



BENEMÉRITA UNIVERSIDAD

AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

"Información BW en tecnologías móviles basadas en agentes"

**Tesis profesional para obtener el título de:
Lic. en Ciencias de la Computación.**

**Presenta:
Giovanni Miguel Jaimez Cruz**

**Asesor:
Dr. Fernando Zacarías Flores**

Puebla, Pue.

Otoño 2008

Índice	Pág.
Introducción	
Motivación	I
Objetivo general	II
Objetivos específicos	II
Organización de la memoria	III
Capítulo I. Base de Datos	
1.1 Bases de Datos	1
1.2 Diagrama de clases de la Base de Datos	4
1.3 MySQL 5.0 como sistema gestor de la base de datos	5
1.4 Historia de MySQL	5
Capítulo II. El entorno de desarrollo Java	
2.1 Java	8
2.2 Interfaz principal del sistema	11
2.3 Conexión de Java con MySQL	12
2.4 SQLyog	13
Capítulo III. Especificación, análisis y diseño del sistema	
3.1 Descripción del sistema	14
3.2 Arquitectura principal	18
3.3 Casos de uso	18
3.4 Modelo conceptual	36
3.5 Clases principales	38
Capítulo IV. Herramientas de software utilizadas	
4.1 Telefonía móvil	43
4.2 GPRS (General Packet Radio Service)	45
4.3 NetBeans de Sun Microsystems	46

4.4	El servidor de Servlets: Tomcat 5	52
4.5	NetBeans Versión 5.5.1 con MySQL	52
Capítulo V.	Planeación basada en regresión lineal	
5.1	Pronósticos estadísticos	56
5.2	Regresión lineal	60
5.3	Mínimos cuadrados	63
Pruebas		66
Conclusiones		67
Referencias		69

Agradecimientos

A mi Madre:

Eustolia Cruz Arroyo

Como una pequeña compensación a sus sacrificios.

A mi Padre:

Miguel Jaimez Arriaga

Como un agradecimiento a su gran apoyo.

A mi Hermano y Hermana:

Erick y Edalid Jaimez Cruz

Con cariño y respeto.

A mis Amigos:

Porque me demostraron que los hermanos también se eligen.

A mi novia:

Rosalba Alvarado García

Por el presente que me brinda y el futuro que nos espera...

Introducción

Motivación

“Algo peor que no tener información disponible es tener mucha información y no saber qué hacer con ella.” [1]

Actualmente 41% de los ejecutivos pasan el 20% fuera de las oficinas de ahí la necesidad de contar con herramientas de inteligencia de negocios, es primordial para tomar decisiones basadas en información, completa, veraz, oportuna y de calidad. [2]

De acuerdo con Gartner [3], la tendencia indica que para el 2011 el 50% de los teléfonos celulares sean inteligentes (BI móvil), estos dispositivos tienen las funciones necesarias para poder ser herramientas inteligentes de negocio.

La Inteligencia de Negocios o Business Intelligence (BI) se puede definir como el proceso de analizar los bienes o datos acumulados en la empresa y extraer una cierta inteligencia o conocimiento de ellos, generar escenarios, pronósticos y reportes que apoyen a la toma de decisiones, lo que se traduce en una ventaja competitiva. La clave para BI es la información y uno de sus mayores beneficios es la posibilidad de utilizarla en la toma de decisiones.

BI apoya a los tomadores de decisiones con la información correcta, en el momento y lugar correcto, lo que les permite tomar mejores decisiones de negocios. La información adecuada en el lugar y momento adecuado incrementa efectividad de cualquier empresa.

En el contexto de la informática, un **almacén de datos** (del inglés *data warehouse*) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones clave en la entidad en la que se utiliza. Se trata, sobre todo, de un expediente completo de una organización, más allá de la información transaccional y operacional, almacenado en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos (especialmente OLAP, *procesamiento analítico en línea*). [4]

Se pretende utilizar el Business Warehousing (BW), como herramienta principal para que, a partir de información proporcionada por el Hospital Universitario (HU), se cree una Base de Datos (BD) que facilite la introducción y sobre todo la extracción de información relevante para la creación de escenarios y a su vez, la creación de pronósticos que ayuden en la toma de decisiones clave, apoyados en la visualización de los resultados tanto de forma escrita, como de forma gráfica, esto para dar una idea del panorama futuro que tendrá cada uno de los escenarios que se consulten, recordando que será una herramienta solo de apoyo, y que la decisión recaerá en la persona más capacitada para tal acción.

Objetivo General

El objetivo general de esta tesis es utilizar la Tecnología Móvil, para crear:

1. Una Aplicación Móvil.
2. Un Servidor de Aplicaciones.
3. Una conexión a la Base de Datos desde el servidor.

Objetivos Específicos

Para poder alcanzar el Objetivo General es importante establecer una serie de objetivos específicos:

- Análisis y selección de los datos relevantes del documento de información proporcionado por el HU.
- Creación de una BD con los datos seleccionados anteriormente, de tal forma que la BD pueda ser transportada e implementada fácilmente en cualquier PC.
- Implementación de la BD en un Sistema Gestor de Base de Datos, en este caso MySQL.
- Creación de un Sistema para el Almacenamiento, Eliminación y Consulta de la información en JAVA.
- Análisis, selección y aplicación de protocolos de comunicación vía http para la comunicación entre el dispositivo móvil y el servidor de aplicaciones.
- Creación y configuración de un Servidor de Aplicaciones Servlets para que pueda recibir parámetros desde el dispositivo móvil, consultar la BD de forma local, y enviar un resultado de vuelta vía GPRS.
- La parte de la creación de la Aplicación Móvil, y el Servidor de Aplicaciones, será apoyada por NetBeans 5.5.1.

Organización de la memoria

Este documento está estructurado del siguiente modo:

Capítulo 2: que es una base de datos (BD), y que sistema escogimos para gestionar la BD, en este caso MySQL, un poco de su historia, características y ventajas, además de el modelo Entidad-Relación de nuestra BD.

Capítulo 3: explicación detallada del entorno de desarrollo Java, desde el porqué se ha escogido Java para este proyecto hasta sus orígenes, pasando por sus características más elementales, la interfaz principal de nuestro sistema para actualizar la BD, además de la forma y requisitos para que se conecte con MySQL, y por ultimo una herramienta de apoyo grafica para realizar pruebas.

Capítulo 4: datos sobre el conjunto de aplicaciones producidas para realizar este documento. Análisis, descripciones y explicaciones de los programas escritos.

Capítulo 5: una completa mirada a las tecnologías, y herramientas software utilizadas en este trabajo.

Capítulo 6: introducción, explicación e historia acerca del método de pronóstico utilizado en esta herramienta.

Capítulo 7: conclusiones de éste que firma el proyecto y futuras líneas de trabajo.

Capítulo I

Base de datos

1.1 Base de Datos

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo en la toma de decisiones clave 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 lograr ventajas competitivas. La información integrada y compartida constituye una ventaja importante de los sistemas de base de datos en ambientes grandes [5].

Las ventajas de un sistema de base de datos son las siguientes:

- **Compacto.** No hacen falta archivos de papeles que pudieran ocupar mucho espacio.
- **Rápido.** En la máquina se puede tener y modificar datos mucho más rápido que cualquier ser humano.
- **Manejable.** Se elimina la tarea de mantener archivos a mano. Las tareas mecánicas siempre serán mejor realizadas por las máquinas.
- **Actualizado.** Se dispone en cualquier momento de información precisa y al día.

Conceptos básicos

Definición 1.1 Una base de datos es 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 [27].

Modelos de Datos

Existen tres alternativas disponibles para el diseño de base de datos: el modelo jerárquico, el modelo de red y el modelo relacional [27]:

- **Modelo Jerárquico.** La forma de esquematizar la información se realiza a través de representaciones jerárquicas o relaciones de padre/hijo, de manera similar a la estructura de un árbol. Así, el modelo jerárquico puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.
- **Modelo de Red.** El modelo de red evita esta redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector.
- **Modelo Relacional.** El modelo relacional se basa en dos ramas de las matemáticas: “La teoría de conjuntos y la lógica de predicados de primer orden”[6]. El hecho de que el modelo relacional se encuentre basado en teorías matemáticas, es lo que lo hace ser seguro, robusto y predecible. Al mismo tiempo, estas ramas de la matemática proporcionan los elementos básicos

necesarios para crear una base de datos relacional con una buena estructura, además de proporcionar las herramientas necesarias para crear eficientes metodologías de diseño. Las bases de datos relacionales son las que han tenido mas uso comercial, consisten de tablas (relaciones), cada tabla lleva el nombre de una relación y contiene filas (tupla) y columnas(atributos). Estas filas y columnas son lo que conocemos como campos y registros. La manera de acceder a cualquiera de ellos se realiza mediante un campo “llave”; éste se elige después de analizar todos los campos del registro. La característica esencial de un campo llave es que es distinto para cada registro. Al procedimiento de seleccionar el campo llave y en general la estructuración de la base de datos se le llama normalización.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas. Las operaciones básicas son:

- Seleccionar renglones de alguna tabla.
- Seleccionar columnas de alguna tabla.
- Unir o juntar información de varias tablas.

Relación, Tupla y Atributo

Una relación es representada gráficamente como una tabla y se define informalmente como: una tabla con filas y columnas. Un Sistema Gestor de Base de Datos (SGBD) sólo necesita que el usuario perciba a la base de datos como un conjunto de tablas, dicha percepción sólo se aplica a la estructura lógica de la base de datos y no a su estructura física la cual puede implementarse con distintas estructuras de almacenamiento. La definición formal de relación es la siguiente.

Definición 1.2 El Administrador de Base de Datos o DBA es la persona con la responsabilidad central de los datos. La labor del DBA es decidir en primer término cuales datos deben almacenarse en la base de datos y establecer políticas para mantener y manejar los datos una vez almacenados.

Definición 1.3 Una base de datos relacional es una base de datos que es percibida por el usuario como una colección de relaciones o una tabla de dos dimensiones. El lenguaje estructurado de consulta (SQL) es empleado para manipular base de datos relacionales. El Instituto Nacional Americano de Estándares o ANSI (American National Standard Institute) ha establecido a SQL como el lenguaje estándar para operar base de datos relacionales [28].

Definición 1.4 El Sistema Administrador de Base de Datos o DBMS (de sus siglas en inglés) es un conjunto de programas que maneja todo acceso a la base de datos. Cuando un usuario solicita un acceso, empleando algún lenguaje de datos determinado el DBMS analiza e interpreta la solicitud, inspecciona el esquema externo del usuario, el esquema conceptual y la definición de la estructura de almacenamiento. Finalmente el DBMS ejecuta las operaciones necesarias.

Definición 1.5 El Lenguaje de Definición de Datos o DDL (de sus siglas en inglés) es un componente que permite al DBMS definir datos.

Definición 1.6 El Lenguaje Manipulador de Datos o DML (de sus siglas en inglés) permite al DBMS entender solicitudes del usuario, ya sea para consultar, agregar o eliminar datos.

Definición 1.7 El término entidad se usa para referirse a cualquier objeto distinguible en la base de datos de la cual se desea registrar información (empleados, productos, etc.). Generalmente en estas entidades existen también interrelaciones que las vinculan, las cuales se representan mediante líneas o actos de conexión y pueden relacionar varias entidades.

Definición 1.8 Un relación es bi-direccional y representa la asociación entre dos entidades, o entre una entidad consigo misma (relaciones recursivas).

Definición 1.9 Una Llave Primaria o PK (de sus siglas en inglés) es una columna o un grupo de columnas que identifican de manera única a cada renglón de la tabla. Cada tabla debe tener una llave primaria y esta debe ser única. No se aceptan duplicados de la llave primaria y generalmente no se puede cambiar.

Definición 1.10 Una Llave Foránea o FK (de sus siglas en inglés) es una columna o combinación de columnas de una tabla, que se refiere a una llave primaria en la misma (recursiva) o en otra tabla. Las FK son utilizadas para hacer uniones (joins) entre tablas y esta se basa en los valores de los datos y son puramente lógicos. La FK puede tener un valor repetido o puede ser nulo. Además la FK debe coincidir con la PK existente.

Normalización

La normalización minimiza la redundancia e inconsistencia en los datos. Una base de datos sin normalizar es redundante. La redundancia de datos causa problemas de integridad. Las transacciones de actualización y borrado pueden no ser consistentes en todas las copias de los datos cuando hay inconsistencia en la base de datos. Ésta también ayuda a identificar entidades, relaciones y tablas mal diseñadas [7], [27].

Para normalizar una base de datos es necesario seguir las formas normales descritas en la Figura 1.1. Aunque en muchas ocasiones no es necesario implementar la Cuarta Forma Normal (4FN) y Quinta Forma Normal (5FN) no necesariamente con sólo aplicar las primeras tendremos resueltos de todos los problemas de consistencia e integridad.

FORMA NORMAL	DESCRIPCIÓN
Primera Forma Normal (1FN)	La tabla debe tener un sólo valor para cada renglón. La tabla no puede contener grupos repetitivos.
Segunda Forma Normal (2FN)	La tabla debe estar en 1FN. Cada columna que no es llave debe ser dependiente de la PK completa.
Tercera Forma Normal (3FN)	La tabla debe estar en 2FN. Una columna que no es PK debe ser funcionalmente dependiente de otra columna, no llave.
Cuarta Forma Normal (4FN)	En las relaciones varios a varios, entidades independientes no pueden ser almacenadas en la misma tabla.
Quinta Forma Normal (5FN)	La tabla debe ser reconstruida desde las tablas resultantes en las cuales ha sido particionada.

Figura 1.1 Formas Normales

Diagrama del Modelo Relacional para nuestro caso

Cuando se diseña una base de datos mediante el modelo relacional, al igual que ocurre en otros modelos de datos, se tienen distintas alternativas, es decir, se pueden obtener diferentes esquemas relacionales y no todos son equivalentes, ya que algunos van a representar la realidad mejor que otros. Es necesario conocer qué propiedades debe tener un esquema relacional para representar adecuadamente una realidad y cuáles son los problemas que se pueden derivar de un diseño inadecuado.

1.2 Diagrama de Clases de la Base de Datos

La base de datos presentada es una base de datos MySQL, concretamente su versión 5. Hemos creado un modelo entidad/relación (Figura 1.2) para mostrar las cinco entidades creadas junto con la relación existente entre ellas, es el siguiente [29]:

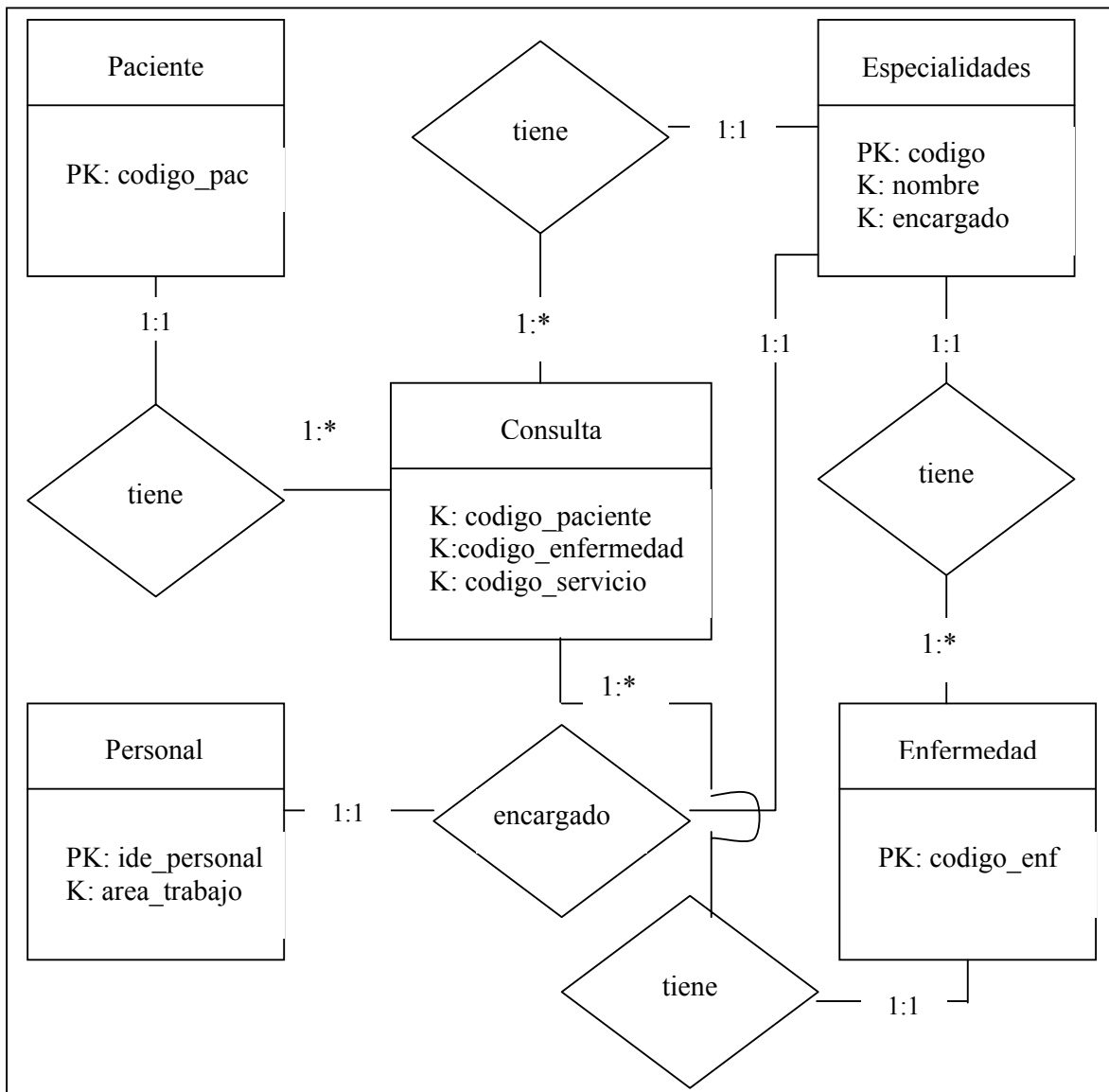


Figura 1.2 Diagrama Entidad-Relación.

1.3 MySQL 5.0 como Sistema Gestor de bases de Datos

Como Sistema Gestor de bases de datos hemos decidido utilizar MySQL 5.0. Y esto es no sólo por su facilidad de uso y su gratuidad, ya que cuenta con licencia *GPL* (*General Public License*), sino que, además, es probablemente el Sistema Gestor de bases de Datos SQL más popular y extendido en la actualidad.

En su nacimiento, MySQL no tenía elementos esenciales de las bases de datos relacionales, como podían ser la falta de *integridad referencial* y de *transacciones*. Pero gracias a su simplicidad y a su licencia de código abierto, todas las características que le faltaban fueron sumadas al proyecto tanto por la empresa que la mantiene, MySQL AB, como por desarrolladores de software libre. [8], [28].

Características a destacar son:

- Amplio subconjunto del lenguaje SQL.
- Indexación y búsqueda de campos de texto.
- Manejo seguro de claves foráneas y transacciones.
- Flexibilidad: esta disponible en multitud de plataformas y sistemas.
- Conectividad asegurada.

Otro dato a tener en cuenta es que MySQL fue seleccionada para formar parte de las llamadas *Soluciones WAMP* (Windows, Apache, MySQL, PHP/Perl/Python), que describen las aplicaciones web creadas por la combinación anterior de herramientas. Fue popularizada por la editorial O'Reilly[13], de gran peso en el mundo informático.

1.4 Historia de MySQL

SQL (*Lenguaje de Consulta Estructurado*) [28] fue comercializado por primera vez en 1981 por IBM (International Business Machines), el cual fue presentado a ANSI (por sus siglas en inglés: American National Standards Institute) y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael Widenius en la década de los 90 trató de usar *mSQL* [14] para conectar las tablas usando rutinas de bajo nivel ISAM (Indexed Sequential Access Method), sin embargo, *mSQL* no era rápido y flexible para sus necesidades. Esto lo conllevó a crear una API SQL (Application Programming Interface SQL) denominada MySQL para bases de datos muy similar a la de *mSQL* pero más portable.

La procedencia del nombre de MySQL no es clara. Desde hace más de 10 años, las herramientas han mantenido el prefijo *my*. También, se cree que tiene relación con el nombre de la hija del cofundador Monty Widenius quien se llama *My*.

Por otro lado, el nombre del delfín de MySQL es Sakila y fue seleccionado por los fundadores de MySQL AB en el concurso "Name the Dolphin". Este nombre fue

enviado por Ambrose Twebaze, un desarrollador de Open source Africano, derivado del idioma SiSwate, el idioma local de Swazilandia y corresponde al nombre de una ciudad en Arusha, Tanzania, cerca de Uganda la ciudad origen de Ambrose.

Lenguajes de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP (Sistemas, Aplicaciones y Procesamiento de datos), lenguaje ABAP(Advanced Business Application Programming).

Aplicaciones

MySQL es muy utilizado en aplicaciones web como MediaWiki [15], Drupal [16] o phpBB [13], en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla [14]. Su popularidad como aplicación web está muy ligada a PHP [16], que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM [16], pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Plataformas

MySQL funciona sobre múltiples plataformas, incluyendo:

- AIX
- BSD
- FreeBSD
- HP-UX
- GNU/Linux
- Mac OS X
- NetBSD
- Novell Netware
- OpenBSD
- OS/2 Warp
- QNX
- SGI IRIX
- Solaris
- SunOS
- SCO OpenServer
- SCO UnixWare

-
- Tru64
 - eBD
 - Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.
 - OpenVMS

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en un red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada [30].

Tipos de compilación del servidor

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

Especificaciones del código fuente

MySQL está escrito en una mezcla de C y C++. Hay un documento que describe algunas de sus estructuras internas en <http://dev.mysql.com/doc/internals/en/> (en inglés).

Capítulo II

El entorno de desarrollo: JAVA

2.1 JAVA

Java es un lenguaje de programación orientado a objetos, con una sintaxis similar a C o C++ pero ofreciendo una mayor simplicidad y robustez en el ciclo de desarrollo: las construcciones y características más complicadas de C y C++ han sido eliminadas y el lenguaje contiene mecanismos implícitos para garantizar la seguridad de las aplicaciones construidas con él. [17]

También incorpora dos mecanismos a la hora de escribir programas simples, potentes y robustos: un tratamiento interno de multitarea y un sistema de excepciones que normaliza el procesado de errores por parte del programador.

La principal característica de Java es que es independiente de la plataforma, pudiendo ejecutarlo sobre distintas arquitecturas y sistemas operativos sin que sea necesario modificar el código del programa. Esta independencia se logra gracias a que el lenguaje está soportado por dos elementos fundamentales: el compilador y la máquina virtual. El compilador traduce los programas a un formato especial llamado *bytecodes*, que es el formato que se le pasa a la máquina virtual. Tanto el compilador Java como la máquina virtual son específicos para cada plataforma, por lo que para poder ejecutar un programa Java en una determinada plataforma debe existir previamente una máquina virtual para ella, por lo que Sun Microsystems dispone de un entorno de ejecución para la mayoría de las plataformas.

Otra ventaja es que cuenta con un gran número de clases preexistentes que no deja de aumentar, presentando una gran riqueza en cuanto al tipo de funciones que permiten realizar [33], [34].

En la Tabla 2.1 se exponen algunos ejemplos.

Tabla 2.1 Ejemplos de clases de Java

Clase	Utilidad
<i>Math</i>	Proporciona métodos para realizar las operaciones matemáticas más habituales
<i>Date</i>	Se utiliza para manejar fechas y horas
<i>Integer</i>	Tiene como miembro una variable de tipo entero, y proporciona los métodos para trabajar con ella
<i>String</i>	Proporciona a Java la capacidad para el manejo de cadenas de caracteres
<i>Vector</i>	Representa un array de objetos que puede crecer y reducirse. Además, permite acceder a los elementos con un índice
<i>Frame</i>	Es una ventana de la que pueden depender otras ventanas y que puede tener una barra de menús
<i>FileDialog</i>	Muestra una ventana de diálogo en la que se puede seleccionar un fichero según el modo de apertura del fichero (para lectura o escritura)

Sun describe el lenguaje Java como “*simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico*”[18]:

Simple: Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. Los lenguajes más difundidos son C y C++, pero adolecen de falta de seguridad, característica muy importante para los programas que se usan en Internet, por ello Java se diseñó para ser parecido a ellos y así facilitar un rápido y fácil aprendizaje. Java elimina muchas de las características de otros lenguajes para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles, como el *garbage collector* (reciclador de memoria dinámica), que se encarga de liberar memoria no usada. Java reduce en un 50% los errores más comunes de programación respecto a lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- No se admite la aritmética de punteros.
- No existen referencias.
- No existe la definición de registros (*struct*).
- No existe la definición de tipos (*typedef*).
- No existe la definición de macros (*#define*).
- No existe la necesidad de liberar memoria (*free*).

