

**PROTOTIPO DE UN SISTEMA SOFTWARE DE “MARKETING DE
PROXIMIDAD” MEDIANTE LA TECNOLOGÍA BLUETOOTH**

**OSCAR YESID ANAYA OVIEDO
DAVID EDUARDO FLÓREZ GÓMEZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FISICO MECÁNICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2010

**PROTOTIPO DE UN SISTEMA SOFTWARE DE “MARKETING DE
PROXIMIDAD” MEDIANTE LA TECNOLOGÍA BLUETOOTH**

**OSCAR YESID ANAYA OVIEDO
DAVID EDUARDO FLÓREZ GÓMEZ**

Trabajo de grado para obtener el título de Ingeniero de Sistemas

**Director:
Dr. SERGIO FERNANDO CASTILLO CASTELBLANCO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2010

AGRADECIMIENTOS

Los más sinceros agradecimientos:

Al Doctor Sergio Fernando Castillo Castelblanco, Ingeniero de Sistemas, Ph.D., por su colaboración, apoyo, asesoría y dirección en este proyecto.

A La escuela de ingeniería de sistemas e informática EISI por brindar las herramientas y el soporte a lo largo del proyecto para la consecución del mismo.

A Los compañeros que aportaron sus conocimientos y asesorías en cuanto a desarrollo de software, manejo de la base de datos, elaboración de diagramas de UML, pruebas técnicas, sugerencias y redacción del informe.

A nuestras familias por brindar su apoyo económico y respaldar de forma permanente el desarrollo de este proyecto.

A la Universidad Industrial de Santander - UIS por ofrecer todas las herramientas necesarias para nuestra formación como profesionales.

A todas las personas que apoyaron y contribuyeron de alguna manera a la realización de este proyecto.

A Dios padre por sus bendiciones y pruebas ya que gracias a ellas he madurado como persona y como profesional y porque al seguir sus enseñanzas procuro ser un profesional integral fortalecido en valores y ética.

A mis padres por su apoyo moral y económico, además por convertirse en la fuente de mi inspiración y en la causa para demostrarles que puedo salir adelante y que puedo ser el mejor de sus hijos.

A mis hermanos y mis sobrinos porque gracias a su afecto y compañía me han hecho sentir que siempre puedo contar con su apoyo y que también han sido parte de mi inspiración y compromiso por sacar mi carrera adelante.

A mi hermosa novia Ivonne Juliana quien ha sido gran apoyo y que gracias a su amor, compañía y comprensión pude realizar satisfactoriamente la etapa final de este proyecto.

A mis queridos amigos que conforman una segunda familia donde son los únicos miembros que han sido elegidos para ser parte de esta gran hermandad. Gracias a ellos por estar conmigo en los momentos más cruciales de mi vida y por brindarme su amistad incondicional, gracias a: Beatriz, Luz Marina, Leidy, Rosa, Damián, David, Claudia, Ivar, a mis amigos de la sede UIS Málaga, a mi preciosa Jessica que desde el cielo recibe con agrado este logro y a todos aquellos con los cuales he compartido momentos especiales en mi carrera.

A mi compañero de proyecto David por su apoyo, ayuda, comprensión y por compartir esta bonita experiencia de desarrollar nuestro proyecto de grado en un ambiente lleno de cordialidad, respeto y de amistad.

A mis jefes y compañeros de trabajo porque gracias a ellos he recibido muchas enseñanzas acerca de la etapa laboral que estoy viviendo y porque han contribuido en mi adaptación a la misma.

Oscar Yesid

Al Dios de la vida, que ha puesto en mi camino tantas cosas bellas y maravillosas enseñándome siempre el buen camino, creciendo como persona y aprendiendo día a día a ser mejor.

A mis padres Héctor Javier y Blanca Rosa a quienes les debo todo. Sus enseñanzas brindadas desde niño, el acompañamiento y respaldo en cada paso de mi vida, su amor y su buen ejemplo, el cual ha sido mi guía para asegurarme de tener siempre el mejor modelo a seguir.

A mis hermanos Javier Leonardo e Ivan Dario, a ellos les debo su fraternidad, su compinchería, su alegría y tantos buenos momentos que hemos compartido y que Dios lo quiera seguiremos compartiendo.

A mi familia. Junto con ellos, conformando una familia que alcanza un nuevo significado de unión familiar. El amor y respaldo brindado por cada uno de mis tíos y tías, hace sentir más que orgullo por pertenecer a este hogar.

A mi novia Yudy Mabel, quien ha estado a mi lado desde el inicio de mi carrera y que hoy por hoy continúa brindándome lo mejor de sí. Amor, eres mi inspiración y la persona con la que deseo compartir todos los días de mi vida.

A mi amigo y compañero de proyecto Oscar Yesid, con quien en los últimos años he compartido muy buenos y gratos momentos llenos de risas, amabilidad y compañerismo.

A mis amigos, a todos ellos que siempre han estado a mi lado, Jairo, Julián, Nohora, Sonia, Magaly, Sandra, Cesar, Jhonatan, Chau, Wilfredo, Néstor y Angélica y a todas esas personas maravillosas con las que he compartido a lo largo de mi vida y de mi carrera universitaria. A mi jefe y amigo Eduwin, quien siempre ha estado presto a compartir su conocimiento y enseñanza de la vida laboral y profesional. Agradezco a ustedes los buenos momentos, los consejos, la compañía y el poder compartir un espacio de su vida conmigo.

David Eduardo

TABLA DE CONTENIDO

	PÁG.
INTRODUCCIÓN	19
1. GENERALIDADES DEL PROYECTO	21
1.1 RELEVANCIA DEL TEMA	21
1.2 OBJETIVOS	22
1.2.1 Objetivo General	22
1.2.2 Objetivos Específicos	22
1.3 JUSTIFICACIÓN	24
1.4 METODOLOGÍA	25
1.5 ESTRUCTURA DEL TRABAJO	26
2. MARCO TEÓRICO	27
2.1 TECNOLOGÍAS NECESARIAS	27
2.1.1 BLUETOOTH	27
2.1.1.1 Descripción general	27
2.1.1.2 Comportamiento de un dispositivo Bluetooth.	27
2.1.1.3 Especificaciones técnicas:	29
2.1.1.4 Arquitectura de protocolos Bluetooth	29
2.1.1.5 Ventajas y desventajas otros	31
2.1.2 JAVA JSR-82	32
2.1.2.1 JAVAX.OBEX	33
2.2 DESCRIPCIÓN DE LA METODOLOGÍA APLICADA	33
2.2.1 Modelo En Espiral.	34
2.2.1.1 Planificación	36
2.2.1.2 Análisis de Riesgo	36
2.2.1.3 Ingeniería	36

2.2.1.4 Evaluación del Cliente	36
2.3 MARKETING	36
2.3.1 Análisis Del Mercado.	37
2.3.1.1 Productos	38
2.3.1.2 Comunicación	38
2.3.2 Publicidad	39
2.3.3 Medios Publicitarios	40
2.3.3.1 Los medios impresos	41
2.3.3.2 Los medios electrónicos	41
2.3.3.3 Los medios digitales interactivos	42
2.3.3.4 Los medios externos	43
2.3.4 Marketing De Proximidad	43
3. ESPECIFICACIONES DEL SOFTWARE DESARROLLADO	45
3.1 DESCRIPCIÓN DEL SOFTWARE	45
3.2 ANÁLISIS DE REQUERIMIENTOS.	46
3.3 ARQUITECTURA SOFTWARE.	47
3.3.1 Arquitectura 3 Capas.	47
3.4 HERRAMIENTAS SOFTWARE NECESARIAS	48
3.4.1 Bluecove	48
3.4.2 Apache Derby	50
3.4.3 Jfreechart	51
3.5 PLATAFORMA DE DESARROLLO (IDE)	53
3.5.1 Eclipse.	53
4. DISEÑO DEL PROTOTIPO	55
4.1 DIAGRAMAS DE LENGUAJE DE MODELADO UNIFICADO - UML	55
4.1.1 Logueo	55
4.1.2 Configuración De Campañas.	56
4.1.3 Carga Y Transferencia De Archivos	57

4.1.4 Usuarios	57
4.1.5 Reportes	58
4.2 DIAGRAMAS DE SECUENCIA.	59
4.3 ESTRUCTURA Y DATOS	60
4.3.1 Base De Datos	60
4.3.2 Core Del Sistema:	62
4.3.3 Interfaz Grafica Gui:	63
4.3.4 Manejo Base De Datos:	63
4.3.4.1 Tabla De Usuarios	64
4.4 PROGRAMACIÓN	64
4.4.1 Proceso De Envío De Archivos Mediante La Tecnologia Bluetooth	64
4.4.1.1 Búsqueda de Dispositivos	65
4.4.1.2 Verificación de servicios en el dispositivo: Después de encontrar los	67
4.4.1.3 Crear Conexión:	69
4.4.1.4 Envío del archivo.	70
5. DESARROLLO DEL PROTOTIPO	72
5.1 DESCRIPCIÓN DEL PRIMER PROTOTIPO	72
5.2 DESCRIPCIÓN DEL PROTOTIPO FINAL	73
5.2.1 Login	74
5.2.2 Prototipo En Ejecución	75
5.2.2.1 Panel Principal	75
5.2.2.2 Configuración De La Campaña	77
5.2.3 Reportes.	84
5.2.4 Usuarios	87
5.3 EVALUACIÓN	90
CONCLUSIONES	91
RECOMENDACIONES	93

LISTA DE FIGURAS

	PÁG.
Figura 1. Ejemplo de una Piconet	28
Figura 2. Pila de Protocolos Bluetooth	30
Figura 3. Modelo en Espiral	35
Figura 4. Picadilly (Londres)	39
Figura 5. Medios Digitales	42
Figura 6. Marketing de Proximidad	43
Figura 7. Ejemplos de Figuras JfreChart	52
Figura 8. IDE - Logo de Eclipse	53
Figura 9. Diagrama UML de Logueo	55
Figura 10. Diagrama UML de Configuración de Campañas	56
Figura 11. Diagrama UML de Carga y Transferencia de Archivos	57
Figura 12. Diagrama UML Acciones de los Usuarios	58
Figura 13. Diagrama UML de los Reportes	59
Figura 14. Diagrama de Secuencia del Logueo	60
Figura 15. Diseño de la base de datos para el manejo de campañas, estadísticas y usuarios	61
Figura 16. Core del Sistema	62
Figura 17. Clase Base de Datos	63
Figura 18. Tabla de Usuarios - JTable	64
Figura 19. Ventana de Login	74
Figura 20. Ventana de Inicio de ODDOBLUE	75
Figura 21. Ventana ODDOBLUE - Menú Archivo	76
Figura 22. Nueva Campaña	77
Figura 23. Selección de Archivo a Enviar	78
Figura 24. Formato Seleccionado de Imagen	79
Figura 25. Formato Seleccionado de Audio	80

Figura 26. Formato Seleccionado de Juego	80
Figura 27. Formato Seleccionado de Video	81
Figura 28. Otro Tipo de Formato Seleccionado	81
Figura 29. Selección de Fecha para Inicio de la Campaña	82
Figura 30. Carga y Ejecución del Prototipo	83
Figura 31. Inicio del Envío de la Campaña	84
Figura 32. Reporte Estadístico	85
Figura 33. Reportes	86
Figura 34. Administración de Usuarios	87
Figura 35. Creación/Edición de Usuarios	88
Figura 36. Confirmación de la Contraseña de Usuario	89
Figura 37. Administración de Usuarios - Listado	89

LISTA DE TABLAS

	PÁG.
Tabla 1. Servicios Bluecove	49
Tabla 2. Método de DiscoveryListener	72

RESUMEN

TÍTULO: **PROTOTIPO DE UN SISTEMA SOFTWARE DE “MARKETING DE PROXIMIDAD” MEDIANTE LA TECNOLOGÍA BLUETOOTH*.**

AUTORES: **OSCAR YESID ANAYA OVIEDO**
 DAVID EDUARDO FLOREZ GOMEZ**

PALABRAS CLAVE:

Bluetooth, Java JSR 82, Marketing, Marketing de Proximidad.

Es conveniente reconocer que actualmente las tecnologías de información y comunicación – TIC, están ejerciendo una gran influencia en las distintas formas de comunicación entre las personas y las organizaciones, apoyando al mismo tiempo a la sociedad, en el proceso de transición de la era industrial a la nueva era de la información.

El trabajo que aquí se presenta, está orientado a fortalecer el cambio que actualmente se observa con respecto a la forma de tratar la información, la cual ya no es tratada como el conjunto de datos que se almacenan para dar cuenta de procesos, actividades, etc., sino como el principal bien que mediante su gestión apoya tanto la toma de decisiones como los procesos de comunicación, de las personas en general.

Este proyecto presenta el diseño y desarrollo de una herramienta Software, que permite la transferencia de información entre las personas a través de sus dispositivos móviles, los cuales deberán tener necesariamente dentro de sus componentes electrónicos, la tecnología Bluetooth (encargada de transmitir de forma inalámbrica la información).

La herramienta Software desarrollada esta soportada en el lenguaje JAVA y en la plataforma IDE Eclipse, utilizados en la programación de los distintos componentes que conforman el sistema de marketing de proximidad, que tiene como objetivo el envío de campañas publicitarias.

Es aquí donde se presenta una oportunidad de hacer envío de información publicitaria a los dispositivos móviles de una manera limpia y sin costo para el cliente. Permitiendo así que esta alternativa sea un nuevo medio de difusión que posibilite llegar al público.

* Trabajo de grado s

** Facultad De Físico Mecánicas. Escuela De Ingeniería De Sistemas E Informática. Director Dr. Sergio Fernando Castillo Castelblanco

ABSTRACT

TITLE: PROTOTYPE SOFTWARE SYSTEM OF A PROXIMITY MARKETING THROUGH BLUETOOTH TECHNOLOGY*.

AUTHORS: OSCAR YESID ANAYA OVIEDO
DAVID EDUARDO FLOREZ GOMEZ**

KEY WORDS:

Bluetooth, Java JSR 82, Marketing, Marketing de Proximidad.

Is appropriate to acknowledge at present the information and communication technologies (ICT) are exerting a big influence on the different ways of communication between organizations and people, while supporting at the same time the society, in the process of transition from the industrial age to the new age or information.

The work presented here, it's oriented to strengthen change currently observed with respect to the way of how to tray the information, which is not longer treated as the data set that is stored to be executed in processes, activities, etc., but as the main well supported both by it's management decision-making as the process of communication, people in general.

This project presents the design and development of a software tool, which allow transfer of information between people through its mobile device, which must necessarily have within their electric components, the Bluetooth technology (responsibly for transmitting the information wirelessly)

Software tool development is developed in JAVA language and development platform IDE Eclipse, used in programming of different compounds that make up the proximity marketing system, which aims at sending advertising campaigns.

It is here where one presents an opportunity to do sending of advertising information to the mobile devices of a clean way and without cost for the client. Allowing so this alternative is a new way of diffusion that it makes possible to come to the public.

* Work of degree s

** Physicist's faculty Mechanics. School Of Systems engineering And Computer science. The director Dr . Sergio Fernando Castillo Castelblanco

INTRODUCCIÓN

Los dispositivos móviles o celulares han sufrido una gran evolución desde su aparición en 1983 cuando Motorola lanzó al mercado el DynaTAC 8000x que pesaba 800gr, la batería duraba sólo una hora de conversación y ocho de Stand-By (en espera), era analógico y tenía un pequeño display de leds, las personas que lo adquirieron pagaron un valor de US\$3995.

Los celulares hoy día, cuentan con un peso y tamaño más reducido, son digitales, tienen una duración de batería más prolongada, grandes y vistosas pantallas táctiles, y por supuesto sirven para algo más que hablar por medio de él. Los fabricantes los han diseñado para convertirse en un objeto personal e indispensable, ya que tienen otras tareas que facilitan la vida de quien lo posee, entre las que se destacan: Alarma, Agenda, cámara fotográfica, Walkman, hasta de reloj personal y transferencia de archivos por Bluetooth, llegando de esta forma a ocupar un lugar importante en el usuario y en el entorno comercial. Se nombra éste último, ya que estas nuevas tecnologías brindan una nueva oportunidad al comercio de pautar y llegar a clientes potenciales directamente a su celular.

Es aquí donde se presenta una oportunidad de hacer envío de información publicitaria a los dispositivos móviles de una manera limpia y sin costo para el cliente. Permitiendo así que esta alternativa sea un nuevo medio de difusión que posibilite llegar al público. Además de que ofrece diferentes ventajas respecto a otros tipos de medios utilizados para hacer publicidad que, actualmente se encuentran en el mercado.

Utilizar dispositivos móviles que dispongan de la tecnología Bluetooth, permite hacer publicidad de manera limpia y dirigida, y solamente cuando el cliente acepte

la invitación a recibir la información de publicidad, se hará la transferencia del archivo, es decir que no es invasivo. Tarea que resulta ser opuesta al manejo que se dan a los mensajes de texto que llegan a los clientes, los cuales son enviados por las compañías de telefonía celular, sin tener un mecanismo que les permita moderar la aceptación del mensaje por parte del cliente.

Observando la necesidad de contar con un servicio que permita a las organizaciones y personas en general, enviar sus mensajes de publicidad a los clientes, tomando en cuenta su interés de recibirlos o no, se presenta como alternativa de solución, el desarrollo del prototipo de un sistema software de "Marketing de Proximidad" basado en la tecnología Bluetooth, que permitirá tomar en cuenta la decisión de los clientes en cuanto a aceptar o rechazar la información que le ha sido enviada.

1. GENERALIDADES DEL PROYECTO

1.1 RELEVANCIA DEL TEMA

La publicidad es el medio que utilizan las empresas para dar a conocer sus productos al mundo. La forma de hacer publicidad es muy variada, para llegar a aplicarla se han utilizado algunos elementos como los panfletos, que se reparten en las calles o los medios de comunicación (Radio, TV) cuyo objetivo es llamar la atención del los clientes, también se encuentran las vallas publicitarias, las cuáles han cambiado el entorno visual de las ciudades, debido a que los comerciantes han llenado de anuncios, colores y luces, los espacios más concurridos por las personas del común.

Hoy en día se utilizan las nuevas tecnologías han llegado para quedarse, un ejemplo de ellas es el uso del teléfono celular, que día a día se ha ido masificando entre organizaciones y personas, a tal punto que se ha vuelto un elemento de uso cotidiano para la gran mayoría de las quienes las manejan. Este es un medio que no ha sido explotado en un sentido comercial, para convertirse en un medio de publicidad. Es en este punto donde el uso de las nuevas tecnologías y los modelos para hacer marketing, convergen.

Se encuentra una oportunidad de utilizar las nuevas tecnologías y así desarrollar un aplicativo que cumpla la función de poder enviar publicidad a clientes potenciales, de tal forma que pueda ser utilizado para el área del marketing de proximidad.

