



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ARAGÓN

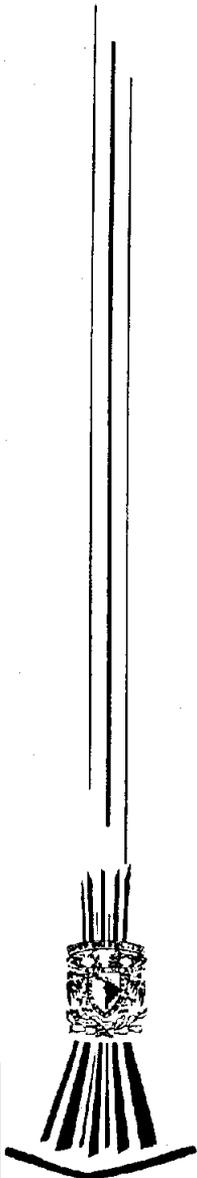
DESARROLLO DE APLICACIONES MÓVILES UTILIZANDO JAVA 2 ME COMO LENGUAJE DE PROGRAMACIÓN EN TELÉFONOS CELULARES Y/O POCKET PC.

PROYECTO DE TESIS
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A N :
ANGUIANO VARGAS JOSÉ LUIS
ROCHA GARFÍAS OSCAR IGNACIO

ASESOR:
ING. ALEJANDRO GONZÁLEZ PONCE

MÉXICO D.F.

2005





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIAS.

A Dios.

Por permitir que llegara este día y porque siempre esta conmigo.

A mis Padres.

Porque gracias a ellos soy lo que soy y a los que amo con todo mi ser, para ellos es este logro y espero lo disfruten tanto como yo pues es dedicado a ustedes. Los amo, respeto y admiro mucho.

A mis hermanas.

Rocío por ser una excelente madre, esposa y hermana. Eres un ser humano hermoso. Te quiero mucho.

Rosario por tu ejemplo de inteligencia, capacidad, independencia y ganas de hacer las cosas bien. Te quiero mucho.

A Alejandra.

Por estar conmigo en las buenas y en las malas, por tu comprensión, cariño, paciencia, apoyo, y tu amor incondicional. Te Amo. También a tu familia por brindarme su cariño, su confianza y su apoyo, gracias.

A Alejandro.

Por ser mi sobrino consentido. Te quiero mucho hijo y espero esto te sirva de ejemplo para que tu también te superes como ser humano.

A mi amigo Oscar.

Por ser mi mejor amigo y estar conmigo en esta aventura. Gracias por tus consejos, apoyo y la amistad que siempre me has brindado.

A todos mis amigos, maestros y a la FES Aragón.

Porque todos colaboraron de manera importante en este triunfo.

DEDICATORIAS.

A Dios.

Por permitir que llegara este día y porque siempre esta conmigo.

A mis Padres.

Porque gracias a ellos soy lo que soy y a los que amo con todo mi ser, para ellos es este logro y espero lo disfruten tanto como yo pues es dedicado a ustedes. Los amo, respeto y admiro mucho.

A mis hermanas.

Rocio por ser una excelente madre, esposa y hermana. Eres un ser humano hermoso. Te quiero mucho.

Rosario por tu ejemplo de inteligencia, capacidad, independencia y ganas de hacer las cosas bien. Te quiero mucho.

A Alejandra.

Por estar conmigo en las buenas y en las malas, por tu comprensión, cariño, paciencia, apoyo, y tu amor incondicional. Te Amo. También a tu familia por brindarme su cariño, su confianza y su apoyo, gracias.

A Alejandro.

Por ser mi sobrino consentido. Te quiero mucho hijo y espero esto te sirva de ejemplo para que tu también te superes como ser humano.

A mi amigo Oscar.

Por ser mi mejor amigo y estar conmigo en esta aventura. Gracias por tus consejos, apoyo y la amistad que siempre me has brindado.

A todos mis amigos, maestros y a la FES Aragón.

Porque todos colaboraron de manera importante en este triunfo.

Desarrollo de aplicaciones móviles utilizando Java 2 ME como lenguaje de programación en teléfonos celulares y/o Pocket PC.

ÍNDICE

Introducción.	
Metodología.	
Planteamiento del Problema.	
Objetivos.	
Justificación.	
Capítulo 1. Conceptos Básicos.	
1.1 Lenguajes de Programación (Panorama general).	1
1.1.1 Evolución de los Lenguajes de Programación.	2
1.1.2 Clasificación de los Lenguajes de Programación.	6
1.2 ¿Qué es Microsoft.NET?	10
1.2.1 Servicios Web.	11
1.2.2 Componentes de Microsoft .NET.	13
1.3 ¿Qué es J2ME?	14
1.3.1 Arquitectura de J2ME.	17
1.4 Pocket PC y Smartphone.	23
1.5 Redes Inalámbricas.	24
1.5.1 Ventajas de las Redes Inalámbricas.	25
1.6 Bases de Datos.	26
1.6.1 Ventajas de las Bases de Datos.	26
1.7 ¿Qué es WAP?	27
1.8 ¿Qué es Java Servlets?	28

1.9 ¿Qué es Java JSP?	28
1.10 ¿Qué es HTML?	29
1.11 ¿Qué es HTTP?	29
Capítulo 2. Los Dispositivos Móviles.	
2.1 Los Dispositivos Móviles.	30
2.1.1 Antecedentes de los Dispositivos Móviles.	31
2.1.2 Futuro de los Dispositivos Móviles.	32
2.2 Sistema Operativo para Dispositivos Móviles.	33
2.2.1 ¿Qué es el Sistema Operativo?	33
2.2.2 Historia de los Sistemas Operativos.	34
2.2.3 Objetivos de los Sistemas Operativos.	37
2.2.4 Funciones de los Sistemas Operativos.	38
2.2.5 Facilidades del Sistema Operativo.	41
2.2.6 Tipos de Sistemas Operativos.	42
2.2.7 Características del Sistema Operativo.	43
2.3 Diferencias entre el Sistema Operativo Windows CE y Windows XP, 2000, ME y 98	46
2.3.1 Windows CE.	46
2.3.2 Windows XP.	47
2.3.3 Windows 2000.	49
2.3.4 Windows ME.	50
2.3.5 Windows 98.	51

2.4 Aplicaciones para Dispositivos Móviles.	52
2.4.1 Características de las Aplicaciones Móviles.	53
2.5 Tecnologías de Comunicación en Dispositivos Móviles.	56
Capítulo 3. Las aplicaciones móviles.	
3.1 Métodos de comunicación.	58
3.1.1 Internet como nuevo "Canal de Comunicación".	59
3.1.2 Internet como canal (secundario) de Prensa escrita.	61
3.1.3 Internet como canal (secundario) de Radio.	62
3.1.4 Internet como canal (secundario) de TV.	63
3.1.5 Internet como nuevo "Medio de Comunicación".	63
3.2 Necesidades de las aplicaciones móviles.	64
3.3 Características de la Tecnología Microsoft.NET.	66
3.3.1 Herramientas de desarrollo de Aplicaciones.	66
3.3.2 .NET Framework.	68
3.3.3 .NET Compact Framework.	69
3.3.4 .NET Mobile Internet Toolkit.	70
3.3.5 Seguridad en Microsoft .NET.	71
3.3.5.1 Creación de un entorno de confianza.	72
3.3.5.1.1 Creación de una firma.	73
3.3.5.1.2 Comprobación de una firma.	73
3.3.5.2 Seguridad del almacén de objetos.	74
3.3.5.3 Seguridad de bases de datos.	74

3.3.5.4 Interfaces API de confianza.	75
3.3.5.5 Uso de SSL para la comunicación de red segura.	75
3.3.5.6 Cifrado de datos con CryptoAPI.	76
3.3.5.7 Uso de la API de almacén protegido.	77
3.3.5.8 Generación segura de datos aleatorios.	78
3.3.5.9 Aislamiento de datos confidenciales en una tarjeta inteligente.	78
3.4 Características de la Tecnología Java 2 ME.	80
3.4.1 Entorno de desarrollo J2ME Wireless Toolkit.	82
3.4.2 Entorno de desarrollo Sun One Mobile Edition.	83
3.4.3 Entorno de desarrollo JBuilder X.	83
3.4.4 Entorno de desarrollo NetBeans.	84
3.4.5 Entorno de desarrollo Nokia Developer Suite.	84
3.4.6 Seguridad en J2ME.	85
3.4.6.1 JSR-177 (Security and Trust Services API for J2ME).	85
3.4.6.2 MIDP 2.0	85
3.4.6.3 Modelo Sandbox de seguridad.	86
3.5 Características de Java Servlets y JSP.	87
3.5.1 Características de Java Servlets.	87
3.5.2 Funcionamiento de los Servlets.	88
3.5.3 Puesta en Funcionamiento de los Servlets.	89
3.5.4 Características de Java JSP.	90
3.5.5 Funcionamiento de los JSP.	91

3.5.6 Estructura de Directorios.	92
3.5.7 Características de WAP-WML.	93
Capítulo 4. Implementación de una aplicación móvil para telefonía celular y/o Pocket PC.	
4.1 Aplicaciones J2ME "Zombies en México" y "Recetario".	96
4.1.1 Juego Java "Zombies en México".	98
4.1.2 "Recetario".	102
4.2 Aplicación en HTML, WAP-WML, Java Servlets y JSP's "Agenda Virtual".	104
4.2.1 "Agenda Virtual".	106
Conclusiones.	119
Apéndice A Código del Juego Java "Zombies en México".	122
Apéndice B Código de la Aplicación Java "Recetario".	129
Apéndice C Código de la Aplicación "Agenda Virtual".	137
Glosario.	169
Bibliografía.	173

INTRODUCCIÓN

La presente investigación tiene como objeto de estudio el análisis de los lenguajes de programación que permiten crear aplicaciones para dispositivos móviles, los cuales hoy en día se han convertido en parte fundamental de la vida diaria de las personas.

Estos dispositivos son producto del desarrollo de la Tecnología móvil así que estudiando y analizando su funcionamiento desarrollaremos tres aplicaciones que de alguna manera sean útiles a los usuarios.

Para programar dichas aplicaciones hablaremos de las tecnologías que permiten hacerlo en particular la tecnología Java la cual pensamos es la más conveniente para nuestro proyecto, basándonos en el costo y optimización del código.

Como bien se sabe estos dispositivos utilizan redes inalámbricas e Internet para sus comunicaciones, por lo tanto abarcaremos estos temas para dar las bases sobre las cuales edificaremos nuestras aplicaciones.

No pretendemos crear nada nuevo únicamente queremos mostrar un panorama general en el desarrollo de aplicaciones entorno al mundo móvil, que

en nuestro criterio en un futuro no muy lejano será la tecnología predominante en las áreas de desarrollo de software.

METODOLOGÍA

La investigación que realizaremos es de tipo científica a nivel licenciatura, para la cual proponemos un cronograma de actividades que seguiremos estrictamente en el cumplimiento de los tiempos para entregas parciales o término de una etapa del proyecto.

El proyecto lo hemos dividido en dos grandes etapas que se distribuyen en los cuatro capítulos a desarrollar en el proyecto:

En la primera etapa llevaremos a cabo una investigación documental, para la cual visitaremos bibliotecas, sitios en Internet o cualquier otro lugar y fuente que nos puedan facilitar información teórica del tema especificado.

Para la segunda etapa la investigación será técnica-práctica. Para esta etapa analizaremos los entornos de programación y elegiremos el más adecuado para programar las aplicaciones necesarias.

PLANTEAMIENTO DEL PROBLEMA Y DELIMITACIÓN

Descripción:

Llevar a cabo la investigación y el análisis de las aplicaciones móviles en Teléfonos Celulares y/o Pocket PC con el fin de mostrar sus características, estructura y funcionamiento y así desarrollar una aplicación que muestre los beneficios y alcances de las nuevas tecnologías para el desarrollo de aplicaciones móviles.

Elementos:

Nuestra investigación contemplará 3 elementos importantes los cuales son:

Tecnologías para el desarrollo de aplicaciones móviles. Se describirán las características de las diferentes tecnologías, como son Java 2 ME y Microsoft.NET.

Dispositivos móviles. Historia de los dispositivos móviles y características que deben tener para la implementación de estas nuevas tecnologías o sea SmaPhone y Pocket PC.

Métodos de comunicación. Se mostrará la tecnología inalámbrica así como Internet como medio de comunicación.

Alcances y límites:

Al finalizar la investigación se tendrá el análisis de las diferentes tecnologías para el desarrollo de aplicaciones móviles, así como conocimientos generales de lo que son los dispositivos móviles y sus características; todo esto como marco de estudio para el desarrollo e implementación de una aplicación que muestre los alcances de dichas tecnologías.

Será una investigación teórica y se presentará en el dispositivo o si no en un emulador.

OBJETIVOS

Objetivo general.

Analizar las características de las tecnologías Microsoft.NET y Java 2 ME para el desarrollo de aplicaciones móviles y sus alcances, así como el funcionamiento de los dispositivos basados en la tecnología móvil con el fin de desarrollar e implementar una aplicación que muestre estos alcances.

Objetivos específicos.

1. Describir teóricamente los fundamentos de las nuevas tecnologías para el desarrollo de aplicaciones móviles así como sus características.
2. Conocer el funcionamiento de los dispositivos móviles actuales para conocer sus características y alcances.
3. Desarrollar una aplicación basándonos en los alcances de las aplicaciones móviles.

JUSTIFICACIÓN

Se llevará acabo ésta investigación con la finalidad de sustentar y construir nuestra Tesis que dará cierre a una etapa formativa como es la carrera de Ingeniería en Computación, a la vez que constituya un aporte que hacemos a la misma.

Nuestra investigación está orientada al análisis de las nuevas tecnologías Microsoft.NET y Java 2 ME para el desarrollo de aplicaciones móviles, las cuales hoy en día están en pleno crecimiento.

Como Ingenieros en Computación tenemos como objetivo aplicar nuestros conocimientos con el fin de facilitar o mejorar la calidad de vida automatizando o simplificando muchas de las actividades de las personas, por lo tanto pensamos mostrar los alcances de nuevas tecnologías de comunicación las cuales pueden mejorar la calidad de vida.

La importancia de nuestra investigación radica en la información de los alcances del desarrollo de software para las nuevas tecnologías, ya que dicha tecnología ha alcanzado en los últimos años gran importancia en las áreas de las comunicaciones pues en la actualidad muchos de los aparatos de

comunicación móvil están relacionados con las computadoras y con ellas se ha optimizado la forma de comunicación.

Los dispositivos móviles son actualmente necesarios para la comunicación a distancia, ya que con la ayuda de estos, se realiza un mejor intercambio y manipulación de información.

CAPÍTULO 1. CONCEPTOS BÁSICOS.

Para el desarrollo de nuestra Tesis utilizaremos conceptos como: Lenguaje de Programación, Redes Inalámbricas, Bases de Datos, Pocket PC, Smartphone, entre otros; en este primer capítulo se definen dichos conceptos con el fin de dar las bases necesarias sobre las cuales edificaremos nuestro proyecto.

1.1 Lenguajes de Programación.

Se le llama Lenguaje de Programación al conjunto de sintaxis y reglas semánticas que definen los programas del computador. Un lenguaje le da la capacidad al programador de especificarle al computador qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando lenguajes máquina, lenguajes ensambladores y lenguajes de alto nivel.

Lenguajes Máquina. Es el "lenguaje natural" de una computadora en particular y esta relacionado intimamente con el diseño del hardware de esa computadora además de que consisten por lo general en cadenas de números.

Lenguajes Ensambladores. Estos en vez de utilizar las cadenas de números usan abreviaturas similares al inglés para representar las operaciones

elementales de la computadora (utiliza programas de traducción denominados ensambladores para convertir los programas en lenguaje máquina).

Lenguajes de Alto Nivel. Desarrollados para acelerar el proceso de programación, en estos se pueden escribir simples enunciados para llevar acabo tareas sustanciales (los programas de traducción que convierten los programas de lenguaje de alto nivel al lenguaje máquina se llaman compiladores).

Es obvio que los lenguajes de alto nivel son mucho más deseables desde el punto de vista del programador, ya que utilizan un lenguaje relativamente próximo al lenguaje humano.

En conclusión un programa escrito en un lenguaje de programación necesita pasar por un proceso de compilación, interpretación o intermedio, es decir, ser traducido al lenguaje de máquina para que pueda ser ejecutado por el ordenador.

1.1.1 Evolución de los Lenguajes de Programación.

A medida que la complejidad de los programas ha ido aumentando, se han requerido de nuevos lenguajes para poder describir esos problemas y que sean resueltos por el ordenador.

Las generaciones de los lenguajes de programación, se han venido dando debido a que las necesidades que plantean los problemas son cada día más grandes y complejas, a continuación se hace un pequeño resumen de cada una de las generaciones de los lenguajes de programación.

Primera Generación.

Los lenguajes de primera generación o también conocidos como lenguajes máquina, son los que utilizan el código binario (unos y ceros) para comunicarse con la computadora, esta generación de lenguajes es muy compleja, ya que al usar pocos signos, no puede expresar cosas muy elaboradas.

Los lenguajes máquina son dependientes del ordenador, es decir, un lenguaje máquina particular puede ser utilizado en solo un tipo de computadora.

Segunda Generación.

Los lenguajes de esta segunda generación son conocidos también como ensambladores, y se distinguen de los lenguajes máquina por su eficiencia (en comparación con sus antecesores). Estos lenguajes ensambladores se basan en lo que es la comprensión de varias palabras en una sola, por ejemplo:

ADC significara "sumar con reserva"(en ingles: Add with Carry)

Haciendo notoria la aclaración, de que esta serie de instrucciones serán traducidas al lenguaje máquina por un programa de traducción denominado ensamblador.

Tercera Generación.

Los lenguajes de la tercera generación utilizan instrucciones de alto nivel y además son independientes de las máquinas. Una vez escrito el programa con instrucciones de alto nivel, se utiliza un programa llamado traductor que convierte a lenguaje máquina los programas escritos en lenguajes de alto nivel. Estos programas son parecidos a los ensambladores de la segunda generación, solo que a menudo tienen que compilar, reunir varias instrucciones de máquina para formar secuencias cortas que simularan la actividad solicitada por la instrucción de alto nivel.

Con la aparición de los lenguajes de tercera generación se alcanzó, en gran medida la meta de la independencia respecto a las máquinas. Como las instrucciones de estos lenguajes no hacían referencia a los atributos de ninguna máquina en particular, se podían compilar con la misma facilidad en una u otras máquinas. Unos de los primeros lenguajes de la tercera generación son FORTRAN y COBOL, por lo que algunas clasificaciones les colocan en la

segunda generación después de estos aparecen otros como BASIC y en los 70's aparecen lenguajes de alto nivel como PASCAL, C y más recientes son C++ y Java.

Cuarta Generación.

"Lenguajes orientados a problemas (4GL). Resultan más eficaces para la resolución de un tipo de problemas a costa de una menor eficiencia para otros. Requieren poca capacitación especial de parte del usuario, son considerados de muy alto nivel, diseñados para resolver problemas específicos incluye: lenguajes de consulta y generador de aplicaciones:

1. Lenguajes de consulta: Permiten a no programadores usar ciertos comandos de fácil comprensión para la búsqueda y generación de reportes a partir de una base de datos.
2. Generador de aplicaciones: Quiere decir que cuando se diseña uno de estos lenguajes, se tiene en cuenta que su finalidad es la resolución de problemas, prescindiendo de la arquitectura del computador. Contiene varios módulos que han sido preprogramados para cumplir varias tareas."¹

¹ Internet, [http:// www.mailxmail.com/curso/informatica/programacionestructurada/capitulo4.htm](http://www.mailxmail.com/curso/informatica/programacionestructurada/capitulo4.htm)

Quinta Generación.

"Lenguajes orientados a aplicaciones en inteligencia artificial, como lisp y prolog. Dentro de este campo destacan las aplicaciones en sistemas expertos, juegos, visión artificial (Jurasic Park) y robótica. Lisp es un lenguaje para procesamiento de listas y manipulación de símbolos. Prolog es un lenguaje basado en la lógica, para aplicaciones de bases de datos e Inteligencia Artificial. Podemos decir entonces, que los lenguajes de alto nivel, tienen las ventajas de mayor legibilidad de los programas, portabilidad, facilidad de aprendizaje y facilidad de modificación."²

1.1.2 Clasificación de los Lenguajes de Programación.

De acuerdo al número de instrucciones necesarias para realizar una tarea específica podemos clasificar a los lenguajes de programación en dos grandes bloques:

1. Lenguajes de Bajo Nivel.
2. Lenguajes de Alto Nivel.

Lenguajes de Bajo Nivel. Es el tipo de lenguaje que cualquier computadora es capaz de entender. Se dice que los programas escritos en

² Idem

forma de ceros y unos están en lenguaje de máquina, porque esa es la versión del programa que la computadora realmente lee y sigue.

Lenguajes de Alto Nivel. Son lenguajes de programación que se asemejan a las lenguas humanas usando palabras y frases fáciles de entender.

En un lenguaje de bajo nivel cada instrucción corresponde a una acción ejecutable por el ordenador, mientras que en los lenguajes de alto nivel una instrucción suele corresponder a varias acciones.

Características de los lenguajes de alto nivel. Son independientes de la arquitectura física de la computadora. Permiten usar los mismos programas en computadoras de diferentes arquitecturas (portabilidad), y no es necesario conocer el hardware específico de la máquina. La ejecución de un programa en lenguaje de alto nivel, requiere de una traducción del mismo al lenguaje de la computadora donde va a ser ejecutado. Una sentencia en un lenguaje de alto nivel da lugar, al ser traducida, a varias instrucciones en lenguaje entendible por el computador. Utilizan notaciones cercanas a las usadas por las personas en un determinado ámbito. Se suelen incluir instrucciones potentes de uso frecuente que son ofrecidas por el lenguaje de programación.

De acuerdo a las distintas filosofías que han seguido y la forma de trabajar que implican los Lenguajes de Programación, también llamados Paradigmas de Programación, se pueden clasificar en:

"Paradigma Imperativo: Son aquellos que facilitan los cálculos por medio de cambios de estado, entendiendo como estado la condición de una memoria de almacenamiento. Los lenguajes estructurados en bloques, se refieren a los ámbitos anidados, es decir los bloques pueden estar anidados dentro de otros bloques y contener sus propias variables. La RAM representa una pila con una referencia al bloque que está actualmente activo en la parte superior.

Paradigma heurístico: Define un modelo de resolución de problemas en el que se incorpora algún componente heurístico, sobre la base de una representación más apropiada de la estructura del problema, para su resolución con técnicas heurísticas.

Paradigma concurrente: La programación distribuida ha sido dividida en dos amplias categorías, sistemas acoplados en forma débil o fuerte. El término distribuido se refiere por lo general a lenguajes para sistemas acoplados débilmente que soportan un grupo de programadores trabajando en un programa particular de manera simultánea y comunicándose a través de paso de mensajes mediante un canal de comunicación. Un sistema acoplado fuertemente permite que más de un proceso en ejecución tenga acceso a la

misma ubicación de memoria. Un lenguaje acoplado con el sistema debe sincronizar el uso compartido de la memoria, de modo que solo un proceso escriba una variable compartida a la vez, y de modo que un proceso pueda esperar hasta que ciertas condiciones se satisfagan por completo antes de continuar la ejecución. La memoria compartida tiene la ventaja de la velocidad, por que no se necesita pasar mensajes.

Paradigma Funcional. Como su nombre lo dice operan solamente a través de funciones. Cada función devuelve un solo valor, dada una lista de parámetros. No se permiten asignaciones globales, llamados efectos colaterales. La programación funcional proporciona la capacidad para que un programa se modifique así mismo, es decir que pueda aprender.

Paradigma Lógico. Esta programación se basada en un subconjunto del cálculo de predicados, incluyendo instrucciones escritas en formas conocidas como cláusulas de Horn. Este paradigma puede deducir nuevos hechos a partir de otros hechos conocidos. Un sistema de cláusulas de Horn permite un método particularmente mecánico de demostración llamado resolución.

Paradigma Orientado a Objetos. Describen los lenguajes que soportan objetos en interacción. Un objeto es un grupo de procedimientos que comparten un estado. El término de orientado a objetos fue utilizado originalmente para

distinguir aquellos lenguajes basados en objetos que soportaban clases de objetos y la herencia de atributos de un objeto padre por parte de sus hijos.”³

Para finalizar mencionaremos que para el desarrollo de aplicaciones sobre dispositivos móviles Microsoft ofrece su plataforma .NET mientras que SUN Microsystems pone a disposición su plataforma J2ME (Java 2 Micro Edition). Vamos a analizar cada una.

1.2 ¿Qué es Microsoft .NET?

“Microsoft .NET es un conjunto de tecnologías de software que permiten conectar entre sí al mundo de la información, usuarios, sistemas y dispositivos. Permite un nivel sin precedente de integración de software a través del uso de Servicios Web, pequeños y discretos bloques de aplicaciones construidos que se conectan entre ellos, así como a otras aplicaciones grandes, vía Internet.”⁴

La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, sistema operativo y lugar, a los servicios Web y sus aplicaciones.

³ Internet, <http://www.fri.utn.edu.ar/sistemas/paradigmas/page24.html>

⁴ Internet, <http://www.microsoft.com/latam/net/basics/whatis.asp>

La idea central de la plataforma .NET es la de crear software como servicio así como construir, instalar e integrar en federaciones estos servicios para que puedan ser accedidos mediante Internet. Esto es posible debido a que tenemos la infraestructura de comunicación global, Internet, cada vez más rápida y a un costo cada vez menor, y además, a la capacidad de los procesadores que continúa incrementándose año tras año. El usuario de Internet debe con su explorador no solamente acceder a contenido de texto, imágenes y sonido, sino también poder hacer uso de los servicios Web.

Por otro lado .NET también trata de aprovechar el incremento en la capacidad de procesamiento de los nuevos dispositivos móviles llamados "Smart Devices" (dispositivos inteligentes) para que el usuario haga uso de la funcionalidad que proveen los servicios Web con interfases cada vez más sencillas y naturales como la voz o la escritura.

1.2.1 Servicios Web.

"Los Servicios Web son servicios con un interfaz definido y conocido al que se puede acceder a través de Internet. Al igual que una página Web está definida por un URL (Uniform Resource Locator), un Servicio Web está definido por un URI (Uniform Resource Identification) y por su interfaz, a través del cual se puede acceder a él."⁵

⁵ *Idem.*

Los Servicios Web se dividen en servicios de transporte, de mensajería, de descripción y de descubrimiento.

Los servicios de transporte establecen la conexión y el puerto usado. Los protocolos que utilizan son HTTP, SMTP, FTP o BEEP, este último es un protocolo específico para Servicios Web, que a diferencia de los anteriores, no es cliente-servidor sino "entre pares"; los dos ordenadores entre los que se establece la comunicación actúan como clientes y servidores a la vez. Es además extensible y está especificado en XML.

Los servicios de mensajería, especifican cómo se tiene que codificar el mensaje que contiene los datos que se intercambian entre el cliente y el servidor. El protocolo más usado en esta capa es el SOAP.

Los servicios de descripción WSDL (Web Services Description Language), permiten especificar la dirección de un servicio y el interfaz que se usa para acceder a él, sea SOAP o HTML.

Los servicios de descubrimiento UDDI (Universal Description, Discovery, and Integration), permiten no sólo describir Servicios Web, sino productos, la empresa en sí, y cómo está dispuesta a llevar a cabo transacciones.

También es importante mencionar, aunque fuera de la anterior clasificación a los Servicios Web XML.

Los Servicios Web XML permiten que las aplicaciones compartan información y que además llamen a funciones de otras aplicaciones independientemente de cómo se hayan creado, sin importar el sistema operativo o la plataforma en que se ejecuten y los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden relacionarse y formar un grupo de colaboración para realizar una tarea determinada.

1.2.2 Componentes de Microsoft .NET.

"El componente principal de .NET, es el Common Language Runtime (CLR) o máquina virtual común. Se trata de un programa que se puede ejecutar, en principio, en cualquier sistema operativo y que provee de una serie de servicios que se pueden usar desde diferentes lenguajes de programación."⁶

Los demás componentes de .Net permiten extender a todos los productos de Microsoft la funcionalidad de .Net:

- ASP.NET: Active Server Pages, en su versión para .Net.

⁶ *Idem*.

- VB.NET: versión para el CLR del Visual Basic, el lenguaje común a todas las aplicaciones de Microsoft.
- ADO.NET, acceso a objetos de datos (Access to data objects), que permite acceder de forma orientada a objetos a bases de datos; también da una serie de servicios para acceso a bases de datos y otros repositorios de objetos desde dentro de la CLR.
- Perl.NET, Python.NET son desarrollos de ActiveState, que se integran con el entorno Visual Studio.NET y permiten desarrollar programas en esos lenguajes.
- WinForms y WebForms, diseño gráfico de ventanas dentro de .NET.

Por lo tanto, una vez que hemos terminado de hablar de Microsoft y su plataforma .NET, hay que dar paso a la plataforma propuesta por SUN Microsystems Java 2 Micro Edition, que como todos los productos de SUN, es gratis y por consiguiente la que ocuparemos en nuestro proyecto.

1.3 ¿Qué es J2ME?

Java es un lenguaje de programación de alto nivel con el que se puede escribir tanto programas convencionales como para Internet.

Una de las ventajas significativas de Java sobre otros lenguajes de programación es que es independiente de la plataforma, tanto en código fuente

como en binario. Esto quiere decir que el código producido por el compilador Java puede transportarse a cualquier plataforma que tenga instalada una máquina virtual Java y ejecutarse. Pensando en Internet ésta característica es crucial ya que esta red conecta ordenadores muy distintos.