Orientado a objetos: Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta encapsulación, herencia y polimorfismo. Hace uso de la definición de entidades formadas por métodos y variables que reciben el nombre de clases, la instancia de una clase recibe el nombre de objeto.

Distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan serlo.

Interpretado: la máquina virtual Java es un programa que se ejecuta sobre el sistema operativo del ordenador (por lo que es dependiente de la plataforma) y ejecuta directamente el código objeto mediante la interpretación de los *bytecodes*, aunque no se trata de un intérprete tradicional pues éstos ya han pasado por las etapas de validación del compilador Java.

Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los *bytecodes*, que son el resultado de la compilación de un programa Java.

Seguro: La seguridad en Java tiene dos facetas. Por una parte se eliminan características de C y C++ para prevenir el acceso ilegal a la memoria como, por ejemplo, punteros o el *casting* implícito. Por otra parte, el código Java pasa muchos tests antes de ejecutarse en la máquina virtual. Pasa a través de un verificador de *bytecodes* que comprueba el formato de los fragmentos de código y aplica un comprobador de teoremas para detectar fragmentos de código que falsee punteros, viole derechos de acceso sobre objetos o intente cambiar el tipo o clase de un objeto. Si los *bytecodes* pasan la verificación sin generar ningún mensaje de error, entonces sabemos que:

- El código no produce desbordamiento de operandos en la pila.
- El tipo de los parámetros de todos los códigos de operación son conocidos y correctos.
- No ha ocurrido ninguna conversión ilegal de datos.
- El acceso a los campos de un objeto se sabe que es legal.
- No hay ningún intento de violar las reglas de acceso y seguridad.

El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros local del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior. Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida. En resumen, las aplicaciones Java resultan extremadamente seguras [31].

Arquitectura neutral: El compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará (este fichero contiene los *bytecodes*). Cualquier máquina que tenga el sistema de ejecución (*run-time*, que sí es dependiente de la máquina) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado.

Portable: Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros de 32 bits en complemento a 2 y las cadenas de caracteres utilizan Unicode (no ASCII). Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos diferentes (Unix, Pc, Mac, etc.).

Altas prestaciones: Para los casos en que la velocidad del intérprete Java no resulte suficiente, existen mecanismos como los compiladores **JIT** (*Just In Time*), que se encargan de traducir, a medida que va siendo necesario, los *bytecodes* a instrucciones de código máquina. También existen otros mecanismos como los compiladores incrementales y sistemas dedicados para tiempo real.

Multitarea: Java permite muchas actividades simultáneas en un programa. Los hilos (*threads*) son pequeños procesos o piezas independientes de un gran proceso. Al estar los *threads* construidos en el lenguaje, son más fáciles de usar y más robustos que sus

homólogos en otros lenguajes que no los soportan de manera nativa. Esta característica permite mejorar el rendimiento interactivo y el comportamiento en tiempo real.

Dinámico: Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales (siempre que mantengan el API anterior). Esto permite actualizar el código “en caliente” y facilita el mantenimiento del software. Por otro lado, Java proporciona mecanismos para cargar dinámicamente clases desde la red, de manera que nuevos contenidos de información podrán ser tratados por manejadores específicos.

2.2 Interfaz principal del sistema de gestión de la Base de Datos

Para poder actualizar nuestra Base de Datos, hemos creado un Sistema desarrollado en Java que permite administrar de forma rápida y efectiva:

- Personal
- Pacientes
- Especialidades
- Enfermedades
- Y Consultas

La parte principal de este sistema se muestra en la Figura 2.1 y es la de Consultas, ya que es a través de esta interfaz que nosotros podemos dar Altas, y Consultar, la información que se necesita para que funcione nuestro motor de pronósticos.

Figura 2.1 Pantalla Principal del Sistema que actualiza la BD.

En la figura 2.1, podemos crear escenarios tan variados, que constan, en primer lugar de los 31 días del mes, después de un mes específico, o todo el año, seguido del

año o conjunto de años a consultar, los cuales se ingresaron previamente de acuerdo con la información proporcionada por el HU, y que constan de los años 1996 hasta el 2006, y por ultimo una especialidad en especifico, entre las cuales tenemos 40 en total, o todas juntas, todo depende del grado de complejidad con el que se requiera nuestro escenario de consulta.

2.3 Conexión de Java con MySQL

Ya que nuestro sistema requiere conectarse con un Sistema Gestor de Base de Datos, que para nuestro uso es MySQL, en este capitulo se explica de manera general, como es que se puede configurar dicha conexión.

WAMP

(Windows-Apache-MySQL- PHP/Python/PERL). El término hace referencia al sistema creado por la conjunción de esas aplicaciones libres (de código abierto) y el sistema operativo Windows. Este grupo de aplicaciones generalmente son usados para crear servidores web.

WAMP provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un sistema operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script web (PHP, Python o PERL) [32].

Este conjunto de aplicaciones es muy útil y practico, ya que nos evitamos de engorrosos manuales de instalación y configuración necesarios para el funcionamiento de estos programas en forma separada, al instalar este paquete, automáticamente quedan configurados los parámetros necesarios para su uso, evitándonos así mucho tiempo y esfuerzo.

Conector de Java con MySQL

Es un controlador nativo de Java que convierte las llamadas de JDBC (por sus siglas en ingles de: Java Database Connectivity) dentro del protocolo de red usado por la Base de Datos MySQL. Esto permite a los desarrolladores trabajar con el lenguaje de programación Java fácilmente construyendo programas que interactúan con MySQL y conectan todos sus datos almacenados incluso en un ambiente heterogéneo. MySQL Connector/J es un controlador de tipo IV JDBC y tiene un conjunto de funciones completa que soporta las capacidades de MySQL.

Dentro del capitulo 4.5 existe una clase de prueba, para la conexión de Java con MySQL, es la clase mas sencilla que existe, y su función es checar si se estableció la conexión, mandar un letrero satisfactorio, o de lo contrario desplegar un error, todo ello desde línea de comandos.

Habiendo descargado e instalado las dos aplicaciones aquí mencionadas es posible probar nuestra función, y con ello quedara establecida la comunicación de Java con nuestro Sistema Gestor de Base de Datos.

2.4 SQLyog

SQLyog es una herramienta gráfica fácil de usar, compacta y muy rápida para administrar su base de datos MySQL en cualquier parte del mundo. SQLyog es una herramienta que permite administrar bases de datos MySQL, y con esto me evite realizar mi trabajo con la BD desde línea de comandos, así, de una manera grafica y fácil, pude visualizar los avances y realizar mis pruebas para posteriormente realizar la conexión de MySQL con Java[19].

Características destacadas:

- Soporte Unicode/UTF8 completo
- Productividad desarrollador / usuario
- Documentación de esquema HTML
- Atajos para generar sentencias SQL DML desde definición de esquema
- Editor de consultas con pestañas múltiples y editor de resultados
- Editor de consultas con pestañas múltiples y editor de resultados
- Ejecución múltiple de consultas
- Ejecución de consultas multi-hilo - Posibilidad de detener consultas grandes
- Plantillas SQL
- Interfaz de cuadrícula estilo Excel para ver/actualizar los resultados
- Editor de Blob multi-formato
- Se pueden ver datos en modo cuadrícula o texto
- Exportación de Resultado/Datos a CSV / XML / HTML / Excel
- Soporte completo de versiones desde 3.23.38 hasta la última 5.x
- Exportación de resultados/datos de tabla al portapapeles/archivo amigable con Excel
- Editor de tabla y resultado sin diálogo
- Restaura / Importa volcados SQL largos
- Soporta objetos MySQL 5.x
- Gestión de MySQL hospedado
- Administración de índices
- Administrador de relaciones/claves foráneas
- Reordenar columnas
- Copiar objetos a otro servidor con un solo clic
- Diagnostico de tablas
- Herramientas para despejar
- Explorador de objetos
- Crear/borrar bases de datos
- Optimizado para la gestión de MySQL hospedado
- Administración MySQL de alta velocidad. Utiliza una API MySQL nativa en C - la forma más rápida de comunicarse con un servidor MySQL
- 100% manejable con el teclado
- Binario pequeño y compacto
- Uso mínimo del registro - migración fácil de las preferencias del usuario arrastrando y soltando archivos de configuración.
- Aspecto visual ordenado, se pueden mostrar/ocultar paneles

Capítulo III

Especificación, análisis y diseño del sistema

3.1 Descripción del sistema

Explicar con palabras el software que se quiere desarrollar resulta muy útil a la hora de diseñar y analizar una aplicación. La definición textual del sistema software deseado es la siguiente:

Se necesita desarrollar un sistema de software capaz de pronosticar escenarios de consultas para apoyar a la toma de decisiones clave, por medio de un teléfono móvil. Para ello se desarrollará, además del programa para el móvil correspondiente (en adelante midlet), una base de datos del Hospital Universitario (HU), un sistema que gestione nuestra base de datos, pudiendo tener Altas, Bajas y Consultas de la información más relevante previamente seleccionada y proporcionada por el HU, y un servidor de servlets que se conectará de manera local a la BD y que será accesado por el midlet a través de Internet vía GPRS.

El software permitirá, a un usuario previamente seleccionado ya que solo a ellos se les instalará el software, poder tener una herramienta al alcance de su mano que lo apoye en la toma de decisiones clave mediante el pronóstico de consultas por mes o por año, de uno o varios años futuros y además, de uno o varios servicios, además de obtener los resultados de forma escrita, los podrá visualizar de manera gráfica para tener un mayor panorama del pronóstico, desde cualquier lugar y con el único uso de un teléfono móvil con conexión a Internet.

Básicamente, al iniciar el midlet, el usuario introducirá sus datos del escenario, que, serán enviados al servidor y, una vez recibidos, se procederá al pronóstico solicitado.

En este punto podrá, seleccionar un mes específico o todo un año, un servicio específico o todos juntos entre los que se encuentren disponibles en la información que nos proporcione previamente el HU. De este modo habilitamos el software para trabajar, no con un solo un año, sino con un conjunto de ellos, favoreciendo la flexibilidad de las aplicaciones.

Seguidamente se procederá a la selección del año a pronosticar, pudiendo, además, poder poner rango de años para ampliar la información a pronosticar.

Después de elegir estos datos, el usuario seleccionará si su consulta será por un solo año o por rango, tomando solo el primer año introducido o ambos.

En un último paso se procederá a confirmar la conexión a Internet por medio del teléfono, y recibirá primeramente de manera escrita los resultados, desplegados en una o varias páginas según el rango de pronóstico, pudiendo navegar fácilmente entre cada una de ellas, posteriormente al llegar a la última página de resultados, aparecerá por primera vez la opción de poder graficar los resultados obtenidos, esta opción se mantendrá después de aquí al navegar entre las páginas de los resultados escritos.

Midlet Pronósticos

El núcleo central de este proyecto y hecho diferenciador de otros muchos es, sin duda, esta aplicación para teléfonos móviles que hemos llamado, no sabemos si acertadamente, “Herramienta para la toma de decisiones clave”. Pasamos ahora a describir su diseño.

Este midlet no cuenta con muchas clases, debido a que se limita a enviar y a recibir información, y posteriormente a realizar el despliegue de esta, por lo que el realizar un diagrama de clases UML al uso podría resultar un excelente apoyo para conocer el completo funcionamiento del midlet [31], [35].

Midlet: las clases

En este apartado trataremos de explicar cada una de las clases que componen el midlet, para qué sirven, cuál es su funcionamiento interno y cómo están estructuradas.

Pantalla principal: HTTPMIDlet

Esta es la clase principal del midlet, además de ser la pantalla en la que introduciremos nuestros datos para crear el escenario a pronosticar.

El primer dato es un Grupo de selección donde elegiremos el mes del escenario.

El segundo dato es un Grupo de selección donde tenemos los 40 servicios disponibles en la información que nos proporciona previamente el HU, además de que el primer nombre que es TODOS, es decir, el pronóstico se realizará tomando en cuenta todos, o cada uno de los servicios.

El tercer dato es un Grupo de Selección donde elegiremos si el pronóstico se realizará tomando en cuenta el primer dato o todo el año, se inicializa una variable con el valor del índice seleccionado, se realiza una estructura de control SWITCH a esta variable, en caso que la variable sea igual a “0”, el primer operador enviado al servlet será el del primer dato elegido, y en caso que la variable sea igual a “1”, se inicializa otra variable llamada “meses” con el valor de “TODOS”, que significa que el pronóstico se realizará tomando en cuenta todos los meses, es decir, por año, y este será el valor del primer operador enviado al servlet.

El segundo operador a enviar al servlet será una variable inicializada según el caso del segundo dato introducido. La variable se llama “servicio”, y si el segundo dato es igual a “0”, el valor de esta será “TODOS”, si no, se obtiene el índice del nombre elegido en el segundo dato y se le suma 200700, para obtener el número de identificador del servicio elegido.

El cuarto dato es un Campo de texto donde introduciremos el año base de nuestro pronóstico, y será el valor de el tercer parámetro enviado a nuestro servlet.

El quinto dato es un Campo de texto donde introduciremos el año final de nuestro pronóstico, y será el valor de el cuarto parámetro enviado al nuestro servlet.

El sexto dato es un Grupo de selección donde elegiremos si nuestro pronóstico será de un solo año, o de un rango de años, obtendremos el valor del índice seleccionado y ese será el valor de nuestro quinto y último parámetro enviado a nuestro servlet, si el valor fuera “0” (Solo un año), el pronóstico será solo de el valor de nuestro tercer parámetro, o si fuera “1” (Rango), el pronóstico será desde el valor del tercer parámetro, hasta el valor del cuarto parámetro.

De esta manera se inicializa la variable “url” con la dirección de nuestro servlet, y de manera implícita, el envío de nuestros cinco parámetros para poder realizar el pronóstico.

Después de presionar el boton de “OK” se conectará al servlet y si la conexión tiene éxito, se mostrarán los resultados en pantalla.

Clase secundaria: Resultados

Esta clase es la encargada de la descomposición de la cadena de resultados en todos los elementos que la componen, ya que estos se encuentran separados solo por una diagonal (/). Después a través de su método llamado “visualiza()”, envía de ocho en ocho los resultados obtenidos ya por separado a la clase CanvasResultados(), para que esta los despliegue.

Además verifica si hay mas datos que desplegar, pone en pantalla los botones de adelante (--->>>), o de atrás (<<<---), según se de el caso, además del botón de “Atrás” que regresa a la pantalla principal, y de “Graficar” que muestra la gráfica de los resultados obtenidos.

Clase secundaria: CanvasResultados

Esta clase recibe los datos a “pintar” en pantalla de la clase Resultados(), y los pinta siempre en medio para no tener problemas de espacio.

Clase secundaria: MiCanvas

Esta clase es la encargada de pintar la gráfica de resultados del escenario pronosticado.

Recibe también la cadena de resultados, la descompone, y escala el tamaño de la barra según el tamaño que se obtenga para mantener una apariencia constante, además ajusta el ancho de las barras, al número de estas que se tengan que desplegar en pantalla, a mayor número de barras, menor será el ancho, y viceversa.

Tiene también un botón de “Atrás” que regresa a la pantalla anterior a la gráfica.

Servlet: las clases

En este apartado se trata de explicar cada una de las clases que componen el servlet, para qué sirven, cuál es su funcionamiento interno y cómo están estructuradas.

Servlet: PronosticoServlet

Este es el servlet al que se conecta el midlet, este es el encargado de recibir los 5 parámetros y de enviarlos a la clase Prueba(), recibir el resultado y enviarlo al midlet en forma de cadena.

Clase secundaria: Prueba

Esta clase quizás pensarán muchas personas que no tiene sentido, y que está de mas utilizarla, para nuestro caso se creo para mandar manualmente los escenarios a pronosticar, y poder desplegar en background los resultados, de esta forma se pudo realizar modificaciones y configuraciones para el funcionamiento del servlet, y la información que mande de regreso al midlet sea la correcta.

Lo que hace es únicamente servir de puente entre PronosticoServlet y la clase de ConsultaPronostico, además de abrir la conexión con la Base de Datos.

Clase secundaria: ConsultaPronostico

Esta es la clase más importante del servlet, ya que es la encargada de realizar la consulta a la Base de Datos, y realizar el pronóstico, según los parámetros recibidos del midlet.

Esta clase toma en cuenta las diferentes combinaciones según sea el caso del escenario a pronosticar, es decir, si la consulta la realizara por uno o todos los meses, por uno o todos los servicios, y todos los años base que anteriormente se dieron de alta en la BD de acuerdo a la información proporcionada por el HU.

La consulta es realizada a través de su método llamado Busqueda(), y los datos obtenidos son enviados al método llamado EscribeResultado() el cual calcula la recta de regresión lineal a través de la fórmula de mínimos cuadrados, y crea la cadena de resultados separándolos por una “/”.

Clase secundaria: LoadDriver

Esta clase es la encargada de establecer la conexión con la BD, y además de crear las sentencias necesarias para la consulta a la base de datos según los escenarios creados en el midlet.

3.2 Arquitectura General

En la figura 3.1 se puede visualizar la arquitectura general de la herramienta de Toma de Decisiones Clave, con todos los componentes que intervienen en su funcionamiento.

De esta manera se tiene un panorama completo del camino que siguen los datos así como la fuente desde donde se extraen, interviniendo los emisores y receptores.

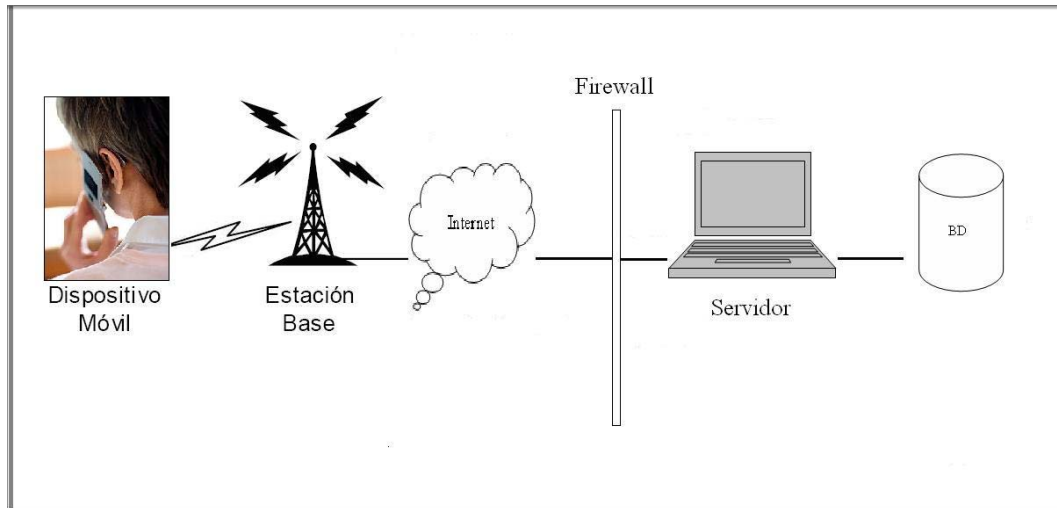


Figura 3.1 Arquitectura General de la Herramienta para la Toma de Decisiones Clave.

3.3 Casos de Uso

Uno de los lenguajes que en los últimos años ha tenido mayor aceptación es UML, esto se debe a que este novedoso lenguaje permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Otra de las razones por las que UML se ha convertido en el estándar de facto de la industria, es debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE) [37], [40].

Uno de los objetivos principales de la creación de UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común y sencilla para que fuera aceptada por la gran mayoría. En la Figura 1 se puede ver cuál ha sido la evolución de UML hasta la creación de UML 1.3, en el que se basa este documento. Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación [38], [39].

En las siguientes páginas describimos los casos de uso del sistema que se creó, tanto el que actualiza la Base de Datos, como las aplicaciones móvil y web.

Con esto se pretende dar una idea de cada clase en particular a través de su descripción, para que, de forma narrativa se entiendan los procesos que se desarrollan en la ejecución de nuestro sistema.

Además se complementa con imágenes en tiempo de ejecución de acuerdo al caso de uso del que se está haciendo referencia y 4 ejemplos que servirán para que el panorama sea aun mayor acerca de su funcionamiento.

CASO DE USO 1: Gestión de la Base de Datos

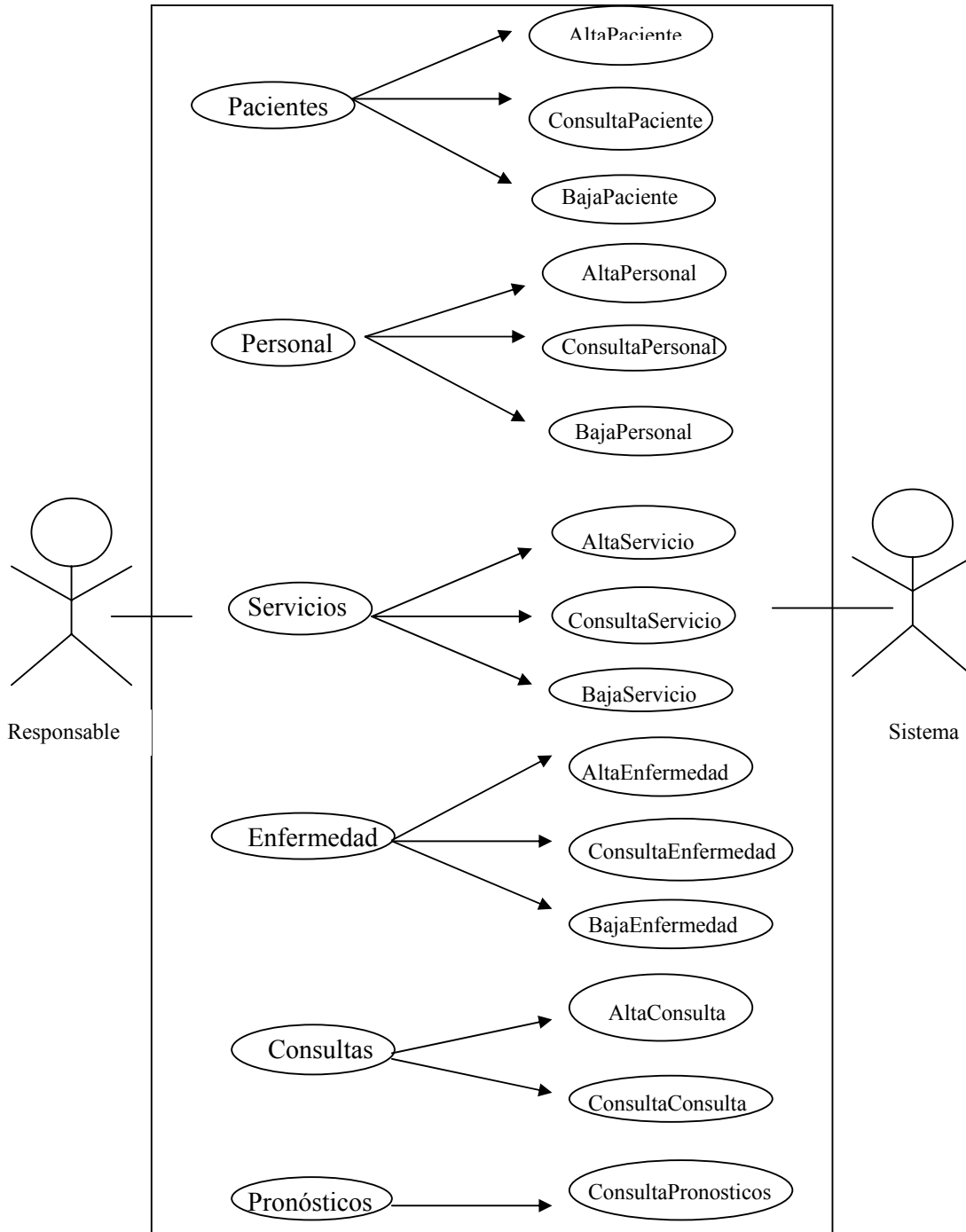


Figura 3.2 Caso de Uso Gestión de la Base de Datos.

Tabla 3.1 Caso de Uso de Pacientes.

