

**SISTEMA DE GESTIÓN DE APRENDIZAJE
MÓVIL PARA UN POCKET PC**

**Ana Lucia Jurado Narváez
Gustavo Adolfo Hurtado Sarasti**

**Director
MSc. Carlos Alberto Cobos Lozada**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Línea de Tecnologías de la Información
Popayán
2004**

SISTEMA DE GESTIÓN DE APRENDIZAJE MÓVIL PARA UN POCKET PC



**Ana Lucia Jurado Narváez
Gustavo Adolfo Hurtado Sarasti**

**Director
MSc. Carlos Alberto Cobos Lozada**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Línea de Tecnologías de la Información
Popayán
2004**

Nota de Aceptación

Firma Presidente del Jurado

Firma de Jurado

Firma de Jurado

Popayán 30 de Noviembre del 2004

Dedico este trabajo a mis padres Fernando y Josefa, por su amor incondicional y por enseñarme que cada meta que la vida te pone, por pequeña que sea, se debe enfrentar como el mayor de los retos.

A mi sobrina Valentina por haber llegado a iluminar mis días.

A mi tía Guiomar por sus sabios consejos, que me enseñan a llevar siempre la frente en alto.

A todas aquellas personas han estado conmigo en cada instante de mi vida enseñándome como vivir cada día como el último de mis días.

Gustavo Hurtado

Dedico este trabajo a aquellas importantes personas de mi vida, para quienes mi ausencia ha sido mi legado en el afán de alcanzar mis metas:

Mis padres, Jaime Jurado y Zoila Narváez, quienes me regalaron el gran tesoro de la educación, acompañado de su ilimitable amor y comprensión, fuente de luz y energía para continuar mi camino.

Mis hermanos, Yenny y Jaime, quienes depositan en mi su confianza y cariño y para quienes lo único que tengo para ofrecerles es mi ejemplo.

Y a todos los que estuvieron presentes en mi vida, para regalarme un poco de su ser.

Ana Jurado

A nuestros compañeros quienes han estado presentes para brindarnos su apoyo y demostrarnos que en la universidad también se pueden hacer buenos amigos.

Ana y Gustavo.

AGRADECIMIENTOS

Por el apoyo que recibimos en el desarrollo del presente trabajo de investigación, queremos expresar nuestros agradecimientos a las siguientes personas e instituciones:

COLCIENCIAS, Instituto colombiano para el desarrollo de la ciencia y la tecnología, por la cofinanciación en su proyecto 1103-11-14461 "Propuesta Curricular para el Desarrollo de la Pedagogía de la Investigación en Ciencias con Enfoque en Estudios CTS + I para la Educación Media (Colegio INEM).

CARLOS ALBETO COBOS LOZADA, MSc Computación. Quien además de ser nuestro guía y maestro durante cada día de trabajo, es nuestro amigo, compañero y ejemplo a seguir.

ANDRES ORTIZ GONZALES, Ingeniero de Sistemas. Quien fue nuestro compañero y soporte en el momento necesario.

TABLA DE CONTENIDO

1	INTRODUCCIÓN.....	1
1.1	PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	1
1.2	OBJETIVOS.....	5
1.2.1	OBJETIVO GENERAL	5
1.2.2	OBJETIVOS ESPECÍFICOS.....	5
1.3	APORTES DEL PROYECTO.....	6
2	MARCO TEÓRICO.....	9
2.1	ELEARNING Y EDUCACIÓN EN LÍNEA.....	9
2.1.1	DEFINICIÓN, CONTEXTO, VENTAJAS Y DESVENTAJAS	9
2.1.2	MODELO GENERAL Y CARACTERÍSTICAS	16
2.1.3	ESTÁNDARES Y ESPECIFICACIONES	19
2.1.4	ARQUITECTURA SCORM	20
2.1.5	MODELO DE AGREGACIÓN DE CONTENIDOS (CAM).....	22
2.1.6	METADATOS DE LOS SCO (LOM).....	23
2.1.7	EMPAQUETAMIENTO DE CONTENIDOS (CP)	25
2.1.8	TENDENCIAS.....	25
2.2	MLEARNING.....	26
2.2.1	DEFINICIÓN	27
2.2.2	ACCESO INALÁMBRICO A INTERNET	28
2.2.3	CAPACIDADES Y CARACTERÍSTICAS	28
2.2.4	IMPLICACIONES TEÓRICAS.....	30
2.2.5	IMPLICACIONES PEDAGÓGICAS	31
2.2.6	IMPLICACIONES DE DISEÑO	31
2.2.7	MLEARNING EN UN AMBIENTE DE APRENDIZAJE MÁS AMPLIO	33
2.2.8	MLEARNING EN UN AMBIENTE DE APRENDIZAJE INDEPENDIENTE	33
2.2.9	PROYECTOS DE I+D RELACIONADOS	34
2.3	MICROSOFT .NET	35
2.3.1	EL MODELO DE COMPUTACIÓN DISTRIBUIDA	35
2.3.2	LA PLATAFORMA .NET	36
2.3.3	EL MICROSOFT .NET FRAMEWORK	37

2.3.4	EL COMMON LENGUAJE RUNTIME (CLR)	39
2.3.5	LENGUAJES DE PROGRAMACIÓN.....	39
2.3.6	MÓDULOS ADMINISTRADOS	40
2.3.7	ENSAMBLADOS.....	40
2.3.8	LA BIBLIOTECA DE CLASES DEL .NET FRAMEWORK	41
2.3.9	COMPILACIÓN Y EJECUCIÓN DE UNA APLICACIÓN .NET	41
2.4	SMART CLIENTS	42
2.4.1	DESCRIPCIÓN	42
2.4.2	CARACTERÍSTICAS IDEALES DE UN SMART CLIENT.....	44
2.5	DESARROLLO DE APLICACIONES MOVILES CON MICROSOFT .NET	44
2.5.1	EL MICROSOFT .NET COMPACT FRAMEWORK (.NET CF).....	45
2.5.2	TRABAJANDO CON XML.....	52
2.5.3	MANEJO DE ENTRADA y SALIDA	54
2.5.4	OBJETOS DE DATOS - ADO.NET.....	55
2.5.5	SERVICIOS WEB XML.....	61
2.5.6	MULTITHREADING	70
2.6	OPEN NET COMPACT FRAMEWORK – Open Net CF.....	72
2.7	ORACLE DATA PROVIDER FOR .NET (ODP.NET).....	72
3	DESCRIPCIÓN DEL SISTEMA.....	75
3.1	ARQUITECTURA DEL SISTEMA.....	75
3.1.1	DESPLIEGUE DE LA APLICACIÓN	77
3.2	DIAGRAMA DE CASOS DE USO	85
3.2.1	DESCRIPCIÓN GENERAL DEL DIAGRAMA DE LOS CASOS DE USO	85
3.3	DESCRIPCION DE LOS PRINCIPALES CASOS DE USO	87
3.4	DIAGRAMAS DE SECUENCIA	93
3.5	DIAGRAMA DE CLASES.....	96
3.6	MODELO DE LA BASE DE DATOS	99
3.6.1	PROGRAMA ACADÉMICO DEL USUARIO	100
3.6.2	META MODELOS PARA LA GENERACIÓN DE SCO	100
3.6.3	PROCESO DE EVALUACIÓN	102
3.6.4	FORO.....	103
3.6.5	CHAT	104
3.7	PROBLEMAS ENCONTRADOS Y SOLUCIONES SELECCIONADAS	105

3.8	RECOMENDACIONES TÉCNICAS.....	109
3.8.1	MANEJO DE DATOS.....	109
3.8.2	CONEXIÓN Y COMUNICACIÓN.....	109
3.8.3	INTERFAZ.....	110
3.8.4	ARQUITECTURA.....	110
3.8.5	ACTUALIZACIONES Y DESPLIEGUE	110
3.8.6	SEGURIDAD.....	110
4	CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO.....	111
4.1	CONCLUSIONES	111
4.2	RECOMENDACIONES.....	112
4.3	TRABAJO FUTURO	114
5	REFERENCIAS BIBLIOGRAFICAS.....	115

LISTA DE FIGURAS

Figura 1 Modelo general de educación en línea.....	17
Figura 2 Arquitectura de un LMS utilizando SCORM	21
Figura 3 Modelo de Agregación de Contenidos	22
Figura 4 Meta datos de los objetos de aprendizaje	24
Figura 5 Arquitectura general del acceso a la red alámbrica e inalámbrica	29
Figura 6 Arquitectura general de la plataforma .Net.....	37
Figura 7 Comparación de Aproximaciones.....	43
Figura 8 Ambiente de Ejecución de una aplicación con el .Net CF	45
Figura 9 Estructura del Ambiente de Ejecución.....	47
Figura 10 Arquitectura de ADO.NET [42]	57
Figura 11 Arquitectura de una aplicación .NET utilizando Servicios Web.....	62
Figura 12 Acceso a datos Oracle.NET [47]	73
Figura 13 Arquitectura para el cliente Pocket PC y Windows PC	77
Figura 14 Arquitectura para el cliente Web	77
Figura 15 Despliegue de Easy Learning.....	78
Figura 16 Ingreso al sistema Pocket PC	79
Figura 17 Selección de asignaturas	79
Figura 18 Visualización de SCOs	80
Figura 19 Uso del Servicio Web de Google.....	80
Figura 20 Resultados de la búsqueda	81
Figura 21 Administración del repositorio local	81
Figura 22 Visualizar foros	82
Figura 23 Agregar un aporte al foro.....	82
Figura 24 Administrar foros	83
Figura 25 Chat	83
Figura 26 Servicios de evaluación.....	84
Figura 27 Ingreso al sistema Web	84
Figura 28 Gestión de Usuarios	85
Figura 29 Diagrama General de los Casos de Uso	86
Figura 30 Diagrama de Secuencia Seleccionar Asignatura y Visualizar Agregaciones.....	95

Figura 31 Diagrama de Secuencia Visualizar SCO	96
Figura 32 Diagrama de Clases de Interfaz y Lógica del Cliente PPC	97
Figura 33 Diagrama de Clases de Lógica del Negocio y Lógica de Servicios	98
Figura 34 Modelo físico del programa académico de un usuario	101
Figura 35 Modelo físico de los meta modelos para la generación de SCO.....	102
Figura 36 Modelo físico para el modelo de evaluación	103
Figura 37 Modelo físico del foro	104
Figura 38 Modelo físico del Chat	105

LISTA DE ANEXOS

ANEXO A. Artículo presentado en las Jornadas Chilenas de Computación 2004.

ANEXO B. Descripción total de los casos de uso de diseño.

RESUMEN

TITULO: Sistema de Gestión de Aprendizaje Móvil para un Pocket PC*

AUTORES: HURTADO S, Gustavo Adolfo y JURADO N. Ana Lucía**

PALABRAS CLAVE: Aprendizaje Móvil, Sistemas de Gestión de Aprendizaje, SCORM

DESCRIPCION:

La presente monografía inicia con una breve introducción acerca del contexto de la solución planteada justificando su concepción. Seguidamente se plantean los objetivos que se persiguieron para su desarrollo y los aportes que fueron posibles obtener en el transcurso del proyecto. Además se plantea una breve descripción acerca de eLearning o aprendizaje electrónico, estrategia que conduce a los estudiantes a hacer de sus actividades educativas procesos de aprendizaje autónomos que más adelante con la inclusión de las tecnologías móviles, dan origen al aprendizaje móvil o mLearning, en el cuál hay más interacción entre el docente y el estudiante, y sobre todo existe más libertad en como los estudiante llevan a cabos sus procesos de aprendizaje. Luego, se presenta la herramienta utilizada, el tipo de solución obtenida y la descripción de los principales aspectos de diseño que nos condujo a ella. Finalmente las conclusiones a las que se llegaron y las recomendaciones que de ellas se dedujeron.

* Trabajo de Investigación para optar a grado de ingeniero de sistemas

** Facultad de Ingeniería Electrónica y Telecomunicaciones, Programa de Ingeniería de Sistemas, COBOS L, Carlos Alberto (Director).

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

La revolución digital trajo consigo una gran variedad de cambios, fundamentalmente en la forma como la gente piensa, trabaja, se comunica y gana su sustento diario, esto gracias al avance de las tecnologías de la información y las telecomunicaciones. Una de las principales consecuencias de esta revolución digital lo marco la llegada de Internet; puesto que se logró un acercamiento global a la información y a los conocimientos, los cuales se han convertido en los pilares para alcanzar el desarrollo de las sociedades modernas. Es ahora posible obtener y compartir, una gran cantidad de contenidos de diferentes géneros: científicos, políticos, económicos, sociales, entre otros; en tan solo una mínima cantidad de tiempo.

Tener un acceso más fácil a la información, ha generando en las naciones una mirada a una economía dependiente del conocimiento, lo que implica una mayor demanda de personas capacitadas en un menor tiempo y con la capacidad de poder retroalimentarse constantemente, sin tener que dejar a un lado sus labores cotidianas, lo que a su vez se dificulta debido a la falta de tiempo, o a la perdida del mismo por desplazamientos y esperas en estaciones, bancos, entre otros. Es de ahí donde se hace necesario la inclusión de ambientes de aprendizaje que suplan estas limitaciones, ya que para muchos el asistir a un salón de clase diariamente en un horario determinado es casi imposible. Estos aspectos fomentaron el desarrollo del aprendizaje electrónico (eLearning, electronic learning) y en especial del aprendizaje en línea (online learning).

Cada día alrededor del mundo existen más instituciones educativas que ofrecen servicios de aprendizaje electrónico a sus estudiantes, estos servicios van desde simples tutoriales hasta cursos de pregrado y postgrado en prestigiosas universidades; tales como la Universidad Abierta de Hong Kong [1], la Universidad Abierta del Reino Unido [1], entre otras. Estos cursos han tenido tanta acogida y tanto éxito que sus usuarios afirman que no es necesario estar sentado muchas horas en un salón de clase con un profesor, para obtener altas calificaciones y tener las mismas oportunidades que cualquier persona egresada de programas presenciales; tan buenos han sido los resultados de estos

estudiantes que se comprueba que el aprendizaje electrónico es tan efectivo como la educación tradicional [1]. Una herramienta indispensable dentro de este nuevo ambiente de aprendizaje son los llamados Sistemas de Gestión de Aprendizaje (LMS – Learning Management System), debido a que éstos son los encargados de controlar el progreso de los estudiantes en los cursos que toman, distribuir los contenidos educativos, la actualización de los mismos y además realizar los procesos de gestión que soportan las actividades de aprendizaje en o fuera de línea [2].

A pesar de las innumerables ventajas que el aprendizaje electrónico trae consigo aún hay varias oportunidades por aprovechar, generadas a partir de las necesidades de aquellas personas para las que su único tiempo libre transcurre entre largas trayectorias hasta llegar a sus hogares, interminables colas bancarias, tediosas esperas en el aeropuerto, desplazamientos a lugares apartados y cientos de situaciones en las que es difícil contar con un PC y una conexión a Internet, lo que hace necesario, contar con un medio a través del cual se puedan solventar estas limitaciones y que permita incrementar significativamente la posibilidad de que las personas aprovechen estos espacios de tiempo en su formación; de esta manera se ha motivado el origen del llamado aprendizaje móvil (mLearning), que se apoya en los asistentes personales digitales (dispositivo de mano y bolsillo), los teléfonos inteligentes, los computadores portátiles y las conexiones inalámbricas para solucionar las necesidades de tiempo y espacio que limitan el aprendizaje electrónico; pues a través de estos dispositivos es posible desarrollar procesos de aprendizaje bajo las circunstancias antes mencionadas, además el uso de los dispositivos móviles puede ser más oportuno, debido a que realmente son personales y cada usuario lo puede configurar y utilizar de la manera que más cómoda y práctica le parezca, facilitando así el control sobre el mismo, contribuyendo entonces a aumentar el tiempo disponible para el aprendizaje y dando la posibilidad de llegar a todos aquellos interesados en su educación.

Al igual que el aprendizaje electrónico, el aprendizaje móvil está generando grandes expectativas en buena parte de la comunidad académica, a tal punto de augurársele ser el futuro del aprendizaje. Y es que el aprendizaje móvil reúne ciertas características que le permiten ser independiente de la ubicación en espacio y tiempo, ya que promete llevar a cabo procesos de aprendizaje en el lugar y hora deseados y adicionalmente sobre

cualquier dispositivo. De esta forma, habrá mayor disponibilidad de los contenidos, será más fácil la distribución de estos contenidos hasta el dispositivo y de la mano a ello el rendimiento académico podrá ser más satisfactorio, si se aprovecha al máximo las ventajas que este sistema de aprendizaje trae consigo. Pero mientras eso sucede hay algunos aspectos desfavorables a tener en cuenta: la capacidad de almacenamiento de los dispositivos, la imposibilidad de estar conectado en un 100% a la red, la generación de contenidos educativos por parte de los profesores, y la administración del aprendizaje. Por lo anterior se hace necesario contar con un Sistema de Gestión de Aprendizaje Móvil (MLMS) a través del cual sea posible soportar estos aspectos, lo que incluye: la distribución de los cursos hasta el dispositivo en pequeños volúmenes, disminuyendo así la cantidad de espacio requerido para albergar todo un curso; la gestión de la conexión a la red, evitando la necesidad de estar conectado permanentemente; el desarrollo y estructuración de los contenidos temáticos, el seguimiento y gestión de los procesos de aprendizaje que el estudiante realiza con el fin de alcanzar los objetivos del curso. Todo esto en conjunto permite obtener un modelo de aprendizaje móvil más sólido y útil, sobre el cual se habrá disminuido las limitaciones con las que inicialmente se planteo. (Ver anexos)

Para lograr los objetivos trazados por el MLMS es necesario hacer uso de un estándar que permita modelar los contenidos educativos, a partir del cual se unifiquen y distribuyan sin problemas de compatibilidad. Uno de los mayores aportes en el desarrollo de contenidos lo ha hecho ADL, Advanced Distributed Learning, cuya visión es proveer acceso a la educación de calidad y entrenamiento, adaptado a las necesidades individuales, entregadas de forma rentable en cualquier lugar y en cualquier momento [4], esto a partir del Modelo de Referencia para Objetos de Contenidos Intercambiables (SCORM) con el fin de crear objetos de aprendizaje¹ reutilizables y realizar posteriormente una fácil actualización de los mismos si así se desea. SCORM cuenta con 3 criterios fundamentales: soportar completamente la unión de lineamientos que puedan ser entendidos e implementados por la elaboración de objetos de contenidos distribuibles; debe ser adoptado, entendido y usado por una amplia variedad de desarrolladores como

¹ Objeto de Aprendizaje: Entidad digital o no, la cual puede ser usada, rehusada o referenciada en una tecnología que soporta el aprendizaje (SCORM – LTSC – IEEE).

sea posible; debe permitir el mapeo del modelo específico de un desarrollador para diseñar sistemas instructivos y desarrollado dentro de sí mismo [4]. Lo que asegura que SCORM sea acogido para brindar alta calidad en la educación y el correcto soporte para el desarrollo de objetos de aprendizaje. Adicionalmente, estos objetos de aprendizaje deberán garantizar: durabilidad, que no requieran modificaciones en sus versiones debido a cambio de sistema; interoperabilidad, que operen a través de una amplia variedad de hardware, sistemas operativos y navegadores Web; accesibilidad, que puedan ser indexados y encontrados cuando sean necesarios, y reusabilidad, que puedan ser modificados y usados por una gran variedad de desarrolladores de herramientas. Con lo cual es posible garantizar que los contenidos cumplan con las características de alto nivel propuestas por ADL, además de promover el aprendizaje basado en la tecnología y reducir los costos de inversión entre un 50% y 80% aproximadamente, entre otros [6].

Este escenario generado a partir del aprendizaje móvil ha incentivado a instituciones educativas, empresas del sector de las tecnologías de la información y desarrolladores de software a realizar sus propios aportes en este campo, como es el caso de AdvanceWork con su Curso Móvil de Inglés (Professional English) mediante Pocket PC de Microsoft para los profesionales de negocios que necesitan adquirir un inglés fluido, combina práctica de conversación en vivo con instructores hablantes nativos de inglés y lecciones multimedia [7]; la Escuela de Secundaria Waldo Emerson Lessenger en Detroit en conjunto con la Universidad de Michigan que prueban la integración de computadores de mano en las clases de ciencia del sexto grado, los estudiantes realizan lecciones en software personalizado, actividades externas donde tienen la oportunidad de registrar observaciones y subirlas a una base de datos que usa el salón de clases [8]; INSEAD, NOKIA e ICUS han implantado un curso utilizando en forma combinada la Web y la tecnología móvil, INSEAD suministró el contenido del curso, ICUS aplicó los principios de diseño instruccional al curso y NOKIA suplió la experiencia en uso de tecnología WAP [9]; SmartForce con su plataforma MySmartForce que le permite al cliente ejecutar operaciones fuera de línea, incluyendo rastreo en el LMS [10]; entre otras valiosas contribuciones en el desarrollo del aprendizaje móvil. A partir de las cuales es posible desarrollar aplicaciones que implementen nuevas y mejores funcionalidades de las ya presentadas, teniendo en cuenta: necesidades comunes, no limitándose a un área de estudio específica, brindando así la posibilidad de llegar a más usuarios; contenidos,

basados en un estándar que permitan obtener una mayor ventaja de éstos; captura de datos, a partir de los cuales sea posible obtener información valiosa para el desarrollo del aprendizaje, y por supuesto gestionar el acceso en o fuera línea a los cursos.

En Colombia actualmente no existen cursos que puedan ser llevados a cabo sobre dispositivos móviles y mucho menos sistemas que los administre, debido a que nuestro actual sistema educativo esta basado principalmente en el desarrollo de procesos de aprendizaje a partir de clases presénciales y unos cuantos esfuerzos de instituciones por ofrecer programas académicos semipresenciales y de aprendizaje electrónico, sin aprovechar el uso que es posible dar a las tecnologías que en la actualidad están entrando al país y excluyendo a buena parte de la población, debido a las limitaciones que estos sistemas educativos presentan. Se genera entonces la necesidad de explorar en el desarrollo de un MLMS que sea capaz de apoyar los procesos de aprendizaje desde la perspectiva del aprendizaje móvil, que haga posible tomar ventaja de las oportunidades y beneficios que brinda este sistema, promoviendo así el desarrollo del aprendizaje móvil en Colombia, con un producto innovador en nuestra región.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Se desarrolló un Sistema de Gestión de Aprendizaje Móvil² (Mobile Learning Management System, M-LMS) que permite estructurar y cursar³ contenidos educativos basados en el Modelo de Referencia para Objetos de Contenidos Intercambiables (Sharable Content Object Reference Model, SCORM), a través de un Asistente Personal Digital (Personal Digital Assistant, PDA) Pocket PC.

1.2.2 OBJETIVOS ESPECÍFICOS

Los objetivos específicos logrados con este proyecto fueron:

² Mobile Learning Management System (M-LMS): Sistema para administrar los procesos de aprendizaje a través de un dispositivo móvil.

³ Teniendo en cuenta los modelos de divulgación y evaluación de un LMS definidos en el Proyecto Unicauca Virtual [11] que servirán como apoyo a la definición de requisitos del M-LMS a obtener en la fase de análisis.

- Definir el modelo del Sistema de Gestión de Aprendizaje Móvil utilizando el paradigma orientado a objetos, una instanciación del Proceso Unificado y los diagramas⁴ definidos en el Lenguaje Unificado de Modelado (Unified Modeling Language, UML).
- Desarrollar el Sistema de Gestión de Aprendizaje Móvil utilizando Microsoft Visual Studio .NET 2003 como herramienta de desarrollo, en una arquitectura de 3 niveles comunicadas por SOAP y XML, de la siguiente forma:
 - ◆ Una interfaz para estructurar los contenidos del curso por parte del profesor en un PC y otra en la que el estudiante cursa la asignatura a través del PDA.
 - ◆ Un conjunto de servicios Web XML que soportan la lógica de los procesos de aprendizaje en/o fuera de línea que llevan a cabo cada uno de los usuarios del sistema.
 - ◆ Una base de datos relacional en Oracle 9i[®] que permite dar persistencia a la información manejada en el sistema, acompañada de un directorio virtual de Internet Information Server (IIS) que almacena los contenidos de los cursos.
- Demostrar el uso del sistema, definiendo un bloque temático de un curso, adicionando los objetos de contenido compartible para el mismo y distribuyéndolos hasta el PDA.

1.3 APORTES DEL PROYECTO

Al finalizar la investigación [12] y el desarrollo del proyecto se destacan los siguientes aportes y productos que en su mayoría son adicionales a los establecidos en los objetivos, los cuales son:

- Participación en el Concurso Mundial Imagine Cup 2004, el cual es uno de los premios a la excelencia que entrega Microsoft, dirigido a aquellos diseñadores y programadores que apliquen sus conocimientos en el desarrollo de aplicaciones comercial y socialmente viables. El proyecto fue evaluado a nivel nacional en la ciudad de Bogotá el día 8 de Mayo de 2004, por un conjunto de 4 jueces colombianos designados por Microsoft Colombia, de esta primera evaluación el proyecto fue

⁴ Los diagramas a utilizar se definen en cada actividad y fase del ítem 4. [Actividades y Cronograma] del documento.

seleccionado como el Mejor Proyecto en la Categoría de Software Design para Colombia. Posteriormente el proyecto se evaluó a nivel Andino por parte de 4 jueces y el día 22 de Mayo de 2004 se le notificó al grupo del proyecto que en esta evaluación fue seleccionado como el Mejor proyecto en la Categoría Software Design para la Región Andina. Finalmente, el proyecto se presentó en la final mundial del Imagine Cup 2004 en Sao Paulo, Brasil, del 1 al 8 de Julio de 2004 en donde se realizó la defensa del proyecto ante 10 jurados internacionales. Adicionalmente se ubicó un stand en el centro de convenciones del World Trade Center de Sao Paulo (Brasil), sede del Microsoft Professional Developers Conference (PDC) 2004.

- Integración del Sistema de Gestión de Aprendizaje Móvil para Pocket PC en el proyecto de investigación titulado “Propuesta Curricular para el Desarrollo de la Pedagogía de la Investigación en Ciencias con Enfoque en Estudios CTS + I para la Educación Media (Colegio INEM)” y cofinanciado por COLCIENCIAS. Con esta aplicación software los estudiantes de décimo grado del Colegio INEM podrán realizar procesos de aprendizaje autónomo y colaborativo, entre ellas: revisar contenidos (SCOs), presentar auto evaluaciones, dialogar a través de chats, revisar y discutir en foros, entre otros. Actualmente se cuenta con un primer prototipo operacional al cual se le están desarrollando pruebas y que se convierte en el corazón de futuros desarrollos.
- Presentación del proyecto en el III Seminario Internacional Tecnologías Internet, SITI 2004, organizado por la Comunidad de Estudiantes de Ingeniería de Sistemas, CEIS y el Programa de Ingeniería de Sistemas de la Facultad de Ingeniería Electrónica de la Universidad del Cauca, realizado entre el 7 y el 9 Octubre de 2004. <http://siti2004.unicauca.edu.co>
- Divulgación del proyecto en el VII Congreso Iberoamericano de Informática Educativa, organizado por la Rede Iberoamericana de Informática Educativa, realizado entre el 13 y el 15 de Octubre de 2004 Monterrey, México. <http://www.ribie.org/ribie2004>

- Divulgación del proyecto en las Jornadas Chilenas de Computación 2004, patrocinadas por la Sociedad Chilena de Ciencia de la Computación, realizadas del 8 al 12 de Noviembre, Arica, Chile. <http://lsm.dei.uc.pt/ribie/ribie2004/> (Ver ANEXO A)
- Divulgación del proyecto en el III Coloquio Internacional sobre Currículo 2004, organizado por el Doctorado en Ciencias de la Educación de RUDECOLOMBIA, realizado entre el 3 y el 5 de Noviembre de 2004, Popayán, Colombia. <http://www.coloquiocurriculo.unicauca.edu.co/>
- La presente monografía de grado, en la cual se presenta una síntesis general del desarrollo del proyecto, describe la forma en que se cumplió con los objetivos, los aportes más significativos del proyecto, las conclusiones, recomendaciones y el trabajo futuro.
- Una herramienta software denominada Easy Learning, la cual es un Sistema de Gestión de Aprendizaje Móvil para un Pocket PC, que cuenta con dos interfaces de usuario final, la primera para el estudiante a través de una aplicación Windows Forms en una Pocket PC, y la segunda para el docente (instructor, profesor, guía o tutor) y el administrador también a través de una aplicación de Windows Forms en un PC estándar. La lógica de la aplicación fue construida con servicios Web XML, el contenido relacional se almacenó en una base de datos Oracle9i™ y el almacenamiento de los objetos de contenido compartible (Sharable Content Object, SCO) en un directorio virtual de Internet Information Server (IIS). Para el almacenamiento de los SCOs se utilizó el meta datos de SCORM 1.2 buscando mayor interoperabilidad y reusabilidad de los contenidos

2 MARCO TEÓRICO

2.1 ELEARNING Y EDUCACIÓN EN LÍNEA

Es preciso tener en cuenta que hoy en día la pregunta ya no consiste en determinar si debemos o no usar la educación en línea, lo que realmente interesa es determinar cual o cuales son las mejores formas de usarla. Usarla o no, ya no es importante porque en muchas partes del mundo se esta usando la educación en línea y más aún la calidad académica de sus egresados es tan buena o en algunos casos mejor que la de estudiantes que no usan este tipo de educación. Es por esto que debemos llegar a usar las tecnologías de la información para soportar y mejorar los procesos de enseñanza y aprendizaje en la educación presencial como en la educación a distancia, teniendo en cuenta los factores económicos, culturales y sociales propios de nuestro país.

A continuación se presenta una breve definición de la educación en línea, el contexto en el que se utiliza, las ventajas y las desventajas del uso de esta. Se continuará con la descripción del modelo general de educación en línea que se aplica en varias universidades reconocidas a nivel mundial, teniendo en cuenta no solo el componente software, sino también los procesos, la gente, entre otros. Después se inicia la parte técnica de soporte a la educación en línea, empezando con los estándares y las especificaciones que la Industria y otros entes actualmente están definiendo para mejorar los contenidos y los sistemas administradores de aprendizaje que la soportan, con el objetivo de potenciar el desarrollo de esta área en todo el mundo, se hace una descripción general del Modelo de Referencia de Objetos de Contenido Compartible (SCORM) y se finaliza con la descripción de las tendencias actuales y más importantes en estandarización de la educación en línea.

2.1.1 DEFINICIÓN, CONTEXTO, VENTAJAS Y DESVENTAJAS

La educación en línea se puede describir como una forma flexible y poderosa de mejorar el proceso de enseñanza y aprendizaje usando para ello una red de computadores (o Internet), el uso de contenidos ricos en multimedia, como imágenes, videos, contenidos basados en micro mundos exploratorios, simuladores y juegos, y otras herramientas que

permitan interacción y colaboración entre los estudiantes y entre los estudiantes y el profesor.

Los contenidos que se distribuyen a través de los sistemas de educación en línea pueden ser de variados tipos, entre ellos documentos de texto en cualquier formato, presentaciones interactivas, juegos, simulaciones y micro mundos exploratorios que deben buscar en todos los casos que el estudiante aprenda o refuerce los conocimientos relacionados con un tema del curso, desarrolle una habilidad o apropie un valor específico.

Con respecto a la interactividad y a la colaboración es preciso comentar que en los salones de clase los estudiantes no participan o sólo lo hacen porque el profesor constantemente los está motivando, estimulando o persuadiendo para que lo hagan. Con la educación en línea, a través de una chat, un foro u otras herramientas especializadas de colaboración y comunicación el estudiante puede empezar a desarrollar sus habilidades argumentativas, la capacidad de formular preguntas apropiadas a sus compañeros, perdiendo de esta forma el temor a participar.

Por definición, el eLearning (aprendizaje electrónico) es el suministro de programas educativos y sistemas de aprendizaje a través de medios electrónicos. El eLearning se basa en el uso de una computadora u otro dispositivo electrónico (grabadoras, radios, televisores, asistentes personales digitales, teléfonos móviles, entre otros) para proveer a las personas un material educativo/formativo. La educación a distancia creó las bases para el desarrollo del e-Learning, el cual viene a resolver algunas dificultades en cuanto a tiempos, sincronización de agendas, asistencia y viajes, problemas típicos de la educación tradicional. Cuando el ambiente de educación-aprendizaje involucra principalmente el uso de computadores conectados en línea con servicios de educación se está hablando de educación en línea.

