



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE GESTIÓN REMOTA PARA LA
DIFUSIÓN DE MENSAJES VÍA INALÁMBRICA**

**ALEJANDRO LEONEL AMAYA AGUILAR
ANGEL CAMACHO VILLÁN**

DIRECTOR:

ING. AMAURY HECTOR PEREA MATSUMURA



CIUDAD UNIVERSITARIA, MÉXICO D.F.
2010.

AGRADECIMIENTOS

A Dios por darme la oportunidad de desarrollar todas las metas que me trazo y por permitirme mejorar cada día.

A mis padres Alejandro Amaya y Ramona Aguilar por su apoyo éste y todos los días, las personas más importantes para mí, sólo con su amor y confianza he terminado una de las mejores etapas de mi vida.

A la Universidad Nacional y a la Facultad de Ingeniería que además de proporcionarme conocimiento me dio educación y una identidad.

A mis excelentes profesores de la Facultad: Ángel Victoria, Luis Hernández, Luis Arenas, Patricia, Lanzier, Cintia y Betzabé, profesores que imparten algo más que clase, son ingenieros e ingenierías que creen en la Universidad y en sus alumnos.

A mis amigos de la Universidad: Sergio, Fernando, Jorge, Ángel, José, Héctor, Ángeles todos ellos dejando algo importante en esta etapa de mi vida.

Al equipo nGWISE por todo su apoyo, todos verdaderos emprendedores, un ejemplo de capacidad y perseverancia: Roberto, Amaury, Ulises, Ángel, Alex y Ann.

A los sinodales por sus valiosas observaciones y consejos para el desarrollo de este trabajo.

Al Dr. Víctor García, Jefe del departamento de Telecomunicaciones por el apoyo que brinda el laboratorio LADAI para poder desarrollarnos como grupo estudiantil.

A todo el equipo del LADAI, gente con gran capacidad y que pone en alto el nombre del laboratorio.

A la gente que compartió su conocimiento conmigo: el profesor Víctor que me inculcó la idea de estudiar esta carrera y apoyarme en los momentos difíciles de mi desarrollo como estudiante. A Don Jorge por apoyarme como todo un profesor durante la carrera.

A Daniel, Lucia, Ángel Valdelamar y Judith, gracias por estar siempre para mí.

- Alejandro L. Amaya A.

Índice	Página
<i>Capítulo I. Introducción</i>	1
1.1 Descripción General	
1.2 Requerimientos Específicos	
1.3 Objetivo	
1.4 Alcances	
1.5. Método	2
 <i>Capítulo II. Conceptos básicos del entorno móvil</i>	 4
2.1 Arquitectura de los sistemas de comunicaciones inalámbricas	
2.2 Clasificación de los dispositivos móviles	6
2.3 Servicios móviles	7
2.4 Arquitecturas para aplicaciones móviles.	10
2.4.1 Internet inalámbrico (Thin Clients)	
2.4.2 Clientes inteligentes (Smart Clients)	11
2.4.3 Mensajería	13
2.5 Tecnologías inalámbricas aplicables al sistema.	15
2.5.1 Wireless Personal Area Network (Red Inalámbrica de Área Personal)	
2.5.1.1 Bluetooth	15
2.5.2 Wireless Local Area Network (Red Inalámbrica de Área Local)	22
2.5.2.1 WiFi-802.11b	23
2.5.2.2 802.11g	
2.5.2.3 802.11n	
2.5.3 WIRELESS WIDE AREA NETWORK (Red Inalámbrica de Área Amplia)	25
2.5.3.1 Redes de Primera-Generación 1G	
2.5.3.2 Redes de Segunda-Generación 2G	26
2.5.3.3 Redes de Tercera-Generación 3G	28
 <i>Capítulo III. Fundamentos para el desarrollo de aplicaciones en internet</i>	 30
3.1 Arquitectura Cliente – Servidor	31
3.1.1 Antecedentes	
3.1.2 Arquitectura	
3.2 Desarrollo de sitios Web	34
3.2.1 Introducción	
3.2.2 Requerimientos	
3.2.3 Diseño	35
3.2.4 Pruebas y resultados	37
3.2.5 Mantenimiento	
3.3 Bases de datos	39
3.3.1 Introducción	
3.3.2 Sistema manejador de bases de datos	
3.3.3 Análisis de bases de datos	43
3.3.4 Diseño de bases de datos	50
3.3.5 Construcción	
3.4 Control de versiones	53
3.5 Buenas prácticas para el desarrollo de software	55

<i>Capítulo IV. Análisis</i>	60
4.1 Usuarios	61
4.2 Seguridad	62
4.3 Información	
4.4 Sistema automatizado	64
4.5 Requerimientos de Hardware	65
4.6 Requerimientos de Software	69
<i>Capítulo V. Diseño</i>	85
5.1 Estructura general del sistema	86
5.1.1 Diseño del mapa del sitio	
5.1.2 Diseño de la administración	
5.2 Modelo de casos de uso	96
5.3 Bases de datos	110
<i>Capítulo VI. Desarrollo</i>	113
6.1 Configuración del Servidor	114
6.2 Framework	117
6.2.1 Programación Orientada a Objetos	
6.2.2 Aplicando la arquitectura MVC	
6.2.3 Construcción de la Base de Datos	119
6.2.4 Construcción del Modelo	121
6.2.5 Programación de la Vista	123
6.2.6 Programación del Controlador	124
<i>Capítulo VII Pruebas</i>	125
7.1 Plan de Pruebas	126
7.1.1 Pruebas de Integración	127
7.1.2 Pruebas del Sistema	129
7.1.3 Pruebas del Sistema (Sistema abierto)	134
<i>Capítulo VIII. Resultados y aportaciones</i>	137
<i>Capítulo IX. Conclusiones</i>	143
<i>Anexo A. Manual técnico</i>	146
A.1 Introducción	147
A.2 Objetivo del sistema	
A.3 Contenido técnico	
A.3.1 Reglas del negocio	
A.3.2 Diagramas de flujo de datos	148
A.3.3 Campos requeridos	151
A.3.4 Requerimientos de interface con otros sistemas	
A.3.5 Modelo lógico de datos	153
A.3.6 Modelo físico de datos	154
A.3.7 Áreas de aplicación	155
A.4 Mapa de navegación	
A.4.1 Descripción gráfica	156
A.4.2 Descripción de procesos	163

<i>Glosario de términos</i>	169
<i>Bibliografía</i>	172

Capítulo I

Introducción

1.1 Descripción General

En la actualidad se ha ampliado el uso de dispositivos móviles y forman parte de casi cualquier actividad, es una nueva manera de comunicarse dentro de grupos y comunidades. Más recientemente los desarrollos web y móvil permiten crear aplicaciones que interactúen directamente con el usuario y formen parte de su actividad cotidiana.

Dentro de comunidades bien definidas la información tiene una importancia muy grande, su valor ha hecho necesario el desarrollo de sistemas para compartirla a cada elemento dentro de una organización.

1.2 Requerimientos Específicos

Es necesario proporcionar un sistema que permita administrar la información de una organización en forma de mensajes (fechas, eventos, etc.) de manera que puedan ser consultados de manera práctica e inmediata.

Es necesario considerar que la solución no puede implementar un sistema fijo o uno alámbrico debido a la poca practicidad así como los costos que se tendrían para los requerimientos de software y principalmente de hardware. La tecnología actual permite plantear una solución basada en sistema inalámbrico debido a las facilidades técnicas y claro, el bajo costo además del impacto a un mayor grupo de usuarios.

1.3 Objetivos

- Implementar un servicio para facilitar la transmisión de información dentro de una organización.
- Permitir a los usuarios consultar los contenidos desde un dispositivo móvil de forma inmediata y segura desde un punto de acceso preestablecido.
- Crear una herramienta práctica, de fácil administración y bajo nivel de mantenimiento.
- Facilitar el acceso a la información.
- Un sistema con un impacto en un gran número de usuarios a un bajo costo.
- Considerar las herramientas actuales para la implementación, sin la necesidad de crear nuevos elementos y costos.
- Aplicar los conocimientos de nuestra carrera en un proyecto práctico y de gran impacto para la comunidad universitaria.

1.4 Alcances

Para tal fin se pretende desarrollar un portal Web que presente los componentes necesarios para crear y distribuir contenidos, un sistema que facilite en gran parte las operaciones relacionadas con el hardware así como la lógica que representa el control de la entrega de información de forma que sea una tarea sencilla de realizar sin olvidar la capacidad para responder a todas las tareas indicadas.

Así, el funcionamiento será manipulado por componentes que permitan:

- Administrar la información del sistema (Usuarios, mensajes, dispositivos, etc.)
- Generar reportes.
- Creación dinámica de los contenidos.
- Programar entregas.

Debido a que el sistema empleado no es inmune a ser suplantado por cualquier otro usuario o equipo, por ejemplo enviar información errónea o con intenciones diferentes a las del equipo de trabajo, se integrará un control en el contenido de los mensajes para verificar su autenticidad.

El sistema contempla un control de usuarios, a diferencia de los sistemas convencionales de su misma clase, ya que no consideran la seguridad y facilita la suplantación de los contenidos.

El desarrollo se llevará a cabo con herramientas actuales para el desarrollo Web, se considerará desde funcionalidad hasta presentación. Cabe mencionar que uno de los principales componentes será una base de datos, ya que de aquí parten el desarrollo de todo el sistema, igualmente se hace un énfasis en la transparencia que deberá hacer el sistema respecto al hardware para el usuario.

1.5 Método

Dada la naturaleza del proyecto es necesario considerar un conjunto de buenas prácticas como guía para un desarrollo de calidad desde la perspectiva de un sistema de software.

Como se describirá a detalle en el apartado “3.5 Buenas prácticas para el desarrollo de software” del capítulo 3: “Fundamentos Web”, el desarrollo se dividirá en fases que cubran por completo cada una de las etapas necesarias para contar con un proyecto de calidad, con un alto grado de cumplimiento de los requerimientos y con la finalidad de entregar un sistema que requiera un nivel mínimo de mantenimiento. Así, de manera introductoria se presenta cada una de estas etapas:

Fase 1 Inicio: Toma como base el “Plan de Proyecto” que a su vez considera entre otros puntos: Relación de recursos humanos, roles, ciclos y actividades, tiempos y calendario de actividades.

Fase 2 Requerimientos: Especifica las necesidades que el cliente establece, se asignan roles y actividades para el equipo de trabajo.

Fase 3 Análisis y Diseño: Especifica a los desarrolladores el funcionamiento del producto.

Fase 4 Construcción: Es la parte de desarrollo que toma como base la Fase 3.

Fase 5 Integración y Pruebas: Registro de pruebas de acuerdo al calendario de actividades.

Fase 6 Cierre: Entrega del proyecto.

Capítulo II

Conceptos básicos del entorno móvil

Definición de móvil e inalámbrico

El uso de los términos móvil e inalámbrico es comúnmente intercambiado por que estos entornos se interceptan. Pero hay que resaltar que el primero se refiere a la posibilidad de utilizar una aplicación en movimiento, y el segundo a la posibilidad de comunicación de voz y datos a través de ondas de radio.

2.1 Arquitectura de los sistemas de comunicaciones inalámbricas*Componentes del entorno móvil.*

Una solución móvil requiere unir productos y servicios de diferentes partes. Estos componentes tienen su análogo en una solución empresarial. Los siguientes son los componentes definidos para una solución móvil:

- *Operadores inalámbricos y proveedores de servicio*
Se refiere a las diferentes compañías que ofrecen la infraestructura de comunicación para WLAN's, en este caso es importante escoger un operador que brinde el servicio requerido en el área geográfica en la cual la solución quiera ser implementada.
- *Proveedores de hardware independiente*
Junto con los dispositivos móviles, el hardware es de lo más requerido para la solución inalámbrica, como por ejemplo: módems inalámbricos, tarjetas de red, puntos de acceso.
- *Proveedores de software de infraestructura*
La industria del software inalámbrico ha madurado, al aplicar una solución tendrá que seleccionar un proveedor de software que tiene un probado historial y puede demostrar su oferta en el entorno que ha seleccionado.
- *Proveedores de software independiente (Desarrolladores de la aplicación)*
Pueden brindar soluciones industriales que requiera menos trabajo ya que no parten de un desarrollo desde cero. Si se está desarrollando una solución con aplicación común en varias compañías, es probable que exista una solución clave disponible por un proveedor independiente.
- *Integradores del sistema*
Son compañías que se encargan de unir todos los componentes de la solución, con prácticas enfocadas en soluciones empresariales.
- *Fabricantes de dispositivos*
Los dispositivos se clasifican por su tamaño, funcionalidad y costo, es necesario escoger el dispositivo adecuado para la aplicación que se está desarrollando. Por ejemplo, PDA's y laptops ofrecen interfaces más efectivas y sofisticadas que las que podrían ofrecer algunos teléfonos celulares.

Es muy difícil que un solo proveedor pueda ofrecer todas las partes de la solución, pero con una estrategia de asociación, varios proveedores pueden trabajar en conjunto para ofrecer una solución empresarial completa.

También hay que contemplar que no todos los componentes serán necesarios para la solución. Es posible solo usar una parte de la cadena y así producir la aplicación correcta para la tarea. Por

ejemplo, una aplicación puede crearse solo usando un proveedor de software de infraestructura, desarrolladores de la aplicación, y dispositivos móviles. Si la solución es móvil pero no inalámbrica, el operador inalámbrico no será necesario. Muchas aplicaciones no requieren más hardware que el dispositivo mismo.

2.2 Clasificación de los dispositivos móviles

Una amplia variedad de dispositivos móviles están disponibles para cubrir un enorme rango de aplicaciones y usuarios. El rango se extiende desde dispositivos a precios accesibles que cuenten con acceso a Internet hasta dispositivos de propósito definido como laptops, PDAs, y smartphones, etc. Obviamente el tamaño de los dispositivos define sus características y el desempeño que pueden brindar.

Descripción General de Dispositivos

De todos los componentes de una solución inalámbrica, el dispositivo móvil tiene la mayor atención. Esta es la única parte de la solución con la que el usuario final tiene contacto. Cuando un problema se presenta, la mayoría de las veces algún aspecto del dispositivo debe tenerse en cuenta. Estos aspectos pueden ser la conexión inalámbrica, sistema operativo del móvil, mecanismos de entrada, o características de rendimiento (Tabla 2.1).

Características generales
<ul style="list-style-type: none"> • Tamaño y peso del dispositivo. • Memoria disponible para aplicaciones y datos. • Velocidad del procesador: Esto determina qué tipo de aplicaciones pueden ejecutarse. • Características de la Pantalla: Tamaño, profundidad del color. • Soporte para el sistema operativo: Puede ser actualizado a una nueva versión. • Entradas de expansión para añadir periféricos, como mas memoria, módems inalámbricos, receptores GPS, cámaras digitales, etc. • Duración de la Batería. • Módulos integrados, como teclados, infrarrojos, Bluetooth, etc. • Soporte para el software, incluyendo aplicaciones de terceros, herramientas de desarrollo, navegadores, controladores de hardware, etc.

Tabla 2.1 Características generales

Escoger el dispositivo correcto requiere de una evaluación de criterios, siempre procurando un costo justo.

Factores que se deben considerar para la solución móvil son los mecanismos de entrada de datos y las opciones de conectividad inalámbrica, a continuación se detallan cada uno de ellos:

Mecanismos de entrada.

Los teléfonos móviles son diseñados para ser eficientes y simples, permitiendo la operación con una sola mano aun para usuarios nuevos. Las computadoras son diseñadas para poder y riqueza en aplicaciones. Junto con el tamaño y poder del dispositivo, los mecanismos de entrada de datos es uno de los aspectos más importantes para seleccionar un dispositivo móvil. Esto dependerá

sobre todo de la aplicación, y de los niveles de interacción con el usuario, pero también si se trata de una aplicación donde se maneja gran cantidad de datos.

Cuando se trata de métodos de entrada, varias opciones están disponibles:

- Entrada mediante keypad.
- Entrada mediante pluma (stylus).
- Entrada mediante teclado.
- Entrada mediante voz.

Comunicación inalámbrica

Hay tres conexiones básicas para habilitar la comunicación inalámbrica:

- Dos unidades, requiere de dos piezas de equipo para trabajar juntas, una proporciona la conexión y la otra la usa; estos dispositivos pueden comunicarse mediante cable, infrarrojo, Bluetooth.
- Configuración desmontable, usa un modulo para proveer la conexión, obviamente requiere que el dispositivo cuente con la ranura para este periférico.
- Configuración Integrada, una tendencia emergente es incluir la conectividad inalámbrica en el dispositivo.

Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, se tiene la siguiente clasificación:

- **Dispositivo Móvil de Datos Limitados:** dispositivos que tienen una pantalla pequeña, principalmente basada en pantalla de tipo texto con servicios de datos generalmente limitados a SMS y acceso WAP. Un típico ejemplo de este tipo de dispositivos son los teléfonos móviles.
- **Dispositivo Móvil de Datos Básicos:** dispositivos que tienen una pantalla de mediano tamaño, (entre 120 x 120 y 240 x 240 pixeles), menú o navegación basada en íconos por medio de una "rueda" o cursor, y que ofrecen acceso a e-mails, lista de direcciones, SMS, y un navegador web básico. Un típico ejemplo de este tipo de dispositivos son las BlackBerry y los Teléfonos Inteligentes.
- **Dispositivo Móvil de Datos Mejorados:** dispositivos que tienen pantallas de medianas a grandes (por encima de los 240 x 120 pixeles), navegación de tipo *stylus*, y que ofrecen las mismas características que el Dispositivo Móvil de Datos Básicos más aplicaciones nativas del sistema operativo. Este tipo de dispositivos incluyen los Sistemas Operativos como Windows Mobile, Symbian OS, Palm Os, etc.

2.3 Servicios móviles

Los servicios móviles aprovechan el carácter inalámbrico de las comunicaciones por radio, y su movilidad inherente. La siguiente es una clasificación según su utilización:

- Servicios Móviles Privados PMR (Private Mobile Radio). Utilizan redes de radio propietarias, constituidas por un conjunto de estaciones de radio fijas que aseguran la cobertura territorial deseada, formando grupos cerrados de comunicaciones sin conexión a redes públicas.

- Servicios Móviles Semi-Públicos PAMR (Public Access Mobile Radio). Utilizan redes de radio móviles propiedad de grandes empresas, que alquilan capacidad excedentaria, para uso de terceros, o redes instaladas por un operador de red para esta clase de servicios.
- Servicios de Telefonía Pública PLMN (Public Land Mobile Networks). Se prestan mediante redes telefónicas en las cuales la parte de acceso se realiza vía radio. El núcleo de red incluye las funciones de transporte (transmisión y conmutación) de toda la red telefónica.

Pero también se clasifican según la tecnología empleada:

Tecnología GSM

GSM (Group Special Mobile): Estándar para la comunicación mediante teléfonos móviles que incorporan tecnología digital.

Se concibió desde un principio como una red PLMN multiservicio. En la especificación se han definido numerosos servicios básicos: portadores y servicios de telecomunicaciones, con un amplio conjunto de servicios suplementarios, como son los de voz, mensajes cortos y datos. La mayoría de los servicios portadores tienen los siguientes atributos:

- Modo de transferencia de información: conmutación de circuitos.
- Establecimiento de la comunicación: por demanda.
- Simetría de la comunicación: bidireccional simétrica.
- Configuración de la comunicación: punto a punto.

Servicios GSM

Servicios de voz: Se distingue entre la telefonía propiamente dicha y las llamadas de emergencia. Este servicio proporciona la información vocal y garantiza la replicación de los tonos de señalización audibles correspondientes a la red PSTN, ISDN.

Servicio de Mensajes cortos: Permite el intercambio de mensajes de una longitud máxima de 160 caracteres. El servicio SMS garantiza la recepción de los mensajes aunque los destinatarios no se encuentren disponibles en el momento de la transmisión.

Servicio de Fax:

- Conmutación voz/fax G3, permite la conexión de equipos de facsímil del denominado Grupo 3 (según la ITU-T), a estaciones móviles de la PLMN-GSM para el correcto envío y/o recepción de faxes.
- Fax automático, garantiza solamente el funcionamiento de equipos de facsímil del Grupo 3 en modo autollamada respuesta, permitiendo la conexión entre los equipos de fax situados en las redes de telefonía fija o PLMN-GSM.

Servicio de voz para grupos:

Se pueden clasificar estos servicios en:

- Servicios de llamada de voz para grupos (VGCS Voice Group Call Service)
- Servicio de difusión de voz (VBS Voice Broadcast Service)

Tecnología GPRS

GPRS (General Packet Radio Service): Es un conjunto de nuevos servicios portadores desarrollados en la fase 2 de GSM para permitir la transmisión de información en modo de paquete a través de la red GSM, asegurando a la vez el interfuncionamiento de esta con redes externas. Presenta como características importantes la utilización de códigos perfeccionados y la posición de emplear múltiples intervalos de tiempo de GSM en vez de uno solo.

Al tratarse de una conmutación de paquetes, únicamente se factura por información transmitida y no por tiempo, también es posible la simetría con canales diferentes en los enlaces ascendentes y descendentes. El servicio GPRS se convirtió en el puente para la migración hacia los servicios móviles de tercera generación. Este servicio permite al cliente el envío y recepción de datos en modo paquete. La transferencia de la información se puede utilizar entre una entidad emisora y una o más entidades receptoras.

En GPRS la interfaz de radio no cambia, de forma que el servicio es compatible con el de voz. En cuanto al núcleo, hay que añadirse al núcleo de red GSM dos tipos de nodos: SGSN que realiza funciones de conmutación de paquetes y GGSN que proporciona la funcionalidad de pasarela con redes externas.

Se han definido tres clases de móviles GPRS:

- Clase A: Que pueden soportar simultáneamente comunicaciones GPRS y servicios de conmutación de circuitos, es decir, mantener una comunicación telefónica normal y transferir paquetes GPRS al mismo tiempo.
- Clase B: Que pueden conectarse, simultáneamente a ambos sistemas, GSM y GPRS, y escuchar mensajes de aviso en ambos, pero en cada momento, sólo pueden utilizar servicios de conmutación de paquetes o de circuitos, teniendo preferencia por los segundos.
- Clase C: Que únicamente pueden conectarse a redes de datos. Son terminales que no soportan servicio de voz, orientado sobre todo a Internet móvil.

Servicios Móviles de Tercera Generación

Los principios y objetivos de estos sistemas se pueden resumir en lo siguiente:

- 1) Oferta de todos los servicios que proporcionan las redes fijas de telefonía y datos en una red integrada, con el valor añadido de la movilidad.
- 2) Prestación de servicios de comunicaciones, Anykind-Anywhere-Anytime es decir, de cualquier tipo, en cualquier parte y en cualquier momento.
- 3) Roaming mundial en cobertura, operadores y servicios.
- 4) Elevado grado de personalización de forma que el usuario goce de la máxima libertad de elección de su perfil personal de servicios suscritos, sea de forma permanente o transitoria.
- 5) Flexibilidad para soportar y ofrecer simultáneamente servicios multimedia con diferentes rangos de velocidades.
- 6) Capacidad de tráfico elevada.
- 7) Compatibilidad con los sistemas actuales de la segunda generación.

Utilizan la tecnología WCDMA (Wideband Code Division Multiple Access), que aumenta las tasas de transmisión de datos de los sistemas GSM utilizando la interfaz aérea CDMA (*Acceso Múltiple por División de Código*) en lugar de TDMA (*Acceso Múltiple por División de Tiempo*) y por ello ofrece

velocidades de datos mucho más altas en dispositivos inalámbricos móviles y portátiles que las ofrecidas hasta el momento.

2.4 Arquitecturas para aplicaciones móviles.

Eligiendo la arquitectura

Diferentes factores influyen al momento de seleccionar la arquitectura para una aplicación, entre ellos se encuentran: Hacia quién va dirigida, el tipo de conectividad, los requerimientos de seguridad entre otros, con la finalidad de tener un buen funcionamiento de la aplicación.

Para elegir la arquitectura conveniente, es necesario considerar:

Aplicaciones de usuario: ¿Quién es el usuario final?, ¿Qué habilidades debe tener?

Tipo de dispositivo: ¿El dispositivo ya tiene disponible la tecnología necesaria o habrá que agregar algún componente extra?, ¿Qué funcionalidades debe tener el dispositivo?

Conectividad: ¿Cómo se conectará el dispositivo?, ¿Acceso inalámbrico o cableado?, ¿Si es inalámbrico qué tipo de red usar?

Información: ¿Qué cantidad de información será destinada al usuario?, ¿Dónde reside la información, en algún cliente o en el servidor?, ¿Cuántos dispositivos puede tener un mismo usuario?

Notificaciones: ¿El usuario debe ser notificado sobre la actualización de la información?, ¿Qué pasa si el dispositivo está en modo desactivado y no recibe las notificaciones?

Seguridad: ¿La información está protegida contra posibles ataques?, ¿La autenticación del usuario se hace de en la parte del servidor o en la del dispositivo?

Duración de la batería: ¿Es un punto a considerar?, ¿La conexión inalámbrica consume buena parte de la energía de la batería?, ¿Es posible recargar la batería continuamente?

Una vez hecho el anterior análisis es necesario considerar que ninguna arquitectura cumple con todas las expectativas en un desarrollo, regularmente es necesario implementar componentes extras de acuerdo a las necesidades específicas.

A continuación se detallan las 3 principales arquitecturas: Internet inalámbrico (Thin Clients), Clientes Inteligentes (Smart Client) y Mensajería que tiene la característica de poder formar parte de las dos anteriores o trabajar de manera independiente.

2.4.1 Internet inalámbrico (Thin Clients)

Estas aplicaciones tienen por lo general la misma arquitectura de las aplicaciones de Internet cableado. Los componentes son los mismos y sólo difiere la forma en cómo es enviada la información al usuario final. En la parte del cliente existe un dispositivo con un navegador web conocido como mini-navegador debido a su capacidad y tamaño. Estas aplicaciones de Internet inalámbricas son también llamadas "thin clients".

Los componentes principales de la arquitectura de Internet inalámbrico son:

- *Mini-Navegador cliente:* Este navegador utiliza una URL para conectarse al servidor Web, la información es interpretada y mostrada al usuario final.

- *Servidor Web inalámbrico.* Es quien recibe las peticiones HTTP entrantes y regresa una respuesta al cliente.
- *Fuente de información.* El servidor Web accederá a la información con el mecanismo indicado.

Ventajas.

Mínima configuración de software: Permite el correcto funcionamiento de la aplicación y el correcto despliegue de información sin ninguna configuración adicional del lado del cliente.

Interfaz fácil de usar: Los usuarios están familiarizados con la interfaz de un navegador, lo que permite una fácil adaptación.

Información actualizada: Se asegura que toda la información está actualizada y disponible.

Seguridad: Toda la información está contenida del lado del servidor. Ninguna información está del lado del cliente.

Desventajas.

Conectividad: El acceso a la información está restringido a la conectividad inalámbrica, por lo que puede ser complicado cuando los usuarios no están fijos en un área de alcance.

Interfaz simple: Debido a las limitaciones de los mini-navegadores no se consideran los gráficos para minimizar la cantidad de información descargada del servidor, que de otra forma conllevaría a una disminución de la velocidad en la red.

Fase de pruebas: Las pruebas del desempeño de las aplicaciones pueden ser complicadas debido al amplio número de mini-navegadores disponibles.

Disponibilidad: Cuando un problema existe del lado del servidor, los usuarios no tienen disponible el servicio.

Costo. Puede ser un problema si el usuario debe estar conectado continuamente a la aplicación. Por ejemplo en redes donde el cargo se hace de acuerdo al tiempo que el usuario permanece conectado y no por la cantidad de datos transmitidos.

Ejemplos de aplicaciones.

Servicios de información: Cualquier usuario con un dispositivo inalámbrico y mini-navegador puede fácilmente acceder a una gran cantidad de información, incluyendo noticias, clima, etc.

Entretenimiento: Parecido a los sistemas de información, una gran cantidad de información está disponible, incluyendo juegos interactivos y mensajería instantánea.

m-Commerce: Estas aplicaciones tienen una gran aceptación cuando pueden comprarse u obtenerse bienes inmediatamente, como juegos descargables o timbres para celulares.

2.4.2 Clientes inteligentes (Smart Clients)

Son una buena alternativa para las aplicaciones de Internet inalámbrico, donde se crea un software a la medida en lugar de usar un mini-navegador. Este software normalmente contiene mecanismos de persistencia para los datos, lo que implica que las aplicaciones pueden ser ejecutadas en cualquier momento, aun cuando no haya una conexión disponible. No requieren una conexión inalámbrica y la integración con la fuente de información se hace comúnmente a través de una sincronización. Esta comunicación se puede hacer con conexiones inalámbricas o cableadas.

Los componentes principales de la arquitectura un cliente inteligente son:

- *Cliente Inteligente:* Es la aplicación donde la parte del cliente se ejecuta, puede ser una aplicación nativa del sistema o una desarrollada con tecnología Java. Para proporcionar acceso a datos offline, éstos deben ser contenidos en la aplicación.
- *Servidor de sincronización:* La información es enviada desde la aplicación cliente al servidor de sincronización, puede ser de forma inalámbrica o cableada. El servidor se ocupará de que la menor cantidad de información sea enviada, así como de resolver conflictos de transferencia y resolverlos.
- *Fuente de información:* El servidor de sincronización accederá a la fuente de información de acuerdo a un mecanismo predefinido. Este acceso se realiza durante la sincronización si es necesario para el cliente recibir información. Es un proceso sencillo y requiere una conexión activa al servidor hasta que finalice el procesamiento de los datos.

Ventajas

Disponibilidad: El usuario puede tener acceso a la información y realizar operaciones con los datos aún cuando no haya cobertura de una conexión inalámbrica y sincronizarla después.

Interfaz de gran alcance: Contiene gráficos y es muy similar a las aplicaciones que el usuario está acostumbrado a trabajar en aplicaciones de escritorio.

Rendimiento: Se mantienen un buen nivel debido a que no se pierde procesamiento en una comunicación inalámbrica o acceso a datos externos.

Seguridad: Se tiene un mejor control comparado con la arquitectura de Internet inalámbrica debido a que se puede manipular la información en ambos lados de la comunicación.

Costo: El costo puede disminuir mucho con este tipo de aplicaciones debido a que no se considera el precio por tiempo de transmisión ya que sólo la información del dispositivo es transferida del cliente inteligente al servidor de sincronización.

Desventajas

Complejidad en el desarrollo: Debido a que se desarrolla en lenguaje nativo del dispositivo móvil, es necesario aprender un nuevo API para el cliente y si se desarrolla para múltiples dispositivos con diferentes sistemas operativos puede llegar a ser algo muy complicado. La tecnología Java puede ser una buena opción debido a la popularidad de sus desarrollos.

Múltiples ciclos de desarrollo: Con diferentes dispositivos y diferentes sistemas operativos, el desarrollo y pruebas deben completarse para cada uno de ellos. La información obtenida puede ser todo un conjunto de datos que no tengan criterios en común.

Ejemplos de aplicaciones:

Automatización de ventas.

El éxito de realizar una venta depende en muchos casos de la información y el tiempo en que el vendedor pueda proporcionarla, es por eso que una aplicación de este tipo puede proporcionar: información del contacto, catálogos de productos, lista de precios. Una vez que la información de un pedido es levantada y guardada en el dispositivo está lista para su sincronización posterior (offline).

Cuidado de la salud.

La gente involucrada en el campo de la salud puede obtener muchos beneficios de un cliente inteligente, puede crear citas u obtener información de un paciente con sus huellas digitales. Así, es posible por ejemplo, hacer diagnósticos de forma más efectiva y exacta contando con el historial

del paciente. Es así como se comprueba cómo la persistencia de datos dentro de aplicaciones móviles puede incrementar la productividad y reducir errores.

2.4.3 Mensajería

La mensajería se puede presentar de diferentes formas, desde correos electrónicos, alertas y notificaciones hasta mensajes de aplicación a aplicación. Se puede considerar a la mensajería como una arquitectura de aplicación o complemento de otra.

A continuación se muestran las tres categorías de mensajería más comunes:

Usuario a usuario: Es cuando los mensajes se envían de un usuario a otro con una gran variedad de mecanismos incluyendo correo electrónico, mensajes de texto inalámbrico como los SMS (servicio de mensajes cortos) o mensajes instantáneos (IM). Mensajes de texto enriquecido, que incluye gráficos utilizando el Servicio de Mensaje Mejorado (EMS), así como contenidos multimedia usando el Servicio de Mensaje Multimedia (MMS).

Notificaciones y alertas: Son mensajes urgentes que pueden ser entregados a usuarios con dispositivos inalámbricos. Lo que permite asegurar que la información será recibida en un tiempo corto. Estos mensajes pueden contener direcciones de sitios en Internet donde los usuarios pueden encontrar información adicional.

Existen dos principales tecnologías para las notificaciones y alertas: las alertas (HDML) y WAP-Push, que se enfocan en dispositivos inalámbricos de pantallas pequeñas y en mensajes que incluyen una dirección a un portal WAP respectivamente.

Aplicación a Aplicación: En la mayoría de los casos la interacción con el usuario no es necesaria. Es así como se puede enviar información directamente del servidor a la aplicación cliente sin ningún tipo de interacción por parte del usuario.

Tanto en la mensajería de Usuario a Usuario como en Notificaciones y alertas los requerimientos de software extra son mínimos. En el primer caso el fabricante proporciona el software, para el segundo este funcionamiento forma parte del mini-navegador del dispositivo.

Para la mensajería Aplicación-Aplicación se requiere software para que el dispositivo se pueda comunicar y crear un flujo entre el cliente y el servidor, así como para la comunicación sobre una red cableada o inalámbrica.

Componentes de la arquitectura de mensajes Aplicación-aplicación.

Los principales componentes son:

Mensajería del Cliente: La aplicación cliente tiene una lógica propia así como una cola de entrega. Los mensajes se pueden almacenar en entrada o salida para su acceso posterior. Por ejemplo si una aplicación intenta enviar información al servidor cuando no existe conexión, se puede almacenar en salida y enviar cuando se haya establecido la comunicación con el servidor.

Servidor de mensaje: Es la parte del sistema que se comunica con la mensajería del cliente, así como con los sistemas centrales. Este tipo de mensajería es útil desde la perspectiva de los negocios móviles (m-business) debido a que un usuario que envía no tiene que esperar la respuesta del receptor, permitiendo a éste último continuar con sus actividades mientras se ejecutan las tareas de envío.

Fuente de información: El servidor de mensajería puede interactuar con diferentes sistemas dentro de la misma organización como sistemas de bases de datos, aplicaciones de negocios y otros sistemas de mensajería.

Ventajas.

Capacidad de envío: Ambas arquitecturas, clientes inteligentes e Internet inalámbrico pueden mejorar con el envío de mensajes. Para aplicaciones de Internet inalámbrico, estos mensajes pueden contener una dirección de Internet donde el usuario pueda obtener mayor información. Para aplicaciones en clientes inteligentes el envío de mensajes puede avisar al usuario si debe sincronizar la información del cliente o en algunos casos, el mensaje puede avisar sobre el inicio de una sincronización automática desde el servidor.

Almacén y envío: En redes inalámbricas, es una funcionalidad importante para los mensajes permanecer en espera cuando la conexión no está disponible y que pueden ser entregados cuando se restablece. En el cliente, el usuario puede trabajar como si tuviera una conexión disponible, ya que la información será enviada más tarde. En el servidor, mejora el rendimiento debido a que los mensajes serán enviados cuando el usuario esté conectado a la red, garantizando la entrega de los mensajes.

Entrega personalizada de información: Lo destacado de los productos de mensajería es dar al usuario la capacidad de registrar su dispositivo habilitado para mensajes y programar la entrega del mensaje a ese dispositivo en particular.

Comunicación inalámbrica y cableada: La mensajería de aplicación a aplicación puede funcionar sobre ambos tipos de redes. También permite al usuario conectarse a la red cableada cuando la inalámbrica no está disponible. La mensajería usuario a usuario así como las alertas y notificaciones comúnmente requieren de redes inalámbricas para la comunicación.

Desventajas.

La mensajería es un componente adicional para la mayoría de las aplicaciones, por lo que comúnmente son incorporadas dentro de otras sólo cuando hay un beneficio claro, por lo que se consideran muy pocas desventajas en su implementación.

Sin embargo la mensajería agrega complejidad al desarrollo de las aplicaciones móviles. Para la mensajería aplicación a aplicación el desarrollo es requerido tanto en el cliente como en el servidor. Para la mensajería, notificaciones y alertas, la mayoría sino es que todos los desarrollos, son del lado del servidor.

Aplicaciones.

Comunicación.

La tecnología de mensajería es usada de persona a persona en cuestiones tan simples como programar reuniones o hacer preguntas. Funciona muy bien cuando una llamada no es la mejor opción o cuando el usuario desea la entrega inmediata. El SMS es actualmente la tecnología de mensajerías más común aun que podría cambiar conforme la Mensajería Instantánea crezca, permitiendo un servicio entre aplicaciones inalámbricas y usuarios de Internet.

Dispersión de información.

Es la implementación de mensajería más útil para el consumidor. Son aplicaciones que proporcionan al usuario: información de vuelos, información financiera, clima, etc. Prácticamente cualquier información que pueda ser incorporada a un mensaje puede ser dispersa a través de esta tecnología.

Existe una cuestión en cuanto al envío de mensajes debido a la no aceptación de estos, es algo similar al spam en el correo electrónico, debido a que los usuarios serán interrumpidos continuamente para recibir este mensaje. Además es necesario considerar el espacio de almacenamiento en los dispositivos y sus aplicaciones cliente.

Actualmente estas aplicaciones ya se implementan para enviar publicidad y promociones a usuarios móviles. Este es un gran servicio si se cuenta con una suscripción pero es molesta cuando la persona no está interesada en recibir este tipo de mensajes.

2.5 Tecnologías inalámbricas aplicables al sistema.**2.5.1 Wireless Personal Area Network (Red Inalámbrica de Área Personal)**

En el desarrollo de este tema es necesario hacer notar que se desarrollará desde un punto de vista de programación, es decir, desde herramientas, configuraciones e información útil para el descubrimiento de dispositivos, servicios y comunicación. Es así, que este se considera un avance de lo que se expondrá en el tema “*Herramientas Bluetooth en GNU/Linux*”.

2.5.1.1 Bluetooth

La tecnología bluetooth es una forma de comunicación inalámbrica para dispositivos a corta distancia¹, comúnmente a menos de 10 m.

Un dispositivo para iniciar una conexión necesita elegir un dispositivo objetivo y un protocolo de transferencia, antes de establecer la conexión y transferir información. Los dispositivos que establecen una conexión entrante necesitan elegir un protocolo de transferencia y después escuchar antes de aceptar la conexión y transferir datos.

El dispositivo Bluetooth

Cada dispositivo Bluetooth cuenta con una dirección única de 48-bits conocida como dirección Bluetooth o dirección de dispositivo y es idéntica a la dirección MAC que se utiliza en Ethernet, de hecho son controladas por la misma organización IEEE. Esta dirección asignada por el fabricante es única y se mantiene por todo el tiempo de vida del chip.

Nombre del dispositivo.

Al igual que las direcciones IP, es común que una persona no memorice la dirección MAC de su dispositivo por lo que regularmente tendrá asignado un nombre más fácil de recordar, este es el *nombre para mostrar*. Este nombre se visualiza en lugar de la dirección para identificar el dispositivo, el cual es configurable por el usuario quien puede elegir un nombre arbitrario o una frase cualquiera, no hay algún requerimiento en particular para su selección por lo que en ocasiones su identificación puede llegar a ser confusa.

¹ <http://www.bluetooth.com/Bluetooth/Technology/Basics.htm>

Búsqueda de dispositivos cercanos.

El descubrimiento de dispositivos (inquiry) es el proceso de búsqueda y detección de dispositivos cercanos. Es un proceso sencillo, primero se envía un mensaje (broadcast) de descubrimiento esperando una respuesta. Cada respuesta contiene la dirección del dispositivo que contesta y un identificador del tipo de dispositivo (por ejemplo, teléfono celular, computadora, etc.).

Este proceso de descubrimiento toma en promedio de 5 a 15 segundos, sin embargo durante este tiempo uno de los dispositivos cambia de frecuencia en promedio 1000 veces por segundo, y puede elegir 79 posibles frecuencias por las que puede transmitir.

Descubrimiento y conectividad.

Por cuestiones de privacidad y energía, todos los dispositivos cuentan con 2 opciones que determinan sí o no el dispositivo responde a la búsqueda y petición de conexión. Inquiry Scan controla la búsqueda y el Page Scan controla la petición (Tabla 2.2).

	<u>Inquiry Scan</u>	<u>Page Scan</u>	<u>Descripción</u>
Estado	Inactivo	Inactivo	El dispositivo local no es detectable por otros dispositivos Bluetooth, y no aceptará las peticiones entrantes.
	Inactivo	Activo	El dispositivo no será detectable por otros dispositivos Bluetooth, sin embargo responderá a cualquier petición de conexión de dispositivos que ya cuenta con su dirección. Esta es la configuración más común.
	Activo	Inactivo	El dispositivo local es detectable por otros dispositivos bluetooth pero no aceptará ninguna petición de conexión.
	Activo	Activo	El dispositivo local es detectable por otros dispositivos Bluetooth y aceptará peticiones de conexión entrantes.

Tabla 2.2 Descubrimiento y conectividad.

A continuación se muestra un ejemplo de la configuración de un dispositivo Bluetooth.

```
ladai:~# hciconfig
hci0:  Type: USB
      BD Address: 00:50:C2:7F:4A:2A ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNING PSCAN ISCAN
      RX bytes:704 acl:0 sco:0 events:23 errors:0
      TX bytes:327 acl:0 sco:0 commands:20 errors:0
```

```
ladai:~#
```

Protocolo de transporte.

