



BENEMÉRITA UNIVERSIDAD

AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

“Objetos de Aprendizaje Móvil”

Tesis Profesional que para obtener el título de:

Ing. en Ciencias de la Computación

Presenta:

Felipe Hernández Rosales

Asesor:

Dr. Fernando Zacarías Flores

Puebla, Pue.

Otoño 2009

INDICE

	Pág.
INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA	3
JUSTIFICACIÓN	4
OBJETIVOS	5
CAPÍTULO 1: OBJETOS DE APRENDIZAJE	6
Definición de objetos de aprendizaje	6
1.2 Propiedades de los objetos de aprendizaje	8
1.3 Enseñanza basado en objetos de aprendizaje	10
1.3.1 Compartir recursos	11
1.4 Ventajas de los objetos de aprendizaje	14
1.5 Objetos de aprendizaje en el desarrollo docente	16
CAPÍTULO 2: NETBEANS EN EL DESARROLLO DE APLICACIONES	19
2.1 NetBeans	19
2.2 NetBeans y módulo para movilidad	20
2.3 Tecnología móvil	21
2.4 Implicaciones de la tecnología móvil	22
2.5 J2ME y NetBeans	24
2.5.1 Componentes de J2ME	24
2.5.2 Máquina virtual	25
2.5.3 Configuración	25
2.5.4 Perfil	26
2.6 Arquitectura J2ME	27
2.7 MIDlet	28
2.8 GPRS	31
2.8.1 Ventajas del GPRS para el usuario	32
2.9 Conexión a redes	32
2.9.1 Entrada/Salida desde el MIDlet	35
2.9.2 La clase InputStream	36
2.9.3 La clase OutputStream	37
2.9.4 Haciendo conexión	37
2.9.5 Obtener un Input Stream	38
2.9.6 Leyendo desde el Stream	38

CAPÍTULO 3: OBJETOS DE APRENDIZAJE MÓVILES	40
3.1 Revolución móvil	41
3.2 Objetos de aprendizaje y e-Learning.....	41
3.3 Objetos de Aprendizaje Móviles y m-Learning.....	42
3.4 El estándar SCORM de la ADL.....	43
3.5 Los Sistemas de Gestión de Aprendizaje o LMS	45
3.6 Editores de contenido	46
Capacidades y características	47
Implicaciones pedagógicas	48
Aplicación Móvil	49
3.10 Especificaciones Técnicas	51
CAPÍTULO 4: DESARROLLO DE APLICACIÓN WEB	53
4.1 Concepto de Servlet	53
4.2 Ventajas de los Servlets	54
4.3 Arquitectura de los Servlets	55
4.4 Ejemplos de aplicación de los Servlets	59
CAPÍTULO 5: REPOSITORIO DE OBJETOS DE APRENDIZAJE MÓVIL	60
5.1 Antecedentes	60
5.2 Servidor Web	60
5.3 Requisitos de Hardware y Software	63
5.4 Construcción del repositorio	64
5.5 Funcionalidades del repositorio	66
5.6 Resultados	69
CONCLUSIONES Y TRABAJO A FUTURO	70
BIBLIOGRAFÍA	72

INTRODUCCIÓN

El presente trabajo pretende explotar el potencial que las tecnologías de la información y comunicaciones nos ofrecen actualmente. De manera particular pretendemos llevar el conocimiento a la mano de cualquier persona que desee involucrarse en un aprendizaje continuo y permanente (Life-Long-Learning en inglés) a través de objetos de aprendizaje móviles.

El desarrollo de los objetos de aprendizaje móviles simples facilita el aprendizaje, ya que pueden ser más atractivos si ofrecen la posibilidad de ser vistos utilizando diferentes dispositivos móviles, desde un teléfono móvil, por ejemplo.

Se empleará como plataforma de desarrollo Netbeans y el kit de desarrollo Wireless Toolkit de Sun Java™. Además, se desarrollarán objetos de aprendizaje de acuerdo a estándares SCORM de la ADL, y así permitir a los usuarios la descarga algún curso como una aplicación Java según los recursos del dispositivo.

Se espera que los objetos de aprendizaje móviles lleguen a más aprendices, aprovechando el auge que los teléfonos móviles han provocado en los últimos días, pues de entrada contaremos con 55 millones de usuarios potenciales (en México), es decir, el desarrollo de objetos de aprendizaje móvil podrá estar disponible en la mano (teléfono móvil) de cualquier aprendiz interesado en su educación.

Existen diversas propuestas para llevar el conocimiento a todas las personas de nuestro país. El motivo principal, lograr que la gente aprenda, adquiera conocimiento y genere ideas. Se han planteado propuestas en nuevos medios, tal es el caso del libro electrónico. Con esta posibilidad ha sido posible llevar este nuevo tipo de libros a diversos contextos. Algunos de estos son: Libros electrónicos en dispositivos especialmente diseñados para este fin, pero que resultan costosos. Libros electrónicos en bases de datos públicas, las cuales tiene un costo de membresía o bien un costo por descarga. También, existen librerías digitales las cuales venden libros de manera impresa (tradicional) o de manera digital (a través de la descarga desde internet). Sin embargo, todas

estas propuestas presentan algunos inconvenientes, ya que son vistas como actividades ajenas o extras a nuestras actividades cotidianas. Por tal razón, nuestro propósito es el poder llevar ya sean libros digitales, audio-libros y videos de aprendizaje como objetos de tal forma que nos permitan el intercambio de contenidos, la formación de nuevos cursos de manera rápida y sencilla. Además, el traerlos en dispositivos móviles que ya son parte de nuestras actividades cotidianas, como lo es el teléfono móvil, amplía el horizonte de aplicación y uso en el aprendizaje continuo y básico de la gente. Este dispositivo ya forma parte de los accesorios básicos y necesarios en el trabajo, en el estudio, en el hogar, para la diversión, etc. Por lo que es importantísimo poder agregarles un plus en su uso, convirtiéndolo en un medio de aprendizaje.

PLANTEAMIENTO DEL PROBLEMA

¿Cuántas personas no estudian por considerar que no tienen tiempo para asistir a la escuela, centro de capacitación, inscribirse, costos elevados, etc.?

Para responder a esta pregunta sólo basta con conocer las estadísticas sobre el número de usuarios de telefonía celular o móvil. Hasta el año 2007, las estadísticas de usuarios de una sola compañía indicaban 55 millones de usuarios. Esto significa que el desarrollo de libros electrónicos tiene el potencial para llegar a todos estos millones de usuarios cautivos y muy probablemente, ávidos de conocimiento.

Por consiguiente, es de vital importancia el desarrollo de este trabajo de tesis, con el objetivo principal de contar con tecnología que permita explotar los dispositivos móviles que de antemano el usuario ya cuenta con ellos y, no será necesario que adquieran este tipo de dispositivos.

JUSTIFICACIÓN

Uno de los problemas actuales en nuestro país es la falta de preparación en todos los niveles: básica, media, superior y de posgrado. El proyecto propuesto en la presente tesis tiene como objetivo, brindar la oportunidad a todos los usuarios de telefonía móvil el beneficio de traer en su teléfono objetos de aprendizaje que les permitan experimentar un aprendizaje continuo. La idea general es la siguiente: Hacer parte de su vida cotidiana la disponibilidad de material de auto-aprendizaje de tal forma que puedan estudiar en todo momento y en cualquier lugar. Para esto, nuestra aplicación constará de objetos de aprendizaje que conformen un curso básico de aprendizaje. De esta manera, cualquier usuario de telefonía móvil podrá llevar su material ya sea audio-libro, video o libro de texto en su teléfono con los beneficios que este tipo de tecnologías brinda.

OBJETIVOS

Objetivo General

- Demostrar la viabilidad de los Objetos de Aprendizaje en Dispositivos Móviles.

Objetivos Específicos

- Implementar una aplicación móvil (MIDP) para realizar las operaciones de búsqueda y descarga de Objetos de Aprendizaje Móviles (OAm).
- Construir una base de datos de OAm.
- Construir una aplicación web (SERVLET) la cual dará respuesta a las consultas generadas desde la aplicación móvil y a su vez proporcionará los OAm a descargar. Servirá como intermediario en la comunicación entre la Aplicación Móvil y la Base de Datos.
- Implementar la comunicación entre las peticiones de la Aplicación Móvil y la aplicación Web vía WAP-GPRS.
- Construir un Repositorio de Objetos de Aprendizaje Móvil.
- Diseño de Objetos de Aprendizaje Móviles.

Capítulo 1:

OBJETOS DE APRENDIZAJE

Uno de los problemas más viejos y complejos en el contexto de la enseñanza es el correspondiente a la generación de material didáctico, es decir, la construcción de libros, tutoriales, sistemas interactivos, etc. Sin embargo, la construcción de estos materiales es muy costosa y exhaustiva. Estos inconvenientes han dado origen a uno de los nuevos paradigmas llamado “objetos de aprendizaje”. Este nuevo paradigma intenta atacar el problema del costo, de lo complejo de su construcción y algunos otros problemas colaterales que conlleva la construcción de materiales didácticos.

1.1 Definición de Objeto de Aprendizaje

Una de las definiciones que más se han adoptado es la debida a Wiley. Escribe Wiley, [1, 2] que el término objetos de aprendizaje se debe al trabajo que realizó la CedMA en el año 94 [9], dirigido entonces por Wayne Hodgins. Dicho trabajo llevaba por título “*Learning Architectures, API’s, and Learning Objects*”.

El término “objeto de aprendizaje” (u OA como la denominaremos de aquí en adelante) resulta controvertida en tanto que el concepto adoptado por Hodgins resultó excesivamente amplio. El planteamiento inicial de la CedMA fue describir los objetos de aprendizaje como entidades, digitales o no digitales, que pudieran ser utilizadas, reutilizadas y consultadas durante el aprendizaje mediado por ordenador.

La CedMA considera como OA a:

- Un sistema de formación basado en la computadora (CBT: Computer- based training)
- Un ambiente de aprendizaje interactivo
- Sistemas inteligentes, de ayuda, y basados en el ordenador
- Sistemas de aprendizaje a distancia
- Ambientes de aprendizaje colaborativo

El LTSC (Comité de Estándares norteamericano de gran relevancia e influencia) decide unificar el concepto y acuerda que grupos externos al Comité desarrollen descripciones en torno al término para tratar de estrechar así su ámbito de aplicación. Lo que en su inicio pareció una solución no lo es cuando el LTSC incorpora no sólo la definición de Hodgins sino todas las descripciones dadas haciendo confuso el término y difícil su comunicación [1, 4]. El LTSC entrega la siguiente definición:

“Un Objeto de Aprendizaje es cualquier entidad, digital o no digital, la cual puede ser usada, re-usada o referenciada durante el aprendizaje apoyado por tecnología. Ejemplos de aprendizajes apoyados por tecnologías incluyen sistemas de entrenamiento basados en computador, ambientes de aprendizaje interactivos, sistemas inteligentes de instrucción apoyada por computador, sistemas de aprendizaje a distancia y ambientes de aprendizaje colaborativos. Ejemplos de Objetos de Aprendizaje incluyen contenidos multimedia, contenido instruccional, objetivos de aprendizaje, software instruccional y herramientas de software, y personas, organizaciones, o eventos referenciados durante el aprendizaje apoyado por computador.” [15]

Uno de los efectos contrarios que se genera tras la delimitación de los OA es el considerarlos, de forma restringida, como elementos que describen características de programas informáticos o sencillamente *applets* Java, tal como reconocen las empresas NETg.INC o Asymetrix.

El otro problema añadido al término OA se debe al no acuerdo terminológico, es decir, se llama de diferente manera a algo que es similar. Así, Merrill [5] habla de componentes de conocimiento y objetos de conocimiento, Boyle [6] de contextos de aprendizaje, el proyecto europeo ARIADNE de documentos pedagógicos o la empresa Cisco de RIOs, (Objetos de Información Reutilizables).

En la tabla 1.1 exponemos un listado de términos que guardan relación con el OA. Además, en la figura 1.2 se muestra la conceptualización de lo que concebimos como un objeto de aprendizaje.

D. ERRIL (2001)	Objeto de conocimiento Componente de instrucción Componente instruccional
Proyecto ARIADNE	Documento pedagógico
Proyecto ESCOT	Componentes de Programas educacionales
Proyecto MERLOT	Material de aprendizaje en línea
ALI, de Apple	Recursos
BARRON (2000)	RIOs: Objetos de Información reutilizables

Tabla 1.1 Objetos de aprendizaje. Términos relacionados

Wiley [1, 2] indica que hablar de objetos de aprendizaje nos conduce hacia nuevos elementos en el terreno de la instrucción basada en el ordenador (CBI) y en la orientación de objetos. El diseño orientado a objetos es aquel en el cual se definen inicialmente las entidades prototipo y posteriormente se repiten y utilizan como parte de un programa informático [7]. La orientación de objetos implica la creación de componentes (llamados “objetos”) que pueden ser reutilizados en múltiples contextos de aprendizaje.



Figura 1.2 Conceptuación de un objeto de aprendizaje

1.2 Propiedades de los Objetos de Aprendizaje

Por otro lado, Wiley también destaca algunas propiedades importantes que los objetos de aprendizaje contienen tales como: la granularidad, la digitalidad, la posibilidad que ofrecen de adaptación a diferentes contextos, la inmediatez con la que se hacen viables múltiples cursos de formación al relacionar OA diferentes de acuerdo con necesidades diferentes, la posibilidad que ofrecen de

trabajar desde diferentes partes del mundo (interoperabilidad), y el uso que de ellos se hace tantas veces como sea necesario (reusabilidad):

- **Granularidad:** en tanto que hablamos de pequeñas unidades de información.
- **Digitalidad:** de tal forma que es un material disponible en una red informática.
- **Diversidad.** El OA puede ser utilizado por múltiples personas y para múltiples contextos de aprendizaje.
- **Interoperabilidad:** posibilidad de trabajar simultáneamente una amplia comunidad de usuarios.
- **Inmediatez:** en tanto que la aportación sobre OA supone colaboración y ello a su vez implica beneficios inmediatos en las nuevas versiones.
- **Reusabilidad:** suponiendo, con ello, de un lado, un ahorro en esfuerzo y tiempo a la persona implicada al evitar que éste tenga que descomponer la información en pequeñas partes. De otro, la posibilidad de que el OA sea utilizado tantas veces como sea necesario por todas aquellas personas que lo necesiten y que puedan tener acceso a la red informática.

Banks [8] en su definición de OA destaca dos características esenciales en el concepto. De un lado la *reusabilidad*, y de otro, la *coherencia*. Deteniéndose en la reusabilidad el autor matiza esta característica diferenciando tres escenarios de reusabilidad:

a) Reusabilidad por el alumno: Se advierte el empleo de los OA cuanto el estudiante comienza el proceso de formación.

Se considera que un OA es reutilizable por el estudiante cuando el objeto de aprendizaje se encuentra disponible y es fácilmente accesible para él. De esta manera, cada estudiante puede elaborar o construir su propio paquete de objetos atendiendo a sus intereses personales y haciendo posible, de esta forma, que sus metas de aprendizaje queden cubiertas.

b) Reusabilidad por los proveedores de aprendizaje: haría referencia al diseño y a la planificación de aquello que se pretende enseñar.

Banks [8] expone la posibilidad de elaborar cursos por medio de un banco de OA. El proveedor de aprendizaje parte inicialmente de las necesidades

formativas de un grupo de estudiantes y desde ellas solicita al banco de metadatos (banco con datos sobre datos) aquellos criterios de formación que se ajustan a su grupo de alumnos. De forma rápida, y económica, se produce un curso que llega a ser un proyecto completo de enseñanza/ aprendizaje ajustado a las necesidades particulares del grupo de trabajo.

c) Creación por el principiante: aprendizaje adquirido por el estudiante y muestra de ello a través de su propia construcción.

Finalmente también se habla de reusabilidad cuando el estudiante, una vez que ha hecho su propio paquete de OA, genera uno diferente, propio y particular. Podríamos decir que es aquí donde la palabra aprendizaje asume su propio significado en tanto que este objeto es, en sentido estricto, el producto del aprendizaje del estudiante.

En este momento el estudiante es visto como un participante activo en la construcción del conocimiento siendo él quien lleva a cabo su propia construcción y la aporta a la comunidad de aprendizaje.

En los tres escenarios de reusabilidad expuestos por Banks éste reconoce la importancia de la *coherencia*, ya que para ser reutilizados conviene que estén dotados de sentido, dando la sensación de que algo pueda, sea o vaya a ser aprendido. Es por este motivo por el que el autor utiliza la palabra “*identificables*” puesto que si no hay forma de identificar o describir lo que debería ser aprendido difícilmente podría sugerirse el ofrecer “*piezas*” coherentes de aprendizaje.

1.3 Enseñanza basada en objetos de aprendizaje

La educación mediada por las tecnologías, llamada generalmente e-Learning, es una alternativa para elevar el nivel educativo y ampliar la cobertura en educación. Prueba de ello son las iniciativas para fomentar la educación virtual que se vienen adelantando tanto en Estados Unidos como en Canadá, México, Australia y la Comunidad Económica Europea.