1.2 OBJETIVOS

Los objetivos alcanzados en el trabajo de grado son los que a continuación se mencionan:

1.2.1 Objetivo General

Desarrollar un prototipo de un sistema software que permita el envío de información (general, comercial, contextual) desde un terminal a un dispositivo móvil cliente, por medio de la tecnología Bluetooth, permitiendo la interacción del usuario con el sistema, enmarcado dentro del contexto del sector comercial.

1.2.2 Objetivos Específicos

Desarrollar una aplicación en el servidor, mediante la cual se pueda configurar y adicionar los contenidos que la entidad fuente desea difundir a sus usuarios. Para ello la herramienta debe:

- Ofrecer una interfaz en ambiente Web amigable y fácil de utilizar, ya que la persona que va a hacer la configuración de los contenidos a difundir, va a ser cualquier usuario con conocimientos básicos en informática.
- Permitir la adecuada administración de los recursos, tales como imágenes, videos, archivos de audio, aplicaciones java, entre otros. Que serán utilizados para el envío mediante el sistema.
- Detectar de forma automática el dispositivo móvil que entra en el área de cobertura del terminal Bluetooth, para iniciar la conexión y verificar que no se envíe la información si el dispositivo previamente ha recibido los contenidos sin que se haya salido del área de cobertura.

- Facilitar la portabilidad al brindar la posibilidad de tener la aplicación en un computador establecido en el lugar donde se implemente el sistema o de estar en un servidor remoto.

Ofrecer siempre que exista una conexión entre el punto terminal y el dispositivo móvil, la posibilidad de que el usuario elija si desea o no recibir la información que el sistema le está ofreciendo en ese instante.

Desarrollar una aplicación móvil que sea descargable desde el mismo sistema de envío de información, la cual debe permitir la interacción entre el usuario del dispositivo móvil y el sistema como tal. Para ello, se debe:

- Proveer al usuario consultas de información más detallada y más específica de los productos o servicios que está publicitando la entidad fuente.
- Implementar una interfaz sencilla y agradable que sea compatible con la mayoría de los teléfonos celulares existentes en el mercado que cumplan con las características de: soporte Bluetooth y soporte de Java.
- Poseer el soporte para futuras utilizaciones de la aplicación en el móvil, así como también puede actualizarse automáticamente cuando entre en conexión con el sistema.

Contabilizar y registrar la cantidad de peticiones aceptadas y rechazadas que hace el sistema al momento de enviar los contenidos publicitarios cuando se detecten los dispositivos móviles.

1.3 JUSTIFICACIÓN

Actualmente en el mundo las personas están más ligadas a la tecnología gracias a la telefonía celular. Estos dispositivos electrónicos creados inicialmente para transmitir voz, han evolucionado de manera significativa en los últimos 10 años, pasaron de transmitir voz a poseer posicionador global (GPS) incorporados y transmisión de internet inalámbrica.

Para conquistar un mayor número de clientes, los diseñadores han implementado una gama de accesorios que lo hagan ser más solicitado por los consumidores, una de estos atractivos ha sido el Bluetooth, creado para transmitir datos a corta distancia, de manera rápida y eficientemente.

Con respecto a esta tecnología de Bluetooth, y a las características que presenta de transmisión de texto, imágenes, video, ringtones, etc; tomando en cuenta las bondades que presenta la tecnología mencionada, la realización de éste trabajo de grado se orientó a la creación de una plataforma Software que facilita cargar archivos para poder enviarlos vía Bluetooth a otros celulares. Esta plataforma desarrollada en JAVA y utilizando un dispositivo Bluetooth básico para su funcionamiento, permite el admite cargar archivos y enviarlos automáticamente sin necesidad de que una persona esté coordinando esta operación. De manera más específica, a continuación se exponen los distintos aspectos que proporciona el software desarrollado:

- Ofrecerá una alternativa para hacer publicidad de manera directa al cliente donde podrá recibir archivos multimedia sobre el producto publicitar.
- Permite al cliente la escogencia entre aceptar o rechazar el contenido publicitario. Esto es novedoso ya que la publicidad del entorno actual es

transmitida espontáneamente y no permite al cliente entre escoger verla o desecharla.

- Cuando el cliente acepta la invitación para recibir información publicitaria en su teléfono, es una publicidad que llega al cliente de manera directa, el cliente estará dispuesto a observar la información recibida.
- La plataforma informará sobre las campañas que tengan un mayor impacto entre las personas. Así se podrá escoger entre los gustos de los consumidores y atraerlo de manera directa y efectiva.
- Al eliminar la impresión o la publicación en papel de la publicidad se ayuda a la conservación del medio ambiente.

1.4 METODOLOGÍA

La metodología seleccionada para el desarrollo de este proyecto fue el Modelo de Prototipado o también llamado Modelo en Espiral.

Esta metodología fue la que mejor se adaptó a las características del proyecto, ya que permitió al equipo de trabajo realizar prototipos de la herramienta a medida que se iban desarrollando y así en cada iteración se lograron evaluar los objetivos, restricciones, riesgos y nuevos requerimientos que surgían a la par del desarrollo.

Se consideraron las etapas de Planeación, Análisis de riesgo, Ingeniería, Evaluación, Toma de decisiones y Refinamiento, la cuales se ajustaron al desarrollo de este proyecto, y al mismo tiempo permitieron un constante control de la evolución de la herramienta en sus diferentes etapas, logrando con su aplicación convertir los primeros prototipos en aquellos que fundamentaron los

demás ciclos. Al mismo tiempo se obtuvo la calidad del producto en cada una de las etapas.

1.5 ESTRUCTURA DEL TRABAJO

A continuación se presenta una síntesis sobre los temas que trata el presente documento:

En el capítulo Uno se realiza la presentación formal del proyecto, haciendo mención de; descripción del trabajo de grado, los objetivos general y específicos, justificación, metodología aplicada y soporte del mismo.

En el capítulo Dos se conceptualizan los términos que encierran el tema del proyecto de grado desarrollado. Se presenta un estudio de la tecnología Bluetooth, sus características, arquitectura, topología, perfiles, etc., un estudio sobre el Marketing y la publicidad, una descripción de los API's de JAVA para Bluetooth, y la explicación del prototipo de Marketing de Proximidad y sus alcances.

El capítulo Tres presenta las Especificaciones que fundamentaron el desarrollo del software desarrollado. Se realiza una breve explicación de: Análisis de Requerimientos, Arquitectura Software, Herramientas Software utilizadas y Plataforma IDE tomada en cuenta.

El capítulo Cuatro presenta el diseño e implementación de la aplicación, mostrando tanto el primer prototipo, como el prototipo final, así como la evaluación de éste último.

2. MARCO TEÓRICO

2.1 TECNOLOGÍAS NECESARIAS

A continuación se presentan las tecnologías que se tomaron en cuenta para desarrollar el sistema obtenido en éste proyecto.

2.1.1 BLUETOOTH

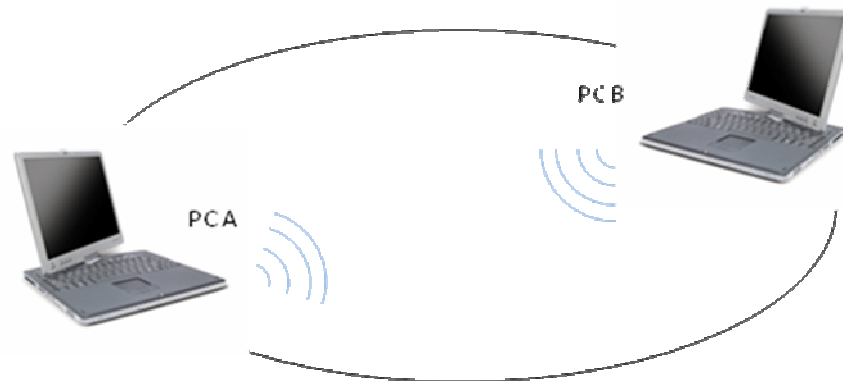
2.1.1.1 Descripción general: La tecnología inalámbrica Bluetooth es un sistema de comunicaciones de corto alcance, cuyo objetivo es eliminar los cables en las conexiones entre dispositivos electrónicos, tanto portátiles como fijos, manteniendo altos niveles de seguridad.

Las características principales de esta tecnología son su fiabilidad, bajo consumo y costo.

2.1.1.2 Comportamiento de un dispositivo Bluetooth. Gracias a su gran aceptación, un dispositivo Bluetooth puede conectarse con casi cualquier otro dispositivo compatible que se halle en las proximidades, eliminando las fronteras en cualquier parte del mundo.

Los dispositivos electrónicos equipados con tecnología Bluetooth pueden conectarse y comunicarse de forma inalámbrica mediante redes de corto alcance denominadas piconets. Cada dispositivo puede conectarse simultáneamente con hasta siete dispositivos más dentro de una misma piconet.

Figura 1. Ejemplo de una Piconet



Un dispositivo puede pertenecer a varias Piconets al mismo tiempo. Las Piconets se establecen de forma dinámica y automática cuando los dispositivos Bluetooth se encuentran en el mismo radio de acción. En esta función un dispositivo toma el papel de Maestro y el otro de Esclavo. Solamente puede haber un sólo Maestro en cada Piconet, y será aquel que sincroniza a todos los esclavos con su reloj y patrón de saltos.

Una de las principales ventajas de la tecnología inalámbrica Bluetooth es su capacidad para gestionar simultáneamente tanto transmisiones de voz como de datos. Esto permite a los usuarios disfrutar de una gran variedad de soluciones innovadoras, tales como el uso de manos libres para atender llamadas, funciones de impresión y fax, ó la sincronización de aplicaciones entre Asistente Personal Digital -PDA, ordenadores y móviles, entre otras muchas.

Hasta el momento han sido lanzadas cuatro versiones de la especificación: la versión 1.2 adoptada en noviembre de 2003. Y la versión 2.0 lanzada un año más tarde. Con velocidad de trasmisión de 2 a 3Mbps y con un rango de 10m de cobertura [1].

Las nuevas versiones son la Bluetooth 3.0 lanzada en Abril de 2009. Esta versión utiliza el protocolo de conexión 802.11, es decir el WiFi¹ del equipo (el equipo debe estar dotado de Bluetooth 3.0 y Wifi) lo que conlleva a una transferencia de datos de hasta 24Mbps. El 17 de Diciembre de 2009 el Grupo de Especial Interés por Bluetooth (Bluetooth Special Interest Group) anunció la adopción de Bluetooth Low Energy Wireless technology. (Tecnología inalámbrica Bluetooth de bajo consumo de energía), llegando a ser esta la versión Bluetooth 4.0. [2]

2.1.1.3 Especificaciones técnicas: Bluetooth opera en una banda de frecuencia industrial omnidireccional, científica y médica (ISM) pensada así por sus desarrolladores, no requiere licencia y se encuadra directamente, entre 2,4 y 2,485 GHz. Además utiliza un espectro ensanchado de salto de frecuencia o FHSS (Frequency Hopping Spread Spectrum), la banda de radio está dividida en 79 subcanales. El radio Bluetooth usa uno de estos canales de frecuencia y salta de canal en canal, demorándose 625 microsegundos en cada canal, alcanzando una velocidad de 1600 saltos por segundo.

2.1.1.4 Arquitectura de protocolos Bluetooth: La pila de Protocolos Bluetooth se basa en el modelo de referencia OSI (Open System Interconnect) de ISO (International Standard Organization), para interconexión de sistemas abiertos. Permite el intercambio transparente de información entre aplicaciones diseñadas con esta especificación.

La pila de protocolos Bluetooth se divide en dos zonas, el Modelo Bluetooth y el Host Bluetooth.

- **Modelo Bluetooth:** Parte Hardware encargada del envío de información a través del interfaz de radiofrecuencia.

¹ Es un sistema de envío de datos que utiliza ondas de radio en lugar de cables.

- **Host Bluetooth:** Parte Software, encargado de enlace y aplicación de las capas superiores.

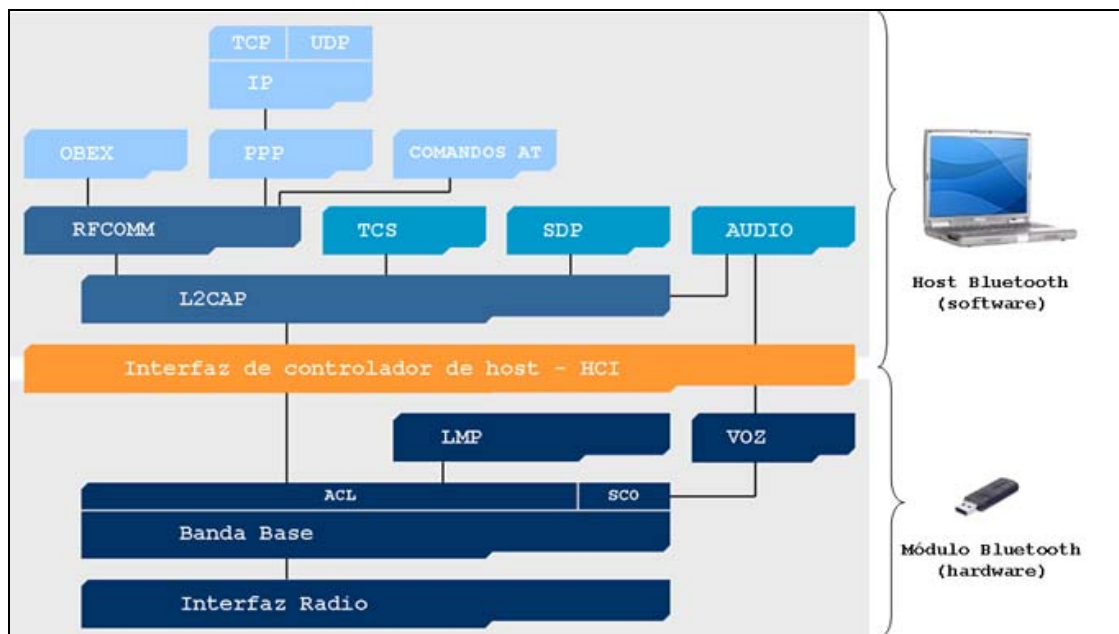
Las dos zonas están comunicadas por el Interfaz de Controlador del Host (HCI).

Las pilas de protocolo Bluetooth más conocidas son:

- Widcomm.
- Toshiba Bluetooth Stack.
- Microsoft Windows XP Bluetooth.
- IVT Blusoleil Stack.

A pesar de la existencia de las diferentes pilas de protocolos desarrollados por varios entes u organizaciones, la especificación Bluetooth exige que se mantenga la interoperabilidad entre los dispositivos que implementen diferentes pilas.

Figura 2. Pila de Protocolos Bluetooth



De la Figura 2 se mencionarán los protocolos más destacados para el desarrollo de éste proyecto.

En la parte inferior de la Figura 2. Se ve la Interfaz de Radio y la capa de Banda base. Estas capas permiten el enlace físico por radiofrecuencia entre unidades Bluetooth dentro de una Piconet.

El protocolo LMP (Link Manager Protocol) es el responsable de la configuración y control de enlace entre dispositivos Bluetooth. Esta comunicación se realiza a través de mensajes, que incluyen mecanismos de seguridad tales como autenticación y cifrado. Por medio de estos mensajes el protocolo LMP controla los modos de administración de energía y los ciclos de trabajo de los dispositivos de radio Bluetooth, así como los estados de conexión dentro de una Piconet.

Otra capa importante dentro de la pila de protocolos Bluetooth, es la capa de interfaz de controlador del Host (HCI). Esta capa actúa como frontera entre las capas relativas al Hardware y al Software. Esta capa permite disponer de una capa de acceso homogénea para los módulos Bluetooth aunque sean de distintos fabricantes. [3]

Al igual que otros estándares de IEEE-802, Bluetooth utiliza direcciones MAC de 6 bytes para el direccionamiento de equipos a nivel de red. Esta dirección es necesaria para transmitir cualquier tipo de información a otros dispositivos.

2.1.1.5 Ventajas y desventajas: La oportunidad de transmitir datos de un equipo a otro sin necesidad de cables hace del Bluetooth la opción ideal para evitar estos inconvenientes. Se trata de un estándar inalámbrico disponible en todo el mundo, con el que se pueden conectar entre sí teléfonos celulares, computadores portátiles, PDA's, entre otros; sin importar la marca del fabricante o las barreras de nacionalidad.

Otra ventaja que posee, es la capacidad para administrar simultáneamente tanto transmisores de voz como de datos. Esto se ve reflejado en dispositivos tales como: los manos libres para celulares, impresiones desde el celular o Laptop sin cableado, ó sincronización de aplicaciones entre PDA's, entre muchos otros.

Además, Bluetooth fue concebida como una tecnología inalámbrica de bajo consumo de energía, poco peso y tamaño, y bajo coste de operación.

Su desventaja ante otros dispositivos de radio, es la tasa de transferencia que resulta ser baja (hasta 3Mbps para la versión 2.0), comparada con otras tecnologías inalámbricas existentes. Aunque con el lanzamiento en 2009 de las tecnologías 3.0 y 4.0 de Bluetooth se ha roto esa brecha, los dispositivos que cuentan con ella no han obtenido la penetración que tienen actualmente las versiones anteriores de Bluetooth.

2.1.2 JAVA JSR-82

La especificación "JSR-82 Java APIs para Bluetooth" es un conjunto de normas enfocados al uso del Bluetooth por medio de la plataforma JAVA. El objetivo de esta especificación es definir la arquitectura y las API's asociadas, y permitir un entorno de desarrollo para aplicaciones que utilicen el Bluetooth como medio de comunicación. Esta JSR fue creada en un principio para poder ser utilizada en J2ME (JAVA 2 MICRO EDITION) pero tiene la particularidad de ser usada en J2SE (JAVA 2 STANDARD EDITION).

La especificación JSR-82 trae dos paquetes: *javax.bluetooth* y *javax.obex*. La API Obex (Object Exchange) está empaquetada por separado, es decir, está definida independiente de la capa de transporte de Bluetooth. Así que cualquier aplicación puede utilizar los dos paquetes ó solamente uno de ellos.

2.1.2.1 JAVAX.OBEX: Obex es un protocolo de transferencia que define los objetos de datos y el protocolo de comunicaciones que deben utilizar dos dispositivos para intercambiarlos. Se basa en un sistema de Cliente-Servidor. El protocolo crea un listado de carpetas que se utiliza para explorar los contenidos de las carpetas desde el dispositivo remoto.

La API Obex es un protocolo de alto nivel, similar al HTTP. Al igual que este último, el protocolo Obex se basa en mensajes compuestos por cabeceras de mensaje y opcionalmente por el cuerpo del mensaje. Además poseen código de respuesta indicando éxito o error. [4].