Esta tecnología utiliza diferentes protocolos de transporte, la mayoría de ellos son de propósito general. A continuación se mencionan lo más representativos:

RFCOMM (Radio frequency communication)

Protocolo de propósito general que trabaja correctamente para emular un puerto serial. Provee un flujo confiable de datos y es útil para las aplicaciones que hacen uso del puerto serie de un dispositivo, así como capa de transporte para OBEX sobre Bluetooth.

Un atributo principal de RFCOMM es que, dependiendo de la aplicación o plataforma de desarrollo, en ocasiones es la única solución en cuanto a protocolo de transporte.

L2CAP(Logical link control and adaptation protocol) (*Protocolo de adaptación y control de enlace lógico*)

Es un protocolo basado en paquetes cuya característica es forzar las órdenes de entrega así como transmitir los paquetes de acuerdo a su orden.

La fiabilidad se logra a través de la transmisión de paquetes que pueden o no retransmitirse. Existen tres políticas de retransmisión:

- Nunca retransmitir (hacer el mejor esfuerzo)
- Siempre retransmitir hasta la entrega o falla en la conexión.
- Dejar la entrega y continuar con datos en espera si un paquete no ha sido entregado después de un cierto tiempo.

L2CAP se utiliza como protocolo de transporte para RFCOMM, así cada conexión RFCOMM se encapsula dentro de una L2CAP.

Puertos.

Un puerto tiene como principal objetivo, permitir a las aplicaciones en el mismo dispositivo utilizar el mismo protocolo de transporte. En RFCOMM, los canales 1 a 30 están disponibles para su uso. En L2CAP los puertos son llamados Protocol Service Multiplexers (PSM) y tiene definidos valores entre 1 y 32,767.

Muchos protocolos de transporte están diseñados para aplicaciones específicas por lo que en lugar de lidiar con su diseño algunos de los puertos ya están reservados. Este conjunto se define comúnmente como “puertos conocidos” o “reservados”. Es por eso que se recomienda que las aplicaciones no usen ninguno de los puertos conocidos a menos que se implemente un servicio estándar asignado a ese puerto, por ejemplo el puerto 80 para TCP/IP es reservado para el tráfico web.

La tecnología bluetooth también considera esta cuestión y L2CAP reserva los puertos 1-1023 para uso estándar, por ejemplo el Protocolo de Servicio de Descubrimiento usa el puerto 1. Sin embargo RFCOMM no tiene ningún puerto reservado, considerando el poco número de puertos con que cuenta.

Protocolo de Servicio de descubrimiento (SDP).

Cada dispositivo Bluetooth mantiene un servidor SDP escuchando en un puerto “conocido”. Cuando una aplicación del servidor comienza, registra una breve descripción de sí misma y un número de puerto con el servidor SDP en el dispositivo local. Es así que cuando un cliente remoto se conecta al dispositivo proporciona una breve descripción del servicio que está buscando, y el servidor SDP proporciona una lista de todos los servicios que coinciden.

De esta forma cuando el servidor inicia, puede elegir un puerto de forma arbitraria, un puerto que no esté en uso en el dispositivo local asegurando así que está libre de conflictos.

Registro de servicios.

La descripción de un servicio se registra mediante el servidor SDP desde la aplicación del servidor, es un registro almacenado tanto en el *registro de servicios* como en el registro SDP. Es una lista de atributos y valores. Cada atributo es un entero de 16 bits, una cadena, o una lista de otros tipos de datos. Prácticamente es un diccionario de varias entradas que describen los servicios ofrecidos.

Entre los atributos existen dos muy importantes dentro de esta lista: Id de servicio y la Lista de Id de clase de servicio que comúnmente son utilizados por el cliente para identificar el servicio requerido.

Atributos de SDP

Algunos de los atributos más comunes y su significado se muestran a continuación:

Lista de Clase de servicio: Es una lista de clase de servicio que el servidor proporciona. Es el único atributo obligatorio que siempre debe aparecer en el registro.

Id de servicio: Es un solo elemento que identifica a un servicio en específico.

Descripción del servicio: Es una cadena de texto que describe el servicio proveído.

Lista descriptora de protocolo: Es una lista de protocolos y puertos usados por el servicio

Lista de descripción de perfil: Es una lista de descriptor de perfiles con el cual el servicio cumple. Cada descriptor consta de un UUID y un número de versión.

Registro de servicio: Es un entero que sólo identifica el registro dentro del dispositivo.

Comunicación vía Sockets.

Una vez con la dirección del dispositivo, el protocolo de transporte y el puerto la programación es en cierta forma sencilla.

Un socket en la programación representa un punto final dentro de un enlace en una comunicación. La idea desde el punto de vista de software, es que todos los datos que pasen a través del enlace deben ir dentro o salir del socket.

Existen 2 formas de trabajar con los sockets, ya sea como cliente o servidor.

Sockets de tipo cliente.

Los sockets cliente son bastante simples, una vez que se han creado sólo se necesita una configuración extra para contar con un socket conectado. Su creación se hace considerando el puerto del dispositivo objetivo. El sistema operativo se ocupa de los detalles a bajo nivel, reservando recursos en el adaptador Bluetooth, buscando el dispositivo remoto, formando una red y estableciendo la conexión. Una vez que el socket está conectado, ya puede ser usado para la transferencia de datos.

Sockets tipo Servidor.

Este tipo de sockets son un poco más complejos al momento de obtener una conexión. Tres pasos son necesarios para lograr esto: Primero la aplicación *vincula* al socket con los recursos del Bluetooth, especificando el adaptador y el puerto que se usará. Segundo, el comando “Escuchar” se utiliza para activar el socket en modo “Escucha”. Es así como se comunica al sistema operativo que acepte peticiones de conexiones entrantes en el adaptador local y el puerto donde el socket fue vinculado.

Una de las principales diferencias entre los sockets cliente y servidor es que el socket servidor primero es creado por la aplicación y nunca puede ser utilizado para una comunicación como tal. En su lugar cada vez que este socket acepta una conexión entrante usando el comando “Aceptar”, crea un nuevo socket que representa la nueva conexión establecida. El socket servidor entonces regresa y escucha más peticiones de conexión, y la aplicación debe usar el socket recién creado para comunicarse con el cliente.

Rango de alcance.

Los dispositivos Bluetooth están divididos en 3 clases de acuerdo a su rango de operación:

- Clase 3: Tienen un rango mayor a 3 metros y no son muy comunes.
- Clase 2: Tienen un rango de 10 metros, la mayoría de los dispositivos móviles se encuentran en esta clase, celulares, manos libres y laptops.
- Clase 1: Tienen un rango de 100 metros y están enfocadas a la industria.

Los rangos son sólo estimados, en la práctica este rango puede variar significativamente con el ambiente y su el modo de uso. La fuerza de la señal puede ser afectada por: paredes, edificios, escritorios, entre otros. Otras tecnologías inalámbricas usadas cerca de los dispositivos también pueden interferir con la señal.

Velocidad de comunicación.

En teoría dos dispositivos Bluetooth tienen una tasa máxima asimétrica de 732.2 (Kb/s). Asimétrica significa que sólo un dispositivo está transmitiendo, y simétrica significa que ambos están transmitiendo, uno a otro. En la práctica las tasas de transferencia son menos, debido a que siempre existe ruido en los canales de comunicación inalámbricos.

Como todos los métodos inalámbricos de comunicación, la fuerza de la señal se deteriora con la distancia, entre más alejada este de la fuente.

Para la versión 2.0 se tiene una velocidad de 2178.1 Kb/s de forma asimétrica y 1306.9 Kb/s de forma simétrica.

Radio Frecuencias y Salto de Canal.

Todos los dispositivos Bluetooth operan en la banda de frecuencia de 2.4 GHz. Es decir, en la misma que trabajan los microondas, el 802.11 y algunos teléfonos inalámbricos. La diferencia radica en que el Bluetooth divide esta banda de 2.4 GHz en 79 canales y emplea saltos de canal para que los dispositivos estén siempre cambiando las frecuencias en las que están transmitiendo y recibiendo. Es por eso que la interferencia en un canal no afecta demasiado, debido a que no debe existir ninguna después de un salto a un canal nuevo.

A diferencia del estándar 802.11 que trabaja en la misma banda y que divide los canales en sólo 14 por lo que estos canales están más cercanos unos de otros. Otra diferencia radica en que los dispositivos Bluetooth nunca se quedan en el mismo canal. Una comunicación en un dispositivo activo cambia de canal cada 625 μ s (unas 1600 veces por segundo). Esto se trata de hacer de manera aleatoria de forma que ningún canal sea utilizado mucho más que otro. Es claro, que dos dispositivos que se están comunicando uno con otro, deben saltar de canal juntos, para que siempre puedan estar transmitiendo o recibiendo sobre la misma frecuencia.

Esto se hace con la única finalidad de hacer a Bluetooth más robusto contra las interferencias de fuentes cercanas de ondas de radio y permite a muchas redes bluetooth coexistir en el mismo lugar. Las versiones más nuevas van más allá y utilizan *salto de frecuencia adaptativa*, donde los dispositivos evaden específicamente los canales ruidosos y que tienen alta interferencia.

Seguridad – PIN's

Dos dispositivos Bluetooth pueden tener un procedimiento de autenticación, con el que pueden verificar sus identidades. Una vez autenticados, pueden tener la opción de cifrar todos los paquetes que intercambian. La autenticación también se puede dar sin cifrado, y es útil cuando dos dispositivos sólo necesitan estar seguros de que se están comunicando con la máquina correcta, y no necesariamente preocuparse de si alguien más está atento a la comunicación.

El PIN - secuencia alfanumérica de más de 16 caracteres de longitud, que sólo dos dispositivos comparten y ningún otro dispositivo tiene – Es una especie de contraseña secreta que se comparte en Bluetooth para establecer un canal seguro de comunicación.

La primera vez que dos dispositivos se autentican o cifran sus paquetes, es conocido como “pareo”. Primero, un PIN común debe ser proporcionado a los dos dispositivos, después cada dispositivo usa ese PIN para generar una llave de enlace. La llave de enlace es guardada en ambos dispositivos, y es de hecho la que se usa para cifrar la información transmitida entre los dos (es necesario hacer notar que durante el pareo, el PIN nunca es transmitido).

Ya que los dos dispositivos han hecho el proceso de pareo, las aplicaciones en cada dispositivo pueden requerir la autenticación o cifrado que necesiten. El cifrado trabaja en un nivel bajo (la capa enlace) y si los paquetes para una conexión entre los dos dispositivos son cifrados, entonces los paquetes para todas las comunicaciones entre ellos son cifrados. Es por eso que, la autenticación y el cifrado son comúnmente manejados por el sistema operativo directamente, en lugar de las aplicaciones.

Modos de seguridad.

El modo de seguridad de un dispositivo Bluetooth determina qué políticas implementar al utilizar esta tecnología, así como las peticiones de cifrado de información de dispositivos conectados. A continuación se describen los distintos modos de seguridad:

Modo de seguridad 1(no seguro): Si un dispositivo B requiere el cifrado o la autenticación, el dispositivo A debe aceptar, es decir el dispositivo A nunca iniciará un procedimiento de seguridad.

Modo de seguridad 2 (seguridad impuesta a nivel de servicio): Es el modo asignado por defecto por el sistema operativo. Este modo implementa políticas de acceso para aplicaciones con distintos requerimientos de seguridad.

Modo de seguridad 3(seguridad impuesta a nivel de enlace): Aquí el dispositivo inicia la autenticación y el cifrado siempre que cualquier conexión es establecida, y rechaza aquellas con dispositivos que no realicen el proceso de pareo.

Pareo simple.

Anteriormente con las primeras versiones de Bluetooth existía una muy notable complicación con el intercambio de PIN, debido a que el usuario los elegía de forma incorrecta, por ejemplo 1111 o 0000. Un segundo problema fue que dispositivos sin posibilidades de proporcionar información al dispositivo dado que no existía un elemento de entrada, como los manos libres, regularmente tiene un PIN común como 0000 grabado por el fabricante lo que obviamente desestima el objetivo del PIN. Además, aún cuando los usuarios ya han elegido un PIN y desean iniciar una comunicación tienen que lidiar con interfaces confusas y difíciles de utilizar.

Para tratar estos problemas, la versión 2.1 de Bluetooth introduce el *Pareo Simple*. El cambio más significativo en cuanto al pareo, es que el usuario no necesita introducir un PIN. En lugar de esto, el PIN es automáticamente generado y el usuario sólo recibe una notificación para aceptar o rechazar el pareo. Además de esto el Pareo Simple provee técnicas de cifrado más robustas que actualmente son comunes en protocolo de internet como SSH, IPsec, PGP y SSL.

Perfiles Bluetooth

Un Perfil Bluetooth es un conjunto de especificaciones y métodos útiles para desarrollar aplicaciones específicas y de más alto nivel enfocados a la programación. Son formas

estandarizadas para realizar tareas como transferencias de archivos, trabajo con impresoras o reproducción música.

Algunos de los perfiles más conocidos y utilizados son:

- *Obex Object Push*: Se entiende como OBEX a “Intercambio de objeto” y permite a los dispositivos enviar y recibir datos, que son comúnmente documentos, imágenes, sonidos que pueden ser enviados (pushed) o recibidos (pulled).
- *File transfer*: A diferencia del perfil OBEX que sólo permite un intercambio simple, este perfil permite a un dispositivo acceder al sistema de archivos de otro y enviar o recibir archivos.
- *Hands-Free Audio*: Permite la transmisión de audio entre dispositivos Bluetooth como teléfonos celulares o computadoras.
- *Personal Area Network*: Permite a los dispositivos Bluetooth formar una red IP y compartir la conexión de internet de un dispositivo con otro.
- *Human Interface Device*: Permite a ciertos dispositivos Bluetooth de entrada como teclados o ratones interactuar con computadoras equipadas con Bluetooth.

La interface controladora de host (*Host Controller Interface -HCI*)

La interface controladora de host define cómo una computadora (Host) interactúa y se comunica con el adaptador local Bluetooth (el controlador). Todas las comunicaciones entre los dos son encapsuladas dentro de paquetes HCI y se definen cuatro tipos:

Command Packet: Este tipo de paquetes son enviados desde la computadora al adaptador Bluetooth, y se utilizan para controlar su operación. Este tipo de paquetes también se utilizan para realizar el descubrimiento de dispositivos cercanos, conectar a un dispositivo remoto, configurar los parámetros de conexión entre otros.

Event Packet: El adaptador Bluetooth se encarga de generar y enviar este tipo de paquetes a la computadora (host) siempre que ocurre un evento de interés, por ejemplo: detección de dispositivos, establecimiento de conexión, información acerca del adaptador local Bluetooth, entre otros.

ACL data packet: Los paquetes HCI de este tipo encapsulan información destinada para recibir desde un dispositivo Bluetooth remoto. En este sentido HCI es un protocolo de transporte para ACL, L2CAP, RFCOMM y protocolos de transporte de más alto nivel. Una vez que los paquetes pasan a través del adaptador Bluetooth, las cabeceras HCI son divididas y el paquete es transmitido por el aire.

Synchronous data packet: Los paquetes SCO son también encapsulados dentro de paquetes HCI la primera vez que se transmiten desde el host al adaptador Bluetooth, y son envueltos dentro de este tipo de paquetes. Como con ACL, las cabeceras HCI son divididas desde su paquete cuando son enviadas sobre ondas de radio, y son envueltas de nuevo después de ser recibidas.

Para poder realizar operaciones a bajo nivel, una aplicación necesita acceso a la interfaz controladora del Bluetooth. Muchos entornos de desarrollo no facilitan el acceso a esta capa y proporcionan otras a más alto nivel, que son más sencillas pero también más restrictivas.

Es importante recalcar que la interfaz controladora sólo trabaja para una computadora y su adaptador Bluetooth local. No es posible tener una interfaz que conecte dos dispositivos Bluetooth separados, tampoco es posible que una computadora realice una conexión a un dispositivo Bluetooth remoto que no esté físicamente agregado al host.

Acotaciones

A pesar de que Bluetooth es una de muchas tecnologías inalámbricas, no tienen las mismas características y capacidades que las demás, a continuación una lista de las limitantes que se pueden considerar:

- *Anunciar la presencia de un dispositivo:* No es posible para un dispositivo Bluetooth informar sobre su presencia. Sólo es posible haciendo un descubrimiento de dispositivos cercanos.
- *Detectar cuando un dispositivo remoto está ejecutando un descubrimiento.* Aun que cuenta con notificaciones de eventos de bajo nivel, no tiene alguno que alerte cuando un mensaje de descubrimiento es detectado. El Bluetooth puede detectar este evento, pero no hace visible la información a la computadora (host).
- *Determinar la dirección Bluetooth de un dispositivo que ejecuta el descubrimiento.* Un dispositivo Bluetooth que ejecuta el descubrimiento nunca trasmite la información que lo identifica. Aun cuando fuera posible por software detectar que un dispositivo remoto ejecuta el descubrimiento, no sería posible identificar a ese dispositivo.
- *Distancia de un dispositivo Bluetooth remoto.* Algunos dispositivos Bluetooth registran la potencia de la señal de algún nuevo dispositivo encontrado o de alguno ya conectado. Sin embargo la señal de radio ajusta la señal ya sea para reducir los errores de transmisión o para conservar energía. Es por eso que la interferencia por fuentes de radio, atenuación de la señales causada por diferentes materiales llegan a modificar las medidas de potencia de la señal.
- *Mensajes Broadcast.* No es posible que un dispositivo Bluetooth envíe un mensaje generalizado a todos los dispositivos cercanos.

2.5.2 Wireless Local Area Network (Red Inalámbrica de Área Local)

Las soluciones de redes inalámbricas son uno de los más crecientes segmentos de la industria de telecomunicaciones. La conclusión de estándares industriales, y la correspondiente liberación de productos para redes inalámbricas por líderes fabricantes, inició la implementación de soluciones para redes inalámbricas en muchos puntos del mercado, incluyendo la pequeña oficina, grandes corporaciones, fabricas, y lugares públicos como aeropuertos, centros de convenciones, hoteles e incluso cafeterías.

En algunas instancias la tecnología WLAN es usada para ahorrar en costos y evitar los cables, mientras que en otros es la única opción para proporcionar alta velocidad en el acceso a Internet. Cualquiera que sea la razón, las WLAN están emergiendo en todas partes.

Para direccionar su creciente demanda, las compañías han empezado a liberar múltiples productos, que implementen algunos de los múltiples estándares de esta tecnología. Para evaluar estos productos algunas áreas deben ser consideradas:

- **Rango/Cobertura:** El rango de los productos WLAN está entre los 50 y los 150 metros.
- **Rendimiento:** El rango de la tasa de transferencia de datos va de los 1 Mbps a 54 Mbps
- **Interferencia:** Algunos estándares presentaran interferencia por los equipos electrodomésticos, y otras tecnologías de red inalámbricas.
- **Consumo de Poder:** El monto del poder consumido por el adaptador inalámbrico varía por el producto y por el estándar soportado.
- **Costo:** El costo puede variar significativamente dependiendo de los requerimientos del desarrollo y los estándares implementados.

Estándares de Redes inalámbricas

Estándar 802.11

La especificación IEEE 802.11 fue aprobada en julio de 1997, haciéndolo el primer estándar de redes inalámbricas en ser definido. Utiliza los mismos protocolos que una red Ethernet, pero permite la comunicación sin cables, utilizando una frecuencia de 2.4 GHz.

2.5.2.1 WiFi - 802.11b

Es el estándar más popular de la familia. La especificación fue aprobada al mismo tiempo que el estándar 802.11a en 1999, y desde ese entonces ha tenido una amplia aceptación en el mercado, usando el espectro de 2.4 GHz. Además el espectro de 2.4 GHz está disponible globalmente para configuraciones de redes inalámbricas, mientras que el espectro de 5 GHz que usa el 802.11a es para uso limitado en varios países.

El estándar 802.11b es capaz de alcanzar una máxima capacidad de 11Mbps. Lo que sobrepasa la velocidad del estándar Ethernet original, haciendo a 802.11b una alternativa practica para sustituir ó extender una red cableada. Para ayudar a fomentar la interoperabilidad entre los productos 802.11b, se ha establecido una certificación para la mencionada Wireless Fidelity. Obtener esta certificación asegura que el producto es apto para trabajar con otros productos Wi-Fi. Esta certificación, sumado a la liberación de productos 802.11b por parte de compañías líderes como Cisco, Lucent, Agere Systems, Proxim y 3Com, han convertido al 802.11b al estándar líder.

El uso del espectro de 2.4 GHz tiene sus pros y sus contras. Dentro de las ventajas, es que el espectro es casi universalmente disponible para configuración de redes inalámbricas. Y las señales pueden atravesar barreras físicas como muros y techos de forma más efectiva que otras frecuencias. Del lado de las desventajas, esta la congestión. Desde que es sin licencia, significa que

cualquiera pueda usarlo, por ejemplo los audífonos inalámbricos, el horno de microondas. Con el uso generalizado de su espectro, existe la posibilidad de congestión, provocando demasiada interferencia para la efectiva comunicación de datos.

En configuraciones típicas para oficinas, un 802.11b puede comunicarse con dispositivos alrededor de 100 metros de distancia. Los dispositivos alrededor de 30 metros alcanzan una velocidad de transferencia de 11Mbps, de los 30 a los 65 metros, tienen una tasa de transferencia de 5Mbps, después de 65 metros la tasa baja a 1-2 Mbps.

Seguridad en el estándar 802.11b

Cuando la IEEE creó la especificación para el 802.11, implemento una característica llamada WEP (Wired Equivalent Privacy), con la intención de brindar niveles básicos de autenticación y cifrado de datos.

Desafortunadamente la seguridad WEP presenta algunos inconvenientes. Para autenticación WEP utiliza muy pocas llaves y no posee un mecanismo de actualización de éstas. Lo que provoca mismas llaves se utilicen por diferentes clientes y puntos de acceso. Esto significa que un usuario malicioso puede captar el flujo de información, mediante software libre disponible. WEP está destinado para usarse en conjunto con mecanismos de firewall, redes privadas virtuales, y seguridad a nivel de aplicación.

Actualmente se recomienda usar el cifrado WPA2 (Wi-Fi Protected Access 2) que está basado en el algoritmo AES (Advanced Encryption Standard).

2.5.2.2 Estándar 802.11g

Brinda alta velocidad a la comunicación inalámbrica en una banda de 2.4 GHz, al mismo tiempo que mantiene la compatibilidad con el 802.11b. Esto es realizado en dos capas. En la primera 802.11g opera en la misma frecuencia de 2.4 GHz con el mismo tipo de modulación DSSS, para velocidades arriba de 11 Mbps. Para 54 Mbps, 802.11g utiliza el más eficiente tipo de modulación OFDM, aun en la banda de 2.4 GHz.

Aunque 5 GHz tiene muchas ventajas, también sus desventajas. El más importante de estos es la compatibilidad. Las diferencias entre las frecuencias significan que los productos 802.11a no podían interoperar con los 802.11b, por lo que la IEEE desarrollo el 802.11g, el cual extiende la velocidad y el rango de la 802.11b para que sea completamente compatible con los sistemas de versiones anteriores.

La ventaja obviamente es que mantiene la compatibilidad con el 802.11b y su aceptación mundial, como también ofrece altas tasas de transferencia comparables con 802.11a. El número de canales disponibles, sin embargo, no se ha incrementado, por que los canales están en función del ancho de banda y no de modulación de la señal, en este sentido 802.11a gana con 8 canales comparado con los 3 en 802.11g. Otra desventaja al trabajar en 2.4GHz, presenta los mismos problemas de interferencia que 802.11b.

2.5.2.3 Estándar 802.11n

Entre sus innovaciones clave, el estándar 802.11n añade una tecnología llamada «multiple-input multiple-output» (MIMO), una señal de procesamiento y una antena inteligente para transmitir varias secuencias de datos a través de diversas antenas. El resultado es un rendimiento de hasta 2,5 veces superior y hasta el doble de alcance en comparación con el anterior estándar 802.11g.

2.5.3 WIRELESS WIDE AREA NETWORK (Red Inalámbrica de Área Amplia)

A partir de la década de los 90's este tipo de redes se han utilizado para la comunicación de datos, a diferencia de las WPAN's (Redes inalámbricas de área personal) y las WLAN's (Redes inalámbricas de área local), las redes inalámbricas de gran alcance operan sobre frecuencias reguladas, lo que significa que deben pagar por el espectro que utilizan. En algunos casos es extremadamente caro por lo que para recuperar el gasto, una tarifa debe ser cobrada.

Aún así, las redes de largo alcance proveen un componente importante en las soluciones inalámbricas. Cuando la cobertura de un amplio rango es requerido, las WWAN's son la solución. Pueden proveer cobertura de comunicación nacional y en ocasiones internacional de datos y voz. La calidad y velocidad dependen de la tecnología utilizada.

Señales analógicas y digitales.

Las primeras redes inalámbricas usaron señales analógicas para transmitir sonido. Estas señales cambian constantemente mientras la voz es transmitida, similar a la fluctuación de la propia voz. Debido a la naturaleza fluctuante de las ondas analógicas, son comúnmente representadas con una onda seno.

Las redes inalámbricas evolucionaron y comenzaron a transmitir datos tanto como voz, es así que surgió la necesidad de crear una comunicación digital. Las transmisiones digitales son un flujo de 1's y 0's. Debido a que la información contenida en una computadora es puramente digital, las redes digitales proveen más eficiencia tanto en espectro como en el consumo de energía. Toda la segunda generación de redes inalámbricas con capacidad para manipular datos usa tecnología digital.

Los siguientes son algunos de los muchos beneficios alcanzados para cambiar a las redes digitales:

- *Rendimiento.* Las redes digitales pueden transferir más datos sobre la misma cantidad de espectro; también permiten una compresión para una mayor eficiencia. Adicionalmente las señales digitales consumen menos energía que las analógicas.
- *Seguridad.* Las señales analógicas pueden fácilmente ser escuchadas con un sintonizador de radio, aun cifradas pueden ser escuchadas fácilmente. Es mucho más difícil con las señales digitales debido a que pueden ser cifrados varias veces, dependiendo del nivel de privacidad requerida. Además, las técnicas de distribución las hacen más difícil de decriptar.
- *Calidad.* Las señales digitales tienen una mejor calidad de sonido con menos interferencia. Filtros avanzados pueden ser usados para quitar cualquier ruido.
- *Características.* La tecnología digital proporciona mayores facilidades para los servicios de voz como contestadoras e identificadores de llamadas, así como los servicios básicos para el tráfico de datos.

2.5.3.1 Redes de Primera-Generación (1G)

Las redes inalámbricas de primera generación fueron utilizadas para comunicación de voz y registraban un alto nivel de interferencia, lo que hacía variar su calidad. Al principio estas redes

sufrían caídas de conexiones, baja capacidad e implementaban poca seguridad. Los dispositivos debían tener también un tamaño grande para incorporar los receptores de radio necesarios para capturar la señal analógica.

A pesar de estas dificultades, las redes de primera generación se desplegaron comercialmente en muchos países en la década de los 80's, Estados Unidos y México entre otros desplegaron redes utilizando el Servicio Avanzado de Telefonía Móvil (AMPS).

2.5.3.2 Redes de Segunda-Generación (2G)

Las redes de segunda generación introdujeron las capacidades digitales a las redes inalámbricas a principios de los 90's. Los que llevaron a una mayor calidad en la transmisión de voz así como en los servicios básicos. Además, otras características como el correo de voz, llamada en espera, identificador de llamadas fueron introducidas al servicio de voz.

Otro importante beneficio de cambiar a la tecnología digital fue el incremento de capacidad. La tecnología digital permite a más usuarios comunicarse en la misma cantidad de espectro, aumentando así la eficiencia. Adicionalmente las redes digitales proporcionan una mayor seguridad, una deficiencia de las redes analógicas de la primera generación.

Todas las redes de segunda generación proveen soporte para la comunicación de datos (con excepción de TDMA, que utiliza Cellular Digital Packet Data (CDPD) para su servicio de datos), las tasas de transferencia oscilan entre 9.6 y 19.2 Kbps que es lo suficientemente rápido para las aplicaciones simples de Internet basadas en el Protocolo de Aplicaciones Inalámbricas (WAP) o el Servicio de Mensajes Cortos (SMS).

Son cuatro los principales sistemas de segunda generación que se encuentran en uso:

Digital AMPS (DAMPS conocido como TDMA), que puede aprovechar el servicio AMPS de primera generación.

Code Division Multiple Access (CDMA IS -95) creado por Qualcomm en 1995.

Global System for Mobile Communications (GSM) que es el más popular de las redes 2G.

Personal Digital Cellular (PDG), utilizada ampliamente en Japón.

Todas estas redes están basadas en estándares diferentes por lo que son incompatibles unas de otras; aun utilizando el mismo protocolo de red las frecuencias son otra desventaja ya que la regulación varía de un país o contiene a otro.

Redes de Segunda-Generación (2.5G)

Las redes 2G son un avance hacia las redes de tercera generación, sin formar parte de esta última. Proveen la principal característica que los usuarios requieren para utilizar el Internet móvil: los paquetes de datos. La principal diferencia entre las redes 2G y 2.5G es la implementación de la Conmutación por Paquetes por la Conmutación por circuitos. A continuación se describe cada una de estas técnicas:

La conmutación es el proceso por el cual un usuario se comunica con otro, por medio de un conjunto de elementos para lograr una transferencia de información.

Conmutación de circuitos. Se realiza a través del establecimiento de un canal de comunicación de inicio a fin antes de realizar el envío y tiene la característica de mantener activo este canal durante todo el tiempo que dura la comunicación.

Este procedimiento es común cuando se realiza la conmutación entre dispositivos similares, además no realiza operaciones extra como control de velocidad o de código sin embargo su principal desventaja es el tiempo que toma para el establecimiento de la conexión.

Conmutación de paquetes. Esta técnica trata a la información a través de bloques que contienen una dirección de origen y otra de destino, su principal característica es el empleo de mensajes cortos y de longitud fija (paquetes). Es una técnica resulta muy rápida debido a que los paquetes permanecen muy poco tiempo en la memoria del Procesador de Intercambio de Mensajes (IMP) para transmitirse después como un único proceso. Otra de sus ventajas es la capacidad que tiene el medio de transmisión para ser compartido entre varios usuarios, lo que implica que los recursos se utilizan cuando se envían o reciben datos.

Este avance trajo consigo muchas características positivas, la principal es la alta velocidad de transferencia de datos a más de 144 Kbps, casi 10 veces más que la red 2G.

Dos son los principales protocolos de las redes 2.5G: General Packet radio Services (GPRS), Enhanced Data Rates for Global Evolution (EDGE). Ambos ofrecen muchas mejoras sobre las redes 2G, algunas de sus principales características son:

- *Rendimiento:* Mayor uso de espectro para compartir conexiones entre una gran cantidad de usuarios para comunicación de voz y datos. Esto permite a los operadores tener más usuarios en la misma red.
- *Velocidad:* Al implementar algoritmos más eficientes de modulación y al ser capaz de usar múltiples canales simultáneos para transferencia de datos.
- *Disponibilidad.* Los usuarios pueden permanecer conectados a la red 2.5G sin tener que pagar un cargo por minuto, ya que se hará en una red de conmutación de circuitos (todo el tiempo). Esto permite a los usuarios tener acceso a servicio de datos conforme lo necesiten sin incurrir en cargos por recursos que no se estén utilizando. También permite a las nuevas aplicaciones enviar datos al usuario, en lugar de que los usuarios tengan que pedirlos.
- *Actualización de sistemas 2G.* Ambas tecnologías 2.5G, GPRS Y EDGE, son actualizaciones de redes celulares existentes. Los usuarios tienen las mismas capacidades de voz de antes, pero no tienen acceso a datos a alta velocidad con la misma cobertura de la red. En ocasiones, la migración a 2.5G de 2G involucra una actualización de software para el operador. A nivel técnico, 2.5G proporciona la base con la que la red 3G será construida.

Aplicaciones 2.5G

Con las redes mejoradas y los nuevos dispositivos llegaron nuevas aplicaciones inalámbricas. En algunos casos, las aplicaciones 2.5G son mejoras de las aplicaciones utilizadas en 2G; en otros son algunas que no fueron prácticas en las redes 2G. Un ejemplo son las aplicaciones de internet inalámbrico. En las redes 2G, las aplicaciones WAP son comúnmente basadas en texto con

limitaciones de gráficos. Con las redes 2.5, estas aplicaciones pueden contener graficas con algunas capacidades multimedia. Estas aplicaciones se ejecutarán más eficientemente debido a que las redes son de conmutación de paquetes (por paquete).

Otras aplicaciones de las redes 2.5 incluyen:

- Correo electrónico.
- Mensajería instantánea.
- Acceso a redes LAN
- Aplicaciones para compartir archivos.

2.5.3.3 Redes de Tercera-Generación (3G)

La finalidad de las redes 3G fue llegar a un estándar global para los servicios de alta velocidad de datos y alta calidad para voz. La meta de formar un solo protocolo para los sistemas 3G nunca se logró debido a los diferentes desarrollos tanto en Norte América, Europa y Japón, sin embargo un grupo de trabajo se crearía para avanzar en su desarrollo. Es así que se crearon varias divisiones: Wideband CDMA (WCDMA), y CDMA2000. Para el año de 1999 la ITU (Unión Internacional de Telecomunicaciones) aprobó un estándar para los sistemas inalámbricos de tercera generación, el IMT-2000 (International Mobile Telecommunications).

Sus principales características son:

Mayores tasas de datos. Los sistemas 3G proveen transferencia de datos que van de 144 Kbps a 2 Mb, dependiendo del nivel de movilidad. El IMT-2000 define 3 niveles de movilidad. Todas las redes 3G deben tener soporte a los siguientes requerimientos:

- *Alta movilidad:* 144 Kbps para usuarios rurales que viajan a altas velocidades, más de 120 km/h.
- *Movilidad Completa:* 384 Kbps para usuarios viajando a menos de 120 km/s en áreas urbanas.
- *Movilidad limitada:* 2 Mbps para usuarios que se mueven a menos de 10 km/h.
- *Mejora de la calidad de servicio:* QoS (Quality Of Service) es soportado de fin a fin en los sistemas 3G, a diferencia de su predecesores 2.5G. Esto permite a los usuarios establecer acuerdos con ciertas características de los operadores como transferencia de datos y latencia en red.

Un punto importante a considerar en la tasa de transferencia de datos en sistemas 3G depende de la distancia del usuario de la estación base. Entre más lejos el usuario se mueva de la estación base, es más difícil conseguir altas velocidades.

Aplicaciones 3G.

Muchas de las aplicaciones realizadas para redes 3G son similares a las desarrolladas para sistemas 2.5G. Ambas generaciones de redes soportan aplicaciones IP y proveen una fácil migración para

los desarrolladores. Es así que se crean aplicaciones habilitadas específicamente para hacer uso del incremento de tasas de transferencia que las redes 3G ofrecen:

- Aplicaciones de video.
- Descarga de archivos.
- Descarga de software.
- Voz sobre IP.
- Servicio de localización.
- Servicios de mensajes multimedia.

Capítulo III

Fundamentos para el desarrollo de aplicaciones en Internet

3.1 Arquitectura Cliente/ Servidor

3.1.1 Antecedentes

La evolución de los sistemas ha permitido a los desarrolladores considerar nuevas formas de procesar la información con nuevas tecnologías de distribución de funciones que representan avances en las comunicaciones en red, ampliando el almacenamiento y mejorando notablemente el manejo de las bases de datos que en conjunto muestran un contraste respecto a sistemas anteriores.

La evolución de estos sistemas se debe a cuestiones como:

- Rendimiento de los servidores respecto a su eficiencia y confiabilidad.
- Eficiencia del procesamiento centralizado, debido a la disminución de costos en relación a la cantidad de usuarios.
- Aumento de la productividad con el uso de nuevas tecnologías acordes a la dinámica actual.

En la actualidad este modelo proporciona soluciones a las necesidades de negocios y empresas ya que es versátil y modular, lo que permite la escalabilidad e interacción con otros sistemas. Entre sus ventajas se puede considerar: el mejor aprovechamiento tanto de la red de comunicación como del hardware, mayor seguridad y autonomía, mejor manejo de concurrencia además de que facilita herramientas de alto nivel a los usuarios mientras el sistema se ocupa de procesos centrales.

3.1.2 Arquitectura

La arquitectura denominada Cliente-Servidor, es un sistema distribuido el cual se divide en: Servidores donde residen los datos y Clientes donde son ejecutadas las aplicaciones. Es básicamente una división lógica en donde al cliente también se le denomina frontend y el servidor backend.

Sus principales componentes son (Figura 3.1):

Servidores: Son estaciones de trabajo que conservan los datos de la organización además de mantener en red a todos los demás equipos que se encuentran debajo de la jerarquía (clientes).

Ejemplos de servidores:

- *Servidores de transacciones:* El cliente envía una petición a un procedimiento, función o método en el servidor y el resultado es devuelto al cliente como una unidad lógica (transacción).
- *Servidores de archivos:* El cliente hace una petición para un registro y el servidor responde con la transmisión de éste.

- *Servidores de bases datos*: El cliente envía peticiones por medio de transacciones², el servidor ejecuta la(s) transacción(es) y regresa los resultados al cliente. El servidor provee también mecanismos de concurrencia, seguridad y consistencia de datos.

Entre sus funciones más comunes se encuentran:

- Aceptar las peticiones de los clientes.
- Procesar los requerimientos.
- Procesar y transmitir los resultados de los requerimientos al cliente.

Clientes: Son computadoras que se encuentran en un nivel inferior en la arquitectura, las cuales realizan peticiones a los servicios que ofrecen los servidores.

Entre sus funciones se encuentran:

- Interacción con el usuario.
- Generar las peticiones hacia el servidor.
- Recibir los resultados desde el servidor.
- Presentar la interfaz del usuario.

Red: Sistema de comunicación que permite a los clientes tener acceso a los servicios.

El proceso en general de una petición común es:

1. El usuario crea una petición.
2. La petición es enviada a través de la red.
3. El servidor verifica los privilegios del usuario.
4. El servidor procesa la petición y regresa un resultado.
5. El cliente recibe la respuesta y la muestra al usuario.

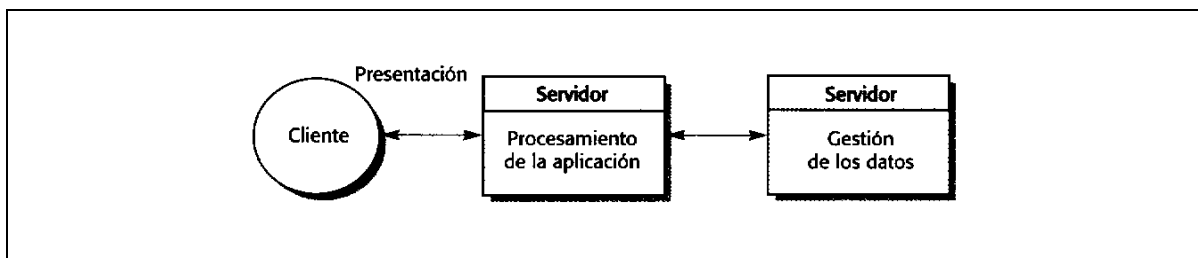


Figura 3.1 Principales componentes de la arquitectura cliente-servidor

La implementación más simple de este tipo de arquitectura se denomina de dos capas, en la que una aplicación se organiza como un servidor (o varios idénticos) y un conjunto de clientes; existen dos tipos:

- *Modelo de cliente ligero (thin -client)*: En este modelo el procesamiento de las aplicaciones y la administración de los datos se lleva a cabo del lado del servidor. El cliente es el responsable de

² Transacción: Conjunto de sentencias en Lenguaje de Consulta Estructurado (SQL).

la capa de presentación. El modelo de cliente ligero facilita la implementación de los clientes en dispositivos de red en lugar de máquinas o estaciones de trabajo. El dispositivo ejecuta un mini navegador de Internet y la interfaz de usuario se implementa a través del sistema. Un desventaja de este modelo es la carga del procesamiento hacía el servidor y la red, debido a la generación de tráfico entre el cliente y el servidor.

- Modelo de cliente enriquecido (fat client): En este modelo, el servidor es responsable de la administración de los datos. El software del cliente implementa la lógica de la aplicación y la interacción con el usuario. El modelo de cliente enriquecido utiliza la capacidad de procesamiento de los sistemas modernos distribuyendo tanto el procesamiento de la lógica como de la presentación al cliente, mientras el servidor se ocupa de ser un administrador de transacciones para las operaciones de la base de datos.
- La principal desventaja del modelo de 2 capas se debe a que las tres capas lógicas (presentación, procesamiento y administración de datos) se deben asociar con dos
- computadoras, el cliente y el servidor. La solución es la implementación del modelo de tres capas, de esta manera la presentación, el procesamiento de la aplicación y la administración de los datos son separados sobre procesadores diferentes. Por ejemplo un sistema bancario en línea es una implementación de la arquitectura cliente servidor de tres capas; la administración de los datos del cliente se encuentran comúnmente en un mainframe; un servidor web proporciona los servicios de aplicación como transferencias y finalmente la computadora del usuario con un navegador de internet es el cliente.

Uso de diferentes arquitecturas cliente-servidor³ (Tabla 3.1)

Arquitectura C/S de dos capas con clientes ligeros	Aplicaciones de sistemas heredados en donde no es práctico separar el procesamiento de la aplicación y la gestión de los datos. Aplicaciones que requieren cálculos intensivos tales como compiladores y poca o ninguna gestión de los datos. Aplicaciones que requieran manejar una gran cantidad de datos (navegar y consultar) con poco o ningún procesamiento de la aplicación.
Arquitectura C/S de dos capas con clientes ricos	Aplicaciones en donde el procesamiento de la aplicación se proporciona por software comercial (por ejemplo, Microsoft Excel) sobre el cliente. Aplicaciones que requieren un procesamiento de datos computacionalmente intensivo (por ejemplo visualización de datos) Aplicaciones con una funcionalidad para el usuario final relativamente estable usada en un entorno de gestión del sistema bien establecido
Arquitectura C/S de tres capas o multicapa	Aplicaciones a gran escala con cientos o miles de clientes. Aplicaciones en donde todos los datos como la aplicación son volátiles. Aplicaciones en donde se integran datos de múltiples fuentes.

