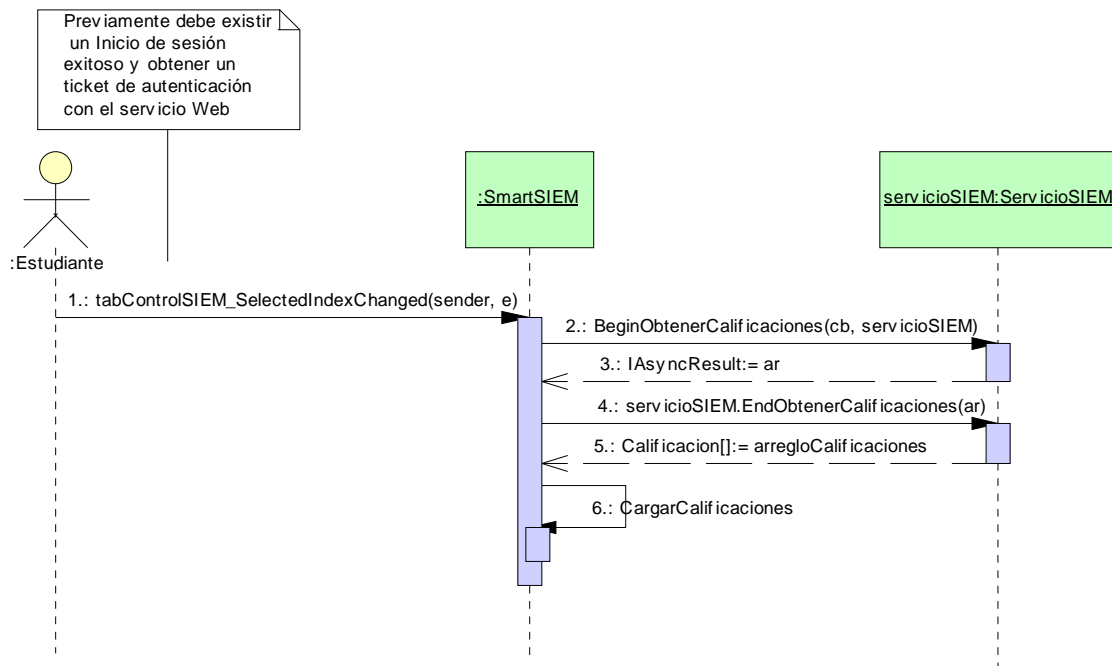


Figura 4.18 Diagrama de secuencia Consultar calificaciones

La Figura 4.18 muestra la secuencia seguida para conocer las calificaciones de un estudiante, luego de haber iniciado una sesión exitosa y obtenido un *ticket* de autenticación válido.

El estudiante da clic sobre *Calificaciones*, entonces la aplicación solicita *BeginObtenerCalificaciones* al objeto *servicioSIEM* y puede realizar otras tareas mientras espera la respuesta. El *servicio Web SIEM* le informa a la aplicación que la respuesta está lista y la envía. La información recibida es almacenada en un arreglo de objetos *Calificacion*. Luego la aplicación llama a *CargarCalificaciones*, para colocar la información en los controles respectivos.

Finalmente la aplicación muestra al estudiante sus calificaciones.

4.2.5.5 Diagrama de secuencia Consultar información de materia

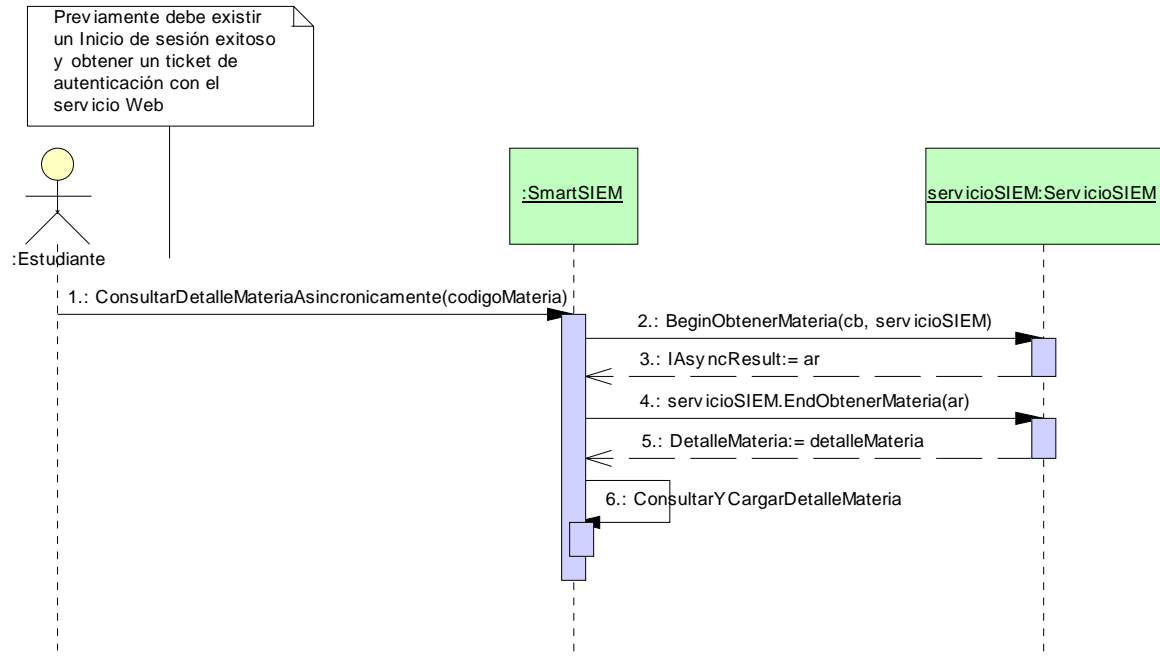


Figura 4.19 Diagrama de secuencia Consultar información de materia

El diagrama de la Figura 4.19 muestra la secuencia para conocer información detallada de una materia, luego de haber iniciado una sesión exitosa y obtenido un *ticket* de autenticación válido.

El estudiante da clic sobre *el nombre de la materia*, la aplicación controla el evento y solicita *BeginObtenerCalificaciones* al objeto *servicioSIEM* y puede realizar otras tareas mientras espera la respuesta. El *servicio Web SIEM* le informa a la aplicación que la respuesta está lista y la envía. La información recibida es almacenada en el objeto *detalleMateria* de la clase *DetalleMateria*. Luego la aplicación llama a *ConsultarYCargarDetalleMateria*, para colocar la información en los controles respectivos.

Finalmente la aplicación muestra al estudiante la información detallada de una materia.

4.2.5.6 Diagrama de secuencia Cerrar sesión

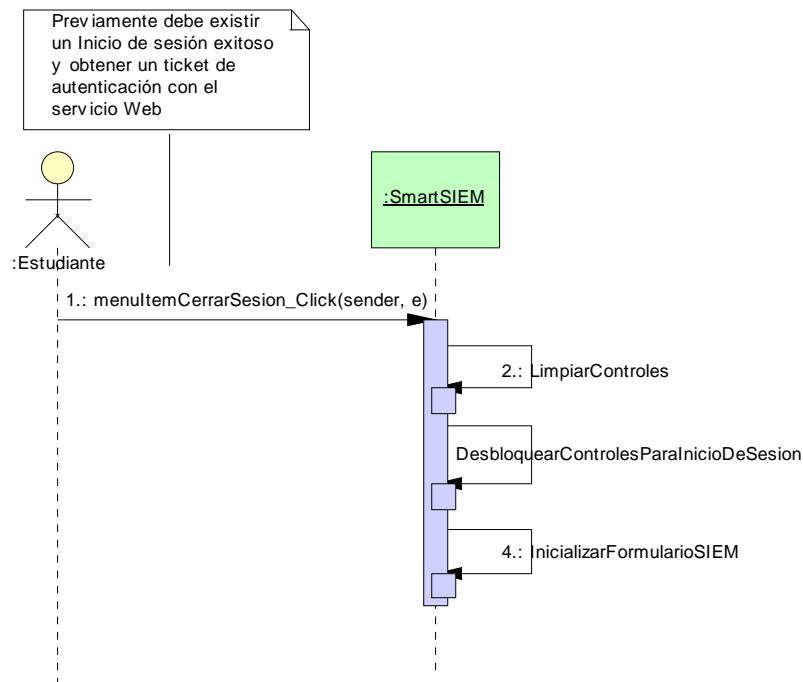


Figura 4.20 Diagrama de secuencia Cerrar sesión

El diagrama de la Figura 4.20 muestra la secuencia para cerrar la sesión con la aplicación y con el servicio Web.

El estudiante da clic sobre *Cerrar sesión* del menú *Archivo*. La aplicación *SmartSIEM* controla el evento, en donde, se elimina el *ticket* de autenticación del servicio Web *SIEM*, se llama a *DesbloquearControlesParaInicioDeSesion* para habilitar los controles de inicio de sesión, y por último se llama a *InicializarFormularioSIEM* para mostrar al usuario la página de inicio de sesión.

4.3 IMPLEMENTACIÓN

Para la implementación de la aplicación cliente inteligente o *Smart Client* se utilizó un proyecto de *Visual Studio 2005* de *Pocket PC 2003*; esta aplicación es básicamente un interfaz que se ejecuta en el móvil para el acceso a datos del servicio Web.

Para empezar se debe crear un proyecto para dispositivo inteligente o *Pocket PC 2003 Smart Device*. Para esto se siguen los siguientes pasos:

1. Clic Archivo, Nuevo, Proyecto o bien Ctrl+Mayús.+N.

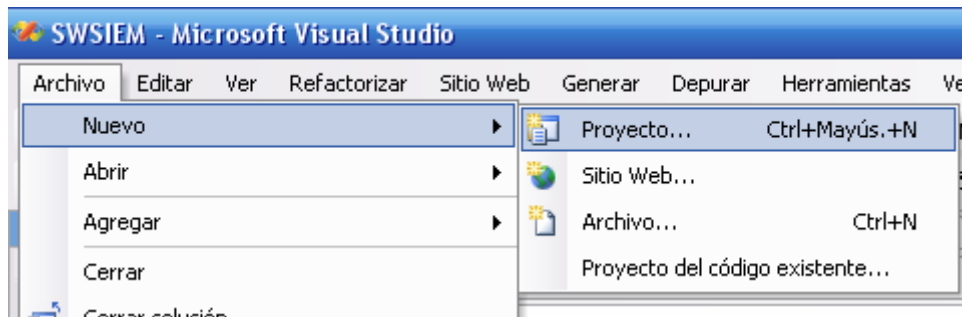


Figura 4.21 Crear proyecto cliente inteligente

2. Seleccionar *Pocket PC 2003* y *Aplicación de dispositivo*. Asignar el nombre para el proyecto en este caso *SmartSIEM* y clic en aceptar.

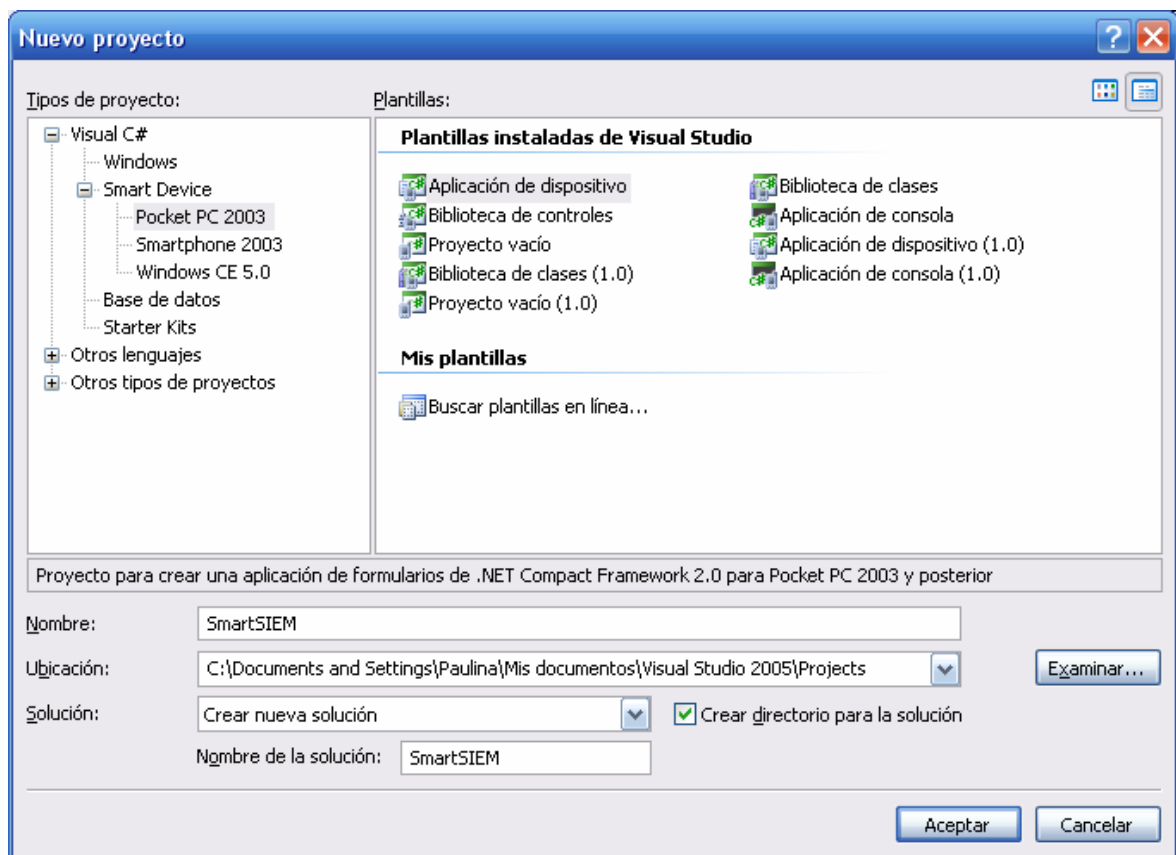


Figura 4.22 Seleccionar plantilla adecuada de dispositivo

Con el proyecto creado se tiene acceso a una representación del dispositivo móvil de tipo *Pocket PC*, el cual tiene el formulario sobre el cual se implementará la aplicación, este formulario se muestra en la Figura 4.23.

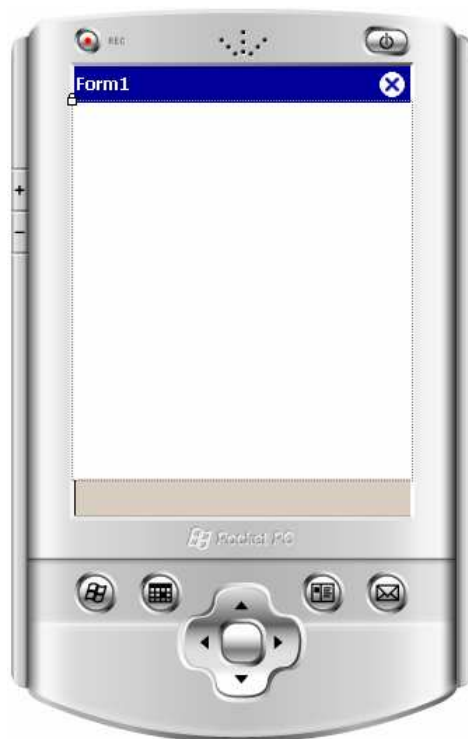


Figura 4.23 Formulario para proyecto *Pocket PC*

Visual Studio 2005 utiliza un emulador del dispositivo móvil para depurar la aplicación y este emulador debe conectarse al servicio Web, por tanto el emulador debe tener la capacidad de acceder a la red para lograr depurar la aplicación. Para lograr conectar el emulador a la red se deben seguir los siguientes pasos.

1. Instalar el programa de sincronización de dispositivos inteligentes *Microsoft ActiveSync*. En el presente proyecto se utilizó la versión 4.5 del programa. Ver Figura 4.24.
2. En el *Visual Studio 2005*, hacer clic en Herramientas, Administrador de emuladores de dispositivo. Ver Figura 4.25.