En este protocolo el cliente envía comandos al servidor, entre los que se destacan:

- **CONNECT:** Inicia Sesión.
- **PUT:** Envía un archivo al servidor.
- **GET:** Solicita un archivo al servidor.
- **DELETE:** Solicita la eliminación de un archivo.
- **SETPATH:** El cliente desea cambiar el directorio actual dentro del sistema de archivos del servidor.
- **DISCONNECT:** Finaliza la sesión.

2.2 DESCRIPCIÓN DE LA METODOLOGÍA APLICADA

El desarrollo de Software no es tarea fácil, prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Una metodología es un conjunto de procedimientos que permiten producir y mantener un producto software, esto es, define una serie de pasos a seguir para obtener un software de calidad.

Las metodologías de Software revelan cómo construir técnicamente software, son guías y abarcan tareas como planeación de proyectos, recolección y análisis de requerimientos, diseño, implementación de la aplicación, pruebas de funcionamiento, documentación y mantenimiento.

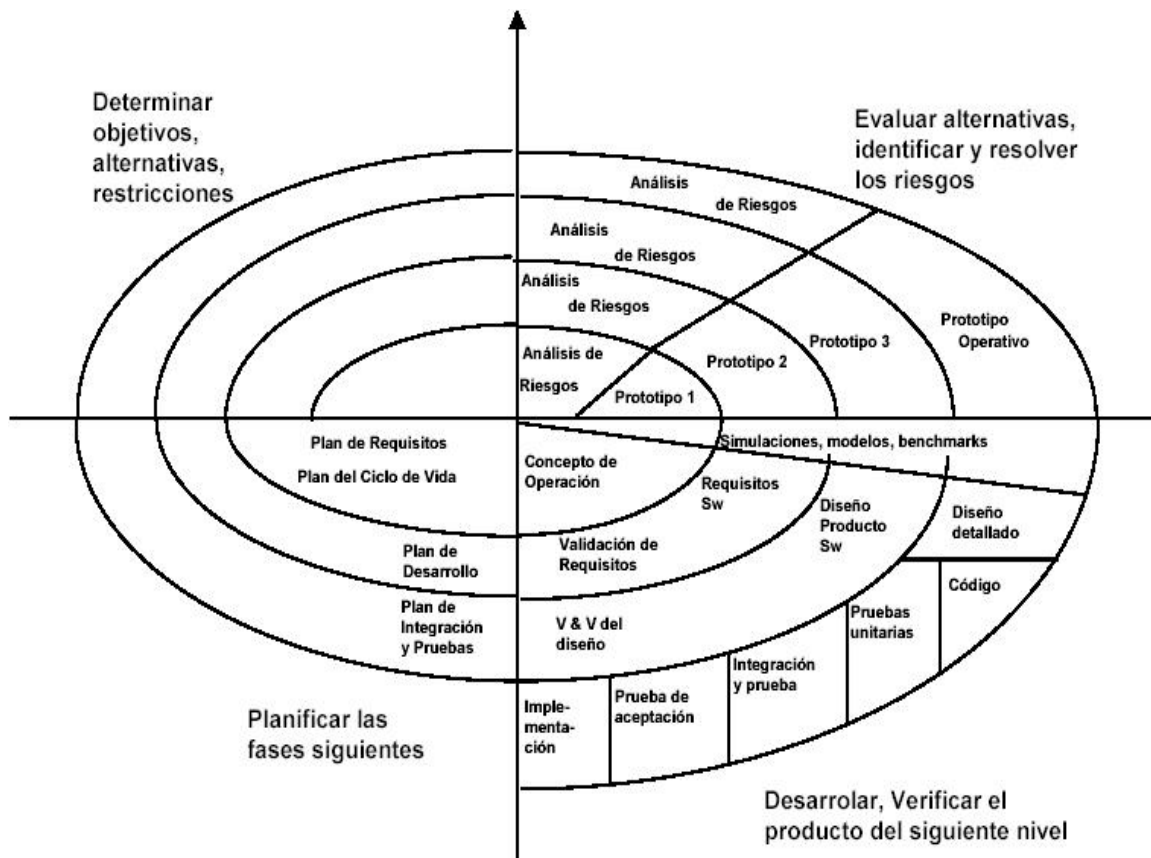
Existen muchos modelos de ciclo de vida de un proyecto con diferentes enfoques, cada uno con sus ventajas e inconvenientes.

La metodología seleccionada para el desarrollo de este proyecto es el Modelo de Prototipado o también llamado Modelo en Espiral.

2.2.1 Modelo En Espiral.

Introducido por B. Boehm (A spiral model for software development and enhancement, 1988). Del modelo se puede decir que es una mezcla de todos los demás modelos de desarrollo de software, que va refinando los requisitos del sistema a medida que ellos se van descubriendo. Boehm, precisa que la gran mayoría de los requerimientos surgen en fase de diseño e implementación.

Figura 3. Modelo en Espiral



El modelo en espiral captura algunos principios básicos.

- **Planificación:** Determinación de objetivos, alternativas y restricciones. Se revisa y evalúa lo realizado hasta cierto punto, y se toma la decisión de continuar con las siguientes fases y planificar actividades.
- **Análisis de riesgo:** Análisis de alternativas e identificación/solución de riesgos.
- **Ingeniería:** Desarrollo del producto “Siguiete nivel”.
- **Evaluación del cliente:** Valoración de los resultados de la ingeniería.

2.2.1.1 Planificación: El objetivo de esta fase es delimitar los aspectos generales del software. El desarrollador y el cliente especifican y definen las características globales del software. Se identifican los requerimientos y cuales deben tener una mayor atención, también se profundiza sobre las mejores opciones.

2.2.1.2 Análisis de Riesgo: Se hace un estudio de los posibles riesgos que se puedan presentar en el desarrollo del proyecto. Se realiza una selección de las propuestas planteadas, de tal forma que se reduzcan o eliminen los riesgos. Esta etapa sirve también para verificar y validar los resultados obtenidos.

2.2.1.3 Ingeniería: Se investiga sobre las tecnologías a utilizar, los requerimientos del sistema (Hardware) y la arquitectura Software a utilizar e implementar. En esta etapa del desarrollo del software se toman los ítems anteriores y se hacen avances en el prototipo.

2.2.1.4 Evaluación del Cliente: Se hace una presentación preliminar del software a los interesados, mostrando los avances del mismo. En esta fase se hace una evaluación de los objetivos pactados inicialmente y se compara con la dirección que está tomando el desarrollo. Aquí se termina el primer ciclo, dando inicio al segundo bucle de este con las mismas características del ciclo anterior. Esto ayuda a crear un software estable, con metas fijas y lineamientos claros en cada una de las etapas y ciclos que se cumplan.

2.3 MARKETING

El término Marketing tiene varias definiciones. En términos generales, el Marketing es un conjunto de actividades destinadas a satisfacer las necesidades y deseos de los mercados a cambio de una utilidad o beneficio para las empresas u

organizaciones que la ponen en práctica. Razón por la cual, nadie duda que el Marketing es indispensable para lograr el éxito de los mercados actuales. [5]

En una empresa, el Marketing debe desempeñar unas funciones orientadas a analizar y comprender el mercado donde se mueve la empresa, identificar las necesidades de los clientes y desarrollar una demanda para favorecer los productos de la empresa.

2.3.1 Análisis Del Mercado.

La empresa debe conocer con detalle cuáles son sus clientes potenciales y reales. Se debe investigar sobre cuáles son sus necesidades, deseos, hábitos de compra, capacidad de compra, etc. Esto permitirá a la empresa que desee ingresar a ese mercado, ofrecer los productos más adecuados a sus clientes.

Se han identificado cinco roles que influyen en la compra de un producto por parte de un cliente:

- **El iniciador:** sugiere o motiva la compra.
- **El influenciador:** ofrece una influencia positiva o negativa sobre la compra.
- **El Decisor:** toma la decisión de hacer o no la compra.
- **El comprador:** realiza la compra.
- **El usuario:** disfruta finalmente el producto.

El análisis de mercado puede hacerse de manera más o menos sistemática. Cuando un observador “se da cuenta” que hace falta un nuevo producto que él puede ofrecer, probablemente haya realizado un análisis inconsciente del mercado. Por otro lado están las grandes empresas que están dispuestas a invertir grandes sumas de dinero y cuentan con personal idóneo para realizar dicho estudio.

2.3.1.1 Productos: Para las empresas, es importante realizar un estudio interno sobre sus productos frente a la situación actual del mercado. Es ahí donde el Marketing hace su contribución.

Inicialmente una empresa comienza con un producto, pero pronto empezará con la introducción de nuevos productos. Es así como surgen una gran variedad de productos dentro de dichas organizaciones comerciales y posteriormente se especializan en el desarrollo tecnificado de los mismos.

Dentro de este marco es importante estudiar el ciclo de vida de los productos. Dado que todo producto que se introduce en un mercado, se desarrolla y finalmente desaparece, por ello se debe estar atento y detectar el momento de su decadencia, para así poder tomar medidas correctivas.

La modificación de las características de un producto suele ser una de las formas de influir en el mercado. Cambiando los empaques, los embalajes, etc. Así también se pueden usar materiales de mayor calidad, añadir un manual de operaciones o realizar pequeñas modificaciones a los productos finales que, en este caso, sería el gancho para atraer a una mayor cantidad de clientes.

2.3.1.2 Comunicación: No basta con tener un buen producto a un buen precio y listo para ser entregado, es importante que los posibles clientes sepan que el producto existe.

Dentro de las políticas de comunicación se agrupan varias técnicas para dar a conocer un producto. Tales como:

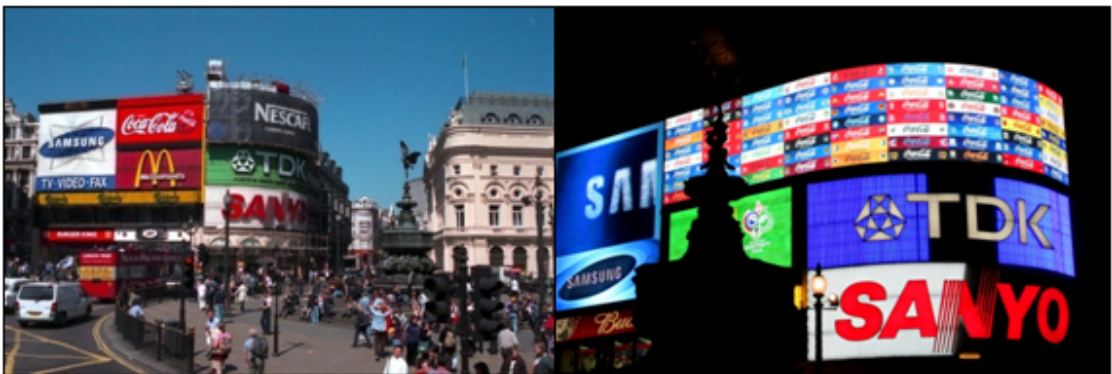
- La publicidad.
- La venta personal.
- La promoción.

- Las relaciones públicas.
- La propaganda.

2.3.2 Publicidad

En el contexto del Marketing, la publicidad es la política de comunicación más conocida y la que constituye la inversión más cuantiosa.

Figura 4. Picadilly (Londres)



Es la forma destinada a difundir o informar al público sobre un bien o un servicio a través de los medios de comunicación. El objetivo primordial de la publicidad es motivar al público hacia una acción de consumo. Ésta informa al consumidor sobre los beneficios de un determinado producto o servicio.

En el año 1711, el periódico The Spectator, descubre que la venta de publicidad permitiría abaratar los costos del ejemplar ya que los anunciantes serían quienes financien los costos de la edición.

La publicidad es pagada por las empresas para dar a conocer sus productos y sin ella quebrarían los medios informativos y de entretenimiento, cuya supervivencia depende del apoyo financiero.

La publicidad favorece a una economía sana y sirve para crear consumidores que están mejor informados, son más conocedores y exigentes.

Antes de iniciar una campaña publicitaria se deben tener en cuenta los análisis del Marketing sobre el tipo de campaña a emprender y, tener como objetivo un público o escenario definido. Una vez conocido y estudiado este objetivo se determinan los medios de comunicación a utilizar para alcanzar las metas o la realización de las ventas proyectadas.

La publicidad orientada a los productos estimula la acción a corto plazo de un producto o servicio, además de crear conocimiento de la compañía. A continuación se presentan las tres opciones de publicidad más representativas:

- **Publicidad de línea de precios regulares:** Da a conocer al mercado los precios que maneja la empresa. El cliente se encarga de hacer las comparaciones.
- **Publicidad orientada a ofertas:** Precios especiales, dos por uno, descuentos, etc.
- **Publicidad orientada a liquidación:** Rotación de productos lentos, mercancía a punto de expirar, mercancía en mal estado, cierre del negocio, etc.

La publicidad institucional no sólo trata de generar una imagen favorable del producto sino de introducir una imagen favorable de la empresa a largo plazo, promociona una idea acerca de la compañía, de tal forma que se llegue a crear una actitud favorable hacia ella.

2.3.3 Medios Publicitarios

El medio transmisor del mensaje del patrocinador es el enlace entre la compañía que ofrece un producto y el cliente que desea comprarlo. Aunque el término

Medios esté relacionado generalmente con los canales de comunicación masiva como Televisión, radio, prensa y revistas, también denota otros vehículos de promoción como vallas publicitarias, carteles u otros elementos visuales. Además alberga a las nuevas tecnologías de la comunicación como el Internet, los medios electrónicos, el fax, etc.

Los medios publicitarios se pueden clasificar en los siguientes tipos:

- Los Medios impresos.
- Los Medios electrónicos.
- Los medios digitales interactivos.
- Los Medios externos.

2.3.3.1 Los medios impresos: Con la expresión Medios impresos se hace alusión a cualquier tipo de medio impreso de publicación comercial (periódicos, revistas, folletos), que venden espacios publicitarios en sus páginas interiores. Es la forma más tradicional del marketing.

El listado de los medios impresos que utilizan la publicidad es extenso. Cada vez que haya un punto de interés donde un medio de comunicación impreso esté involucrado, es seguro que llevará publicidad en su interior.

2.3.3.2 Los medios electrónicos: A la radio y a la televisión se les consideraba medios de radiodifusión. Pero la tecnología ha evolucionado y existe la televisión por cable transmitiendo electrónicamente y no por aire. En Colombia existen actualmente 15 canales de televisión. Entre ellos los más importantes son RCN, CARACOL y CANAL UNO.

2.3.3.3 Los medios digitales interactivos: La llegada del Internet ha dado origen a una nueva forma de medios publicitarios. La evolución que han sufrido desde sus inicios en la década de los años 80's ha sido total.

Hoy día, los usuarios de Internet han encontrado en esta herramienta la mejor manera de comunicarse e informarse. Precisamente ese mercado fue abordado por los cibernautas desde un principio y las grandes compañías no tardaron en involucrarse con este nuevo medio que vendría a revolucionar las telecomunicaciones.

Las opciones en Internet van desde simples Banners, hasta microvideos publicitarios. La evolución del Internet se da a medida que las redes informáticas evolucionan y permiten una mayor velocidad en la tasa de transmisión de datos.

La efectividad de este medio radica en la gran población que puede acceder a él. Si un banner es publicado y visto por 200 personas y de esa cantidad 10 siguen el enlace, se puede hablar de un 5% de efectividad del medio.

Figura 5. Medios Digitales



En los últimos años se ha visto una migración por parte de los medios impresos a los digitales. En un caso más específico, los diarios del mundo han entrado al

mundo de la informática y la Internet para publicar sus noticias y así poder atender la demanda. Aunque se elimina la imprenta, estos medios no han podido encontrar el modelo económico que les represente las mismas utilidades que obtenían con el medio impreso.

2.3.3.4 Los medios externos: Se enmarcan las vallas publicitarias o los dibujos realizados en grandes paredes que son visibles a los consumidores. Este tipo de medio es muy utilizado en campañas electorales donde lo más importante para un candidato es la imagen.

2.3.4 Marketing De Proximidad

Llamado también “Movable Marketing” o “Marketing 2.0”, es una estrategia que hace envíos de publicidad a los teléfonos celulares a través de Bluetooth. Esta estrategia de marketing es la que enmarca el desarrollo de este proyecto.

El poder enviar publicidad directamente a los clientes para que acepten la invitación a recibirla en su celular, es una muestra del interés que ellos muestran hacia el producto.

Figura 6. Marketing de Proximidad



El sistema permite el envío de información constante a dispositivos Bluetooth que estén en el área, y será enviada a los usuarios que acepten el envío. Los dispositivos Bluetooth están integrados a los dispositivos móviles que están en el mercado, así en un área de espacio público por ejemplo de un centro comercial, se puede iniciar una campaña enviando información sobre locales comerciales cercanos.

El envío de la información podrá realizarse las 24 del día, no se gastará en impresión o papel, conservando así el medio ambiente y se podrán llevar datos estadísticos sobre la cantidad de archivos enviados. Según el Dr Juan Carlos Mario "*Las ventajas son innumerables*" [6]

3. ESPECIFICACIONES DEL SOFTWARE DESARROLLADO

3.1 DESCRIPCIÓN DEL SOFTWARE

El Marketing de proximidad está orientado al aprovechamiento de las tecnologías relacionadas con la transferencia de información a través de dispositivos móviles, y el software que a continuación se presenta como resultado del proyecto en mención, está enfocado a apoyar la expansión del uso de tecnologías asociadas con el envío de archivos multimedia (Videos, Música, Fotos) vía Bluetooth, a usuarios de teléfonos móviles que tengan incorporado este dispositivo en ellos.

El sistema evita aplicar en sus envíos el comportamiento que demuestran los Sistemas de Mensajería Multimedia - MMS, los cuales realizan el envío de información a través de las redes de los operadores celulares directamente a sus usuarios, evitando así la interacción con el usuario que finalmente debe aceptar la recepción de la mensajería a su celular obligatoriamente.

El Marketing de proximidad permite una interacción con el usuario final, ya que al momento de realizarse un enlace entre dos dispositivos que manejen Bluetooth, se facilita al dispositivo receptor, que participe de una consulta orientada a preguntar al usuario, si acepta o no la transferencia de datos.

Para llevar a cabo esta consulta de aceptación sobre la transferencia de datos, en el software desarrollado se diseñó una interfaz amigable, de tal manera que le permita al usuario identificar cada uno de los siguientes aspectos² que toma en cuenta el sistema:

² Estos pasos se mostrarán con detalle más adelante.

- Carga del archivo (Publicidad) que se desee enviar.
- Programación del tiempo de emisión.
- Inicio del sistema y la transmisión de datos.
- Almacenamiento de configuraciones ya realizada, de tal forma que puedan ser utilizadas posteriormente. (Reciben el nombre de *Campañas*).