Las instituciones de educación superior de estos países incorporan en su estrategia de enseñanza-aprendizaje la más avanzada tecnología de telecomunicaciones, redes electrónicas e hipermedios. Ofrece programas

educativos de licenciatura, maestría y educación continua, no solo para estudiantes de su país, sino a nuevos ámbitos educativos en otros países. Su principal característica es la innovación en los modelos educativos para que generen mayor calidad y motivación hacia el aprendizaje, así como la innovación en el uso de la tecnología en sus programas académicos.

Uno de los grandes retos de los miembros de las comunidades educativas es aprovechar e intercambiar ampliamente los recursos digitales. Generalmente, los recursos o materiales digitales desarrollados apenas son usados para el objetivo específico para el que fueron hechos, pero es claro el potencial que radica en el hecho de poder ponerlos a disposición de los demás miembros no solo de la misma institución, sino de la comunidad educativa en general, de manera que estos puedan ser aprovechados por el grueso de la misma como recursos posibles y disponibles para el desarrollo de esquemas de enseñanza aprendizaje.

Los países más desarrollados enfrentados con este reto, han hecho avances significativos en la definición de estándares y metodologías que garanticen los objetivos de accesibilidad, interoperabilidad, durabilidad y reutilización de los materiales desarrollados. Esta es una forma cómoda y viable de empaquetar los recursos y contenidos, tanto para los estudiantes que los usan para su estudio, como para los docentes que los utilizan en distintos contextos, especialmente en la preparación de cursos, y para los desarrolladores que tienen que construir nuevas herramientas y mejorar las actuales.

Entre los trabajos más importantes en este sentido, se encuentran los estándares emitidos por el IMS Global Learning Consortium (IMS), la Advanced Distributed Learning Initiative (ADL) y el Institute of Electrical and Electronics Engineers (IEEE). El elemento común y central relacionado con estos estándares es la propuesta de organizar el contenido educativo en la forma de objetos de aprendizaje.

1.3.1 Compartir recursos

En un sentido más estricto, un objeto de aprendizaje debe poseer ciertos atributos esenciales que lo distinguan de ciertas piezas de información: ser

educativo, proporcionar una cantidad de conocimiento o habilidad relativamente pequeña, ser auto-contenido, ser útil en más de una secuencia de instrucción, ser fácil de identificar y por lo tanto de buscar, ser independiente de un sistema administrador del aprendizaje (LMS) específico, ser accesible desde una gran variedad de plataformas.

Los tipos de materiales referidos como objetos de aprendizaje pueden ser cursos, videos, simulaciones, documentos, textos, multimedia, fotografías, tutoriales, etc.

Con el fin de facilitar la búsqueda de objetos de aprendizaje en un banco de objetos o repositorio, es necesario adjuntar al objeto, una descripción general sobre el mismo, así como se describen los libros (título, autor, fecha, editorial, etc.) en una ficha dentro de los catálogos de una biblioteca. Esta ficha o etiqueta del objeto recibe el nombre de “metadato”.

Los grupos de trabajo en estándares como los anteriormente mencionados (IMS, ADL y IEEE), son los encargados de definir los estándares asociados al metadato: establecer la información suficiente y necesaria para describir un objeto de aprendizaje: tipo de objeto, autor, contenido, idioma, versión, etc.

Es a través de este metadato como los motores de búsqueda y repositorios o bancos en línea pueden localizar, identificar y hacer el intercambio de estos objetos.

Un estudio realizado por la Comisión para la Educación basada en la web del Congreso de los Estados Unidos (Web-Base Education Commission, 2000), resume las ventajas de usar Internet para la educación en tres aspectos principales: centrar el proceso de enseñanza-aprendizaje en el estudiante en lugar de en el salón de clase, enfocarse en las fortalezas y potencialidades de cada estudiante y hacer de la educación continua una realidad.

La tendencia tecnológica en la producción de contenido educativo en la web se orienta hacia los objetos de aprendizaje, debido a su enorme potencial de reusabilidad, capacidad generativa, adaptabilidad y escalabilidad.

Como se señaló, el objetivo central de los objetos de aprendizaje consiste en posibilitar que los alumnos y los docentes puedan adaptar los recursos formativos en concordancia con sus intereses, necesidades, estilos y objetivos de formación y de aprendizaje.

Aunque se menciona que un OA es “una pieza pequeña” o un recurso “modular” no se puede especificar una dimensión precisa. El tamaño de un OA es variable y esto se conoce como granularidad.

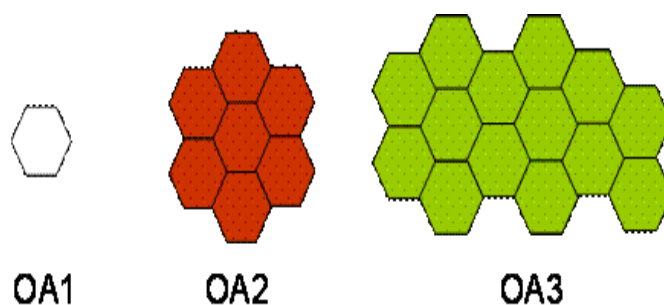


Figura 1.3 Objetos de Aprendizaje en distinta granularidad

En la Figura 1.3 se han representando, a través de hexágonos, lo que pueden ser unidades de contenido o elementos que componen al OA. Por ejemplo, el OA1 podría ser una imagen y el OA2 podría ser una página web que incluye texto e imágenes. El OA3 puede ser un recurso multimedia en el que se incluyen más unidades de contenido que en los objetos anteriores.

No es posible definir la cantidad de información o elementos que un OA debe contener, esto dependerá de las necesidades y habilidades del autor para trabajar y conceptualizar trozos de contenidos que irán formando un curso, el reto es crear objetos que mantengan la unidad y sean auto-contenidos. Por ejemplo (Figura 1.4), un curso se divide en módulos, un módulo en lecciones y las lecciones en temas; si la unidad mínima en que se puede fraccionar ese curso es “tema” entonces la construcción de OA para dicho curso estará orientado a la fracción o granularidad “tema”, en este ejemplo se construiría un OA para el TEMA1 y otro para el TEMA2, que seguramente tendrán un tamaño distinto.

La forma en la que los recursos se agregan o unen entre sí puede ayudar a definir su granularidad, también lo puede ser su tamaño en relación al número

de páginas, de duración o tamaño del archivo. Sin embargo, el mejor criterio para definir la granularidad de un objeto es por sus propósitos u objetivos [10].

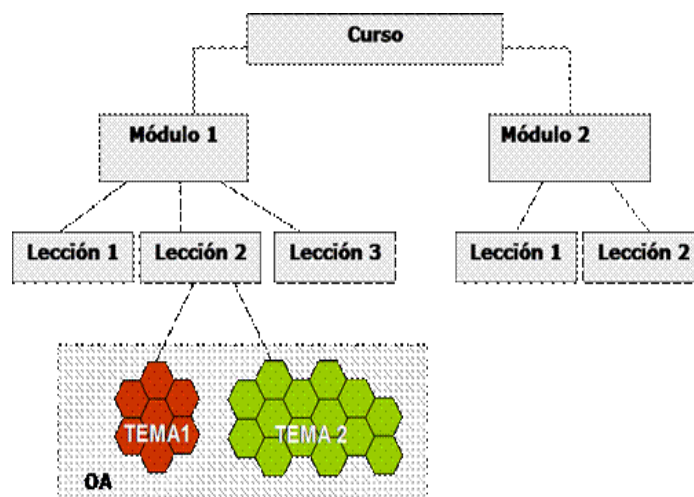


Figura 1.4 Taxonomía de un objeto de aprendizaje

De manera general, para llegar a la granularidad de un objeto, los contenidos se pueden visualizar en una estructura jerárquica. La amplitud y profundidad que esta estructura jerárquica tenga dependerá de los objetivos educativos. La mayor jerarquía tiene los contenidos o conceptos más generales y hacia abajo estarán los particulares, de éstos últimos se llega a la granularidad que deberá darse al OA.

Se considera una buena práctica que los OA cubran un único objetivo de aprendizaje y para lograrlo deben mantener independencia del contexto y no requerir de otros recursos, es decir, que sean autosuficientes y contengan en sí mismos los recursos necesarios para poderse interpretar.

1.4 Ventajas de los objetos de aprendizaje

Si en última instancia entendemos que el OA implica recursos para ser utilizados dentro de contextos de enseñanza-aprendizaje diferentes, podemos preguntarnos en este punto qué características tienen y/o les hacen ser motivo de atención dentro de las comunidades científicas. Respondiendo a esta cuestión se coincide con Boyle [6] en que los atributos a considerar a la hora

de elaborar materiales multimedia para la enseñanza y el aprendizaje deben caracterizarse por:

- Ser **universales**, tratando de hacer posible la asimilación de contenido en base a tradiciones diversas y evitando las particularidades ideológicas.
- Ser **extensibles y abiertos**, ofreciendo una base conceptual que permita incorporar los nuevos avances de forma estructurada, de acuerdo con el cuerpo de conocimiento establecido.
- Ser **formales**, permitiendo incorporar con precisión conceptos y representaciones.
- Ser **útiles**, en tanto que la representación sistemática del conocimiento debe estar dirigida *hacia* o *para* el diseño educacional y no exclusivamente *sobre* el diseño educacional.

La *universalidad*, atendiendo ahora al planteamiento que expusimos más arriba tomado de Banks [8] viene descrita por éste al considerar en su definición que los OA deben ser reutilizables, tal y como lo reconoce también Wiley [1].

Por otro lado, la apertura, formalismo y utilidad de los OA que expone Boyle [6], podrían resumirse en palabras de Banks [8] a través de la coherencia. En este sentido, el OA para ser reutilizado debe, de alguna manera, estar dotado de sentido dando la sensación de que algo ha sido aprendido. De ahí que los OA necesiten ser identificables (por medio de los metadatos por ejemplo) ya que si en realidad no hay forma de describir o identificar lo que debería ser aprendido difícilmente se podría tener la esperanza de disponer de una pieza coherente de aprendizaje.

El desarrollo del OA se justifica Longmire [11], Wiley [1] y Banks [8] por sus posibilidades a la hora de ser un material *flexible*. Los objetos tienden a ser pequeñas unidades de información por lo que la suma de varias unidades permiten obtener un producto para el aprendizaje distinto, siempre y cuando la combinación de OA se altere en al menos uno de los OA empleados o bien en el orden de aplicación.

Otra característica que apunta Longmire [11] es la facilidad con que los OA pueden ser *actualizados, investigados y dirigidos hacia un contenido concreto*.

En tanto que hablamos de unidades de información digitales, la posibilidad de estar colocadas en un banco de datos de una red telemática, facilita el acceso a multitud de personas de diferentes lugares.

Al tiempo, tanto el estudiante como el formador puede reformularlos, ampliarlos, redirigirlos e incorporar nuevos OA a toda la comunidad educativa (interoperabilidad e inmediatez) que dependiendo del contexto y de sus intereses los organizarán hacia un contenido u otro. Recordemos que esta posibilidad también era descrita por Banks [8] cuando diferenciaba la reusabilidad partiendo de tres momentos dentro del proceso de enseñanza/aprendizaje.

Siguiendo con las ventajas, los OA se caracterizan por la viabilidad a la hora de ser *personalizados*. Longmire [11] matiza que los OA permiten la adaptación al usuario al ser viable atender a las necesidades específicas del usuario y adaptar, en función de ellas, el material de formación rápidamente (materiales *just-in-time*). Los OA por ser pequeños elementos dotados de significado y por ellos modulares, granulares, amplían el potencial de *software* que personaliza el contenido (programas a medida del usuario) permitiendo la recombinación de materiales al nivel de granularidad deseado.

Los mismos OA permiten organizarse atendiendo al mismo tiempo a diferentes paquetes de especificaciones relacionadas con el diseño, desarrollo y presentación de los contenidos dentro de sistemas de aprendizaje distintos.

De otro lado, la granularidad o modularidad contribuye, como se indicó en el apartado anterior, al aprendizaje basado en competencias. El estudiante, por tanto, dentro de su personal proceso de aprendizaje tendrá la posibilidad de favorecer la adquisición de destrezas, entre ellas las relacionadas con el acceso al conocimiento, y el desarrollo de actitudes asociadas a dichas competencias (Longmire, [11]).

Finalmente, es de destacar, desde un planteamiento estrictamente mercantilista eso sí, la rentabilidad económica (Longmire [11]; Banks, [8]). La elaboración de un OA puede resultar lenta inicialmente, pero una vez que la pieza ha sido identificada cualquier usuario puede hacer uso de ella y

reutilizarla tantas veces como quiera. El valor del contenido en sí mismo se incrementa en la misma proporción en que dicho objeto es utilizado, ya sea por una misma persona en una o varias veces, ya sea por varias personas en uno o diferentes momentos.

1.5 Objetos de aprendizaje en el desarrollo docente

Los objetos de aprendizaje son una herramienta que ha mostrado su importancia en el desarrollo docente en los diversos países donde se ha incorporado su uso. Tal como lo describen los autores [11]. Un número de especialistas en el área educativa sostienen que el crecimiento profesional de los docentes está íntimamente relacionado con su deseo de aprender. Head y Taylor [12] consideran que el *desarrollo profesional* del docente debe diferenciarse de su *entrenamiento* como tal ya que el primero está motivado por una fuerza interna dirigida hacia la exploración de la propia práctica con el fin de mejorarla, mientras que el segundo está directamente asociado con una agenda de perfeccionamiento externa basada en técnicas. Judith Kennedy [13] comparte el mismo criterio al sostener que se puede considerar al entrenamiento como un número finito de estrategias de enseñanza. El desarrollo profesional, por otra parte, se centra en la experiencia de cada docente en particular como base para el cambio –orientándose hacia la reflexión y la modificación de actitudes. Es a través de la crítica sensible y constructiva, en lugar de la crítica negativa, que los viejos hábitos de enseñanza pueden abrir la puerta hacia nuevas maneras de ser y de hacer. Si bien el entrenamiento y el desarrollo se pueden diferenciar con cierta claridad, ambos son dos aspectos de una misma realidad: se complementan, en lugar de excluirse mutuamente.

Michael Wallace [14] considera de suma importancia que el docente se comprometa a mejorar su práctica progresivamente a lo largo de su vida laboral. Es así que su crecimiento profesional se presenta como un proceso de aprendizaje que ha de durar toda la vida. Asimismo, Kathleen Bailey [15] sostienen que el aprendizaje continuo es esencial para aquellos que desean crecer y cambiar a través del perfeccionamiento de sus competencias y de la práctica reflexiva.

Para alcanzar dicho objetivo, los docentes pueden recurrir –ya sea como parte de un plan de acción o quizás casi inconscientemente– a un número de estrategias que pueden variar en su grado de formalidad: las membresías en asociaciones profesionales, los comentarios informales con colegas en espacios comunes, la participación en conferencias y seminarios, las reuniones de personal, la participación en equipos de trabajo, los cursos de postgrado y la reflexión privada acerca de la propia práctica –entre muchas otras opciones [14]. El hecho de que exista todo este espectro de estrategias implica que puede haber tantas interpretaciones del concepto de desarrollo profesional y tantas formas de concretarlo como docentes que buscan canalizar el deseo que los motiva a aprender. Por ende, son los mismos docentes quienes tienen la oportunidad de armar su propia agenda de crecimiento o dinámica profesional en respuesta a sus necesidades de hacer de su experiencia laboral algo más valioso [12] (Head & Taylor).

Se ha subrayado, entonces, que la búsqueda continua de crecimiento profesional es un aspecto intrínseco en la vida de un docente profesional y que dicha búsqueda está normalmente motivada por el interés en mejorar la propia práctica. Asimismo, se ha mencionado que las actividades a las que un docente puede recurrir para cumplir con ese objetivo son casi ilimitadas. Dentro de las mismas, se puede incluir el aceptar nuevos retos laborales.

En este sentido, muchos docentes estarán de acuerdo en que la incorporación de TICs al proceso de enseñanza-aprendizaje se presenta como un verdadero desafío laboral. Dentro de las distintas tecnologías disponibles, una de las más utilizadas para la difusión de conocimiento es la Web. El uso de *Objetos de Aprendizaje* (aquí, OA) alojados en la Web implica un cambio en la metodología de enseñanza-aprendizaje y, al mismo tiempo, la posibilidad de adaptar el diseño de los cursos de acuerdo a las diversas necesidades de los estudiantes en distintos contextos educativos –pudiendo potencialmente enriquecer a todos los agentes involucrados.

Capítulo 2:

NETBEANS EN EL DESARROLLO DE APLICACIONES

Para conseguir trabajar en aplicaciones móviles (J2ME) una opción es la instalación de herramientas (Toolkits) Java [16, 17]. Sin embargo, si se recurre a ese tipo de instalaciones, se está bastante limitado para desarrollos posteriores y además no te darán una visión global del lenguaje Java. Por ello, en el presente trabajo de tesis instalamos herramientas profesionales, algunas bastante complejas, pero tocando solamente los aspectos J2ME para facilitar su uso. Se necesita, para ello, la instalación de los siguientes componentes: El Java Standard Edition (J2SE), en su versión Java Development Kit (JDK), en su versión 4 o superior. Lo puedes descargar desde la web de Sun: www.sun.com no confundir con el J2SE (Java Standard Edition), en su versión Java Runtime Environment (JRE), ya que este no nos serviría, pues es solo el entorno de ejecución, y nosotros lo que necesitamos es el entorno de desarrollo). También, empleamos un IDE completo, gratuito cuyo nombre es NetBeans (ver figura 2.1).