Gestión de la Base de Datos	
DESCRIPCION DE CASOS DE USO	
Nombre:	Pacientes
Actores:	Responsable
Función:	Permitir el control de los pacientes.
Descripción:	El Responsable puede dar de Alta a un paciente introduciendo los datos más relevantes, posteriormente puede consultarlo por nombre, apellido paterno o materno, o por matrícula, y en la ventana de consulta aparece la opción de dar de Baja al paciente seleccionado.
Referencias:	De requerimientos: Ninguno
	De Casos: Ninguno

The screenshot shows a web application window titled 'Sistema de Control del Hospital Universitario de Puebla'. The main menu includes 'Nuestra Institucion', 'Pacientes', 'Personal', 'Servicios', 'Enfermedad', 'Consultas', 'Pronosticos', and 'Informacion Tecnica'. The 'Pacientes' menu is active, showing sub-options 'Alta Paciente' and 'Consulta Paciente'. The main content area is titled 'Ingresar Paciente a la Base de Datos' and contains the following form fields:

- Nombre :
- Apellido Paterno :
- Apellido Materno :
- Matricula :
- Direccion :
- Telefono :
- Derechohabiente :

At the bottom of the form are two buttons: 'Limpiar Forma' and 'Ingresar Paciente'.

Figura 3.3 Pantalla de Alta de Pacientes del Caso de Uso Pacientes.

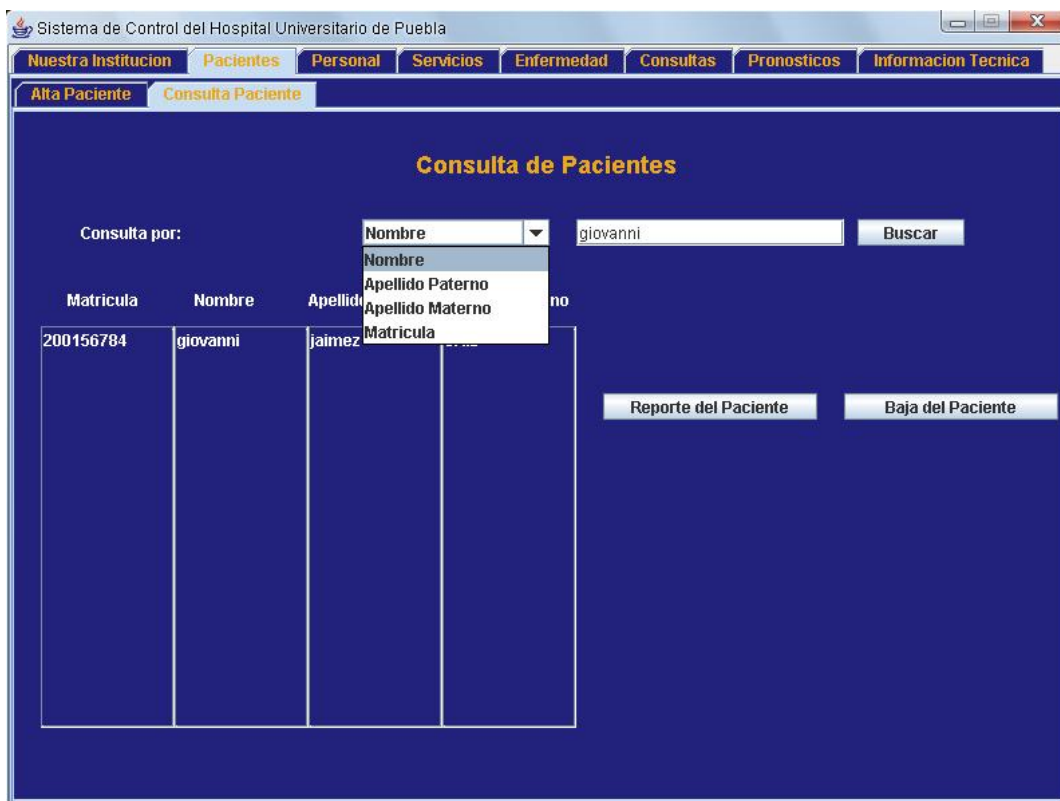


Figura 3.4 Pantalla de Consulta de Pacientes del Caso de Uso Pacientes.

Tabla 3.2 Caso de Uso de Personal.

Gestión de la Base de Datos	
DESCRIPCION DE CASOS DE USO	
Nombre:	Personal
Actores:	Responsable
Función:	Permitir el control del personal que labora en la institución.
Descripción:	El Responsable puede dar de Alta a un trabajador introduciendo los datos más relevantes, posteriormente puede consultarlo por nombre, apellido paterno o materno, o por clave, y en la ventana de consulta aparece la opción de dar de Baja al trabajador seleccionado.
Referencias:	De requerimientos: Ninguno
	De Casos: Ninguno

The screenshot shows a web application window titled 'Sistema de Control del Hospital Universitario de Puebla'. The navigation menu includes 'Nuestra Institucion', 'Pacientes', 'Personal', 'Servicios', 'Enfermedad', 'Consultas', 'Pronosticos', and 'Informacion Tecnica'. The 'Personal' menu is active, and the sub-menu 'Alta Personal' is selected. The main content area is titled 'Ingresar Personal a la Base de Datos' and contains the following form fields:

- Nombre :
- Apellido Paterno :
- Apellido Materno :
- Matricula :
- Direccion :
- Telefono :
- Extension :
- Cubiculo :
- Estado Civil :
- Puesto :
- E-mail :
- Grado de Estudio :
- Area de Trabajo :

At the bottom of the form are two buttons: 'Limpiar Forma' and 'Ingresar Personal'.

Figura 3.5 Pantalla de Alta de Personal del Caso de Uso de Personal.

The screenshot shows the 'Consultas' screen in the same application. The navigation menu is identical to the previous screen. The sub-menu 'Consulta Personal' is selected. The main content area is titled 'Consultas' and features a search form:

Consulta de Personal por

A dropdown menu is open over the search input, showing the following options: 'Nombre', 'Apellido Paterno', 'Apellido Materno', and 'Clave'. The 'Nombre' option is currently selected.

Below the search form is a table with the following data:

Clave	Nombre	Apellido	Apellido Materno	Clave
70037	Rosalba	ALvarado		

At the bottom right of the screen are two buttons: 'Reporte del Personal' and 'Baja del Personal'.

Figura 3.6 Pantalla de Consulta de Personal del Caso de Uso Personal.

Tabla 3.3 Caso de Uso de Servicios.

Gestión de la Base de Datos	
DESCRIPCION DE CASOS DE USO	
Nombre:	Servicios
Actores:	Responsable
Función:	Permitir el control de los servicios que se prestan en la institución.
Descripción:	El Responsable puede dar de Alta un servicio, posteriormente puede consultarlo por código o por nombre, y en la ventana de consulta aparece la opción de dar de Baja al servicio seleccionado.
Referencias:	De requerimientos: Ninguno
	De Casos: Personal

Figura 3.7 Pantalla de Alta de Servicios del Caso de Uso Servicios.

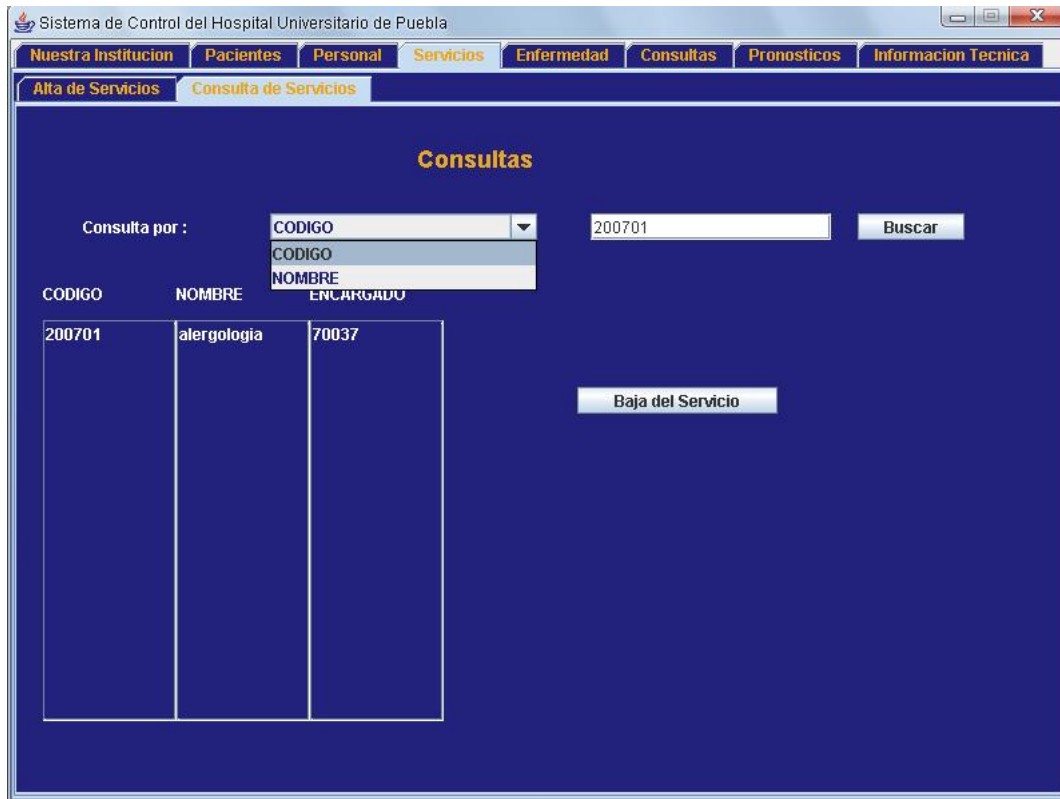


Figura 3.8 Pantalla de Consulta de Servicios del Caso de Uso de Servicios.

Tabla 3.4 Caso de Uso de Enfermedad.

Gestión de la Base de Datos	
DESCRIPCION DE CASOS DE USO	
Nombre:	Enfermedad
Actores:	Responsable
Función:	Permitir el control de las enfermedades en las que se dividen los servicios que se prestan en la institución.
Descripción:	El Responsable puede dar de Alta una enfermedad, posteriormente puede consultarla por especialidad o por nombre, y en la ventana de consulta aparece la opción de dar de Baja la enfermedad seleccionada.
Referencias:	De requerimientos: Ninguno
	De Casos: Servicios

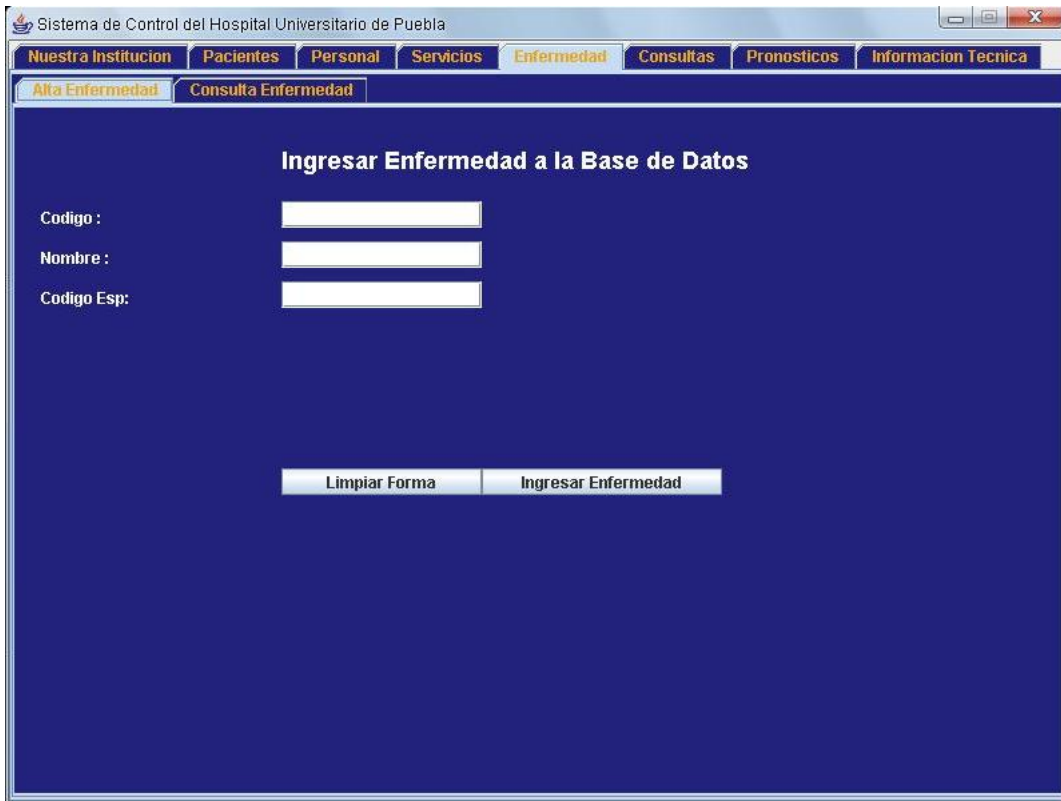


Figura 3.9 Pantalla de Alta de Enfermedades del Caso de Uso de Enfermedad.

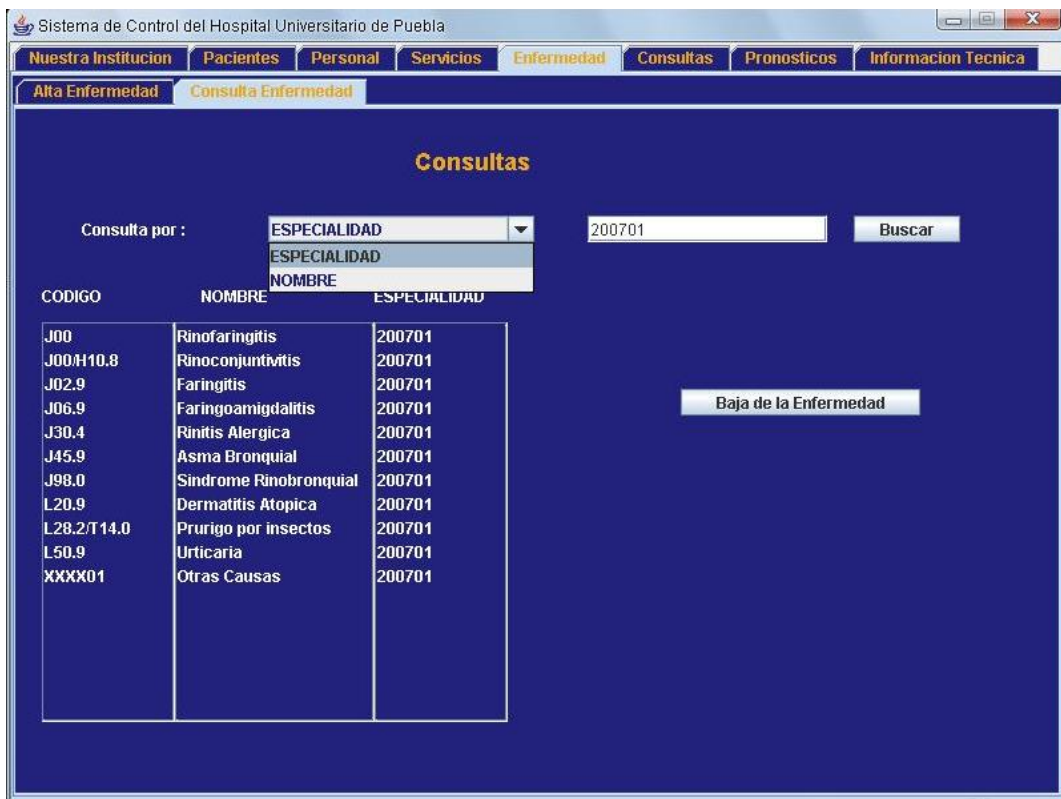


Figura 3.10 Pantalla de Consulta de Enfermedades del Caso de Uso Enfermedad.

Tabla 3.5 Caso de Uso de Consultas.

Gestión de la Base de Datos	
DESCRIPCION DE CASOS DE USO	
Nombre:	Consultas
Actores:	Responsable
Función:	Permitir el control de las consultas realizadas en la institución.
Descripción:	El Responsable puede dar de Alta una consulta, posteriormente puede crear escenarios de consulta, los cuales contemplan día, mes, año y servicio.
Referencias:	De requerimientos: Ninguno
	De Casos: Pacientes, Enfermedad, Servicios

Figura 3.11 Pantalla de Alta de Consultas del Caso de Uso de Consultas.

DIA	MES	AÑO	COD PACIENTE	COD ENFERMEDAD	COD SERVICIO	SUBTOTAL
0	1	1996	0	XXXX01	200701	238
0	1	1996	0	XXXX02	200702	17
0	1	1996	0	XXXX03	200703	6
0	1	1996	0	XXXX04	200704	84
0	1	1996	0	XXXX05	200705	92
0	1	1996	0	XXXX06	200706	83
0	1	1996	0	XXXX07	200707	62
0	1	1996	0	XXXX08	200708	23
0	1	1996	0	XXXX09	200709	75
0	1	1996	0	XXXX10	200710	0
0	1	1996	0	XXXX11	200711	174
0	1	1996	0	XXXX12	200712	61
0	1	1996	0	XXXX13	200713	0
0	1	1996	0	XXXX14	200714	274
0	1	1996	0	XXXX15	200715	17
0	1	1996	0	XXXX16	200716	437
0	1	1996	0	XXXX17	200717	26

Figura 3.12 Pantalla de Consulta de Consultas del Caso de Uso de Consultas.

Tabla 3.6 Caso de Uso de Pronósticos.

Gestión de la Base de Datos	
DESCRIPCION DE CASOS DE USO	
Nombre:	Pronósticos
Actores:	Responsable
Función:	Crear escenarios de consulta para su posterior pronóstico.
Descripción:	El Responsable crea escenarios de consulta para poder pronosticarlos posteriormente.
Referencias:	De requerimientos: Ninguno
	De Casos: Consultas

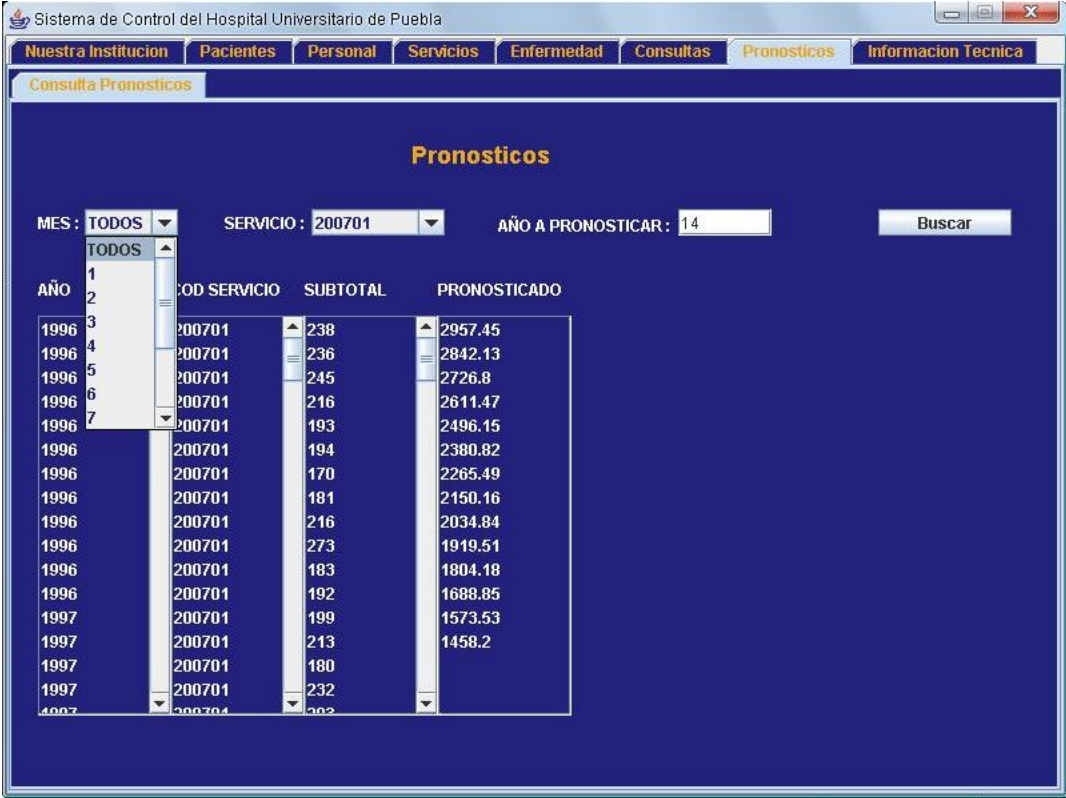


Figura 3.13 Pantalla de Pronósticos del Caso de Uso de Pronósticos.

CASO DE USO 2: Aplicación Móvil

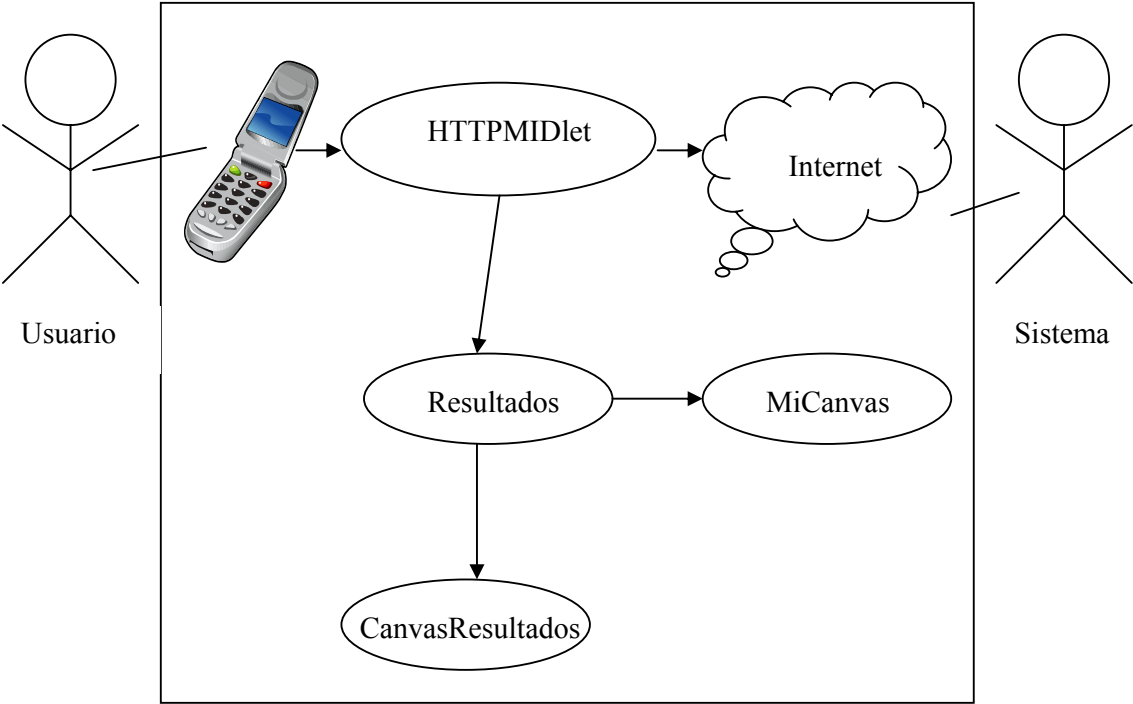


Figura 3.14 Caso de Uso de la Aplicación Móvil

Tabla 3.7 Caso de Uso de HTTPMIDlet.

Aplicación Móvil	
DESCRIPCION DE CASOS DE USO	
Nombre:	HTTPMIDlet
Actores:	Usuario
Función:	Captura los datos y Realiza la conexión con el servidor web.
Descripción:	El usuario crea el escenario a pronosticar, la función envía los parámetros, recibe respuesta en forma de cadena y la envía a la función Resultados.
Referencias:	De requerimientos: R1
	De Casos: Ninguno.

Para entender mejor el funcionamiento del sistema móvil se han impreso algunos ejemplos, los cuales tienen cada una de las posibles combinaciones.