3.2 ANÁLISIS DE REQUERIMIENTOS.

- A. La aplicación desarrollada en el presente proyecto tuvo en cuenta de manera inicial los siguientes aspectos, que facilitan su apropiación a los usuarios:
- Portabilidad.
 - Ligera y con una interfaz amigable.
 - Entendible para el usuario final.
- B. Sumado a esto, para el desarrollo del prototipo software se requería utilizar una plataforma que presentará como principal ventaja la portabilidad, y Java brinda esa oportunidad ya que presenta las siguientes características:
- A través de su Máquina Virtual (VM) permite ejecutar aplicaciones Java en cualquiera de los sistemas operativos soportados por la plataforma.
 - Es un potente lenguaje de programación basado en C++.
 - Las aplicaciones desarrolladas bajo este lenguaje de programación son de bajo consumo de recursos máquina.
- C. La interfaz se creó de tal forma que fuera entendible a la hora de darle manejo al software desarrollado, permitiendo así visualizar todas sus funciones claramente.

D. La base de datos que exigía el sistema, debía ser pequeña y embebida, ayudando de esta manera a la portabilidad del sistema. En este sentido, *Apache Derby* fue la opción más apropiada ya que no necesita de un servidor dedicado para funcionar correctamente y se adapta a las necesidades del prototipo.

3.3 ARQUITECTURA SOFTWARE.

Debido a que desde el inicio se planteó la idea de desarrollar una aplicación de escritorio amigable, se eligió la Arquitectura de tres capas, ya que es la que más se acomoda a los requerimientos del sistema, los cuales exigen como principales puntos de atención los siguientes aspectos:

- Trabajar con una base de datos en la que la información persista.
- Diseñar una interfaz gráfica entendible y amigable.
- Facilitar la interacción del usuario con el prototipo final.

3.3.1 Arquitectura 3 Capas.

Se llama arquitectura 3 capas, por estar separada en tres características importantes que se mencionan a continuación:

- **Capa Presentación:** O interfaz del usuario. Está conformada por los formularios y los controles que se encuentran en los formularios. Es la capa que interactúa con los usuarios.
- **Capa Negocio:** Se reciben las peticiones del usuario y se envían las respuestas tras el proceso.

- **Capa de datos:** Contiene las clases que interactúan con la base de datos. Permite realizar todas las operaciones con la base de datos de forma transparente para la capa Negocio.

El prototipo encaja dentro de esta arquitectura ya que presenta todas las características de la misma.

3.4 HERRAMIENTAS SOFTWARE NECESARIAS

Para la construcción de este sistema se hizo necesario acudir a algunas herramientas que facilitaron el funcionamiento del sistema desarrollado. A continuación se enumeran cada una de éstas herramientas y su respectiva descripción.

3.4.1 Bluecove³

Es una implementación del JSR -82⁴ para Java 2 Standard Edition – J2SE, que posee las clases y los métodos para poder acceder a los protocolos java.Bluetooth y Java.Obex.

Este framework⁵ soporta Mac OSX, WIDCOMM, BluSolei, Windows nativo y Linux con BlueZ. Está bajo la licencia “The Apache Software Licence 2.0”, y funciona en cualquier máquina virtual de java (JVM) desde la versión 1.1.

³ Para obtener más detalles de éste framework, visitar el enlace: <http://bluecove.org/> donde se puede encontrar la descripción detallada del proyecto, documentación y descarga del mismo.

⁴ Librerías Java para Bluetooth.

⁵ Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en lo cual otro proyecto de software puede ser organizado y desarrollado.
<http://www.wikipedia.com>

Se pueden distinguir cuatro elementos principales de Bluecove, que se presentan en la siguiente tabla:

Tabla 1. Servicios Bluecove

ELEMENTO DE BLUECOVE	DESCRIPCIÓN
getLocalDevice	Pertenece a la clase LocalDevice y proporciona la capacidad de generar un objeto que entrega la identificación del dispositivo Bluetooth.
DiscoveryAgent	Es una interfaz que provee métodos para descubrir dispositivos y servicios.
DiscoveryListener	Esta clase proporciona el método deviceDiscovered() que se utiliza cada vez que se encuentra un dispositivo en un proceso de búsqueda.
ServiceRecord	Es una clase que proporciona métodos para registrar el dispositivos y los servicios encontrados

La interfaz más importante de este framework es la llamada DiscoveryListener, que permite ser implementada por una clase encargada de recibir los eventos de reconocimiento dispositivos y servicios. Esta interfaz provee cuatro métodos, dos para descubrir dispositivos y dos para identificar servicios. A continuación se presenta una breve descripción de los métodos utilizados por ésta interfaz.

- **deviceDiscovered(RemoteDevice btDevice, DeviceClass cod)** Llamado cuando un dispositivo es encontrado durante una búsqueda.
- **inquiryCompleted(int discType)** Llamado cuando una búsqueda es completada.

- **servicesDiscovered(int transID, ServiceRecord[] servRecord)** Llamado cuando servicio(s) son encontrados durante una búsqueda de servicios.
- **serviceSearchCompleted(int transID, int respCode)** Llamado cuando una búsqueda de servicios es completada o fue terminada por cause de un error.

3.4.2 Apache Derby⁶

Apache Derby es un subproyecto de base de datos de Apache, es un sistema gestor de base de datos relacional de código abierto, implementado completamente en Java y se encuentra bajo la licencia "Apache licence, version 2.0".

Tiene un tamaño bastante reducido (aproximadamente 2MB), y posee la gran ventaja de contener un controlador embebido, que permite tener al sistema gestor, embebido dentro de cualquier aplicación java, haciendo que la base de datos sea completamente portable y evitando que este necesariamente en modo Cliente/servidor. Aunque también posee la capacidad para funcionar en modo cliente/servidor, si no se desea que la base de datos esté embebida dentro de la aplicación java.

Es fácil de instalar, configurar y utilizar, está basado en los estándares de Java, JDBC y SQL.

La forma en que se registra el driver embebido para cualquier aplicación java es muy sencilla basta con solo una línea de código como la que se muestra a continuación:

Class.forName("org.apache.derby.jdbc.EmbeddedDriver").newInstance();

⁶ Para información de este proyecto y descarga del mismo se puede visitar el sitio oficial de Apache Derby en <http://db.apache.org/derby/>.

Para crear una conexión se hace con la siguiente instrucción:

```
Connection conexion =  
DriverManager.getConnection("jdbc:derby:oddblueDB;create=true",  
"oddblue", "");
```

Donde el parámetro del método *getConnection* presenta algunas partes que serán comentadas a continuación:

“**create=true**” significa que si la base de datos no existe entonces ésta será creada, si se está seguro de que la base de datos ya está creada entonces se cambia “true” por “false” en el argumento del parámetro.

“**oddblueDB**” significa el nombre de la base de datos.

Los últimos dos campos que van entre comillas corresponden al nombre de usuario y contraseña para la base de datos a la cual se desea hacer la conexión.

Para las consultas, actualizaciones, inserciones y ejecuciones de demás instrucciones SQL simplemente se utiliza un objeto de la clase **Statement** y mediante los métodos **execute(sql)** y **executeQuery(sql)**, donde **sql** es un String con la instrucción SQL que se desea ejecutar.

3.4.3 Jfreechart

Es una librería para gráficos escrita 100% en Java, facilita mostrar gráficos de calidad profesional en aplicaciones, ya sean web o de escritorio.

Entre las características principales de esta biblioteca se encuentran:

- Un API consistente y bien documentado con soporte para un amplio rango de tipos de gráficos.

- Un diseño flexible fácilmente extendible, y la posibilidad de ser usado tanto en tecnologías de servidor (aplicaciones Web) y de cliente (Swing, por ejemplo).
- Soporte para varios tipos de salida, incluyendo componentes Swing, archivos de imagen de formato PNG y JPEG, y formatos gráficos de vectores (incluyendo PDF, EPS y SVG).
- JFreeChart es open source, más específicamente Software Libre (está distribuido bajo la licencia de software libre Lesser General Public License - LGPL, que permite el uso en aplicaciones propietarias).
- Requiere de la plataforma Java 2 (JDK versión 1.3 o superior)

A continuación se presenta la Figura 10 que muestra ejemplos de los diferentes diagramas que pueden ser ejecutados con esta librería. Dando a conocer que no sólo permite generar estadísticas utilizando el típico pastel, sino que puede representar una gran cantidad de figuras dependiendo de las necesidades del programador. [7]

Figura 7. Ejemplos de Figuras JfreChart



3.5 PLATAFORMA DE DESARROLLO (IDE)

El Integrated Development Environment ó Entorno de Desarrollo Intregado – IDE, seleccionado para el prototipo fue ECLIPSE, ya que permite crear aplicaciones en diferentes lenguajes de programación como Java, PHP, C/C++ o Python.

3.5.1 Eclipse.

Dentro de los entornos de desarrollo más comunes y completos están los siguientes IDE: Netbeans y Eclipse. Sin embargo se ha elegido el IDE Eclipse por su fácil manejo y por el previo conocimiento que los autores del proyecto tienen sobre éste. Una de sus características principales según el Ingeniero Senior David Galardo⁷ es que *“Eclipse se organiza en torno al concepto de una perspectiva que hace más fácil conseguir el trabajo hecho, porque los instrumentos adecuados (y sólo las herramientas apropiadas) están a la mano.”*

Figura 8. IDE - Logo de Eclipse



Eclipse, es una plataforma de software de código abierto. Típicamente se ha dado a conocer para desarrollar entornos integrados de desarrollo o IDE, así como el IDE de Java llamado *Java Development Toolkit* (JDK) y el compilador *Eclipse Compiler for Java* (ECJ).

⁷ David Gallardo. Co-Autor *“Eclipse in Action: A Guide For Java Developers” (Eclipse en acción: Guía para desarrolladores Java)*.

Eclipse se ha formado como un ecosistema abierto independiente, debido a estar libre de regalías y ser una plataforma universal para integración de herramientas de desarrollo software.

Las herramientas de Eclipse, dan a los desarrolladores la libertad de escoger un IDE Multilenguaje, Multiplataforma y Multiproveedor de ambiente. Eclipse provee un Plug-in basado en un Framework que facilita la creación, integración y utilización de herramientas software, permitiendo al mismo tiempo ahorrar tiempo y dinero.

Gracias a esta herramienta, los desarrolladores de software pueden aprovechar la reutilización de la plataforma y concentrarse en las competencias básicas para crear nuevas tecnologías de desarrollo. [8]

4. DISEÑO DEL PROTOTIPO

El diseño de este prototipo debía ser realizado pensando en el usuario común, cualquier persona con conocimientos básicos en sistemas podría estar capacitado para poder utilizar el software sin mayores complicaciones.

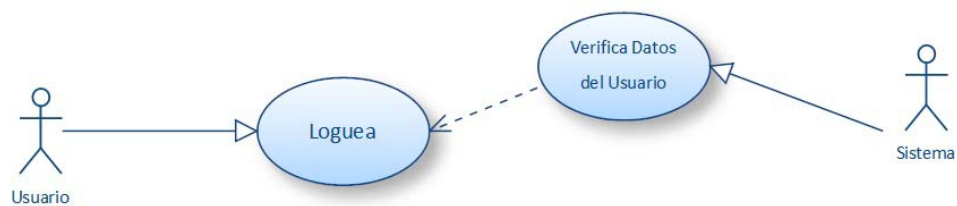
Se aplican modelos UML para obtener una visión general sobre las acciones que debe realizar el prototipo y así buscar los actores que entran a participar de este proceso.

4.1 DIAGRAMAS DE LENGUAJE DE MODELADO UNIFICADO - UML

4.1.1 Logueo

La primera interacción que tendrá el usuario con el prototipo es en la parte del "LOGIN". Es allí donde se cargarán los datos del usuario del prototipo.

Figura 9. Diagrama UML de Logueo

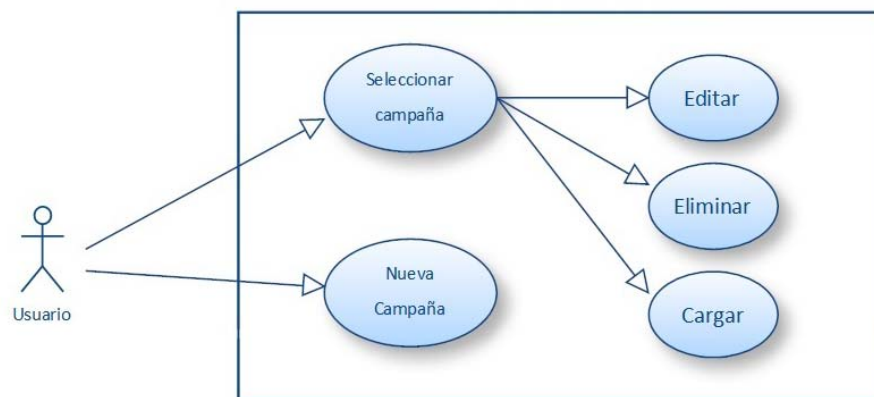


La figura 9 muestra como el usuario hace el ingreso al prototipo digitando su contraseña, es decir “loguearse” y tener acceso a él. El sistema hace una verificación de los datos, aceptando ó no su ingreso.

4.1.2 Configuración De Campañas.

Una vez el usuario ingresa al sistema debe configurar una campaña. Se debe entender por campaña aquella configuración que será cargada en el sistema para su distribución por Bluetooth.

Figura 10. Diagrama UML de Configuración de Campañas



En este caso el usuario tendrá dos opciones:

- Seleccionar campaña.
- Nueva campaña.

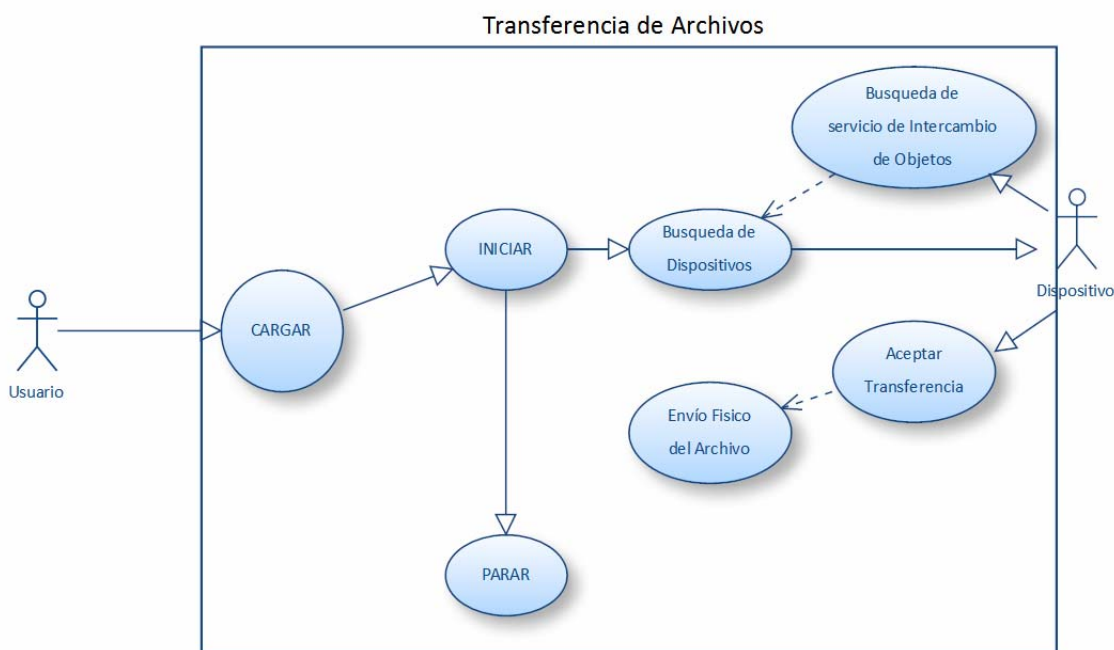
Si el usuario selecciona la opción de nueva campaña podrá:

- Editar.
- Eliminar.
- Cargar.

4.1.3 Carga Y Transferencia De Archivos

Una vez se ha configurado la campaña se debe hacer una carga al sistema, de tal forma que este preparado para iniciar la transferencia de archivos. El diagrama UML es el siguiente.

Figura 11. Diagrama UML de Carga y Transferencia de Archivos

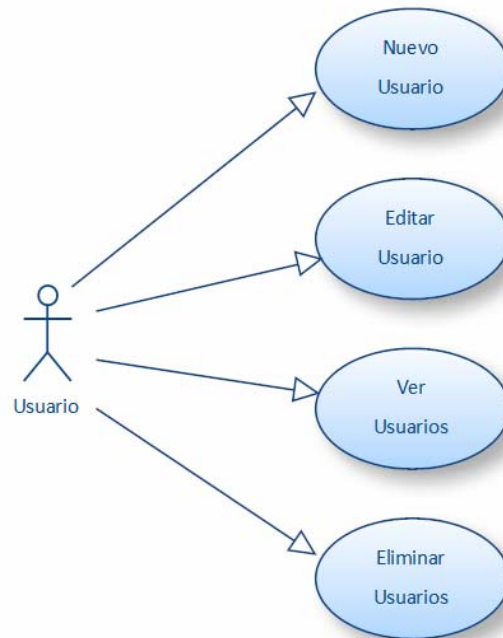


En este diagrama se muestra la interacción del sistema con los demás dispositivos que están en el área de cobertura.

4.1.4 Usuarios

Los usuarios del sistema deben hacer un registro personal para poder acceder a los componentes que hacen parte del prototipo. Una vez registrados tendrán la opción de crear campañas.

Figura 12. Diagrama UML Acciones de los Usuarios



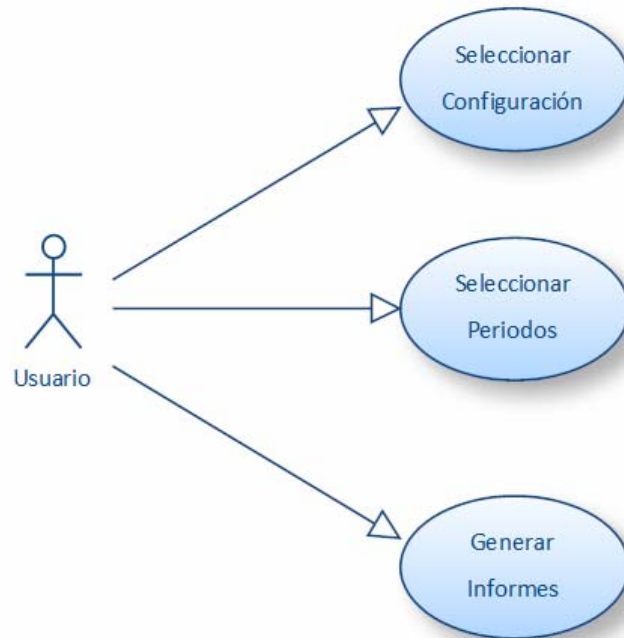
El diagrama muestra las diferentes acciones que puede ejecutar el usuario del sistema en la interfaz de usuarios.