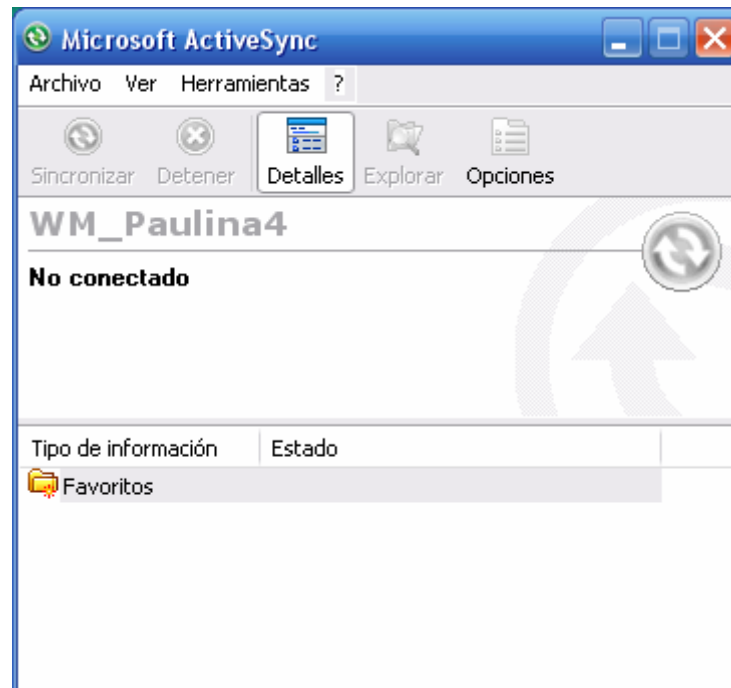


Figura 4.24 Microsoft ActiveSync

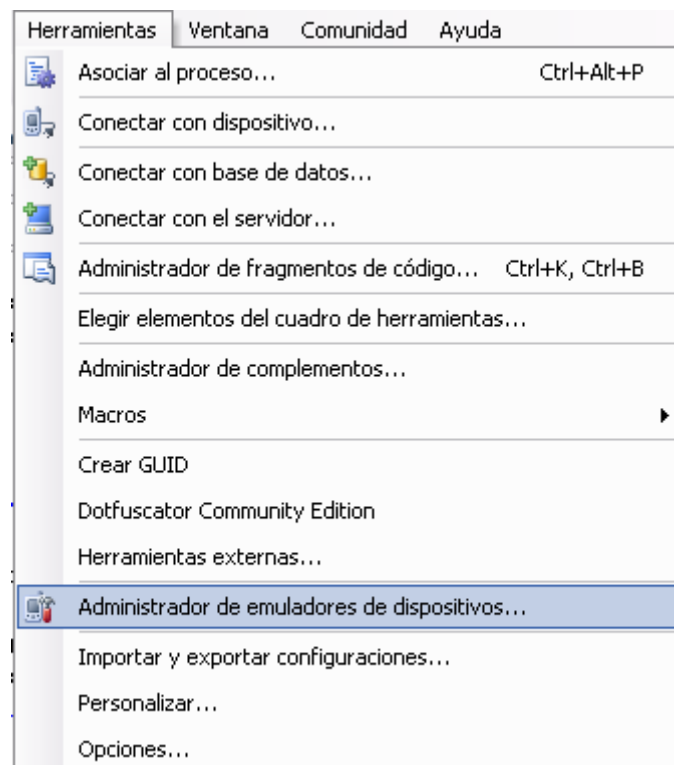


Figura 4.25 Administrador de emuladores de dispositivo

3. Clic derecho en *Pocket PC 2003 SE – Emulador* y clic en Conectar. Esta acción conecta al emulador con *Visual Studio 2005*.

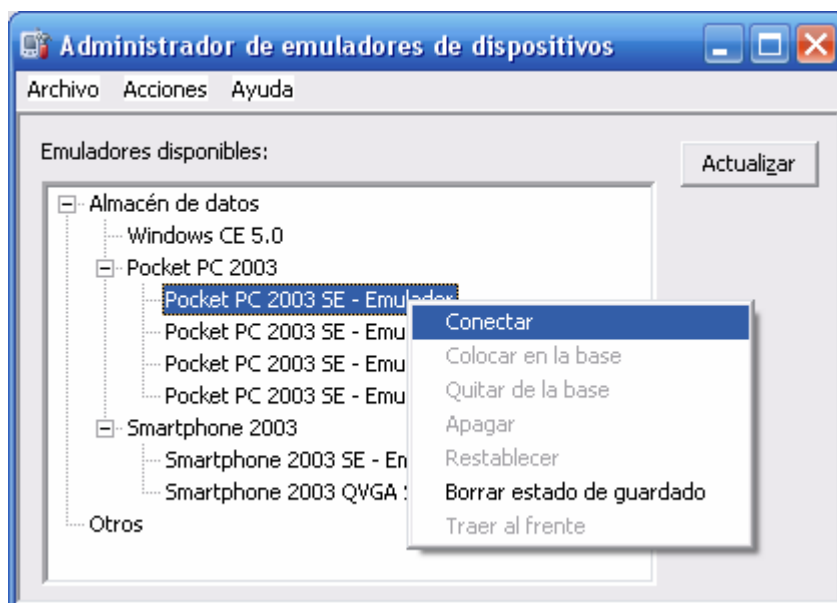


Figura 4.26 Conectar un emulador

4. Clic derecho en *Pocket PC 2003 SE – Emulador* y clic en Colocar en la base, esta acción permite conectar el emulador de dispositivo con las funciones de red a través de la aplicación de sincronización *ActiveSync*.

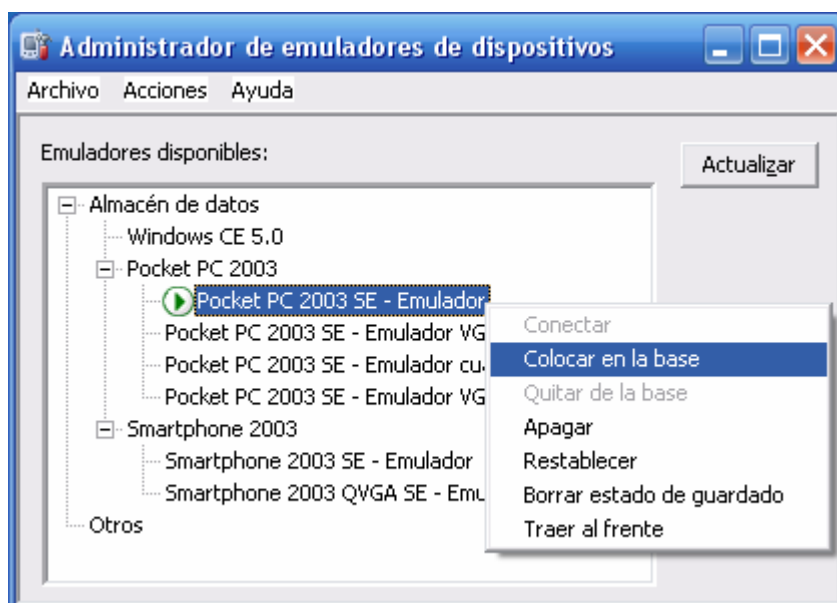


Figura 4.27 Colocar en la base un emulador

Una vez logrado que el emulador se conecte a la red, es hora de agregar controles al formulario que se presenta en la Figura 4.23 Formulario para proyecto *Pocket PC*.

Realizar esto es sencillo, se logra efectuando los siguientes pasos:

1. Clic en Ver, Cuadro de herramientas o bien Ctrl + W, X.

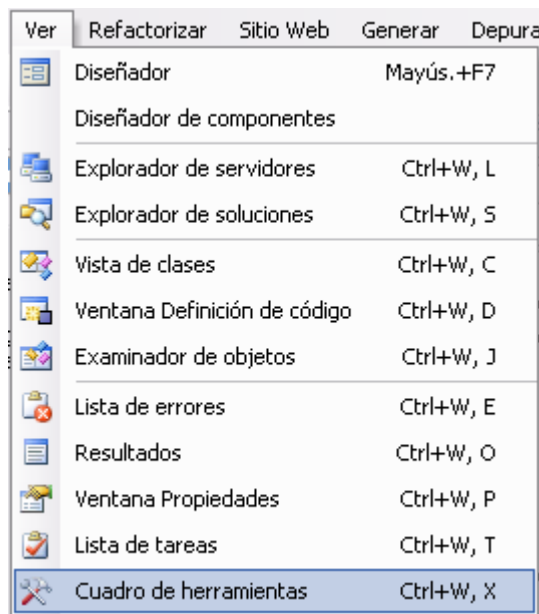


Figura 4.28 Ver cuadro de herramientas

2. Agregar los controles al formulario. Este proceso se debe repetir hasta tener todos los controles necesarios en el formulario. Ver Figura 4.29.

En este punto es necesario programar las acciones que cada uno de los controles tomará, por ejemplo es necesario escribir código para obtener información desde el servicio Web y mostrar estos datos en el formulario. Este código se presenta como anexo al presente proyecto en un CD (Anexo D).

Como se mencionó anteriormente la aplicación toma los datos del servicio Web, esto se logra agregando una referencia Web al proyecto, los pasos para agregar una referencia Web al proyecto se señalan en la sección Implementación del capítulo tercero. Cuando la referencia Web es asignada es posible acceder a los métodos del servicio Web.

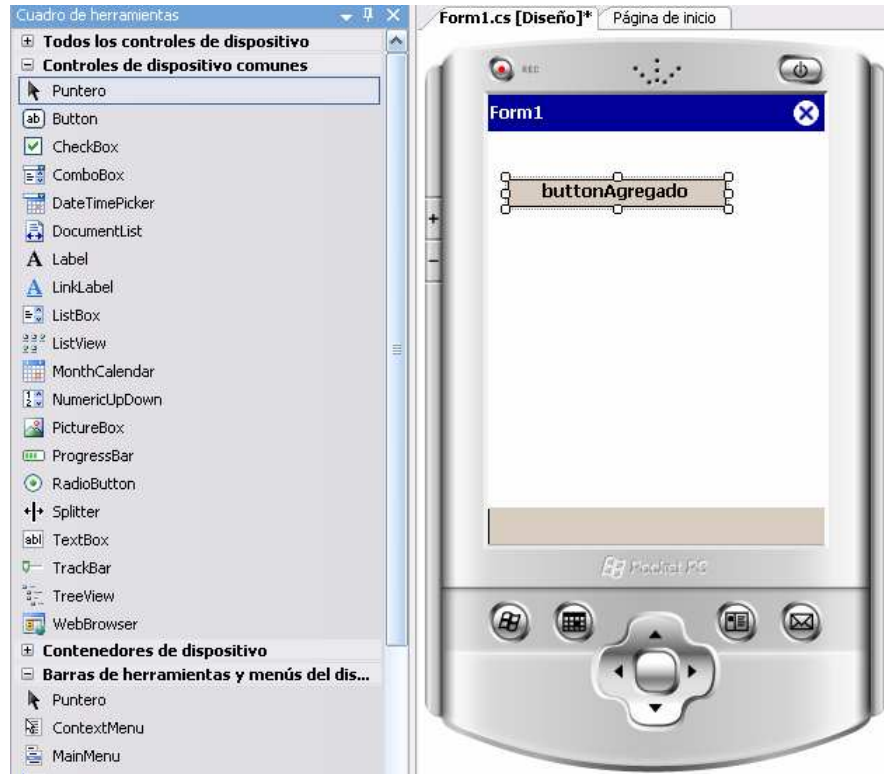


Figura 4.29 Agregar controles al formulario

El proceso de autenticación de la aplicación *Smart Client* consta de los siguientes pasos:

1. El usuario ingresa su número de cédula y su número único.
2. Esta información es enviada al servicio Web, para un proceso de autenticación con el servicio Web, si el proceso es exitoso se obtiene un *ticket* de autenticación, el mismo que es asignado a la clase *proxy* para posteriores peticiones.

El siguiente código implementa la lógica de autenticación:

```
//Clase proxy para conexión con el servicio Web
Service servicioSIEM;
//Se obtiene el ticket de autenticación
string strTicket =
servicioSIEM.IniciarSesion(cedulaIdentidad, numeroUnico);
//Se crea el ticket con la información recibida del servicio Web
TicketDeAutenticacion ticketAutenticacion = new TicketDeAutenticacion();
ticketAutenticacion.ticket = strTicket;
//Se asigna el ticket a la clase proxy
servicioSIEM.TicketDeAutenticacionValue = ticketAutenticacion;
```

Para el caso de la aplicación *Smart Client* los métodos del servicio Web *SIEM* son llamados en modo asíncrono, esto quiere decir que la aplicación cliente no se bloqueará hasta recibir respuesta del servicio Web. Esto es muy útil para brindar una mejor experiencia al usuario de la aplicación.

Esto se logra siguiendo los siguientes pasos:

1. Llamar al método *Begin* correspondiente del servicio Web, este método inicia el requerimiento asíncrono, como ejemplo se tomará el método que permite obtener la información de un estudiante *BeginObtenerEstudiante*. Se debe establecer el método que responderá al *CallBack* del servicio Web para este caso *ConsultarEstudianteCallBack*
2. Llamar al método *End* correspondiente, este método finaliza el requerimiento asíncrono, para este caso *EndObtenerEstudiante*.
3. Invocar el delegado para la actualización de interfaz de usuario, este método actualiza los datos mostrados al usuario con la información consultada asíncronicamente.

El siguiente fragmento de código implementa lo expuesto.

```
//Clase proxy para comunicarse
Service servicioSIEM;

//Información del estudiante
Estudiante estudiante;

//Delegado para actualización de interface
private delegate void ActualizarInterface();
delegadoActualizarEstudiante = new ActualizarInterface(CargarEstudiante);

//Función que llama a Begin y asigna el callback
private IAsyncResult ConsultarEstudianteAsincronicamente()
{
    //Callback
    AsyncCallback cb =
    new AsyncCallback(ConsultarEstudianteCallBack);
    //Lamar a begin
    IAsyncResult ar =
    servicioSIEM.BeginObtenerEstudiante(cb, servicioSIEM);
    return ar;
}

//Función de callback
private void ConsultarEstudianteCallBack(IAsyncResult ar)
{
    //llamado al método End
    estudiante = servicioSIEM.EndObtenerEstudiante(ar);
}
```

```
....//llamado a la función de actualización de interfaz
    this.Invoke(delegadoActualizarEstudiante);
}

//Función a la que es apuntada por el delegado de actualización
//actualiza la interfaz de usuario
private void CargarEstudiante()
{
    textBoxNombreEstudiante.Text = estudiante.Nombre;
    textBoxCedulaEstudiante.Text = estudiante.Cedula;
    textBoxEmailEstudiante.Text = estudiante.Email;
    textBoxIdMilitar.Text = estudiante.LibretaMilitar;
    textBoxNumeroUnicoEstudiante.Text = estudiante.NumeroUnico;
}
```

Capítulo 5

Pruebas y Comparación

5 PRUEBAS Y COMPARACIÓN

En este capítulo se probarán y compararán las aplicaciones *Online* y *Smart Client*.

La fase de pruebas, tiene como fin asegurar el correcto funcionamiento de las aplicaciones; estas pruebas se detallan en las secciones 5.1 y 5.2 del presente capítulo.

El siguiente paso es comparar las aplicaciones entre sí; las secciones 5.3 y 5.4 describen el proceso de comparación y los resultados de este proceso.

5.1 CASOS DE PRUEBAS

Se probarán las aplicaciones *Online* y *Smart Client*, para verificar el correcto funcionamiento de las mismas, con este objetivo se definen casos o escenarios de pruebas.

Los casos de pruebas de las aplicaciones son las posibles interacciones que el usuario puede tener con las mismas. Así, para el caso de las aplicaciones *Online* y *Smart Client*, los casos de prueba son los siguientes:

- Iniciar sesión, el estudiante inicia una sesión, para esto utiliza su número de cédula y su número único.
- Consultar información de cuenta, el estudiante puede revisar la información de su sesión.
- Consultar horario, el estudiante consulta el horario de clases.
- Consultar calificaciones, el estudiante consulta sus calificaciones académicas.
- Consultar información de materia, se obtiene información como nombre, número de créditos, etc. de las materias tomadas por el estudiante.
- Cerrar sesión, el estudiante termina la sesión.

5.2 REALIZACIÓN DE PRUEBAS

En esta sección se realizan las pruebas de funcionamiento de las aplicaciones. Se accederá al servicio Web utilizando la aplicación *Online*, así como, la aplicación *Smart Client*. En ambos casos se comprobará que se obtengan los resultados esperados.

5.2.1 AMBIENTE DE PRUEBAS FUNCIONALES

Para realizar las pruebas de funcionamiento del servicio Web y las aplicaciones, es importante definir algunas características del ambiente en el cual se desarrollarán.

Las características del servidor, tanto en hardware como en software, se muestran en la Tabla 5.1.

Hardware	Procesador	<i>Intel Core 2 2.0 GHz</i>
	Memoria RAM	1 MB
	Disco Duro	300 GB
	Red	100 Mbps
Software	Sistema operativo	<i>Windows 2003 Server</i>
	Servidor Web	<i>Internet Information Server 6.0</i>
	Servidor de base de datos	<i>SQL Express Edition</i>
	Software de medición	<i>Management Console</i>

Tabla 5.1 Características del servidor

Hardware	Procesador	<i>Intel PXA270 624 MHz</i>
	Memoria RAM	64 MB
	Memoria ROM	256 MB
	Red	<i>Wi-Fi b</i> <i>Bluetooth</i> <i>Infrarrojo</i>
Software	Sistema operativo	<i>Windows Mobile 5.</i>

Tabla 5.2 Características del dispositivo móvil

El equipo cliente será un *Pocket PC* de la marca *Dell*, las características de este dispositivo móvil se describen en la Tabla 5.2.

La Figura 5.1, muestra la configuración del ambiente de pruebas para la aplicación *Online* y *Smart Client*.

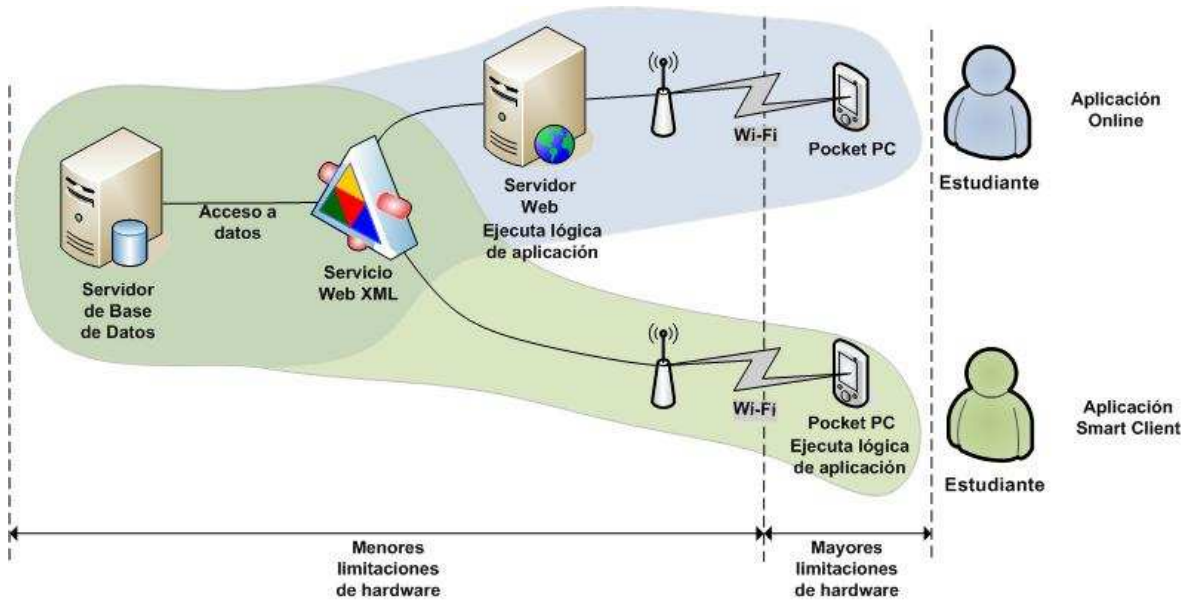


Figura 5.1 Configuración de ambiente de pruebas para la aplicación *Online*

Para estas pruebas se utilizó el laboratorio de la Carrera de Electrónica y Redes de Información. En el laboratorio se instaló las aplicaciones y utilizando la red inalámbrica se probó con la ayuda del dispositivo móvil *Pocket PC* el correcto funcionamiento de las aplicaciones *Online* y *Smart Client*.

