



**BENEMÉRITA UNIVERSIDAD**

---

**AUTÓNOMA DE PUEBLA**

**Facultad de Ciencias de la Computación**

**“Uso de tecnologías móviles para  
aprendizaje para la vida”**

**Tesis Profesional que para obtener el título de:**

**Ingeniero en Ciencias de la Computación**

**Presenta:**

**Francisco Javier Lozano Torralba**

**Asesor:**

**Dr. Fernando Zacarías Flores**

**Primavera 2010**

# Índice

Introducción .....	4
Objetivo General .....	5
Objetivos Específicos .....	5
Capítulo I: Marco Teórico .....	6
1.1 Descripción de la Plataforma Java .....	6
1.2 Características del lenguaje de programación Java .....	8
1.3 Java Platform, Micro Edition (Java ME) .....	9
1.3.1 Configuraciones .....	10
1.3.2 Perfiles .....	14
1.3.3 MIDlets .....	17
1.3.4 Paquetes opcionales .....	19
1.4 Descripción de herramientas utilizadas .....	19
1.4.1 NetBeans IDE .....	19
1.4.2 Java Wireless Toolkit .....	21
1.4.3 Apache HTTP Server .....	22
1.4.4 Adobe Dreamweaver .....	24
1.4.5 EasyPHP .....	24
Capítulo II: Análisis de Requerimientos .....	26
2.1 Aprendizaje para la vida (Life Long Learning) .....	26
2.2 eLearning .....	27
2.3 Modelado de aplicaciones con UML .....	28
2.4 Diagramas de casos de uso .....	30
2.4.1 Administrador .....	31
2.4.2 Usuario .....	37
2.4.3 Agente .....	40
Capítulo III: Diseño del sistema .....	42
3.1 Enseñanza de un segundo idioma .....	42
3.2 Aprender en cualquier lugar y en cualquier momento .....	43
3.3 El aprendizaje necesita acción .....	43
3.4 El entorno de aprendizaje ubicuo .....	45
3.5 El modelo u-Teacher .....	47
Capítulo IV: Implementación del sistema .....	50
4.1 Arquitectura del sistema .....	50
4.2 Desarrollo de aplicaciones móviles en NetBeans .....	55
4.2.1 Mobility Pack de NetBeans .....	56
4.3 Clases .....	58
Conclusiones y trabajo a futuro .....	64
Bibliografía .....	65

## Índice de figuras

Figura 1: Ejemplo de clases en Java .....	8
Figura 2: APIs de CLDC 1.0 .....	12
Figura 3: APIs de CLDC 1.1 .....	12
Figura 4: API de CDC 1.1.2 .....	14
Figura 5: API de MIDP 1.0 .....	16
Figura 6: API de MIDP 2.0 .....	17
Figura 7: Componentes de Java ME .....	19
Figura 8: Interfaz para desarrollo en móviles de NetBeans .....	21
Figura 9: Emulador de dispositivo móvil del Wireless Toolkit .....	22
Figura 10: Página de la configuración de Apache (generada mediante PHP) .....	23
Figura 11: Interfaz de Dreamweaver .....	24
Figura 12: Página en el directorio WWW de EasyPHP, nótese el URL .....	25
Figura 13: Diagrama de Booch (Grady Booch) .....	28
Figura 14: Diagrama OMT (James Rumbaugh et. al.) .....	29
Figura 15: Diagrama OOSE (Ivar Jacobson) .....	29
Figura 16: Evolución de UML .....	30
Figura 17: Ejemplo de herramienta UML para casos de uso .....	31
Figura 18: Diagrama de casos de uso del Administrador .....	32
Figura 19: Diagrama de casos de uso del Usuario .....	37
Figura 20: Descarga de contenido .....	38
Figura 21: Envío y recepción de datos .....	40
Figura 22: Diagrama de casos de uso del Agente .....	40
Figura 23: Diagrama de casos de uso del sistema .....	41
Figura 24: Arquitectura general del sistema .....	51
Figura 25: Interfaz web .....	52
Figura 26: Pantalla inicial .....	53
Figura 27: Interfaz web e interfaz móvil mostrando el mismo contenido .....	53
Figura 28: Prácticas de videos .....	54
Figura 29: Envío de datos .....	54
Figura 30: Chat .....	55
Figura 31: Pantallas de la aplicación en Visual Mobile Designer .....	56
Figura 32: Flujo de pantallas y editor de código .....	57

## Introducción

El desarrollo de tecnologías móviles tiene una gran demanda actualmente, y esto obedece a el potencial que este tipo de dispositivos ofrecen al poder concentrar varias actividades cotidianas en un solo dispositivo. Algunas de sus características más relevantes son: diversos protocolos de comunicación entre dispositivos como Bluetooth, FireWire, puerto infrarojo, USB, acceso a Internet mediante WLAN, GSM EDGE (3G), GPRS, WAP, etc., capacidades de almacenamiento contadas ahora hasta en gigabytes que les permiten poder guardar la información generada en diversas actividades y grandes capacidades de procesamiento, permitiendo manejar contenido multimedia y rivalizando en algunas ocasiones con las de las PCs de hace algunos años, además de esto, los dispositivos móviles también pueden procesar datos a través de su máquina virtual Java. Esta máquina permite el desarrollo de múltiples programas que le brinden al usuario de estos un valor agregado. Todas estas capacidades en conjunto han hecho que actualmente existan alrededor de 3000 millones de usuarios de este tipo de tecnologías en el mundo, lo que les permite ser un nicho de oportunidad impresionante.

Las tecnologías de información y comunicación toman roles cada vez mas importantes en nuestras vidas, la sociedad y la economía se están adaptando a la ola de innovaciones asociadas con este cambio, resultando en nuevas actividades y nuevas formas de realizar las cosas. En la sociedad moderna basada en el conocimiento, el nuevo material en que se basan las actividades sociales y económicas es la información y la buena comunicación con nuestros socios.

La tendencia social actual hacia el aprendizaje durante el tiempo de ocio, junto con el rápido avance de la tecnología móvil han dado pie a la visión de permitir a la sociedad aprender en cualquier lugar y en cualquier momento. Es importante recordar que día a día, los usuarios demandan el poder continuar su forma de trabajo computacional y comunicaciones en cualquier lugar donde se encuentren de una manera sencilla e ininterrumpida.

América Latina se ha dado cuenta de las diferencias en infraestructura, recursos financieros y capacidad de desarrollo que se tienen con otras partes del mundo y que son necesarias para la realización de este tipo de sistemas, necesitamos invertir en investigación, en desarrollo y en educación y sobre todo en aprender a comunicarnos en otros idiomas, el más importante de estos el Inglés. Nuestro futuro crecimiento depende de la competitividad internacional de la

sociedad y de su habilidad para tomar nuevos mercados, crear nuevos productos y la aplicación creativa de nuevas tecnologías.

## Objetivo General

Los avances en la tecnología invariablemente traen con ellos oportunidades adicionales para facilitar y enriquecer las experiencias de los estudiantes, más aun, los usuarios ahora tienen ciertos sentimientos arraigados a sus dispositivos móviles: “No podría vivir sin mi teléfono; es parte de mi vida”.

Aprovechando esta nueva tendencia, nuestro proyecto trata de una novedosa propuesta basada en un aprendizaje apoyado en las actividades cotidianas que todos los usuarios de teléfonos móviles practican. La idea de esta propuesta es que el estudiante aprenda el idioma Inglés de una manera divertida y cotidiana, es decir, a través de canciones, de juegos, de puzzles, etc.

El objetivo general de este proyecto de tesis es el desarrollo de un sistema móvil que permita a los usuarios aprender un nivel de inglés básico mediante la descarga de de lecciones en sus teléfonos.

## Objetivos Específicos

- Sintaxis básica de oraciones (texto y voz).
- Ejercicios de pronunciación en donde los estudiantes puedan escuchar las pronunciaciones correctas.
- Videos donde los estudiantes ven y escuchan diálogos breves
- Formatos que validen la comprensión de los videos.
- Juegos que ayuden al aprendizaje.

## Capítulo I: Marco Teórico

Este capítulo expondrá conceptos teóricos que corresponden a las tecnologías utilizadas y mencionadas en la presente tesis.

### 1.1 Descripción de la Plataforma Java

El usuario final de cualquier tecnología suele demandar cada vez mayor capacidad a esta a un costo más bajo. El desarrollo e implantación de la conectividad global a través de la telefonía móvil supuso que el usuario necesitara y demandara mejores servicios de ésta y es ahí donde entra Java. Un dispositivo que en un principio solo servía para comunicarse por medio del habla, ahora sirve para enviar mensajes de texto, como grabadora, cámara de fotos o vídeo, etc., pero para ciertos usos, como el de incluir conectividad a internet o despliegado de documentos en el dispositivo, no se puede depender solamente del hardware sino que se necesita software. Java fue concebido en su origen para dar la posibilidad a cualquier dispositivo electrónico de consumo de tener un software y poder mejorar así su rendimiento. Es posible incluir Java en casi cualquier tipo de hardware existente, por pequeño que sea este.

Java es un número de productos de software y especificaciones creadas por Sun Microsystems, que juntos crean un entorno para desarrollo de aplicaciones independientes de la plataforma. Java puede ser utilizado en una amplia variedad de plataformas computacionales, como pueden ser teléfonos celulares, dispositivos móviles, PCs, servidores web, empresariales y de aplicaciones, e incluso en supercomputadoras. Por la parte de Internet, los applets o pequeñas aplicaciones creados en Java e incluidos dentro del código HTML se utilizan para mejorar el contenido y proveer seguridad al navegar por la World Wide Web.

Las sintaxis para programación en el lenguaje Java son sumamente parecidas a las de C y C++, pero eliminando algunas estructuras de bajo nivel como los apuntadores y el manejo detallado de memoria, aspectos manejados automáticamente mediante la máquina virtual de Java (JVM), permitiendo una programación más sencilla pero con resultados no menos complejos.

La máquina virtual de Java (JVM) es el componente crítico de la plataforma Java, y es el único componente que debe crearse para ejecutarse en una plataforma específica de hardware o software, esta máquina virtual permite

interpretar, compilar y ejecutar el código de un programa (bytecode), no importando en que plataforma se haya programado, asegurando la portabilidad de la plataforma Java y su fácil adaptación a las tecnologías emergentes.

Algunos productos que utilizan ampliamente tecnologías Java son los siguientes:

- Android (Sistema Operativo para móviles).
- Amazon Kindle (Lector de E-books).
- Discos Blu-Ray.
- Google Maps.
- Mifos (Plataforma para instituciones de microfinanzas).
- SAP NetWeaver (Servidor de aplicaciones).
- RuneScape (Juego Multijugador Masivo en Línea).

Existen tres grandes versiones de Java, cada una para un mercado diferente:

- **Java Platform, Standard Edition (Java SE):** Permite crear applets y aplicaciones para computadoras personales y servidores.
- **Java Platform, Enterprise Edition (Java EE):** Es el estándar Java para la industria, se utiliza para crear aplicaciones para empresas y web.
- **Java Platform, Micro Edition (Java ME):** Permite crear aplicaciones para dispositivos móviles y con memoria limitada.

Otras tecnologías de Java que vale la pena mencionar son:

- **JavaFX:** Esta versión de Java está optimizada para crear aplicaciones altamente interactivas específicas para Internet (Rich Internet Applications).

- **Java Runtime Environment (JRE):** De manera autónoma provee las librerías, la máquina virtual de Java y los componentes necesarios para ejecutar applets y aplicaciones creados con Java.

## 1.2 Características del lenguaje de programación Java

**API:** El Application Programming Interface (API) de Java cuenta con un gran número de paquetes que incluyen clases predefinidas que no deja de aumentar, presentando una gran riqueza en cuanto al tipo de funciones que se puede realizar.

<b>java.applet</b>	Provee las clases necesarias para la creación de applets.
<b>java.awt</b>	Contiene las clases requeridas para crear interfaces y manejo de gráficos e imágenes.
<b>java.io</b>	Contiene las clases necesarias para el manejo de entrada y salida de datos.
<b>java.security.cert</b>	Provee las clases necesarias para el manejo de certificados.
<b>java.sql</b>	Estas clases permiten el acceso y procesamiento de datos de un RDBMS.

**Figura 1: Ejemplo de clases en Java**

**Simplicidad:** Java ofrece toda la funcionalidad de un lenguaje como C y C++ pero sin las características menos usadas e inseguras como definiciones de tipos, macros y punteros, e incluye nuevas funciones como el manejo automático de memoria y un *garbage collector*, el cual realiza una limpieza dinámica de la memoria.

**Orientado a objetos:** Java trabaja los datos como objetos y con interfaces entre estos, soporta encapsulación, herencia y polimorfismo.

**Interconexión:** Java cuenta con capacidades de interconexión TCP/IP, y puede trabajar con distintos protocolos como http, udp y ftp. Esto permite el intercambio de información mediante redes, haciéndolo ideal para crear aplicaciones distribuidas.

**Arquitectura Neutral:** La ejecución de la máquina virtual de Java es como un programa en el sistema operativo y ejecuta directamente el bytecode de los programas ya compilados específicamente para esta, cualquier arquitectura que cuente con una máquina virtual de Java puede ejecutar este programa, sin importar en donde fue creado. Otros de los estándares que implementa Java para

asegurar su portabilidad son que los enteros siempre se manejan en 32 bits y las cadenas de caracteres utilizan Unicode en vez de ASCII.

**Robustez:** Java busca problemas en el código tanto durante la compilación como en tiempo de ejecución, esto ayuda a reducir la posibilidad de errores, maneja la memoria automáticamente, previniendo los accesos ilegales a memoria y verifica el bytecode resultado de la compilación antes de ejecutarlo en la máquina virtual.

Algunas de las verificaciones que se realizan en el código son:

- El código no produzca desbordamientos.
- Todos los parámetros son conocidos y correctos.
- No han ocurrido conversiones ilegales de datos.
- El acceso a los campos de otros objetos es legal.
- No hay intentos de violar las reglas de acceso y seguridad.

**Threading:** Java puede realizar varias actividades simultáneamente en un programa de forma nativa mediante el manejo de hilos (threads), estos son pequeños procesos generados por un proceso grande que realizan diversas tareas para este. Esta característica permite mejorar el rendimiento y el comportamiento del programa.

### 1.3 Java Platform, Micro Edition (Java ME)

Java ME fue presentado en 1999 con el propósito de crear aplicaciones Java para dispositivos pequeños; inicialmente implementando solamente una máquina virtual. Teniendo en cuenta las limitaciones asociadas a los dispositivos pequeños como memoria limitada, tamaño de pantalla y consumo energético, Sun definió las características de la tecnología Java Platform, Micro Edition (Java ME) para adecuarse a esos entornos tan limitados.

Java ME es la tecnología del futuro para la industria de los dispositivos móviles y, actualmente, prácticamente la totalidad de las compañías telefónicas están incluyendo los medios necesarios para implantarla en sus productos.