Tabla 3.1 Arquitecturas Cliente-Servidor

³ Arquitecturas de objetos distribuidos (Uso de diferentes arquitecturas cliente-servidor.) Ingeniería del software, Séptima edición - Ian Sommerville

La arquitectura Cliente-Servidor se volvió popular conforme fueron aumentando las estaciones de trabajo, sin embargo existen diferentes cuestiones que complican su implementación:

- Aumento de tráfico y saturación de las comunicaciones al aumentar el número de usuarios.
- Problemas de bloqueos por operaciones simultáneas.
- Complejidad de la administración bajo un ambiente distribuido.
- Definición de las reglas de negocio.

3.2 Desarrollo de sitios web.

3.2.1 Introducción

Internet es una red de máquinas que ofrece una gran cantidad de servicios. Existen servidores para correo electrónico, transferencia de archivos, proveedores de acceso entre muchos más. La web es un servicio que está constituido por un conjunto de páginas enlazadas que presentan además de texto y enlaces, imágenes y videos lo que lo convierte en el servicio con más diversidad en internet.

Un sitio web es una entidad transparente para el usuario que sin embargo es un conjunto de archivos e imágenes, todas enlazadas a través de código fuente. La decisión de construir un sitio web debe basarse en una investigación, análisis y una planeación bien definida para poder satisfacer todas las necesidades de la organización.

Si no se logra alcanzar un acuerdo y generar un plan de trabajo, el futuro del desarrollo será incierto. En ocasiones es muy común pensar en el diseño del sitio como un conjunto de componentes como: botones, colores y animaciones; sin embargo es necesario tener siempre presente las necesidades de los usuarios y la facilidad de navegación en el sitio.

3.2.2 Requerimientos

El primer paso para desarrollar un sitio web es realizar el análisis, interactuando y conociendo a los usuarios finales ya que la principal meta es ayudarlos a encontrar sus necesidades y después ofrecer solución a dichas necesidades. Se puede comenzar con un perfil general del usuario, éste puede considerar información personal, ambiente de trabajo (hardware y software), así como requerimientos de alto nivel que sean solicitados. Es una buena práctica considerar diferentes tipos de perfiles ya que dentro de la misma organización pueden llegar a presentarse diferentes tipos de usuarios.

Objetivo del sitio Web

Se debe tener claro el (los) objetivo(s) que se desean cubrir con el desarrollo del sitio web. De igual modo es importante establecer el alcance, sus principales características, necesidades de los usuarios finales y administradores. Además de estas necesidades primordiales, existe un conjunto de objetivos que todo software debe de cumplir:

- **Facilidad:** Toda tarea por más compleja que sea, deberá parecer sencilla para el usuario
- **Eficiencia:** Utilizar los recursos disponibles para desarrollar de la mejor manera posible todos los procesos relacionados con la temática del sitio.

- Usabilidad: El sitio por si mismo debe de tener la capacidad de ser aprendido y usado de forma sencilla.
- Amigable: El usuario debe considerar a la herramienta como una ayuda para mejorar sus tareas actuales y no como un conjunto de componentes que le complican más el desarrollo de sus actividades.
- Rápido: Los resultados se deben obtener de forma más rápida comparados con los obtenidos con procesos ordinarios.

Una vez considerados los requerimientos es necesario establecer un plan de trabajo, mediante el cual se establecen las tareas de cada uno de los integrantes del equipo de trabajo, con el cual se aclaran todos los puntos correspondientes a la implementación para evitar futuras discrepancias en cuestión de requerimientos específicos.

3.2.3 Diseño

De acuerdo con las necesidades establecidas, lo más común es preguntar a los usuarios que utilizarán el sitio sus necesidades particulares que pudieran no haberse contemplado y registrar sus respuestas y posibles soluciones. En ocasiones es útil comparar soluciones ya existentes para completar o detallar la futura implementación. Una vez tomadas todas estas consideraciones ya es posible realizar la estructura del sitio, es decir plantear un diseño apropiado.

En este apartado es necesario considerar:

- *Secciones*: Idear la organización y estructura del sitio web de acuerdo al contenido que se haya analizado.
- *Mapa del sitio*: Hacer una representación general de las secciones del sitio, así como los enlaces entre ellas.
- *Diseño general de una página*: Prototipo de la estructura y contenido en general de cada página del sitio, se pueden tener en cuenta, la cabecera, el cuerpo de la página, el pie de página así como menús y algunas otras opciones.

Actualmente los componentes gráficos son esenciales en la construcción de sitios web, su uso en general se ha maximizado gracias al aumento en cuanto a procesamiento de los equipos de comunicaciones, sin embargo es necesario resaltar que estos componentes son sólo una ayuda en la presentación del sitio y no su funcionalidad como tal.

El abuso de su implementación presenta desventajas como: cargas lentas del servicio, contenido difícil de visualizar o una completa desorientación en la navegación. Además es necesario considerar las situaciones a las que el usuario se enfrentará ya que puede creer que la página tarda demasiado en desplegarse y no está dispuesto a cambiar la velocidad de su conexión, además de que no quiere lidiar con más de un navegador de internet.

La finalidad de los gráficos en el diseño es hacer agradable la navegación del usuario, comenzando por ideas sencillas para llegar a concretar un diseño entre colores, imágenes, etc.

La implementación del diseño debe ser una tarea donde se identifiquen bien las herramientas que se emplearán, actualmente la fase de construcción se facilita si se hace uso de un estándar, una de las principales ventajas de hacerlo es el gran número de usuarios que podrán usar el servicio sin

importar el navegador de internet que utilicen o con qué dispositivo consulten la página; cuando un desarrollo no se basa en un estándar es seguro que el proyecto generará más costos en tiempo y recursos debido a la necesidad de hacer que el sitio sea compatible con la mayoría de los navegadores y dispositivos existentes.

El Lenguaje Web.

Un sitio web está representado en código HTML la cual es una herramienta estándar que indica el orden y la presentación de todos los elementos en la página.

Existen dos tipos de páginas en la web:

Estáticas – Sólo muestran texto y enlaces, se pueden crear de manera sencilla y regularmente el texto presentado es plano con algunas imágenes o videos.

Dinámicas – Existe interacción con el usuario y tienen complementos de diseño que mejoran los efectos en el contenido. Es necesario utilizar otros lenguajes de programación y aumenta el grado de complejidad en su desarrollo.

Algunas de las ventajas de construir una página dinámica son:

- Almacenamiento permanente de la información en una base de datos.
- Automatizar procesos de actualización.

Existen dos variantes de este tipo de páginas, las que se ejecutan en el cliente y las que lo hacen en el servidor.

Las primeras hacen todo el procesamiento y ejecutan todo en la parte del cliente, es decir, en el navegador.

Están conformadas por objetos y funcionalidades dentro de un script que es interpretado por el navegador; se desarrollan principalmente con Java Script y Visual Basic Script (VBScript). Tienen la característica de depender de la máquina donde se ejecutan, siendo esta su principal desventaja debido a que cada navegador tiene un funcionamiento independiente, lo que implica que uno puede interpretar de una forma completamente diferente de lo que lo haría otro. Una ventaja es que utilizan recursos de la máquina local para ejecutar algunas operaciones del usuario lo que incrementa la velocidad de respuesta.

Las páginas dinámicas de servidor son aquellas que son interpretadas y ejecutadas por el servidor en sí, son prácticas cuando existe información que sólo puede ser contenida en el servidor. Su funcionamiento es muy similar a las páginas que se ejecutan del lado del cliente, cuando se envía una petición del cliente se ejecutan los scripts y se regresa un resultado con código que puede interpretar cualquier navegador, es decir código HTML. Para estos desarrollos se utilizan lenguajes como: Active Server Pages (ASP), Hipertext Preprocesor (PHP) y Java Server Pages (JSP).

Una de las principales ventajas de esta implementación es la seguridad debido a que el cliente no tiene acceso a todos los datos del servidor y los scripts sólo regresan código HTML. La desventaja es la disminución en la capacidad del servidor debido al número de clientes que puede atender.

Desarrollo Web mediante scripts

Ya se ha explicado que las páginas en internet se desarrollan con HTML, sin embargo este no es un lenguaje de programación, es una herramienta que junto a otros lenguajes hacen posible la

versatilidad y funcionalidad de una página , así como mejorar sus automatización y responder a los estándares actuales de sitios más elaborados.

Java script.

Es un lenguaje de programación del lado del cliente que se utiliza para desarrollar programas que respondan a ciertas acciones dentro de una página web.

Es útil para proporcionar interactividad con el usuario, así como para efectos más elaborados a la página, una de sus principales características es que proporciona al programador todos los elementos de la página para poder trabajar con ellos de manera dinámica.

Los desarrollos Web se denominan applets.

Visual Basic Script

Lenguaje de programación que se ejecuta del lado del cliente, es útil para crear efectos especiales en las páginas así como para acceder a sus atributos. Su sintaxis se basa principalmente en el lenguaje Visual Basic, por lo que su programación es similar. Su principal desventaja es que los desarrollos están enfocados en el navegador Internet Explorer, por lo que actualmente se prefiere utilizar Java Script para ampliar la compatibilidad con más navegadores.

ASP (Active Server Page)

Es una herramienta que utiliza HTML para el desarrollo de páginas dinámicas del lado del servidor. Como principales características tiene: Acceso a datos en el servidor, ejecutar peticiones para ser enviadas al cliente, ejecutado del lado del servidor por lo que puede ser interpretado por cualquier navegador, interacción con otros lenguajes de programación, acceso al sistema de archivos, es posible incluir componentes desarrollados con anterioridad para resolver ciertos requerimientos, facilidad de programación con una gran variedad de editores, es una herramienta gratuita, buen rendimiento cuando se implementa en plataformas Windows. Esta herramienta actualmente ha evolucionado a ASP.NET con una ligera diferencia en la sintaxis.

PHP (Hypertext Preprocessor)

Es un lenguaje de programación que posee elementos tales como: uso de variables, sentencias condicionales y de control, operaciones y funciones. Es ejecutado del lado del servidor y se enfoca al desarrollo de sitios web. Una de sus principales características es la facilidad de su manejo con las bases de datos, independencia de la plataforma además de tener una gran diversidad de bibliotecas con funciones de utilidad.

3.2.4 Pruebas y resultados.

Es una fase en la que se realizan pruebas a los módulos o al sistema en general en busca de algún detalle que pudiera comprometer su funcionamiento. La fase de pruebas va más allá de la simple revisión de la presentación del sitio, considera también el rendimiento y funcionamiento de los servicios. Es de gran ayuda cuando un futuro usuario colabora directamente con el equipo de trabajo realizando recomendaciones e interactuando con el encargado para realizar mejoras constantes durante todo el proyecto.

En el plan de pruebas se asignarán los roles a cada integrante del equipo, se deberá controlar el reporte de las pruebas y considerar: En qué modulo se realizan las pruebas, como se describen las

fallas encontradas, a quien se tienen que notificar, cómo se resolverán, como organizar los reportes de acuerdo a su naturaleza y las reuniones para descurtir los reportes generados.

Algunas de las pruebas más comunes para realizar en un desarrollo web son:

- De usabilidad: Es una prueba que tiene como principal objetivo registrar la interacción del usuario. Una forma de desarrollarla es observar el comportamiento del usuario mientras navega por el sitio.
- De Integración: Verifica que la comunicación entre los diferentes módulos y acciones de todo el sitio sea correcta y los procesos realicen las tareas que deben hacer cuando se comunican entre ellos.
- De Seguridad: En este apartado se buscan huecos de seguridad, es de suma importancia en sitios que ofrecen servicios donde las transacciones son comunes, además de manipular información importante para la organización.
- Funciones de sistema: Se revisan las funciones asociadas a cada modulo del desarrollo, se consideran desde los procesos y la navegación hasta la comunicación y transacciones con el servidor.

La fase de pruebas se considera costosa en cuanto a recursos, sin embargo tiene un menor costo comparado con una posible eventualidad que pueda ocurrir cuando se libera el producto, es por eso que debe hacerse ver al equipo como una utilidad y no como un trabajo sin sentido.

3.2.5 Mantenimiento

Dada la naturaleza de un proyecto web es muy común que después del desarrollo (ya una vez hecha su presentación y puesto al día para brindar el servicio) requiera de mantenimiento constante ya que regularmente se generan nuevos requerimientos por lo que en algunas organizaciones además del grupo de desarrollo se cuenta con un grupo de mantenimiento que por lo general sólo implementa pequeñas mejoras.

Es aquí cuando la documentación y el trabajo en equipo reflejan el verdadero valor a estos elementos en conjunción, la información que se tiene y sobre todo las experiencias que se puedan compartir evitan una futura falla una vez que se haya entregado una versión completa del proyecto. Una buena práctica son las reuniones donde se podrán compartir soluciones a problemas encontrados con anterioridad con la finalidad de agilizar el proceso de respuesta y así disminuir los periodos de revisión para el mantenimiento.

Errores

Existen un conjunto de errores muy comunes durante la navegación en un sitio web, revisando los archivos "logs" (o de bitácora), con herramientas de análisis se pueden generar reportes de los módulos que enlisten problemas dentro de sitio. Los errores más comunes se describen a continuación:

Error 500 "internal server error": Es un error que indica que el servidor es incapaz de responder a una petición valida, por ejemplo cuando un usuario llena un formulario y al momento de enviarlo la página despliega este error, en estos casos es necesario revisar los permisos de los scripts involucrados.

Error 404 “file not found”: Se debe a que los recursos solicitados no fueron encontrados, en ocasiones se debe a un error tipográfico con la dirección URL.

Error 403 “forbidden”: Este error indica que el recurso solicitado requiere autenticación, es común cuando el usuario no ingresa correctamente sus datos para su ingreso, si no es el caso es probable que exista una falla en la codificación.

Monitoreo

El monitoreo del sitio es útil para conocer sus puntos fuertes y débiles. Es común la creación de un reporte como base para futuros análisis considerando los puntos pico y/o puntos de menor tráfico de acuerdo a rangos de tiempo definidos, por ejemplo: días, semanas, meses, etc. Teniendo dicha información como respaldo para detectar en qué momento ocurren determinados cambios, lo cual ayudará a detectar y solucionar de manera oportuna las eventualidades que se vayan presentando.

En la actualidad existen herramientas que proporcionan este tipo de análisis, ya sea que formen parte del servicio de alojamiento del sitio web o como parte de un conjunto completo de herramientas especializadas. La forma de implementar ambas herramientas está ligada al nivel deseado de análisis y del número de usuarios a los que se les vaya a proporcionar el servicio, esto pensando en que el sitio web vaya a crecer en cuanto al número de usuarios o bien su alcance territorial vaya a crecer.

3.3 Bases de datos

3.3.1 Introducción

Las últimas décadas se han caracterizado por un rápido crecimiento en el número e importancia de las aplicaciones que hacen uso de bases de datos. Las bases de datos son una herramienta esencial para los sistemas de información, las cuales nos permiten interactuar y manipular información con y para el usuario.

Actualmente debido a la gran cantidad de datos y a la necesidad de obtener información de ellos, se pueden listar muchas de las ventajas que ofrece una base de datos en comparación al uso de sistemas de archivos, aquí se mencionan algunas:

- *Reducción*: No hay necesidad de archivar en grandes volúmenes.
- *Velocidad*: Los dispositivos de almacenamiento recuperan y actualizan la información más rápidamente que cualquier persona.
- *Información actualizada*: En cualquier momento, tendremos a la mano la información más reciente y precisa.

De tal manera que podemos definir a una base de datos como un conjunto de datos relacionados con cierto orden y guardados en algún lugar, considerando los beneficios específicos como:

- *Los datos pueden compartirse*: Satisfacer los requerimientos de datos de aplicaciones sin tener que agregar información a la base de datos.
- *Reducir la redundancia*: Optimizar el espacio de almacenamiento, debido a la repetición innecesaria de los datos.

- *Evitar la inconsistencia (hasta cierto grado):* Controlando o evitando la redundancia, se garantiza que la base de datos será consistente.
- *Manejo de transacciones:* Garantizar que una operación se realice de principio a fin.
- *Mantener la integridad:* Asegurar que los datos de la base estén correctos.
- *Seguridad:* Es posible establecer restricciones para cada tipo de acceso

Considerando estos puntos podemos definir una Base de Datos Relacional como:

Conjunto de datos interrelacionados, almacenados en una computadora; de modo que estos datos guardan independencia física y lógica, consistentes, íntegros y con redundancia controlada.

Una buena Base de Datos no es algo que simplemente suceda; el esquema debe diseñarse con cuidado, ya que facilitará la administración y se vuelve un valioso generador de información, mientras que una base mal diseñada se volverá un generador de datos redundantes.

3.3.2 Sistema manejador de bases de datos

Con la globalización del uso de sistemas de información se desarrollaron aplicaciones específicas así como herramientas para la administración de datos denominadas DBMS (Data Base Management Systems) ó SMBD (Sistemas manejadores de bases de datos), estos sistemas permiten a los usuarios en general, plasmar sus diseños de manera esquemática con lo cual facilitan:

- Almacenar físicamente los datos.
- Garantizar consistencia en los datos.
- Garantizar integridad de los datos.
- Atomicidad transaccional.
- Manejo de vistas de información.

Los componentes de un sistema de bases de datos, son los siguientes:

- Hardware: Infraestructura física de la base de datos.
- Software: Son los programas de aplicación.
- Datos: La información contenida.
- Usuarios: Cualquier persona que interactúe con la base.

El SMBD maneja todas las solicitudes de acceso a la Base de Datos formuladas por los usuarios, para la adición, eliminación y actualización de archivos ó tablas. Así el SMBD separa a los usuarios de la Base de Datos de detalles a nivel de equipo.

Conceptualmente lo que sucede es lo siguiente:

a) Un usuario solicita acceso, empleando algún sublenguaje de datos determinado.

- b) El SMDB interpreta esa solicitud y la analiza.
- c) El SMDB ejecuta las operaciones necesarias sobre la base de datos almacenada.
- d) El resultado de la consulta es regresado al usuario.

La arquitectura de la base de datos se basa en los tres niveles que define el grupo ANSI/SPARC, con la finalidad de separar los programas de aplicación de la base de datos física.

- *Nivel Físico (Interno)*: Es el nivel que define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.
- *Nivel Conceptual*: Trata la representación de entidades, atributos, relaciones, operaciones de los usuarios y restricciones.
- *Nivel Vista (Externo)*: Es el nivel de mayor abstracción, en este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios, es la parte del modelo conceptual a la que cada uno tiene acceso.

Con esta arquitectura podemos definir dos tipos de independencia de datos:

- *Independencia lógica*: Es la capacidad de modificar el esquema conceptual sin alterar los esquemas externos o los programas de aplicación.
- *Independencia física*: Es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual.

Componentes de un SMDB Relacional (Tabla 3.2)

Lenguaje de Definición de Datos	Permite la definición de los objetos de la base de datos. Puede usarse para crear, alterar o borrar relaciones (tablas), vistas, restricciones de integridad (por ejemplo, llaves primarias y llaves foráneas), etc. Define los objetos (tablas, tipos de datos, índices, reglas, defaults, vistas, triggers, procedimientos almacenados).
<p>Ejemplo:</p> <p>Create <code>CREATE USER usuario WITH PASSWORD 'hola123';</code></p> <p>Alter <code>ALTER USER usuario WITH PASSWORD 'adios321';</code></p> <p>Drop <code>DROP USER usuario;</code></p>	
Lenguaje de Control de Datos	Permite la asignación de los diferentes permisos sobre los objetos existentes en una base de datos. Permite la definición de lo que los usuarios pueden hacer dentro de la base de datos.

Tabla 3.2 - Componentes de un SMDB Relacional

<p>Ejemplo: Grant GRANT INSERT ON messages TO PUBLIC; Revoke REVOKE INSERT ON messages FROM PUBLIC;</p>	
<p>Lenguaje de Manipulación de Datos</p>	<p>Permite el manejo o procesamiento de los objetos de la base de datos. Puede usarse para consultar, modificar, borrar, o agregar tuplas (renglones) a las relaciones existentes. Los DML se distinguen por sus sublenguajes de recuperación subyacentes: el procedural (manipula registros individualmente) y el no procedural (manipula un conjunto de registros).</p>
<p>Ejemplo: Insert INSERT INTO messages VALUES ('Mensaje1', 'Contenido de prueba', 100, 1, 3, t, f, '2009-09-10 19:00:00', '2009-09-10 12:01:00'); Update UPDATE messages SET role_id = 1 WHERE role_id = 3; Select SELECT message_id, message_title FROM messages WHERE role_id = 3; Delete DELETE FROM devices WHERE device_id <> 1;</p>	
<p>Lenguaje de Manipulación de Datos</p>	<p>Permite el manejo o procesamiento de los objetos de la base de datos. Puede usarse para consultar, modificar, borrar, o agregar tuplas (renglones) a las relaciones existentes. Los DML se distinguen por sus sublenguajes de recuperación subyacentes: el procedural (manipula registros individualmente) y el no procedural (manipula un conjunto de registros).</p>

Tabla 3.2 - Componentes de un SMBD Relacional (Cont.)

Un ejemplo es SQL:

Insert

```
INSERT INTO messages VALUES
  ('Mensaje1', 'Contenido de prueba', 100, 1, 3, t, f, '2009-09-10
19:00:00', '2009-09-10 12:01:00');
```

Update

```
UPDATE messages SET role_id = 1 WHERE role_id = 3;
```

Select

```
SELECT message_id, message_title
  FROM messages
 WHERE role_id = 3;
```

Delete

```
DELETE FROM devices WHERE device_id <> 1;
```

Diccionario de Datos

En un sistema de bases de datos relacional se necesita saber información acerca de los datos y sus relaciones. Esta información se denomina *diccionario de datos, o catálogo de sistema*.

El diccionario de datos almacena información acerca de la estructura de la base de datos y la información de autorización, como las restricciones de la clave.

Tabla 3.2 - Componentes de un SDBD Relacional (cont.)

3.3.3 Análisis de bases de datos

Un SDBD utiliza un esquema para definir toda la estructura de la base. El primer punto a considerar en la creación de la base es el tipo de información que almacenará, igual de importante es considerar la información de la primera instancia y la necesaria para esta primera etapa. Una vez organizada toda la estructura y los componentes de la base se puede garantizar una correcta administración y uso de la base.

El diseño de una base de datos relacional consta de tres modelos que en conjunto forman una abstracción de los datos, definiendo la forma en que serán organizados y relacionados.

Modelos de datos.

Un modelo es una representación de la realidad que contiene las características generales de algo que se va a abstraer. Un modelo de datos es una colección de herramientas conceptuales para

describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia. Es el resultado del análisis de la información, y consiste en la representación conceptual de esta.

Modelo Conceptual.

En este modelo se realiza el análisis de los datos, se consideran los desarrollos existentes y aplicables a futuro. Se hace uso de este en la fase de análisis y define aspectos importantes para el negocio y la asociación entre dicha información. A continuación se listan un conjunto de herramientas para el desarrollo del modelo:

- Diagramas de flujo.
- Diagramas entidad relación.
- Árboles de decisión.
- Pseudocódigo

El modelo entidad-relación

Es un modelo que fue presentado por Peter Pin-Shan Chen en 1976. Se identifica por sus siglas E-R. Se basa en una representación del mundo real en la que los datos se mapean en entidades, relaciones y atributos mostrando siempre una independencia respecto a la base de datos y el sistema operativo. El modelo entidad-relación se utiliza en la fase de análisis de un proyecto. Mientras que el modelo relacional se emplea en la fase de diseño de un proyecto. Sus componentes son:

Entidades: Son una abstracción de un objeto a partir del cual se obtendrá información relevante para el análisis, por ejemplo, auto, empleado, etc. En un Diagrama Entidad-Relación (DER) la representación se realiza con un rectángulo con el nombre de un sustantivo(Figura 3.2).

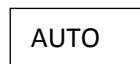


Figura 3.2

Estas entidades pueden pertenecer a cualquier de las dos clasificaciones existentes:

- Regulares: Son aquellas entidades independientes, es decir, su existencia no depende de alguna otra, por ejemplo: ROLES
- Débiles: Son aquellas cuya existencia depende de la ocurrencia de un elemento de esta en alguna otra entidad, por ejemplo: AUTO depende de la entidad MOTOR.

Atributos: Las características de las entidades se definen como atributos, la instancia de cada ejemplar para una entidad tiene las mismas características, la única diferencia entre ellas es el valor que toma para cada uno de los atributos. Son cuatro los tipos de atributos:

- **Obligatorios:** Son aquellos a los que debe asignársele un valor siempre.
- **Opcional:** Son aquellos a los que puede o no asignársele un valor.
- **Monovalorados:** Son aquellos a los que se les asigna un único valor.
- **Multivalorados:** Son aquellos a los que se les asigna varios valores, ó bien ninguno.

Dominio: Es el conjunto de valores que puede tomar un atributo en una entidad, se puede considerar una restricción considerando valores predeterminados.

Claves: Son atributos que identifican de forma única la instancia de un ejemplar en la entidad. Estos identificadores se definen como “Clave Primaria” o “Primary Key”.

Relaciones: Son aquellas que vinculan o asocian una ó mas entidades con otras. Por ejemplo, entra las entidades AUTO y MOTOR se pueden vincular como “POSEE”. Su representación en un DER se realiza con una línea que conecta a las entidades relacionadas, es común que un verbo se asocie a esta relación para identificarla.

Grado de una relación: Es el número de entidades asociaciones en la relación, por ejemplo una relación binaria indica el grado igual a 2.

Cardinalidad

La cardinalidad especifica el tipo de asociación entre las relaciones. Básicamente existen tres casos, a continuación se detallan cada uno de ellos:

Considerando las entidades “PERSONA” Y “EMPLEO”

1:1 .- Indica que para cada PERSONA labora en un solo EMPLEO (Figura 3.3).



Figura 3.3

M: 1 .- Cada empleo puede tener a muchas personas laborando. Sólo se puede asociar un empleo con cada persona, pero varias personas se asocian con el mismo empleo (Figura 3.4)



Figura 3.4

M: M .- Cada empleo puede tener a muchas personas laborando y cada persona puede estar laborando en más de un empleo(Figura 3.5).



Figura 3.5

La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus instancias requiere la existencia de, al menos, una instancia de la otra entidad participante. Si no, la participación es *opcional (parcial)*.

Modelo lógico

Son los requerimientos impuestos por el SMBD. Es un conjunto de reglas de integridad, que exige la ocurrencia de cierto tipo de objetos en cualquier base de datos. En el contexto relacional se compone de entidades, atributos y relaciones.

El modelo relacional fue propuesto originalmente por E.F. Codd en el famoso artículo "A Relational Model of Data for Large Shared Data Banks" en 1970. La estructura fundamental del modelo relacional es precisamente esa "relación", es decir, una tabla bidimensional constituida por renglones (tuplas) y columnas (atributos). Cada atributo de una relación se caracteriza por un nombre y por un dominio.

Componentes del Modelo Relacional

Es la representación de los datos y sus relaciones a través de tablas, donde los renglones representan un registro y las columnas los atributos. Los siguientes son los componentes que conforman al modelo relacional:

Entidad: Es una persona, lugar, evento u objeto identificado de forma única, del cual se registra información además de mantenerla, correlacionarla y desplegarla. Estas pueden ser de dos tipos:

- **Tangibles:** Empleados, Alumnos, artículos.
- **Intangibles:** Un suceso, actividad, cuentas.

Relaciones: En el contexto relacional a una tabla se conoce como relación. Una relación es un conjunto de tuplas (registros) y atributos (campos).

Atributos: Son las características de la entidad. Los atributos se modelan como columnas de la entidad. La forma de diferenciar las entidades es por medio de sus atributos y cada una de ellas debe diferir al menos en uno.

Tupla: Conjunto de valores que componen un renglón de la relación. Es el renglón n en una tabla.

Grado de una tupla: Número de atributos que tiene una tupla.

Cardinalidad: Número de tuplas de una relación

Dominio: Conjunto de valores validos para un atributo.

Tipo de llaves.

- *Llave primaria:* Es el atributo que identifica de manera única a un registro de una tabla.
- *Llave candidata:* Atributo o conjunto de atributos que podrían servir como llave primaria.
- *Llave secundaria:* Son aquellas llaves que tienen todas las características para ser primarias, pero al existir otras que cumplieron mejor con los objetivos no fueron elegidas.
- *Llave extranjera o foránea:* Es una llave primaria de otra tabla y al mismo tiempo forma parte de otra tabla como atributo.

Asociaciones.

Una asociación es la unión de dos o más relaciones, generalmente las asociaciones requieren de al menos 2 tablas. Existen 3 tipos de asociaciones.

- Uno a uno (1:1): Si una instancia de una relación se pueden enlazar sólo a una instancia de la otra relación.
- Uno a muchos (1: M): Si cada instancia de una relación está relacionada con muchas instancias de otra relación.
- Muchos a Muchos (M: M): Cuando se asocia una instancia en una relación, con muchas instancias en otra relación y viceversa.

Integridad de datos.

La integridad de datos se refiere a su corrección y consistencia. Los Constraints (Restricciones) de integridad de datos aseguran que los usuarios realicen únicamente operaciones en las cuales dejarán a la base de datos en un estado correcto y consistente. A continuación se presentan sus diferentes tipos (Tabla 3.3):

<u>Tipo de Restricción</u>	<u>Descripción</u>
Integridad de Entidades	Ninguna parte de la llave primaria puede ser nula.
Integridad referencial	Una llave foránea debe coincidir con un valor de una llave primaria.
Integridad de Columnas	Una columna debe solo contener solo valores consistentes con el formato de datos definidos para la columna.
Integridad definida por el Usuario	Los datos almacenados en la base de datos deben cumplir con las reglas del negocio.

Tabla 3.3 – Tipos de integridad

Básicamente un dato es inconsistente si existen múltiples copias (no fundamentadas para su uso) de un registro y no todas las copias han sido actualizadas. Una Base de datos inconsistente provee información incorrecta o contradictoria a los usuarios.

Normalización.

Una vez que se ha implementado el diseño, es posible encontrar algunos detalles o fallas en el diseño como redundancia o ambigüedades, en estos casos la solución es aplicar las reglas de normalización.

La normalización es el proceso de organizar las entidades y sus relaciones en formatos de tabla utilizando los conceptos relacionales; consiste en agrupar a los campos de datos en un conjunto de tablas o relaciones que representen a las entidades, sus características y enlaces con otras relaciones de manera adecuada.

La principal razón para aplicar este proceso es asegurar que el modelo conceptual funcionará. Es necesario considerar que un grupo no normalizado no funcionará, por el contrario existirán complicaciones al momento de desarrollar aplicaciones de alto nivel que impliquen modificaciones a la base de datos.

Las formas normales son un conjunto de restricciones para evitar anomalías al momento de ejecutar operaciones sobre las tuplas. Con el fin de lograr que una relación cumpla con alguna forma normal se efectúa un proceso de descomposición.

Las ventajas de descomposición son:

- ✓ Evitar anomalías en inserciones, modificaciones y eliminaciones.
- ✓ Mejora la independencia de datos.
- ✓ Mejor organización de los datos de acuerdo a sus dependencias funcionales, es decir, de acuerdo a sus relaciones lógicas.

La normalización se lleva a cabo bajo una serie de pasos, cada uno corresponde a una forma normal con ciertas propiedades. Conforme se avanza en la normalización, las relaciones tienen un formato más fuerte y estricto, es así que son menos vulnerables a las anomalías de actualización.

Formas normales.

Estas reglas obligan en la mayoría de los casos a dividir una tabla en 2 o más para solucionar los posibles conflictos.

Es necesario apuntar que una tabla puede cumplir con la primera forma normal y no con la segunda pero no viceversa, entonces se puede decir que son más restrictivas las formas conforme se avanza en su implementación.

Primera forma normal (1FN)

Una relación normalizada es una que sólo tiene valores elementales (o simples) en la intersección de cada renglón y columna. Así, una relación no tiene grupos repetitivos.

Para normalizar una relación que contiene un solo grupo repetitivo, se elimina el grupo y se forman dos nuevas relaciones. Una relación está en primera forma normal si no contiene grupos repetitivos. De manera estricta se puede definir como sigue:

Una entidad R está en primera forma normal (1FN) si los valores, para cada atributo $A \in R$, son atómicos. Esto implica, que los valores en el dominio no deberán ser listas o conjuntos de valores.

Segunda forma normal (2FN)

Una relación está en segunda forma normal si, está en 1FN y además cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria, es decir si se han eliminado las dependencias parciales. Se entiende por dependencia parcial cuando los atributos no llave dependen sólo de una parte de una llave compuesta.

Tercera forma normal (3FN)

Ocurre cuando una tabla está en 2FN y ningún atributo no clave depende transitivamente de las claves de la tabla. En otras palabras, no ocurre cuando algún atributo depende funcionalmente de atributos que no son clave.

Pasos para el Análisis Relacional*1.- Identificar entidades.*

- a) Examinar los sustantivos
- b) Nombrar a cada entidad
- c) ¿Existe información de interés para la organización acerca de la entidad?
- d) ¿Cada entidad es identificable de manera única? ¿Cuáles atributos sirven como UID (identificador único)?

2.- Descripción de entidades

Escribir la descripción de la entidad.

3.- Establecer las relaciones entre entidades

Cada dirección de una relación.

- a) Un nombre

b) Una opción

c) Un grado (cardinalidad)

Algunos puntos importantes a considerar en esta fase:

- Una entidad debe tener un UID, ó no es entidad.
- Todos los componentes de un UID deben ser marcados con #* ó # únicamente.
- Una entidad puede ser identificada de manera única a través de una relación.
- Usar una barra UID para indicar que la relación es parte del identificador único de la entidad.
- Una entidad puede ser identificada de manera única a través de múltiples relaciones.

De igual modo es importante considerar:

- *Diferencias entre atributos y entidad:* Si un atributo, tiene atributos propios, entonces es una entidad.
- *Opcionalidad de atributos:* Se utiliza un mapa de instancias para validar que sean correctas las marcas para atributos (opcionales y obligatorios).
- Entidad débil o subordinada: Entidad que no tiene los atributos para formar una llave primaria.
- Resolución de las relaciones M: M: Se debe agregar una entidad intersección con los atributos que identifican a cada una de las entidades.
- Resolución de una relación recursiva M: M : con una entidad intersección y 2 relaciones M: 1 hacia diferentes instancias de la entidad original.
- Análisis Jerárquico de datos: Una representación jerárquica de datos es un conjunto de relaciones M: 1. Los UID para un conjunto de entidades jerárquicas pueden ser propagados a través de relaciones múltiples.
- Modelado de relaciones complejas: Modelar una relación entre tres ó más entidades como una entidad intersección con relaciones obligatorias con esas entidades.
- Relaciones que modelan roles: las relaciones permiten que una sola instancia de entidad asuma múltiples roles.

3.3.4 Diseño de bases de datos

El diseño de la Base de Datos se lleva a cabo por medio de dos actividades:

1. Pasar el modelo Entidad-Relación a tablas
2. Refinar el diseño inicial para producir un diseño completo de la Base de Datos.

En el diseño se definen las tablas, índices, vistas y espacio de almacenamiento.

Pasos en el diseño inicial de la Base de Datos

1. Mapear las entidades para las tablas

Crear un mapa de instancias para la nueva tabla. Registrar únicamente el nombre de la tabla (Tabla 3.4)

NOMBRE DE LA TABLA

<u>COLUMNA</u>						
TIPO DE LLAVE						
NULOS/UNICOS						
EJEMPLOS						

Tabla 3.4 – Mapeo de las entidades

2. Mapear atributos para columnas y establecer los atributos obligatorios como No Nulos, únicos y dar datos de ejemplo (Tabla 3.5)

<u>COLUMNA</u>	<u>Columna 1</u>	<u>Columna 2</u>	<u>Columna n</u>
TIPO DE LLAVE						
NULOS/UNICOS	NN	NN		NN/U	NN	
EJEMPLOS						

Tabla 3.5 – Mapeo de los atributos

3. Mapear identificadores únicos a llaves primarias (Tabla 3.6)

<u>COLUMNA</u>	<u>Columna 1</u>	<u>Columna 2</u>	<u>Columna n</u>
TIPO DE LLAVE	PK					
NULOS/UNICOS	NN	NN		NN/U	NN	
EJEMPLOS						

Tabla 3.6 – Mapeo de identificadores

4.-Mapear relaciones foráneas (Tabla 3.7)

<u>COLUMNA</u>	<u>Columna 1</u>	<u>Columna 2</u>	<u>Columna n</u>
TIPO DE LLAVE	PK				PK,FK1	FK2
NULOS/UNICOS	NN	NN		NN/U	NN	
EJEMPLOS						

Tabla 3.7 – Mapeo de relaciones foráneas

Donde:

- N: Campo nulo
- NN: Campo no nulo

U: Campo único

PK: Primary Key (Llave primaria)

FK: Foreign Key (Llave foránea o extranjera)

Normalización en el Diseño

Una vez terminada esta fase es necesario validar cada columna en las tablas mediante las reglas de normalización, en caso de violarlas se deben reubicar.

Modelo Físico

Especifica cómo se almacenarán los datos, el espacio que será ocupado, métodos de acceso, etc. Es así que el diseño es la siguiente etapa después del análisis. En este proceso las entidades se convierten en tablas, los atributos en columnas, los identificadores únicos en llaves primarias, y las relaciones en llaves foráneas, las marcas de los atributos de la siguiente manera: # por PK, o por N y * por NN. En la siguiente figura se muestra el Modelo Top-Down de diseño de bases de datos⁴ que ejemplifica las fases del diseño (Figura 3.6)

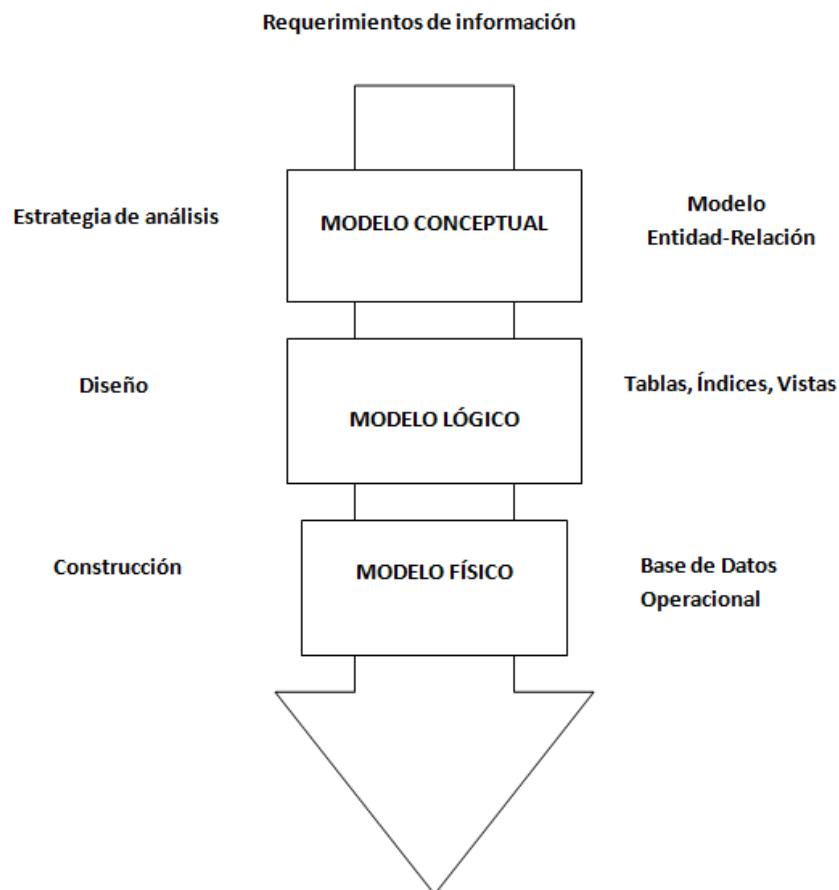


Figura 3.6 – Modelo Top-Down de diseño de bases de datos.

⁴ MODELO TOP-DOWN DE DISEÑO DE BASES DE DATOS – L.I Zamora Nunfio Lidia Lorelí, DGSCA/ UNAM 2009

3.4 Control de versiones.

El control de versiones permite la colaboración conjunta para la manipulación de información. Actualmente existen diferentes herramientas que permiten implementar esta tarea con gran facilidad.

Subversion (SVN) es un sistema de control de versiones de código abierto, con la que es posible revisar un historial de modificaciones así como recuperar versiones anteriores en un desarrollo. Existe la posibilidad de trabajar desde diferentes copias locales en distintas máquinas y trabajar de manera concurrente. A continuación se listan algunas de sus características:

- *Control de versiones para un sistema de archivos.*
Es la capacidad de controlar todas las carpetas y archivos de manera jerárquica
- Control de historial.
SVN permite agregar, eliminar, copiar y renombrar archivos y directorios actualizando el historial de modificaciones para evitar conflictos con antiguos elementos con el mismo nombre.
- *Modificaciones seguras.*
Las modificaciones se realizan de forma íntegra o no se realizan, lo que implica que sólo las actualizaciones se reflejan en el repositorio siempre y cuando sean correctas.
- Interoperabilidad.
SVN tiene la capacidad de operar desde línea de comandos, lo que implica poder ejecutar tareas a través de SSH o como complemento de un servidor HTTP, de esta forma adquiere ventajas como autenticación y privilegios.

Conceptos generales.

Repositorio.

El repositorio guarda la información como un sistema de archivos de manera jerárquica. Aquí todos los clientes realizan una conexión para hacer las operaciones de escritura, de esta manera todos los cambios hechos en una copia local serán visibles y se podrán recibir las modificaciones de otros usuarios por medio de la lectura.