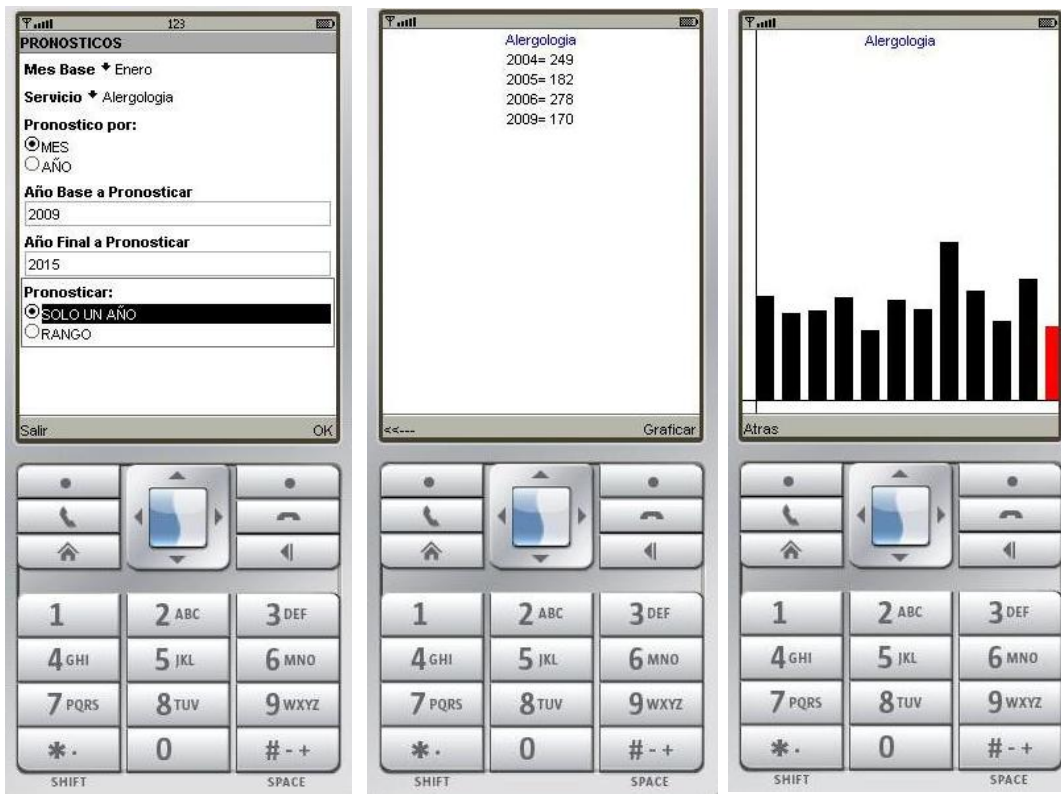


Figura 3.15 Ejemplo 1.

Ejemplo 1: Servicio de los 40 posibles: Alergología, Pronostico por: Mes, Mes: Enero, Año Base a pronosticar: 2009

Tabla 3.8 Caso de Uso de Resultados.

Aplicación Móvil	
DESCRIPCION DE CASOS DE USO	
Nombre:	Resultados
Actores:	Sistema
Función:	Recibe la cadena de respuesta de HTTPMIDlet
Descripción:	Esta función descompone la cadena almacenando los resultados en un arreglo, posteriormente envía 8 datos como máximo, además checa si existen mas resultados o no para enviar a la función CanvasResultados
Referencias:	De requerimientos: R1
	De Casos: HTTPMIDlet

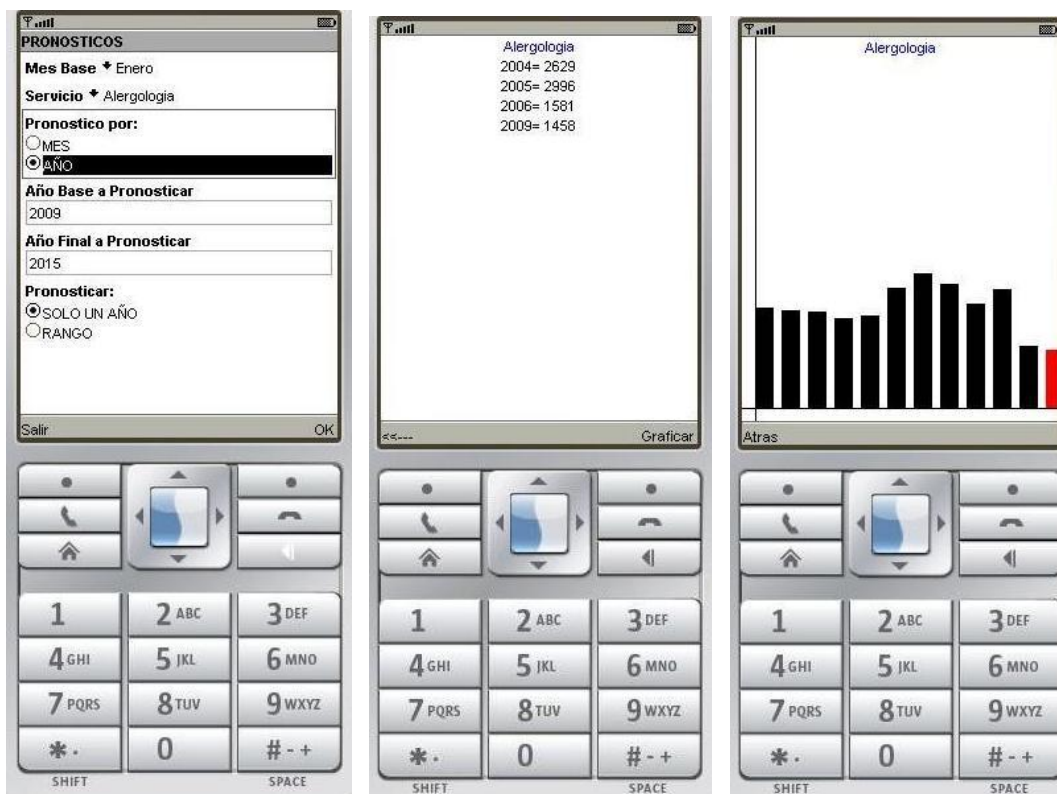


Figura 3.16 Ejemplo 2.

Ejemplo 2: Servicio de los 40 posibles: Alergología, Pronostico por: Año, Año Base a pronosticar: 2009.

Tabla 3.9 Caso de Uso de CanvasResultados.

Aplicación Móvil	
DESCRIPCION DE CASOS DE USO	
Nombre:	CanvasResultados
Actores:	Sistema
Función:	Recibe los datos de la función Resultados.
Descripción:	Esta función recibe los datos y los “pinta” en pantalla. Además de dar paso a la opción de Graficar.
Referencias:	De requerimientos: R1
	De Casos: Resultados

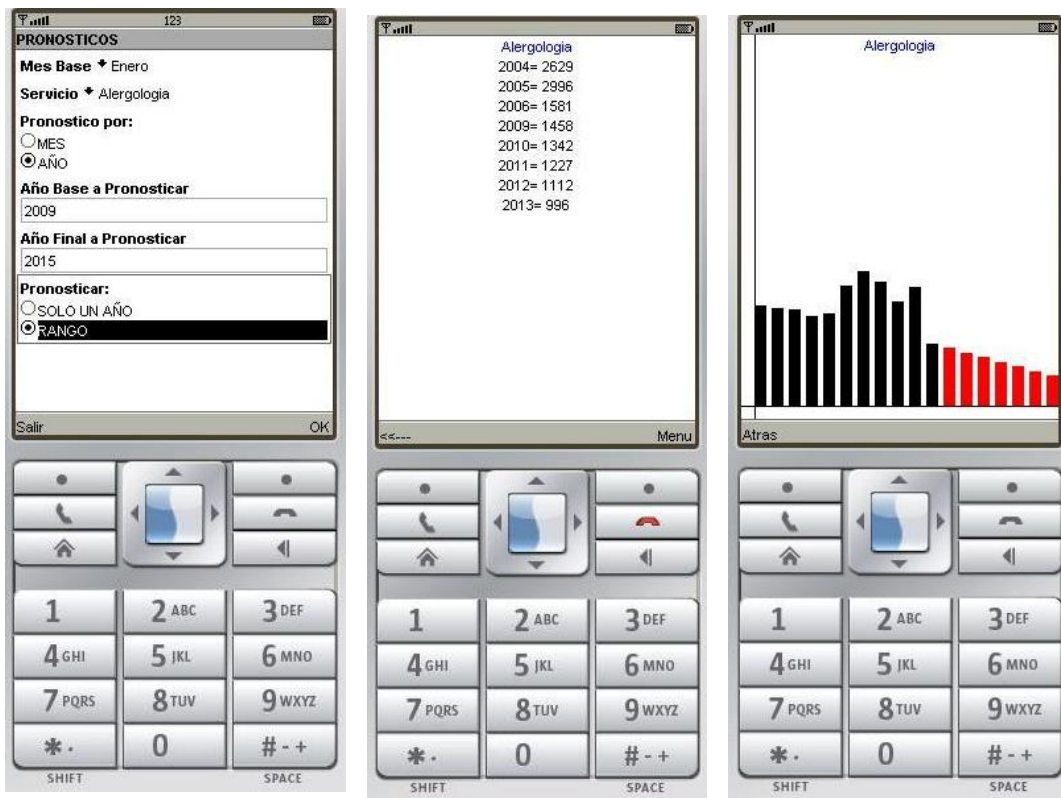


Figura 3.17 Ejemplo 3.

Ejemplo 3: Servicio de los 40 posibles: Alergología, Pronostico por: Año, Rango a pronosticar: 2009-2015.

Tabla 3.10 Caso de Uso de MiCanvas.

Aplicación Móvil	
DESCRIPCION DE CASOS DE USO	
Nombre:	MiCanvas
Actores:	Sistema
Función:	Recibe la cadena de respuesta de la función Resultados.
Descripción:	Esta función recibe la cadena de respuesta, la descompone en los datos y pinta en pantalla un Grafico de Barras escalado de acuerdo al dato mayor existente y al número de barras a desplegar.
Referencias:	De requerimientos: R1
	De Casos: Resultados

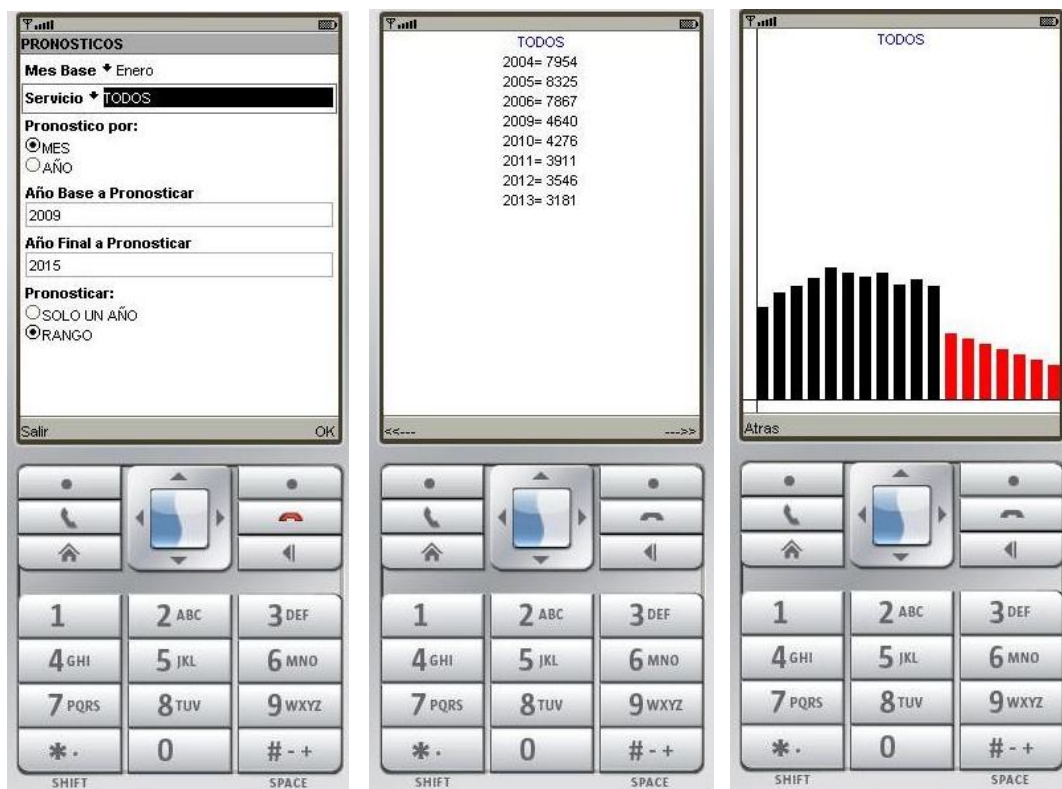


Figura 3.18 Ejemplo 4.

Ejemplo 4: Servicio de los 40 posibles: TODAS, Pronostico por: Mes, Mes: Enero, Rango a pronosticar: 2009-2015.

CASO DE USO 3: Servidor

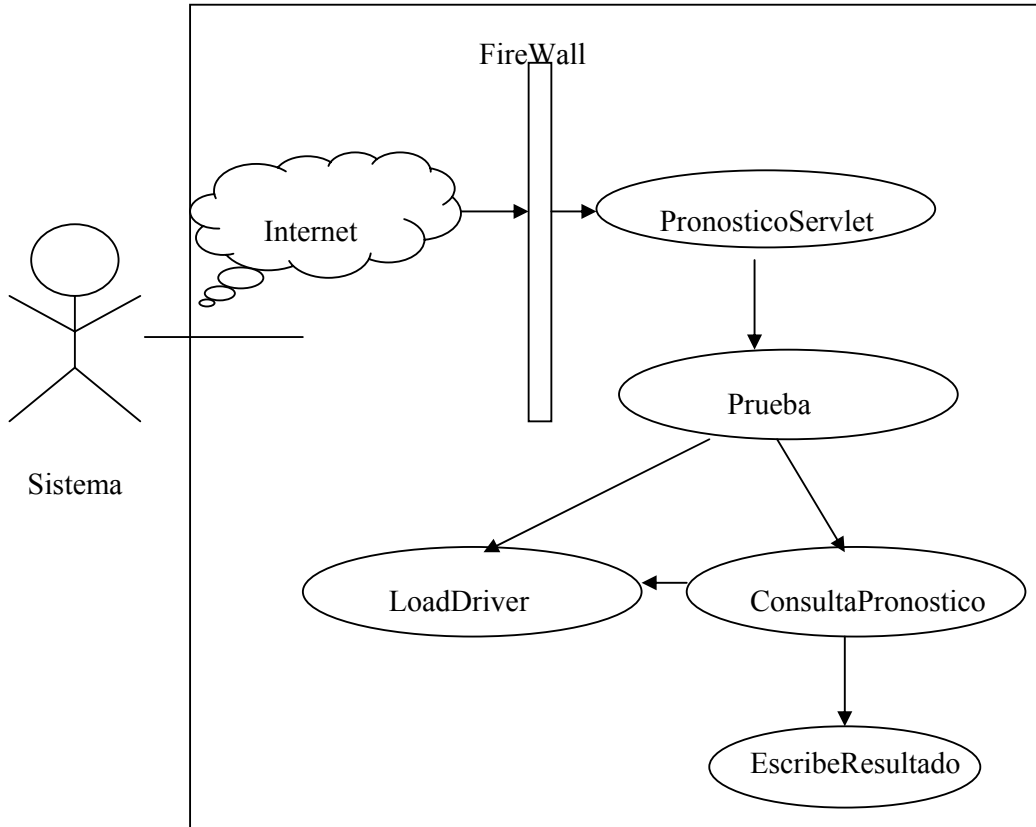


Figura 3.19 Caso de Uso del Servidor.

Tabla 3.11 Caso de Uso de PronosticoServlet.

Servidor	
DESCRIPCION DE CASOS DE USO	
Nombre:	PronosticoServlet
Actores:	Sistema
Función:	Recibe los parámetros enviados a través de Internet.
Descripción:	Este Servlet recibe los parámetros que introdujo el usuario en la pantalla principal del teléfono y que posteriormente se enviaron a través de Internet, luego los envía a la función Prueba.
Referencias:	De requerimientos: R2
	De Casos: HTTPMIDlet

Tabla 3.12 Caso de Uso de Prueba.

Servidor	
DESCRIPCION DE CASOS DE USO	
Nombre:	Prueba
Actores:	Sistema
Función:	Recibe los parámetros enviados por el servlet.
Descripción:	Esta función sirve de puente para poder dar mantenimiento a la parte del servlet, ya que podemos enviarle manualmente los parámetros de prueba o función, a la función que realiza la consulta.
Referencias:	De requerimientos: R2
	De Casos: PronosticoServlet

Tabla 3.13 Caso de Uso de LoadDriver.

Servidor	
DESCRIPCION DE CASOS DE USO	
Nombre:	LoadDriver
Actores:	Sistema
Función:	Establece la conexión con la Base de Datos
Descripción:	Esta función es la encargada de abrir y cerrar la conexión con la BD, además de ejecutar las sentencias creadas en la función ConsultaPronostico.
Referencias:	De requerimientos: R1,R2, Servicio de WAMP.
	De Casos: Prueba,ConsultaPronostico

Tabla 3.14 Caso de Uso de ConsultaPronostico.

Servidor	
DESCRIPCION DE CASOS DE USO	
Nombre:	ConsultaPronostico
Actores:	Sistema
Función:	Recibe los parámetros, y realiza la consulta en la Base de Datos.
Descripción:	Esta función recibe los parámetros introducidos por el usuario, y crea las sentencias necesarias para la consulta a la BD, posteriormente envía estos resultados al método de pronostico.
Referencias:	De requerimientos: R2
	De Casos: Prueba, LoadDriver

Tabla 3.15 Caso de Uso de EscribeResultados.

Servidor	
DESCRIPCION DE CASOS DE USO	
Nombre:	EscribeResultados
Actores:	Sistema
Función:	Calcula el pronóstico del escenario de consultas.
Descripción:	Este método recibe los resultados de la consulta a la BD, y calcula la línea de regresión del pronóstico, además de que envía de regreso una cadena con todos los datos resultados del pronostico separados con una diagonal "/".
Referencias:	De requerimientos: R1, R2
	De Casos: LoadDriver, ConsultaPronostico

3.4 Modelo Conceptual

En los siguientes diagramas se puede observar la conexión que existe entre las clases y la comunicación existente entre ellas, además del papel que desempeña cada una en el correcto funcionamiento del sistema.

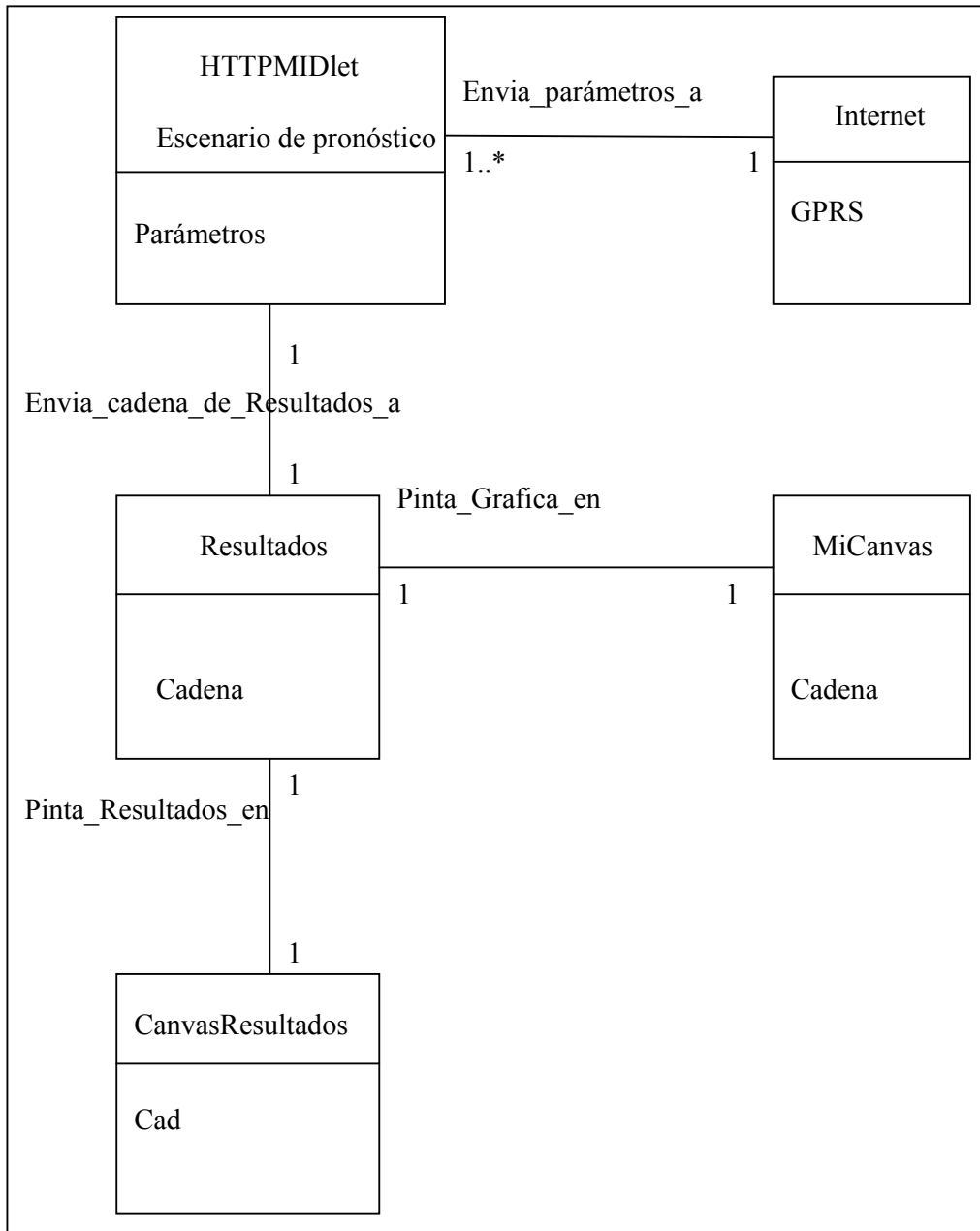


Figura 3.20 Diagrama UML de la Aplicación Móvil.

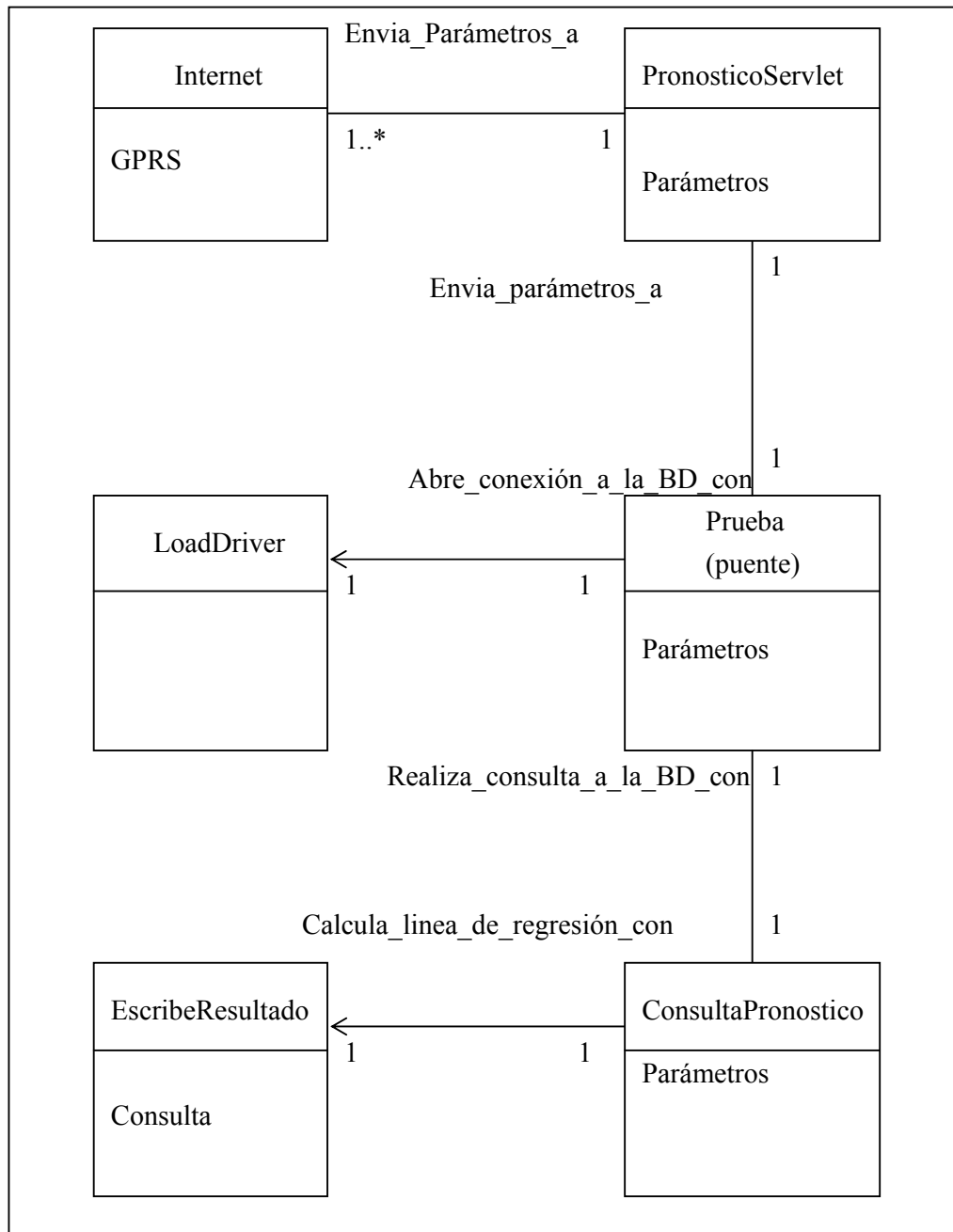


Figura 3.21 Diagrama UML del Servidor.