4.1.5 Reportes

Es uno de los componentes más importantes del sistema. Con esta opción, el usuario podrá conocer el funcionamiento y la cantidad de descargas que se registran de determinada campaña.

Para ello se realizó el siguiente diagrama.

Figura 13. Diagrama UML de los Reportes

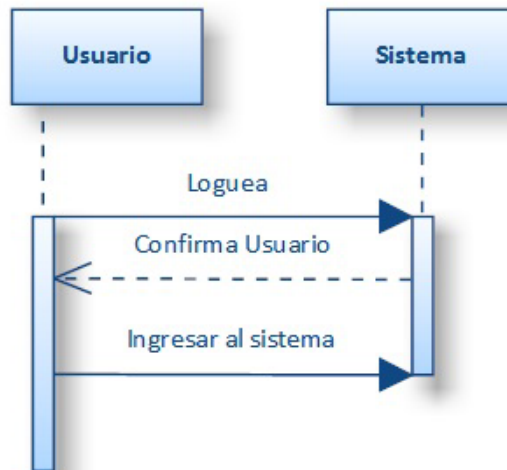


El usuario debe llevar a cabo las acciones paso a paso. En este caso el sistema no interviene, solamente ejecuta las peticiones que el usuario solicite.

4.2 DIAGRAMAS DE SECUENCIA.

Una vez que se conocen los actores del sistema, se procede a determinar los casos de uso. Tal y como se mostraron los diagramas de UML, el primer diagrama de secuencia es el de Logueo.

Figura 14. Diagrama de Secuencia del Logueo



Los demás diagramas de secuencia se encuentran en el Anexo 1.

4.3 ESTRUCTURA Y DATOS

4.3.1 Base De Datos

Para el manejo de los requerimientos del sistema se diseñó un modelo sencillo, básico y funcional para la base de datos. El modelo consta de tres tablas:

- **CAMPANHA:**

En esta tabla se guardan las campañas que se han configurado en el sistema, donde cada una de ellas representa un registro en esta tabla. Aquí se guarda el nombre de la campaña, una llave numérica que internamente identifica la campaña, la ruta del archivo que se va a enviar, el periodo (dado en minutos) en el que se repetirá el proceso de búsqueda de dispositivos y envío del archivo, la fecha de inicio de la campaña y las notas o comentarios acerca de la campaña.

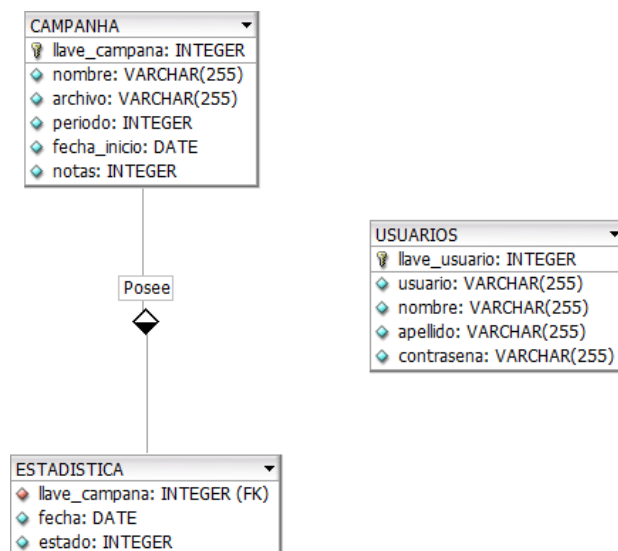
- **ESTADISTICA:**

Esta tabla guarda registros estadísticos acerca de los aciertos y/o rechazos de los archivos enviados en cada una de las campañas configuradas. La tabla se compone de una llave foránea a la tabla CAMPANHA (relaciona la campaña a la cual se está guardando el registro académico), la fecha en la que se realiza la solicitud de envío del archivo y, por último se tiene un campo numérico que representa el estado de la solicitud, en este caso 0 indica “aceptado” y 1 indica “rechazado”.

- **USUARIOS:**

Mediante esta tabla se maneja la información acerca de los usuarios habilitados para utilizar el sistema, se compone de una llave numérica para la identificación interna del usuario, el usuario que se identifica en el sistema, el nombre del usuario, el apellido del usuario y la contraseña con la que el usuario puede acceder al sistema.

Figura 15. Diseño de la base de datos para el manejo de campañas, estadísticas y usuarios

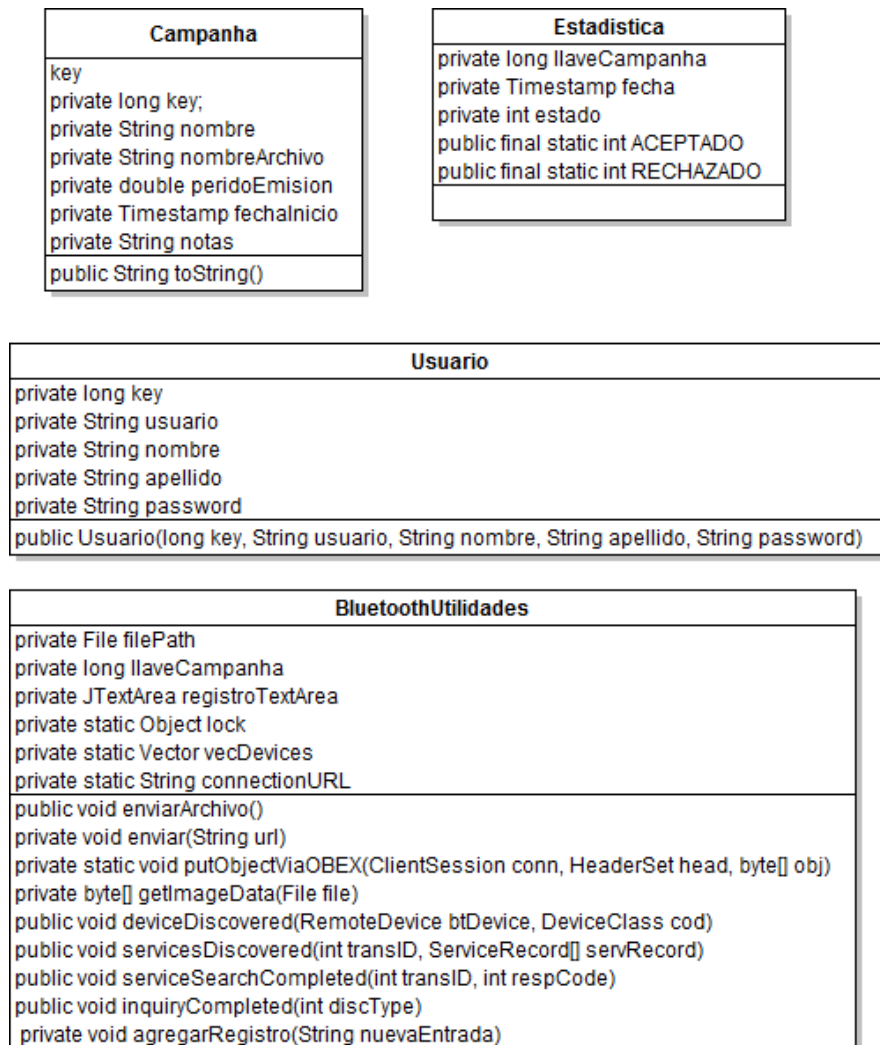


4.3.2 Core Del Sistema:

El paquete principal del sistema se sustenta en las clases principales que modelan los objetos representativos del sistema, tal y como son las clases: Estadística, Usuario y Campanha.

La utilidad principal para el reconocimiento de dispositivos y envío de archivos mediante Bluetooth se condensa en la clase BluetoothUtilidades.

Figura 16. Core del Sistema



4.3.3 Interfaz Grafica Gui:

Estos diagramas se omiten puesto que hacen referencia al manejo de eventos en la interfaz gráfica de usuario a través de Swing.

4.3.4 Manejo Base De Datos:

Para la interacción del sistema con la base de datos se creó una clase especial que definiera estas transacciones y se llamó *BaseDeDatos*. La particularidad de esta clase es que posee atributos y métodos estáticos, es decir, que no se necesita crear un objeto para poder utilizar los métodos que contiene.

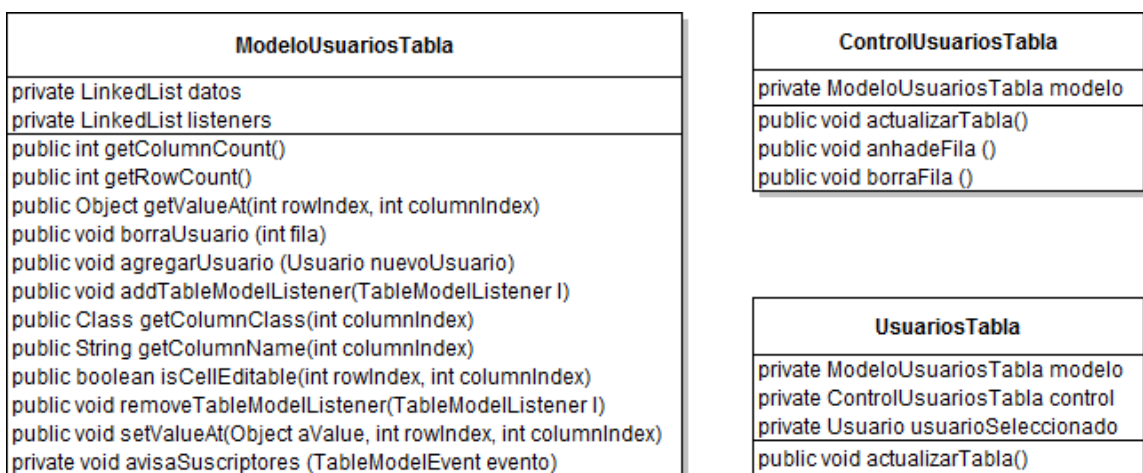
Principalmente se caracteriza por manejar los procesos básicos del mantenimiento de la base de datos, tales como; inserción de datos, borrado y edición de los mismos. Otra funcionalidad interesante en esta clase es que como se está utilizando el driver embebido para el manejador de la base de datos Apache Derby, posee métodos para el registro del driver, la creación de la base de datos y la creación de las tablas.

Figura 17. Clase Base de Datos

BaseDeDatos
<pre>private static Connection conexion private static Statement statement private static ResultSet resultSet public static void IniciarBaseDeDatos() public static boolean existeTablaCampana() public static void crearTablaCampana() public static List<Campahna> obtenerCampanas() public static int maximaLlaveCampana() public static boolean guardarCampana(Campahna campana) public static boolean actualizarCampana(Campahna campana) public static boolean borrarCamapana(long campanaKey) public static boolean existeTablaUsuarios() public static void crearTablaUsuarios() public static boolean IngresaUsuario(String usuario, String password) public static List<Usuario> obtenerUsuarios() public static int maximaLlaveUsuario() public static boolean guardarUsuario(Usuario usuario) public static boolean actualizarUsuario(Usuario usuario) public static boolean borrarUsuario(long usuarioKey) public static boolean existeTablaEstadistica() public static void crearTablaEstadistica() public static List<Estadistica> obtenerEstadisticas() public static List<Estadistica> obtenerEstadisticasPorCampanha(long llaveCampanha) public static List<Estadistica> obtenerEstadisticasPorPeriodo(long llaveCampanha, Timestamp fechaInicio, Timestamp fechaFinal) public static boolean guardarEstadistica(Estadistica estadistica) public static void cerrarConexion()</pre>

4.3.4.1 Tabla De Usuarios: Uno de los componentes más interesantes pero a la vez complejo del paquete de Swing en java es la tabla (JTable). Para obtener un óptimo rendimiento y excelente manejo de este componente es conveniente implementar clases para el modelo y control de la tabla. En este paquete se encuentran tres clases que se implementan para el correcto manejo de los usuarios mediante una tabla.

Figura 18. Tabla de Usuarios - JTable



4.4 PROGRAMACIÓN

4.4.1 Proceso De Envío De Archivos Mediante La Tecnología Bluetooth

El *framework* BlueCove implementa una serie de clases e interfaces que facilitan el proceso de detección de los dispositivos que tiene el Bluetooth dentro del radio de alcance, una de las utilidades que ofrece este *framework* es el reconocimiento de los servicios que se encuentran activos en el dispositivo detectado y además, permite el envío de archivos y datos mediante los protocolos de intercambio de la pila Bluetooth.

4.4.1.1 Búsqueda de Dispositivos: La búsqueda de dispositivos es el primer paso en el proceso es la detección de los dispositivos, esto es posible mediante la implementación de la interfaz “*DiscoveryListener*”, contenida en el *framework* BlueCove. El código que realiza este proceso de descubrimiento de dispositivos es el siguiente:

```
//objeto usado para la espera
private static Object lock=new Object();
//vector que contiene los dispositivos descubiertos
private static Vector vecDevices=new Vector();
// mostrar dispositivo local con su nombre y direccion
LocalDevice localDevice = LocalDevice.getLocalDevice();

System.out.println("Address:
    "+localDevice.getBluetoothAddress());
System.out.println("Name: "+localDevice.getFriendlyName());

//encontrar dispositivos
DiscoveryAgent agent = localDevice.getDiscoveryAgent();

//empezar busqueda
System.out.println("Starting device inquiry...");
agent.startInquiry(DiscoveryAgent.GIAC, this);

try {
    synchronized(lock){
        lock.wait();
    }
}
catch (InterruptedException e) {
```

```

        e.printStackTrace();
    }

    //busqueda finalizada
    System.out.println("Device Inquiry Completed. ");

    //obtener el numero de dispositivos encontrados en el vector de dispositivos
    int deviceCount=vecDevices.size();

    if(deviceCount <= 0){
        System.out.println("No Devices Found .");
    }

    /**
     * Called when a bluetooth device is discovered.
     * Used for device search.
     */
    public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
        //add the device to the vector
        if(!vecDevices.contains(btDevice)){
            vecDevices.addElement\(btDevice\);
        }
    }

    /**
     * Called when the device search is over.
     */
    public void inquiryCompleted(int discType) {
        synchronized(lock){
            lock.notify();
        }
    }

```

```
}  
  
}
```

4.4.1.2 Verificación de servicios en el dispositivo: Después de encontrar los dispositivos disponibles, en cada uno de estos dispositivos se verifica que tenga activo el servicio requerido para el intercambio de objetos (que pueden ser; imágenes, videos, archivos, tarjetas virtuales o citas).

Este servicio se encuentra en el perfil de Bluetooth conocido como "Object Push".

Un perfil Bluetooth es la descripción de ciertas características que son propias para cada dispositivo y que son utilizadas para la comunicación en forma estandarizada. El perfil "Object Push" se caracteriza porque el emisor es quien inicia siempre la comunicación y utiliza las API's de OBEX para las operaciones de; conexión y desconexión, envío, recepción y cancelación.

Este proceso se realiza mediante con el siguiente código:

```
private static String connectionURL;  
for (int i = 0; i <deviceCount; i++) {  
    RemoteDevice remoteDevice=(RemoteDevice)vecDevices.elementAt(i);  
    UUID[] uuidSet = new UUID[1];  
    uuidSet[0]=new UUID("0008",true);  
  
    System.out.println("\nSearching for service...");  
    agent.searchServices(null,uuidSet,remoteDevice,this);  
  
    try {  
        synchronized(lock){
```

```

        lock.wait();
    }
}
catch (InterruptedException e) {
    e.printStackTrace();
}

if(connectionURL==null){
    System.out.println("Device does not support Object Push.");
}

else{
    System.out.println("Device supports Object Push.");

    enviar(connectionURL);
}
}

/**
 * Called when a bluetooth service is discovered.
 * Used for service search
 */
public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
    if(servRecord!=null && servRecord.length>0){
        connectionURL=servRecord[0].getConnectionURL(
            ServiceRecord.NOAUTHENTICATE_NOENCRYPT,false);
    }
    synchronized(lock){
        lock.notify();
    }
}

```

```

}

/**
 * Called when the service search is over.
 */
public void serviceSearchCompleted(int transID, int respCode) {
    synchronized(lock){
        lock.notify();
    }
}

```

El método `enviar(connectionURL)` método será descrito en el siguiente paso del proceso.

4.4.1.3 Crear Conexión: Si el dispositivo soporta el perfil “Object Push”, ya se encuentra preparado para crear la conexión desde el emisor. Para ello se crea un objeto de la clase `javax.obex.ClientSession` que representa la sesión, ésta es creada mediante la dirección URL que se obtiene en el paso anterior. El código que crea la conexión es el siguiente:

```

ClientSession sesión;
session = (ClientSession) Connector.open(url, Connector.READ_WRITE);

```

Seguidamente se crean las cabeceras para la comunicación mediante OBEX, esto se hace a través de la interfaz `HeaderSet`, la cual define métodos que establecen y dan los valores a las cabeceras OBEX.

```

// create a new set of headers
HeaderSet headers = session.createHeaderSet();

```

Para el enviar el archivo, éste se convierte en un arreglo de bytes mediante el siguiente método:

```
private byte[] getImageData(File file) {  
    InputStream in;  
    try {  
        in = new FileInputStream(file);  
    } catch (FileNotFoundException e1) {  
        // TODO Auto-generated catch block  
        e1.printStackTrace();  
        return null;  
    }  
    byte[] buff = new byte[1024];  
    ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);  
    try {  
        while (true) {  
            int length = in.read(buff);  
            if (length == -1) {  
                break;  
            }  
            baos.write(buff, 0, length);  
        }  
    } catch (IOException e) {  
        System.err.println(e);  
        return null;  
    }  
    return baos.toByteArray();  
}
```

4.4.1.4 Envío del archivo: Después de haber preparado las cabeceras Obex, se agrega el archivo a éstas cabeceras y seguidamente se ejecuta el envío. Mediante

la interfaz "Operation" se puede hacer el envío de las cabeceras Obex. Para ello la interfaz Operation posee un método llamado PUT para enviar estas cabeceras.

El código para este paso es el siguiente:

```
headers.setHeader(headers.NAME, this.filePath.getName());  
headers.setHeader(headers.TYPE, this.filePath.getName());  
putObjectViaOBEX(session, headers, imgData);
```

```
private static void putObjectViaOBEX(ClientSession conn, HeaderSet head,  
byte[] obj) throws IOException {
```

```
    // Include the length header
```

```
    head.setHeader(head.LENGTH, new Long(obj.length));
```

```
    conn.connect(head);
```

```
    // Initiate the PUT request
```

```
    Operation op = conn.put(head);
```

```
    // Open the output stream to put the object to it
```

```
    DataOutputStream out = op.openDataOutputStream();
```

```
    // Send the object to the server
```

```
    out.write(obj);
```

```
    // End the transaction
```

```
    out.close();
```

```
    op.close();
```

```
    conn.close();
```

5. DESARROLLO DEL PROTOTIPO

5.1 DESCRIPCIÓN DEL PRIMER PROTOTIPO

El primer paso consistió en realizar una conexión con los servicios Obex habilitados entre dos dispositivos Bluetooth, para así, enviar información sobre una campaña publicitaria a través de Bluetooth. Se inició con la búsqueda y reconocimiento de los dispositivos que están en el área de descarga.