"[...] Sun Microsystems ha definido el lenguaje java simplemente como Plataforma Java 2 y ha redistribuido a ésta en tres ramas en función del sector al que va orientada la edición correspondiente, cada rama con su conjunto de API's y herramientas de desarrollo propias [...]:

- Java 2 Standard Edition (J2SE). Orientada a ordenadores de sobremesa. Comprende el JDK hasta ahora distribuido por SUN, en donde Swing se ha convertido en pieza clave y al que se han incorporado clases adicionales para facilitar el desarrollo de aplicaciones Java en donde la interfaz de usuario tiene una importancia muy especial.
- Java 2 Enterprise Edition (J2EE). Engloba al J2SE y lo potencia añadiéndole clases para la programación en entornos corporativos. Esta edición está orientada al desarrollo de aplicaciones para servidores utilizando Enterprise JavaBeans, aplicaciones Web, Servlets, JavaServer Pages, CORBA y Extensible Markup Language (XML).

- Java 2 Micro Edition (J2ME). Es un subconjunto de J2SE orientado al desarrollo de aplicaciones Java destinadas a dispositivos con pocos recursos.⁷

Java 2 Micro Edition (J2ME) fue presentada en 1999 por Sun Microsystems con el propósito de habilitar aplicaciones Java para pequeños dispositivos.

Java Micro Edition es la versión del lenguaje Java que está orientada al desarrollo de aplicaciones para dispositivos pequeños con capacidades restringidas tanto en pantalla gráfica, como de procesamiento y memoria (teléfonos móviles, PDA's, Handhelds, Pagers, etc). Esta edición tiene componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar), inclusión de un pequeño y rápido recolector de basura y otras diferencias.

La aparición de esta tecnología se debe a que las necesidades de los usuarios de telefonía móvil han cambiado mucho en estos últimos años y cada vez demandan más servicios y prestaciones por parte tanto de los terminales como de las compañías. Además el uso de esta tecnología depende del

⁷ Froufe Quintas, Agustín y Jorge Cárdenes, Patricia, *J2ME Java 2 Micro Edition Manual de Usuario y Tutorial*, México, 2004, p. 2.

asentamiento en el mercado de otras, como GPRS, íntimamente asociada a J2ME y que no ha estado a nuestro alcance hasta hace poco.

1.3.1 Arquitectura de J2ME.

Hay tres elementos claves en los que se puede dividir la arquitectura de J2ME, que son los siguientes:

1. Configuraciones.
2. Máquinas Virtuales.
3. Perfiles.

Configuraciones.

Una configuración en J2ME se define como el conjunto mínimo de API's Java que permiten desarrollar aplicaciones para un grupo de dispositivos determinados, a través de estas configuraciones se indican qué características básicas son comunes a todos los dispositivos de una determinada configuración.

Se han definido dos tipos diferentes de configuraciones para dar soporte a un gran número de dispositivos que están ya presentes en el mercado o vendrán en un futuro no muy lejano, son las configuraciones CDC y CLDC. En la práctica actualmente solo esta última, la configuración CLDC, está disponible, ya que es

la única que tiene dispositivos en el mercado sobre los cuales se pueden realizar desarrollos.

Configuración de dispositivos con conexión, CDC (Connected Limited Configuration).

Está orientada a dispositivos con cierta capacidad tanto de proceso como de memoria, como pueden ser determinados electrodomésticos inteligentes (televisores con Internet, neveras o sistemas de navegación). Sus características serían:

- El procesador es de 32 bits.
- Tienen al menos 2 Mb o más de memoria total.
- Poseen la funcionalidad completa de la Máquina Virtual Java 2, además de conectividad a algún tipo de red.
- Se basa en J2SE v1.3 e incluye varios paquetes Java de la edición estándar.

Configuración de dispositivos limitados con conexión, CLDC (Connected Limited Device Configuration).

Se orienta a dispositivos dotados de conexión y con limitaciones gráficas, de proceso y memoria, refiriéndonos a teléfonos móviles, PDA's, etc. Los dispositivos que hacen uso de esta configuración se caracterizan por:

- Memoria total disponible entre 160 Kb y 512 Kb. Se pide como mínimo 128 Kb de memoria no volátil para la máquina virtual Java y las bibliotecas CLDC, y 32 Kb de memoria volátil para la máquina virtual en tiempo de ejecución.
- Procesador de 16 o 32 bits y a 25 Mhz de velocidad mínima para que las aplicaciones puedan ejecutarse sin problemas.
- Tener conexión a algún tipo de red de manera intermitente y con un ancho de banda limitado. El tipo de conexión normalmente es inalámbrica.
- Bajo consumo (trabajan normalmente con baterías).

Máquinas Virtuales J2ME.

Cuando un programa es compilado en la plataforma Java se produce un código intermedio denominado bytecode, que es portable entre los entornos, de manera que es la máquina virtual del entorno donde queramos ejecutar el código

la que se encarga de traducir los bytecodes al código máquina de la plataforma de ejecución. Además de realizar ésta función también se encarga de otras funciones muy importantes como son las llamadas al Sistema Operativo y mantener las reglas de seguridad.

Sucede que la implementación de la JVM tanto para J2SE como para J2EE es muy pesada en términos de memoria y en necesidad de proceso. Por lo que fue necesario definir una JVM para J2ME que sea adecuada a las características limitadas y restringidas de los dispositivos móviles. Existen dos configuraciones con sus peculiaridades, por eso se definieron dos tipos de máquinas virtuales que dan soporte a cada una de las configuraciones, la KVM y la CVM.

La máquina virtual KVM.

Da soporte a la configuración CLDC y es la máquina virtual más pequeña y compacta de la que SUN dispone. El nombre de KVM proviene de la palabra Kilobyt, ya que hace referencia a su baja ocupación de memoria que oscila entre 35 Kb y 85 Kb. Una de las peculiaridades que tiene la KVM es la de estar escrita en lenguaje C.

Es una máquina virtual muy portable y muy modular, y en su diseño se han tratado de aunar dos factores muy importantes: el de ser completa y el de ser rápida, sin eliminar ninguna característica básica fundamental.

La máquina virtual CVM.

La CVM o Compact Virtual Machine es la máquina virtual que nos vamos a encontrar sobre la configuración CDC y dispone de las mismas características que la que existe para J2SE. Está muy orientada a dispositivos con procesadores de 32 bits y 2 Mb o más de RAM.

Perfiles.

Los perfiles se utilizan para establecer que API's definen las características de un dispositivo en particular, además agrupa los dispositivos por la funcionalidad que dan al usuario y los tipos de aplicaciones que se ejecutarán en ellos. Del mismo que las configuraciones están asociadas a las máquinas virtuales, existen diferentes perfiles que se asientan sobre cada una de las configuraciones vistas. Para la configuración CLDC se tienen los perfiles:

1. PDA Profile.
2. Mobile Information Device Profile (MIDP).

Mientras que para la configuración CDC los perfiles son:

1. Foundation Profile.
2. Personal Profile.
3. RMI Profile.

Los perfiles CLDC.

PDA Profile. Abarca las denominadas PDA's de gama baja como son las Palm. Actualmente se encuentra en fase de definición por lo que la información sobre este perfil es muy escasa.

Mobile Information Device Profile (MIDP). Fue el primer perfil definido para CLDC y se ajusta fundamentalmente a los teléfonos móviles, aunque también abarca cualquier otro dispositivo con conectividad, capacidad gráfica, de proceso y memoria reducida. Actualmente es el perfil más elaborado y el que dispone de dispositivos accesibles al usuario doméstico.

Los perfiles CDC.

Foundation Profile. Pensados para dispositivos que carecen de interfaz gráfica por lo que, aunque dispone de toda la funcionalidad J2SE, elimina los paquetes AWT y Swing que se usan para crear la interfaz gráfica de usuario.

Personal Profile. A diferencia del perfil anterior, éste incluye el soporte completo para la creación de gráficos mediante AWT.

RMI Profile. Este perfil se monta sobre una implementación del Foundation Profile al que añade un subconjunto de las API's de RMI.

Ya que hablamos de los lenguajes que nos permitirán programar aplicaciones móviles tenemos que decir que son los famosos dispositivos móviles Pocket PC y Smartphone.

1.4 Pocket PC y Smartphone.

La Pocket PC ofrece lo mejor del desktop de una PC de manera que se tenga en la palma de la mano y con versiones Pocket de las aplicaciones Windows Powered, Outlook, Word, Excel, Microsoft Reader, Windows Media Player, Internet Explorer. Pocket PC es una poderosa computadora portátil.

El Smartphone o "Teléfono Inteligente" combina las características de un teléfono celular con la funcionalidad de un asistente personal digital (PDA), en forma de teléfono compacto e inalámbrico que se puede manejar con una sola mano. Entre sus funciones se incluye voz, SMS y servicios de mensajería instantánea; correo electrónico que se puede recibir desde diversos orígenes, como los servicios del cliente de mensajería y colaboración Outlook, Exchange,

IMAP y POP3, y aplicaciones de administrador de información personal (PIM), como las de calendario y contactos.

Para estos dispositivos la manera de comunicación es a través de Redes Inalámbricas, así que ahora hablaremos de ellas.

1.5 Redes Inalámbricas.

Las redes inalámbricas son formas versátiles de transferir datos. Pueden operar sobre una variedad de ondas de radio, desde el espectro infrarrojo hasta las bandas del teléfono celular.

Especialmente las redes de área local inalámbrica no son del todo inalámbricas (aunque si existen las que son totalmente inalámbricas). Con mucha frecuencia, las redes de área local inalámbricas se arman de manera similar a las redes celulares, con diversos puntos de acceso inalámbrico conectados a una red Ethernet estándar.

Cuando una computadora conectada a la red de área local inalámbrica se mueve alrededor de la red, la computadora detecta qué punto de acceso está más cercano a la laptop y usa ese punto de acceso. El rango local de redes de espectro extendido cubre hasta cerca de 50,000, pies cuadrados (es un cuadro

de 225 pies por lado). Algunos fabricantes hacen sistemas que con el uso de una antena externa, pueden transmitir hasta cinco millas.

1.5.1 Ventajas de las Redes Inalámbricas.

Las LAN inalámbricas ofrecen un buen número de ventajas para quienes trabajan en red, en particular en grandes organizaciones. A continuación se describen varios de los principales beneficios:

- Simplifica altas, movimientos y cambios. Actualmente este proceso en las redes alambreadas es comparativamente incómodo.
- Despliegue rápido. Las redes inalámbricas funcionan bien en situaciones en las que los requerimientos establecen que la red debe ser transportable, de rápido despliegue y muy confiable.
- Ofrece movilidad a los usuarios que no trabajan en una sola ubicación, no hay necesidad de una estación de trabajo de red fija.

Por último hablaremos de las Bases de Datos ya que, toda la información que viaja a través de las redes inalámbricas, que es generada por los dispositivos móviles y administrada por las aplicaciones desarrolladas con los lenguajes de programación, debe estar guardada en algún lugar para su correcta organización y así aprovecharla al máximo.

1.6 Bases de Datos.

La historia de las bases de datos surge desde mediados de los años sesenta sin embargo en 1970 Codd propuso el modelo relacional el cual ha marcado la línea de investigación por muchos años, pero ahora se encuentran también los modelos orientados a objetos.

“Las bases de datos son una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los Sistemas de Información de una empresa o negocio en particular. Las bases de datos proporcionan la infraestructura requerida para los Sistemas de Apoyo a la Toma de Decisiones y para los Sistemas de Información Estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas.”⁸

1.6.1 Ventajas de las Bases de Datos.

La utilización de bases de datos se ha incrementado notablemente en los últimos años y esto se debe a las ventajas que ofrece el uso de ellas, entre las que podemos mencionar:

* Internet, <http://elizabethpeguero.8m.com/Eliza.htm>

- Globalización de la información: permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- Eliminación de información inconsistente.
- Permite compartir información.
- Permite mantener la integridad en la información: la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.
- Independencia de datos: el concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos.

1.7 ¿Qué es WAP?

WAP (Wireless Application Protocol) es un protocolo basado en los estándares de Internet desarrollado para permitir a teléfonos celulares navegar a través de Internet. Con la tecnología WAP se pretende que desde cualquier teléfono celular WAP se pueda acceder a la información que hay en Internet así como realizar operaciones de comercio electrónico.

1.8 ¿Qué es Java Servlets?

"Un servlet es un programa en código Java que se ejecuta en un servidor Web. Puede ser invocado utilizando el protocolo http, un servlet es cargado y ejecutado directamente en el servidor Web y regresa resultados que podrán ser mostrados mediante páginas HTML y/o WAP.

Sin embargo esta no es la única función de un servlet; puede realizar otras tareas como comunicarse con otro servlet para ayudar en su trabajo, o bien facilitar el acceso a una Base de Datos."⁹

1.9 ¿Qué es Java JSP?

"Es una página Java en servidor (JSP, acrónimo del inglés JavaServer Page) es una plantilla para una página Web que emplea código Java para generar un documento HTML y/o Wap dinámicamente. Las páginas JSP se ejecutan en un componente del servidor conocido como contenedor de JSP que las traduce a Servlets Java equivalentes.

Por esta razón los Servlets y las páginas JSP están íntimamente relacionadas. Lo que se puede hacer con una tecnología es, en gran medida, también posible con la otra; aunque cada una tiene capacidades propias."¹⁰

⁹ Ceballos, F. Javier, JAVA 2 Curso de Programación 2ª Edición, México 2003, pag. 737

¹⁰ Hanna, Phil, Manual de Referencia JSP, México 2002, pag. 47

1.10 ¿Qué es HTML?

"Lenguaje de marcado de hipertexto (HTML), es el lenguaje empleado para describir el interior de los documentos Web , el protocolo de transferencia de hipertexto (http), es el lenguaje empleado para describir como se envían estos documentos por Internet."¹¹

1.11 ¿Qué es HTTP?

"Http proporciona las normas para que los navegadores hagan peticiones y los servidores entreguen respuestas. Este conjunto de normas, o protocolo, incluye la manera de:

1. Solicitar un documento por su nombre.
2. Ponerse de acuerdo en el formato de los datos.
3. Determinar quien es el usuario.
4. Decir como manejar recursos obsoletos.
5. Indicar los resultados de una petición."¹²

¹¹ Hanna, Phil, Manual de Referencia JSP, México 2002, pag. 11

¹² Hanna, Phil, Manual de Referencia JSP, México 2002, pag. 11

CAPÍTULO 2. LOS DISPOSITIVOS MÓVILES.

En este capítulo hablaremos de lo que son los dispositivos móviles, su historia, sus características y el Sistema Operativo que utilizan. Además explicaremos que son y como se crean las aplicaciones para estos dispositivos.

2.1 Los Dispositivos Móviles.

El mundo de "lo móvil" está de moda, los usuarios de telefonía móvil se han multiplicado convirtiéndose en el mayor y más difundido exponente de ese mundo. El teléfono no es el único juguete móvil que últimamente adquiere el usuario de un cierto nivel técnico, ahí están los reproductores MP3, las consolas de juegos, las agendas, asistentes personales y los ordenadores portátiles.

A ese mundo en miniatura hay que añadir a los sistemas informáticos móviles, conocidos como Palmsize PC, Handheld, Pocket y similares. Sus características técnicas limitan hasta cierto punto las posibilidades de estos sistemas respecto a un ordenador corriente, pero hay que tener en cuenta que muchos de ellos tienen una potencia de proceso y capacidad similares a los equipos de sobremesa de hace pocos años. La diferencia es que esa potencia y capacidad ahora pueden transportarse en un bolsillo.

2.1.1 Antecedentes de los Dispositivos Móviles.

A mediados de los ochenta, la empresa Psion lanzó su Psion Organiser que tenía unas dimensiones reducidas, cabía en la palma de la mano y contaba con una pantalla de una sola línea de texto y un teclado completo.

Apple presentó en 1993 el Newton MessagePad de dimensiones algo mayores que el Psion Organiser, lo que llamaba la atención era la ausencia de teclado y las grandes dimensiones de la pantalla. Ésta, además, presentaba una interfaz gráfica, no sólo líneas de texto. Utilizando un lápiz, el usuario podía escribir sobre la pantalla, que era táctil. La selección de opciones resultaba igualmente fácil pulsando con ese lápiz.

Pocos años más tarde, a mediados de los noventa, la empresa US Robotics, tras adquirir Palm Computing, presentó la serie de dispositivos Palm Pilot. Este dispositivo carecía de teclado y contaba con una pantalla táctil. El éxito de este PDA (Personal Digital Assistant) dio lugar al actual puesto preferente de Palm en el mundo de los dispositivos móviles.

El último concursante en entrar en el campo de los dispositivos móviles fue Microsoft, con su sistema operativo Windows CE. En 1996, con la presentación de la primera versión de este sistema, varias decenas de fabricantes de hardware adquirieron el compromiso de diseñar dispositivos que

utilizaran dicho sistema operativo. A diferencia de Apple, Psion o Palm, Microsoft no es una empresa que fabrica un dispositivo con un sistema operativo en su interior, sino que pone ese sistema a disposición de terceros fabricantes que, como Compaq y HP, fabrican el hardware.

2.1.2 Futuro de los Dispositivos Móviles.

A corto plazo, todo parece indicar que los actuales PDA, los Smartphone, las Pocket PC e incluso las consolas de juegos, irán acercando posibilidades hasta convertirse en un mismo dispositivo, que nos permitirá navegar por la red más rápido y sin necesidad de cables ni módems.

El entretenimiento, más que la productividad, irá adquiriendo una mayor importancia en estos dispositivos. Es la ley del mercado la que ha hecho posible que los teléfonos móviles no sean una herramienta de ejecutivos, sino un electrodoméstico más de uso cotidiano. Estos PDA del futuro no serán sólo para usuarios de hojas de cálculo y correo electrónico, sino también para la diversión de jóvenes que oirán la última música obtenida de Internet, se comunicarán por video con sus amigos y jugarán en grupos con sus dispositivos gracias a Bluetooth y la posibilidad de crear redes sin cables.

2.2 Sistema Operativo para Dispositivos Móviles.

"La plataforma de Pocket PC y Smartphone se basa en el sistema operativo Windows CE 3.0 de Microsoft que incluye muchas de las características y funciones basadas en Win32 (incluidas las API de Win32, como TAPI y Winsock), además se incorporan numerosas aplicaciones como son: correo electrónico, herramientas PIM y software de Web Pocket Internet Explorer. Este tipo de software es compatible con los formatos HTML, WAP (WML) y XML."¹³

2.2.1 ¿Qué es el Sistema Operativo?

El Sistema Operativo es un conjunto integrado de programas que se utilizan para administrar los recursos y operaciones en general de un sistema de cómputo. El SO permite supervisar automáticamente sus propias operaciones llamando a los programas de aplicación traduciendo cualquier otro programa de servicio y administrando los datos necesarios para producir los resultados esperados por el usuario. Por lo tanto el SO tiende a aislar al hardware del usuario.

Se define al SO como una colección de programas de control, de administración y servicios que controlan todos los aspectos del ambiente en que

¹³ Internet. <http://www.microsoft.com/spanish/msdn/articulos/archivo/121202/voices/devappsp.asp>

funcionarán los programas de aplicación y que junto con el hardware los ejecutan.

2.2.2 Historia de los Sistemas Operativos.

Los Sistemas Operativos, al igual que el Hardware de las computadoras, han sufrido una serie de cambios revolucionarios llamados generaciones.

Generación Cero (década de 1940).

Los primeros sistemas computacionales no poseían sistemas operativos. Los usuarios tenían completo acceso al lenguaje de la máquina. Todas las instrucciones eran codificadas a mano.

Primera Generación (década de 1950).

Los sistemas operativos de los años cincuenta fueron diseñados para hacer más fluida la transición entre trabajos. Este fue el comienzo de los sistemas de procesamiento por lotes, donde los trabajos se reunían por grupos o lotes. Cuando el trabajo estaba en ejecución, éste tenía control total de la máquina, al terminar cada trabajo, el control era devuelto al sistema operativo, el cual limpiaba, leía e iniciaba el trabajo siguiente.

La introducción del transistor a mediados de los 50's cambió la imagen radicalmente.

Para poder correr un programa, se tenía que escribir en Fortran o en lenguaje ensamblador y después se perforaría en tarjetas, enseguida se llevaría la pila de tarjetas al cuarto de introducción al sistema y la entregaría a uno de los operadores. Cuando la computadora terminara el trabajo, un operador se dirigiría a la impresora y desprendería la salida y la llevaría al cuarto de salida, para que la recogiera el programador.

Segunda Generación (a mitad de la década de 1960).

La característica de los sistemas operativos fue el desarrollo de los sistemas compartidos como es la multiprogramación, y los principios del multiprocesamiento. En los sistemas de multiprogramación, varios programas de usuario se encuentran al mismo tiempo en el almacenamiento principal y el procesador se cambia rápidamente de un trabajo a otro. En los sistemas de multiprocesamiento se utilizan varios procesadores en un sólo sistema computacional, con la finalidad de incrementar el poder de procesamiento de la máquina.

Tercera Generación (mitad de década 1960 a mitad década de 1970).

Se inicia en 1964 con la introducción de la familia de ordenadores Sistema/360 de IBM. Los ordenadores de esta generación fueron diseñados como sistemas para usos generales. Eran sistemas grandes, voluminosos, de modos múltiples, algunos de ellos soportaban simultáneamente procesos por lotes, tiempo compartido, procesamiento de tiempo real y multiprocesamiento. Estos sistemas introdujeron mayor complejidad a los ambientes computacionales.

Cuarta Generación (mitad de década de 1970 en adelante).

Los sistemas de la cuarta generación constituyen el estado actual de la tecnología.

Con la ampliación del uso de redes de computadoras y del procesamiento en línea, los usuarios tienen acceso a computadoras alejadas geográficamente a través de varios tipos de terminales. Los sistemas de seguridad se han incrementado mucho ahora que la información pasa a través de varios tipos vulnerables de líneas de comunicación. La clave de cifrado está recibiendo mucha atención; han sido necesario codificar los datos personales para que, aún si los datos son expuestos, no sean de utilidad a nadie más que a los receptores adecuados.

Los sistemas de bases de datos han adquirido gran importancia. Nuestro mundo es una sociedad orientada hacia la información, y el trabajo de las bases de datos es hacer que esta información sea consultada de una manera controlada para aquellos que tienen derechos de acceso.

2.2.3 Objetivos de los Sistemas Operativos.

El objetivo fundamental de los SO consiste en reducir el costo de ejecutar los programas aumentando la utilización de varios componentes del sistema limitando el tiempo perdido, además:

- Optimizan el rendimiento y simplifican el uso efectivo del sistema.
- Administran los recursos del hardware y proporcionan un aspecto más accesible a los usuarios en lo relacionado a sus programas.
- Mayor productividad del operador.
- Adaptabilidad de los programas a los cambios de componentes del sistema.
- Posibilidad de extensión de las capacidades funcionales del SO de manera que satisfagan circunstancias cambiantes.
- Tiempos de respuesta mejorados que satisfagan más fácilmente requerimientos de tiempo real.
- Aumentan la eficiencia y con ello disminuyen el costo del uso de sistema.

- Mejoran el uso del hardware al automatizar la corriente de trabajos y realizar decisiones sobre el manejo de los recursos del sistema en la escala del tiempo del sistema y no en la escala del tiempo hombre.

2.2.4 Funciones de los Sistemas Operativos.

Los SO a través de sus programas realizan múltiples funciones como lo son:

- Decidir cuales usuarios pueden tener acceso a los recursos y durante cuanto tiempo.
- Suplir una caída o colapso si algún dispositivo sufre alguna falla.
- Asigna el camino a un dispositivo pasando por los procesadores de entrada y salida.
- Administra las rutinas preempaquetadas (compiladores, ensambladores, traductores de clasificación y búsqueda, funciones trigonométricas, etc.).
- Ayuda a la elaboración y ejecución de programas del usuario.
- Proporciona un medio ambiente en el que puedan trabajar determinados paquetes de software.
- Administra y lleva registro de los recursos (memoria, procesador, dispositivos, información), los asigna y los recupera.
- Operaciones de ordenamiento físico y temporal de las operaciones y tareas de control de tráfico.

- Autoprotección contra el usuario y protección de cada usuario contra los demás.
- Administración de almacenamiento secundario (disco duro).
- Manejo de las operaciones de entrada/salida y manejo de los programas problema.
- Organiza los archivos de información requerida por los distintos usuarios.
- Organiza el almacenamiento en disco en una estructura de archivos lógicos.
- Recibe las señales de los dispositivos electrónicos.
- Permite la recepción, procesamiento y despacho de mensajes remotos.
- Controla y supervisa a todos los programas. Fija un límite de tiempo para cada programa y un límite al número de líneas de salida.
- Asigna memoria a los programas, guarda la dirección absoluta de un programa en donde comienza y su tamaño.
- Reporta al procesador la entrada de programas a la memoria.
- Permite que un programa sea independiente de:
 - a) Nombres particulares de dispositivos de entrada/salida. Se usan nombres simbólicos para los dispositivos.
 - b) Nombres de archivos que permiten el nombramiento simbólico de archivos.
 - c) Tipos de dispositivos en los que las operaciones de entrada/salida pueden ser independientes del tipo de dispositivo en el que reside

el archivo actual, es decir, coordina donde está un archivo y a donde lo manda.

- Suministra una utilización eficiente del procesador, así como de los dispositivos de entrada/salida.
- Suministra una programación y la carga automáticamente de los programas a la memoria.
- Suministra la formación de colas y el envío de resultados de operaciones de entrada/salida.
- Proporciona un medio de tener control después de que se introdujo un programa salvando el estado del programa interrumpido y determinando la rutina requerida para procesar. A este proceso se le conoce como interrupción (ejecución no secuencial de un programa).
- Suministra una bitácora de lo que sucedió durante el proceso de todos los programas y el tiempo requerido para cada trabajo.
- Tratamiento de errores (de hardware y software).
- Suministra un programa para el manejo de interrupciones, coordina la transferencia de control entre los programas después de una interrupción. Tales interrupciones pueden ser del tipo entrada/salida, externas, de programa y las requeridas por el SO.

2.2.5 Facilidades del Sistema Operativo.

Multiprogramación: es la técnica de poder atender una o más tareas en forma concurrente.

Técnica que permite tratar varias tareas o fragmentos de ellas diferentes ejecutándose la intercalación de las mismas bajo el control de un programa supervisor, es decir, sobre esas tareas o fragmentos de tareas se efectúa un tratamiento en paralelo generalmente con un sólo procesador.

Asignación dinámica de espacio en disco: el sistema signa espacio a los archivos en una sección de disco llamada FAT (Tabla de Asignación de Archivos) que es un directorio que contiene la ubicación de los archivos.

Enlace y carga dinámica: en la mayoría de los sistemas y áreas de datos se cargan y agregan estaticamente el espacio de direcciones de trabajo durante el proceso de enlace y carga antes del inicio de ejecución.

Multiprocesamiento: es la ejecución de dos o más procesos en un sistema de procesadores multiple.

Protección de memoria: el SO compartirá la memoria con varios otros programas que son cargados en el resto de la memoria. Para evitar

interferencias mutuas se guarda la dirección absoluta en la cual empieza el programa en el registro base y en el registro límite el tamaño del bloque del programa.

2.2.6 Tipos de Sistemas Operativos.

SO tipo Batch. Cuando se desarrollaron por primera vez los sistemas Batch los trabajos se agrupaban y ejecutaban, pero sin embargo por contra únicamente con targetas y con cintas, el acceso a los datos y programas era secuencial es decir, uno a la vez.

SO Interactivo. Facilita la comunicación en línea entre el usuario y el sistema. El usuario da instrucciones al sistema operativo recibiendo una respuesta inmediata y generalmente emplea un teclado y una impresora o pantalla para la salida.

La mayoría de los sistemas tienen editores de texto interactivos para producir programas y un depurador interactivo para la pila de depuración de programas, un sistema interactivo se caracteriza por el tipo de respuesta muy breve.

SO en tiempo real. Se utiliza frecuentemente como dispositivo de control en una aplicación especializada. Unos sensores proporcionan los datos al

procesador. El procesador se encarga de analizar los datos ajustando en lo posible los controles para modificar las entradas del sensor. Los sistemas de tiempo real son empleados en los sistemas médicos, sistemas industriales de control y algunos sistemas de tráfico. Un sistema en tiempo real tiene restricciones fijas bien definidas, el procesamiento tiene que realizarse dentro de los límites definidos de lo contrario el sistema fallará.

SO de tiempo compartido. Se caracteriza por utilizar la planificación del procesador y la multiprogramación, para dotar a cada usuario de una pequeña parte del procesador compartido. Cada usuario tiene en memoria un programa independiente cuando se ejecuta lo hace normalmente durante un breve tiempo antes de acabar o mandar las instrucciones de entrada/salida.

2.2.7 Características del Sistema Operativo.

Eficiencia:

- a) Mínimo tiempo para cambiar de proceso. Debe ser capaz de tomar un tiempo mínimo para el cambio de un proceso a otro, ya sea en ambiente multiprogramación o en dar los resultados de un proceso e iniciar otro.
- b) Utilizar al máximo el procesador. Debido a que en un sistema quien realiza los procesos es el procesador, del uso máximo de él dependerá la productividad, la producción es con base al número de trabajos por hora,

en tonces el SO debe evitar en todo momento que el procesador este ocioso.

- c) Tiempo de proceso mínimo de los trabajos. El SO debe proporcionar el uso óptimo del procesador y los recursos para evitar duplicidad de operaciones, un ejemplo puede verse si el SO cambiara de procesos en forma indiscriminada, no tendría productividad a pesar de trabajar demasiado.
- d) Mínimo de tiempo de respuesta. Se refiere a la utilización de los recursos combinando con los diferentes procesos para obtener una mayor utilización y dejar al mínimo tiempo las unidades ociosas.
- e) Mayor productividad. Esto significa que el sistema debe producir el mayor número de trabajos posibles por hora, lo cual se logra con una optima administración de todos los recursos.

Confiabilidad.

El SO debe ser confiable, ya que todos los procesos obedecen a una secuencia discreta de operaciones.