5.2.2 RESULTADOS DE LA APLICACIÓN *ONLINE*

Para cada uno de los casos de pruebas definidos en la sección 5.1, los resultados de las pruebas utilizando los datos del estudiante Machado Luis son los siguientes:

- Iniciar sesión, la Figura 5.2 muestra la pantalla de inicio de sesión para la aplicación *Online*, donde se deben ingresar el número único y el número de cédula del estudiante.

- Consultar información de cuenta, al autenticarse el usuario, la aplicación muestra la pantalla de información de cuenta (Figura 5.3), en donde el estudiante podrá ver sus datos personales. Esta pantalla además, cuenta con los vínculos para acceder a la información de horarios y calificaciones.

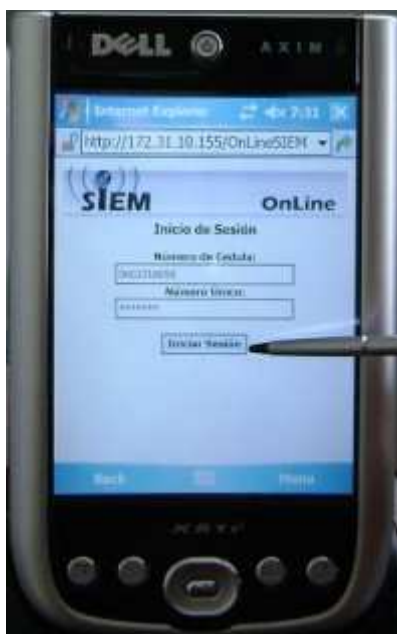


Figura 5.2 Prueba iniciar sesión en la aplicación *Online*



Figura 5.3 Prueba Consultar información de cuenta en la aplicación *Online*

- Consultar calificaciones, esta pantalla presenta las calificaciones obtenidas por el estudiante, los nombres de las materias son vínculos que permiten visualizar información detallada de dicha materia. Los resultados se muestran en la Figura 5.5.



Figura 5.6 Prueba Consultar información de materia en la aplicación *Online*



Figura 5.7 Prueba cerrar sesión en la aplicación *Online*

- Consultar información de materia, esta pantalla muestra información detallada de una materia. A esta pantalla se accede a través de las pantallas de calificaciones y horarios. Se puede observar los resultados de la prueba en la Figura 5.6.
- Cerrar sesión, esta funcionalidad se encuentra en la pantalla Obtener información de cuenta. Permite al estudiante terminar la sesión. El resultado de esta funcionalidad se puede observar en la Figura 5.7.

5.2.3 RESULTADOS DE LA APLICACIÓN *SMART CLIENT*

Para los casos de pruebas definidos en la sección 5.1 y utilizando la cuenta del estudiante Hernández Carlos, se obtuvieron los siguientes resultados:

- Iniciar sesión, la pantalla de inicio de sesión permite ingresar al sistema, para lo cual se debe proporcionar el número de cédula y número único del estudiante. El resultado de la prueba se muestra en la Figura 5.8.



Figura 5.8 Prueba iniciar sesión en la aplicación *Smart Client*

- Consultar información de cuenta, en esta pantalla el estudiante puede ver sus datos personales. El resultado de esta prueba se muestra en la Figura 5.9.
- Consultar horario, el estudiante accede a esta pantalla seleccionando la opción Horario del menú, la misma que muestra el horario de clases organizado por días. Los nombres de las materias son vínculos que permiten obtener información detallada de la materia. Se puede observar el resultado de esta prueba en la Figura 5.10.



Figura 5.9 Prueba Consultar información de cuenta en la aplicación *Smart Client*



Figura 5.10 Prueba Consultar horario en la aplicación *Smart Client*

- Consultar calificaciones, el estudiante accede a esta pantalla al seleccionar la opción Calificaciones del menú. En esta pantalla (Figura 5.11) se muestran las calificaciones obtenidas por el estudiante. Los nombres de las materias son vínculos que permiten obtener información detallada de la materia.



Figura 5.11 Prueba Consultar calificaciones en la aplicación *Smart Client*



Figura 5.12 Prueba Consultar información de materia en la aplicación *Smart Client*

- Consultar información de materia, se accede a la pantalla de información detallada de una materia al seleccionar el nombre de la materia ya sea desde la pantalla de horario o desde la pantalla de calificaciones. La información que se obtuvo en esta prueba se muestra en la Figura 5.12.
- Cerrar sesión, el estudiante puede cerrar su sesión ingresando al menú archivo y seleccionando la opción Cerrar sesión como se muestra en la Figura 5.13. Mediante esta acción el estudiante termina la sesión.



Figura 5.13 Prueba cerrar sesión en la aplicación *Smart Client*

5.3 MEDIDAS DE COMPARACIÓN

En esta sección se indicarán las pruebas de rendimiento a realizarse con el objetivo de determinar qué aplicación tiene un mejor desempeño.

Las mediciones permiten determinar el rendimiento de cada una de las aplicaciones, ayudando a encontrar “cuellos de botella” que afecten el correcto funcionamiento de las aplicaciones.

Para realizar una comparación entre las aplicaciones se utilizarán métricas que ayuden a determinar la eficiencia, el tiempo de respuesta o latencia y la utilización de recursos, que cada aplicación produce.

5.3.1 HERRAMIENTAS A UTILIZARSE

Para efectuar las pruebas de rendimiento se utilizan dos tipos de herramientas. Una para generar tráfico o requerimientos a los servidores, y otra para medir la actividad de los servidores.

Para generar tráfico se utilizó la herramienta *Microsoft Web Application Stress (WAS)*; esta aplicación tiene como objetivo simular múltiples requerimientos de

usuario hacia un servidor Web. Es usada por desarrolladores Web para probar sus aplicaciones y determinar sus límites. La interfaz de esta aplicación se muestra en la Figura 5.14.

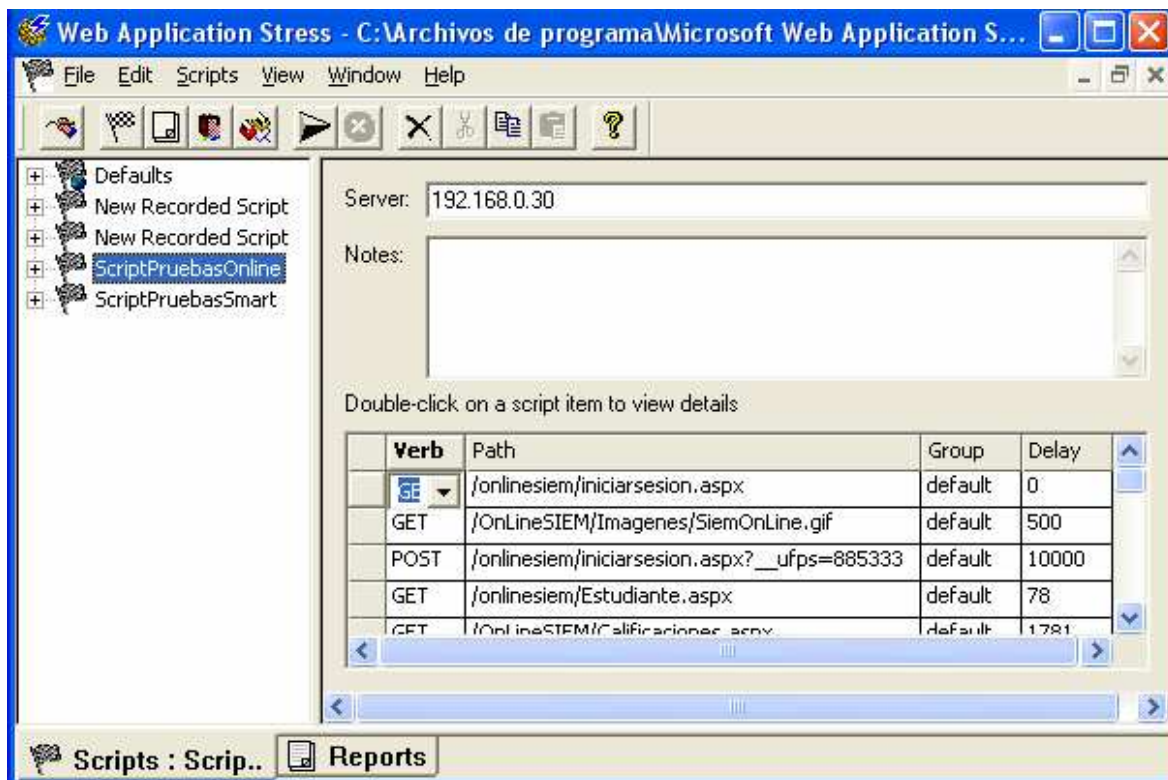


Figura 5.14 Web Application Stress

Información adicional sobre la herramienta *Web Application Stress* se encuentra en el anexo C-1.

Cuando se generan requerimientos hacia el servidor es necesario medir la actividad que el servidor tiene, para esto se utiliza la herramienta *Microsoft Management Console*. Esta aplicación cuenta con dos utilidades que permiten medir parámetros de rendimiento en el servidor; estas utilidades son el *Monitor del sistema* y la utilidad de *Registro y alertas de rendimiento*.

La herramienta proporciona información detallada del uso de los recursos del sistema y permite conocer el efecto que tiene la carga de requerimientos sobre los recursos del servidor Web. La Figura 5.15 muestra la interfaz de la aplicación *Microsoft Management Console*.

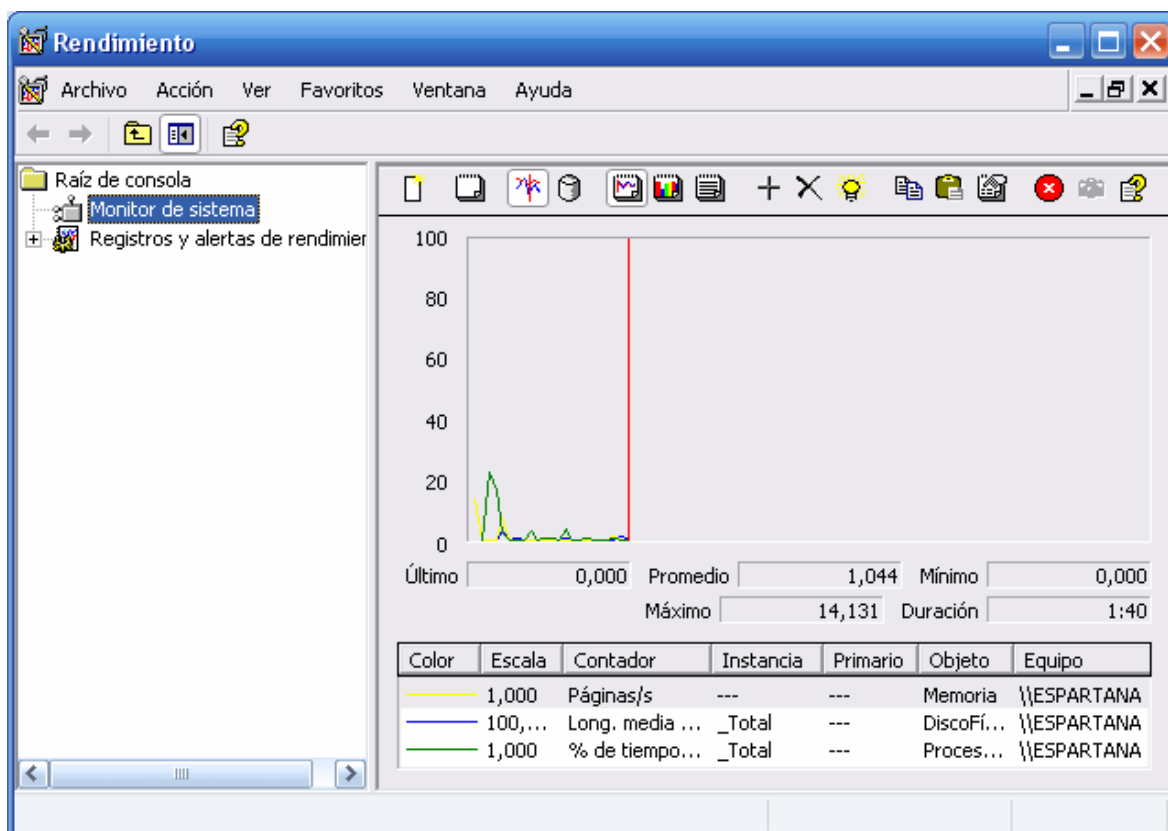


Figura 5.15 Microsoft Management Console

Información adicional sobre la herramienta *Management Console* se encuentra en el anexo C-2.

5.3.2 AMBIENTE DE PRUEBAS DE RENDIMIENTO

Para realizar la comparación de rendimiento de las aplicaciones *Online* y *Smart Client*, es importante definir algunas características del ambiente en el cual se desarrollarán.

Las características del servidor y los equipos cliente, se muestran en la Tabla 5.3 y la Tabla 5.4 respectivamente.

Hardware	Procesador	<i>Intel Core 2 2.0 GHz</i>
	Memoria RAM	1 GB
	Disco Duro	300 GB

	Red	100 Mbps
Software	Sistema operativo	<i>Windows 2003 Server</i>
	Servidor Web	<i>Internet Information Server 6.0</i>
	Servidor de base de datos	<i>SQL Express Edition</i>
	Software de medición	<i>Management Console</i>

Tabla 5.3 Características del servidor

Hardware	Procesador	<i>Intel Pentium 4 2.4 GHz</i>
	Memoria RAM	512 MB
	Disco Duro	100 GB
	Red	100 Mbps
Software	Sistema operativo	<i>Windows XP</i>
	Generador de tráfico	<i>Web Application Stress</i>

Tabla 5.4 Características del equipo cliente

La Figura 5.16, muestra un diagrama de la configuración del ambiente de pruebas de rendimiento para la aplicación *Online*.

La Figura 5.17 muestra un diagrama de la configuración del ambiente de pruebas de rendimiento para la aplicación *Smart Client*.