Para realizar la búsqueda de dispositivos Bluetooth cercanos se utilizó el Objeto *DiscoveryListener*. Es una interfaz que se utiliza para que el dispositivo notifique eventos a la aplicación una vez sea descubierto un dispositivo Bluetooth en el área de cobertura.

Estos son los cuatro métodos de la interfaz *DiscoveryListener*:

Tabla 2. Método de DiscoveryListener

MÉTODOS	DESCRIPCIÓN
deviceDiscovered()	Procesos de búsqueda de dispositivos. Cuando se encuentra un dispositivo se llama <i>deviceDiscovered()</i> . Una vez se haya concluido la búsqueda de dispositivos se llama <i>inquiriComplete()</i> .
InquiriComplete()	
servicesDiscovered()	Mediante estos procesos, una vez se tengan identificados los dispositivos disponibles, se averigua por los servicios disponibles en cada uno de los dispositivos.
serviceSearchCompleted()	

En este prototipo se presenta una interfaz sencilla donde el usuario tiene la capacidad de escoger un archivo ubicado físicamente en la maquina donde se corre el aplicativo, para luego ser enviado mediante Bluetooth por petición del usuario, a los dispositivos que estén al alcance dentro del área de descarga.

El proceso en teoría es sencillo, primero se ubica la ruta del archivo mediante un dialogo de navegación hasta el archivo seleccionado, luego el *path* o ruta del archivo se utiliza para cargar físicamente el archivo, a continuación éste archivo se convierte en un *array* de bytes que, posteriormente se envía en el formato de transmisión de objetos OBEX.

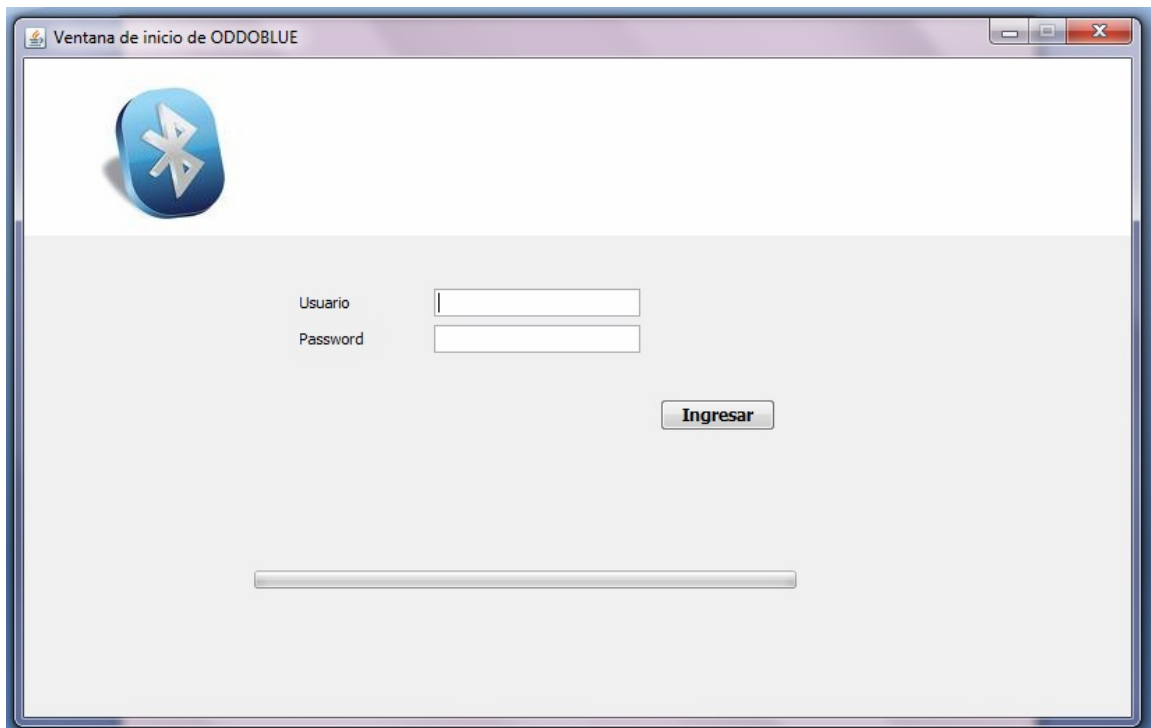
5.2 DESCRIPCIÓN DEL PROTOTIPO FINAL

Se adapta el conjunto de servicios desarrollados en el primer prototipo en una Interfaz Gráfica de Usuario - GUI. El Software que se desprende del desarrollo de este proyecto está representado por una interfaz gráfica sencilla que puede ser utilizada por el usuario final de manera fácil, rápida e intuitiva.

Éste prototipo se desarrolló enteramente en Java bajo la especificación JSR-82 Bluetooth, para que su ejecución pueda efectuarse desde cualquier Sistema Operativo.

5.2.1 Login

Figura 19. Ventana de Login



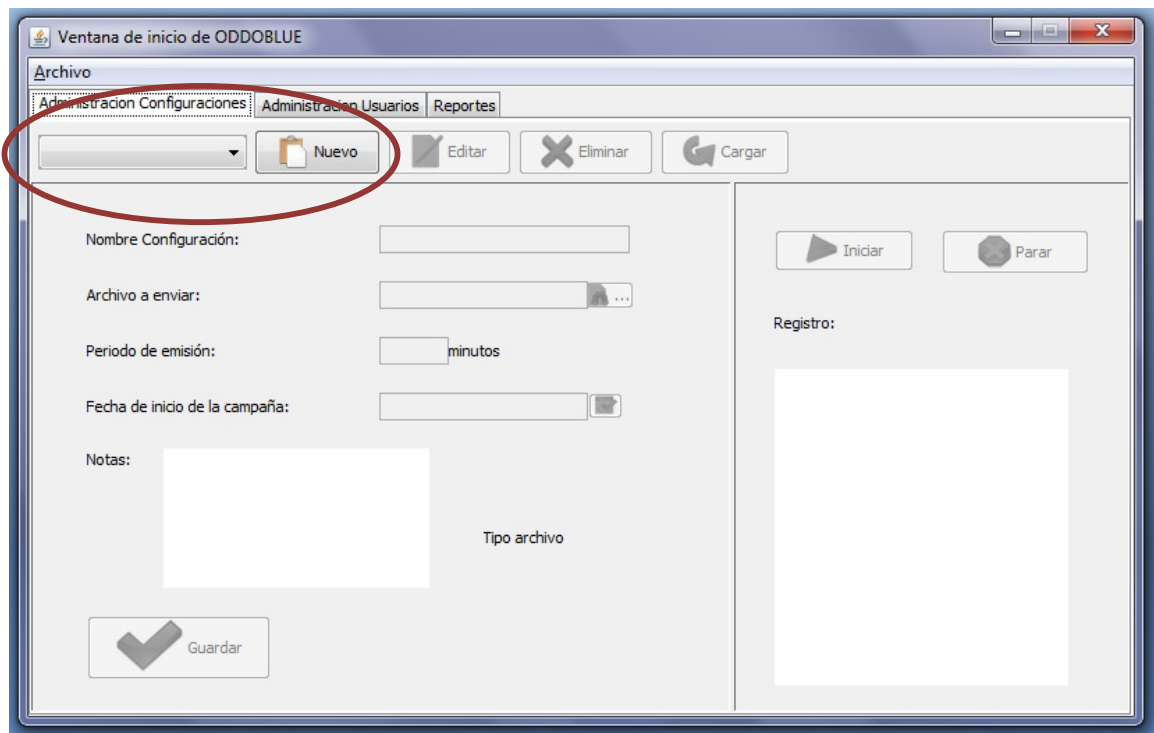
Al ejecutar el programa aparecerá el formulario LOGIN. Como su nombre lo indica es aquel formulario que permite el ingreso validado al prototipo.

Los datos tanto del administrador como de los usuarios que estén listados en la base de datos, podrán utilizar los diferentes componentes del prototipo. El administrador, sin embargo, es el único que puede modificar el listado de los usuarios que podrán intervenir en el sistema.

5.2.2 Prototipo En Ejecución

5.2.2.1 Panel Principal

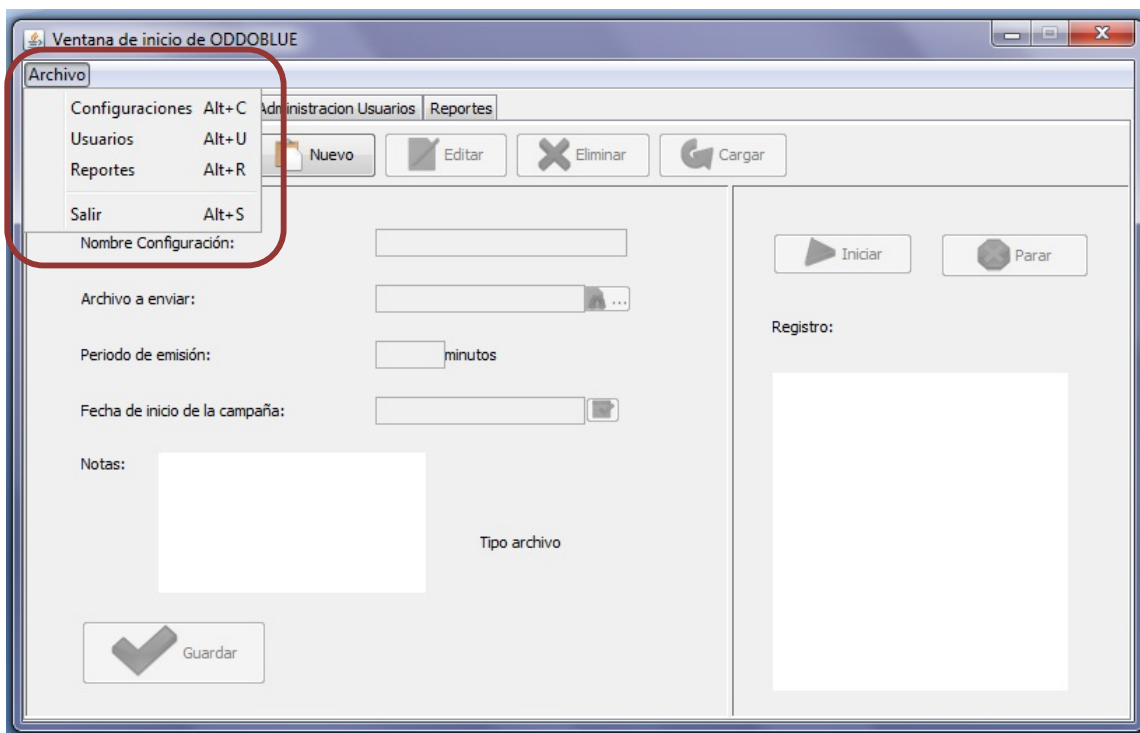
Figura 20. Ventana de Inicio de ODDOBLUE



En esta ventana aparecen todos los elementos del prototipo aunque están desactivados, esto es útil en el sentido de la seguridad que tiene el prototipo, ya que no permite modificar los elementos que están en el panel principal. El botón "Nuevo" y la pestaña desplegable son los únicos botones activos en el formulario.

Archivo es un menú desplegable, que puede transportar al usuario sobre las diferentes pestañas que trae el formulario además de la opción de salir.

Figura 21. Ventana ODDOBLUE - Menú Archivo

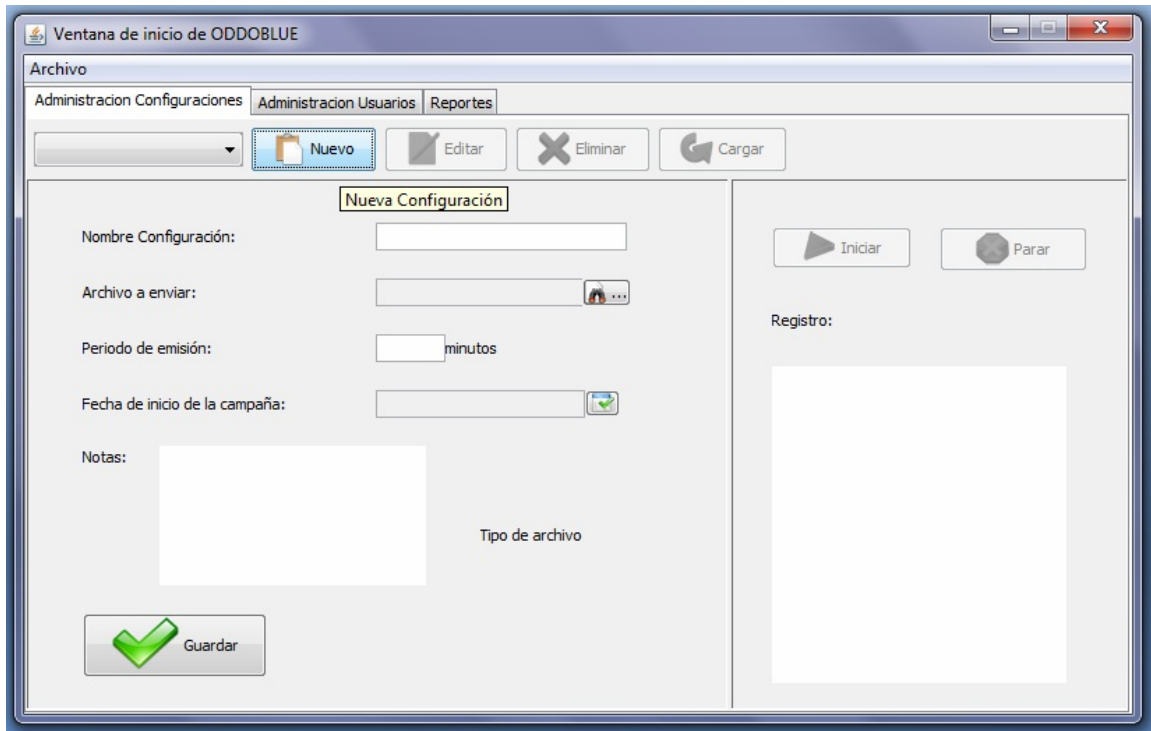


En este proyecto se tratará el término “Campaña” como el todo de la configuración de los componentes que trae el formulario y, que serán guardados en la base de datos. Se guardará el nombre, la ruta donde se encuentra el archivo a enviar, la periodicidad con que se desee realizar la búsqueda de dispositivos, la fecha de inicio y un aparte de notas para que el usuario pueda hacer una pequeña descripción de la configuración que se acaba de realizar.

Campaña hace referencia a “Campaña publicitaria”, ya que el prototipo está enfocado al envío de publicidad, aunque puede ser configurada para enviar cualquier clase de archivo y no necesariamente que esté relacionada con el mercadeo.

5.2.2.2 Configuración De La Campaña

Figura 22. Nueva Campaña



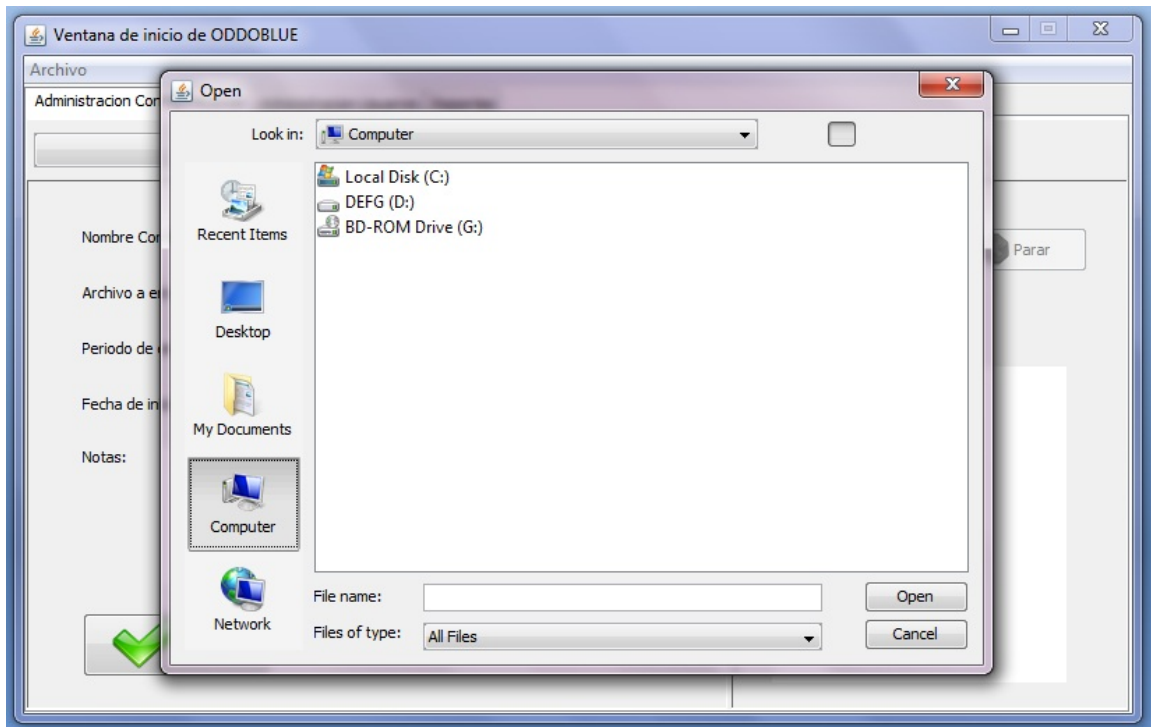
Cuando se crea una campaña nueva, se habilitan al instante los siguientes campos:

- Nombre Configuración.
- Archivo a enviar.
- Periodo de emisión.
- Fecha de la campaña.
- Notas.
- Guardar.

Archivo a enviar: Para poder enviar un archivo por Bluetooth para la nueva campaña publicitaria, lo primero que se requiere es hacer la selección del mismo

dentro de la máquina donde está alojado el programa. Para ello se despliega un navegador que permite ubicar el archivo que se enviará, que resulta ser cómodo al utilizarse.

Figura 23. Selección de Archivo a Enviar



Una vez seleccionado el archivo, el programa reconoce el formato del archivo y lo identifica dependiendo si es un archivo de: audio, video, imagen u otro tipo, se muestra una imagen característica de este tipo de archivo, con base en el formato elegido.

Esto ayuda al usuario con sólo mirar el programa, a identificar qué tipo de archivo está cargado y si es el elegido por el usuario a enviar en ese momento.

Cada uno de los formatos que se pueden visualizar, dependiendo de la selección del archivo que el usuario enviará, serán los que a continuación se presentan en cada una de las figuras.