En los inicios de la educación a distancia los materiales fueron escritos o impresos y enviados por correspondencia, en esto, eran muy similares al ambiente de educación presencial, en donde se utilizaban libros, mapas, entre otros. Los primeros desarrollos para montar ambientes de aprendizaje electrónico consistieron en tomar los libros y

pasarlos a formato digital o analógico; con el tiempo esos materiales han ido evolucionando y han dejado de ser sólo texto para pasar a contener videos, animaciones y otras formas de presentación de contenidos que son más adecuados a las circunstancias de lo que se quiere enseñar.

En un tiempo, los video/audio cassettes también fueron una herramienta muy útil para desarrollar procesos de aprendizaje, en el aprendizaje por radio o programas de aprendizaje basados en televisión. Todos estos sistemas tenían un gran problema, el retorno o feedback, ya que el estudiante recibía mucha información, pero si se generaban dudas, no había la posibilidad de despejarlas fácilmente. Para ello algunas instituciones organizaban reuniones en alguna parte de la ciudad o colegio y hacían un refuerzo presencial con el profesor; otras implementaron servicios de atención telefónica para dar soporte al proceso de enseñanza que estaba basado en estos medios.

Con la aparición del computador y el CD-ROM surgió el entrenamiento basado en computador (Computer Based Training, CBT), en donde específicamente, el entrenamiento estaba ligado a lo que hoy en día hacen grandes empresas y corporaciones, se entrega un CD o material para aprender una tarea sobre un trabajo específico y luego se presenta un examen de validación. Adicionalmente, existe algo más amplio que el entrenamiento y es la instrucción basada en el computador (Computer Aided Instruction, CAI), que es usar los CD-ROMs para llevar a cabo procesos de enseñanza pero basado en un instructor, es decir, hay un profesor que da elementos complementarios que pueden llevarse a cabo en clases presenciales. Este mecanismo se sale del marco del entrenamiento en el sentido que lo utilizan empresas de entrenamiento, pero también se utilizó mucho en carreras de pregrado, técnicas y demás.

Otras tecnologías conocidas son la audio conferencia, la tele conferencia, la video conferencia que son desarrollos que aumentaron el grado de interactividad, pues las personas empiezan a participar un poco más en su propio proceso de aprendizaje. Dentro de esto, está el aprendizaje basado en el Web (Web Based Learning, WBL) o la **educación en línea**, en donde el estudiante se sienta frente al computador y a través de la red baja contenidos y además tiene la posibilidad de desarrollar actividades de aprendizaje; estas actividades pueden ser colaborativas o actividades de ejercitación,

entre otras. Normalmente cuando se habla de instrucción se está hablando de una sola parte de lo que es formar o educar a una persona, pero cuando se habla de aprendizaje o educación en el Web está guiada a todos los aspectos necesarios para formar no sólo un profesional, si no también una persona.

Partiendo de este contexto y de las cosas que se pueden hacer con la educación en línea, tenemos muchas ventajas, la primera por ejemplo, vemos que los servicios de educación en línea o las plataformas tecnológicas que se desarrollan están centradas en el estudiante, éste en un cambio significativo, debido a que en la educación tradicional, en los salones de clase, se utiliza mayoritariamente el modelo conductista, en donde el profesor tiene el control sobre todas las actividades que se desarrollan en clase y el estudiante hace lo que el profesor dice, es decir el docente es quien marca el ritmo de la clase. Pero en la educación en línea, cambia este paradigma, ya que se trabaja un modelo constructivista y mucho más personalizado, pues el eje central es el estudiante y también es él quien toma su propio ritmo de trabajo, realiza los procesos de aprendizaje de acuerdo a sus capacidades y tiene la libertad de escoger el horario en el que desea llevar a cabo sus actividades educativas, lo que le da cierta autonomía; también le permite cierta independencia geográfica, dado que “puede estudiar en cualquier lugar”, bueno o en cualquier lugar donde haya un computador con conexión a Internet, además es posible que el estudiante configure la aplicación para que el ambiente de trabajo sea más amigable.

La educación en línea propicia un ambiente más agradable y menos intimidatorio para el estudiante no solo por la autonomía que le brinda sino también porque dado el caso de que surja alguna inquietud o comentario durante el desarrollo del proceso de aprendizaje, es posible resolverlo sin temor a los comentarios de los demás compañeros, ya que el estudiante se encuentra solo durante el desarrollo de las actividades académicas y la comunicación se puede realizar directa y únicamente con el docente por medio de conversaciones (chat), correos electrónicos, entre otros, lo que permite que haya más interacción y comunicación estudiante–profesor. Estos tipos de comunicación también es posible llevarlos a cabo entre estudiante y estudiante, lo que permite que haya más colaboración y cooperación entre compañeros.

El utilizar este mecanismo de educación logra uniformidad y una mejor estructuración de los cursos que se desarrollan. Esto porque ya no se van a encontrar tres cursos diferentes, si la asignatura es dada por tres docentes diferentes, por el contrario, se busca crear un solo curso con sus respectivos contenidos y recursos y que éste sea reutilizado por todos los docentes y estudiantes relacionados al mismo, con el fin de que haya uniformidad en la información que los estudiantes reciben. Además, para los docentes el proceso de actualización del material educativo es más sencillo y duradero, ya que en el momento en que se necesiten hacer las actualizaciones y/o correcciones del material académico, es posible hacerlo y estos cambios se verán reflejados automáticamente en el sistema para todos los usuarios que tengan acceso al mismo.

Otra ventaja importante que se encuentra con las plataformas utilizadas para llevar a cabo procesos de aprendizaje en línea, es que su interfaz es muy fácil de usar y están diseñadas para que los usuarios (estudiantes, profesores y otros) reciban una corta capacitación para empezar a usarlas, además del soporte (help desk) que normalmente se puede recibir a cualquier momento (24 x 7 x 365).

Las plataformas de educación en línea permiten realizar un seguimiento del progreso de los estudiantes de forma centralizada y automática, debido a que todos los resultados de las actividades educativas y evaluaciones de los estudiantes estarán registradas en el sistema, esto hace posible que el reporte de estos resultados se pueda obtener en cualquier momento y así poder verificar el rendimiento y el nivel en el que los estudiantes se encuentran, para tomar medidas correctivas a tiempo, si es el caso.

Para implementar este ambiente de educación no hay necesidad de implantar nuevas salas de sistemas especializadas, es posible utilizar la infraestructura existente de la institución, lo que permitirá reducir costos. También es posible reutilizar múltiples herramientas que facilitan el aprendizaje de contenidos, como simuladores, juegos, entre otros.

Muchas herramientas utilizadas para llevar a cabo educación en línea también soportan actividades de administración. Las actividades como los procesos de matrículas se tornan a veces difíciles tanto para las personas encargadas de realizarlos como para los

estudiantes, ya que si algún estudiante vive muy lejos de la institución o se le presenta alguna eventualidad en el momento de desplazarse a la institución educativa a realizar su registro, es posible hacerlo desde su casa o su lugar de trabajo sin problemas, y así para los administrativos de la institución encargados del proceso será más fácil porque todos los datos estarán registrados en el sistema desde el principio.

Así como encontramos grandes ventajas en la educación en línea, también encontramos desventajas o factores que hay que tener en cuenta como los siguientes. Existe una fuerte dependencia de un proveedor de servicios de Internet (Application Server Provider o Internet Service Provider), ya que para llevar a cabo procesos de aprendizaje en este ambiente, es necesario trabajar todo el tiempo conectado a una red, es preciso contar con un proveedor de alta calidad y sobretodo que la conexión que proporcione sea constante y no intermitente, esto no solo evitará que ocurran errores durante el funcionamiento de la aplicación, sino también que haya inconsistencia de los datos en el sistema. Otra desventaja notoria es la baja velocidad de conexión, ya que en la mayoría de los casos ésta se realiza vía módem, lo que forja a que los contenidos tengan que ser diseñados de tal forma que el tiempo de descarga sea muy bajo.

Una de las amenazas a la que estamos expuestos al trabajar todo el tiempo conectado a Internet son los virus informáticos, ya que no se sabe si la institución educativa sea portadora de uno de ellos, y ésta sin saberlo sea quien lo distribuya. Como también se puede dar el caso de que al compartir documentos con compañeros, lo que en realidad se esté compartiendo sea un virus.

Otro factor que puede repercutir en la decisión de optar o no por ésta ambiente de educación es el costo de los equipos y acceso a Internet, para las personas que no los posean; ya que es cierto que hoy en día este costo ha disminuido considerablemente, pero no todas las personas se encuentran en capacidad de adquirir y financiar estos implementos.

Un elemento muy importante en este enfoque educativo son los contenidos que se distribuirán a los estudiantes, y si tenemos en cuenta que para la realización de los mismos se requiere coordinación entre los docentes encargados de la asignatura, y

además de que cuando se inicia la implementación este ambiente, la inexperiencia es evidente, se pueden lograr contenidos de baja calidad y que no son suficientes para alcanzar los objetivos propuestos por la asignatura.

Si tenemos en cuenta el contexto generado por la educación en línea y el eLearning se puede notar que se pierde el control sobre el ambiente de aprendizaje en el que se encuentra el estudiante. En un salón de clase el docente es quien controla que se den las condiciones apropiadas para las actividades educativas, pero en este ambiente el estudiante logra cierta independencia y es él quien decide su ambiente de aprendizaje, lo que no implica que sea el adecuado. Por ejemplo existe la posibilidad de que pierda tiempo divagando en el Web, pues no está exento de encontrar cosas que llamen más su atención como e-mails, mensajes en el Chat, entre otros y pierda la visión de lo que estaba estudiando y no lo haga más. También podemos notar que la interacción cara a cara del estudiante con el docente se pierde, debido a que aquí todo se convierte en un ambiente virtual, lo que implica que será difícil verificar si las actividades académicas en realidad fueron desarrolladas por el estudiante.

Debido al enfoque que maneja este ambiente de aprendizaje, se incrementan las actividades de coordinación del profesor, ya que la labor del docente es tan o más que las de un docente del aprendizaje tradicional. El docente requiere formación adicional, porque no es solamente aprender acerca de la nueva tecnología, si no realmente estar centrado en que este nuevo ambiente de educación tiene viabilidad, además de que se necesita encontrar la manera de llegar a los estudiantes, para que ellos se interesen en el aprendizaje, no es solo preocuparse por como utilizar el computador e ingresar los contenidos, es todo un proceso personal. También es necesario generar el contenido completo en línea, así sea por etapas, ya que los estudiantes van a su propio ritmo, y se pueden presentar casos en los que los estudiantes no necesitan profundizar tanto en algunos temas y avanzan con más rapidez a otros, y lo que ellos esperan es tener todo el contenido para estudiar y no tener que dar tiempo a que el docente termine el contenido de los capítulos siguientes.

En la realización de los cursos se pueden encontrar problemas de derechos de autor, porque aunque es cierto que a estos cursos sólo tienen acceso los usuarios registrados,

es muy sencillo para los estudiantes guardar los contenidos y redistribuirlos a terceros, sin tener en cuenta el trabajo realizado por el o los docentes al tener que estructurarlos.

Es verdad que se puede hacer reutilización de la infraestructura existente en la institución educativa, pero también es inevitable una fuerte infraestructura de soporte, necesaria no solo para la realización de las actividades académicas en línea, si no también para dar soporte a los usuarios, administrar los servidores, administrar la calidad de la transmisión de los datos en la red, entre otros.

2.1.2 MODELO GENERAL Y CARACTERÍSTICAS

En un ambiente de educación en línea se deben tener en cuenta ciertos componentes esenciales, que son: Una plataforma tecnológica de soporte, unas unidades técnicas, académicas y administrativas de soporte, la gente y los roles, los datos y la información que se espera recuperar, y finalmente el conjunto de procesos y tareas que se desarrollan.

En la Figura 1 se presenta un modelo general de los componentes que se deben tener cuando se monta un ambiente de educación en línea. En primera instancia se tiene una plataforma tecnológica de soporte o aplicación en la que el docente ingresa los contenidos para que el estudiante realice las actividades autónomas, colaborativas o competitivas de aprendizaje, baje los simuladores etc., a esta plataforma se le llama Learning Content Management System (LCMS) o Learning Management System (LMS), es un sistema administrador de contenidos que le permite al estudiante gestionar su aprendizaje y al docente generar y publicar los contenidos a través de él.

Esta plataforma tecnológica no funciona si el estudiante no es activo y autónomo en su proceso de aprendizaje; en muchos casos cuando se habla de educación a distancia se utiliza el término adulto, esto es porque este término da cierto grado de responsabilidad, pero una persona joven la mayoría de las veces está preocupada por cosas vanas que le dan diversión, y no consideran tan importante su proceso de aprendizaje en el momento, en este ambiente de educación se necesita una persona con autonomía y responsabilidad, activa a la hora de realizar las actividades académicas, que cuestione y proponga cosas al docente. Adicionalmente el profesor, como tutor, es un guía que ayuda

a la formación del estudiante, es una persona que se da cuenta como sus estudiantes están aprendiendo y ve que necesitan para un mejor aprendizaje, se da cuenta de las debilidades y fortalezas, cuáles son los estudiante que participan, es decir realiza un seguimiento del progreso de los estudiantes. Todas estas actividades se pueden realizar a través de la plataforma tecnológica.

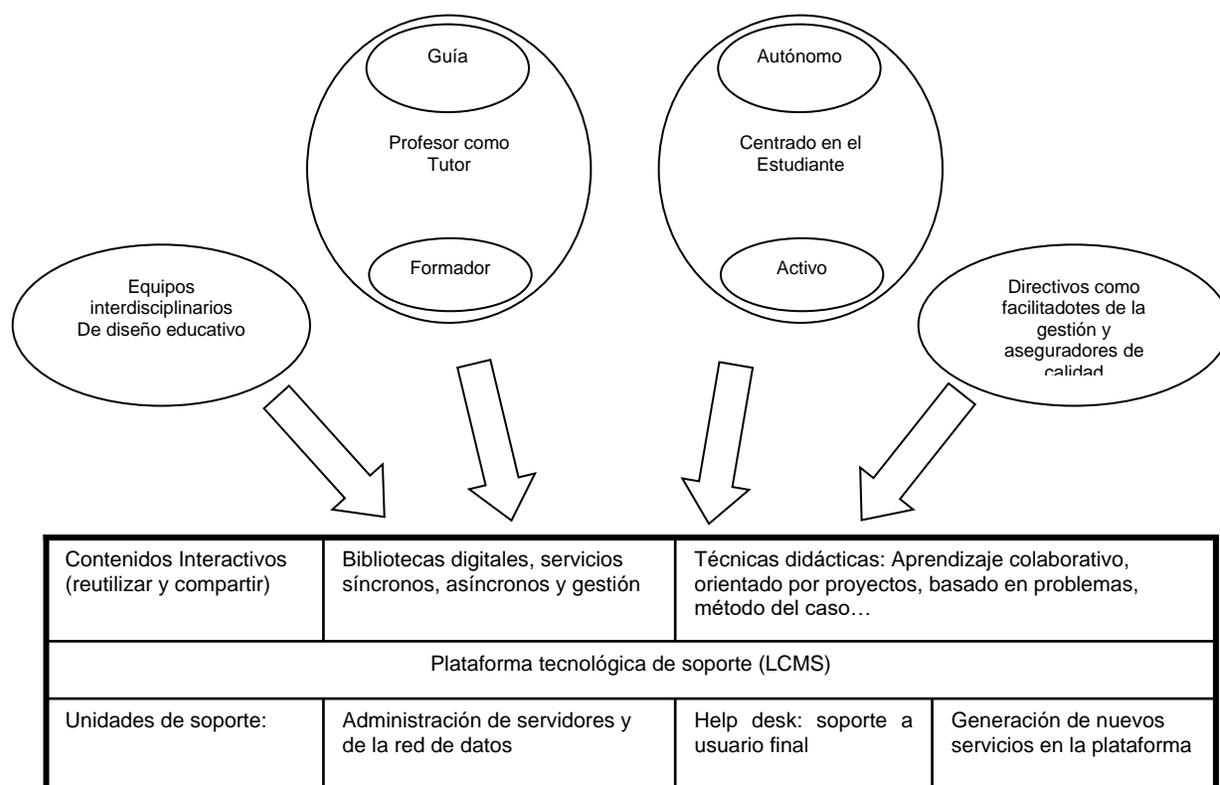


Figura 1 Modelo general de educación en línea

Para el desarrollo de los contenidos se tiene que hablar de equipos interdisciplinarios de diseño educativo y en general de un equipo de desarrollo curricular. Los equipos interdisciplinarios de diseño educativo contemplan: Director del proyecto, quienes previenen la reutilización, definen la estructura general del contenido y la secuenciación de los mismos en conjunto con los expertos en pedagogía y los expertos en el tema; y coordina el equipo para que se mantenga la integridad del proyecto. Programadores / desarrolladores de contenido, quienes empaquetan el contenido, escriben los meta datos de los recursos educativos u objetos de contenido compatible (SCOs) y programan la interacción con el LMS (Learning Management System). Diseñadores gráficos, su labor

consiste en Identificar, ubicar o generar gráficos, videos, animación y audio, además de describir los meta datos de estos medios. Expertos en el tema, son quienes trabajan con los pedagogos para dividir el contenido en las agregaciones correspondientes con sus reglas de secuenciación. Finalmente, bibliotecarios de contenidos, quienes cargan los contenidos en los repositorios, actualizan los meta datos y localizan material que se pueda reutilizar.

Se ha previsto por varios estudios realizados, que el desarrollo de un curso en línea tiene dos puntos que definen su éxito, la primera es que el contenido esté muy bien desarrollado, es decir, una persona no ingresa a un curso en donde tiene que esperar más de cinco minutos para descargar un contenido, los contenidos no deben ser diseñados solo para trabajarlos a través de la Web, sino también para que cuando un estudiante tenga la posibilidad de desarrollarlo tenga la suficiente motivación para trabajar y participar en las actividades propuestas para el; por tal motivo los equipos interdisciplinarios por lo menos necesitan una persona experta en el tema o que desempeñe ese rol, para cumplir satisfactoriamente con los requisitos. El otro punto importante es que se requiere un fuerte soporte pedagógico para que los contenidos educativos soporten todo el aprendizaje que se proponen brindar, utilizando las reglas de secuenciación adecuadas; máxime si se van a empezar a desarrollar procesos que están buscando calidad. Adicionalmente hay otras dos personas que piensan un poco en que los contenidos que se van a desarrollar estén bien estructurados y que además puedan ser reutilizados después en otros cursos, esto está dado por el estándar de SCORM.

También hay un conjunto de servicios que permiten manejar mucha información pero bien estructurada que está en Internet y algo muy importante, que es la tendencia actual, es que esta plataforma tecnológica de la posibilidad de desarrollar diferentes técnicas didácticas, entre ellas aprendizaje colaborativo, orientado por proyectos, basado en problemas, etc. Porque al no tener esto lo que obtenemos es un tutor que le va mostrando al estudiante contenidos, y eso no es más que replicar el ambiente presencial.

Hay unidades de soporte como la administración de servidores, red de datos, el soporte al usuario final y personal que esté constantemente agregando servicios a la plataforma, pero también viendo la forma como el cliente se puede apropiar de esos servicios.

En este modelo vemos que los directivos se convierten en facilitadores de la gestión, es gente que está preocupada principalmente por asegurar la calidad de la educación.

2.1.3 ESTÁNDARES Y ESPECIFICACIONES

Hoy en el mundo hay por lo menos 200 plataformas registradas, lo que nos pone a pensar que si se sigue un camino diferente, cada quien va a armar su isla de información, entonces se necesita un lenguaje común que permita dar un mayor crecimiento a ese trabajo. Muchas instituciones han tratado de armar un estándar básico para que todo el que haga una herramienta de desarrollo de educación en línea tenga unas bases comunes para intercambiar recursos de aprendizaje que sirvan en diferentes instituciones. Entre estos estándares tenemos el IMS Global Learning Consortium, Inc., the Aviation Industry CBT (Computer-Based Training) Committee (AICC), the Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE), como los más representativos, ellos y con el trabajo hecho en diferentes áreas por las universidades e instituciones, han desarrollado muchos conceptos y generaron un lenguaje común que lo debe cumplir cualquier curso, las partes del mismo y demás.

El departamento de Defensa de los Estados Unidos, después de realizar un trabajo similar, logró una propuesta llamada ADL (Advanced Distributed Learning) que agrupó las anteriores. Se realizaron pruebas con lo que se ha construido entre todos, se plantearon laboratorios, se hizo divulgación y vieron si realmente se llegaba a un punto consistente para todos los interesados en la educación en línea. Con los resultados obtenidos generaron modelos de referencia y esos modelos pasaron después a una organización mundialmente reconocida por manejar y definir estándares como lo es la IEEE.

Dentro de la iniciativa ADL, este trabajo se viene manejado bajo el nombre de SCORM (Sharable Content Object Reference Model) – Modelo de Referencia para Objetos de Contenidos Compartibles. Este modelo se viene trabajando desde 1998 y agrupa muchas empresas y universidades que trabajan con educación, en él se define un Modelo de Agregación de Contenidos (CAM, Content Aggregation Model) para un sistema de aprendizaje basado en el Web (WBL, Web Base Learning) y un ambiente de ejecución de objetos de aprendizaje. Su objetivo principal consiste en proveer una descripción técnica de los objetos de contenido de tal forma que sean fácilmente compartidos a través de

múltiples ambientes de aprendizaje en línea, para ello se define el Modelo de Objetos de Aprendizaje (LOM, Learning Object Model). El LOM es básicamente la descripción de un contenido, por ejemplo quién lo hizo, cuándo lo hizo, el objetivo que persigue ese contenido, y otras características importantes que permiten conocerlo sin tener que leer el contenido, esta descripción en el lenguaje de SCORM se llama *Meta datos*. De igual forma se está estableciendo un estándar que permita comunicar a los objetos de aprendizaje con el LMS, la cual básicamente resumen un conjunto de funciones, estados y reglas para el registro del desempeño del estudiante. Esto también permite que si el estudiante solamente leyó la mitad del contenido, éste le reporte al LMS el progreso del estudiante, es decir que la próxima vez que el estudiante ingrese, el sistema le recordará que ese contenido solamente fue leído hasta la mitad.

2.1.4 ARQUITECTURA SCORM

Cuando se hace un sistema gestor de contenidos de aprendizaje, se tiene un LMS o sistema gestor de aprendizaje que cumple con unas tareas específicas en cuanto al seguimiento del progreso educativo del estudiante, un CMS o sistema administrador de contenidos que cumple con otras tareas específicas como las de generar los contenidos, enlazarlos, además tiene las herramientas de autoría de contenidos para los profesores (ver Figura 2).

Entre otros elementos importantes de esta arquitectura está el visor, que es el que le permite al estudiante ver los contenidos de aprendizaje, desarrollar las actividades académicas, comunicarse con sus compañeros y demás. Se tiene también una API de comunicación entre el visor, el contenido cargado en él y el LMS; esta API es llamada CMI, es el protocolo con el que se comunica el contenido con el LMS, y es el encargado de enviar y reportar el progreso del estudiante en el contenido cargado por el visor, es decir que si el contenido también tiene actividades académicas a realizar, es el API quien envía los resultados al sistema.

definen los prerequisites necesarios, en cuanto estudio de contenidos para pasar a un contenido siguiente. Entre otros servicios relacionados a los contenidos, tenemos los servicios de pruebas y exámenes, el servicio de administración del curso como tal, el servicio de administración del perfil del aprendiz, en dónde se puede observar el progreso académico del estudiante.

2.1.5 MODELO DE AGREGACIÓN DE CONTENIDOS (CAM)

El CAM es un mapa (estructura del contenido) que se usa para agregar SCOs (Sharable Content Objects), dentro de unidades cohesivas de instrucción llamadas agregaciones, como los módulos, capítulos, lecciones, entre otros. Los SCOs son los recursos de aprendizaje como tal, es decir, ellos son los que en realidad tienen el contenido educativo que el estudiante seguirá.

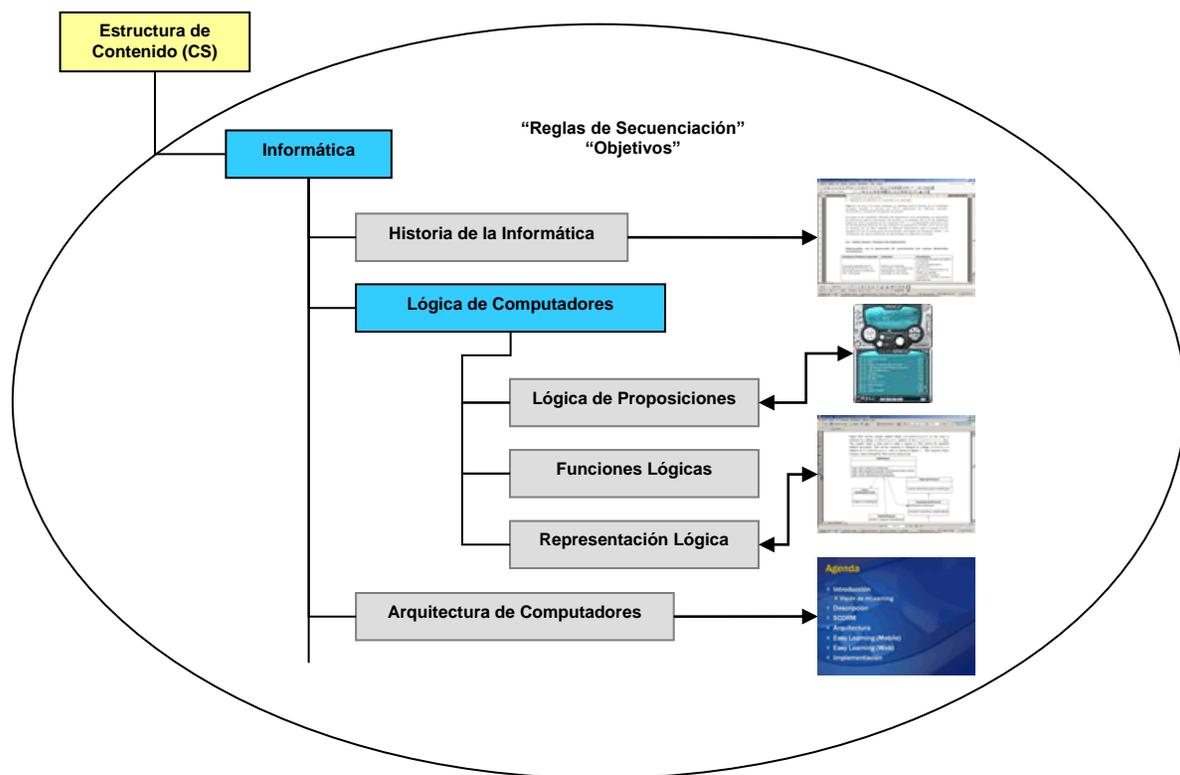


Figura 3 Modelo de Agregación de Contenidos

Estos SCOs son descritos por Meta datos, estos meta datos permite conocer el SCO sin tener la necesidad de acceder a él, es decir, dan una descripción detallada del recurso, por ejemplo quien lo hizo, en qué fecha lo hizo, cómo se llama el contenido, las palabras

claves del mismo, el formato del recurso, que permite saber en que tipo de aplicación será visto, el objetivo que persigue el contenido y otras características importantes del mismo. Los SCO se hacen con medios (assets en la terminología SCORM), es decir, texto, videos, imágenes, animaciones que al agruparlos le dan un sentido al SCO, estos medios son llamados assets. El CAM además define las reglas de secuenciación de los SCO, es decir el orden de cómo ellos serán presentados.

2.1.6 METADATOS DE LOS SCO (LOM)

El LOM permite describir y conocer un contenido para posteriormente reutilizarlo en cualquier curso y compartirlo si es necesario. SCORM y la IEEE han definido nueve categorías para la definición de los objetos de aprendizaje, ellas son (Figura 4):

- La categoría general que agrupa información general del recurso como un todo (identificador, título, categoría, lenguaje, descripción, palabras claves, cubrimiento, estructura y nivel de agregación).
- La categoría del ciclo de vida que agrupa las características relacionadas con la historia del actual estado del recurso y quienes lo han modificado durante su evolución (versión, estado y la gente o las organizaciones involucradas).
- La categoría del meta-meta datos que agrupa información sobre el propio registro de los meta datos, en lugar del recurso de aprendizaje que este registro describe (identificador, entrada del catálogo, personas u organizaciones que han modificado este meta datos, la versión de SCORM usada para el registro del meta datos y el lenguaje del meta datos).
- La categoría técnica que agrupa los requerimientos técnicos y las características del recurso (formato, tamaño, localización, requerimientos, proceso de instalación y duración de su ejecución).
- La categoría educacional que agrupa las características educacionales y pedagógicas del recurso (tipo de interactividad, tipo de recurso, nivel de interactividad, densidad semántica, usuario de destino, ambiente de aprendizaje típico, edad típica del usuario de destino, tiempo típico de trabajo con el recurso, dificultad, como usar el recurso y el lenguaje natural del recurso).

- La categoría de derechos que agrupa los derechos de propiedad intelectual y las condiciones de uso del recurso (costo, restricciones de derechos de autor que aplican y las condiciones de uso).
- La categoría de relación que agrupa las características que definen la interrelación entre este recurso y otros recursos de destino (Los recursos con los que se relaciona y el tipo de relación, entre ellas, “es parte de”, “esta basado en”, entre otras).
- La categoría de anotaciones que provee comentarios en el uso educativo del recurso e información de quienes y cuando se crearon esos comentarios (anotador, fecha y descripción).
- La categoría de clasificación que describe donde se ubica este recurso dentro de una clasificación particular (propósito, ubicación en la taxonomía, fuente de la taxonomía, descripción y palabras clave).

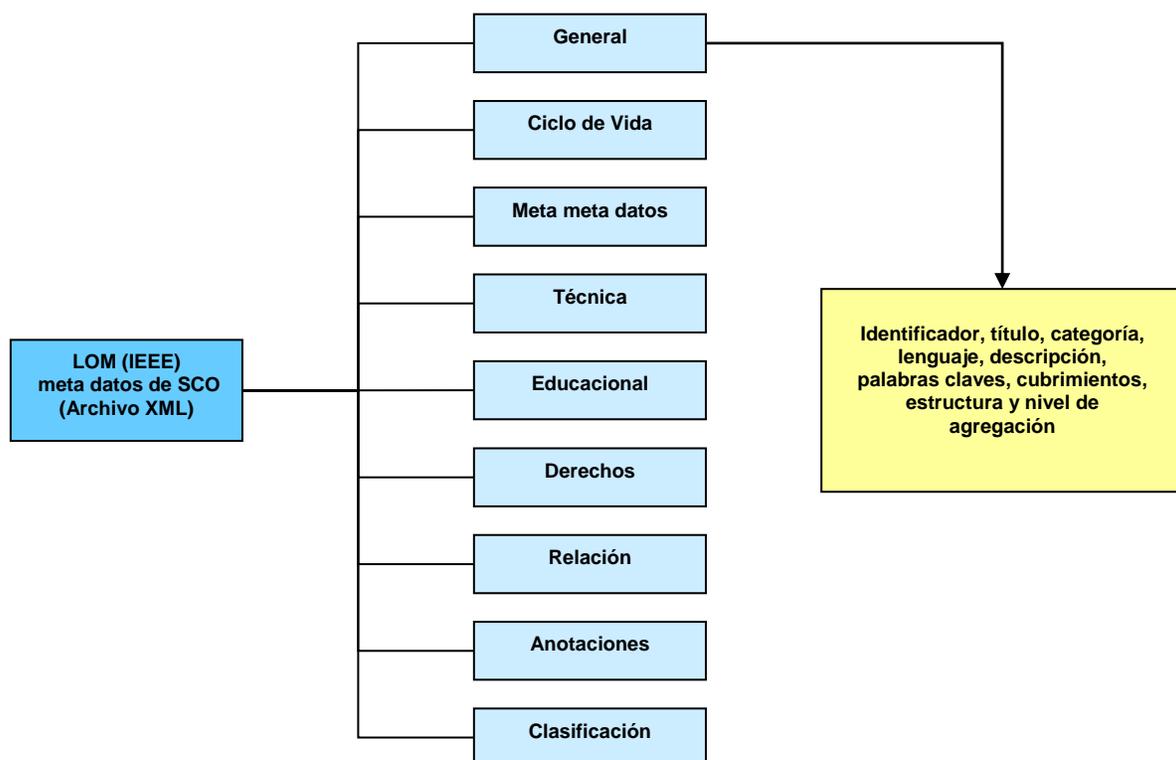


Figura 4 Meta datos de los objetos de aprendizaje

2.1.7 EMPAQUETAMIENTO DE CONTENIDOS (CP)

SCORM define el empaquetamiento de contenidos (Content Packaging, CP) como una forma de estandarizar el intercambio digital de cursos entre diferentes sistemas y herramientas. EL CP puede definir la estructura del curso/contenido (organización) y el comportamiento de los mismos (secuenciación y navegabilidad). Uno de los conceptos más importantes en el CP, es el archivo de manifiesto, este describe el paquete por si mismo, conteniendo lo siguiente:

- Identificador y versión.
- meta datos acerca del paquete.
- Una sección de organización que define la estructura del contenido (ítem por ítem, con sus relaciones jerárquicas y sus prerrequisitos) y el comportamiento.
- Una lista de referencias a los recursos del paquete.
- Submanifiestos.