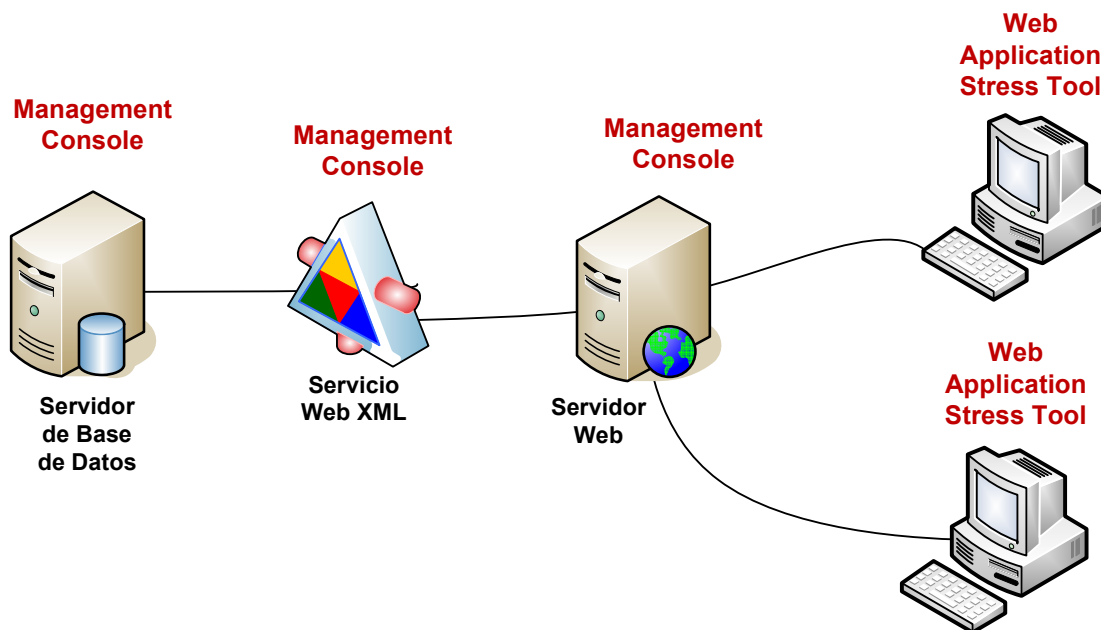


Figura 5.16 Configuración del ambiente de pruebas para la aplicación *Online*

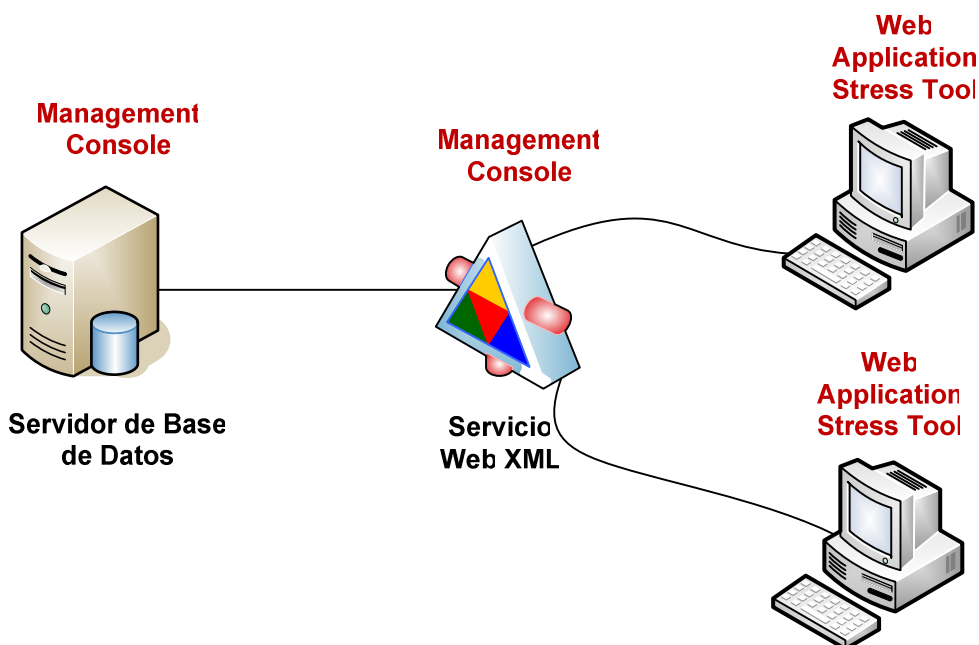


Figura 5.17 Configuración del ambiente de pruebas para la aplicación *Smart Client*

5.3.3 CARGA DE TRABAJO DE LAS PRUEBAS DE RENDIMIENTO

La carga de trabajo que genera un usuario en el servidor, consiste en el conjunto de acciones que realiza para obtener la información deseada del sistema. Para las pruebas de rendimiento se consideran las siguientes actividades o carga de trabajo por parte del usuario:

- Iniciar sesión.
- Revisar su información personal.
- Revisar su horario.
- Revisar sus calificaciones.
- Revisar el detalle de cada materia tomada.
- Salir del sistema.

Son convenientes cargas de trabajo que se incrementen gradualmente, de tal manera que permita establecer la actividad en el servidor dependiendo de la carga de trabajo.

El número de usuarios o carga de trabajo será incrementado progresivamente con una cantidad de 300 usuarios hasta llegar a 3600 usuarios, teniendo un total de 13 pruebas para cada aplicación; cada una de estas pruebas tiene una duración de 30 minutos.

5.3.4 MÉTRICAS PARA LAS PRUEBAS

Las métricas consisten de los parámetros que serán censados del lado del servidor y del lado del cliente; se realizarán las siguientes mediciones:

- Tiempos de respuesta.
- Eficiencia.
- Utilización de recursos.

A continuación se detallarán las métricas para cada uno de los casos señalados.

5.3.4.1 Tiempos de respuesta

El tiempo de respuesta o latencia, es el tiempo que el servidor tarda en responder a un requerimiento del cliente.

Este tiempo es la suma de la latencia en el servidor, la espera en la cola para que el requerimiento sea atendido y la latencia en la red. Es el tiempo de espera que el usuario experimenta.

El tiempo de respuesta será evaluado por las métricas descritas en la Tabla 5.5.

Métrica	Descripción	Umbral
TTFB (<i>Time To First Byte</i>)	Tiempo que tarda en llegar el primer byte de respuesta al cliente. Este tiempo es medido en milisegundos.	10 segundos.
TTLB (<i>Time To Last Byte</i>)	Tiempo que tarda en llegar el último byte de la respuesta al cliente. Este tiempo es medido en milisegundos.	10 segundos.

Tabla 5.5 Parámetros de medición de tiempos de respuesta ⁽⁷⁾

“El tiempo de espera aceptable para usuarios de sitios Web es 3 segundos, hasta un máximo de 10 segundos”. ⁽¹¹⁾

5.3.4.2 Eficiencia

La eficiencia se mide por el número de requerimientos que una aplicación puede servir exitosamente por unidad de tiempo.

Las métricas utilizadas para medir la eficiencia en el servidor se describen en la Tabla 5.6.

Debe señalarse que no existe un umbral o límite establecido para las métricas de eficiencia, pues éstas dependen de la aplicación y el hardware del servidor sobre el que se ejecuta la aplicación. En otras palabras, una aplicación que ejecute una lógica complicada tendrá una eficiencia menor, o menor número de usuarios atendidos por segundo, pero esto puede variar si la misma aplicación es ejecutada en un hardware de mejores características.

Objeto	Métrica	Descripción	Umbral
Aplicaciones ASP.NET	Solicitudes en ejecución	Número de requerimientos existentes concurrentemente. Este contador se incrementa cuando el requerimiento está siendo procesado y se decrementa cuando el requerimiento ya ha sido procesado.	Depende de la aplicación y el hardware del servidor.
	Solicitudes por segundo	Número de solicitudes ejecutadas por segundo. Ayuda a medir la eficiencia del servidor de aplicaciones ASP.NET.	Depende de la aplicación y el hardware del servidor.
Servicio Web	Peticiones de extensión ISAPI/s	Mide la tasa de peticiones de extensión ISAPI ¹⁰ que están siendo procesadas de forma simultánea por el servicio Web.	Depende de la aplicación y el hardware del servidor.

Tabla 5.6 Parámetros de medición de eficiencia ⁽⁷⁾

5.3.4.3 Utilización de recursos

5.3.4.3.1 Procesador

Objeto	Métrica	Descripción	Umbral
Procesador	% de tiempo de procesador	Esta medida es el principal indicador de la actividad del procesador. Indica el grado de utilización del procesador.	85%.
	% Tiempo privilegiado	Indica el porcentaje de tiempo en que un hilo se ejecuta en modo privilegiado, esto sucede cuando la aplicación utiliza determinados servicios del SO ¹¹ .	75%.
	% de tiempo de interrupción	Indica el porcentaje de tiempo que el procesador ocupa para atender y servir interrupciones de hardware.	Depende del procesador.
Sistema	Longitud de la cola del procesador	Indica cuántos hilos de ejecución están listos para ejecutarse, pero están en cola de espera porque otro hilo está siendo ejecutado.	2.
	Cambios de s ¹² .	Ocurre cuando un hilo de alta prioridad toma el lugar de un hilo de más baja prioridad que se está ejecutando o cuando un hilo de alta prioridad se bloquea. Indica el grado de competencia que tienen los hilos para la utilización del procesador.	Hasta 15000 / segundo / procesador

Tabla 5.7 Parámetros de medición de utilización del procesador ⁽⁷⁾

¹⁰ ISAPI: *Internet Server Application Programming Interface*. Es una DLL que puede ser cargada y llamada por un servidor HTTP.

¹¹ SO: Sistema Operativo.

¹² Sistema Cambios de s: Cambios de contexto que ocurren en el sistema operativo.

La Tabla 5.7 muestra las métricas a ser cuantificadas en el procesador.

El porcentaje de tiempo de interrupción en el procesador, no tiene un límite establecido, pero se debe mencionar que este valor debe ser bajo, ya que indica las interrupciones que son generadas por dispositivos periféricos; si este valor es alto puede deberse a problemas con el hardware.

5.3.4.3.2 Memoria

En la Tabla 5.8 se indican las métricas a ser tomadas para medir la utilización de memoria.

Métrica	Descripción	Umbral
MBytes disponibles	Indica la memoria física disponible para la ejecución de los procesos.	Menor al 25 % de la RAM instalada.
Errores de caché/s.	Indica con qué frecuencia el sistema operativo busca datos en caché y no los encuentra.	No existe un valor definido.

Tabla 5.8 Parámetros de medición de utilización de memoria ⁽⁷⁾

5.3.4.3.3 E/S Disco

Métrica	Descripción	Umbral
Long. Media de la cola de disco	Indica el promedio de requerimientos de lectura escritura que están en la cola para un determinado disco.	2
Long. promedio de cola de lectura	Indica el número promedio de requerimientos de lectura esperando en la cola del disco.	2
Long. promedio de cola de escritura	Indica el número promedio de requerimientos de escritura esperando en la cola del disco.	2
Media en segundos/lectura	Indica el tiempo promedio en realizar una operación de lectura desde disco.	No existe un valor establecido.
Escrituras en s ¹³ .	Velocidad de operación de escritura en el disco.	No existe un valor establecido.

Tabla 5.9 Parámetros de medición de utilización de E/S Disco ⁽⁷⁾

¹³ Escrituras en s: Escrituras de disco por segundo.

En la Tabla 5.9 se describen las métricas a ser tomadas para medir la utilización del disco duro.

5.3.4.3.4 Recursos de red

En la Tabla 5.10 se hace una descripción de las métricas a ser tomadas para medir la utilización de los recursos de red.

Métrica	Descripción	Umbral
Total de bytes/s.	Este contador indica la velocidad con la que los bytes son enviados y recibidos sobre la red.	80% de la velocidad de la red
Bytes recibidos/s.	Este contador indica la velocidad con la que los bytes son recibidos sobre la red.	Sumados a bytes enviados/s no debe sobrepasar el 80% de la capacidad de la red.
Bytes enviados/s.	Este contador indica la velocidad con la que los bytes son enviados sobre la red.	Sumados a bytes recibidos/s no debe sobrepasar el 80% de la capacidad de la red.

Tabla 5.10 Parámetros de medición de utilización de E/S Red ⁽⁷⁾

5.3.5 MÉTRICAS OBTENIDAS DE LA APLICACIÓN ONLINE

En el anexo C-3 se presentan tablas detalladas donde se encuentran todas las métricas obtenidas al medir la aplicación *Online*.

5.3.5.1 Tiempos de respuesta

Los resultados tabulados para medir los tiempos de respuesta de la aplicación *Online* se los puede observar en la Tabla 5.11. Como se puede apreciar, los tiempos de respuesta que se obtuvieron van aumentando según el número de usuarios. Al comparar estos valores con el umbral de 10 segundos se puede observar que a partir de los 2700 usuarios, este tiempo es superado.

Se puede decir entonces, que la aplicación puede atender menos de 2700 usuarios para poder responder a las solicitudes sin que el usuario experimente un tiempo de respuesta mayor a los 10 segundos.

Número de Usuarios Simultáneos	TTFB Avg (ms)	TTLB Avg (ms)
1	4.8791	5.0478
300	573.6700	580.2277
600	2450.3885	2491.9800
900	3415.3597	3516.5692
1200	4849.6346	5006.8955
1500	4932.8238	5148.9962
1800	6459.8123	6689.5243
2100	7152.7579	7444.3896
2400	9919.4201	10283.6004
2700	19636.0350	20333.9476
3000	11613.7397	12172.4421
3300	12391.9894	12986.1127
3600	13681.7838	14329.3168

Tabla 5.11 Tiempos de respuesta de la aplicación *Online*

5.3.5.2 Eficiencia

La Tabla 5.12 muestra los resultados obtenidos de las métricas que ayudan a determinar la eficiencia de la aplicación *Online*.

Se debe señalar que la herramienta *Management Console* hace mediciones en el servidor cada 15 segundos. La métrica solicitudes en ejecución, mide los requerimientos en ejecución al instante en que la herramienta realiza la medida, mientras que las otras dos métricas son el resultado de un promedio de las medidas tomadas durante los 15 segundos.

Como se señaló anteriormente, las métricas no tienen un valor umbral establecido, pero a través de las pruebas se puede indicar que el tope de eficiencia del servidor es de 221 solicitudes/s que corresponde a la carga generada por 600 usuarios; esto significa que cuando los usuarios superan los 600, los tiempos de respuesta hacia los clientes irán aumentando.

Número de usuarios simultáneos	Aplicaciones ASP.NET		Servicio Web
	Solicitudes en ejecución	Solicitudes por segundo	Peticiones de extensión ISAPI/s
1	0.0000	0.6867	0.6789
300	56.4750	175.0731	173.5489
600	294.7250	221.4875	219.7073
900	654.6417	227.3451	225.3617
1200	1019.2583	238.5760	236.2987
1500	1460.0833	238.7540	236.3327
1800	2129.2083	243.5451	241.1715
2100	2547.8167	249.9970	248.0283
2400	2296.3917	276.5132	274.1443
2700	3079.5083	265.3588	265.3599
3000	3220.3000	276.2657	273.9423
3300	3734.6500	280.4383	277.8987
3600	4119.2750	286.2162	283.7392

Tabla 5.12 Eficiencia de la aplicación *Online*

5.3.5.3 Utilización de recursos

5.3.5.3.1 Procesador

La Tabla 5.13 muestra la utilización del procesador cuando ejecuta la aplicación *Online*.

Al comparar cada métrica con el umbral establecido en la sección 5.3.4.3.1, se puede señalar que:

- El porcentaje de tiempo de interrupción del procesador se mantiene bajo, por lo que se puede decir que no existen inconvenientes con el hardware periférico del servidor.
- El porcentaje de tiempo del procesador en ningún caso supera el 85% establecido como umbral.
- El porcentaje de tiempo privilegiado en ninguna de las pruebas alcanza el 75% establecido como umbral.

- Los cambios de contexto del sistema superan el valor límite establecido cuando existen 1800 o más usuarios simultáneos. Esto quiere decir que existen muchos procesos con la misma prioridad que compiten por el procesador.
- Con respecto a la longitud de la cola del procesador se puede observar que desde los 300 usuarios los procesos empiezan a encolarse, y superan el umbral de 2 procesos en cola.

Número de usuarios simultáneos	Procesador			Sistema	
	% de tiempo de interrupción	% de tiempo de procesador	% Tiempo privilegiado	Cambios de s.	Longitud de la cola del procesador
1	0.0061	0.1385	0.0330	588.6926	0.0417
300	0.5104	16.6389	4.0317	7048.3181	3.5167
600	0.6853	22.1923	5.3879	8451.0641	2.6417
900	0.6016	21.6250	5.1892	9213.3948	3.1417
1200	0.6163	22.6392	5.3785	11696.5333	5.4167
1500	0.6636	22.8179	5.3761	11599.2367	3.8167
1800	0.7560	26.0826	5.9100	15400.6500	6.4583
2100	0.7934	28.4327	6.3321	17300.3734	7.3000
2400	0.9531	30.2610	6.8723	27351.4915	9.0583
2700	1.0022	33.5226	7.3095	25254.9849	3.4250
3000	1.2283	37.4975	8.4202	31263.5467	10.3583
3300	1.2574	37.1832	7.9970	34135.2275	9.4583
3600	1.4779	39.2639	8.3832	42870.8063	11.6500

Tabla 5.13 Utilización del procesador de la aplicación *Online*

5.3.5.3.2 Memoria

La Tabla 5.14 muestra la utilización del recurso de memoria cuando la aplicación *Online* está ejecutándose.

El límite de la métrica MBytes disponibles es menor al 25% de la RAM instalada, para este caso es de 256 MBytes, y como se puede ver, ningún valor es menor a 256 MBytes, lo que indica que no existe una deficiencia de memoria.

Número de usuarios simultáneos	Errores de caché/s.	MBytes disponibles
1	30.5435	518.0917
300	35.1604	592.5917
600	35.8646	551.1667
900	35.2860	503.9750
1200	35.9214	492.5500
1500	35.7918	487.5583
1800	36.0229	472.0417
2100	36.2468	466.5583
2400	37.5553	475.2750
2700	36.9845	460.8167
3000	37.2961	460.0333
3300	37.4575	450.9250
3600	37.8197	445.8417

Tabla 5.14 Utilización de memoria de la aplicación *Online*

5.3.5.3.3 E/S Disco

Número de usuarios simultáneos	Escrituras en s.	Longitud media de la cola de disco	Longitud promedio de cola de escritura de disco	Longitud promedio de cola de lectura de disco	Media en segundos/lectura
1	2.27730	0.00296	0.00240	0.00056	0.00281
300	3.04345	0.00562	0.00505	0.00057	0.00286
600	3.16227	0.00561	0.00345	0.00215	0.00272
900	2.84672	0.00310	0.00288	0.00022	0.00305
1200	3.01508	0.00390	0.00365	0.00025	0.00264
1500	2.94776	0.00333	0.00301	0.00033	0.00375
1800	2.95402	0.00332	0.00283	0.00050	0.00183
2100	2.87451	0.00295	0.00274	0.00020	0.00266
2400	3.07723	0.00306	0.00281	0.00025	0.00213
2700	3.09233	0.00567	0.00301	0.00265	0.00279
3000	2.87337	0.00383	0.00370	0.00013	0.00119
3300	2.89181	0.00317	0.00289	0.00028	0.00288
3600	2.95239	0.00320	0.00301	0.00019	0.00100

Tabla 5.15 Utilización de disco de la aplicación *Online*

En la Tabla 5.15 se puede observar los datos de utilización de disco cuando se ejecuta la aplicación *Online*.

Comparando cada métrica con sus valores umbrales se puede decir que en ningún caso se sobrepasa este límite.

Analizando las métricas de longitud promedio de cola de escritura de disco y longitud promedio de lectura de disco se observa que los datos obtenidos son menores a 2, lo que indica que no existen problemas con el disco.

5.3.5.3.4 Recursos de Red

Los datos de la utilización de recursos de red al manejar la aplicación *Online* se muestran en la Tabla 5.16.