Figura 24. Formato Seleccionado de Imagen

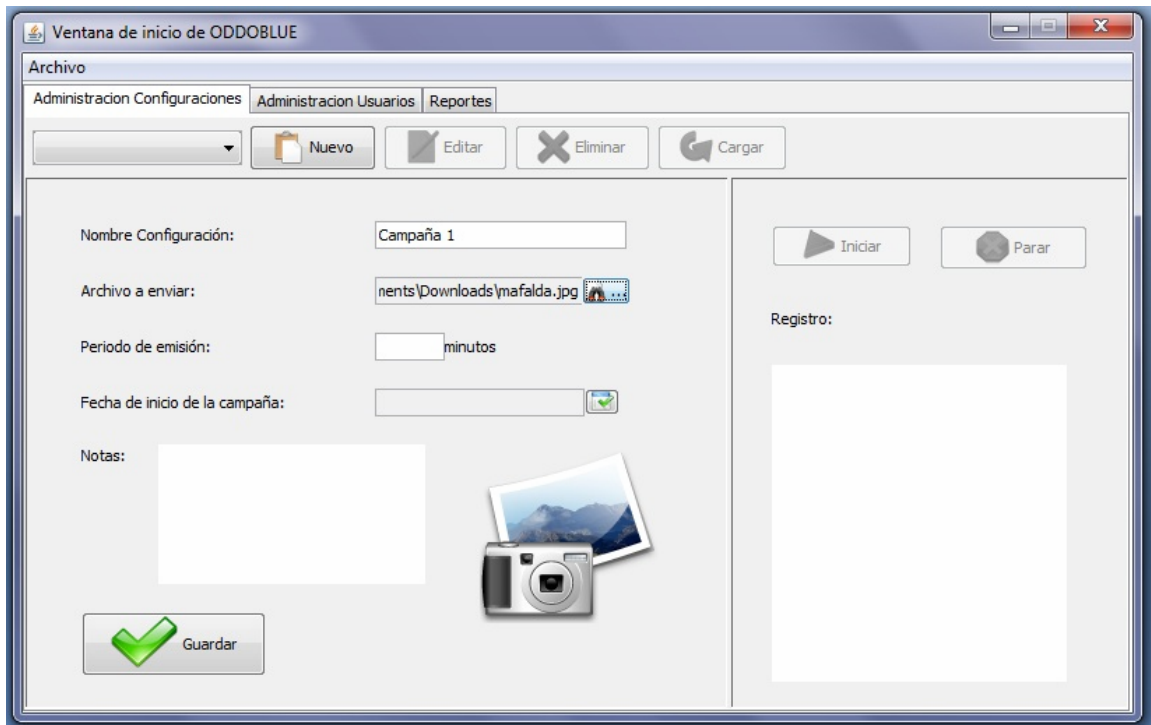


Figura 25. Formato Seleccionado de Audio

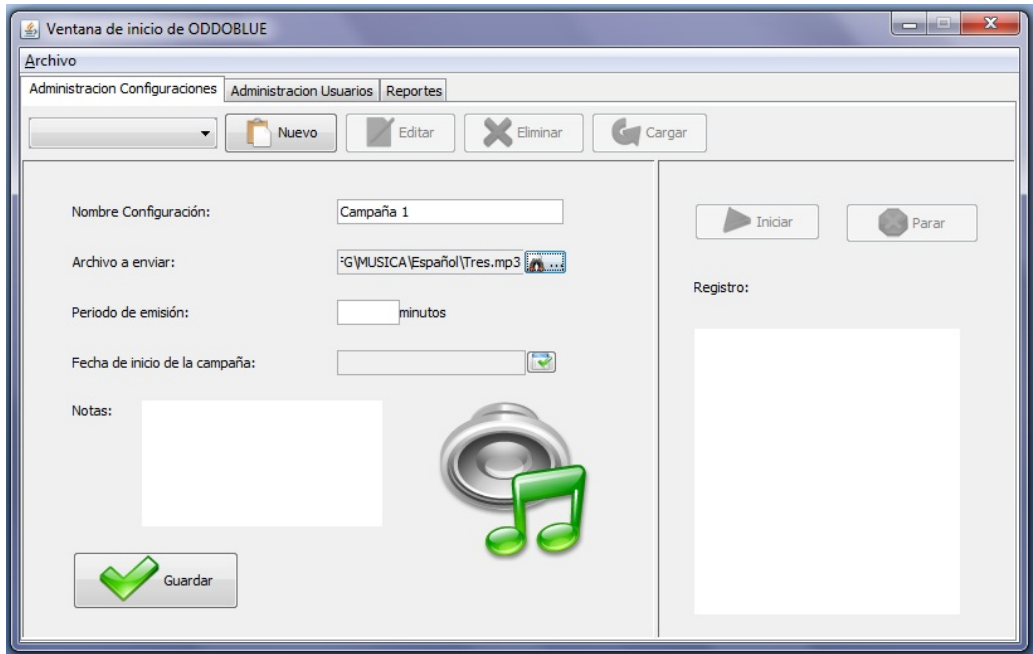


Figura 26. Formato Seleccionado de Juego

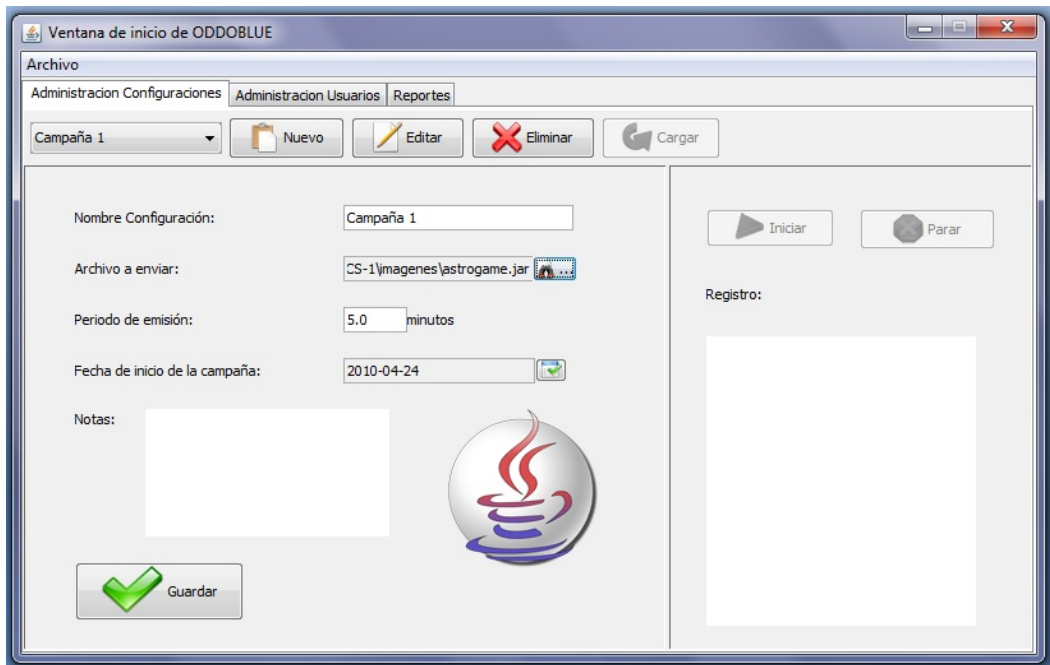


Figura 27. Formato Seleccionado de Video

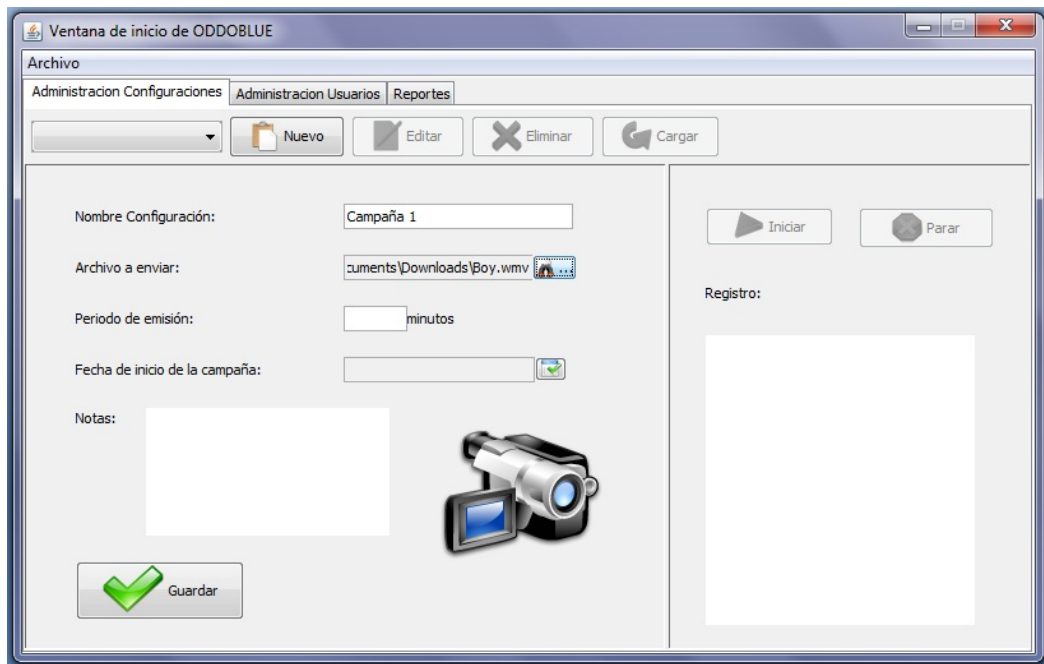
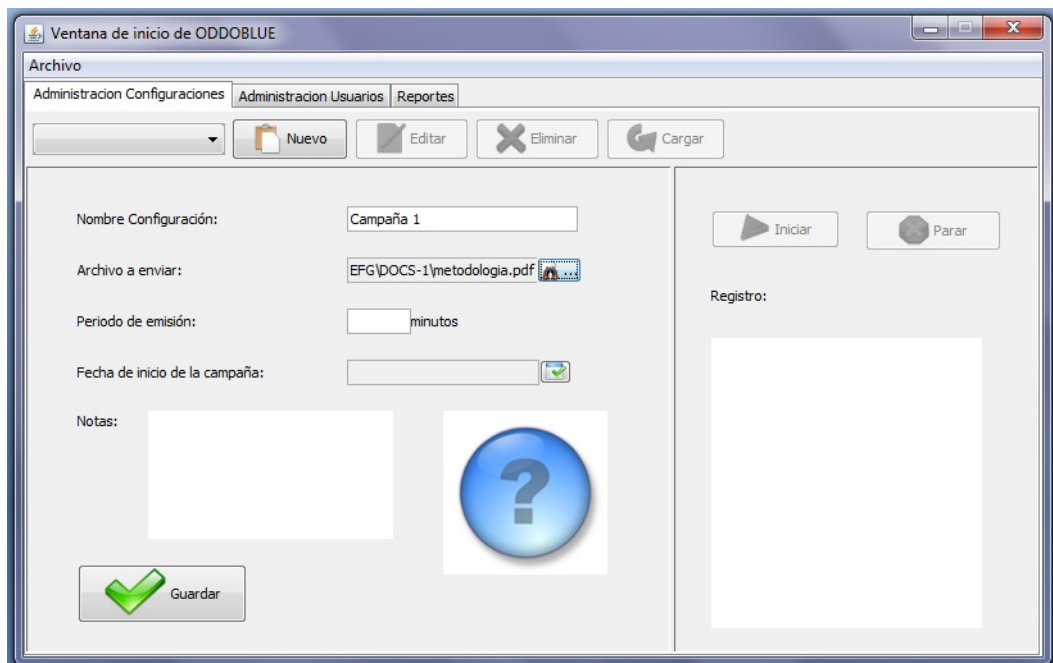


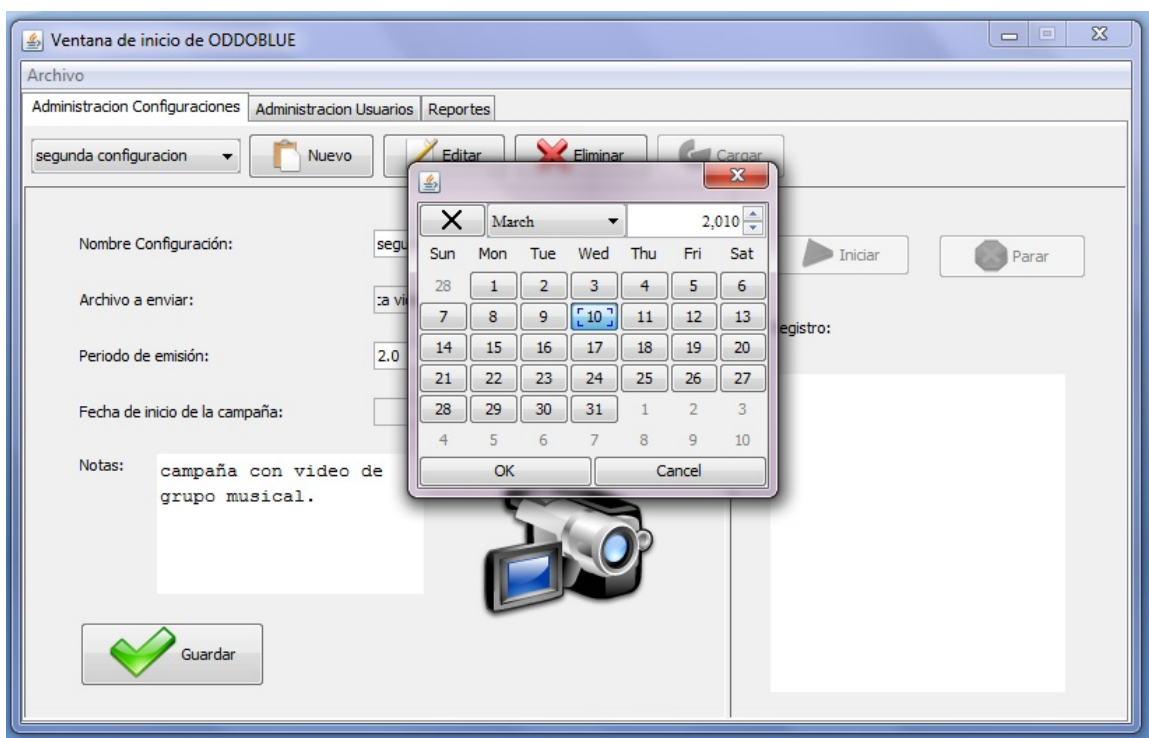
Figura 28. Otro Tipo de Formato Seleccionado



Periodo de emisión: Se debe entender como Periodo de emisión la periodicidad con que el sistema realiza la búsqueda de dispositivos cercanos para entablar una conexión y todo el proceso que implica la transferencia de la información que contiene una campaña.

Fecha de inicio de la campaña: Esta fecha debe utilizarse a partir de la fecha “hoy” o una posterior. Se utiliza para no permitir el inicio de la campaña antes de la fecha programada, esto evita los errores de usuarios que utilizan campañas publicitarias antes de que estén completamente adecuadas al mercado.

Figura 29. Selección de Fecha para Inicio de la Campaña

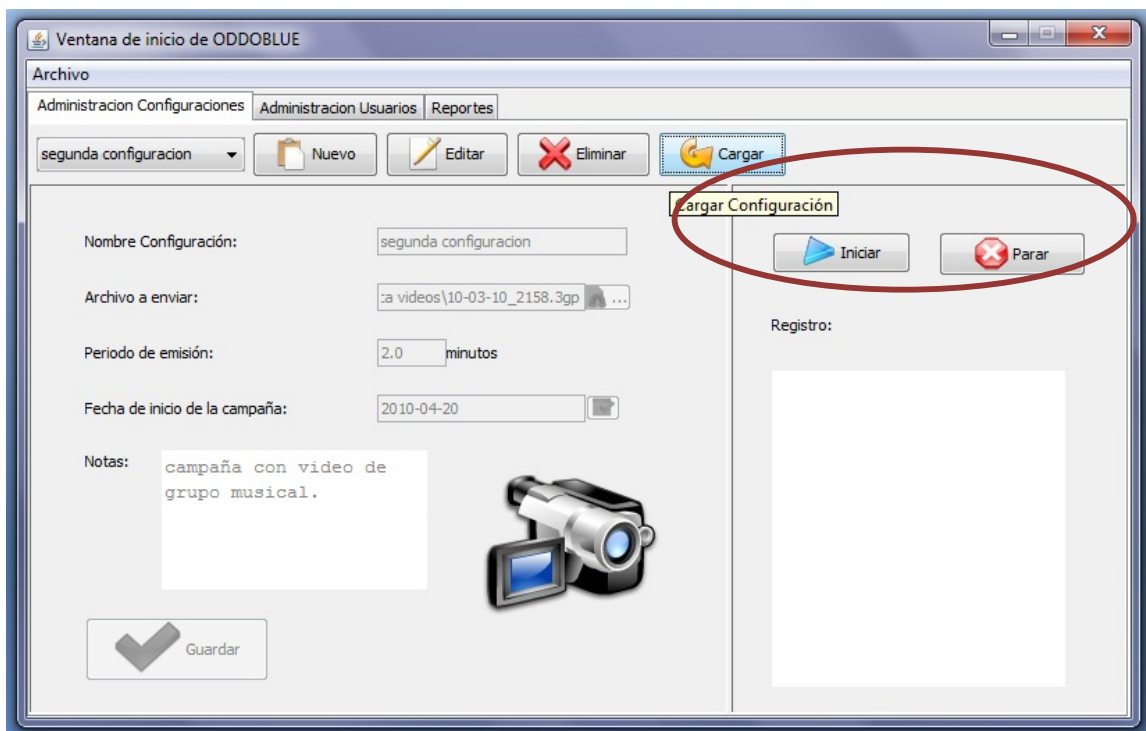


Cuando todos los campos que tiene el prototipo han sido llenados y se ha predeterminado el archivo a enviar, se guarda la configuración en la base de datos, esto se hace con el botón Guardar, una vez se ha guardado la campaña,

podrá verse en el listado de campañas que pueden ser cargadas automáticamente para su uso.

Carga y ejecución del prototipo: Una vez seleccionada la campaña a enviar, debe ser cargada al sistema. Este procedimiento se realiza para tomar las configuraciones que le fueron efectuadas a la campaña, se hace necesaria una validación de los datos que ingresa el usuario en la configuración de la campaña, por ejemplo la “fecha de inicio de campaña” la cual debe ser una fecha igual al día “hoy” o una fecha anterior. Una vez efectuada la validación, se activan automáticamente los botones de Iniciar y Parar.