Java ME es una colección de tecnologías y especificaciones que pueden ser combinadas para construir un entorno de aplicación completo para adecuarse a los requerimientos de cualquier dispositivo móvil. Esto ofrece una gran

flexibilidad y compatibilidad que cooperan para brindar al usuario final la mejor experiencia posible.

No se pueden crear aplicaciones Java ME sin conocer de antemano la configuración y el perfil al que van destinados, o sin conocer qué paquetes opcionales están disponibles. En estos conceptos se encuentran las definiciones que determinan las características de Java que se pueden emplear, qué APIs usar y cómo se empaquetan, así como sobre cómo se realiza la distribución de las aplicaciones.

En general, las siguientes preguntas resumen los requerimientos para un desarrollo móvil en Java:

- ¿En qué tipo de hardware correrá la aplicación?
- ¿Qué funcionalidad se incluirá en la máquina virtual de Java?
- ¿Qué características del lenguaje Java se incluirán?
- ¿Qué APIs serán necesarias para el desarrollo de la aplicación móvil?

### **1.3.1 Configuraciones**

En resumen, una configuración, que provee las librerías y capacidades de la máquina virtual más básicas para utilizarse en una familia de dispositivos con capacidades similares.

Una configuración consiste en un entorno de ejecución Java completo que define el entorno de ejecución básico de Java ME. Su objetivo es adecuarse a las necesidades de una familia de dispositivos con capacidades similares.

Cualquier configuración está formada por tres elementos:

- Una máquina virtual Java para ejecutar el bytecode de la aplicación.
- Código nativo para realizar la interfaz entre Java y el sistema operativo del dispositivo.
- Un conjunto de clases Java que constituyen el entorno de ejecución.

Aunque una configuración proporcione un entorno Java completo, su conjunto base de clases es reducido y debe ser ampliado por medio de perfiles o bien por clases propias definidas por el desarrollador.

Las configuraciones no son pensadas como una solución completa, sino como un punto de partida, una base común para crear APIs adicionales. Actualmente están definidas dos configuraciones: Connected Device Configuration (CDC) y Connected Limited Device Configuration (CLDC).

- **Connected Limited Device Configuration (CLDC):** Esta Configuración para Dispositivos con Conexión Limitada (CLDC) fue creada para poder mantener las características de Java sobre dispositivos de capacidades de procesamiento y memoria muy limitadas como teléfonos celulares, pagers o PDAs.

Para este tipo de dispositivos se dispone de una máquina virtual Java específica denominada Kilo Virtual Machine (KVM), donde Kilo corresponde a Kilobyte. Como su nombre lo indica, CLDC no requiere muchos recursos de hardware, siendo algunos de estos:

- Entre 160 KB y 512 KB de memoria (incluida RAM y ROM).
- Conexión a red, intermitente, de baja velocidad, 9600 bps o menor.

La descripción del API CLDC tiene ciertas limitaciones a nivel de la máquina virtual, las más importantes son:

- No hay soporte para operaciones de punto flotante (No obstante, la revisión 1.1 de la especificación incluye cierto soporte para operaciones con punto flotante).
- Las constantes y tipos en punto flotante no están soportadas.
- Las clases y métodos que necesitaban argumentos Float o Double han sido eliminadas.
- No hay finalización de objetos. El método finalize() ha sido eliminado de la clase Object. El recolector se limita a

recuperar objetos no referenciados y a eliminarlos.

- Sólo se definen estos errores: Error, OutOfMemoryError, VirtualMachineError y NoClassDefFoundError. Cualquier error no asimilado por la implementación será tratado por la máquina virtual concluyendo la aplicación.
- No hay soporte para la serialización de objetos, así como grupos de tareas, threading, o daemons.
- No se permiten cargadores de clases específicos. La aplicación no puede influir en cómo se cargan las clases.

<b>java.io</b>	Contiene las clases necesarias para el manejo de entrada y salida de datos.
<b>java.lang</b>	Contiene las clases fundamentales para el lenguaje de programación Java.
<b>java.util</b>	Contiene clases para colección de datos, hora y fecha.
<b>javax.microedition.io</b>	Contiene el entorno de trabajo genérico para conexiones de E/S en entornos con recursos limitados.

**Figura 2: APIs de CLDC 1.0**

<b>java.io</b>	Contiene las clases necesarias para el manejo de entrada y salida de datos.
<b>java.lang</b>	Contiene las clases fundamentales para el lenguaje de programación Java.
<b>java.lang.ref</b>	Provee soporte para referencias de objetos.
<b>java.util</b>	Contiene clases para colección de datos, hora y fecha.
<b>javax.microedition.io</b>	Contiene el entorno de trabajo genérico para conexiones de E/S en entornos con recursos limitados.

**Figura 3: APIs de CLDC 1.1**

- **Connected Device Configuration (CDC):** La Configuración para Dispositivos Conectados (CDC), está orientada a dispositivos con cierta capacidad computacional y memoria, además deben poder

conectarse a la red, de modo generalmente inalámbrico. CDC usa una máquina virtual Java similar a la de J2SE pero con limitaciones gráficas y de memoria. Esta máquina virtual es llamada CVM (Compact Virtual Machine).

Dispositivos de este tipo son los Smartphones, los Set-Top (televisión digital interactiva), terminales digitales de video o fotografía, impresoras, reproductores de sonido MP3, etc. El API CDC es un subconjunto muy amplio de J2SE. Incluye todas las clases del API CLCD, a las que añade algunas características adicionales:

- Soporte para punto flotante.
- Soporte para multithreading (multihilos).
- Soporte para la manipulación de sistemas de archivos.
- Soporte para la serialización de objetos.
- Soporte para conexiones HTTP.
- Soporte para la mayoría de las colecciones del API de J2SE.
- Cargador de clases definido por el usuario.
- Soporte de red.
- Soporte para los paquetes de J2SE.

<b>java.io</b>	Contiene las clases necesarias para el manejo de entrada y salida de datos.
<b>java.lang</b>	Contiene las clases fundamentales para el lenguaje de programación Java.
<b>java.lang.ref</b>	Provee soporte para referencias de objetos con interacción limitada con el Garbage Collector.
<b>java.lang.reflect</b>	Clases e interfaces para obtener información sobre clases y objetos.
<b>java.math</b>	Clases para aritmética de enteros y punto flotante.
<b>java.net</b>	Clases para implementar aplicaciones

	de red.
<b>java.security</b>	Clases e interfaces para el manejo de seguridad.
<b>java.security.cert</b>	Provee las clases necesarias para el manejo de certificados.
<b>java.text</b>	Clases para el manejo de texto, fechas, números y mensajes.
<b>java.util.jar</b>	Clases para leer y escribir archivos JAR.
<b>java.util.zip</b>	Clases para leer y escribir archivos ZIP y GZIP.
<b>java.util</b>	Contiene clases para colección de datos, internacionalización, hora y fecha.
<b>javax.microedition.io</b>	Contiene el entorno de trabajo genérico para conexiones de E/S en entornos con recursos limitados.

**Figura 4: API de CDC 1.1.2**

### 1.3.2 Perfiles

Mientras que la configuración define las características de las APIs de una familia de dispositivos, los perfiles definen las características de las APIs de cada dispositivo. Por eso, a la hora de construir una aplicación se necesitan tanto las APIs del perfil del dispositivo como de la configuración. Un perfil dado dota de funcionalidad específica a una configuración.

- **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 una interfaz gráfica completa y con posibilidad de ejecución de applets. Incluye además las bibliotecas del Abstract Windows Toolkit (AWT); añadiendo así, una interfaz de usuario básica.

- **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 una interfaz un poco más desarrollada que aquellos que posean los dispositivos donde PP va dirigido.
- **PDA Profile (PDAP):** Diseñado para PDAs que tengan pantallas de mayor resolución y más memoria que los teléfonos móviles.
- **Game Profile (GP):** Ofrece una plataforma para escribir juegos en dispositivos CDC.

Existe también el perfil MIDP (Mobile Information Device Profile), construido sobre CLDC. El perfil MIDP fue el primer perfil definido para Java ME y está orientado a dispositivos de ciertas características:

- Reducida capacidad computacional y de memoria.
- Conectividad limitada.
- Capacidad gráfica reducida.
- Entrada alfanumérica de datos reducida.
- 128 Kb de memoria no volátil (ROM) para componentes MIDP.
- 8 Kb de memoria ROM para datos persistentes de aplicaciones.
- 32 Kb de memoria volátil (RAM) en tiempo de ejecución para la pila de la máquina virtual de Java.

Desde un punto de vista práctico MIDP es el único perfil disponible para trabajar efectivamente en la actualidad. Las aplicaciones realizadas usando el perfil MIDP son llamadas MIDlets.

<b>javax.microedition.lcdui</b>	Provee características para la implementación de interfaces de usuario para aplicaciones MIDP.
---------------------------------	--

<b>javax.microedition.rms</b>	Provee un mecanismo para que los MIDlets puedan almacenar y recolectar datos de forma persistente.
<b>javax.microedition.midlet</b>	Define las aplicaciones MIDP y la interacción entre la aplicación y el entorno en el que esta se ejecuta.
<b>javax.microedition.io</b>	Incluye coporte generic de conexiones basado en el CLDC.
<b>java.io</b>	Contiene las clases necesarias para el manejo de entrada y salida de datos.
<b>java.lang</b>	Contiene las clases fundamentales para el lenguaje de programación Java.
<b>java.util</b>	Contiene clases para colección de datos, hora y fecha.

**Figura 5: API de MIDP 1.0**

<b>javax.microedition.lcdui</b>	Provee características para la implementación de interfaces de usuario para aplicaciones MIDP.
<b>javax.microedition.lcdui.game</b>	Provee clases para el desarrollo de juegos para dispositivos móviles.
<b>javax.microedition.rms</b>	Provee un mecanismo para que los MIDlets puedan almacenar y recolectar datos de forma persistente.
<b>javax.microedition.midlet</b>	Define las aplicaciones MIDP y la interacción entre la aplicación y el entorno en el que esta se ejecuta.
<b>javax.microedition.io</b>	Incluye coporte generic de conexiones basado en el CLDC.
<b>javax.microedition.media</b>	Permite el manejo de medios en móviles.
<b>javax.microedition.media.control</b>	Define los tipos de control específicos para reproductores de medios.
<b>javax.microedition.pki</b>	Certificados para autenticar información para conexiones seguras.

<b>java.io</b>	Contiene las clases necesarias para el manejo de entrada y salida de datos.
<b>java.lang</b>	Contiene las clases fundamentales para el lenguaje de programación Java.
<b>java.util</b>	Contiene clases para colección de datos, hora y fecha.

**Figura 6: API de MIDP 2.0**

### 1.3.3 MIDlets

Las aplicaciones MIDP se denominan MIDlet, un MIDlet es un programa en lenguaje de programación Java para dispositivos embebidos (se dedican a una sola actividad), más específicamente para la máquina virtual de Java ME. Generalmente son aplicaciones y juegos que corren en un teléfono móvil y están desarrollados bajo el perfil MIDP.

El ciclo de desarrollo de un MIDlet es el siguiente:

- Crear y 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 los archivos de clases (.class) que se han generado en la compilación del programa. Además puede contener los recursos necesarios para el MIDlet como son 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 del MIDlet contenido en el archivo JAR.

Existen varias formas para transferir los archivos JAR y JAD al dispositivo móvil con soporte Java ME, esto dependiendo de la marca y modelo del dispositivo.

Si el dispositivo 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.

Otra posibilidad es poner los archivos JAR y JAD en un servidor WAP o HTTP y descargarlo directamente desde el móvil. Para ello es necesario que el dispositivo tenga un navegador WAP o web y soporte GPRS, EDGE o 3G para una descarga fiable.

Para que una aplicación MIDP trabaje en un dispositivo móvil, se requiere un dispositivo que implemente Java ME y MIDP para ser utilizada. Como otros programas desarrollados en Java, tienen la característica, "Escribir una vez, ejecutar en cualquier parte" ("Write once, run anywhere"). Para escribir los programas se puede obtener el Sun Java Wireless Toolkit o NetBeans con el Mobility Pack. Para las distribuciones 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 estar empacado dentro de un archivo .JAR.
- El archivo .JAR necesita ser pre-verificado usando un preverificador.
- En algunos casos, el archivo .JAR necesita ser firmado digitalmente por un proveedor de dispositivos móviles.

### 1.3.4 Paquetes opcionales

Cuando una serie de clases o interfaces no se ajustan a un solo dispositivo, sino que podemos incluirla en varios de ellos, creamos un paquete. Un claro ejemplo de esto es el soporte para la tecnología Bluetooth. Los paquetes opcionales no definen un entorno de ejecución completo sino un conjunto de APIs para características determinadas.

En resumen, un paquete es un conjunto de APIs específicas que extiende las capacidades de un entorno de aplicación Java ME.



Figura 7: Componentes de Java ME

## 1.4 Descripción de herramientas utilizadas

A continuación se describen brevemente cada una de las herramientas de software utilizadas en la creación de este proyecto.