Los procesos del SO son muy complejos y elaborados, por lo tanto es innecesario realizar pruebas estratégicas y generalmente esto se consigue con la práctica y desarrollo de programas de aplicación.

Mantenimiento.

Se refiere a dos puntos fundamentales:

- a) De corrección de errores y fallas en las cuales la solución se requiere de inmediato ya que puede obstaculizarse el desarrollo de otros procesos.
- b) El crecimiento. Que implica adaptarse a nuevos aditamentos, por ejemplo agregar nuevos compiladores, nuevas máquinas de ayuda, manejadores de impresión, manejadores de bases de datos, etc.

Reducción de tamaño.

Para que pueda cumplir con todas las características anteriores un SO lógicamente debe ser muy grande, sin embargo, se requiere que ésta función no exista o al menos disminuya en lo más posible. Es necesario entonces realizar rutinas y procedimientos simples en la elaboración del SO ya que, ésta facilita cumplir con las características solicitadas y disminuir su tamaño que además permitirá la utilización de menos recursos (memoria, espacio en disco, tiempo de procesamiento).

2.3 Diferencias entre el Sistema Operativo Windows CE y Windows XP, 2000, ME y 98.

Comenzaremos mencionando que Windows CE es el Sistema Operativo desarrollado por Microsoft especialmente para los Dispositivos Móviles y que la familia de Windows 98, ME, 2000 y XP es la evolución que ha tenido el Sistema Operativo para PC's. A continuación describiremos a detalle las características de cada uno de estos Sistemas Operativos.

2.3.1 Windows CE.

Windows CE es el sistema operativo modular de tiempo real para dispositivos móviles de 32 bits inteligentes y conectados. Windows CE combina la compatibilidad y los servicios de aplicación avanzados de Windows con soporte para múltiples arquitecturas de CPU y opciones incluidas de comunicación y redes proporcionando una rica y escalable fundación abierta para crear una amplia variedad de productos. Windows CE impulsa a los dispositivos electrónicos del cliente, terminales Web, dispositivos de acceso a Internet, controladores industriales especializados, computadoras de bolsillo y dispositivos de comunicación incrustados. Esta plataforma altamente modular permite a los desarrolladores crear de una manera confiable y flexible la nueva generación de dispositivos móviles de 32-bits que se integran fácilmente con Windows y la Internet.

2.3.2 Windows XP.

Desde que apareció Windows95 las sucesivas versiones han sido una evolución de la original, sin embargo con Windows XP se ha producido un cambio de mayor importancia ya que se ha cambiado el núcleo o Kernel del sistema operativo, así que podemos considerarlo como un nuevo sistema y no una versión más de Windows.

Hasta ahora Microsoft disponía de dos sistemas operativos diferentes, para el entorno personal o doméstico tenía Windows98 y para el entorno profesional el Windows NT/2000. Con Windows XP se produce una convergencia entre ambas versiones ya que se ha partido del núcleo del sistema de Windows 2000 para crear Windows XP y a partir de ahí se han realizado algunos retoques para diferenciar dos versiones de Windows XP, una para el ámbito personal llamada Windows XP Home Edition, y otra para el ámbito profesional denominada Windows XP Professional.

El principal beneficio de esta estrategia para los usuarios domésticos es que Windows XP ha adquirido la robustez y estabilidad de Windows NT/2000, lo que quiere decir que Windows XP se quedará menos veces bloqueado, y la mejora para los usuarios profesionales es que Windows XP tiene mayor compatibilidad con el hardware de la que gozaba Windows NT/2000.

Windows XP dispone de un nuevo sistema de usuarios completamente diferente respecto a Windows98. Ahora se pueden definir varios usuarios con perfiles independientes. Esto quiere decir que cada usuario puede tener permisos diferentes que le permitirán realizar unas determinadas tareas. Cada usuario tendrá una carpeta "Mis documentos" propia que podrá estar protegida por contraseña y un menú de inicio diferente. También se dispone de una carpeta a la que tienen acceso todos los usuarios y donde se pueden colocar los documentos que se quieren compartir con los demás usuarios.

Windows XP es más seguro, ésta versión incluye un Firewall básico que impide el acceso de intrusos por Internet, ofrece varias opciones para resguardar los archivos privados y tiene una herramienta que permite 'regresar' una PC con fallas hasta un punto en el cual funcionaba bien.

Para usar varias de las opciones de seguridad de Windows XP es necesario convertir el disco duro al sistema de archivos NTFS (NT File System), que es otra de las novedades de Windows XP. Un sistema de archivos es el método que emplea un sistema operativo para almacenar y organizar los archivos en la computadora, NTFS es el sistema de archivos que se utiliza en Windows NT/2000, y es más avanzado que el sistema de Windows 95, 98 y ME (FAT y FAT32). NTFS permite bloquear el acceso a carpetas o archivos específicos, además, puede manejar archivos y discos de cientos o miles de GB.

2.3.3 Windows 2000.

Windows 2000 representa un esfuerzo por unificar lo que hasta ahora eran dos sistemas operativos distintos, Windows 9x y Windows NT. Windows 2000 es la nueva versión de Windows NT 4.0 WorkStation y NT Server, pero también incorpora la sencillez de manejo de la serie 9x. Dicho en otras palabras, Windows 2000 ofrece lo mejor de ambos mundos: la solidez y la seguridad de NT, junto a la facilidad de manejo, soporte de hardware y multimedia de Windows 98.

Entre lo mejor de Windows 98 que ofrece la versión 2000, se encuentra el soporte de hardware, la interfase renovada, incluso, la presencia de Internet Explorer 5 y del Reproductor de medios, y soporte para las nuevas tecnologías como USB, FAT32, Administración Avanzada de Energía, etc.

Otros rasgos importantes son la abundancia de herramientas de conectividad, madurez de la interfaz, buen reconocimiento del hardware y estabilidad. Se añade a esto el soporte de nuevas tecnologías, las mejoras en sus funciones de informática remota, aplicaciones centralizadas de servicio y reinicios obligatorios drásticamente reducidos.

La familia Windows 2000 está integrada por cuatro versiones:

- Windows 2000 Professional: sucesor de NT Worksta-tion, está destinado a ser un cliente de red seguro y una estación de trabajo corporativa.

Soporta hasta 2 procesadores y es útil, como sistema operativo autónomo, para correr aplicaciones de alta performance, especialmente en diseño gráfico.

- Windows 2000 Server: sucesor de NT Server, soporta hasta 4 procesadores y está destinado a ser el servidor de impresión, archivos, aplicaciones e, incluso, Web de una empresa pequeña a mediana.
- Windows 2000 Advanced Server: sucesor de NT Server Enterprise Edition, soporta hasta 8 procesadores y será el servidor departamental de aplicaciones en empresas medianas a grandes, con más de un dominio y tareas de misión crítica.
- Windows 2000 Data Center Server: soporta hasta 32 procesadores y sólo se entregará sobre pedido. Está destinado a grandes empresas que requieran data warehousing, análisis econométricos, simulaciones científicas a gran escala, etc.

2.3.4 Windows ME.

Con características específicamente diseñadas para usuarios en casa, Windows Millennium Edition (Windows Me) hace del uso de su PC más fácil y divertido. Windows Me da el poder de poner en red a sus computadoras caseras, comunicarse más eficientemente por Internet y trabajar con rico contenido multimedia, como es el caso de fotografías, videos y música. Asistentes hechos a su medida y un rediseñado centro de ayuda (Help Center)

hacen de la configuración y de la solución a problemas algo más rápido y sencillo.

Con las herramientas incluidas en Windows Me, se puede compartir todo desde películas caseras hasta álbum de fotografías. Es más sencillo que nunca almacenar, administrar y compartir fotografías digitales, escenas de películas y archivos de audio. Windows Me ofrece animaciones más vivas, con mejor respuesta, mejor control del sonido y calidad, rápidas y sencillas maneras de importar fotografías, amplias capacidades de plug and play y una herramienta integrada para edición de video.

Windows Me facilita las cosas a los usuarios de PC's de diversas maneras, por medio de asistentes, tours, recursos de ayuda y soporte para el hardware más nuevo. Windows Me también tiene una mejora sobre varias versiones de Windows mediante la proporción automática de lo último en actualizaciones y parches (batches) a su escritorio, y permitiéndole a su PC recuperar fácilmente la configuración si es que algo malo sucediera con ellas. Para mayor seguridad, los archivos que se necesitan para que su PC esté en óptimas condiciones están siempre protegidos.

2.3.5 Windows 98

Este Windows soporta todos los tipos de Hardware, incluyendo a los estándares más recientes como el DVD-ROM y la capacidad para ver televisión en la PC.

También ofrece características plug and play, Windows 98 el siguiente escalón en la familia de sistemas operativos Windows de escritorio. De cierta forma es la continuación que podía esperar de Windows 95. Como era obvio predecir, esta nueva versión continúa soportando 32 bits en su total dimensión aunque todavía se debe esperar para que se incorpore toda la funcionalidad de seguridad presente en los 32 bits y que hoy es una característica de la familia NT.

2.4 Aplicaciones para Dispositivos Móviles.

Con la creciente popularidad de los dispositivos informáticos móviles, entre los que se encuentran los Asistentes Digitales Personales (PDA), los Teléfonos Móviles (Smartphone) y las Pocket PC, existe una demanda cada vez mayor de aplicaciones que puedan ejecutarse en varios dispositivos.

Existen varios conjuntos de herramientas y bibliotecas que se pueden utilizar para crear aplicaciones de gran riqueza para dispositivos móviles

inteligentes como los PDA, los Smartphone y las Pocket PC. No obstante, el enfoque aumentará en dos métodos principales: .NET Compact Framework, que está dirigido a Microsoft Pocket PC y a dispositivos impulsados por Windows CE, .NET y J2ME, el cual no está limitado a los sistemas operativos Microsoft.

2.4.1 Características de las Aplicaciones Móviles.

"El desarrollo de aplicaciones destinadas a dispositivos móviles, desde el punto de vista de la Ingeniería del Software, no debe diferir sustancialmente de los pasos a seguir cuando se construyen aplicaciones para ordenadores de escritorio o estaciones de trabajo. Así, se pueden establecer los siguientes pasos previos:

1. Análisis de Requerimientos.
2. Diseño de la Aplicación.
3. Implementación de la Aplicación."¹⁴

1. Análisis de Requerimientos. Se deben determinar las necesidades de los usuarios y los requerimientos de la aplicación. Por ejemplo, en el caso de un análisis para una aplicación que se ejecutará en un dispositivo móvil, algunos de estos requerimientos generales pueden ser la facilidad de uso, ejecución en diferentes dispositivos, conexión a entidades mayores para obtener o devolver datos, que almacene información de forma persistente, etc.

¹⁴ Internet, http://leo.ugr.es/~fjgc/INTRO/intro_9.htm

2. Diseño de la Aplicación. Es muy importante en este tipo de aplicaciones el crear programas separados por cada uno de los posibles usos que se le dé a la aplicación. De esta manera cada programa será más pequeño y se adaptará mucho mejor a las características de los dispositivos móviles.

En cuanto al diseño del interfaz de usuario, se debe decidir la correspondencia entre la aplicación y la pantalla. Se recomienda en esta etapa que se haga un "Story Board" conteniendo en cada viñeta los requerimientos para la pantalla correspondiente. En otra fase se decidirá qué tipo de controles vamos a utilizar para realizar entradas de datos y cómo vamos a presentar la información.

El almacenamiento persistente es un aspecto a tener en cuenta en nuestro diseño. Una primera opción es aquel formato que se emplea para enviar y recibir datos entre el dispositivo J2ME y el sistema externo, con esto evitamos una fase de conversión de formatos. Si el dispositivo posee sistemas de ficheros, entonces se puede optar por la creación de un fichero con una estructura más o menos compleja y usar las bibliotecas de Java para acceder a ellos. Otra alternativa puede ser emplear sistemas de gestión de bases de datos relacionales.

Finalmente, se debe tener en cuenta dentro del diseño, aspectos relacionados con la conectividad y con la entrada/salida, ya que son puntos importantes para determinar la portabilidad de la aplicación.

3. Implementación de la Aplicación. En esta etapa se hace la elección del lenguaje, plataforma y herramientas de desarrollo que se utilizarán para desarrollar la aplicación.

Los pasos a seguir en esta fase hasta instalar el programa en el dispositivo serán los siguientes:

- Escritura del código.
- Compilación de la aplicación.
- Eliminación de información de clases innecesaria. Esta etapa es opcional y en ella se renombran clases, métodos, interfaces, con objeto de hacerlo ambiguo. Un paquete obtenido de esta fase lo protege de la descompilación y de la ingeniería inversa. Además, reduce el tamaño de los ficheros de clase, dando lugar a ficheros JAR más pequeños.
- Ejecución del preverificador para añadir la información de "clase verificada" a los ficheros de clase.
- Empaquetamiento de la aplicación: creación del fichero JAR y JAD.
- Ejecución en un emulador apropiado.

- Instalación en el dispositivo y ejecución. En este caso existen dos modos de hacerlo: en el primero, se descargará la aplicación a través de una conexión de red, se cargará en memoria, se ejecutará la aplicación y finalmente se eliminará cualquier traza de ésta en el dispositivo; en la segunda, y siempre que el dispositivo lo permita, se instalará físicamente.

2.5 Tecnologías de Comunicación en Dispositivos Móviles.

Una de las características más importantes de un teléfono celular es su posibilidad de comunicarse con el exterior. Las capacidades y protocolos que actualmente pueden poseer los teléfonos de acuerdo con su marca y modelo son:

- Infrarrojo. La conexión por infrarrojo permite comunicarnos con otro dispositivo a través de un enlace visual entre ambos. Así podremos conectar nuestro teléfono con otro móvil con un Notebook, Palm o equipo de escritorio.
- Bluetooth. La tecnología Bluetooth (incluida en los teléfonos de altas prestaciones) nos permite conectarnos de manera inalámbrica con otros dispositivos que estén en un cierto alcance del mismo.

- Puertos Serial y USB. Esta es la forma más antigua de conexión de un equipo celular y es la única que no es inalámbrica. Permite conectarle un cable en la base del teléfono que podrá conectarse a nuestra PC a un puerto Serial o USB así como conectarnos a otros equipos como Palm.
- SMS. Forma de comunicarse que poseen dos equipos celulares a través de mensajes de texto SMS. Utilizando un mensaje como medio de transporte se puede enviar información de un equipo a otro por ejemplo, un juego multiplayer basado en turnos, en el que pueden jugar dos personas, como un juego de naipes, puede enviar sus jugadas a través de SMS y al ser recibidas por el otro equipo procesarlas por el mismo juego instaladas en el teléfono receptor.
- HTTP. Este protocolo nos permite comunicarnos, a Internet mediante un servidor web.
- TCP/UDP. Estos protocolos son de bajo nivel, lo que significa que deberemos programar más para lograr la conexión, pero tienen sus ventajas, se puede reducir la cantidad de información transmitida, se puede mantener el enlace abierto entre cliente y servidor lo que facilita la programación de juegos multiplayer y optimiza la información transmitida.

CAPÍTULO 3. LAS APLICACIONES MÓVILES.

En este capítulo abarcaremos todo lo referente a las Aplicaciones Móviles como son los Métodos de Comunicación, las características de las Tecnologías para desarrollar estas Aplicaciones y la Seguridad de las mismas.

3.1 Métodos de Comunicación.

"Los métodos a través de los cuales se transfieren las comunicaciones son usualmente invisibles para los usuarios. Las telecomunicaciones se iniciaron con el envío de señales telefónicas mediante cables sujetos por poleas. Si bien este método se sigue usando ampliamente, otros métodos de transmisión se han vuelto comunes hoy en día, como los cables de fibra óptica y las señales de radio y microondas enviadas por satélite o transmisores terrestres."¹⁵

En la mayoría de las telecomunicaciones se utilizan cables físicos. Se pueden usar distintos tipos de cables dependiendo de la clase y disponibilidad de la información transmitida. Entre mayor sea el número de bytes por segundo que un cable pueda transferir, más rápida será la transmisión de la información. Los cables telefónicos ordinarios no suelen ser muy rápidos para la transmisión de grandes volúmenes de información. Normalmente se utilizan cables especializados de alta velocidad cuando este tributo es importante. Normalmente se utilizan las líneas de redes digitales de servicios integrados

¹⁵ *Internet*. <http://www.aceproject.org/main/espanol/et/etf01b.htm>

(ISDN por sus siglas en inglés) para transmitir grandes volúmenes de información a altas velocidades. Los cables de fibra óptica también se pueden utilizar para transmisiones a gran velocidad. Los cables ópticos transmiten información utilizando pulsaciones de luz láser.

En algunos casos, especialmente en sitios remotos, los cables físicos pueden no ser apropiados o no estar disponibles. En estos casos, la información puede ser transmitida por microondas u ondas de radio, utilizando satélites o transmisores terrestres. Todos estos medios de telecomunicación son complejos y están siendo constantemente mejorados.

3.1.1 Internet como nuevo "Canal de Comunicación".

Internet es un 'canal' a través del cual puede transitar el tráfico de información de los medios de comunicación ya existentes. Internet este 'nuevo canal' tiene algunas peculiaridades extraordinarias:

- Es un canal universal, es decir, un canal que soporta sin dificultad el tráfico de todos los medios de comunicación.
- Es un canal omnifuncional, capaz de desempeñar funciones 'conectoras' (comunicaciones de uno a uno), funciones 'distribuidoras' (de uno a muchos) y funciones 'colectoras' (de muchos a uno); por lo tanto es un canal que se puede personalizar.

- Es un canal bidireccional y por consiguiente interactivo; no sólo que admite la interactividad, sino que en su funcionamiento, la facilita.
- Es un canal de alcance prácticamente ilimitado, cuyo ámbito es mundial.

Este nuevo canal presenta grandes ventajas que hacen posible un desarrollo de esos medios impensable con las limitaciones de los canales tradicionales inconvenientes pueden ser mitigados o anulados por mejoras en la tecnología del canal, que deberán producirse durante los próximos años.

1. Una característica de Internet, de gran trascendencia, radica en el hecho de que la comunicación de esta, aunque es masiva en el sentido de que puede llegar y llega a muchos es siempre comunicación de uno a uno. El equívoco radica en que con frecuencia tiende a imaginarse al servidor de información como un 'emisor' de prensa, radio o televisión que difunde en cada uno de sus actos de comunicación un mensaje dado a una audiencia masiva. Pero las cosas en Internet no son así, sino que el servidor de esta es capaz de 'atender' a muchos usuarios a la vez.

2. Otra característica de Internet se encuentra en la posición activa del 'usuario' de los servicios que se ofrecen en ésta. En el comportamiento típico de Internet el usuario decide cada paso y está decidiendo en cada momento si continúa recibiendo una información y en qué condiciones y con qué características la recibe.

Esta 'interactividad', que actualmente emerge también en la audiencia televisiva es en el caso de Internet prácticamente obligada (al menos hasta ahora). Esto representa una exigencia adicional impuesta al usuario, la actitud pasiva típica de las audiencias tradicionales de radio y televisión no es posible en Internet.

3. Constituye un canal de potencia y alcance prácticamente ilimitados. Por las características técnicas de este canal, Internet es un medio masivo, de alcance universal. Hay actualmente, limitaciones muy importantes, que sufren los usuarios de Internet. Pero son técnicas y comerciales que el desarrollo tecnológico permite vencer con crecimientos exponenciales en el rendimiento de los recursos. Otras más radicales derivan de las condiciones económicas y culturales

3.1.2 Internet como canal (secundario) de Prensa escrita.

En Diciembre de 1994, primer periódico electrónico en la Web:

Los inconvenientes de la prensa electrónica derivan de las características físicas del 'receptor' de la información y su conexión necesaria a la red telefónica: incomodidad de la lectura en pantalla, falta de movilidad, etc. Otro es la reducción de la audiencia, restringida necesariamente a las personas con conexión a Internet.

Las ventajas:

- La facilidad e inmediatez de la recuperación de la información.
- La posibilidad de una reactualización, corrección, ampliación.
- La facilidad, para el usuario, de edición, copia, archivado.
- Acceso a los números atrasados.
- Infinitas consultas y recuperaciones posibles.

En la segunda perspectiva:

- Remisión ilimitada a otras informaciones.
- Posibilidades abiertas por la interactividad.
- Posibilidad de difundir localmente información de interés local.
- La posibilidad de personalización.

3.1.3 Internet como canal (secundario) de Radio.

En Abril de 1995 aparece el primer software de Real Audio. Los inconvenientes de la radio 'on-line' son la incomodidad del 'receptor' y la limitación de la audiencia.

Las ventajas:

- La desaparición de los problemas de frecuencia, derivados de la limitación de banda disponible.
- La desaparición de las limitaciones de alcance y los problemas asociados de potencia.

Desarrollo de los servicios, posibilitado por el nuevo canal:

- Posibilidad de 'emitir' simultáneamente múltiples canales y programas.
- Posibilidad de acceso a fondos de fonoteca.
- Posibilidades derivadas de la interactividad: de los programas 'abiertos al público' tradicionales a los 'foros', la participación en línea.
- Posibilidad de personalización.

3.1.4 Internet como canal (secundario) de TV.

Es la TV digital por cable limitada por la infraestructura de la Red Telefónica Básica que para desempeñar la función de canal secundario de TV, Internet tiene que transmitir las señales video en tiempo real, transmisión dificultada por la escasa capacidad de las líneas.

3.1.5 Internet como nuevo "Medio de Comunicación".

Las posibilidades de Internet no se reducen a su condición de nuevo canal para los medios de comunicación tradicionales, sino que convierten a Internet en un nuevo 'medio de comunicación', cualitativamente distinto de los medios existentes. Sin dejar de seguir siendo un nuevo canal va creando un nuevo lenguaje, complejo, cuyo desarrollo está convirtiendo Internet en un nuevo medio de comunicación, tan característico y diferenciado de los demás.

Las principales posibilidades de este 'nuevo medio' son:

- Ser multilinguaje, capaz de utilizar simultánea y articuladamente los lenguajes propios de todos los medios.
- Ser hipertextual, capaz de niveles y ramificaciones de referencias en número indefinido.
- Ser personalizable, capaz de ofrecer las informaciones que cada usuario individualmente demande, en el volumen que pida y con el formato que prefiera.
- El nuevo medio puede acceder desde sus marcos a todos los demás medios, seleccionando sus materiales e integrándolos en su propia oferta informativa.

Como se ha visto Internet se ha convertido en el método de Comunicación más usado y como los dispositivos móviles también han implementado el uso de este método surge la idea de crear aplicaciones que exploten al máximo esta herramienta.

3.2 Necesidades de las Aplicaciones Móviles.

Actualmente ya existe una amplia oferta de dispositivos móviles en el mercado y los fabricantes distribuyen nuevos modelos constantemente. Como resultado de esta gran variedad de dispositivos se deben afrontar los siguientes problemas cada vez que se desarrolla una aplicación Web móvil:

- Diferentes lenguajes de marcado. Incluidos HTML, HTML compacto (cHTML) para los teléfonos i-mode japoneses y WML para los teléfonos WAP (protocolo de aplicaciones inalámbricas).
- Diferentes factores de forma. Incluidos los diversos tamaños de pantalla, su orientación (horizontal o vertical) y los tipos de pantalla en color, escala de grises o blanco y negro. Estas variables afectan a la paginación del contenido y al tipo de gráficos que se generan.
- Diferentes capacidades de los dispositivos. Si pueden mostrar imágenes, establecer una llamada telefónica o recibir mensajes de notificación.
- La administración de estados. Incluida la compatibilidad con las cookies.

Hasta ahora, los desarrolladores que han querido crear aplicaciones que funcionen a lo largo de una amplia variedad de dispositivos estuvieron limitados por los ambientes de desarrollo disponibles para soluciones del más bajo denominador común, que sólo permiten a los usuarios aprovechar las funciones más simples disponibles en todos los dispositivos, incluyendo los más delgados o menos capaces. Mientras esto proporciona a las aplicaciones un largo alcance en los dispositivos, no les permite, a ellos ni a los usuarios, usar la capacidad de cualquier dispositivo en particular. El resultado es una experiencia de bajo rendimiento para el usuario.

La visión de movilidad de .NET, y los productos y servicios que lo soportan, es lo suficientemente flexible para satisfacer prácticamente cualquier

necesidad de desarrollo móvil. Los desarrolladores y sus usuarios obtienen tanto la amplia cobertura del dispositivo como un soporte de dispositivo enriquecido. Los elementos de infraestructura necesaria, tales como seguridad y capacidad de ampliación, están integrados profundamente en la plataforma, proporcionando a los desarrolladores una plataforma confiable sobre la cual crear.

3.3 Características de la Tecnología Microsoft.NET.

Existen varios conjuntos de herramientas y bibliotecas para crear aplicaciones de gran riqueza para dispositivos móviles inteligentes como las Pocket PC y los Teléfonos Móviles. No obstante, el enfoque aumentará en dos métodos principales: .NET Compact Framework, que está dirigido a Microsoft Pocket PC y a dispositivos impulsados por Windows CE .NET, y J2ME, el cual no está limitado a los sistemas operativos Microsoft.

3.3.1 Herramientas de desarrollo de Aplicaciones.

Visual Studio .NET 2003 es la última versión del entorno y el juego de herramientas de desarrollo estándares de Microsoft. Incorpora la compatibilidad con aplicaciones en desarrollo para dispositivos móviles inteligentes que utilizan .NET Compact Framework. Con Visual Studio .NET 2003, millones de desarrolladores de aplicaciones de escritorio existentes obtienen un entorno de desarrollo uniforme a través de escritorio, PDA y otros dispositivos de bolsillo.

Esto hace que el desarrollo de aplicaciones sea más fácil y menos costoso, permite que se vuelva a emplear el código y como consiguiente, faculta la producción de aplicaciones más robustas. De igual manera, impulsa el desarrollo de aplicaciones de "informática de conexión ocasional", las cuales se ajustan dinámicamente a la disponibilidad de una conexión inalámbrica y logran trabajar fuera de línea de forma útil hasta que se establece la conexión de nuevo.

Con la primera versión de las herramientas de desarrollo móvil de .NET Compact Framework y Visual Studio .NET se pueden crear aplicaciones en Visual Basic .NET o C#; es probable que una versión posterior admita otros lenguajes .NET de Windows. Para las aplicaciones que necesitan una base de datos, SQL Server CE, la cual es una base de datos relacional compacta, se ejecuta en dispositivos CE de Windows CE y es compatible con Windows SQL Server.

Las aplicaciones de .NET Compact Framework se pueden depurar mediante un emulador que se ejecuta en la máquina de desarrollo o bien, en un dispositivo físico conectado a la máquina de desarrollo a través de un puerto serie (o USB) o una conexión de red.

Si se requiere el uso de aplicaciones basadas en web, Microsoft ASP.NET Mobile Controls (anteriormente Mobile Internet Toolkit), que ahora es parte de

Visual Studio .NET 2003, puede utilizarse para desarrollar aplicaciones web móviles, las cuales adaptan la presentación de página a una gama de dispositivos, tales como micronavegadores en PDA, teléfonos inteligentes y teléfonos WAP.

3.3.2 .NET Framework.

“.NET Framework ofrece un modo robusto y eficaz de diseño de aplicaciones Windows para PC. Este método reemplaza los anteriores, en los que los API de los sistemas operativos Windows (Win32) se accedían directa o indirectamente a través de bibliotecas específicas al lenguaje utilizadas por desarrolladores de Visual Basic o Visual C++.”¹⁶

Existe un gran número de ventajas cruciales en el uso de .NET Framework:

- Provee compatibilidad completa para el diseño de suntuosas aplicaciones cliente o basadas en web, al igual que servicios web XML.
- Ofrece un conjunto de estándares de interfaces u operaciones de programación a través de los cerca de 20 lenguajes admitidos. No obstante, en la práctica la mayoría de las nuevas aplicaciones .NET se escriben ya sea en Visual Basic .NET* o C#. Visual Basic .NET está basado en Visual Basic 6 con algunas mejoras importantes como la

¹⁶ Internet. <http://www.intel.com/espanol/update/contents/mo11031.htm>

compatibilidad completa con la programación orientada a objetos y algunos aspectos inevitables de la compatibilidad con versiones anteriores. C#, al igual que Java, está basado en C++ y es bastante similar a Java.

- .NET Framework incluye un entorno común de tiempo de ejecución que verifica el código de las aplicaciones a medida se cargan y las administra durante su ejecución, controlando los errores, la recopilación de residuos y la seguridad, entre otros.
- Las aplicaciones se compilan en un lenguaje intermedio y se convierten a un código de máquina ejecutable justo antes de la ejecución por parte del entorno de tiempo de ejecución. En potencia, esto significa que el tiempo de ejecución de .NET Framework se puede transportar a otros sistemas operativos y plataformas de hardware sin tener que volver a escribir o a compilar las aplicaciones existentes.

3.3.3 .NET Compact Framework.

“.NET Compact Framework es un subconjunto de .NET Framework. Ofrece las ventajas de un entorno robusto y seguro para la ejecución de código cliente y servicios web XML para dispositivos inteligentes, tales como Pocket PC, Teléfonos Móviles y cajas de conexión.”¹⁷

¹⁷ *Idem.*

De forma específica es compatible con:

- Pocket PC (y Pocket PC 2002 Phone Edition, para los usuarios que desean hacer llamadas telefónicas con PDA), el cual en la actualidad se basa en el sistema operativo Windows CE 3.0.
- Las soluciones integradas que se ejecutan en Windows CE .NET, la cual es la última versión del sistema operativo de Microsoft para dispositivos móviles inteligentes.
- Se espera que sea compatible en un futuro cercano con Microsoft Smartphone 2002, el cual se utiliza principalmente para llamadas de voz pero también ofrece la mayoría de las funciones de PDA.

.NET Compact Framework (al igual que la versión completa de .NET Framework) es una abstracción de las funciones subyacentes del sistema operativo y puede satisfacer una gran mayoría de los requisitos de las aplicaciones a través de su funcionalidad estándar. No obstante, si se requiere el acceso a una función única de dispositivo, se puede admitir ésta con facilidad debido a que .NET Compact Framework está diseñado para ampliarse.