Además del archivo de manifiesto, el empaquetado de contenido debe contener los archivos físicos que son referenciados en el manifiesto. La organización de los contenidos sigue una estructura jerárquica como la que se observa en la figura, donde el concepto de Modulo, Capitulo y Lección se puede generalizar al Concepto de Agregación y de esta forma adecuarlo a otras jergas menos ortodoxas como por ejemplo, fase, etapa, subcurso, punto de enseñanza, tarea, entre otros.

La secuenciación y la navegación dentro de las agregaciones y los SCOs se expresan por medio de reglas que actúan como información para que el LMS los convierta en requisitos. Estas reglas se pueden formar por expresiones que involucran agregaciones o SCOS que son operados con Y lógico, O Lógico, Negación, Igualdad, Diferencia, Conjuntos, Separador, Mínimo y precedencia.

2.1.8 TENDENCIAS

El mismo desarrollo que está realizando SCORM puede definir ciertas tendencias de trabajo, tanto en la plataforma como en los contenidos, este estándar principalmente ha suministrado la construcción de los meta datos, el cómo empaquetar los contenidos, y en

este momento se encuentra trabajando fuertemente en la evaluación. Hoy, ya se consiguen borradores a cerca de cómo elaborar evaluaciones estándar dentro de la educación en línea.

En cuanto a la evaluación el IMS ya publico un documento en borrador, el mismo que está trabajando SCORM, en el que se definen todos los tipos de evaluación, por ejemplo evaluaciones de múltiples respuestas con opción única, opción múltiple, evaluaciones abiertas, cerradas, preguntas al estilo de las sopas de letras, y otros tipos de evaluaciones posibles de trabajar en los diferentes sistemas.

También se está trabajando a cerca de la identificación del perfil del aprendiz, muchas herramientas definen el perfil como los desarrolladores lo han visto, otras aplicaciones no lo toman en cuenta; así que se está trabajando en como identificar y diferenciar los perfiles de los estudiantes. Otros trabajos que se están realizando es cómo utilizar la simulación dentro del proceso de aprendizaje, como unificar los repositorios para compartir contenidos, como hacer la presentación y la navegación en forma estándar, y específicamente en la parte del dominio del aprendizaje como estandarizar el aprendizaje procedimental, el soporte al rendimiento, el aprendizaje de competencias, el aprendizaje adaptativo, los sistemas de aprendizaje inteligente, los sistemas de juegos, los sistemas de aprendizaje móvil y demás.

2.2 MLEARNING

McManus [3] esta convencido de que toda nueva tecnología trae consigo cierto potencial que puede ser usado en procesos de enseñanza aprendizaje. La pregunta esencial que se debe responder para saber si ese potencial es real, es: ¿Que puede hacer el instructor y el aprendiz con esta nueva tecnología que no puede hacer con las otras que ya existen? Además en su curso de doctorado en la Universidad de Oulu establece unas preguntas claves que se deben tomar en cuenta en el desarrollo de proyectos relacionados con mLearning, estas son: ¿Qué son las tecnologías móviles?, ¿Cuáles son las capacidades y características únicas de esta tecnología?, ¿Cuáles son las implicaciones teóricas?, ¿Cuales son las implicaciones pedagógicas?, ¿Cuales son las implicaciones de diseño de soluciones con esta tecnología?, ¿Cómo se puede usar esta tecnología como parte de un ambiente de aprendizaje más amplio?, ¿Cómo se puede usar esta tecnología en un

ambiente de aprendizaje independiente?, ¿Qué investigaciones se han realizado o se están realizando para resolver estas preguntas? y ¿Cuáles han sido los resultados de estas investigaciones?.

2.2.1 DEFINICIÓN

Quinn [2] afirma que el mLearning es eLearning a través de dispositivos computacionales móviles: Dispositivos Asistentes Personales (Personal Digital Assistant, PDA, como las Palm y las Pocket PC), Máquinas Windows CE (Entre ellos los computadores de mano o handheld, los computadores portátiles o Laptop's y los Table PC) y teléfonos celulares, o como la intersección de la computación móvil y el eLearning, la cual se caracteriza por la capacidad de acceder a recursos de aprendizaje desde cualquier lugar, en cualquier momento, con altas capacidades de búsqueda, alta interacción, alto soporte para un aprendizaje efectivo y una constante valoración basada en el desempeño. Considera de esta forma al mLearning como eLearning verdaderamente independiente de la ubicación en espacio y tiempo. Es preciso comentar que estos dispositivos también son conocidos como Dispositivos Inteligentes o Dispositivos de Información (DI).

La visión de Quinn sobre la computación móvil se basa en computación portátil con alta interactividad, conectividad total y alto procesamiento. Un dispositivo pequeño que esta siempre en red, que permite una fácil entrada de datos a través de plumas, dictados o por un teclado si es necesario, y la habilidad de ver imágenes con alta resolución y alta calidad de sonido.

García [14] comenta que es una práctica común combinar los cursos basados en la Web con otros medios, tales como CD-ROMS, videos, clases satelitales y clases presenciales. Esto se hace porque, a pesar de las múltiples ventajas y la versatilidad de la Web, sus capacidades son limitadas, hecho que en ciertas oportunidades obliga al diseñador a acudir a diferentes medios y formatos para la distribución de los contenidos del curso. Desde este punto de vista define el mLearning como el uso de la Web junto con tecnología móvil.

Además Harris [14] define el mLearning como el punto en que la computación móvil y el aprendizaje electrónico se interceptan para producir una experiencia de aprendizaje en cualquier momento y en cualquier lugar. El mLearning incluye el uso del teléfono celular, una Palm Pilot u otro PDA [15] para que el usuario tenga al alcance de su mano los recursos del curso disponibles dondequiera que se encuentre, es decir, pueda consultar

material de aprendizaje (guías o lecturas), chequear las actividades de aprendizaje, buscar información actualizada sobre un curso que esté tomando o tomar una prueba que le permita evaluar su nivel de conocimientos. De hecho es posible que el usuario pueda bajar el material que le interese, trabajar con éste fuera de línea (si los desea), reconectarse para enviar los resultados y recibir la retroalimentación.

En resumen, se puede decir que el mLearning se constituye en la utilización de las "tecnologías móviles" al servicio de los procesos asociados con la enseñanza y el aprendizaje.

2.2.2 ACCESO INALÁMBRICO A INTERNET

Un acceso inalámbrico a Internet usa nodos fijos y móviles [16]. Desde el punto de vista de una aplicación, un nodo móvil se asemeja a un nodo fijo. Los protocolos de transporte como TCP, UDP e IP normalmente se usan tanto en los nodos fijos como en los móviles.

En el nodo móvil, la primera propiedad distintiva es la conexión inalámbrica. Hace referencia al protocolo de comunicación que permite conectar los dispositivos a la red a través de tecnología hardware inalámbrica. En la actualidad se usa el estándar IEEE 802.11(a/b/g) que es independiente del vendedor y se han empezado a usar otras tecnologías como el estándar infrarrojos IrDa, la HiperLan basada en radio ETSI y Bluetooth. La segunda propiedad distintiva consiste en la administración móvil. Esto es, que el usuario puede moverse desde un lugar a otro manteniendo la comunicación con otros nodos en la red sin interrupciones. Desde el punto de vista de la aplicación, esta funcionalidad es transparente. Sin embargo, algunas aplicaciones pueden beneficiarse de saber que los usuarios están ubicados en un nodo móvil. La tercer y última propiedad distintiva consiste en que existen clases de nodos especiales en el borde de la red móvil que dan el acceso a la red fija, ellos son el punto de acceso a través de los cuales la red alámbrica y la inalámbrica se conectan (Figura 5).

2.2.3 CAPACIDADES Y CARACTERÍSTICAS

Los DI dan la posibilidad de aumentar la personalización de los contenidos y de la misma aplicación de aprendizaje, ya que son más económicos, más fáciles de adquirir y son realmente personales. Hoy día, sus costos varían entre 100 y 900 dólares aproximadamente.

Dos aspectos importantes que se deben tener en cuenta al momento de distribuir los contenidos de mLearning son, el costo de la comunicación inalámbrica en cada país y las imitaciones propias de cada dispositivo. Algunas de las limitaciones más importantes se relacionan con el tamaño de las pantallas, las limitaciones de los teclados o de las plumas que se utilizan para la escritura de datos, entre otras.

Cuando se selecciona una computadora de mano [17], determinar previamente el uso que se le va a dar, ayuda a tomar una mejor decisión en cuanto a las características del equipo. Algunas de las características claves son: Software. Los sistemas operativos más populares son Palm OS, Microsoft Pocket OS y Microsoft Windows CE. Si es importante tener acceso a Word, Excel y otras aplicaciones entonces Pocket OS y Windows CE pueden ser la mejor opción. Los teléfonos celulares en su mayoría disponen de un navegador Web basado en WAP. Despliegue. Las pantallas vienen a uno y varios colores, y en diferentes resoluciones. Algunas aplicaciones instructivas pueden requerir color. Conectividad. Si necesita acceder a información remota será necesario un modem alámbrico o inalámbrico, una tarjeta de red WiFi o Bluetooth, entre otros. Y Capacidad de expansión. Algunos computadores de mano tienen ranuras de expansión. Adicionar hardware, tal como un teclado desprendible puede ser crítico para algunas aplicaciones instructivas. Otras permiten almacenar hasta 256MB en almacenamiento externo y esto permite transportan gran volumen de información fuera de línea.

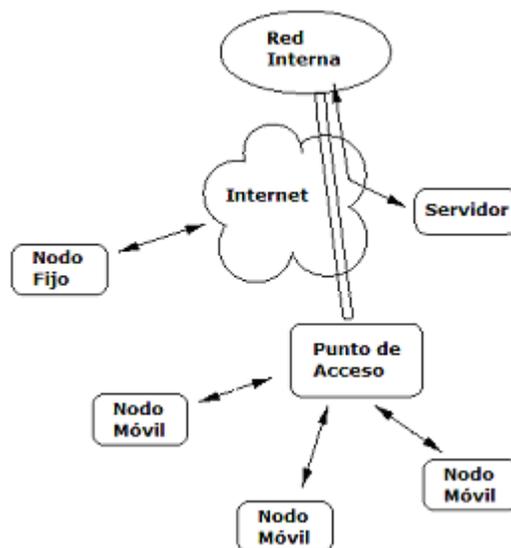


Figura 5 Arquitectura general del acceso a la red alámbrica e inalámbrica

Otros factores claves que se deben tener en cuenta son el tiempo de uso de la batería, la garantía, las actualizaciones y las reparaciones. Otros DI como cámaras y scanners pueden ayudar en el proceso de aprendizaje. Quinn [2] afirma que pronto no habrá distinción entre eLearning y mLearning, debido principalmente a que día a día los dispositivos se vuelven más poderosos y la diferencia entre un DI y un PC de escritorio desaparecerá.

2.2.4 IMPLICACIONES TEÓRICAS

Las aplicaciones (front-end) de aprendizaje en los DI deben asegurar que exista comunicación con el Sistema Gestor de Aprendizaje (Learning Management System, LMS) para registrar el progreso del estudiante, así como para habilitar los procesos relacionados con la certificación [2]. De lo anterior se desprenden dos implicaciones importantes:

- La primera, consiste en cómo se debe administrar el aprendizaje a través de una conexión intermitente. Aunque es preferible que los individuos tomen responsabilidad de su propio aprendizaje, muchas circunstancias sugieren que es beneficioso tener sistemas que rastreen y administren el progreso del aprendizaje por parte de los aprendices. Para solucionar esto se pueden plantear dos alternativas. La primera consiste en adicionar a los visores Web que corren en los DI algunos plug-ins o applets, y la segunda consiste en realizar soluciones de aprendizaje autónomas (standalone) y propietarias. El problema con los plug-ins y los applets es que trabajan un poco diferente de acuerdo al visor y las plataformas, lo cual genera ciertos inconvenientes para mantener el sistema. Las soluciones propietarias en algunos casos dificultan el manejo de contenidos diferentes y limitan su flexibilidad. Con el objetivo de facilitar el intercambio de contenidos entre aplicaciones propietarias, se han desarrollado algunas iniciativas por estandarizar la distribución de los cursos, los objetos de contenido (Sharable Content Objects, SCO), las preguntas y evaluaciones, entre otras cosas (SCORM, LTSC, IEEE) [4].
- La segunda, se relaciona con la necesidad de contar con soluciones de plataforma cruzada, significa que todos los aprendices deben tener acceso a todo el material independiente de las preferencias de un sistema operativo, o un ambiente en particular. Para el desarrollo de este tipo de soluciones se está trabajando en formas avanzadas de representación de la información separando el contenido del formato,

un ejemplo de esto es XML, el cual permite especificar el contenido y también como se presentara de acuerdo a cada dispositivo. Esto requiere algunas revisiones importantes en el proceso de desarrollo de contenidos y la utilización de las herramientas adecuadas que cumplan con esta promesa. En esta misma línea están trabajando Java y Microsoft .NET, los cuales están promoviendo el uso de una jerarquía de clases base y un interprete universal que permitan desarrollar aplicaciones que se ejecuten en diferentes DI, sin tener que modificar el código.

2.2.5 IMPLICACIONES PEDAGÓGICAS

El tiempo potencial para la educación se puede incrementar [16], el uso de DI puede ser más oportuno para procesos de aprendizaje autónomo, la ubicación del usuario puede afectar el contenido educativo que éste recibe, por ejemplo en una exhibición, la descripción de un sitio cercano puede aparecer, los usuarios pueden acceder fácilmente a noticias cortas, por ejemplo, preguntar por la información de un lugar mientras se esta moviendo hacia esta ubicación, el equipo terminal puede proveer una manera conveniente de identificar al usuario y usarse como un método flexible de pago y debido a que el equipo terminal es personal, el usuario tiene un amplio control sobre éste, conocer muy bien como usarlo y personalizarlo de acuerdo a sus requerimientos. Finalmente la aplicación le puede proveer además de ayudas en su proceso de aprendizaje autónomo, un ambiente virtual de aprendizaje colaborativo.

En un ambiente verdaderamente abierto seria posible personalizarlo hasta el punto donde el usuario puede convertirse en desarrollador del ambiente y de la educación.

2.2.6 IMPLICACIONES DE DISEÑO

En las redes fijas, el usuario no necesita poner atención en el costo de la comunicación o la disponibilidad del servicio, puesto que los recursos son abundantes. Por otra parte, cuando se usa acceso inalámbrico, se debe tomar en consideración las características especiales del ambiente inalámbrico y móvil [16], ellas son: Las comunicaciones inalámbricas tienen una capacidad limitada, la calidad de la conexión es fluctuante, los nodos tienen recursos limitados, los nodos pueden quedar temporalmente fuera de alcance y los costos de comunicación son altos. Esto tiene un impacto significativo en la estructura del contenido para educación y deben ser reconsiderados en sistemas móviles. Todos los diseños de aplicaciones de educación convencionales para redes fijas pueden

ser usados en redes inalámbricas, suministrando los recursos adecuados y haciendo algunos cambios debido a las comunicaciones móviles.

El ambiente móvil causa un conjunto de condiciones que deben ser consideradas cuando se desarrolla contenido educativo, ellas son: La aplicación no se debe atar a un lugar, equipo o tiempo en particular. El mismo paquete de educación debe poderse usar con dispositivos de diversas capacidades. Si es necesario, el contenido puede ser pre-procesado en el servidor para diversos terminales. El contenido se debe adaptar a las características fluctuantes de capacidad de las redes inalámbricas. Es decir, las fluctuaciones en la capacidad de transmisión no deben afectar la robustez de la aplicación educativa. El contenido debe soportar operaciones fuera de línea y operaciones de extrema limitación de ancho de banda.

A pesar de la flexibilidad requerida en la manipulación del contenido, el dueño del contenido debe tener la capacidad de proteger sus derechos de propiedad intelectual. Esto puede ser implementado usando contenedores de información seguros, los cuales permiten acceder la información solo a los usuarios autorizados. Típicamente, esto requiere un mecanismo de delegación de acceso, el cual restringe el acceso a la información independiente de su ubicación.

Hasta el momento se destacan dos formas de utilizar las tecnologías móviles en el aprendizaje:

- Una se refiere a la posibilidad de "descargar" de la red objetos de aprendizaje o cursos completos a un PC o un DI. Cuando la persona finaliza su estudio, conecta de nuevo el dispositivo a la red, y "carga" los resultados de las actividades y las evaluaciones al sistema de administración del aprendizaje (LMS). En eLearning se requería que el usuario estuviera siempre conectado a la red (en línea) para poder estudiar.
- La otra tendencia se refiere a la conexión permanente por medio de una red inalámbrica, lo que permite al usuario desplazarse con su DI por cualquier lugar y permanecer conectado. Por ejemplo en la universidad Carnegie-Mellon un estudiante puede alquilar un PC durante 45 minutos, desplazarse por cualquier sitio de la universidad y estar siempre conectado a la red.

2.2.7 MLEARNING EN UN AMBIENTE DE APRENDIZAJE MÁS AMPLIO

García [14] referencia una experiencia interesante acerca de las compañías INSEAD, NOKIA e ICUS. El objetivo de estas tres empresas, consistía en implantar un curso utilizando en forma combinada la Web y la tecnología móvil. Los estudiantes podían acceder al 80% del contenido del curso programado para 20 horas a través de su PC o de su teléfono celular. El curso fue desarrollado para la Web y para WAP. El formato en WAP se desarrolló como un sistema de menús jerárquico y requirió de textos más cortos, pantallas adicionales y mayor número de títulos. Se reportó que los estudiantes accedieron sólo entre el 40 y 50% del material WAP, específicamente por las limitaciones que presentan los teléfonos en cuanto a tamaño de la pantalla, limitadas capacidades gráficas, lenta velocidad de procesamiento y poca capacidad de almacenamiento, a pesar de las ventajas en cuanto a facilidad de acceso a la información.

Es indudable que el uso combinado de tecnología Web, con la tecnología móvil está todavía en sus inicios. Tiene que superar esas limitaciones para convertirse en una herramienta potencialmente útil que mejore la flexibilidad de los sistemas de enseñanza en línea. Por ahora, su uso se limita a ciertas tareas como: Avisos, noticias, alertas y tips diarios, medio básico de comunicación, realización de evaluaciones rápidas, revisión de material de apoyo en línea, búsqueda de información acerca de un tópico específico, revisión de otros sitios WAP, registro y matrícula de cursos y programas académicos, entre otros.

Sin embargo, hay mucha gente que le apuesta a los PDAs para potenciar la eficiencia de los sistemas de aprendizaje basados en la Web. Desde esta perspectiva se podría decir que el mLearning se convierte en una ampliación del eLearning o del aprendizaje mixto.

2.2.8 MLEARNING EN UN AMBIENTE DE APRENDIZAJE INDEPENDIENTE

Desde la perspectiva de usar los diversos DI dentro de un ambiente que provee la totalidad del espacio de aprendizaje computacional que se le ofrece al estudiante, se han desarrollado un conjunto variado de aplicaciones. Algunas de las desarrolladas para Palm OS [15], son

- Astro Info (v2.11). Grados 6 – 12. Es un calendario astronómico que despliega información grafica del sol, la luna y ocho planetas más en múltiples lenguajes.

- Children's Illustrated eTales (v1.01). Kinder - 2. Visualiza cuentos electrónicos cortos con figuras a todo color que facilitan su lectura.
- Cooties v4.0. Grados 9 – 12. Es un programa de simulación de transferencia de virus. El programa incorpora la solución del problema y alienta el trabajo cooperativo entre los estudiantes. Los profesores tienen la posibilidad de determinar el tiempo de incubación, niveles de inmunidad individual y muchas otras características. Los estudiantes colaboran entre ellos para determinar quien es el portador inicial y rastrean el camino del contagio.
- Herbert's Math Time (v1.0). Kinder – 5. Es un programa que ayuda a los niños a practicar sus habilidades matemáticas. Incluye suma, resta, multiplicación y división.

2.2.9 PROYECTOS DE I+D RELACIONADOS

Fleischman [17] referencia como la Escuela de Secundaria Waldo Emerson Lessenger en Detroit ha introducido en forma exitosa el uso de las Palm para las clases de ciencias en sexto grado. Usando las Palm, los estudiantes realizan lecciones de máquinas y fuerza en software personalizado. Los estudiantes también usan las Palms como parte de actividades externas donde tienen la oportunidad de registrar observaciones y compartirlas con todo el salón de clases.

En el Institute for Learning and Research Technology de la Universidad de Bristol [18] han desarrollado un proyecto de investigación con mLearning y dan las siguientes recomendaciones: Usar formato HTML preferiblemente, usar lo menos posible otros formatos y de esta forma evitar la instalación de programas en la PDA, que en algunos casos se deben comprar, disminuir la entrada de datos por parte del estudiante, habilitar la consulta de información en Internet, permitir el envío y recepción de mail en las PDA, compartir documentos a través de FTP u otro sistema y Complementar las lecturas, por ejemplo con sesiones de preguntas y respuestas (una nueva dimensión a la lectura).

En The Mobile Learning Environment Project de la University of North Carolina at Wilmington están desarrollando una ambiente de mLearning fácil de usar y para múltiples dispositivos y plataformas, en el cual integran colaboración, comunicación y computación, con transferencia de datos basados en Web Services. Actualmente se conocen adelantos en la construcción de la aplicación pero no de pruebas de campo.

2.3 MICROSOFT .NET

Las tecnologías Microsoft .Net fueron escogidas para la elaboración del proyecto debido al enfoque arquitectural que implementan en el desarrollo de aplicaciones basadas en Internet, el cual adopta un modelo fundamentado en servicios Web, a través de los cuales es posible brindar al los usuarios la información personalizada y precisa que requieren en cualquier lugar, en todo momento y con cualquier dispositivo.

Los servicios Web son componentes de Software, o islas de funcionalidad programáticamente accesibles en Internet que utilizan protocolos abiertos de Internet, incluyendo HTTP (Hyper Text Transfer Protocol) para transporte, SOAP (Simple Object Access Protocol) para describir como interoperar las aplicaciones y XML (eXtensible Markup Language) para la representación de información variada y flexible. El modelo de programación de los servicios Web es completamente independiente de la plataforma y se adapta de manera ideal a la naturaleza heterogénea de Internet. Además el modelo de servicios Web basados en XML no se encuentra vinculado a arquitecturas particulares del hardware, gracias a los modelos abiertos e independientes de la plataforma sobre la que están contruidos.

2.3.1 EL MODELO DE COMPUTACIÓN DISTRIBUIDA

La industria de la computación está evolucionando rápidamente y está adoptando a Internet como una infraestructura avanzada y rica en comunicaciones en la que se pueden desarrollar aplicaciones distribuidas a gran escala, en donde se posibilite una nueva manera de hacer e integrar las aplicaciones. Lo anterior permite romper las barreras entre distintas aplicaciones que tienen información, barreras entre sistemas, barreras entre los sistemas y la gente que los utiliza, barreras entre las organizaciones.

Una gran variedad de nuevos dispositivos hardware como: celulares, Tablet PCs, Pocket PCs, TV entre otros dispositivos hogareños, están invadiendo el mercado, los cuales estarán conectados entre sí, con servidores y distintas aplicaciones, a través de Internet como elemento integrador. A partir de esta necesidad de permitir un modelo de conectividad que integre una gran variedad dispositivos y sistemas se va dando forma al nuevo modelo de computación distribuida llamado servicios Web basados en XML. El

objetivo es entonces consiste en permitir comunicar entre si a sistemas heterogéneos dentro y fuera de la empresa. Esta comunicación es independiente del sistema operativo, lenguaje o modelo de programación. Para conseguir esto se desarrollaron estándares adicionales a XML y SOAP como:

- UDDI: (Descubrimiento, Descripción e Integración Universal) Lenguaje que permite publicar, encontrar y usar los Servicios Web basados en XML. Es la 'Página Amarilla' de los servicios Web es decir un directorio para poder encontrarlos. Puede ser accedido con un explorador en <http://www.uddi.org/> o programáticamente ya que UDDI es también un servicio Web.
- WSDL: (Lenguaje de Descripción de Servicios Web) Lenguaje por medio del cual un servicio Web describe entre otras cosas qué hace o qué funcionalidad implementa.

En resumen, con el uso de los servicios Web basados en XML se integra la información que puede ser accedida desde distintos dispositivos, desde distintas plataformas de hardware o software y que puede estar guardada en distintos formatos.

2.3.2 LA PLATAFORMA .NET

Con base en los estándares abiertos de Internet, Microsoft .NET es la plataforma de servicios Web XML para potenciar aplicaciones, servicios y dispositivos a fin de que trabajen juntos para permitir el acceso y acción en la información en cualquier momento, en todo lugar y desde cualquier dispositivo; proporcionando desarrollo simplificado y experiencias ricas del usuario, así como oportunidades sin precedentes para los negocios (ver Figura 6).

El objetivo de la plataforma .NET consiste en simplificar el desarrollo de aplicaciones Web, distribuyendo software como servicios. Proveer las herramientas y tecnologías para transformar a Internet en una plataforma de computación distribuida en gran escala. Esta plataforma además soporta los estándares sobre los cuales se basan los servicios Web. La plataforma .NET utiliza tecnologías existentes, productos modificados para su uso dentro de la plataforma y elementos nuevos. Los puntos claves de .NET son:

- La Plataforma Microsoft .NET incluye la infraestructura y las herramientas para crear y poner en funcionamiento una nueva generación de servicios.

- Los productos y servicios de Microsoft .NET incluyen a Microsoft Windows .NET, MSN.NET, los servicios de suscripción personales, Office.NET, Visual Studio.NET y Microsoft bCentral para .NET.
- Los servicios .NET para terceros son una gran variedad de socios y desarrolladores que tendrán la oportunidad de producir servicios verticales y corporativos creados en la plataforma .NET.

Que es la Plataforma .NET

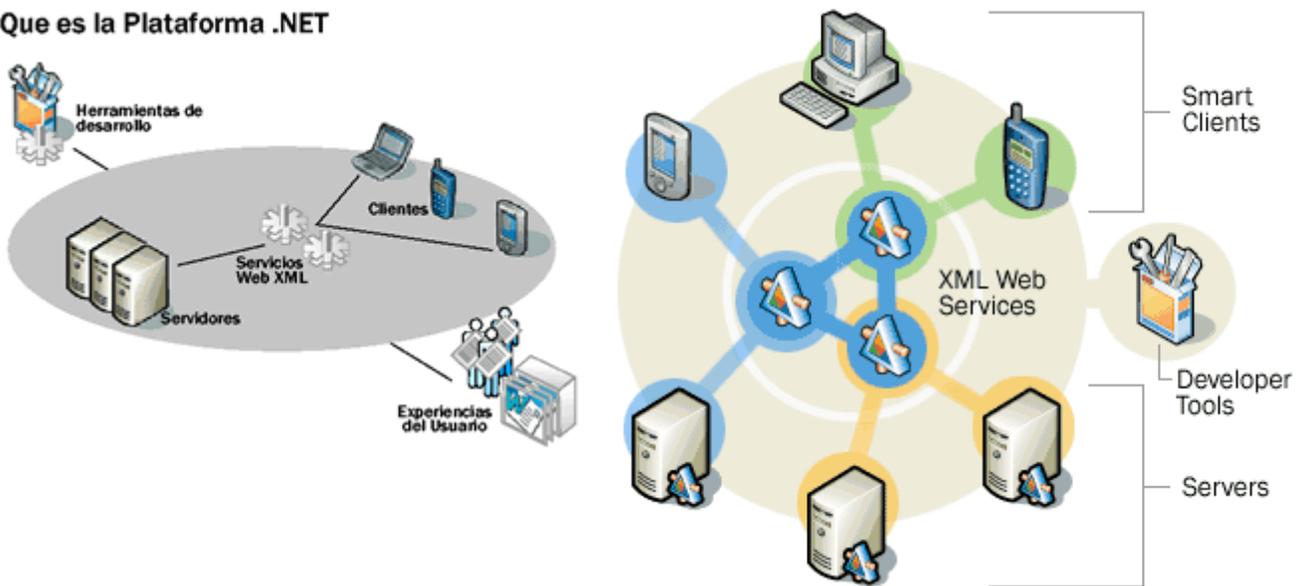


Figura 6 Arquitectura general de la plataforma .Net

2.3.3 EL MICROSOFT .NET FRAMEWORK

El .Net Framework es un ambiente administrado y seguro para el desarrollo y ejecución de aplicaciones. El .Net Framework administra todos los aspectos de la ejecución de los programas que requieren de este para ejecutarse. Realiza la asignación de memoria para el almacenamiento de datos e instrucciones, otorga y deniega los permisos apropiados para las aplicaciones, inicia y administra el ambiente de ejecución, y administra la reasignación de memoria de los recursos que ya no son utilizados. Sus principales componentes son el Common Language Runtime (CLR) y la Biblioteca de Clases del .Net Framework (NET Framework Class Library, FCL).

El CLR abstrae los servicios del sistema operativo y sirve como un motor de ejecución para las aplicaciones administradas (aplicaciones cuyas acciones están sujetas a aprobaciones del CLR). Este provee el núcleo de servicios, tales como compilación de código, asignación de memoria, administración de hilos, y recolección de basura. A través del Sistema de Tipos Comunes (*CTS*, Common Type System), refuerza el tipo de seguridad estricta y se asegura que el código es ejecutado en un ambiente confiable, reforzando también la seguridad de acceso de código.

La FCL proporciona el núcleo para las funciones de programación, tales como las que antes solo estaban disponibles a través de la API de Windows, y las funciones de nivel de aplicación usadas para el desarrollo Web, Windows, acceso a datos, seguridad y administración remota. Provee una colección de tipos útiles y reusables que son diseñados para integrarse con el CLR. Estos tipos son orientados a objetos y totalmente extensibles. EL FCL provee acceso a todas las características del CLR y se encuentra organizado dentro de espacios de nombre o *namespaces*. Cada namespace contiene una funcionalidad relacionada a un grupo de clases.

Las aplicaciones .Net no son ejecutadas de la misma forma como las aplicaciones Windows tradicionales. Contrario a estas que deben ser compiladas dentro de un ejecutable que contiene código nativo, el código de las aplicaciones .Net es compilado dentro de un metalenguaje llamado Lenguaje Intermedio de Microsoft (Microsoft Intermediate Language, *MSIL*) o también conocido como Lenguaje Intermedio Común (Common Intermediate Language, *CIL*), para luego ser almacenado en un archivo llamado ensamblado (*assembly*). MSIL es un formato binario que contiene instrucciones de bajo nivel que pueden ser solamente entendidas por el compilador incluido en el .Net llamado compilador *JIT* (Just in Time). Cuando se ejecuta una aplicación de este tipo, el JIT verifica que el código MSIL sea completamente seguro y a continuación lo transforma en código de máquina teniendo en cuenta el sistema operativo y el procesador, y lo ejecuta. A toda aplicación o biblioteca generada en MSIL y que se ejecuta bajo el control del CLR se le denomina *código administrado*, el cual resuelve muchos de los problemas de programación en Windows, como el registro de componentes y versiones (algunas veces llamado infierno de las DLL), debido a que los ensamblados contienen toda la información de tipos y versiones que el CLR necesita para correr la aplicación.

2.3.4 EL COMMON LENGUAJE RUNTIME (CLR)

El motor de ejecución de las aplicaciones en el .NET Framework, proporciona una serie de servicios, entre los que se incluyen:

- Administración del código (carga y ejecución).
- Acceso a los meta-datos que describen los tipos, los miembros y las referencias del código, para localizar y cargar clases, colocar instancias en memoria, resolver invocaciones a métodos, generar código nativo, exigir mecanismos de seguridad y establecer los límites del contexto del tiempo de ejecución.
- Administración de memoria para los objetos administrados.
- Control de excepciones, incluyendo excepciones en varios lenguajes.
- Interoperabilidad entre el código administrado, los objetos COM y las DLL anteriores (datos y código no administrados).
- Aislamiento de la memoria de la aplicación.
- Aplicación de seguridad de acceso al código.
- Comprobación de la seguridad de los tipos.
- Conversión de IL a código nativo.
- Automatización del diseño de objetos.
- Soporte para servicios de desarrolladores (creación de perfiles, depuración, etc.).