De acuerdo con las características de hardware en donde se ejecuta la aplicación el 80% de la velocidad de la red que se establece como umbral es 10485760 bytes/s, y como se puede ver las medidas realizadas en ningún momento alcanzan este valor, es decir no existe ningún problema con la red.

Número de usuarios simultáneos	Bytes enviados/s.	Bytes recibidos/s.	Total de bytes/s.
1	1979.1319	1269.1850	3248.3169
300	508904.4218	321968.9593	830873.3810
600	644617.6254	408332.6631	1052950.2885
900	658129.9072	419048.7289	1077178.6361
1200	688103.4260	439326.6780	1127430.1040
1500	681975.5686	439409.4870	1121385.0556
1800	695697.7310	448609.9496	1144307.6806
2100	709730.1791	459920.7199	1169650.8990
2400	791827.0306	508875.0605	1300702.0911
2700	751912.5113	488320.2098	1240232.7212
3000	782837.4036	507675.6006	1290513.0043
3300	793018.7639	516073.6314	1309092.3954
3600	805209.0937	525530.5183	1330739.6120

Tabla 5.16 Utilización de red de la aplicación *Online*

5.3.6 MÉTRICAS OBTENIDAS DE LA APLICACIÓN *SMART CLIENT*

En el anexo C-4 se presentan las tablas de las métricas obtenidas al medir la aplicación *Smart Client*.

5.3.6.1 Tiempos de respuesta

La Tabla 5.17, muestra los tiempos de respuesta por número de usuarios simultáneos de la aplicación *Smart Client*.

Al comparar con el umbral los tiempos de respuesta obtenidos con la aplicación *Smart Client*, se puede ver que los valores no se acercan a dicho umbral. El mayor tiempo de espera es menor a 1 segundo, lo que significa que el tiempo de respuesta que los usuarios perciben está dentro de los límites aceptables.

Número de usuarios simultáneos	TTFB Avg (ms)	TTLB Avg (ms)
1	8.5710	8.6710
300	8.6760	8.7065
600	9.2614	9.2935
900	12.0922	12.1330
1200	18.8275	18.8723
1500	20.5768	20.6258
1800	23.8455	23.8958
2100	31.1445	31.1858
2400	38.9518	38.9999
2700	46.1104	46.1553
3000	55.2760	55.3311
3300	152.2034	152.2997
3600	106.9617	107.0240

Tabla 5.17 Tiempos de respuesta de la aplicación *Smart Client*

5.3.6.2 Eficiencia

La Tabla 5.18 expone los resultados obtenidos de las medidas de eficiencia de la aplicación *Smart Client*.

Número de usuarios simultáneos	Aplicaciones ASP.NET		Servicio Web
	Solicitudes en ejecución	Solicitudes por segundo	Peticiones de extensión ISAPI/s
1	0.0000	0.1550	0.1539
300	0.1500	52.1122	51.5566
600	0.3667	104.4277	103.4487
900	0.5333	156.5891	155.1685
1200	0.7000	208.6992	206.8149
1500	0.8250	260.6431	257.9949
1800	1.2417	312.3321	308.4261
2100	1.4750	364.1462	359.3985
2400	1.4500	415.1493	410.0642
2700	2.1750	466.1868	460.4969
3000	2.5167	516.7388	510.9508
3300	2.4000	566.0034	560.9098
3600	4.3167	612.2243	607.8380

Tabla 5.18 Eficiencia de la aplicación *Smart Client*

Se observa un incremento lineal de la eficiencia del servidor, es decir a mayor número de usuarios mayores solicitudes por segundo atendidas correctamente.

5.3.6.3 Utilización de recursos

5.3.6.3.1 Procesador

Los resultados de utilización del procesador al ejecutar la aplicación *Smart Client*, se exponen en la Tabla 5.19.

Los valores obtenidos de porcentaje de tiempo de interrupción no son altos, es decir, que no existen inconvenientes con el hardware donde se instaló la aplicación. El porcentaje de tiempo del procesador en todas las mediciones con diferente cantidad de usuarios simultáneos no llega al umbral. Los valores obtenidos de porcentaje de tiempo privilegiado se mantienen por debajo del umbral. Con respecto a los cambios de contexto, se puede ver que se supera el umbral cuando se tiene 2400 o más usuarios simultáneos utilizando la aplicación

Smart Client, éste es un indicador de que la actividad del procesador se incrementa a medida de que aumenta el número de usuarios.

Con los datos obtenidos se puede ver que el sistema empieza a poner en cola los procesos a partir de los 900 usuarios, pero se sobrepasa el umbral a partir de los 1800 usuarios.

Número de usuarios simultáneos	Procesador			Sistema	
	% de tiempo de interrupción	% de tiempo de procesador	% Tiempo privilegiado	Cambios de s.	Longitud de la cola del procesador
1	0.0091	0.1519	0.0543	361.2977	0.6000
300	0.1858	5.5029	2.5082	2172.9037	0.8000
600	0.3498	11.3906	5.1593	3618.5951	0.6083
900	0.5026	16.1287	7.4105	5313.9534	1.0417
1200	0.7969	24.9374	11.2754	7056.2742	1.0917
1500	0.8628	26.5213	11.9961	8898.0328	1.6333
1800	1.0760	33.9023	15.3390	11100.9496	2.1417
2100	1.4401	42.7730	19.3712	12662.1500	2.5583
2400	1.4358	42.6996	19.6793	15534.3808	3.4333
2700	1.6927	53.3295	24.0465	16642.5339	3.9583
3000	1.7053	51.2122	23.4549	18923.3901	3.0250
3300	1.9349	63.2284	28.2824	18812.4540	4.1250
3600	2.1318	64.6322	29.7966	22881.8453	5.0083

Tabla 5.19 Utilización del procesador de la aplicación *Smart Client*

5.3.6.3.2 Memoria

La Tabla 5.20 muestra la utilización del recurso de memoria cuando se ejecuta la aplicación *Smart Client*.

Los MBytes disponibles en memoria, van disminuyendo mientras se incrementa el número de usuarios, pero no se llega al umbral, por lo que se puede señalar que existe suficiente memoria.

Número de usuarios simultáneos	Errores de caché/s.	MBytes disponibles
1	6.3774	690.5333
300	4.8014	671.7583
600	11.1647	667.9000
900	9.0906	662.6667
1200	11.7201	658.7750
1500	14.6464	655.5667
1800	17.1819	651.5583
2100	20.8884	650.3500
2400	23.6304	647.9083
2700	26.1980	644.9000
3000	27.6662	642.1500
3300	30.3998	639.5833
3600	32.5366	636.8500

Tabla 5.20 Utilización de memoria de la aplicación *Smart Client*

5.3.6.3.3 E/S Disco

Número de usuarios simultáneos	Escrituras en s.	Longitud media de la cola de disco	Longitud promedio de cola de escritura de disco	Longitud promedio de cola de lectura de disco	Media en segundos/lectura
1	2.4423	0.0030	0.0023	0.0006	0.0005
300	2.6928	0.0031	0.0024	0.0007	0.0028
600	2.9556	0.0037	0.0034	0.0003	0.0025
900	2.9223	0.0029	0.0028	0.0002	0.0022
1200	3.0112	0.0035	0.0029	0.0006	0.0020
1500	3.1296	0.0032	0.0031	0.0001	0.0003
1800	3.2061	0.0034	0.0032	0.0001	0.0007
2100	3.6238	0.0036	0.0034	0.0003	0.0024
2400	3.6822	0.0041	0.0039	0.0001	0.0011
2700	3.8317	0.0039	0.0038	0.0001	0.0008
3000	3.8474	0.0043	0.0039	0.0003	0.0028
3300	4.0657	0.0044	0.0042	0.0002	0.0008
3600	4.1233	0.0067	0.0039	0.0027	0.0014

Tabla 5.21 Utilización de disco de la aplicación *Smart Client*

La Tabla 5.21 muestra la utilización del disco cuando se ejecuta la aplicación *Smart Client*. Como se puede observar los valores de cola de disco son bajos y no alcanzan el umbral establecido de 2.

5.3.6.3.4 Recursos de Red

Los datos obtenidos de utilización de red cuando se ejecuta la aplicación *Smart Client*, se muestran en la Tabla 5.22. Se puede notar que los Bytes totales/s, se incrementan con el número de usuarios simultáneos, pero no llegan al umbral de 10485760 bytes/s.

Número de usuarios simultáneos	Bytes enviados/s.	Bytes recibidos/s.	Total de bytes/s.
1	200.9022	185.3392	386.2414
300	65883.8888	61418.2889	127302.1777
600	131902.0171	123008.2241	254910.2412
900	197839.7704	184490.5231	382330.2935
1200	263733.7581	245845.9859	509579.7440
1500	329550.9695	307126.7945	636677.7639
1800	395006.8549	367943.6360	762950.4909
2100	460516.9872	428898.4947	889415.4819
2400	525429.6320	489082.5434	1014512.1754
2700	590102.8694	549219.7005	1139322.5699
3000	654319.3858	608715.9702	1263035.3560
3300	718018.1565	667194.7799	1385212.9364
3600	776980.0820	722192.8864	1499172.9684

Tabla 5.22 Utilización de red de la aplicación *Smart Client*

5.4 COMPARACIÓN

Las aplicaciones *Online* y *Smart Client* fueron probadas en igualdad de condiciones, obteniéndose el análisis de resultados que se presenta a continuación.

Las figuras que se muestran en esta sección fueron construidas con los datos que se señalan en las secciones 5.3.5 y 5.3.6.

5.4.1 TIEMPOS DE RESPUESTA

Los tiempos de respuesta fueron evaluados en función de las métricas TTFB y TTLB, estos resultados se muestran en la Figura 5.18 y la Figura 5.19 respectivamente.

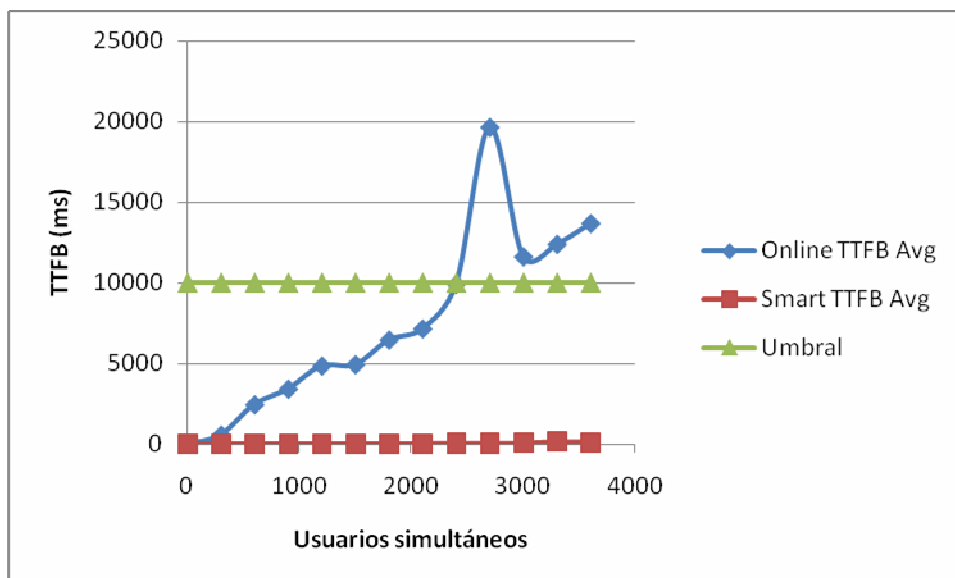


Figura 5.18 Comparación de *Time to first byte* de las aplicaciones

Se nota que al incrementar el número de usuarios los tiempos de respuesta aumentan en cada una de las aplicaciones, pero al comparar los valores entre las aplicaciones, se puede apreciar que en la aplicación *Online* los tiempos son mayores con respecto a la aplicación *Smart Client*. También se puede observar que el umbral establecido en la aplicación *Online* se alcanza cuando se tienen 2400 usuarios y se lo sobrepasa al incrementar el número de usuarios. Mientras que en la aplicación *Smart Client* los valores se mantienen por debajo del umbral.

Entonces al utilizar la aplicación *Smart Client*, el usuario experimenta un tiempo de espera menor que si utiliza la aplicación *Online*, especialmente cuando se sobrepasan los 900 usuarios. Además que al tener 2400 usuarios o más, la aplicación *Online* tiene un tiempo de respuesta mayor al umbral.

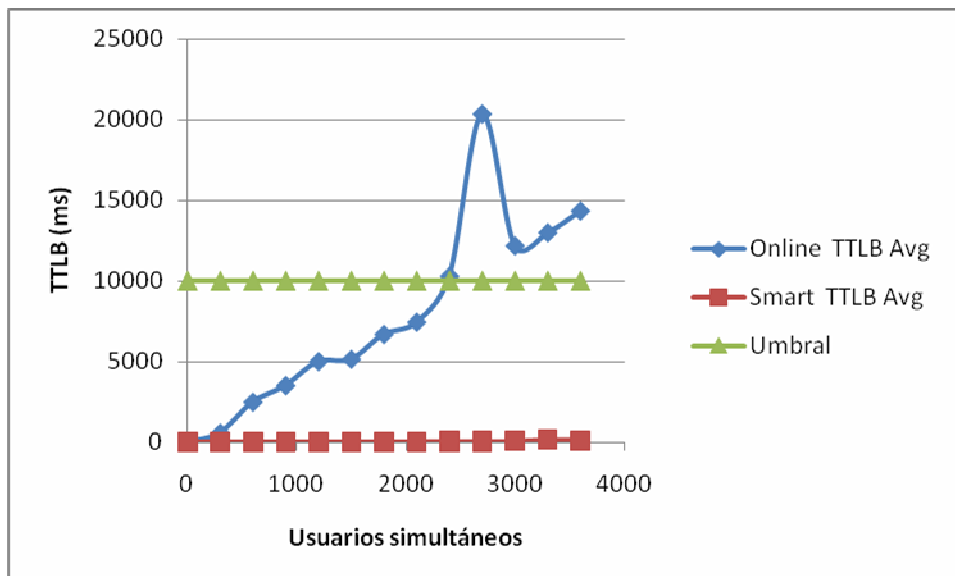


Figura 5.19 Comparación de *Time to last byte* de las aplicaciones

5.4.2 EFICIENCIA

Para comparar la eficiencia entre las aplicaciones se utilizó datos de los resultados de las métricas tanto de la aplicación *Online* como de la aplicación *Smart Client*.

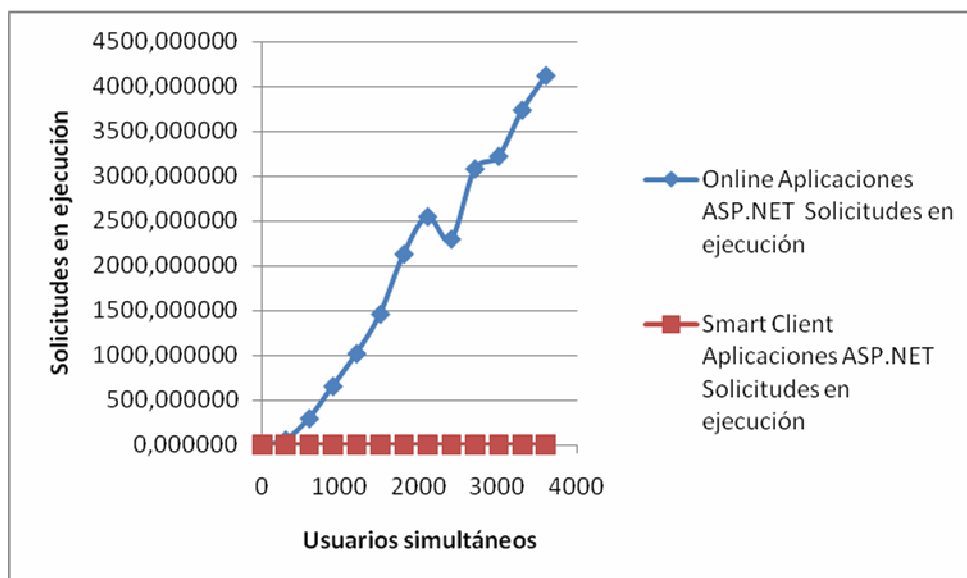


Figura 5.20 Comparación de Solicitudes en ejecución de las aplicaciones

La Figura 5.20 muestra una gráfica de las solicitudes en ejecución de las dos aplicaciones. Como se puede ver, a medida que se incrementan los usuarios simultáneos, el número de solicitudes en ejecución también aumenta en la aplicación *Online*, llegando a tener hasta 4119 solicitudes en ejecución en el caso de 3600 usuarios. Mientras que en la aplicación *Smart Client*, el número de solicitudes también aumenta al incrementar el número de usuarios, pero a diferencia de la aplicación *Online*, el número de solicitudes en ejecución es bajo. Lo cual señala que el servidor puede evacuar con mayor velocidad los requerimientos originados en la aplicación *Smart Client*.

La Figura 5.21 permite ver una comparación de los valores de la métrica solicitudes por segundo de las dos aplicaciones. Se puede observar que a medida que se incrementa el número de usuarios, las solicitudes por segundo que puede atender el servidor, en el caso de la aplicación *Smart Client* también aumentan, mientras que en la aplicación *Online*, el servidor aumenta el número de solicitudes por segundo hasta 600 usuarios, de ahí en adelante, ejecuta casi el mismo valor de solicitudes por segundo.