La principal característica del control de versiones es su capacidad para mantener un historial de cada modificación a cada archivo y/o directorio. Existe también la facilidad de ver los cambios de forma cronológica y no sólo la última versión por lo que es posible visualizar información adicional como quién la modificó y qué parte se actualizó.

Copia de trabajo.

Es un sistema de archivos ordenados de forma jerárquica almacenados en el sistema local. Es un conjunto de archivos con los cuales se trabaja como colaborador de un proyecto, las modificaciones hechas no serán vistas por otros colaboradores y tampoco las modificaciones de otros usuarios serán visibles para el usuario local a menos que se indique explícitamente una publicación o una actualización respectivamente.

La forma de obtener una copia de trabajo se realiza a través de la operación *checkout*, por ejemplo:

```
$svn co file:///home/blue/svn/blue/trunk/ .
```

Donde "blue" es el proyecto en cuestión

Una vez realizada la operación de checkout, el control de versiones no tiene forma de actualizar el repositorio de forma automática, por lo que es necesario definirlo explícitamente, svn proporciona una operación para realizar dicho proceso y se identifica como *commit*.

```
$svn ci archivo_modificado -m "actualización de archivo n"
```

De esta forma los cambios se reflejan en el repositorio y la última versión será visible para los demás colaboradores.

La operación que debe ejecutar otro colaborador del proyecto para poder actualizar y trabajar sobre los últimos cambios será: *update*. De esta forma tendrá disponibles los cambios en su copia de trabajo, lo mismo ocurrirá para cualquier colaborador que actualice su copia.

```
$svn up
```

Adicionalmente existe la operación *status* que proporciona información sobre el estado actual del repositorio.

```
$svn status
```

URLs para Acceso a repositorios.

Existen diferentes métodos para acceder a los repositorios, desde el disco duro local hasta diferentes protocolos de comunicación, a continuación se listan los disponibles (Tabla 3.8):

Protocolo	Descripción
http://	Acceso a través de un servidor Apache
file:///	Accedo directo del disco local
https://	Trabaja de la misma forma que http:// pero con encriptación SSL
svn://	Acceso a través de un servidor svnserver
svn+ssh://	Igual a svn:// pero a través de un túnel SSH.

Tabla 3.8 – Métodos de acceso a repositorio

Ejemplos:

Acceso al repositorio desde el sistema de archivos local

```
svn co file:///home/blue/svn/blue/trunk/ ./
```

Desde el servidor Apache para la conexión con Eclipse

```
svn+ssh://login@132.248.59.180/home/blue/svn/blue
```

Revisiones.

Una revisión es una nueva actualización en el sistema de archivos del repositorio y se crea cada vez que se ejecuta la tarea para hacer permanentes los cambios (commit). Cada una de estas

actualizaciones tiene un identificador único (numérico) auto incrementable. Es así que la primera revisión comienza en cero cuando el directorio que contendrá al sistema de archivos está vacío.

Es así que el conjunto de todos estos elementos permiten controlar las versiones de todo un sistema de archivos lo que facilita el desarrollo y mantenimiento de un proyecto. Los colaboradores del desarrollo no tienen que ocuparse de lidiar con el control de concurrencia o con conflictos en cada una de las modificaciones que se realicen.

3.5 Buenas prácticas para el desarrollo de software.

En la actualidad la calidad de un producto es sinónimo de ahorro de recursos y costos. Cuando se comienza el desarrollo de un sistema siempre existe la necesidad de optimizar procesos, facilitar su mantenimiento, mejorar sus características y propiedades. Las buenas prácticas para el desarrollo consideran cubrir los objetivos y alcanzar la máxima funcionalidad de un sistema con la finalidad de tener un grado alto de cumplimiento de las especificaciones.

La implementación de un conjunto de buenas prácticas de desarrollo depende en gran parte, del compromiso de los colaboradores por lo que es necesario exponer las ventajas dentro de la organización.

Una vez que se ha decidido implementar un conjunto de estas buenas prácticas se deben de tomar las siguientes consideraciones:

- Comunicar al grupo de trabajo la implementación del modelo de calidad con la finalidad de hacer partícipes a todos los elementos. De igual importancia es expresar la toma de esta medida como parte de mejora constante y no verlo como un trámite sin sentido.
- Facilitar la información necesaria a la organización para su conocimiento.
- Establecer la ubicación de la información, desde formatos hasta documentos.
- Establecer el estándar de la documentación, abreviaturas, descripciones, etc.
- Definir el plan de trabajo donde se definen los objetivos.
- Definir los procesos que permitan establecer una serie de pasos a seguir para desarrollar u obtener un producto en la organización.

¿Qué modelo elegir?

De acuerdo con complejidad del desarrollo y sobre todo a los antecedentes y experiencia propia se puede elegir entre varios modelos para un desarrollo en particular.

En este caso se consideró a MoProSoft⁵ como el conjunto de buenas prácticas a seguir en una organización que se enfoca en el desarrollo de software, entre otras cosas por su facilidad y forma práctica de exponer las necesidades de cada proyecto.

⁵ MoProSoft (Modelo de Procesos para la Industrial del Software)

Es un conjunto integrado de prácticas de Administración e Ingeniería de Software que puede servir de referencia a las organizaciones que se dedican al desarrollo y mantenimiento de sistemas para lograr mejoras en su forma de trabajar y su desempeño.

<http://www.comunidadmoprosoft.org.mx/> ¿Qué es MoProSoft? Hanna Oktaba.

Implementación

De acuerdo al análisis de los requerimientos se implementaron fases acordes a cada etapa del proyecto bajo las recomendaciones de Moprosoft. Es necesario recalcar que sólo se hizo uso de las más representativas como guía para facilitar la producción y disminuir en lo posible el mantenimiento futuro. A continuación se describe el conjunto de recomendaciones que se utilizaron durante el desarrollo.

Plan de proyecto

Es el documento con la descripción de actividades para gestionar los proyectos y establece:

- Relación de recursos humanos: Grado de estudios y capacitación.
- Materiales y equipo necesario.
- Herramientas de software.
- Roles: Para cada uno de los integrantes del proyecto.
- Ciclos y actividades: Definen el proyecto en diferentes etapas con la finalidad de hacerlo escalable y dinámico.
- Tiempo y actividades: Se define el tiempo estimado para cada ciclo.
- Calendario: Define las fechas de cada uno de los ciclos.
- Plan de manejo de riesgos.
- Riesgos técnicos: Define el riesgo, el impacto que tendrá en el desarrollo, sus consecuencias y las acciones que se deben considerar para minimizar su impacto.

Plan de Desarrollo

Se describen las distintas fases del desarrollo y por las cuales deberá pasar cada uno de los ciclos descritos en el plan de proyecto, a continuación se describe el contenido de cada una.

Fase 1: Inicio

Se consideran los puntos del “Plan de Proyecto”, para el desarrollo del proyecto.

Fase 2: Requerimientos.

Se define la descripción del proyecto en base en base a los requerimientos de alto nivel establecidos por el cliente. Se definen los requerimientos específicos (casos de uso específicos). Se asignan los roles de acuerdo al trabajo que desempeñaran y por último se establece un calendario de actividades para cada requerimiento.

Fase 3: Análisis y Diseño

Se documenta el análisis y diseño. Los roles son asignados para cada actividad y se establece un calendario de actividades.

Fase 4: Construcción

Se desarrolla el proyecto en base al documento de Análisis y Diseño y se conforma el equipo de trabajo.

Fase 5: Integración y pruebas.

Se genera un calendario de actividades para las pruebas que serán documentadas en un “Reporte de validación y verificación”.

Fase 6: Cierre

Se entrega el proyecto totalmente integrado y funcionando.

Especificación de Requerimientos

Este documento pretende profundizar los requerimientos de alto nivel establecidos por el cliente; ya que éste será el que apruebe/desapruebe el avance del proyecto, es decir se establece qué es lo que hará el proyecto.

- Introducción: Descripción general del proyecto y su uso en el ámbito en el negocio del cliente.
 - Antecedentes: Describirá los factores que afectan al producto y sus requerimientos.
 - Propósito: Establece la finalidad del documento así como la audiencia a la que va dirigido.
 - Alcance: La descripción de todo aquello con lo que debe cumplir el sistema.
 - Definiciones, acrónimos y abreviaciones: Se definen todos los términos propios, acrónimos y abreviaciones requeridas para comprender el documento y el proyecto en general.
- Descripción general del sistema.
 - Perspectiva del producto: Relacionar el producto con algún otro.
 - Definición del sistema: Es un resumen de todas las funciones principales que realizará el software sin incluir detalles de cada una de ellas.
 - Características del usuario: Detalle las características generales de cada tipo de usuario incluyendo grados de estudio, experiencia y nivel de aptitud técnica.
 - Restricciones generales: Incluye una descripción general de cualquier aspecto que limitará las opciones del desarrollador.
 - Suposiciones y dependencias: Es un listado de los factores que afectan los requerimientos, establecidas. Es importante considerar que estos factores no son una restricción para el diseño del software.
 - Aplazamiento de requisitos.: Listado de cada uno de los requerimientos que se desarrollaran a futuro y que no forman parte del desarrollo actual.
- Requerimientos específicos.

Deberá contener todos los requerimientos de software con un nivel de detalle suficiente para permitir al diseñador desarrollar el sistema que satisfaga tales requerimientos así como para los encargados en pruebas para comprobar que se satisfagan.

Cada requerimiento establecido en esta sección debe ser percibido externamente por un usuario, operador u otro sistema externo. Los requerimientos deben incluir una descripción de cada entrada y salida del sistema así como toda función realizada por el este.
- Requerimientos de interfaces externas.

Es una descripción detallada de todas las entradas y salidas del sistema.

 - Interfaces de usuario: Es una lista de todos los posibles prototipos que el usuario podrá emplear dentro del sistema.
 - Interfaces: De hardware, de software, de comunicación.

- **Requerimientos funcionales**
Los casos de uso deberán:
 - Describir una tarea del negocio que sirva para una de sus metas.
 - Tener un nivel de detalle adecuado.
 - Sencillo para facilitar la tarea al desarrollador.
 - Permitir al analista centrarse en las necesidades del usuario, qué espera éste lograr con el sistema, evitando que los especialistas impongan el funcionamiento de acuerdo a sus criterios de desarrollo.

- Para su desarrollo se considerará:
 - Una breve descripción del caso de uso.
 - Acciones previamente realizadas por el actor para que la tarea pueda llevarse a cabo.
 - Acciones previamente realizadas por el sistema para que la tarea pueda llevarse a cabo.
 - Excepciones: Considerando las acciones del actor que desencadenan una acción del sistema.

- **Requerimientos no funcionales**
 - **Requerimientos de rendimiento:** Se especifican los requerimientos numéricos estáticos y dinámicos puestos en el producto de software.
 - **Seguridad:** Seguridad con la que deberá contar el producto.
 - **Fiabilidad:** Fiabilidad en que los datos sean almacenados.
 - **Disponibilidad:** Condiciones sobre disponibilidad del sistema, si afecta o no el servicio que otorga el producto.
 - **Mantenimiento:** Elementos que se requieren para dar mantenimiento al producto.
 - **Portabilidad:** Especifica si el producto podrá adaptarse a otros productos ya desarrollados o que estén por desarrollarse.
 - **Otros requerimientos:** Algún caso especial que no se encuentra en las especificaciones de los otros puntos.

Análisis y diseño

Este documento permite a los desarrolladores saber cómo funciona el producto, lo cual les permitirá enfocarse únicamente en la construcción; de esta forma se puede ver al Análisis y Diseño como los planos del proyecto.

Ciclos

Para el estándar de diseño se considerará:

- **Número de ciclo** al que pertenece el desarrollo del componente.
- **Componente:** Identificar del elemento.
- **Descripción:** Breve explicación para el componente actual.
- **Módulo:** Modulo al que pertenece el componente a construir, de acuerdo a la arquitectura establecida para desarrollar el producto, por ejemplo MVC (Modelo Vista Controlador).
- **Propiedades:** Atributos del componente.

Interfaces de los componentes

Es una descripción detallada de cada una de las interfaces (conjunto de componentes), cada descripción deberá contar con:

- **Interfaz:** Identificador de la interfaz
- **Acción:** Descripción de los procesos ejecutados por la interfaz.
- **Parámetros:** Elementos que conforman la interfaz.

- Id Mensaje de error: Identificador para el mensaje de error.
- Mensaje de error: Mensajes de error que pueden ser lanzados por los componentes en caso de que algún parámetro no pase la validación.

Diagramas de clases

Diagrama de clase o algún otro documento que ayude a la construcción del producto.

Diagrama de base de datos

Diagrama de base de datos junto con el reporte generado por la herramienta de creación, el cual contendrá especificaciones sobre cada relación y atributo.

Arquitectura y diagrama de paquetes principales de sistema.

Arquitectura: Herramientas empleadas para el desarrollo del producto. Deberá considerar:

- Herramienta: Producto que permita una organización en la programación
- Tipo de arquitectura: MVC
- Observaciones

Diagrama de paquetes

De acuerdo con el tipo de arquitectura a emplear, se hace la descripción de cada uno de los paquetes que la conforman.

Diagrama

Diagrama que muestre la interacción entre los paquetes de la arquitectura.

Diagrama de componentes por paquete.

Muestra en forma de paquetes cada uno de los componentes que confirman el sistema y las relaciones que existen entre ellos.

Construcción

En esta fase se dividen los ciclos y actividades de desarrollo de acuerdo a los roles asignados. Básicamente el enfoque va dirigido a la codificación de las soluciones propuestas.

Integración y pruebas

Se proponen las pruebas necesarias para tener un alto grado de cumplimiento de los requerimientos. De acuerdo al enfoque se consideran regularmente los apartados: “Pruebas de integración” y “Pruebas de Sistema”, cuyo registro deberá considerar entre otras características la fecha de realización, la descripción del proceso, el evaluador y resultado encontrado.

Cierre

Es la fase final donde se entrega el producto con todas las características y especificaciones solicitadas.

Capítulo IV

Análisis

En este apartado se desarrollarán las bases que se deberán considerar en la fase de diseño, se buscarán las necesidades principales y la forma de solucionarlas. Como primer paso se analizarán los perfiles de los usuarios que utilizarán el sistema, se considerarán de igual forma los requerimientos específicos tomando en cuenta desde información personal hasta ambiente de trabajo.

4.1 Usuarios

Administrador

Las actividades del administrador serán:

- Registrar a los usuarios que recibirán los contenidos.
- Registrar dispositivos móviles.
- Generar entregas y contenidos para ser enviados (esta última tarea será automatizada por el sistema).

Un administrador tiene la posibilidad de:

- Crear los contenidos para ser enviados a los usuarios.
- Modificar la información que haya registrado.
- Revisar los reportes generados por el sistema.

Entre sus características se encuentran (Tabla 4.1):

<u>Tipo de usuario</u>	<u>Formación</u>	<u>Habilidades</u>	<u>Actividades</u>
Administrador	Informática	Conocimientos de Windows a nivel usuario	Administración de usuarios, mensajes y dispositivos.

Tabla 4.1 – Características del administrador

Receptor

Los usuarios podrán:

- Consultar la información que se les envíe a su celular vía Bluetooth.

Tendrá la posibilidad de:

- Registrar uno o más dispositivos en el sistema.

Entre sus características se encuentran (Tabla 4.2):

<u>Tipo de usuario</u>	<u>Formación</u>	<u>Habilidades</u>	<u>Actividades</u>
Usuario	-	Manejo de multimedia, internet y celular	Consultar información

Tabla 4.2 – Características del receptor

4.2 Seguridad

En la actualidad los sistemas con un funcionamiento similar se ocupan sólo de operaciones comunes como búsqueda de dispositivos y transmisión de información sin tomar en cuenta la seguridad.

Esta tecnología es una de las mejores al momento de trabajar con redes inalámbricas de alcance corto, sin embargo la parte de seguridad no se profundiza en su totalidad, se deben de tomar en cuenta los siguientes puntos al momento de implementar seguridad en nuestro proyecto:

- Un dispositivo no detecta la presencia de otro dispositivo.
- Un dispositivo no detecta cuando otro ejecuta la tarea de descubrimiento de otros dispositivos cercanos.
- Un dispositivo no detecta la dirección del dispositivo que está ejecutando un descubrimiento.

La seguridad es débil al considerar que por ejemplo una persona no puede estar convencida de la fuente ni el contenido de la información, por cuestiones como:

- Suplantación: Otro equipo se hace pasar por la fuente de transmisión.
- Modificación: Ataque contra la integridad de la información

Con estas vulnerabilidades presentes, es necesario considerar un sistema de registro de usuarios con la finalidad de evitar activamente cualquier de los puntos antes mencionados.

Es así que el sistema al contar con un registro tiene acceso a información que el usuario y sólo él conoce, por lo que al interactuar con el sistema tendrá la certeza de que el sistema es quien dice ser y no algún otro.

4.3 Información.

Especificaciones generales.

Mensajes

La creación de los mensajes será tarea única de los administradores. Los mensajes contendrán información en general, fechas de eventos, trámites, etc. Deberán tener un contenido relevante de forma que el usuario al recibirlo en su dispositivo móvil encuentre útil la información, su longitud no deberá rebasar los 225 caracteres. Su diseño (dimensiones y tamaño) deberá estar optimizado debido a que el tiempo de recepción de un mensaje está directamente ligado al tamaño del archivo, entre mayor sea su tamaño, su transmisión hacia el dispositivo desde el equipo central tomará mayor tiempo.

Dispositivos

Los dispositivos deberán registrarse en la base de datos con información necesaria (Dirección Mac y Canal) para la correcta transmisión de los contenidos, así como para realizar análisis periódicos. Los dispositivos que no cuenten con el servicio para el correcto envío de contenidos, no deberán afectar el rendimiento del sistema por lo que se definirá un trato especial para este caso.

Funcionamiento

El sistema contendrá en la base de datos la información que el usuario proporciona al momento de su suscripción al servicio, entre los datos solicitados se tendrá una cadena de texto haciendo las veces de un identificador que sólo el mismo usuario tendrá presente; el sistema cuando deba hacer una entrega tomará en cuenta su información personal y generará un contenido único que será consultado por el usuario y será reconocido por él.

Las bases que se consideran para la seguridad del sistema son:

- El identificador es conocido por el usuario y sólo él.
- Algún sistema que intente suplantar la fuente de la información no considera la información que el usuario proporciona en su registro.
- La información no podrá ser modificada dado que desde un principio la fuente no es la indicada por el sistema.

Envío de la entrega.

En la etapa de búsqueda de dispositivos, el sistema encontrará al dispositivo suscrito en el sistema, en el momento de detectarlo le enviará la entrega o entregas pendientes.

Generación del contenido.

Con la información recopilada del usuario identificada por un rol de usuario, es posible generar el contenido de la entrega. Las entregas estarán plasmadas en imágenes con la información del mensaje y el identificador del usuario debido a que en el caso de suplantación del sistema o modificación de la información, es muy sencillo generar un archivo plano con algunas cadenas de texto.

Prioridad de entregas.

El sistema hará un descubrimiento de dispositivos sin ninguna consideración especial sobre un dispositivo en particular, es así que las entregas se enviarán de inmediato cuando se haya detectado un dispositivo registrado, debido a esto es necesario considerar un control para no formar un ciclo que impida el avance de la secuencia de envíos cuando se realice cancelación de la entrega por parte del usuario, en este caso se contabilizarán estos eventos y si el intento de entrega no es satisfactorio en un par de ocasiones, la entrega se inhabilita para no volver a ser enviada.

Reportes

La generación de los reportes se hará de manera dinámica a partir de los datos contenidos en la base de datos. Los diferentes reportes estarán relacionados con: usuarios, mensajes y entregas.

- Reporte de usuarios:
Mostrará el número total de usuarios agrupados de acuerdo al rol que tengan asignado.
- Reporte de mensajes:
Se tendrá un listado con el número de mensajes creados en un intervalo de tiempo, intervalo que será abierto para proporcionar flexibilidad en el análisis.
- Reporte de entregas:

Mostrará el número total de envíos satisfactorios así como las entregas pendientes y los mensajes cancelados por el usuario, de la misma forma que se hace con el reporte de mensajes, se manejará un intervalo personalizable de tiempo para facilitar la consulta de la información.

Restricciones

Generales:

Los contenidos serán mensajes diversos (fechas de trámites, eventos e información en general) plasmados en una imagen. En un futuro se puede considerar el envío de contenidos multimedia (audio, video, etc.)

Técnicas

Los mensajes sólo podrán ser enviados a usuarios registrados previamente.

El hecho de detectar un dispositivo en la fase de descubrimiento de equipos cercanos, no garantiza el correcto envío y recepción.

4.4 Sistema automatizado.

De acuerdo al análisis anterior, el sistema contará con una administración que facilite el registro de usuarios, dispositivos y mensajes de forma que estos últimos puedan ser enviados de forma inalámbrica vía Bluetooth a los dispositivos móviles de los usuarios. El registro como ya se explicó tendrá la tarea de asegurar tanto la fuente como el contenido de la información. De acuerdo al modelo de trabajo que se plantea y las herramientas previstas para trabajar, es posible considerar el desarrollo de un subsistema con algunas de las características principales del primer sistema (registro de usuarios y dispositivos) pero con algunas facilidades y modificaciones de forma que pueda enfocarse a otros objetivos, sin perder la idea principal de este trabajo.

Consideraciones

La principal idea de esta modificación es automatizar todo el proceso de un envío común, es decir, desde el registro de usuarios y dispositivos así como la generación de entregas y envíos de estas. Todo esto con la finalidad de que cualquier persona que cuente con un dispositivo móvil pueda hacer uso del sistema sin la necesidad de pertenecer a un grupo en específico.

Requerimientos específicos.

- El registro del usuario de un dispositivo móvil no debe ser una fase obligada para poder ser suscriptor del servicio.
- No hay necesidad de generar un contenido personalizado debido a que existe la primicia de que el mismo usuario tiene la certeza de la fuente.
- Debe existir un control de entregas, considerando que al no ser suscriptor del servicio, el usuario encuentre molesto el sistema de entregas.

Alcance

Para poner en contexto la idea principal de este subsistema, se considera el siguiente caso:

Regularmente dentro de las escuelas y facultades de la Universidad existen exposiciones, congresos, etc. En algunos casos la información proporcionada por el grupo organizador no es tan directamente captada por la gente que colabora en estas reuniones, en ocasiones por que se considera información conocida o en otras ocasiones por mantener un nivel práctico. Es aquí cuando un sistema de transmisión de información es útil para lograr mantener a las personas interesadas en una continua retroalimentación, aquí tres ejemplos prácticos:

1. Existe una exposición en un centro cultural, la gente que asiste la encuentra interesante y busca una página de internet con contenidos de la exposición, otra con un tema en específico de alguna obra o necesita los horarios y días para recomendarla; el sistema tendrá contenidos con información que previamente se analizó y se encontró importante para los visitantes, su registro como usuario y de su dispositivo debe ser automatizado con la finalidad de que considere el sistema como parte del servicio y no como un complemento que le quite tiempo que bien puede pasar en la exposición.

2. Un congreso tiene su información en carteles pegados a través de todo el conjunto en todo el campus y en ocasiones la fechas y horarios de los talleres para desapercibidos o alguien puede no considerar sus tiempo y la hora de inicio se pasa, es entonces que un aviso con el nombre del taller, una breve descripción y recordatorio de horario pueden hacer la diferencia entre llegar a la hora o no entrar por un descuido de la hora.

El sistema que además cuenta con un control de horarios puedes ser desactivado 5 minutos después del comienzo de la reunión, por lo que no volverá a ser enviado debido a que la información ahora es irrelevante y sólo causaría molestia entre los usuarios que ya están en el taller o que no desean asistir.

3. Dentro del congreso existen expositores que tienen promociones para suscripciones de revistas, avisos para opciones de trabajo, próximas fechas y lugares donde se presentarán, el sistema puede tomar está información y enviarla a toda la comunidad sin considerar ningún requisito, es aquí cuando el envío se hacer masivo y de nuevo sin la necesidad de un registro previo.

4.5 Requerimientos del Hardware*Equipo de trabajo: El Servidor*

Poner en funcionamiento y administrar un servidor web requiere de especial atención debido a que debe cumplir con dos aspectos muy importantes: la seguridad y la disponibilidad. El primer punto se refiere al envío de la información a quien corresponde, y el segundo a que el servicio responda siempre que se le solicite. Para cumplir con estos aspectos se debe procurar tener en el servidor sólo el software y los procesos necesarios, es decir, que el servidor se utilice para lo que en realidad ha sido puesto en marcha. Por ejemplo, no es necesario tener el servicio de ftp, si no se realizan conexiones mediante este protocolo, ó tener software de paquetería de oficina si nadie lo utiliza en el servidor.

Para delimitar qué es necesario y qué no lo es, se parte del análisis del tipo de contenido que se mostrará al usuario final, así como el alcance del proyecto; lo que también permitirá hacer un estimado de la carga de procesamiento que tendrá el servidor en la fase de producción.

Para el desarrollo de este proyecto es necesario considerar un equipo con un procesador equivalente a Pentium IV o superior, con 512 MB en memoria RAM; con estos elementos se asegura que el sistema tendrá un rendimiento aceptable.

Medio de Transmisión: La Antena

Por lo definido tanto en los objetivos del capítulo uno, como en la investigación de las tecnologías inalámbricas del capítulo dos, se concluyó que se necesita una red inalámbrica de área personal. Un par de argumentos a favor de esta tecnología son:

- No representaría costo alguno para los usuarios.
- Viene incorporada en la mayoría de los dispositivos móviles.

Es así que el servicio no generará ningún gasto y además tendrá un amplio margen de usuarios. Una vez definida la tecnología Bluetooth como la ideal para el proyecto, se encontró que los dispositivos convencionales tienen un alcance muy limitado, por lo que se buscó un dispositivo comercial que proporcionara un mayor alcance. De esta forma, se encontró un dispositivo con las especificaciones suficientes para implementarlo en el desarrollo. A continuación se muestran las características de la antena:

- Compatible con la versión 2.0 de Bluetooth.
- Conector USB estándar.
- Trabaja con Sistemas Operativos Linux, Windows y Mac OS X.
- Rango de un kilómetro para la antena interna.
- Rango de alcance puede extenderse hasta 30 kilómetros.
- No necesita una fuente de alimentación externa.
- Estuche de aluminio para reducir la interferencia e incrementar la sensibilidad.

La antena XR2 AIRcable está equipada con un sensible y poderoso transmisor Bluetooth, puede lograr un alcance no paralelo de hasta 30 km. El alcance extendido es de hasta 2 km para la antena omni-direccional y de hasta 10 km para la antena direccional. Dada su alta sensibilidad permite encontrar dispositivos Bluetooth más débiles como los teléfonos celulares y los auriculares en cientos de metros. Todos los otros equipos sólo son capaces de obtener un rango de 10 metros o menos (en condiciones no ideales), ningún otro dispositivo Bluetooth tiene tanto alcance, y es gracias a esta capacidad que es posible conectarlo a cualquier equipo mediante una conexión Bluetooth y manejar virtualmente cualquier perfil Bluetooth disponible. Las aplicaciones que se pueden implementar mediante el dispositivo incluyen transmisión de datos estándar, comunicación de auriculares estéreo, entre otros.

Especificaciones técnicas⁶

Operating systems	Linux Mac OS X Windows
Industrial strength aluminum case	small size: 70mm x 57mm x 15mm case mounting flanges magnetic bottom Industrial operating temperature range -40°C ...85°C
Power supply	Powered through USB, 5V, max 200mA
Bluetooth with external antenna connector	omni-directional antenna up to 2km with 9dBi directional antenna up to 10km with 18dBi professional installation up to 30km with 24dBi parabol external cable available
Bluetooth profiles	depends on the bluetooth software & drivers on the computer to which it is connected
Certifications	Bluetooth certified (BQB) Bluetooth 2.0 compatible with 802.11b tolerance (AFH) FCC authorization 15C - SQC ACC1600 CE certification
Radio	raw output power: 19.5dBm input sensitivity: -92dBm omni directional antennas: up to 9dBi directional: up to 18dBi professional installation up to 24dBi antenna connector: RP-SMA

Tabla 4.3 Especificaciones técnicas de la antena

Consideraciones.

Dada la naturaleza de las tecnologías inalámbricas es necesario tomar en cuenta aspectos que afectan su rendimiento, cuestiones como: ruido, interferencia y bloqueos, además hay que considerar la movilidad de los usuarios y el ambiente de propagación. Es por estos factores que se establecen límites asociados al rango y velocidad de transmisión en comparación con las comunicaciones alámbricas donde la distorsión y atenuación de la señal es baja.

Respecto a las pérdidas por trayectoria se puede considerar:

- Reducción de la potencia de la señal en el receptor en relación con la potencia transmitida.
- Las pérdidas son proporcionales a la distancia
- Degradación de la señal a causa de obstáculos como paredes, pisos, etc.

Además de esto, existe un conjunto de fenómenos llamados “Factores de propagación” (Figura 4.1) que pueden dañar la señal y distorsionarla, fenómenos como:

- Reflexión: Aparece cuando la señal topa con un área que es capaz de reflejar la energía.

⁶ <http://www.aircable.net/host-xr2.html>

- Difracción (Shadow fading): Cuando la señal cambia su trayectoria al viajar debido a los bordes de objetos intermedios.
- Dispersión: Aparece cuando la señal topa con un área con irregularidades cuyo valor es una fracción de su longitud de onda.

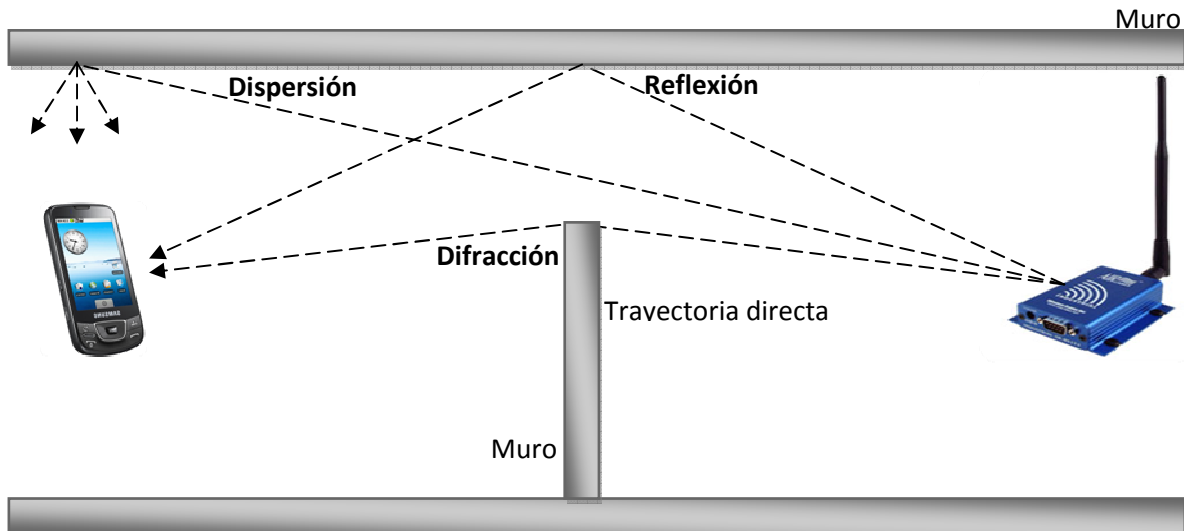


Figura 4.1 Factores de propagación

De igual forma se deben tomar en cuenta las fluctuaciones en la recepción de señales, las más comunes se describen a continuación:

- Path loss (Pérdida por trayectoria): Define la disminución de la señal tomando en cuenta la distancia existente entre transmisor y receptor.
- Shadow fading (Distorsiones lentas): Determina la atenuación de la señal considerando las características del terreno así como obstáculos intermedios como por ejemplo edificios.
- Raleigh fading (Distorsiones rápidas): Caracteriza a las fluctuaciones rápidas.

Así, con todos estos factores afectando directamente el rendimiento de los sistemas inalámbricos es necesario considerar para el análisis un conjunto de estructuras propuestas de forma estadística para predecir la intensidad promedio de la señal a una distancia arbitraria del transmisor o su variabilidad en una ubicación en particular.

A continuación se realiza un análisis considerando sólo la trayectoria directa, la siguiente formula se utiliza para calcular la potencia de la señal recibida en ambientes internos o exteriores:

$$P_r [dBm] = P_t [dBm] + G_t [dBm] + 20 \log \left(\frac{\lambda}{4\pi} \right) + 10n \log \left(\frac{1}{d} \right)$$

- P_r Potencia de la señal recibida
- P_t Potencia transmitida
- G_t Ganancia de la antena de transmisión
- λ longitud de onda
- d distancia de la separación de las antenas
- n exponente de pérdida de trayectoria

De la cual los dos últimos términos expresan la cantidad de la pérdida de trayectoria (Path loss ó PL), para el caso de Bluetooth se considera una λ de 0,122 m para una frecuencia de portadora de 2,45 GHz, obteniendo la siguiente ecuación:

$$PL = 40 + 10n \log(d)$$

Despejando d de la ecuación anterior y teniendo en cuenta que el valor mínimo aceptable por el receptor coincide con el valor de pérdida de trayectoria se puede calcular el rango útil para los radios Bluetooth.

La tabla 4.4 presenta algunos cálculos utilizando la ecuación de pérdida de la trayectoria asumiendo que: la potencia del transmisor es de 19.5 dBm, la sensibilidad del receptor es de -70 dBm (estándar) ó -80 dBm (extendido) y la ganancia de las antenas tanto del receptor como el transmisor es 0 dBm.

Tipo de saturación	n	Potencia Tx[dBm]	Sensibilidad Rx[dBm]	Rango
Espacio libre	2.0	+19.5	-70	316
			-80	1000
Moderada	3.0	+19.5	-70	316
			-80	1000
<i>Tabla 4.4 Cálculos de la ecuación de perdida por trayectoria</i>				

Tabla 4.4 Cálculos de la ecuación de perdida por trayectoria

4.6 Requerimientos de Software

Framework – Symfony

Symfony es un framework para desarrollo de aplicaciones que implementa de forma automática muchos de los métodos necesarios para la creación de proyectos. Agrega estructura al código, propone al desarrollador una secuencia de trabajo, ayuda a la fácil comprensión del código de forma que lo hace más fácil de mantener.

Symfony esta diseñado para optimizar el desarrollo de aplicaciones web; está encaminado a la construcción de aplicaciones robustas para un contexto empresarial. Permite un control total de la configuración, desde la estructura de los directorios a las bibliotecas importadas.

Symfony está desarrollado y utiliza el lenguaje de programación PHP. Para poder utilizarlo, la lista de requerimientos es mínima, basta con tener un servidor web con PHP instalado y una base de datos. Symfony permite tener estructurada la aplicación mediante la arquitectura MVC y a partir del diseño de la base de datos, se generan automáticamente módulos, clases y métodos, lo que representa un avance importante para iniciar del desarrollo.

Además de esto, Symfony cuenta con una comunidad de usuarios y desarrolladores, que contribuyen con módulos para una amplia variedad de funcionalidades que complementan al framework.

Es importante mencionar que Symfony es completamente libre y publicado bajo la licencia MIT.

Características de Symfony

El framework es fácil de instalar y configurar en múltiples plataformas, es independiente del sistema manejador de bases de datos, compila las mejores prácticas de programación y diseño, además es fácil de extender, permitiendo la integración de bibliotecas externas.

Características de automatización de proyectos web:

- La construcción de capas permite la traducción de datos e interfaces, así como la localización del contenido.
- La presentación usa plantillas y diseños que pueden ser construidos sin tener conocimientos del framework. Reduce el código de presentación mediante el encapsulamiento en simples funciones.
- Soporte para la validación de formularios, para asegurar la integridad de datos en la base, y una mejor experiencia de usuario.
- Métodos de autenticación y credenciales para facilitar la creación de secciones restringidas y administración de la seguridad del usuario.
- Herramientas para entorno de desarrollo.
- Proporciona un panel de depuración que acelera el mantenimiento gracias a que muestra toda la información que los desarrolladores necesitan en la página en que estén trabajando.
- Es posible hacer cambios en la configuración al tiempo de pruebas.
- La interfaz desde línea de comandos automatiza el desarrollo entre dos servidores.

Arquitectura MVC

La arquitectura MVC consiste en tres niveles:

- El Modelo: representa la información dentro la cual operan las aplicaciones.
- La Vista: despliega el modelo en la página web, para interacción con el usuario.
- El Controlador: responde a las acciones de usuario, e invoca cambios en el modelo ó la vista según sea necesario.

Una vez identificadas las capas de aplicación se definen cada uno de los paquetes necesarios para el desarrollo del sistema. A continuación se listan cada uno de ellos:

Servidor Web Apache

La arquitectura del servidor Apache se caracteriza por ser modular. El servidor consta de un sistema base y diversos módulos que aportan una gran diversidad de funciones que se consideran básicas para un servidor web. Algunos de estos módulos son:

- `mod_rewrite`: reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como PHP en páginas estáticas HTML para así engañar a los navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).

- `mod_deflate` : Compresión transparente con el algoritmo deflate del contenido enviado al cliente.

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- `mod_perl` : Páginas dinámicas en Perl.
- `mod_php`:- Páginas dinámicas en PHP.
- `mod_security`: Filtrado a nivel de aplicación, para seguridad.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas a internet. Es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP, Perl, Python y Ruby.

Este servidor web es redistribuido como parte de varios paquetes propietarios de software. Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en forma segura y confiable. Un ejemplo se puede observar al compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web comúnmente utilizan una versión local de Apache en orden de pre visualizar y probar código mientras éste es desarrollado.

Imagemagick

ImageMagick es un software Libre, su licencia es compatible con la licencia GPL. Funciona en la mayoría de los sistemas operativos para la edición de imágenes. Las imágenes se pueden recortar, girar, cambiarlas de color o incluso combinarlas con otras. Además se pueden aplicar varios efectos y agregar a las imágenes texto, líneas, polígonos, elipses y curvas de Beizer.

La mayor parte de la funcionalidad de ImageMagick puede ser aprovechada en modo de comando; sin embargo, es más común que se use desde programas escritos en los siguientes lenguajes de programación: C, C++, Java, Lisp, Pascal, Perl, PHP, Python, Ruby y Tcl/Tk. Para cada uno de ellos, ImageMagick posee una interfaz la cual hace posible modificar o crear imágenes de forma automática y dinámica.

ImageMagick soporta una gran variedad de formatos de imagen (más de 100) incluyendo formatos como GIF, JPEG, JPEG-2000, PNG, PDF, PhotoCD, TIFF y DPX.

Algunas de las acciones que Imagemagick puede ejecutar:

- Conversión entre formatos; convertir una imagen desde un formato a otro (ej. PNG a JPEG).
- Transformaciones: cambiar de tamaño, girar, recortar y girar una imagen.
- Transparencias: determinar partes de una imagen como invisible.
- Dibujar: Agregar formas, figuras y texto a una imagen.
- Decorar: Agregar un marco o borde a una imagen.
- Efectos especiales: difuminar, esmerilar o trazar una imagen.
- Texto y comentarios: Insertar una descripción o un texto artístico a una imagen.
- Identificación de imagen: Describe los atributos de formato de una imagen.
- Animación: Crea un animación GIF a partir de un grupo de imágenes.
- Composiciones: Sobrepone una imagen sobre otra.

Cómo crear una imagen animada GIF con Imagemagick

Coloca las imágenes de la secuencia un directorio. Las imágenes deben estar numeradas en el orden que se desea que se muestren. Mediante la terminal accede al directorio y ejecuta el siguiente comando:

```
convert -delay 100 -loop 5 1.jpg 2.jpg 3.jpg ... imagen.gif
```

Donde `-delay` es el tiempo en milisegundos entre imágenes, `-loop` las veces que se repetirá la secuencia, `1.jpg 2.jpg 3.jpg ...` hace referencia a todas las imágenes de la secuencia, e `imagen.gif` define el nombre de nuestra imagen animada GIF, recién creada.

Cómo agregar marcas de agua con Imagemagick

Para generar la marca de agua, ejecutar el siguiente comando:

```
convert -size 150x15 -gravity Center -background none label:"texto ejemplo" water_mark.png
```

Donde `-size` define el tamaño en pixeles, `-gravity` la orientación del elemento, `-background` una imagen de fondo, en este caso se pone ninguna, `label` para especificar la etiqueta de la marca de agua y por ultimo el nombre.

Para agregar la marca de agua a una imagen:

```
composite -gravity South -geometry +0+10 -dissolve 60 -quality 90 water_mark.png imagen.jpg imagen_marcada.png
```

Donde `-gravity` define la orientación del elemento, `-geometry` para definir la ubicación en pixeles a partir de la orientación, `-dissolve` para disolver las imágenes en un porcentaje, `-quality` para definir el nivel de compresión, seguido de esto se especifican los nombres de la marca de agua, la imagen y por ultimo el nombre de la imagen con marca de agua.

PostgreSQL

PostgreSQL tiene la mayoría de las características presentes en grandes SDBD, como transacciones, subselects, disparadores, vistas, claves foráneas integridad referencial, y un sofisticado bloqueo. Pero además posee algunas características que ellos no tienen, como tipos definidos por el usuario, herencia, normas, y control de concurrencia multi-versión para reducir la contención de bloqueo. La siguiente es una breve lista de algunas de esas características:

Licencia

PostgreSQL es distribuido bajo la licencia BSD clásica. Básicamente, esto permite a los usuarios realizar cualquier tipo de modificación al código fuente, incluyendo la reventa de archivos binarios. La única restricción es que PostgreSQL no es legalmente responsable de los problemas que usted tenga con el software. También existe el requisito de que este copyright aparezca en todas las copias del software.

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays. PostgreSQL es altamente extensible; soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL

PostgreSQL soporta la especificación ANSI-SQL: 2008. Tiene soporte completo para subconsultas y transacción serializable a nivel de aislamiento.