2.3.5 LENGUAJES DE PROGRAMACIÓN

El .Net Framework está diseñado para tener compatibilidad a través de lenguajes, gracias al CLR, así que un lenguaje es meramente un medio sintáctico para producir MSIL, y con unas pocas excepciones, cualquier cosa que se haga en un lenguaje, podrá hacerlo en todos los otros también. Así que una aplicación escrita en Microsoft Visual Basic .Net puede referenciar una librería de enlace dinámico (DLL) escrita en Microsoft Visual C#, la cual a su vez puede acceder a recursos escritos en Microsoft Visual C++ administrado o en algún otro lenguaje .Net. Esta interoperabilidad entre lenguajes se extiende totalmente a la herencia de orientación a objetos. Microsoft provee actualmente compiladores CIL para 5 lenguajes: C#, J#, C++, Visual Basic y JScript. El SDK (Software Development Kit) del .Net Framework igualmente incluye un ensamblador CIL llamado ILASM, así que es posible escribir código CIL puro si se desea.

2.3.6 MÓDULOS ADMINISTRADOS

Cuando se construye un programa con algún compilador capaz de general MSIL, el compilador produce un modulo administrado, el cual es simplemente un ejecutable diseñado para ejecutarse a través del CLR. Dentro de un módulo administrado hay cuatro elementos importantes:

- Un archivo cabecera Ejecutable Portable de Windows (PE, Windows Portable Executable)
- Una cabecera CLR que contiene información importante acerca del módulo, tales como la localización de su MSIL y Meta data.
- Un Meta data que describe cada cosa dentro del modulo y de sus dependencias externas, esta meta data no es opcional y es la que hace que cada modulo administrado sea auto describible.
- Las instrucciones CIL generadas a partir del código fuente.

La meta data es además importante porque el CLR debe ser capaz de determinar que tipos están presentes en cada módulo administrado que es cargado. Pero esta es también importante para los compiladores y las herramientas que tratan con ejecutables administrados.

2.3.7 ENSAMBLADOS

Es la unidad primaria de una aplicación .Net, un ensamblado es una colección de código auto descriptible, recursos y meta data. El manifiesto del ensamblado contiene información acerca de que esta contenido dentro de este. El manifiesto provee:

- Información de identificación, tales como el nombre del ensamblado y el numero de versión
- Una lista de todos los tipos expuestos por el ensamblado
- Una lista de todos los ensamblados requeridos por el ensamblado
- Una lista de las instrucciones de seguridad de código de acceso, incluyendo permisos requeridos por el ensamblado y los permiso denegados al ensamblado

Cada ensamblado tiene un solo manifiesto, y este contiene toda la descripción de la información para el ensamblado. Sin embargo, el manifiesto del ensamblado puede ser contenido en su propio archivo o dentro de uno de los módulos del ensamblado.

2.3.8 LA BIBLIOTECA DE CLASES DEL .NET FRAMEWORK

Es una colección de tipos orientados a objetos e interfaces que proveen modelos de objetos y servicios para muchas de las tareas de programación complejas cuando se escriben aplicaciones. Muchos de los tipos presentados por la biblioteca de clases del .Net Framework son totalmente extensibles, permitiendo la construcción de tipos que incorporen funcionalidad propia dentro del código administrado. La biblioteca de clases esta organizada dentro de Espacios de Nombres (namespaces). Por ejemplo, el namespace System.Windows.Forms contiene los tipos que hacen formas Windows y los controles usados en esas formas. Los namespaces son agrupaciones lógicas de clases relacionadas y se encuentran organizados jerárquicamente. El namespace raíz del .Net Framework es el namespace System.

2.3.9 COMPILACIÓN Y EJECUCIÓN DE UNA APLICACIÓN .NET

Cuando se compila una aplicación .Net, esta no es compilada a código binario de máquina; en vez de ello, es convertido a MSIL. La forma en que se despliegan las aplicaciones es a partir de uno o más ensamblados consistentes de archivos ejecutables y archivos DLL en forma MSIL. Al menos uno de estos ensamblados contendrá un archivo ejecutable que ha sido diseñado como punto de entrada para la aplicación.

Cuando se inicia la ejecución de un programa, el primer ensamblado es cargado dentro de la memoria. En este punto, el CLR examina el manifiesto del ensamblado y determina los requerimientos para ejecutar el programa. Este examina los permisos de seguridad requeridos para el ensamblado y los compara con las políticas de seguridad del sistema. Si las políticas de seguridad del sistema no permiten los permisos requeridos, la aplicación no puede correr. Si la aplicación aprueba las políticas de seguridad del sistema, el CLR ejecuta la aplicación. Este crea un proceso para que la aplicación se corra dentro de este e inicia la ejecución de la aplicación. Cuando se inicia la ejecución, el primer bit de código que se necesita para ser ejecutada la aplicación es cargada dentro de la memoria y compilado de MSIL a código binario nativo por el compilador JIT del CLR. Una vez compilado, el código es ejecutado y almacenado en memoria como código nativo. De este modo, cada porción de código es compilada solo una vez cuando una aplicación es ejecutada. Dondequiera que la ejecución del programa dependa de código que aun no ha

sido ejecutado, el compilador JIT lo compila e inicia su ejecución y lo almacena en memoria como código nativo. De esta forma, el desempeño de la aplicación es maximizado debido a que solo las partes del programa que han sido ejecutadas son compiladas.

2.4 SMART CLIENTS

Una aplicación *Smart Client* (Cliente Inteligente) es más que solo una aplicación de formas Windows de .Net. Esta ciertamente usa recursos locales, CPUs locales, así que obtiene los beneficios de una rica experiencia de usuario, respuesta y desempeño que es posible ver en una aplicación Windows u Office. Pero además, las aplicaciones Smart Client encajan dentro de una arquitectura orientada a servicios o abierta, consumiendo servicios Web por si mismos, aunque no pueden ofrecerlos. El atributo clave de una aplicación Smart Client es su panorama desconectado. No se cuenta con esa cualidad en un navegador (browser), pues este requiere una conexión permanente a Internet. Los Smart Clients son aplicaciones que pueden trabajar, en un tren, en un avión o donde quiera que se desee. En el momento en que es posible la conectividad a la red de nuevo, los datos son automáticamente sincronizados. Otra de las ventajas con las que cuentan las aplicaciones Smart Client es su facilidad de despliegue.

2.4.1 DESCRIPCIÓN

Los *Smart Clients* [28] son un paso evolutivo sobre las aplicaciones cliente enriquecido y cliente Web, capturando los beneficios de ambos, y convirtiéndose así en parte de la siguiente generación de clientes software, los Smart Clients están definidos por los siguientes atributos (ver Figura 7):

- Toman total ventaja de los dispositivos de usuario final tales como: PCs, Tablet PCs, PDAs, teléfonos, entre otros.
- Brindan una experiencia de usuario de alta fidelidad
- Explotan la última tecnología en gráficas e interfaces de usuario
- Personalizan e individualizan los perfiles de cada usuario basado en el contexto
- Conexión Inteligente
- Soportan una rica experiencia de usuario tanto en línea o fuera de línea, tomando ventaja del procesamiento y almacenamiento de datos locales

- Distribución, capacidad para consumir servicios Web a través de los cuales sea posible acceder a datos almacenados en centrales corporativas y pre/post procesados en un servidor
- Manejan información centralizada
- Acceden a los datos de forma transparente
- Facilitan el retorno, almacenamiento y procesamiento datos
- Cuentan con un acceso a datos flexible y granular dependiendo de que, cuando y como los datos están siendo accedidos y afectados
- Están diseñadas para las operaciones del día a día
- Soportan un fácil despliegue que es típicamente asociado con las aplicaciones basadas en Web.
- Cuentan con niveles de seguridad
- Tienen capacidad de procesamiento local apalancado inteligentemente
- Son fácilmente versionables

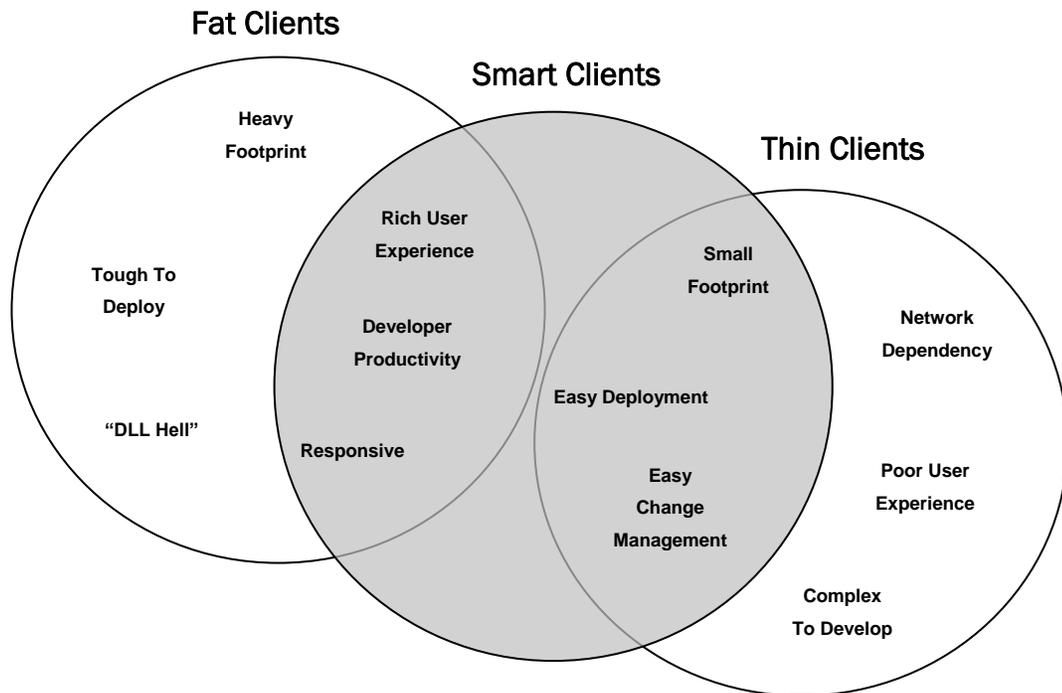


Figura 7 Comparación de Aproximaciones

2.4.2 CARACTERÍSTICAS IDEALES DE UN SMART CLIENT

- Rica experiencia de usuario
 - ◆ Presentar al usuario las mejores interfaces posibles teniendo en cuenta las capacidades del sistema operativo y del hardware subyacentes
 - ◆ Interactuar y controlar dispositivos especializados como impresoras, lectores de código de barras, lectores RFID, tablas de dibujo, teléfonos, etc.
- Gran modelo de despliegue
 - ◆ Actualizar en un determinado lugar y automáticamente cada vez que el usuario use la siguiente vez la aplicación (justo como en la Web)
 - ◆ Desplegar y actualizar de tal forma que no se creen conflictos en los dispositivos de los usuarios finales
 - ◆ Desconocer de parte de los usuarios finales algún aprendizaje especializado para trabajar con la aplicación
- Administración transparente de datos
 - ◆ Los usuarios deben poder utilizar la aplicación sin importar el estado actual de la conexión, quizás con una funcionalidad reducida en el momento de no tener una conexión presente
 - ◆ La sincronización de los datos debe suceder sin que el usuario tenga que explícitamente estar en línea o fuera de línea
- Seguridad
 - ◆ Contar con un modelo de seguridad que asegure la integridad de los datos

2.5 DESARROLLO DE APLICACIONES MOVILES CON MICROSOFT .NET

El poder de desarrollo de aplicaciones que la plataforma .Net ofrece también se extiende hacia las soluciones software para dispositivos móviles, ofreciendo entre otros, la posibilidad de implementar aplicativos a través del Microsoft .Net Compact Framework (.NET CF) (actualmente se encuentra en la versión 1.0 SP2), el cual ofrece la capacidad de mostrar, recopilar, procesar y enviar datos a un dispositivo móvil, datos que a su vez pueden ser locales, remotos o una combinación de ambos. .NET CF suministra al igual que el .Net Framework una máquina de ejecución con lenguaje común CLR y una BCL. .Net CF se integra al IDE del Microsoft Visual Studio .Net 2003 (VS.NET 2003) con el fin

de simplificar el desarrollo de aplicaciones para dispositivos inteligentes y de ofrecer mayor potencialidad.

2.5.1 EL MICROSOFT .NET COMPACT FRAMEWORK (.NET CF)

El .Net CF es una versión compacta del .Net Framework, el cual incluye un subconjunto de la BCL y unas nuevas clases que son específicamente diseñadas para el desarrollo con dispositivos móviles. Cuenta también con una nueva implementación del CLR (Compact CLR), teniendo en cuenta las restricciones de memoria, CPU y batería de los dispositivos, con el objetivo de trabajar más eficientemente. VS.NET 2003 incluye el software y las herramientas necesarias para crear aplicaciones que corran en dispositivos móviles donde el .Net CF este instalado. Las aplicaciones desarrolladas a partir del .Net CF corren en el dispositivo haciendo uso de los servicios del sistema operativo, del CLR y la BCL del .Net CF (ver Figura 8).

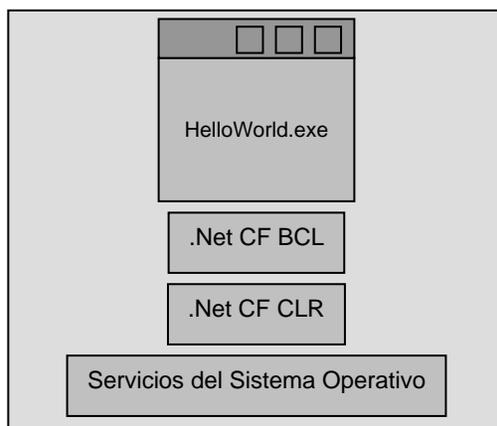


Figura 8 Ambiente de Ejecución de una aplicación con el .Net CF

El .Net CF está dirigido a las siguientes plataformas: Pocket PC 2000, Pocket PC 2002, Pocket PC 2002 Phone Edition, Pocket PC .Net, Windows CE .Net 4.1 y superiores.

Aunque el .Net CF brinde un gran soporte para el desarrollo de aplicaciones hay ciertas situaciones en las cuales los desarrolladores deben trabajar fuera de este:

- Tener acceso directo a la API de WIN32 con el fin de utilizar funcionalidad que no se provee a través de la librería de clases del .Net CF.
- Acceder a objetos COM.
- Interactuar con DLLs que proveen terceros para el manejo de periféricos.

Con el fin de obtener, enviar y sincronizar datos desde y hacia el dispositivo móvil, apoyando así la implementación de aplicaciones en escenarios desconectados el .Net CF ofrece la capacidad de trabajar como un cliente de Servicios Web XML.

Ambiente de ejecución del .NET CF: Cuando se desarrollan aplicaciones para dispositivos móviles usando el VS.Net 2003 y el .Net CF, es necesario de un paso adicional para poder desplegar o depurar la aplicación, pues esta no corre sobre el equipo de desarrollo. VS.NET 2003 crea el ejecutable compilado (ensamblado) y lo envía junto con las librerías necesarias hasta el dispositivo o emulador donde se ejecutará la aplicación; la cual correrá dentro de una instancia del CLR.

Common Lenguaje Runtime Compacto: El CLR compacto es la parte más importante del .Net CF, y al igual que el CLR del .Net Framework es el responsable de tomar los ensamblados que han sido compilados a MSIL, configurar los dominios de aplicación para correrlos dentro de este, usar el compilador JIT (Just In Time) para compilar los métodos a código nativo con el fin de ser ejecutados en el procesador huésped, administrar la memoria asignada, recolectar la basura, controlar la seguridad, y cargar las clases.

El CLR consiste actualmente de 2 partes distintas, el motor de ejecución, el cual es la interfaz con los servicios del sistema operativo subyacente, y la biblioteca de clases, las cuales son los bloques de construcción básica de una aplicación .Net.

La Figura 9 muestra las capas básicas del ambiente de ejecución del .Net CF. Las aplicaciones, las bibliotecas de clases específicas del dispositivo y la mayoría de clases base son todas compiladas a MSIL y corren bajo el control del CLR Compacto. El motor de ejecución es por si mismo un ejecutable nativo, al igual que la PAL, la cual es una capa de abstracción entre el motor de ejecución y los servicios del sistema operativo subyacente. El motor de ejecución y la PAL son empaquetados juntos dentro de un ejecutable llamado Mscoree.dll.

Hosts de dominio de aplicación: El cargador de la aplicación shell de Windows CE se actualiza en las plataformas donde el .Net CF se instala; así que cuando se inicia una aplicación, el cargador de la aplicación reconoce que ésta debe correr a través del .Net

CF. Este cargador de aplicación, el cual es código nativo, es un host de dominio de aplicación. El cual se encarga de cargar el CLR dentro de un proceso, crear el dominio de aplicación dentro del proceso, y cargar el código de la aplicación dentro del dominio de aplicación.

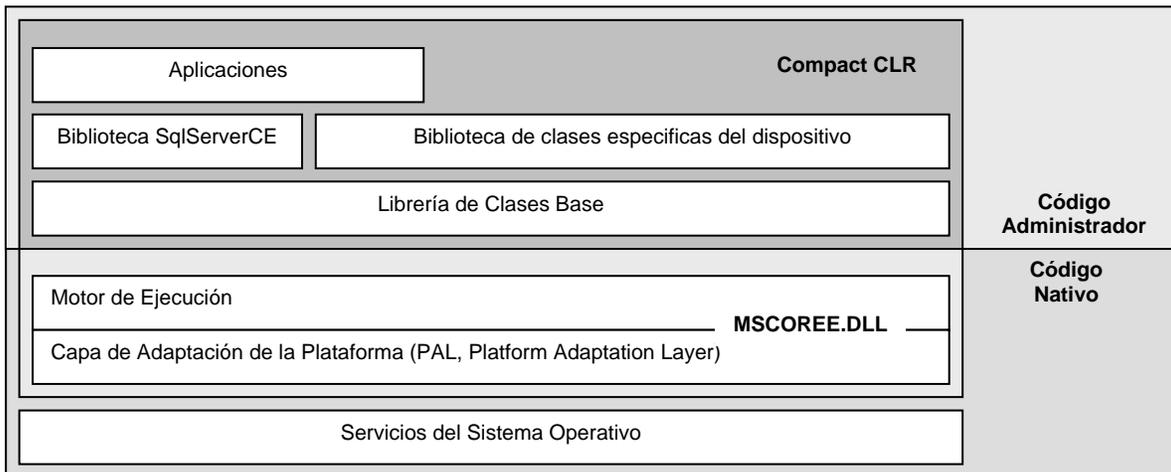


Figura 9 Estructura del Ambiente de Ejecución

Dominios de Aplicación: Un dominio de aplicación es una entidad análoga a un proceso regular del sistema operativo, excepto que el dominio de aplicación está completamente bajo el control del CLR, en vez de estar bajo el control del sistema operativo. Cada aplicación .Net se ejecuta dentro de un dominio de aplicación, y el CLR se asegura que todos los recursos usados por la aplicación durante su operación sean liberados cuando la aplicación finalice. Un único proceso del sistema operativo hospeda una instancia del CLR, el cual puede administrar múltiples dominios de aplicación. Cada dominio de aplicación ejecuta una única aplicación. Los dominios de aplicación ofrecen las siguientes ventajas para el aislamiento de aplicaciones en tiempo de ejecución comparado con la opción de usar procesos separados:

- El código administrado debe pasar a través de un proceso de verificación antes de que se le permita ejecutarse.
- Se puede iniciar una nueva aplicación en un nuevo dominio de aplicación, y es posible detener una aplicación en un dominio de aplicación sin iniciar o detener un proceso. Lo cual es mucho más barato que tener una aplicación ejecutándose en su propio proceso. Sin embargo, en la actual implementación del shell cargador de Windows CE se inicia un nuevo proceso y un nuevo CLR por cada aplicación, en vez de dar una

mirada para ver si existe un proceso corriendo el CLR, crear un nuevo dominio de aplicación dentro de este y corre la aplicación ahí. Por defecto entonces hay un dominio de aplicación por proceso.

- El CLR previene llamadas directas entre objetos en diferentes dominios de aplicación.

Delegados y eventos: Un evento es un mensaje que envía un objeto cuando ocurre una acción. Los eventos proporcionan un medio apropiado para que los objetos puedan señalar cambios de estado que pueden resultar útiles para los clientes de ese objeto.

Los eventos se declaran mediante delegados, así por ejemplo cuando se escribe código que se ejecuta cuando un botón es presionado, realmente se escribe código en un método manejador de evento en la clase de la forma Windows. La clase botón publica un evento *Click*, y este también define un método delegado, el cual define la firma del método de algún manejador de eventos para ese evento. Un manejador de evento se conforma de la definición de la firma en el delegado usando el mismo número y tipo de parámetros. Entonces la clase se suscribe al evento *Click* del botón de la clase, así que cuando esta clase botón enciende un evento *Click*, este llama el código de la clase del manejador de evento y el manejador de eventos de algún otro suscriptor. Un delegado es como una interfase, pero solo define la firma de un método único.

El ejemplo siguiente permite notificar cuando se incrementa el valor del campo miembro *_number* de la clase *Incrementor* a un cliente que se suscribe con el evento *Increment*.

```
using System;

namespace NETCFDevelopersReference
{
    public class Incrementor
    {
        private int _number = 0;

        public void AddOne( )
        {
            _number++;
            // Evento Increment no es nulo si hay algún suscriptor
            if(Increment != null)
            {
                // Creando nuevo objeto IncrementorEventArgs para enviar
                // a los suscriptores
                IncrementorEventArgs e = new IncrementorEventArgs(_number);
                Increment(this, e);
            }
        }
    }
}
```

```

    }

    // Declarando la firma requerida del manejador de eventos
    // de los métodos
    public delegate void IncrementEventHandler(
        object sender, IncrementorEventArgs e);

    // Declarando el evento y enlazándolo al delegado
    public event IncrementEventHandler Increment;
}

// Definiendo la clase que es pasada en el evento para informar al
// suscriptor acerca del evento, la cual hereda de la clase EventArgs
public class IncrementorEventArgs : EventArgs
{
    private int _value;

    public IncrementorEventArgs(int newValue)
    {
        _value = newValue;
    }

    public int Value
    {
        get
        {
            return _value;
        }
    }
}

// Declarando la clase de prueba
public class EventClientClass
{
    static void Main(string[] args)
    {
        EventClientClass thisclass = new EventClientClass( );
        thisclass.DoStuff( );
    }

    public void DoStuff( )
    {
        Incrementor myInc = new Incrementor( );
        // Suscribiéndose al evento
        myInc.Increment += new Incrementor.
            IncrementEventHandler(myIncrementEventHandler);
        myInc.AddOne( );
        myInc.AddOne( );
        myInc.AddOne( );
        Console.ReadLine( );
    }

    void myIncrementEventHandler

```

```

        (object sender, IncrementorEventArgs e)
    {
        Console.WriteLine("My event handler got called!" +
            " The value is {0}.", e.Value.ToString());
    }
}
}

```

Archivos de configuración de aplicaciones: En el .Net Framework, el namespace *System.Configuration* contiene clases que permiten a una aplicación traer constantes de un archivo de configuración XML. Lo que permite una más fácil configuración, ya que las constantes no quedan embebidas en el código sino en un archivo externo. Por ejemplo, un uso de esto podría ser definir la URL donde se alojan los Servicios Web.

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="URLWS" value="http://server/servicio.asmx"></add>
  </appSettings>
</configuration>

```

El inconveniente con el .Net CF es que este no soporta el acceso a archivos de configuración directamente a través de la clase del *System.Configuration.ConfigurationSettings.AppSettings*, sin embargo es posible hacer una implementación básica de una clase que permita hacer este acceso fácilmente. Como se muestra a continuación:

```

using System;
using System.Collections;
using System.IO;
using System.Reflection;
using System.Xml;

namespace Custom.Configuration
{
    /// <summary>
    /// Clase personalizada que simula la operación de la clase
    /// System.Configuration.ConfigurationSettings.AppSettings
    /// </summary>
    public class ConfigurationSettings
    {
        const string sectionTag = "appSettings";

        public static Hashtable AppSettings
        {
            get
            {
                Hashtable settings = new Hashtable(5);
            }
        }
    }
}

```

```

// Se obtiene la ruta para el archivo de configuración
// como una URI
string assemblyPath =
    Assembly.GetCallingAssembly().GetName().CodeBase;
string configFileUrl = assemblyPath + ".config";

// Se parsea el archivo de configuración
XmlDocument cfgDoc = new XmlDocument();
FileStream fs = new FileStream(configFileUrl,
    FileMode.Open, FileAccess.Read);
cfgDoc.Load(new XmlTextReader(fs));
fs.Close();

// Se Obtienen los nodos descendientes dentro de las secciones
// de las etiquetas del archivo de configuración XML
XmlNodeList nodes =
    cfgDoc.GetElementsByTagName(sectionTag);
foreach(XmlNode node in nodes)
{
    foreach(XmlNode childnode in node.ChildNodes)
    {
        XmlAttributeCollection attributes =
            childnode.Attributes;
        settings.Add(attributes["key"].Value,
            attributes["value"].Value);
    }
}
return settings;
}
}
}

// Se declara la clase de prueba
class TestAppSettings
{
    static void Main(string[] args)
    {
        Console.WriteLine("TestString: " +
            ConfigurationSettings.AppSettings["TestString"]);
        Console.ReadLine();
    }
}
}
}

```

La otra forma de lograr leer un archivo es a través del namespace `OpenNETCF.Configuration`, que hace parte del Open Net Compact Framework, haciendo uso del método estático `AppSettings` de la clase `ConfigurationSettings`.

```
OpenNETCF.Configuration.ConfigurationSettings.AppSettings["URLWS"];
```

Control InputPanel: Con el propósito de permitir al usuario ingresar información a través de los controles de captura de datos en un formulario Windows a través del dispositivo móvil, es necesario programar la funcionalidad del control InputPanel, el cual permite al usuario ingresar la información a través de un teclado en pantalla o un panel de reconocimiento de escritura manual, el cual aparece y se contrae dependiendo del lugar donde se encuentre el foco de entrada de texto, lo que se conoce como Soft Input Panel (SIP). El siguiente ejemplo muestra los pasos para mostrar y ocultar el control InputPanel cuando el control TextBox recibe o pierde el foco de entrada de texto.

```
/// <summary>
/// Automáticamente sube el SIP (Soft Input Panel)
/// </summary>
private void textBox_GotFocus( object sender, EventArgs e )
{
    inputPanel.Enabled = true;
}

/// <summary>
/// Automáticamente oculta el SIP
/// </summary>
private void textBox_LostFocus( object sender, EventArgs e )
{
    inputPanel.Enabled = false;
}
```

2.5.2 TRABAJANDO CON XML

El .Net CF provee soporte para trabajar con XML en dos formas: la posibilidad de leer y escribir XML usando las clases *XmlReader* y *XmlWriter*, y una implementación del Document Object Model (DOM) en la clase *XmlDocument*. Las dos posibilidades se encuentran en el namespace *System.Xml*. La clase *XmlDocument* construye una especie de modelo de árbol de un documento en memoria; se puede navegar por el documento en todas las direcciones y la modificar los datos XML. El costo de esta flexibilidad es que la clase *XmlDocument* requiere una substancial cantidad de memoria para grandes documentos debido a que esta lee el documento entero en memoria. Las clases *XmlReader* y *XmlWriter* proveen mucha más velocidad, más características de eficiencia de memoria, pero solo permiten navegar a través del documento hacia adelante.

Trabajando XML con la clase XMLWriter: La clase *XmlWriter* es una clase abstracta que provee métodos para salida de un archivo XML a un archivo o un flujo (stream) de forma rápida, solo hacia adelante, y sin manejo de cache. Con el propósito de realizar

estas tareas se puede usar la clase `XmlTextWriter` la cual deriva de `XMLWriter`, y así crear y persistir documentos XML. La clase `XmlTextWriter` provee un número de métodos que pueden ser usados para crear documento XML, con sus elementos, atributos, valores, etc. El siguiente ejemplo muestra la forma de crear un documento XML que posee dos elementos con sus respectivos valores.

```
XmlTextWriter scoWriter = new XmlTextWriter( fileURL,
                                             System.Text.Encoding.ASCII);
scoWriter.WriteStartDocument( );
scoWriter.WriteStartElement("SCOs");
scoWriter.WriteStartElement("SCO");
scoWriter.WriteElementString( "SCONOMBRE", nodo.Text );
scoWriter.WriteElementString( "SCOFECHA",
                               DateTime.Today.ToShortDateString( ) );
scoWriter.WriteEndElement( );
scoWriter.WriteEndElement( );
scoWriter.WriteEndDocument( );
scoWriter.Flush( );
scoWriter.Close( );
```

Leyendo XML con la clase `XmlDocument`: El primer paso para leer un documento XML con la clase `XmlDocument` es cargar el XML. Para lo cual se puede usar el método `Load` de la clase `XmlDocument`. Para cargar un documento XML de un objeto `Stream`, `TextReader`, o `XmlReader` se puede usar el método `LoadXML` cargando todo en un `string`. Por ejemplo es posible cargar un documento XML de las siguientes formas:

```
---
XmlDocument xmlDoc = new XmlDocument( );
// Cargando el documento desde un objeto Stream
// (FileStream deriva de Stream )
xmlDoc.Load( new FileStream( "archivo.xml", FileMode.Open ) );
---
XmlDocument xmlDoc = new XmlDocument( );
// Cargando el documento desde un objeto TextReader
// ( StreamReader deriva de TextReader )
xmlDoc.Load( new StreamReader( "archivo.xml" );
---
XmlDocument xmlDoc = new XmlDocument( );
// Cargando el documento desde un objeto XmlReader
// ( XmlTextReader deriva de XmlReader )
xmlDoc.Load( new XmlTextReader ( "archivo.xml" );
---
XmlDocument xmlDoc = new XmlDocument( );
// Cargando el documento desde un string que contiene el XML
xmlDoc.LoadXML( xmlString );
```

Navegando un documento XML: Una vez el documento ha sido cargado es posible navegar a través de sus nodos. Para retornar el elemento raíz del documento, es posible

hacerlo simplemente a través de la propiedad `DocumentElement` de la clase `XmlDocument`, de la siguiente forma:

```
XmlNode elementoRaiz = xmlDoc.DocumentElement;
```

Todos los nodos en un árbol de documentos XML son extensiones de la clase abstracta `XmlNode`. Es posible a través de una variable de tipo `XmlNode` navegar a través de todos los nodos en un documento. La clase `XmlNode` tiene adicionalmente métodos y propiedades para soportar navegación en todas las direcciones, comprobar la existencia de un nodo, obtener la lista de nodos, conocer el padre de un nodo específico, obtener el primer y último nodo secundario de un nodo, acceder valores de los elementos, acceder atributos, modificar y guardar el documento XML, entre otros. El ejemplo siguiente muestra algunos de estos métodos y propiedades:

```
// Fijando el camino (path) del documento XML de los SCOs descargados
string sFileURL = this.strPath + @"\"SCOsDownload";

// Cargando el Documento XML
XmlDocument xmlDoc = new XmlDocument( );
StreamReader xmlFile = new StreamReader( sFileURL );
xmlDoc.Load( xmlFile );
xmlFile.Close( );

// Obteniendo el elemento raíz del documento
XmlNode SCOs = xmlDoc.DocumentElement;

// Recorriendo los nodos hijos del elemento raíz del documento
foreach(XmlNode scoNodo in SCOs.ChildNodes)
{
    // Buscando el SCO a eliminar
    XmlNode scoNombre = scoNodo.FirstChild;
    if(scoNombre.FirstChild.Value == SCONombre)
    {
        // Removiendo el SCO del documento
        SCOs.RemoveChild(scoNodo);
    }
}
xmlDoc.Save( sFileURL );
```

2.5.3 MANEJO DE ENTRADA y SALIDA

El .Net CF provee poderosas características para leer y escribir todo tipo de archivos, desde simples archivos de texto, hasta archivos binarios puros que deben ser leídos como bytes. Con características que permiten leer/escribir punteros a través de un archivo y para copiar, mover y borrar archivos enteros.

El .Net CF soporta operaciones de entrada/salida en dos niveles: acceso de byte de bajo nivel usando objetos *Stream*, y acceso de caracteres usando objetos *StreamReader* y *StreamWriter*. La clase abstracta *Stream* provee funcionalidad de bajo nivel para leer y escribir a alguna fuente de datos de byte. Esta fuente de datos puede ser un archivo local, un archivo remoto o algún objeto que se pueda acceder de manera similar a un archivo aunque no lo sea. El hecho más importante concerniente al uso de un stream es que lee y escribe en formato de bytes. La clase abstracta *Stream* es la clase base para las clases *FileStream*, *MemoryStream* y *NetworkStream*.

Para crear un objeto del tipo *FileStream* es posible hacerlo a través del método *File.Open* o a través de su constructor, el segundo tiene varias sobrecargas, pero en la más básica de estas recibe mínimo dos parámetros, el primero indica el nombre del archivo y el segundo especifica un valor de la enumeración *FileMode*. El propósito del parámetro *FileMode* es indicar si el archivo va a ser creado o abierto, y que acción se realizará sobre el archivo. El siguiente ejemplo muestra la forma de crear un nuevo archivo:

```
FileStream flujo = new FileStream( @"\\myFile.txt", FileMode.Create );
```

FileStream posee además los métodos *ReadByte* y *WriteByte* para leer y escribir bytes en el archivo. También es posible escribir y leer tipos de datos primitivos tales como enteros o dobles directamente hacia y desde streams sin necesidad de realizar conversiones para o desde caracteres, a través de las clases *BinaryReader* y *BinaryWriter*. El siguiente ejemplo muestra la creación de un nuevo archivo y la escritura sobre este de un string, un entero y un double.