### 1.4.1 NetBeans IDE

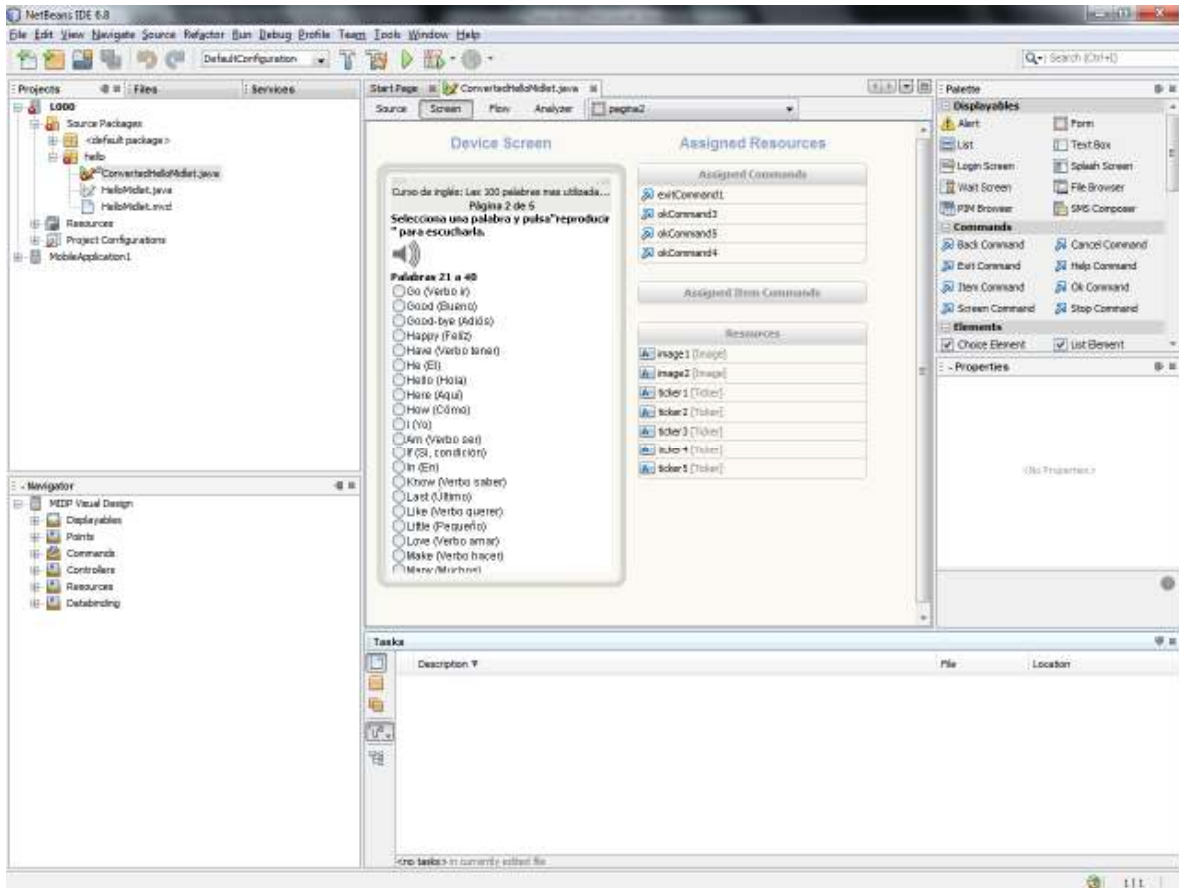
NetBeans es un entorno de desarrollo de aplicaciones (IDE) desarrollado en Java por Sun Microsystems, este brinda a los desarrolladores todas las herramientas necesarias para crear aplicaciones de escritorio, empresariales, web y móviles, NetBeans es fácil de instalar y

utilizar y corre en varios sistemas operativos, incluyendo Windows, Linux, Mac OS X y Solaris.

Para su funcionamiento, NetBeans necesita una máquina virtual de Java instalada en el sistema operativo, la cual puede ser una previamente instalada o la versión de Java 2 Standard Edition (J2SE) incluida con NetBeans.

Algunas características de NetBeans son las siguientes:

- Gratuito.
- Fácil construcción de aplicaciones tanto de escritorio como móviles combinado con un constructor de interfaces gráficas de usuario (GUI).
- Creación de aplicaciones web con Cascading Style Sheets (CSS), JavaScript y JavaServer Pages (JSP).
- Además de aplicaciones Java y web, puede utilizarse para programar aplicaciones en C, C++, PHP, Symfony, Groovy, Grails, Ruby, Ruby on Rails y Python.
- Documentación en línea.



**Figura 8: Interfaz para desarrollo en móviles de NetBeans**

### 1.4.2 Java Wireless Toolkit

El Sun Java Wireless Toolkit es un toolbox para probar aplicaciones para dispositivos móviles basadas en Java Micro Edition y utilizando el Connected Limited Device Configuration (CLDC) y Mobile Information Device Profile (MIDP).

Este toolkit incluye entornos de emulación de dispositivos, características de optimización, documentación y ejemplos para crear aplicaciones móviles eficientes.



**Figura 9: Emulador de dispositivo móvil del Wireless Toolkit**

### **1.4.3 Apache HTTP Server**

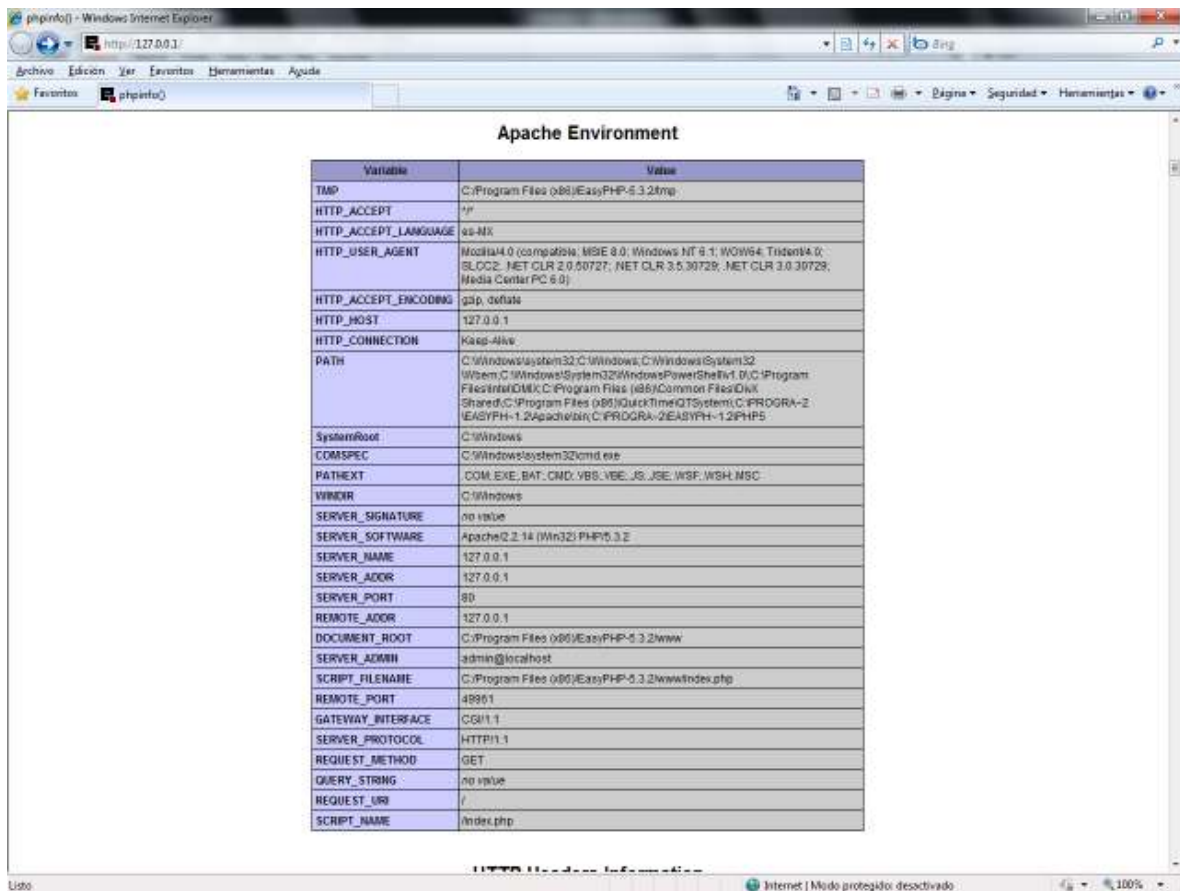
El Apache HTTP Server Project es un desarrollo de software libre con el propósito de crear una implementación robusta y gratuita de un servidor HTTP.

Apache se usa principalmente como servidor para contenido estático y dinámico de páginas web pero su función no se limita a eso, ya que por ejemplo, se puede configurar para compartir archivos desde un servidor o una computadora personal hacia Internet. Como todo servidor HTTP, Apache puede ser configurado y personalizado en gran medida mediante archivos de configuración, genera logs de sus actividades para análisis de errores y detección de fallas y soporta encriptación de hasta 2048 bits para

sus comunicaciones mediante los protocolos de Secure Socket Layer (SSL) y Transport Layer Security (TLS).

Apache HTTP Server es una descarga gratuita que también se distribuye en muchas distribuciones de Linux, así como también integrado en muchos paquetes de software propietario como Oracle, IBM WebSphere Application Server y Mac OS X.

Para este proyecto, se utilizó Apache de manera local principalmente para realizar pruebas y para simular un servidor en Internet en el cual se encuentra hosteada la parte de la página web del proyecto y desde la cual se puede descargar el contenido móvil.



Variable	Value
TMP	C:\Program Files (x86)\EasyPHP-5.3.2\tmp
HTTP_ACCEPT	*/*
HTTP_ACCEPT_LANGUAGE	es-MX
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_HOST	127.0.0.1
HTTP_CONNECTION	Keep-Alive
PATH	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Program Files\Intel\DMIC;C:\Program Files (x86)\Common-Files\Disk Shared;C:\Program Files (x86)\QuickTime\QTSystem\;C:\PROGRA~2\EASYPH~1.2\apache\bin;C:\PROGRA~2\EASYPH~1.2\PHP5
SystemRoot	C:\Windows
COMSPEC	C:\Windows\system32\cmd.exe
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
WINDOW	C:\Windows
SERVER_SIGNATURE	no value
SERVER_SOFTWARE	Apache/2.2.14 (Win32) PHP/5.3.2
SERVER_NAME	127.0.0.1
SERVER_ADDR	127.0.0.1
SERVER_PORT	80
REMOTE_ADDR	127.0.0.1
DOCUMENT_ROOT	C:\Program Files (x86)\EasyPHP-5.3.2\www
SERVER_ADMIN	admin@localhost
SCRIPT_FILENAME	C:\Program Files (x86)\EasyPHP-5.3.2\www\index.php
REMOTE_PORT	48961
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	no value
REQUEST_URI	/
SCRIPT_NAME	/index.php

Figura 10: Página de la configuración de Apache (generada mediante PHP)

## 1.4.4 Adobe Dreamweaver

Dreamweaver es el editor elegido para la creación del componente web del proyecto. Dreamweaver incluye, además de un editor tradicional de HTML un editor “What You See Is What You Get” (WYSIWYG) y está disponible para Windows y Mac OS X, soporta tecnologías para la creación de páginas web tales como Cascading Style Sheets (CSS), JavaScript y Hypertext Preprocessor (PHP) entre otras, facilitando el proceso de la creación de una página web.

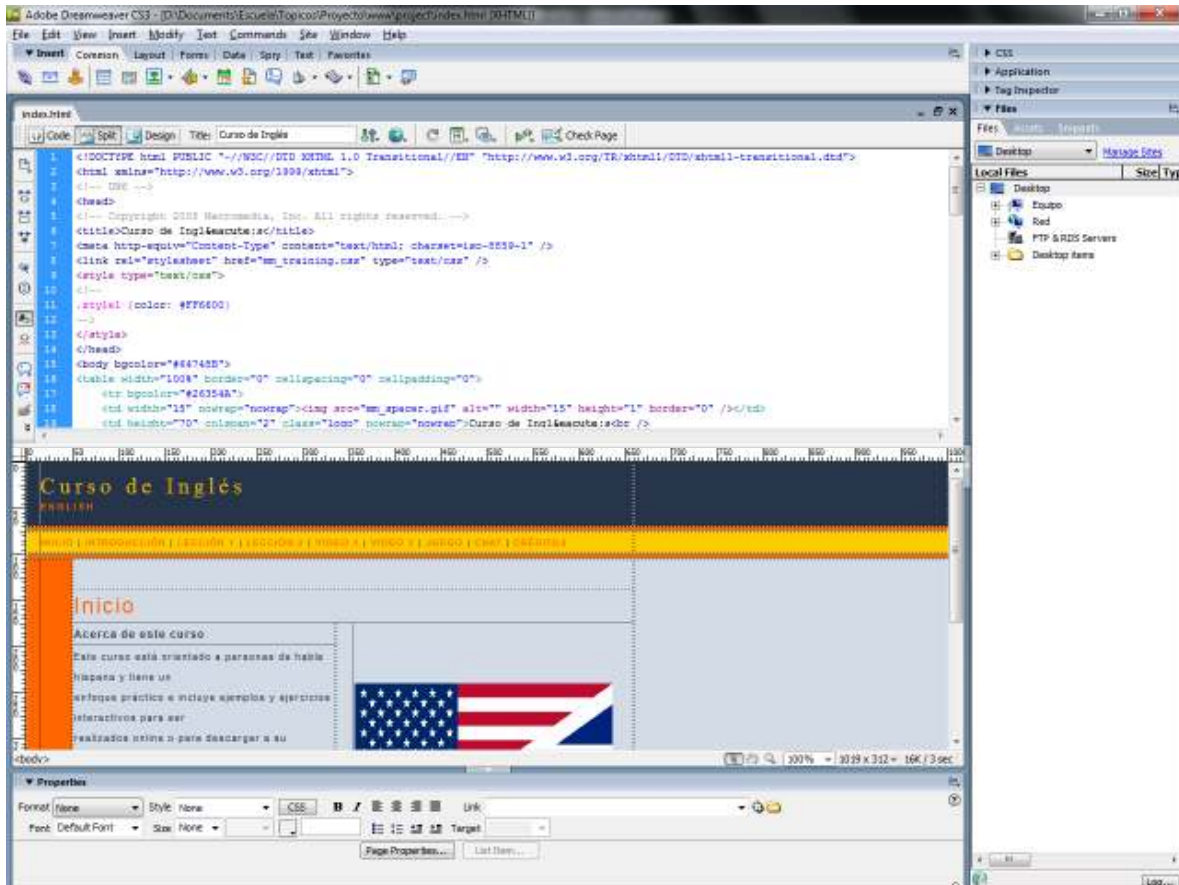


Figura 11: Interfaz de Dreamweaver

## 1.4.5 EasyPHP

EasyPHP es un paquete AMP (Apache MySQL PHP), este incluye el Apache HTTP Server, el manejador de bases de datos MySQL y el lenguaje PHP y permite montar fácilmente un servidor HTTP (Apache) en una máquina sin mayor configuración, haciéndolo muy útil como un ambiente de

pruebas para desarrollos web. Para utilizarlo, simplemente debe activarse y este realizará toda la configuración necesaria para ejecutar Apache.

Para visualizar los nuevos desarrollos web que se hagan solo se deben copiar los archivos al directorio “www” dentro del directorio de instalación de EasyPHP y abrir un browser en la dirección del localhost.



Figura 12: Página en el directorio WWW de EasyPHP, nótese el URL

## Capítulo II: Análisis de Requerimientos

En este capítulo se realiza el análisis de los requerimientos del sistema. Se comienza el análisis a partir de la necesidad planteada para posteriormente continuar con los diagramas de caso de uso.

### 2.1 Aprendizaje para la vida (Life Long Learning)

Una de las áreas que mayor crecimiento y transformación han sufrido en los últimos años es el aprendizaje. Incluso, actualmente ha proliferado un sinnúmero de propuestas que se basan en las recientes tecnologías de la información.

Desde la publicación del informe de la Unión Europea en Lisboa, en el que se publicaron las seis prioridades mundiales en materia de formación, se han creado muchos proyectos relacionados con el sector educativo, con el fin de hacer una sociedad más competitiva, dinámica y basada en el conocimiento.