API Flexible y Lenguajes Procedurales

PostgreSQL tiene soporte para trabajar en conjunto con más de una docena de idiomas de programación, como Java, Perl, Python, Ruby, Tcl, C / C + +, y su propio PL/pgSQL, que es similar a la de Oracle PL/SQL. PostgreSQL posee su biblioteca de funciones estándar (cientos de funciones integradas que van desde matemáticas básicas y las operaciones de cadena a la criptografía y la compatibilidad de Oracle). Del mismo modo, PostgreSQL incluye un framework que permite a los desarrolladores definir y crear sus propios tipos de datos, junto con funciones de apoyo y los operadores que definen su comportamiento.

MVCC Control de Concurrencia Multi-Versión

Es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Algunos SMDB con capacidades SQL, tal como MySQL o Access, las peticiones de lectura tienen que esperar para acceder a información de la base de datos. La espera está provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Cliente/Servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

PHP

Es un lenguaje de programación de propósito especial, utilizado mayormente en la generación de páginas dinámicas en internet. Cuenta con las siguientes características:

PHP es interpretado

Esto ayuda a que el desarrollo con PHP sea rápido y dinámico, puesto que no hay que compilar el código cada vez que se haga algún cambio. Cuando normalmente se habla de PHP, se hace referencia al conjunto de reglas de sintaxis del lenguaje, los diferentes módulos que incluye y por supuesto, al intérprete o motor, es decir, a la distribución completa.

El motor de PHP se llama Zend. Es el encargado de interpretar el código fuente de los scripts PHP, esto es, realiza un parseo (recorre el código identificando elementos) el código y ejecuta sus instrucciones.

PHP está enfocado a Internet

PHP no nació como un lenguaje de propósito general para programar cualquier cosa, sino con un claro objetivo: Internet. Provee una sintaxis sencilla, combinado entre las etiquetas de HTML, y ofrece una gran facilidad para generar aplicaciones web con contenidos dinámicos.

Las aplicaciones web escritas en PHP constan de un conjunto de scripts que interactúan con las fuentes que les proveen el contenido (bases de datos, archivos en disco, archivos XML, etc.), procesan estos contenidos y generan una salida HTML que es la que se envía al navegador. Asimismo, los datos que el usuario envía al sitio web, pueden ser procesados y almacenados en estas fuentes de datos, proporcionando dinamismo en ambos sentidos: usuario y aplicación web. Ejemplos de este dinamismo de la aplicación en ambos sentidos son foros, contadores, carritos de compras, etc.

Aunque es cierto que la principal utilidad de PHP es generar HTML dinámico, el lenguaje es también útil para generar otros tipos de salida, entre los que destacan principalmente archivos de texto a disco (generalmente usados como caché), archivos XML y también imágenes PNG y JPG.

PHP es extensible por medio de módulos

PHP ofrece de por sí unas enormes posibilidades que se tienen disponibles por defecto. No obstante, si se requiere alguna otra funcionalidad, es posible cargar el módulo que ofrece en el script en el que se necesite; o bien dejarlo cargado en PHP por defecto si se va a utilizar frecuentemente.

PHP tiene además una API bien documentada, que permite, en casos extremos, crear módulos de usuario.

Programación Orientada a Objetos (POO)

PHP permite utilizar la POO para hacer código estructurado y fácil de implementar, que facilita el desarrollo de aplicaciones web entre varios desarrolladores, pudiendo cada uno de ellos tomar

una parte del proyecto, responsabilizarse de ella y no tener que trabajar todos sobre el mismo código. De esta manera, luego se implementaría la aplicación de manera sencilla haciendo uso de las clases ya creadas. Otra ventaja es que si se desea cambiar la funcionalidad de una tarea sólo se tendrá que hacer en la definición de las clases. No obstante, la POO que ofrece PHP no es una POO real del todo. Todavía no implementa todas las características propias de esta forma de programación, aunque lo hará en un futuro, cuando salga la nueva versión del motor de PHP, Zend2.

Herramientas Bluetooth en GNU/Linux

Cuando se comienza un sistema de esta naturaleza, es muy útil para el desarrollador ser capaz de encontrar dispositivos y servicios, conectarse a ellos, revisar el estado de las conexiones y monitorear las comunicaciones a través de herramientas, aplicaciones o comandos existentes o al menos que hayan mostrado su efectividad en desarrollos parecidos.

En Linux existen 3 principales partes del subsistema Bluetooth (las rutinas a nivel de kernel, las bibliotecas de desarrollo libbluetooth y los comandos a nivel usuario). El kernel se ocupa de la manipulación del hardware Bluetooth que se encuentra montado en la máquina, del adaptador Bluetooth además de mostrar una interfaz al resto del sistema que permite a cualquier aplicación Bluetooth trabajar con el hardware.

La biblioteca de desarrollo libbluetooth proporciona un conjunto de estructuras y funciones que utilizan los programadores. Son una abstracción de algunas de las operaciones más comunes (tales como detectar dispositivos cercanos) y proporcionan funciones simples que pueden ser invocadas para efectuar tareas comunes.

Finalmente las herramientas a nivel usuario son un conjunto de programas que efectúan tareas muy específicas y que pueden ser invocadas desde la línea de comandos.

Linux mantiene un conjunto de procesos (demonios) que se ejecutan constantemente utilizando la biblioteca Bluetooth de desarrollo para manipular los recursos Bluetooth del sistema según la configuración que considere el usuario.

Actualmente existe un conjunto de herramientas conocidas como BlueZ que han logrado hacer de las tareas un proceso muy sencillo, aportando también suficiente flexibilidad considerando las necesidades de cada usuario.

BlueZ fue inicialmente desarrollada por Max Krasnyansky en Qualcomm y en 2001 fue liberada bajo la licencia GPL. El anuncio fue enviado a la lista de correo de desarrollo de Bluetooth Axis y comenzó a ser la base Bluetooth oficial para Linux. Un mes más tarde, fue considerada por Linus Torvalds para incluirla en el Kernel Linux, empezando con la versión 2.4.6.

En Enero de 2004 el mantenimiento de BlueZ fue entregado a Marcel Holtmann. Esto ocurrió un mes después de Torvalds liberase la primera versión del kernel 2.6.

El objetivo de una calificación oficial de BlueZ como un subsistema para Bluetooth SIG fue alcanzada en Abril de 2005 con ayuda de Tom Tom BV⁷.

⁷ <http://www.bluez.org/about/history>

Proporciona soporte para las capas base de Bluetooth y protocolos. Es flexible, eficiente y tiene una implementación modular. Presenta las siguientes características:

- Completa implementación modular
- Multiproceso simétrico seguro
- Soporte para múltiples dispositivos bluetooth
- Abstracción real del hardware
- Interfaz de conexión estándar a todas las capas
- Soporte de seguridad a nivel de servicio y dispositivo

Actualmente BlueZ cuenta con una gran diversidad de módulos:

- Subsistema base de kernel Bluetooth
- Capas de kernel de audio
- Implementaciones de Kernel
- Controladores de dispositivos virtuales
- Biblioteca y Procesos SDP y Bluetooth
- Utilidades de prueba y configuración
- Herramientas de análisis y protocolos de descodificación

Plataformas:

Los módulos, bibliotecas y utilidades de BlueZ son conocidos por trabajar de manera muy eficiente en muchas arquitecturas soportadas por Linux. Esto incluye plataformas multiproceso como también sistemas hyper multiprocesamiento:

- Intel y AMD x86
- AMD64 y EM64T(x86-64)
- SUN SPARC 32/64bit
- PowerPC 32/64bit
- Intel StrongARM and Xscale
- Hitachi/Renesas SH processors
- Motorola DragonBall

Distribuciones:

BlueZ es soportado por muchas distribuciones Linux y en general es compatible con cualquier sistema Linux en el mercado.

- Debian GNU/Linux
- Ubuntu Linux
- Fedora Core/Red Hat Linux
- OpenSUSE/SuSE Linux
- Mandrake Linux

Existen seis herramientas en línea de comandos provistas por BlueZ que son indispensables al momento de configurar un dispositivo Bluetooth en una máquina así como para depurar aplicaciones. Tres de los comandos comienzan con “hci” y uno con “sdp”:

“hci”: Identifica la Interfaz Controladora del Equipo (Host Controller interface)

“sdp”: Protocolo de Servicio de Descubrimiento (Service Discovery Protocol).

Estas herramientas tienen una gran variedad de configuraciones y comandos. Algunos de estos sólo regresan información, mientras que otros modifican alguna configuración o realizan alguna acción.

Algunos de los comandos se muestran a continuación:

Hciconfig

Este comando es utilizado para configurar las propiedades básicas de los adaptadores Bluetooth del sistema. Como el nombre lo indica, provee un componente a nivel usuario para la interfaz de configuración. Cuando se ejecuta sin ningún argumento, muestra el estado de los adaptadores conectados a la máquina. En todos los demás casos, su uso es como se muestra:

```
# hciconfig <dispositivo> <comando> <argumentos...>
```

Donde <dispositivo> es regularmente hci0 (hci1 especifica el segundo adaptador Bluetooth si se tienen 2, hci2 el tercero, etc.) La mayoría de los comandos requieren privilegios de súper usuario.

Consulta del estado de los adaptadores Bluetooth

En la forma más básica, sin parámetros muestra la siguiente información.

```
Debian-1:~# hciconfig
hci0:Type: USB
      BD Address: 00:50:C2:7F:4A:2A ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNING PSCAN ISCAN
      RX bytes:1080 acl:0 sco:0 events:34 errors:0
      TX bytes:358 acl:0 sco:0 commands:25 errors:0
```

Es así, que cada adaptador Bluetooth reconocido por el conjunto de herramientas BlueZ despliega su información. En este caso se muestra:

El adaptador: hci0

Dirección: 00:50:C2:7F:4A:2A

La parte “UP RUNNING” indica que el adaptador está activo. “PSCAN” e “ISCAN” indica Inquiry Scan y Page Scan respectivamente. El resto de la descripción son estadísticas y algunas propiedades del dispositivo.

Habilitado/Deshabilitando un Adaptador

Los comandos up y down se utilizan para habilitar y deshabilitar un adaptador Bluetooth:

```
Debian-1:~# hciconfig hci0 down
Debian-1:~# hciconfig
hci0: Type: USB
      BD Address: 00:50:C2:7F:4A:2A ACL MTU: 310:10 SCO MTU: 64:8
      DOWN
      RX bytes:991 acl:0 sco:0 events:27 errors:0
```

```
TX bytes:345 acl:0 sco:0 commands:24 errors:0
```

```
Debian-1:~# hciconfig hci0 up
Debian-1:~# hciconfig
hci0: Type: USB
      BD Address: 00:50:C2:7F:4A:2A ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNING PSCAN ISCAN
      RX bytes:1934 acl:0 sco:0 events:50 errors:0
      TX bytes:687 acl:0 sco:0 commands:47 errors:0
```

Configuración Inquiry Scan y Page Scan

Los ajustes de Inquiry scan y Page scan para un adaptador Bluetooth determinan si es detectable por dispositivos cercanos y si el adaptador aceptará peticiones de conexiones entrantes, respectivamente. La siguiente tabla resume los comandos para configurar ambas opciones (Tabla 4.5).

<u>Inquiry</u>	<u>Page Scan</u>	<u>Command</u>
On	On	Piscan
Off	On	Pscan
On	Off	Iscan
Off	Off	Noscan

Tabla 4.5 Configuración de dispositivo

```
Debian-1:~# hciconfig hci0 noscan
Debian-1:~# hciconfig
hci0: Type: USB
      BD Address: 00:50:C2:7F:4A:2A ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNING
      RX bytes:2201 acl:0 sco:0 events:53 errors:0
      TX bytes:697 acl:0 sco:0 commands:50 errors:0
```

*El comando anterior deshabilita tanto Inquiry Scan y Page Scan para el adaptador hci0.

Configuración permanente

Los cambios hechos por hciconfig son sólo temporales, y las configuraciones son borradas después de un reinicio o cuando el dispositivo es inhabilitado y habilitado de nuevo. La forma recomendable para mantener las modificaciones de forma permanente es editar el archivo /etc/bluetooth/hcid.conf. Este archivo es comúnmente cargado al arranque del sistema, y es usado para configurar los dispositivos Bluetooth:

```
# HCI daemon configuration file.
#
# HCID options
options {
    # Automatically initialize new devices
    autoinit yes;
```

```

.
.
.
.
# Default settings for HCI devices
device {
.
.
# Inquiry and Page scan
    iscan enable; pscan enable;
.
.
.
}

```

Hcitol

La herramienta hcitol básicamente tiene dos funciones. La primera es buscar y detectar dispositivos Bluetooth cercanos, la segunda es probar y mostrar información sobre las conexiones Bluetooth en modo de depuración.

Detectando Dispositivos bluetooth Cercanos.

hcitol scan busca los dispositivos Bluetooth cercanos y muestra su dirección y su nombre.

```

Debian-1:~# hcitol scan
Scanning ...
    01:02:03:B7:D6:12    palmen

```

En esta ejecución, n dispositivos son encontrados. La operación de detección de la dirección de dispositivos cercanos y búsqueda de nombre, son dos procesos diferentes y cuando se busca este último regularmente toma un poco más de tiempo. Si el nombre no es importante, el comando *hcitol inq* es útil para mostrar sólo la primera parte de la tarea que es encontrar la dirección de dispositivos cercanos.

```

Debian-1:~# hcitol inq
Inquiring ...
    01:02:03:B7:D6:12    clock offset: 0x7d6b    class: 0x100114
    00:03:7A:D9:66:7C    clock offset: 0x6509    class: 0x1c010c
    00:1D:FD:E9:CB:A7    clock offset: 0x7700    class: 0x5a0204

```

Sdptool

La herramienta sdptool tiene dos usos principales. El primero es buscar y encontrar el Servicio de Descubrimiento (Service Discovery Protocol, SDP) de dispositivos cercanos. Esto es útil para ver qué perfiles Bluetooth están implementados por otro dispositivo, tales como teléfonos celulares. El segundo es para la configuración básica de los servicios SDP ofrecidos por la máquina local.

Búsqueda avanza de servicios

sdptool browse [addr] lista un registro de servicios ofrecidos por el dispositivo Bluetooth con la dirección *addr*. Cuando se omite *addr*, sdptool revisa todos los dispositivos cercanos. Si *local* es

usado como la dirección, entonces el servidor SDP local es verificado. Cada registro de servicio de la lista es brevemente descrito.

Un registro común se muestra a continuación:

```
Debian-1:~# sdptool browse local
Browsing FF:FF:FF:00:00:00 ...
Service Name: Headset Audio Gateway
Service RecHandle: 0x10000
Service Class ID List:
  "Headset Audio Gateway" (0x1112)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
  Channel: 12
Profile Descriptor List:
  "Headset" (0x1108)
  Version: 0x0100
Service Name: Audio Source
Service RecHandle: 0x10001
Service Class ID List:
  "Audio Source" (0x110a)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  PSM: 25
  "AVDTP" (0x0019)
  uint16: 0x100
Profile Descriptor List:
  "Advanced Audio" (0x110d)
  Version: 0x0100
Service Name: AVRCP TG
Service RecHandle: 0x10002
Service Class ID List:
  "AV Remote Target" (0x110c)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  PSM: 23
  "AVCTP" (0x0017)
  uint16: 0x103
Profile Descriptor List:
  "AV Remote" (0x110e)
  Version: 0x0103
Service Name: AVRCP CT
Service RecHandle: 0x10003
Service Class ID List:
  "AV Remote" (0x110e)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  PSM: 23
  "AVCTP" (0x0017)
  uint16: 0x103
Profile Descriptor List:
  "AV Remote" (0x110e)
  Version: 0x0103
```

Aquí, el dispositivo local muestra por ejemplo, un servicio llamado “Headset Audio Gateway” que opera en el canal 12 con RFCOMM. El servicio tiene un Identificador Único Universal (UUID) 0x1112, y también agrega al Perfil de Headset, indicado por el descriptor de perfil listado al final. En general, esta información es suficiente para que una aplicación determine si es ó no el servicio que se está buscando (UUID 0x1112) y como conectarse (usando RFCOMM canal 12).

sdptool search puede ser utilizado para buscar dispositivos cercanos con un servicio específico. El servicio para buscar puede ser especificado ya sea con un UUID o detectando la cadena como “SP” (Serial Port) u “OPUSH” (OBEX Object Push). Por ejemplo, para buscar el servicio OBEX Object Push, es necesario ejecutar:

```
Debian-1:~# sdptool search OPUSH
Inquiring ...
Searching for OPUSH on 01:02:03:B7:D6:12 ...
Service Name: OBEX Object Push
Service RecHandle: 0x10001
Service Class ID List:
  "OBEX Object Push" (0x1105)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
  "OBEX" (0x0008)
Language Base Attr List:
  code_ISO639: 0x656e
  encoding: 0x8cc
  base_offset: 0x100
```

```
# sdptool search 0X1101
```

```
Debian-1:~# sdptool search 0x1105
Class 0x1105
Inquiring ...
Searching for 0x1105 on 01:02:03:B7:D6:12 ...
Service Name: OBEX Object Push
Service RecHandle: 0x10001
Service Class ID List:
  "OBEX Object Push" (0x1105)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
  "OBEX" (0x0008)
Language Base Attr List:
  code_ISO639: 0x656e
  encoding: 0x8cc
  base_offset: 0x100
```

Configuración Básica de Servicio

sdptool add <name> puede ser usado para publicar un conjunto predefinido de servicios, todos, perfiles estandarizados Bluetooth. No puede ser usado para publicar un servicio arbitrario con un UUID definido por el usuario. En esos casos la definición puede ser realizada a través de programación.

sdptool del <handle> puede ser utilizado para quitar la publicación de un servicio local. El servicio SDP mantiene un identificador para cada servicio que lo relaciona al servidor, básicamente es un apuntador al registro del servicio. Para identificarlo, se puede encontrar en la descripción del servicio utilizando *sdptool browse* y buscar la línea “Service Rechandle:”. Con el ejemplo que se

muestra a continuación, el servicio de Puerto Serial tiene el identificador 0x10000, en la máquina, el siguiente comando detendrá la publicación del servicio:

```
Debian-1:~# sdptool del 0x10000
```

sdptool también provee comandos para modificar los registros de servicio (por ejemplo: para cambiar un UUID). Esto, junto con los comandos *add* y *del* existen para que los programadores puedan ver el código fuente de sdptool y ejemplos de cómo realizar lo mismo con aplicaciones propias.

Hcidump

hcidump puede ser utilizado para interceptar y desplegar todos los paquetes Bluetooth enviados y recibidos por la máquina local. Esto puede ser muy útil para determinar y examinar en qué proceso la conexión falla así como para la depuración a bajo nivel de conexiones y transferencia de datos. Hcidump requiere privilegios de súper usuario.

Cuando se ejecuta sin ningún parámetro, *hcidump* muestra un resumen de los paquetes Bluetooth intercambiados entre la máquina local y el adaptador Bluetooth. Esto incluye paquetes de configuración del dispositivo, descubrimientos de dispositivos y establecimientos de conexión entre otros. Los paquetes entrantes anteceden un símbolo “>” (mayor que), y los paquetes salientes con el símbolo “<” (menor que). La longitud de cada paquete (plen) también se muestra. Por ejemplo, *hcidump* comienza dentro de una terminal y el comando *hcitool inq* en otra, la salida de *hcidump* se muestra a continuación (Tabla 4.6)

<u>Terminal 1</u>	<u>Terminal 2</u>
<pre>ladai:~# hcidump HCI sniffer - Bluetooth packet analyzer ver 1.32 device: hci0 snap_len: 1028 filter: 0xffffffff < HCI Command: Inquiry (0x01 0x0001) plen 5 > HCI Event: Command Status (0x0f) plen 4 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 . . . > HCI Event: Inquiry Result with RSSI (0x22) plen 15 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 > HCI Event: Inquiry Complete (0x01) plen 1</pre>	<pre>alejandro.amaya.blue@ladai:~\$ hcitool inq Inquiring ... 00:1C:35:90:98:6C clock offset: 0x1082 class: 0x5a0204 00:1E:DC:6A:3E:ED clock offset: 0x3a6c class: 0x5a0204</pre>

Tabla 4.6 – Ejecución del comando *hcidump*

Aquí, se puede ver que el primer comando (Inquiry) fue ejecutado en el adaptador Bluetooth para buscar dispositivos cercanos, 24 paquetes de tamaño igual a 15 bytes, 1 paquete de 4 bytes y 1 paquete de 1 byte fueron enviados: información sobre el estado del comando, el resultado de descubrimiento indicando que varios dispositivos cercanos fueron detectados, y otro paquete del estado una vez que el descubrimiento fue completado. De esta forma, *hcidump* provee sólo un resumen básico de los paquetes, que no son siempre suficientes para la depuración. Una opción es

usar la bandera -X, que causa que hcidump despliegue el contenido de cada paquete en formato hexadecimal junto con sus códigos ASCII. Se muestra un ejemplo a continuación (Tabla 4.7):

Terminal 1	Terminal 2
<pre>ladai:~# hcidump -X HCI sniffer - Bluetooth packet analyzer ver 1.32 device: hci0 snap_len: 1028 filter: 0xffffffff < HCI Command: Inquiry (0x01 0x0001) plen 5 0000: 33 8b 9e 08 00 3.... > HCI Event: Command Status (0x0f) plen 4 0000: 00 01 01 04 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 b1 cd 3d 4b 17 00 01 02 04 02 52 ce 1a b0 ...=K.....R... > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 b1 cd 3d 4b 17 00 01 02 04 02 52 ce 1a b0 ...=K.....R... > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 b1 cd 3d 4b 17 00 01 02 04 02 52 ce 1a ac ...=K.....R... > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 ed 3e 6a dc 1e 00 01 02 04 02 5a 26 3a bf ..>j.....Z&: . . > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 b1 cd 3d 4b 17 00 01 02 04 02 52 ce 1a af ...=K.....R... > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 ed 3e 6a dc 1e 00 01 02 04 02 5a 25 3a c2 ..>j.....Z%: > HCI Event: Inquiry Result with RSSI (0x22) plen 15 0000: 01 b1 cd 3d 4b 17 00 01 02 04 02 52 ce 1a ae ...=K.....R... > HCI Event: Inquiry Complete (0x01) plen 1 0000: 00</pre>	<pre>alejandro.amaya.blue@ladai:~\$ hcitool inq Inquiring ... 00:1C:35:90:98:6C clock offset: 0x107d class: 0x5a0204 00:1E:DC:6A:3E:ED clock offset: 0x3a75 class: 0x5a0204</pre>

Tabla 4.7 Ejecución del comando hcidump -X

La opción -V para hcidump -V despliega toda la información como puede reunir de cada paquete, y resume los que no puede interpretar. Si se utiliza junto con -X, provee toda la información para paquetes que puede decodificar, así como también los datos hexadecimales para todos los demás paquetes. El ejemplo se muestra a continuación (Tabla 4.8):

Terminal 1	Terminal 2
<pre>ladai:~# hcidump -X -V HCI sniffer - Bluetooth packet analyzer ver 1.32 device: hci0 snap_len: 1028 filter: 0xffffffff < HCI Command: Inquiry (0x01 0x0001) plen 5 lap 0x9e8b33 len 8 num 0 > HCI Event: Command Status (0x0f) plen 4 Inquiry (0x01 0x0001) status 0x00 ncmd 1 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 bdaddr 00:1A:75:41:17:34 mode 1 clkoffset 0x5ecc class 0x520204 rssi -82 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 bdaddr 00:1A:75:41:17:34 mode 1 clkoffset 0x5ecc class 0x520204 rssi -80 > HCI Event: Inquiry Result with RSSI (0x22) plen 15 bdaddr 00:1A:75:41:17:34 mode 1 clkoffset 0x5ecb class 0x520204 rssi -80 > HCI Event: Inquiry Result with RSSI (0x22) plen 15</pre>	<pre>alejandro.amaya.blue@ladai:~\$ hcitool inq Inquiring ... 00:1C:35:90:98:6C clock offset: 0x107c class: 0x5a0204 00:1E:DC:6A:3E:ED clock offset: 0x3a76 class: 0x5a0204</pre>

<u>Terminal 1</u>	<u>Terminal 2</u>
<pre> bdaddr 00:21:86:63:0A:08 mode 1 clkoffset 0x496c class 0x3e0104 rssi -84 . . . > HCI Event: Inquiry Result with RSSI (0x22) plen 15 bdaddr 00:1E:DC:6A:3E:ED mode 1 clkoffset 0x3a27 class 0x5a0204 rssi -68 > HCI Event: Inquiry Complete (0x01) plen 1 status 0x00 </pre>	

Tabla 4.8 – Ejecución del comando `hcidump -X -V`

l2ping

La herramienta `l2ping` envía paquetes a otro dispositivo Bluetooth y espera una respuesta. Es un tipo especial de paquete L2CAP que contiene información sin significado: cuando un dispositivo Bluetooth recibe un paquete, debe mandar el paquete de regreso al origen. Esto es útil para probar y analizar comunicaciones L2CAP con otro dispositivo Bluetooth. Si dos dispositivos se comunican, y se notan algo lentos, entonces `l2ping` pueden proporcionar información en tiempo sobre cuanto toma enviar y recibir paquetes de cierto tamaño. El único parámetro requerido es la dirección del dispositivo Bluetooth.

Por ejemplo, para enviar paquetes al dispositivo 01:02:03:B7:D6:12

```

Debian-1:~# l2ping -c 4 01:02:03:B7:D6:12
Ping: 01:02:03:B7:D6:12 from 00:50:C2:7F:4A:2A (data size 44) ...
20 bytes from 01:02:03:B7:D6:12 id 0 time 14.89ms
20 bytes from 01:02:03:B7:D6:12 id 1 time 31.89ms
20 bytes from 01:02:03:B7:D6:12 id 2 time 27.90ms
20 bytes from 01:02:03:B7:D6:12 id 3 time 27.82ms

```

4 sent, 4 received, 0% loss

Sistema Operativo Debian

Debian es una distribución Linux, es un conjunto de herramientas y programas básicos que hacen las veces de Sistema Operativo en la computadora. El proyecto Debian comenzó en Agosto de 1993 con el ideal de ser una distribución abierta a través de contribuciones de desarrolladores a escala internacional. Una de sus principales características es la implementación del “micro-empaquetado” que utiliza un manejo detallado de las dependencias de cada paquete para garantizar la consistencia del sistema. Entre sus principales características sobresale el soporte para diferentes tipos de procesadores: Intel i386 y superiores, los procesadores Alpha, ARM, Intel IA-64, Motorola 68k, MIPS, PA-RISC, PowerPC, Sparc (y UltraSparc), IBM S/390 y Hitachi SuperH.

Capítulo V

Diseño

En esta etapa se generan los documentos que permitirán a los desarrolladores conocer el correcto funcionamiento del producto, estos documentos son un conjunto de guías que los ayudarán a enfocarse sólo en la construcción.

5.1 Estructura general del sistema

La administración contará con opciones necesarias para manipular cualquier información almacenada por el sistema.

5.1.1 Diseño del mapa del sitio (Figura 5.1)

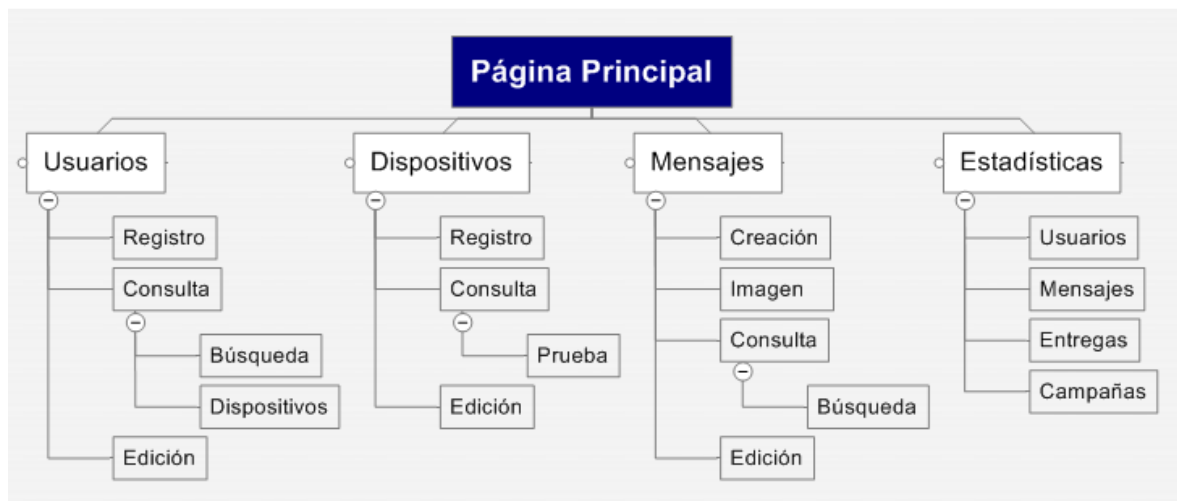


Figura 5.1 - Diseño del mapa del sitio

5.1.2 Diseño de la administración

La administración contará con una estructura básica con la finalidad de hacer de todas las tareas una actividad sencilla de realizar. La estructura de las páginas será la siguiente:

CABECERA (Figura 5.2)

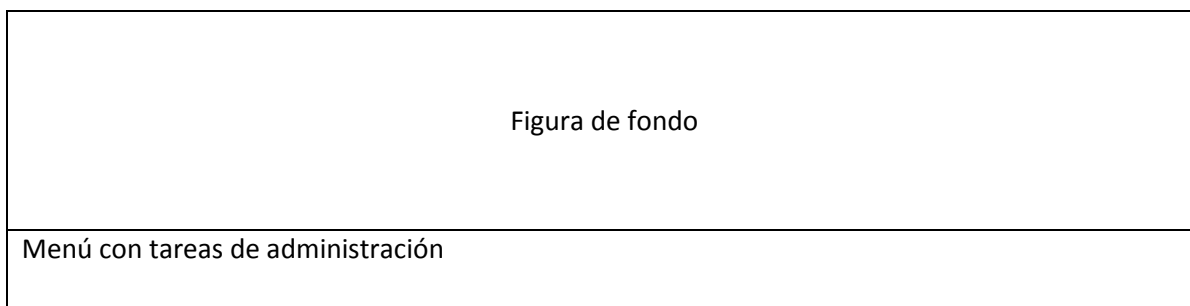


Figura 5.2 - Cabecera

CUERPO DE LA PÁGINA (Figura 5.3)

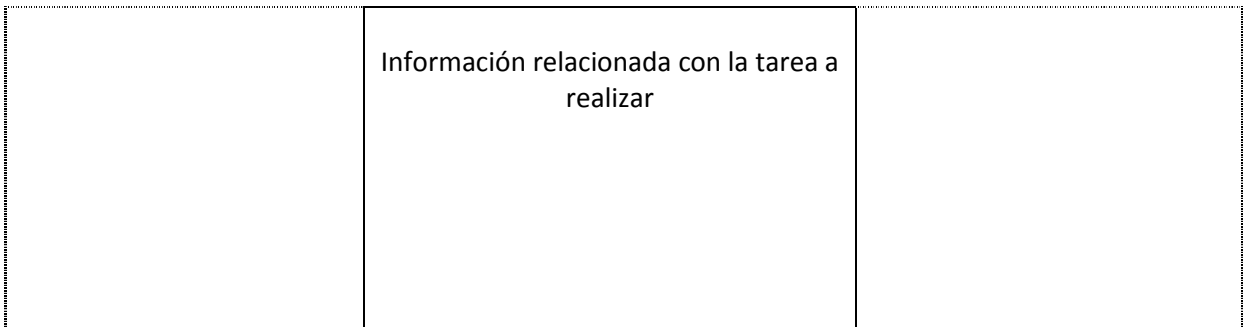


Figura 5.3 - Cuerpo de la página

PIE DE PÁGINA (Figura 5.4)

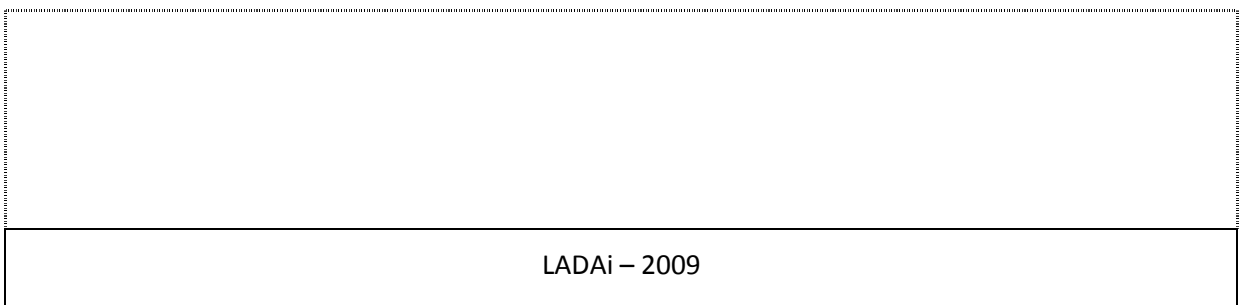


Figura 5.4 - Pie de página

Estándar de diseño: Nombrado de componentes

Componente Inicio (Tabla 5.1)

Componente	Inicio
Descripción	Formulario que muestra pantalla para iniciar sesión en el sistema
Módulo	Vista
Propiedades	Se compone de: campo de texto (usuario), campo de texto (contraseña), botón_continuar

Tabla 5.1 - Componente Inicio

Componente Lista de Usuarios (Tabla 5.2)

Componente	Listado de usuarios
Descripción	Lista de los usuarios registrados que muestra la información general del registro.
Módulo	Vista
Propiedades	Se compone de un bloque que mostrará a manera de lista, los siguientes datos: Nombre, apellido, usuario, activo, correo, nombre del rol, password, fecha de creación.

Tabla 5.2 – Componente Lista de Usuarios

Componente Registro de dispositivos (Tabla 5.3)

Componente	Registro de dispositivos
Descripción	Formulario para ingresar los datos para registrar un dispositivo y asociarlo a un usuario.
Módulo	Vista
Propiedades	Se compone de: campo de texto (Dirección Mac, usuario), campo de selección (tipo, canal), casilla selección (activo).

Tabla 5.3 - Componente Registro de dispositivos

Componente Listado de dispositivos (Tabla 5.4)

Componente	Listado de dispositivos
Descripción	Lista de dispositivos registrados.
Módulo	Vista
Propiedades	Se compone de un bloque que mostrará a manera de lista, los siguientes datos: Id_dispositivo, usuario, dirección mac, tipo, canal, activo, fecha de creación, fecha de actualización y una liga de prueba.

Tabla 5.4 - Componente Listado de dispositivos

Componente Crear Mensaje (Tabla 5.5)

Componente	Crear_Mensaje
Descripción	Formulario que permite crear mensajes nuevos
Módulo	Vista
Propiedades	Se compone de: campo de texto (Título del mensaje, contenido, número de entregas), campo de selección (tipo de usuario, fecha de entrega, fecha de selección).

Tabla 5.5 – Componente Crear_Mensaje

Componente Listado de Mensajes (Tabla 5.6)

Componente	Listado de Mensajes
Descripción	Listado de mensajes describiendo la información general del registro.
Módulo	Vista
Propiedades	Se compone de un bloque que mostrará a manera de lista, los siguientes datos: Título, contenido del mensaje, status, activo, entregas, fecha de creación, fecha de actualización, fecha de entrega, fecha de terminación. Así como ligas para realizar operaciones sobre esta información (Eliminar, editar).

Tabla 5.6 - Componente Listado de Mensajes

Componente Figura_Mensaje (Tabla 5.7)

Componente	Figura_Mensajes
Descripción	Formulario que permite seleccionar una Figura para generar los archivos a entregar y el registro de entrega
Módulo	Vista
Propiedades	Casilla de selección(Figura), botón_generar

Tabla 5.7 – Componente Figura_Mensajes

Componente Mensaje_a_Enviar (Tabla 5.8)

Componente	Mensaje_a_Enviar
Descripción	Bloque de texto que confirma la información enviada
Módulo	Vista
Propiedades	Texto (mensaje de confirmación), botón_regresar.

Tabla 5.8 - Componente Mensaje a Enviar

Componente Escaneo (Tabla 5.9)

Componente	Escaneo
Descripción	Encuentra dispositivos dentro del rango de alcance
Módulo	Controlador
Propiedades	Parámetros de programa: Proceso de escaneo, acción de escaneo, búsqueda de servicio.

Tabla 5.9 – Componente Escaneo

Componente Envío_de_Mensaje (Tabla 5.10)

Componente	Envío_de_Mensaje
Descripción	Envío de archivo a dispositivos vía bluetooth.
Módulo	Controlador
Propiedades	Parámetros de programa: Protocolo de transmisión, información del dispositivo (dirección mac, canal, activación de pareo automático) y archivo.

Tabla 5.10 – Componente Envío_de_Mensajes

Componente Reportes (Tabla 5.11)

Componente	Reportes
-------------------	----------

Descripción	Sección de reportes para: dispositivos, mensajes, entregas y usuarios.
Módulo	Vista
Propiedades	Se compone de: hipervínculos(criterios para generar reporte), calendarios dinámicos (fecha de inicio y fecha de fin), botón_graficar.

Tabla 5.11 - Componente Reportes

Interfaces de los componentes

Componente: Inicio (Tabla 5.12)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Inicio	Muestra un formulario	<ul style="list-style-type: none"> Cajas de texto(login y password) y botón para enviar la información 	
Botón_Enviar	Enviar la información para la autenticación del usuario.	<ul style="list-style-type: none"> Se requiere una url para la autenticación. 	1

Tabla 5.12 – Componente Inicio

Componente: Mensajes de error (Tabla 5.13)

<u>Id</u>	<u>Mensajes de error</u>
1	Error de autenticación

Tabla 5.13 – Componente Mensajes de error

Componente: Registro de usuarios (Tabla 5.14)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Registro de usuario	Muestra un formulario para el registro de los usuarios	<ul style="list-style-type: none"> Nombre Apellido Correo Password Rol Active Fecha de creación Fecha de actualización 	
Botón_Crear	Enviar la información para registro del usuario	<ul style="list-style-type: none"> Se requiere una url para llevar a cabo el registro en la base de 	2

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
		datos.	

Tabla 5.14 – Componente Registro de usuarios

Mensajes de error (Tabla 5.15)

<u>Id</u>	<u>Mensajes de error</u>
2	Se encontraron los siguientes campos no validos.

Tabla 5.15 – Componente Mensajes de error

Componente: Listado de usuarios (Tabla 5.16)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Listado de usuarios	Muestra un listado con todos los usuarios	<ul style="list-style-type: none"> • Nombre • Apellido • Usuario • Activo • Correo • Rol • Password • Fecha de creación 	
Botón_Nuevo	Enviar la información para ejecutar la consulta.		3

Tabla 5.16 – Componente Listado de usuarios

Mensajes de error (Tabla 5.17)

<u>Id</u>	<u>Mensajes de error</u>
3	No fue posible realizar la consulta a la base de datos

Tabla 5.17 – Componente Mensajes de error

Componente: Registro de dispositivos (Tabla 5.18)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Registro de dispositivos	Muestra un formulario para el registro dispositivos	<ul style="list-style-type: none"> • Usuario que será asociado al dispositivo • Dirección_Mac • Tipo de dispositivo • Canal de dispositivo 	

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
		<ul style="list-style-type: none"> • Active • Fecha de creación • Fecha de actualización 	
Botón_Crear	Enviar la información para el registro del dispositivo.	<ul style="list-style-type: none"> • Se requiere una url para llevar a cabo el registro en la base de datos. 	4

Tabla 5.18 - Componente: Registro de dispositivos

Mensajes de error (Tabla 5.19)

<u>Id</u>	<u>Mensajes de error</u>
4	Se encontraron los siguientes campos no validos.

Tabla 5.19 – Componente Mensajes de error

Componente: Listado de dispositivos (Tabla 5.20)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Listado de dispositivos	Muestra un listado con todos los dispositivos	<ul style="list-style-type: none"> • Id_dispositivo • Usuario • Dirección_Mac • Tipo • Canal • Activo • Fecha_de_Creación • Fecha_de_Actualización • Prueba 	
Botón_Nuevo	Enviar la información para ejecutar la consulta.		5

Tabla 5.20 – Componente Listado de dispositivos

Mensajes de error (Tabla 5.21)

<u>Id</u>	<u>Mensajes de error</u>
5	No fue posible realizar la consulta a la base de datos

Tabla 5.21 – Componente Mensajes de error

Componente: Registrar Mensajes (Tabla 5.22)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Registro de mensaje	Muestra un formulario para el registro de mensaje	<ul style="list-style-type: none"> • Título del mensaje • Contenido • Tipo de usuario • Activo • Entrega • Finalización 	
Botón_Crear	Enviar la información para el registro del dispositivo.	<ul style="list-style-type: none"> • Se requiere una url para llevar a cabo el registro en la base de datos. 	6

Tabla 5.22 – Componente Registrar Mensajes

Mensajes de error (Tabla 5.23)

<u>Id</u>	<u>Mensajes de error</u>
6	Se encontraron los siguientes campos no validos.

Tabla 5.23 – Componente Mensajes de error

Componente: Listado de Mensajes (Tabla 5.24)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Listado de mensajes	Muestra un listado con todos los mensajes	<ul style="list-style-type: none"> • ID del mensaje • Título del mensaje • Contenido del mensaje • Status • Tipo de usuario • Activo • Fecha de entrega • Fecha de actualización • Fecha de finalización 	
Botón_Nuevo	Permite desplegar la plantilla que contiene al componente listado de mensajes.		7

Tabla 5.24 – Componente Listado de Mensajes

Mensajes de error (Tabla 5.25)

<u>Id</u>	<u>Mensajes de error</u>
7	No fue posible realizar la consulta a la base de datos

Tabla 5.25 – Componente Mensajes de error

Componente: Figura de Mensaje (Tabla 5.26)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Figura de Mensaje	Muestra una sección de imágenes predefinidas.	<ul style="list-style-type: none"> Figura para asignar al mensaje. 	8
Botón_Aceptar	Enviar la información para la creación de imágenes y entregas.		