2.1 NetBeans



Figura 2.1 Relaciones entre ediciones de Java

NetBeans es Open Source Integrated Development Environment, escrito en Java, usado como plataforma para desarrollar programas en Java. Es decir, con NetBeans no solo podremos desarrollar para J2ME, sino también

programas complejos para diferentes entornos y sistemas operativos (Windows, Linux, Mac...) y aplicaciones cliente servidor. La instalación de todos estos programas es sobre el sistema operativo Windows XP presentando un ambiente completo, tal como se muestra en la figura 2.2.

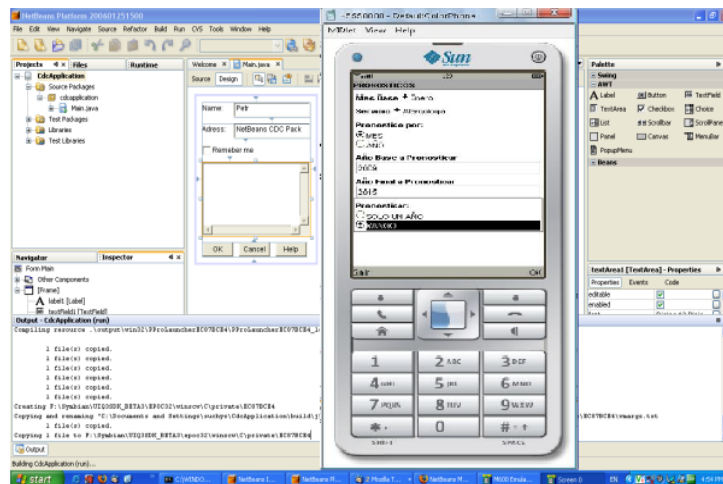


Figura 2.2 Ambiente de desarrollo Netbeans

NetBeans es Open Source Integrated Development Environment, escrito en Java, usado como plataforma para desarrollar programas en Java. Es decir, con NetBeans no solo podremos desarrollar para J2ME, sino también programas complejos para diferentes entornos y sistemas operativos (Windows, Linux, Mac...) y aplicaciones cliente servidor. La instalación de todos estos programas es sobre el sistema operativo Windows XP presentando un ambiente completo, tal como se muestra en la figura 2.1.

Con NetBeans podremos:

- Crear un proyecto
- Ejecutar un proyecto
- Depurar un proyecto.

2.2 NetBeans y modulo para movilidad

Con NetBeans Mobility Pack [18] se tiene la capacidad para desarrollar aplicaciones para dispositivos móviles, con características de optimización para el Mobile Information Device Profile (MIDP), y para el Connected Limited Device Configuration (CLDC). Además, integra en NetBeans características de

Visual Mobile Designer (diseño visual móvil) (ver figura 2.3); Wireless Connection Wizards: para acceder fácilmente a servicios vía servlets, con soporte para el estándar JSR-172; Fragmentación de Dispositivos: de modo que se puedan desarrollar aplicaciones para ejecutarlas en un dispositivo en concreto; Integración con el J2ME Wireless Toolkit: la herramienta de certificación y desarrollo J2ME, que integra emulación OTA (Over-The-Air), mensaje inalámbrico y APIs multimedia, y WMA emulación para mensajes SMS y CBS. Además, integra un Ofuscador (criptógrafo de código), y OTA Download Testing, que emula dicha capacidad sobre un móvil real.

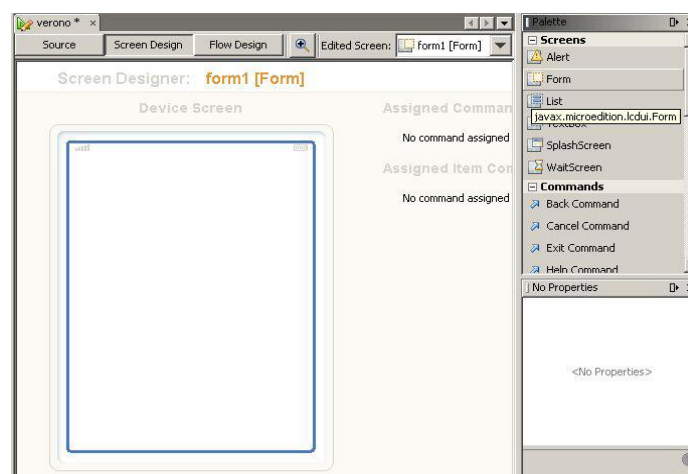


Figura 2.3 Interfaz para diseño visual móvil

Todas estas cualidades hacen que el Mobility Pack sea una herramienta potente y completa para desarrollar aplicaciones en J2ME.

2.3 Tecnología móvil

El uso de la tecnología móvil inalámbrica tal como PDA's, telefonía móvil, ipods o computadoras Notebook (ver figura 2.4) tanto en nuestros medios de trabajo como en nuestra vida diaria [19, 20, 21] está haciendo nuestras tareas cotidianas más flexibles y lo que es más importante, están pudiendo ser empleadas en donde sea y en cualquier momento. El desarrollo de aplicaciones móviles es novedoso ya que este facilita la entrega de información a la persona correcta, en el momento oportuno y en el lugar correcto usando dispositivos electrónicos portables. En un futuro cercano, las aplicaciones móviles vendrán a ser consideradas una parte normal de nuestra vida cotidiana. En este proyecto se desarrollan y evalúan metodologías innovadoras

para la construcción de sistemas basados en dispositivos móviles apoyados por aplicaciones en el Web.



Figura 2.4 Dispositivos móviles

La investigación muestra que los dispositivos móviles pueden ser más fácilmente integrados a la currícula que los dispositivos de escritorio. Esto es posible dado que muchos de los estudiantes cuentan actualmente con dispositivos móviles más que con computadoras de escritorio. Además, los dispositivos móviles no necesitan infraestructura como los dispositivos de escritorio. Otro punto a favor de los dispositivos móviles es que las investigaciones y estudios muestran ganancia en la motivación del estudiante, habilidades analíticas y organizacionales, y algo muy importante, la comprensión; habilidades de escritura y gramática y originalidad. La movilidad permitida por los dispositivos móviles puede crear un sentimiento mayor de propiedad de trabajo para los estudiantes. Es claro, que es necesario un tiempo de adaptación y familiaridad adecuada con los nuevos dispositivos móviles y aquí es donde el staff técnico debe estar disponible para apoyar.

2.4 Implicaciones de la tecnología móvil

Es importante notar como el desarrollo tecnológico evoluciona de manera vertiginosa logrando cambiar el comportamiento de las sociedades de manera radical. Apenas hace unos años, los usuarios de dispositivos móviles no eran más que unos simples consumidores, es decir, su actividad se centraba en el entretenimiento y aplicaciones multimedia, sin embargo, estos han revolucionado y han pasado de ser unos simples consumidores a ser en la

actualidad “prosumidores”, lo que significa que ahora, los usuarios de dispositivos móviles no solo ven video clips, y escuchan música, si no que ahora también socializan, comprando, buscando, creando contenidos, etc.

Actualmente, el término prosumidor (tomado de Wikipedia Foundation Inc. <http://es.wikipedia.org/wiki/Prosumidor>), se aplica en aquellos usuarios que fungen como canales de comunicación humanos, lo que significa que al mismo tiempo de ser consumidores, son a su vez productores de contenidos. Un “prosumer” no tiene fines lucrativos, sólo participa en un medio digital de intercambio de información, tal es el caso del P2P, redes pares intercambiables. La palabra prosumer describe perfectamente a millones de participantes en la revolución del Web 2.0, ya que son cada vez más las personas involucradas que suben información a la red y a su vez son consumidores de la misma, creando así un abanico de información en todos los sentidos.

Uno debería pensar que el concepto de integrar estándares, protocolos y mejores prácticas debería dominar todos y cada uno de los aspectos de un entorno integrado de desarrollo (IDE), pero los desarrolladores de aplicaciones a menudo descubren piezas críticas ausentes en sus presuntamente extensos conjuntos de herramientas. En la figura 2.5 podemos observar una aplicación desarrollada en la herramienta móvil de Netbeans. Como se puede apreciar, la calidad y desempeño de la aplicación son bastante aceptables que cualquier usuario se sentirá a gusto con la aplicación móvil.

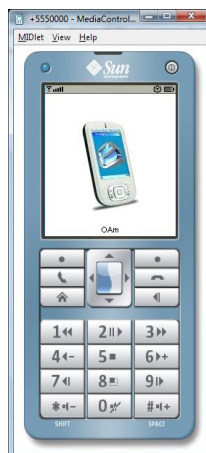


Figura 2.5 Aplicación móvil generada con NetBeans

2.5 J2ME y NetBeans

Sólo de forma muy simplista puede considerarse que J2ME y J2EE son versiones reducida y ampliada respectivamente, de J2SE: en realidad, cada una de las ediciones enfoca ámbitos de aplicación radicalmente distintos e independientes entre sí. Después de todos, las necesidades computacionales y APIs de programación requeridas por, por ejemplo, un juego gráfico ejecutándose sobre una PDA no tienen mucho que ver con los de un servidor distribuido de aplicaciones basado en EJB.

En lo que sigue presentamos con más detalle los componentes que integran la plataforma J2ME, la taxonomía propuesta por Sun de entornos de ejecución basados en las capacidades gráficas y computacionales de los dispositivos y la funcionalidad para la que están diseñados, la integración en J2ME de otras tecnologías Java relacionadas y la relevancia de la iniciativa J2ME para el desarrollo de aplicaciones y servicios embebidos en dispositivos móviles 2,5 y 3G [17].

2.5.1 Componentes de J2ME

Al contrario que en otras tecnologías orientadas a PCs y ordenadores convencionales, en J2ME el espectro de dispositivos considerados varía enormemente en cuanto a capacidad computacional, memoria y capacidades gráficas. Ante la imposibilidad de establecer una arquitectura común que se adecue a esta variedad de entornos hardware, J2ME define una serie de componentes (building blocks) a partir de los cuales se construye una implantación concreta para un dispositivo determinado. Estos componentes se agrupan en los siguientes tipos [22, 23]:

- Máquina virtual
- Configuración
- Perfil
- Paquetes opcionales

Un entorno de ejecución determinado de J2ME se compone entonces de una selección de máquina virtual, configuración y perfil, y posiblemente otros paquetes opcionales. En la figura 2.6 se muestran las relaciones entre los distintos componentes de un entorno de ejecución de J2ME.

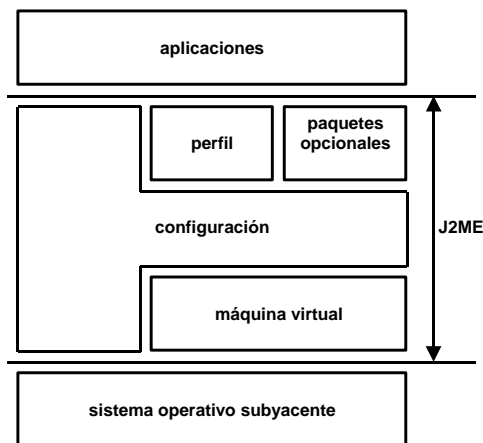


Figura 2.6 Relaciones entre componentes

2.5.2 Máquina virtual

Una máquina virtual de Java (JVM, *Java Virtual Machine*) es un programa encargado de traducir código intermedio (*bytecode*) de los programas Java precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java[22]. De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo subyacente.

Las implementaciones tradicionales de JVMs son en general muy pesadas en cuanto a memoria ocupada (*footprint*) y requerimientos computacionales. J2ME define varias JVMs de referencia adecuadas al ámbito de los dispositivos electrónicos, que en algunos casos se apartan de la especificación estándar relajando algunas exigencias con el fin de obtener implementaciones más ligeras.

2.5.3 Configuración

Una configuración es un conjunto de APIs básicas de Java que definen un entorno generalizado de ejecución [22]. Una configuración no está orientada a ningún campo específico de aplicación, sino que más bien trata de agrupar los dispositivos en cuanto a sus capacidades computacionales y entorno genérico de operación (por ejemplo, si tienen conectividad o no). En particular, las

librerías de interfaz gráfica nunca pertenecen a la definición de una configuración J2ME.

La configuración define la plataforma mínima necesaria para un grupo de dispositivos que tienen similar memoria y capacidades de procesamiento. Se compone de una máquina virtual, unas características del lenguaje Java y un conjunto mínimo de clases que soporta ese grupo de dispositivos.

La configuración tomada para la realización de nuestro proyecto es:

- Configuración del dispositivo: CLDC-1.0

2.5.4 Perfil

Un perfil es un conjunto de APIs orientadas a un ámbito de aplicación determinado, para una configuración J2ME dada. Los perfiles identifican un grupo de dispositivos por la funcionalidad que proveen (por ejemplo, teléfonos móviles, electrodomésticos) y el tipo de aplicaciones que típicamente correrán sobre ellos. La librería de interfaz gráfica es generalmente una componente muy importante de la definición de un perfil [22]. Las interfaces gráficas difieren mucho entre distintos perfiles (displays textuales en teléfonos móviles, interfaces para PDAs, etc.), y en algunos perfiles ni siquiera existen (sistemas embebidos).

Un perfil siempre se define a partir de una configuración dada. Así, cabe pensar en un perfil como un conjunto de APIs que dotan a una configuración J2ME de funcionalidad específica para una familia de dispositivos concreta y para un tipo de aplicaciones determinado.

Con objeto de ofrecer un completo entorno de ejecución específico para cada categoría de dispositivo, las configuraciones se deben combinar con un conjunto de APIs de alto nivel (perfiles), que definen el modelo de ciclo de vida de la aplicación, el interfaz de usuario y el acceso a las propiedades específicas del dispositivo. Algunos de los perfiles existentes son (varios todavía en la etapa de definición):

- **Mobile Information Device Profile (MIDP).** Diseñado para teléfonos móviles y PDAs, incluye la interfaz de usuario, conectividad de red, almacenamiento local de datos y gestión de aplicaciones. Combinado con CLDC, MIDP aporta un entorno de ejecución para Java que minimiza los consumos de memoria y de procesador.

- **Foundation Profile (FP).** Este perfil es el de nivel más bajo asociado a CDC, ya que los perfiles CDC pueden ser vistos como capas que se añaden a para proveer funcionalidades a los diferentes dispositivos. FP suministra una implementación de CDC con capacidades de acceso a red que se utiliza para aplicaciones embebidas en alto grado y sin interfaz de usuario.

- **Personal Profile (PP)** es un perfil diseñado expresamente para dispositivos con un interfaz gráfico completo y con posibilidad de ejecución de applets. Incluye además las bibliotecas del Abstract Windows Toolkit (AWT). Añade, por tanto, un interfaz de usuario básico a FP.

- **Personal Basis Profile (PBP)** es un subconjunto de PP que suministra un entorno para dispositivos que se puedan conectar a una red y que dispongan de un interfaz un poco más desarrollado que aquellos que posean los dispositivos donde PP va dirigido.

- **PDA Profile (PDAP)** es similar al MIDP pero diseñado para PDAs que tengan mejores pantallas y más memoria de los teléfonos móviles.

- **Game Profile (GP)** ofrece la plataforma para escribir juegos en dispositivos CDC.

Actualmente el perfil más utilizado es MIDP, que será el utilizado para el desarrollo de este proyecto, y en concreto usaremos el Perfil del Dispositivo MIDP-2.0.

Las aplicaciones desarrolladas para un determinado perfil serán portables a cualquier dispositivo que soporte ese perfil. Cabe destacar también que un mismo dispositivo puede soportar varios perfiles y que sobre una configuración también pueden residir diversos perfiles.

2.6 Arquitectura J2ME

Como se ha explicado anteriormente, el programa de estandarización ha definido diversos entornos de ejecución de J2ME identificados por una

selección particular de máquina virtual, configuración y perfil J2ME. Es de suponer que esta arquitectura evolucionará en dos sentidos: refinamiento de los entornos de ejecución definidos (nuevas versiones de los componentes) y definición de nuevos entornos de ejecución.

En el siguiente gráfico podemos ver un esquema modular con las diferentes configuraciones, perfiles y máquinas virtuales:

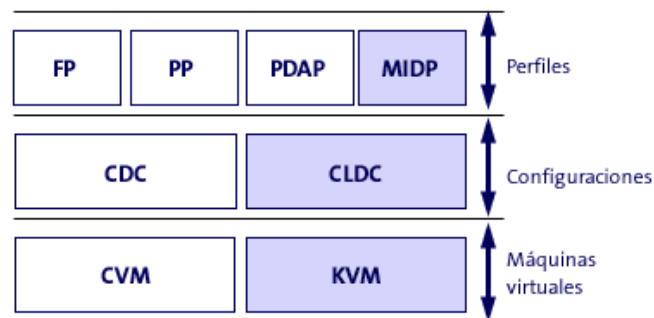


Figura 2.7 Arquitectura J2ME

En la figura 2.7 se muestra el estado actual de la arquitectura J2ME (figura 2.8 ejemplos de Sun). Solamente se han incluido en el diagrama los componentes de J2ME que están estandarizados o muy próximos a ser estandarizados y que tienen especial relevancia para el mercado de los dispositivos para comunicaciones móviles.