No hay duda de que la idea de “un aprendizaje para toda la vida” está ya anticuada. Por ello, la sociedad, los empresarios y educadores deben ser más flexibles para avanzar al mismo ritmo que los desarrollos tecnológicos y la globalización, el envejecimiento de la población y las nuevas prácticas. Las instituciones educativas pueden contribuir de forma significativa al proceso de aprendizaje continuo. Debemos anticiparnos a los sistemas escolares actuales, ya que sus cambios no son lo suficientemente dinámicos como para ajustarse a las exigencias de los mercados laborales y sus rápidas evoluciones, o incluso para superar las expectativas de los “clientes”. Queremos construir el acceso al aprendizaje continuo de forma que resulte atractivo tanto para los alumnos como para la sociedad en general.

Nuestra ambición consiste en ofrecer a la sociedad una plataforma educativa que se adapte a sus intereses tanto escolares como particulares y profesionales. Teniendo en cuenta que sin duda, nos estamos moviendo hacia una sociedad global, también es fundamental sentirse cómodos al trabajar con personas de otras culturas, así como adaptarse a las diferencias culturales. Como hemos mencionado, la construcción del acceso al aprendizaje debe verse desde distintos puntos de vista. Aún así, los objetivos principales deberían ser siempre la

promoción de propuestas orientadas al aprendizaje y su integración dentro del método de formación de los alumnos.

Sin embargo, el acceso a oportunidades de formación no es suficiente; las personas también necesitan apoyo y orientación para desarrollar sus propios planes de formación y para participar en la sociedad. El acceso al aprendizaje continuo puede ser una forma de crear el camino hacia la integración social. Debemos subrayar la calidad y efectividad de la formación proporcionada y su disponibilidad a la diversidad de estudiantes.

## **2.2 eLearning**

Resulta evidente que el eLearning ha supuesto una extraordinaria revolución en la manera de concebir la educación a distancia. Actualmente, sin embargo, empiezan a proliferar dudas respecto a esta consideración inicial y comúnmente aceptada. Una identificación entre formación a distancia y eLearning produce, en realidad una infravaloración de esta última modalidad formativa, precisamente porque la identificación entre ambas realidades limita muchas de las posibilidades nuevas que aporta la formación en red en su aplicación y que en realidad se aproximan más a la formación presencial, incluso superando muchas de las barreras inherentes a esta última (capacidad de interacción, construcción social del conocimiento, etc.).

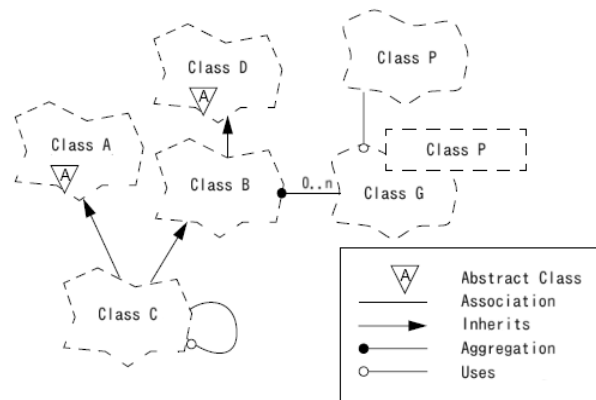
Actualmente, los esquemas de comunicación evolucionan de un contexto “uno-a-uno” de la formación a distancia y el de “uno-a-muchos” de los contextos presenciales a una modalidad en red, “muchos-a-muchos”, propia del eLearning; por otra parte, el alumno es el centro del aprendizaje en la educación a distancia, pues él es el único responsable de asimilar en solitario los contenidos que se le proporcionan, mientras que en la formación presencial se deposita la mayor parte de la responsabilidad en la labor del docente.

En el eLearning se da una situación intermedia en la que el alumno posee un papel fundamental, no individual sino colectivo, pues forma parte de una comunidad, pero a la vez existe una figura docente que es clave y funciona como catalizador de todo el proceso de enseñanza-aprendizaje. Se trata de la figura del tutor online.

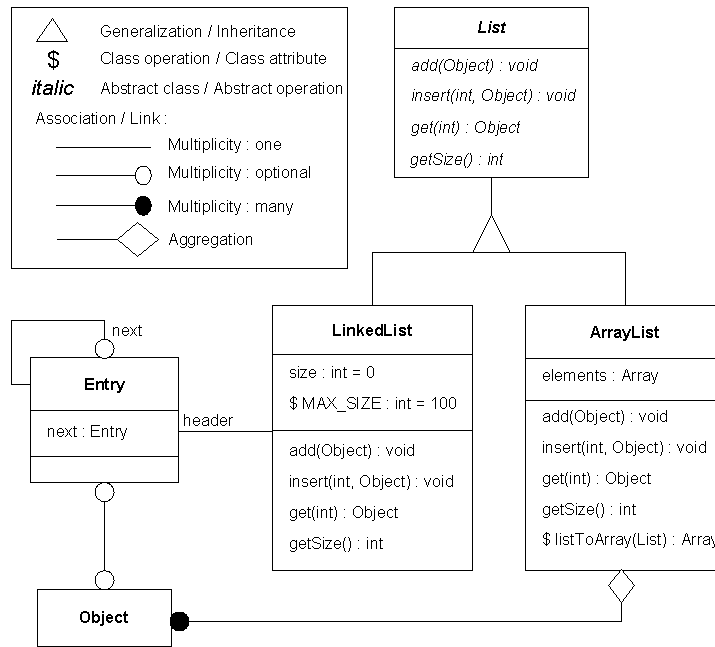
La formación continua es, sin duda alguna, uno de los ámbitos de aplicación que genera mayor demanda y experimenta mayor crecimiento en relación con el eLearning, tanto en contextos presenciales, no presenciales o mixtos.

## 2.3 Modelado de aplicaciones con UML

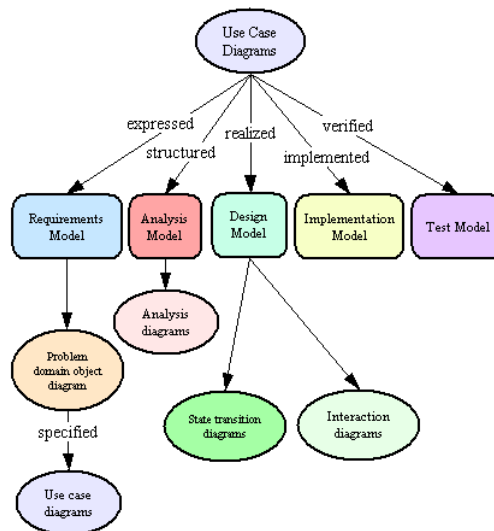
Uno de los lenguajes que en los últimos años ha tenido mayor aceptación para el modelado de aplicaciones y que ahora se ha convertido en un estándar de facto es el Unified Modeling Language (UML), esto se debe a que este novedoso lenguaje no solo permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos como su arquitectura, estructura y comportamiento, sino que también permite el modelado de sus procesos de negocios y estructura de datos. UML combina técnicas de modelado de datos, modelado de negocios, modelado de objetos y modelado de componentes e incorpora y sintetiza las ventajas de los modelos de Booch, Object-Modelling Technique (OMT) y Object-Oriented Software Engineering (OOSE).



**Figura 13: Diagrama de Booch (Grady Booch)**



**Figura 14: Diagrama OMT (James Rumbaugh et. al.)**



**Figura 15: Diagrama OOSE (Ivar Jacobson)**

Para el desarrollo de UML se estableció que debía existir un balance entre la expresividad y la simplicidad. Si fuese demasiado simple, podría impedir abordar determinadas clases de problemas. Si resultase muy complejo, sería mortal para los diseñadores. En el caso de unificar metodologías existentes, había que considerar la sensibilidad respecto de las documentaciones existentes. Si se producían muchos cambios, se confundirían a los usuarios actuales; ello

provocaría resistencia para adoptarlo, y se perdería la oportunidad de difundir el nuevo lenguaje y hacerlo de uso masivo.

Uno de los objetivos principales de la creación de UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE (Computer-Aided Software Engineering) orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común y sencilla para que fuera aceptada por la gran mayoría.

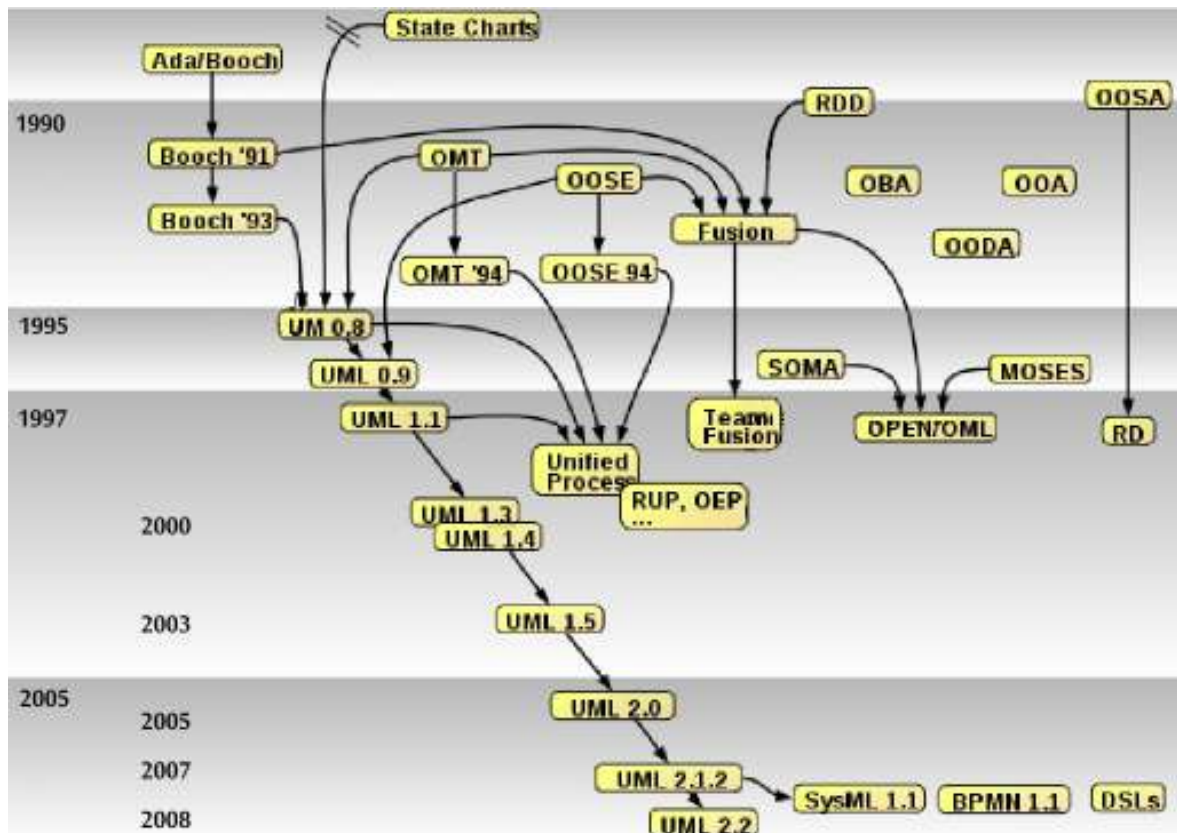
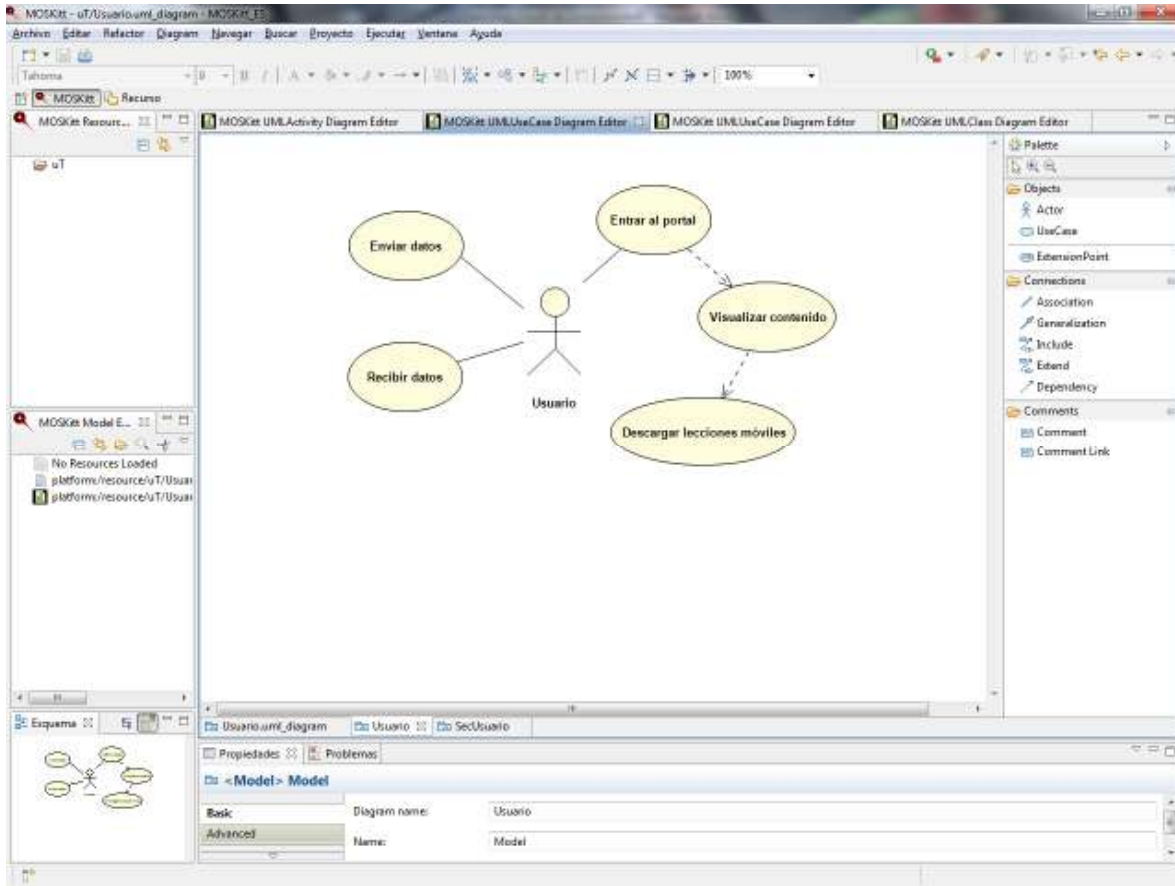


Figura 16: Evolución de UML

## 2.4 Diagramas de casos de uso

En UML, un diagrama de casos de uso (Use Case Diagram) es un tipo de diagrama de comportamiento cuyo propósito es presentar una representación gráfica de la funcionalidad provista por un sistema desde el punto de vista de los actores, sus metas (representadas como casos de uso), sus interacciones y las dependencias entre esos casos de uso. El propósito principal de un diagrama de

casos de uso es el de mostrar que funciones se realizarán por cada actor (que tomará un cierto rol) dentro de un sistema y como se relacionará con su entorno.