3.3.4 .NET Mobile Internet Toolkit.

Desarrollado en .NET Framework y su ambiente de desarrollo de aplicación Web ASP.NET, Mobile Internet Toolkit es una colección de controles

del lado del servidor para crear aplicaciones Web móviles. Cada uno de estos controles automáticamente percibe las funciones del dispositivo conectado y emite el lenguaje de marcación correcto, permitiendo a los desarrolladores crear una sola aplicación Web móvil que puede de manera automática servir a cualquier cantidad de dispositivos, desde teléfonos móviles basados en WML (WAP) hasta teléfonos móviles basados en cHTML (Imode), hasta HTML para PDAs, tales como la Pocket PC. Mobile Internet Designer, que permite a los desarrolladores arrastrar y soltar controles móviles y así crear Web Forms móviles, se integra en Microsoft Visual Studio.NET, dando a los desarrolladores del Web móviles acceso a las herramientas de desarrollo de aplicación más poderosas de la industria. Mobile Internet Toolkit también da soporte de capacidad de ampliación para nuevos dispositivos y controles.

3.3.5 Seguridad en Microsoft .NET.

En este tema describiremos las diferentes características y mecanismos de seguridad que pueden utilizar los desarrolladores y fabricantes de dispositivos para crear aplicaciones y dispositivos seguros basados en Windows CE .NET.

3.3.5.1 Creación de un entorno de confianza.

Los dispositivos de Microsoft Windows CE envían, reciben y procesan información que se debe proteger de aplicaciones potencialmente inseguras. Para proteger su dispositivo, puede implementar medidas de seguridad que eviten que el sistema operativo cargue módulos desconocidos, restringir el acceso a interfaces de programación de aplicaciones (API) del sistema e impedir el acceso de escritura a secciones del registro del mismo. A la hora de certificar aplicaciones, podrá designar a un módulo como de confianza o no confianza. El núcleo puede utilizar esta información para evitar que se carguen aplicaciones no autorizadas o limitar su acceso al sistema.

Para crear un entorno de confianza, deberá implementar las siguientes funciones:

- OEMCertifyModuleInit. Permite que el cargador del sistema operativo notifique al OEM (fabricante del equipo original) que se está cargando un nuevo módulo. Permite que el OEM decida si desea comprobarlo por motivos de seguridad.
- OEMCertifyModule. Permite que el cargador del sistema operativo pase el código del módulo (por ejemplo, DLL, EXE y OCX) al OEM para comprobar que la ejecución en el sistema es segura.

Antes de que el núcleo cargue una aplicación, la función OEMCertifyModule comprueba la firma de ésta para proteger el sistema contra aplicaciones desconocidas. De este modo se garantiza que la plataforma basada en Windows CE .NET carga una aplicación sólo si contiene una firma digital válida.

3.3.5.1.1 Creación de una firma.

Para crear una firma digital desde un archivo, aplique una función hash al archivo y, a continuación, firme el hash resultante con una clave privada. Un modo sencillo de crear una firma digital desde un archivo consiste en utilizar el archivo Signfile.exe, que se incluye en Microsoft Platform Builder. Este archivo es una herramienta para la firma de archivos ejecutables con una clave privada proporcionada por un proveedor de servicios de cifrado (CSP).

3.3.5.1.2 Comprobación de una firma.

Con el método OEMCertifyModule, para calcular una firma durante el proceso de comprobación, el núcleo utiliza la misma fórmula de hash empleada durante el proceso de generación de la firma. OEMCertifyModule compara la firma calculada a partir del hash con la firma incluida en el archivo. Si las firmas no coinciden, OEMCertifyModule no permitirá que se cargue el archivo.

3.3.5.2 Seguridad del almacén de objetos.

El almacén de objetos ofrece varios elementos de seguridad en un entorno de confianza. Los archivos de sistema se encuentran protegidos para evitar que aplicaciones que no sean de confianza puedan leerlos o modificarlos. Este tipo de archivos tienen definido el atributo de sistema.

Asimismo, el sistema protege un conjunto de claves de registro para evitar que las modifiquen aplicaciones que no sean de confianza. Aunque todas las aplicaciones pueden leer las claves y valores del registro, sólo aquellas de confianza pueden modificar los valores o las subclaves que se encuentran bajo claves protegidas. El sistema protege un conjunto básico de claves, que puede ser ampliado por el OEM.

3.3.5.3 Seguridad de bases de datos.

Windows CE .NET permite que las aplicaciones de confianza incluyan un indicador de sistema en las bases de datos para así denegar el acceso a llamadores que no sean de confianza. Las aplicaciones que no son de confianza no pueden abrir, leer ni modificar las bases de datos que incluyen un indicador de sistema. Para proteger una base de datos, los llamadores de confianza pueden establecer el indicador CEDB_SYSTEMDB dentro de la estructura CEDBASEINFOEX pasada a CeCreateDatabaseEx2 o CeSetDatabaseInfoEx2.

3.3.5.4 Interfaces API de confianza.

Además de las funciones del OEM, las API CeGetCurrentTrust y CeGetCallerTrust permiten que una biblioteca DLL consulte el nivel de confianza de una aplicación que llama. Pueden utilizarse estas funciones para comprobar los niveles de confianza de las aplicaciones.

3.3.5.5 Uso de SSL para la comunicación de red segura.

Para la comunicación de red segura, Windows CE .NET admite los protocolos de seguridad de nivel de socket seguro SSL 2.0, SSL 3.0 y SSL 3.1, que se encuentran disponibles a través de WinInet o directamente en WinSock. La versión SSL 3.1 se denomina también Seguridad de la capa de transporte (TLS). Estas aplicaciones utilizan sockets seguros para enviar y recibir datos codificados a través de las líneas de comunicación.

La implementación de sockets seguros se basa en la autenticación para determinar si un host remoto es de confianza. Los hosts remotos establecen el grado de confianza mediante la obtención de un certificado de una entidad emisora de certificados (CA). Esta entidad puede, a su vez, recibir la certificación de otra superior y así sucesivamente, con lo que se crea una cadena de confianza. Para determinar si un certificado es de confianza, la aplicación debe determinar la identidad de la CA raíz y, a continuación, decidir si es segura.

SSL basado en Windows CE .NET utiliza entidades emisoras de certificados de confianza. Cuando una aplicación intenta realizar una conexión segura, SSL extrae el certificado raíz de la cadena de certificaciones y lo comprueba con la base de datos de CA. Envía el certificado del servidor a la aplicación a través de una función de devolución de llamada para la validación de certificados, junto con los resultados de la comparación con las bases de datos de CA.

3.3.5.6 Cifrado de datos con CryptoAPI.

CryptoAPI proporciona servicios que permiten agregar a las aplicaciones esquemas de cifrado y descifrado de datos, autenticar con el uso de certificados digitales y codificar o descodificar hacia y desde ASN.1 en las aplicaciones basadas en Microsoft Win32. Se pueden utilizar las funciones de CryptoAPI sin necesidad de poseer conocimientos profundos de la implementación subyacente. CryptoAPI funciona con determinados proveedores de servicios de cifrado (CSP) que realizan las funciones criptográficas reales, como el cifrado, descifrado y la seguridad y almacenamiento de claves.

Los tres elementos del sistema de cifrado de Microsoft son el sistema operativo, la aplicación y el CSP. Las aplicaciones se comunican con el sistema operativo a través de la capa CryptoAPI y el sistema operativo se comunica con los CSP a través de la interfaz de proveedor de servicios de cifrado (CSPI).

Los CSP son módulos independientes, normalmente bibliotecas DLL que contienen algoritmos y realizan todas las operaciones de cifrado. En teoría, los CSP se escriben para ser independientes de una aplicación concreta, de modo que cualquier aplicación se pueda ejecutar con varios CSP.

Los CSP predefinidos incluidos en Windows CE .NET son:

- RSA Base Provider. Admite la firma digital y el cifrado de datos. Se considera una herramienta de cifrado para fines generales.
- RSA Enhanced Provider. Admite el cifrado de clave de 128 bits. Proporciona mayor seguridad a través de claves más largas y algoritmos adicionales.
- Smart Card CSP. Admite el uso de tarjetas inteligentes para Windows. Se puede encontrar un código fuente de CSP de tarjeta inteligente de ejemplo en el directorio.

3.3.5.7 Uso de la API de almacén protegido.

Para proteger la información confidencial y para evitar la manipulación de datos, la API de almacén protegido proporciona una práctica solución para los problemas de cifrado, administración de claves y experiencias de usuario. Las dos funciones de CryptoAPI, CryptProtectData y CryptUnprotectData, toman las credenciales de inicio de sesión del usuario para bloquear y desbloquear los datos privados.

Las ventajas que ofrece el almacén protegido son las siguientes:

- Una aplicación fácil de usar que toma los datos y las contraseñas opcionales, u otros valores de entropía, y recibe datos ocultos.
- Los datos se encuentran protegidos frente a otros usuarios con capacidad de inicio de sesión en el mismo dispositivo.
- Los datos están protegidos frente a manipulaciones mientras el dispositivo está desconectado.
- El uso transparente de las credenciales de inicio de sesión a fin de proporcionar la entropía necesaria para la protección de datos.
- Extensibilidad de OEM que permite utilizar tokens de hardware como las tarjetas inteligentes o los dispositivos biométricos.

3.3.5.8 Generación segura de datos aleatorios.

Se utiliza CryptGenRandom generar datos aleatorios. Esta función presenta dos de las propiedades de un buen generador de números aleatorios: no es previsible y presenta una distribución uniforme de los valores.

3.3.5.9 Aislamiento de datos confidenciales en una tarjeta inteligente.

Puede agregar una capa de seguridad a un dispositivo de Windows CE .NET mediante el uso de tarjetas inteligentes que permiten almacenar

información de autenticación o un mecanismo de firma digital. Puede escribir un proveedor de CryptoAPI personalizado que aproveche la capacidad de la tarjeta inteligente de almacenar información de forma segura.

El subsistema de tarjeta inteligente de Windows CE .NET admite CryptoAPI a través de proveedores de servicios de tarjeta inteligente (SCSP), que son bibliotecas DLL que permiten el acceso a servicios específicos. Este subsistema proporciona un vínculo entre el hardware lector de la tarjeta inteligente y las aplicaciones.

Windows CE .NET no proporciona proveedores SCSP; generalmente, es el proveedor de tarjetas inteligentes el que proporciona los SCSP correctos. Sin embargo, Windows CE .NET ofrece las siguientes interfaces:

- Administrador de recursos (Scard.dll). Utiliza las API de Win32 para administrar el acceso a varios lectores y tarjetas inteligentes.
- Biblioteca auxiliar del administrador de recursos (Winscard.dll). Expone servicios PC/SC para el uso de tarjetas inteligentes y lectores de tarjetas inteligentes.
- Biblioteca auxiliar del lector de tarjetas inteligentes (Smclib.lib). Proporciona, según sea necesario, las rutinas de compatibilidad habituales para el controlador de tarjetas inteligentes y compatibilidad adicional con los protocolos T=0 y T=1 para controladores específicos.

Controladores de lectores de tarjetas inteligentes de ejemplo (Pscr.dllbulletip3.dllstcusb.dll). Controlador de lector de PC SwapSmart. Controlador de lector en serie. Controlador de lector de bus serie universal (USB).

Esta es la propuesta de Microsoft para el desarrollo de Aplicaciones Móviles pero como sabemos el costo siempre es elevado, así que para nuestro proyecto nos inclinamos por la propuesta de SUN Microsystems que es Free (gratis), su nombre es Java 2 MicroEdition. A continuación se mencionan sus características.

3.4 Características de la Tecnología Java 2 ME.

Para Java existen tres ediciones:

- La plataforma Java 2 Standard Edition (J2SE) para el diseño de aplicaciones suntuosas de escritorio o basadas en web.
- La plataforma Java 2 Enterprise Edition (J2EE), un superconjunto de J2SE, para el diseño de componentes de transacción escalables y basados en servidor: Enterprise Java Beans (EJB).
- La plataforma Java 2 Micro Edition (J2ME), un subconjunto de J2SE, para el diseño de aplicaciones integradas y aplicaciones para dispositivos móviles inteligentes.

Éstas solamente son especificaciones, los proveedores de software deben producir las implementaciones reales. Cada especificación define un conjunto de estándares para interfaces o bibliotecas que puedan utilizar las aplicaciones Java. Las aplicaciones Java resultantes se compilan en un lenguaje de código de byte intermedio.

El entorno de tiempo de ejecución, Java Virtual Machine (JVM), carga y ejecuta la aplicación para convertir el código de byte en código de máquina ejecutable. JVM tiene la responsabilidad de administrar las aplicaciones a medida que se ejecutan. Existen varios paralelos a .NET Framework.

La plataforma Java 2 Micro Edition (J2ME) está disponible en dos configuraciones principales que incorporan una máquina virtual (VM) y API núcleos:

La primera es Connect Limited Device Configuration (CLDC) para dispositivos de bajo nivel y con recursos y conectividad limitados. Existen dos opciones llamadas perfiles:

- Mobile Information Device Profile (CLDC-MIDP) se utiliza ampliamente en cientos de millones de teléfonos en la actualidad.
- Personal Digital Assistant Profile (CLDC-PDAP) está diseñado para PDA futuros de bajo nivel que funcionan principalmente como PIM.

La segunda configuración es Connected Device Configuration (CDC), la cual es relativamente nueva. Está diseñada para dispositivos nuevos más sofisticados, entre los que se incluyen los dispositivos PDA. Existen tres perfiles que se basan entre sí y que se relacionan principalmente con las crecientes capacidades de la interfaz de usuario. El más sofisticado es Personal Profile (CDC-PP), el cual tiene capacidades equivalentes a las de J2SE. También es la competencia natural de .NET Compact Framework.

También existe una gran cantidad de paquetes opcionales que amplían los perfiles; éstos incluyen Wireless Messaging API, Mobile Media API, J2ME RMI Optional Package y el paquete opcional JDBC para CDC Foundation Profile, al igual que otros que aún están en el proceso de especificación, tal como J2ME Web Services.

3.4.1 Entorno de desarrollo J2ME Wireless Toolkit.

J2ME Wireless Toolkit no es propiamente un entorno de desarrollo sino más bien un entorno de compilación, preverificación y ejecución de los MIDlets creados. Es una herramienta que completa el entorno de desarrollo basado en línea de comandos al permitir realizar ciertas tareas desde su interfaz de usuario, la más importante, ejecutar la aplicación en uno de los emuladores que posee para esta tarea. Sin embargo, carece de un editor integrado que permita codificar directamente en él.

Un aspecto muy importante de este entorno es que se puede integrar con otros entornos de desarrollo (NetBeans y Sun One Mobile).

3.4.2 Entorno de desarrollo Sun One Mobile Edition.

Son dos las versiones que existen de este entorno de trabajo creado por SUN y basado en la plataforma NetBeans por lo tanto es muy similar a éste.

Realizar aplicaciones desde Sun One Mobile es muy fácil y, gracias a las plantillas incorporadas al entorno, podemos crear desde clases simples hasta suites.

3.4.3 Entorno de desarrollo JBuilder X.

JBuilder X de Borland Mobile Edition es la plataforma que Borland ha desarrollado para dar soporte a J2ME y es, junto con el SUN Java Studio Mobility 6 Early Access, lo que entraría a formar parte de la serie de productos no free.

Las posibilidades que tiene esta herramienta en cuanto al desarrollo son infinitas y vas desde el editor integrado de código, el depurador, las opciones de optimización, hasta las posibilidades de integración con Together y el acceso a todas sus opciones

3.4.4 Entorno de desarrollo NetBeans.

Probablemente NetBeans sea uno de los proyectos Open Source con más proyección de futuro, es un entorno que ha sabido adaptarse a las necesidades de la comunidad internacional. Existen dos tipos de herramientas. Lo que es el entorno de desarrollo (IDE) y lo que se conoce como plataforma NetBeans para la creación de aplicaciones de escritorio.

Las ventajas que aporta NetBeans son muchas, la más importante es disponer de un autentico entorno de desarrollo y ejecución de aplicaciones, un editor de código muy completo, herramientas automáticas para la actualización y multitud de módulos que abarcan la casi totalidad de las áreas disponibles en Java (desde web services hasta herramientas de testing y rendimiento).

3.4.5 Entorno de desarrollo Nokia Developer Suite.

Un aspecto importante es que para poder instalar la aplicación hay que estar dados de alta en el forum de Nokia para que envíen la clave de activación, otro aspecto interesante es que puede instalarse de tres maneras distintas: como programa independiente (standalone), integrada con JBuilder o integrada con Sun One Studio.

3.4.6 Seguridad en J2ME.

La seguridad ha pasado a considerarse un factor cada vez más importante en el mundo de los dispositivos conectados inteligentemente, J2ME no es la excepción por lo que aporta las siguientes características.

3.4.6.1 JSR-177 (Security and Trust Services API for J2ME).

Este JSR define un conjunto de APIs cuyo propósito es ofrecer servicios de seguridad para aplicaciones J2ME.

Aspectos claves son el soporte de certificados digitales, plataformas de pago, criptografía en las comunicaciones, asegurar la integridad de los datos, proveer servicios de credenciales y de información personal, pago por medios digitales, etc.

3.4.6.2 MIDP 2.0

MIDP 2.0 mejora de manera considerable el modelo de seguridad del MIDP 1.0 a través del paquete `javax.microedition.pki`. Este paquete provee el manejo de certificados que son usados para la autenticación de la información en conexiones seguras.

En MIDP 2.0 es posible realizar firmado de aplicaciones y manejar dominios privilegiados. Con el firmado de aplicaciones podemos comprobar la identidad del proveedor de la aplicación y por lo tanto confiar en la misma. A través de los privilegios de dominio los proveedores de aplicaciones y los vendedores de dispositivos, pueden definir qué APIs son consideradas como restringidas. Estas nuevas características protegen al dispositivo frente accesos no autorizados por parte de las aplicaciones. Los protocolos seguros estándar como HTTPS, TLS/SSL y WTLS permiten la transmisión segura de datos a través de la encriptación de los mismos.

3.4.6.3 Modelo Sandbox de seguridad.

J2ME cuenta con su propio modelo Sandbox de seguridad. Este modelo establece que un applet sólo puede ejecutar ciertas operaciones que se consideran seguras. Hay otra serie de operaciones que están fuera del Sandbox y que no pueden ser ejecutadas por el applet.

La filosofía de este modelo de seguridad es dotar a los desarrolladores de grandes posibilidades para hacer aplicaciones potentes, pero por otra parte minimizar los riesgos que supone un dispositivo que ejecuta aplicaciones descargadas de la red. Esto define las líneas maestras del modelo Sandbox de seguridad:

- Los ficheros de clases Java deben ser verificados como aplicaciones Java válidas.
- Sólo se permite el uso de APIs autorizadas por CLDC.
- No está permitido cargar clases definidas por el usuario.
- Sólo características nativas que entren dentro del CLDC pueden ser accedidas.

3.5 Características de Java Servlets y JSP.

Los Servlets y las páginas JSP están relacionadas por lo tanto lo que se puede hacer con una tecnología es también posible con la otra; aunque cada una tiene capacidades propias.

3.5.1 Características de Java Servlets.

Los Servlets son programas en java que deben ser precompilados y puestos en un servidor de páginas o servidor web el cual los carga, compila y ejecuta en el momento que un usuario remoto lo solicita a través de una página html y/o wml.

Cuando el servidor web carga y ejecuta un servlet, lo mantiene activo en memoria en su propio mientras no se aplique reset al servidor.

Por lo tanto, los servlets nos permiten crear paginas web activas en el sentido de que la respuesta que ofrecen puede variar en funcion de los datos que proporcione el cliente y la situación de contexto existente.

Las características principales de los Servlets son:

- Al estar escritos en java son independientes de la plataforma.
- Consumen menos recursos por que solo son cargados la primera vez que se solicitan sus servicios. Las siguientes peticiones crean hilos de ejecución.
- Son portables debido a que se ejecutan bajo la maquina virtual de java y el administrador de seguridad de java (Java Security Manager).
- No requieren soporte para Java en el explorador del cliente, ya que operan en el dominio del servidor y envían el resultado en HTML y/o wml.

3.5.2 Funcionamiento de los Servlets.

El funcionamiento de los servlets sigue los estos pasos:

1. "El cliente realiza una petición http haciendo uso de un navegador, en esta petición pueden existir parámetros con sus valores.
2. La petición y sus datos asociados los recoge el servidor web.

3. El servidor Web detecta que hay que ejecutar un servlet y delega esta acción al contenedor de servlets, el cual se encarga de llevarla a cabo.
4. Una vez ejecutado el servlet, se traspasan los resultados (en forma de pagina web) al servidor web.
5. El servidor web envía la pagina web de resultado a través del ordenador servidor.
6. La pagina web de respuesta le llega al cliente, que la visualiza haciendo uso del navegador."¹⁸

3.5.3 Puesta en Funcionamiento de los Servlets.

Desarrollar y ejecutar por primera vez un servlet no resulta tan sencillo como un programa de java basado en SDK estándar (J2SE), primero es necesario instalar una versión de java 2 Enterprise Edition (J2EE) donde se encuentran los paquetes `javax.servlet` y `javax.servlet.http`. El primero de ellos contiene las interfaces y clases mas genéricas que soportan el funcionamiento de los servlets.

Asiando referencia a los apartados anteriores, para ejecutar un servlet necesitamos un "contenedor de servlets".

En resumen para poder compilar y ejecutar un servlet es necesario realizar las siguientes acciones:

¹⁸ Hanna. Phil, Manual de Referencia JSP, México 2002, pag. 94

- Instalar y configurar una versión de Java 2 Enterprise Edition (J2EE).
- Instalar y configurar algún contenedor de servlets.

3.5.4 Características de Java JSP.

Las paginas JSP (Java Server Pages) es una tecnología basada en scripts que utilizan una variante del lenguaje java para generar páginas web de forma dinámica en el servidor.

Las paginas JSP permiten generar dinámicamente HTML, WML, XML o algún otro tipo de página web. En las jsp, se escribe el texto que va a ser devuelto en la salida (HTML, WML, XML) intercalado con el código java dentro de él para poder modificar o generar contenido dinámicamente.

Las páginas JSP se ejecutan en un componente del servidor conocido como contenedor de JSP que las traduce a Servlets Java equivalentes.

Como son servlets, las JSP tiene todas las ventajas de los servlets:

- Tiene un mayor rendimiento y capacidad de adaptación (escalabilidad).
- No es necesaria una configuración especial por parte del cliente.
- Incorporan soporte para http, lo que hace posible la programación de aplicaciones.

- Tiene pleno acceso a la tecnología Java.
- Se vuelven a compilar automáticamente cuando es necesario.
- Como las paginas JSP son similares al html, tienen mayor compatibilidad con las herramientas de desarrollo web.

3.5.5 Funcionamiento de los JSP.

Las paginas JSP pasan por tres etapas:

1. Código fuente JSP. Este código es que realmente escribe el desarrollador. Se encuentra en un archivo de texto con extensión .jsp y consiste en un mezcla de código de plantilla html, instrucciones en lenguaje java, directivas JSP y acciones que describen como generar una pagina web para dar servicio a una petición concreta.
2. Código fuente Java. El contenedor de JSP traduce el código fuente JSP al código fuente de un servlet java equivalente. Este código fuente se guarda en un área de trabajo y suele ser útil en el proceso de depuración de errores.
3. Clase Java compilada. Como cualquier otra clase Java el código servlet generado se compila en código de bytes en un archivo .class preparado para ser cargado y ejecutado.

El contenedor de JSP administra cada una de estas etapas de las páginas JSP automáticamente, basándose en la situación temporal de cada archivo.

3.5.6 Estructura de Directorios.

Una aplicación web tiene una estructura de archivos preestablecida que todos los motores de servlets compatibles entienden. El nivel superior o raíz de la aplicación (application root), contiene documentos html, páginas JSP, imágenes y algunos otros recursos que forman el contenido de la aplicación, también podemos encontrar subdirectorios con contenidos de la aplicación.

El directorio raíz debe contener un directorio llamado WEB-INF. Este directorio y sus subdirectorios no son visibles para los usuarios de la aplicación.

El directorio WEB-INF contiene servlets, clases, archivos .jar y los datos de configuración, todos estos forman la parte operativa de la aplicación.

Hay tres partes de especial interés en WEB-INF que son:

1. Directorio clases este contiene servlets y otras clases. Estas clases son halladas automáticamente por el cargador de clases servlet, como si estuviese en la ruta de clases (CLASSPATH) de la aplicación, además

puede contener subdirectorios que correspondan a la estructura de paquetes.

2. Directorio lib contiene los archivos .jar. Las todas las clases dentro de los archivos jar se ponen automáticamente a disposición del cargador de clases sin tener que estar explícitamente listadas en una ruta de clases.
3. Archivo web.xml es un documento XML llamado descriptor de despliegue, tiene una estructura rigurosamente establecida que se usa para configurar los servlets y otros recursos que forman la aplicación web.

3.5.7 Características de WAP-WML.

WAP es una serie de tecnologías que consisten en: WML, que es el lenguaje de etiquetas, WMLScript es un lenguaje de script, lo que vendría a ser JavaScript y el Wireless Telephony Application Interface (WTAI).

Las características principales de WML son:

- Soporte para imágenes y texto, con posibilidad de texto con formato.
- Tarjetas agrupadas en barajas. Una pagina WML es como una página HTML en la que hay una serie de cartas, al conjunto de estas cartas se les suele llamar baraja.
- Posibilidad de navegar entre cartas y barajas de la misma forma que se navega entre páginas Web.

- Manejo de variables y formularios para el intercambio de información entre el teléfono celular y el servidor.

WML es un lenguaje de marcas similar al HTML. WML es compatible con XML 1.0. Las páginas WML son llamadas barajas ya que están compuestas por cartas, un navegador WAP, solo puede mostrar una carta al mismo tiempo.

El lenguaje WML es para los entendidos un extensión de la definición XML donde se podrá definir el contenido de las paginas.

Una página WML trabaja con marcas (como HTML), en forma de apertura como `<wml>` y cierre como `</wml>`.

Una página WML podrá contener texto, tablas reducidas, algún tipo de formato de texto, hipervínculos o links (ya sea otras páginas WML o a otras cards del mismo WML), formularios para el envío de información, imágenes (en formato WBMP) y cierta programación de scripts que se ejecutarán directamente en el navegador WAP en el teléfono.

El mayor cuidado que debemos tener cuando desarrollamos páginas WML es que, al estar el navegador ejecutándose sobre equipos celulares, no debemos olvidarnos que estos poseen poca capacidad de memoria y de procesamiento, por lo cual ante cualquier mínimo error en el archivo WML, no se

podrá procesar y lo mismo sucederá si hacemos nuestro WML demasiado grande.

Las imágenes en las páginas WAP soportan el formato WBMP que es un formato de BMP extendido que reduce el tamaño de una imagen de pequeño tamaño. Recordemos que deben ser imágenes muy pequeñas para que se adapten a la pantalla móvil. Además, se utilizan en blanco y negro monocromáticos. Recién en la versión 2.0 de WAP se da soporte para imágenes a color.

CAPÍTULO 4. IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA TELEFONÍA CELULAR Y/O POCKET PC.

Se decidió desarrollar todas las Aplicaciones en las tecnologías Open Sources (Java 2, Java2 ME, MySQL) y no en la propuesta de Microsoft .NET que como sabemos su costo siempre es elevado.

Hemos desarrollado dos Aplicaciones para Celulares utilizando Java 2 ME como lenguaje de programación y una tercera Aplicación utilizando WAP-WML, HTML, Servlets y JSP's para la generación de WML y HTML dinámico, además MySQL como motor de Base de Datos.

4.1 Aplicaciones J2ME "Zombies en México" y "Recetario".

Hasta hace poco no era muy común pensar en "Sistemas Operativos para celulares"; sin embargo, cada vez más se diferencian los equipos por esta característica. En un principio, cada compañía instalaba su propio SO desarrollado para sus equipos, con el tiempo y para lograr mayor compatibilidad entre aplicaciones comenzaron a surgir distintos SO en el mercado móvil. Entre ellos podemos mencionar SymbianOS, PalmOS, Windows CE y Windows Mobile 2003.

El lenguaje que es compatible con todos estos SO e incluso con muchos otros, es Java2ME y es sobre el cual están hechas nuestras aplicaciones "Zombies en México" y "Recetario", las aplicaciones desarrolladas y compiladas en este lenguaje podrán ser ejecutadas en la mayoría de los equipos celulares actuales.

Una vez compiladas y probadas nuestras aplicaciones se crearán el paquete JAR (Java Archive) que contendrá las clases a ejecutar, junto con las imágenes PNG o archivos de datos que se utilizarán dentro de la aplicación y el paquete JAD (Java Descriptor) el cual describe textualmente la aplicación, sus propiedades e instrucciones para que el equipo pueda instalarlo correctamente. Estos dos archivos se pueden cargar en un equipo a través del puerto serial, infrarrojo y bluetooth o bien, pueden ser descargados de un servidor web desde Internet, esta operación se llama Over the Air y se le conoce con las siglas de OTA.

En este caso el teléfono primero solicita al servidor web el archivo JAD y cuando este es descargado y leído, obtendrá la dirección del archivo JAR, el cual será solicitado vía Internet al servidor web. Una vez finalizada la descarga del archivo JAR, comparará que la información del JAD concuerde con éste y si es correcto lo instalará sobre el equipo.

La única consideración a tener en cuenta es que el servidor web deberá tener configurados en forma correcta los tipo MIME para que automáticamente

entregue el tipo correcto de formatos. Los tipos MIME son: para archivos .JAD text/vnd.sun.j2me.app-descriptor y para archivos .JAR application/java-archive.