J2ME presenta dos configuraciones: CLDC y CDC. La primera se dedica a dispositivos con estrictas limitaciones de memoria, capacidad de cálculo, consumo y conectividad de red. Por otro lado, CDC se encarga de dispositivos con más potencia. Parte de CLDC es un subconjunto de CDC, por lo que la portabilidad de aplicaciones se puede conseguir cuando nos movemos de un entorno más restringido a otro más rico. De la misma manera, y siguiendo en el hilo de la portabilidad, una aplicación en J2ME podrá ejecutarse en J2SE normalmente, salvo que se utilicen las bibliotecas específicas de J2ME.

2.7 MIDlet

A las aplicaciones MIDP se denominan "MIDlet", el cual es un programa en lenguaje de programación Java para dispositivos embebidos (se dedican a una

sola actividad), más específicamente para la máquina virtual Java MicroEdition (Java ME). Generalmente son juegos y aplicaciones que corren en un teléfono móvil y está desarrollada bajo la especificación MIDP (perfil para información de dispositivo móvil) [26].

El ciclo de desarrollo de un MIDlet es el siguiente:

- Editar
- Compilar
- Pre-verificar MIDlet
- Ejecución en el emulador
- Ejecución en el dispositivo

Una vez realizados los MIDlets tienen que ser distribuidos en dos archivos especiales. Son los archivos JAR (Java Archive) y los archivos JAD (Java Application Descriptor). Un archivo JAR es un archivo comprimido (en formato ZIP) que contiene las clases (.class) que ha generado la compilación de nuestro programa. Además puede contener los recursos necesarios para el MIDlet como sonidos, gráficos, etc... Finalmente, contiene un archivo con extensión .mf, es lo que se llama un archivo de manifiesto. Este archivo contiene información sobre las clases contenidas en el archivo JAR.

El segundo archivo necesario para la distribución de MIDlets son los archivos JAD. El archivo JAD contiene información necesaria para la instalación de los MIDlets contenidos en el archivo JAR. Un archivo puede contener más de un MIDlet. Cuando ocurre esto, hablamos de un MIDlet suite (véase figura 2.8). Podemos editar los parámetros contenidos en el archivo JAD mediante el botón Settings de KToolBar. Aquí podemos editar información del MIDlet como el nombre, la versión o el autor del MIDlet (o de los MIDlets).

Sólo nos resta transferir los archivos JAR y JAD al dispositivo móvil con soporte J2ME. Hay varias formas de hacerlo dependiendo de la marca y modelo del dispositivo.

Si el teléfono tiene soporte de infrarrojos o bluetooth, y la computadora tiene puerto IrDA o bluetooth, se podrá transferir fácilmente el archivo sin necesidad

de cable alguno. Si no, se tendrá que recurrir a un cable de datos (consulta el manual del teléfono).

Otra posibilidad es poner los archivos JAR y JAD en un servidor WAP o un espacio web y descargarlo desde tu móvil. Para ello es necesario que el dispositivo tenga un navegador WAP o web y soporte GPRS para una descarga fiable.



Figura 2.8 Ejemplos de aplicaciones MIDlets de Sun

Para que una aplicación MIDP trabaje en un dispositivo móvil, se requiere un dispositivo que implemente JavaME y MIDP (versiones 1.0 y posteriores) para ser utilizada [26]. Como otros programas desarrollados en Java, tienen la característica, "Escribir una vez, ejecutar en cualquier parte" ("Write once, run anywhere"). Para escribir se puede obtener Sun Java Wireless Toolkit o NetBeans con la extensión Mobility Pack. Para las distribución son necesarios dos archivos, archivo .JAR conteniendo el bytecode del programa y un archivo .JAD que describe los contenidos del archivo .JAR.

Un MIDlet tiene que cumplir los siguientes requisitos para poder funcionar en un dispositivo móvil:

- La clase principal necesita ser una subclase de `javax.microedition.midlet.MIDlet`.
- El MIDlet necesita ser empacado dentro de un archivo .JAR.
- El archivo .JAR necesita ser pre-verificado usando un pre-verificador.
- En algunos casos, el archivo .JAR necesita ser firmado digitalmente por un proveedor de dispositivos móviles.

En la Figura 2.9 podemos observar el diagrama de estados el cual muestra el comportamiento de un Midlet en su ejecución.

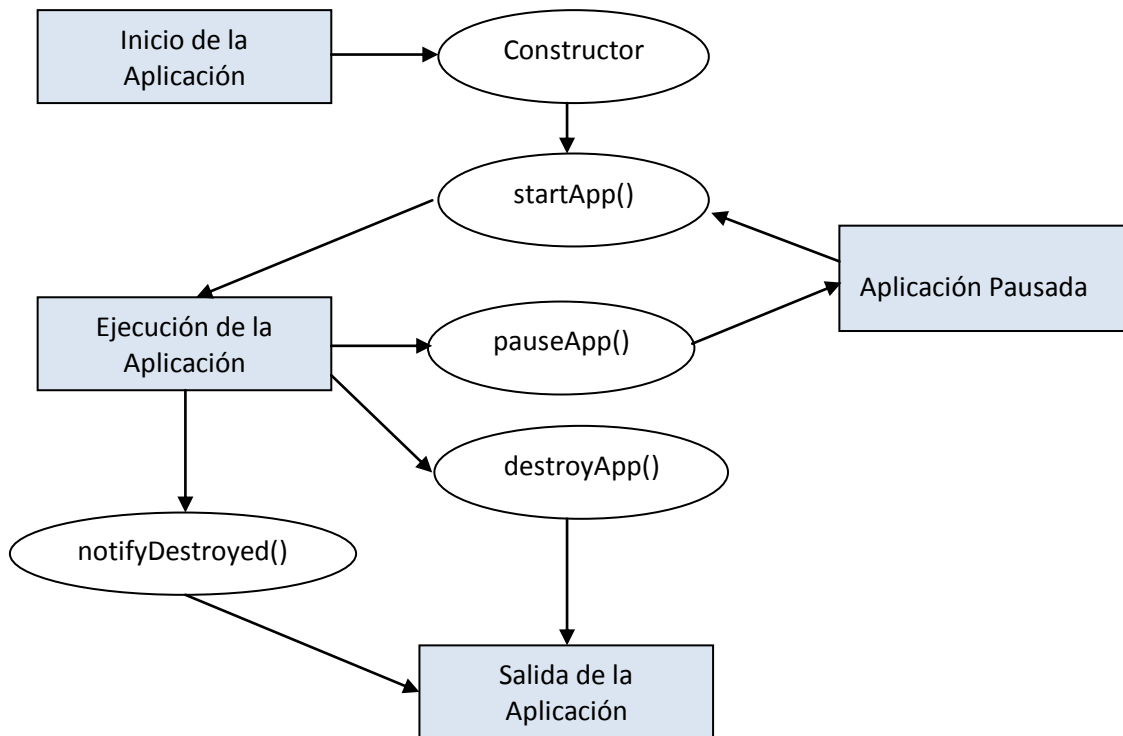


Figura 2.9. Diagrama de Estados de un Midlet

2.8 GPRS

Nuestra aplicación móvil utiliza el servicio GPRS para realizar la descarga de los Objetos de Aprendizaje Móviles (los cuales nombraremos OAm), dicho servicio deberá estar configurado el dispositivo móvil que contiene nuestra aplicación.

El sistema GSM es el sistema de comunicación de móviles digital de 2ª generación basado en células de radio. Apareció para dar respuestas a los problemas de los sistemas analógicos.

Fue diseñado para la transmisión de voz por lo que se basa en la conmutación de circuitos, aspecto del que se diferencia del sistema GPRS. Al realizar la transmisión mediante conmutación de circuitos los recursos quedan ocupados durante toda la comunicación y la tarificación es por tiempo. Más adelante veremos como estas limitaciones hacen ineficiente la transmisión de datos con GSM y como GPRS lo soluciona.

Nuestra aplicación realiza la descarga de los Objetos de Aprendizaje Móviles vía GPRS el cual es una nueva tecnología que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de 'paquetes'. La conmutación de paquetes es un procedimiento más adecuado para transmitir datos, hasta ahora los datos se habían transmitido mediante conmutación de circuitos, procedimiento más adecuado para la transmisión de voz.

2.8.1 Ventajas del GPRS para el usuario.

Las ventajas que obtiene el usuario con el sistema GPRS son consecuencia directa de las características vistas en el punto anterior.

- **Característica de "Always connected"**: un usuario GPRS puede estar conectado todo el tiempo que desee, puesto que no hace uso de recursos de red (y por tanto no paga) mientras no esté recibiendo ni transmitiendo datos.

- **Tarifación por volumen de datos transferidos**, en lugar de por tiempo (Por ejemplo: una compañía mexicana cobra \$ 0.04 pesos por Kilobyte descargado).

- **Coste nulo de establecimiento de conexión a la red GPRS**, frente a los quantum de conexiones existentes actualmente en GSM.

- **Mayor velocidad de transmisión**: en GSM sólo se puede tener un canal asignado (un "timeslot"), sin embargo, en GPRS, se pueden tener varios canales asignados, tanto en el sentido de transmisión del móvil a la estación base como de la estación base al móvil. La velocidad de transmisión aumentará con el número de canales asignados. Además, GPRS permite el uso de esquemas de codificación de datos que permiten una velocidad de transferencia de datos mayor que en GSM.

- **Posibilidad de realizar/recibir llamadas** de voz mientras se está conectado o utilizando cualquiera de los servicios disponibles con esta tecnología.

- **Modo de transmisión asimétrico**: más adaptado al tipo de tráfico de navegación html o wml. Un terminal GPRS 4+1 (4 slots downlink y 1 uplink) tendrá cuatro veces mayor capacidad de transmisión de bajada que de subida.

2.9 Conexión a redes

Uno de los aspectos más interesantes de los MIDlets es su acceso a Internet a través de una conexión inalámbrica. Las clases e interfaces usadas en MIDP para acceso a red son muy similares a las usadas en J2SE y J2EE.

El CLDC delega las funciones de red en el GCF (Generic Connection Framework). El propósito del GCF es proporcionar una abstracción de los servicios de red. En lugar de forzar a todos los dispositivos a soportar todos los protocolos de red, el GCF describe un marco de trabajo en red del que el dispositivo seleccionará los protocolos y servicios que es capaz de soportar. En realidad, no es el dispositivo el que selecciona las capacidades de red que soportará, sino que esto lo hará el perfil del dispositivo. La responsabilidad del GCF es describir las capacidades de red disponibles en todos los dispositivos así como un sistema extensible en el que los diferentes perfiles de dispositivo pueden soportar un subconjunto de dichas capacidades.

El GCF describe una clase fundamental denominada *Connector*, que será usada para establecer todas las conexiones de red. Los tipos específicos de conexiones son modelados por las interfaces del GCF que son obtenidos a través de la clase *Connector*. Todas estas clases e interfaces están dentro del paquete *javax.microedition.io*. Tal y como vimos anteriormente, las interfaces son las siguientes [23]:

- **Connection:** Una conexión básica que sólo puede ser abierta y cerrada.
- **ContentConnection:** Un flujo (stream) de conexión que proporciona acceso a datos web.
- **DatagramConnection:** Una conexión para manejar comunicaciones orientadas a paquetes.
- **InputConnection:** Una conexión de entrada para las comunicaciones del dispositivo.
- **OutputConnection:** Una conexión de salida para las comunicaciones del dispositivo.
- **StreamConnection:** Una conexión en ambas direcciones para las comunicaciones del dispositivo.

- **StreamConnectionNotifier:** Una conexión especial para notificaciones, que es usada para esperar que se establezca una conexión.

La descripción de las capacidades específicas en cada caso, la tenemos en los perfiles. Por ejemplo, el API de MIDP construye sobre estas interfaces el soporte para el trabajo en red de los MIDlets. El API de MIDP añade la interfaz `HttpConnection`, que proporciona un componente nuevo en el GCF para conexiones HTTP. Estas conexiones permiten a los MIDlets conectarse con páginas web. La especificación MIDP indica que las conexiones HTTP son el único tipo de conexiones obligatorio en las implementaciones de MIDP.

Siempre usaremos la clase `Connector` para establecer conexiones, independientemente del tipo de conexión. Todos los métodos de la clase `Connector` son estáticos. El más importante de ellos es el método `open()`. Las tres versiones de éste método son:

static Connection open (String name)
static Connection open (String name, int mode)
static Connection open (String name, int mode, boolean timeouts)

El primer parámetro de estos métodos es la cadena de conexión. Este es el parámetro más importante porque determina el tipo de conexión que se realizará. La cadena de conexión tendrá el siguiente formato:

Protocolo:Destino[Parámetros];

La primera parte define el protocolo de red, como `http` o `ftp`. El destino es típicamente el nombre de la dirección de red. El último parámetro es una lista de parámetros asociados a la conexión.

Algunos ejemplos de diferentes tipos de cadenas de conexión [25]:

- **HTTP:** `http://www.usal.es/`
- **Socket:** `socket://www.usal.es:80`
- **Datagram:** `datagram://:9000`
- **File:** `file:/datos.txt`
- **Port:** `comm:0;baudrate=9600`

La segunda y la tercera versión del método `open()` esperan un segundo parámetro denominado modo (`mode`), que describe el modo de conexión. Este modo indica si la conexión es abierta para leer, escribir o para ambas cosas. Las siguientes constantes, definidas en la clase `Connector`, pueden ser usadas para describir el tipo de conexión:

- `READ`
- `WRITE`
- `READ_WRITE`

La última versión del método `open()` acepta un tercer parámetro, que es un flag que indica si el código que ha llamado el método es capaz de controlar una excepción por tiempo excedido (`timeout`). Esta excepción es lanzada si el periodo de `timeout` para establecer la conexión falla. Los detalles de este periodo y cómo es gestionado, es labor del protocolo específico, por lo que no tenemos control sobre el mismo.

El método `open()` devuelve un objeto de tipo `Connection`, que es la interface básica para el resto de interfaces de conexión. Para usar un determinado tipo de interface de conexión, debemos convertir la interfaz `Connection` que devuelve el método `open()` al tipo apropiado:

```
StreamConnection conn = (StreamConnection)Connector.open(http://www.usal.es/);
```

2.9.1 Entrada/Salida desde el MIDlet

La clase `Connector` y las interfaces asociadas a esta son usadas para obtener una conexión a la red. Una vez que la conexión está establecida, debemos usar las clases de entrada/salida para leer y escribir datos a través de la conexión. Estas clases son aportadas por el paquete `java.io`:

- **`ByteArrayInputStream`**: Un flujo (stream) de entrada que se gestiona internamente como una matriz de bytes.
- **`ByteArrayOutputStream`**: Un flujo (stream) de salida que se gestiona internamente como una matriz de bytes.
- **`DataInputStream`**: Un flujo (stream) desde el cual se leen datos como tipos primitivos.

- **DataOutputStream**: Escribe datos en tipos primitivos en un flujo (stream) en formato binario.
- **InputStream**: La clase base para todos los flujos (streams) de entrada
- **InputStreamReader**: Un flujo (stream) desde el que se puede leer caracteres de texto.
- **OutputStream**: La clase base para todos los flujos (streams) de salida
- **OutputStreamWriter**: Un flujo (stream) en el que se pueden escribir caracteres de texto.
- **PrintStream**: Un flujo (stream) de escritura que facilita el “envío” de datos en forma de tipos primitivos.
- **Reader**: Una clase abstracta para leer flujos (streams) de lectura
- **Writer**: Una clase abstracta para leer flujos (streams) de escritura

Las dos clases más importantes de esta lista son `InputStream` y `OutputStream`, que proporcionan la funcionalidad básica para recibir y enviar datos, las cuales serán descritas más adelante.

2.9.2 La clase `InputStream`

La clase `InputStream` es una clase abstracta que sirve como base para otras clases de streams de entrada en MIDP. Esta clase define un interfaz básico para leer bytes en modo de stream. El uso normal será crear un objeto de tipo `InputStream` desde una conexión y entonces recoger información a través del método `read()`. Si no hay información disponible en este momento, el stream de entrada usa una técnica como conocida como bloqueo (bloquing) para esperar hasta que haya datos disponibles.

La clase `InputStream` define los siguientes métodos:

- `read()`
- `read(byte b[])`
- `read(byte b[], int off, int len)`
- `skip(long n)`
- `available()`
- `mark(int readlimit)`

- reset()
- markSupported()
- close()

2.9.3 La clase **OutputStream**

La clase `OutputStream` es el homólogo de salida de la clase `InputStream` y sirve como base para todas las clases de streams de salida de MIDP. La clase `OutputStream` define el protocolo básico para escribir datos a través de un stream de salida. Normalmente crearemos un objeto de tipo `OutputStream` en una conexión y entonces llamaremos al método `write()` [23, 24]. La clase `OutputStream` usa la técnica de bloqueo de forma similar a lo visto para la clase `InputStream`, es decir, se bloquea hasta que los datos son escritos en el stream. Durante el bloqueo la clase `OutputStream` no permite que más datos sean escritos.