3.5 Clases Principales

La descripción de todas y cada una de las clases que componen a nuestro sistema están descritas en los Diagramas UML que a continuación se presentan [36].

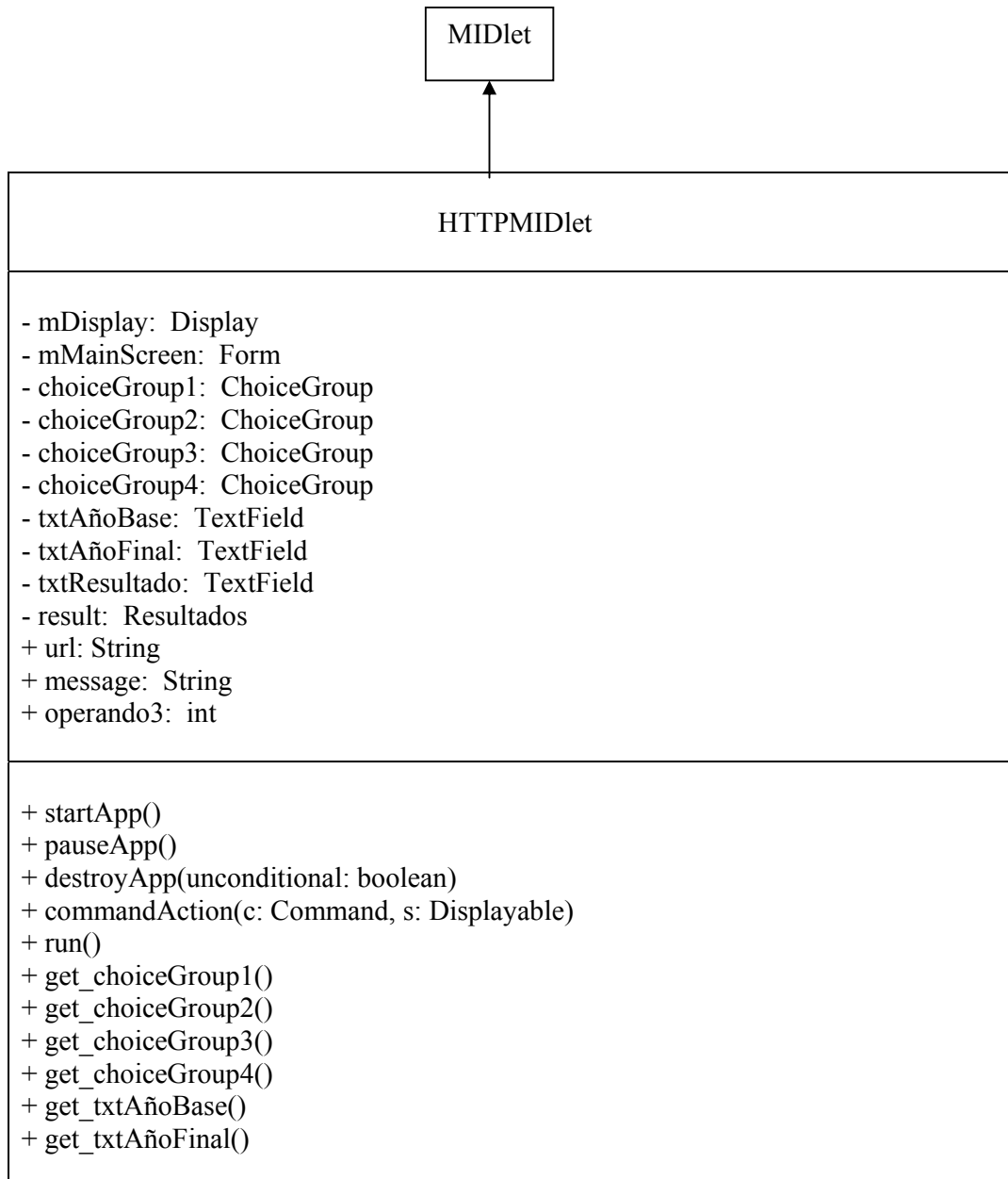


Figura 3.22 Diagrama UML de la clase HTTPMIDlet.

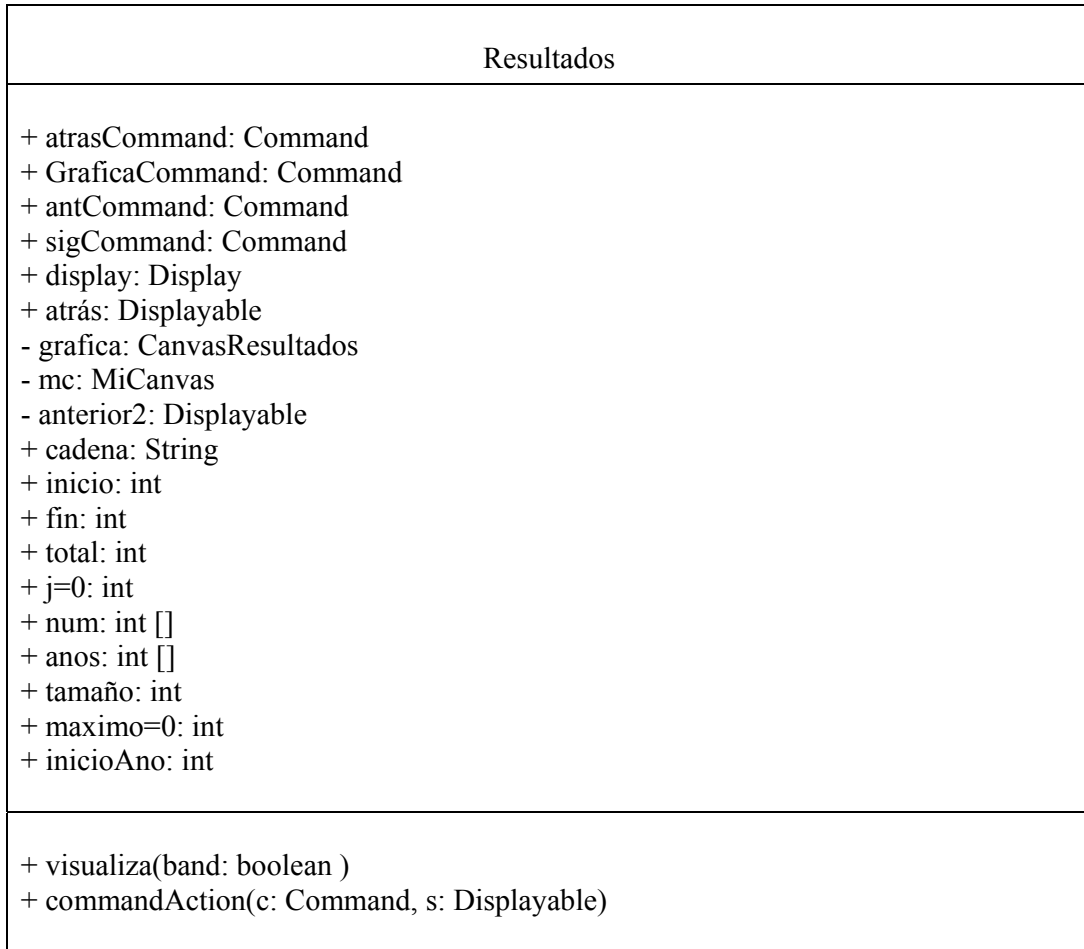


Figura 3.23 Diagrama UML de la clase Resultados.

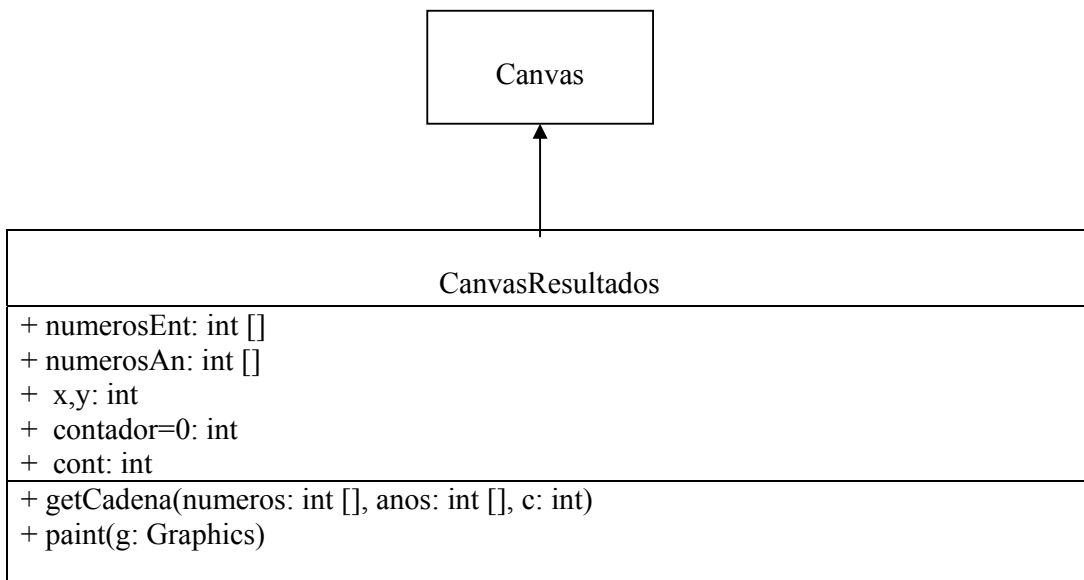


Figura 3.24 Diagrama UML de la clase CanvasResultados.

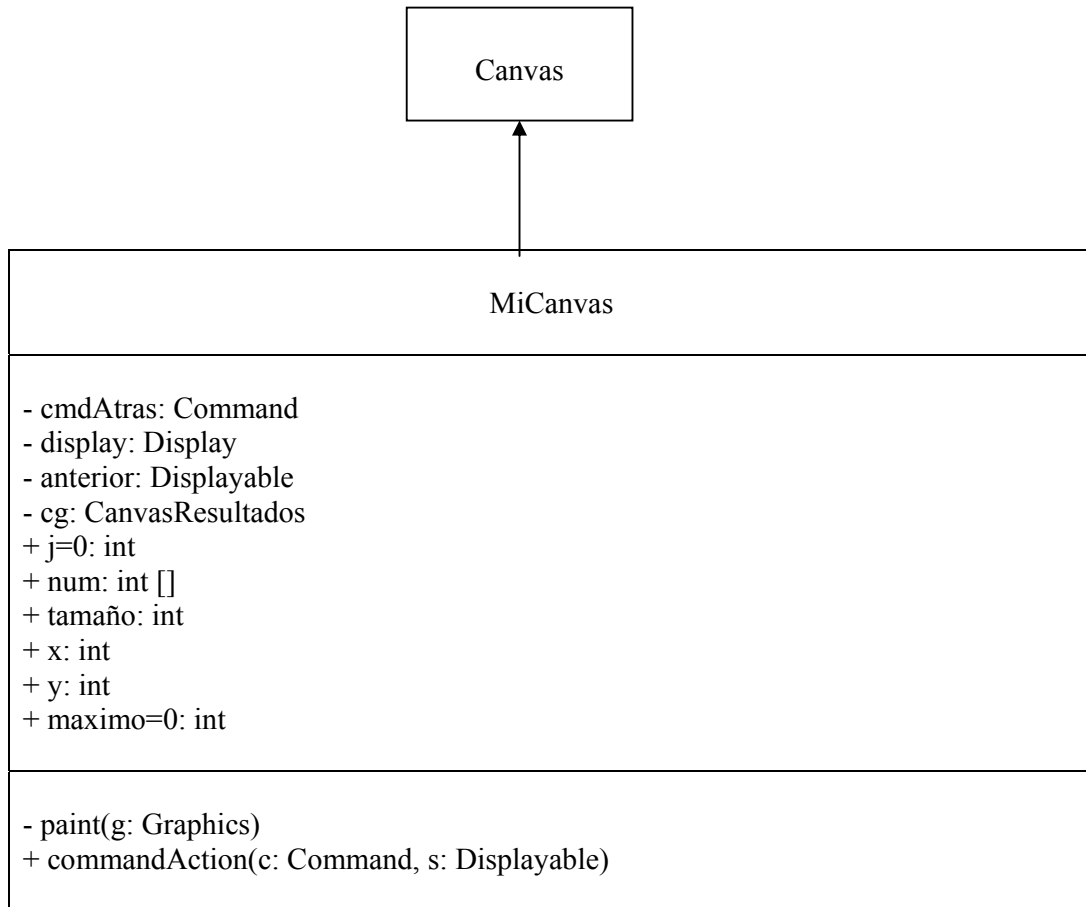


Figura 3.25 Diagrama UML de la clase MiCanvas.

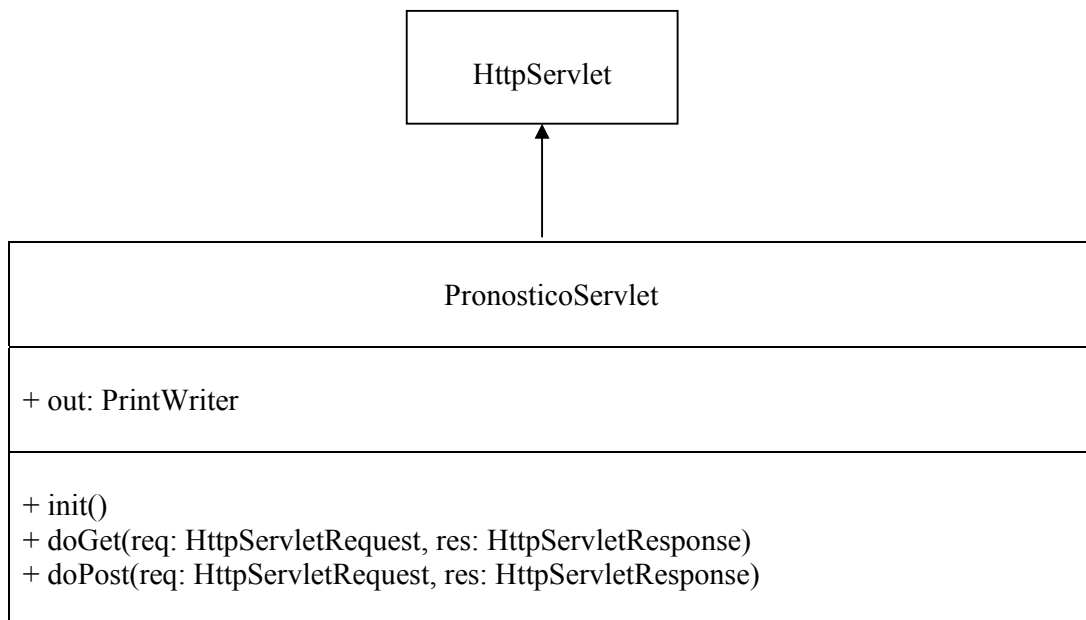


Figura 3.26 Diagrama UML de la clase PronosticoServlet.

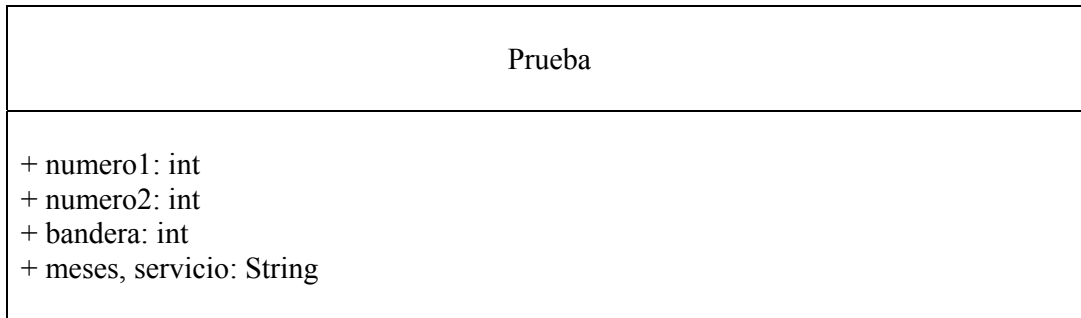
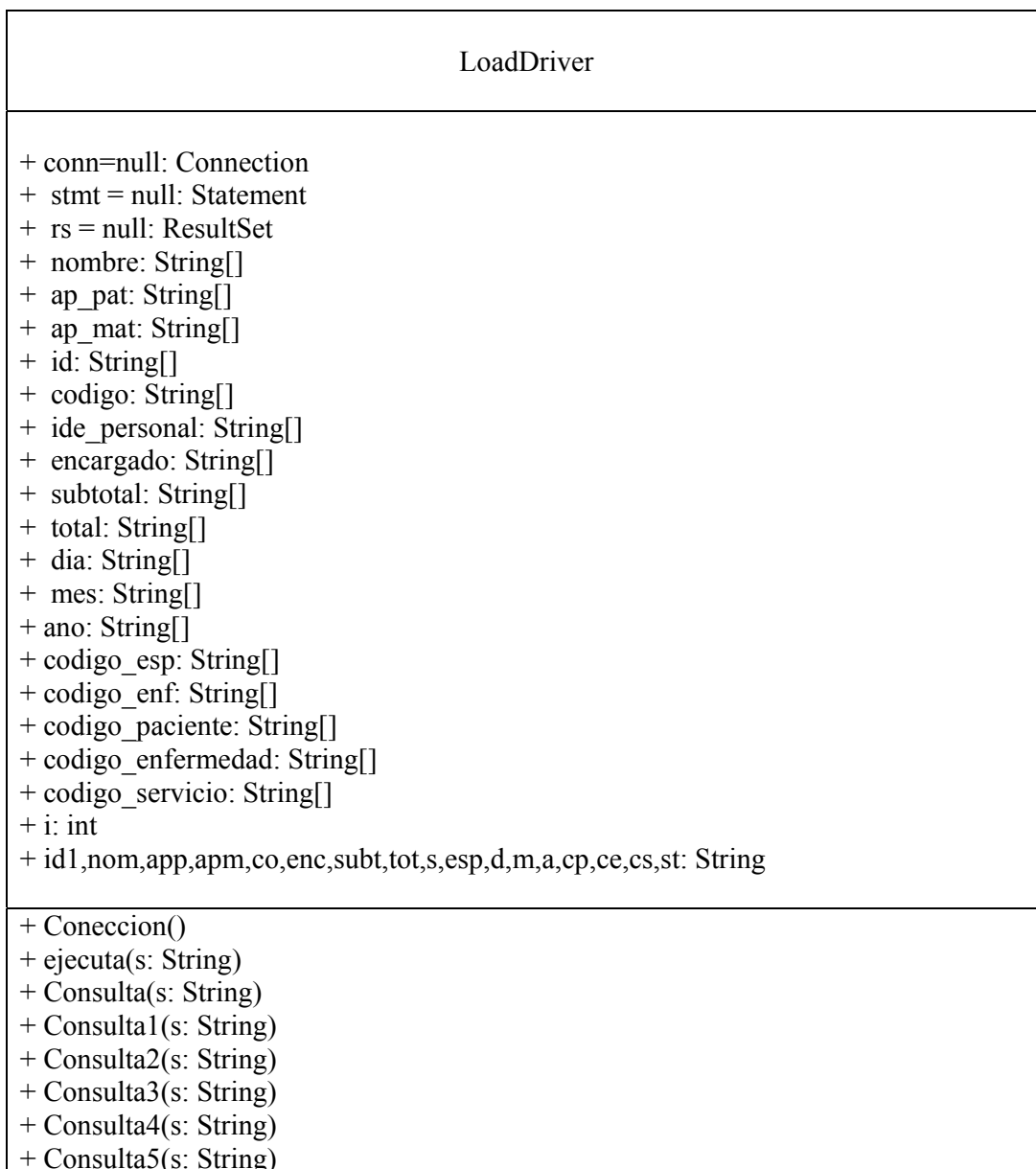


Figura 3.27 Diagrama UML de la clase Prueba.



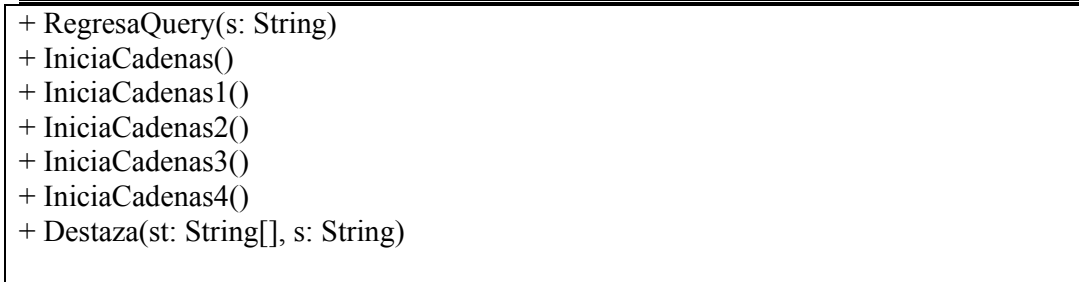


Figura 3.28 Diagrama UML de la clase LoadDriver.

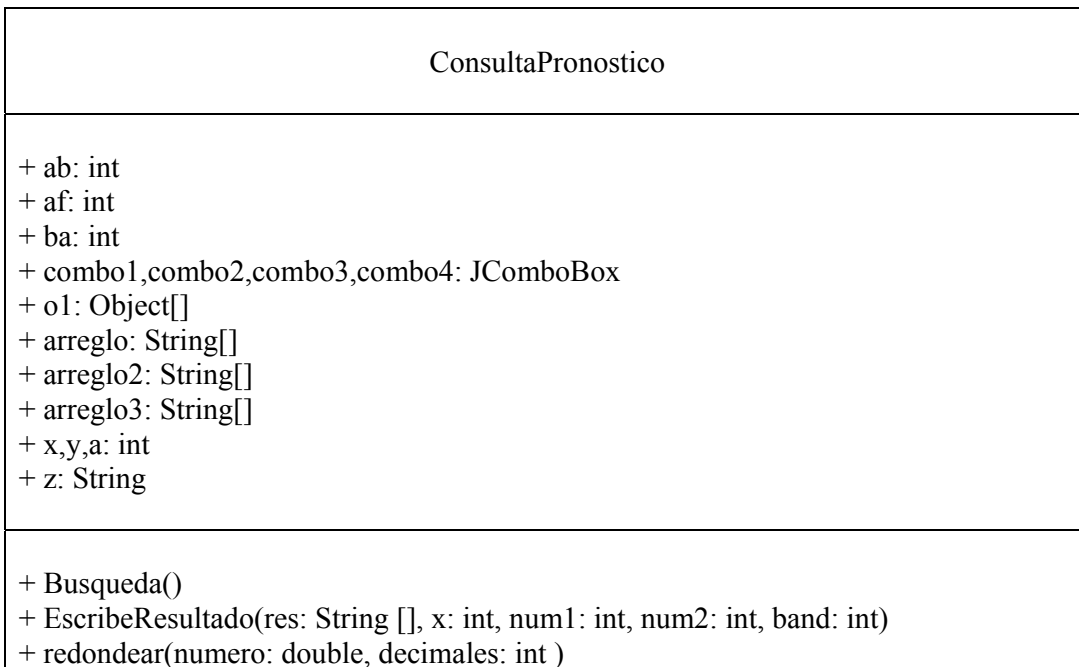


Figura 3.29 Diagrama UML de la clase ConsultaPronostico.

Capítulo IV

Herramientas de software utilizadas

4.1 Telefonía Móvil [20]

G1

La primera generación de telefonía móvil (**G1**) funcionaba por medio de comunicaciones analógicas y dispositivos portátiles que eran relativamente grandes. Esta generación utilizaba principalmente los siguientes estándares:

- **AMPS** (Sistema telefónico móvil avanzado): Se presentó en 1976 en Estados Unidos y fue el primer estándar de redes celulares. Utilizada principalmente en el continente americano, Rusia y Asia, la primera generación de redes analógicas contaba con mecanismos de seguridad endebles que permitían hackear las líneas telefónicas.
- **TACS** (Sistema de comunicaciones de acceso total): Es la versión europea del modelo AMPS. Este sistema fue muy usado en Inglaterra y luego en Asia (Hong-Kong y Japón) y utilizaba la banda de frecuencia de 900 MHz.
- **ETACS** (Sistema de comunicaciones de acceso total extendido): Es una versión mejorada del estándar TACS desarrollado en el Reino Unido que utiliza una gran cantidad de canales de comunicación.