```
String fileName = myFileDialog.FileName;
FileStream myFileStream = new FileStream( fileName,
                                         FileMode.OpenOrCreate,
                                         FileAccess.Write,
                                         FileShare.None );
BinaryWriter writer = new BinaryWriter( myFileStream, Encoding.ASCII );
writer.Write( "Order" );
writer.Write( 42 );
writer.Write( 3.47 );
writer.Close( );
```

2.5.4 OBJETOS DE DATOS - ADO.NET

En el .NET Framework, Microsoft introdujo ADO.NET, una evolución de la arquitectura de acceso a datos provista por el modelo de programación ADO (ActiveX® Data Objects).

ADO.NET no reemplaza a ADO, es un conjunto de clases que exponen servicios de acceso a datos al programador de .NET, además proporciona un conjunto variado de componentes para crear aplicaciones distribuidas con uso compartido de datos, proporciona acceso a datos relacionales, datos XML y datos de aplicaciones. Es compatible con diversas necesidades de programación, incluida la creación de clientes de bases de datos y objetos empresariales de nivel medio utilizados por aplicaciones, herramientas, lenguajes o exploradores de Internet.

ADO.NET separa claramente el acceso a los datos de la manipulación de los mismos y crea componentes que se pueden usar por separado o conjuntamente. ADO.NET incluye proveedores de datos para conectarse a una base de datos, ejecutar comandos y recuperar resultados. Esos resultados se procesan directamente o se colocan en un objeto DataSet de ADO.NET con el fin de exponerlos al usuario para un propósito específico, junto con datos de varios orígenes, o de utilizarlos de forma remota entre niveles. El objeto DataSet de ADO.NET también puede utilizarse independientemente de un proveedor de datos para administrar datos que son locales de la aplicación o que proceden de un origen XML.

Componentes de ADO.NET: Los componentes de ADO.NET están diseñados para separar el acceso a datos de la manipulación de datos. ADO.NET tiene dos componentes principales que cumplen esta función: el DataSet y el proveedor de datos, que es un conjunto de componentes entre los que se incluyen los objetos Connection, Command, DataReader y DataAdapter. El DataSet de ADO.NET es el componente central de la arquitectura sin conexión de ADO.NET. El DataSet está expresamente diseñado para el acceso a datos independientemente del origen de datos. Como resultado, se puede utilizar con múltiples y distintos orígenes de datos, con datos XML o para administrar datos locales de la aplicación, además contiene una colección de uno o más objetos DataTable formados por filas y columnas de datos, así como información sobre claves principales, claves externas, restricciones y relaciones relativas a los datos incluidos en los objetos DataTable.

El otro elemento central de la arquitectura de ADO.NET es el proveedor de datos, cuyos componentes están diseñados expresamente para la manipulación de datos y para el

acceso rápido a datos de sólo lectura. El objeto Connection proporciona conectividad con un origen de datos. El objeto 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. El objeto DataReader proporciona una secuencia de datos de alto rendimiento desde el origen de datos. Por último, el objeto 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.

Es posible escribir proveedores de datos para cualquier origen de datos, por ejemplo Oracle, Interbase y otros, que han sido escritos por los propios fabricantes de los motores. .NET Framework incluye dos proveedores de datos por defecto: el proveedor de datos para OLE DB y el proveedor de datos para SQL Server. En la Figura 10 se ilustran los componentes de la arquitectura de ADO.NET.

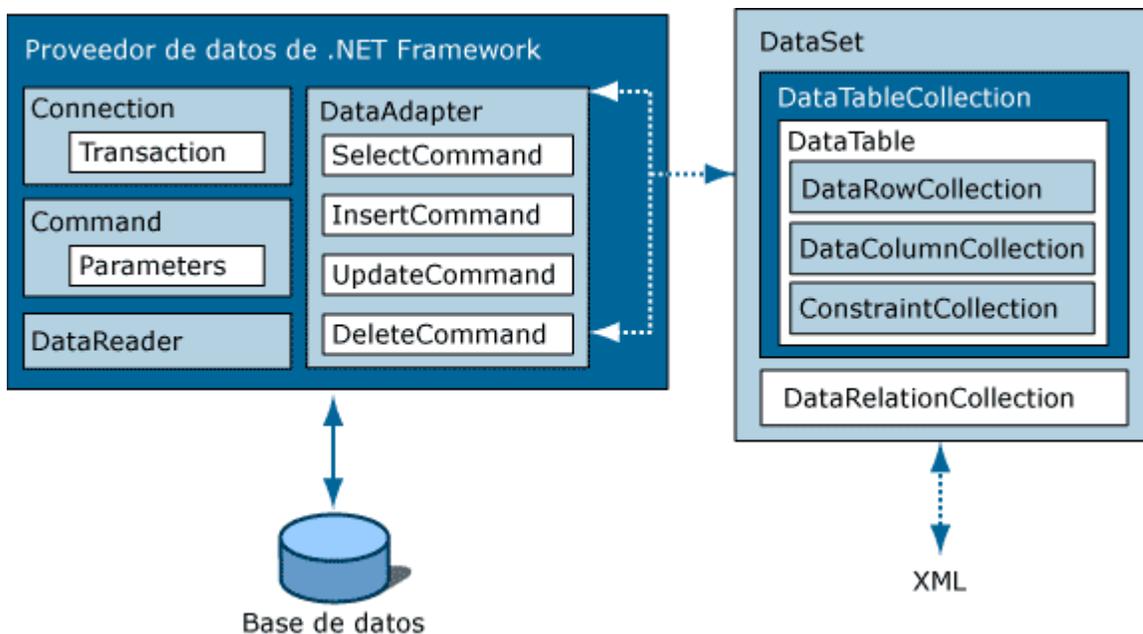


Figura 10 Arquitectura de ADO.NET [42]

Ventajas de ADO.NET: ADO.NET ofrece varias ventajas sobre las anteriores versiones de ADO y sobre otros componentes de acceso a datos, como las siguientes:

- **Interoperabilidad:** Las aplicaciones ADO.NET pueden aprovechar la flexibilidad y la amplia aceptación de XML. Dado que XML es el formato de transmisión de conjuntos de datos a través de la red, cualquier componente que pueda leer el formato XML podrá procesar los datos. En realidad, no es necesario en absoluto que el componente receptor sea un componente ADO.NET. El componente transmisor puede transmitir simplemente el conjunto de datos a su destino, independientemente de cómo esté implementado el componente receptor. El único requisito es que el componente receptor pueda leer XML.
- **Mantenibilidad:** A lo largo de la vida de un sistema es posible hacer cambios modestos, pero raramente se intenta hacer cambios importantes, estructurales, debido a su dificultad. ADO.NET mediante los conjuntos de datos permite que este proceso se haga de una manera más sencilla y sin daños a la funcionalidad de la aplicación. Si es necesario reemplazar un solo nivel por dos, sólo hay que asegurarse de que ellos intercambien información y dado que los niveles pueden transmitir datos por medio de conjuntos de datos con formato XML, la comunicación es relativamente fácil
- **Programabilidad:** Los componentes de datos ADO.NET encapsulan funcionalidad de acceso a datos de diversas formas que ayudan a programar de modo más rápido y con menos errores. Por ejemplo, los comandos de datos condensan la tarea de generar y ejecutar instrucciones SQL o procedimientos almacenados. Del mismo modo, las clases de datos de ADO.NET generadas por las herramientas dan como resultado conjuntos de datos con tipo, lo que permite el acceso a los datos mediante programación con tipo. Por ejemplo, en la siguiente línea de código, se obtiene acceso a un miembro de datos en un conjunto de datos sin tipo:

```
if (TotalCost > DataSet1.Tables["Customers"].Rows[i]["AvailableCredit"]) { }
```

La línea equivalente, con acceso a un miembro de datos de un conjunto de datos con tipo, tiene el siguiente aspecto:

```
if (TotalCost > DataSet1.Customers("Jones").AvailableCredit) { }
```

El código correspondiente al conjunto de datos con tipo es más fácil de leer. También es más fácil de escribir, porque se proporciona la finalización de instrucciones. Por último, el código correspondiente al conjunto de datos con tipo es más seguro, porque proporciona la comprobación de tipos en tiempo de compilación

- **Rendimiento:** Para las aplicaciones desconectadas, los conjuntos de datos ADO.NET ofrecen ventajas de rendimiento frente a los conjuntos de registros ADO desconectados. Cuando se utiliza el cálculo de referencias de COM para transmitir un conjunto de registros desconectado entre niveles, la conversión de los valores del conjunto de registros a tipos de datos reconocibles por COM puede suponer un costo de procesamiento significativo. En ADO.NET, tal conversión de tipos de datos no es necesaria.

XML y ADO.NET: ADO.NET aprovecha la eficacia de XML para proporcionar acceso a datos sin mantener una conexión abierta. ADO.NET fue diseñado teniendo en cuenta las clases de XML incluidas en .NET Framework: ambos son componentes de una única arquitectura. ADO.NET y las clases de XML incluidas en .NET Framework convergen en el objeto DataSet. Es posible llenar un DataSet a partir de una secuencia o un documento XML, Se puede utilizar la secuencia o el documento XML para suministrar datos al DataSet, información de esquema o ambas cosas. La información suministrada desde la secuencia o el documento XML puede combinarse con datos o información de esquema existente ya presentes en el DataSet.

ADO.NET también permite crear una representación XML de un DataSet compatible con el Consorcio World Wide Web (W3C), para transportar el DataSet a través de HTTP con el fin de que lo utilice otra aplicación u otra plataforma compatible con XML. En una representación XML de un DataSet, los datos se escriben en XML y el esquema, si está incluido en línea en la representación, se escribe utilizando el lenguaje de definición de esquemas XML (XSD). XML y el esquema XML proporcionan un formato cómodo para transferir el contenido de un DataSet a y desde clientes remotos

Cargar DataSet desde XML: Para llenar un DataSet con datos desde XML, hay que utilizar el método ReadXml del objeto DataSet. El método ReadXml lee desde un archivo, una secuencia o un XmlReader y toma como argumentos el origen de XML y un argumento XmlReadMode opcional. El método ReadXml lee el contenido de la secuencia o el documento XML y carga datos en el DataSet. También crea el esquema relacional del DataSet dependiendo del XmlReadMode especificado y de si ya existe o no un esquema

relacional. El siguiente ejemplo muestra el formato de un archivo XML sencillo ubicado en "C:\Program Files\easylearning\smartproxy\usuario.xml" que se cargará en un DataSet:

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <user>
    <Cuenta>PepitoPerez</Cuenta>
    <Clave>pperez</Clave>
    <tipoUsuario>ESTUDIANTE</tipoUsuario>
  </user>
</NewDataSet>

//Creando el DataSet
DataSet objDs = new DataSet();

//Cargando el archivo
objDs..ReadXml(@"C:\Program Files\easylearning\smartproxy \usuario.xml");

//Accediendo al contenido
strTipoUsuario = objDs.Tables["user"].Rows[0][" tipoUsuario"].toString();
```

Escribir un objeto DataSet como XML: En ADO.NET es posible escribir una representación XML de un DataSet, con o sin su esquema. Si la información de esquema está incluida en línea con el código XML, se escribirá con el lenguaje de definición de esquemas XML (XSD). El esquema contiene las definiciones de tabla del DataSet, así como las definiciones de relaciones y restricciones. Cuando un DataSet se escribe como datos XML, las filas del DataSet se escriben en sus versiones actuales. Sin embargo, el DataSet también se puede escribir como un DiffGram⁵, de forma que se incluyan los valores actuales y originales de las filas. La representación XML del DataSet se puede escribir en un archivo, una secuencia, un XmlWriter o una cadena. Estas opciones ofrecen una gran flexibilidad en cuanto a la forma de transportar la representación XML del DataSet. En el siguiente ejemplo se muestra como guardar el contenido de un DataSet en un archivo XML:

```
//Creando el DataSet
DataSet objDS = new DataSet();
DataTable objTable = new DataTable("tblUser");
objTable.Columns.Add("Nombre", typeof(string));
objTable.Columns.Add("Apellido", typeof(string));
objTable.Columns.Add("Cedula", typeof(string));

objDS.Tables.Add(objTable);
```

⁵ Un DiffGram es un formato XML que se utiliza para identificar las versiones actual y original de los elementos de datos. El DataSet utiliza el formato DiffGram para cargar y hacer persistente su contenido, así como para serializar su contenido con el fin de transportarlo a través de una conexión de red.

```

// Creando una fila (DataRow) del tipo tblCliente que contendrá una tupla
// con el Nombre, Apellido y cedula del usuario

DataRow objDr;
objDr = Dt.NewRow( );
objDr["Nombre"] = "Pepito";
objDr["Apellido"] = "Perez";
objDr["Cedula"] = "12345678";

// Adicionando la Fila a la tabla
objTable.Rows.Add(objDr);

//Grabando el XML
objDS.WriteXml(@"C:\ProgramFiles\easylearning\smartproxy\usuario\Cliente.
xml");

//El archivo que se obtiene es el siguiente:
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  < tblUser >
    <Nombre>Pepito</ Nombre >
    <Apellido>Perez</ Apellido >
    < Cedula >12345678</ Cedula >
  </ tblUser >
</NewDataSet>

```

2.5.5 SERVICIOS WEB XML

A pesar de la aparente sencillez del XML, éste está transformando la creación y el uso del software. Así como el Web revolucionó la comunicación entre usuarios y aplicaciones, XML revoluciona día a día la comunicación entre aplicaciones, o de forma más general, la comunicación entre equipos, ya que ofrece un formato de datos universal que permite adaptar o transformar fácilmente la información.

Los servicios Web XML son unidades de código discretas, cada una de las cuales se encarga de un conjunto limitado de tareas, permiten que las aplicaciones compartan información, están basados en el lenguaje universal de intercambio de información en Internet, XML; pueden utilizarse en cualquier plataforma o sistema operativo, independientemente del lenguaje de programación y .NET es la plataforma de Microsoft utilizada para crearlos.

Un servicio Web es lógica de aplicación programable que se expone mediante protocolos estándar de Internet, tales como HTTP, SOAP, XML y UDDI. Una parte de la especificación de SOAP define un conjunto de reglas para utilizar XML al representar

datos, otras, definen un formato ampliable de mensajes o convenciones para representar llamadas a procedimientos remotos (RPC) mediante el formato de mensajes SOAP y enlaces al protocolo HTTP. UDDI especifica un servicio que permite a los proveedores de servicios Web anunciar o publicar la existencia de sus servicios Web y a los consumidores les facilita el proceso de encontrar servicios Web de su interés.

Dado que un servicio Web no está asociado a una tecnología de componentes, a una convención de llamada a objetos concretas y a que representan funcionalidad de caja negra que se puede reutilizar sin conocer los detalles de la implementación; permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada. Dada la definición de los Servicios Web, en la Figura 11 se puede ver la arquitectura de una aplicación que hace uso de ellos. En este caso la aplicación cliente pueda crear y consumir los mensajes apropiados y para ello no tiene que conocer los detalles de la estructura interna de los servicios Web que utiliza.

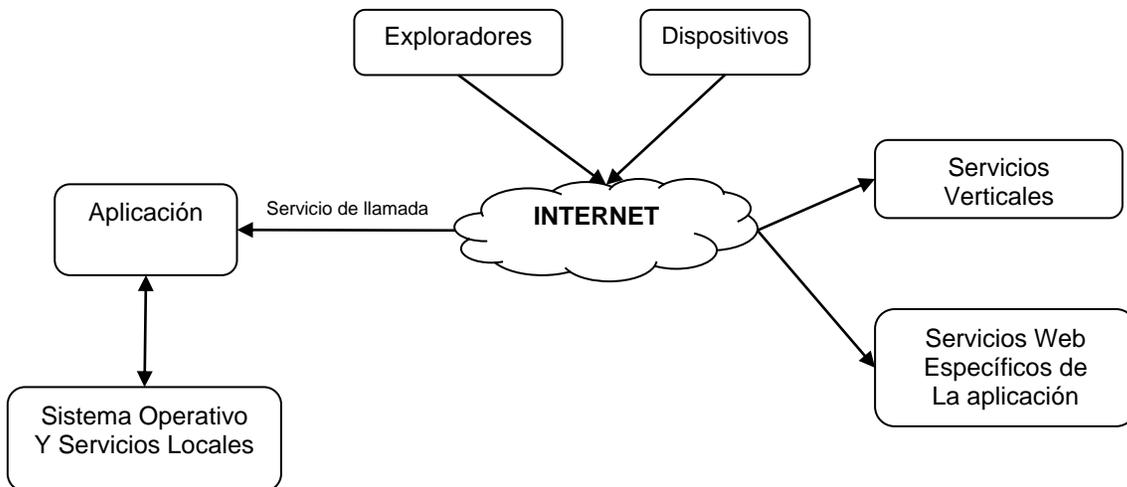


Figura 11 Arquitectura de una aplicación .NET utilizando Servicios Web

Modelo de código para servicios Web XML en código administrado: Los servicios Web XML se componen de dos partes: el punto de entrada al servicio Web XML y el

código que implementa la funcionalidad del servicio Web XML. En ASP.NET, el archivo .asmx es un archivo de texto que sirve de punto de entrada direccionable del servicio Web XML. Hace referencia al código de ensamblados precompilados, a un archivo de código subyacente o a un código contenido en el propio archivo .asmx. La directiva de procesamiento WebService situada en la parte superior del archivo .asmx determina dónde se buscará la implementación del servicio Web XML. De forma predeterminada, VS.NET utiliza archivos de código en segundo plano, como Service1.asmx.vb o Service1.asmx.cs, cuando se crea un servicio Web XML con la plantilla de proyecto de servicio Web ASP.NET. Cuando se crea un servicio Web XML en código administrado, ASP.NET automáticamente proporciona la infraestructura y controla el procesamiento de solicitudes y respuestas del servicio Web XML, incluido el análisis y la creación de mensajes SOAP.

Directiva de procesamiento WebService: La directiva de procesamiento WebService, situada en la parte superior de la página .asmx, contiene información en forma de atributos relacionados con la implementación del servicio Web XML. Esta directiva proporciona información necesaria al entorno ASP.NET, así como la clase que implementará la funcionalidad del servicio Web XML. A continuación se muestra un ejemplo de la directiva de procesamiento WebService en el lenguaje C#:

```
<%@ WebService Language="c#" Codebehind="Service1.asmx.cs"  
    Class="WebService1.Service1" %>
```

El atributo Language indica el lenguaje de programación utilizado para desarrollar el servicio Web XML. Al crear servicios Web XML en código administrado con VS.NET, la implementación del servicio Web XML reside en un archivo de código en segundo plano. El archivo de código subyacente se asocia a la página .asmx mediante el atributo Codebehind. Este atributo ayuda a VS.NET a administrar el proyecto de servicio Web XML. Cuando se utiliza la plantilla de proyecto de servicio Web ASP.NET, el atributo Class indica qué clase del archivo de código en segundo plano implementará la funcionalidad del servicio Web XML.

Clase System.Web.Services.WebService: Esta clase define la clase base de los servicios Web XML, proporciona acceso directo a los objetos ASP.NET comunes, como

los destinados a estados de aplicación y de sesión. De forma predeterminada, los servicios Web XML creados en código administrado con Visual Studio se heredan de esta clase. El servicio Web XML se puede heredar de esta clase para obtener acceso a objetos intrínsecos de ASP.NET, como Request y Session.

Si el servicio Web XML no se hereda de esta clase, puede tener acceso a los objetos intrínsecos de ASP.NET desde System.Web.HttpContext.Current. La clase que implementa el servicio Web XML debe ser pública y debe tener un constructor predeterminado público (sin parámetros). De este modo ASP.NET podrá crear una instancia de la clase del servicio Web XML para procesar las solicitudes de servicio Web XML entrantes. A continuación se muestra un ejemplo de la declaración de la clase de un Servicio Web usando la clase base:

```
using System.Web.Services;
public class serBasedeDatos: System.Web.Services.WebService
{
    // Código de Implementación.
}
```

Atributo WebService: Cada servicio Web XML requiere un espacio de nombres único que permita a las aplicaciones de cliente diferenciar entre los servicios Web XML que utilicen el mismo nombre de método. El espacio de nombres (namespace) predeterminado de los servicios Web XML creados en VS.NET es "http://tempuri.org/WebService1/Service1", donde WebService1 es el nombre del proyecto y Service1 es el nombre de la clase. Aunque el espacio de nombres se asemeja a una dirección URL normal, no es de esperar que se pueda ver en un explorador Web, ya que se trata simplemente de un identificador único. A continuación se muestra un ejemplo de cómo utilizar el atributo WebService, en dónde no se utiliza el valor por defecto del campo namespace.

```
[WebService(Namespace="tgh://mlearning01/wsM_LMS/LogicaDeServicios/",
Description="Web Service de Acceso a una Base de Datos Oracle")]
public class serBasedeDatos : System.Web.Services.WebService
{
    // Código de implementación
}
```

Atributo WebMethod: Para exponer un método como parte de un servicio Web XML, hay que anteponer el atributo WebMethod a la declaración de cada método público que se vaya a exponer, de la siguiente manera:

```
[WebMethod(true)]
public bool AbrirConexion(string Cuenta, string Clave)
{
    // Código de implementación
}
```

El atributo WebMethod contiene varias propiedades para configurar el comportamiento del servicio Web XML. Por ejemplo, en el anterior código se muestra el valor true, con lo cual se establece que el método Web manejará variables de sesión, en caso contrario cualquier variable de sesión que se utilice no mantendrá el valor, es decir que su contenido será null.

Archivo de configuración Web.config: La configuración de los servicios Web XML sigue el mismo paradigma que utilizan todas las aplicaciones Web de ASP.NET. La configuración de ASP.NET se basa en una arquitectura eficaz y extensible de configuración de archivos de texto basados en XML. Un archivo de configuración simplemente es un conjunto de elementos XML que representan las opciones de configuración de una determinada característica técnica de Microsoft .NET Framework. En el caso de los servicios Web XML, las opciones de configuración se encapsulan en el elemento XML webServices de un archivo de configuración.

La información de configuración de los recursos de ASP.NET se encuentra en un archivo de configuración denominado Web.config. Todos los archivos de configuración contienen una jerarquía anidada de etiquetas y subetiquetas XML con atributos que especifican las opciones de configuración. Debido a su formato XML, en las etiquetas, subetiquetas y atributos se distingue entre mayúsculas y minúsculas. En los nombres de etiquetas y atributos se utiliza la siguiente combinación de mayúsculas y minúsculas: la primera letra de la etiqueta es minúscula y la primera letra de cada palabra concatenada a continuación para formar el nombre es mayúscula. En los valores de los atributos se utiliza la combinación de mayúsculas y minúsculas de Pascal; es decir, la primera letra del valor es mayúscula y también la primera de cada palabra concatenada a continuación. Los valores true y false, que siempre se escriben en minúsculas, son una excepción.

Toda la información de configuración reside entre las etiquetas XML raíz <configuration> y </configuration>. Cada sección de configuración contiene subetiquetas con atributos que contienen las opciones de esa sección. A menudo, las opciones de configuración se anidan bajo etiquetas de agrupamiento de secciones. Normalmente, estas etiquetas de sección de nivel superior representan el espacio de nombres al que se aplican las opciones de configuración. La etiqueta <system.web> representa las opciones de las clases de ASP.NET. En la infraestructura de configuración de ASP.NET no se atribuye ningún tipo de datos a las opciones de configuración. En el siguiente código se muestra un ejemplo del archivo de configuración:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<system.web>

    <!-- Etiquetas que permiten establecer el comportamiento de la
        Depuración Dinámica De Compilación-->
    <compilation
        defaultLanguage="c#"
        debug="true"
    />

    <!-- Etiquetas que permiten personalizar los mensajes de error -->
    <customErrors
        mode="RemoteOnly"
    />

    <!-- Esta sección establece las directivas de autenticación de la
        aplicación. Los modos existentes son "Windows", "Forms",
        "Passport" y "None" -->
    <authentication mode="Windows" />

    <!-- Esta sección establece las directivas de autorización de la
        aplicación. Puede permitir o denegar el acceso a los recursos
        de la aplicación por usuario o función. Comodines: "*"
        incluye a todo el mundo, "?" incluye a usuarios anónimos (sin
        autenticar) -->
    <authorization>
        <allow users="*" /> <!-- Permitir a todos los usuarios -->
        <!-- <allow users="[lista de usuarios separados por comas]"
            roles="[lista de funciones separadas por comas]"/>
            <deny users="[lista de usuarios separados por comas]"
                roles="[ lista de funciones separadas por comas]"/>-->
    </authorization>

    <!-- Configuración de Estado de Sesión: Las cookies permiten
        identificar qué solicitudes pertenecen a una sesión
        determinada. Si éstas no están disponibles, se puede realizar
        el seguimiento de una sesión agregando un identificador de
        sesión a la dirección URL-->
    <sessionState
```

```
mode="InProc"  
stateConnectionString="tcpip=127.0.0.1:42424"  
cookieless="true"  
timeout="20"  
  
/>  
</system.web>  
</configuration>
```

Modelo de código para obtener acceso a servicios Web XML en código

administrado: Un cliente de servicios Web XML es cualquier componente, servicio o aplicación que haga referencia y utilice un servicio Web XML. Es importante destacar que los clientes de los servicios Web XML no tienen que ser obligatoriamente una aplicación de cliente, como una aplicación de escritorio; la mayoría de las veces los clientes de servicios Web XML pueden ser otras aplicaciones Web o incluso otros servicios Web XML. El uso de un servicio Web XML abarca la comunicación de métodos de servicio Web XML a través de una red mediante protocolos estándar. Sin embargo, para que una aplicación pueda comenzar a comunicarse con métodos de servicio Web XML, primero se deben llevar a cabo cuatro pasos básicos:

- Determinar la existencia de un servicio Web XML y obtener la URL del servicio.
- Descubrir el servicio Web XML. Dada la dirección URL de un documento de descubrimiento que reside en un servidor Web, el programador de una aplicación cliente puede conocer la existencia de un servicio Web XML, sus capacidades y cómo interactuar correctamente con él. Este proceso se conoce como descubrimiento de servicios Web XML.
- Mediante el proceso de descubrimiento de servicios Web XML, se descarga un conjunto de archivos al equipo local, que contienen información acerca de la existencia de servicios Web XML. La información consiste en un documento XML que detalla el servicio Web XML en Lenguaje de descripción de servicios Web (WSDL).
- Dada una descripción de servicio, se debe generar una clase de proxy (Referencia a Web Service) que pueda comunicarse con los métodos de servicio Web XML en función de la definición precisa que contiene la descripción de servicio. Esta clase controla el trabajo de asignar parámetros a elementos XML y, después, envía el mensaje SOAP a través de la red. De forma predeterminada, la clase de proxy utiliza SOAP sobre HTTP para comunicarse con el servicio Web XML

- Crear una aplicación cliente que llame a métodos de la clase de proxy. Los métodos de la clase de proxy se pueden comunicar con los métodos del servicio Web XML a través de Internet mediante protocolos estándar.

El siguiente comando crea un archivo .wsdl para el servicio Web XML ubicado en la dirección URL especificada y una clase proxy cliente en lenguaje C# para el servicio Web XML:

```
wsdl http://hostServer/WebserviceRoot/WebServiceName.asmx?WSDL
```

El siguiente comando crea una clase proxy cliente en lenguaje C# para un servicio Web XML ubicado en la dirección URL especificada. La herramienta guarda la clase proxy cliente en el archivo myProxyClass.cs.

```
wsdl /out:myProxyClass.cs
http://hostServer/WebserviceRoot/WebServiceName.asmx?WSDL
```

El siguiente código instancia un servicio Web y consume (usa) uno de sus métodos:

```
serBasedeDatos objSerBd = new serBasedeDatos();
....
DataSet objDs = new DataSet();
objDs = objSerBd.LlenarDataSet("Select * from tblUsuarios",
                             "tblUsuarios");
```

Manteniendo información de estado: HTTP es un protocolo sin estados. Cada solicitud se atiende a medida que se recibe; una vez procesada la solicitud se descartan todos sus datos. No se mantiene ningún estado entre varias solicitudes, aunque sean del mismo cliente. ASP.NET permite mantener el estado de las aplicaciones y de las sesiones mediante el uso de variables de aplicación y de sesión, respectivamente.

Estado de la aplicación: Las variables de estado de la aplicación son variables globales para cada aplicación de ASP.NET. Estas variables suelen configurarse en el evento Application_OnStart y, después, se tiene acceso a las mismas y se modifican en páginas individuales de ASP.NET. La duración de las variables de aplicación es la misma que la de la aplicación ASP.NET, es decir, hasta que se baje (detenga) la aplicación.

Estado de la sesión: Se puede almacenar valores que tengan que mantenerse durante la sesión de un usuario en las denominadas variables de sesión. Estas variables son

únicas de cada sesión de usuario y se puede tener acceso a ellas en cualquier página ASP.NET de una aplicación. Puede configurar y tener acceso a información de sesión dentro de una aplicación ASP.NET. Por ejemplo:

```
//Asignar un valor a la variable de sesión miVariable.  
Context.Session["miVariable "] = algunValor;  
  
//Recuperar el valor de la variable de sesión miVariable.  
string myString;  
if (Context.Session["miVariable "] != null)  
    myString = (string) Context.Session["miVariable "];
```

Es importante recordar que las variables de sesión son objetos. Así, para evitar un error en tiempo de ejecución, se debe comprobar si la variable está configurada o no antes de intentar tener acceso a ella. Las variables de sesión se descartan (borran) automáticamente después de no utilizarlas durante un tiempo específico, que se define en el archivo Web.config. En cada solicitud se restablece el tiempo de espera. Las variables se pierden cuando se abandona explícitamente la sesión en el código.

Cuando se inicia una sesión en la primer solicitud, el servidor emite un Id. de sesión único para el usuario. Para que persista el Id. de sesión, es aconsejable almacenarlo en una cookie en memoria (que es el valor predeterminado) o dado el caso de que no maneje cookies, se debe incrustar dentro de la dirección URL de la solicitud después del nombre de la aplicación. Para cambiar entre el estado de sesión con cookie y sin cookie, se configure en el archivo Web.config el valor del parámetro cookieless como true o false.

Identificador universal de recursos: La URL (también denominada Uniform Resource Identifier, URI) es un texto corto que identifica unívocamente cualquier recurso (servicio, página, documento, dirección de correo electrónico, etc.) accesible en una red. Normalmente una URI consta de dos partes:

- Identificador del método de acceso (protocolo) al recurso, por ejemplo http:.
- Nombre del recurso, por ejemplo "//www.google.com"

2.5.6 MULTITHREADING

Con el propósito de prestar servicios adicionales al usuario en una aplicación sin que esta detenga el procesamiento de los eventos en la interfaz y se mantenga un adecuado tiempo de respuesta de la aplicación, se requiere la ejecución de trabajos en el fondo, o escondidos para el usuario. El .Net CF provee un poderoso soporte para realizar estas tareas, lo que se conoce como Multithreading, y consiste en la capacidad de desarrollar más de una tarea al mismo tiempo. Cuando se implementa multithreading, la ejecución múltiple de hilos, algunas veces llamados procesos, se realiza dentro de una única aplicación. El sistema operativo es entonces capaz de programar hilos individuales dentro de una aplicación así que es posible simular el desempeño de múltiples tareas simultáneamente

Background process: El .Net CF provee tres técnicas primarias para correr procesos de fondo: servicios Web XML asincrónicos, pool de hilos e hilos creados explícitamente, cada uno es diseñado para un uso en un escenario particular, los cuales requieren de un seguimiento y de una administración apropiada.

Servicios Web XML asincrónicos: Normalmente los servicios Web son llamados a través de una clase Proxy generada a partir de VS .Net adicionando una referencia Web o a través del comando WSDL en la línea de comandos. La clase Proxy generada provee tres métodos por cada método web (Web Method) que el servicio Web proporcione. Un método llamado de la misma forma como la operación definida en el servicio Web, el cual llama sincrónicamente al servicio Web y un par de métodos con el prefijo **BeginNombreMétodo** y **EndNombreMétodo**, para hacer una llamada asincrónica del servicio Web. Así que si el Proxy del servicio Web expone una método llamado Add, adicionalmente tendrá los métodos BeginAdd y EndAdd, El siguiente ejemplo muestra como hacer uso del método asincrónico BeginBusqueda:

```
class Busqueda : Form
{
    private ProxyBusqueda _wsBusqueda = new ProxyBusqueda( );
    public string _result;

    public void btnBuscar_OnClick(object sender, EventArgs e)
    {
        AsyncCallback cb = new AsyncCallback(BusquedaCompleta);
        // Iniciando la llamada al servicio Web
    }
}
```

```

        _wsBusqueda.BeginBusqueda(txtBuscar.Text, cb, null);
    }

    // Funcion que se llama cuando el método asincrónico termina la tarea
    public BusquedaCompleta(IAsyncResult ar)
    {
        // obtenido el resultado del servicio Web XML
        _result = _wsBusqueda.EndBusqueda(ar);
    }
    ...
}

```

En el ejemplo anterior cuando la llamada al servicio Web es iniciada, la aplicación pasa un delegado `AsyncCallback` para el método `BeginBusqueda`. El delegado puede automáticamente ser llamado cuando la llamada al servicio se complete. Así que cuando se finalice la búsqueda el resultado será almacenado en la variable `_result`.

Pool de Hilos: En los casos donde una aplicación necesite desempeñar largos procesos locales tales como cálculos extensos, procesamiento o inicialización de archivos, el .Net CF provee la construcción de un pool de hilos. Este pool de hilos permite realizar trabajos en el fondo sin la sobrecarga frecuente de la creación y destrucción separada de hilos en cada tarea. El pool de hilos del .Net CF se realiza a través de la clase `ThreadPool`. Las tareas son encoladas dentro del pool de hilos envolviendo un método en un delegado `WaitCallback` y pasando el delegado al método estático `ThreadPool.QueueUserWorkItem`.