Los métodos de la clase son:

- `write(int b)`
- `write(byte b[])`
- `write(byte b[], int off, int len)`
- `flush()`
- `close()`

2.9.4 Haciendo conexión

El primer paso en el desarrollo de cualquier tipo de comunicación de red en un MIDlet involucra el establecimiento de la conexión, esto se realiza a través de `static open()` el cual es método de la clase `Connector` () [24]. El siguiente ejemplo muestra como abrir una conexión HTTP de una página web:

```
String url = "http://localhost:8080/mi_servlet/OAmt?por="+por+
            "&tipo="+tipo+"&busqueda="+enviar;
HttpConnection c = (HttpConnection)Connector.open(url);
```

La conexión lanza un objeto `HttpConnection` para que se pueda realizar la comunicación conexión por un `OutputStream` o `InputStream`. El siguiente paso es obtener un `OutputStream` o `InputStream` sobre la conexión, el cual es necesario actualmente para ejecutar cualquier entrada o salida.

2.9.5 Obtener un `InputStream`

El obtener un `Stream` de una conexión es un poco avanzado. Para el propósito de lo desarrollado en este proyecto de tesis, demostraremos como obtener un `InputStream`, el cual podrá ser usado para leer la página web asociada a la conexión HTTP. El código necesario para obtener un `InputStream` de una conexión a través del cual se puedan leer datos, es el siguiente:

```
InputStream in = null;  
in = conn.openInputStream();
```

El método `openInputStream()` es todo lo necesario para establecer un `InputStream`. Lo siguiente a este punto es obtener el `OutputStream` para que se pueda iniciar la lectura de los datos de la conexión HTTP.

2.9.6 Leyendo desde el `Stream`

La clave para la lectura de datos de un objeto `InputStream` es el uso del método `read()`. La versión básica del método `read()` lee un byte de datos desde el `InputStream` y regresa a este como un entero. Las llamadas subsecuentes al método `read()` resulta en bytes adicionales de datos iniciando la lectura desde el `Stream`, continuando hasta el final `Stream`, el cual resulta en el retorno de el método `read()`. El siguiente es un ejemplo de cómo el código dentro de un ciclo (loop) lee líneas de datos de un `InputStream` e imprime la lectura:

```
StringBuffer data = new StringBuffer();  
  
int ch;  
  
while ((ch = in.read()) != -1) {  
  
if (ch != '\n') {  
  
// Lee la línea caracter por caracter
```

```
data.append((char)ch);
}
else {
// Imprime la línea como cadena de datos
System.out.println(data.toString());
// Limpia la cadena para la siguiente línea
data = new StringBuffer();
}
}
```

2.10 ciclo que lee líneas de datos de un *InutStream* e imprime la lectura

En el código 2.10, la variable *data* es un *string buffer* que es usado para retener por un momento una línea de datos. El método *read()* es llamada para leer caracteres del Stream. Dentro del ciclo *while*, el actual caracter es comparado con *'\n'* (nueva línea) para ver si el final de la línea ha sido alcanzado. Si es así, ya sabemos que tenemos acumulada una línea de texto completa, y si no hay error se imprime y se limpia el *string buffer*. [22,24]

Este código representa la premisa básica de cómo podemos leer los datos a través de un *InputStream* de una conexión HTTP, la cual en nuestro caso particular es la usada para conectarse a nuestra a aplicación web ubicada en un servidor HTTP.

Capítulo 3:

OBJETOS DE APRENDIZAJE MÓVILES

Los Objetos de Aprendizaje Móviles contienen las propiedades de los Objetos de Aprendizaje como lo son: Granularidad, Digitalidad, Diversidad, Interoperabilidad, Inmediatez y Reusabilidad (los cuales se describen en el Capitulo 1).

Los Objetos de aprendizaje (OA) simples, pueden ser más atractivos si ofrecen la posibilidad de ser vistos utilizando diferentes dispositivos, desde un teléfono móvil, por ejemplo [19, 20, 21]. Los usuarios de un OA pueden tener la necesidad de navegarlo offline, ya que no disponen de una conexión de Internet, otros quizás prefieran “bajarlos” en formato de archivo de texto, video o sonido, donde tienen la posibilidad de resaltarlo, modificarlo e imprimirlo según sea.

Con la difusión masiva de los teléfonos móviles, la progresiva mejora de los dispositivos, el aumento de sus funciones, y la creciente heterogeneidad en los usos que las personas les dan, los móviles pueden convertirse pronto en otra herramienta para aprender.

La mayoría de los teléfonos móviles en venta actualmente soportan aplicaciones .java (ejemplo son: los juegos, el convertidor de moneda, calculadoras, etc.). El desarrollo de un Objeto de Aprendizaje Móvil (OAm) interactivo, requiere programación en j2me (para java), pero si se trata de un OAm de tipo expositivo, donde el texto es lo más importante, existen programas que pueden transformar un archivo de texto de extensión “txt” en una aplicación java, mjBookMaker 4.7 [28] es uno de ellos; en el caso de los OAm en video se usaremos el programa Total Video Converter para obtener el formato requerido que es .3gp y en los OAm de sonido el programa MP3Converter es útil para convertir los sonidos en un formato compatible para los dispositivos móviles, siendo .mp3 el formato a utilizar.

3.1 La revolución móvil

Si con la llegada de las computadoras e Internet llegó el Aprendizaje Electrónico (mejor conocido en inglés como e-Learning), con la revolución móvil actual llega el Aprendizaje Móvil (conocido como m-Learning). El m-Learning no es más que la evolución del e-Learning a un entorno móvil sobre dispositivos móviles [27] (véase figura 3.1).

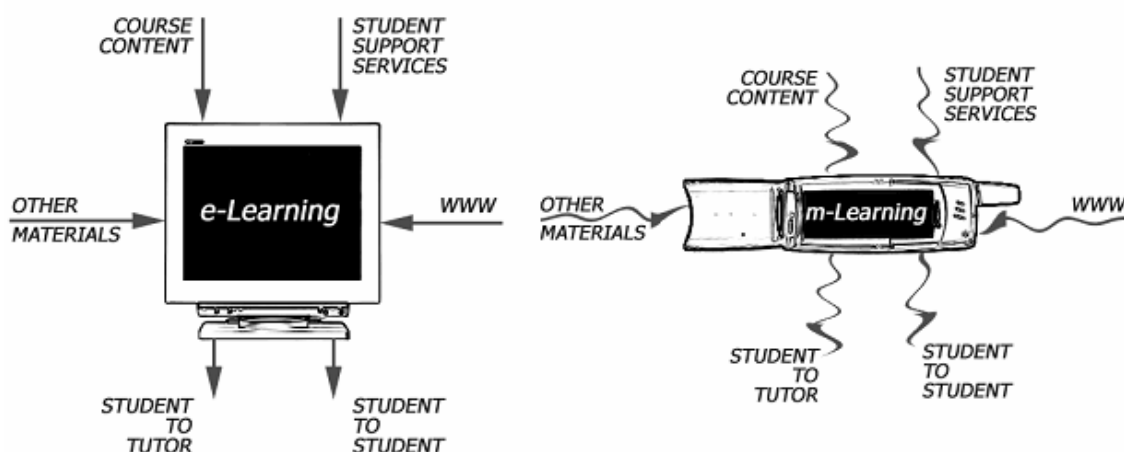


Figura 3.1 e-Learning y m-Learning (Se indican los elementos en inglés, para evitar interpretaciones erróneas de traducción).

Con el desarrollo del Bluetooth, WAP, GPRS y UMTS, el movimiento de la tecnología y la informática hacia un entorno inalámbrico es irreversible. Tecnologías y aplicaciones son sustituidas o completadas por otras nuevas e inalámbricas: e-Business por m-Business, e-Commerce por m-Commerce y también e-Learning por m-Learning.

3.2 Objetos de aprendizaje y e-Learning

El Paradigma de Orientación a Objetos o POO, actualmente muy usado en el área de las ciencias de la computación. En este paradigma, los datos son representados como objetos, los cuales tienen atributos y métodos. Por otra parte en el ámbito de las aplicaciones educativas se ha hecho necesario contar con un nuevo tipo de datos que se adapte de mejor manera, al ámbito de la educación lo que ha dado lugar a los Objetos de Aprendizaje o LO (*Learning Object*) y que tienen las mismas características.

La necesidad de disponer de entidades que sirvan al aprendizaje, no es nueva y fue planteada originalmente en 1991 por David Merrill, cuando en su Teoría de Transacción Instruccional escribía de la necesidad de tener “*Unidades de Conocimiento*”, [35].

Sin embargo, hasta el año 2000, es donde aparecen varios proyectos entre ellos *ARIADNE* e *IMS* los cuales apoyan al grupo de trabajo, Metadato de Objetos de Aprendizaje o LOM (*Learning Objects Metadata*) para obtener en el 2002, el primer estándar del IEEE (*Institute of Electrical and Electronic Engineers*), sobre Objetos de Aprendizaje [34], donde lo definen como: “cualquier entidad digital o no digital que se pueda utilizar, reutilizar o referenciar durante el aprendizaje apoyado por tecnología”. Otros autores como David A. Wiley, hace uso de metáforas y taxonomías para explicar su uso, [36]. Existen además, otros trabajos de los autores donde se describe una relación específica para los LO entre la metáfora del LEGO y el Paradigma de Orientación a Objetos y repositorios de objetos de aprendizaje [31, 32, 33].

3.3 Objetos de Aprendizaje Móviles y m-Learning

El contenido necesita ser desarrollado para que sea independiente del dispositivo. Se usan representaciones que separen contenido de formato. Sobre este modelo de contenido se desarrolla una arquitectura que proporciona conocimiento sobre el usuario y su contexto, como también, sobre los contenidos disponibles que se ajustan a su contexto.

Tomando en consideración lo anterior y sabiendo que un dispositivo móvil dispone de una menor cantidad de recursos informáticos (memoria, procesador, disco duro, pantalla, etc.) se sugiere que el diseño de un entorno de aprendizaje necesita considerar dos dimensiones.

1. La visualización de contenidos debe ser flexible
2. La entrega de objetos debe realizarse para el contenido y formato preciso.

3.4 El estándar SCORM de la ADL

Aprendizaje Distribuido Avanzado o ADL (*Advanced Distributed Learning*) es un programa del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca para desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y en gran escala, de formación educativa sobre nuevas tecnologías Web.

ADL nace como una forma de unificar iniciativas anteriores, pero recogiendo las características más desatacadas de cada una de ellas, es así como: de IMS recoge la descripción de cursos en XML, como forma de separar contenidos de presentación; y de AICC recoge el mecanismo de intercambio de información mediante una API. Con todo ello establece una proposición de estándar llamado SCORM. De esta forma el estándar establece una serie de pasos a seguir para la implementación detallada de contenidos, que permita que los sistemas que lo cumplan, puedan intercambiar contenidos, es decir, logrando interoperabilidad, reusabilidad y adaptabilidad.

Según SCORM, un LMS es un software consistente de un conjunto de funcionalidades, diseñadas para poner a disposición, hacer seguimientos, entregar reportes y gestionar contenidos de aprendizaje, además del progreso e interacción de los aprendices, también se puede aplicar a un sistema simple de gestión de cursos o a entornos de distribución empresarial de alta complejidad.

La última versión del estándar es SCORM 2004, la que básicamente agrega sólo el libro "Secuenciación y Navegación", y hace muy pocas diferencias con la versión anterior SCORM 1.2, donde existe una gran cantidad de LMS y herramientas que la cumplen. Por esta razón se hará referencia a la versión SCORM 1.2, donde sus especificaciones, están organizadas en tres libros.

En la Figura 3.2, se muestra a grandes rasgos los contenidos de cada uno de ellos y su origen. Por otra parte SCORM divide, la tecnología e-learning en componentes funcionales. Los dos principales componentes son:

1. Los LMS, antes mencionados, y los
2. Objetos de Contenido Compartible o SCOs (*Sharable Content Objects*).

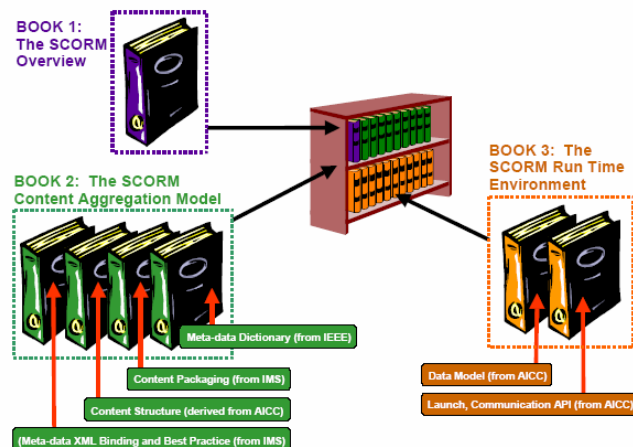


Figura 3.2.- Libros de la Versión 1.2 de SCORM. Fuente [29]. (Se indican los elementos en inglés, para evitar interpretaciones erróneas de traducción).

Los SCO hacen referencia a objetos de aprendizaje reusables y estandarizados, los cuales son una combinación de “Asset” [30]. La Figura 3.3, muestra un SCO, escrito en Java Script y compuesto de seis Assets, cada uno de ellos en un formato distinto (html, txt, jpeg, flash, gif y wav).

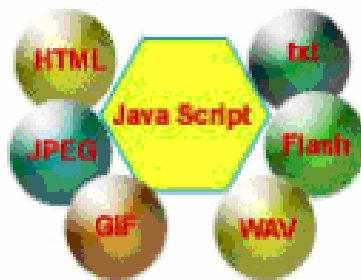


Figura 3.3. Ejemplo de un SCO.

Y por último pero no menos importante, cabe destacar que la ADL realiza dos acciones, las cuales son, “Certifica” y “Valida Conforme” SCORM, contenido y LMS.

El estándar para objetos de aprendizaje SCORM, combinado con XML, proporcionan las bases para que el contenido se desarrolle de forma independiente a los mecanismos de visualización y además, provee de reglas para establecer el ajuste del contenido a la pantalla (Display) del dispositivo. Estándares como EML (Education Modeling Language) proporcionan la especificación de pedagogías apropiadas.

¿Y cómo son estos objetos de aprendizaje? Si queremos que sean tan flexibles... ¿no estamos limitando su diseño? Las restricciones de tamaño físico como el espacio de disco duro o memoria de un dispositivo móvil potencia representaciones básicas. El texto sigue siendo el medio más utilizado. Gráficos, que aparecen en forma de diagramas o como dibujos estáticos, no ocupan mucho espacio y resultan ser objetos sumamente representativos. De la misma manera se puede trabajar con narraciones en audio o videos adaptados a un formato para dispositivos de bajos recursos en muy viable.

3.5 Los Sistemas de Gestión de Aprendizaje o LMS

Para visualizar Objetos de Aprendizaje empaquetados bajo el estándar SCORM es necesario contar con un editor o una plataforma de aprendizaje que lo permita, las más comunes son los LMS.

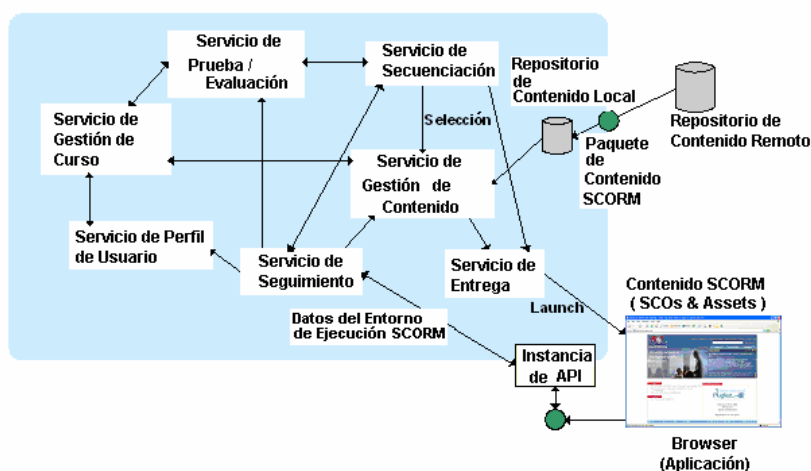


Figura 3.4. Modelo Altamente Generalizado de un LMS. Fuente [29].

En la Figura 3.4 se muestra un modelo generalizado que muestra sus componentes potenciales o los servicios de un LMS. Cabe mencionar que el estándar SCORM no especifica la funcionalidad dentro del LMS. La Universidad Austral de Chile, utiliza como Sistema de Gestión de Aprendizaje, el LMS SIVEDUC4 (Sistema Virtual de EDUCación), véase Figura 3.5, basado en el LMS de código abierto Dokeos Versión 1.5. Cabe destacar que éste LMS permite contenidos IMS 1.1, SCORM 1.2 y SCORM 2004.

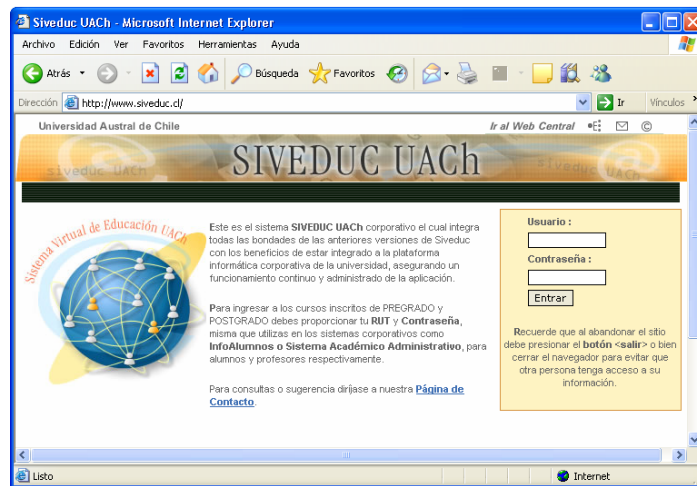


Figura 3.5. Interfaz del Sistema Virtual de EDUCación o SIVEDUC de la Universidad Austral de Chile.