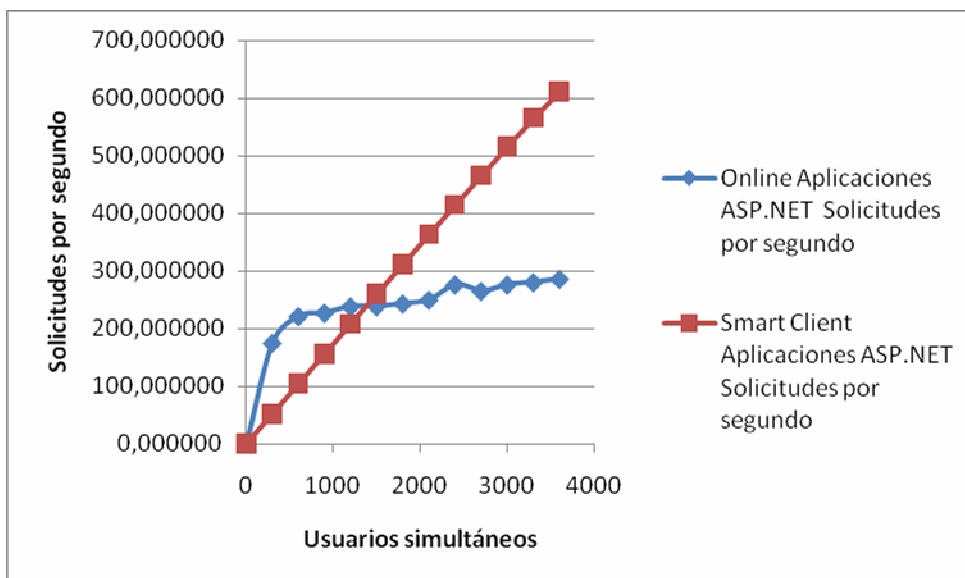


Figura 5.21 Comparación de Solicitudes por segundo de las aplicaciones

La Figura 5.22 permite visualizar los valores obtenidos en ambas aplicaciones de la métrica peticiones de extensión ISAPI/s. Como se puede notar esta gráfica es

similar a la anterior, esto se debe a que ambas métricas dan una medida de la actividad del servidor.

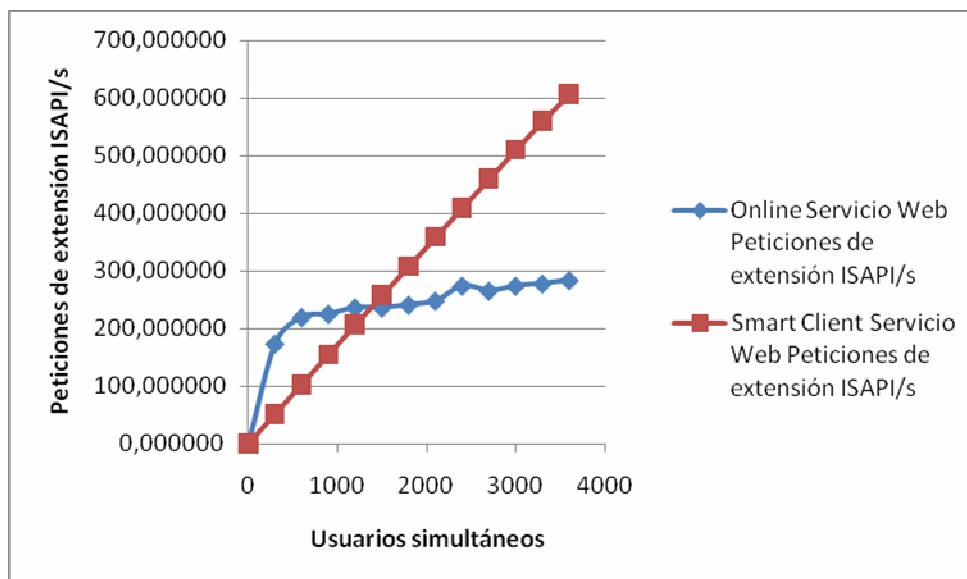


Figura 5.22 Comparación de Peticiones de extensión ISAPI/s de las aplicaciones

A partir del análisis hecho a cada métrica, se puede decir que la aplicación *Online* puede atender menor número de usuarios que la aplicación *Smart Client*, debido a que en el caso de la aplicación *Smart Client* parte de la carga de trabajo es absorbida por el cliente. La carga de trabajo que absorbe el cliente *Smart Client* se debe a la ejecución de la interfaz de usuario.

5.4.3 UTILIZACIÓN DE RECURSOS

5.4.3.1 Procesador

La Figura 5.23 muestra la comparación del porcentaje de tiempo de interrupción del procesador entre las dos aplicaciones. Como se puede ver este valor es bajo en ambas aplicaciones, indicando que no existen inconvenientes en el hardware periférico del servidor.

La Figura 5.24 muestra el gráfico de comparación del porcentaje de tiempo de procesador entre las aplicaciones *Online* y *Smart Client*. Como se puede apreciar ninguna de las dos aplicaciones hace uso del procesador hasta llegar al valor

umbral. También se puede ver que la aplicación *Online* hace mayor uso del procesador hasta cuando tiene una carga de 600 usuarios, de allí en adelante el porcentaje crece lentamente y es menor que en la aplicación *Smart Client*. Mientras que en el caso de la aplicación *Smart Client* el porcentaje de tiempo de procesador crece casi linealmente con respecto al número de usuarios simultáneos.

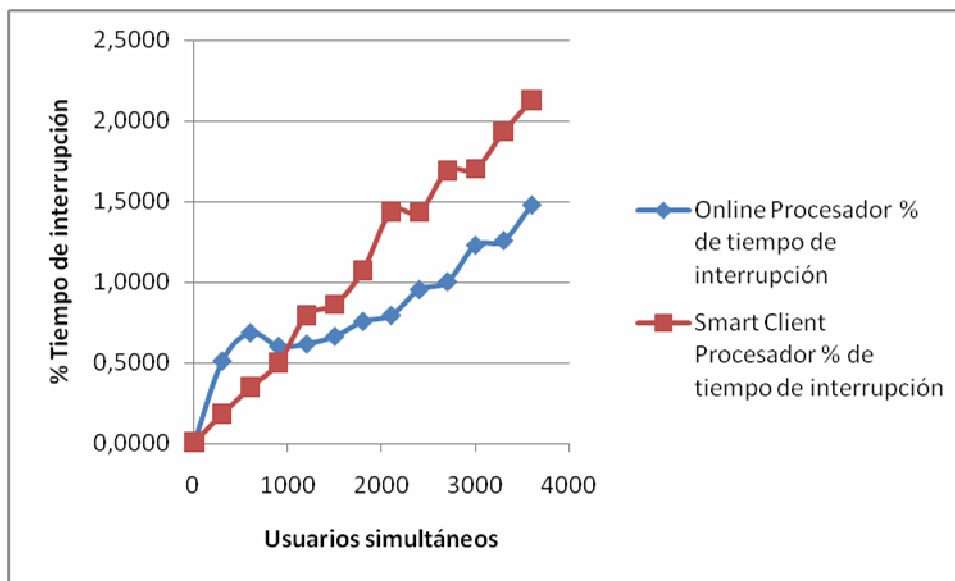


Figura 5.23 Comparación del % de tiempo de interrupción de las aplicaciones

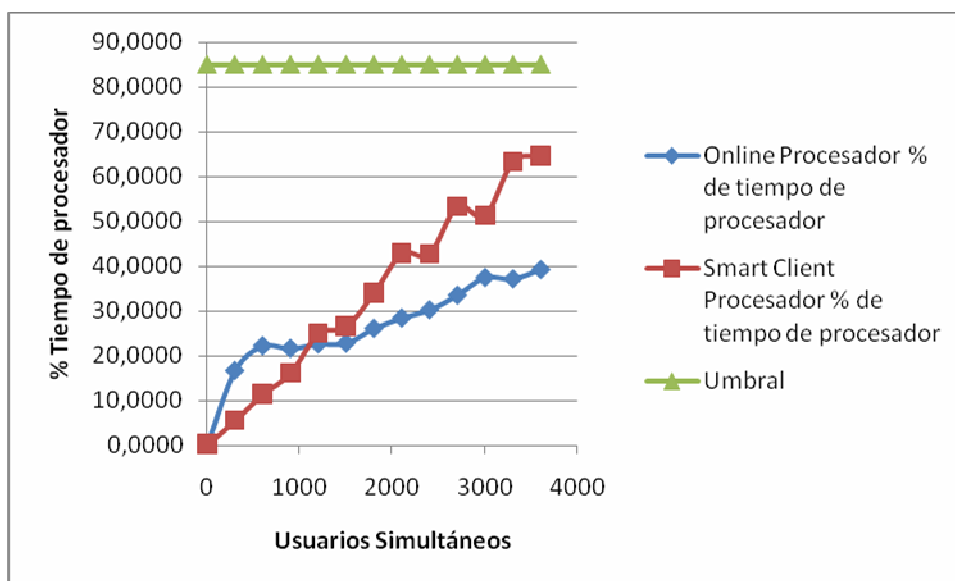


Figura 5.24 Comparación del % de tiempo de procesador de las aplicaciones

La Figura 5.25 permite mostrar el porcentaje del tiempo privilegiado de las aplicaciones *Online* y *Smart Client*. Ninguna de las dos aplicaciones llega al valor umbral. Se puede observar que en la aplicación *Smart Client* los valores son superiores que para los de la aplicación *Online*, lo cual indica que se tiene mayor número de procesos que se ejecutan en modo privilegiado.

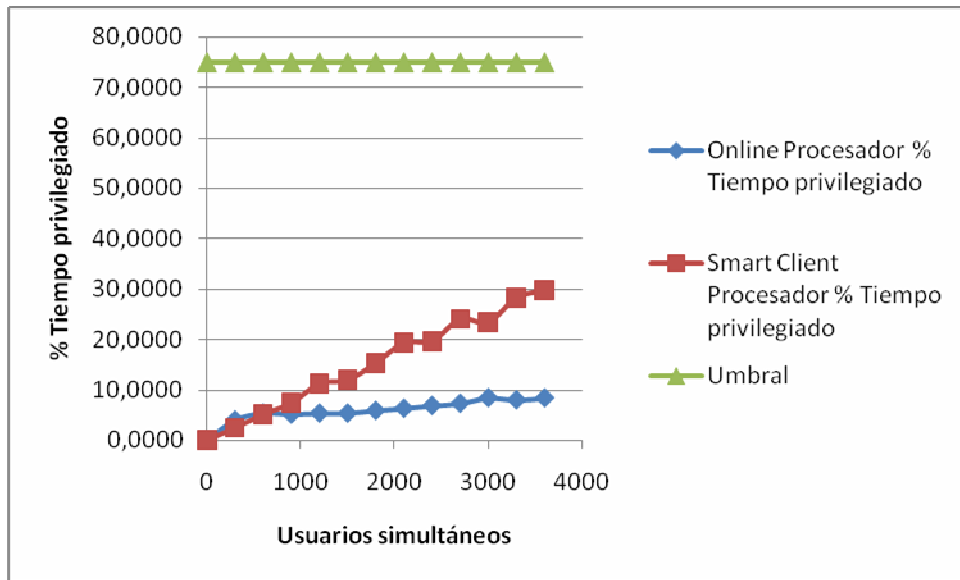


Figura 5.25 Comparación del % de tiempo privilegiado de las aplicaciones

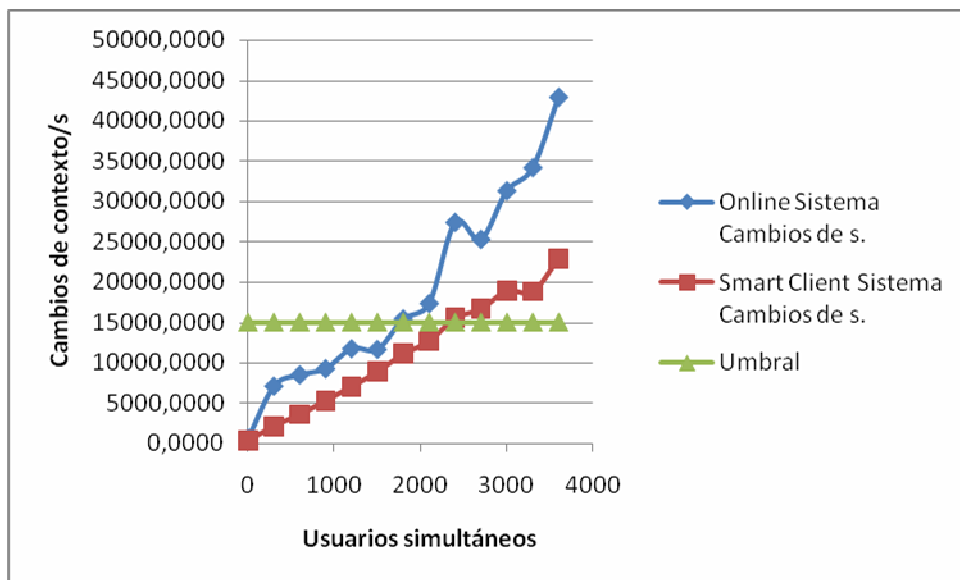


Figura 5.26 Comparación de Cambios de contexto/s de las aplicaciones

En la Figura 5.26 se hace una comparación de los cambios de contexto por segundo del sistema. Como se puede ver estos valores son mayores en la aplicación *Online*, que en la aplicación *Smart Client*, indicando que, existen mayor número de procesos con alta prioridad, haciendo que procesos de baja prioridad que están siendo ejecutados sean interrumpidos para atender a un proceso de alta prioridad, lo que genera mayor cantidad de cambios de contexto. En ambos casos se supera el valor umbral, la aplicación *Online* lo hace a los 1800 usuarios simultáneos y la aplicación *Smart Client* a los 2700 usuarios. Se puede interpretar que existe una competencia por los recursos mayor en la aplicación *Online*, esto a su vez hace que el procesador sea utilizado ineficientemente, pues un proceso es interrumpido enviándolo a la cola de espera.

La Figura 5.27 muestra la longitud de la cola del procesador que presentan ambas aplicaciones. Como se ve, en el caso de la aplicación *Online*, el procesador encola desde los 300 usuarios sobrepasando el valor umbral. La aplicación *Smart Client* se mantiene por debajo del valor umbral hasta llegar a tener una carga de 1800 usuarios, en donde sobrepasa este valor.

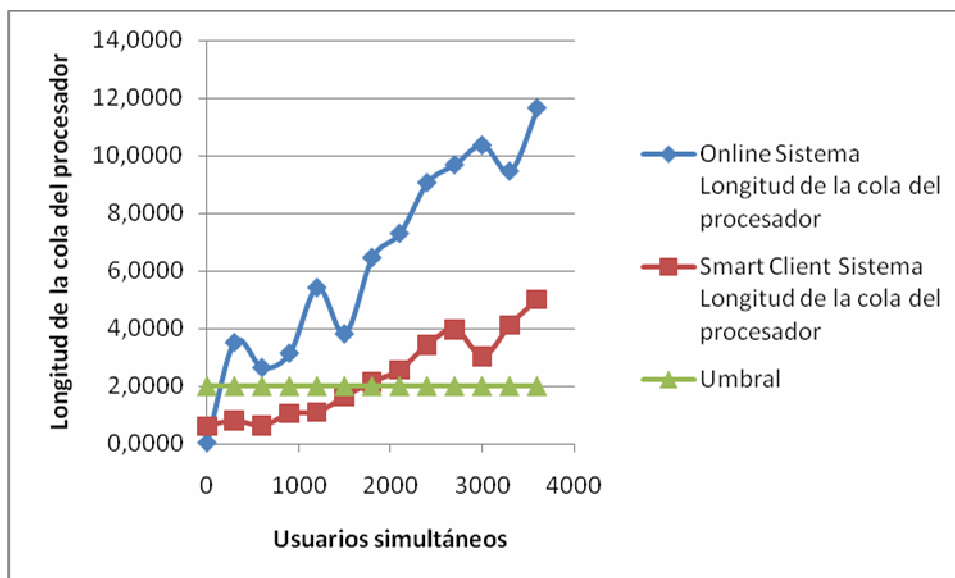


Figura 5.27 Comparación de Longitud de la cola del procesador de las aplicaciones

Los resultados de cola del procesador en las pruebas con diferentes cargas de trabajo son mayores en la aplicación *Online*, esto se debe al mayor número de

cambios de contexto lo que provoca que crezca la cola en el procesador, a pesar de que éste no se ocupe en toda su capacidad.

5.4.3.2 Memoria

La Figura 5.28 expone los resultados obtenidos de la métrica errores de caché/s de la aplicación *Online* así como de la aplicación *Smart Client*. Como se aprecia la aplicación *Online* tiene mayor número de errores de caché en las pruebas con diferente número de usuarios simultáneos que la aplicación *Smart Client*, lo que hace ver que en el caso de la aplicación *Online* el sistema operativo busca por datos en caché y éstos no son encontrados, generándose así mayor número de errores y mayores retardos en el procesamiento. Mientras que en la aplicación *Smart Client* la información buscada en gran parte se encuentra en caché, por lo que ocurre menor número de errores.