**Figura 17: Ejemplo de herramienta UML para casos de uso**

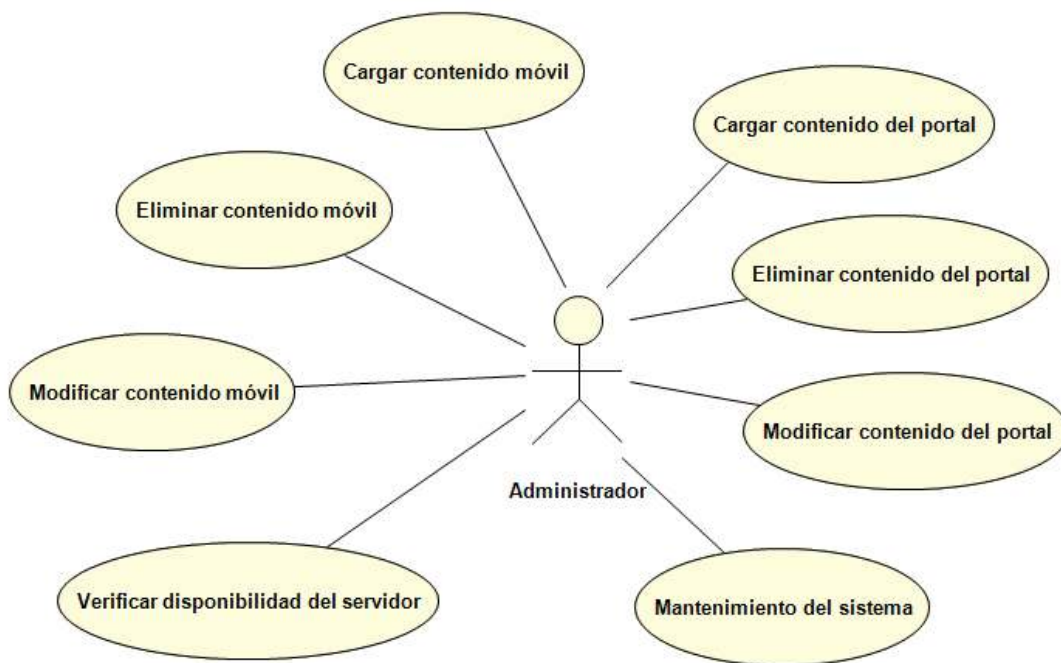
En UML, un actor especifica un rol de un usuario o cualquier otro sistema que interactúa con un caso de uso, se tienen los siguientes actores:

### **2.4.1 Administrador**

El administrador maneja el sistema y es responsable de su disponibilidad.

- Carga nuevo contenido para móviles en el portal.

- Elimina contenido obsoleto para móviles en el portal.
- Modifica las propiedades del contenido móvil en el portal.
- Carga nuevo contenido en el portal.
- Elimina contenido obsoleto en el portal.
- Modifica las propiedades del contenido en el portal.
- Verifica que el sistema se encuentre disponible.
- Da mantenimiento al sistema para mantener la disponibilidad.



**Figura 18: Diagrama de casos de uso del Administrador**

### Cargar contenido móvil

Caso de uso:	Cargar contenido móvil
Actores:	Administrador
Propósito:	Añadir nuevo contenido para dispositivos móviles.

Resumen:	El administrador carga nuevo contenido para dispositivos móviles, en forma de un archivo JAR que podrá ser descargado y ejecutado por estos.	
Acción del actor	1. El usuario administrador dispone de nuevo contenido móvil, por lo que lo añade a la carpeta correspondiente en el servidor.	Respuesta del sistema 1. El sistema ahora deberá mostrar el nuevo contenido.

#### Eliminar contenido móvil

Caso de uso:	Cargar contenido móvil	
Actores:	Administrador	
Propósito:	Elimina contenido para dispositivos móviles.	
Resumen:	El administrador desea eliminar contenido para dispositivos móviles.	
Acción del actor	1. El usuario administrador elimina contenido móvil, de la carpeta correspondiente en el servidor.	Respuesta del sistema 1. El sistema ya no deberá mostrar el nuevo contenido.

#### Modificar contenido móvil

Caso de uso:	Modificar contenido móvil	
Actores:	Administrador	
Propósito:	Edita el contenido para dispositivos móviles.	
Resumen:	El administrador desea cambiar el contenido ya establecido para	

	dispositivos móviles.	
Acción del actor		Respuesta del sistema
1. El usuario administrador sobre escribe con una nueva versión el contenido móvil, en la carpeta correspondiente en el servidor.		1. El sistema deberá mostrar el contenido actualizado en vez del anterior.

### Cargar contenido del portal

Caso de uso:	Cargar contenido del portal	
Actores:	Administrador	
Propósito:	Añadir nuevo contenido para el portal.	
Resumen:	El administrador carga nuevo contenido para el portal, en forma de nuevas páginas de lecciones, secciones, etc.	
Acción del actor		Respuesta del sistema
1. El usuario administrador dispone de nuevo contenido para el portal, por lo que lo añade a la carpeta correspondiente en el servidor y ajustando las demás páginas (en caso de requerirse) para mostrarlo.		1. El sistema ahora deberá mostrar el nuevo contenido del portal.

### Eliminar contenido del portal

Caso de uso:	Eliminar contenido del portal	
Actores:	Administrador	

Propósito:	Eliminar contenido del portal.	
Resumen:	El administrador desea eliminar contenido del portal, en forma de páginas de lecciones, secciones, etc.	
Acción del actor	<ol style="list-style-type: none"> <li>1. El usuario administrador desea eliminar contenido del portal, por lo que elimina los archivos necesarios de la carpeta correspondiente en el servidor y ajustando las demás páginas (en caso de requerirse) para ya no mostrarlo.</li> </ol>	Respuesta del sistema <ol style="list-style-type: none"> <li>1. El sistema ya no deberá mostrar el contenido eliminado del portal.</li> </ol>

#### Modificar contenido del portal

Caso de uso:	Modificar contenido del portal	
Actores:	Administrador	
Propósito:	Edita el contenido del portal.	
Resumen:	El administrador desea cambiar el contenido ya establecido del portal.	
Acción del actor	<ol style="list-style-type: none"> <li>1. El usuario administrador sobre escribe con una nueva versión el contenido del portal o hace ligeros cambios en el contenido ya establecido, en la carpeta correspondiente en el servidor.</li> </ol>	Respuesta del sistema <ol style="list-style-type: none"> <li>1. El sistema deberá mostrar el contenido actualizado en vez del anterior.</li> </ol>

#### Verificar disponibilidad del servidor

Caso de	Verificar disponibilidad del servidor
---------	---------------------------------------

uso:	
Actores:	Administrador
Propósito:	Verificar que el servidor se encuentre disponible.
Resumen:	El administrador debe verificar que el servidor se encuentra en buen estado y no presenta fallas que impidan su funcionamiento correcto.
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> <li>1. El administrador verifica el estado del servidor y la conexión de red.</li> <li>2. El administrador verifica que el servidor HTTP y los servicios requeridos se encuentren disponibles.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema deberá funcionar normalmente.</li> </ol>

#### Mantenimiento del sistema

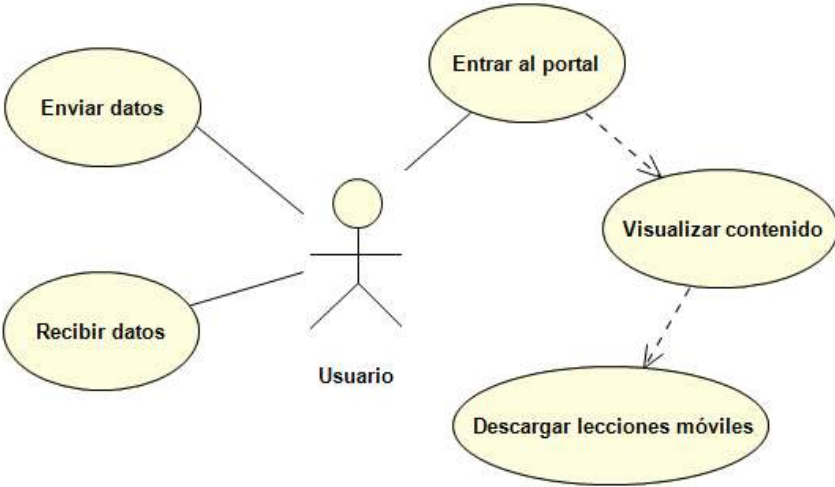
Caso de uso:	Mantenimiento del sistema
Actores:	Administrador
Propósito:	Reparar errores presentados en el servidor.
Resumen:	El administrador debe solucionar los errores presentados en el servidor tanto de manera proactiva como reactiva para su correcto funcionamiento.
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> <li>1. El administrador verifica y soluciona los problemas a nivel de hardware.</li> <li>2. El administrador verifica y soluciona los problemas a nivel</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema deberá funcionar normalmente.</li> </ol>

de software.	
--------------	--

**2.4.2 Usuario**

Es el estudiante que entra al sistema para utilizar el contenido del portal o descargar la parte móvil del sistema.

- Descarga lecciones para su dispositivo móvil del portal.
- Envía datos desde la aplicación móvil para su evaluación por el agente.
- Recibe datos enviados desde el agente hacia la aplicación móvil.
- Entra al portal.
- Dentro del portal puede visualizar el contenido del este.



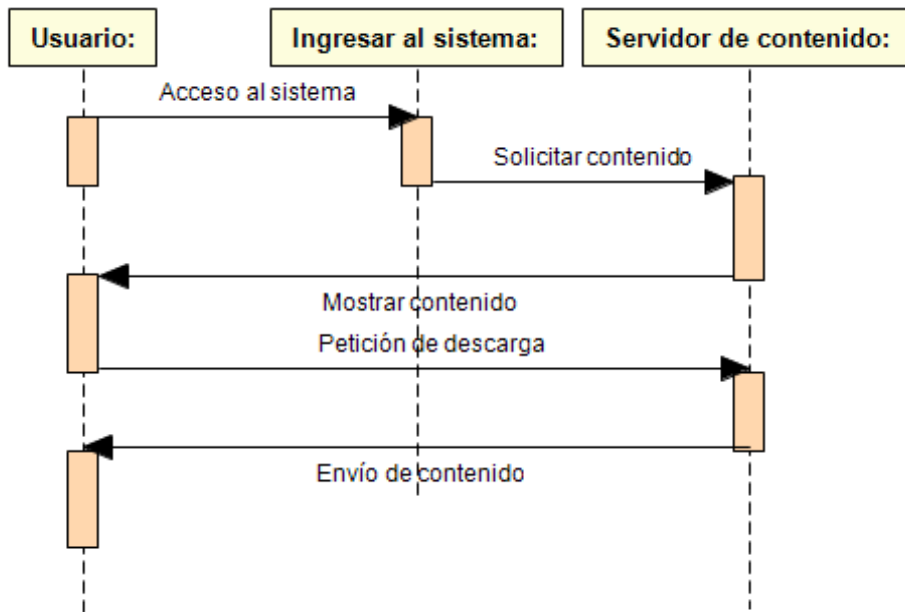
**Figura 19: Diagrama de casos de uso del Usuario**

Descargar lecciones móviles

Caso de uso:	Descargar lecciones móviles
Actores:	Usuario
Propósito:	Obtener contenido para dispositivo móvil desde el portal.

Resumen:	El usuario descarga un archivo JAR que podrá ejecutar en su dispositivo móvil, este archivo incluye el contenido del portal y podrá ser utilizado en cualquier momento.	
Acción del actor		Respuesta del sistema
<ol style="list-style-type: none"> <li>1. El usuario entra al sistema mediante el portal.</li> <li>2. El usuario navega a la sección de su preferencia para buscar el contenido móvil de esta.</li> <li>3. El usuario trata de descargar el contenido para móvil de su preferencia.</li> </ol>		<ol style="list-style-type: none"> <li>1. El sistema muestra el contenido del portal al usuario.</li> <li>2. El sistema muestra el contenido de acuerdo a lo que solicita el usuario.</li> <li>3. El sistema busca el contenido JAR correspondiente y lo transfiere al dispositivo móvil del usuario.</li> </ol>

**sd: Descargar lecciones móviles**



**Figura 20: Descarga de contenido**

Enviar datos

Caso de	Enviar datos
---------	--------------

uso:	
Actores:	Usuario, Agente
Propósito:	El usuario manda los datos de sus respuestas al agente.
Resumen:	El usuario manda los datos de sus respuestas de ejercicios o exámenes desde la aplicación móvil al servidor para que el agente los interprete.
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> <li>1. El usuario envía sus respuestas mediante un MMS o una conexión wireless disponible al servidor para que el agente interprete los datos.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema recibe los datos y los pasa al agente, este los interpretará para obtener los resultados del usuario.</li> </ol>

#### Recibir datos

Caso de uso:	Recibir datos
Actores:	Usuario, Agente
Propósito:	El agente manda los datos interpretados de las respuestas al usuario.
Resumen:	Después de que el usuario manda los datos de sus respuestas de ejercicios o exámenes desde la aplicación móvil al servidor para que el agente los interprete, este los envía de vuelta.
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> <li>1. El usuario debe haber enviado sus datos al agente previamente.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema envía los datos que el agente le ha pasado mediante un correo electrónico que llegará al usuario con los resultados que obtuvo.</li> </ol>

sd: Enviar datos

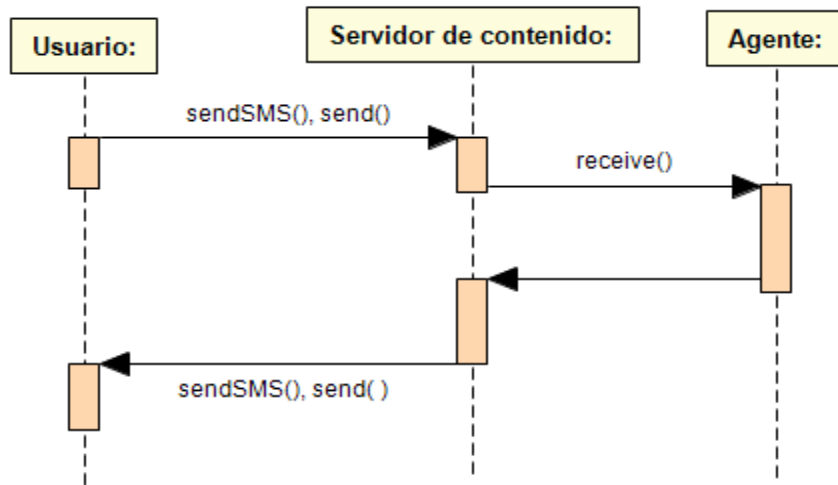


Figura 21: Envío y recepción de datos

### 2.4.3 Agente

El agente obtiene los resultados de los ejercicios de los usuarios, los califica y los devuelve.

- Recibe datos desde la aplicación móvil para procesarlos.
- Procesa los datos que recibió.
- Envía los datos procesados a la aplicación móvil.