3.6 Editores de contenido

Como se dijo anteriormente, para utilizar un Objeto de Aprendizaje bajo el estándar SCORM en un LMS, es necesario su “empaquetamiento” que consiste en etiquetar el contenido de tal modo que pueda ser reconocido como tal por el LMS y permita su carga en el sistema. En el caso de las especificaciones ADL SCORM e IMS (las más extendidas en la actualidad) esto se hace describiendo el contenido en un archivo XML denominado imsmanifest.xml, donde se referencia todos los recursos que agrupa dicho contenido.

Existen varias herramientas para “empaquetar” contenidos y una buena opción gratuita y de código abierto es el editor ReLOAD5 ó (*Reusable eLearning Object Authoring & Delivery*), se puede editar, pre-visualizar y empaquetar contenidos ó sea integrar SCOs, es decir, asegura que el contenido de los cursos empaquetados son, SCORM 1.2 compatibles. En la figura 3.6, se representa el esquema general de un empaquetador de contenidos como el editor ReLOAD.

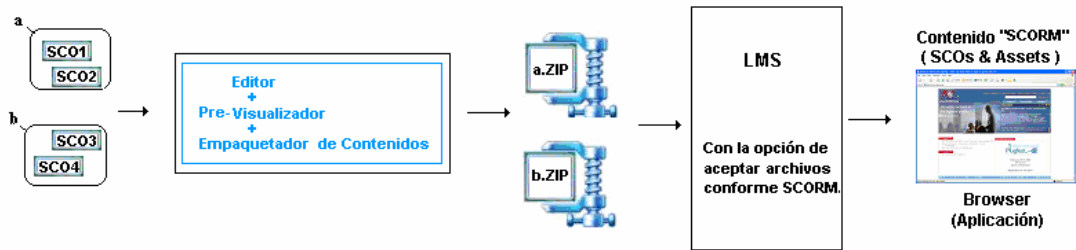


Figura 3.6. Modelo Generalizado del Editor ReLOAD

Una alternativa aún más sencilla, pero que no es gratuita, son las herramientas de autor específica para e-Learning, que ofrecen un entorno completo en el que diseñar los contenidos no requieren de programación, “empaquetando” de forma automática los contenidos con ellas creados y generando todo el código necesario para el seguimiento de la actividad del alumno sobre el contenido en cualquier LMS compatible, claro que al no ser gratuitas sólo se permite descargar versiones de prueba por un tiempo (días) limitado.

3.7 Capacidades y características

Los dispositivos móviles 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.

Dos aspectos importantes que se deben tener en cuenta al momento de distribuir los contenidos de m-Learning 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.

Los teléfonos móviles en su mayoría disponen de un navegador Web basado en WAP así como también:

- **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 (véase Figura 3.7), una tarjeta de red WiFi o Bluetooth, entre otros.

- **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 16Gb en almacenamiento externo y esto permite transportar gran volumen de información fuera de línea.

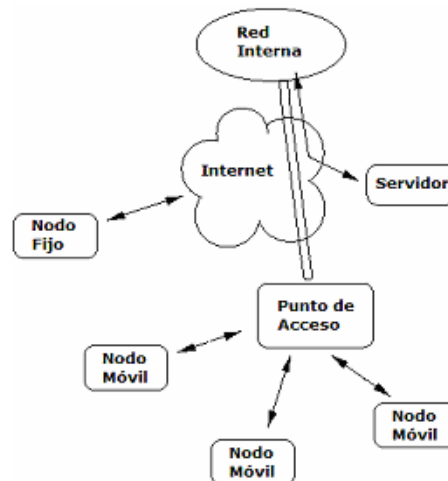


Figura 3.7. Arquitectura general del acceso a la red alámbrica e inalámbrica

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 Dispositivos móviles como cámaras, reproductores mp3 y scanner's pueden ayudar en el proceso de aprendizaje. Quinn [37] afirma que pronto no habrá distinción entre e-Learning y m-Learning, debido principalmente a que día a día los dispositivos se vuelven más poderos y la diferencia entre un dispositivo móvil y una computadora de escritorio desaparecerá.

3.8 Implicaciones pedagógicas

El tiempo potencial para la educación se puede incrementar [38], el uso de dispositivos móviles puede ser más oportuno para procesos de aprendizaje autónomo, la ubicación del usuario puede afectar el contenido educativo que éste recibe.

La aplicación le puede proveer además de ayudas en su proceso de aprendizaje autónomo, un ambiente virtual de aprendizaje colaborativo al usuario. En un ambiente verdaderamente abierto sería posible personalizarlo

hasta el punto donde el usuario puede convertirse en desarrollador del ambiente y de la educación.

3.9 Aplicación Móvil

En esta sección presentamos nuestra aplicación basada en dispositivos móviles. En particular hemos diseñado nuestros objetos móviles para teléfonos móviles. Esto obedece a que actualmente existen más de 3.5 billones de usuarios potenciales en el mundo que ya cuentan con dispositivos móviles y nos ofrece un gran campo de desarrollo. Además, la mayoría de los usuarios existentes son jóvenes y esto le da un valor agregado a las aplicaciones desarrolladas para móviles y más aún a los Objetos de Aprendizaje Móviles.



Figura 3.8 Interfaces principales de OAm

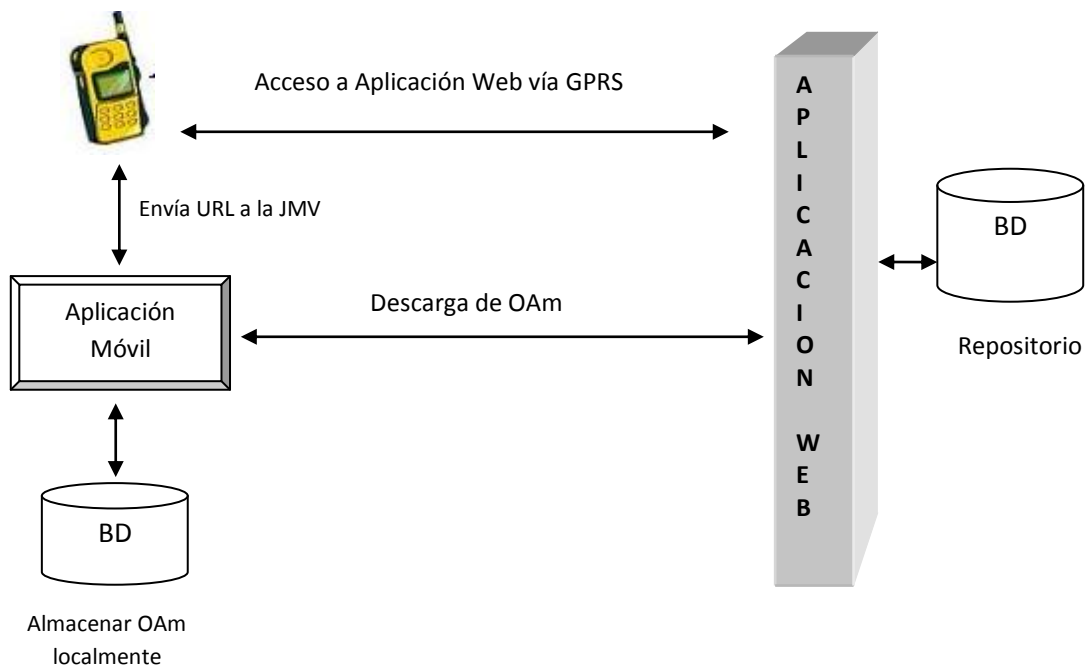
En la figura 3.8 mostramos las interfaces principales de nuestra aplicación. Como se puede observar, la primera es la pantalla de bienvenida, en la segunda pantalla se muestra como la aplicación le permite al usuario elegir buscar los OAm por Autor, Tema o Todos (ésta implica las dos opciones anteriores) y en la tercera pantalla se observa como el usuario puede seleccionar el formato del OAm por Texto, Audio, Video o Todos (implica las opciones anteriores).

Una vez elegido el Objeto de Aprendizaje Móvil, podemos observar detalles del mismo como lo son: nombre del autor, tipo de formato, tema e índice. Si el usuario así lo desea puede seleccionar la opción “Descargar” para obtener el Objeto de Aprendizaje Móvil en su dispositivo móvil (véase figura 3.8) y poder visualizar de forma Offline (sin conexión internet).



Figura 3.8 Selección del Objeto de aprendizaje a descargar (pantalla 1), Objeto de Aprendizaje Móvil ejecutándose (pantalla 2)

Cuando la aplicación móvil realiza una petición de búsqueda de OAm al Servlet, éste a su vez se conecta con la base de datos que contiene los OAm y realiza la búsqueda de dicha petición (figura 3.9), al obtener el resultado éste se envía a la aplicación móvil y se procede a realizar la descarga del OAm vía GPRS.



3.9 Lógica de Conexión entre aplicación móvil, aplicación web y repositorio.

Esta nueva forma de aprendizaje da al usuario muchos beneficios, algunos de estos son:

- El usuario puede aprender en cualquier lugar
- El usuario puede aprender en cualquier momento
- El usuario puede compartir (vía bluetooth) objetos de aprendizaje (conocimiento) de manera sencilla.
- El compartir este tipo de objetos de aprendizaje entre teléfonos móviles lo hace divertido
- Se puede trabajar Online u Offline.
- Incluso se pueden agregar imágenes en los objetos de aprendizaje.

Finalmente, tal como lo hemos mencionado al inicio de esta tesis los objetos de Aprendizaje Móvil abren una nueva era en el área del aprendizaje. La visión general que nos deja este trabajo es: la informática móvil es portable, rica en interactividad, conectividad total y alto poder de procesamiento. Permitiendo una fácil entrada de datos, reproductor de música, imágenes, video, etc.

3.10 Especificaciones Técnicas:

Para el funcionamiento de nuestra aplicación móvil, el dispositivo móvil que se utiliza deberá contar como mínimo con las siguientes características:

APIs:	
<u>Tecnología Java</u>	
JSR 139 Connected, Limited Device Configuration (CLDC) 1.0	
JSR 118 MIDP 2.0	
JSR 185 Java™ Technology for Wireless Industry	
JSR 172 J2ME™ Web Services Specification	
Navegador	
Detalles del Navegador:	WAP 2.0 (con GPRS activado)

Multimedia	
Formato de reproducción de video:	3GP
Formatos de Audio:	MP3
Memoria	
Espacio libre en memoria mínimo:	3 Mb

Capítulo 4:

DESARROLLO DE APLICACIÓN WEB

Nuestra aplicación Web (Servlet) fue desarrollada bajo la plataforma Netbeans (véase Capítulo 2) el cual nos da las herramientas necesarias para su construcción, conexión y ejecución.

Cuando se habla de aplicaciones para Web comúnmente escuchamos hablar de JAVA, y con ello sus aplicaciones más conocidas, los Applets, que son programas que se pueden cargar a través de una red y que se ejecutan de igual forma en cualquier plataforma, todo ello gracias a las potentes características de JAVA. Hasta hace poco, JAVA se utilizaba básicamente para dotar a las páginas WEB de una mayor interactividad mediante los Applets, y por tanto solo actuaba sobre el lado cliente. Pero el lado servidor también puede beneficiarse de todas las ventajas que ofrece JAVA, gracias a los *Servlets*.

Los *Servlets* se diferencian de los Applets básicamente en que se ejecutan en el servidor y en que no presentan ningún tipo de interfaz gráfica, puesto que se encargan de hacer el trabajo oculto.

Los servlets son programas atienden peticiones de un cliente teniendo al servidor como el encargado, pero escritos en Java y con la ventaja de explotar todas las bondades de java. Por ejemplo, un servlet puede ser responsable de tomar los datos de un formulario HTML y enviarlos a una base de datos para actualización de la misma.

4.1 Concepto de Servlet

La palabra Servlet deriva de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web. Por contraposición, un Servlet es un programa que se ejecuta en un servidor.

El uso más común de los Servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

4.2 Ventajas de los Servlets

Las ventajas de los Servlets ante los CGI son:

- **Desempeño:** Los Servlets son más rápidos que los CGI debido a que utilizan *threads* en lugar de *procesos*.

Una de las quejas más comunes de los CGI/Perl es el desperdicio de los recursos del sistema. En un ambiente típico CGI, cada petición crea un nuevo proceso el cual carga el interprete de Perl. El intérprete de Perl carga el script CGI, lo compila, y lo ejecuta. Además, si la aplicación se comunica con la BD, una nueva conexión será necesaria para ser hecha por cada CGI.

Los Servlets inician un nuevo thread, *hilo*, (más que un nuevo proceso) con cada petición. Cada servlet es cargado una vez y usado más y más. Nota que a diferencia de los CGI, los Servlets requieren que se cuente con una Máquina Virtual de Java (JVM) corriendo sobre el servidor todo el tiempo. Para sitios ocupados, ésta permite a los Servlets usar mucho menos los recursos del sistema e incrementar el desempeño.

- **Portabilidad:** Los servlets son tan portables como cualquier otra aplicación de Java.

Muchos sitios están buscando una solución de portabilidad. Tal vez desearían para su desarrollo usar Linux o Windows NT mientras su servidor de producción corra bajo Solaris. Ellos pueden querer vender su aplicación Web para muchos clientes como para plataformas sea posible. Hay versiones de Perl para muchas variantes de Unix como para Windows 95 y NT. Mientras que Perl para Unix no es ciento por ciento compatible con el Perl para NT, los CGIs escritos en Perl pueden ser escritos para trabajar sobre ambos. El llevar CGIs a Windows NT no es difícil de hacer, pero nunca funcionarán sin realizar al menos una modificación. Por otro lado, los CGI pueden contener comandos de Unix que no trabajarán sobre Windows NT.

La portabilidad con los servlets de Java es más simple. Java fue diseñado para ser portable a través de todas las plataformas, permitiendo que las aplicaciones sean movidas fácilmente de un sistema operativo a otro.

- **Seguridad:** La seguridad puede ser un problema cuando se desarrollan CGIs. El área de más preocupación es el proceso de estrada del usuario. Este podría ser desde formas o desde datos dentro de un URL. Muchos CGIs escritos en Perl son vulnerables a ataques donde la finalidad del usuario es trucar el CGI ejecutando algún comando sobre el servidor. Los Servlets no están en riesgo de correr comando de shell no planeados. Los lenguajes compilados como Java (o C) proveen mejor seguridad que los lenguajes que interpretan scripts.

Los Servlets son archivos de clases compilados mientras que un CGI/Perl es manipulado en su forma de código fuente. Dependiendo quien tenga acceso al servidor Web, se puede elegir entre instalar o no el código fuente.

- **Desarrollo:** El desarrollo con Perl es simple. Lo único que se necesita es un editor y mucha documentación de Perl.. Mientras que a muchos desarrolladores les agrada este ambiente, hay una creciente proliferación de herramientas IDE (Integrated Development Environment) para desarrollo. En ese campo, Java por encima de Perl. Hay IDEs disponibles desde Sun, Microsoft, Symantec y muchas otras, proporcionando simples entornos GUI. Las IDEs pueden proveer útiles herramientas fáciles de aprender para nuevos desarrolladores.

Compartiendo el trabajo entre los desarrolladores es además es más fácil con Java, o puede asignarse en paquetes individuales. Cuando se trabaja con una aplicación CGI/Perl donde dos o más desarrolladores estuvieron trabajando simultáneamente, la aplicación normalmente es manejada de un desarrollador a otro. El desarrollo de aplicaciones Java es mucho mejor para proyectos largos.

4.3 Arquitectura de los Servlets

El principal componente de la Servlet API es la interfaz Servlet. Todos los Servlets implementan esta interfaz directamente, por medio de la extensión de la clase que la implementa, **HttpServlet**. Esta interfaz está provista de métodos que manipulan a los servlets y la comunicación con sus clientes. Ésta clase

contiene una serie de métodos predefinidos, que son invocados en respuesta a eventos:

1. Inicialización. Se carga en memoria una instancia del servlet. La instanciación se produce la primera vez que el servlet es cargado en memoria por parte del servidor de aplicaciones. Como consecuencia de este evento se ejecuta el siguiente método:

```
public void init(ServletConfig config) throws ServletException
```

2. Petición: el cliente realiza la petición al servlet. Dentro de la versión 1.1 y en función del método de invocación usado en HTML (DO o GET) se ejecutará uno de los siguientes:

```
public void doGet(HttpServletRequest petición, HttpServletResponse  
                  respuesta) throws ServletException, IOException
```

```
public void doPost(HttpServletRequest petición, HttpServletResponse  
                  respuesta) throws ServletException, IOException
```

El objeto 'petición' es enviado por la JVM del servidor de aplicaciones y representa (encapsula) información de la petición HTTP. Más adelante veremos para que puede servir el objeto 'respuesta', por de pronto nos sirve decir que en este objeto se encapsulan los servicios para generar la salida. También tiene la posibilidad de usar:

```
public void service(HttpServletRequest petición, HttpServletResponse  
                  respuesta) throws ServletException, IOException
```

3. Destrucción del servlet: la contraimagen de *init()*, el servlet va a ser destruido (normalmente por la detención del servidor). No se invoca cuando se termina la petición, sino cuando se va a descargar el servlet de memoria:

```
public void destroy()
```

Cuando un Servlet es llamado desde un cliente, este recibe dos objetos: **ServletRequest** y **ServletResponse**. La interfaz ServletRequest se encarga la comunicación desde el cliente al servidor, mientras que la interfaz ServletResponse atiende la comunicación desde Servlet al cliente.