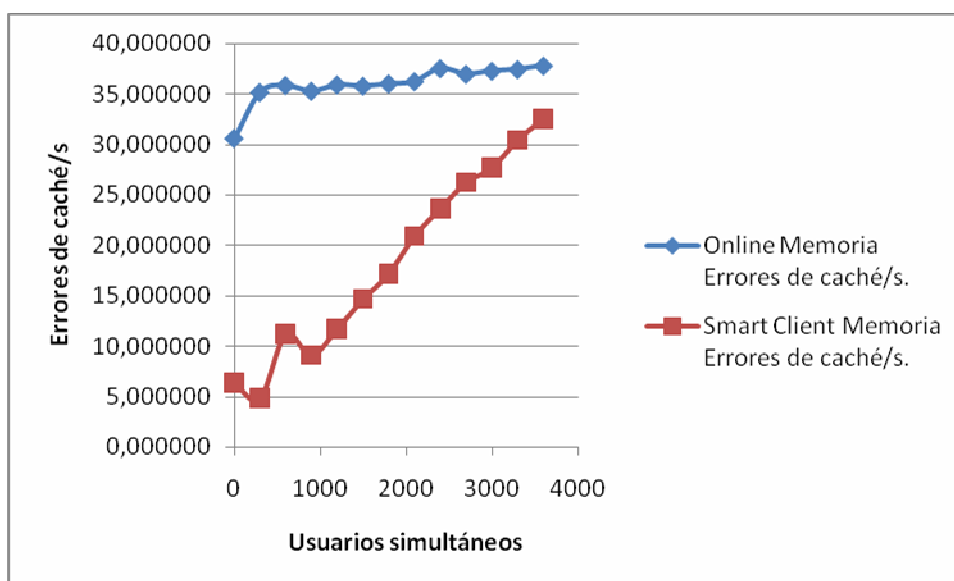


Figura 5.28 Comparación de Errores de caché/s de las aplicaciones

En la Figura 5.29 se hace una comparación de los datos obtenidos de MBytes disponibles de las dos aplicaciones. Lo que se nota en esta gráfica es que en ambos casos existe suficiente memoria y que la aplicación *Online* utiliza más el recurso de memoria que la aplicación *Smart Client*.

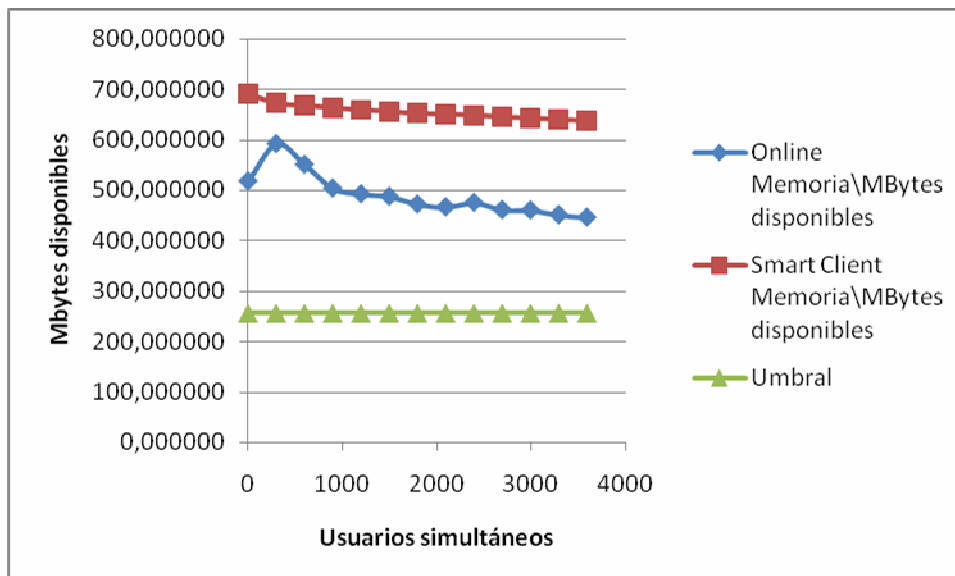


Figura 5.29 Comparación de MBytes disponibles de las aplicaciones

5.4.3.3 E/S Disco

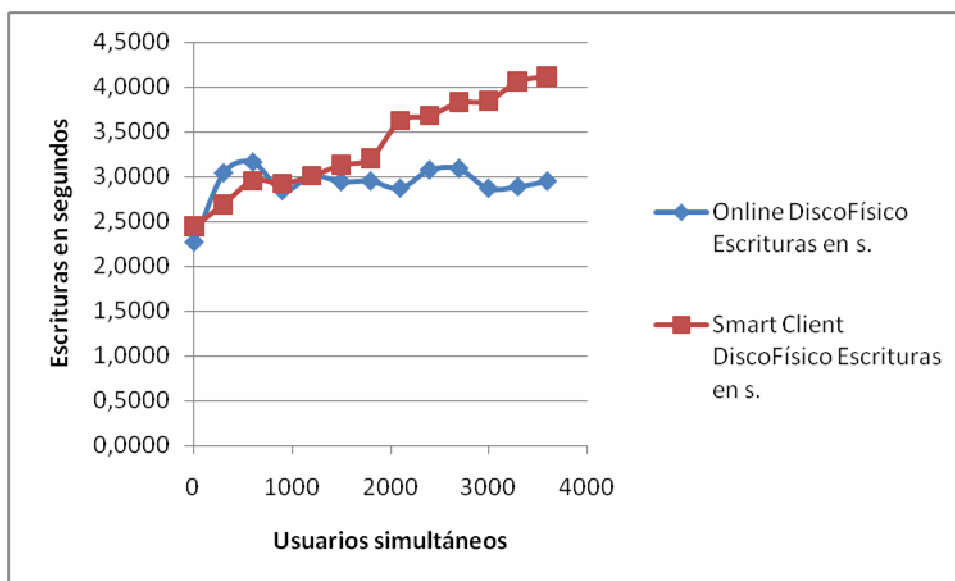


Figura 5.30 Comparación de Escrituras en segundos de las aplicaciones

La comparación de escrituras en segundo en el disco de las aplicaciones se puede apreciar en la Figura 5.30. Aquí se observa que hasta 600 usuarios la aplicación *Online* realiza mayor número de escrituras en el disco, mientras que con mayor número de usuarios, la aplicación *Smart Client* es la que realiza mayor

número de escrituras por segundo en disco. Esta métrica nos indica la velocidad a la que cada aplicación realiza escrituras en disco.

Con respecto a la comparación de métricas de longitudes, total, de escritura y lectura de disco, se puede decir, que en ambos casos no existe cola, ya que los valores obtenidos son menores a 1. Estos resultados se los muestra en la Figura 5.31, Figura 5.32 y la Figura 5.33 respectivamente.

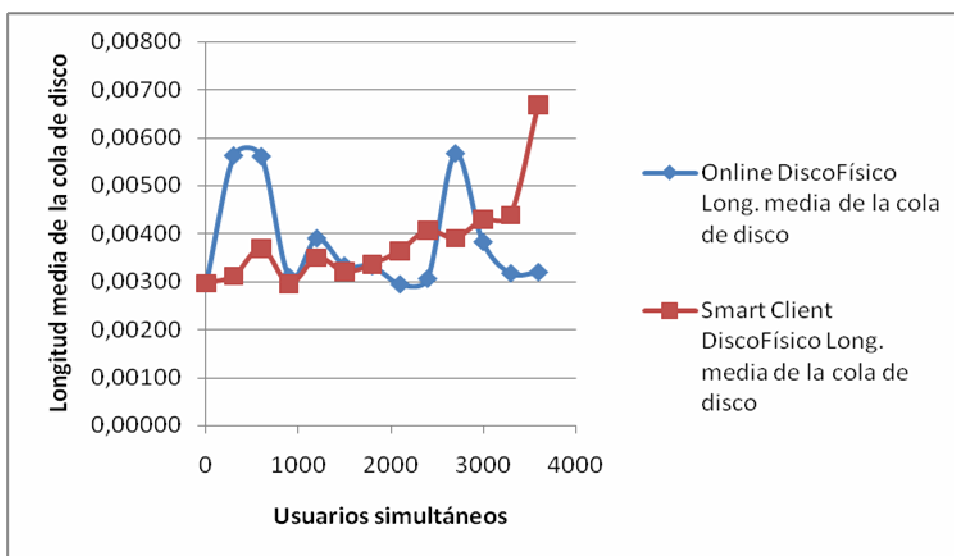


Figura 5.31 Comparación de Longitud media de la cola de disco de las aplicaciones

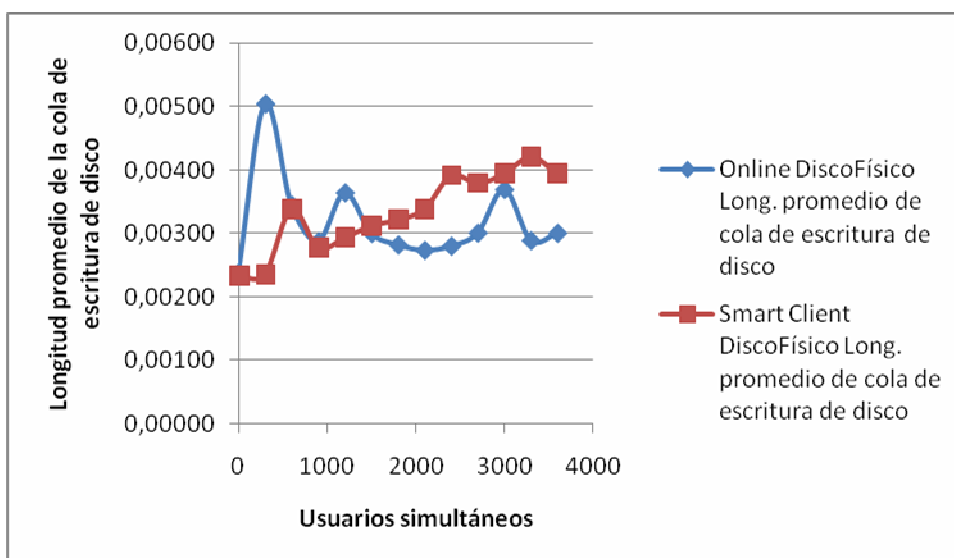


Figura 5.32 Comparación de Long. promedio de cola de escritura de disco de las aplicaciones

La Figura 5.34 muestra una gráfica comparativa de la métrica media en segundos/lectura entre la aplicación *Online* y la aplicación *Smart Client*. Como se puede ver, en la mayor parte de casos la aplicación *Online* tiene valores superiores a los datos de la aplicación *Smart Client*, lo que indica que la aplicación *Smart Client* se demora menos tiempo en realizar una lectura desde disco.

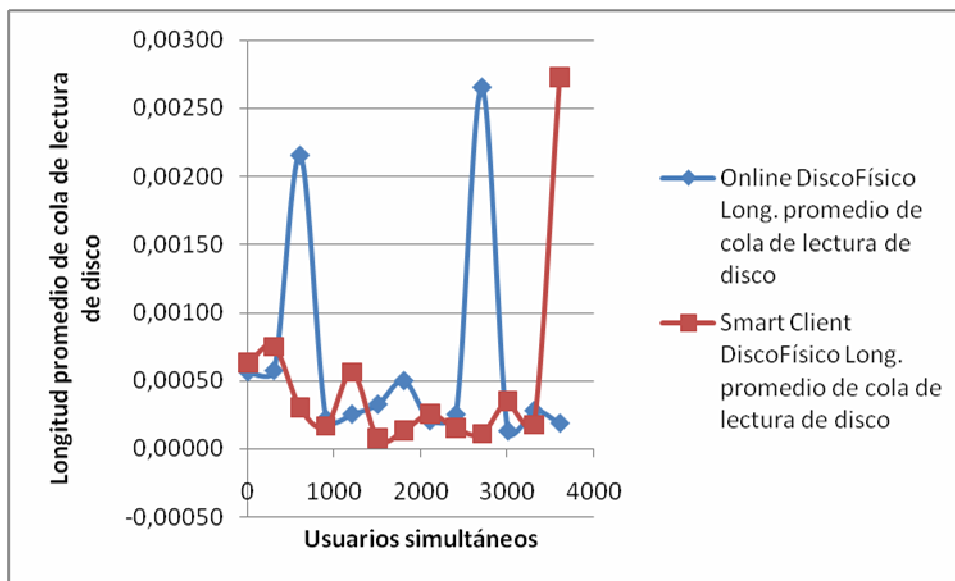


Figura 5.33 Comparación de Long. promedio de cola de lectura de disco de las aplicaciones

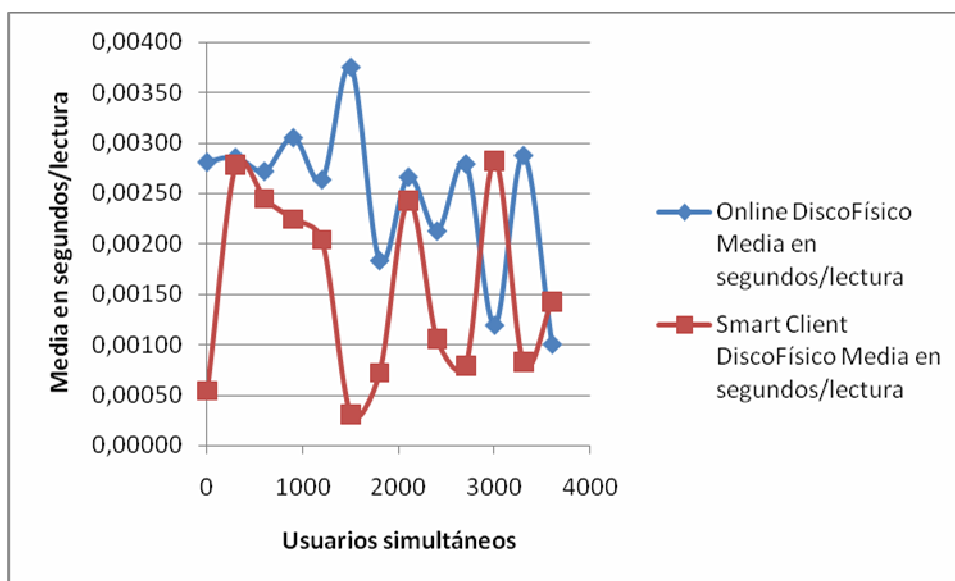


Figura 5.34 Comparación de Media en segundos/lectura de las aplicaciones

5.4.3.4 Recursos de red

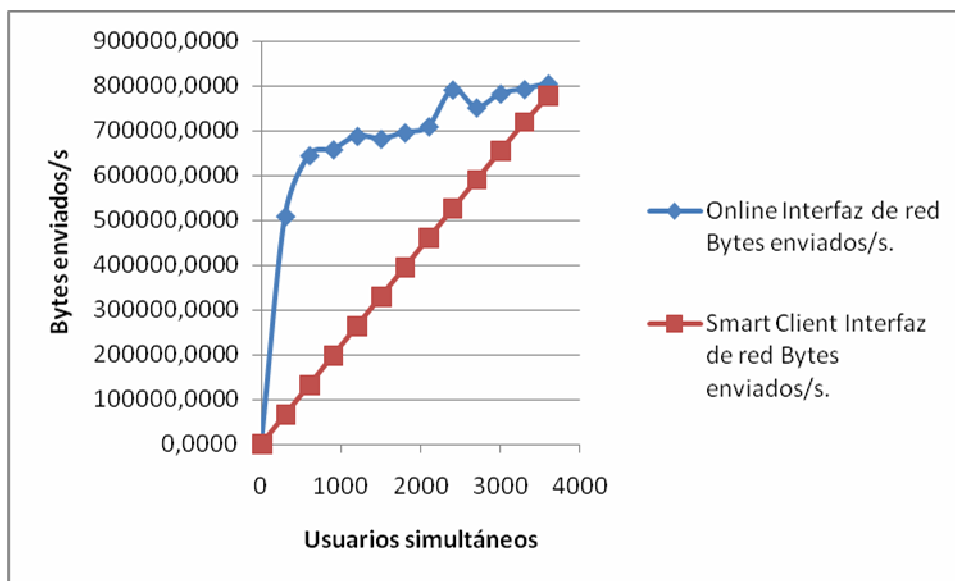


Figura 5.35 Comparación de Bytes enviados/s de las aplicaciones

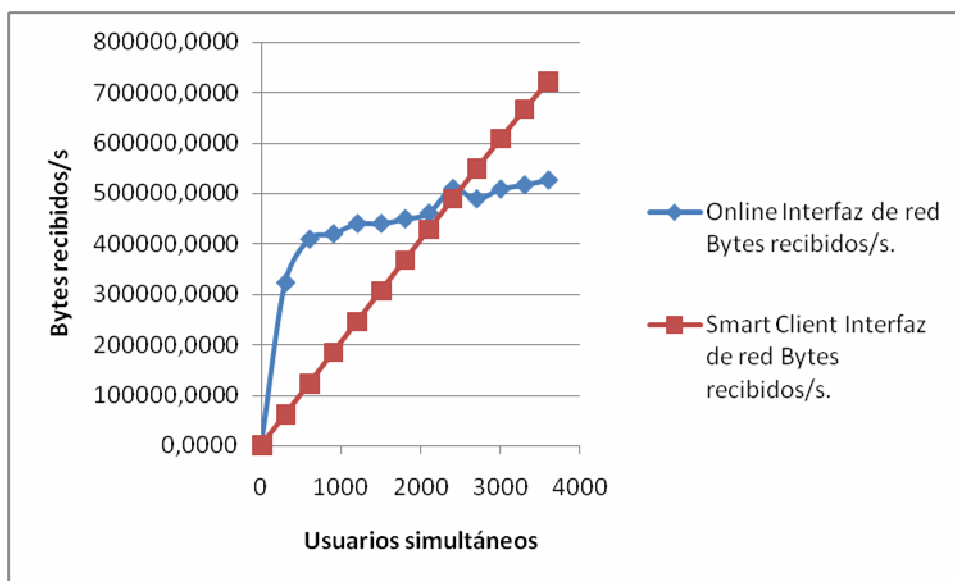


Figura 5.36 Comparación de Bytes recibidos/s de las aplicaciones

Para realizar una comparación de los recursos de red entre las aplicaciones, se realizó gráficas comparativas de las métricas Bytes enviados/s (Figura 5.35), Bytes recibidos/s (Figura 5.36) y Total de bytes/s (Figura 5.37). En ambos casos, se puede decir que no existe ningún problema con el recurso de red, ya que ni la aplicación *Online* ni la aplicación *Smart Client* utilizan la red más del 80% de su

capacidad. También se puede decir que a partir de los 600 usuarios la aplicación *Online* por estar ocupada en el procesamiento de los requerimientos se mantiene en la utilización de recursos de red.