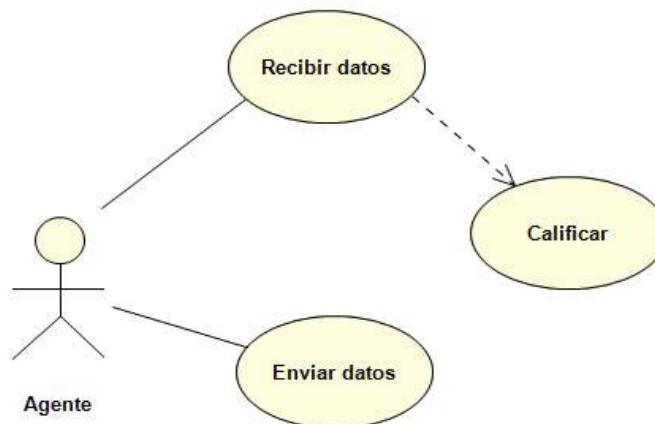
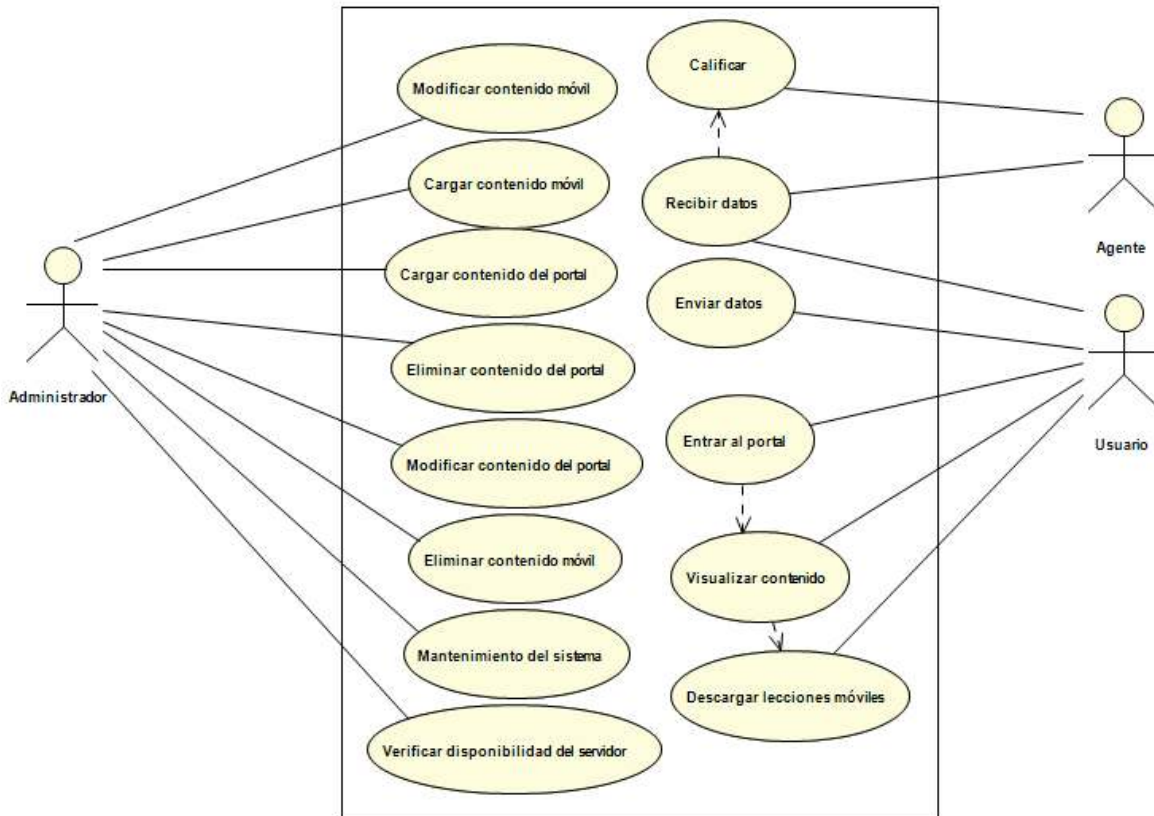


Figura 22: Diagrama de casos de uso del Agente



**Figura 23: Diagrama de casos de uso del sistema**

## Capítulo III: Diseño del sistema

En este capítulo se analiza el porqué de las especificaciones concretas que deberá llevar el sistema y sus necesidades para ser implementadas posteriormente.

### 3.1 Enseñanza de un segundo idioma

En términos de lingüística, el lenguaje es un sistema complejo de comunicación que se analiza en varios niveles. Tales pueden ser: sintaxis, morfología, semántica entre otros. Debido a esto, en ocasiones el proceso de aprendizaje puede resultar complicado para el estudiante cuando comienza a enfrentarse con esos elementos que pueden convertirse en dificultades a lo largo del camino. El aprendizaje de un segundo idioma se define como el proceso de aprender un lenguaje a cualquier nivel cuando ya se ha adquirido alguno anteriormente. Esto es, un segundo idioma es cualquier otro lenguaje aparte de la lengua nativa del aprendiz.

La computación aplicada en el campo educacional y particularmente en la enseñanza de un idioma parece ser prometedora. En los últimos tiempos ha mostrado una gran ventaja sobre la instrucción tradicional. Esta situación se expresa considerando que los programas computacionales han arrojado resultados más efectivos en ésta área. Lo anterior en comparación con aquella instrucción basada simplemente en la utilización de los métodos en los que se estudian y aplican las reglas gramaticales. Es así como surge el término de Aprendizaje de un Idioma Asistido por Computadora (CALL: Computer Assisted Language Learning). En este contexto varios tipos de herramientas computacionales han sido desarrolladas, todas ellas con el objetivo de ayudar a los estudiantes a aprender diferentes idiomas. Por lo que ahora es muy común ver cómo la tecnología se emplea en la educación.

Cuando el diseño del software se basa en algún modelo educativo, se obtiene como resultado programas más completos, y estos pueden llegar a satisfacer las necesidades de un estudiante en particular, ya que pueden adaptarse a su ritmo de aprendizaje.

### **3.2 Aprender en cualquier lugar y en cualquier momento**

Los usuarios entrarían a una página web o aplicación móvil con el mismo contenido, donde al principio en una fase introductoria, aprenderían a mejorar sus capacidades básicas de pronunciación, lectura y redacción. También, considerando la opinión de Tony Buzan, en su libro “Using your Memory”, sugiere que hay tres métodos importantes para el aprendizaje de lenguajes extranjeros que pueden ser muy efectivos:

- Las palabras más importantes que aprender: Señala que deben existir al menos 100 palabras que componen el 50% de todas las frases usadas en una conversación en un lenguaje. El hecho de aprender a utilizar esas 100 palabras ahorra al estudiante un largo camino en su meta de aprender a hablar ese lenguaje, aunque sea en un nivel básico.
- Uso de mnemónicos para ligar palabras: Aquí se utilizan imágenes para ligar palabras de un lenguaje extranjero al propio lenguaje del estudiante, por ejemplo, al estudiar inglés/español se puede tener:
  - Español: Alfombra / Inglés: Carpet

El estudiante puede imaginar una alfombra oriental con algún diseño tejido en el centro.

- Mnemónico del lenguaje cotidiano: Esto depende del hecho de que el vocabulario básico de un idioma se relaciona a cosas comunes: cosas que se pueden encontrar en la ciudad o pueblo. Para usar esta técnica, se debe escoger un lugar con el que se esté familiarizado. Se deben utilizar objetos en ese lugar como claves para recordar las imágenes que se ligan a esas palabras extranjeras.

La mayoría de los métodos se sintetizan en estos tres, pero no servirían de nada si no se tiene un componente audible, un componente audible es absolutamente esencial al aprender un nuevo idioma.

### **3.3 El aprendizaje necesita acción**

Se requieren acciones para aprender inglés. El pronunciar las palabras correctamente es muy importante para aprender inglés. El hecho de que estas acciones se puedan realizar utilizando un portal o dispositivo móvil es un valor

agregado. Se pueden saber todos los tips de aprendizaje, pero si no se comienza a practicar y a hacer cosas, no se conseguirá nada. El hecho es que si se desea aprender inglés bien, el estudiante debe cambiar su vida. Algunas actividades que se pueden realizar son:

- Leer libros en inglés, el leer un libro por una hora y analizar la estructura de las palabras y buscarlas en un diccionario de inglés.
- Escuchar audio libros o grabaciones en inglés, detenerlos frecuentemente y tratar de entender lo que se está diciendo y tratar de imitar la pronunciación de quien está hablando.
- Escribir e-mails en inglés, cuidando que cada palabra esté correctamente escrita.

Con estas acciones en mente, se debe implementar lecciones que cubran estos aspectos, si se desea aprender se debe disfrutar realizando estas actividades, ya que una persona que odie una actividad jamás podrá tener éxito en esta. El mayor problema con el estudio del idioma inglés es que los estudiantes quieren aprender a hablar inglés bien, pero no todos quieren dedicarle el tiempo necesario a estudiar por su cuenta. Esta falta de motivación significa que los estudiantes básicamente no ocupan su tiempo en aprender inglés, o si lo hacen, no lo hacen regularmente. Por ejemplo, un estudiante promedio podría estudiar inglés por 12 horas antes de un examen, pero no leerá un libro de inglés por 30 minutos cada día. Simplemente no siente que el estudio del inglés sea tan satisfactorio y solo lo hace porque debe hacerlo. El problema es que un solo gran esfuerzo no lo deja con nada, mientras que actividades pequeñas diarias tienen un gran beneficio. Se debe trabajar en la motivación, las siguientes actividades pueden ayudar en esto:

- Motivación: volverse una persona a la que le gusta aprender otros idiomas.
- Conseguir un buen diccionario.
- Tratar de evitar errores. Se debe tratar de usar de forma correcta el idioma desde el principio.
- Pronunciación: Aprender a pronunciar el idioma.

- Leer y escuchar muchas oraciones en ese idioma, como leer o ver películas en ese idioma.

El idioma es una forma de comunicación primariamente hablada. Aprendemos nuestro idioma desde niños oyendo e imitándolo. Los métodos más eficaces para aprender idiomas siempre están enfocados más al aspecto auditivo. No se puede ignorar la importancia de la lectura, pero claramente el aspecto fundamental de la comunicación en un idioma es hablar y escuchar.

Los videos son otra parte importante que suplementa las acciones que deben realizarse. A medida que escuchamos conversaciones y oraciones en ese idioma, más rápido avanzaremos en nuestro aprendizaje. Los videos, películas y música estimulan el sentido del oído, esto permite imitar la pronunciación de forma más clara y correcta. Es así que el sistema deberá tender a mejorar el conocimiento del usuario del lenguaje inglés mediante la práctica con ejercicios que deberá resolver. Si el usuario tiene poco o nulo conocimiento del idioma, deberá obtener las bases para ser capaz de empezar a hablarlo y entenderlo. En particular, se agregará la reproducción de videos en dispositivos móviles con el objetivo de ofrecerle al estudiante la oportunidad de practicar en cualquier lugar y en cualquier momento tomando ventaja de la tecnología móvil.

### 3.4 El entorno de aprendizaje ubicuo

A finales de la década de 1980, el investigador de Xerox PARC (Palo Alto Research Center), Mark Weiser, acuñó el término “cómputo ubicuo”. Este se refiere al proceso de integrar computadoras al mundo físico. El cómputo ubicuo incluye tecnología computacional encontrada en procesadores, dispositivos móviles, cámaras digitales y otros, que en conjunto proveen una nueva dimensión al aprendizaje. Las teorías de aprendizaje son importantes en el desarrollo de la tecnología educacional porque ayudan a crear una relación entre la información, el estudiante y el entorno, permitiendo a los estudiantes obtener nuevos conocimientos desde lo que ven, escuchan, leen y perciben.

Las características principales del aprendizaje ubicuo son las siguientes:

- **Permanencia:** Los estudiantes nunca pierden su trabajo a menos que sea borrado a propósito. Adicionalmente, los procesos de aprendizaje se guardan continuamente cada día.
- **Accesibilidad:** Los estudiantes tienen acceso a sus documentos, datos o videos desde cualquier lugar. Esta información es provista basada en sus peticiones. Así, el aprendizaje involucrado es auto dirigido.

- **Inmediación:** Donde quiera que se encuentren los estudiantes, deben poder tener cualquier información inmediatamente. Así, los estudiantes pueden resolver problemas rápidamente. De otra forma, el estudiante podría grabar las preguntas y buscar las respuestas después.
- **Interactividad:** Los estudiantes pueden interactuar con expertos, maestros o compañeros en forma de comunicación síncrona o asíncrona. Así, los expertos pueden ser consultados más fácilmente y el conocimiento está más disponible.
- **Situar las actividades de instrucción:** El aprendizaje podría ser embebido en nuestra vida diaria. Tanto los problemas encontrados como el conocimiento requerido se presentan en su naturaleza y forma auténticas. Esto ayuda a los estudiantes a darse cuenta de las características de las situaciones problemáticas que hacen acciones particulares algo relevante.
- **Adaptabilidad:** Los estudiantes pueden obtener la información necesaria en la forma adecuada.

Más aún, el aprendizaje ubicuo puede ser un tipo de aprendizaje colaborativo soportado por computadora, de esta forma se propone este sistema de enseñanza móvil, explotando para esto la tecnología móvil de los teléfonos celulares.

Para diseñar un entorno de aprendizaje ubicuo es necesario permitir que cada estudiante interactúe con varios objetos embebidos, esta relación es común en esta nueva era de cómputo ubicuo. En un salón de clases ubicuo, los estudiantes se moverán dentro del espacio ubicuo e interactuarán con varios dispositivos. Aquí, cada estudiante tendrá un dispositivo móvil, PDA o teléfono celular, u-Teacher se encontrará disponible en todo momento y en cualquiera de las tecnologías móviles (PDA, teléfono celular o Laptop).

u-Teacher permite a los estudiantes interactuar de dos formas: offline y online. Online mediante la web y offline mediante el teléfono celular o PDA sin alguna conexión a Internet. Cuando el estudiante trabaja online, lo hace utilizando Internet para comunicarse con el servidor; por otro lado, cuando trabaja offline, se comunica mediante mensajes. Además, el estudiante puede descargar lecciones, tareas y exámenes a su dispositivo móvil para estudiar offline en cualquier

momento y en cualquier lugar. Esta característica permite a los estudiantes interactuar mas entre ellos y con sus maestros, así, los maestros no solo enseñan a sus alumnos los tópicos de cada día, si no que los alientan aprender mediante el uso de las nuevas tecnologías, el aprendizaje puede convertirse en una experiencia emocionante al buscar el conocimiento en vez de buscar solo una calificación, esto porque el sistema se basa en una de las tecnologías más aceptadas en el mundo. Los estudiantes pueden planear su tiempo, elegir los días, y completar sus tareas de modo que desarrollan un alto nivel de auto disciplina, responsabilidad y madurez. Más aún, el entorno de aprendizaje ubicuo es una situación de educación omnipresente, educación que sucede alrededor del estudiante pero el estudiante puede no estar consciente del proceso de aprendizaje.