4.1.1 Juego Java "Zombies en México".

Ésta Aplicación se desarrollo en J2ME utilizando el perfil MIDP 2.0 y la configuración CLDC 1.0 que en capitulos anteriores se describieron junto con sus características.

El objetivo del juego es que el personaje principal "Joos" recolecte la mayor cantidad de corazones que pueda sin que los tres diferentes Zombies lo toquen o el tiempo para tomar el corazón se acabe. El juego se desarrolla en un laberinto donde las paredes cambian constantemente de posición.

Una vez iniciada la Aplicación lo primero que aparece es la Portada del juego y la música de fondo como se ve en la Figura 2, oprimiendo la tecla 5 entramos al juego que nos muestra el laberinto, al personaje principal "Joos", a los Zombies que trataran de matarnos y el corazón que debemos tomar, en el tablero se mostrará un punto azul que nos indicará en que dirección se encuentra el corazón. En la parte superior se encuentra la línea de tiempo que nos resta para tomar el siguiente corazón y debajo el numero de corazones que llevamos recolectados, ver Figura 3.

Para mover al personaje principal ocuparemos las teclas 2 (Arriba), 4 (Izquierda), 6 (Derecha) y 8 (Abajo).



Figura 2.

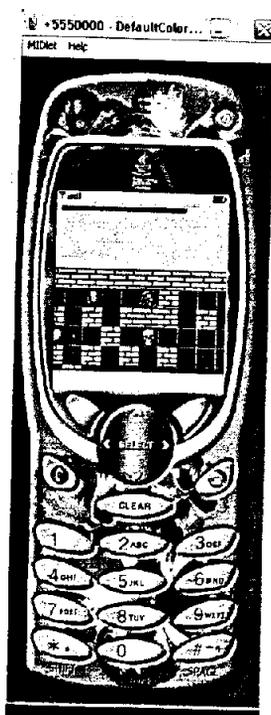


Figura 3.

Si nos llega a tocar un Zombie o se nos acaba el tiempo, el juego termina y aparece un letrero avisando que estamos muertos. Oprimiendo la tecla 5 pasaremos a la pantalla de puntuación, en la parte superior aparece la marca más alta conseguida, en el centro el número de corazones recolectados y en la parte inferior el nombre de los creadores del juego. En el caso de que hayamos superado la marca más alta aparecerá un aviso ¡¡¡Nueva Marca!!! y ese valor

quedará guardado hasta que otro jugador lo supere. Si queremos comenzar de nuevo el juego solo se tiene que oprimir la tecla 5. Ver Figuras 4 y 5.

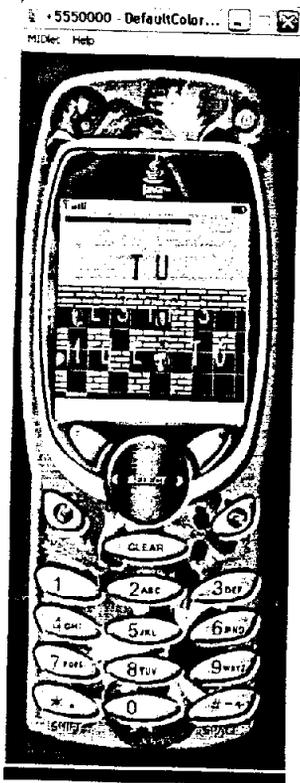


Figura 4.



Figura 5.

El juego esta programado con Java 2 Micro Edtion (J2ME), consta de 5

Clases las cuales son:

1. MiMidlet.java
2. Juego.java
3. GestorSonidos.java

4. miCanvas.java
5. PersistenciaRMS.java

MiMidlet.java. Es la Clase principal llamada Midlet se encarga de importar las librerías necesarias y de arrancar la Aplicación.

Juego.java. Clase que se encarga de gestionar todos lo elementos gráficos del juego como son pintar el tablero y a los personajes, el tablero esta en constante movimiento para cambiar de posición las paredes y darle un grado de dificultad.

GestorSonidos.java. Clase que se encarga de cargar y reproducir los ficheros WAV y MID, que son los sonidos que se reproducirán durante el juego.

miCanvas.java. Es la Clase que se encarga de gestionar los eventos del juego, como son los movimientos del personaje a través de las teclas del telefono.

PersistenciaRMS.java. Es la Clase que se encarga de almacenar en memoria la mejor marca obtenida.

4.1.2 "Recetario".

Ésta Aplicación al igual que la anterior se desarrollo en J2ME utilizando el perfil MIDP 2.0 y la configuración CLDC 1.0 que en capitulos anteriores se describieron junto con sus características.

El objetivo de esta Aplicación "Recetario" es tener en el Celular una base de datos con Recetas de Sopas, Ensaladas, Pescados, Pollos, Carnes, Postres y Bebidas.tan grande como nosotros vayamos incorporando más recetas.

Una vez iniciada la Aplicación aparecerá la portada, en donde se verá una imagen y el nombre de los creadores. Posteriormente se desplegará una lista con las recetas que tengamos capturadas ahí podremos seleccionar alguna y pasar a una pantalla en donde veremos el nombre de la Receta, los ingredientes necesarios, el modo de preparación y el tipo de receta, pero también tenemos la posibilidad de añadir más recetas aparecerá un formulario para capturar los mismos datos antes mencionados. Ver las Figuras 6, 7 y 8.



Figura 6.



Figura 7.

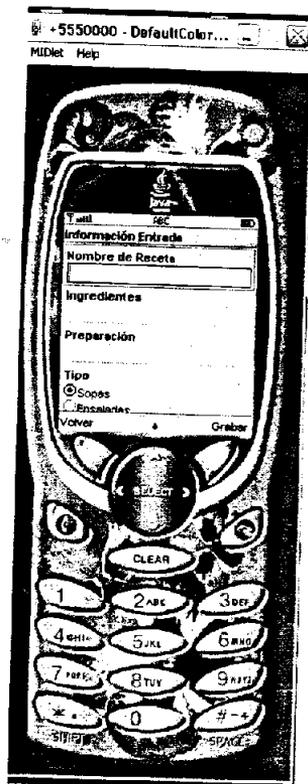


Figura 8.

La Aplicación está programada en Java 2 Micro Edition (J2ME) y consta de tres clases las cuales son:

1. RecetarioMid.java
2. Receta.java
3. RecetarioRMS.java

RecetarioMid.java. Es la Clase principal se encarga de iniciar la Aplicación y además de la parte gráfica para la elaboración de formularios y listas.

Receta.java. Esta Clase se encarga de manipular los datos ingresados en el formulario de captura para poder almacenarlos después.

RecetarioRMS.java. Es la Clase que se encarga del almacenamiento de la información.

4.2 Aplicación en HTML, WAP-WML, Java Servlets y JSP's "Agenda Virtual".

Internet móvil es la capacidad que tiene un equipo celular de navegar por la red Internet. La tecnología que permite esta navegación por Internet es la llamada WAP (Wireless Application Protocol) que hace de interfaz o pasarela entre la red de Internet y el protocolo HTTP con el que se reciben las paginas web en la red celular.

Los navegadores WAP de los equipos celulares soportan solamente paginas en formato WML, que es un versión reducida de HTML y adaptada a las necesidades de contenido de un teléfono celular.

Algunos equipos celulares con mayor capacidad de procesamiento y memoria poseen navegadores HTML que navegan por sitios estándar desarrollados para computadoras personales de tamaño normal. Sin embargo, si pensamos desarrollar para este tipo de equipos, tengamos en cuenta siempre la limitación del tamaño de pantalla, por lo que es recomendable realizar una versión reducida del sitio con menor cantidad de imágenes y optimizada para ser visualizada en dichos equipos.

Es por esto que la "Agenda Virtual" es un sistema que consta de un portal Web y un portal WML los cuales están en un servidor de aplicaciones Jboss y accedan a la misma base de datos MySQL por lo que se pueden ver los mismos datos de contactos en ambas aplicaciones.

El portal Web esta escrito en código HTML, JavaScript, CSS, java JSP, java Servlets para generar páginas HTML dinámicas y se conecta a una base de datos MySQL.

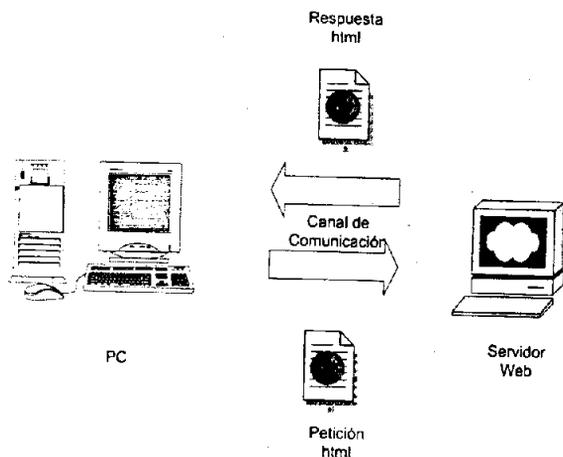
El portal WML fue desarrollado en código WML, java JSP, java Servlets para generar páginas WML dinámicas y utilizando una base de datos MySQL.

4.2.1 "Agenda Virtual".

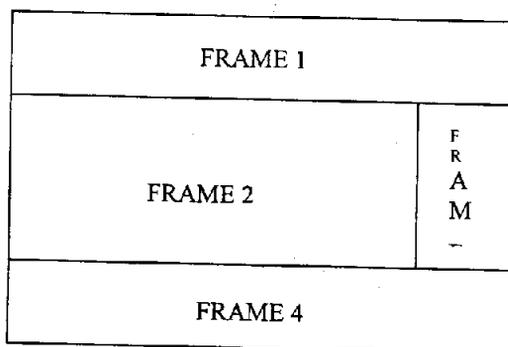
Este es un proyecto en el cual se pretende tener una agenda la cual se pueda acceder desde Internet desde una computadora de escritorio, una pocket PC en la dirección <http://dodgem.homeip.net/Tesis2005> o un teléfono celular y en la dirección <http://dodgem.homeip.net/TesisWml2005/Menu.wml> por lo cual se dividió en dos partes la primera basada en código HTML el cual puede ser accedido desde el browser de la computadora y la pocket PC y la segunda basada en código WML la cual puede ser accedida desde el navegador de un teléfono celular.

PORTAL WEB

El portal Web es un conjunto de paginas HTML, Servlets y JSP's con un formato CSS a las cuales se puede acceder desde Internet ya que se encuentran alojadas en un servidor Web el cual permite que estén publicadas en la red por lo que se puede acceder a ellas desde cualquier computadora o pocket PC con acceso a Internet.



PAGINA INDEX.html. La pagina principal del portal Web es una pagina HTML la cual contiene cuatro frames los cuales apuntan a distintas paginas HTML formateadas con CSS el cual permite dar estilo a las paginas Web.



PAGINA Cabecera.html. El primer frame es la cabecera la cual contiene el titulo de la pagina y un menú secundario en el cual se puede navegar entre distintas paginas HTML las cuales contienen datos del proyecto de Tesis.

Agenda Virtual

Presentación Planteamiento Objetivos Justificación Contactos

Tesis

PAGINA Presentacion.html. El segundo frame es el mas grande y es en el cual se despliega toda la información así como la funcionalidad del portal.

Titulo de la Tesis



Titulo de la Tesis

Este proyecto se desarrollo con el objetivo de obtener el titulo de la carrera de Ingenieria en Computacion de la Universidad Nacional Autonoma de México capus FEST Aragon.

Titulo de la Tesis

Desarrollo de aplicaciones móviles utilizando Java 2 ME como lenguaje de programación en teléfonos celulares y/o Pocket PC.

PAGINA Menú.html

Pagina Menu.html. El tercer frame es el menú principal en el cual se puede navegar por las paginas funcionales del portal las cuales son la inserción de un nuevo contacto, búsqueda, edición y borrado de los datos de contactos en la agenda.

Opciones



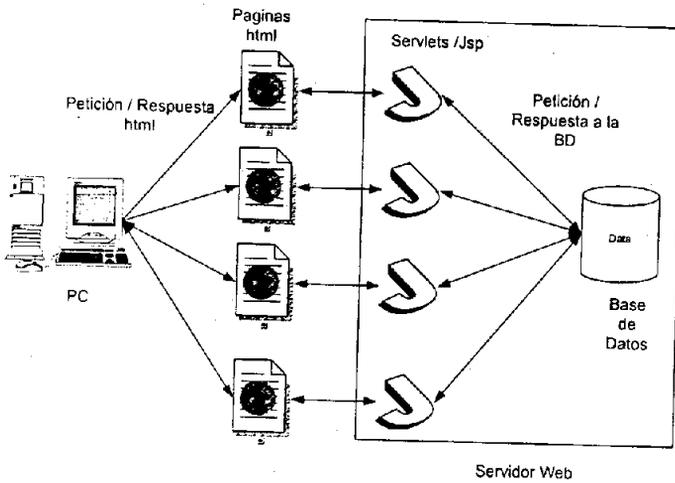
-
-
-
-
-

PAGINA Pie.html. El cuarto frame es el pie de la pagina el cual esta formado de márgenes, nombre de la escuela y programadores.

Trabajo de Tesis de Asignatura: Páginas Interactivas y Redes Gráficas. Carlos Iglesias.
Lima, Perú, 2001.

PAGINAS FUNCIONALES

Son paginas Web en las cuales se distribuyo las distintas funcionalidades de la Agenda como es Insertar un registro, consultar, editar y borrar esto se logra mediante paginas HTML las cuales mandan invocar distintos servlets lo cuales hacen la conexión a la base de datos y realizan las distintas transacciones o en su defecto pagina JSP la cuales internamente se conectan a la base de datos y realizan todas las transacciones ellas mismas.



PAGINA Insert.html. Es el formulario para colocar los datos del cliente en esta página HTML al presionar el botón de Submit se manda llamar el servlet insertar.java el cual inserta los registros en las diferentes tablas de la base de datos.

Inicio Contenido

Inicio Contenido

Nombre:

Apellido Paterno:

Apellido Materno:

Sexo:

Hombre Mujer

Fecha de Nacimiento:

Teléfono Casa:

Teléfono Celular:

Teléfono Trabajo:

E-mail:

Aceptar

Pagina Select.jsp. En esta pagina se realizan las búsquedas de los contactos que están ya registrados, además muestra el detalle de los datos del cliente en la pagina selectDetalle.jsp.

Consulta Contacto

Formulario de búsqueda

Nombre:

Apellido Paterno:

Apellido Materno:

Nombre	Paterno	Materno	E-mail	Tel-Casa	Detalles
Artemio	Alvarado	Aquino		55555555555555	<input type="button" value="Aqui"/>

Pagina Editar.jsp. En esta pagina se realizan las actualizaciones de los contactos que están ya registrados, buscando al contacto en especifico y luego yendo al detalle de los datos del cliente en la pagina editarDetalle.jsp.

Editar Contacto

Formulario de búsqueda

Nombre:

Apellido Paterno:

Apellido Materno:

Nombre	Paterno	Materno	E-mail	Tel-Casa	Editar
Artemio	Alvarado	Aquino		55555555555555	<input type="button" value="Editar"/>

Editar Contacto

Nombre: Artemio
Apellido Paterno: Alvarado
Apellido Materno: Aquino
Sexo: M
Fecha de Nacimiento: 04/08/1980
Telefono Casa: 555555555555
Telefono Celular: 666666666666
Telefono Trabajo: 777777777777
E-mail:

Pagina Borrar.jsp. Realiza el borrado de la información en la base de datos, esto lo hace buscando al contacto en particular que se desea borrar y al dar clic en el contacto deseado invocando al servlet delete.java.

Forma Contacto

Editar el contacto

Nombre: a
Apellido Paterno:
Apellido Materno:

Nombre	Paterno	Materno	E-mail	Tel-Casa	Borrar
Artemio	Alvarado	Aquino		555555555555	<input type="button" value="Borrar"/>

Documento CSS. Las paginas HTML son formateadas para los colores de titulos, tipo de letra, color de letra, color de fondo asi como el color y forma de

botones y barras de desplazamiento por un archivo CSS el cual contiene las especificaciones para los diferentes tags de HTML.

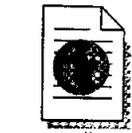
PORTAL WML

El portal WML es un conjunto de paginas WML o también conocidas como WAP además de Servlets y JSP's a las cuales se puede acceder desde el navegador de un teléfono celular ya que este tipo de navegadores solo interpreta paginas programadas en wml .

En este portal únicamente se incluyeron las funcionalidades de insertar un nuevo contacto y consulta de información ya que el borrado y la edición incluyen varias operaciones lo que aumentaría el tiempo de conexión a Internet lo cual resultaría en muy altos costos, por lo que estas dos funcionalidades solo son accesibles en el portal web.

La versatilidad de poderse conectar y buscar un contacto desde el celular sin importar el lugar donde se este permite que esta aplicación sea muy practica ya que esta misma información puede ser accesada o actualizada desde una computadora que se encuentre conectada ha Internet.

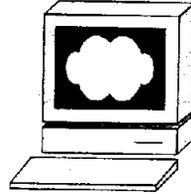
Respuesta Wml



Petición Wml

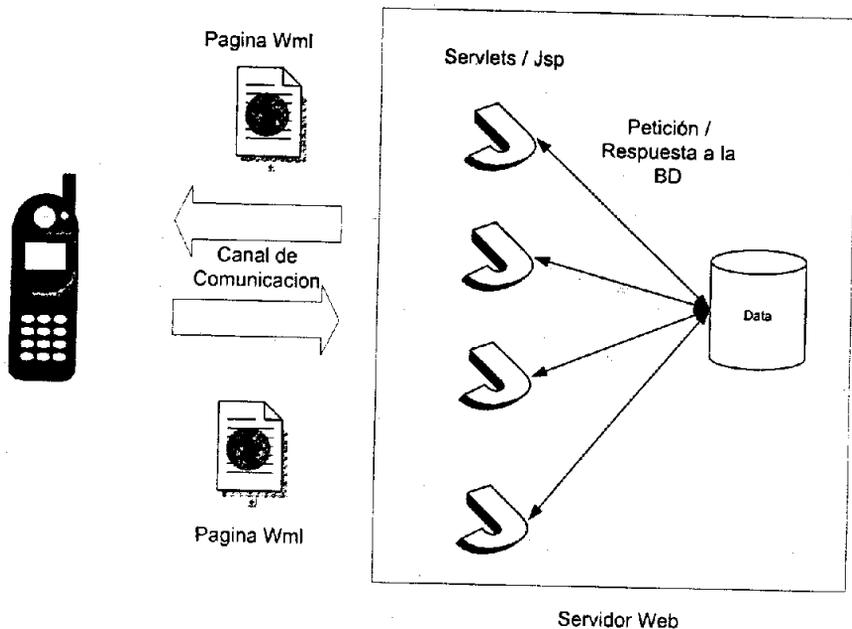


Teléfono
Celular



Servidor
Web

La funcionalidad del portal WML esta dada por distintos JSP's y servlets los cuales acceden a una base de datos para insertar o buscar información.



Las paginas wml también conocidas como wap están conformadas por barajas lo cual al ser accedido desde un teléfono celular da la impresión de navegar o moverse entre distintas paginas lo cual no es correcto ya que se esta moviendo entre las diferentes cartas de una misma pagina. Esto permite que el celular cargue una única pagina pero mejor distribuida ya que cada carta incluye toda la pantalla del celular.

Menu.wml

Card1:

Es la pagina primera carta la cual es una pagina de presentación la cual contiene el nombre de la escuela y el nombre de los programadores.



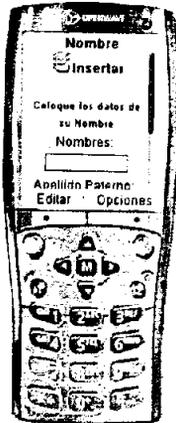
Card2:

En esta car se presenta un pequeño menú el cual nos da las opciones de buscar o insertar un contacto para la agenda o ver las referencias del juego y el recetario programados en j2me.



Insertwml.jsp

Este programa JSP se encarga de crear un formulario wml con los datos del cliente y a su vez se encarga de guardar la información en la base de datos en el momento que se llene el formulario y sea mandada la información.



Selectwml.jsp

Se encarga de crear la pagina wml con los campos requeridos para la búsqueda de información y también realiza la conexión a la base de datos y hace la búsqueda.



Una vez que se realizo la búsqueda se muestran en pantalla los resultados de dicha acción.

Juego.jsp

Es un programa JSP el cual se encarga de escribir la pagina wml con los datos de las aplicaciones J2ME que son el juego "Zombies en México" y el "Recetario".

CONCLUSIONES

Con este trabajo se trató de bosquejar un panorama global de lo que son las tecnologías para dispositivos móviles, que en nuestra opinión es el futuro de las telecomunicaciones y el desarrollo de software, además de ser uno de los campos en los cuales se puede desarrollar un Ingeniero en Computación.

Se estudiaron las plataformas Microsoft .NET y Java 2 que son tecnologías que permiten el desarrollo de aplicaciones para dispositivos móviles y se decidió desarrollar todas las Aplicaciones en esta última debido a que su distribución es gratuita.

Se abordaron temas que si talvez ya no son tan recientes aún son relativamente nuevos para los recién egresados de la carrera y con esto pretendemos sino darles una base sólida por lo menos darles un panorama general de lo que son estas tecnologías.

Para lograr esto nos basamos en los conocimientos adquiridos en nuestra experiencia laboral así como en los adquiridos durante la carrera en materias como son: Computadoras y Programación, Programación Estructurada y Características de Lenguaje, Ingeniería de Sistemas, Programación de Sistemas, Ingeniería de programación, Sistemas Operativos, Bases de Datos, Sistemas de Información, Redes de Computadoras y Seminario de Ingeniería de

Computación, las cuales nos dieron las bases y fundamentos para desarrollar nuestras aplicaciones de la mejor manera posible.

Estamos concientes que si bien nuestras aplicaciones no son muy grandes si cubren las características necesarias para ser la base de desarrollos más complejos como sería un sistema integral de información, ya que un sistema así se basa en el manejo de conectividad a Base de Datos y la manipulación de la información que en estas residen.

Aún así tratamos de crear aplicaciones lo más variadas posibles con el fin de cubrir de la mejor manera los temas investigados en nuestra Tesis.

En cuanto a las aplicaciones J2ME siempre nos parecieron interesantes así que decidimos inventar nuestro propio juego el cual cubre un interés personal y además un recetario que permite simular una pequeña base de datos dentro del teléfono móvil, la cual podría modificarse fácilmente para almacenar cualquier tipo de información.

En nuestra experiencia laboral en las diferentes empresas siempre hay algo en común y es que siempre tienen sistemas en los cuales los usuarios puedan realizar altas, bajas y cambios en información almacenada en Bases de Datos por lo que, si bien nuestra aplicación "Agenda Virtual" no está completamente dirigida a una empresa en particular es fácilmente amoldable a

cualquier rubro, ya que lo único que cambiaría es el significado de los valores de los datos que se manipulan.

Nuestro objetivo personal del por qué realizar este tema de Tesis es nuestro interés por manejar las diferentes tecnologías Java, ya que en el campo laboral al manejar estas tecnologías se abren diferentes posibilidades de empleo. Además de que revisamos otras tecnologías igualmente recientes.

La contribución más importante es a nivel personal pues se tenía la necesidad de conocer el lenguaje de programación Java ya que éste es el que se maneja en nuestra área laboral, la diversidad de las aplicaciones nos permitió practicar los distintos conocimientos adquiridos en diversos cursos y nos ayudó a adquirir algo de experiencia en el manejo de este lenguaje así como aplicar algunas de las cosas aprendidas en nuestra formación profesional al cursar la carrera.

APENDICE A Código del Juego Java "Zombies en México"

A continuación se presenta el código de las diferentes Clases Java que hacen posible que el juego funcione.

MiMidlet.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.game.*;
import javax.microedition.lcdui.*;
import java.io.InputStream;
import javax.microedition.media.*;
import java.util.*;
import javax.microedition.rms.*;

public class MiMidlet extends MIDlet{

    Display dis;

    public void startApp(){

        dis=Display.getDisplay(this);
        dis.setCurrent(new miCanvas());
    }
    public void pauseApp(){

    }
    public void destroyApp(boolean unconditional){
        GestorSonidos.getInstancia().pararFondo();
    }

}
```

Juego.java

```
class Juego{
    Sprite portada;
    Sprite gameOver;
    TiledLayer tablero;
    int estado=0; //0 portada, 1 jugando, 2 gameOver, 3 puntuacion
    int x[],y[];
    int temporizador=0;
    int cuentaAtras=0;
    byte puntos=0;
    int tocaPieza=0;
    boolean nuevoRecord;
```

```

byte record=0;
Juego(){
    try{
        portada=new Sprite(Image.createImage("/Portada.png"));
        tablero=new TiledLayer (19,19,Image.createImage("/Celdass.png"),20,20);
        gameover=new Sprite(Image.createImage("/GameOver.png"));
        temporizador=10;
        PersistenciaRMS pr=new PersistenciaRMS();
        record=pr.getPuntos();
    }catch(Exception e){
        e.printStackTrace();
    }
}

public void iniciarPartida(){
    System.gc();
    int xa[]={1,16,17,17,0};
    int ya[]={1,17,17,16,0};
    x=xa;
    y=ya;
    tablero.setPosition(50,50);

    for(int n=0;n<19;n++)
        for (int m=0;m<19;m++){
            if (n % 2 ==0 && m % 2 == 0)
                tablero.setCell(n,m,1);//Ladrillo
            else
                tablero.setCell(n,m,2); //Vacio
            if (n==0 || n==18 || m==0 || m==18)
                tablero.setCell(n,m,1); //Ladrillo
        }
    tablero.setCell(1,1,3);
    tablero.setCell(16,17,7);
    tablero.setCell(17,17,8);
    tablero.setCell(17,16,9);
    ponerAnillo();
    cuentaAtras=170;
    puntos=0;
    locaPieza=2;
    nuevoRecord=false;
}

public void pintar(Graphics g){
    try{
        if (estado==0){
            portada.paint(g);
        }else if(estado==1 || estado==2){
            if (estado==1){
                cuentaAtras--;
                if (cuentaAtras==0) {
                    morir();
                }
            }
            tablero.setPosition(90-x[0]*20,90-y[0]*20);
            tablero.paint(g);
            g.setColor(255,0,0);
            g.fillRect(5,5,cuentaAtras, 5);

            for (int n=0;n<puntos % 20;n++){
                g.setColor(255,255,0);
                g.fillArc(n*7+5, 12, 5,5,0,360);
            }
            for (int n=0;n<puntos / 20;n++){
                g.setColor(0,128,0);
                g.fillArc(n*7+5, 18, 5,5,0,360);
            }
            //Visualiza una marca azul para saber donde esta el corazón
            int xMarca=x[4]-x[0];
            int yMarca=y[4]-y[0];
            if (xMarca<-4) xMarca=-4;
            if (xMarca>3) xMarca=3;
        }
    }
}

```

```

        if (yMarca<-4) yMarca=-4;
        if (yMarca>3) yMarca=3;
        if (xMarca==4 || xMarca==3 || yMarca==4 || yMarca==3){
            g.setColor(0,0,255);
            g.fillRect(100+xMarca*20, 100+yMarca*20,3,3);
        }
        if (estado==2) gameOver.paint(g);
    }else if(estado==3){
        g.setColor(0,0,0);
        g.fillRect(0,0,200,200);
        g.setColor(255,255,255);

        g.setFont(Font.getFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,Font.SIZE_LARGE));
        g.drawString("Puntos: " + puntos + " corazones",
90,60,Graphics.BASELINE|Graphics.HCENTER);
        if (nuevoRecord){
            g.drawString("¡¡¡NUEVA
MARCA!!!",90,110,Graphics.BASELINE|Graphics.HCENTER);
        }

        g.setFont(Font.getFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,Font.SIZE_SMALL));
        g.setColor(64,128,128);
        g.drawString("Marca: "+ record + "
corazones",90,20,Graphics.BASELINE|Graphics.HCENTER);
        g.setFont(Font.getFont(Font.FACE_SYSTEM,0,Font.SIZE_SMALL));
        g.setColor(64,128,255);
        g.drawString("Por J.Luis Anguiano y Oscar Rocha
",10,160,Graphics.BASELINE|Graphics.LEFT);
        g.drawString("UNAM FES Aragón",10,170,Graphics.BASELINE|Graphics.LEFT);
    }

    temporizador--;
    if (temporizador<0) temporizador=0;

} catch(Exception e){
}
}

private void mover(int dx, int dy, int pose){
    if (temporizador>0) return;
    if (estado==1){
        int valor=tablero.getCell(x[0]+dx,y[0]+dy);
        if (valor==10) {
            puntos++;
            ponerAnillo();
            cuentaAtras=170;
            tocaPieza=2;
            GestorSonidos.getInstance().escucharTomar();
        }
        if (valor==2 || valor==10){
            tablero.setCell(x[0],y[0],2);
            y[0]+=dy;
            x[0]+=dx;
            tablero.setCell(x[0],y[0],pose);
        }
    }
}

private void morir(){
    estado=2;
    temporizador=10;
    if (puntos>record) {
        record=puntos;
        nuevoRecord=true;
        PersistenciaRMS pr=new PersistenciaRMS();
        pr.setPuntos(record);
    }

    GestorSonidos.getInstance().escucharMorir();
}

```

```

    }

    public void pulsaArriba(){
        if (temporizador>0) return;
        if (estado==1) mover(0,-1,5);
    }

    public void pulsaAbajo(){
        if (temporizador>0) return;
        if (estado==1) mover(0,1,3);
    }

    public void pulsaDerecha(){
        if (temporizador>0) return;
        if (estado==1) mover(1,0,4);
    }

    public void pulsaIzquierda(){
        if (temporizador>0) return;
        if (estado==1)mover(-1,0,6);
    }

    public void pulsaDisparo(){
        if (temporizador>0) return;
        if (estado==0) {
            iniciarPartida();
            estado=1;
        }else if (estado==2){
            estado=3;
            temporizador=5;
        }else if (estado==3){
            GestorSonidos.getInstance().escucharFondo();
            estado=0;
            temporizador=5;
        }
    }

    public void ponerAnillo(){
        Random r=new Random();
        do{
            x[4]=Math.abs(r.nextInt()%17)+1;
            y[4]=Math.abs(r.nextInt()%17)+1;
        }while(tablero.getCell(x[4],y[4])!=2);
        tablero.setCell(x[4],y[4],10);
    }

    public void moverEnemigos(){
        if (estado!=1) return;
        Random r=new Random();
        int x2=Math.abs(r.nextInt()%17)+1;
        int y2=Math.abs(r.nextInt()%17)+1;
        if (tablero.getCell(x2,y2)==tocaPieza){
            if (tablero.getCell(x2,y2)==1) {
                tablero.setCell(x2,y2,2);
                tocaPieza=2;
            }
            else if (tablero.getCell(x2,y2)==2) {
                tablero.setCell(x2,y2,1);
                tocaPieza=1;
            }
        }
        int azar, aux;

        for (int n=1;n<4;n++){
            aux=5 - (puntos/20);
            if (aux<1) aux=1;
            azar=Math.abs(r.nextInt()%aux);
            if (azar==0){
                tablero.setCell(x[n],y[n],2);
                if (x[n]<x[0] && tablero.getCell(x[n]+1,y[n])>=2 && tablero.getCell(x[n]+1,y[n])<=6)

```

x[n]++;


```

        try{ if (pMorir!=null) pMorir.start();
            pFondo.setMediaTime(100800000);
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public void escucharPulso(){
        try{ if (pPulso!=null) pPulso.start();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public void pararFondo(){
        //Para el sonido de fondo
        try{
            if (pFondo!=null) pFondo.stop();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

miCanvas.java

class miCanvas extends GameCanvas implements Runnable{

```

    int x=90,y=90;
    Thread hilo;
    public miCanvas (){
        super(true);
        hilo=new Thread(this);
        hilo.start();
    }

    public void run(){
        Juego j=new Juego();
        GestorSonidos.getInstancia().cargarSonidos();
        GestorSonidos.getInstancia().escucharFondo();
        Graphics g=getGraphics();
        int tecla;
        int ancho=getWidth();
        int alto=getHeight();
        while (true){
            g.setColor(20,128,255);
            g.fillRect(0,0,ancho,alto);
            tecla=getKeyStates();
            if (tecla!=0) GestorSonidos.getInstancia().escucharPulso();

            if ((tecla & RIGHT_PRESSED) != 0) j.pulsaDerecha();
            if ((tecla & LEFT_PRESSED) != 0) j.pulsaIzquierda();
            if ((tecla & DOWN_PRESSED) != 0) j.pulsaAbajo();
            if ((tecla & UP_PRESSED) != 0) j.pulsaArriba();
            if ((tecla & FIRE_PRESSED) != 0) j.pulsaDisparo();
            j.moverEnemigos();

            j.pintar(g);
            flushGraphics();
            try{
                Thread.sleep(50);
            }catch(Exception e){
            }
        }
    }
}

```

PersistenciaRMS.java

```
class PersistenciaRMS{
    public void setPuntos(byte puntos){
        try{
            //Abre el registro correspondiente
            RecordStore rs=RecordStore.openRecordStore("Puntuacion", true);
            byte b[]={puntos};
            if (rs.getNumRecords()==1)
                rs.setRecord(1,b,0,1);
            else
                rs.addRecord(b,0,1);
            rs.closeRecordStore();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public byte getPuntos(){
        try{
            RecordStore rs=RecordStore.openRecordStore("Puntuacion", true);
            byte b[];
            if (rs.getNumRecords()==1){
                b=rs.getRecord(1);
                return b[0];
            }else return 0;
        }catch(Exception e){
            e.printStackTrace();
            return 0;
        }
    }
}
```

APENDICE B Código de la Aplicación Java "Recetario"

A continuación se presenta el código de las diferentes Clases Java que hacen posible que la aplicación Recetario pueda ejecutarse en el teléfono.