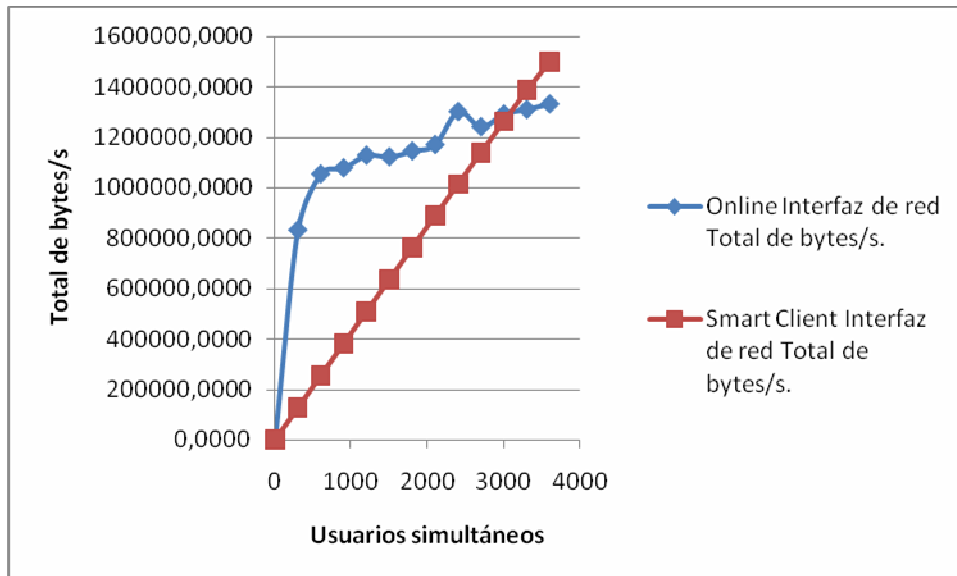


Figura 5.37 Comparación de Total de bytes/s de las aplicaciones

Luego de hacer un análisis comparativo de las distintas métricas que permiten medir la utilización de recursos de las dos aplicaciones, se puede decir que, en el caso de la aplicación *Online*, al sobrepasar los 600 usuarios, la utilización de recursos se estabiliza y se empiezan a acumular peticiones en la cola del procesador, concluyendo que desde este punto se hace un uso ineficiente de los recursos del sistema. Por otro lado, la aplicación *Smart Client*, incrementa el uso de los recursos mientras el número de usuarios se aumenta, llegando a atender mayor cantidad de usuarios que la aplicación *Online*, antes de que el procesador comience a poner procesos en cola.

Entonces se puede decir que la aplicación *Smart Client* hace una utilización de recursos mayor que la aplicación *Online*, pero con la diferencia que los recursos son utilizados de una mejor manera, logrando atender más usuarios. También se puede decir que la aplicación *Smart Client* recibe ayuda de sus clientes, pues éstos se encargan de manejar la interfaz de usuario, lo que aliviana el trabajo del servidor y le permite atender a más usuarios.

5.4.4 FACILIDAD PARA EL DESARROLLO

Las aplicaciones *Online* y *Smart Client* tienen características que las hacen distintas, y por consiguiente el diseño y desarrollo de estas aplicaciones es un tanto diferente.

La aplicación *Online* es una aplicación centralizada que es accedida por medio de un navegador Web; en cambio la aplicación *Smart Client* es una aplicación que se debe distribuir e instalar en cada uno de los dispositivos donde se quiera utilizar.

También se distinguen por el hecho que una aplicación *Online* se ejecuta en un servidor Web, donde se tiene menores limitaciones de hardware que en un dispositivo *Pocket PC*. Por el contrario la aplicación *Smart Client* se ejecuta en un dispositivo Móvil que tiene restricciones de hardware.

En este contexto, el desarrollo de la interfaz de usuario es más sencillo de realizar en la aplicación *Online* por las siguientes razones:

- Los controles de interfaz de usuario de la aplicación *Online* son creados y ejecutados en el servidor, donde existen menores limitaciones de hardware. Por esta razón, los controles tienen muchas funcionalidades ya programadas lo que hace relativamente sencilla la construcción de interfaces de usuario o páginas Web.
- Por otro lado en la aplicación *Smart Client*, hay controles para interfaz de usuario que tienen funcionalidades limitadas, esto hace que sea necesario programar estas funcionalidades.
- Además, para la aplicación *Online* se tiene una amplia variedad de controles de interfaz de usuario, lo que no ocurre para el caso de la aplicación *Smart Client*, donde se cuenta con un número limitado de controles de interfaz de usuario.

5.4.5 FACILIDAD PARA EL MANTENIMIENTO

El mantenimiento del software permite corregir errores que se pudieren haber pasado por alto en la fase de desarrollo, también permite actualizar cierta funcionalidad del software o agregar una funcionalidad adicional.

Los procesos de mantenimiento de una aplicación *Online* y una *Smart Client* son diferentes, esto se debe principalmente al hecho que una aplicación *Online* es centralizada en un servidor y que la aplicación *Smart Client* está distribuida en cada uno de los dispositivos en donde se utiliza.

Con esta premisa, el mantenimiento de la aplicación *Online* es más sencillo y rápido, debido a que si existe un error en el software, o hay que agregar y/o modificar una funcionalidad del software, este hecho es solucionado en una sola localidad (servidor Web). Mientras que en el caso de la aplicación *Smart Client*, si se detecta un error o se desea modificar o actualizar la funcionalidad del software, éste deberá corregirse y cargarlo nuevamente en todos los dispositivos *Pocket PC* en donde se utilice la aplicación.

5.5 COSTO DEL SISTEMA

Para analizar el costo del sistema se tomó en cuenta las actividades necesarias para desarrollar los sistemas *Online* y *Smart Client*. Cada una de las actividades tiene recursos asignados y por lo tanto costos.

5.5.1 ACTIVIDADES

El desarrollo del proyecto consta de seis actividades principales:

1. Fase de inicio del proyecto, en esta fase se recopila la documentación existente y se establecen estándares para el desarrollo del proyecto.
2. Análisis, esta fase consiste en recopilar información sobre el problema, analizar esta información para establecer los casos de uso del sistema.
3. Diseño, en esta fase se identifican clases y las relaciones entre éstas, además permite identificar los métodos y atributos de cada unas de estas clases. También en esta fase se determina la arquitectura o estructura de capas del sistema.
4. Implementación, esta fase consiste en realizar el diseño planteado; la implementación se realizó utilizando *.NET*.

5. Pruebas, en esta fase se realizan las pruebas de las aplicaciones para asegurar su correcto funcionamiento y establecer cuál logra un mejor desempeño.
6. Puesta en marcha, en esta fase se instala el sistema para su funcionamiento.

5.5.1.1 Diagrama Gantt de actividades

La Figura 5.38 muestra en detalle las actividades del proyecto y su duración.

5.5.2 RECURSOS

Las actividades señaladas anteriormente serán ejecutadas por el personal de ingeniería que se indica en la Tabla 5.23. La información de costos mensuales del personal de ingeniería fue proporcionada por la empresa de desarrollo de software *KOALA Servicios, Tecnología y Sistemas S. A.*

Personal de Ingeniería	Costo mensual en dólares
Arquitecto <i>.NET</i>	1200
Programador <i>.NET</i>	800

Tabla 5.23 Personal de Ingeniería

5.5.2.1 Recursos por actividad

El personal de ingeniería tiene un papel determinado que cumplir. En la Tabla 5.24 se muestra cómo se asignan las tareas al personal de ingeniería.

Actividad	Personal a cargo
Fase de inicio del proyecto	Arquitecto <i>.NET</i>
Análisis	Arquitecto <i>.NET</i>
Diseño	Arquitecto <i>.NET</i>
Implementación	Programador <i>.NET</i>
Pruebas	Programador <i>.NET</i>
Puesta en marcha	Programador <i>.NET</i>

Tabla 5.24 Recursos por actividad

Figura 5.38 Diagrama Gantt de actividades

5.5.3 RESUMEN DE COSTOS

El costo del sistema se determina entonces por el costo de las actividades necesarias para desarrollar el sistema y además por gastos varios. Los gastos varios son el alquiler de oficina, agua, luz, teléfono, Internet y gastos de papelería.

Los gastos de actividades se resumen en la Tabla 5.25.

Actividad	Duración (días)	Recurso	Costo Mensual (dólares)	Costo por actividad (dólares)
Fase de inicio del Proyecto	4	Arquitecto .NET	1.200,00	240,00
Análisis	14	Arquitecto .NET	1.200,00	840,00
Diseño	14	Arquitecto .NET	1.200,00	840,00
Implementación	62	Programador .NET	800,00	2.480,00
Pruebas	10	Programador .NET	800,00.	400,00
Puesta en marcha	5	Programador .NET	800,00	200,00
TOTAL	109			5.000,00

Tabla 5.25 Costo de actividades

La duración del proyecto es de cinco meses en los que hay que pagar los gastos varios descritos en la Tabla 5.26.

Gastos varios	Duración (meses)	Costo Mensual (dólares)	Costo por actividad (dólares)
Alquiler oficina	5	180,00	900,00
Luz	5	40,00	200,00
Agua	5	10,00	50,00
Teléfono	5	20,00	100,00
Internet	5	50,40	252,00
Suministros de oficina (papel, tinta, carpetas, cartas, copias, fotos)	5	20,00	100,00
TOTAL		320,40	1.602,00

Tabla 5.26 Gastos varios

El costo total del proyecto es la suma de los gastos por actividad y los gastos varios; la Tabla 5.27 muestra el costo total del proyecto.

Gastos	Costo (dólares)
Actividades	5.000,00
Varios	1.602,00
TOTAL	6.602,00

Tabla 5.27 Costo total del proyecto

El costo del desarrollo del software depende de su complejidad, porque esto determina el tiempo de duración del proyecto así como el número de personas involucradas en el desarrollo del software.

También existen ocasiones en las cuales el software ya ha sido desarrollado y sólo es necesaria su adaptación a ciertos requerimientos, esto permite aumentar las ganancias, pues se revende una aplicación fabricada, esto generalmente sucede con paquetes de software contables.

Capítulo 6

Conclusiones y Recomendaciones

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Los dispositivos móviles son una herramienta clave en la cotidianidad y la vida laboral; debido a los avances de hardware estos dispositivos que en su comienzo servían como terminales de voz, hoy se fortalecen al ser capaces de ejecutar aplicaciones distribuidas empresariales.

Microsoft está interesado en formar parte activa del competitivo mercado de los dispositivos móviles; lanzó el sistema operativo *Windows Mobile* y una versión para móviles del paquete *Office*. Además desarrolló el *.NET Compact Framework* y *ASP.NET Mobile* que permiten desarrollar aplicaciones para dispositivos móviles y propuso un prototipo de *PDA* al que denominó *Pocket PC*.

El diseño de una aplicación comienza con una descripción de los procesos que se siguen para realizar determinada tarea, esta descripción se denomina requerimientos del negocio, y permite identificar los requerimientos del sistema y establecer los procesos a ser automatizados con el sistema de software a diseñarse.

Los requerimientos del sistema son los procesos que debe cumplir la implementación del sistema, para cubrir los requerimientos de usuario. En esta fase se llega inclusive a un diseño de interfaces de usuario.

Los servicios Web son el equivalente a un servidor de páginas Web, pues ambas tecnologías permiten obtener información desde una localidad de Internet, con la diferencia que los servicios Web interactúan con otra aplicación y las páginas Web con usuarios humanos.

La autenticación basada en formularios diseñada para ser utilizada por personas es adaptable para la autenticación de aplicaciones. Esto se logra sobrescribiendo el método *Application_AuthenticateRequest* en el archivo *Global.asax* y pasando las credenciales de autenticación en los encabezados *SOAP* de los requerimientos.

Al diseño del sistema se consideró necesario dividirlo en tres capas: capa de acceso a datos, en donde se maneja las interacciones con la base de datos; la capa de reglas del negocio, encargada de proporcionar la funcionalidad al sistema; y la capa de interfaz de usuario, la cual interactúa con el estudiante.

UML (Unified Modeling Language) consiste de un conjunto de diagramas utilizados para modelar sistemas orientados a objetos. Es flexible y no es mandatorio seguir rigurosamente los procesos que plantea *UML*, ya que existen decisiones que se deben tomar dependiendo de cada aplicación. En general *UML* permite analizar, diseñar y documentar aplicaciones.

Las aplicaciones *Online* tienen la ventaja de estar disponibles para todos los dispositivos que cuenten con un navegador Web sin necesidad de ningún proceso de instalación adicional. También al ser una aplicación centralizada cualquier mantenimiento, actualización o corrección de algún error en la aplicación, se ve inmediatamente reflejado para todos los usuarios.

La desventaja de las aplicaciones *Online*, consiste en que éstas se encargan de ejecutar la lógica de la aplicación y de manejar el interfaz de usuario; en resumen casi toda la carga de trabajo es ejecutada en el servidor Web. Si los usuarios simultáneos crecen también crece el trabajo en el servidor, por tanto aumenta su tiempo de respuesta y disminuye su eficiencia. En resumen puede atender a menos usuarios simultáneos que una aplicación *Smart Client*.

Las aplicaciones *Smart Client* o cliente inteligente ejecutan parte de la carga de trabajo, y exclusivamente se conectan al servidor para obtener datos. Al encargarse del manejo de la interfaz de usuario aliviana el trabajo del servidor; por consiguiente éste puede atender a más usuarios y con tiempos de respuesta adecuados.

La desventaja de las aplicaciones *Smart Client*, es que necesitan ser instaladas en cada uno de los dispositivos en los que se utilizará; esto es un problema, pues no todos los dispositivos cuentan con un mismo sistema operativo, y mismas características de hardware. Esto determina que se necesiten distintas versiones de la aplicación para los diferentes dispositivos; esto a su vez dificultaría el

mantenimiento de las aplicaciones pues si se encuentra un error en la aplicación ésta debería ser actualizada en cada una de sus versiones y actualizada en cada dispositivo.

6.2 RECOMENDACIONES

Es recomendable diseñar un sistema con una estructura de capas funcionales. Se debe separar en capas las funcionalidades de acceso a datos, lógica de negocio e interfaz de usuario. El organizar de esta manera el código, permite agilizar los procesos de mantenimiento de software. Además es posible, como en el presente proyecto de titulación, hacer diferentes implementaciones de la capa de interfaz de usuario; las aplicaciones *Online* y *Smart Client* son implementaciones de esta capa.

Se recomienda utilizar procedimientos almacenados para el acceso a datos, ya que éstos son pre-compilados y se ejecutan con mayor rapidez, permitiendo obtener los datos de una manera eficiente.

Es recomendable utilizar un emulador de dispositivo en la fase de desarrollo de software, pero para probar el funcionamiento de la versión final es aconsejable realizarlo en un dispositivo real.

Antes que una aplicación Web salga a un ambiente de producción, ésta debe pasar un proceso de pruebas de rendimiento, que ayude a establecer los límites de esta aplicación o encontrar “cuellos de botella” que afecten el desempeño de la aplicación Web. El que una aplicación Web falle es desastroso, pues afecta al número de usuarios conectados simultáneamente y dependiendo los servicios que preste puede causar un malestar masivo en los usuarios finales.

Para realizar las pruebas se deben considerar períodos de prueba mayores a 20 minutos y asegurarse que los cliente que generen tráfico tengan un porcentaje de procesamiento menor al 75%.

Es recomendable que aplicaciones de uso masivo con una funcionalidad bien definida, donde el desempeño es muy importante, sean desarrolladas como

aplicaciones inteligentes o *Smart Client*; un claro ejemplo de una aplicación *Smart Client* es *Microsoft Messenger*. Mientras que aplicaciones de comercio electrónico, en donde van a existir cambios constantes de catálogos, precios, etc., que pueden ser fácilmente manejados en un servidor centralizado, se deben pensar como aplicaciones *Online*.

Se recomienda que los servicios en línea, que en la actualidad se prestan para PCs, migren para ser accesibles a dispositivos móviles, pues debido al masivo uso de móviles estos servicios serán muy solicitados. Para que ocurra esto la conectividad de datos para los dispositivos móviles deberá tener costos atractivos para el usuario final.

REFERENCIAS BIBLIOGRÁFICAS

1. *BlogdeMóviles*. [En línea] 2007.

<http://www.blogdemoviles.com.ar/category/celulares-smartphones/>.

2. **Maximiliano, Firtman**. Desarrollos móviles con .NET.

3. Wikipedia. [En línea] 2007. <http://es.wikipedia.org>.

4. **Arienza, Jorge L.** El Observatorio de Internet Móvil. *El Observatorio de Internet Móvil*. [En línea] 7 de marzo de 2007.

5. Gartner. [En línea] 2006. www.gartner.com.

6. **Microsoft Corporation**. *Estrella 1 Introducción a Microsoft .NET*. 2006.

7. **Microsoft Corporation**. *MSDN Library Visual Studio*. 2005.

8. **Microsoft Corporation**. *Estrella 2 Introducción al desarrollo de aplicaciones Web con ASP.NET*. 2006.

9. **Connolly, Randy**. *Core Web application development with ASP .NET 2.0*.

10. **Lam, Hoang y Thai, Thuan L.** *.NET Framework Essentials*.

11. **Mhlabane, M.J., Petkov, D. y Nepal, T.** *An investigation into performance testing for e-commerce web-based*. 2005.