Tabla 5.26 – Componente Figura de Mensaje

Mensajes de error (Tabla 5.27)

<u>Id</u>	<u>Mensajes de error</u>
8	No fue posible mostrar las imágenes de muestra.

Tabla 5.27 – Componente Mensajes de error

Componente: Figura de Envío (Tabla 5.28)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Prototipo de entrega	Mostrar la forma genérica de la Figura.		9
Botón_Regresar	Direccionar al listado de los mensajes		

Tabla 5.28 - Componente Figura de Envío

Mensajes de error (Tabla 5.29)

<u>Id</u>	<u>Mensajes de error</u>
9	No fue posible crear las entregas

Tabla 5.29 - Componente Mensajes de error

Componente: Reportes (Tabla 5.30)

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
Listado de reportes	Mostar gráfica con información contenida en la base de datos.	<ul style="list-style-type: none"> Fecha_inicio Fecha_fin 	
Botón_Gráficar	Enviar la información para procesar las fechas y el		10

<u>Interfaz</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Id Mensaje de error</u>
	critero de la consulta.		

Tabla 5.30 – Componente Reportes

Mensajes de error (Tabla 5.31)

<u>Id</u>	<u>Mensajes de error</u>
7	No fue posible realizar la consulta a la base de datos.

Tabla 5.31 – Componente Mensajes de error

5.2 Modelo de casos de uso

Diagrama general de casos de uso: administrador

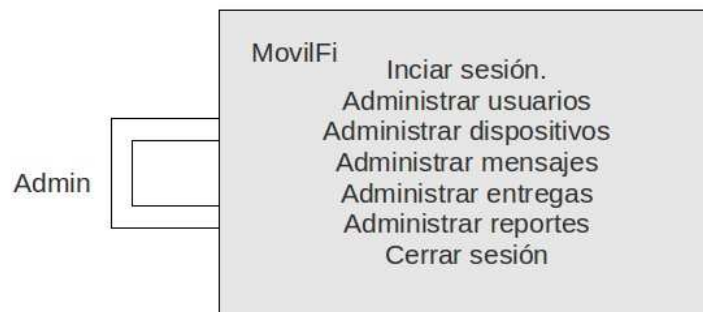


Imagen 5.5 - Diagrama general de casos de uso: administrador

Caso de uso: Inicio de sesión (Tabla 5.32)

Actor	Administrador
Descripción	Iniciar sesión en el sistema Web/administración
Precondición	El usuario ingresó a la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.32 - Caso de uso Inicio de sesión

Flujo principal: Iniciar sesión (Tabla 5.33)

<u>Flujo principal</u>				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar nombre de usuario y contraseña y dar clic en el botón enviar.	2	Recibe y valida los datos	E1
		3	Despliega la pantalla de inicio de sesión del usuario.	

Tabla 5.33 - Flujo principal Iniciar sesión

Excepciones: Iniciar sesión (Tabla 5.34)

Excepciones		
ID	Nombre	Acción
E1	Datos inválidos	Despliega un mensaje de error en la pantalla

Tabla 5.34 – Excepciones Iniciar Sesión

Pos condición: Iniciar sesión (Tabla 5.35)

Pos condición	El usuario ha iniciado sesión
---------------	-------------------------------

Tabla 5.35 - Pos condición Iniciar Sesión

Caso de uso: Administrar usuarios (Tabla 5.36)

Actor	Administrador
Descripción	Registrar usuario
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.36 - Caso de uso Administrar usuarios

Flujo principal: Administrar usuarios (Tabla 5.37)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresa al formulario de registro de usuarios.	2	Despliega formulario completo con los campos necesarios para el registro del usuario.	
3	Captura datos para el registro del usuario	4	Despliega mensaje indicando que tuvo éxito la acción.	E2, E3

Tabla 5.37 - Flujo principal Administrar usuarios

Excepciones: Administrar usuarios (Tabla 5.38)

Excepciones		
ID	Nombre	Acción
E2	Datos inválidos	Despliegan un mensaje de error en la pantalla
E3	Registro fallido	Despliega un mensaje al no ser posible registrar en la base de datos

Tabla 5.38 – Excepciones Administrar usuarios

Pos condición: Administrar usuarios (Tabla 5.39)

Pos condición	Se ha registrado usuario
---------------	--------------------------

Tabla 5.39 - Pos condición Administrar usuario

Caso de uso: Editar usuario (Tabla 5.40)

Actor	Administrador
Descripción	Editar usuario
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.40 - Caso de uso Editar usuario

Flujo principal: Editar usuario (Tabla 5.41)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresa al formulario de edición de usuarios.	2	Despliega formulario con los campos activos para su edición.	
3	Captura los datos para editar la información del usuario	4	Despliega mensaje indicando que tuvo éxito la acción.	E4, E5

Tabla 5.41 - Flujo principal Editar usuario

Excepciones: Editar usuario (Tabla 5.42)

Excepciones		
ID	Nombre	Acción
E4	Datos inválidos	Despliega un mensaje de error en la pantalla
E5	Actualización fallida	Despliega un mensaje al no ser posible registrar en la base de datos

Tabla 5.42 - Excepciones: Editar usuario

Pos condición: Editar usuario (Tabla 5.43)

Pos condición	Se han actualizado los datos del usuario
---------------	--

Tabla 5.43 - Pos condición: Editar usuario

Caso de uso: Administrar mensajes (Tabla 5.44)

Actor	Administrador
Descripción	Crear mensaje
Precondición	El usuario ingresó e inició sesión en la página de administración La base deberá tener al menos un usuario y un dispositivo registrado
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.44 - Caso de uso: administrar mensajes

Flujo principal: Administrar mensajes (Tabla 5.45)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar a al formulario para el registro del mensaje	2	Despliega formulario completo con los campos necesarios para el registro del mensaje.	
3	Captura los datos para el registro del mensaje	4	Muestra confirmación de la acción realizada y muestra la lista de los mensajes existentes	E6,E7,E8

Tabla 5.45 - Flujo principal: administrar mensajes

Excepciones: Administrar mensajes (Tabla 5.46)

Excepciones		
ID	Nombre	Acción
E6	Datos inválidos	Despliega un mensaje de error en la pantalla
E7	Registro/Modificación en la base de datos, en la tabla de "Mensajes"	Despliega mensaje: "No fue posible modificar/leer en la base de datos"
E8	Imposible generar el mensaje	Despliega mensaje: "No fue posible crear el mensaje".

Tabla 5.46 - Excepciones: administrar mensajes

Pos condición: Administrar mensajes (Tabla 5.47)

Pos condición	Se ha creado mensaje.
---------------	-----------------------

Tabla 5.47 - Pos condición: administrar mensajes

Caso de uso: Editar mensaje (Tabla 5.48)

Actor	Administrador
Descripción	Editar mensaje
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.48 - Caso de uso: Editar mensaje

Flujo principal: Editar mensaje (Tabla 5.49)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar al formulario de edición de mensaje.	2	Despliega formulario con los campos que son modificables.	

3	Captura los datos para la edición del mensaje	4	Despliega confirmación de la acción realizada.	E9,E10
---	---	---	--	--------

Tabla 5.49 - Flujo principal: Editar mensaje

Excepciones: Editar Mensaje (Tabla 5.50)

Excepciones		
ID	Nombre	Acción
E9	Datos inválidos	Despliega un mensaje de error en la pantalla
E10	Actualización fallida	Despliega mensaje: "No fue posible modificar/leer en la base de datos"

Tabla 5.50 - Excepciones: Editar Mensaje

Pos condición: Editar Mensaje (Tabla 5.51)

Pos condición	Se han actualizado los datos del mensaje.
---------------	---

Tabla 5.51 - Pos condición: Editar Mensaje

Caso de uso: Administrar dispositivos (Tabla 5.52)

Actor	Administrador
Descripción	Añadir dispositivo
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.52 - Caso de uso: Administrar dispositivo

Flujo principal: Administrar dispositivos (Tabla 5.53)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar a al formulario para el registro del dispositivo	2	Ejecuta el proceso "Información Dispositivos"	E11
		3	Despliega la información encontrada con la ejecución del proceso " Información Dispositivos"	E12
4	Captura los datos para el registro del dispositivo	5	Despliega confirmación del registro de los dispositivos.	E13,E14

Tabla 5.53 - Flujo principal: Administrar dispositivo

Excepciones: Administrar dispositivos (Tabla 5.54)

Excepciones		
ID	Nombre	Acción
E11	No fue posible recolectar información del dispositivo	Notificar que el dispositivo no cuenta con el servicio necesario para la comunicación.
E12	El dispositivo que realiza el descubrimiento no responde	Notificar que el dispositivo que realiza el descubrimiento no realiza la tarea solicitada
E13	Datos no validos	El sistema muestra un mensaje de error en la pantalla
E14	Inserción fallida	No fue posible registrar en la base de datos

Tabla 5.54 - Excepciones: Administrar dispositivo

Pos condición: Administrar dispositivos (Tabla 5.55)

Pos condición	El dispositivo se ha registrado correctamente.
---------------	--

Tabla 5.55 - Pos condición: Administrar dispositivos

Caso de uso: Editar Dispositivo (Tabla 5.56)

Actor	Administrador
Descripción	Editar dispositivo
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.56 - Casos de uso: Editar dispositivo

Flujo principal: Editar Dispositivo (Tabla 5.57)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresa al formulario de edición de dispositivo.	2	Despliega formulario con los campos activos para su edición.	
3	Captura los datos para la edición del dispositivo	4	El sistema muestra confirmación de la acción realizada.	E15,E16

Tabla 5.57 – Flujo principal: Editar dispositivo

Excepciones: Editar Dispositivo (Tabla 5.58)

Excepciones		
ID	Nombre	Acción
E15	Datos inválidos	Despliega un mensaje de error en la pantalla
E16	Actualización fallida	Despliega mensaje: “No fue posible modificar/leer en la base de datos”

Tabla 5.58 - Excepciones: Editar dispositivo

Pos condición: Editar Dispositivo (Tabla 5.59)

Pos condición Se han actualizado los datos del dispositivo.

Tabla 5.59 - Pos condición: Editar usuario

Caso de uso: Entregas (Tabla 5.60)

Actor	Administrador
Descripción	Generar entregas
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.60 - Caso de uso: Entregas

Flujo principal: Entregas (Tabla 5.61)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar a la administración de entregas	2	Despliega una lista de mensajes para generar las entregas	
3	Selecciona la opción: "Generar" para crear las entregas e imágenes	4	Ejecuta la tarea: "Generar"	E17,E18
		5	Despliega la confirmación de la acción realizada.	

Tabla 5.61 - Flujo principal: Entregas

Excepciones: Entregas (Tabla 5.62)

Excepciones		
ID	Nombre	Acción
E17	Imposible generar imágenes asociadas al mensaje	Despliega mensaje "No fue posible crear las imágenes asociadas al mensaje"
E18	Registro/Modificación en la base de datos, en la tabla de "Entregas"	Despliega mensaje "No fue posible modificar/leer en la base de datos"

Tabla 5.62 - Excepciones: Entregas

Pos condición: Entregas (Tabla 5.63)

Pos condición Se generó correctamente las entregas

Tabla 5.63 - Pos condición: Entregas

Caso de uso: Iniciar/terminar las entregas (Tabla 5.64)

Actor	Administrador
-------	---------------

Descripción	Administrar entregas
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.64 - Caso de uso: Iniciar/terminar las entregas

Flujo principal: Iniciar/terminar las entregas (Tabla 5.65)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar a la administración de entregas	2	Despliega los mensajes para generar las entregas	
3	Selecciona la tarea: "Check"	4	Ejecuta la tarea: "Check" que activará o desactivará las entregas de acuerdo a la fecha registrada en el mensaje.	E17
		5	Despliega confirmación de la acción realizada.	

Tabla 5.65 - Flujo principal: Iniciar/terminar las entregas

Excepciones: Iniciar/terminar las entregas (Tabla 5.66)

Excepciones		
ID	Nombre	Acción
E17	Ejecución incorrecta	Muestra mensaje "No fue posible ejecutar la tarea correctamente"

Tabla 5.66 - Excepciones: Iniciar/terminar las entregas

Pos condición: Iniciar/terminar las entregas (Tabla 5.67)

Pos condición	Las entregas se han actualizado correctamente.
---------------	--

Tabla 5.67 - Pos condición: Iniciar/terminar las entregas

Caso de uso: Envío de entregas (Tabla 5.68)

Actor	Administrador
Descripción	Enviar entregas
Precondición	El usuario ingresó e inició sesión en la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.68 - Caso de uso: Envío de entregas

Flujo principal: Envío de entregas (Tabla 5.69)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar a la	2	Despliega una lista de los mensajes para generar las	

	administración de entregas		entregas	
3	Selecciona la tarea:"Send"	4	Ejecuta la tarea: "Send" que enviará las entregas pendientes a los dispositivos que haya detectado dentro del área de cobertura.	E18
		5	Despliega confirmación de la acción realizada.	

Tabla 5.69 - Flujo principal: Envío de entregas

Excepciones: Envío de entregas (Tabla 5.70)

Excepciones		
ID	Nombre	Acción
E18	No se realiza el envío de las entregas	Muestra mensaje "No fue posible realizar el envío"

Tabla 5.70 - Excepciones: Envío de entregas

Pos condición: Envío de entregas (Tabla 5.71)

Pos condición	Las entregas se han enviado correctamente.
---------------	--

Tabla 5.71 - Pos Condición: Envío de entregas

Caso de uso: Administrar escaneo (Tabla 5.72)

Actor	Administrador
Descripción	Administrar escaneo
Precondición	El usuario ingresó e inició sesión en una consola del servidor
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.72 - Caso de uso: Administrar escaneo

Flujo principal: Administrar escaneo (Tabla 5.73)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresar a la administración de escaneos			
2	Selecciona opción "Scan"	3	Realiza el escaneo y a obtiene las direcciones MAC de los dispositivos en intervalos fijos de tiempo	E19
		4	Despliega de la ejecución de tarea	

Tabla 5.73 - Flujo principal: Administrar escaneo

Excepciones: Administrar escaneo (Tabla 5.74)

Excepciones		
ID	Nombre	Acción
E19	El módulo que realiza el escaneo no regresa información.	Por medio de un mensaje se notifica el posible fallo del dispositivo encargado del escaneo.

Tabla 5.74 - Excepciones: Administrar escaneo

Pos condición: Administrar escaneo (Tabla 5.75)

Pos condición	Los dispositivos cercanos se registran en la base de datos.
---------------	---

Tabla 5.75 - Pos condición: Administrar escaneo

Caso de uso: Consultar reportes (Tabla 5.76)

Actor	Administrador
Descripción	Consultar reportes
Precondición	El usuario ingresó e inició sesión en una consola del servidor
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.76 - Caso de uso: Consultar reportes

Flujo principal: Consultar reportes (Tabla 5.77)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Ingresa a la sección de reportes	2	Muestra diferentes criterios de consulta	
3	Selecciona criterio y ejecuta "Graficar"	4	Despliega los datos de acuerdo al criterio y fechas seleccionados.	E20

Tabla 5.77 - Flujo principal: Consultar reportes

Excepciones: Consultar reportes (Tabla 5.78)

Excepciones		
ID	Nombre	Acción
E20	El sistema no grafica de los datos indicados.	Se notifica la posible pérdida de conexión con la base de datos.

Tabla 5.78 - Excepciones: Consultar reportes

Pos condición: Consultar reportes (Tabla 5.79)

Pos condición	Los reportes se generan correctamente.
---------------	--

Tabla 5.79 - Pos condición: Consultar reportes

Caso de uso: Cerrar sesión (Tabla 5.80)

Actor	Administrador
Descripción	Cerrar sesión en el sistema Web/administración
Precondición	El usuario ingresó a la página de administración
Requisitos	Sistema Web disponible y base de datos activa

Tabla 5.80 - Caso de uso: Cerrar sesión

Flujo principal: Cerrar sesión (Tabla 5.81)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Cierra sesión desde el hipervínculo “Salir” del menú de navegación.	2	Termina la sesión actual	E21
		3	Despliega la pantalla de inicio de sesión	

Tabla 5.81 - Flujo principal: Cerrar sesión

Excepciones: Cerrar sesión (Tabla 5.82)

Excepciones		
ID	Nombre	Acción
E21	La sesión no termina correctamente	Se reporta la excepción.

Tabla 5.82 Excepciones: Cerrar sesión

Pos condición: Cerrar sesión (Tabla 5.83)

Pos condición	La sesión se cierra correctamente.
----------------------	------------------------------------

Tabla 5.83 - Pos condición: Cerrar sesión

Diagrama de casos de uso: Receptor (Figura 5.6)

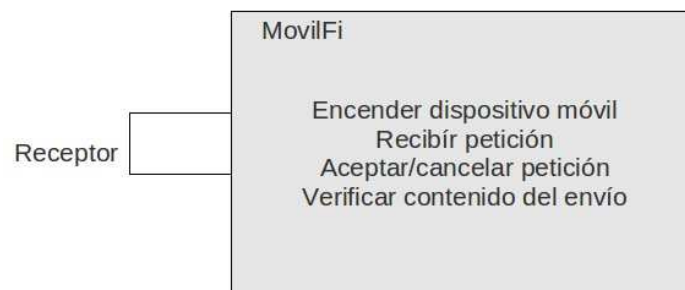


Figura 5.6 - Diagrama de casos de uso: Receptor

Caso de uso: Recibir entrega (Tabla 5.84)

Actor	Receptor
Descripción	Recibir petición de entrega
Precondición	El usuario está registrado en el sistema
Requisitos	El usuario se encuentra dentro del área de alcance.

Tabla 5.84 - Caso de uso: Recibir entrega

Flujo principal: Recibir entrega (Tabla 5.85)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Recibe una petición para la entrega de un archivo desde el equipo central			E22
2	Acepta petición para recibir la entrega	3	Responde enviando el mensaje.	E23
		4	El sistema actualiza el estado de esa entrega	E24 E25

Tabla 5.85 - Flujo principal: Recibir entrega

Excepciones: Recibir entrega (Tabla 5.86)

Excepciones		
ID	Nombre	Acción
E22	No recibe la petición	Generar reporte para revisar el estado del dispositivo
E23	No recibe contenido	Informar sobre el estado del dispositivo
E24	No actualiza la base de datos	Generar reporte para revisar estado de los procesos relacionados con la base de datos
E25	No recibe el contenido indicado	Generar reporte de seguimiento.

Tabla 5.86 Excepciones: Recibir entrega

Pos condición: Recibir entrega (Tabla 5.87)

Pos condición	La entrega se recibe correctamente
---------------	------------------------------------

Tabla 5.87 - Pos condición: Recibir entrega

Caso de uso: Cancelar/ No responder a la entrega (Tabla 5.88)

Actor	Receptor
Descripción	Recibir petición de entrega
Precondición	El usuario está registrado en el sistema

Requisitos El usuario se encuentra dentro del área de alcance.

Tabla 5.88 - Caso de uso: Cancelar/No responder a la entrega

Flujo principal: Cancelar/ No responder a la entrega (Tabla 5.89)

Flujo principal				
Paso	Eventos actor	Paso	Eventos sistema	Excepción
1	Recibe una petición para la entrega de un archivo desde el equipo central			E22
2	Cancela/ No responde a la petición para recibir la entrega	3	Responde asignando prioridad al mensaje.	E25
		4	El sistema actualiza el estado de esa entrega	E24

Tabla 5.89 - Flujo principal: Cancelar/No responder a la entrega

Excepciones: Cancelar/ No responder a la entrega (Tabla 5.90)

Excepciones		
ID	Nombre	Acción
E22	No recibe la petición	Generar reporte para revisar el estado del dispositivo
E25	No recibe contenido	Generar informe sobre la asignación de prioridad de la entrega
E24	No actualiza la base de datos	Generar reporte para revisar estado de los procesos relacionados con la base de datos.

Tabla 5.90 - Excepciones: Cancelar/No responder a la entrega

Capítulo VI

Desarrollo

En esta etapa se realiza la construcción del proyecto en base al documento de Análisis y Diseño. Se asignan las tareas de acuerdo al perfil de cada uno de los integrantes del grupo de trabajo y se asignan fechas de entrega.

6.1 Configuración del Servidor

Prácticas de administración de servidores

Existe un conjunto de buenas prácticas administrativas cuya finalidad es realizar las tareas de manera que se pueda facilitar el mantenimiento futuro, por ejemplo:

- Realizar respaldos de la configuración, del sistema de archivos y de los archivos de los proyectos periódicamente. Siempre que se modifique un archivo de configuración, se debe realizar un respaldo de dicho archivo con la fecha de modificación en el nombre, así en caso de que la nueva configuración no funcione correctamente, es posible restaurar el servidor a un punto interior.
- Realizar respaldos de las Bases de Datos.
- Los respaldos deben realizarse en horas en las que el servidor no se encuentre muy solicitado.
- Mantener al día las actualizaciones de seguridad y del sistema.
- Dejar abiertos solo los puertos necesarios. Para seguridad del sistema, además de realizar la actualización, por ejemplo solo habilitar los puertos 80, 8080 para el acceso Web, el puerto 25 para el acceso mediante ssh, el puerto 5432 para el acceso a la base de datos.
- Realizar un monitoreo de las bitácoras del sistema, de los servicios y el acceso al servidor. Detectar en ellos los registros inusuales e identificar la causa.
- Realizar un monitoreo del espacio libre en disco. Asignar cuotas de espacio en disco específicas para cada usuario.

Es posible dejar estas acciones como una tarea del sistema, y programar su ejecución en un determinado periodo de tiempo.

Para el servidor se decidió instalar el sistema operativo Debian GNU/Linux <<etch>>, el cual permite tener un sistema libre, completo y actual. La instalación se realizó vía red, obteniendo los paquetes desde el servidor de la Facultad de Ciencias.

Administración de paquetes Debian

Los Repositorios son referencias a servidores donde se encuentran archivos necesarios para brindar la robustez al sistema operativo, éstos están definidos en el archivo `/etc/apt/sources.list`, cuya descripción es la siguiente:

Una línea del archivo puede iniciar con la palabra “`deb`” que define los paquetes binarios ó la palabra “`deb-src`” que define las fuentes de los paquetes. En seguida, el primer argumento es la URL del servidor, el segundo argumento es el nombre de la distribución, los siguientes argumentos son una lista de componentes Debian válidos.

El archivo `sources.list` del servidor cuenta con los siguientes repositorios:

```
deb http://mmc.igeofcu.unam.mx/debian/ etch main
deb-src http://mmc.igeofcu.unam.mx/debian/ etch main
```

```
deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib

deb http://www.symfony-project.org/get debian/

deb http://packages.dotdeb.org etch all
deb-src http://packages.dotdeb.org etch all
```

Dichos repositorios fueron necesarios para tener las versiones más actuales del software requerido.

Configuración de Servidor Web Apache

Los archivos de configuración de Apache son `apache2.conf` y `httpd.conf`, se encuentran en la carpeta `/etc/apache2/`.

En el archivo `apache2.conf` se indica el directorio raíz del proyecto:

```
<VirtualHost 127.0.0.1:80>
  DocumentRoot "/home/blue/public_html/web"
  DirectoryIndex index.php
  <Directory "/home/blue/public_html/web">
    AllowOverride All
    Allow from All
  </Directory>
</VirtualHost>
```

Donde `"/home/blue/public_html/"` es la ruta del sistema de archivos del proyecto. Después de realizar una modificación en la configuración, los cambios sólo serán permanentes reiniciando el servicio de Apache, para hacerlo se ejecuta el siguiente comando:

```
/etc/init.d/apache2 restart
```

Configuración Symfony

Instalación del Framework Symfony

De la página del proyecto (<http://www.symfony-project.org>) se descargó el paquete de la última versión de Symfony (1.2).

Dentro del directorio `/lib` del servidor se creó la carpeta `vendor` y dentro de ésta se descomprimió el archivo descargado, creando una carpeta a la cual se renombró como `symfony`.

Para verificar la configuración, Symfony cuenta con un script llamado `check_configuration.php` con el cual se verifica de forma automática si el equipo cuenta con los requerimientos necesarios. En primera instancia la salida fue la siguiente:

```
** Mandatory requirements **
OK          PHP version is at least 5.2.4
OK          php.ini has zend.zel_compatibility_mode set to off

** Optional checks **

OK          PDO is installed
OK          PDO has some drivers installed: pgsql
OK          PHP-XML module is installed
OK          XSL module is installed
```

```

OK          The token_get_all() function is available
OK          The mb_strlen() function is available
OK          The iconv() function is available
OK          The utf8_decode() is available
[[WARNING]] A PHP accelerator is installed: FAILED
            *** Install a PHP accelerator like APC (highly recommended)
***
[[WARNING]] php.ini has short_open_tag set to off: FAILED
            *** Set it to off in php.ini ***
[[WARNING]] php.ini has magic_quotes_gpc set to off: FAILED
            *** Set it to off in php.ini ***
OK          php.ini has register_globals set to off
OK          php.ini has session.auto_start set to off
OK          PHP version is not 5.2.9

```

Está información indica que se cuenta con los recursos necesarios para ejecutar symfony sin ninguna complicación. De manera opcional es posible modificar la configuración e instalar el modulo APC para cumplir con el resto de los requerimientos.

Creación del proyecto

Dentro de la carpeta public_html ubicada en el home del usuario blue (nombre del proyecto), es necesario ejecutar la siguiente instrucción que generará el sistema de archivos siguiente:

```
php /lib/vendor/symfony/data/bin/symfony generate:project blue
```

```

apps      : Mantiene todas las aplicaciones del proyecto
cache     : Cache generado por el framework
config    : Archivos de configuración
lib       : Bibliotecas y clases del proyecto
log       : Bitácoras del framework
plugins   : Plugins instalados
test      : Archivos de prueba por unidad y funcionales
web       : Directorio raíz de la aplicación

```

El sistema es un conjunto de aplicaciones, la instrucción necesaria para la creación de una aplicación es:

```
php symfony generate:app frontend      #donde frontend es la aplicación
```

```

apps/
  frontend/
    config    : Archivos de configuración de la aplicación.
    lib       : Bibliotecas y clases de la aplicación.
    modules   : Código de la aplicación (Dividido como arquitectura MVC)
    templates: Archivos plantilla para toda la aplicación.

```

Cada una de las aplicaciones se compone de módulos, la instrucción necesaria para su creación es la siguiente:

```
php symfony init-module frontend messages
```

#Donde messages es el nombre del módulo

```

modules/
  messages/
    actions   : Archivo que contiene el código que hace las veces de

```

controlador en el módulo correspondiente.
 Templates : Plantillas correspondiente a cada una de las "acciones"
 del controlador.

Configuración de PHP

El archivo de configuración de PHP, php.ini se encuentra en la carpeta /etc/php5. Para el proyecto es necesario modificar el archivo php.ini y activar el modulo [gd] para manipular las imágenes, activar el modulo [apc] para optimizar el uso del cache, desactivar las opciones magic_quotes_gpc y short_open_tag para cumplir con los requerimientos del framework. Para consultar la configuración de PHP se puede utilizar la función phpinfo() y así verificar los módulos habilitados.

Configuración de Adobe Flash Player

Adobe Flash Player es una aplicación para reproducir elementos multimedia, creado inicialmente por Macromedia y actualmente distribuido por Adobe Systems. Estos elementos multimedia se reproducen en un entorno, como por ejemplo el navegador en cuyo formato del reproductor es el de un Plugin.

Para la instalación, se descargo la última versión de Adobe Flash Player desde la página <http://www.adobe.com> donde se eligió la versión adecuada para Linux. El archivo viene en varios formatos para diferentes distribuciones. En nuestro caso se descargo la versión 10 en un archivo tar.gz, el cual se desempaqueto en la carpeta /usr/lib/mozilla/plugins/.

6.2 Framework

6.2.1 Programación Orientada a Objetos

El paradigma de Programación Orientada a Objetos proporciona a los lenguajes modernos en general la facilidad para solucionar problemas a través de objetos, los cuales se comunican con mensajes con la finalidad de proveer buenas prácticas de desarrollo.

Los mecanismos básicos de la programación orientada a objetos son:

- *Objetos*: Un objeto es una entidad con propiedades particulares y métodos para manipularlo.
- *Mensajes*: Es la forma como los objetos reciben, interpretan y envían respuesta a otros objetos.
- *Métodos*: Un método se implementa en un determinado tipo de objeto y establece el comportamiento que debe de tener cuando recibe un mensaje.
- *Clase*: Es un tipo de objeto que el usuario define y son una abstracción de la realidad.
- *Características de la programación orientada a objetos*.
- *Abstracción*: Es la generalización de una idea, no considera particularidades y permite desarrollar una idea general del asunto en cuestión.
- *Encapsulamiento*: Comúnmente se define para hacer ver a un objeto como una caja negra en donde se almacena toda la información del objeto.
- *Herencia*: Es la reutilización de código ya definido, considerando siempre la jerarquía con que se definen las clases. Cuando una o más clases están en la parte inferior de esta clasificación, entonces se considera que heredan de aquellas que se encuentran en la parte superior de la jerarquía.
- *Polimorfismo*: Es la capacidad de un método de comportarse de acuerdo al clase que lo invoca. Es así que es posible tener una gran variedad de objetos utilizando el mismo medio de acceso.

Las características de la POO de los lenguajes modernos permiten una programación sencilla, tiene como utilidades la lógica de encapsulamiento, la herencia, y una convención clara para los nombres. En caso contrario si se implementa la arquitectura MVC a un lenguaje no orientado a objetos, ocasiona duplicación de código.

6.2.2 Aplicando la arquitectura MVC

La arquitectura MVC separa la lógica del negocio (modelo) y la presentación (vista), permitiendo su fácil mantenimiento.

Por ejemplo si una aplicación debe ejecutarse en navegadores convencionales y dispositivos móviles, sólo se requiere una nueva vista y se puede mantener el controlador original así como el modelo. El controlador ayuda a ocultar los detalles del protocolo utilizado por la petición del modelo y la vista, de este modo el modelo abstrae la lógica de los datos, los cuales hacen que la vista y la acción sean independientes por instancia del tipo de base de datos usada por la aplicación.

El objetivo de la arquitectura MVC consiste en separar el código en tres capas, la lógica de datos se coloca con el modelo, el código de la presentación con la vista, y la lógica de aplicación con el controlador.

Los objetivos que se deben cumplir al aplicar la arquitectura MVC son:

- Manejo de los posibles errores
- El código HTML y PHP deben estar separados
- El código debe ser independiente de la base de datos

Otros patrones de diseño pueden hacer la experiencia de codificar más fácil. El modelo, la vista y el controlador pueden ser aun más subdivididos.

Abstracción de la Base de Datos

La capa del modelo puede ser separada en:

- La capa de acceso a los datos
- La capa de abstracción de la base de datos

De esta manera, las funciones de acceso a los datos no usarán consultas dependientes de la base, y llamarán algunas otras funciones que ejecutarán las consultas por sí mismas. Si se cambia el sistema manejador de base datos, solo la capa de abstracción necesitará ser actualizada.

Elementos de la vista

La capa de vista también puede beneficiarse de separar el código. La mayoría de las páginas web contienen elementos consistentes ó fijos a través de una aplicación: las cabeceras de la página, el diseño gráfico, el pie y la navegación global. Sólo la parte central de las páginas cambia. Esto es porque la vista está separada en diseño y plantilla. El diseño es global para la aplicación ó para un

grupo de páginas. La plantilla sólo pone en forma las variables disponibles por el controlador. Una parte de lógica de programación es necesaria para hacer que componentes (las plantillas y el diseño) trabajen juntos.

De acuerdo a estos principios, la vista puede separarse en tres partes:

- Plantillas
- Lógica
- Diseño

Acción y Controlador

El controlador tiene la mayor carga de trabajo, una parte importante de este trabajo es común a todos los controladores de la aplicación. Las tareas incluyen manejo de peticiones, control de la seguridad, carga de la configuración de la aplicación, y tareas similares. Esto es porque el controlador es frecuentemente dividido en:

- Controlador de acceso, el cual es único para toda la aplicación.
- Acciones, las cuales contienen sólo el controlador de código específico para una página.

Una de las grandes ventajas de un controlador de acceso es la facilidad que proporciona con un único punto de entrada para toda la aplicación. Si se decide bloquear el acceso a la aplicación sólo hay que modificar el script del controlador de acceso. En una aplicación sin controlador de acceso, se debe desactivar de forma individual cada punto de entrada.

6.2.3 Construcción de la Base de Datos

ORM y la capa de abstracción

Para tener acceso a la base de datos en un contexto orientado a objetos se requiere una interfaz que traduzca la lógica de objeto a la lógica relacional. Esta herramienta es llamada ORM ó Mapeo Objeto Relación y sirve para crear objetos con acceso a datos así como para mantener la lógica del negocio. El mayor beneficio de la capa de abstracción es la portabilidad, obligando a utilizar una sintaxis específica para las consultas mientras realiza el trabajo de definición de acuerdo al DBMS particular.

Sinfonía usa PROPEL como ORM, éste mapea clases PHP con tablas de la base de datos. Para crear el modelo de datos objetos que usará Sinfonía, es necesario traducir el modelo relacional de la base de datos.

A continuación se detallará el proceso necesario para la construcción del modelo (cabe mencionar que este proceso es genérico al desarrollar con el framework symfony).

1. Creación de la Base de Datos

Mediante la terminal del sistema operativo, es necesario acceder como usuario postgres y crear la Base de Datos del proyecto.

```
postgres=# CREATE DATABASE blue;
```

1.1 Creación del usuario.

Es necesario crear un usuario asignado a esta base, al cual se le otorgan todos los privilegios necesarios para trabajar.

```
postgres=# CREATE USER blue WITH PASSWORD 'xxxxxx';
```

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE blue TO blue;
```

1.2 Esquema de la Base de Datos

En este punto es necesario mapear las entidades de la bases de datos en formato XML. Como ejemplo, tomando la entidad Deliveries del modelo entidad-relación (Figura 6.1), la estructura será la siguiente:

Deliveries
delivery_id: INTEGER(10) [PK]
device_id: INTEGER(10) [FK]
message_id: INTEGER(10) [FK]
status: BOOLEAN(10)
active: BOOLEAN(1)
created_at: TIMESTAMP(10)

Figura 6.1 - Entidad Deliveries

```
<table name="deliveries" phpName="Deliveries">
  <column name="delivery_id" type="integer" required="true"
primaryKey="true" autoincrement="true" />
  <column name="device_id" type="integer" />
  <foreign-key foreignTable="devices">
    <reference local="device_id" foreign="device_id"/>
  </foreign-key>
  <column name="message_id" type="integer" />
  <foreign-key foreignTable="messages">
    <reference local="message_id" foreign="message_id"/>
  </foreign-key>
  <column name="active" type="boolean" />
```



```

    <column name="status" type="boolean" />
    <column name="created_at" type="timestamp" />
</table>

```

6.2.4 Construcción del Modelo

Una vez mapeadas todas las entidades, la construcción del modelo se realiza ejecutando el siguiente comando:

```
php symfony propel:build-model
```

2.1 Construcción del esquema SQL

El siguiente paso es generar un script con las sentencias SQL que definen las tablas y relaciones del modelo:

```
php symfony propel:build-sql
```

La información de este script, creará la estructura de toda la base de datos:

```
php symfony propel:insert-sql
```

De esta manera en postgres es posible ver la tabla deliveries creada:

Tabla «public.deliveries»

Columna	Tipo	Modificadores
delivery_id	integer	not null default nextval('deliveries_delivery_id_seq'::regclass)
device_id	integer	
message_id	integer	
active	boolean	
status	boolean	
created_at	timestamp without time zone	

Índices:

```
«deliveries_pkey» PRIMARY KEY, btree (delivery_id)
```

Restricciones de llave foránea:

```
«deliveries_FK_1» FOREIGN KEY (device_id) REFERENCES devices(device_id)
```

```
«deliveries_FK_2» FOREIGN KEY (message_id) REFERENCES messages(message_id)
```

Carga de datos inicial

Symfony también permite definir una precarga de datos mediante el archivo `/data/fixtures/fixtures.yml` cuyo contenido es un arreglo de tablas y campos con valores asignados.

Para hacer la precarga se ejecuta el siguiente comando:

```
php symfony propel:data-load
```

Esto es útil para tener datos de prueba sin la necesidad de insertar continuamente mediante sentencias SQL.

Al crear el modelo, se generan las clases correspondientes a las tablas que están definidas en el esquema así como los métodos de acceso estándar - `/lib/om` -. Cuando existe la necesidad de implementar métodos o funciones más específicas es cuando se trabaja con las clases del directorio `/lib/model` ya que heredan de aquellas que son base - `/lib/om` - .

Haciendo un listado del directorio `/lib` del proyecto se muestran las clases asociadas a los módulos:

```
/lib/model
    AccessPointsPeer.php
    AccessPoints.php
    DeliveriesPeer.php
    Deliveries.php
    DevicesPeer.php
    Devices.php
    map
    MessagesPeer.php
    Messages.php
    om
    RolesPeer.php
    Roles.php
    Scans_DeliveriesPeer.php
    Scans_Deliveries.php
    ScansPeer.php
    Scans.php
    Users_DevicesPeer.php
    Users_Devices.php
    UsersPeer.php
    Users.php
```

De esta forma el framework genera clases con métodos como(Tabla 6.1):