```

void eleminarSCO(object val)
{
    // Código que realiza la tarea especifica.
}

public void btnEliminar_Click(object sender, EventArgs e)
{
    // Envolviendo eleminarSCO en el delegado y enviándolo al Thread Pool
    WaitCallback w = new WaitCallback(eleminarSCO) ;
    ThreadPool.QueueUserWorkItem(w) ;
}

```

En el anterior ejemplo el método `eleminarSCO` será encolado dentro del pool de hilos y se ejecutará en uno de los hilos disponibles. La llamada a `QueueUserWorkItem` después de un tiempo coloca la petición en la cola.

Hilos Creados Explícitamente: En algunos casos, un proceso de fondo puede correr por mucho más tiempo en la vida de la aplicación, como es el caso de continuas lecturas a un GPS (Global Positioning System) o monitorear piezas de equipos. En estos casos es mucho mejor crear un hilo para una tarea específica. Estos hilos se manejan a través de instancias de la clase `Thread` y son creados envolviendo un método en un delegado `ThreadStart`, pasando el delegado al constructor de la clase `Thread` y llamando al método `Thread.Start`.

```
void actualizarConexion( )
{
    // Revisando si existe conexión con el sistema central
}

public void timerConexion_Tick(object sender, System.EventArgs e)
{
    ThreadStart startMethod = new ThreadStart(actualizarConexion);
    Thread conexionThread = new Thread(startMethod);
    conexionThread.Start( );
}
```

En este ejemplo, el método `actualizarConexion` está ejecutándose en un nuevo hilo creado. El hilo es dedicado al método `actualizarConexion` y terminará una vez el método termine.

2.6 OPEN NET COMPACT FRAMEWORK – Open Net CF

El Open Net Compact Framework [26] es un framework de aplicación que enriquece y extiende el .Net compact Framework. El Open Net CF implementa nuevos controles y librerías de clases, entre las que se encuentran namespaces de: control y acceso a datos, manejo de gráficos, networking, trabajo con XML, manejo de formularios, interoperabilidad con código nativo, implementación de Web Services Enhancements 2 (WSE2), entre otros. Este último en especial es el que permite lograr utilizar la implementación de WSE2 sobre dispositivos móviles.

2.7 ORACLE DATA PROVIDER FOR .NET (ODP.NET)

ODP.NET ofrece acceso optimizado a los datos de una base de datos Oracle desde el ambiente de .NET, permite a los desarrolladores tomar ventaja de la avanzada funcionalidad de las bases de datos Oracle, incluyendo cluster de aplicación real

(Configuración que puede tomar Oracle, en dónde la base de datos es manejada por más de una instancia), XML DB (alto desempeño de Oracle recuperando y almacenando XML nativo) y avanzada seguridad; además puede ser usado desde cualquier lenguaje de .NET. ODP.NET está diseñado para desarrollar soluciones escalables proporcionando completo soporte para Unicode y para transacciones locales y distribuidas, además, hace el uso de Oracle desde el ambiente de .NET más flexible, rápido y estable, ya que incluye características no disponibles desde los drivers de .NET, porque entre los tres métodos de acceso de los datos, OLEDB.NET, ODBC.NET, ODP.NET; ODP.NET es el más nativo al ambiente de NET y ofrece acceso de alto rendimiento a la base de datos Oracle. Una distinción importante es que ODP.NET no utiliza un puente adicional para acceso a los datos; por lo que tiene buen desempeño, mejor que OLE DB. NET y ODBC .NET. Además, porque estos puentes del NET que se construyen para las fuentes de datos genéricas, no proporcionan el acceso completo a la funcionalidad avanzada de la base de datos, como se puede ver en la Figura 12.

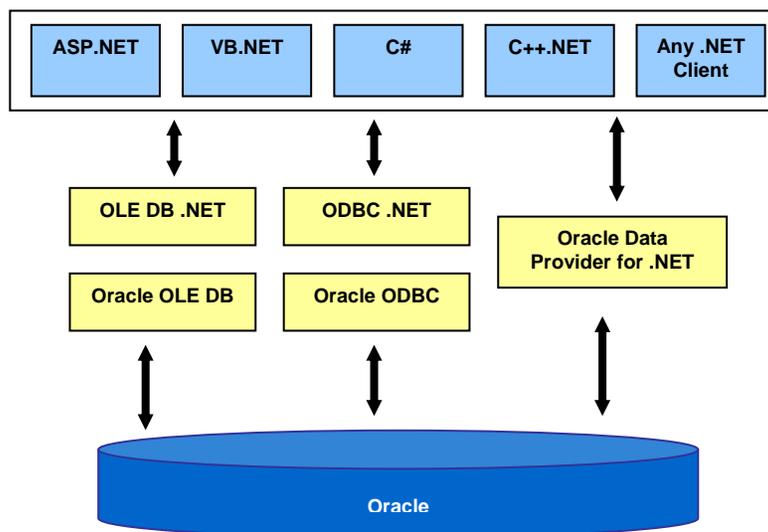


Figura 12 Acceso a datos Oracle.NET [47]

Los Oracle LOBs, BLOBs, y BFiles, son objetos binarios de gran tamaño de Oracle y son usados para almacenar varios tipos de datos binarios o caracteres de gran tamaño, incluyendo imágenes y documentos de texto. ODP.NET provee métodos nativos para recuperar y administrar los LOBs, lo cual mejora el desempeño del código que los maneja y hace que el código y proceso de escritura sea más fácil para los programadores.

ODP.NET expone muchas otras características de las bases de datos Oracle, incluyendo PL/SQL, PL/SQL transaccional y soporte a Unicode. Los usuarios de este proveedor de datos pueden ejecutar completamente procedimientos almacenados PL/SQL y funciones en la base de datos. Los PL/SQL pueden ser empaquetados o no empaquetados, además tiene la habilidad de retornar conjuntos de múltiples resultados desde un procedimiento almacenado.

ODP.NET tiene completo soporte para Unicode, así que los usuarios pueden globalizar sus aplicaciones fácilmente en múltiples lenguajes, esto hace el despliegue de aplicaciones en ambientes de múltiples lenguajes fácil y rápido.

3 DESCRIPCIÓN DEL SISTEMA

Easy Learning es un sistema de gestión de aprendizaje móvil para Pocket PC que combina algunas de las estrategias del aprendizaje autónomo. Los contenidos que se distribuyen con Easy Learning se basan en SCORM 1.2, pero no únicamente con este, ya que su diseño es flexible y permite usar nuevas versiones de SCORM y otros estándares de educación en línea.

Easy Learning fue diseñado para tres roles principales, el administrador, el docente (instructor, profesor, guía o tutor) y el estudiante. Las tareas que puede llevar a cabo el administrador son: entrar al sistema, administrar los roles, los usuarios del sistema, las asignaturas y los cursos. Las tareas que puede realizar el docente son: entrar al sistema, seleccionar los cursos que va a orientar, administrar las agregaciones (módulos, capítulos, lecciones y otros), objetos de contenido compartible (SCOs) y evaluaciones (test, preguntas, fechas y demás) e interactuar con sus estudiantes y otros colegas a través de salas de chat y foros. Finalmente, las tareas que puede llevar a cabo el estudiante son: entrar al sistema, seleccionar el curso que desea estudiar, seleccionar las agregaciones específicas que desea revisar, ver los SCOs e interactuar con sus compañeros y sus profesores a través de salas de chat y foros.

3.1 ARQUITECTURA DEL SISTEMA

La arquitectura de la aplicación tiene tres capas, Interfaz (Objetos de frontera), Lógica (Objetos de control) y Almacenamiento (Objetos de entidad). El almacenamiento de datos fue hecho en una base de datos Oracle9i™, y el almacenamiento de los archivos de contenidos fue hecho en un directorio virtual de Microsoft Internet Information Server (IIS). El servidor de aplicaciones corre bajo IIS y fue hecho con servicios Web XML en tres capas internas, lógica de servicios, lógica de negocios y una fachada para todos los servicios Web.

La capa de lógica de servicios es un conjunto de servicios Web que permiten manejar la persistencia de los objetos en la base de datos relacional y en el directorio virtual de IIS. El diseño permite a la aplicación cambiar las fuentes de información a otros tipos de fuente, por ejemplo otro motor de base de datos, sin tener que cambiar ninguna otra capa

de la aplicación. La capa de lógica del negocio en el servidor se encarga de manejar y entender las distintas relaciones que existen entre los datos relacionales, los contenidos, los roles de usuarios, los usuarios, los cursos y en general todos los conceptos que manipula la aplicación.

El cliente Pocket PC que es utilizado por los estudiantes y el cliente Windows PC que utilizan los docentes, tienen una arquitectura similar dividida en tres capas internas, una capa Smart que contiene clases de administración y control de conexión, las cuales se encargan de controlar la conexión inalámbrica/alámbrica, ya que estas monitorean si existe o no la conexión con el servidor y con base en eso habilita/deshabilita los servicios que se pueden utilizar; además, se encargan de manejar una copia de los archivos de contenido y de algunas tuplas relacionales que se necesitan para trabajar fuera de línea a través de un cache de directorio local. Una capa para la lógica de interfaz, que es la encargada de dar el formato adecuado de la información que recibe y retorna, y una capa para los formularios Windows Pocket PC y Windows PC para la visualización e interacción del estudiante y el docente (ver Figura 13), la diferencia entre estos clientes es que los estudiantes realizan exclusivamente procesos de aprendizaje, mientras que los docentes realizan procesos de seguimiento y administración a los procesos de aprendizaje de los estudiantes .

El cliente Web que es utilizado por los administradores también cuenta con tres capas internas, una capa para los formularios Web, otra para las páginas de servidor que manejan la lógica de la interfaz y por último una capa para realizar un Proxy de la fachada de los servicios Web de la lógica de la aplicación. Esta aplicación sólo trabaja en línea, y presta exclusivamente servicios de administración del sistema, en la Figura 14 se puede ver un resumen de su arquitectura.

Aprovechando las capacidades actuales del servidor Oracle9i™ se utilizó una configuración de servidor multihilo o compartido y conexiones de usuarios que perduran durante la conexión, por medio de variables de sesión. De esta forma el tiempo de respuesta de la aplicación es mucho mejor que establecer conexiones intermitentes o que se abren para cada solicitud y el costo de memoria por cada conexión de usuario es muy bajo.

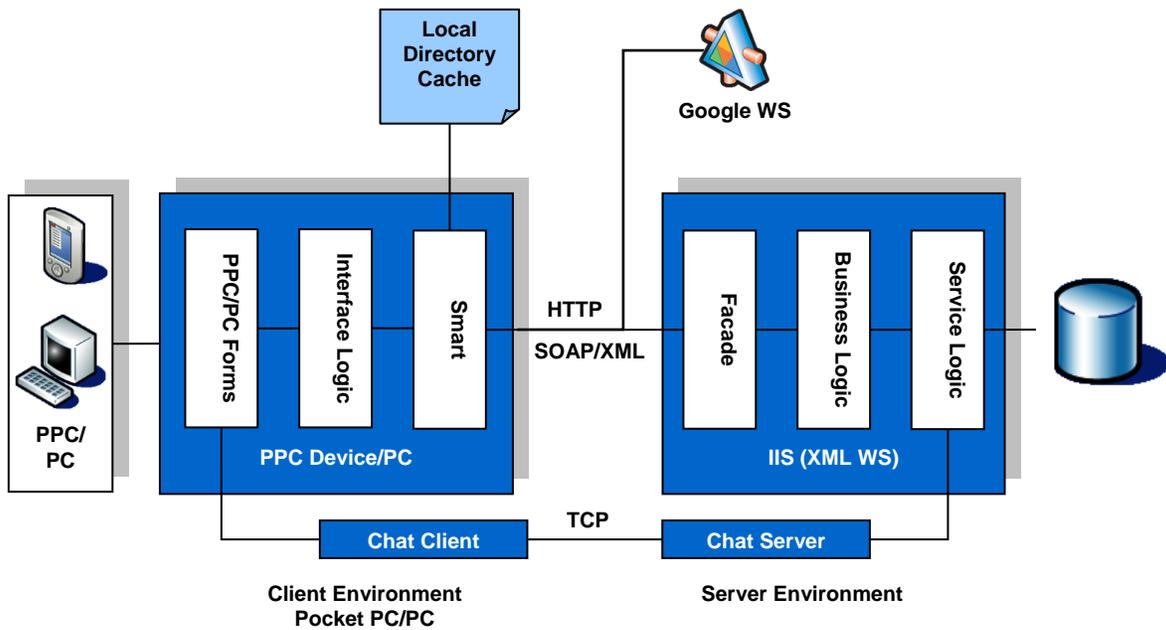


Figura 13 Arquitectura para el cliente Pocket PC y Windows PC

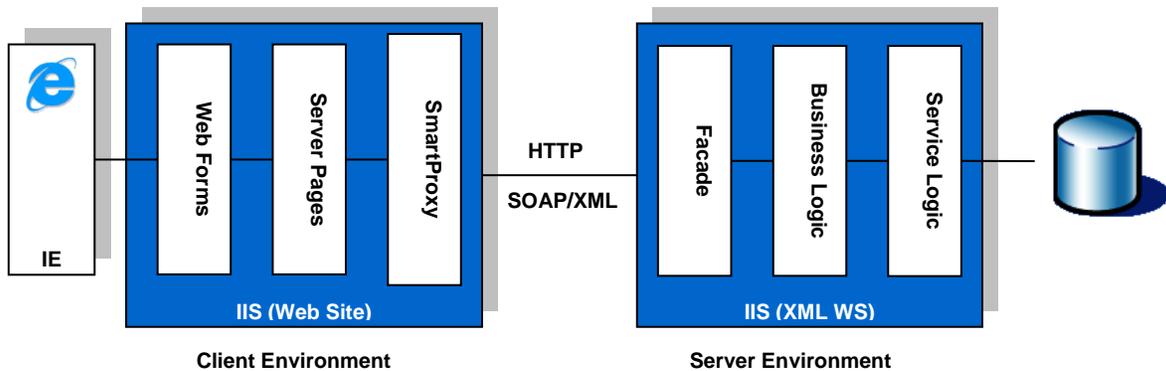


Figura 14 Arquitectura para el cliente Web

3.1.1 DESPLIEGUE DE LA APLICACIÓN

La Figura 15 muestra las tres aplicaciones (el servidor distribuido en dos máquinas con una red de alta velocidad que las conecta, la aplicación cliente Pocket PC que usa el estudiante, la cual se conecta con el servidor a través de una red inalámbrica, la aplicación Windows PC que utiliza el docente y la aplicación Web del administrador que también se conectan a través de una red alámbrica, aunque estas últimas también se podrían conectar a través de una red inalámbrica). En la Figura 15 se muestra la aplicación con el uso de una red inalámbrica, que para nuestro caso es una red basada

en 802.11 a/b/g. En otros ambientes o ubicaciones sólo se necesita que una compañía de telecomunicaciones provea una comunicación inalámbrica que permita usar en el nivel de aplicación soap, http, udp y tcp.

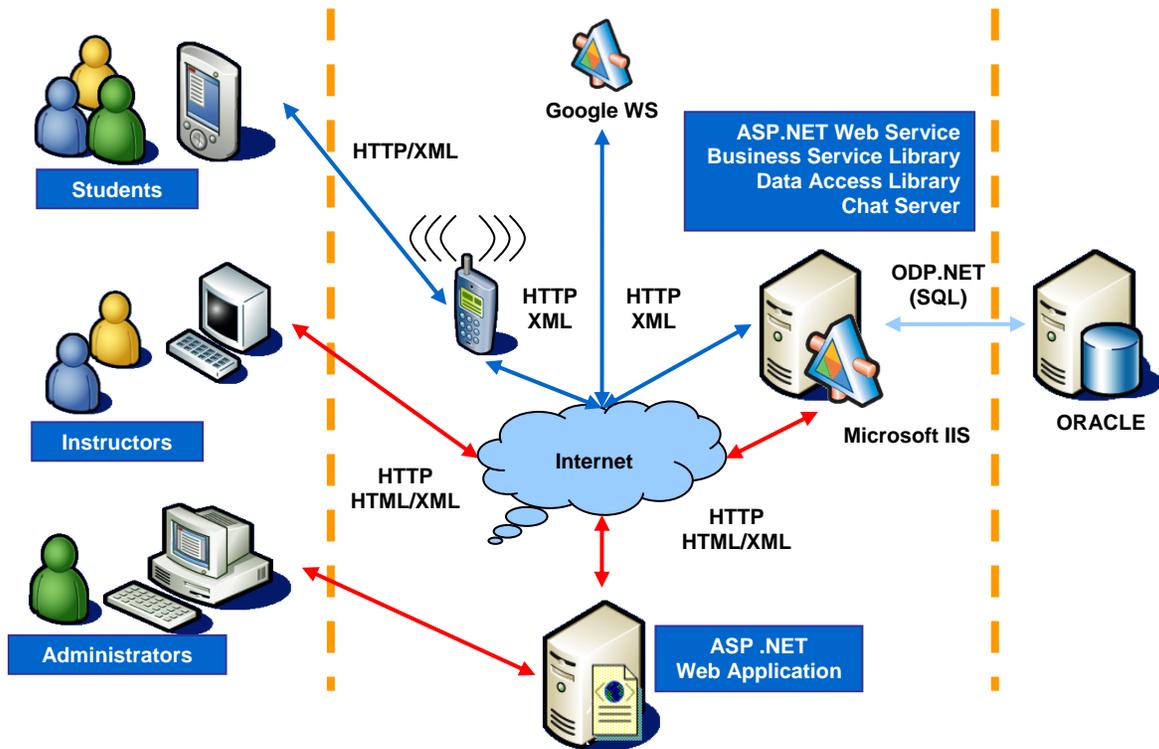


Figura 15 Despliegue de Easy Learning

En la Figura 16 se puede observar el formulario de ingreso al sistema a través del cual el estudiante digita su nombre de usuario y clave. En este caso el sistema determina si existe conexión con el servidor, si es así lo registra frente al servidor, si no, revisa en el caché de directorio local (fuera de línea) si existe la información necesaria para que el estudiante ingrese al sistema.



Figura 16 Ingreso al sistema Pocket PC

En las Figura 17 se muestran las asignaturas que el estudiante tiene matriculadas y las agregaciones (capítulos, módulos o lecciones) para la asignatura seleccionada; en este caso se visualiza la asignatura Informática y sus agregaciones.



Figura 17 Selección de asignaturas

En la Figura 18 se muestra el objeto de contenido (SCO) que el estudiante quiere visualizar/estudiar, en este caso “Lógica de Proposiciones” de la asignatura de Informática.



Figura 18 Visualización de SCOs

En la Figura 19 se puede observar la opción de búsqueda a través del servicio Web XML de Google. Es preciso tener en cuenta que no se está navegando a la página Web de Google, se está usando una interfaz de clases y métodos ofrecidas por Google para realizar las búsquedas.



Figura 19 Uso del Servicio Web de Google

En la Figura 20 se muestra el resultado de la búsqueda efectuada, para ello se utiliza el navegador Web que está incrustado en Easy Learning, es decir la aplicación no abre un navegador externo, logrando con esto concentrar el trabajo del estudiante con una sola herramienta.



Figura 20 Resultados de la búsqueda

En la Figura 21 se muestra el módulo de administración del repositorio local; en donde se pueden listar y eliminar manualmente archivos de SCOs descargados en el dispositivo. La principal utilidad de este módulo, es la de informar al usuario los SCOs que podrá revisar al desconectarse de la red, si el prevé que va a ser por un largo tiempo, es decir descargue lo que necesita estudiar y desconéctese de la red. Esto da mucha flexibilidad y disminuye costos, si el acceso a la red inalámbrica se realiza a través de una empresa de telefonía celular o móvil que cobre por tiempo de conexión.



Figura 21 Administración del repositorio local

En la Figura 22 se muestra la interfaz principal del foro, en ésta el estudiante puede visualizar todos los mensajes organizados en forma de árbol (jerárquica), mostrando el título de cada del mensaje. Al seleccionar (dar clic) en un nodo (rama o hilo de conversación) del árbol se muestra todo el mensaje con información detallada, la fecha de publicación y la persona que lo público. Un estudiante puede seleccionar un nodo y oprimir el botón Responder o Preguntar, en ese momento se despliega la Figura 23 que le permite registrar un aporte. El sistema automáticamente registra el título del mensaje, el nombre del usuario, la fecha, el tipo de aporte, y el texto en si mismo de ese aporte.



Figura 22 Visualizar foros



Figura 23 Agregar un aporte al foro

En la Figura 24 se presenta la interfaz que le permite al estudiante autorizado crear y administrar sus foros, especificando la fecha de expiración de éste y los usuarios autorizados para accederlo



Figura 24 Administrar foros

En la Figura 25 se muestra la interfaz del chat a través del cual el estudiante podrá interactuar con otros compañeros y profesores.



Figura 25 Chat

En la Figura 26 se presenta la interfaz que le permite al estudiante realizar evaluaciones o auto evaluaciones de la asignatura seleccionada. En esta se presenta la calificación (en

porcentaje) obtenida luego de finalizar la prueba. El sistema registra los resultados de la evaluación pregunta por pregunta para dar mayor feedback al profesor de la asignatura.



Figura 26 Servicios de evaluación

En la Figura 27 se puede observar el formulario Web de ingreso al sistema a través del cual el administrador digita su nombre de usuario y clave.

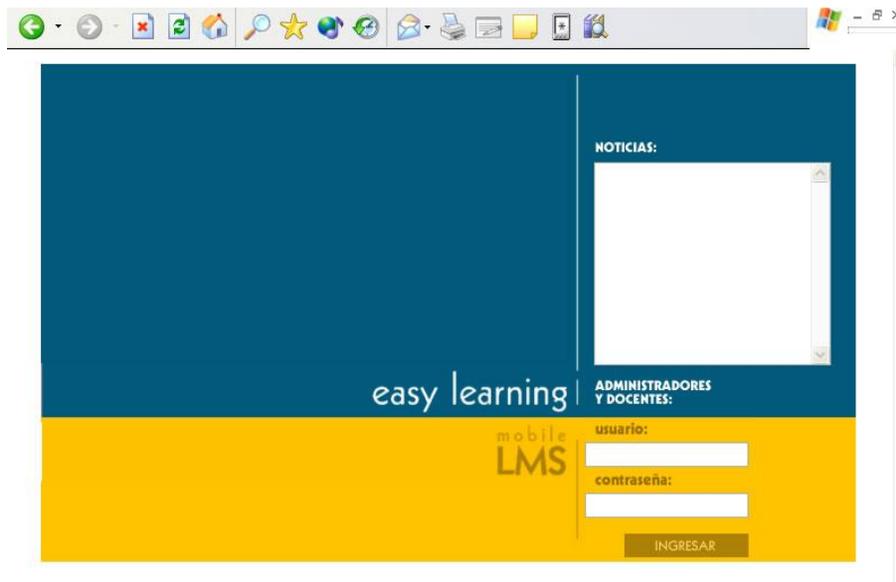


Figura 27 Ingreso al sistema Web

En la Figura 28 se presentan interfaz Web a través de la cual el administrador puede gestionar o manipular los usuarios y el rol al que pertenecen.

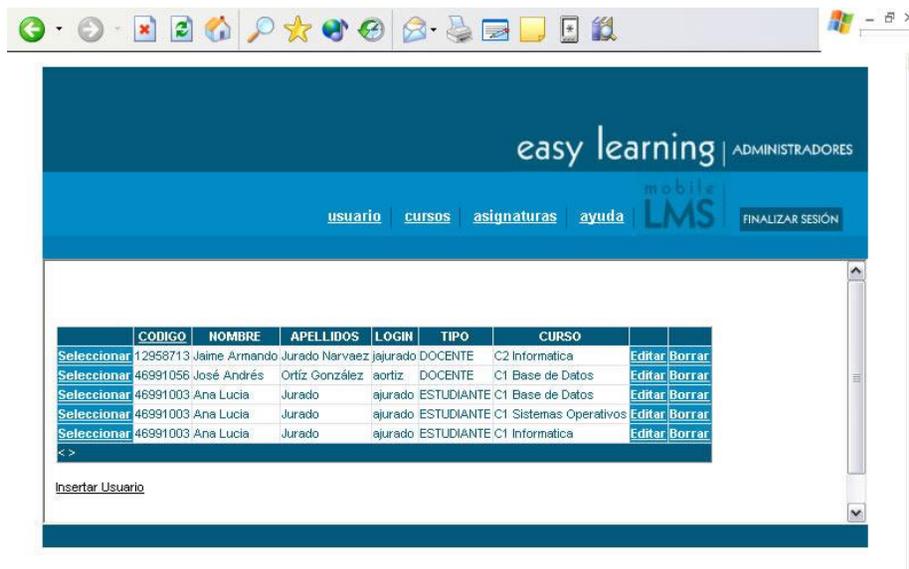


Figura 28 Gestión de Usuarios

3.2 DIAGRAMA DE CASOS DE USO

A continuación en la Figura 29 se presenta el diagrama general de los casos de uso:

3.2.1 DESCRIPCIÓN GENERAL DEL DIAGRAMA DE LOS CASOS DE USO

El sistema cuenta con tres actores principales: el estudiante, el docente y el administrador; los cuales desempeñan las siguientes tareas con el sistema:

Administrador, es el encargado de:

- Crear, modificar, eliminar y otorgar perfiles a los usuarios.
- Crear y eliminar cursos.
- Crear y eliminar asignaturas.
- Realizar el mantenimiento de la aplicación.

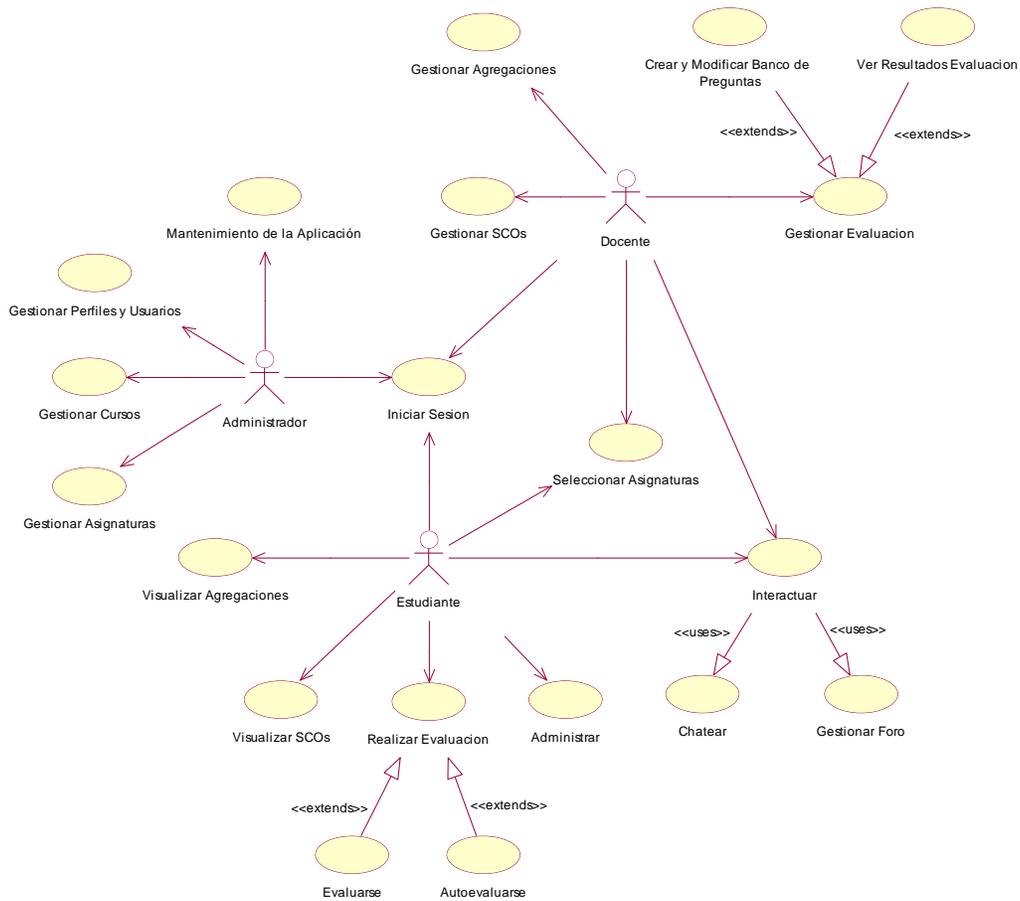


Figura 29 Diagrama General de los Casos de Uso

Docente, es el encargado de:

- Generar y gestionar los contenidos programáticos de sus asignaturas adicionando agregaciones y SCOs.
- Realizar y publicar las evaluaciones y auto evaluaciones para los estudiantes.
- Crear, gestionar y participar en los foros oficiales de los cursos.
- Participar en el chat.
- Realizar seguimiento a las evaluaciones y auto evaluaciones presentadas por los estudiantes.

Estudiante, es el encargado de:

- Visualizar las asignaturas en las que se encuentra registrado, con su contenido y recursos correspondientes.

- Realizar las evaluaciones y auto evaluaciones publicadas por el docente.
- Interactuar con sus compañeros y docentes a través de foros y chats.
- Crear sus propios foros y otorgar permisos de participación de estos foros.
- Administrar los recursos descargados y establecer parámetros de configuración del servidor de los servicios Web, del chat, el puerto por el que se atienden las peticiones del chat y el lugar donde se descargan los recursos.

3.3 DESCRIPCION DE LOS PRINCIPALES CASOS DE USO

En este ítem se presentan unos casos de uso, ya que mostrar la totalidad de los mismos se sale de los alcances de este documento (ver ANEXO B).

Caso de uso:	Seleccionar Asignaturas y Visualizar Agregaciones
Actores:	Estudiante (iniciador)
Propósito:	Seleccionar las asignaturas del estudiante y visualizar las agregaciones de la selección.
Resumen:	El estudiante visualiza e interactúa con un árbol jerárquico de agregaciones de una materia, e información de los SCOs pertenecientes a dichas agregaciones.
Tipo:	Primario Real
Precondiciones:	Se debe haber registrado previamente en el sistema
	
Curso Normal de los eventos	

Acción de los actores	Respuesta del sistema
1. Este caso de uso inicia cuando el estudiante ingresa al sistema.	
	2. Si el usuario está conectado con el sistema obtiene las asignaturas en las cuales el usuario esta matriculado desde la base de datos central, y las carga en el control ComboBox. En caso contrario obtiene las asignaturas desde el cache de directorio local.
	3. Presenta el formulario de las asignaturas.
4. Obtiene las asignaturas en las cuales se encuentra matriculado. Puede seleccionar del control ComboBox entre las diferentes asignaturas en las que se encuentra matriculado actualmente.	
5. El estudiante selecciona la asignatura de la cual desea ver sus agregaciones.	6. Si el usuario está conectado al sistema obtiene las agregaciones (módulos, capítulo y subcapítulos) de la asignatura y la información de los SCOs (recursos) disponibles para las agregaciones desde la base de datos central y guarda en el repositorio local (Cache de Directorio Local) en un archivo XML las agregaciones y en otro archivo XML la información de los SCOs.
	7. Si el usuario no esta conectado al sistema busca la información de las agregaciones y sus SCOs en los archivos XML del cache de directorio local.

	8. Muestra gráficamente en forma jerárquica de árbol las agregaciones pertenecientes a la materia seleccionada y la información de sus SCOs relacionados.
9. El estudiante puede navegar a través del árbol de agregaciones y expandir los nodos del árbol para visualizar la información de los SCOs pertenecientes a cada agregación.	
10. Si el estudiante selecciona un nodo perteneciente a un SCO de una agregación se inicia el caso de uso Visualizar SCOs .	
Flujos de Excepción:	
<ul style="list-style-type: none"> • Línea 7: Si no es posible obtener la información desde el repositorio local se muestra el mensaje: "No fue posible obtener los temas y contenidos de la materia". 	

Caso de uso:	Visualizar SCOs
Actores:	Estudiante (iniciador)
Propósito:	Permitirle al estudiante realizar procesos de aprendizaje
Resumen:	El estudiante visualiza los SCOs (recursos) pertenecientes a las agregaciones de una asignatura, seleccionándolos a partir de un árbol jerárquico.
Tipo:	Primario Real
Precondiciones:	Debe haber seleccionado una asignatura y una agregación de la asignatura que contenga al menos un SCO.

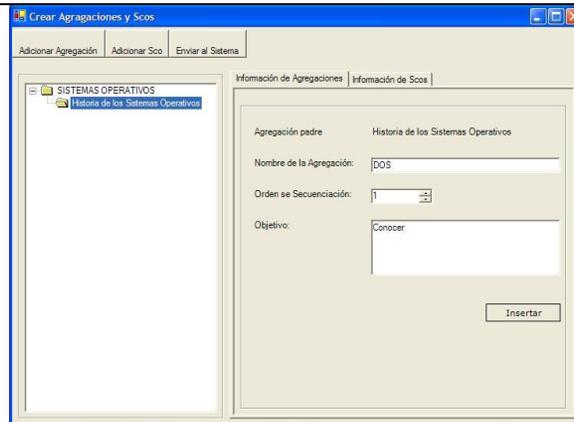


Curso Normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso comienza cuando un estudiante desea visualizar un SCO (recurso) perteneciente a una agregación de una asignatura.	
2. El estudiante selecciona de un árbol jerárquico la agregación de la cual desea visualizar el SCO. El estudiante selecciona el SCO que desea visualizar.	3. Obtiene el SCO (recurso) seleccionado por el estudiante. Guarda en el repositorio local (Cache de Directorio Local) el archivo de recurso seleccionado. Guarda en el repositorio local un archivo XML con información acerca de los SCOs que el estudiante ha descargado, el cual se actualiza cada vez que se realiza una nueva descarga. Muestra gráficamente a través de un visor de SCOs el recurso seleccionado.
4. El estudiante revisa el recurso a través	5. Si el estudiante selecciona un nuevo

del visor y puede regresar al árbol de agregaciones y contenidos para seleccionar un nuevo recurso.	recurso se repite la línea 3.
Cursos Alternativos:	
<ul style="list-style-type: none"> • Línea 3: El sistema verifica primero en el archivo XML de SCOs descargados, si el SCO ya se encuentra en el dispositivo, de ser así carga el recurso local. Si no se encuentra el recurso localmente realiza la descarga desde el servidor. Si el archivo no se encuentra en el servidor devuelve un mensaje: "Recurso no disponible". 	