RecetarioMid.java

```
package recetario;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.lcdui.List;
import java.util.*;
import java.io.*;
import javax.microedition.io.*;
import javax.microedition.rms.*;

public class RecetarioMid extends MIDlet implements CommandListener {
    // Definimos la variable de la pantalla del dispositivo
    private Display display;
    // Definimos el formulario en donde se permite la edición y
    // presentación de los datos correspondientes a una receta
    private Form formulario;
    // Definimos la pantalla que presenta la lista de recetas que
    // están almacenados en la zona RMS
    private List contactos;
    // Campos de texto para la entrada de datos correspondientes a cada
    // receta
    private TextField receta,ingredientes,pasos;
    // Grupo de selección para clasificar la receta
    private ChoiceGroup tipoReceta;
    // Flag de control de la edición
    private boolean enEdición = false;
    // Instancia de la clase de control de la zona RMS
    private RecetarioRms rs;
    // Vector que contiene los identificadores de los registros de la
    // zona RMS que corresponden a cada uno de los contactos de la agenda
    private Vector ids = new Vector();
    // Array de imágenes de los tipos de receta
    private Image[] imagenes;
    // Comandos de control de la pantalla y acciones del Recetario
    private Command salir,nuevo,volver,grabar,borrar;
    // Identificador que asignamos a la zona de almacenamiento
    static final String ID_RMS = "recetario";

    // Constructor del midlet que inicializa todas las variables,
    // comandos, campos de entrada de datos y carga el array de
    // imágenes de la clasificación de las recetas
    public RecetarioMid() {
        // Clasificación de las recetas
        String[] tipos = { "Sopas","Ensaladas","Pescados","Pollos","Carnes","Postres","Bebidas" };
        // Creamos los comandos que aparecerán en la pantalla
        salir = new Command( "Salir",Command.EXIT,2 );
        nuevo = new Command( "Añadir",Command.SCREEN,3 );
        volver = new Command( "Volver",Command.BACK,2 );
        grabar = new Command( "Grabar",Command.OK,3 );
        borrar = new Command( "Eliminar",Command.SCREEN,3 );
        // Creamos la pantalla principal para la lista de recetas
        contactos = new List( "Recetas",List.IMPLICIT );
        // Creamos el formulario que contendrá la información de una
```

```

// receta y los campos de texto para permitir la entrada de
// datos
formulario = new Form( "Información Entrada" );
receta = new TextField( "Nombre de Receta", "", 30, TextField.ANY );
ingredientes = new TextField( "Ingredientes", "", 150, TextField.ANY );
pasos = new TextField( "Preparación", "", 150, TextField.ANY );
tipoReceta = new ChoiceGroup( "Tipo", Choice.EXCLUSIVE, tipos, null );

// Cargamos las imágenes de clasificación de los tipos de recetas
try {
    imagenes = new Image[7];
    imagenes[0] = Image.createImage( "/Sopas.png" );
    imagenes[1] = Image.createImage( "/Ensaladas.png" );
    imagenes[2] = Image.createImage( "/Pescados.png" );
    imagenes[3] = Image.createImage( "/Pollos.png" );
    imagenes[4] = Image.createImage( "/Carnes.png" );
    imagenes[5] = Image.createImage( "/Postres.png" );
    imagenes[6] = Image.createImage( "/Bebidas.png" );
} catch( IOException e ) {
    e.printStackTrace();
}

// Abrimos la zona de almacenamiento donde están guardadas todas las recetas
try {
    rs = new RecetarioRms( ID_RMS );
} catch( Exception e ) {
    e.printStackTrace();
}

// Recorremos la zona y vamos recuperando la lista de los
// identificadores que están asignados a cada uno de los registros
RecordEnumeration registros = null;
try {
    registros = rs.getNumRegistrosRecetario();
    while( registros.hasNextElement() )
        // Añadimos el identificador del registro al vector
        ids.addElement( new Integer( registros.nextRecordId() ) );
} catch( Exception e ) {
    e.printStackTrace();
}

// Volvemos a recorrer la zona, creando ahora las instancias de
// objetos de tipo Receta que corresponden a cada uno de los
// registros que contiene la zona de almacenamiento
registros.reset();
try {
    while( registros.hasNextElement() ) {
        Receta contacto = new Receta( registros.nextRecord() );
        contactos.append( contacto.getReceta() + ", "+
            contacto.getIngredientes().imagenes[contacto.getTipo()] );
    }
} catch( Exception e ) {
    e.printStackTrace();
}
}

// Método de arranque del midlet
protected void startApp() {
    // Recuperamos el display del objeto estático del Sistema
    display = Display.getDisplay( this );
    // Incorporamos los comandos que se presentarán en la pantalla
    // que presenta la lista de recetas
    contactos.addCommand( salir );
    contactos.addCommand( nuevo );
    contactos.setCommandListener( this );
    // Incorporamos al formulario los campos de texto correspondientes
    // a cada uno de los datos de una receta
    formulario.append( receta );
    formulario.append( ingredientes );
    formulario.append( pasos );
    formulario.append( tipoReceta );
}

```

```

// Incorporamos los comandos que se presentarán en la pantalla
// que presenta la información correspondiente a cada receta
formulario.addCommand( volver );
formulario.addCommand( grabar );
formulario.addCommand( borrar );
formulario.setCommandListener( this );

// Si es la primera vez que ejecutamos el midlet, presentamos
// la pantalla del logotipo
Displayable d = display.getCurrent();
if( d == null ) {
    try {
        Alert splash = new Alert( null,
            "Recetario J.Luis Anguiano y Oscar Rocha",
            Image.createImage("/Recetas.png"),AlertType.INFO );
        splash.setTimeout( 5000 );
        // La dejamos que se vea unos segundos, e indicamos que la
        // transición es a la pantalla de datos
        display.setCurrent( splash,contactos );
    } catch( Exception e ) {
        // Presentamos la pantalla con la lista de recetas
        display.setCurrent( contactos );
    }
}
else {
    // Presentamos la pantalla con la lista de recetas
    display.setCurrent( contactos );
}
}

// Suspende las acciones en segundo plano y libera recursos
// mientras el midlet no está activo
protected void pauseApp() {}

// Detiene toda actividad del midlet, cierra la zona RMS y
// libera los recursos acaparados por el midlet
protected void destroyApp( boolean flag ){
    try {
        rs.cerrar();
    } catch( Exception e ) {
        System.out.println( "Problemas al cerrar la zona RMS." );
    }
}

// Método encargado del control de todos los comandos de las
// pantallas de presentación de la lista de recetas y de la
// presentación de los datos correspondientes a cada una de las
// recetas registradas en el Recetario
public void commandAction( Command c,Displayable d ) {
    // Comando para abandonar la aplicación
    if( c == salir ) {
        destroyApp( true );
        notifyDestroyed();
    }
    // Comando para crear una nueva receta y añadirla al Recetario
    else if( c == nuevo ) {
        // Borramos el contenido de los campos de texto
        receta.setString( "" );
        ingredientes.setString( "" );
        pasos.setString( "" );
        tipoReceta.setSelectedIndex( 0,true );

        // Quitamos el comando de borrado de la pantalla
        formulario.removeCommand( borrar );
        // Indicamos que no es edición, sino una nueva receta
        enEdicion = false;
        // Presentamos en la pantalla del dispositivo móvil el
        // formulario
        display.setCurrent( formulario );
    }
}

```

```

// Si se ha seleccionado alguna de las recetas que aparecen en
// la pantalla de presentación de la lista de recetas
else if( c == List.SELECT_COMMAND ) {
    // Recuperamos el identificador que corresponde a la receta
    int indice = contactos.getSelectedIndex();
    int id = ((Integer)ids.elementAt(indice)).intValue();
    // Recuperamos el contenido del registro de la zona RMS
    Receta receta = rs.getRegistroRecetario( id );

    // Asignamos a la receta el contenido de cada uno de los
    // campos de formulario de entrada de datos
    receta.setString( receta.getReceta());
    ingredientes.setString(recetario.getIngredientes());
    pasos.setString(recetario.getPasos());
    tipoReceta.setSelectedIndex( receta.getTipo(),true );

    // Colocamos el comando de borrado del registro en la pantalla
    formulario.addCommand( borrar);
    // Indicamos que se permite la edición del contenido del
    // formulario
    enEdicion = true;
    // Presentamos en la pantalla del dispositivo móvil el
    // formulario con los datos de la receta
    display.setCurrent( formulario );
}

// Si se ha seleccionado el comando para eliminar el registro
// que actualmente se encuentra seleccionado, lo hacemos
else if( c == borrar ) {
    // Recuperamos el identificador que corresponde a la receta
    int indice = contactos.getSelectedIndex();
    int id = ((Integer)ids.elementAt(indice)).intValue();
    // Eliminamos el registro que corresponde a la receta en la
    // zona de almacenamiento RMS
    rs.eliminaRegistroRecetario( id );
    ids.removeElementAt( indice );
    contactos.delete( indice );
    // Presentamos en la pantalla del dispositivo móvil la lista
    // de todas las recetas que contiene el Recetario
    display.setCurrent( contactos );
}

// Si queremos pasar de la presentación del formulario de datos
// de una receta a la lista de recetas del Recetario
else if( c == volver ) {
    display.setCurrent( contactos );
}

// Si queremos guardar los datos que contiene el formulario de
// entrada/edición del contenido de una receta determinada
else if( c == grabar ) {
    // Controlamos el flag de edición. Si se está editando una
    // receta, entonces solamente actualizamos la información
    // correspondiente al registro existente en la zona RMS para
    // esa receta
    if( enEdicion ) {
        // Recuperamos el identificador que corresponde a la receta
        int indice = contactos.getSelectedIndex();
        int id = ((Integer)ids.elementAt(indice)).intValue();
        // Creamos una nueva receta con los datos que contiene el
        // formulario
        Receta contacto = new Receta( receta.getString(),
                                     ingredientes.getString(),
                                     pasos.getString(),
                                     tipoReceta.getSelectedIndex() );

        // Grabamos los nuevos datos en el registro correspondiente
        rs.setRegistroRecetario( id,contacto.carga() );
        // Actualizamos los datos de la receta en la lista de
        // selección
        contactos.set( indice,contacto.getReceta()+" "+
                    contacto.getPasos(),imagenes[contacto.getTipo()] );
    }
}

```

```

// Si no estamos en edición, se crea un nuevo registro para la
// receta que se incorpora al Recetario
else {
// Creamos una nueva receta con los datos que contiene el
// formulario
    Receta contacto = new Receta( receta.getString(),
    ingredientes.getString(),
    pasos.getString(),
    tipoReceta.getSelectedIndex() );

// Incorporamos al vector de identificadores el número de
// identificación que asigna el sistema al registro a la hora
// de crearlo en la zona de almacenamiento RMS
ids.addElement(
    new Integer(rs.nuevoRegistroRecetario(contacto.carga())) );
// Añadimos los datos de la receta a la lista de selección

contactos.append( contacto.getReceta()+" "+
    contacto.getPasos(),imagenes[contacto.getTipo()] );

}

// Presentamos la lista de recetas en la pantalla del
// dispositivo móvil
display.setCurrent( contactos );
}
}
}

```

Receta.java

```

package recetario;
import java.util.*;

import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.TextField;

public class Receta {

    private String receta,ingredientes,pasos;
    private int tipo;

// Constructor básico al que se pasan como parámetro cada uno de
// los datos de la receta. Se utiliza cuando se crea una receta
// desde la pantalla del dispositivo
public Receta( String _receta,String _ingredientes,String _pasos,int _tipo ){
    receta = _receta;
    ingredientes = _ingredientes;
    pasos = _pasos;
    tipo = _tipo;
}

// Constructor que obtiene la información de una receta a partir
// de un array de bytes. Se utiliza para crear recetas a partir
// de la información de los registros de la zona RMS. Cada uno de
// los datos está separado del siguiente mediante un ",".
public Receta( byte[] datos ) {
    descarga( new String(datos) );
}

// Método que carga cada una de las variables de la clase con la
// información correspondiente a la receta obtenida del registro
// de la zona RMS correspondiente
public void descarga( String datos ) {
    int inicio = 0;
    int fin = datos.indexOf( ',' );

```

```

receta = datos.substring( inicio,fin );
inicio = fin+1;
fin = datos.indexOf( ";",inicio );
ingredientes = datos.substring( inicio,fin );
inicio = fin+1;
fin = datos.indexOf( ";",inicio );
pasos = datos.substring( inicio,fin );
inicio = fin + 1;
tipo = Integer.parseInt( datos.substring(inicio,datos.length()) );
}

// Método que agrupa toda la información correspondiente a la receta
// en una cadena, separando cada uno de los datos entre sí mediante
// un ";".
public String carga() {
    return( receta +";" + ingredientes +";" + pasos +";" +
           ((String)Integer.toString(tipo)) );
}

// Conjunto de métodos que devuelven cada uno de los datos que
// corresponden a una receta, de forma individual
public String getReceta() {
    return( receta );
}
public String getIngredientes() {
    return( ingredientes );
}
public String getPasos() {
    return( pasos );
}

public int getTipo() {
    return( tipo );
}
}

```

RecetarioRMS.java

```

package recetario;
import javax.microedition.rms.*;
import java.util.Enumeration;
import java.util.Vector;
import java.io.*;

public class RecetarioRms {
    RecordStore rs = null;

    public RecetarioRms( String nombre ) {
        // Abrimos la zona de almacenamiento que corresponde al identificador
        // que nos pasan como parámetro
        try {
            rs = abrir( nombre );
        } catch( RecordStoreException e ) {
            e.printStackTrace();
        }
    }

    // Método utilizado para abrir/crear la zona de almacenamiento
    // utilizada por el recetario
    public RecordStore abrir( String nombre ) throws RecordStoreException {
        // Si no existe la zona de almacenamiento con el identificador que
        // nos pasan, la creamos de nuevo
        return( RecordStore.openRecordStore(nombre,true) );
    }
}

```

```

// Método utilizado para cerrar la zona de almacenamiento utilizada
// por el recetario
public void cerrar() throws RecordStoreNotOpenException,
RecordStoreException {
// Si la zona de almacenamiento está vacía, no contiene ningún
// registro, entonces la borramos completamente
if( rs.getNumRecords() == 0 ) {
// Obtenemos el identificador de la zona RMS
String nombre = rs.getName();
// La cerramos
rs.closeRecordStore();
// La eliminamos completamente
rs.deleteRecordStore( nombre );
}
// Si contiene registros, entonces solamente la cerramos
else {
rs.closeRecordStore();
}
}

// Este es el método invocado para recuperar el contenido de uno
// de los registros de la zona RMS. Se le pasa el identificador
// asignado en la creación
public Receta getRegistroRecetario( int id ) {
// Creamos una nueva receta con la información que contiene el
// registro al que apunta el identificador que se pasa
try {
return( new Receta(rs.getRecord(id)) );
} catch( RecordStoreException e ) {
e.printStackTrace();
}
return( null );
}

// Metodo utilizado para volcar en el registro que se indique en
// el identificador que se pasa al método como parámetro, la
// información correspondiente a la receta que pasa a integrar
// el recetario. La acción es sincronizada para que la zona esté
// bloqueada mientras se realiza la operación. Se utiliza para
// actualizar la información que corresponde a la receta.
public synchronized void setRegistroRecetario( int id,String contacto ) {
// Convertimos la cadena que contiene la información de la receta
// en un array de bytes
byte[] datos = contacto.getBytes();
// Grabamos la información de la receta en la zona RMS
try {
rs.setRecord( id,datos,0,datos.length );
} catch( RecordStoreException e ) {
e.printStackTrace();
}
}

// Crea un nuevo registro en la zona RMS con la información de la
// receta que se pasa como parámetro. Se utiliza para crear
// nuevas recetas en la zona RMS.
public synchronized int nuevoRegistroRecetario( String reg ) {
// Convertimos la cadena que contiene la información de la receta
// en un array de bytes
byte[] datos = reg.getBytes();
// Añadimos la información de la receta en la zona RMS
try {
return( rs.addRecord(datos,0,datos.length) );
} catch ( RecordStoreException e ) {
e.printStackTrace();
}
return( -1 );
}

// Metodo que elimina el registro indicado por el identificador que

```

```
// se pasa como parámetro de la zona RMS
public synchronized void eliminaRegistroRecetario( int id ) {
    // Eliminamos el registro
    try {
        rs.deleteRecord( id );
    } catch( RecordStoreException e ) {
        e.printStackTrace();
    }
}

// Método que devuelve el número de recetas que contiene
// el Recetario
public synchronized RecordEnumeration getNumRegistrosRecetario()
throws RecordStoreNotOpenException {
    return( rs.enumerateRecords( null, null, false ) );
}
}
```

APENDICE C Código de la Aplicación "Agenda Virtual"

A continuación se presenta el código de las Clases Java, JSP y Documentos HTML que hacen posible que la aplicación Agenda Virtual funcione correctamente.

Idex.html

```
<HTML>
<HEAD>
<TITLE> FRAME HORIZONTAL </TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</HEAD>
<body bgcolor="#FFFFFF" text="#000000">
<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 BORDER=NO>
  <TR><TD><IFRAME SRC=cabecera.html NAME="cabecera" WIDTH="900" HEIGHT="104" scrolling="NO"
noreferrer marginwidth="0" marginheight="0" FRAMEBORDER="0"></IFRAME></TD></TR>
  <TR>
    <TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0 BORDER=NO>
      <TD><IFRAME SRC=presentacion.html NAME="escritorio" WIDTH="650" HEIGHT="354"
scrolling="AUTO" marginwidth="0" marginheight="0" FRAMEBORDER="0"></IFRAME></TD>
      <TD><IFRAME SRC=menu.html NAME="menu" scrolling="NO" HEIGHT="354" noreferrer
marginwidth="0" marginheight="0" FRAMEBORDER="0"></IFRAME></TD>
    </TABLE>
  </TR>
  <TR><TD><IFRAME SRC=pie.html NAME="pie" WIDTH="900" HEIGHT="52" scrolling="NO" noreferrer
marginwidth="0" marginheight="0" FRAMEBORDER="0"></IFRAME></TD></TR>
</TABLE>
</body>
</HTML>
```

Cabecera.html

```
<HTML>
<HEAD>
<TITLE>business</TITLE>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<STYLE type=text/css>
BODY { scrollbar-face-color:#000099; scrollbar-shadow-color: #000000; scrollbar-highlight-color: #0000CC; scrollbar-
3dlight-color:#0000FF; scrollbar-darkshadow-color:#000000; scrollbar-track-color: #000000; scrollbar-arrow-color:#ffffff;}
</STYLE>
</HEAD>
<BODY bgColor=#ffffff>
<DIV align=center>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="74%">
  <TR>
    <TD width="2%">&nbsp;</TD>
    <TD width="2%">&nbsp;</TD>
    <TD width="3%">&nbsp;</TD>
    <TD bgColor=#006500 width="2%">&nbsp;</TD>
    <TD width="5%">&nbsp;</TD>
    <TD width="2%">&nbsp;</TD>
    <TD width="2%">&nbsp;</TD>
```

```

<TD bgColor=#ce3000 width="2%">&nbsp;</TD>
<TD width="11%">&nbsp;</TD>
<TD width="2%">&nbsp;</TD>
<TD width="5%">&nbsp;</TD>
<TD bgColor=#ff9a00 width="2%">&nbsp;</TD>
<TD width="7%">&nbsp;</TD>
<TD width="2%">&nbsp;</TD>
<TD width="16%">&nbsp;</TD>
<TD bgColor=#00a200 width="2%">&nbsp;</TD>
<TD width="12%">&nbsp;</TD>
<TD width="2%">&nbsp;</TD>
<TD width="16%">&nbsp;</TD>
<TD bgColor=#006500 width="3%">&nbsp;</TD>
</TR>
</TABLE>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD bgColor=#006600 height=13 rowSpan=2 width="18%">&nbsp;</TD>
<TD bgColor=#ce3000 rowSpan=2 width="8%">&nbsp;</TD>
<TD bgColor=#ff9900 height=13 rowSpan=2 width="15%">&nbsp;</TD>
<TD bgColor=#949231 height=13 rowSpan=2 width="14%">&nbsp;</TD>
<TD bgColor=#00a200 height=13 rowSpan=2 width="44%">&nbsp;</TD>
</TR>
<TR></TR>
</TABLE>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD height=50><DIV align=center class=CompanyName><B>Agenda Virtual</B></DIV></TD>
<TD vAlign=bottom><DIV align=right><B><FONT size=2>Tesis</FONT></B></DIV></TD>
</TR>
<TR>
<TD colspan=2>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="100%">
<TR><TD bgColor=#ffcc00 height=1><DIV align=center><IMG height=1 src="/image/pixel.gif"
width=1></DIV></TD></TR>
</TABLE>
</TD>
</TR>
<TR>
<TD colspan=2>
<DIV align=center><B>
<A href="/presentacion.html" TARGET="escritorio"><FONT
color=#3169a5>Presentación</FONT></A><FONT color=#ff9900 size=3> | </FONT>
<A href="/planteamiento.html" TARGET="escritorio"><FONT
color=#3169a5>Planteamiento</FONT></A><FONT color=#ff9900 size=3> | </FONT>
<A href="/objetivos.html" TARGET="escritorio"><FONT
color=#3169a5>Objetivos</FONT></A><FONT color=#ff9900 size=3> | </FONT>
<A href="/justificacion.html" TARGET="escritorio"><FONT
color=#3169a5>Justificación</FONT></A><FONT color=#ff9900 size=3> | </FONT>
<A href="/contactos.html" TARGET="escritorio"><FONT color=#3169a5>Contactos</FONT></A>
</B></DIV>
</TD>
</TR>
<TR>
<TD colspan=2>
<DIV align=left>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="100%">
<TR><TD bgColor=#ffcc00 height=2><DIV align=center><IMG height=1 src="/image/pixel.gif"
width=1></DIV></TD></TR>
</TABLE>
</DIV>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Presentacion.html

```
<HTML>
<HEAD>
<TITLE>business</TITLE>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<STYLE type=text/css>
</STYLE>
</HEAD>
<BODY bgColor=#ffffff>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD height=300 vAlign=center width=300><BR><IMG height=299 src="/image/back.gif" width=300></TD>
<TD height=300 vAlign=top width="34%"><BR>
<TABLE bgColor=#ffffff border=0 cellPadding=0 cellSpacing=0 width="100%">
<TR>
<TD><DIV align=center><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Trabajo de
Tesis</FONT></B></DIV></TD>
</TR>
</TABLE>
<P><FONT color=#000000>Este proyecto se desarrollo con el objetivo de obtener el titulo de la carrera de Ingenieria
en Computacion de la Universidad Nacional Autonoma de México capus FEST Aragon.<BR></FONT></P>
<TABLE bgColor=#ffffff border=0 cellPadding=0 cellSpacing=0 width="100%">
<TR>
<TD><DIV align=center><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Titulo de la
Tesis</FONT></B></DIV></TD>
</TR>
</TABLE>
<P><FONT color=#000000>Desarrollo de aplicaciones móviles utilizando Java 2 ME como lenguaje de programación
en teléfonos celulares y/o Pocket PC.<BR></FONT></P>
</TD>
</TABLE>
</BODY>
</HTML>
```

Menú.html

```
<HTML>
<HEAD>
<TITLE>business</TITLE>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<STYLE type=text/css>
</STYLE>
</HEAD>
<BODY bgColor=#ffffff>
<TABLE align=right border=0 cellPadding=0 cellSpacing=0 width="75%">
<TD bgColor=#ffcc00 height=393 vAlign=top width=2><IMG height=1 src="/image/pixel.gif" width=1></TD>
<TD bgColor=#ffffff height=393 vAlign=top width=135>
<DIV align=right>
<P align=left><B><FONT color=#ff0000 size=3><BR><DIV align=center>Opciones</DIV></FONT></B></P>
<P align=left><B><FONT color=#ff0000><IMG height=140 src="/image/mylogo.gif" width=140></FONT></B> </P>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="100%">
<TR>
<TD height=25 width="20%"><DIV align=right><IMG height=12 src="/image/point.gif" width=12></DIV></TD>
<TD><DIV align=center><B><A href="/insert.html" TARGET="escritorio"><FONT
color=#ff9c00>Insertar</FONT></A></B></DIV></TD>
</TR>
<TR>
<TD height=25 width="20%"><DIV align=right><IMG height=12 src="/image/point.gif" width=12></DIV></TD>
<TD><DIV align=center><B><A href="/select.jsp" TARGET="escritorio"><FONT
color=#ff9c00>Buscar</FONT></A></B></DIV></TD>
</TR>
```

```

<TR>
  <TD height=25 width="20%"><DIV align=right><IMG height=12 src="/image/point.gif" width=12></DIV></TD>
  <TD><DIV align=center><B><A href="/editar.jsp" TARGET="escritorio"><FONT
color=#ff9c00>Editar</FONT></A></B></DIV></TD>
</TR>
<TR>
  <TD height=25 width="20%"><DIV align=right><IMG height=12 src="/image/point.gif" width=12></DIV></TD>
  <TD><DIV align=center><B><A href="/borrar.jsp" TARGET="escritorio"><FONT
color=#ff9c00>Borrar</FONT></A></B></DIV></TD>
</TR>
<TR>
  <TD height=25 width="20%"><DIV align=right><IMG height=12 src="/image/point.gif" width=12></DIV></TD>
  <TD><DIV align=center><B><A href="/column.htm" TARGET="escritorio"><FONT color=#ff9c00>Column
5</FONT></A></B></DIV></TD>
</TR>
</TABLE>
</DIV>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Pie.html

```

<HTML>
<HEAD>
<TITLE>business</TITLE>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<STYLE type=text/css>
</STYLE>
</HEAD>
<BODY bgColor=#ffffff>
  <DIV align=center>
    <TABLE border=0 cellPadding=0 cellSpacing=0 width="80%">
      <TR bgColor=#ffcf00>
        <TD height=1><DIV align=right><IMG height=1 src="/image/pixel.gif" width=1></DIV></TD>
      </TR>
    </TABLE>
  </DIV>
  <DIV align=center>
    <BR><FONT color=#006400 face="Times New Roman, Times, serif">Trabajo de Tesis de Anguiano
Vargas Jose Luis y Rocha Garfias Oscar Ignacio</FONT><BR>
  </DIV>
  <DIV align=center>
    <TABLE border=0 cellPadding=0 cellSpacing=0 width="90%">
      <TR bgColor=#949231>
        <TD height=2><DIV align=right><IMG height=1 src="/image/pixel.gif" width=1></DIV></TD>
      </TR>
    </TABLE>
  </DIV>
  <DIV align=center>
    <FONT color=#3169a5 face="Arial, Helvetica, sans-serif" size=2>UNAM Fest
Aragon.</FONT><FONT color=#3169a5></FONT><BR>
  </DIV>
  <TABLE border=0 cellPadding=0 cellSpacing=0 width="100%">
    <TR bgColor=#006500>
      <TD height=3><DIV align=right><IMG height=1 src="/image/pixel.gif" width=1></DIV></TD>
    </TR>
  </TABLE>
</DIV>
</BODY>
</HTML>