El uso de tecnología inalámbrica y móvil permite un acceso sencillo y contribuye a la funcionalidad educativa, ya que esta debería ser ofrecida en cualquier lugar a cualquier hora y de cualquier forma.

Además de lo descrito anteriormente, existen cinco propiedades de los dispositivos móviles que producen oportunidades educativas únicas:

- **Portabilidad:** Se puede llevar uno de estos dispositivos a cualquier lugar.
- **Interacción social:** Se puede intercambiar datos y colaborar con otras personas frente a frente.
- **Sensibilidad al contexto:** Se puede obtener datos únicos de la localización actual, entorno y tiempo, incluidos datos reales y simulados.
- **Conectividad:** Se puede conectar dispositivos móviles a dispositivos de adquisición de datos, otros móviles, o a una red que crea un verdadero entorno compartido.
- **Individualidad:** Se puede proveer una plataforma de investigación personalizada para al gusto de cada persona.

### 3.5 El modelo u-Teacher

Teniendo como base los razonamientos anteriores, podemos explicar las necesidades del sistema con la siguiente definición:

Se necesita desarrollar un sistema de software capaz de ser utilizado por medio de red y de dispositivos móviles. Para ello se desarrollará un portal web y además el programa para dispositivos móviles correspondiente, esta plataforma permitirá el aprendizaje de un idioma extranjero, en este caso el inglés y debe incluir el mismo contenido tanto en la versión del portal como la versión móvil.

El software permitirá a un usuario explorar las diferentes secciones contenidas en el curso para aprender de forma básica el lenguaje inglés, y también podrá descargar desde el portal la versión móvil del curso, la cual deberá tener el mismo contenido del portal y podrá ser consultada en cualquier lugar y en cualquier momento sin necesidad de conectarse al servidor. Dentro de la aplicación móvil, el usuario iniciará esta y después del saludo de entrada, podrá acceder a la lección, ejercicio o examen correspondiente, los resultados escritos por el usuario serán mandados al servidor y evaluados por un agente.

Las tareas diseñadas para este sistema son:

- Descargar lecciones para ayudar a los estudiantes a aprender los fundamentos básicos del idioma inglés online o en sus dispositivos móviles (offline).
- Sintaxis básica de oraciones (texto y voz).
- Ejercicios de pronunciación donde los estudiantes pueden escuchar la pronunciación correcta.
- Videos donde los estudiantes vean y escuchen diálogos breves.
- Formatos que validen lo que se entendió en los videos.
- Prácticas colaborativas donde a lo mas cuatro estudiantes puedan interactuar con preguntas y respuestas.
- Textos y diálogos y sus correspondientes actividades de validación.
- Juegos que ayuden a reafirmar lo aprendido.
- Chat de texto y voz entre diferentes dispositivos que permita el aprendizaje del lenguaje escrito.

En conjunto, las herramientas descritas para la implementación de u-Teacher podrán dar los siguientes beneficios:

- Proporciona el conocimiento básico para el idioma inglés: Las personas que no cuentan con algún conocimiento del idioma pueden utilizar esta herramienta. Esto con el fin de tener un panorama básico de la pronunciación de palabras y la escritura de éstas. Todo esto sin la necesidad de tener a un maestro al lado, lo que le permite tener un avance de acuerdo a sus necesidades y posibilidades de aprendizaje.
- Conveniencia para los usuarios al tener un curso completo en formato digital portátil y reducción de la dependencia de materiales físicos.
- Acceso al curso de forma flexible, independiente de la localización, minimizando así la dependencia de un laboratorio o biblioteca y su horario de atención.
- Se obtiene una herramienta fácil de usar para aprendizaje en la cual se puede estudiar y realizar tareas o exámenes en cualquier lugar y momento.
- Mayor involucramiento por parte del usuario e interés por el material, prácticas y discusiones y una mayor concentración y disciplina.
- Mayor soporte para las necesidades y preferencias de estudio individual.

## Capítulo IV: Implementación del sistema

En este capítulo se explica la arquitectura del proyecto y su implementación a nivel web y a nivel aplicación, mostrando las pantallas finales para la interacción con el usuario.

### 4.1 Arquitectura del sistema

La arquitectura general del sistema consiste básicamente de los siguientes elementos:

- Servidor de aplicaciones
- Servidor web
- Dispositivo móvil
- Servicio de comunicaciones de dispositivos móviles
- Firewall para evitar intrusiones
- Agente para evaluar los resultados de los estudiantes

Primero, el servidor de aplicaciones contiene la aplicación y permite que esta sea descargada hacia el dispositivo móvil, de igual forma, mantiene el servicio de la aplicación para los estudiantes.

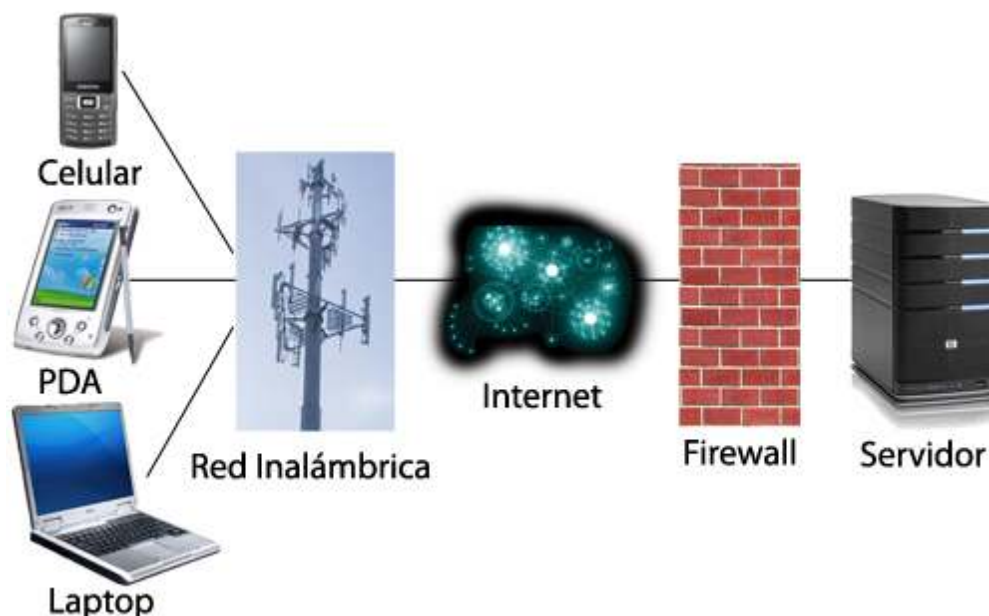
Segundo, dentro del mismo servidor de aplicaciones se encuentra el servidor web, este es el responsable de la operación de la parte web de la aplicación, permitiendo subidas y descargas de información.

Tercero, el dispositivo móvil se comunica mediante mensajería multimedia o directamente con conectividad web, de esta forma permite descargar las lecciones y ejercicios, resolverlos fuera de línea y volver a mandarlos al servidor mediante un mensaje o conectándose a la web.

Cuarto, el servicio de comunicaciones para dispositivos móviles permite la comunicación entre los dispositivos y el servidor.

Quinto, El firewall detiene a los intrusos, programas dañinos o cualquier otro peligro y permite la operación segura del servidor.

Finalmente, la aplicación tiene un agente que tiene dos tareas: evaluar las respuestas de los alumnos y supervisar la versión del programa, en caso de que se encuentren nuevas versiones disponibles.



**Figura 24: Arquitectura general del sistema**

A continuación se presenta u-Teacher como una aplicación para teléfonos celulares para la enseñanza del idioma inglés. El dominio del lenguaje inglés para un profesional que trata de ingresar en una compañía con un mínimo de vocación exportadora o que es miembro de un grupo más amplio de personas con culturas diversas no solo es un requisito indispensable para las pruebas de selección de personal, sino también como un instrumento de comunicación y negocios. Por estas razones, los estudiantes del idioma necesitan un portal de Internet que les ofrezca un conjunto de lecciones para ayudarlos a aprender los conceptos básicos del lenguaje inglés en línea, y lo que es más importante, que puedan acceder a estos conocimientos en cualquier lugar y en cualquier momento mediante sus dispositivos móviles (como teléfonos celulares) con los siguientes objetivos específicos (online y offline respectivamente):

- Descarga de un conjunto de lecciones para ayudar a los estudiantes a aprender los conceptos básicos del lenguaje inglés en línea o desde sus dispositivos móviles.

- Sintaxis básica de oraciones con texto y voz.
- Ejercicios de pronunciación donde los estudiantes puedan escuchar e imitar la pronunciación correcta.
- Videos donde el usuario pueda ver y escuchar diálogos breves con el fin de acostumbrarse a las conversaciones en inglés.
- Formatos que validen el nivel de entendimiento de los videos.
- Establecer diálogos y actividades para los estudiantes.
- Juegos que ayuden al aprendizaje.



**Figura 25: Interfaz web**

En la pantalla inicial se da una descripción del curso y se puede acceder a las diferentes lecciones de este, cada lección tiene su contraparte móvil.

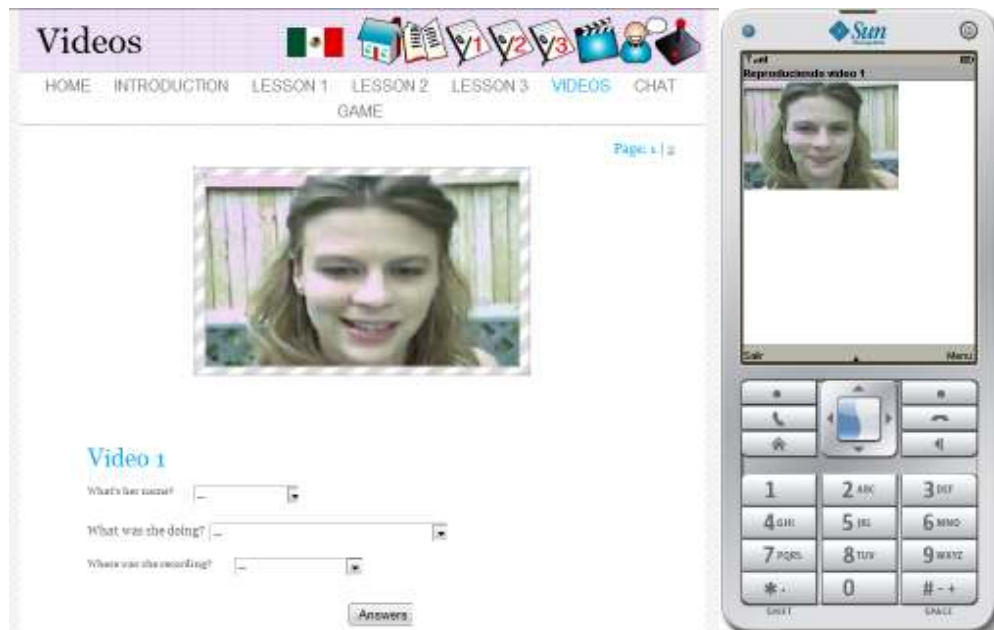


Figura 26: Pantalla inicial

Dentro de cada lección se encuentra un vínculo para descargar una versión móvil de esta para el dispositivo móvil, que podrá ser transferida a este. Las lecciones son idénticas e incluyen el contenido tanto de texto como imágenes, sonidos y videos.



Figura 27: Interfaz web e interfaz móvil mostrando el mismo contenido



**Figura 28: Prácticas de videos**



**Figura 29: Envío de datos**

Con respecto a la evaluación, se implementa un agente cuya misión es la de validar que las respuestas sean correctas. Aquí, el estudiante contesta las preguntas offline (en cualquier lugar y momento) y se mandan por medio de un

archivo de texto para su evaluación. El resultado de la evaluación se devuelve de la misma manera.

En cuanto al chat, por su naturaleza se encuentra disponible solo para la parte web, ya que para esto se requiere encontrarse continuamente online.



Figura 30: Chat

## 4.2 Desarrollo de aplicaciones móviles en NetBeans

Existen muchos métodos para trabajar en aplicaciones móviles Java ME, algunas muy simples, como la instalación de Toolkits Java (como el Wireless Toolkit). Sin embargo, si se recurre solamente a ese tipo de instalaciones, se está bastante limitado para desarrollos posteriores y además no se obtendrá una visión global del lenguaje Java. Por ello, en el presente trabajo de instalamos herramientas profesionales, algunas bastante complejas, pero tocando solamente los aspectos de Java ME para facilitar su uso.

NetBeans es un Open Source Integrated Development Environment, escrito en Java, usado como plataforma para desarrollar programas en Java y otros lenguajes de programación. Es decir, con NetBeans no solo podremos desarrollar para J2ME, sino también programas complejos para diferentes entornos y

sistemas operativos (Windows, Linux, Mac OS X, etc.) y aplicaciones cliente-servidor.

#### 4.2.1 Mobility Pack de NetBeans

El Mobility Pack de NetBeans expande las capacidades estándar de NetBeans para poder desarrollar aplicaciones para dispositivos móviles con Java ME, con características de optimización para el Mobile Information Device Profile (MIDP), y para el Connected Limited Device Configuration (CLDC). Además, integra en NetBeans las siguientes características para Java ME:

- **Visual Mobile Designer (Diseñador Visual Móvil):** Permite desarrollar GUIs (Graphical User Interface) de manera rápida y precisa mediante el arrastre desde una paleta de componentes directamente a la interfaz de la aplicación; la paleta incluye componentes comunes para las pantallas de la aplicación móvil y permite crear diferentes pantallas interconectadas a las cuales se pueden agregar también gráficos y sonidos de manera personalizada.