Caso de uso:	Gestionar Agregaciones – Insertar Agregaciones
Actores:	Docente (iniciador)
Propósito:	Insertar agregaciones a una nueva asignatura
Resumen:	El docente inserta agregaciones a una asignatura, adicionando nodos a un árbol jerárquico.
Tipo:	Primario Real
Precondiciones:	Debe haber un nodo seleccionado

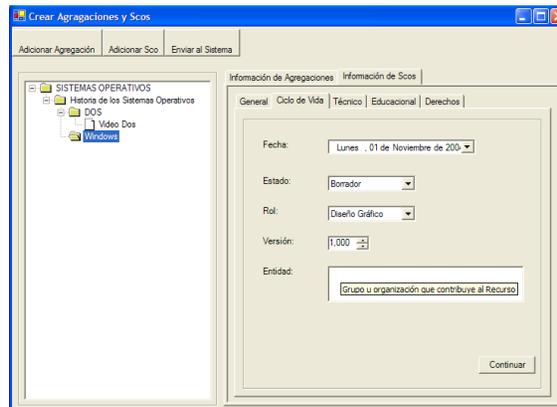


Curso Normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el docente hace click sobre el botón Adicionar Agregación de la barra de herramientas del formulario	2. Hace visible el TabPage Información de Agregaciones, que contiene los campos que se deben suministrar para la inserción de la agregación y

	adiciona el nombre de la agregación seleccionada en el campo Agregación Padre.
3. Ingresar los datos y dar clic sobre el botón insertar.	4. Recoge la información y la inserta en el árbol como nodo hijo del nodo seleccionado y expande el árbol. Luego limpia los valores de los objetos y hace invisible el TabPage Información de Agregaciones.
5. El docente puede seguir adicionando nodos agregaciones como lo requiera.	
Flujos de Excepción:	
<ul style="list-style-type: none"> • Línea 1. Si el nodo seleccionado es de tipo SCO y no de tipo Agregación se muestra el siguiente mensaje de error, “El nodo debe ser de tipo Agregación” • Línea 4. Si la información no está completa, presenta un mensaje de error. 	

Caso de uso:	Gestionar SCOs – Insertar SCO
Actores:	Docente (iniciador)
Propósito:	Insertar SCOs a una agregación.
Resumen:	El docente inserta un SCO a una agregación de una asignatura, adicionando nodos a un árbol jerárquico
Tipo:	Primario Real
Precondiciones:	Se debe haber seleccionado una agregación.



Curso Normal de los eventos	
Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el docente hace clic sobre el botón, Adicionar SCO	2. Verifica que el nodo sea de tipo Agregación y hace Visible el TabPage Información de SCOs, que contiene los objetos en donde se debe ingresar la información del SCO, además adiciona el nombre del la agregación seleccionada en el campo Agregación Padre.
3. El docente ingresa la información y da clic en el botón continuar del TabPage Información de SCOs.	4. Activa el siguiente TabPage Información de SCOs.
5. El docente termina de llenar la información cuando encuentre el botón Insertar y da clic en él.	6. Recoge la información y la adiciona en forma de Nodo, como nodo hijo de la agregación seleccionada en el árbol. Luego expande el árbol, limpia los valores de los objetos y oculta el TabPage Información de SCOs.
7. El docente puede ingresar tantos nodos SCO como lo requiera	
Flujos de Excepción:	
	<ul style="list-style-type: none"> • Línea 1. Si el nodo seleccionado es de tipo SCO y no de tipo Agregación se muestra el siguiente mensaje de error, “El nodo debe ser de tipo Agregación” • Línea 6. Si la información no está completa, presenta un mensaje de error.

3.4 DIAGRAMAS DE SECUENCIA

La Figura 30 muestra el diagrama de secuencia de los casos de uso *Seleccionar Asignatura y Visualizar Agregaciones*, el cual se inicia luego de haber cargar las asignaturas del estudiante, cuando el este selecciona una asignatura, lo que envía un

mensaje al objeto *frmAsignaturas* para cambiar la asignatura, este objeto a su vez envía un mensaje al objeto *SmartProxy* para que se le retornen las agregaciones de la materia seleccionada, como el objeto *SmartProxy* no es el apropiado para llevar a cabo esta labor directamente le envía el mensaje al objeto *Asignatura*, el cual se comunica con el objeto *serFachada*, que como su nombre lo indica es una fachada, que contiene todos los métodos fachada de las clases de lógica del negocio, por lo cual comunica el mensaje a la clase experta en la labor o petición que se le haga, en este caso envía el mensaje al objeto *serAsignaturas*, el que finalmente se comunica directamente con el objeto de la clase que maneja la lógica de servicios, *serBaseDeDatos*, para que obtenga las agregaciones de la materia seleccionada de la base de datos y las retorne.

En el camino de retorno si fue posible obtener las agregaciones, estas son guardadas localmente en el repositorio local del dispositivo en un archivo XML a través del objeto *PersistenciaXML*. En el caso de no poder obtener las agregaciones desde el servidor se revisa el repositorio local para ver si se encuentran ya grabadas en el archivo XML, si es así se obtienen también a través del objeto *PersistenciaXML*, de lo contrario se presenta un mensaje de error al usuario y se finaliza la secuencia de mensajes.

Si fue posible obtener las agregaciones, por cada una de estas se envía una secuencia de mensajes en un orden similar para solicitar la información de los SCOs, por cada una de las agregaciones, la que al igual que las agregaciones es posible guardar y obtener desde el repositorio local. Finalmente a medida que se obtiene la información de los SCOs de las agregaciones se arma el árbol con la información correspondiente. De no ser posible obtener la información de los SCOs se presenta un mensaje de error al usuario y se termina la secuencia de mensajes.

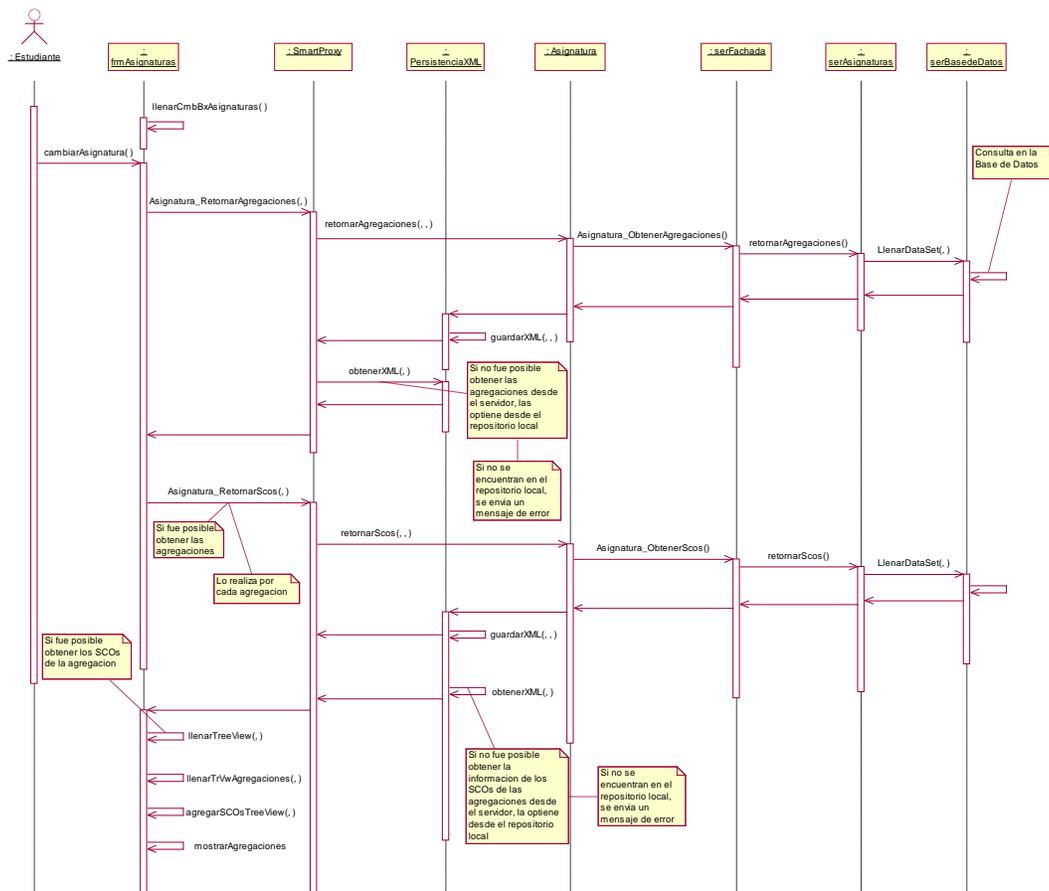


Figura 30 Diagrama de Secuencia Seleccionar Asignatura y Visualizar Agregaciones

La Figura 31 muestra el diagrama de secuencia del caso de uso visualizar SCOs, el cual se inicia cuando el estudiante selecciona un nodo en el árbol de agregaciones e información de sus SCOs, lo que envía un mensaje al objeto *frmAsignaturas* para cargar (mostrar) el SCO, el cual revisa en el repositorio local si el SCO (recurso) ya se encuentra descargado, de ser así se carga y muestra el SCO al estudiante.

En caso contrario el objeto *frmAsignaturas* envía un mensaje al objeto *SmartProxy* para que se retorne el recurso seleccionado, el objeto *SmartProxy* no es el apropiado para llevar a cabo esta labor directamente así que reenvía el mensaje al objeto *SCO*, el cual se comunica con la el objeto *serFachada*, que comunica el mensaje a la clase experta en la petición que se esta haciendo, en este caso envía el mensaje al objeto *serAsignaturas*, el que finalmente busca el SCO en repositorio de SCOs del servidor y lo retorna.

En el camino de retorno si fue posible obtener el SCO, se guarda la información de descarga se ese SCO en el repositorio local del dispositivo en un archivo XML, a través del objeto *PersistenciaXML*. Para finalmente cargar y mostrar el SCO al usuario. En el caso de no poder obtener el SCO desde el servidor se presenta un mensaje de error al usuario y se finaliza la secuencia de mensajes.

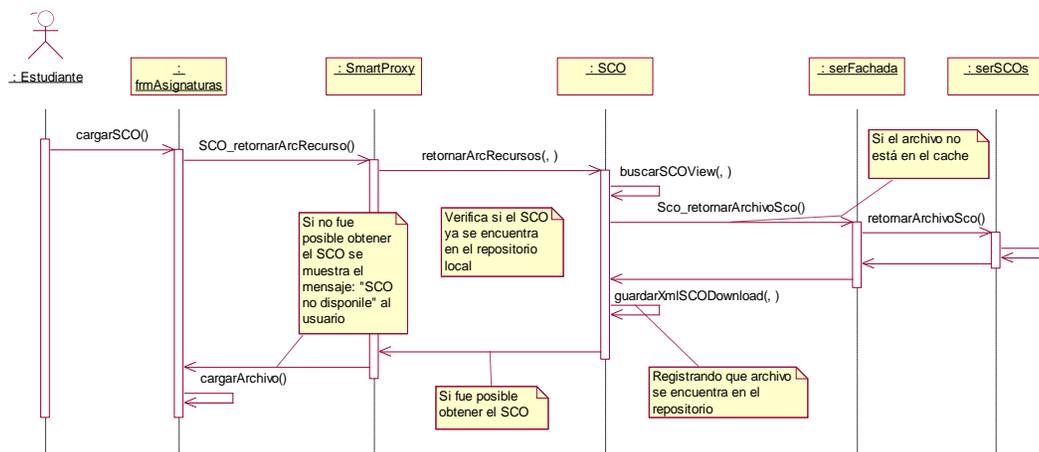


Figura 31 Diagrama de Secuencia Visualizar SCO

3.5 DIAGRAMA DE CLASES

La Figura 32 presenta el diagrama de clases de interfaz del cliente Pocket PC (PPC) al igual que el diagrama de lógica del cliente PPC. En la parte correspondiente al diagrama de clases de interfaz del cliente PPC se muestran las clases encargadas de realizar la visualización e interacción del estudiante con el sistema, a través de la interfaz gráfica de la clase *frmLogon* el estudiante ingresa login y password para acceder al sistema, esta clase tiene una relación de dependencia con la clase *frmAsignaturas* ya que se crea para capturar la información del estudiante y se destruye luego de enviarla a la clase *frmAsignaturas* y obtener ingreso al sistema. La clase *frmAsignaturas* es además la encargada de crear las clases *frmChat*, *frmForo*, *frmAdminForo* y *frmEvaluación*, así que tienen una relación de composición, aunque estas manejan su propia lógica de visualización e interacción con el estudiante y se visualiza su interfaz gráfica a partir de las interacciones del usuario con la clase *frmAsignaturas*.

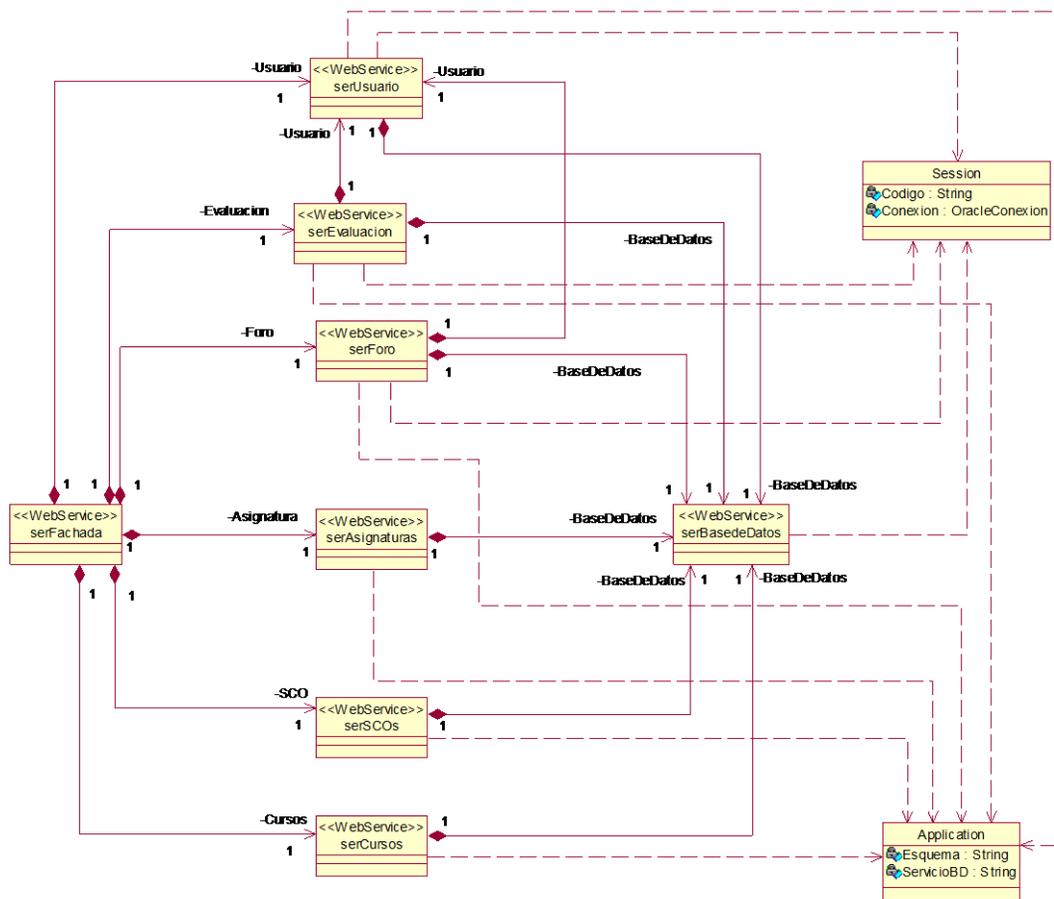


Figura 33 Diagrama de Clases de Lógica del Negocio y Lógica de Servicios

La lógica del negocio se compone de las clases: serUsuario, encargada de gestionar toda la información correspondiente a un usuario; serEvaluación quien es la encargada de tratar los datos correspondientes a las evaluaciones; serForo quien es la responsable de administrar, modificar, eliminar y crear foros en el sistema para los diferentes cursos; serAsignaturas que contiene los métodos de gestión de las asignaturas y serSCOs responsable de trabajar con los datos de los SCOs. Estas clases de lógica del negocio son servicios Web, tienen una relación de composición con serFachada, y para poder completar correctamente sus tareas se comunican con la base de datos Oracle por medio de la lógica de servicios, serBaseDeDatos, quien también es un servicio Web. serBaseDeDatos contiene todos los métodos necesarios para interactuar con la base de datos y es la única que tiene comunicación con ella.

Las clases de lógica del negocio utilizan la variable de aplicación *Esquema*, para las sentencias de consulta, inserción y modificación en la base de datos, esto porque para tener acceso a los objetos de la base de datos de otros esquemas se debe anteponer el nombre del esquema dueño al objeto; a su vez *serBaseDeDatos* utiliza la variable de aplicación *ServicioBD*, quien almacena el nombre de servicio al cual los usuarios se conectan.

Adicionalmente los servicios Web de lógica del negocio manejan una variable de sesión *Código*, en donde se guarda el identificador del usuario y *serBaseDeDatos* trabaja con la variable de sesión *Conexión*, con la que realiza las peticiones a la base de datos durante la sesión del usuario.

3.6 MODELO DE LA BASE DE DATOS

La persistencia de los datos del sistema se realiza en una base de datos relacional en Oracle 9i configurada como servidor compartido o multihilos, en donde al crearse la instancia de la base de datos, se lanza un pool de procesos servidor (apropiada para configuración de procesamiento analítico en línea, OLTP). Un proceso de servidor atiende a varios procesos cliente y para gestionar el turno de los procesos cliente hay un proceso despachador (uno por protocolo), una cola de peticiones y una de respuesta. Cuando llega una petición el proceso despachador verifica si hay procesos de servidor disponibles para atenderla, dado el caso de que no haya uno, la petición pasa a la cola de peticiones hasta que llegue su turno, cuando el proceso servidor termina con la petición coloca la respuesta en la cola de respuestas, en donde el despachador está pendiente para retornarla al cliente. Se tomó esta configuración porque la base de datos tiene tantos usuarios, como usuarios tiene el sistema, además del esquema *mlearning*, el cual es el propietario de los objetos, como por ejemplo las tablas, vistas y secuencias. Con este esquema se maneja por una sesión una conexión a la base de datos y se controlan mejor las operaciones de los usuarios en el sistema sin sobrecargar el motor de la base de datos. Los usuarios esquemas están diferenciados por el rol otorgado, este puede ser *ADMINMLMS* para administradores, *MLEARNING_DOCENTE* para los docentes y *MLEARNING_ESTUDIANTE* para los estudiantes. A continuación se ve el modelo físico de la base de datos separado por vistas.

3.6.1 PROGRAMA ACADÉMICO DEL USUARIO

En la Figura 34 se presenta el modelo físico que representa la información correspondiente al programa académico de un usuario. En éste se puede apreciar la tabla tblUsuario que guarda los datos correspondientes a un usuario, como su nombre y apellidos, su login, el cual utiliza para tener acceso al sistema, su identificación ante la institución, que para el caso de los estudiantes es el código asignado y para los docentes la cédula, dado el caso que la institución no les otorgue un código, y un campo que indica si el usuario está o no bloqueado para ingresar al sistema. El password que el usuario utiliza corresponde al usuario esquema y por tal no está en la tabla. Se puede ver también la tabla tblCurso que tiene el nombre de un curso y la asignatura relacionada al curso. La relación entre el usuario y el curso está dada por la tabla tblUsuarioCurso quien tiene un campo adicional que guarda la relación entre el usuario y el curso, es decir es estudiante o docente. También se tienen las tablas relacionadas al programa académico de un curso, entre ellas tblAsignatura que guarda las asignaturas existentes en el sistema, la tabla recursiva tblAgregaciones que guarda todas las agregaciones o contenidos y módulos pertenecientes a una asignatura, de éstas se guarda, a parte de su nombre y la agregación padre, el objetivo que la agregación persigue y su orden, que permite llevar a cabo reglas de secuenciación. La tabla tblSco guarda la información general acerca de un SCO o recurso de la asignatura, entre ellas el nombre del recurso, la ubicación del recurso dentro del servidor, además de la fecha en la que el recurso fue publicado. Por último se puede diferenciar en la gráfica la tabla tblAgregacionSco, quien representa la relación existente entre una agregación y un SCO.

3.6.2 META MODELOS PARA LA GENERACIÓN DE SCO

En la Figura 35 se muestran las tablas que permiten utilizar cualquier estándar para la generación y almacenamiento de los objetos de contenido intercambiable, SCO. El modelo empieza con tblPlantillas, esta tabla contiene los datos del modelo o estándar, como su nombre, la versión utilizada, la fecha en que se ingresa el modelo, su identificador dentro del sistema y el identificador del campo raíz del mismo.

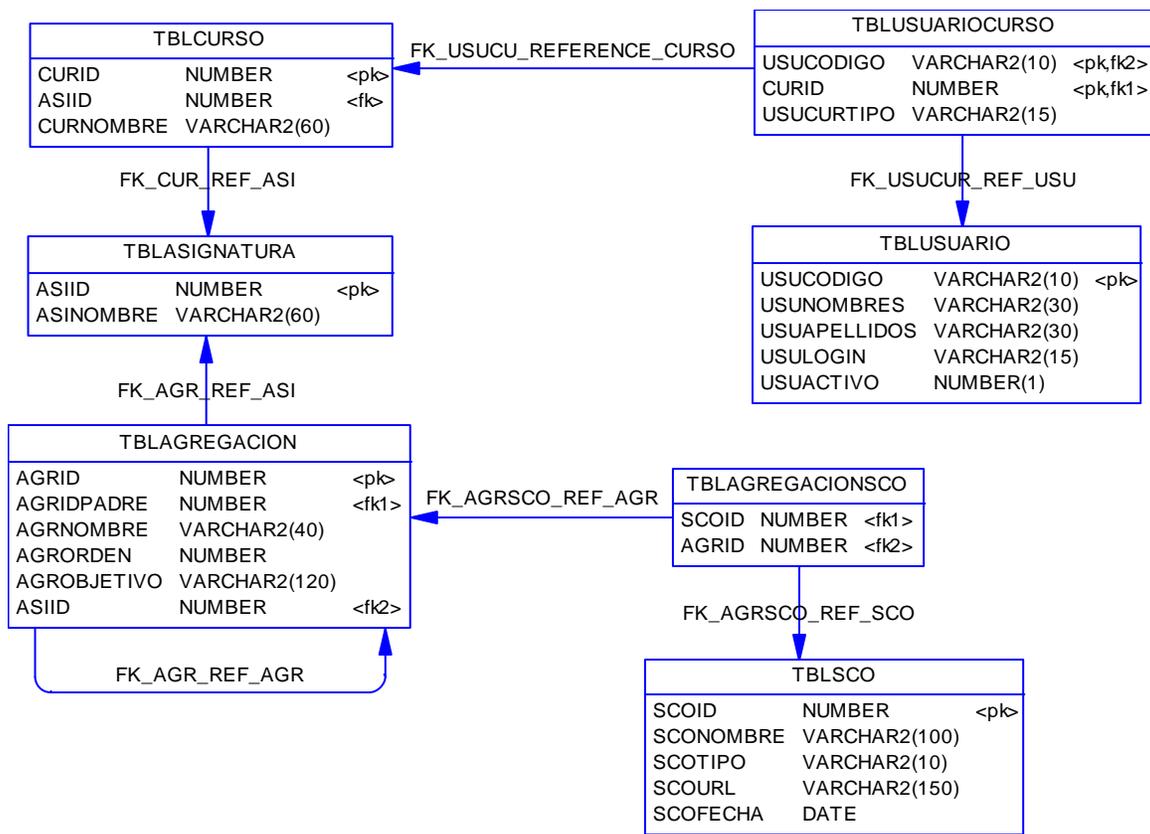


Figura 34 Modelo físico del programa académico de un usuario

En tblCamposMetadatos se almacenan los campos de todas las plantillas disponibles en el sistema, en esta tabla se guarda el nombre e identificador del campo, este último valor es dado por la secuencia seqCamposMetadatos, el nombre del campo, el tipo de dato del campo está dado por la referencia a la tabla tblTipoDatosMetadatos, en donde se guarda los posibles tipos de datos de un campo con su identificados, por ejemplo string, int, flota, etc; y como también es posible que el valor del campo pueda estar definido entre un conjunto de valores se creo la tabla tblMultivaloresMetadatos, que guarda el identificador del campo, el identificador del valor y el nombre de los valores posibles para el campo.

La asociación entre las plantillas y sus campos está dada por la tabla tblMetadatosPlantillas, esta es una tabla recursiva que guarda el identificador del campo, el identificador del campo padres, para que se genera una jerarquía si es necesario, un valor que indica si es obligatorio o no y el orden en que se deben colocar los campos dentro del modelo. tblMetadatosDatosPlantillas guarda el valor como tal, que toma el

campo dentro de una plantilla, y se relaciona con tblSCO porque toda la información que se guarda es acerca de él.

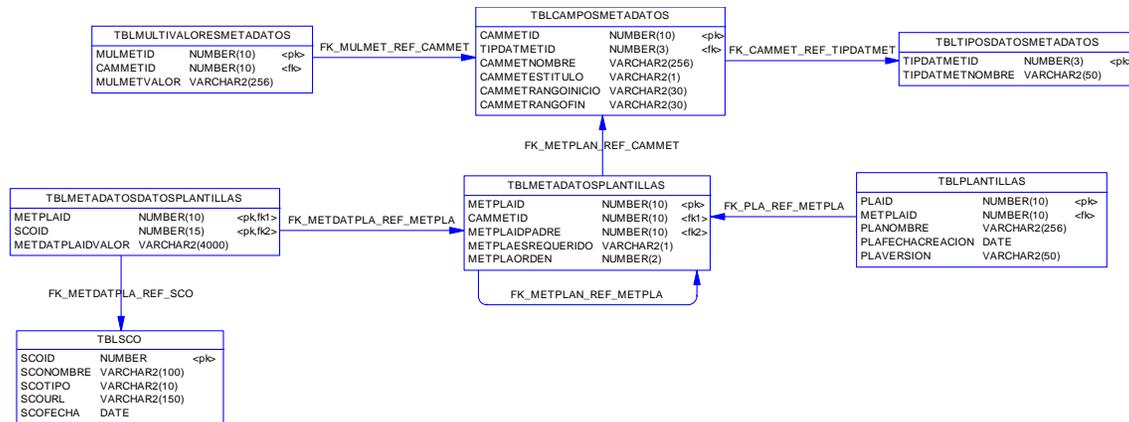


Figura 35 Modelo físico de los meta modelos para la generación de SCO

3.6.3 PROCESO DE EVALUACIÓN

En la Figura 36 se muestran las tablas que manejan las evaluaciones dentro del sistema, se utilizan las tablas tblPregunta, tblRespuestas, tblEvaluacion, tblResultadoPreguntas, y otras que representan asociaciones entre ellas.

tblPreguntas contiene la información de las preguntas colgadas en el sistema, como su identificador determinado por la secuencia seqPregunta, el tipo de pregunta, por ejemplo si es de selección múltiple o única y el texto de la pregunta, tiene además una referencia a tblUsuario que indica el docente que colgó la pregunta en el sistema, y una referencia a tblAgregacion que indica qué agregación abarca la pregunta. Las posibles respuestas a la pregunta se guardan en tblRespuestas, esta tabla guarda el identificador de la respuesta dado por la secuencia seqRespuesta, el texto de la respuesta y el valor de la respuesta, es decir, en el caso de una pregunta con opción única, si la respuesta es la acertada su valor será 100 y por último el campo que hace referencia a la pregunta a la que pertenecen las respuestas.

Cuando un docente genera una evaluación o el estudiante pide una auto evaluación, ésta se guarda en tblEvaluación y además de guardarse el tipo e examen, se almacena la fecha en que se publica o pide y la cantidad de preguntas que la prueba tiene. La asociación entre las preguntas y la evaluación está dada por tblEvaluacionPregunta.

tblEvaluacionUsuario es la tabla que asocia la evaluación y al usuario que la desarrolló y almacena además el resultado obtenido en la mismo. Por último se ve tblResultadoPreguntas, esta tabla relaciona una evaluación con las preguntas de la evaluación, que el estudiante las desarrollo y la calificación obtenida en las preguntas.

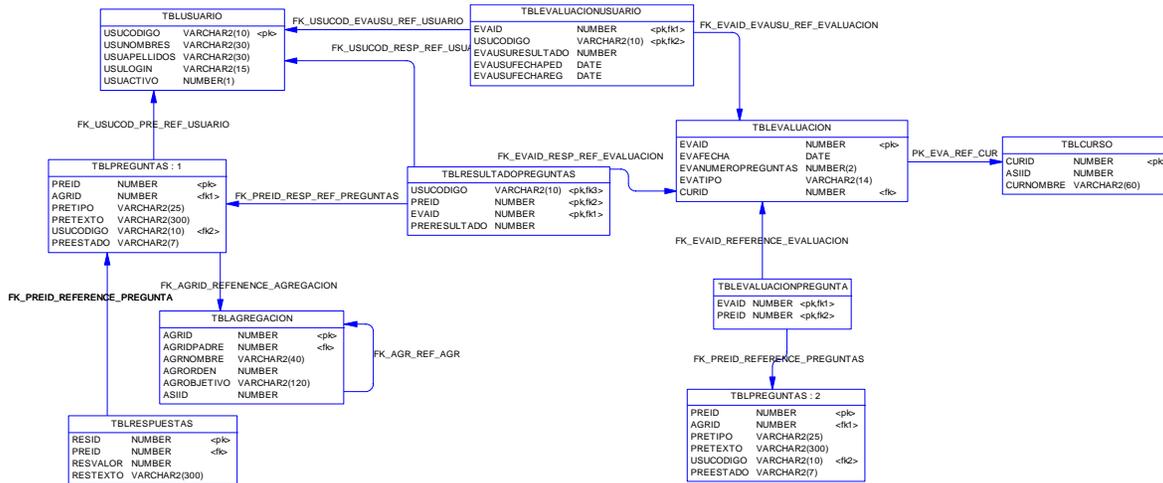


Figura 36 Modelo físico para el modelo de evaluación

3.6.4 FORO

Uno de los principales medios de interacción que provee EasyLearning es el foro (ver Figura 37). El modelo físico que lo representa está dado por tblForo, la cual es la tabla principal que guarda el identificador del foro determinado por la secuencia seqForo, el tema del cual trata el foro, la fecha de creación y cierre del foro y una referencia del curso al cual pertenece. La asociación entre el foro y los usuarios que tienen acceso al mismo está dado por tblForoUsuario quien tiene un campo adicional que indica el tipo de usuario dentro del foro, es decir, la persona que crea el foro tiene como valor en el campo tipo, administrador y es la única persona que puede hacer modificaciones en cuanto al nombre, o los usuarios que tienen acceso, estos últimos tienen en el campo tipo el valor usuarios y son personas que tienen acceso al foro y pueden colgar mensajes pero no lo pueden modificar. Por último tblMensajes, tabla recursiva que guarda todos los mensajes colgados en el foro, sus campos guardan el identificador del mensaje, la fecha en la que fue publicado, el título del mensaje, el texto, una referencia a tblUsuario que indica quien publicó el mensaje, el mensaje padre, dado el caso que sea una respuesta o comentario a otro mensaje y una referencia a tblForo que identifica el foro al que los mensajes pertenecen.

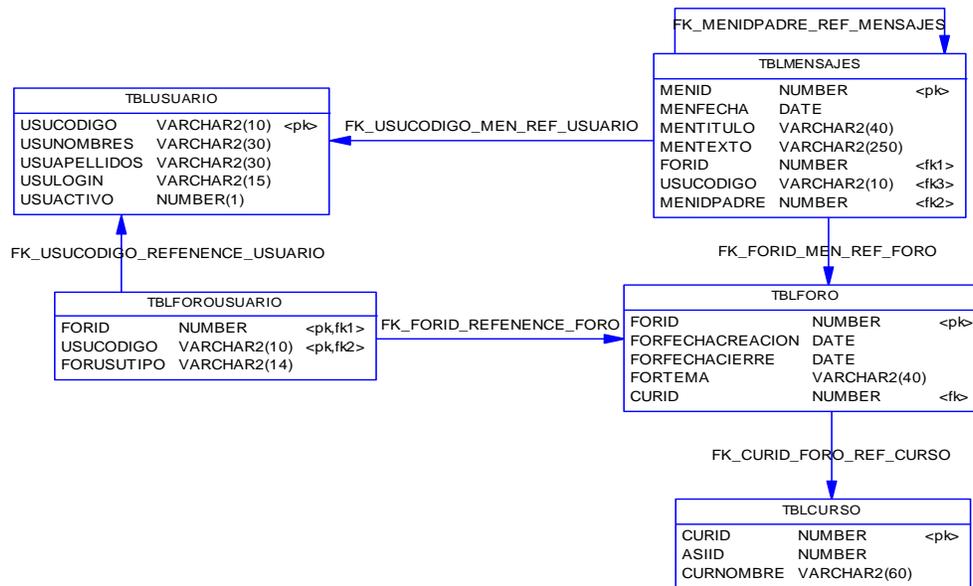


Figura 37 Modelo físico del foro

3.6.5 CHAT

El otro mecanismo de comunicación es el Chat, este tiene tres tablas que representan su modelo (ver Figura 38). `tblChat`, es la tabla que guarda el identificador del Chat o sala de Chat, la fecha de creación de la sala, el nombre y el identificador del curso al que pertenece. `tblConversaciones` almacena todos los mensajes enviados por el Chat, además de una referencia a `tblusuario` que indica la persona que envía el mensaje, tan bien hay un campo que define el tipo de mensaje enviado, público o privado, dado el caso que sea privado en `tblConversacionesPrivadas` se guarda el identificador del mensaje y el código de los usuario (s) destino del mensaje.

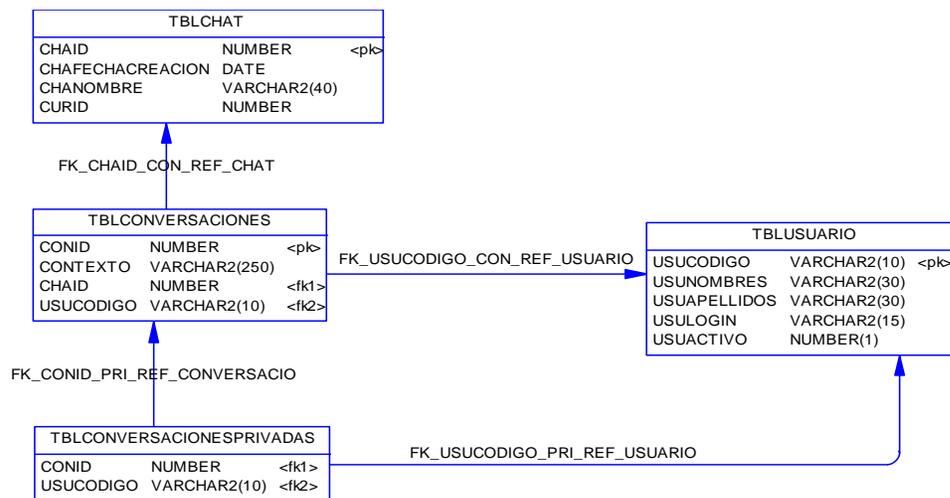


Figura 38 Modelo físico del Chat

3.7 PROBLEMAS ENCONTRADOS Y SOLUCIONES SELECCIONADAS

A continuación se explican los principales problemas que se tuvieron en el desarrollo de la aplicación:

Mantener el estado de la sesión de conexión con la base de datos en el dispositivo:

Con el propósito de poder realizar peticiones desde del cliente PPC a los servicios Web de la lógica del negocio, a través de la clase serFachada, es necesario mantener la conexión del usuario con la base de datos, para lo cual no hay soporte directo con el .Net CF, ya que carece del uso de cookies. Por lo cual fue necesario mantener el estado de la conexión con la base de datos de la siguiente forma:

1. Se modifica la propiedad cookieless de false (valor por defecto) a true del archivo web.config de los servicios Web XML.
2. Se hace uso de la clase System.Uri creando una instancia de esta con el fin de proporcionar una representación de objeto de un identificador de recursos uniforme (URI) y un acceso sencillo a las partes del identificador URI.
3. En el momento de realizar la primera petición al servicio Web de la clase serFachada se inicializa el objeto URI con la URL perteneciente al servidor de servicios Web.
4. Se invoca el método de inicio de sesión del servicio Web serFachada, el cual la primera vez que se intenta conectar, lanza una excepción del tipo WebException.

5. Si la excepción es del tipo `HttpWebResponse`, se verifica si el encabezado `Location` está asociado a la respuesta obtenida, si es así se redefine el objeto `Uri` asignándole la nueva localización especificada en la respuesta, de la siguiente forma:

```
objUri = new Uri(this.wsURL, resp.Headers["Location"]);
```

6. Finalmente se reinvoca el método de inicio de sesión (el que contiene el cuerpo de código que se muestra a continuación), para que redefina la `Url` del servicio `Web serFachada` con la `Uri` absoluta del objeto `Uri`.

A continuación se muestra una fracción del código fuente de la solución:

```
// Fijando la Url en el Proxy del WS serFachada
if (objUri == null)
    objUri = new Uri(WS.Url);
else
    WS.Url = objUri.AbsoluteUri;
try
{
    usuario = WS.Usuario_IniciarSession(Login, Password, "POCKET");
    blnConectado = true;
    ...
}
// Captura el "redirect" que sucede cuando hay una sesión sin cookies
// -> [cookieless sessions]
catch(WebException we)
{
    // Necesitamos una HttpWebResponse para chequear el código de
    // estado del HTTP
    if(typeof(HttpWebResponse).IsInstanceOfType(we.Response))
    {
        HttpWebResponse resp = (HttpWebResponse)we.Response;
        if(resp.StatusCode == HttpStatusCode.Found)
        {
            // Capturando la nueva localización especificada en la respuesta
            this.wsURL = new Uri( objUri, resp.Headers["Location"] );
            // Reinvocando a si mismo el método IniciarSession
            return iniciarSession( Login, Password, path );
        }
    }
    blnConectado = false;
}
```

Desligar la aplicación de una referencia Web estática: Cuando se trabaja con una aplicación que es cliente de servicios Web (que consume servicios Web), cabe la posibilidad que los servicios Web cambien de servidor, lo que obligaría a que se recompilara y distribuyera de nuevo la aplicación cliente. Con el propósito de evitar esto es necesario desligar la aplicación de una referencia Web que sea estática en el código,

lo cual es posible solucionar a través del método estático AppSettings de la clase System.Configuration.ConfigurationSettings en el .Net Framework, que es capaz de obtener la información contenida en un archivo XML de configuración. El inconveniente radica en que la versión actual del .Net CF (v1.0) no tiene una implementación para esta clase, por lo que para solucionar el problema se hizo uso del Open Net CF, un framework de aplicación que enriquece y extiende el .Net compact Framework, y que si tiene una implementación de la clase ConfigurationSettings, con lo que ya es posible hacer uso de un archivo de configuración XML en el que se define la URL donde se alojan los servicios Web, se obtiene y asigna dicha información a la URL del servicio Web, como se muestra en el marco teórico.

Mantener e informar acerca del estado de la conexión con el servidor: Debido a que no solo es necesario mantener el estado de la sesión de conexión con la base de datos en el dispositivo, sino también establecer cuando esta conexión se pierde por razones como: caída del servidor, culminación de la sesión, problemas en la conexión; es pertinente chequear continuamente el estado en que se encuentra la conexión con el servidor. Lo que es posible realizar a través de un control Timer que se active cada determinado tiempo, en el caso de la aplicación cada minuto, e invoca un método que verifica el estado de la conexión y la sesión, notificando su resultado a la interfaz a través de un evento.

Descargar los SCOs (contenidos): Debido a que los SCOs no son siempre un simple archivo de texto, si no que pueden abarcar diferente cantidad de archivos en variados formatos, es necesario entonces definir la forma en que sea posible descargarlos hasta el dispositivo. Lo cual se hace con un método Web de los servicios Web de lógica del negocio que implemente esta funcionalidad, como se muestra a continuación:

```
[WebMethod(true)]
public Byte[] retornarArcRecursos(string nombrearc)
{
    // Estableciendo la ruta donde se encuentra el archivo en el servidor
    string ruta = Server.MapPath(".") + @"\" + nombrearc;

    // Conteniendo el archivo en un flujo
    FileStream flujo = new FileStream(ruta, FileMode.Open,
                                     FileAccess.Read, FileShare.Read);

    // Leyendo el archivo desde el flujo a través de un lector binario
```

```

BinaryReader lector = new BinaryReader(flujo);

// Creando un arreglo de bytes con la longitud del flujo
Byte[] datos = new Byte[int.Parse(flujo.Length.ToString())];

// Almacenando el archivo en un arreglo de bytes
datos = lector.ReadBytes(int.Parse(flujo.Length.ToString()));

// Cerrando el flujo
flujo.Close( );

// retornando el archivo en un arreglo de bytes
return datos;
}

```

Cuando se invoque este método a través del servicio Web que lo implementa se hará entonces el proceso contrario para obtener y guardar el SCO.

Visualizar los SCOs (contenidos): Debido a que el formato de los SCOs puede variar se requiere de un visor que los soporte, y que se pueda incrustar en las formas de la aplicación, con el fin de que el usuario no requiera de pasos adicionales para la visualización de los recursos que descarga. El .Net CF no cuenta con control *ActiveX Explorador del Web de Microsoft* con el que cuenta el .Net Framework, el cual sería ideal para la visualización de los SCOs, así que para poder utilizar un visor de este tipo, se hizo uso del control *HTMLViewer* [27] creado por Intelliprolog, el cual se incrusta visualmente en las formas y cuenta con la funcionalidad necesaria para la visualización.

Hacer Data Binding a controles que implementen el evento SelectedIndexChanged:

La utilización de data binding es un método muy efectivo cuando se realiza carga de datos en controles gráficos. El problema viene cuando estos controles implementan el evento *SelectedIndexChanged*, debido a que en el momento que se inicia la carga de los datos en el control se dispara automáticamente este evento, si se ha programado previamente. La solución está en realizar primero toda la carga de datos y luego implementar el evento, como se muestra a continuación.

```

// Realizando DataBinding al ComboBox cmbBxAsignaturas
// Asignando la fuente de datos
cmbBxAsignaturas.DataSource = dsAsignaturas.Tables["tblAsignaturas"];

// Asignando la propiedad del origen de datos a mostrar
cmbBxAsignaturas.DisplayMember = "ASINOMBRE";

```

```

// Asignando la propiedad del origen de datos a partir de la cual se
// obtendra el valor a mostrar
cmbBxAsignaturas.ValueMember = "ASIID";

// Seleccionando el Item 0
cmbBxAsignaturas.SelectedIndex = 0;

// Asignando el manejador del evento SelectedIndexChanged
// del control ComboBox cmbBxAsignaturas
cmbBxAsignaturas.SelectedIndexChanged += new System.EventHandler(
    cmbBxAsignaturas_SelectedIndexChanged);
cmbBxAsignaturas_SelectedIndexChanged(null, null);

```

3.8 RECOMENDACIONES TÉCNICAS

3.8.1 MANEJO DE DATOS

- Si se va a mantener un gestor de base de datos constante y la posibilidad de cambiarlo es prácticamente nula, sería recomendable dejar algunas tareas al gestor utilizando triggers y procedimientos almacenados, ya que esto agiliza el proceso de las tareas y minimiza la carga y peticiones al servidor. Así mismo, una buena opción sería implementar el repositorio local del dispositivo como una solución propietaria (por ejemplo Oracle Database Lite) que lo mantenga y haga la sincronización respectiva con el sistema central.
- Sacar mayor provecho de las potencialidades que ofrece el objeto DataSet de ADO.NET para realizar tareas en el dispositivo, debido a que este representa una colección de tablas y se las puede manipular como tal, por ejemplo en el caso de la gestión realizada sobre datos en archivos XML.

3.8.2 CONEXIÓN Y COMUNICACIÓN

- Una forma de garantizar que los mensajes que se envíen entre el cliente PPC y el cliente Windows a la lógica del negocio sería la implementación de Microsoft Windows Message Queuing, en las peticiones y respuestas que se hacen de un lado a otro. Con lo que además se garantizaría: priorización de mensajes, capacidades de trabajo fuera de línea, mensajes transaccionales y seguridad, de ser necesario.
- Hacer uso de técnicas avanzadas de utilización de servicios Web, como es el caso de la especificación hecha por la Arquitectura XML Global (Global XML Architecture, GXA), a partir de esta se generó el paquete Web Service Enhancement (WSE), que en su versión 2.0, el cual es una librería de clases para construir servicios Web usando

los últimos protocolos de servicios Web, incluyendo: WS-Security, WS-SecureConversation, WS-Trust, WS-Policy, WS-SecurityPolicy, WS-Addressing, y WS-Attachments [49].

3.8.3 INTERFAZ

- Una técnica que actualmente se esta madurando en cuanto al manejo de interfaz de aplicaciones para clientes inteligentes es el reconocimiento de comandos de voz, la cual esta teniendo mucho éxito, debido a que es ideal para dispositivos cuya entrada de datos es limitada, la que entonces sería ideal en el ambiente del cliente PPC.
- Es conveniente ajustar la interfaz gráfica del cliente PPC a distintos tamaños y orientaciones de pantalla, redefiniendo en el evento resezing de las clases de los formularios, la forma y cada uno de los controles.
- Dependiendo de donde se utilice la aplicación sería conveniente implementar diferentes idiomas en los menús, lo que se conoce como aplicaciones internacionales, esto implica utilizar técnica de globalización y localización.

3.8.4 ARQUITECTURA

- Para un mejor manejo de los formularios Windows de la aplicación cliente PPC, podría implementarse una clase que controle y administre la carga, visualización y despliegue de las formas.

3.8.5 ACTUALIZACIONES Y DESPLIEGUE

- Realizar las actualizaciones y despliegue del cliente Windows a través de técnicas como No-touch deployment. La cual es una aproximación a realizar copia de la aplicación en un servidor Web, que luego el .Net Framework automáticamente descargará junto con sus ensamblados dependientes, dando un simple a clic a un enlace de una pagina Web.

3.8.6 SEGURIDAD

- Mantener un registro de las actividades y eventos del usuario realizadas en el sistema, con el fin de monitorearlos y tomar medidas de lo que ocurra en el sistema, como es el caso de perdida de información, ataques de virus, entre otros.

4 CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

4.1 CONCLUSIONES

- El mLearning es una realidad que puede ampliar el tiempo potencial de aprendizaje del que disponen los estudiantes. Además, da la posibilidad de usarlo en ambientes integrados de eLearning o como un ambiente aislado de aprendizaje (standalone).
- Es posible definir un conjunto de servicios que son adecuados de implementar en mLearning y otros que no lo son tanto, para ello es preciso tener en cuenta las características de los dispositivos y las implicaciones pedagógicas, teóricas y prácticas.
- La conexión intermitente a Internet (o a una Intranet) que se tiene en redes inalámbricas (por su costo, o por su alcance), la poca capacidad de almacenamiento y el despliegue limitado que se obtiene en las pantallas son las tres limitaciones más importantes de los dispositivos móviles en relación con su uso en los procesos de aprendizaje.
- Tener claras las limitaciones de los dispositivos móviles y sus ventajas u oportunidades son la base para la construcción de aplicaciones de mLearning que realmente brinden un valor agregado a los estudiantes y profesores en sus procesos de enseñanza y aprendizaje.
- Implementar una arquitectura dividida en capas bien definidas facilita la implementación en paralelo de la aplicación por parte del equipo de desarrollo, aumentando la escalabilidad de esta y disminuyendo la posibilidad de grandes errores al finalizar los ciclos de desarrollo.
- El uso de SCORM como estándar de Generación de Contenidos fortalece el alcance de la aplicación de mlearning, debido a que el uso del estándar potencia la reutilización de recursos de aprendizaje, no solo entre los contenidos del mismo sistema, sino también facilita el intercambio con otros repositorios de contenidos.

- El aprendizaje móvil es una nueva forma de acercar las tecnologías de la información al servicio de la educación y los procesos de aprendizaje. Tiene enormes ventajas que la hacen un campo de exploración, investigación y desarrollo muy importante en la tele educación o la educación electrónica. Es de especial interés del aprendizaje móvil, su mayor posibilidad de aumentar el tiempo potencial para realizar procesos de enseñanza-aprendizaje.
- Muchos de los contenidos y actividades de aprendizaje que se han venido desarrollando en los últimos años en el aprendizaje en línea, se pueden y deben usar en el marco del aprendizaje móvil, pero para esto es preciso tener en cuenta las actuales limitaciones de visualización, digitación y conectividad.
- Easy Learning es un sistema de gestión de aprendizaje móvil para Pocket PC que promete ser muy útil para llevar a cabo procesos de enseñanza-aprendizaje basado en contenidos que pueden llevarse a cabo en forma autónoma y actividades de aprendizaje autónomo y colaborativo.
- Con el uso de herramientas de desarrollo como Visual Studio .Net, se aprovecha el potencial de los desarrolladores enfocándolos en una solución ingenieril del problema y no en una solución enfocada a la implementación.
- En nuestro contexto, fue importante aplicar la técnica Divide y Vencerás, planteando el desarrollo a partir de módulos funcionales, atacando así la solución en subproblemas. Esto permitió obtener resultados rastreables a corto plazo, y al final del proceso, en la unión de las subsoluciones, alcanzar los objetivos planteados

4.2 RECOMENDACIONES

- Debido a las características actuales de los dispositivos móviles, sería apropiado utilizar, o en su defecto generar, una herramienta que permita crear recursos de aprendizaje que se adecuen a los mismos sin complicaciones y que además se obtenga un mayor provecho de las capacidades del dispositivo.

- Con el propósito de cumplir uno de los objetivos de un Sistema Gestor de Aprendizaje es necesario implementar reglas de secuenciación para el desarrollo de los procesos de aprendizaje del estudiante en el dispositivo móvil, con el propósito de llevar un completo registro de su proceso y rendimiento académico.
- Debido a que el sistema parte de los requerimientos establecidos para los LMSs actuales, es decir, no cuenta con todos los servicios que dentro del mLearning se pueden utilizar, se recomienda adicionar funcionalidades como el servicio de agenda para los estudiantes y docentes o el desarrollo de actividades colaborativas, entre otras, que den un mayor valor agregado a la aplicación. ANEXO A
- Debido a las limitadas capacidades de espacio y resolución en pantalla de los dispositivos móviles actuales, es recomendable recurrir a un diseñador gráfico experto en buenas prácticas de diseño de interfaces gráficas para la interacción humano – dispositivo móvil.
- Un factor muy importante consiste en implementar los requerimientos necesarios para controlar y asegurar la calidad en la creación de contenidos educativos, en donde existan grupos interdisciplinarios de trabajo que no sólo aportan en el desarrollo de las unidades de conocimiento, si no también en la forma en que ellas se deben entregar a los estudiantes.
- Es importante realizar las gestiones necesarias para registrar los derechos de autor de las aplicaciones que se desarrollan como proyecto de grado, pues muchos de estos pueden servir como proyectos de emprendimiento de empresas, lo que permite asegurar su continuidad.
- Las evaluaciones son un elemento crucial dentro del seguimiento al desarrollo de procesos de aprendizaje de los estudiantes, puesto que de una u otra forma reflejan el nivel en el que se encuentran; es por esto que se recomienda que el proceso de creación de evaluaciones, su ponderación, la correcta estructuración y clasificación de

las preguntas se realicen bajo el seguimiento de la especificación Preguntas e Interoperabilidad de Pruebas (Question and Test Interoperability, QTI)⁶.

- Con el propósito de controlar el acceso a los contenidos y recursos de los cursos, se recomienda dar un periodo de vigencia a los mismos.

4.3 TRABAJO FUTURO

- Realizar la experimentación de la aplicación para medir el grado de satisfacción de los usuarios en los procesos de aprendizaje que realicen, y confrontar si en realidad cumple con las características planteadas por un Sistema de Gestión de Aprendizaje.
- Continuación con la investigación en mLearning en la segunda fase del proyecto titulado “Propuesta Curricular para la formación de Ingenieros desde el Enfoque en Estudios CTS + I en la Universidad del Cauca”, que ya esta aprobado por la Universidad del Cauca y Colciencias.

⁶ IMS Global Learning Consortium. <http://www.imsglobal.org/question/index.cfm>

5 REFERENCIAS BIBLIOGRAFICAS

- [1] From eLearning to mLearning. Ericsson. 2001.
<http://learning.ericsson.net/leonardo/thebook/chapter1.html>.
- [2] Quinn, C. (2000). mLearning: Mobile, Wireless, In-Your-Pocket Learning. Line zine. Learning in the new economy <http://www.linezine.com/2.1/features/cqmmwiyp.htm>
- [3] McManus, T. F. (2002). Exploring Mobile Learning Environments. Research Unit for Education Technology. University of Oulu. <http://edtech.oulu.fi/edtech/35ov01-02/mlearning/readings.htm>.
- [4] Advanced Distributed Learning Home Page. <http://www.adlnet.org>. The SCORM Overview and Content Aggregation Model. Version 1.2
- [5] Advance Distributed Learning (ADL) Initiative – The SCORM Overview v1.2, Octubre 1 – 2001, pag. 1-21
- [6] Advance Distributed Learning (ADL) Initiative – The SCORM Overview v1.2, Octubre 1 – 2001, pags. 1-3, 1-21
- [7] AdvanceWork Presenta Primer Curso Móvil de Inglés mediante Pocket PC de Microsoft. SYRACUSE, Nueva York 26 de septiembre de 2001.
<http://www.advancework.com/english/press/092601SPANVSN.htm>
- [8] Going Mobile: New Technologies in Education. John Fleischman. 2001.
<http://www.convergemag.com/magazine/story.phtml?id=2530000000001969>
- [9] Instrucción basada en la Web. Dra. Silvia C. García Urrea. Venezuela. 2001.
http://www.unesr.edu.ve/Publicaciones/Articulo_2.htm
- [10] Going Mobile. Paul Harris. Learning Circuits. 2001.
<http://www.learningcircuits.com/2001/jul2001/harris.html>
- [11] LARMAN, Craig. UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos. Ed. Prentice Hall. México, 1999. Pag. 536.
- [12] RESTREPO, Bernardo. Conceptos y Aplicaciones de la Investigación Formativa, y Criterios para Evaluar la Investigación científica en sentido estricto. Consejo Nacional de Acreditación. http://www.cna.gov.co/cont/doc_aca/index.htm.
- [13] COBOS, Carlos; Niño, Miguel Ángel; Mendoza Martha Eliana; Molina, Luis, Chávez Wilson, Ardila Mario. Unicauca Virtual Fase I: Estandarización y el Método Funcional.

- [14] García, S. (2001). Instrucción basada en la Web. Venezuela. http://www.unesr.edu.ve/Publicaciones/Articulo_2.htm.
- [15] Palm Applications in Education. The Concord Consortium. http://pie.concord.org/index.php3?PALM_Session=147517deef19645824969c2421dace4c.
- [16] Malinen, J., Kari, H. & Tiusanen, M. (1999). Wireless networks and their impact on network-based learning content. Enabling network-based learning. Helsinki University of Technology. <http://www.enable.evitech.fi/enable99/papers/malinen/malinen.html>.
- [17] Fleischman, J. (2001). Going Mobile: New Technologies in Education. <http://www.convergemag.com/magazine/story.phtml?id=2530000000001969>.
- [18] Ramsden, A. Evaluating a Low Specification Wirelessly Connected PDA as a Means Supporting Learning. Learning Technology Support Service. <http://www.ltss.bristol.ac.uk>.
- [19] WIGLEY, Andy; WHEELWRIGHT, Stephen. Microsoft .Net Compact Framework (Core Reference). Ed. Microsoft Press. Redmond, Washington, 2003. Pag. 860.
- [20] BUHLER, Erich. Visual Basic .Net, Guía de Migración y Actualización. Ed. MCGRAW-HILL. Interamericana de España S.A.U. Pag. 976.
- [21] STOECKER, Matthew. Developing Windows-Based Applications with Microsoft Visual Basic .Net y Visual C# .Net. Ed. Microsoft Press. Redmond, Washington, 2003. Pag. 585.
- [22] WEBB, Jeff. Developing Web Applications with Microsoft Visual Basic .Net y Visual C# .Net. Ed. Microsoft Press. Redmond, Washington, 2002. Pag. 864.
- [23] PROSISE, Jeff. Programming Microsoft .NET eBook (Core Reference). Ed. Microsoft Press. Redmond, Washington, 2002. Pag. 816.
- [24] Programa Desarrollador 5 Estrellas DCE – Microsoft Corporation Latinoamérica. <http://www.microsoft.com/spanish/msdn/comunidad/dce/default.asp>
- [25] Microsoft .NET Compact Framework QuickStart Tutorial – Got Dot Net <http://samples.gotdotnet.com/quickstart/CompactFramework/>
- [26] Open Net Compact Framework <http://www.opennetcf.org/CategoryView.aspx?category=Home>
- [27] Intelliprogram Inc - Products for .NET Compact Framework – HTMLViewer <http://www.intelliprogram.com/netcf/index.html>
- [28] Smart Client Developer Center <http://msdn.microsoft.com/smartclient>

- [29] Quality Login - Pocket PC Application Handbook – Designed for Windows Mobile.
http://www.qualitylogic.com/certprograms/pocketpc_spec.html
- [30] Qué son los Servicios Web XML?
<http://www.microsoft.com/latam/net/basics/xmlservices.asp>
- [31] KIRTLAND, Mary. Una plataforma para servicios Web. Enero de 2001.
http://msdn.microsoft.com/library/spa/default.asp?url=/library/spa/dntaloc/html/websvcs_platform.asp
- [32] Modelo de código para servicios Web XML en código administrado.
http://msdn.microsoft.com/library/spa/default.asp?url=/library/spa/vbcon/html/vbconcod_emodelforwebservicesinmanagedcode.asp
- [33] Formato de los archivos de configuración de ASP.NET.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconaspnetconfiguration.asp>
- [34] Modelo de código para obtener acceso a servicios Web XML en código administrado.
http://msdn.microsoft.com/library/spa/default.asp?url=/library/spa/vbcon/html/vbconcod_emodelforaccessingwebservicesinmanagedcode.asp
- [35] Generar clientes de servicios Web XML.
http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpcon_BuildingASPNETWebServiceClients.asp
- [36] Introducción a la administración de estados de ASP.NET.
<http://support.microsoft.com/default.aspx?scid=kb:es:307598#2>
- [37] Obtener acceso a un servicio Web XML en código administrado.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/spa/vbcon/html/vbtskcreatingclientsforwebservices.asp>
- [38] Microsoft .NET Compact Framework Background Processing Techniques.
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetcomp/html/CompactfxTechArt.asp>
- [39] Acceso a datos con ADO.NET. Guía del desarrollador de .Net Framework.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconaccessingdatawithadonet.asp>
- [40] Introducción al acceso a datos con ADO.NET.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcon/html/vbconBenefitsOfADO.asp>

- [41] Ventajas de ADO.NET.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcon/html/vbconBenefitsOfADO.asp>
- [42] Arquitectura de ADO.NET.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconwhyadonet.asp>
- [43] XML y DataSet.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconsynchronizingdatasetwithxmladatadocument.asp>
- [44] DiffGrams.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconwritingdatasetasxml.asp>
- [45] Cargar DataSet desde XML.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconloadingdatasetfromxml.asp>
- [46] Escribir un objeto DataSet como datos XML.
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconloadingdatasetfromxml.asp>
- [47] Feature Overview Oracle data provider for .Net. Agosto 2002.
<http://www.oracle.com/technology/tech/windows/odpnet/ODP.NET-FOV.html>
- [48] HARRINGTON, Gregg D, LIPSCHUTZ, Robert P. Accessing Oracle with ODP.NET. Marzo de 2003. <http://www.oracle.com/technology/oramag/oracle/03-mar/o23odp.html>
- [49] Web Services Developer Center - Web Services Enhancements (WSE)
<http://msdn.microsoft.com/webservices/building/wse/default.aspx>