```
lib/model/om
```

<u>Método</u>	<u>Función</u>
save()	Guarda un registro en la base de datos.
doSelectOne()	Consulta un registro considerando un criterio
doCountJoinUsers()	Realiza un conteo al hacer una unión con la tabla Users

Tabla 6.1 - Métodos generados por el framework

Y en ocasiones es necesario implementar métodos más específicos (Tabla 6.2): lib/model

<u>Método</u>	<u>Función</u>
findOrCreateD ()	Encuentra las entregas de cada dispositivo para ser enviadas
registerStatusbyImage ()	Registra el estado de una entrega cuando se le asigna una imagen

Tabla 6.2 - Métodos definidos por el desarrollador

6.2.5 Programación de la Vista

Haciendo un listado para las plantillas generales del proyecto se pueden ver los elementos principales para el componente de la vista (Tabla 6.3).

<u>Plantilla</u>	<u>Descripción</u>
footer.php	Pie de la página
header.php	Cabecera
layout.php	Plantilla base para las página
menuAdmin.php	Menú de administración
menu.php	Menú de opciones

Tabla 6.3 - Plantillas generales

Después del análisis en la parte de la vista, el framework genera una estructura básica escalable para la presentación de la información, crea listados y consultas, formularios para registrar los datos y para realizar modificaciones (Tabla 6.4).

<u>Plantilla</u>	<u>Descripción</u>
editSuccess.php	Formulario de edición para un registro dado.
_form.php	Realiza la interacción entre el formulario de edición y creación
indexSuccess.php	Listado del contenido de la tabla correspondiente al módulo
newSuccess.php	Formulario para la creación de un nuevo registro
showSuccess.php	Muestra la consulta de un registro en específico

Tabla 6.4 - Plantillas de estructura básica

Adicionalmente se crean plantillas que no son implementadas directamente por el framework y que deben cubrir necesidades específicas de la presentación (Tabla 6.5).

Plantilla	Descripción
showInfoSuccess.php	Muestra la información de los dispositivos cercanos (Nombre, Dirección Mac y Canal)
imageSuccess.php	Muestra un conjunto de imágenes para asignar al mensaje
usersSuccess.php	Muestra una gráfica con el análisis de los usuarios del sistema

Tabla 6.5 - Plantillas especificadas por el desarrollador

6.2.6 Programación del Controlador

El controlador dentro de una aplicación Web se encarga de las peticiones, seguridad, carga de configuraciones entre otras tareas. En este caso en particular el controlador está contenido dentro de las clases *action* para cada uno de los módulos.

El Framework tiene una correspondencia con la vista por lo que se crearan tantas acciones como plantillas existan (no necesariamente) (Figura 6.6).

Acción	Modulo	Función
actions.class.php/executeIndex	Messages	Contiene la lógica para realizar un listado de los registros contenidos del modulo con el cual está asociado.
actions.class.php/executeDelete	Scans	Contiene la lógica para eliminar un registro del modulo al cual está asociado.
actions.class.php/executeUpdate	Deliveries	Contienen la lógica para actualizar un registro.

Tabla 6.6 - Actions generadas por el framework

Adicionalmente se deben crear las acciones para necesidades específicas como (Tabla 6.7):

Acción	Modulo	Función
actions.class.php/executeDeliver	Messages	Crear las entregas que serán enviadas a los dispositivos cuando el sistema los detecte de acuerdo al rol con que esté registrado ese dispositivo.
actions.class.php/executeScan	Scans	Contiene la lógica para escanear en busca de dispositivos
actions.class.php/executeDeliveriesChartData	Deliveries	Genera todos los datos que serán plasmados en la graficas cuyo criterio será el número de entregas en un rango de tiempo abierto para facilitar su análisis.

Tabla 6.7 - Actions especificadas por el desarrollador

Capítulo VII

Pruebas

7.1 Plan de Pruebas

El plan se genera a partir de la identificación de pruebas requeridas para el cumplimiento de los requerimientos. Las tareas se deberán de distribuir de acuerdo al rol.

Entre las tareas que se realizan se encuentran: Integrar los componentes, documentar los resultados, corregir los defectos encontrados y actualizar el registro de rastreo. Los resultados que se encuentren en esta fase son utilices para verificar y en su caso actualizar el Manual técnico o el Manual de Usuario.

De manera más particular para las aplicaciones Web es necesario considerar:

- **Compatibilidad:** Dada su naturaleza es necesario realizar pruebas en una gran variedad de navegadores, aún cuando se considera el estándar.
- **Gramática:** Es necesario revisar la correcta redacción y el lenguaje que se utiliza dentro de la página.
- **Vínculos sin referencia:** Es de gran importancia desde antes del lanzamiento del sitio.
- **Formularios y programación:** Todos los formularios así como las prestaciones que ofrece el sistema deberán ser probadas para comprobar que en realidad hacen lo que se supone deban hacer.

El reporte de cada una de las pruebas deberá considerar:

- Registro de participantes
- Fecha
- Lugar
- Duración
- Defectos encontrados

7.1.1 Pruebas de Integración (Tabla 7.1)

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
27/04/09	Alejandro Amaya	Ingresar a la URL de inicio	En el explorador ingresar la URL deseada.	Visualizar la página principal de 'blue'.		Se obtuvo el resultado esperado	Correcto
30/04/09		Ingresar a la página para iniciar sesión.	Ingresar datos para iniciar sesión.	Visualizar la aplicación 'frontend'	Mostrar más opciones en el menú: reportes, usuarios, etc	Se obtuvo el resultado esperado	Correcto
		Ingresar a la sección 'Usuarios' en el menú "Crear"	Del menú seleccionar el vínculo 'Mensajes'	Visualizar la página de mensajes	Implementar el vínculo para los mensajes.	El link está inactivo	Pendiente *P1
		Ingresar a la sección 'Mensajes' del menú "Crear"	Del menú seleccionar el vínculo 'Usuarios'	Visualizar la página de usuarios	Implementar el vínculo para los usuarios.	El link está inactivo	Pendiente. *P2
		Ingresar a la sección 'Dispositivos' del menú "Crear"	Del menú seleccionar el vínculo 'Dispositivos'	Visualizar la página de dispositivos	Implementar el vínculo para los dispositivos	El link está inactivo	Pendiente *P3

Tabla 7.1 - Pruebas de Integración

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
30/04/09	Alejandro Amaya	Ingresar a la sección 'Usuarios' en el menú "Consultar"	Del menú seleccionar el vínculo 'Mensajes'	Visualizar la página de mensajes	Implementar el vínculo para los mensajes.	Se obtuvo el resultado esperado	Correcto
		Ingresar a la sección 'Mensajes' del menú "Consultar"	Del menú seleccionar el vínculo 'Usuarios'	Visualizar la página de usuarios	Implementar el vínculo para los usuarios.	Se obtuvo el resultado esperado	Correcto
		Ingresar a la sección 'Dispositivos' del menú "Consultar"	Del menú seleccionar el vínculo 'Dispositivos'	Visualizar la página de dispositivos	Implementar el vínculo para los dispositivos	Se obtuvo el resultado esperado	Correcto
		Ingresar a la sección 'Estadísticas'	Del menú seleccionar el vínculo 'Estadísticas'	Visualizar la página de estadísticas		Se obtuvo el resultado esperado	Correcto

Tabla 7.1 - Pruebas de Integración (cont.)

Actualizaciones (Tabla 7.2)

Corrección	Fecha	Evaluador	Descripción	Resultado
P1	01/05/09	Alejandro Amaya	Se actualizó la URL de re direccionamiento	Se obtuvo el resultado esperado
P2				
P3				

Tabla 7.2 - Actualizaciones

7.1.2 Pruebas del Sistema

Funciones Usuario (Administrador) (Tabla 7.3)

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
27/04/09	Alejandro Amaya	Iniciar sesión	Evento actor 1.1.1.	Datos correctos: envió a mensajes Datos incorrectos: error al iniciar sesión	Se trabaja actualmente con el plugin de symfony sfGuardAuth	Symfony no re direcciona correctamente al momento de iniciar sesión	Pendiente *C1
		Crear nuevo usuario	Evento actor 1.2.3	Usuario registrado en la tabla de usuario. *Visible en el listado de usuarios	Las validaciones son correctas	Usuario registrado y visible en el listado	Correcta
		Editar usuario	Evento actor 1.3.3	Actualización correcta de los nuevos datos.	Las validaciones son correctas	Datos de usuario actualizados.	Correcto
		Crear mensaje	Evento actor 1.4.3	Un nuevo mensaje para entregar. .	La validación funciona correctamente.	El mensaje se crea correctamente.	Correcto
		Editar Mensaje	Evento actor 1.5.3	Cancelar mensaje	Crear action para cancelar mensaje y ponerlo con active=0	Mensaje cancelado	Pendiente *C2
		Registra dispositivo	Evento actor 1.6.4	Registrar el dispositivo con los datos indicados en el formulario		El registro por dispositivo es correcto	Correcto
		Editar información de dispositivo	Evento actor 1.7.3	Editar la información de algún dispositivo		Le edición se realiza correctamente	Correcto

Tabla 7.3 - Funciones Usuario (Administrador)

Fecha	Evaluator	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
		Ejecutar tarea para generar las entregas de un mensaje	Evento actor 1.8.3	Se generan las entregas del mensaje seleccionado con el contenido indicado		En ocasiones las entregas no se generan	Pendiente *C3
		Ejecutar tarea para iniciar/terminar las entregas de acuerdo a la fecha	Evento actor 1.9.3	Las entregas se iniciar/cancelan conforme la fechas que se registraron en su creación.	La tarea deberá automatizarse con un 'task' del framework	Los mensajes de inician/finalizan a la hora indicada.	Correcto *A1
		Ejecutar tarea para realizar los envíos pendientes	Evento actor 1.10.3	Se comienzan a enviar las entregas registradas en el sistema	La tarea deberá automatizarse con un 'task' del framework	Se envían correctamente las entregas	Correcto *A2
		Ejecutar tarea para realizar el escaneo de dispositivos cercanos	Evento actor 1.11.3	Se encuentran nuevos dispositivos y se registran en la base	La tarea deberá automatizarse con un 'task' del framework.	Los dispositivos se registran correctamente	Correcto *A3
		Cerrar sesión	Evento actor 1.13.1	Cerrar sesión del sistema	Todas las operaciones de administración no estarán disponibles una vez que se termine la sesión	La sesión se cierra correctamente	Correcto

Tabla 7.3 - Funciones Usuario (Administrador) (Cont.)

Correcciones (Tabla 7.4)

Corrección	Fecha	Evaluador	Descripción	Resultado
C1	04/05/09	Alejandro Amaya	Se modificó la URL de redireccionamiento	Al iniciar sesión se muestra correctamente la página indicada
C2	04/05/09	Alejandro Amaya	Se modificó la acción que actualiza el estado de los mensaje	Los mensajes se controlan de acuerdo a banderas preestablecidas y se envían de forma correcta de acuerdo a la fecha de su registro.
C2	04/05/09	Ángel Camacho	Se revisaron y actualizaron los permisos de los directorios con las imágenes.	Se crearon corre correctamente todas las imágenes de cada mensaje

*Tabla 7.4 - Correcciones***Actualizaciones (Tabla 7.5)**

Actualización	Fecha	Evaluador	Descripción	Observaciones	Resultado
A1	06/05/09	Alejandro Amaya	Automatizar tarea para iniciar/terminar las entregas de forma automática	Se implementó un 'task' del framework	El procedimiento está automatizado.
A2			Automatizar tarea para realizar el escaneo de dispositivos cercanos		
A3			Automatizar tarea para realizar los envíos pendientes		

Tabla 7.5 – Actualizaciones

Funciones del sistema (Tabla 7.6)

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
28/04/09	Alejandro Amaya	Iniciar escaneo en busca de dispositivos cercanos	Evento actor 1.11.3	Encontrar dispositivos dentro del área de alcance		Se encuentran dispositivos dentro del área de alcance	Correcto
28/04/09	Alejandro Amaya	Iniciar/terminar las entregas de acuerdo a la fecha	Evento actor 1.9.3	Actualizar el estado de las entregas de acuerdo a la fecha		El sistema actualizar el estado de la entrega de acuerdo a la fecha	Correcto
28/04/09	Alejandro Amaya	Realizar los envíos pendientes	Evento actor 1.10.3	Enviar las entregas pendientes		Se envían correctamente las entregas	Correcto
		Actualizar registro después de la entrega.	Evento actor 2.1.4	Actualizar la entrega después que el usuario la recibió correctamente.		El sistema actualiza la entrega después de haberla enviado	Correcto
		Actualizar registro después que el usuario cancela la entrega	Evento actor 2.1.6	Inhabilitar la entrega cuando el usuario la cancela 2 veces		La entrega se inhabilita cuando el usuario la canceló.	Correcto
		Actualizar registro cuando el dispositivo no responde.	Evento actor 2.1.8	Cancelar la entrega cuando el dispositivo no acepta o cancela.		La entrega se inhabilita cuando el usuario no acepta o cancela.	Correcto
		<i>La ejecución de la tarea debe realizarse automáticamente cada minuto</i>	Evento actor 1.11.4	La tarea debe ejecutarse de acuerdo al intervalo de tiempo programado	Si la tarea ya se ejecuta y se cancela para no interrumpir la actual.	La tarea se ejecuta cada minuto.	Correcto

Tabla 7.6 - Funciones del sistema

Funciones Usuario (Receptor) (Tabla7.7)

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
28/04/09	Alejandro Amaya	Recibir petición de entrega	Evento actor 2.1.1	Recibir notificación de petición de entrega.		La notificación se recibe correctamente.	Correcto
		Acepta la petición	Evento actor 2.1.3	El mensaje es recibido correctamente	El mensaje es enviado sólo a los usuario registrados con el rol indicado	El mensaje es enviado al usuario con el rol de su registro.	Correcto
		Cancela la petición	Evento actor 2.1.5	No recibe el archivo	Si el usuario cancela un par de ocasiones la entrega del archivo, se le cancela la respectiva entrega	No recibe el archivo debido a su cancelación	Correcto
		Confirmar contenido del mensaje e ID del usuario	Evento actor 2.1.7	Los datos contenidos en el mensaje deben ser los contenidos en el registro de usuario		El mensaje contiene la información correcta para cada usuario.	Correcto

Tabla 7.7 - Funciones Usuario (Receptor)

7.1.3 Pruebas del Sistema (Sistema abierto)**Funciones Usuario (Administrador) (Tabla 7.8)**

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
27/04/09	Alejandro Amaya	Iniciar sesión	Evento actor 1.1.1.	Datos correctos: envío a mensajes Datos incorrectos: error al iniciar sesión	Se trabaja actualmente con el plugin de symfony sfGuardAuth	Symfony no re direcciona correctamente al momento de iniciar sesión	Correcto
		Crear mensaje	Evento actor 1.4.3	Un nuevo mensaje para entregar.	La validación funciona correctamente.	El mensaje se crea correctamente.	Correcto
		Editar Mensaje	Evento actor 1.5.3	Editar Mensaje	La edición del mensaje contempla sólo la modificación de las fechas	La edición del mensaje es correcta	Correcto
		Consultar reportes	Evento actor 1.12.1	Visualizar reportes de acuerdo al criterio elegido	El intervalo de tiempo quedará abierto para facilitar el análisis	Las graficas muestran la información correctamente de acuerdo al criterio seleccionado	Correcto

Tabla 7.8 - Funciones del usuario

Funciones del sistema (Tabla 7.9)

Fecha	Evaluador	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
28/04/09	Alejandro Amaya	Iniciar escaneo en busca de dispositivos cercanos	Evento actor 1.11.3	Encontrar dispositivos dentro del área de alcance		Se encuentran dispositivos dentro del área de alcance	Correcto
28/04/09		Iniciar/terminar las entregas de acuerdo a la fecha	Evento actor 1.9.3	Actualizar el estado de las entregas de acuerdo a la fecha		El sistema actualizar el estado de la entrega de acuerdo a la fecha	Correcto
28/04/09		Realizar los envíos pendientes	Evento actor 1.10.3	Enviar las entregas pendientes		Se envían correctamente las entregas	Correcto
		Actualizar registro después de la entrega.	Evento actor 2.1.4	Actualizar la entrega después que el usuario la recibió correctamente.		El sistema actualiza la entrega después de haberla enviado exitosamente.	Correcto
		Actualizar registro después que el usuario cancela la entrega	Evento actor 2.1.6	Inhabilitar la entrega cuando el usuario la cancela 2 veces		La entrega se inhabilita cuando el usuario la canceló.	Correcto
		Actualizar registro cuando el dispositivo no responde.	Evento actor 2.1.8	Cancelar la entrega cuando el dispositivo no acepta o cancela.		La entrega se inhabilita cuando el usuario no acepta o cancela.	Correcto
		La ejecución de la tarea debe realizarse automáticamente cada minuto		La tarea debe ejecutarse de acuerdo al intervalo de tiempo programado	Si la tarea ya se ejecuta y se cancela para no interrumpir la actual.	La tarea se ejecuta cada minuto.	Correcto

Tabla 7.9 - Funciones del sistema

Fecha	Evaluable	Descripción	Procedimiento	Resultado esperado	Observaciones	Resultado encontrado	Validación
		Cerrar sesión	Evento actor 1.13.1	Cerrar sesión del sistema	Todas las operaciones de administración no estarán disponibles una vez que se termine la sesión	La sesión se cierra correctamente	Correcto

Tabla 7.9 - Funciones del sistema (Cont.)

Funciones Usuario (Receptor)

Las funciones del Receptor se detallan en el apartado “7.1.2 Pruebas del Sistema - Funciones Usuario (Receptor)”.

Capítulo VIII

Resultados y aportaciones

Una vez realizada la implementación del sistema dentro del laboratorio se consideró una comunidad con un gran número de dispositivos para poder obtener resultados sobre el rendimiento de la aplicación. Es así que se buscó la oportunidad de trabajar con instituciones con diferentes enfoques y criterios. La primera institución que colaboró con la implementación fue el Instituto de Ciencia y Tecnología del Distrito Federal a través de “La Semana de la Ciencia y la Innovación 2009”. La segunda institución que apoyó la implementación del sistema fue la Facultad de Medicina de la UNAM en su departamento de servicios escolares. A continuación se describe el trabajo que se desarrolló con cada una de las instituciones.

*Instituto de Ciencia y Tecnología del Distrito Federal
Semana de la Ciencia y la Innovación 2009.*

Durante “La Semana de la Ciencia y la Innovación 2009” que se realizó en el Palacio de Minería del 21 al 25 de septiembre del presente año, el equipo de trabajo tuvo la oportunidad de brindar el servicio de envío de contenidos a dispositivos móviles a través de Bluetooth con el sistema desarrollado con el trabajo realizado en este proyecto.

Durante cinco días se enviaron contenidos de talleres, exposiciones, conferencias, actividades e información en general relacionada con el evento. En este tiempo se pudo comprobar la capacidad del sistema en cuanto a: número de envíos, velocidad de transmisión, tiempo de envío, actualización de la base de datos entre otras cuestiones relacionadas con la capacidad del sistema. A continuación se muestra la información de dispositivos, mensajes y envíos recopilada durante el evento (Tabla 8.1):



Tabla 8.1 - Mensajes

Dispositivos

Se registraron de forma automática un total de 411 dispositivos.

Lista de dispositivos

Dispositivo	Usuario	Direccion Mac	Tipo	Canal	Activo	Creado	Actualizado	Acción
2	Invitado	00:1D:FD:E9:CB:A7	Celular	9	1	2009-09-21 12:17:45	2009-09-21 12:17:45	prueba
	Invitado	00:16:20:FD:35:4C	Celular	6	1	2009-09-21 12:18:48	2009-09-21 12:18:48	prueba
	Invitado	00:17:E5:84:B0:26	Celular	9	1	2009-09-21 12:19:45	2009-09-21 12:19:45	prueba
	Invitado	00:22:98:67:7D:88	Celular	3	1	2009-09-21 12:21:22	2009-09-21 12:21:22	prueba
	Invitado	00:23:39:F6:62:4C	Celular	7	1	2009-09-21 12:24:43	2009-09-21 12:24:43	prueba
	Invitado	00:16:BC:FC:F2:7E	Celular	9	1	2009-09-21 12:25:40	2009-09-21 12:25:40	prueba
	Invitado	00:1F:5D:E8:F3:C9	Celular	9	1	2009-09-21 12:26:21	2009-09-21 12:26:21	prueba
	Invitado	00:18:C5:64:F6:ED	Celular	9	1	2009-09-21 12:29:39	2009-09-21 12:29:39	prueba
10	Invitado	00:22:66:83:9B:12	Celular	9	1	2009-09-21 12:32:57	2009-09-21 12:32:57	prueba
1	Invitado	IN:VI:TA:DO:OP:EN	PDA	1	1	2009-09-21 12:11:31	2009-09-21 12:11:31	prueba


411 dispositivos (1 al 10) 

Tabla 8.1 (Cont.)– Dispositivos registrados

Envíos

Se enviaron un total de 1296 entregas, de las cuales 803 fueron satisfactorias y 493 fueron canceladas o no fueron recibidas por el usuario.



Tabla 8.1 (Cont.) – Grafica de número entregas

Este evento fue de gran ayuda al momento de comprobar la capacidad del sistema para cumplir los objetivos planteados en un principio. De igual forma ayudó a identificar un conflicto relacionado con la administración de dispositivos, una vez solucionado este punto el sistema tuvo un tiempo de respuesta menor.

Facultad de Medicina UNAM
Servicios Escolares

La implementación se desarrolló en dos etapas, la primera fue realizada por el equipo de trabajo, administrando y monitoreando el comportamiento de la aplicación con la finalidad de registrar toda la información. Durante este periodo se obtuvo la siguiente información (Tabla 8.2):

Mensajes										
					Se registró un primer mensaje con información relacionada con las inscripciones con una duración de 2 semanas aproximadamente.					
Lista de Mensajes										
Mensaje	Título del mensaje	Contenido	Entregas	Imagen	Tipo de usuario	Activo	Creado	Entrega	Actualizado	Finalización
2	Mensaje MI	Recuerda que en Noviembre son las inscripciones de tercer y cuarto año	500	1	3	1	2009-10-20 12:11:00	2009-10-20 12:00:00	2009-10-23 13:14:07	2009-11-15 19:00:00
1	Bienvenida	LADAI - Cursos gratuitos	100	0	3	0	2009-10-20 12:09:42		2009-10-20 12:09:42	
2 mensajes (1 al 2)										

Tabla 8.2 – Detalles de la primera campaña


Dispositivos									
Se registraron de forma automática un total de 3145 dispositivos.									
Lista de dispositivos									
Dispositivo	Usuario	Dirección Mac	Tipo	Canal	Activo	Creado	Actualizado	Acción	
2901	Invitado	00:1F:DF:34:4E:EE	Celular	9	1	2009-11-02 12:37:50	2009-11-02 12:37:50	prueba	
2902	Invitado	00:1D:3B:23:74:2D	Celular	9	1	2009-11-02 12:41:03	2009-11-02 12:41:03	prueba	
2903	Invitado	00:15:2A:6C:10:26	Celular	9	1	2009-11-02 12:46:09	2009-11-02 12:46:09	prueba	
2904	Invitado	00:21:AB:0B:04:A6	Celular	9	1	2009-11-02 12:52:07	2009-11-02 12:52:07	prueba	
2905	Invitado	00:16:20:EE:9D:EE	Celular	5	1	2009-11-02 12:52:54	2009-11-02 12:52:54	prueba	
2906	Invitado	00:1F:5D:E5:18:EA	Celular	9	1	2009-11-02 12:58:53	2009-11-02 12:58:53	prueba	
2907	Invitado	00:18:C5:65:4D:63	Celular	9	1	2009-11-02 13:00:44	2009-11-02 13:00:44	prueba	
2908	Invitado	00:1B:98:33:8B:10	Celular	7	1	2009-11-02 13:02:32	2009-11-02 13:02:32	prueba	
2909	Invitado	00:16:98:B1:76:2E	Celular	1	1	2009-11-02 13:04:25	2009-11-02 13:04:25	prueba	
2910	Invitado	00:16:98:A5:CB:56	Celular	1	1	2009-11-02 13:05:14	2009-11-02 13:05:14	prueba	
3145 dispositivos (2871 al 2880) 									

Tabla 8.2 (Cont.)– Dispositivos registrados

En este punto es necesario mencionar que del total de dispositivos, existe un conjunto que no presenta las facilidades para la transferencia de archivos (Tabla 8.3)

Envíos

Se enviaron un total de 296 entregas exitosas y se cancelaron o no fueron recibidas por el usuario 1436 entregas.



Tabla 8.3 – Grafica de entregas en Medicina

En esta etapa el número de entregas es bajo comparado con el número de cancelaciones o entregas fallidas debido a las siguientes cuestiones:

- La antena se encuentra en un lugar de paso, es decir, un dispositivo que deba recibir una entrega en un momento puede estar cerca del punto de acceso y poco tiempo después puede salir del área de cobertura.
- Por el momento no se ha publicado el servicio ofrecido por el departamento a la comunidad e la Facultad por lo que los alumnos no están seguros de la fuente de la información.

Para la segunda fase de la implementación, se facilitó toda la documentación, manuales y la administración del sistema al departamento de servicios escolares con la finalidad de que un equipo independiente lo manipulara y proporcionara al equipo de trabajo sus observaciones sobre el comportamiento, usabilidad y en general información útil para la mejora del sistema. De esta manera el equipo de trabajo obtuvo información muy valiosa y pudo hacer los ajustes indicados.

Al implementar el sistema en ambas instituciones se buscó siempre cumplir con los objetivos que se listan en el capítulo I (1.3 *Objetivos*): permitir a los usuarios consultar contenidos desde un dispositivo móvil de forma inmediata, facilitar el acceso a la información, tener un gran impacto respecto al número de usuarios a un costo mínimo así como una administración de fácil manejo y bajo mantenimiento.

En general el sistema tuvo un comportamiento aceptable y la información obtenida ayudó a analizar su rendimiento bajo condiciones diferentes comparadas con las que hasta ese momento

se habían considerado por lo que se afirma que se logró una alta tasa de cumplimiento de los requerimientos.

Por otra parte la implementación del sistema dentro de la misma Universidad, en este caso en la Facultad de Medicina hace notar el compromiso de los alumnos y la Facultad de Ingeniería hacia su comunidad. Es así que el desarrollo de este proyecto remarcan los principios de la ingeniería: Innovar y mejorar los procesos y servicios con el objetivo de ahorrar recursos.

Capítulo IX

Conclusiones

En la actualidad las aplicaciones móviles tienen un crecimiento muy marcado gracias a la alta capacidad de procesamiento de los nuevos dispositivos y a su uso cada vez más intuitivo, es por eso que se puede notar un crecimiento en general de la publicidad e inversión en los medios digitales.

La tecnología Bluetooth ha facilitado la comunicación entre dispositivos de manera muy práctica y sobre todo a una velocidad considerable, la fiabilidad en la transferencia de información ha permitido que su implementación se generalice en casi todos los dispositivos y se continúe avanzando en su desarrollo contrario a lo que sucedió con otras tecnologías como la de rayos infrarrojos.

Por otra parte las prestaciones que otorgan las herramientas de desarrollo de software en conjunto con los sistemas manejadores de bases de datos permiten crear aplicaciones robustas que facilitan a los usuarios finales una manipulación de información de forma práctica y rápida sin dejar de lado la integridad de ésta. Un punto importante en el desarrollo de los sistemas de software actuales es la capacidad de crear sistemas a partir de herramientas libres, sistemas operativos como Linux y herramientas en general que permiten el control de hardware desde un nivel muy bajo que facilita la manipulación de su rendimiento de manera muy eficiente.

Para el caso que aquí nos ocupa y con estos antecedentes es importante hacer notar la eficiencia de un lenguaje enfocado al desarrollo web para manipular dispositivos externos y la capacidad de éste para obtener información 100% fiable sobre su operación.

Limitaciones de la tecnología.

Es necesario plantear los límites que se deben considerar al utilizar este tipo de tecnología en la parte de comunicaciones debido a la facilidad que existe para acceder a información restringida del dispositivo, esto enfocado a las implicaciones éticas y/o legales involucradas en estas acciones. Es por eso que no se puede hacer uso de la información recopilada en esta investigación con fines distintos a los definidos por la propia organización, ya que esto estaría en contra de los principios del grupo de trabajo.

El futuro de las aplicaciones móviles

Como se ha hecho notar durante este trabajo, las aplicaciones móviles tienen un auge que será permanente durante bastante tiempo, esto debido a sus capacidades técnicas y sobre todo a la conectividad entre un gran número de ellos.

La tecnología Bluetooth deberá tener un mayor desarrollo en elementos como la seguridad y multi-conectividad que hasta el momento sólo se logra implementando herramientas externas de software pero que no es propia del estándar que define a esta tecnología.

El futuro de las aplicaciones web.

La web es el servicio de mayor apogeo en internet, su versatilidad permite tenerlo presente casi en cualquier ámbito, su capacidad de penetrar en todo tipo de comunidades ha facilitado su desarrollo. En los próximos años se verá una tendencia aún mayor respecto a la interacción con los usuarios finales y la información que éstos desean compartir.

El éxito de este servicio se debe en gran medida a los estándares y buenas prácticas para el desarrollo de aplicaciones, estas guías han permitido tener un gran alcance en los servicios que se ofrecen. Queda claro que el desarrollo web no es algo nuevo, mucho menos los estándares para su implementación, sin embargo mediante este trabajo se quiere hacer notar la importancia de seguir estas buenas prácticas con la única finalidad de lograr un trabajo notable.

Por último, debido a que anteriormente no existía algún precedente de aplicaciones similares en la comunidad universitaria, deseamos que este desarrollo pueda ser utilizado para crear o mejorar aplicaciones relacionadas con las tecnologías de la información y Bluetooth, si bien es cierto que en la Facultad de Ingeniería no existe un camino bien definido para los desarrollos de este tipo, es cierto también que los egresados de esta Facultad tienen la capacidad de aplicar sus conocimientos para innovar y/o mejorar procesos ya existentes con la finalidad de disminuir los recursos tanto económicos como humanos; deseamos de igual forma aportar una manera particular de trabajo, una que represente las buenas prácticas de laborar de los universitarios y que sea una característica sobresaliente en especial de la Facultad de Ingeniería de la UNAM.

Manual técnico

A.1. Introducción

El sistema es un conjunto de herramientas con la capacidad de enviar información de manera inalámbrica a dispositivos equipados con Bluetooth (Celulares, PDA, Laptops, etc.). Tiene la facilidad de controlar toda la información, generar los contenidos y mostrar reportes a través de una administración web amigable para el usuario.

A.2 Objetivo del sistema

El sistema tiene como objetivo entregar contenidos a un gran número de receptores haciendo uso de una herramienta de alto nivel a bajo costo tanto para su implementación como para los usuarios con dispositivos móviles.

Objetivos Específicos

Entre los principales objetivos que cubre el sistema se encuentran:

- Administración Web de alto nivel para la información: Usuarios, Dispositivos, Mensajes, Entregas.
- Automatizar los procesos involucrados con la manipulación del hardware de forma que su manipulación sea sencilla y transparente para el administrador.
- Proporcionar información (gráficas) relévate para los análisis necesarios como: Entregas, usuarios, dispositivos.

A.3 Contenido técnico**A.3.1 Reglas del negocio***Administrador*

- Cualquier modificación de información será posible sólo después de iniciar sesión.
- El registro de usuario deberá contemplar todos los campos del formulario, poniendo especial atención en la asignación del rol.
- La edición de información para un usuario contempla todos los campos del formulario, exceptuando su identificar único.
- El registro de los dispositivos deberá contemplar todos los campos y la información proporcionada por la sección "Información de los dispositivos" poniendo especial atención en los campos "Dirección Mac" y "Canal" debido a que su correcto registro hará posible la correcta comunicación entre el receptor y el equipo central.
- La edición de los dispositivos sólo contempla la modificación del campo "Usuario" debido a que la información del dispositivo no varía con el tiempo.
- El registro de los mensajes deberá contemplar todos los campos del formulario poniendo especial atención en las fechas de entrega y finalización de la campaña.
- La edición de mensajes contempla la modificación de la fecha de finalización de la campaña para ampliar o reducir su duración.
- Una vez que el administrador termina su sesión no será posible manipular cualquier tipo de información del sistema.

Sistema

- El sistema inicia y finaliza la campaña y todas las entregas correspondientes a un mensaje de acuerdo a las fechas de su registro.
- El sistema realiza automáticamente el escaneo en busca de dispositivos cercanos.
- El sistema envía las entregas cuando detecta al dispositivo considerando las fechas válidas para la campaña.
- El sistema actualiza automáticamente el estado de las entregas una vez que el receptor acepta o rechaza el archivo que se envía desde el equipo central, actualiza la información de igual forma cuando el dispositivo no responde a la petición.

- Una entrega toma el estado inactivo una vez que el usuario cancela la recepción del archivo un par de ocasiones.

Receptor

- El dispositivo del usuario es detectado por el sistema siempre y cuando se encuentre dentro del área de cobertura.
- El dispositivo estará activo en la base de datos si y sólo si cuente con el servicio de transferencia de archivos.
- Una vez que el sistema detecta el dispositivo buscará alguna entrega pendiente y en caso afirmativo recibe una petición de entrega.
- El receptor tiene la posibilidad de aceptar o rechazar la entrega y el sistema se encargará de registrar su respuesta.
- De acuerdo a la respuesta del punto anterior, el sistema determinará una prioridad para la entrega, es decir, si el receptor cancela en dos ocasiones su entrega correspondiente, automáticamente se cancela. En caso contrario si la recibe exitosamente, ésta entrega no volverá a ser enviada.

A.3.2 Diagramas de flujo de datos

Elementos del Diagrama de Flujo de Datos (DFD)⁸

A continuación se describen cada uno de los elementos del DFD:

Flujo de dato: Representación del flujo de dato de origen al destino (Figura A.1).



Figura A.1 Flujo de dato

Función: Transformación del flujo de datos de entrada en la salida (Figura A.2).



Figura A.2 Función

Almacenamiento: Son los datos pasivos, comúnmente tablas, etc. (Figura A.3).



Figura A.3 Almacenamiento

⁸ Murguía Gutierrez, Edgar, Elementos del DFD, <http://www.fismat.umich.mx/~emurgia/mipagina/tesis/node44.html> , 07 de Agosto de 2009.

- Inicio de sesión (Figura A.4)



Figura A.4 Inicio de sesión

El inicio de sesión proporciona acceso a los elementos para la administración de la información y los datos de acceso son proporcionados por el meta administrador en un registro previo.

- Administración de usuarios (Figura A.5)

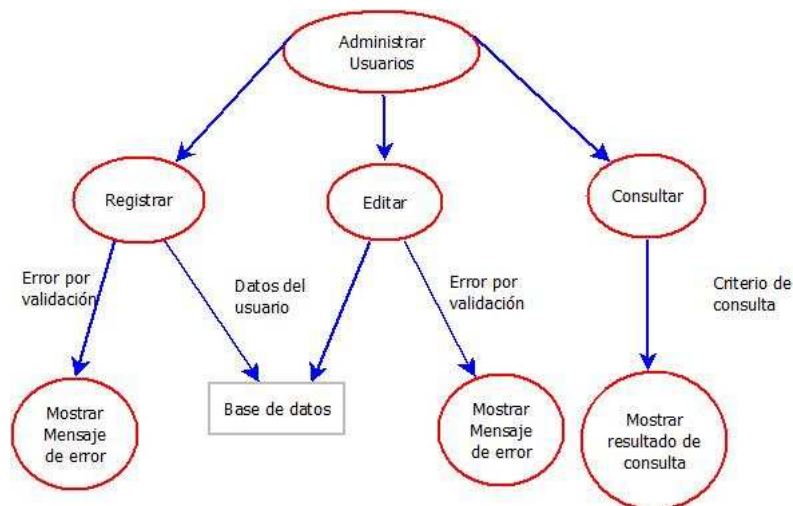


Figura A.5 Administración de usuarios

Esta sección permite la manipulación de la información relacionada con el usuario. El registro y la edición se controlan a partir de validadores que verifican el contenido de los campos. La consulta se realiza de manera sencilla haciendo uso de los filtros de búsqueda.

- Administración de mensajes (Figura A.6)

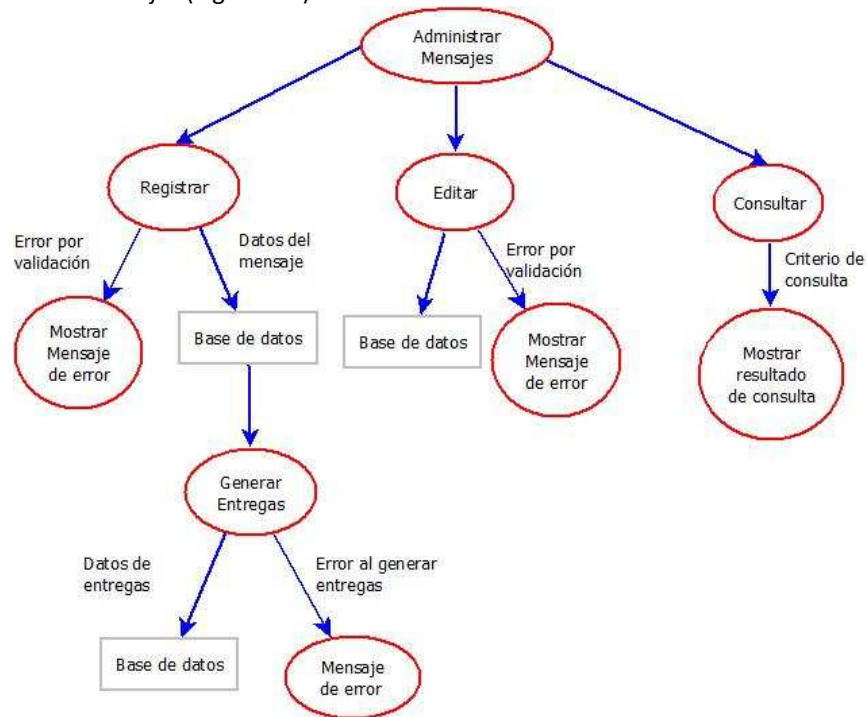


Figura A.6 Administración de mensajes

La administración de mensajes facilita la creación de las entregas debido a que sólo es necesario crear el contenido y asignar la fechas ya que el sistema automatiza las tareas de envío y entrega así como el control de prioridad.

- Administración de Reportes (Figura A.7)

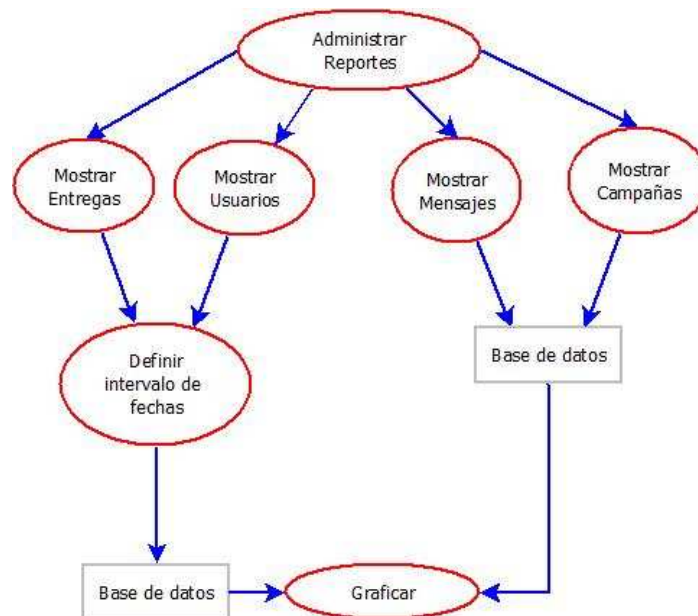


Figura A.7 Administración de reportes

La sección de reportes es útil para analizar a partir de gráficas la información en la base de datos. En la sección “Campañas” existe una grafica con una serie de columnas donde se muestra el comportamiento de un mensaje, datos como número de entregas o cancelaciones.

- Administración de dispositivos (Figura A.8)

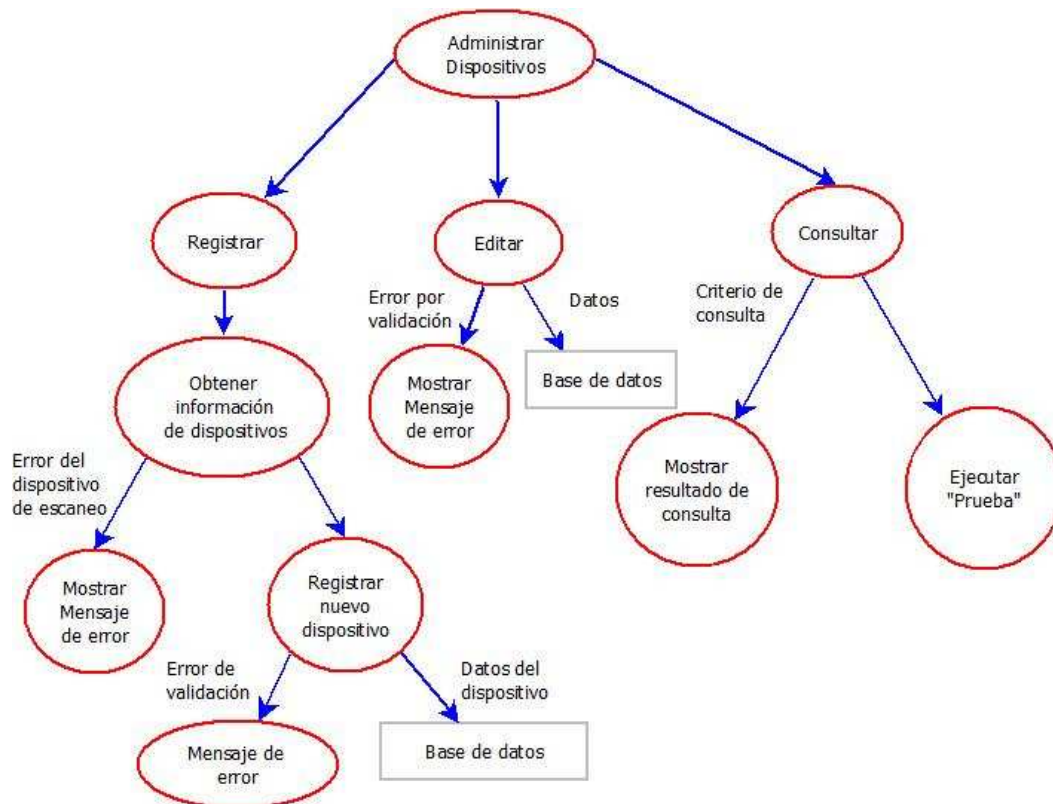


Figura A.8 Administración de dispositivos

La administración de dispositivos tiene como finalidad proporcionar al administrador información para el registro de los aparatos de tal forma que el sistema central genere un enlace y así pueda comunicarse. Una utilidad importante dentro de la consulta es la herramienta “Prueba” ya que en cualquier momento se puede ejecutar para enviar un contenido al dispositivo para comprobar la correcta comunicación.

A.3.3 Campos requeridos

Apartado Diseño: Diseño de base de datos

A.3.4 Requerimientos de interface con otros sistemas

- *Utilidades Bluetooth*

El sistema trabaja en conjunto con el paquete de herramientas BlueZ dentro de la distribución Debian de Linux (“Desarrollo – Requerimientos de Software”). La interacción se realiza a través de scripts PHP que toman la información de la ejecución de tal conjunto de herramientas. A continuación se describen los casos:

- *Descubrimiento de dispositivos:* El dispositivo central realiza un escaneo en busca de dispositivos cercanos, una vez con la información procesada, un script php actualiza la información de la base de datos.
- *Envío de contenido:* Una vez que el dispositivo encuentra algún dispositivo que tiene una entrega pendiente, realiza el envío el cual se obtiene de la carpeta contenedora de los archivos.

Como se comentó en el punto “Resumen hciconfig” del apartado “Herramientas Bluetooth en GNU/Linux” del capítulo 4, después de un reinicio es posible que la configuración vuelve a su estado de inicio por lo que es necesario asignarle las propiedades correctas de nuevo, en este caso es necesario ejecutar una tarea desde la terminal del sistema:

```
$MovilFi#: hciconfig piscan
```

De esta forma el dispositivo trabajará de forma adecuada nuevamente.

- *Base de datos*
 - La interacción entre la base y la administración web está soportada por el framework por lo que es transparente a nivel programación, es así que la única tarea que debe ejecutarse es la creación de la base y la asignación de permisos. El detalle de los requerimientos se muestra en el apartado “Desarrollo – Construcción de la base de datos”.
 - Para iniciar la operación del sistema se deben considerar algunos datos base para su correcto funcionamiento, la tarea se detalla en “Desarrollo – Construcción de la base de datos (Carga de datos)”.
- *ImageMagick*
 - La creación de las entregas está directamente ligada a la correcta operación de la biblioteca ImageMagick (“Desarrollo – Requerimientos de Software”). Después de su instalación el sistema configura la herramienta de forma automática.

A.3.5 Modelo lógico de datos (Figura A.9)

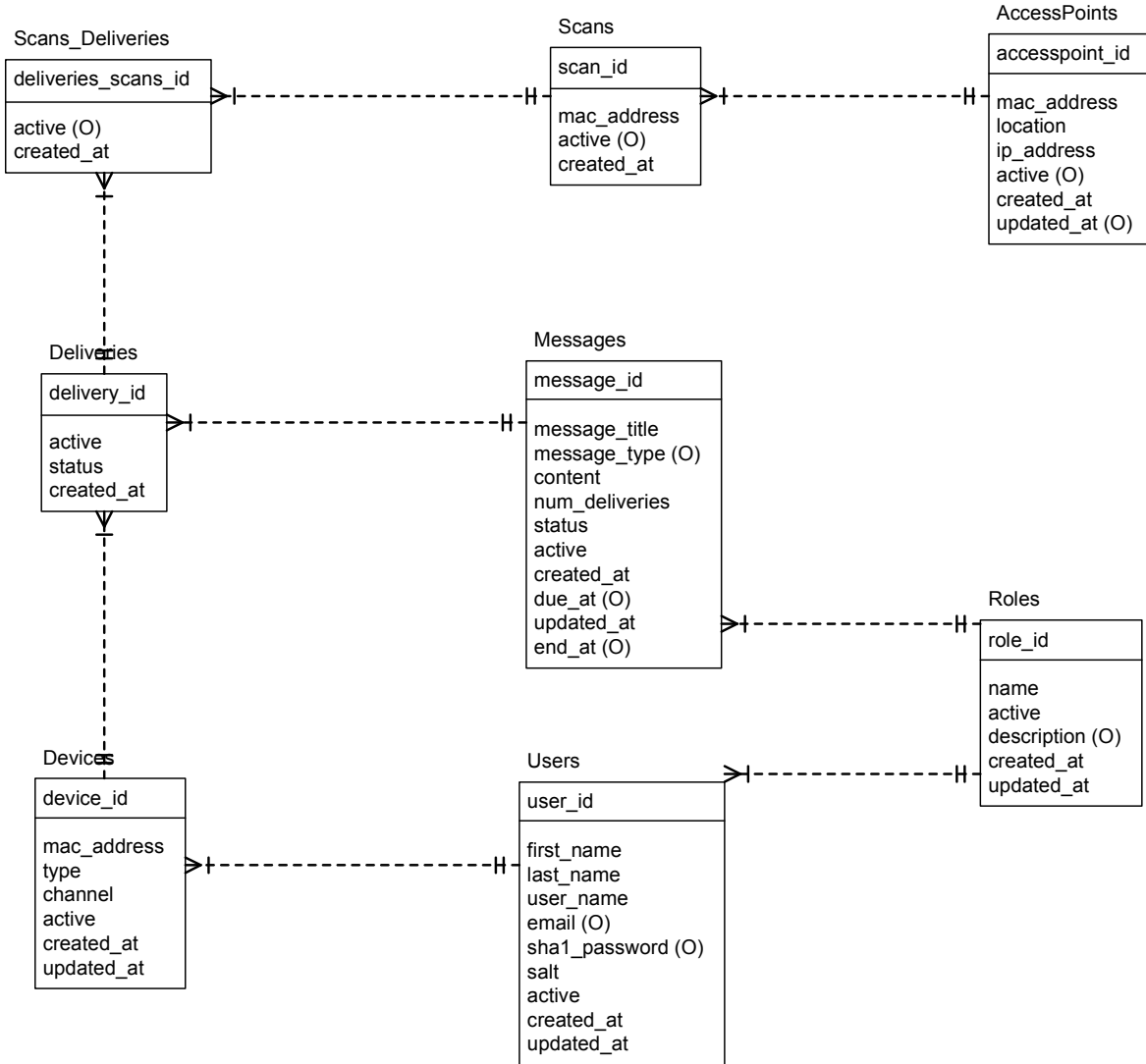


Figura A.9 - Modelo lógico de datos.

A.3.6 Modelo físico de datos (Figura A.10)

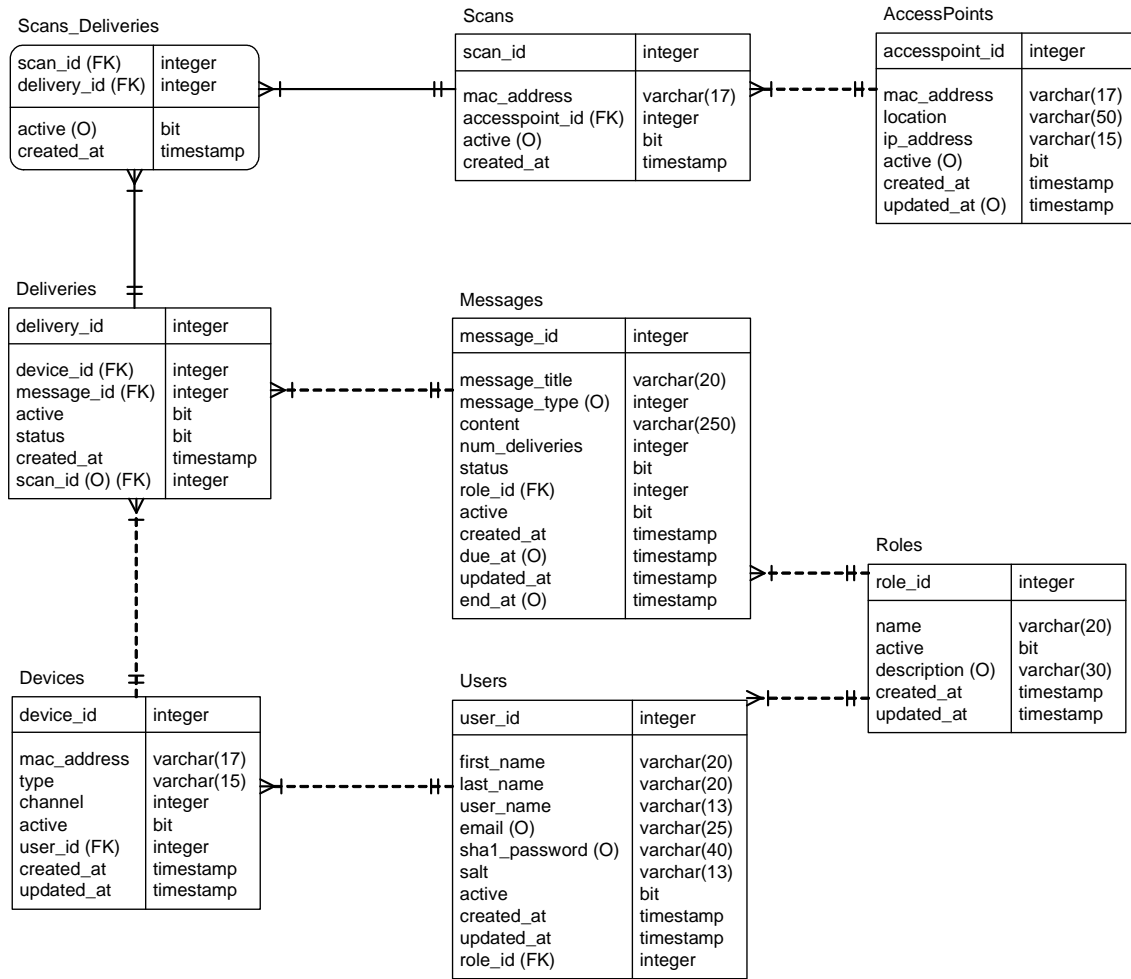


Figura A.2 - Modelo físico de datos

A.3.7 Áreas de aplicación

El sistema realizará correctamente cada una de sus tareas en espacios abiertos en entornos cerrados (bibliotecas, salas de conferencias, etc). Su rendimiento será el máximo cuando el sistema se implemente en este tipo de ubicaciones.

Consideraciones:

- La intensidad de la señal y por tanto el área de cobertura se verán afectadas por interferencias causadas por celulares, antenas WiFi y dispositivos de microondas. De igual forma los muros de gran tamaño disminuirán su rendimiento debido a la interferencia entre la antena y el dispositivo móvil.
- Es recomendable dirigir la antena hacia un lugar visible y de fácil acceso para los receptores.
- El tiempo de entrega de los contenidos variará de acuerdo al peso del archivo así como de la distancia existente entre la antena y el dispositivo.

A.4. Mapa de navegación (Figura A.11)

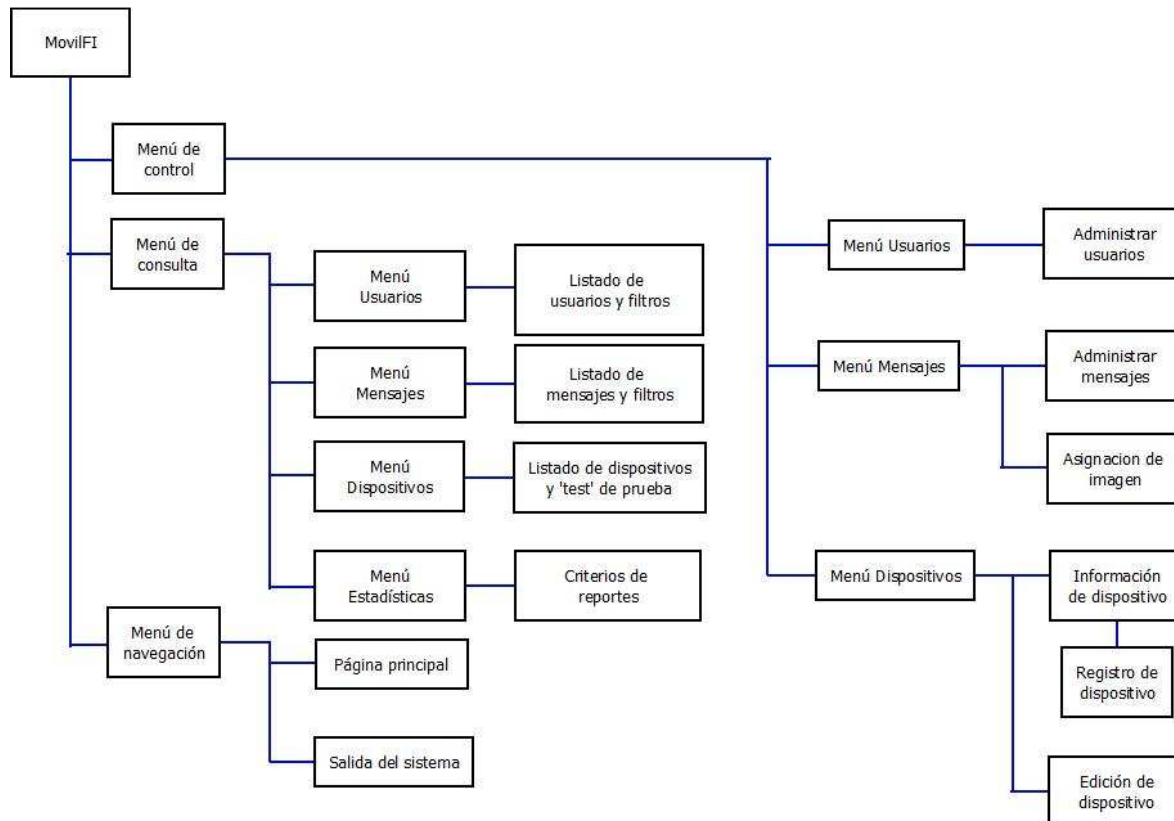


Figura A.11 Mapa de navegación

A.4.1 Descripción gráfica

Inicio de sesión.

La siguiente será la interfaz de acceso al sistema (Figura A.12). La única precondition, será haber registrado en el sistema al administrador para poder iniciar sesión.

Figura A.12 - Inicio de sesión

Menú principal

Una vez iniciada sesión, se mostrará la siguiente página (Figura A.13) con dos menús principales.

1. Menú de navegación (A1).
2. Menú de administración (A2).

Se mostrará también la administración de mensajes como sección principal. (A3)

Figura A.13 - Menú principal

A continuación se muestra cada uno de los elementos del menú y una descripción de las funciones asociadas.

Usuarios

El menú “Usuarios” en la sección “Crear” (B1) tiene como finalidad el registro de usuario al sistema. Se muestra el formulario (B2) y las operaciones necesarias (Figura A.14).

Frontend | Salir

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

↑ B1

Nuevo Usuario

Nombre

Apellido

Usuario

Correo

Password

Role

Activo

"2009 Blue. Todos los derechos Reservados. LADAI"

← B2

Figura A.14 - Nuevo usuario

Mensajes

El menú “Mensajes” en la sección “Crear” (C1, Figura A.15), tiene como finalidad el registro de los mensajes (C2, Figura A.16) y asignación de contenido (C3, Figura A.17).

Frontend | Salir

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

↑ C1

Lista de Mensajes

Mensaje	Titulo del mensaje	Contenido	Entregas	Imagen	Tipo de usuario	Activo	Creado	Entrega	Actualizado	Finalizacion
0 mensajes (1 al 0)										

[Nuevo](#)

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.15 – Mensajes

Frontend | Salir

Crear: Usuarios | Mensajes | Dispositivos | Consultar: Usuarios | Mensajes | Dispositivos | Estadísticas

Frontend Blue

Nuevo Mensaje

Titulo del mensaje

Contenido ← C2

Num deliveries

Tipo de usuario

Entrega / / : :
mes(mm)/dia(dd)/año(aaaa)

Finalizacion / / : :

Miércoles, 26 de Agosto de 2009

Figura A.16 - Nuevo mensaje

Frontend | Salir

Crear: Usuarios | Mensajes | Dispositivos | Consultar: Usuarios | Mensajes | Dispositivos | Estadísticas

Frontend Blue



CyT 1



CyT 2



CyT 3

← C3

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.17 -Selección de Figura

Una vez creado el mensaje y asignada la plantilla, se desplegará una muestra del contenido (Figura A.18).

Frontend | Salir

Crear: Usuarios | Mensajes | Dispositivos | Consultar: Usuarios | Mensajes | Dispositivos | Estadísticas

Frontend Blue

Contenido de prueba para el sistema abierto



[Regresar](#)

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.18 - Vista de la Figura generada

Dispositivos

El menú “Dispositivos” en la sección “Crear” (D1, Figura A.19) tiene como finalidad el registro de dispositivos en sistema. Se muestra el formulario (D2, Figura A.19) y una sección para obtener la información de los dispositivos cercanos a registrarse (D3, Figura A.19).



Figura A.11 - Nuevo Dispositivo

Consulta: Usuarios

El menú “Usuarios” en la sección “Consultar” (E1) mostrará diferentes criterios de búsqueda (E2) para agilizar la consulta de un usuario en particular (Figura A.20).

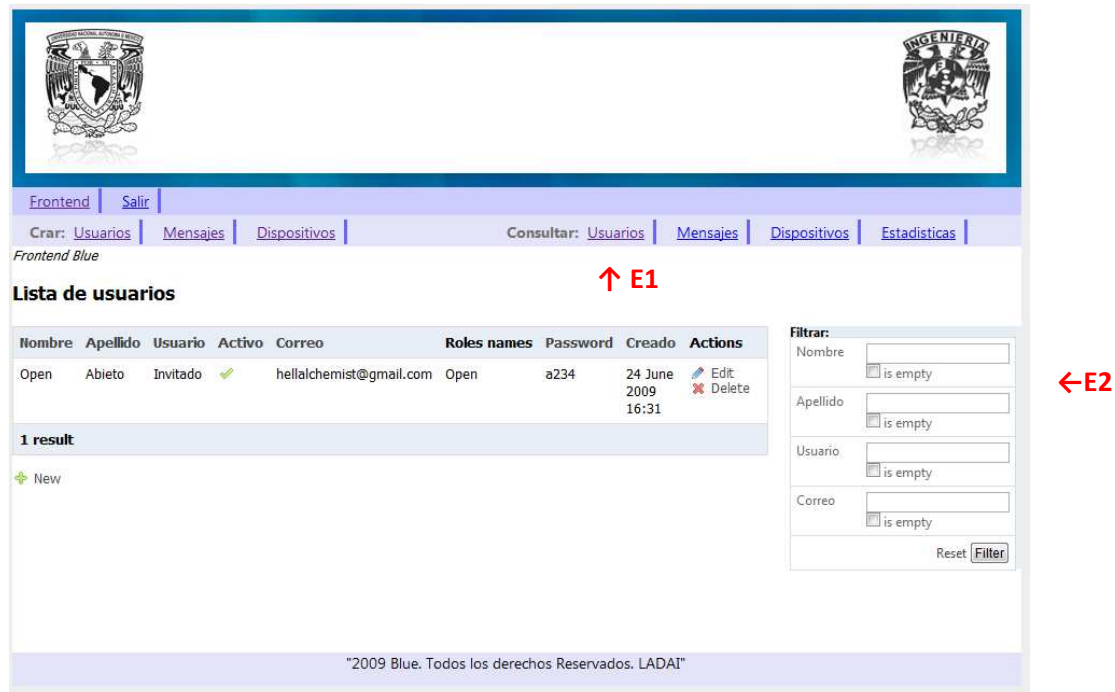


Figura A.20 - Consultar usuarios

Consulta: Dispositivos

El menú "Dispositivos" en la sección "Consultar" (F1) mostrará un listado de todos los dispositivos registrados así como al usuario al que pertenece. Dentro del listado existe la opción "Prueba" (F2) que tiene como finalidad realizar un envío automático con un contenido de prueba para comprobar la correcta comunicación entre el sistema y el dispositivo (Figura A.21).

Frontend | Salir

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

↑ F1

Lista de dispositivos

Dispositivo	Usuario	Direccion Mac	Tipo	Canal	Activo	Creado	Actualizado	Acción
1	Invitado	IN:VI:TA:DO:OP:EN	PDA	1	1	2009-06-24 16:31:40	2009-06-24 16:31:40	prueba

← F2

1 dispositivos (1 al 1)

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.21 - Consultar dispositivos

Reportes

El menú "Estadísticas" en la sección "Consultar" (G1) mostrará diferentes criterios (G2) para generar reportes de la información contenida en la base (Figura A.22).

Frontend | Salir

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

↑ G1

[Entregas](#)

[Mensajes](#) ← G2

[Usuarios](#)

[Campanias](#)

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.22 - Consultar estadísticas

Reportes: Entregas

La sección "Entregas" mostrará una gráfica con el número de entregas realizadas durante el rango de tiempo definido con la fecha de "Inicio" y de "Fin" (H1, Figura A.23).

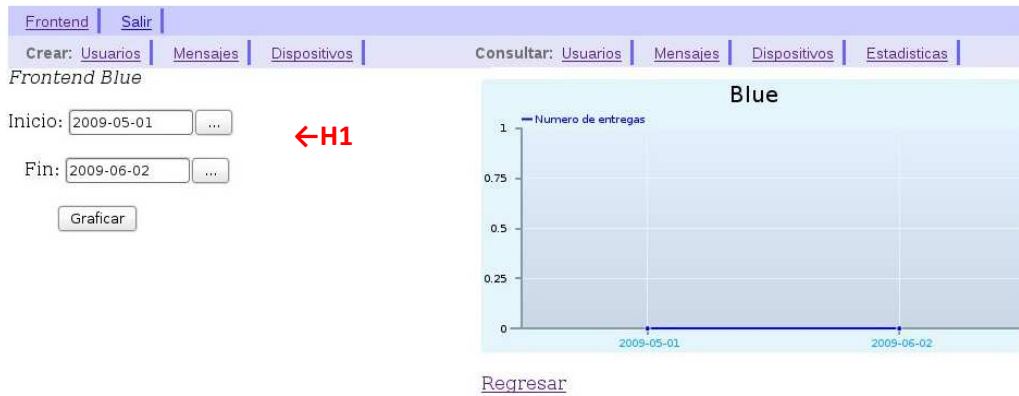


Figura A.23 - Reporte de entregas

Reportes: Mensajes

La sección “Mensajes” mostrará una gráfica con el número de mensajes realizados cada mes durante el año actual (Figura A.24).



Figura A.24 - Reporte de mensajes

Reportes: Usuarios

La sección “Usuarios” mostrará una gráfica con el porcentaje de usuarios registrados en el sistema de acuerdo al rol asignado y al rango de fechas (Figura A.25).



Figura A.25 - Reporte de usuarios

Reportes: Campañas

La sección “Campañas” mostrará una lista con cada una de las campañas registradas en el sistema (J1, Figura A.26). Cada una de estas mostrará a su vez una gráfica de barras (J2, Figura A.27)

Frontend | **Salir**

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

Campania

[Mensaje de Prueba](#)

[Bienvenida](#) ← J1

[Regresar](#)

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.26 – Campañas

Frontend | **Salir**

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

Numero entregas

← J2

Graficar

[Regresar](#)

Figura A.27 - Grafica por campaña

Cerrar sesión.

Para cerrar sesión en el sistema es necesario ejecutar la acción asociada al elemento “Salir” (K1, Figura A.28) del menú de navegación.

Frontend | **Salir** ←K1

Crear: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | Consultar: [Usuarios](#) | [Mensajes](#) | [Dispositivos](#) | [Estadísticas](#)

Frontend Blue

Lista de Mensajes

Mensaje	Título del mensaje	Contenido	Entregas	Imagen	Tipo de usuario	Activo	Creado	Entrega	Actualizado	Finalizacion
										0 mensajes (1 al 0)

[Nuevo](#)

"2009 Blue. Todos los derechos Reservados. LADAI"

Figura A.28 - Cerrar sesión

A.4.2 Descripción de procesos

La siguiente descripción se realizará a través de un formato donde se identifiquen las características de los elementos principales así como las acciones que en conjunto ejecuten.

Inicio de sesión.

Los componentes del formulario de inicio de sesión (Figura A.12) se muestran a continuación:

Elemento	Tipo	Acción	Observaciones
Nombre de usuario	Cuadro de texto	Iniciar sesión	
Password	Cuadro de texto		
Remember	CheckBox		Útil para iniciar sesión y no volver a capturar los datos
Sign in	Botón		

Tabla A.1 Componentes de formulario de inicio de sesión

Página principal

Menú de navegación (Figura A.13):

Elemento	Tipo	Acción	Observaciones
Frontend	Hipervínculo	Mostrar página principal	
Salir	Hipervínculo	Cerrar sesión	

Tabla A.2 Componentes del menú de navegación

Menú de administración (Figura A.13):

Elemento	Tipo	Acción	Observaciones
Usuarios	Hipervínculo	Mostrar formulario de usuarios	Creación
Mensajes	Hipervínculo	Mostrar administración de mensajes	
Dispositivos	Hipervínculo	Mostrar administración de dispositivos	
Usuarios	Hipervínculo	Mostrar listado de usuarios y filtros de búsqueda	Consulta
Mensajes	Hipervínculo	Mostrar listado de mensajes y filtros de búsqueda	
Dispositivos	Hipervínculo	Mostrar listado de dispositivos y "test" de prueba para comunicación	
Estadísticas	Hipervínculo	Muestra las opciones para la generación de reportes	

Tabla A.3 Componentes del menú de administración

Usuarios

El registro de usuarios muestra el formulario con los campos necesarios (Figura A.14)

Elemento	Tipo	Acción	Observaciones
Nombre	Cuadro de texto		Nombre del usuario
Apellido	Cuadro de texto		Apellido
Usuario	Cuadro de texto		Identificar del usuario "nickname"
Correo	Cuadro de texto		Correo electrónico del usuario
Password	Cuadro de texto		Identificador elegido por el usuario
Role	Select		Grupo u organización a la que pertenece el usuario
Activo	CheckBox		Bandera que identifica el estado de un usuario: Activo o Inactivo

Elemento	Tipo	Acción	Observaciones
Cancel	Hipervínculo	Cancelar el registro actual y mostrar un listado de los usuarios actuales.	
Save	Botón	Guardar el registro actual	
Save and Add	Botón	Guardar el registro actual ya agregar uno nuevo	

Tabla A.4 Componentes del formulario de nuevo usuario

La consulta de usuario muestra un listado de los registros actuales y un conjunto de opciones para filtrar las búsquedas (Figura A.20)

Elemento	Tipo	Acción	Observaciones
Listado de datos	Tabla	Mostrar en orden los registro de la base	
Edit	Hipervínculo	Mostrar formulario para la edición del registro	El formulario de edición mostrará una sección con los dispositivos que el usuario tiene registrado en el sistema. La sección se puede identificar cómo : <i>Dispositivos del usuario Nombre_Usuario</i>
Delete	Hipervínculo	Eliminar el registro actual	Al ejecutar la acción, se mostrará una ventana preguntando si se está de acuerdo con la eliminación del registro.
Nombre	Cuadro de texto	Filtrar la búsqueda por nombre de usuario.	Es posible utilizar operadores para manipular cadenas y generalizar la búsqueda. Uno de ellos es el operador %, que tiene la siguiente estructura: %palabra_cicompleta% - Buscará en el campo indicado la cadena incompleta, por ejemplo: %niel% encontrará todas los nombres que tengan en su cuerpo, la secuencia de caracteres 'niel'.
Apellido	Cuadro de texto	Filtrar la búsqueda por Apellido	
Usuario	Cuadro de texto	Filtrar la búsqueda por Usuario	
Correo	Cuadro de texto	Filtrar la búsqueda por Correo	
Reset	Hipervínculo	Eliminar los criterios de búsqueda y mostrar listado	
Filter	Botón	Realizar la búsqueda de acuerdo al criterio seleccionado.	

Tabla A.5 Componentes del formulario de consulta de usuarios

Mensajes

El registro de mensajes muestra el formulario con los campos necesarios (Figura A.15, A.16)

Elemento	Tipo	Acción	Observaciones
Título del mensaje	Cuadro de texto		Se mostrará un mensaje de advertencia cuando el título sea demasiado corto (2 caracteres) o demasiado extenso (35 caracteres)
Contenido	Cuadro de texto		Se mostrará un mensaje de advertencia cuando el contenido sea demasiado corto (2 caracteres) o demasiado extenso (125 caracteres)
Número de entregas	Cuadro de texto		Campo numérico que limita el número de entregas por registro
Tipo de usuario	Select		El tipo de usuario indica hacia que usuario va dirigido el mensaje, por ejemplo: profesor o alumno
Entrega	Select		La entrega del registro será activada automáticamente por el

			sistema de acuerdo a la fecha de entrega.
Finalización	Select		El sistema automáticamente finalizará el envío de entregas para el registro una vez que se haya alcanzado la fecha de finalización
Cancel	Hipervínculo	Cancelar el registro actual y mostrar un listado de los mensajes	
Save	Botón	Guardar el registro actual	

Tabla A.6 Componentes del formulario de mensajes

Una vez creado el mensaje es necesario asignarle una plantilla para crear todas las entregas y comenzar el envío (Figura A.17)

Elemento	Tipo	Acción	Observaciones
Plantillas	CheckBox		
Generar Figura	Botón	Crear todas la entregas de ese mensaje y asociarlas a todos los usuarios de acuerdo a su rol	Todas las entregas generadas (archivos) se encuentran dentro de la carpeta “web/images/envíos”

Tabla A.7 Componentes de selección de plantilla

Finalmente se desplegará en una página nueva una muestra de la entrega (Figura A.18)

Elemento	Tipo	Acción	Observaciones
Muestra	Figura		El contenido de la entrega es: Plantilla, mensaje e identificar del usuario
Regresar	Hipervínculo		Vínculo a la página principal.

Tabla A.8 Componentes de la vista previa de la Figura generada

Dispositivos

El registro de dispositivos muestra el formulario con los campos necesarios (Figura A.19)

Elemento	Tipo	Acción	Observaciones
Dirección Mac	Cuadro de texto		La información para este campo se mostrará en la sección “ INFORMACIÓN DE LOS DISPOSITIVOS”
Type	Select		Muestra una lista con los tipos de dispositivos para registrar
Channel	Select		Muestra una lista con los canales para la comunicación entre el dispositivo y el sistema. La información para este campo se mostrará en la sección “ INFORMACIÓN DE LOS DISPOSITIVOS
Activo	CheckBox		Indica el estado del dispositivo
Usuario	Cuadro de texto		El cuadro de texto tiene la funcionalidad de “Auto complementar” el nombre de el usuario que previamente se registró en el sistema
Cancel	Hipervínculo		Cancelar el registro actual y mostrar el listado de los dispositivos actuales
Save	Botón		Realiza la acción de registrar el nuevo dispositivo con la información del formulario
INFORMACIÓN DE LOS DISPOSITIVOS	Frame		Mostrará el resultado en forma ordenada de la ejecución del escaneo.
Información	Hipervínculo		Ejecuta un escaneo en busca de información de los dispositivos cercanos para ser registrados

Tabla A.9 Componentes del formulario de dispositivos

La consulta de dispositivos muestra un listado de los registros actuales y un conjunto de opciones para filtrar las búsquedas (Figura A.21)

Elemento	Tipo	Acción	Observaciones
Listado	Tabla	Mostrar en orden los registro de la base	Cuando el listado de exceda 10 registros se mostrará un paginador para facilitar la navegación y ubicación de los dispositivos.
Acción: Prueba	Hipervínculo	Enviar al dispositivo asociado a ese registro un archivo de prueba para comprobar la correcta comunicación entre el sistema y el dispositivo	Al terminar esta operación se mostrará un mensaje confirmado el envío.
Dispositivo	Hipervínculo	El hipervínculo está asociado al detalle del registro y al formulario de edición del dispositivo.	Es posible utilizar operadores para manipular cadenas y generalizar la búsqueda. Uno de ellos es el operador %, que tiene la siguiente estructura: %palabra_cicompleta% - Buscará en el campo indicado la cadena incompleta, por ejemplo: %niel% encontrará todas los nombres que tengan en su cuerpo, la secuencia de caracteres 'niel'.
Usuario	Tabla Columna	A partir de este campo se genera un criterio de ordenamiento para el listado de los dispositivos	El ordenamiento de los registro es de forma descendente a partir de la letra con que inicie el nombre de usuario desde la A y hasta la Z

Tabla A.10 Componentes de menú de estadísticas

Reportes

Los elementos de la sección de estadísticas son (Figura A.22):

Elemento	Tipo	Acción	Observaciones
Entregas	Hipervínculo	Mostrar elementos a partir de las entregas.	
Mensajes	Hipervínculo	Mostrar una gráfica a partir de los mensajes.	
Usuarios	Hipervínculo	Mostrar una gráfica a partir de los usuarios.	
Campañas	Hipervínculo	Mostrar una gráfica a partir de las campañas.	

Tabla A.11 Componentes de menú de estadísticas

Los componentes de la sección de entregas son (Figura A.23):

Elemento	Tipo	Acción	Observaciones
Inicio	Calendario	Mostrar calendario para seleccionar la fecha de inicio para la consulta.	
Fin	Calendario	Mostrar calendario para seleccionar la fecha de fin para la consulta.	
Graficar	Botón	Da la consulta con la fecha de inicio y fin, gráfica con datos que obtiene de la base de datos.	
Regresar	Hipervínculo	Asociado al menú principal de reportes	
Gráfica			Al pasar el apuntador sobre la gráfica esta muestra los valores de los puntos de interés

Tabla A.12 Componentes de los reportes de las entregas

Los componentes de la sección de mensajes son (Figura A.24):

Elemento	Tipo	Acción	Observaciones
Regresar	Hipervínculo	Asociado al menú principal de reportes	
Gráfica			Al pasar el apuntador sobre la gráfica esta muestra los valores de los puntos de interés

Tabla A.13 Componentes de los reportes de mensajes

Los componentes de la sección de usuarios son (Figura A.25):

Elemento	Tipo	Acción	Observaciones
Inicio	Calendario	Mostrar calendario para seleccionar la fecha de inicio para la consulta.	
Fin	Calendario	Mostrar calendario para seleccionar la fecha de fin para la consulta.	
Graficar	Botón	Da la consulta con la fecha de inicio y fin, gráfica con datos que obtiene de la base de datos.	
Regresar	Hipervínculo	Asociado al menú principal de reportes	
Gráfica			Al pasar el apuntador sobre la gráfica esta muestra los valores de los puntos de interés

Tabla A.14 Componentes de los reportes de usuarios

La sección de campañas muestra un listado de cada uno de los mensajes registrados (Figura A.26)

Elemento	Tipo	Acción	Observaciones
Campaña	Tabla	Muestra cada uno de los mensajes registrados en la base de datos	
Regresar	Hipervínculo	Asociado al menú principal de reportes	

Tabla A.15 Listado de las campañas

Los componentes para cada una de las campañas son (Figura A.27):

Elemento	Tipo	Acción	Observaciones
Gráfica	Columna		Al pasar el apuntador sobre la gráfica esta muestra los valores de los puntos de interés
Entregas	Columna	Muestra el número de entregas enviadas y recibidas por el usuario correctamente	
Campaña terminada	Columna	Muestra el número de entregas no enviadas debido a la finalización de la campaña	
En Espera	Columna	Muestra en el número de entregas no enviadas dada la fecha de finalización pero que se crearon al momento de registro del mensaje	
Cancelado	Columna	Muestra el número de entregas enviadas correctamente pero canceladas por el usuario	La cancelación se considera una vez que el usuario no acepta la recepción del archivo o no responde a la petición en un par de ocasiones
Regresar	Hipervínculo		Asociado al menú principal de campañas

Tabla A.16 Componentes de la grafica

Cerrar sesión.

El elemento se encuentra en el menú de navegación (Figura A.28):

Elemento	Tipo	Acción	Observaciones
Salir	Hipervínculo	Cerrar sesión del administrador del sistema.	Por seguridad cuando el sistema no realiza ninguna acción después de cierto tiempo, la sesión se cierra automáticamente por lo que es necesario volver a ingresar para continuar.

Tabla A.17 Componente para cerrar sesión

Glosario de términos

<u>Término</u>	<u>Definición</u>
ACL	Asynchronous Connection Oriented - Conexión Orientada
Alcance	Rango de distancia entre los dispositivos para la comunicación.
ANSI/SPARC	ANSI-SPARC (American National Standards Institute, Standards Planning And Requirements Committee – Instituto Nacional de Normalización Estadounidense, Comité de Planeación de Estándares y Requerimientos)
API	Interfaz que define la forma como un programa de aplicación solicita servicios de bibliotecas. Puede incluir especificaciones para rutinas, estructura de datos, clases de objetos y protocolos.
Cabeceras	Es uno de los componentes de un paquete, aquí se encuentran los campos necesarios para el protocolo a nivel de red (Dirección de las estaciones de origen y destino del identificador del circuito virtual)
Capa de enlace	Es la capa número dos del Modelo OSI, se ocupa del direccionamiento físico, de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control de flujo.
Censado	Escaneo del dispositivo central en busca de otros dispositivos móviles.
Cifrado	Es el proceso de ocultar información con la finalidad de hacerlo ilegible para cualquier persona no relacionada con la fuente o el destino de un mensaje.
Circuito virtual	Sistema de comunicación por el cual los datos de un usuario origen pueden ser transmitidos a otro usuario destino a través de más de un circuito de comunicaciones real durante un cierto periodo de tiempo.
Curvas de Bezier	Sistema que se desarrolló hacia los años 1960, para el trazado de dibujos técnicos, en el diseño aeronáutico y de automóviles. Su denominación es en honor a Pierre Bezier, quien ideó un método de descripción matemática de las curvas que se comenzó a utilizar con éxito en los programas de CAD (Diseño asistido por Computadora).
Datos offline	No es necesaria una conexión para obtener información
Dirección IP	Es un identificador lógico de un dispositivo (comúnmente una computadora) dentro de una red que implementa el protocolo IP (Internet Protocol)
Dirección MAC	Identificador único de cada dispositivo móvil.
EDGE	Enhanced Data Rates for Global Evolution (Tasas de Datos Mejoradas para la evolución de Global)
Emparejar	Comunicar un dispositivo nuevo intercambiando una clave con el sistema para poder transferir archivos. *La operación se realiza una sola vez.

Espectro	En el campo de la física se define como una gama de frecuencias.
Ethernet	Estándar de redes de computadoras de área local
Facsímil	Un facsímil, también llamado facsímile, es una copia o reproducción muy precisa, casi perfecta, de un documento.
Filtro	Elemento que separa una determinada frecuencia o conjunto de frecuencias de una señal eléctrica que pasa por él.
Framework	Estructura definida y organizada para el desarrollo de software.
FTP	File Transfer Protocol - Protocolo para la transferencia de archivos.
GPL	General Public License.- Licencia creada por la Free Software Foundation orientada a proteger la libre distribución, modificación y uso de software.
GPRS	General Packet Radio Services - Servicio General de Paquetes vía Radio
HTML	HyperText Markup Language (Lenguaje de Marcas de Hipertexto)
IEEE	The Institute of Electrical and Electronics Engineers, una asociación técnico-profesional dedicada a la creación de estándares de tecnologías actuales y nuevas.
IPSec	Protocolo de Seguridad de Internet: Es un protocolo para la seguridad de las comunicaciones del Protocolo de Internet (PI), que se apoya en la autenticación y cifrado de los paquetes PI de un flujo de datos.
ISDN	Integrated Services Digital Network - Red Digital de Servicios Integrados
Keypad	Bloque de botones que contienen regularmente dígitos sin considerar el conjunto completo del alfabeto.
L2CAP	Logical Link Control and Adaptation Protocol
LAMP	Conjunto de aplicaciones formado por: Linux, Apache, MySQL, PHP.
Licencia BSD	Licencia Berkeley Software Distribución.- Licencia de software libre permisiva y al contrario que la GPL permite el uso del código fuente en software no libre.
Licencia MIT	Licencia Massachusetts Institute of Technology - Licencia que permite reutilizar el Software así licenciado tanto para ser software libre como para ser software no libre, permitiendo no liberar los cambios realizados al programa original.
Mainframe	Computadoras utilizadas comúnmente por grandes organizaciones para almacenar información crítica, procesar grandes volúmenes de información y/o realizar grandes transacciones financieras.
Modelo OSI	Es el modelo de referencia de interconexión de sistemas abiertos.
OBEX	OBject Exchange (Intercambio de Datos)
PGP	Herramienta útil para firmar, cifrar y descifrar información para incrementar la seguridad en comunicaciones.

PIN	Personal Identification Number (Número de Identificación Personal)
PLMN	Public Land Mobile Network (Red móvil terrestre pública)
Plugin	Programa desarrollado de forma independiente que ejecuta una tarea específica.
PSTM	Public Switched Telephone Network (La red telefónica pública conmutada)
RFCOMM	Radio Frequency Communication Comunicación por Radio Frecuencia
SCO	Synchronous Connection Oriented – Conexión Orientada Síncrona
Script	Conjunto de instrucciones que se ejecutan secuencialmente.
SDP	Service Discovery Protocol (Protocolo de Servicio de Descubrimiento)
SMS	Short Message Service (Servicio de Mensajes Cortos)
Spam	Mensajes no solicitados de forma directa, con contenido comúnmente publicitario y que es enviado en grandes cantidades.
SSH	Protocolo de red que permite el intercambio de información a través de un canal seguro entre dos dispositivos
SSL	Secure Sockets Layer: Es un protocolo de cifrado que proporciona seguridad en comunicaciones en redes como internet.
Stylus	Accesorio de escritura, generalmente PDA (Personal Digital Assistant -Asistente Digital Personal) o teléfono inteligente.
Task	Tarea realizada por el sistema y que se ejecuta desde la línea de comandos bajo un entorno de desarrollo del framework symfony.
Template	Plantilla que forma parte de la vista en la arquitectura MVC.
Triggers	Disparador en una base de datos - Es un procedimiento previamente definido que se ejecuta cuando se cumple cierta condición al tener una operación INSERT, UPDATE o DELETE.
UIT-T	La Unión Internacional de Telecomunicaciones (UIT) es el organismo especializado de las Naciones Unidas encargado de regular las telecomunicaciones, a nivel internacional, entre las distintas administraciones y empresas operadoras
UUID	Universally Unique Identifiers – Identificadores Universales Únicos

Bibliografía

Debian, Debian Reference, <http://www.debian.org/doc/manuals/debian-reference>, fecha de consulta: 4 de Agosto de 2009.

Elmasri y Navathe, *Sistemas de Bases de datos, Conceptos Fundamentales*, Segunda edición, Addison Wesley Iberoamericana, México 1994.

Evans , Meryl , *Web Design Process and Workflow*, <http://www.peachpit.com/guides/content.aspx?g=webdesign&seqNum=24> , fecha de consulta: 13 de Agosto de 2009

Figueiras, Aníbal R., *Panorámica de las Telecomunicaciones*, Prentice Hall, Madrid 2002.

Huang, Albert S. and Rudolph Larry, *Bluetooth Essentials for Programmers*, Cambridge University Press, United States of America, 2007.

Huidobro, José M. y Roldán, David, *Integración de Voz y Datos, Call centers, Tecnología y Aplicaciones*, McGraw Hill, España, 2003.

ImageMagick Studio, Examples of ImageMagick Usage, <http://www.imagemagick.org>, fecha de consulta: 10 de Agosto de 2009.

Lizarraga Betzabé, *Apuntes de Bases de Datos*, UNAM - Facultad de Ingeniería, Fecha de Consulta 08 de Julio de 2009

Mallick, Martyn, *Mobile and Wireless Design Essentials*, Wiley, United States of America, 2003.

Oktaba Hanna et al., *Modelo de Procesos para la Industria de Software ,MoProSoft por Niveles de Capacidad de Procesos, Versión 1.3*, México 2005, <http://comunidadmoprosoft.org.mx> , fecha de consulta: 17 de Diciembre de 2008.

Olguín Romo, Heriberto, *Dirección, Organización y Administración de Centros de Tecnología de Información*, UNAM Facultad de Ingeniería, México 2005.

Rodríguez Muñoz, David, *Sistemas Inalámbricos de Comunicación Personal*, Alfaomega, México, 2002.

Roger S., Pressman, *Ingeniería de software-Un Enfoque Práctico*, McGraw Hill, Cuarta Edición, España, 1998.

Vázquez, Pablo, *Creación de Sitios Web*, Grandi S.A., Argentina 2006.

Symfony, The Definitive Guide to symphony Chapter 2 - Exploring Symfony's Code, http://www.symfony-project.org/book/1_2/02-Exploring-Symfony-s-Code, fecha de consulta: Agosto de 2009.

Mischa Schwartz, *Mobile Wireless Communications*, Cambridge University Press, United Kingdom, 2005.

Agradecimientos

A mis padres:

Silverio Camacho Sánchez y María Livia Villán Morales

A mis hermanos:

José de Jesús, Evelin, Edgar Juan y Erick Oswaldo

A mis amigos:

Néstor Gaytan, Juan Carlos Pérez, Mauricio Mayen, Angel Trinidad, Carlos Téllez, Esaú Casarrubias y Familia, Víctor Guzmán, Rubén Garnett, Eduardo Morales, Sergio Becerril, Oscar Iglesias, Ricardo Zavaleta

A la Universidad, mis profesores y sinodales:

Betza, Amaury, Patricia, Fernando

A mi compañero de tesis:

Alejandro L. Amaya Aguilar

Gracias CamiLa

Se dice que un sueño tarda 5 años en volverse un recuerdo más.

Gracias a todos por este sueño.

Atte.

Angel Camacho Villán