Con la aparición de una segunda generación totalmente digital, la primera generación de redes celulares se volvió obsoleta.

G2

La segunda generación de redes móviles (**G2**) marcó un quiebre con la primera generación de teléfonos celulares al pasar de tecnología analógica a digital. Los principales estándares de telefonía móvil de G2 son:

- **GSM** (*Sistema global para las comunicaciones móviles*): El estándar más usado en Europa a fines de siglo XX y también se admite en Estados Unidos. Este estándar utiliza las bandas de frecuencia de 900 MHz y de 1800 MHz en Europa. Sin embargo, en Estados Unidos la banda de frecuencia utilizada es la de 1900 MHz. Por lo tanto, los teléfonos móviles que pueden funcionar tanto en Europa como en Estados Unidos se denominan teléfonos de **tribanda**.

- **CDMA** (*Acceso múltiple por división de código*): Utiliza una tecnología de espectro ensanchado que permite transmitir una señal de radio a través de un rango de frecuencia amplio.
- **TDMA** (*Acceso múltiple por división de tiempo*): Emplea una técnica de división de tiempo de los canales de comunicación para aumentar el volumen de los datos que se transmiten simultáneamente. Esta tecnología se usa, principalmente, en el continente americano, Nueva Zelanda y en la región del Pacífico asiático.

Gracias a la G2, es posible transmitir voz y datos digitales de volúmenes bajos, por ejemplo, mensajes de texto (**SMS** siglas en inglés de *Servicio de mensajes cortos*) o mensajes multimedia (**MMS** siglas en inglés de *Servicio de mensajes multimedia*). El estándar GSM permite una velocidad de datos máxima de 9,6 kbps.

Se han hecho ampliaciones al estándar GSM con el fin de mejorar el rendimiento. Una de esas extensiones es el servicio **GPRS** (*Servicio general de paquetes de radio*) que permite velocidades de datos teóricas en el orden de los 114 Kbits/s pero con un rendimiento cercano a los 40 Kbits/s en la práctica. Como esta tecnología no se encuentra dentro de la categoría "G3", se la llama **G2.5**.

El estándar **EDGE** (*Velocidades de datos mejoradas para la evolución global*) anunciado como **G2.75**, cuadruplica las mejoras en el rendimiento de GPRS con la tasa de datos teóricos anunciados de 384 Kbps, por lo tanto, admite aplicaciones de multimedia. En realidad, el estándar EDGE permite velocidades de datos teóricas de 473 Kbits/s pero ha sido limitado para cumplir con las especificaciones IMT-2000 (*Telecomunicaciones móviles internacionales-2000*) de la ITU (*Unión internacional de telecomunicaciones*).

G3

Las especificaciones IMT-2000 (*Telecomunicaciones móviles internacionales para el año 2000*) de la Unión internacional de telecomunicaciones (ITU) definieron las características de la **G3** (tercera generación de telefonía móvil). Las características más importantes son:

- Alta velocidad de transmisión de datos :
- 144 Kbps con cobertura total para uso móvil.
- 384 Kbps con cobertura media para uso de peatones.
- 2 Mbps con áreas de cobertura reducida para uso fijo.
- Compatibilidad mundial.
- Compatibilidad de los servicios móviles de G3 con las redes de segunda generación.

La G3 ofrece velocidades de datos de más de 144 Kbit/s y de este modo brinda la posibilidad de usos multimedia, por ejemplo, transmisión de videos, video conferencias o acceso a Internet de alta velocidad. Las redes de G3 utilizan bandas con diferentes frecuencias a las redes anteriores: 1885 a 2025 MHz y 2110 a 2200 MHz.

El estándar G3 más importante que se usa en Europa se llama **UMTS** (*Sistema universal de telecomunicaciones móviles*) y emplea codificación **W-CDMA** (*Acceso múltiple por división de código de banda ancha*). La tecnología UMTS usa bandas de 5 MHz para transferir voz y datos con velocidades de datos que van desde los 384 Kbps a los 2 Mbps. El **HSDPA** (*Acceso de alta velocidad del paquete de Downlink*) es un protocolo de telefonía móvil de tercera generación, apodado "G3.5", que puede alcanzar velocidades de datos en el orden de los 8 a 10 Mbps. La tecnología HSDPA usa la banda de frecuencia de 5 GHz y codificación W-CDMA.

Tabla 4.1 Cuadro sinóptico de Tecnologías Móviles.

Estándar	Generación	Banda de frecuencia	Rendimiento	
GSM	G2	Permite la transferencia de voz o datos digitales de bajo volumen.	9,6 kbps	9,6 kbps
GPRS	G2.5	Permite la transferencia de voz o datos digitales de volumen moderado.	21,4 a 171,2 kbps	48 kbps
EDGE	G2.75	Permite la transferencia simultánea de voz y datos digitales.	43,2 a 345,6 kbps	171 kbps
UMTS	G3	Permite la transferencia simultánea de voz y datos digitales a alta velocidad.	0,144 a 2 Mbps	384 kbps

4.2 GPRS (General Packet Radio Service)

Para que el dispositivo móvil pudiera efectuar los protocolos de comunicación mencionados a lo largo de la presente Tesis, se contrato el servicio de navegación en Internet con la empresa prestadora del servicio, en modo GPRS, lo cual consta de un paquete de datos que cuesta 4 Centavos por cada Kb recibido, para tal efecto nuestra aplicación recibe alrededor de .775 Kb, demostrando así que aunque es una aplicación para altos ejecutivos, su costo de operación no es excesivo [41].

Definición

Protocolo inalámbrico, no de voz, que ofrece acceso a redes de datos mediante la conmutación de paquetes. El servicio ofrece tasas de transmisión de datos de hasta 170 Kbps. La característica más notable de esta tecnología es que provee una conexión permanente entre la red y la Terminal móvil. El término permanente se refiere a que la transmisión de datos no es orientada a conexión. Es decir, en este tipo de tecnología no necesita abrir y cerrar una conexión, únicamente debe preocuparse por el envío de paquetes.

El uso de esta tecnología requiere de equipos móviles capaces de interactuar con sitios web que ofrecen servicios multimedia y que contienen aplicaciones que permitan el manejo de estos servicios. [21]

Arquitectura

La popularidad que ha adquirido GPRS junto a GSM es notable. Desde hace algunos años la tecnología GSM ha empezado su auge en México. GSM en conjunción

con GPRS brinda a los usuarios nuevos servicios nunca antes brindados mediante la tecnología celular digital. La arquitectura del sistema que permite ofrecer estos servicios está representada en la figura (4.2.1).

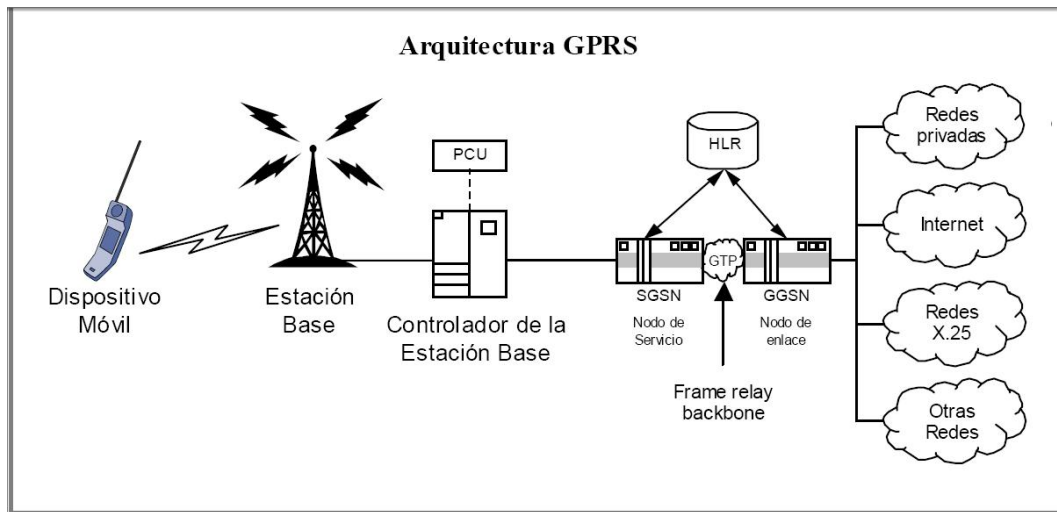


Figura 4.1 Arquitectura GPRS.

Los servicios que provee GPRS utilizan la misma interfaz aérea que GSM. Con el fin de proveer la transmisión de datos GPRS define una infraestructura extra, sobre la infraestructura GSM existente. Así, un dispositivo móvil se comunica con el Controlador de la Estación Base mediante la Estación Base. El CEB de GSM existente solo maneja voz, adicionalmente requiere de una unidad de control de paquetes (PCU), quien será encargada de desviar los paquetes GPRS hasta el nodo de soporte de servicios GPRS (SGSN). El nodo SGSN se encarga de mandar y recibir los datos a los dispositivos móviles, además interactúa con un registro de ubicación local (HLR) para identificar a los dispositivos disponibles en su área de servicio y así hacerse cargo del hand-off de las sesiones GPRS. Este nodo a su vez se conecta al nodo de soporte gateway GPRS (GGSN) mediante frame relay. En este punto la comunicación se realiza mediante un protocolo llamado GPRS tunnel protocol (GTP) que encapsula paquetes IP o X.25 para ser transmitidos entre el SGSN y el GGSN. Finalmente, el GGSN es quien mantiene al sistema conectado a redes como Internet, X.25 y otras redes privadas. Este dispositivo actúa, además, como firewall protegiendo a la infraestructura GPRS de cualquier ataque que pueda provenir por la Internet. También participa junto con el HLR para desviar el tráfico SGSN adecuado, ya que puede haber más de un SGSN en la infraestructura GPRS [41], [42].

4.3 NetBeans de Sun Microsystems

NetBeans es un IDE (*Integrated Development Environment* o, en español, Entorno Integrado de Desarrollo) escrito en Java que agrupa un conjunto de utilidades que facilitan la edición, compilación, depuración, análisis y ejecución de un programa Java. [22]

Con él se pueden diseñar interfaces gráficas de usuario casi sin escribir código, al más puro estilo visual de Delphi o C++, sólo con hacer clic en los componentes sobre la ventana seleccionada.

Además de ser gratuito y de acogerse al proyecto “*open-source project*”, NetBeans se caracteriza por su entorno de desarrollo intuitivo, personalizable, modular y extensible. Es precisamente esta última característica la que permite ampliar el paquete de clases que trae por defecto NetBeans 5.5.1 con una extensión especial para programadores J2ME que, no sólo cuenta con todas las clases necesarias para crear un *midlet* sino que añade al IDE una serie de herramientas para facilitar la integración con las aplicaciones J2SE o J2EE [44].

Al ser NetBeans la herramienta que mas se nos facilito de aprender, y para la que encontramos mas tutoriales sobre su aprendizaje, para llevar a buen puerto este proyecto, creemos conveniente exponer aquí unas nociones sobre cómo crear una pequeña aplicación visual.

Creación de una aplicación Web

Una aplicación Web, es un contenedor de servlets, a los cuales se puede acceder desde nuestra aplicación móvil, y a continuación se describe la manera de crearlo fácil y rápido, utilizando las pantallas que nos proporciona nuestro IDE.

Dar click en Nuevo Proyecto-->Web--->Aplicación Web y Botón de Siguiente (Figura 4.2).

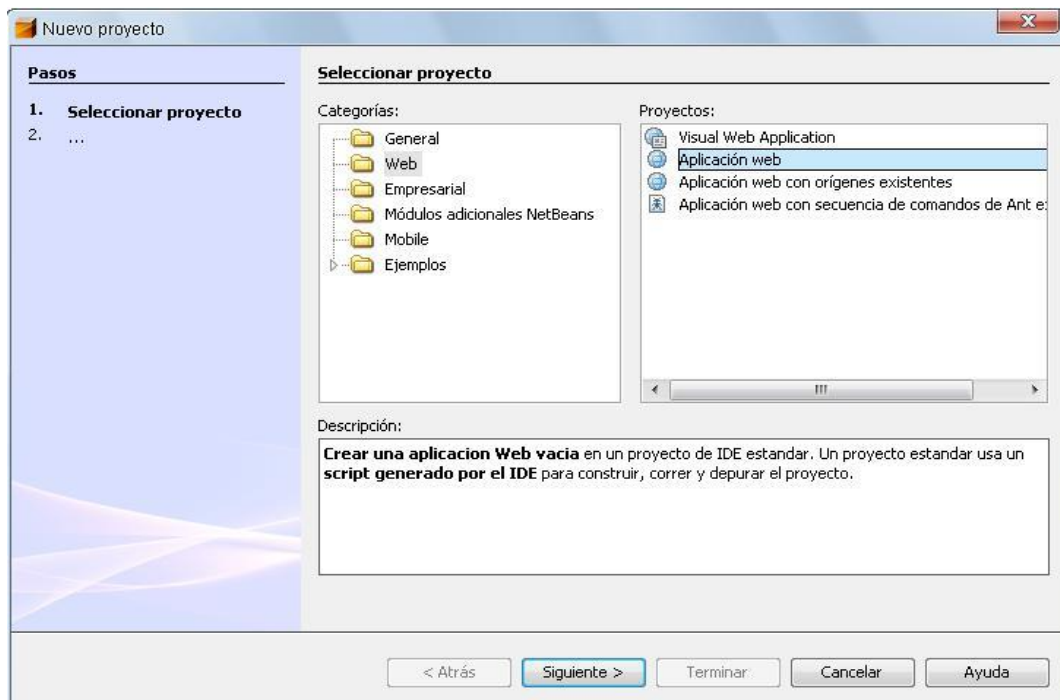


Figura 4.2 Pantalla Nuevo Proyecto.

Introducimos el nombre que tendrá, y la dirección donde estará alojada, lo demás lo dejamos como esta por default, y por ultimo damos en el botón de Finalizar (Figura 4.3)

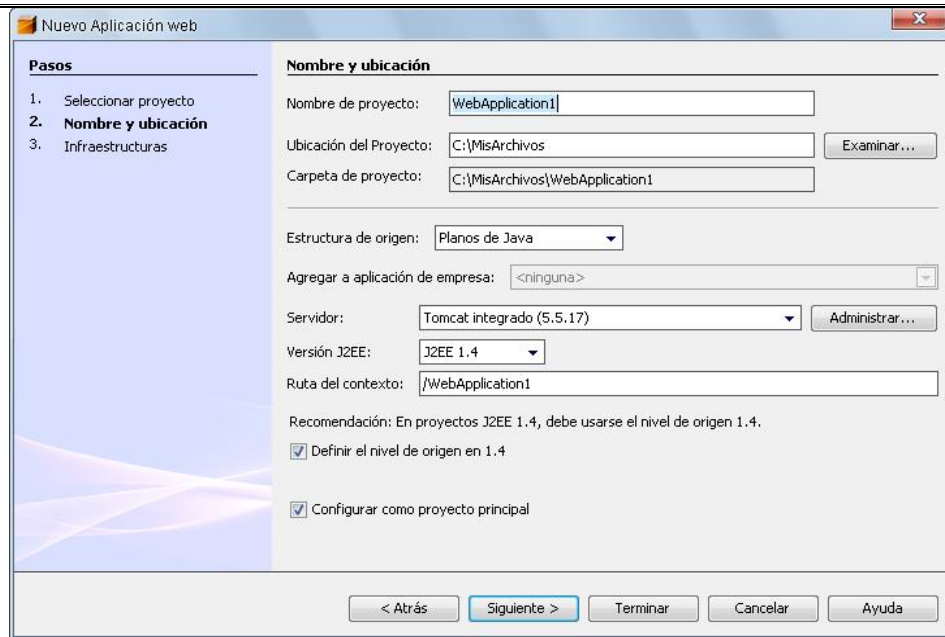


Figura 4.3 Pantalla Nuevo Aplicación Web.

Añadir nuevos servlets

Una vez que hemos creado y ejecutado nuestra aplicación web sin ningún error, podemos agregarle los servlets necesarios para establecer la comunicación con nuestra aplicación web.

Desplegamos nuestro árbol de Proyecto de nuestra aplicación Web, desglosamos nuestro Paquete de Origen, y en el Paquete Determinado damos click derecho --->>> Nuevo --->>> Servlet. (Figura 4.4).

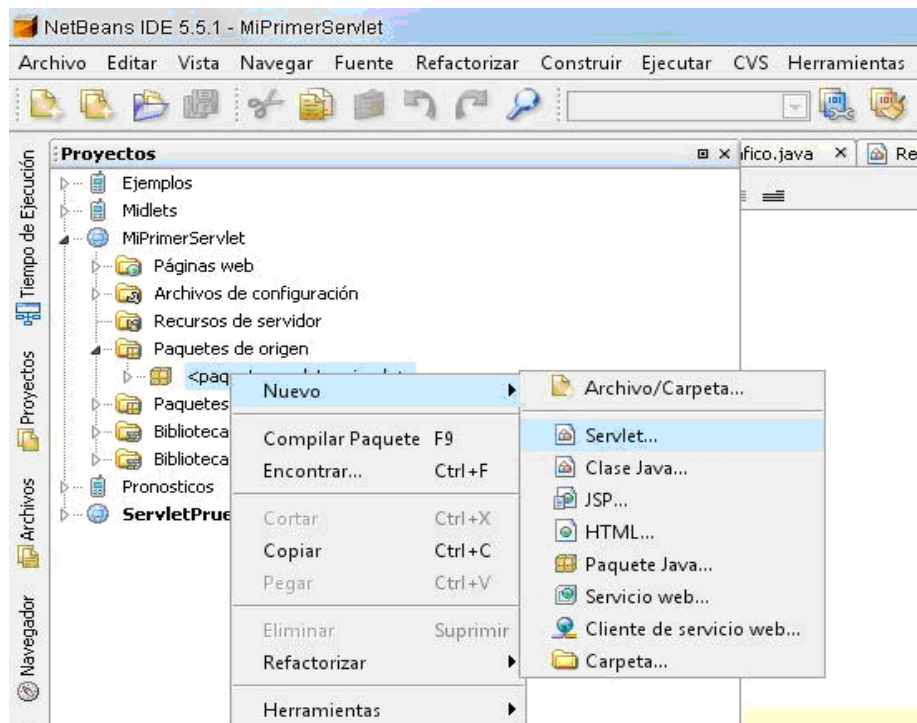


Figura 4.4 Pantalla IDE Nuevo→Servlet.

Escribimos el nombre de nuestro servlet, y damos click en finalizar, y nos aparecerá el esqueleto de nuestro servlet, el cual podremos modificar (Figura 4.5).

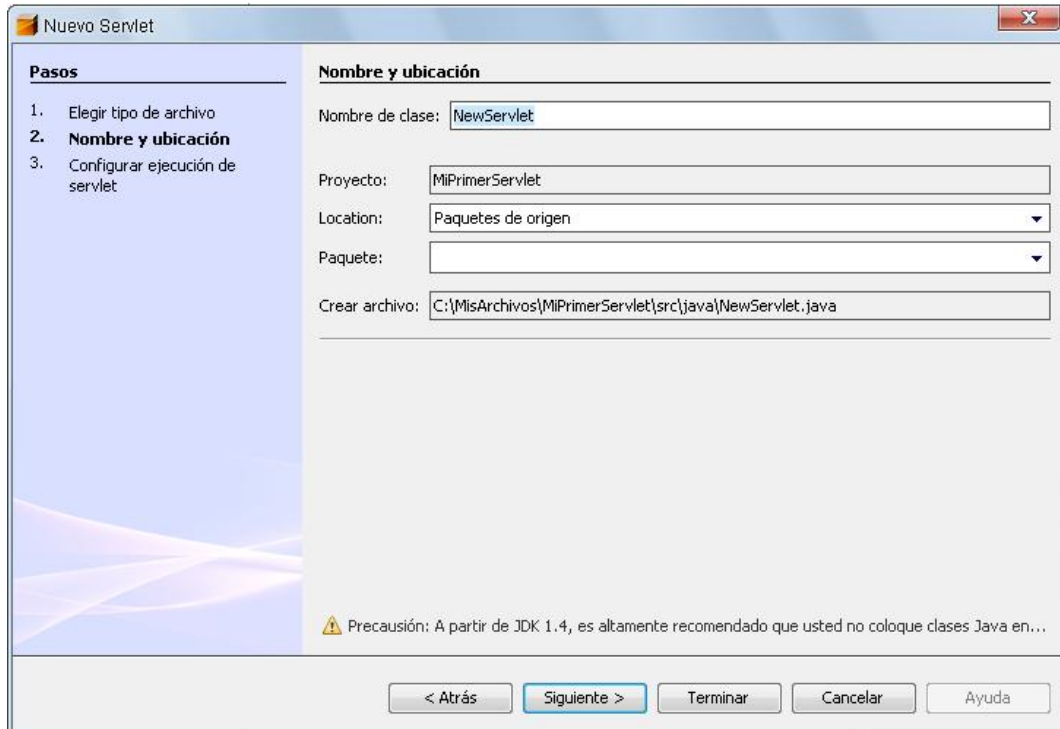


Figura 4.5 Pantalla Nuevo→Servlet.