La interfaz ServletRequest permite al servlet acceder a información como, los nombres de parámetros pasados por el cliente, el protocolo usado por el cliente, y los nombres de los host remoto que hacen la solicitud y el servidor que la recibe. Esta interfaz permite a los Servlets el acceso a métodos que permiten manejar la presentación de la respuesta como salida en el navegador, a través de los cuales consiguen los datos desde el cliente que usa protocolos como HTTP POST, etc.

La interfaz ServletResponse proporciona al Servlet los métodos para contestarle al cliente. Permite al Servlet configurar la forma de salida de los datos para el cliente, ServletOutputStream que permite enviar la replica de datos como respuesta. Las subclases de ServletResponse le dan más capacidad al servlet para responder.

Por ejemplo:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ObjetosAprendizaje extends HttpServlet{
    PrintWriter out;
    public void init(){ } // motor del servlet para cargar la petición

    public void doGet(HttpServletRequest req,HttpServletResponse res)throws
    ServletException,IOException{
    try{
        String tipo=req.getParameter("formato ");
        String busqueda=req.getParameter("busqueda");
        res.setContentType("text/html");
```

```

        PrintWriter out = response.getWriter();
        out.println("Busqueda:" + busqueda + "\n Formato: "+formato);
        out.close();

    } catch(Exception ee){out.write(ee.toString());}

Public void doPost(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException{

try{
    doGet(request, response);
} catch (Exception e) { e.printStackTrace(); }
}