```

Insert.html

```
<HTML>
<HEAD>
<TITLE>business</TITLE>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<link href="estilo.css" rel="stylesheet" rev="stylesheet" type=text/css>
</HEAD>
<BODY bgColor=#ffffff>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
</TABLE>
</DIV>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD colspan=2 vAlign=top>
<DIV align=center><BR>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%>
<TR>
<TD><DIV align=center><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Nuevo
Contacto</FONT></B></DIV></TD>
</TR>
</TABLE><BR><BR>
<FORM name="param" action="/Tesis2005/servlet/insertar" METHOD="POST">
<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width="90%">
<TR>
<TD><DIV align=left><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Datos del
Contacto:</FONT></B></DIV></TD>
</TR>
</TABLE><BR>

<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="60%">
<TR>
<TD><P><FONT color=#000000>Nombre: </FONT></P></TD><TD><INPUT TYPE="text" NAME="nombre"
class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="appat" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Sexo: </FONT></P></TD><TD><INPUT TYPE="radio" NAME="sexo"
VALUE="M"> Hombre <INPUT TYPE="radio" NAME="sexo" VALUE="F">Mujer</TD>
</TR><TR>
<TD><P><FONT color=#000000>Fecha de Nacimiento: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="fech" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Telefono Casa: </FONT></P></TD><TD><INPUT TYPE="text" NAME="telc"
class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Telefono Celular: </FONT></P></TD><TD><INPUT TYPE="text" NAME="telcl"
class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Telefono Trabajo: </FONT></P></TD><TD><INPUT TYPE="text" NAME="telj"
class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>E-mail: </FONT></P></TD><TD><INPUT TYPE="text" NAME="mail"
class="input"></TD>
</TR>
</TR>
</TABLE><BR><BR>
<input type="submit" value="Aceptar" style="border: 1px solid; font-size: 8pt; font-family:
Verdana, background-color: rgb(233,233,233)">
</FORM>
<BR>
```

```

</DIV>
</TD>
</TR></TABLE>
</BODY>
</HTML>

```

Insertar.java

```

package pkgDb;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;
import com.mysql.jdbc.Driver;
import java.sql.*;

public class insertar extends HttpServlet {
public insertar() {
super();
}
public void destroy() {
super.destroy
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
Connection conn = null;
String url = "jdbc:mysql://localhost:3306/DB_Tesis";
String user = "root";
String pwd = "admin";
String sentencia = "insert into t_usuario(P_NOM, AP_PAT, AP_MAT, SEXO) values (?, ?, ?, ?)";
String sentencia1 = "select ID_USU FROM T_USUARIO WHERE P_NOM = ? AND AP_PAT = ? AND AP_MAT = ?";
String sentencia2 = "insert into t_email(ID_USU,EMAIL) values (?,?)";
String sentencia3 = "insert into t_fecnac(ID_USU,fecha_nac) values (?,?)";
String sentencia4 = "insert into t_telefonos(ID_USU, ID_TEL,NUM_TEL ) values (?, ?, ?)";

int result = 0;
ResultSet result1;
int IdUsu = 0;
String nom1 = request.getParameter("nombre");
String nom2 = request.getParameter("apmat");
String nom3 = request.getParameter("apmat");
String sex = request.getParameter("sexo");
String fech = request.getParameter("fecn");
String telc = request.getParameter("telc");
String telcl = request.getParameter("telcl");
String tell = request.getParameter("tell");
String mail = request.getParameter("mail");

try{
DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn = DriverManager.getConnection(url,user,pwd);
if (conn != null) {
PreparedStatement smt = conn.prepareStatement(sentencia);
smt.setString(1, nom1);
smt.setString(2, nom2);
smt.setString(3, nom3);
smt.setString(4, sex);
result = smt.executeUpdate();
smt.close();
PreparedStatement smt1 = conn.prepareStatement(sentencia1);
smt1.setString(1, nom1);
smt1.setString(2, nom2);
smt1.setString(3, nom3);
result1 = smt1.executeQuery();
result1.next();
IdUsu = result1.getInt("ID_USU");
result1.close();
smt1.close();
PreparedStatement smt2 = conn.prepareStatement(sentencia2);

```

```

smt2.setInt(1, IdUsu);
smt2.setString(2, mail);
result = smt2.executeUpdate();
smt2.close();
PreparedStatement smt3 = conn.prepareStatement(sentencia3);
smt3.setInt(1, IdUsu);
smt3.setString(2, fech);
result = smt3.executeUpdate();
smt3.close();
PreparedStatement smt4 = conn.prepareStatement(sentencia4);
smt4.setInt(1, IdUsu);
smt4.setInt(2, 1);
smt4.setString(3, telc);
result = smt4.executeUpdate();
smt4.setInt(1, IdUsu);
smt4.setInt(2, 2);
smt4.setString(3, telc);
result = smt4.executeUpdate();
smt4.setInt(1, IdUsu);
smt4.setInt(2, 3);
smt4.setString(3, tell);
result = smt4.executeUpdate();
smt4.close();
conn.close();
}
}catch(SQLException e){

```

```

response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");
out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");
out.println("<HTML>");
out.println("<HEAD>");
out.println("<TITLE>business</TITLE>");
out.println("<META content='CutePage 2.0' name=GENERATOR>");
out.println("<META content='text/html; charset=iso-8859-1' http-equiv=Content-Type>");
out.println("<link href='estilo.css' rel='stylesheet' rev='stylesheet' type=text/css");
out.println("");
out.println("</HEAD>");
out.println("<BODY bgColor=#ffffff");
out.println("<TABLE border=0 cellPadding=0 cellSpacing=0 width='75%'>");
out.println("</TABLE>");
out.println("</DIV>");
out.println("<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width='75%'>");
out.println("<TR>");
out.println("<TD colSpan=2 vAlign=top");
out.println("<DIV align=center><BR>");
out.println("<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%>");
out.println("<TR>");
out.println("<TD><DIV align=center><B><FONT color=#999933 face='Times New Roman, Times, serif"
size=3>Nuevo Contacto</FONT></B></DIV></TD>");
out.println("</TR>");
out.println("</TABLE><BR><BR>");
out.println("<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width='90%'>");
out.println("<TR>");
out.println("<TD><DIV align=left><B><FONT color=#999933 face='Times New Roman, Times, serif"
size=3>Datos del Contacto:</B></DIV></TD>");
out.println("</TR>");
out.println("</TABLE><BR>");
out.println("</TABLE>");
out.println("<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width='60%'>");
out.println("<TR>");
out.println("<TD><P><FONT color=#000000>Nombre: </FONT></P></TD><TD>"+ nom1 + "</TD>");
out.println("</TR><TR>");
out.println("<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD>"+ nom2 + "</TD>");
out.println("</TR><TR>");
out.println("<TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD><TD>"+ nom3 + "</TD>");
out.println("</TR><TR>");
out.println("<TD><P><FONT color=#000000>Sexo: </FONT></P></TD><TD>"+ sex + "</TD>");

```



```

out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apatt" class="inputt"></TD>
");
out.println(" </TR><TR>
");
out.println(" <TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" class="inputt"></TD>
");
out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>Sexo: </FONT></P></TD><TD><INPUT TYPE="radi"
NAME="sexo" VALUE="M" > Hombre <INPUT TYPE="radi" NAME="sexo" VALUE="F" >Mujer</TD> ");
out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>Fecha de Nacimiento: </FONT></P></TD><TD><INPUT
TYPE="text" NAME="fecn" class="inputt"></TD>
");
out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>Telefono Casa: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="telc" class="inputt"></TD>
");
out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>Telefono Celular: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="telcl" class="inputt"></TD>
");
out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>Telefono Trabajo: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="teltr" class="inputt"></TD>
");
out.println(" </TR><TR> ");
out.println(" <TD><P><FONT color=#000000>E-mail: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="mail" class="inputt"></TD>
");
out.println(" </TR> ");
out.println(" </TR> ");
out.println(" </TABLE><BR><BR> ");
out.println("<BR><BR> ");
out.println("<input type="submit" value="Aceptar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt;
FONT-FAMILY: Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
");
out.println("<INPUT TYPE="submit" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
");
out.println("<INPUT TYPE="Reset" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
");
out.println(" ");
out.println("</FORM> ");
out.println("<BR> ");
out.println(" </DIV> ");
out.println(" </TD> ");
out.println(" </TR></TABLE> ");
out.println("</BODY>");
out.println("</HTML>");
out.flush();
out.close();
}
public void init() throws ServletException {
// Put your code here
}
}

```

Select.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%@ page import="java.io.IOException" %>
<%@ page import="java.io.PrintWriter" %>
<%@ page import="javax.servlet.ServletException" %>
<%@ page import="javax.servlet.http.HttpServlet" %>
<%@ page import="javax.servlet.http.HttpServletRequest" %>
<%@ page import="javax.servlet.http.HttpServletResponse" %>
<%@ page import="com.mysql.jdbc.Driver" %>
<%@ page import="java.sql.*" %>

<%
String path = request.getContextPath();
String basePath = request.getScheme()+"/"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

```

```

<HTML>
<HEAD>
<title>Agenda 'select.jsp' starting page</title>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<base href="<%=basePath%>">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">

<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->

<link rel="stylesheet" type="text/css" href="estilo.css">
<SCRIPT language=javascript>
    function ventanaSecundaria (URL){
        window.open(URL,"Detalle","width=400,height=450,scrollbars=NO,resizable=NO,false");
        return true;
    }
</SCRIPT>

</head>

<BODY bgColor=#ffffff>

<%
    String nom1 = request.getParameter("nombre");
    if (nom1 == null){
        nom1 = "";
    }
    String nom2 = request.getParameter("apmat" );
    if (nom2 == null){
        nom2 = "";
    }
    String nom3 = request.getParameter("apmat" );
    if (nom3 == null){
        nom3 = "";
    }
%>

<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
</TABLE>
</DIV>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD colSpan=2 vAlign=top>
<DIV align=center><BR>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%>
<TR>
<TD><DIV align=center><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Consulta
Contacto</FONT></B></DIV></TD>
</TR>
</TABLE><BR><BR>

<FORM method = "POST">
<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width="90%">
<TR>
<TD><DIV align=left><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Datos del
Contacto:</FONT></B></DIV></TD>
</TR>
</TABLE><BR>

<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="60%">
<TR>

```

```

<TD><P><FONT color=#000000>Nombre: </FONT></P></TD><TD> <INPUT TYPE="text" NAME="nombre"
value="<%=nom1%>" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" value="<%=nom2%>" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" value="<%=nom3%>" class="input"></TD>
</TR><TR>
</TABLE><BR>
<BR>
<INPUT TYPE="submit" value="Buscar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
<!--
<a href="javascript:ventanaSecundaria('http://www.desarrolloweb.com')"> Pincha en este enlace para abrir la ventana
secundaria</a>
<input type="button" value="Aqui" onClick="ventanaSecundaria ('./select.jsp');">
<INPUT TYPE="Reset" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY: Verdana;
BACKGROUND-COLOR: rgb(233,233,233)">
-->
</FORM>

```

```

<BR>
<% if (!nom1.equals("") || !nom2.equals("") || !nom3.equals("")) {
Connection conn = null;
String url = "jdbc:mysql://localhost:3306/DB_Tesis";
String user = "root";
String pwd = "admin";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a ";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a where a.p_nom
LIKE '%?%?' and a.ap_pat LIKE '%?%?' and a.ap_mat LIKE '%?%?'";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a where a.p_nom=? and a.ap_pat =
? and a.ap_mat = ?";
String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,b.EMAIL,c.num_tel,a.id_usu FROM T_USUARIO a,f_email
b,t_telefonos c where a.id_usu = b.id_usu and a.id_usu = c.id_usu and c.id_tel = 1 and a.p_nom LIKE ? and a.ap_pat
LIKE ? and a.ap_mat LIKE ?";

```

```

ResultSet result;

DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn = DriverManager.getConnection(url,user,pwd);

```

```

%>
<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="100%">
<TR class="celda0">
<TD width="14%"><P><FONT color=#000000>Nombre </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Paterno </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Materno </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>E-mail </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Tel-Casa </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Detalles </FONT></P></TD>
</TR>

```

```

<%
if (conn != null) {
PreparedStatement smt = conn.prepareStatement(sentencia);
nom1 = "%"+ nom1 + "%";
nom2 = "%"+ nom2 + "%";
nom3 = "%"+ nom3 + "%";
smt.setString(1, nom1);
smt.setString(2, nom2);
smt.setString(3, nom3);
result = smt.executeQuery();
String contador = "celda2";
while (result.next()) {

```

```

String nombre1 = result.getString("p_nom");
String nombre2 = result.getString("ap_pat");
String nombre3 = result.getString("ap_mat");
String email1 = result.getString("email");
String tel1 = result.getString("num_tel");

String id = result.getString("id_usu");
String url1 = "ventanaSecundaria (./selectDetalle.jsp?id=" + id + ")/";

if (contador == "celda1"){
    contador = "celda2";
}else {contador = "celda1";}

%>
<TR class=<%= contador%>>
<TD width="14%"><P><FONT color=#000000><%= nombre1%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= nombre2%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= nombre3%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= email1%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= tel1%></FONT></P></TD>
<%--
<TD width="14%"><input type="button" value="Aqui" onClick="ventanaSecundaria (./selectDetalle.jsp?id=<%=
id%>);"></TD>
--%>
<TD width="14%"><input type="button" value="Aqui" onClick="<%= url1%>" ></TD>
</TR>
<%>
conn.close();
}%>
<TR class="celda0">
<TD width="14%"><P></TD>
<TD width="14%"><P></TD>
<TD width="14%"><P></TD>
<TD width="14%"><P></TD>
<TD width="14%"><P></TD>
<TD width="14%"><P></TD>
<TD width="8%"><P></TD>
</TR>
</TABLE><BR><BR>
<% } %>
<BR>
</DIV>
</TD>
</TR></TABLE>
</BODY>
</HTML>

```

SelectDetalle.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%@ page import="java.io.IOException" %>
<%@ page import="java.io.PrintWriter" %>
<%@ page import="javax.servlet.ServletException" %>
<%@ page import="javax.servlet.http.HttpServlet" %>
<%@ page import="javax.servlet.http.HttpServletRequest" %>
<%@ page import="javax.servlet.http.HttpServletResponse" %>
<%@ page import="com.mysql.jdbc.Driver" %>
<%@ page import="java.sql.*" %>

<%
String path = request.getContextPath();
String basePath = request.getScheme()+"/"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<HTML>
<HEAD>

```

```

<title>Agenda 'selectDetalle.jsp' starting page</title>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<base href="<%=basePath%>">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">

<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->

<link rel="stylesheet" type="text/css" href="estilo.css">
<SCRIPT language=javascript>
    function ventanaSecundaria (URL){
        window.open(URL,"Detalle","width=400,height=450,scrollbars=NO,resizable=NO,false");
        return true;
    }
</SCRIPT>

</head>

<BODY bgColor=#ffffff>

<%
        String id = request.getParameter("id");
        if (id == null){
            id = "";
        }
%>

<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
</TABLE>
</DIV>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD colSpan=2 vAlign=top>
<DIV align=center><BR>

<FORM method = "POST">
<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width="90%">
<TR>
<TD><DIV align=left><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Detalles del
Contacto:</FONT></B></DIV></TD>
</TR>
</TABLE><BR>

<%if (!id.equals("")){
Connection conn = null;
        String url = "jdbc:mysql://localhost:3306/DB_Tesis";
        String user = "root";
        String pwd = "admin";
        String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,a.sexo,b.fecha_nac,c.num_tel tcasa,d.num_tel
tcel,e.num_tel tofi,f.email FROM t_usuario a,t_fecnac b,t_telefonos c,t_telefonos d,t_telefonos e,t_email f where
a.id_usu = b.id_usu and a.id_usu = c.id_usu and a.id_usu = d.id_usu and a.id_usu = e.id_usu and a.id_usu =
f.id_usu and c.id_tel = 1 and d.id_tel = 2 and e.id_tel = 3 and a.id_usu like ?";

ResultSet result;

        DriverManager.registerDriver(new com.mysql.jdbc.Driver());
        conn = DriverManager.getConnection(url,user,pwd);

        if (conn != null) {
            PreparedStatement smt = conn.prepareStatement(sentencia);
            smt.setString(1, id);
            result = smt.executeQuery();

```

```

result.next();

String nombre1 = result.getString("p_nom");
String nombre2 = result.getString("ap_pat");
String nombre3 = result.getString("ap_mat");
String email1 = result.getString("email");
String sex1 = result.getString("sexo");
String nac1 = result.getString("techa_nac");

```

```

String tel1 = result.getString("tcasa");
String tel2 = result.getString("tcel");
String tel3 = result.getString("tofi");

```

%>

```

<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="60%">
<TR>
<TD><P><FONT color=#000000>Nombre: </FONT></P></TD> <TD><P><FONT color=#000000><%=
nombre1%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD> <TD><P><FONT color=#000000><%=
nombre2%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD> <TD><P><FONT color=#000000><%=
nombre3%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Sexo: </FONT></P></TD> <TD><P><FONT color=#000000><%=
sex1%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Fecha de Nacimiento: </FONT></P></TD> <TD><P><FONT
color=#000000><%= nac1%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Telefono Casa: </FONT></P></TD> <TD><P><FONT color=#000000><%=
tel1%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Telefono Celular: </FONT></P></TD> <TD><P><FONT color=#000000><%=
tel2%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>Telefono Trabajo: </FONT></P></TD> <TD><P><FONT color=#000000><%=
tel3%> </FONT></P></TD>
</TR><TR>
<TD><P><FONT color=#000000>E-mail: </FONT></P></TD> <TD><P><FONT color=#000000><%=
email1%> </FONT></P></TD>
</TR>
</TR>
</TABLE><BR><BR>

```

```

<%conn.close();
}}%>

```

```
<BR>
```

```

</FORM>
</BODY>
</HTML>

```

Editar.jsp

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%@ page import="java.io.IOException" %>
<%@ page import="java.io.PrintWriter" %>
<%@ page import="javax.servlet.ServletException" %>
<%@ page import="javax.servlet.http.HttpServlet" %>
<%@ page import="javax.servlet.http.HttpServletRequest" %>
<%@ page import="javax.servlet.http.HttpServletResponse" %>
<%@ page import="com.mysql.jdbc.Driver" %>
<%@ page import="java.sql.*" %>

<%
String path = request.getContextPath();
String basePath = request.getScheme()+"/"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<HTML>
<HEAD>
<title>Agenda 'editar.jsp' starting page</title>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<base href="<%=basePath%>">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">

<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->

<link rel="stylesheet" type="text/css" href="estilo.css">
<SCRIPT language=javascript>
function ventanaSecundaria (URL){
window.open(URL,"Detalle","width=400,height=450,scrollbars=NO,resizable=NO,false");
return true;
}
</SCRIPT>

</head>

<BODY bgColor=#ffffff>

<%
String nom1 = request.getParameter("nombre");
if (nom1 == null){
nom1 = "";
}
String nom2 = request.getParameter("apmat" );
if (nom2 == null){
nom2 = "";
}
String nom3 = request.getParameter("apmat" );
if (nom3 == null){
nom3 = "";
}
%>

<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
</TABLE>
</DIV>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD colSpan=2 vAlign=top>
<DIV align=center><BR>
```

```

<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%>
<TR>
<TD><DIV align=center><B><FONT color=#999933 face="Times New Roman, Times. serif" size=3>Editar
Contacto</FONT></B></DIV></TD>
</TR>
</TABLE><BR><BR>
<FORM method = "POST">
<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width="90%">
<TR>
<TD><DIV align=left><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Datos del
Contacto:</FONT></B></DIV></TD>
</TR>
</TABLE><BR>
<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="60%">
<TR>
<TD><P><FONT color=#000000>Nombre: </FONT></P></TD><TD><INPUT TYPE="text" NAME="nombre"
value="<%=nom1%>" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" value="<%=nom2%>" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" value="<%=nom3%>" class="input"></TD>
</TR><TR>
</TABLE><BR>
<BR>
<INPUT TYPE="submit" value="Buscar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
<!--
<a href="javascript:ventanaSecundaria('http://www.desarrolloweb.com')"> Pincha en este enlace para abrir la ventana
secundaria</a>
<input type="button" value="Aqui" onClick="ventanaSecundaria ('./select.jsp);">
<INPUT TYPE="Reset" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY: Verdana;
BACKGROUND-COLOR: rgb(233,233,233)">
-->
</FORM>
<BR>
<%if (!nom1.equals("") || !nom2.equals("") || !nom3.equals("")) {
Connection conn = null;
String url = "jdbc:mysql://localhost:3306/DB_Tesis";
String user = "root";
String pwd = "admin";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a ";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a where a.p_nom
LIKE '%?%' and a.ap_pat LIKE '%?%' and a.ap_mat LIKE '%?%'";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a where a.p_nom=? and a.ap_pat =
? and a.ap_mat = ?";
String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,b.EMAIL,c.num_tel,a.id_usu FROM T_USUARIO a,t_email
b,t_telefonos c where a.id_usu = b.id_usu and a.id_usu = c.id_usu and c.id_tel = 1 and a.p_nom LIKE ? and a.ap_pat
LIKE ? and a.ap_mat LIKE ?";

ResultSet result;

DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn = DriverManager.getConnection(url,user,pwd);
%>
<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="100%">
<TR class="celda0">
<TD width="14%"><P><FONT color=#000000>Nombre </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Paterno </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Materno </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>E-mail </FONT></P></TD>

```

```
<TD width="14%"><P><FONT color=#000000>Tel-Casa </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Editar </FONT></P></TD>
</TR>
```

```
<%
```

```
if (conn != null) {
    PreparedStatement smt = conn.prepareStatement(sentencia);
    nom1 = "%" + nom1 + "%";
    nom2 = "%" + nom2 + "%";
    nom3 = "%" + nom3 + "%";
    smt.setString(1, nom1);
    smt.setString(2, nom2);
    smt.setString(3, nom3);
    result = smt.executeQuery();
    String contador = "celda2";
    while (result.next()) {

        String nombre1 = result.getString("p_nom");
        String nombre2 = result.getString("ap_pat" );
        String nombre3 = result.getString("ap_mat" );
        String email1 = result.getString("email" );
        String tel1 = result.getString("num_tel" );

        String id = result.getString("id_usu" );
        String url1 = "ventanaSecundaria (\"./editarDetalle.jsp?id=" + id + "\");";

        if (contador == "celda1"){
            contador = "celda2";
        }else {contador = "celda1";}
    }
}
```

```
%>
<TR class=<%= contador%>>
<TD width="14%"><P><FONT color=#000000><%= nombre1%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= nombre2%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= nombre3%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= email1%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= tel1%></FONT></P></TD>
<%=
<TD width="14%"><input type="button" value="Editar" onClick="ventanaSecundaria ('./selectDetalle.jsp?id=<%=
id%>');"></TD>
--%>
```

```
<TD width="14%"><input type="button" value="Editar" onClick="<%= url1%>" ></TD>
</TR>
```

```
<%
conn.close();
}%>
```

```
<TR class="celda0">
<TD width="14%"><P></TD>
<TD width="8%"><P></TD>
</TR>
```

```
</TABLE><BR><BR>
```

```
<% } %>
<BR>
```

```
</DIV>
</TD>
</TR></TABLE>
</BODY>
</HTML>
```

EditarDetalle.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%@ page import="java.io.IOException" %>
<%@ page import="java.io.PrintWriter" %>
<%@ page import="javax.servlet.ServletException" %>
<%@ page import="javax.servlet.http.HttpServlet" %>
<%@ page import="javax.servlet.http.HttpServletRequest" %>
<%@ page import="javax.servlet.http.HttpServletResponse" %>
<%@ page import="com.mysql.jdbc.Driver" %>
<%@ page import="java.sql.*" %>

<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<HTML>
<HEAD>
<title>Agenda 'editarDetalle.jsp' starting page</title>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<base href="<%=basePath%>">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">

<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->

<link rel="stylesheet" type="text/css" href="estilo.css">
<SCRIPT language=javascript>
    function ventanaSecundaria (URL){
        window.open(URL,"Detalle","width=400,height=450,scrollbars=NO,resizable=NO,false");
        return true;
    }
</SCRIPT>

</head>

<BODY bgColor=#ffffff>

<%
    String id = request.getParameter("id");
    if (id == null){
        id = "";
    }
    String editar = request.getParameter("editar");
    if (editar == null){
        editar = "";
    }

    String ednom1 = request.getParameter("ednombre1");
    String ednom2 = request.getParameter("ednombre2");
    String ednom3 = request.getParameter("ednombre3");
    String edsexo = request.getParameter("edsexo");
    String edfech = request.getParameter("ednac");
    String edtelc = request.getParameter("edtelc1");
    String edtelci = request.getParameter("edtelc2");
    String edtelit = request.getParameter("edtelc3");
    String edmail = request.getParameter("edmail");
    if (ednom1 == null){
        ednom1 = "";
    }
    if (ednom2 == null){
        ednom2 = "";
    }
%>

```

```

    }
    if (ednom3 == null){
    ednom3 = "";
    }
    if (edsex == null){
    edsex = "";
    }
    if (edfech == null){
    edfech = "";
    }
    if (edtelc == null){
    edtelc = "";
    }
    if (edtelcl == null){
    edtelcl = "";
    }
    if (edteilt == null){
    edteilt = "";
    }
    }
    if (edmail == null){
    edmail = "";
    }
}

```

%>

```

<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
</TABLE>

```

</DIV>

```

<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>

```

```

<TD colSpan=2 vAlign=top>
<DIV align=center><BR>

```

```

<FORM method = "POST" >

```

```

<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width="90%">
<TR>

```

```

<TD><DIV align=left><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Editor
Contacto:</FONT></B></DIV></TD>

```

```

</TR>

```

```

</TABLE><BR>

```

```

<%if (!id.equals("")) {

```

```

Connection conn = null;

```

```

Connection conn1 = null;

```

```

String url = "jdbc:mysql://localhost:3306/DB_Tesis";

```

```

String user = "root";

```

```

String pwd = "admin";

```

%>

```

<%if (!id.equals("")) &&

```

```

//&& !editar.equals("")

```

```

(!ednom1.equals("")) ||

```

```

!ednom2.equals("") ||

```

```

!ednom3.equals("") ||

```

```

!edsex.equals("") ||

```

```

!edfech.equals("") ||

```

```

!edtelc.equals("") ||

```

```

!edtelcl.equals("") ||

```

```

!edteilt.equals("") ||

```

```

!edmail.equals("") )

```

}

```

String sentencia2 = "update t_usuario set p_nom=? ,ap_pat=? ,ap_mat=? ,sexo=? where id_usu =? ";

```

```

String sentencia3 = "update t_telefonos set num_tel=? where id_tel=1 and id_usu =?";

```

```

String sentencia4 = "update t_telefonos set num_tel=? where id_tel=2 and id_usu =?";

```

```

String sentencia5 = "update t_telefonos set num_tel=? where id_tel=3 and id_usu =?";

```

```

String sentencia6 = "update t_fecnac b set fecha_nac=? where id_usu =?";

```

```

String sentencia7 = "update t_email set email=? where id_usu =?";

//String ednom1 = request.getParameter("ednombre1");
//String ednom2 = request.getParameter("ednombre2" );
//String ednom3 = request.getParameter("ednombre3" );
//String edsex = request.getParameter("edsexo" );
//String edfech = request.getParameter("ednac" );
//String edtelc = request.getParameter("edtelc1" );
//String edtelcl = request.getParameter("edtelc2" );
//String edtelte = request.getParameter("edtelc3" );
//String edmail = request.getParameter("edmail" );

int result = 0;

DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn1 = DriverManager.getConnection(url,user,pwd);
if (conn1 != null) {

    PreparedStatement smt2 = conn1.prepareStatement(sentencia2);

    smt2.setString(1, ednom1);
    smt2.setString(2, ednom2);
    smt2.setString(3, ednom3);
    smt2.setString(4, edsex);
    smt2.setString(5, id);

result = smt2.executeUpdate();
    smt2.close();

    PreparedStatement smt3 = conn1.prepareStatement(sentencia3);

    smt3.setString(1, edtelc);
    smt3.setString(2, id);

    result = smt3.executeUpdate();
    smt3.close();

    PreparedStatement smt4 = conn1.prepareStatement(sentencia4);

    smt4.setString(1, edtelcl);
    smt4.setString(2, id);

    result = smt4.executeUpdate();
    smt4.close();

    PreparedStatement smt5 = conn1.prepareStatement(sentencia5);

    smt5.setString(1, edtelte);
    smt5.setString(2, id);

    result = smt5.executeUpdate();
    smt5.close();

    PreparedStatement smt6 = conn1.prepareStatement(sentencia6);

    //smt6.setString(1, edfech);
    //smt6.setString(2, id);

    //result = smt6.executeUpdate();
    //smt6.close();

    PreparedStatement smt7 = conn1.prepareStatement(sentencia7);

    smt7.setString(1, edmail);
    smt7.setString(2, id);

    result = smt7.executeUpdate();
    smt7.close();
}