Creación de un MIDlet con conexión a Internet

Ya que hemos agregado nuestro primer servlet de prueba, ahora crearemos un MIDlet para poder acceder de manera local desde nuestro emulador [43].

```
import java.io.*;
import javax.microedition.io.*;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

/**
 * An example MIDlet to invoke a CGI script.
 */

public class FirstMidletServlet extends MIDlet {

    private Display display;

    String url = "http://localhost:8084/ServletPrueba/HelloServlet";

    public FirstMidletServlet() {
        display = Display.getDisplay(this);
    }

    /**
```

```
* Initialization. Invoked when we activate the MIDlet.
*/
public void startApp() {
    try {
        invokeServlet(url);
    } catch (IOException e) {
        System.out.println("IOException " + e);
        e.printStackTrace();
    }
}

/**
 * Pause, discontinue ....
 */
public void pauseApp() {

}

/**
 * Destroy must cleanup everything.
 */
public void destroyApp(boolean unconditional) {
}

/**
 * Prepare connection and streams then invoke servlet.
 */

void invokeServlet(String url) throws IOException {
    HttpConnection c = null;
    InputStream is = null;
    StringBuffer b = new StringBuffer();
    TextBox t = null;
    try {
        c = (HttpConnection)Connector.open(url);
        c.setRequestMethod(HttpConnection.GET);
        c.setRequestProperty("IF-Modified-Since", "20 Jan 2001 16:19:14 GMT");
        c.setRequestProperty("User-Agent", "Profile/MIDP-1.0 Configuration/CLDC-1.0");
        c.setRequestProperty("Content-Language", "en-CA");
        is = c.openDataInputStream();
        int ch;
        while ((ch = is.read()) != -1) {
            b.append((char) ch);
            System.out.println((char)ch);
        }
        //String message;
        t = new TextBox("First Servlet", b.toString(), 1024, 0);
        //message = b.toString();
    }
    finally {
        if(is != null) {
            is.close();
        }
        if(c != null) {
            c.close();
        }
    }
    display.setCurrent(t);
}
}
```

Y por el otro lado, nuestra aplicación web, contendrá el servlet HelloServlet:

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

/**
 * The simplest possible servlet.
 */

public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("Servlet invoked!");
        out.println(new Date());
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String firstName = request.getParameter("firstName").toString();
        System.out.println("firstName = " + firstName);
        processRequest(request, response);
    }

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String firstName = request.getParameter("firstName").toString();
        String surname = request.getParameter("surname").toString();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet GreetingServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet GreetingServlet at " + request.getContextPath () +
"</h1>");
        out.println("<p>Welcome " + firstName + " " + surname + "</p>");
        out.println("</body>");
        out.println("</html>");

        out.close();
    }
}
```

J2ME versus NetBeans

Aprovechando las ventajas que nos ofrece Nuestro IDE, hemos decidido, además de albergar nuestra aplicación Web, también utilizarlo para crear nuestro proyecto, ya que de otra manera debemos, en primer lugar crear un archivo con nuestro MIDlet, después compilarlo, y por último ejecutarlo en nuestro emulador, pero cada una de estas fases se presentan por separado, siendo muy engorroso al trabajar de esta manera, es así como NetBeans nos ofrece de manera gráfica y en una sola pantalla cada una de estas fases, pasando fácilmente de una a la otra sin necesidad de abrir pantallas adicionales para nuestro proyecto.

4.4 El servidor de servlets: Tomcat 5

Tomcat (también llamado *Jakarta Tomcat* o *Apache Tomcat*) funciona como contenedor de servlets y es desarrollado bajo el proyecto Jakarta [22] en la *Apache Software Foundation*. Implementa las tecnologías *Java Servlet 2.4* y *JavaServer Pages 2.0* (JSP) de *Sun Microsystems*.

Tomcat es un servidor de aplicaciones que, a diferencia de un servidor Web, como es por ejemplo *Apache*, incluye un contenedor Web que puede servir páginas dinámicas (a diferencia del servidor Web, que solo sirve páginas HTML estáticas). Incluye el compilador Jasper, que compila páginas JSPs y las convierte en servlets. Además, funciona con cualquier sistema operativo que disponga de máquina virtual Java, ya que fue escrito en este mismo lenguaje.

Historia

James Duncan Davidson se puede considerar el padre de Tomcat. Trabajaba como arquitecto de software para Sun Microsystems en el momento en que decidió donar el proyecto a la Apache Software Foundation .

Eligió el nombre de Tomcat (gato) pretendiendo representar la capacidad del programa de ser independiente, de cuidarse por sí mismo. Aunque el verdadero motivo es que Duncan esperaba que Tomcat se convirtiese en *open source* y que O'Reilly (famosa editorial norteamericana dedicada a la informática) publicase un libro sobre su proyecto. O'Reilly es conocida por asociar animales a las portadas de sus libros.

En el momento de escribir este documento Apache Tomcat se encuentra en su versión 5.5.1

4.5 NetBeans Versión 5.5.1 con MySQL

Conexión de NetBeans 5.5.1 con MySQL

Ahora que tenemos configurado nuestra aplicación Móvil con nuestra aplicación Web, necesitamos configurar la conexión de nuestro IDE con MySQL, ya que haremos uso de nuestra Base de Datos para realizar las consultas necesarias para generar los datos base para crear nuestro escenario de pronóstico.

Para ello necesitamos el mismo conector que utilizamos para Java, y aquí se explica brevemente en que consiste:

La intención es crear una nueva librería a nuestro proyecto de modo que contendrá todas y cada una de las clases necesarias para realizar la conexión con la BD.

1. Primero crearemos un proyecto normal (Aplicación Java)
2. Abrimos el manejador de bibliotecas. **Menu Herramientas -> Administrador de Librerías** (Figura 4.6).

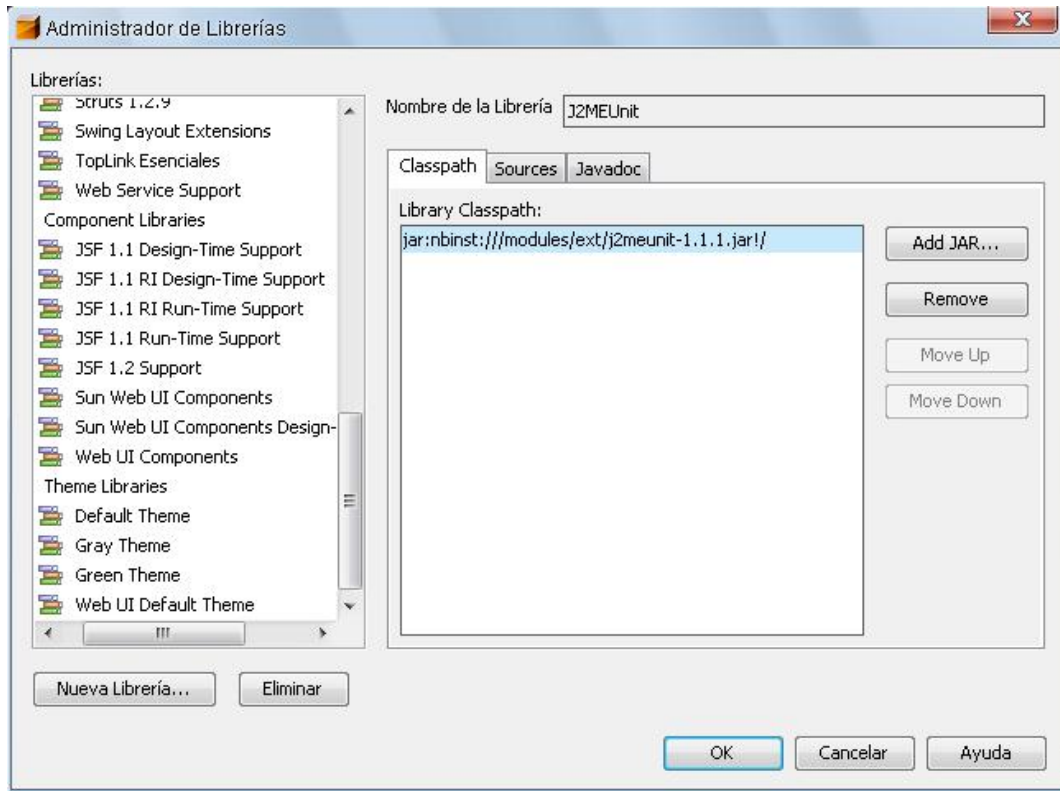


Figura 4.6 Administrador de Librerías.

3. Nos aparecerá el cuadro de diálogo Administrador de Librerías. Ahora debemos dar de alta una nueva biblioteca dando click en **Nueva Librería...**
4. En el cuadro que nos aparece, debemos llenarlo de la siguiente forma: (Figura 4.7).

Nombre de la Librería: MySQL (no puede tener espacios)

Tipo de Librería: Librería de clases.

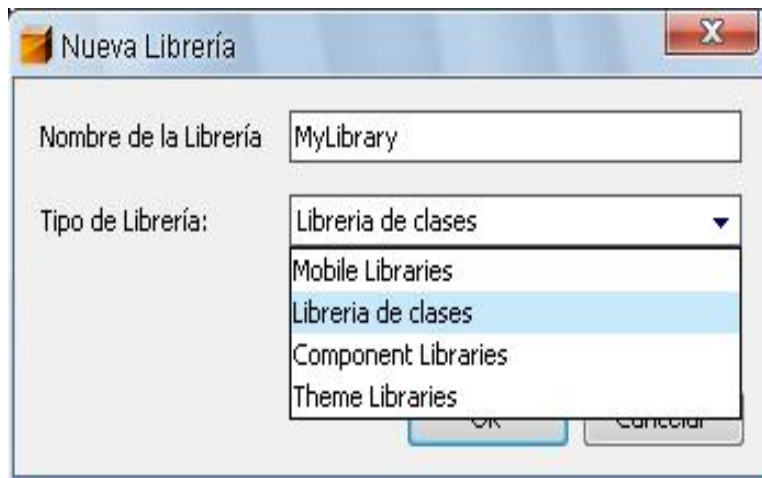


Figura 4.7 Nueva Librería.

Click en el botón OK

5. Ahora debemos agregar unos detalles a la biblioteca. Asegúrese que la nueva biblioteca creada este seleccionada (izquierda).
6. Seleccionamos la pestaña Class Path y damos clic en el botón **Agregar archivo JAR/carpeta**
7. Se abre un dialogo llamado Browse JAR/Folder que permite seleccionar archivos. Buscamos el jar descargado para conectarnos a MySQL (mysql-3.1.11.jar) los seleccionamos y damos clic en el botón **Agregar archivo JAR/carpeta**
8. Aparecerá en la sección Class Path el archivo que acabamos de agregar, damos clic en Ok del cuadro de dialogo Administrador de Librerías.
9. Ahora debemos agregar la librería al proyecto. Dentro del proyecto (explorador del proyecto) seleccionar el nodo Bibliotecas. (Figura 4.8)

Dar click derecho para mostrar el menú contextual

- Seleccionar opción Agregar Biblioteca
- Seleccionar de la lista, la biblioteca recién creada con el nombre MySQL

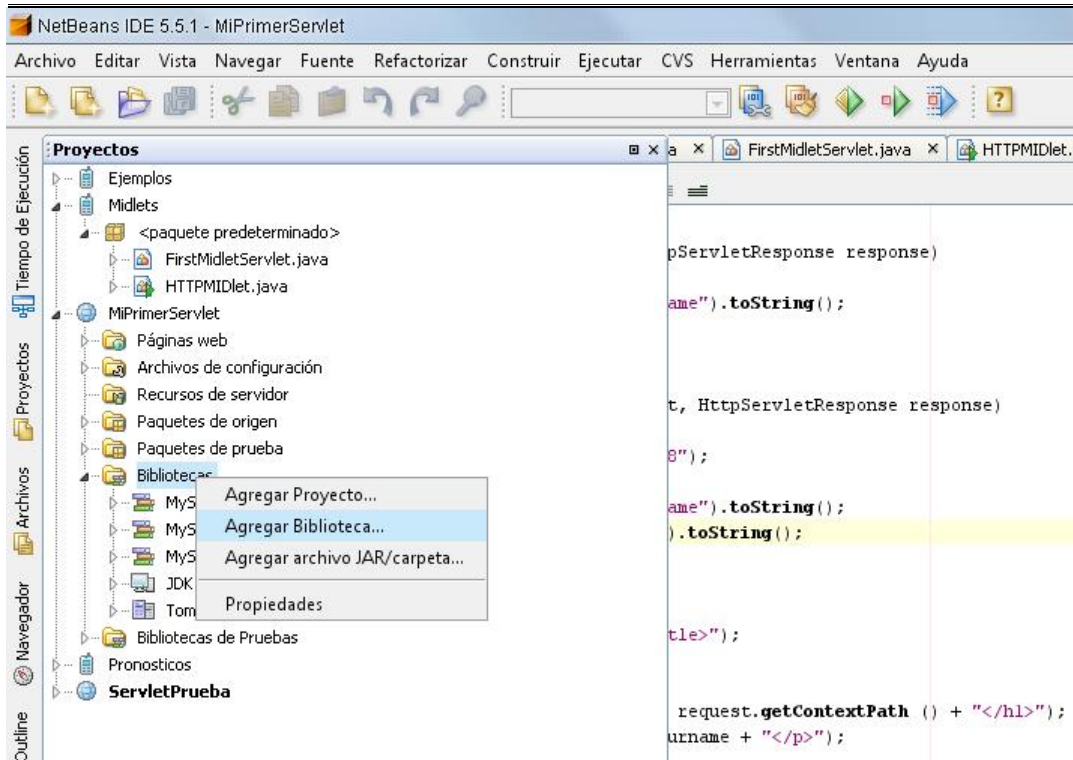


Figura 4.8 Agregar Biblioteca.

Debe aparecer como una nueva entrada dentro del nodo Bibliotecas. Y Listo.

Podemos escribir el siguiente programa para verificar que todo funciona.

```
import java.sql.*;
public class TestConnection {
    static String bd = "tu_BD";
    static String login = "usuario";
    static String password = "contraseña";
    static String url = "jdbc:mysql://localhost/"+bd;