Public void destroy(){ //destruimos y liberamos los recursos abiertos
}

```

3.1 Estructura básica de un Servlet

Tanto `doGet` como `doPost` reciben como parámetros objetos de tipo `HttpServletRequest` y `HttpServletResponse`.

De la clase `HttpServletRequest` nos interesa el método `getParameter(String nombreParametro)`, que devuelve el valor del parámetro de `Búsqueda` y `Formato`. Por ejemplo, si visitáramos la URL

`http://localhost:8080/mi_servlet/OAm?formato=MP3&busqueda=REDES`

(Vemos que es una petición GET, porque los parámetros se pasan en la URL) el Servlet “ObjetosAprendizaje” (*Figura 3.1*) podría llamar a `getParameter("formato")` para obtener el Formato o `getParameter("Busqueda")` para obtener la Búsqueda.

Las clases e interfaces descritas conforman a un Servlet básico. Pero existen métodos adicionales que provee la API con la capacidad para controlar sesiones o múltiples conexiones, entre muchas más aplicaciones

El ciclo de vida de un Servlet se divide en los siguientes puntos:

1. El cliente solicita una petición a un servidor vía URL.
2. El servidor recibe la petición.
 1. Si es la primera, se utiliza el motor de Servlets para cargarlo y se llama al método `init()`.
 2. Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un Servlet puede manejar múltiples peticiones de clientes.
 3. Se llama al método `service()` para procesar la petición devolviendo el resultado al cliente.
 4. Cuando se apaga el motor de un Servlet se llama al método `destroy()`, que lo destruye y libera los recursos abiertos.

4.4 Ejemplos de aplicación de los Servlets

Algunas de las aplicaciones de Servlets incluyen:

Procesamiento de datos enviados con HTTPS a partir de una forma HTML, incluyendo la orden de compra o datos de una tarjeta de crédito. Un Servlet como este podría ser parte de una orden de entrada al sistema de procesamiento de datos, haciendo la respectiva actualización en la base de datos, y quizá hasta un sistema de pago en línea.

Interacción múltiple entre personas. Un Servlet puede manipular múltiples peticiones al mismo tiempo; podrían sincronizar dichas solicitudes de sistemas que dan soporte, por ejemplo conferencias en línea.

Redireccionamiento de peticiones. Los Servlets pueden reenviar las peticiones a otros servidores y Servlets. Esto permite que al ser usados se balancee la carga entre varios servidores que atiendan al misma tarea, según el tipo de proceso que deban atender.

Dado que pueden manejar múltiples peticiones en forma concurrente, es posible implementar aplicaciones de colaboración, como por ejemplo una aplicación de videoconferencia.

Capítulo 5:

REPOSITORIO DE OBJETOS DE APRENDIZAJE MÓVIL

5.1 Antecedentes

Como ya se ha mencionado, nuestro sistema está dividido en dos etapas; una de ellas es la Aplicación Móvil y la otra un Repositorio de Objetos de Aprendizaje Móvil.

Actualmente las universidades utilizan software para administrar cursos, éstos traen herramientas de colaboración que generalmente no son reutilizables en otros ambientes, por esto se diseña y construye un repositorio para almacenar herramientas de colaboración bajo el concepto de objetos de aprendizaje reutilizables permitiendo a cualquier usuario registrado contribuir en su construcción agregando recursos. El repositorio clasifica los objetos de acuerdo al uso pedagógico y existen funcionalidades como buscar objetos, agregar, registrarse como usuario entre otras.

En este capítulo se presenta la construcción de un Repositorio de Objetos de Aprendizaje Móviles, los que incluyen objetos del tipo JAR, MP3 y 3GP (formatos soportados por la mayoría de Dispositivos móviles).

A partir de los antecedentes anteriores se diseñó y construyó un repositorio de OAm como herramientas de apoyo al proceso de enseñanza/aprendizaje adaptable a las nuevas estrategias metodológicas [39] y complemento de nuestro proyecto. De esta forma para desarrollar, agregar y administrar objetos en un repositorio es necesario hacer una clasificación de éstos. Para ello se usó una clasificación de acuerdo a su formato (tipo de archivo) en este caso.

5.2 Servidor Web

El repositorio fue construido con la utilización de las herramientas: Apache, MySql y PHP. Estas herramientas en su conjunto se instalaron bajo el software WevServ el cual fue utilizado en este caso y se describen a continuación:

➤ **WebServ**

WebServ es un software gratuito que se compone de varios software y que pretende hacernos la vida más fácil. Se compone del mejor servidor Web, el Apache 2. Como servidor de Base de Datos (BD) tiene MySQL 5. También trae soporte PHP para el Apache, lo que permite hacer nuestra computadora en una mesa de pruebas de nuestras páginas web en PHP. También trae otras herramientas muy útiles como PHPMyAdmin.

➤ **Apache**

(Acrónimo de "a patchy server"). Servidor web de distribución libre y de código abierto, siendo el más popular del mundo desde abril de 1996, con una penetración actual del 50% del total de servidores web del mundo (agosto de 2007).

La principal competencia de Apache es el IIS (Microsoft Internet Information Services) de Microsoft. Apache fue la primera alternativa viable para el servidor web de Netscape Communications, actualmente conocido como Sun Java System Web Server. Apache es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la Apache Software Foundation. La aplicación permite ejecutarse en múltiples sistemas operativos como Windows, Novell NetWare, Mac OS X y los sistemas basados en Unix.

Apache es principalmente usado para servir páginas web estáticas y dinámicas en la www. Apache es el servidor web del popular sistema XAMP, junto con MySQL y los lenguajes de programación PHP/Perl/Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP, etc.

Características de Apache

- Soporte para los lenguajes perl, python, tcl y PHP.
- Módulos de autenticación: mod_access, mod_auth y mod_digest.
- Soporte para SSL y TLS.
- Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- Permite autenticación de base de datos basada en SGBD.

➤ **Mysql**

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

➤ **PHP**

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

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

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

5.3 Requisitos de Hardware y Software

Los requisitos de hardware y software, para el adecuado funcionamiento del repositorio, para el cliente deben ser como mínimo:

- Software para el Cliente *Web*:
 - Internet Explorer versión 5.5 o superior.
 - Netscape 6.0 o superior.
- Hardware para el Cliente *Web*:
 - 32 MB RAM.
 - Pentium u otro.

5.4 Construcción del Repositorio

El repositorio se construyó usando lenguaje de modelado unificado (UML). A continuación podemos ver el diagrama (ver figura 5.1) que describe los casos de uso y actores del sistema.

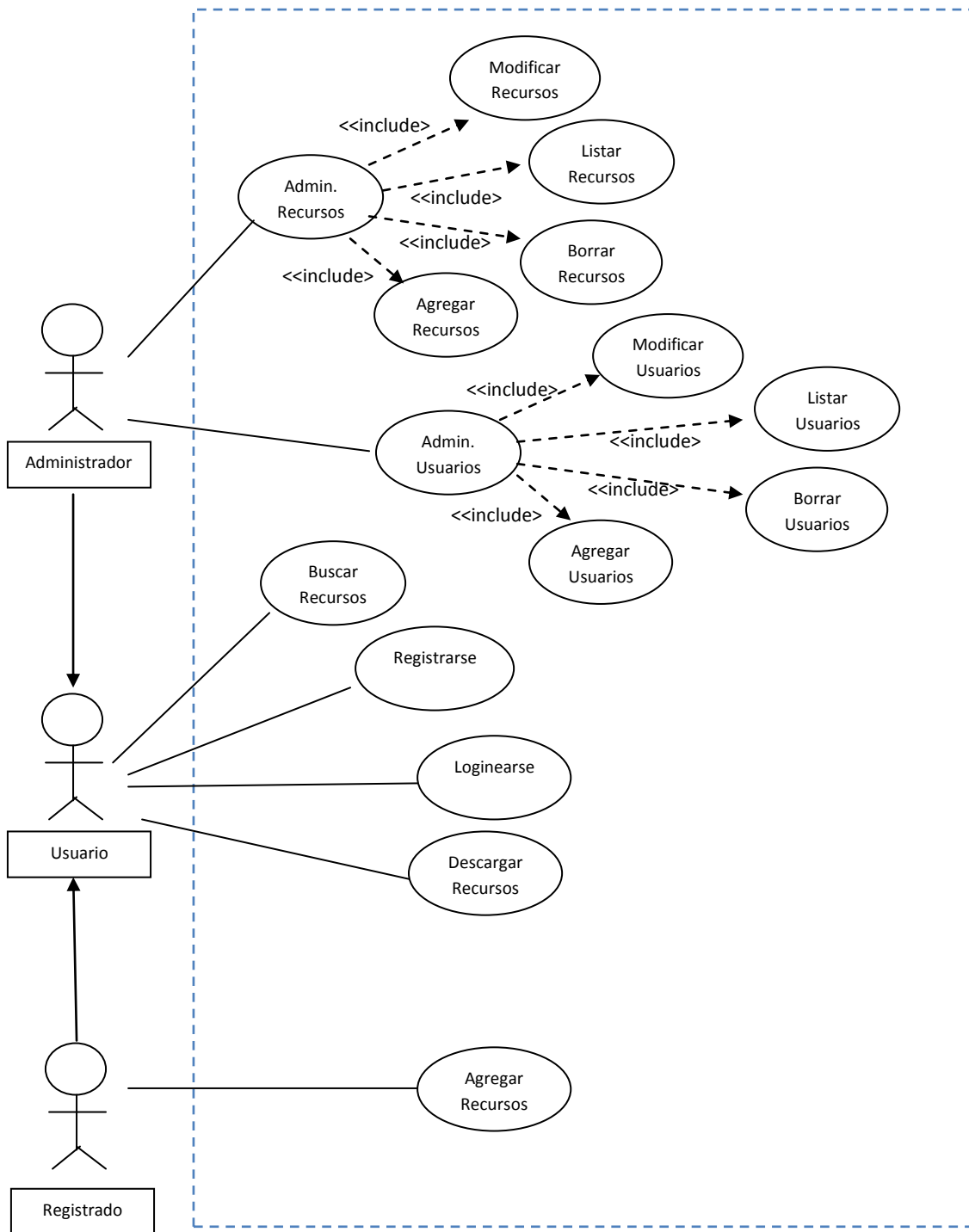


Figura 5.1. Diagrama de Casos de Uso y Actores del Sistema.

La figura 2 muestra un diagrama de secuencia donde se detallan la secuencia de eventos del caso de uso “Buscar Recurso” (ver figura 5.2). Del mismo modo en la figura 5.3 se desarrolló un siguiente diagrama de secuencia que detalla el caso de uso “Agregar Recurso” (ver figura 5.3). Mientras que en la en la figura 5.4 se describe el caso de uso “loginearse” (ver figura 5.4).

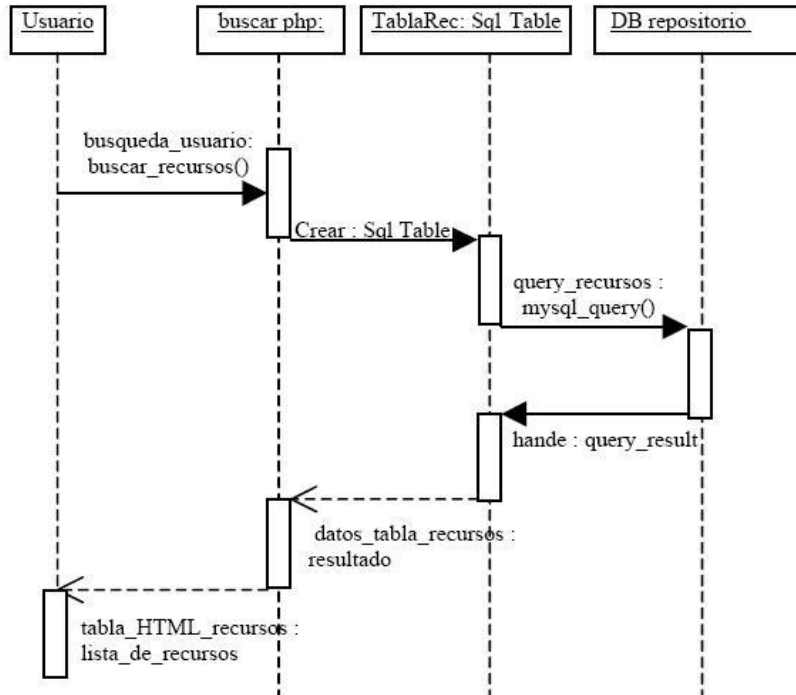


Figura 5.2 Diagrama de secuencia de Buscar Recurso

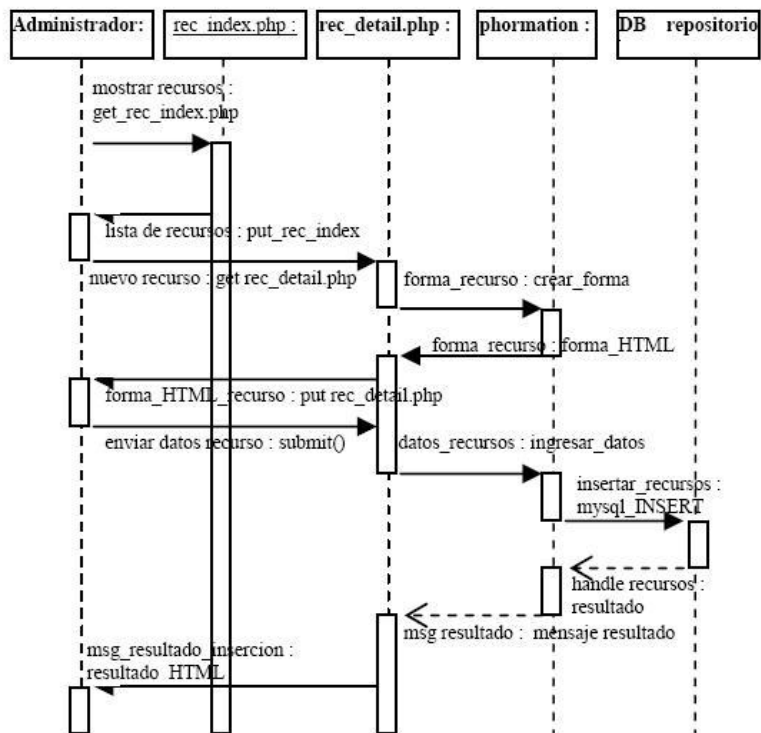


Figura 5.3 Diagrama de secuencia de Agregar Recurso

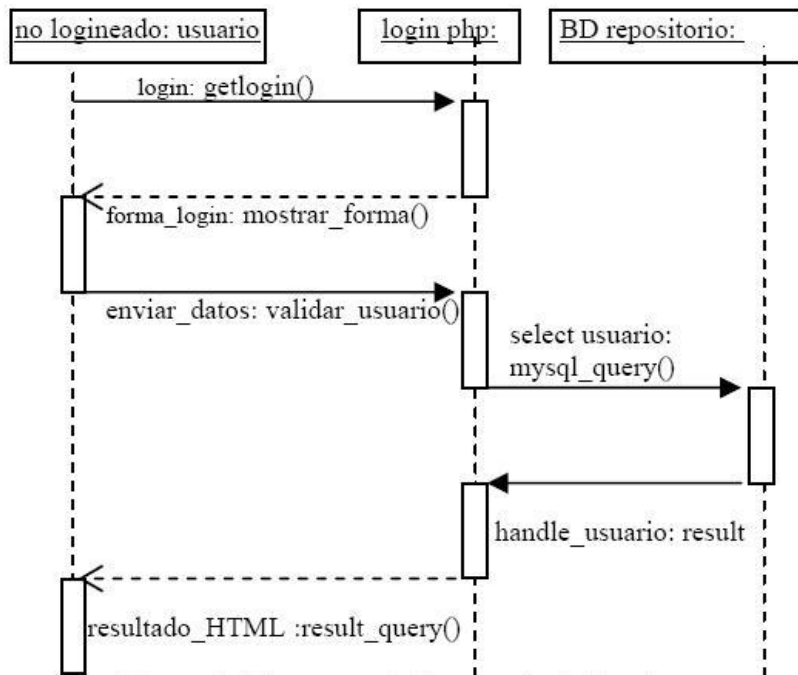


Figura 5.4 Diagrama de secuencia de Loginearse

5.4 Funcionalidades del Repositorio

El repositorio cuenta con las siguientes funcionalidades para el usuario:

- Registro de usuarios.
- Buscar recursos.
- Agregar recursos.

La dirección URL de acceso al repositorio es:

http://kompessorx.homeip.net/mi_servlet, en donde la página inicial se ilustra en la figura 5.5

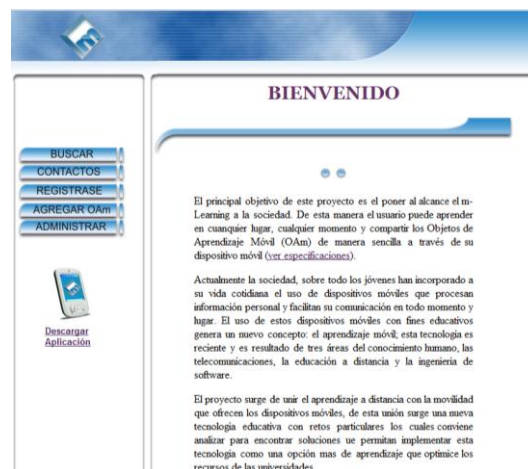


Figura 5.5 Página Principal

- **Registro de Usuarios**

Para registrarse como usuario en el repositorio (ver figura 5.6) sólo son necesarios los datos personales del usuario.

Figura 5.6 Ventana de Registro Nuevo Usuario

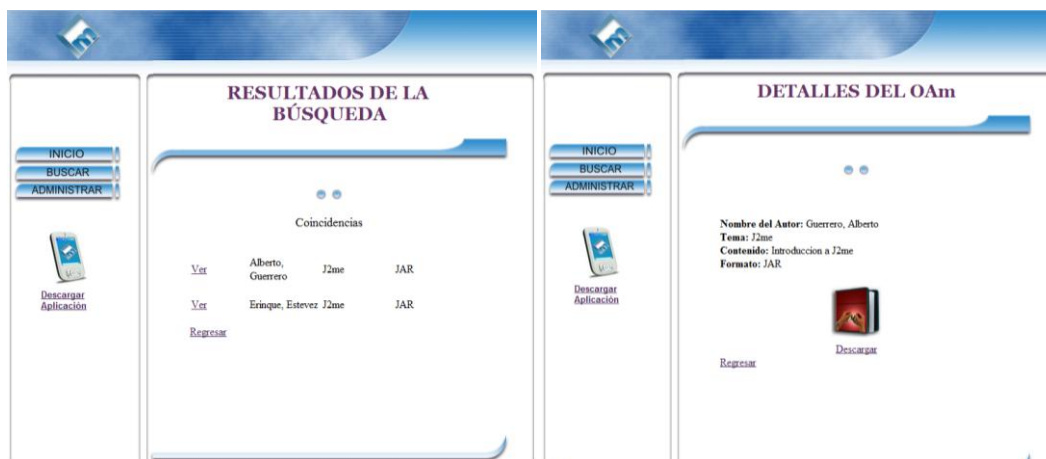
- **Buscar Recursos.**

Para realizar la búsqueda de un OAm en el repositorio, este cuenta con las mismas opciones que contiene la aplicación móvil, como lo es la elección de búsqueda por Tema, Autor y Todos, así como también la elección del Formato (Audio, Texto, Video, Todos). (figura 5.7).

Figura 5.7 Sistema de Búsqueda del repositorio

De los resultados obtenidos (figura 5.8 primera imagen), sólo basta con hacer clic sobre la opción “descargar” al recurso que se desee para acceder a descargarlo en la computadora, inmediatamente se despliega la ventana que

se aprecia en figura 5.8 (segunda imagen). Esta ventana cuenta con la descripción general del recurso como lo son: Nombre del autor, tema, contenido del OAm y formato.



5.8 Ventanas respuesta de consulta (primera imagen) y descarga del recurso (segunda imagen)

- **Agregar Recursos.**

El usuario registrado agrega recursos al repositorio a través de una ventana especial (ver figura 5.9), debe agregar la descripción general (índice) del OAm, el tipo de archivo al cual pertenece, nombre completo del autor y tema. Para poder realizar esto el usuario debe estar registrado.



Figura 5.9 Ventana Agregar Recurso

5.6 Resultados

Se construyó un repositorio con un entorno amigable que permite a los usuarios encontrar recursos para el aprendizaje. Éstos pueden seleccionar los

OAm que mejor se acomoden a sus estrategias de enseñanza/aprendizaje. Los OAm para uso pedagógico, permiten hacer una clasificación representativa de herramientas de apoyo al aprendizaje. Utilizando éste concepto se facilita la selección de recursos que estarán disponibles en el repositorio.

El modelado con lenguajes como UML nos permitió modelar un entorno flexible y compatible con las nuevas especificaciones relacionado a los ambientes de recursos de aprendizaje en la Web. Un repositorio de herramientas de aprendizaje tiene la ventaja de ir mejorando en el tiempo en base a la colaboración de recursos entre los usuarios.

En conclusión, después de poner en prueba dicho repositorio de OAm con 20 usuarios, los cuales se registraron y disfrutaron de nuestro repositorio al subir y buscar OAm de manera fácil, rápida y de acuerdo a sus necesidades e intereses de aprendizaje. De ésta manera, el repositorio cumplió con el objetivo principal de proporcionarle al usuario un almacén, consulta y disposición de OAm para su descarga y uso en sus dispositivos móviles. Ésta prueba la podemos resumir en la siguiente tabla:

	Cantidad
Usuarios Totales	20
Consultas Realizadas	32
Usuarios registrados	20
OAm agregados al repositorio	12
OAm descargados	24
Dispositivos Móviles	17*

Nota: Los 17 dispositivos móviles utilizados cubrían los requerimientos mínimos para el buen funcionamiento de los OAm.

CONCLUSIONES Y TRABAJO A FUTURO

A lo largo de esta tesis, hemos presentado una serie de nuevas tecnologías de la información y comunicación que están revolucionando muchas de las áreas que cotidianamente hemos cultivado. De manera particular, en el presente trabajo nos hemos dado a la tarea de desarrollar un proyecto en el área de la educación incorporando para esto las tecnologías móviles.

Hemos logrado integrar un conjunto de herramientas de actuales que nos permiten desarrollar y llevar el conocimiento a los usuarios de teléfonos móviles de una manera sencilla. La tecnología NetBeans nos permite incorporar un ambiente de desarrollo para dispositivos móviles, particularmente teléfonos móviles, permitiéndonos desarrollar la herramienta de visualización de objetos de aprendizaje móviles.

Por otro lado, en el presente trabajo de tesis usamos algunas herramientas gestoras de contenido que nos permitieron crear Objetos de Aprendizaje que posteriormente convertimos a aplicaciones que podamos visualizar en nuestros teléfonos móviles.

Además, hemos presentado una aplicación novedosa que incorpora nuevas características al aprendizaje actual. Estas características son generadas por una nueva área que ha crecido ampliamente en los últimos años gracias al desarrollo de las tecnologías móviles. Esta nueva propuesta ha cambiado radicalmente el aprendizaje que sin embargo reflejan buenos resultados en su uso pues de 20 usuarios que han empleado nuestra herramienta, todos se vieron atraídos por esta nueva herramienta de aprendizaje que incluso les permite estudiar en todo momento, en cualquier lugar y en “casi” cualquier dispositivo móvil. Todos nuestros usuarios disfrutaron y compartieron sus diferentes OAm haciendo que la actividad del aprendizaje sea aceptada de una manera natural. Esto obedece básicamente a que esta propuesta traslada el aprendizaje a ser una actividad más de nuestra vida diaria.

De esta manera hemos demostrado la viabilidad de los Objetos de Aprendizaje en Dispositivos Móviles.

En un futuro se deja a consideración el incrementar los formatos de los OAm desarrollados, así como también el diseño y mejoras al repositorio de OAm.

BIBLIOGRAFÍA

- [1] Wiley, D.A. "Connecting learning objects to instructional theory: A definition, a metaphor, and a taxonomy". En D. A. Wiley (Ed.) The Instructional Use of Learning Objects. Bloomington, IN: Association for Educational Communications and Technology. 2001
- [2] Wiley, D. A. "Creating Learning Objects". En Kovalchick & Dawson (Eds.). Educational Technology. An Encyclopedia. Santa Bárbara: ABC-CLIO. En <http://coe.sdsu.edu/EET/>, 2009
- [3] Hodgins, W. Web. <http://www.learnativity.com/waynehodgins.html>, 2001
- [4] Koper, R. "Modeling units of study from a pedagogical perspective. The pedagogical meta-model being EML". Versión 2. En <http://dspace.ou.nl/bitstream/1820/36/1/Pedagogical%20metamodel%20behind%20EMLv2.pdf>, 2001
- [5] Merrill, M. D. "Components of Instruction. Towards a Theoretical Tool for Instructional Design". <http://cito.byuh.edu/merrill/text/papers/Components.PDF> 2001
- [6] Boyle, T. "Towards a Theoretical Base for Educational Multimedia Design". <http://www-ijme.open.ac.uk/2002/2/boyle-02-2.pdf>, 2002
- [7] Downes, S. "Learning Objects: Resources for Distance Education Worldwide". International Review of Research in Open and Distance Learning. En <http://www.irrodl.org/content/v2.1/downes.html>,_Vol 10, No 4 (2009)
- [8] Banks, B. "Learning Theory and Learning Objects". <http://myweb.tiscali.co.uk/benadey/NEW%20FDLWEBSITE/html/company/papers/l-theory-l-objects.pdf>, 2001
- [9] CedMA (Computer Education Managers Association) <http://www.cedma.org>, 2009
- [10] Duncan, C. Granularization. En A. Littlejohn (Ed.), "Reusing Online Resources: A Sustainable Approach to E-learning" (pp 12-19). Open & Flexible Learning Series. London , Sterling : VA Taylor & Francis, 2003
- [11] Miguel Zapata Ros. "Secuencing of contents and learning objects psrtrt. III". RED: Revista de educación a distancia. ISSN (versión en línea): 1578-7680, España <http://redalyc.uaemex.mx/redalyc/pdf/547/54701502.pdf>, 2006

- [12] Head y Taylor. "Los objetos de aprendizaje como potencial herramienta para un desarrollo docente intercultural". III Congreso ONLINE - Observatorio para la cibersociedad. 2007
- [13] G. Kennedy, T. Petrovic, M. Keppell: "The development of multimedia evaluation criteria and a program of evaluation for commanded learning", pp 407-415, ASCILITE'98
- [14] Michael Wallace, "Learning objects as a potential means of intercultural teacher development". Action Research for Language Teachers, Cambridge, UK: Cambridge University Press. 2006
- [15] LTSC: "Learning Technology Standards Committee".
<http://www.ieeeltsc.org:8080/Plone>
- [16] Jamie Jaworski, "Java 2 Platform Unleashed", ISBN:0672316315, 2000
- [17] John W. Muchow, "Core J2ME Technology, The Sun Microsystems Press", Java Series, 2002.
- [18] Adam Myatt, "Pro NetBeans IDE 5.5", Enterprise Edition, 2002.
- [19] F. Zacarias, F. Lozano, R. Cuapa and A. Vázquez, "English's Teaching based on New Technologies". The International Journal of Technology, Knowledge & Society, Northeastern University in Boston, Massachusetts, USA. ISSN: 1832-3669, Common Ground Publishing, USA 2008.
- [20] F. Zacarias, F. Lozano, R. Cuapa, A. Vázquez and D. Zacarias, u-Teacher: "Ubiquitous Learning Approach Springer's Lecture Notes of Computer Science (LNCS)" ISSN: 0302-9743, ISBN: 3-540-20965-4. Vol. 5093, Nanjin, China 2008.
- [21] F. Zacarias, A. Sánchez J., D. Zacarías, A. Méndez, R. Cuapa, "Financial Mobile System based on Intelligent Agents". Austrian Computer Society book series, 2006.
- [22] Sergio Gálvez Rojas, Lucas Ortega Díaz. "Java a Tope: j2me (java 2 micro edition).Edición Electrónica" Depósito Legal: MA-1769-2003 ISBN: 84-688-4704-6. Universidad de Málaga. 2003
- [23] Manuel J. Prieto, "Introducción a J2ME (Java 2 MicroEdition)", 2008
- [24] Jackwind Li Guojie, "Build your stock with J2ME", developerWorks, your source for great tutorials: <http://www.ibm.com/developerworks/>

- [25] Michael Morrison, "Wireless Java with J2ME in 21Days", Edit: SAMS, International Standard Book Number: 0-672-32142-4, 201 West 103rd St., Indianapolis, Indiana, 46290 USA, 2007
- [26] Mario Jaramillo Restrepo, "The Mobile Information Device Profile and MIDlets, Part 5" Ingeniería de Sistemas-Universidad de Antioquia. Mayo – 2006
- [27] Desmond Keegan, "The future on Learning: from e-learning to m-learning". FernUniversitat.
- [28] Konstantinov Andrey, mjSoftware, <http://www.mjsoft.nm.ru>, 2007
- [29] Advanced Distributed Learning, "SCORM® 2004 4th Edition Run-Time Environment (RTE) Version 1.1". Versión en línea: http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/SCORM%202004%204th%20Ed%20V1.1/Documentation%20Suite/SCORM_2004_4ED_v1_1_Doc_Suite.zip, 2001
- [30] Advanced Distributed Learning: "SCORM® 2004 4th Edition Content Aggregation Model (CAM) Version 1.1". Versión en línea: http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/SCORM%202004%204th%20Ed%20V1.1/Documentation%20Suite/SCORM_2004_4ED_v1_1_Doc_Suite.zip, 2009
- [31] Álvarez G. Luís A.- Espinoza, Daniela - Duarte, Mauricio.: "Repositorio de Objetos de Aprendizaje Multimediales basados en el estándar SCORM ® 2004". VIII Congreso de Nuevas Tecnologías y Aplicaciones Informáticas. La Habana – Cuba. 9 al 13 de mayo del 2005. Versión en línea. <http://www.inf.uach.cl/lalvarez/publicaciones.htm>, 2004
- [32] Álvarez G. Luís A.- Mónica del C., Gallardo G.: "Diseño de un Repositorio de Objetos de Apoyo al Aprendizaje Colaborativo". CISCi 2004. Orlando - USA. 21 y 25 de julio del 2004. Versión en línea. [http://www.iiisci.org/journal/CV\\$/risci/pdfs/P157490.pdf](http://www.iiisci.org/journal/CV$/risci/pdfs/P157490.pdf), 2004
- [33] Álvarez G. Luís A.- Mónica del C., Gallardo G.: "Repositorio de objetos de Apoyo al aprendizaje Colaborativo". TISE '03. Santiago - Chile. 26, 27 y 28 de noviembre del 2003. Versión en línea. http://www.tise.cl/archivos/tise2003/papers/repositorio_de_objetos.pdf, 2003
- [34] "Draft Standard for Learning Object Metadata. IEEE 1484".12.1-2002, 15 July 2002. Versión en línea. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, 2002

- [35] Merrill, M.D., Li, Z. & Jones, M. "Instructional transaction theory: An introduction. Educational Technology", Versión en línea. http://www.id2.usu.edu/Papers/ITT_Intro.PDF, 1991
- [36] Wiley, D. A. "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy". In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Versión en línea. desde: <http://reusability.org/read/chapters/wiley.doc>, 2000
- [37] Quinn, C. "mLearning: Mobile, Wireless, In-Your-Pocket Learning". Line zine. Learning in the new economy. <http://www.linezine.com/2.1/features/cqmmwiyp.htm>, 2000
- [38] Malinen, J., Kari, H. & Tiusanen, M. "Wireless networks and their impact on network-based learning content. Enabling network-based learning". Helsinki University of Technology. <http://firgoa.usc.es/drupal/node/5560>, 1999
- [39] Gallardo G., Mónica: "Diseño y Construcción de un Repositorio de Herramientas de Apoyo a la Enseñanza/Aprendizaje Virtual". Tesis de grado para optar al Título Profesional de Ingeniero Civil en Informática. Universidad Austral de Chile. Septiembre 2006.