```

```

    conn1.close();
}

}
%>

<%
String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,a.sexo,b.fecha_nac,c.num_tel tcasa,d.num_tel
tcel,e.num_tel tofi, f.email FROM t_usuario a,t_fecnac b, t_telefonos c, t_telefonos d, t_telefonos e,t_email f where
a.id_usu = b.id_usu and a.id_usu = c.id_usu and a.id_usu = d.id_usu and a.id_usu = e.id_usu and a.id_usu =
f.id_usu and c.id_tel = 1 and d.id_tel = 2 and e.id_tel = 3 and a.id_usu like ?";

```

```

ResultSet result;

DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn = DriverManager.getConnection(url,user,pwd);

if (conn != null) {
    PreparedStatement smt = conn.prepareStatement(sentencia);
    smt.setString(1, id);
    result = smt.executeQuery();

    result.next();

    String nombre1 = result.getString("p_nom");
    String nombre2 = result.getString("ap_pat");
    String nombre3 = result.getString("ap_mat");
    String email1 = result.getString("email");
    String sex1 = result.getString("sexo");
    String nac1 = result.getString("fecha_nac");

    String tel1 = result.getString("tcasa");

    String tel2 = result.getString("tcel");

    String tel3 = result.getString("tofi");
    editar = "si";
    request.setAttribute("editar", editar);
    request.setAttribute("id", id);

    //request.setAttribute("ednom1", nombre1);
    //request.setAttribute("ednom2", nombre2);
    //request.setAttribute("ednom3", nombre3);
    //request.setAttribute("edsex", sex1);
    //request.setAttribute("edfech", nac1);
    //request.setAttribute("edtelc", tel1);
    //request.setAttribute("edtelcl", tel2);
    //request.setAttribute("edteft", tel3);
    //request.setAttribute("edmail", email1);
}

```

```

%>

<TABLE align=center border=0 cellpadding=4 cellspacing=0 width="60%">
<TR>
<TD><P><FONT color=#000000>Nombre: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="ednombre1" value="<%= nombre1%>" class="input"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="ednombre2" value="<%= nombre2%>" class="input"></TD>

```

```

</TR><TR>
  <TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD> <TD><INPUT TYPE="text"
NAME="ednombre3" value="<%= nombre3%>" class="input"></TD>
</TR><TR>
  <TD><P><FONT color=#000000>Sexo: </FONT></P></TD> <TD><INPUT TYPE="text" NAME="edsexo"
value="<%= sex1%>" class="input"></TD>
</TR><TR>
  <TD><P><FONT color=#000000>Fecha de Nacimiento: </FONT></P></TD> <TD><INPUT TYPE="text"
NAME="ednac" value="<%= nac1%>" class="input"></TD>
</TR><TR>
  <TD><P><FONT color=#000000>Telefono Casa: </FONT></P></TD> <TD><INPUT TYPE="text"
NAME="edtelc1" value="<%= tel1%>" class="input"></TD>
</TR><TR>
  <TD><P><FONT color=#000000>Telefono Celular: </FONT></P></TD> <TD><INPUT TYPE="text"
NAME="edtelc2" value="<%= tel2%>" class="input"></TD>
</TR><TR>
  <TD><P><FONT color=#000000>Telefono Trabajo: </FONT></P></TD> <TD><INPUT TYPE="text"
NAME="edtelc3" value="<%= tel3%>" class="input"></TD>
</TR><TR>
  <TD><P><FONT color=#000000>E-mail: </FONT></P></TD> <TD><INPUT TYPE="text" NAME="edmail"
value="<%= email1%>" class="input"></TD>
</TR>
</TR>
</TABLE><BR><BR>

```

```

<%
conn.close();
}}>
<BR>
<INPUT TYPE="submit" value="Aceptar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
<!--
<INPUT TYPE="submit" value="Buscar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">

<a href="javascript:ventanaSecundaria('http://www.desarrolloweb.com')"> Pincha en este enlace para abrir la ventana
secundaria</a>
<input type="button" value="Aqui" onClick="ventanaSecundaria ('./select.jsp');">
<INPUT TYPE="Reset" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY: Verdana;
BACKGROUND-COLOR: rgb(233,233,233)">
-->
</FORM>
</BODY>
</HTML>

```

Borrar.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%@ page import="java.io.IOException" %>
<%@ page import="java.io.PrintWriter" %>
<%@ page import="javax.servlet.ServletException" %>
<%@ page import="javax.servlet.http.HttpServlet" %>
<%@ page import="javax.servlet.http.HttpServletRequest" %>
<%@ page import="javax.servlet.http.HttpServletResponse" %>
<%@ page import="com.mysql.jdbc.Driver" %>
<%@ page import="java.sql.*" %>

<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<HTML>

```

```

<HEAD>
<title>Agenda 'borrar.jsp' starting page</title>
<META content="CutePage 2.0" name=GENERATOR>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<base href="<%=basePath%>">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">

<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->

<link rel="stylesheet" type="text/css" href="estilo.css">
<SCRIPT language=javascript>
function ventanaSecundaria (URL){
    window.open(URL,"Detalle","width=400,height=450,scrollbars=NO,resizable=NO,false");
    //window.location.href = URL;
}

    return true;
}
</SCRIPT>

</head>

<BODY bgColor=#ffffff>

<%
    String nom1 = request.getParameter("nombre");
    if (nom1 == null){
        nom1 = "";
    }
    String nom2 = request.getParameter("appal" );
    if (nom2 == null){
        nom2 = "";
    }
    String nom3 = request.getParameter("apmat" );
    if (nom3 == null){
        nom3 = "";
    }
    String var_id = request.getParameter("id" );
    if (var_id == null){
        var_id = "";
    }
%>

<TABLE border=0 cellPadding=0 cellSpacing=0 width="75%">
</TABLE>
</DIV>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="75%">
<TR>
<TD colspan=2 vAlign=top>
<DIV align=center><BR>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%>
<TR>
<TD><DIV align=center><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Borrar
Contacto</FONT></B></DIV></TD>
</TR>
</TABLE><BR><BR>

<FORM method = "POST">
<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width="90%">
<TR>
<TD><DIV align=left><B><FONT color=#999933 face="Times New Roman, Times, serif" size=3>Datos del
Contacto:</FONT></B></DIV></TD>
</TR>

```

```

</TABLE><BR>
<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="60%">
<TR>
<TD><P><FONT color=#000000>Nombre: </FONT></P></TD><TD><INPUT TYPE="text" NAME="nombre"
value="<%=nom1%>" class="inputt"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Paterno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="appat" value="<%=nom2%>" class="inputt"></TD>
</TR><TR>
<TD><P><FONT color=#000000>Apellido Materno: </FONT></P></TD><TD><INPUT TYPE="text"
NAME="apmat" value="<%=nom3%>" class="inputt"></TD>
</TR><TR>
</TABLE><BR>
<BR>
<INPUT TYPE="submit" value="Buscar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY:
Verdana; BACKGROUND-COLOR: rgb(233,233,233)">
<!--
<a href="javascript:ventanaSecundaria('http://www.desarrolloweb.com')"> Pincha en este enlace para abrir la ventana
secundaria</a>
<input type="button" value="Aqui" onClick="ventanaSecundaria ('./select.jsp');">
<INPUT TYPE="Reset" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt; FONT-FAMILY: Verdana;
BACKGROUND-COLOR: rgb(233,233,233)">
-->
</FORM>
<BR>
<%if (!nom1.equals("") || !nom2.equals("") || !nom3.equals("")) {
Connection conn = null;
String url = "jdbc:mysql://localhost:3306/DB_Tesis";
String user = "root";
String pwd = "admin";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a ";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a where a.p_nom
LIKE '%?%?' and a.ap_pat LIKE '%?%?' and a.ap_mat LIKE '%?%?'";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat FROM T_USUARIO a where a.p_nom =? and a.ap_pat =
? and a.ap_mat = ?";
String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,b.EMAIL,c.num_tel,a.id_usu FROM T_USUARIO a,t_email
b,t_telefonos c where a.id_usu = b.id_usu and a.id_usu = c.id_usu and c.id_tel = 1 and a.p_nom LIKE ? and a.ap_pat
LIKE ? and a.ap_mat LIKE ?";

ResultSet result;

DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn = DriverManager.getConnection(url,user,pwd);

%>
<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width="100%">
<TR class="celda0">
<TD width="14%"><P><FONT color=#000000>Nombre </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Paterno </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Materno </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>E-mail </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Tel-Casa </FONT></P></TD>
<TD width="14%"><P><FONT color=#000000>Borrar </FONT></P></TD>
</TR>
<%
if (conn != null) {
if (!var_id.equals("")){
int result1 = 0;
int result2 = 0;
int result3 = 0;
int result4 = 0;
String sentencia1 = "delete from t_email where id_usu = ?";

```

```

        PreparedStatement smt1 = conn.prepareStatement(sentencia1);
        smt1.setString(1, var_id);
result1 = smt1.executeUpdate();
        smt1.close();

        String sentencia2 = "delete from t_fecnac where id_usu = ?";

        PreparedStatement smt2 = conn.prepareStatement(sentencia2);
        smt2.setString(1, var_id);
result2 = smt2.executeUpdate();
        smt2.close();

        String sentencia3 = "delete from t_telefonos where id_usu = ";

        PreparedStatement smt3 = conn.prepareStatement(sentencia3);
        smt3.setString(1, var_id);
result3 = smt3.executeUpdate();
        smt3.close();

        String sentencia4 = "delete from t_usuario where id_usu = ?";

        PreparedStatement smt4 = conn.prepareStatement(sentencia4);
        smt4.setString(1, var_id);
result4 = smt4.executeUpdate();
        smt4.close();

    }else {

        PreparedStatement smt = conn.prepareStatement(sentencia);

        nom1 = "%" + nom1 + "%";
        nom2 = "%" + nom2 + "%";
        nom3 = "%" + nom3 + "%";

        smt.setString(1, nom1);
        smt.setString(2, nom2);
        smt.setString(3, nom3);
        result = smt.executeQuery();
        String contador = "celda2";
        while (result.next()) {

            String nombre1 = result.getString("p_nom");
            String nombre2 = result.getString("ap_pat");
            String nombre3 = result.getString("ap_mat");
            String email1 = result.getString("email");
            String tel1 = result.getString("num_tel");

            String id = result.getString("id_usu");
            //String url1 = "ventanaSecundaria (/editarDetalle.jsp?id=" + id + "%)";

            //String url1 = "/borrar.jsp?id=" + id ;
            //String url1 = "ventanaSecundaria (/Tesis2005/servlet/delete?id=" + id + "%)";
            String url1 = "/Tesis2005/servlet/delete?id=" + id ;
            if (contador == "celda1"){
                contador = "celda2";
            }else {contador = "celda1";}

        }

    }

}

%>
<TR class=<%= contador%>>
<TD width="14%"><P><FONT color=#000000><%= nombre1%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= nombre2%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= nombre3%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= email1%></FONT></P></TD>
<TD width="14%"><P><FONT color=#000000><%= tel1%></FONT></P></TD>
<%=
<TD width="14%"><input type="button" value="Editar" onClick="ventanaSecundaria (/selectDetalle.jsp?id=<%=
id%>);"></TD>
<TD width="14%"><input type="button" value="Borrar" onClick="<%= url1%> "></TD>

```

```

-%>
<FORM name="param" action="<%= url1%>" METHOD="POST">
<TD width="14%"><input type="submit" value="Borrar" Style="BORDER: rgb(128,128,128) 1px solid; FONT-SIZE: 8pt;
FONT-FAMILY: Verdana, BACKGROUND-COLOR: rgb(233,233,233)"> </TD>
</form>
</TR>

<%}}
conn.close();
}%>

<TR class="celda0">
<TD width="14%"><P></TD>
<TD width="8%"><P></TD>
</TR>

</TABLE><BR><BR>
<% } %>
<BR>
</DIV>
</TD>
</TR></TABLE>
</BODY>
</HTML>

```

Delete.java

```

package pkgDb;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.mysql.jdbc.Driver;
import java.sql.*;

public class delete extends HttpServlet {

/**
 * Constructor of the object.
 */
public delete() {
super();
}

/**
 * Destruction of the servlet. <br>
 */
public void destroy() {
super.destroy(); // Just puts "destroy" string in log
// Put your code here
}

```

```

public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    /*****/
    String var_id = request.getParameter("id" );
    if (var_id == null){
        var_id = "";
    }

    Connection conn = null;
    String url = "jdbc:mysql://localhost:3306/DB_Tesis";
    String user = "root";
    String pwd = "admin";
    //String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,b.EMAIL,c.num_tel,a.id_usu FROM T_USUARIO a,t_email
    b,t_telefonos c where a.id_usu = b.id_usu and a.id_usu = c.id_usu and c.id_tel = 1 and a.p_nom LIKE ? and a.ap_pat
    LIKE ? and a.ap_mat LIKE ?";

    ResultSet result;

    try{

        DriverManager.registerDriver(new com.mysql.jdbc.Driver());
        conn = DriverManager.getConnection(url,user,pwd);

        if (conn != null) {
            if (!var_id.equals("")){
                int result1 = 0;
                int result2 = 0;
                int result3 = 0;
                int result4 = 0;
                String sentencia1 = "delete from t_email where id_usu = ?";

                PreparedStatement smt1 = conn.prepareStatement(sentencia1);
                smt1.setString(1, var_id);
                result1 = smt1.executeUpdate();
                smt1.close();

                String sentencia2 = "delete from t_fecnac where id_usu = ?";

                PreparedStatement smt2 = conn.prepareStatement(sentencia2);
                smt2.setString(1, var_id);
                result2 = smt2.executeUpdate();
                smt2.close();

                String sentencia3 = "delete from t_telefonos where id_usu =";

                PreparedStatement smt3 = conn.prepareStatement(sentencia3);
                smt3.setString(1, var_id);
                result3 = smt3.executeUpdate();
                smt3.close();

                String sentencia4 = "delete from t_usuario where id_usu = ?";

                PreparedStatement smt4 = conn.prepareStatement(sentencia4);
                smt4.setString(1, var_id);
                result4 = smt4.executeUpdate();
                smt4.close();

                //Variables en lel redireccionamiento
                //String v_f = null;
                //request.setAttribute("f", v_f );

                //RequestDispatcher view = request.getRequestDispatcher("/borrar.jsp");
                // view.forward(request,response);

                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
            }
        }
    }
}

```

```

out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");
out.println("<HTML>");
out.println("<HEAD>");
out.println("<TITLE>business</TITLE>");
out.println("<META content='CutePage 2.0*' name=GENERATOR>");
out.println("<META content='text/html; charset=iso-8859-1' http-equiv=Content-Type>");
out.println("<link href='estilo.css' rel='stylesheet' rev='stylesheet' type=text/css");
out.println(">");
out.println("</HEAD>");
out.println("<BODY bgColor=#ffffff>");
out.println("<TABLE border=0 cellPadding=0 cellSpacing=0 width='75%'>");
out.println("<TABLE>");
out.println("</DIV>");
out.println("<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width='75%'>");
out.println("<TR>");
out.println("<TD colSpan=2 vAlign=top>");
out.println("<DIV align=center><BR>");
out.println("<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%>");
out.println("<TR>");
out.println("<TD><DIV align=center><B><FONT color=#999933 face='Times New Roman, Times, serif"");
out.println("size=3>Borrar Contacto</FONT></B></DIV></TD>");
out.println("</TR>");
out.println("</TABLE><BR>");
out.println("<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width='90%'>");
out.println("<TR>");
out.println("<TD><DIV align=left><B><FONT color=#999933 face='Times New Roman, Times, serif"");
out.println("size=3>Los Datos del Contacto:</FONT></B></DIV></TD>");
out.println("</TR>");
out.println("</TABLE><BR>");
out.println("<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width='60%'>");
out.println("<TR>");
out.println("<TD><P><FONT color=#000000>Fueron borrados correctamente </FONT></P></TD>");
out.println("</TR>");
out.println("</TABLE><BR><BR>");
out.println("<BR><BR>");
out.println("</DIV>");
out.println("</TD>");
out.println("</TR></TABLE>");
out.println("</BODY>");
out.println("</HTML>");

```

```

return;
}
RequestDispatcher view = request.getRequestDispatcher("/borrar.jsp");
view.forward(request,response);
return;
}
}catch(SQLException e){

```

...../

```

response.setContentType("text/html");
PrintWriter out = response.getWriter();

```

```

out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");
out.println("<HTML>");
out.println("<HEAD><TITLE>Error en el Delete</TITLE></HEAD>");
out.println("<BODY>");
out.print(" This is ");
out.print(this.getClass());
out.println(", using the GET method, se presento un error en el borrado ");
out.println("</BODY>");
out.println("</HTML>");
out.flush();
out.close();
}
}

```

```

public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

```

```

/*****/
String var_id = request.getParameter("id" );
if (var_id == null){
var_id = "";
}

```

```

Connection conn = null;
String url = "jdbc:mysql://localhost:3306/DB_Tesis";
String user = "root";
String pwd = "admin";
//String sentencia = "SELECT a.p_nom,a.ap_pat,a.ap_mat,b.EMAIL,c.num_tel,a.id_usu FROM T_USUARIO a,t_email
b,t_telefonos c where a.id_usu = b.id_usu and a.id_usu = c.id_usu and c.id_tel = 1 and a.p_nom LIKE ? and a.ap_pat
LIKE ? and a.ap_mat LIKE ?";

```

```

ResultSet result;

```

```

try{
DriverManager.registerDriver(new com.mysql.jdbc.Driver());
conn = DriverManager.getConnection(url,user,pwd);

if (conn != null) {
if (!var_id.equals("")){
int result1 = 0;
int result2 = 0;
int result3 = 0;
int result4 = 0;
String sentencia1 = "delete from t_email where id_usu = ?";

PreparedStatement smt1 = conn.prepareStatement(sentencia1);
smt1.setString(1, var_id);
result1 = smt1.executeUpdate();
smt1.close();
}

```

```

String sentencia2 = "delete from t_fecnac where id_usu = ?";

PreparedStatement smt2 = conn.prepareStatement(sentencia2);
smt2.setString(1, var_id);
result2 = smt2.executeUpdate();
smt2.close();

```

```

String sentencia3 = "delete from t_telefonos where id_usu =";

PreparedStatement smt3 = conn.prepareStatement(sentencia3);
smt3.setString(1, var_id);
result3 = smt3.executeUpdate();
smt3.close();

```

```

String sentencia4 = "delete from t_usuario where id_usu = ?";

```

```

PreparedStatement smt4 = conn.prepareStatement(sentencia4);
smt4.setString(1, var_id);
result4 = smt4.executeUpdate();
smt4.close();

*/

//Variables en lei redireccionamiento
//String v_f = null;
//request.setAttribute("f", v_f);

//RequestDispatcher view = request.getRequestDispatcher("/borrar.jsp");
//view.forward(request,response);

response.setContentType("text/html");
PrintWriter out = response.getWriter();

out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"");
out.println("<HTML");
out.println("<HEAD");
out.println("<TITLE>business</TITLE>");
out.println("<META content='CutePage 2.0' name=GENERATOR");
out.println("<META content='text/html; charset=iso-8859-1' http-equiv=Content-Type");
out.println("<link href='estilo.css' rel='stylesheet' rev='stylesheet' type=text/css");
out.println("");
out.println("</HEAD");
out.println("<BODY bgColor=#ffffff");
out.println("<TABLE border=0 cellPadding=0 cellSpacing=0 width='75%'");
out.println("</TABLE");
out.println("</DIV");
out.println("<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width='75%'");
out.println("<TR");
out.println("<TD colSpan=2 vAlign=top");
out.println("<DIV align=center><BR");
out.println("<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=70%");
out.println("<TR");
out.println("<TD><DIV align=center><B><FONT color=#999933 face='Times New Roman, Times, serif"");
out.println("<size=3>Nuevo Contacto</FONT></B></DIV></TD");
out.println("</TR");
out.println("</TABLE><BR><BR");
out.println("<TABLE align=CENTER border=0 cellPadding=0 cellSpacing=0 width='90%'");
out.println("<TR");
out.println("<TD><DIV align=left><B><FONT color=#999933 face='Times New Roman, Times, serif"");
out.println("<size=3>Datos del Contacto:</FONT></B></DIV></TD");
out.println("</TR");
out.println("</TABLE><BR");
out.println("<TABLE align=center border=0 cellPadding=4 cellSpacing=0 width='60%'");
out.println("<TR");
out.println("<TD><P><FONT color=#000000>Registro: Borrado correctamente </FONT></P></TD");
out.println("</TR");
out.println("</TR");
out.println("</TABLE><BR><BR");
out.println("<BR><BR");
out.println("</DIV");
out.println("</TD");
out.println("</TR></TABLE");
out.println("");
out.println("");
out.println("");
out.println("</BODY");

```

```

out.println("</HTML>
");

return;
}
RequestDispatcher view = request.getRequestDispatcher("/borrar.jsp");
view.forward(request,response);
return;
}
}catch(SQLException e){

/*****/

response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");
out.println("<HTML>");
out.println(" <HEAD><TITLE>Error en el Delete</TITLE></HEAD>");
out.println(" <BODY>");
out.print(" This is ");
out.print(this.getClass());
out.println(" using the POST method, se presento un error en el borrado ");
out.println(" </BODY>");
out.println("</HTML>");
out.flush();
out.close();

}
}

public void init() throws ServletException {
// Put your code here
}

}

```

CSS

```

body {scrollbar-3dlight-color:orange;
scrollbar-arrow-color:brown;
scrollbar-track-color:orange;
scrollbar-darkshadow-color:brown;
scrollbar-face-color:gold;
scrollbar-highlight-color:gold;
scrollbar-shadow-color:orange}

InputT {
-moz-border-radius: 15px;
font-family: Tahoma, Verdana, Arial;
font-size: 11px;
background-color: white;
border:gold;
border-style: solid;
border-top-width: 1px;
border-right-width: 1px;
border-bottom-width: 1px;
border-left-width: 1px;
padding-Left : 7px;
padding-Right : 7px;
}
Select {
background-color: #cc0000;
color: #cccccc;
}
textarea {
background-color: #33f00;
color: #000000;
}

```

```

form {
border: 1px solid gold;
padding: 6px;
}
.celda0{
border: 1px dotted 5E93B5;
padding: 10px;
background-color: gold;
}
.celda1{
border: 1px dotted 5E93B5;
padding: 10px;
background-color: #FFFFCC;
}
.celda2{
border-top: 2px solid 5E93B;
border-right: 1px dotted #416D89;
border-left: 1px dotted #416D89;
border-bottom: none;
padding-top: 10px;
padding-right: 3px;
padding-left: 20px;
padding-bottom: 12px;
background-color: #FFCC99;
}

```

Menu.wml

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.1/EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>

  <card id="main" title="Presentacion WML">
    <p align="center"><strong>U.N.A.M</strong><br/>
      <!--
      <br/>
      -->
      <b>Campus Aragon</b><br/>
      <small>Jose Lius Anguiano Vargas</small><br/>
      <small>Oscar Ignacio Rocha Garfias</small> <br/>
      <a href="#menu">continuar</a>
    </p>
  </card>

  <card id="menu" title="Presentacion WML">
    <p align="center">
      <strong> Menu</strong><br/><br/>
       <b><a href="#juego1">Juego</a></b><br/>
       <b><a href=".insert.wml">Insertar</a></b><br/>
       <b><a href="#buscar1">Buscar</a></b><br/>
    </p>
  </card>

  <card id="juego1" title="Nombre">
    <p align="center">
       <strong>Juego</strong><br/><br/>
      <small>Para bajar el juego dar click</small><br/>
      <!--
      <a href="bin/Juego2.jar">Baja el juego</a>
      -->
    </p>
  </card>

  <card id="buscar1" title="Nombre">
    <p align="center">
       <strong>Buscar</strong><br/><br/>
      <small>Coloque los datos de su Nombre</small><br/>
      Nombres: <input name="nombre" size="10" maxlength="10" /><br/>
      Apellido Paterno: <input name="apat" size="10" maxlength="10" /><br/>
    </p>
  </card>

```

```

        <a href="#buscar2">Buscar</a>
    </p>
</card>
<card id="buscar2" title="Nombre">
    <p align="center">
         <strong>Buscar</strong><br/><br/>
        <small>Resultados</small><br/>
        Nombres: Prueba<br/>
        Apellido Paterno: prueba<br/>
        <do type="accept" label="Mandar">
            <!-- -->
            <go method="post" href="/AgendaWml/servlet/Respuesta">
                <!-- -->
                <go method="post"
href="http://dodgem.homeip.net:8080/AjendaWap/Datos.wml">
                    <!--
                    <postfield name="v_nom" value="\$(nombre)"/>
                    <postfield name="v_apat" value="\$(v_apat)"/>
                    <postfield name="v_amat" value="\$(v_amat)"/>
                    -->
                </go>
            </do>
        </p>
</card>
</wml>

```

GLOSARIO

- ADC.** ADd with Carry, sumar con reserva.
- Microsoft .NET.** Plataforma de Microsoft para programar sobre dispositivos móviles.
- J2ME.** Java 2 Micro Edition Plataforma de SUN Microsystems para programar sobre dispositivos móviles.
- Smart Devices.** dispositivos inteligentes.
- URL.** Uniform Resource Locator, localizador uniforme de recurso.
- URI.** Uniform Resource Identification, identificador unificado de recursos.
- HTTP.** Hiper Text Transfer Protocol, Protocolo de transferencia de hipertexto
- SMTP.** Simple Mail Transfer Protocol, protocolo simple de transferencia de correo electrónico.
- FTP.** File Transfer Protocol, protocolo de transferencia de archivos.
- XML.** Extensible Language Markup, lenguaje de marcado ampliable o extensible.
- SOAP.** Simple Object Access Protocol, Protocolo simple de acceso a objetos.
- WSDL.** Web Services Description Language, lenguaje de descripción de servicios Web.
- UDDI.** Universal Description, Discovery, and Integration, Servicio Web en línea que se puede utilizar desde las aplicaciones para descubrir de forma dinámica otros servicios en línea.
- CLR.** Common Language Runtime o máquina virtual común.
- J2SE .** Java 2 Standard Edition.

- J2EE.** Java 2 Enterprise Edition.
- KVM.** Kilo Virtual Machine.
- PDA.** Digital Personal Assistants, asistentes personales digitales.
- GPRS.** General Packet Radio Service, paquete de servicios generales de radio.
- CDC.** Connected Device Configuration Configuración de dispositivos con conexión.
- CLDC.** Connected Limited Device Configuration, Configuración de dispositivos limitados con conexión.
- JVM.** Java Virtual Machine, Máquina virtual de Java.
- CVM.** Compact Virtual Machine, máquina virtual compacta.
- API.** Application Programming Interface, Interfaz de Programación de Aplicaciones.
- MIDP.** Mobile Information Device Profile, Perfil de información del dispositivo móvil.
- SMS.** Short Message Service, Servicio de mensajes cortos.
- LAN.** Local Area Network, Red de área local.
- HTML,** Hypertext Markup Language, lenguaje de marcación de hipertexto.
- WAP.** Wireless Application Protocol, Protocolo de aplicaciones inalámbricas.
- WML.** Wireless Markup Language.
- WTAI.** Wireless Telephony Application Interface.
- CSP.** Communicating Sequential Processes, Proceso secuencial de comunicaciones.
- OEM.** Original Equipment Manufacturer.

SSL. Secure Sockets Layer.

WEP. Wired Equivalency Privacy, Privacidad de aislamiento equivalente.

IEEE. Institute of Electrical and Electronics Engineers, Inc.

TLS. Transport Layer Security, Seguridad de la capa de transporte.

SCSP. Server Cache Synchronization Protocol.

BIBLIOGRAFÍA

1. Cazares Hernández Laura Et. Al. Técnicas Actuales de Investigación Documental. Editorial Trillas UAM 1980
2. Hernández Sampieri. Metodología de la Investigación. Editorial Mc Graw-Hill
3. Zorrilla Arena Santiago. Introducción a la Investigación. Editorial Aguilar León y Cal. 1999
4. López R. Miguel. Normas Técnicas y de Estilo para el trabajo académico. Editorial UNAM.
5. J.G. Brookshear. Introducción a las Ciencias de la Computación. Alianza Editorial. 2a. edición.
6. Castro. Curso de Programación. et. al. McGraw-Hill. 1993.
7. Bobadilla Jesús, Sancho Adela. JAVA a través de ejemplos. Editorial Alfaomega.
8. Ceballos F. Javier. JAVA 2 Curso de Programación. 2ª Edición Editorial Alfaomega.
9. Quintas A. Froufe Cárdenas P. Jorge. J2ME Java 2 Micro Edition. Editorial Alfaomega.
10. Balena Francesco. Programación Avanzada con Microsoft Visual Basic .NET. Editorial Mc Graw-Hill.
11. Maslakowski Mark. Aprendiendo MySQL en 21 Días. Editorial PrenticeHall.
12. Hanna Phil. JSP. Editorial Mc Graw-Hill Osborne Media.

REVISTAS DE INFORMATICA

1. Solo Programadores.
2. Todo Programación.

REFERENCIAS DE INTERNET

1. <http://www.microsoft.com/latam/net/basics/whatis.asp>
2. <http://elizabethpeguero.8m.com/Eliza.htm>
3. [http://www.microsoft.com/spanish/msdn/articulos/archivo/121202/voices/devap
psp.asp](http://www.microsoft.com/spanish/msdn/articulos/archivo/121202/voices/devap
psp.asp)
4. <http://www.monografias.com/trabajos12/hisis/hisis.shtml>
5. http://leo.ugr.es/~fjgc/INTRO/intro_9.htm
6. <http://www.aceproject.org/main/espanol/et/etf01b.htm>
7. <http://www.intel.com/espanol/update/contents/mo11031.htm>