    public static void main(String[] args) throws Exception { Connection conn = null;
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection(url,login,password);
        if (conn != null) {
            System.out.println("Conexión a base de datos "+url+" ... Ok");
            conn.close();
        }
    }
    catch(SQLException ex) {
        System.out.println("Hubo un problema al intentar conectarse con la base de datos "+url);
    }
    catch(ClassNotFoundException ex) {
        System.out.println(ex);
    }
    }//MAIN
} //CLASS
```

Deberá aparecer conexión a base de datos "+url+" ... Ok

Capítulo V

Planeación basada en regresión lineal

Una de las estrategias empleadas y ad-hoc para predecir el comportamiento de muchas de las actividades de una institución de salud es la regresión lineal. Esta estrategia permite a través del coeficiente de correlación determinar que tan certero es el pronóstico generado por la regresión lineal.

5.1 Pronósticos estadísticos [23]

Actualmente en nuestro estado, se vive una gran demanda de servicios hospitalarios, es por eso que el Gobierno del Estado en conjunción con los gobiernos Locales y Federales ha puesto en marcha la construcción de mas y mejores Hospitales para poder solventar dicha demanda. Es por tal motivo que el Hospital Universitario de la BUAP (HU) se ve obligado a buscar mayor eficiencia en sus procesos médicos, específicamente en poder avanzar para brindar un mejor servicio y no quedarse rezagado ante el continuo avance que se tiene en el aspecto de infraestructura Medica. Es por eso que se ve en la necesidad de pronosticar la demanda de sus servicios para establecer un plan de operación que le permita atender oportunamente y de la mejor manera a todos sus pacientes. Los procesos orientados al suministro (compras, ampliación y distribución) están enfocados a garantizar disponibilidad de servicios con eficiencia y al mejor costo y su desempeño depende que las diferentes áreas en que se divide dicho Hospital pronostiquen lo mejor posible la demanda. En este sentido, un tema que actualmente interesa es como pronosticar con mas certeza la demanda de servicios Hospitalarios en el HU.

Panorama general

Tomando como base los procesos empresariales de elaboración de pronósticos, en esta tesis se tiene la intención de poder adaptarlo a la demanda Hospitalaria para poder tener una buena planificación de recursos y así poder brindar un mejor servicio en el HU. No es nuevo para nadie que el reto es lograr disponibilidad en cantidad necesaria y cuando lo requiere el paciente.

Para entregar un mejor nivel de servicio de manera rentable el HU requiere desarrollar una cultura de predicción y planeación. Cuando se elabora un mal pronóstico, la planeación se viene abajo y todas las áreas del Hospital se vuelven ineficientes. Esto se puede observar directamente en el bajo desempeño y los servicios insuficientes que se brindan a los pacientes. Consultas negadas, excesos de inventarios de productos que no requieren los pacientes, reducción de margen al vender con descuentos para lograr los objetivos, costos más altos en las compras, producción y/o distribución para reaccionar a emergencias, etc., ... estos son los síntomas.

Pronosticar la demanda con buena exactitud normalmente no es fácil. No existen recetas de cómo hacerlo y cada empresa tiene que determinar la mejor forma de elaborar sus pronósticos.

El tema de pronosticar es extenso y requiere de técnicas *ad hoc* para cada situación. Por ejemplo, pronosticar consultas de servicios de alta rotación requiere diferentes técnicas que pronosticar servicios de áreas de bajo movimiento o de demanda intermitente. Pronosticar la demanda de servicios nuevos requiere consideraciones diferentes. Por otro lado, en ciertas ocasiones es conveniente pronosticar agrupando áreas similares y en ciertas ocasiones por especialidad o por año.

En ciertas ocasiones el uso de herramientas estadísticas es de muy buena ayuda y en otras ocasiones es mejor elaborar pronósticos en colaboración con los pacientes.

Si el éxito de la planeación depende de pronósticos certeros, entonces es conveniente revisar si se elaboran pronósticos en el HU y determinar si es posible mejorar la exactitud.

Un buen comienzo para mejorar la exactitud de los pronósticos es entender los factores que influyen en el comportamiento de la demanda y tener mejor idea de qué ofrecen las diferentes técnicas de pronósticos [45].

¿Qué son los Pronósticos?

El pronóstico no es una predicción de lo que irremediablemente pasará en el futuro. Un pronóstico es información con cierto grado de probabilidad de lo que pudiera pasar.

La probabilidad de éxito del plan estadístico está en función directa de la elaboración de los pronósticos. Dicho de otra forma, el resultado de la planeación y operación del HU está directamente ligado a la certeza de los pronósticos.

Para pronósticos de negocios las mejores prácticas sugieren una combinación de técnicas cuantitativas y cualitativas, pero en nuestro caso, únicamente haremos uso de las técnicas cuantitativas para determinar el número de consultas probables de un determinado servicio en algún periodo de tiempo. Apoyados de información grafica para extender aun mas el panorama futuro que resulte del escenario pronosticado.

Se ha comprobado que las técnicas de pronósticos estadísticas son muy útiles, ya que cuantifican de manera muy exacta ciertos componentes de la demanda como tendencia, patrones de estacionalidad o de eventos.

Pronósticos y Planeación: Procesos críticos del negocio

El papel de los directivos y gerentes es administrar los elementos del negocio que conducen al logro de los objetivos. De una u otra manera los directivos “presienten” lo que pasará. Sin embargo, en la mayoría de las ocasiones, sus decisiones son mucho mejores si se apoyan en cifras cuantificadas por una herramienta estadística ya que de esta manera se parte de una cifra base más conservadora, y mas aun cuando esta herramienta se tiene al alcance de la mano en todo momento y en todo lugar. Por otro lado, cada vez es más necesario diferenciar las demandas de los pacientes de un mismo servicio, lo que requiere más tiempo y argumentos.

¿Cuál es el costo de malos Pronósticos?

Tenemos garantía que los pronósticos no van a ser 100% exactos y que además la desviación de los pronósticos tiene un costo implícito, ya sea que los pronósticos fueron altos o fueron bajos respecto a la realidad.

El punto fundamental en los pronósticos es ser consistente y lograr la menor desviación respecto a los objetivos:

- i) Pronosticar por arriba de la demanda tiene entre sus consecuencias exceso de inventario, obsolescencia, reducción de margen para promover su venta.
- ii) Pronosticar por debajo de la demanda tiene entre sus consecuencias comprar y producir más caro algo que no estaba planeado, incluso pérdida de pacientes y margen si no reaccionamos a tiempo.

La elaboración de los pronósticos requiere información de la planeación.

Quien elabora los pronósticos debe considerar múltiples factores que propician índices de enfermedad, como lo son: estación del año, clima, construcción de más Hospitales, campañas de vacunación, brotes de epidemia o, incluso, si hubo algún evento extraordinario en la historia reciente que pueda desviar fuertemente las estimaciones. Dejar esto a la memoria seguramente causará que nuestros pronósticos sean menos exactos. Además es necesario implantar alguna forma de documentar la historia para medir los impactos de los eventos y considerarlos o no como parte del pronóstico si se realizaran nuevamente.

La conclusión de un estudio llevado a cabo el pasado año por el Institute of Business Forecasting (IBF) titulado “Why Forecasting?” (www.ibf.org) sugiere que “hoy en día es ineludible un proceso más formal de elaborar los pronósticos sin importar en qué tipo de negocio y/o industria se localice la empresa o qué función realiza. Siempre hay una necesidad de estimar el futuro sobre la cual construir un plan”.

Este estudio también establece que diversas áreas del HU establecen planes diferentes a partir de los pronósticos [45]:

- El área de servicios requiere pronósticos para determinar qué nuevos productos o servicios introducir o discontinuar; en qué sectores de la sociedad tener presencia o salir; qué servicios promover, etc.
- El área de ventas requiere pronósticos para establecer cuotas u objetivos de venta.
- El área de cadena de suministro requiere pronósticos para planear la compra, el abastecimiento y planes logísticos.
- El área de finanzas requiere de pronósticos para hacer un mejor plan de ingresos y gastos.

Finalmente, el estudio concluye que aunque es difícil cuantificar los beneficios del pronóstico de consultas, en la cadena de suministro hay ciertos componentes que pueden ser cuantificados, mostrando las ventajas de éste. Una forma para medir estos beneficios es considerar cuál hubiera sido la pérdida si el pronóstico no fuera exacto. Otra manera de medirlos es cuestionando cuánto habría sido la ganancia (o ahorro) con pronósticos mejorados.

¿Cómo Pronosticar?

El HU actualmente está recurriendo al uso de paquetes de pronósticos estadísticos y establecer un proceso más formal en la planeación de servicios y operación.

Antes de continuar la tesis de pronósticos estadísticos es conveniente entender aspectos relativos al proceso de los pronósticos:

-
- A) Cómo funcionan las técnicas estadísticas.
 - B) Cuántos datos se requieren.
 - C) Cómo se puede medir el impacto de la desviación de los pronósticos.
 - D) Cómo pronosticar cientos de consultas de manera rápida y más exacta.
 - E)Cuál es el perfil sugerido de quien elabora los pronósticos, etc.

Esto le permitirá evaluar si tiene oportunidad de mejorar su proceso mediante el uso de la herramienta que se quiere desarrollar, y si esta a su vez logra convertirse en un apoyo fundamental en la toma de decisiones por su funcionalidad y además por su fácil manejo. Hacer buenos pronósticos de demanda es un proceso que agrega valor ya que está íntimamente relacionado con la toma de decisiones que impactaran en el rendimiento del HU.

Exactitud del pronóstico como indicador de desempeño clave

Se requiere madurez para establecer la exactitud de los pronósticos como un indicador clave ya que siempre habrá desviaciones entre el pronóstico y la demanda. Además es necesaria la intervención de expertos que estén íntimamente ligados con la historia del HU, pues sus aportes serán de gran utilidad para darle mayor sentido a los datos que se tienen y sobre todo, a la hora de analizar los resultados arrojados por la herramienta. Es necesario documentar y aprender cuales fueron las razones que nos llevaron a tanta desviación en una estimación, y si esto no esta disponible, se tendrá la experiencia de los expertos para ayudar a comprender los factores que nos llevaron a dichos resultados, como lo pueden ser: fusión de dos o mas áreas, desaparición de un área, migración de áreas hacia otro Hospital, brotes epidemiológicos, campañas de vacunación, todo esto será de gran apoyo para analizar las variaciones de datos a lo largo de los años. Solo mediante la medición obtenemos una referencia que nos pueda indicar nuestro desempeño y/o tomar acciones inmediatas para corregir el rumbo [46], [47].

Mejores prácticas en la elaboración de Pronósticos

Las mejores prácticas sugieren una combinación de pronósticos estadísticos con pronósticos por experiencia. Esta práctica ayuda a reducir los efectos de influencia del plan, influencias emocionales y además a determinar una mejor estimación y no solo un simple promedio. Una mejora en la exactitud de los pronósticos la podrá confirmar cuando cada mes se estén logrando los resultados de los objetivos. Esto también se confirma cuando las diferentes áreas están alineadas a partir de un pronóstico consensuado.

Apoyando a esta herramienta se tendrá todo un historial clínico, el cual se documenta año tras año, además de los aportes que nos brinden los encargados de cada especialidad para tomar en cuenta todos los factores disponibles y causantes de alguna variación en el numero de consultas realizadas en el HU a través de los años.

Plan de Servicios y Operación. (PSO o S&OP)

De nada sirve un buen pronóstico si todas y cada una de las áreas en que se divide el HU no están alineadas para entregar de manera oportuna los servicios que se requieren. Para evitar esto se debe establecer un proceso formal de Planeación de

Servicios y Operación de la empresa llamado S&OP (Services and Operations Planning) o también conocida como junta PSO (Planeación de Servicios y Operación).

De manera simple, el proceso S&OP consiste en una reunión mensual que a partir de los pronósticos e información de las áreas cercanas a la demanda se definirán los objetivos y las actividades de forma consensuada. La junta se lleva a cabo después de haber realizado con diferentes responsables juntas previas de servicios nuevos, servicios activos y aspectos relativos sobre el suministro.

La creación de una herramienta de elaboración de pronósticos es la clave para la planeación y operación del HU. La toma de decisiones y las utilidades de dicho Hospital pueden ser mejores si cuenta con un buen pronóstico.

5.2 Regresión lineal [24]

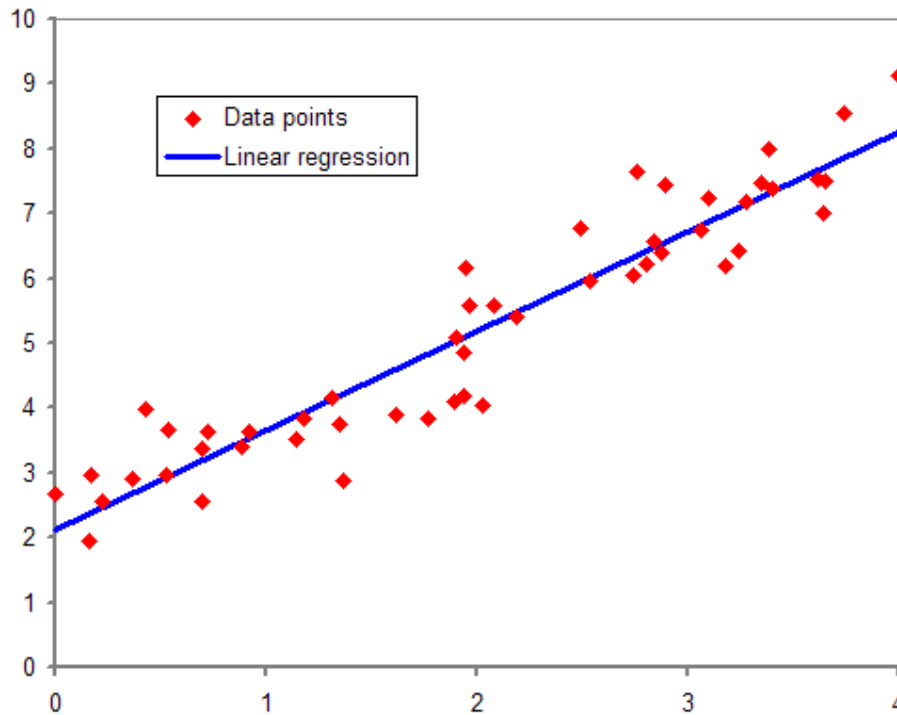


Figura 5.1 Ejemplo de una regresión lineal con una variable dependiente y una variable independiente.

En estadística la **regresión lineal** o **ajuste lineal** es un método matemático que modeliza la relación entre una variable dependiente Y , las variables independientes X_i y un término aleatorio ε . Este modelo puede ser expresado como:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon \quad (1)$$

donde β_0 es la intersección o término "constante", las β_i son los parámetros respectivos a cada variable independiente, y p es el número de parámetros independientes a tener en cuenta en la regresión. La regresión lineal puede ser contrastada con la regresión no lineal.

Historia

La primer forma de regresiones lineales documentada fue el método de los mínimos cuadrados, el cual fue publicado por Legendre en 1805, y por Gauss en 1809. El término "mínimos cuadrados" proviene de la descripción dada por Legendre "*moindres carrés*". Sin embargo Gauss aseguró que conocía dicho método desde 1795.

Tanto Legendre como Gauss aplicaron el método para determinar, a partir de observaciones astronómicas, las órbitas de cuerpos alrededor del sol. En 1821, Gauss publicó un trabajo en dónde desarrollaba de manera más profunda el método de los mínimos cuadrados, y en dónde se incluía una versión del teorema de Gauss-Márkov.

Etimología

El término *regresión* se utilizó por primera vez en el estudio de variables antropométricas: al comparar la estatura de padres e hijos, resultó que los hijos cuyos padres tenían una estatura muy superior al valor medio tendían a igualarse a éste, mientras que aquellos cuyos padres eran muy bajos tendían a reducir su diferencia respecto a la estatura media; es decir, "regresaban" al promedio. La constatación empírica de esta propiedad se vio reforzada más tarde con la justificación teórica de ese fenómeno.

El término *lineal* se emplea para distinguirlo del resto de técnicas de regresión, que emplean modelos basados en cualquier clase de función matemática. Los modelos lineales son una explicación simplificada de la realidad, mucho más ágil y con un soporte teórico por parte de la matemática y la estadística mucho más extenso.

Supuestos del modelo de regresión lineal

Para poder crear un modelo de regresión lineal, es necesario que se cumpla con los siguientes supuestos [48], [49]:

1. La relación entre las variables es lineal.
2. Los errores son independientes.
3. Los errores tienen varianza constante.
4. Los errores tienen una esperanza matemática igual a cero.
5. El error total es la suma de todos los errores.

Tipos de modelos de regresión lineal

Existen diferentes tipos de regresión lineal que se clasifican de acuerdo a sus parámetros:

Regresión lineal simple

Sólo se maneja una variable independiente, por lo que sólo cuenta con dos parámetros. Son de la forma:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \quad (2)$$

donde ε_i es el error asociado a la medición del valor X_i y siguen los supuestos de modo que $\varepsilon_i \sim N(0, \sigma^2)$ (media cero, varianza constante e igual a un σ y $\varepsilon_i \perp \varepsilon_j$ con $i \neq j$).

$$\hat{\beta}_1 = \frac{\sum x \sum y - n \sum xy}{(\sum x)^2 - n \sum x^2} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2} \quad (3)$$

$$\hat{\beta}_0 = \frac{\sum y - b_1 \sum x}{n} = \bar{y} - \hat{\beta}_1 \bar{x} \quad (4)$$

Regresión lineal múltiple

Maneja varias variables independientes. Cuenta con varios parámetros. Se expresan de la forma:

$$Y_i = \beta_0 + \sum \beta_i X_{ip} + \varepsilon_i \quad (5)$$

donde ε_i es el error asociado a la medición i del valor X_{ip} y siguen los supuestos de modo que $\varepsilon_i \sim N(0, \sigma^2)$ (media cero, varianza constante e igual a un σ y $\varepsilon_i \perp \varepsilon_j$ con $i \neq j$).

Rectas de regresión

Las rectas de regresión son las rectas que mejor se ajustan a la nube de puntos (o también llamado diagrama de dispersión) generada por una distribución binomial. Matemáticamente, son posibles dos rectas de máximo ajuste:

- La recta de regresión de Y sobre X :

$$y = \bar{y} + \frac{\sigma_{xy}}{\sigma_x^2} (x - \bar{x}) \quad (6)$$

- La recta de regresión de X sobre Y :

$$x = \bar{x} + \frac{\sigma_{xy}}{\sigma_y^2} (y - \bar{y}) \quad (7)$$

La correlación (" r ") de las rectas determinará la calidad del ajuste. Si r es cercano o igual a 1, el ajuste será bueno; si r es cercano o igual a 0, se tratará de un ajuste malo. Ambas rectas de regresión se intersectan en un punto llamado centro de gravedad de la distribución.

Aplicaciones:

Líneas de tendencia

Una *línea de tendencia* representa una tendencia en una serie de datos obtenidos a través de un largo periodo de tiempo. Este tipo de líneas puede decirnos si un conjunto de datos en particular (como por ejemplo, el PBI, el precio del petróleo o el valor de las acciones) han aumentado o decrementado en un determinado periodo de tiempo. Se puede dibujar una línea de tendencia a simple vista fácilmente a partir de un grupo de puntos, pero su posición y pendiente se calcula de manera más precisa utilizando técnicas estadísticas como las regresiones lineales. Las líneas de tendencia son generalmente líneas rectas, aunque algunas variaciones utilizan polinomios de mayor grado dependiendo de la curvatura deseada en la línea.

Medicina

En medicina, las primeras evidencias relacionando la mortalidad con el fumar tabaco vinieron de estudios que utilizaban la regresión lineal. Los investigadores incluyen una gran cantidad de variables en su análisis de regresión en un esfuerzo por eliminar factores que pudieran producir correlaciones espurias. En el caso del tabaquismo, los investigadores incluyeron el estado socio-económico para asegurarse que los efectos de mortalidad por tabaquismo no sean un efecto de su educación o posición económica. No obstante, es imposible incluir todas las variables posibles en un estudio de regresión. En el ejemplo del tabaquismo, un hipotético gen podría aumentar la mortalidad y aumentar la propensión a adquirir enfermedades relacionadas con el consumo de tabaco. Por esta razón, en la actualidad las pruebas controladas aleatorias son consideradas mucho más confiables que los análisis de regresión.

5.3 Mínimos Cuadrados [25]

Aplicaciones del método:

Este método es una aproximación que nos permite representar un grupo mediante un valor. Así que donde haya un conjunto de valores registrados, no importa la cantidad de estos, ni su tamaño, ahí estará el método de mínimos cuadrados, proporcionándonos una tendencia. Las aplicaciones del método son ilimitadas señores, el límite esta en su imaginación. Desde conocer la tendencia de su éxito con las mujeres, hasta modelar la producción y ventas de una gigantesca y exitosa empresa petrolera. Para la ingeniería, los negocios, los hombres solteros, la investigación y todas las ciencias en general, el método de los mínimos cuadrados, le garantiza su tendencia con el mínimo margen de error.

Qué es el método de mínimos cuadrados?

Es el método más usado para el ajuste de una recta a una serie de datos.

Supongamos el siguiente ejemplo:

Un investigador captura 5 valores de temperatura en la Ciudad de México a lo largo de un día, los valores son los siguientes:(Figura 5.2)

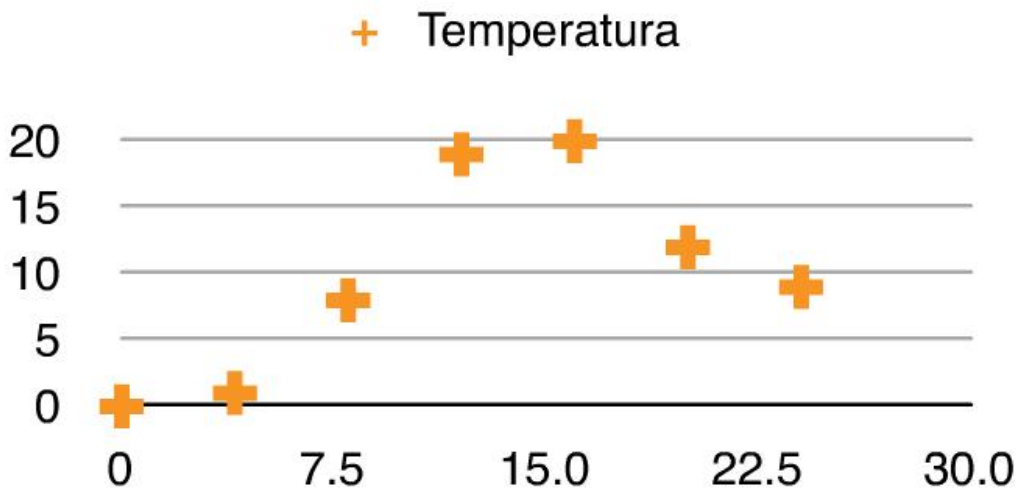


Figura 5.2 Tabla de valores.

Este investigador requiere expresar estos datos de una manera más conveniente y apegada a sus datos reales. Y es aquí donde podemos recurrir a los mínimos cuadrados. Con el método de los mínimos cuadrados se puede calcular en una función una serie de datos registrados.

Cómo aplicar el método de los mínimos cuadrados:

Tomando en cuenta que el método de los mínimos cuadrados consiste en ajustar una recta a valores dispersos, necesitamos entonces conocer las características de la recta, como son, su pendiente y su ordenada al origen, de la cual necesitamos estimar los valores de a y de b de la siguiente ecuación:

$$Y = a + bx$$

Por lo que, sabiendo que el método de los mínimos cuadrados calculará la recta que pasa por la media de todas las observaciones representadas por (x_1, y_1) , (x_2, y_2) , ... (x_n, y_n) , entonces la ecuación de la recta será

$$Y = \bar{y} + b(x - \bar{x})$$

En donde:

\bar{y} = media de y_1, y_2, \dots, y_n

\bar{x} = media de x_1, x_2, \dots, x_n

$$b = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2} \quad (8)$$

Tendremos así entonces la ecuación de la recta en su forma pendiente-ordenada que corresponderá a la recta que satisface la condición:

$$\sum(x - \bar{x})(y - \bar{y}) = (x_1 - \bar{x})(y_1 - \bar{y}) + (x_2 - \bar{x})(y_2 - \bar{y}) + \dots + (x_n - \bar{x})(y_n - \bar{y}) \quad (9)$$

de que las constantes “a” y b” hacen mínima la suma

$$\sum(y - \bar{Y})^2 = \sum(y - \bar{y})^2 - (Y - \bar{y}) \quad (10)$$

de los cuadrados de las desviaciones de los valores observados respecto a dicha línea.

La formula despejada para nuestro caso se muestra en **(11)** [26]:

$$b_1 = \frac{\sum XY - \bar{y} \sum X}{\sum X^2 - \bar{x} \sum X}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

(11)

Pruebas

Se realizaron 10 pruebas, resultando estas satisfactorias, de modo que describiremos aquellos con los que nos encontramos a lo largo del desarrollo del proyecto, los cuales se describen a continuación:

javax.microedition.io.ConnectionNotFoundException:TCP Open:

Esta excepción es arrojada cuando el Servidor (WAP) no está en funcionamiento.

java.net.SocketException: java.net.ConnectException: Connection refused: connect:

Esta excepción es arrojada cuando nuestro servicio de Base de Datos no esta iniciado.

java.net.ConnectException:

Esta excepción es arrojada cuando el teléfono móvil no tiene saldo disponible.

Conclusiones

Si se desea ser un líder eficiente, se necesita ser capaz de realizar decisiones correctas. Si se quiere aprender a hacer esto oportunamente y de manera idónea, entonces debe llevar a su equipo a un suceso espectacular y oportuno. Sin embargo, si se realizan decisiones erróneas, su equipo tomará muchos riesgos y su permanencia como líder probablemente será breve. Por consiguiente en el presente proyecto de tesis hemos realizado la implementación de un novedoso sistema de pronósticos para la toma de decisiones basado en tecnologías móviles (teléfono móvil). Además, hemos incorporado un sistema de pronósticos basado en regresión lineal el cual puede ser utilizado en cualquier momento y desde cualquier lugar.

Por otro lado nuestro desarrollo consta de dos partes: La primera, es una aplicación WAP (Wireless Application Protocol), que funciona como nuestro proveedor de servicios, con la cual se alcanzaron dos de los objetivos generales. La segunda parte corresponde a nuestra aplicación móvil, que se estableció en el primer objetivo general, que funciona en teléfonos móviles Nokia series 60, y LG KP215, Motorola y en general en teléfonos móviles con Java Básico similar al emulador integrado en NetBeans (Wireless ToolKit). Segundo, la parte que corresponde a nuestro proveedor de servicios Web fue desarrollada bajo la configuración de un servidor de Aplicaciones Web (Tomcat version 4.1) propio de NetBeans. Dentro de este implementamos un Servlet el cual es el encargado de la comunicación entre la Base de Datos de nuestro sistema y los usuarios móviles, esto a través, de la generación de un hilo de control para cada uno de nuestros clientes móviles.

Además, la implementación del proveedor de servicios móviles se basa en el sistema de comunicación denominado GPRS. Este sistema (GPRS) se caracteriza por brindar un servicio basado en la cantidad (KB) de información que el usuario móvil descarga desde una aplicación WAP, y no en el tiempo de conexión con esta. Esto hace que el costo por servicio se reduzca drásticamente, teniendo un costo aproximado de 4 centavos por servicio solicitado a nuestro sistema. Este costo obedece a que nuestra aplicación WAP envía información ya procesada la cual el teléfono móvil solo la codifica a una presentación gráfica. Y como es bien sabido una gráfica dice mas que mil palabras.

Por otra parte, la aplicación móvil desarrollada en Wireless ToolKit brinda al usuario una interfaz sencilla, amigable y concreta, lo que facilita el manejo y solicitud de cualquier servicio de pronóstico. Las modalidades en que se implementaron los servicios de pronósticos son los siguientes: a) Mes y especialidad; b) Mes y todas las especialidades; c) año y especialidad; d) año y todas las especialidades; e) especialidad y rango de años; f) todas las especialidades y rango de años.

Finalmente, este proyecto de tesis ha motivado y abierto un nicho de oportunidades en el desarrollo de servicios a usuarios móviles. Particularmente, el HU nos ha solicitado la generación de nuevos servicios y nuevos métodos de pronóstico que incrementen la eficiencia en la toma de decisiones, y por ende, permitan una planeación estratégica acorde a los nuevos tiempos que el HU está enfrentando. Más aun, con este proyecto de tesis hemos logrado la incorporación del uso de tecnologías de información y comunicación móvil a uno de los sectores que más servicios está demandando

actualmente, por lo que se le considera un sector clave en el desarrollo de nuestra sociedad.

Los objetivos específicos que se plantearon al inicio de este proyecto de tesis, se cumplieron al pie de la letra con la aplicación desarrollada, ya que para poder tener un buen trabajo se necesitó analizar, extraer, e introducir la información proporcionada por el HU, además de analizar los posibles protocolos de comunicación entre aplicaciones móviles e Internet, se analizaron detenidamente las herramientas para su desarrollo y se aplicaron para poder darle forma y funcionamiento a nuestro proyecto al fin terminado.

La herramienta que se desarrollo en este proyecto de tesis, se apego estrictamente a las especificaciones proporcionadas por el HU, así que a continuación se resumen algunas extensiones posibles para tal proyecto.

La aplicación desarrollada para el teléfono móvil, el midlet, abarca 11 Kb. Podríamos multiplicar un poco ese espacio y los teléfonos actuales aún soportarían el software, es decir, realizando una serie de extensiones y apegándose a la base que se tiene, la aplicación estaría dentro del límite de espacio para su correcta instalación en el dispositivo móvil.

Una de las posibles extensiones, es el poder realizar el ABC (Altas, Bajas, Consultas) de la Base de Datos del sistema directamente desde nuestro teléfono móvil. Para así tener una comunicación con nuestro sistema, sin la necesidad de estar presentes dentro de la institución.

Respecto a la información grafica, podemos extender nuestra herramienta con opciones de graficado, para tener un panorama más amplio del que se tiene con la única grafica que se presenta. Respecto a este punto, se podría implementar además el almacenamiento persistente en el teléfono móvil (RMS) de los datos de consulta, para comparar resultados consultados en diferentes ocasiones y desplegarlos en tablas comparativas.

Un punto muy importante a considerar y aclarar, es el relacionado con el tema de Agentes, el cual se pretendía abordar al principio del proyecto de tesis, pero debido a los requerimientos que se nos presentaron en el HU, se dejo para trabajo a futuro, ya que la intención principal que desarrollamos fue la de implementar tecnologías móviles con un método de pronostico ya hecho para la toma de desiciones, y no tratar de mejorarlo o implementar uno nuevo. Aquí se tiene un buen elemento a extender a la tesis ya que es un tema muy interesante con el cual se podrían implementar Agentes de Aprendizaje para mejorar los pronósticos arrojados.

Referencias bibliográficas

- [1] To BI or not to BI <http://todobi.blogspot.com/2005/12/to-bi-or-not-to-bi.html>, Lic. Ricardo Sánchez Montoya
Profesor de Cátedra del Departamento Académico de Mercadotecnia del ITESM Campus Monterrey rsm@itesm.mx
- [2] http://www.esemanal.com.mx/articulos.php?id_sec=2&id_art=6328,
Lunes 14 de abril de 2008. Núm. Inteligencia de negocios móvil.
- [3] Wikipedia. <http://en.wikipedia.org/wiki/Gartner>, **Gartner**, Inc. (NYSE: IT) is an information technology research and advisory firm headquartered in Stamford, Connecticut. It was known as **The Gartner Group** until 2001.
- [4] Wikipedia. http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos, Almacén de Datos
- [5] <http://www.monografias.com/trabajos12/basdat/basdat.shtml>, Sistemas de información para la toma de decisiones autor Daniel Cohen McGraw Hill
- [6] <http://www3.uji.es/~mmarques/f47/apun/node44.html>, María Mercedes Marqués Andrés 2001-02-12
- [7] http://www.evidalia.es/trucos/index_v2-279-29.html, Normalización de Bases de Datos.
- [8] MySQL. <http://es.wikipedia.org/wiki/MySQL>
- [9] O'Reilly. http://es.wikipedia.org/wiki/O'Reilly_&_Associates, O'Reilly & Associates, ahora O'Reilly Media, es una empresa editorial estadounidense principalmente enfocada a libros relacionados con la programación informática.
- [10] Wikipedia. <http://en.wikipedia.org/wiki/MSQL>, mSQL or Mini SQL is a lightweight client/server database from Hughes Technologies. Originally developed in 1994, mSQL filled a vacuum that existed

-
- between the embedded desktop databases like Microsoft Access and the enterprise-level commercial databases such as Oracle and DB2. Between 1994 and 1997, it grew in popularity and became the primary database of choice for open source developers.
- [11] Wikipedia. <http://www.mediawiki.org/wiki/MediaWiki/es>. MediaWiki es un software wiki libre escrito originalmente para Wikipedia. Ahora es utilizado por otros proyectos wikis de la Fundación Wikimedia y por otras wikis, incluyendo este sitio web, el hogar de MediaWiki.
- [12] Wikipedia. <http://es.wikipedia.org/wiki/Drupal>. Drupal (pronunciación IPA: [druˈɫ]) es un sistema de gestión de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.
- [13] Wikipedia. <http://es.wikipedia.org/wiki/PhpBB>. phpBB es un sistema de foros gratuito basado en un conjunto de paquetes de código programados en el popular lenguaje de programación web PHP y lanzado bajo la Licencia pública general de GNU, cuya intención es la de proporcionar fácilmente, y con amplia posibilidad de personalización, una herramienta para crear comunidades. Su nombre es por la abreviación de *PHP Bulletin Board*.
- [14] Wikipedia. <http://es.wikipedia.org/wiki/Bugzilla>. Bugzilla es una herramienta basada en Web de seguimiento de errores (bugs), (Bug Tracking System o BTS por sus siglas en inglés), originalmente desarrollada y usada por el proyecto Mozilla. Lanzado como software de código abierto por Netscape Communications en 1998, Bugzilla ha sido adoptado por una variedad de organizaciones para usarlo para

-
- seguimiento de defectos (errores) tanto para software libre como para software propietario.
- [15] Wikipedia. <http://es.wikipedia.org/wiki/.php>. PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente PHP Tools, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.
- [16] <http://es.wikipedia.org/wiki/MyISAM>. MyISAM es la tecnología de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL. Este tipo de tablas están basadas en el formato ISAM pero con nuevas extensiones.
- [17] http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java
- [18] <http://www.desarrolloweb.com/articulos/1670.php>
- [19] <http://www.webyog.com/en/>
- [20] Este documento intitulado “Telefonia movil- Introducción” de Kioskea (es.kioskea.net) esta dispuesto a disposición bajo la licencia Creative Commons. Puede copiar, modificar bajo las condiciones puestas por la licencia, siempre que esta nota sea visible.
- [21] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_d_g/capitulo2.pdf. Capitulo 2.2.3.2
- [22] <http://www.lcc.uma.es/pfc/370.pdf>. Capitulo 3.1
- [23] Las mejores prácticas en pronósticos. Armando González - agonzalez@forecastpro.com
- [24] http://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal

-
- [25] <http://www.scribd.com/doc/2452626/Minimos-Cuadrados>. Arturo Padilla Razo Héctor Zárata Rea
- [26] <http://www.eumed.net/cursecon/medir/estima.htm>
- [27] Date, Introd.a Los Sistemas De Bases De Datos. ISBN: 9684444195, Edit. Pearson, 2000
- [28] Exposing a MySQL Database with RESTful Web Services <http://www.netbeans.org/kb/60/websvc/rest-mysql>
- [29] Cabrera Sánchez. Sistemas gestores de bases de datos. Edit. Paraninfo, 2001.
- [30] Silverschatz Abraha. Principios de bases de datos 5 ed., McGraw Hill, 2007
- [31] Galvez R.S. Ortega D.L. Java a tope: J2ME (Java 2 Micro Edition) Edición electrónica, ISBN: 84-688-4704-6. Universidad de Málaga.
- [32] Chopra Vivek, Profesional apache tomcat 6. ISBN: 9788441523777, Alianza editorial, 2008.
- [33] Deitel y Deitel. 5th edición, Como programar en Java. Prentice Hall 2008.
- [34] Deitel y Deitel. 5th edición, Como programar en C++. Prentice Hall 2006.
- [35] Barrera, Corroppoli, Dans, Ibañez. Introducción al UML 2003
- [36] Joaquin Gracia. UML: Diagramas UML. ¿Qué es UML?, 2005. <http://www.ingenierosoftware.com/analisisydiseno/uml.php>
- [37] Patricio Letelier Torres. Curso: "Desarrollo de Software Orientado a Objetos usando **UML**", DSIC and UPV, 2005.
- [38] Booch, G. et al. **The Unified Modeling Language**. Reading: Addison-Wesley, 1999
- [39] Larman C. *UML y Patrones*, Prentice Hall. Mexico 1999

-
- [40] Bosch J. *Design and Use of Software Architecture*, Addison Wesley, Harlow, England, 2000.
- [41] Christoffer Andersson. *GPRS and 3G Wireless Applications: Professional Developer's Guide*, Wiley, 2001.
- [42] Alan Kavanagh, John Beckmeyer, *GPRS Networks*. Osborne Publishing, 2002.
- [43] Mallick, M., 2003. *Mobile and Wireless Design Essentials*. Wiley Publishing, Inc.,
- [44] Adam Myatt. *Pro NetBeans™ IDE 5.5 Enterprise Edition*, Edior Apress. ISBN: 978-1-59059-788-0, 2007.
- [45] National Public Health Partnership.: *A planning framework for public health practice*. (2000)
- [46] F. Zacarias, R. Cuapa, F. Lozano, A. Vazquez, D. Zacarias, u-Teacher: *Ubiquitous Learning Approach*, pp. 9-20, Edutainment, LNCS 5093, China (2008).
- [47] National Center for Biotechnology Information, <http://www.ncbi.nlm.nih.gov>
- [48] Robert Harris. *Decision Making Techniques*. <http://www.virtualsalt.com/crebook6.htm>. Virtual Salt (1998)
- [49] Nusser, S., L. Miller, K. Clarke, and M. Goodchild, 2003. *Geospatial IT for Mobile Field Data Collection*. *Communications of the ACM*, 46(1), pp. 63-64.