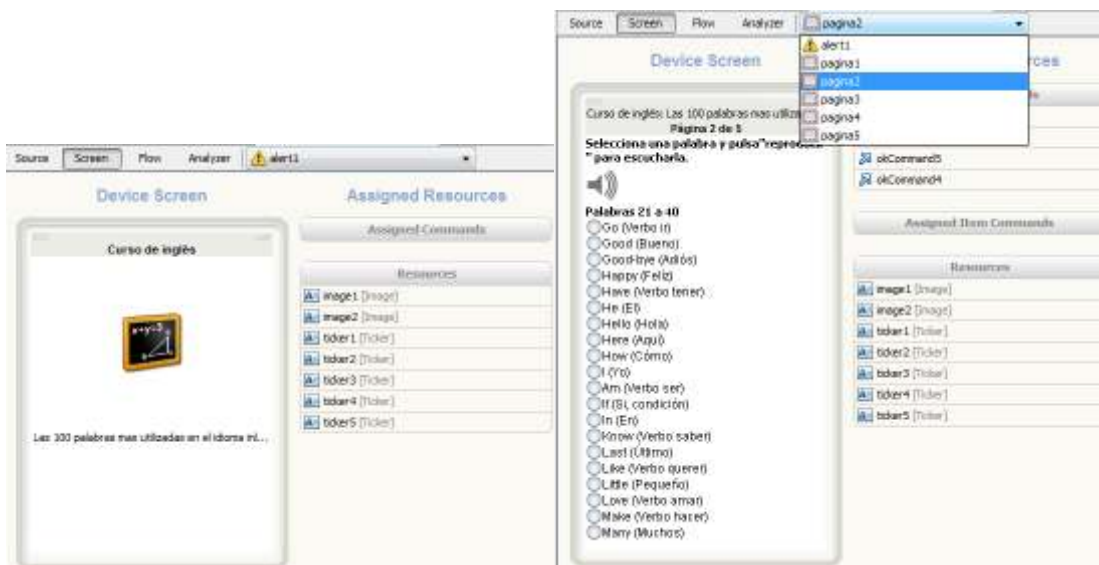
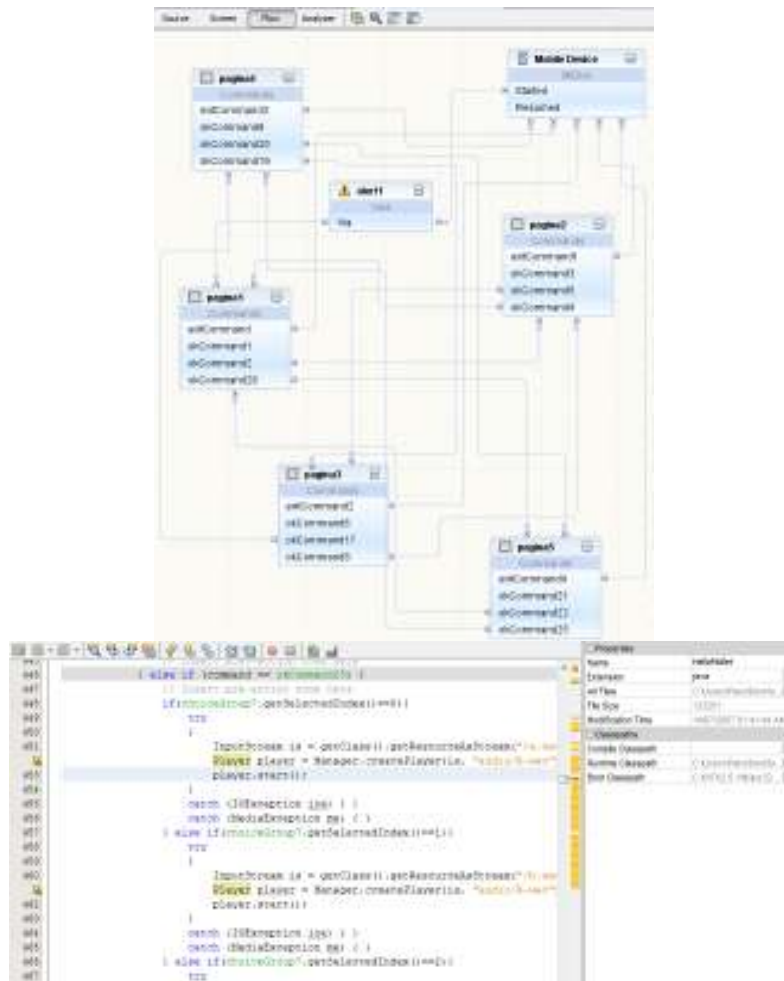


Figura 31: Pantallas de la aplicación en Visual Mobile Designer



**Figura 32: Flujo de pantallas y editor de código**

- **Fragmentación de dispositivos:** Se incluyen muchos tipos de dispositivos y perfiles, de modo que se puedan desarrollar aplicaciones para ejecutarlas en un dispositivo con un hardware específico.
- **Integración con el Java ME Wireless Toolkit:** La herramienta de certificación y desarrollo Java ME, que integra emulación OTA (Over-The-Air), de comunicaciones inalámbricas, APIs multimedia, y emulación para mensajes SMS. Todo esto permite una emulación de dispositivos de forma más realista.

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

### 4.3 Clases

A continuación se muestran las clases Java más relevantes para la creación de la aplicación móvil en NetBeans.

#### **send**

Envía un mensaje por red.

```
public synchronized void run()
{
    while (true)
    {
        //Si no existe con quien comunicarse espera
        if (message == null)
        {
            try
            {
                wait();
            } catch (InterruptedException e) {}
        }
        if (message == null)
        {
            break;
        }
        try
        {
            //Escribe el mensaje en un outputStream, forma básica de E/S
            os.write(message.getBytes());
            os.write("\r\n".getBytes());
            //En caso de que falle entrada o salida, se muestra el error de excepción E/S
        } catch (IOException ioe) {ioe.printStackTrace();}
        //Se completó, se sigue a la espera para mandar borrando el mensaje
        message = null;
    }
}
```

#### **sendSMS**

Envía un mensaje SMS.

```
public boolean sendSms(String number, String message)
{
    boolean result = true;
    try
    {
        //Número para mandar el mensaje
    }
}
```

```

        String addr = "sms://" + number;
        //Crear conexión, MessageConnection es la funcionalidad básica para enviar y
//recibir mensajes.
        MessageConnection conn = (MessageConnection) Connector.open(addr);
        //Prepara mensaje
        TextMessage
msg=(TextMessage)conn.newMessage(MessageConnection.TEXT_MESSAGE);
        //Introduce el mensaje en la variable msg
        msg.setPayloadText(message);
        //Envía mensaje
        conn.send(msg);
        conn.close();
    } catch (SecurityException se)
    {
        //Excepción en caso de error al mandar el mensaje
        result = false;
    } catch (Exception e)
    {
        result = false;
    }
    return result;
}

```

## receive

Recepción de mensajes para el agente.

```

public void run()
{
    try
    {
        //Se crea la conexión
        sc = (SocketConnection)Connector.open("socket://localhost:5000");
        si.setText("Conectado al servidor");
        //Streams de entrada y salida
        is = sc.openInputStream();
        //Loop infinito para recibir los datos
        while (true)
        {
            //Buffer para almacenar el mensaje
            StringBuffer sb = new StringBuffer();
            int c = 0;
            while (((c = is.read()) != '\n') && (c != -1))
            {
                sb.append((char)c);
            }
            if (c == -1)
            {break;}
        }
    }
}

```

```

        //En caso de recibir, notificar que recibió un mensaje
        si.setText("Mensaje recibido - " + sb.toString());
    }
    stop();
    si.setText("Conexión cerrada");
    f.removeCommand(sendCommand);
} catch (ConnectionNotFoundException cnfe) {
    //En caso de no tener conexión muestra error
    Alert a = new Alert("No se encuentra conexión", null, AlertType.ERROR);
    a.setTimeout(Alert.FOREVER);
    a.setCommandListener(this);
    display.setCurrent(a);
} catch (IOException ioe)
{
    if (!stop)
    {
        //Muestra error de la excepción de E/S
        ioe.printStackTrace();
    }
} catch (Exception e)
{
    //Muestra error de la excepción
    e.printStackTrace();
}
}

```

## **commandAction**

Se utiliza para darle acciones al menú de la aplicación.

```

//Si se escoge cierta opción
if(choiceGroup1.getSelectedIndex()==0)
{
    try
    {
        //Trata de obtener el recurso de sonido mediante un InputStream
        InputStream is = getClass().getResourceAsStream("/aan.wav");
        //Reproduce el recurso mediante Player
        Player p = Manager.createPlayer(is, "audio/X-wav");
        p.start();
    }
    //Lanza excepción en caso de no poder reproducir
    catch (IOException ioe) { }
    catch (MediaException me) { }
}

```

```

//Si se escoge cierta opción
if (command == okCommand2)
{
    //Cambia a la pantalla designada
    switchDisplayable(null, getPagina2());
}

```

```

if (command == exitCommand1)
{
    //Salir del MIDlet
    exitMIDlet();
}

```

### **get\_image**

Se utiliza para obtener una imagen de los recursos.

```

public Image get_image1()
{
    if (image1 == null)
    {
        try
        {
            image1 = Image.createImage("/1.PNG");
        } catch (java.io.IOException exception)
        {
            exception.printStackTrace();
        }
    }
    return image1;
}

```

### **get\_pagina**

Crea una pantalla en una nueva forma.

```

public Form get_pagina1()
{
    if (pagina1 == null)
    {
        //Título de la pantalla
        pagina1 = new Form("P\u00E1gina 1", new Item[] {get_stringItem1()});
        //Comandos que se incluirán en la pantalla
        pagina1.addCommand(get_exitCommand1());
        pagina1.addCommand(get_okCommand3());
        pagina1.addCommand(get_okCommand12());
        pagina1.setCommandListener(this);
    }
}

```

```

        pagina1.setTicker(get_ticker2());
    }
    return pagina1;
}

```

## **play**

Reproducción de video en un dispositivo móvil.

```

void play(String url)
{
    try
    {
        VideoControl vc;
        defplayer();
        //Crea un manejador de medios con la dirección de la localización de un medio
        player = Manager.createPlayer(url);
        player.addPlayerListener(this);
        //Checa que ha obtenido la información para cargar el medio
        player.realize();
        //El medio es un video
        vc = (VideoControl)player.getControl("VideoControl");
        if(vc != null)
        {
            //Modo de despliegue por default
            Item video = (Item)vc.initDisplayMode(vc.USE_GUI_PRIMITIVE, null);
            Form v = new Form("Reproduciendo video 1");
            //Checa que ha obtenido la información para cargar el medio
            v.append(video);
            //Muestra el video
            display.setCurrent(v);
        }
        player.prefetch();
        player.start();
    }
    catch(Throwable t)
    {
        reset();
    }
}

```

## **exitMIDlet**

Se utiliza para salir de la aplicación.

```

public void exitMIDlet()
{
    getDisplay().setCurrent(null);
    destroyApp(true);
}

```

```
    notifyDestroyed();  
}
```

## Conclusiones y trabajo a futuro

Se ha presentado a lo largo de esta tesis una serie de nuevas tecnologías de la información y comunicación que han revolucionado la funcionalidad de los dispositivos móviles, de manera particular, nos hemos dado a la tarea de conjuntar estas tecnologías y utilizarlas para realizar una aplicación desarrollada especialmente para dispositivos móviles. Esta aplicación es un primer paso para el desarrollo y uso auténtico de aplicaciones móviles para la enseñanza. Existe todavía muchísimo trabajo que hacer para la mejora de esta aplicación y versiones futuras; esto debido a las limitaciones de las tecnologías móviles en la actualidad.

Esta versión de u-Teacher es todavía muy básica, se espera crear una nueva versión a la que se le pueda incorporar más servicios, incluyendo mejor comunicación y acceso a aplicaciones web como complemento y mejoras en la interfaz para hacerla más atractiva.

De acuerdo a lo presentado en esta tesis, podemos obtener los siguientes puntos de análisis y mejora para futuras versiones de u-Teacher:

Se utilizaron varias tecnologías, principalmente Java ME como la base para esta aplicación debido a sus bondades como portabilidad y facilidad de uso. Java ME sigue evolucionando, y u-Teacher debe seguir evolucionando para hacer uso de las nuevas características que se presenten para Java ME.

La interfaz web pasó por tres fases diferentes, tratando de realizar una lo más atractiva e intuitiva posible para el usuario final, esta interfaz deberá seguir adaptándose a las nuevas tecnologías para desarrollo web que vayan creándose y eventualmente, conforme avancen las capacidades para dispositivos móviles, fusionar el portal web con la aplicación móvil, obteniendo una sola aplicación tanto para dispositivos móviles como para computadoras personales.

De manera particular, se ha observado que las tendencias de aprendizaje a futuro podrían tener un fuerte énfasis en los juegos, siendo estos los que logran captar la atención del usuario de manera más fuerte, se deberá poner especial atención en esto; por ejemplo, otorgando medallas o alguna especie de premio virtual al mejor desempeño en los ejercicios y exámenes, al retener la atención del usuario de esta forma, se garantiza un mejor aprendizaje por parte del usuario realizando una actividad agradable.

## Bibliografía

1. Chen, Y.S., Kao, T.C., Sheu, J.P., Chiang, C.Y.: A Mobile Scaffolding-Aid-Based Bird -Watching Learning System. In: Proceedings of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002), pp. 15-22. IEEE Computer Society Press, Los Alamitos (2002)
2. Curtis, M., Luchini, K., Bobrowsky, W., Quintana, C., Soloway, E.: Handheld Use in K-12: A Descriptive Account. In: Proceedings of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002), pp. 23–30. IEEE Computer Society Press, Los Alamitos (2002)
3. Fischer, G.: User Modeling in Human-Computer Interaction. Journal of User Modeling and User-Adapted Interaction (UMUAI) 11(1/2), 65–86 (2001)
4. Jacobs, M.: Situated Cognition: Learning and Knowledge Relates to Situated Cognition (1999) [verified 31 Oct 2004  
<http://www.gsu.edu/~mstsw/courses/it7000/papers/situated.htm>
5. Jones, V., Jo, J.H.: Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology. In: Atkinson, R., McBeath, C., Jonas-Dwyer, D., Phillips, R. (eds.) Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference, Perth, December 5-8, pp. 468–474 (2004)  
<http://www.ascilite.org.au/conferences/perth04/procs/jones.html>
6. Klopfer, Squire, Holland, Jenkins: Environmental Detectives-the development of an augmented reality platform for environmental simulations, Educational Technology Research and Development. Springer, Heidelberg (2002).
7. Weiser, M.: The computer for the twenty-first century, pp. 94–104. Scientific American (September 1991)
8. M Kontio, Professional Mobile Java, IT Press 2002.
9. Location Intelligence Magazine. Cell phone usage and Multi-Tasking.  
<http://www.directionsmidia.net/newsletters.archive/index.php?ID=712>

- 10.java.com: Java y Tú  
<http://www.java.com/es/>
- 11.The Java Tutorials  
<http://java.sun.com/docs/books/tutorial/index.html>
- 12.Java™ 2 Platform, Standard Edition, v 1.4.2 API Specification  
<http://java.sun.com/j2se/1.4.2/docs/api/>
- 13.Java ME Mobile Information Device Profile (MIDP); JSR 118  
<http://java.sun.com/products/midp/>
- 14.NetBeans Documentation, Training & Support  
<http://netbeans.org/kb/index.html>
- 15.Sun Java Wireless Toolkit  
<http://java.sun.com/products/sjwtoolkit/>
- 16.The Apache HTTP Software Project  
<http://httpd.apache.org/>
- 17.EasyPHP  
<http://www.easyphp.org/>
- 18.Object Management Group – Unified Modeling Language  
<http://www.uml.org/>