Figura 30. Carga y Ejecución del Prototipo

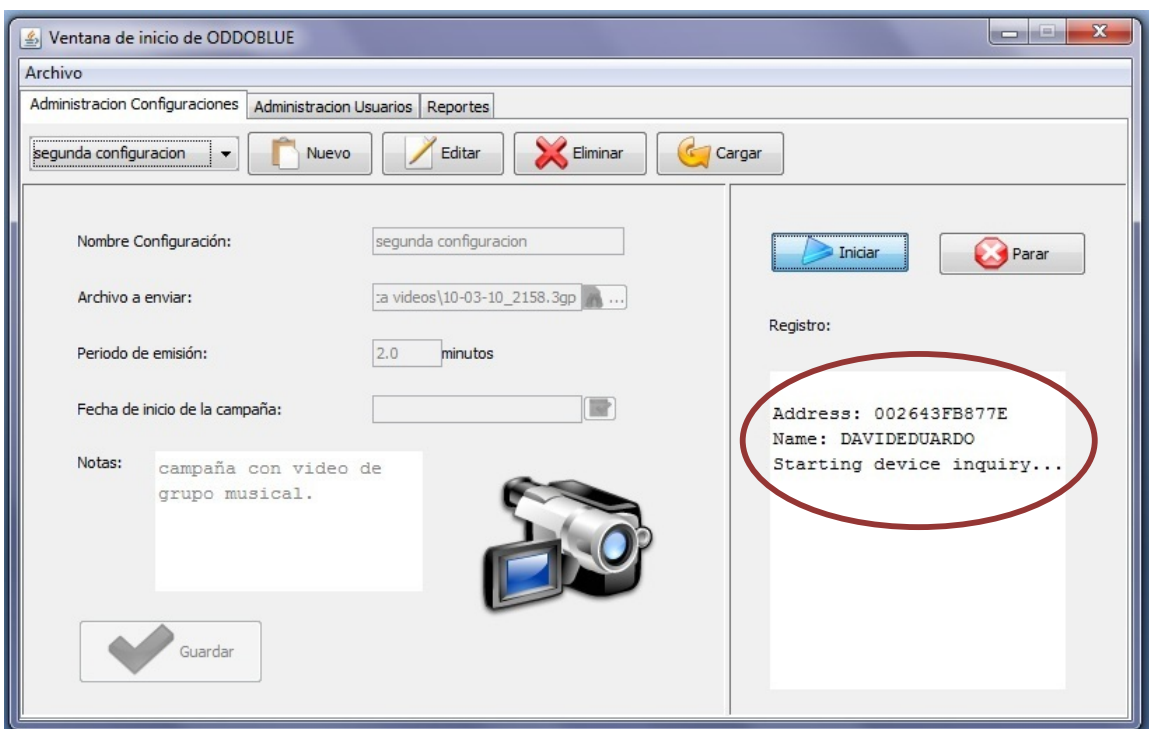


Al dar inicio a la campaña (oprimiendo el botón Iniciar), el prototipo activa el Bluetooth local e inmediatamente comienza con la búsqueda de dispositivos

cercanos para entablar una conexión, y así poder hacer la petición de transferir la información, es decir, comienza la ejecución del prototipo.

La primera acción ejecutada es el inquiry o búsqueda de dispositivos que estén al alcance del dispositivo, esta búsqueda la seguirá efectuando periódicamente con el tiempo determinado para el “periodo de emisión”.

Figura 31. Inicio del Envío de la Campaña



Una vez un dispositivo es encontrado se inicia el proceso del envío del archivo.

5.2.3 Reportes.

Una vez terminada la emisión de una campaña, se hace necesario saber la aceptación que ésta ha obtenido. Para ello se ha creado la Pestaña Reportes, a

través de la cual se puede inspeccionar campaña a campaña la cantidad de envíos exitosos realizados por el sistema.

La pestaña contiene tres componentes, que son:

- **Configuración:** Se elige la campaña para evaluar su desempeño.
- **Fecha inicial - Fecha final:** Indica el periodo en el que se desea saber el comportamiento de la campaña.
- **Mostrar:** Este botón despliega un frame con la representación gráfica en forma de Torta, mostrando los datos que se pidieron anteriormente. Dicho gráfico está dividido en dos secciones: datos aceptados y datos rechazados.

Para conocer los resultados se debe seleccionar la campaña de la cual se requiera la información, hecho esto se activan los campos de “Fecha inicial” y “Fecha Final”, estos dos componentes son importantes para el funcionamiento de este reporte, ya que la información es almacenada día a día y el ubicar fechas precisas permite al usuario determinar cuáles campañas han obtenido mayor aceptación.

Figura 32. Reporte Estadístico

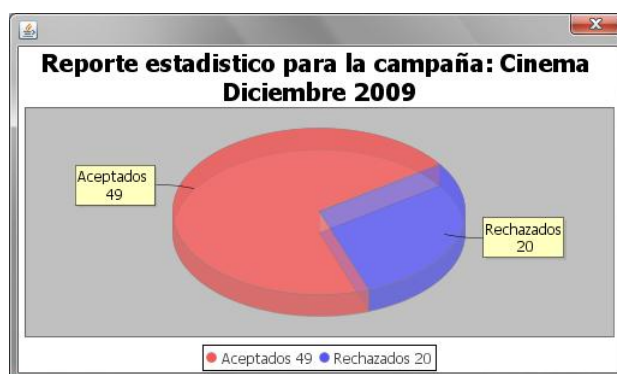
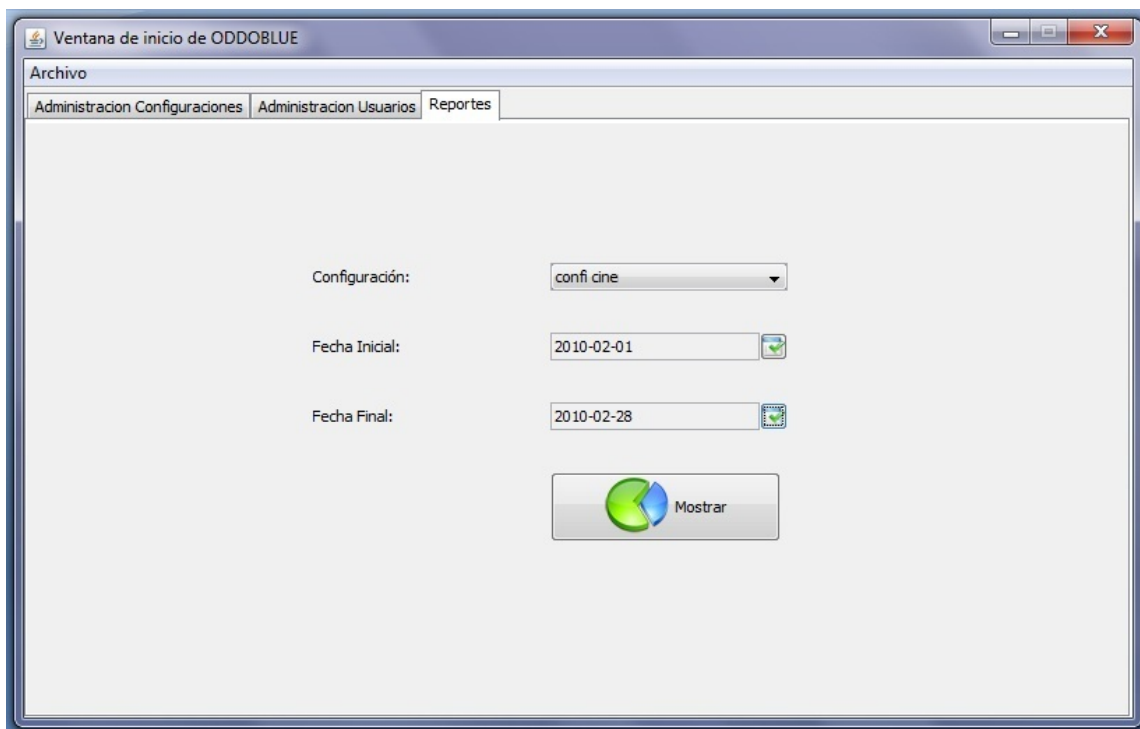


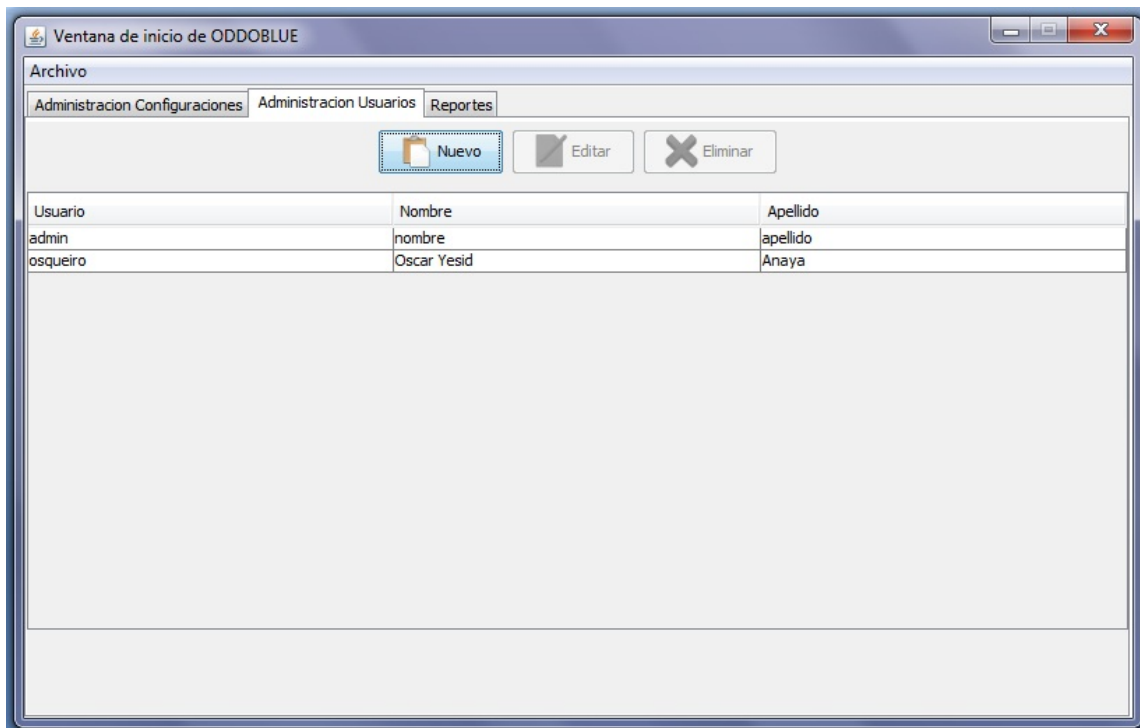
Figura 33. Reportes



Los reportes obtenidos como el mostrado en el ejemplo de la figura 9, se generan automáticamente por el sistema gracias a la librería *jfreechart*, en ella se ve claramente el efecto de la campaña, mostrando en dos colores diferentes la cantidad de peticiones aceptadas por los usuarios así como las rechazadas.

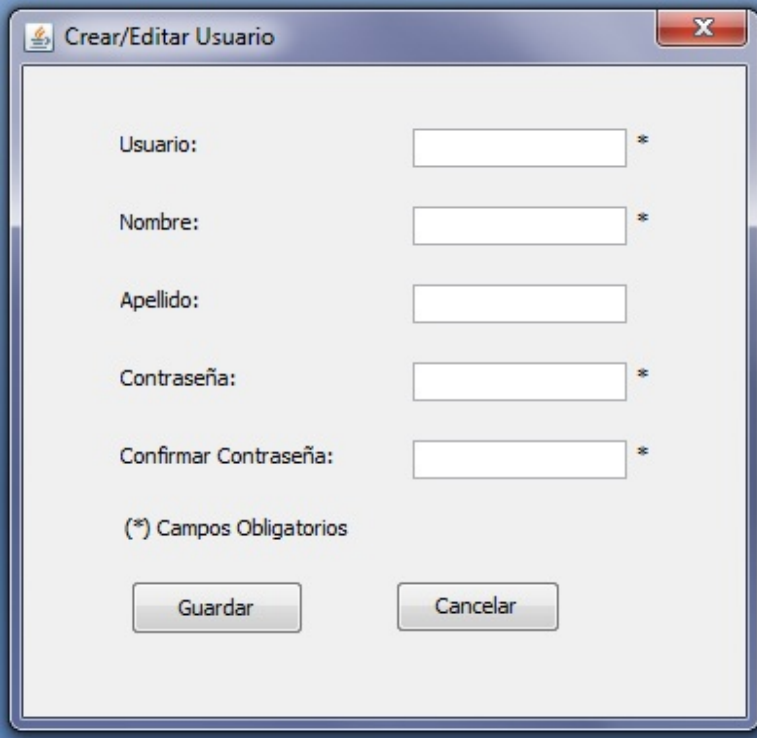
5.2.4 Usuarios

Figura 34. Administración de Usuarios



Los usuarios que estén registrados serán los únicos que puedan acceder al sistema y utilizar todos sus recursos. Para crear una nueva cuenta solamente se da clic en nuevo, donde se despliega un formulario sencillo el cual debe completarse y cuya información será almacenada en la base de datos que posee el sistema.

Figura 35. Creación/Edición de Usuarios



Crear/Editar Usuario

Usuario: *

Nombre: *

Apellido:

Contraseña: *

Confirmar Contraseña: *

(*) Campos Obligatorios

Guardar Cancelar

Los campos que están indicados con un asterisco (*) son campos obligatorios ya que esta información es indispensable

Se ha diseñado un protocolo de seguridad sencillo para los campos de las contraseñas, el campo *Confirmar Contraseña* se pinta de un color diferente hasta cuando coincida con la contraseña dada por el usuario en el campo *Contraseña*, éste último paso ayuda al usuario a completar su configuración de manera adecuada.

Figura 36. Confirmación de la Contraseña de Usuario

Crear/Editar Usuario

Usuario: David *

Nombre: David *

Apellido: Flórez

Contraseña: [redacted] *

Confirmar Contraseña: [redacted] *

(* Campos Obligatorios)

Guardar Cancelar

Una vez son llenados todos los campos correctamente, se guarda y se muestra en el listado de usuarios como se logra ver en la siguiente figura.

Figura 37. Administración de Usuarios - Listado

Ventana de inicio de ODDOBLUE

Archivo

Administración Configuraciones Administración Usuarios Reportes

Nuevo Editar Eliminar

Usuario	Nombre	Apellido
admin	nombre	apellido
osqueiro	Oscar Yesid	Anaya
David	David	Flórez

5.3 EVALUACIÓN

Durante el desarrollo del prototipo se realizaron pruebas con la intención de buscar obtener siempre la funcionalidad del sistema. Su evaluación se llevó a cabo a través del desarrollo mismo. Fueron realizadas pruebas valorando su desempeño y estando atentos a su rendimiento.

Desde el inicio del proyecto se hizo énfasis sobre la operatividad del mismo, a través de la implementación de las librerías necesarias para su buen funcionamiento y, buscando siempre la mejor manera de efectuar los procedimientos y llamados realizados por el sistema.

La transmisión de datos de un dispositivo a otro cercano, por medio Bluetooth utilizando el prototipo, se efectuó de manera exitosa.

CONCLUSIONES

En el entorno nacional se logra percibir que no se han generado herramientas publicitarias enfocadas a explotar las tecnologías que traen consigo los dispositivos móviles, más específicamente y refiriéndose al ámbito que enmarca éste trabajo de grado, la tecnología Bluetooth, aspecto que refleja la innovación y la iniciativa de generar desarrollos de software como el realizado.

El desarrollo del sistema software de marketing de proximidad, permite tanto a las organizaciones como a las personas en general, avanzar hacia la nueva era de la información, a través de la generación de aplicaciones orientadas a facilitar la transferencia de datos, tal y como lo demostró la Tecnología Bluetooth en el prototipo generado, afianzando de esta forma la importancia que representa la Tecnología de la Información y Comunicación – TIC, en la búsqueda de innovación, calidad y evolución de las comunicaciones.

El prototipo del sistema software de marketing de proximidad desarrollado, facilita la transferencia de información entre dispositivos móviles que posean la tecnología Bluetooth, permitiendo a los usuarios interactuar con una interfaz amigable, que presenta:

- Distintas configuraciones de campañas publicitarias, en donde los usuarios pueden acceder a distintas opciones que le permitirán crear y modificar las campañas, de acuerdo a sus necesidades.
- Administración de Usuarios.
- Relación de los reportes estadísticos de cada campaña.

La conexión Bluetooth que realiza el sistema con dispositivos cercanos, se efectúa de manera rápida y constante. Además de cumplir con los horarios establecidos por el usuario para la puesta en marcha la transferencia de información a través del dispositivo Bluetooth, el sistema realiza la búsqueda de dispositivos para posteriormente enviar de información de manera automática.

Durante el desarrollo del trabajo de grado, se logró complementar la formación adquirida durante la carrera de pregrado, a través de la adquisición de las siguientes competencias:

- Comprensión de los aspectos que conforman la tecnología Bluetooth, tales como son; las librerías, los protocolos y los perfiles.
- Entendimiento y utilización del lenguaje de programación JAVA, en donde se profundizó en; las librerías, clases e interfaces que implementa JAVA para el manejo de los protocolos que conforman la tecnología Bluetooth.
- Manejo de una base de datos embebida, ligera y portable, tal y como lo es Apache Derby.
- Utilización de la plataforma IDE Eclipse para desarrollar el prototipo.

RECOMENDACIONES

El prototipo desarrollado podrá ser complementado mediante la ampliación del rango de funcionalidades que hacen parte de la transferencia de archivos, procurando de esta forma lograr una mayor robustez del software, y por consiguiente llegar a convertirlo en una herramienta orientada a satisfacer un amplio número necesidades de los diferentes tipos de usuarios.

Permitir al usuario receptor solicitar información complementaria a aquella que ha recibido en primera instancia, logrando así establecer un intercambio de datos de manera bidireccional.

Adaptar la aplicación desarrollada a nuevas plataformas tales como: la Web y/o aquellas que soporten sistemas operativos móviles (Windows Mobile, Symbian, Android, Mac OS, Palm OS, entre otros).

Permitir la identificación de los dispositivos Bluetooth que estén conectados al equipo desde el que se envían las campañas, de tal forma que puedan utilizarse cada uno de manera individual para enviar una campaña distinta a través de ellos, logrando de esta forma aprovechar en un mejor nivel las prestaciones del software.

Con respecto a trabajos que puedan desarrollarse en un futuro y, que estén basados en este proyecto, es necesario dar a conocer la adaptabilidad para las versiones 3.0 y 4.0 de Bluetooth, las cuales tuvieron su lanzamiento de la última versión (*realese Version*) en Noviembre de 2009.

REFERENCIAS

[1] Bluetooth.

Disponible en: http://es.wikipedia.org/wiki/Bluetooth#Bluetooth_contra_Wi-Fi.

[Fecha consulta: Noviembre 2009]

[2] SIG INTRODUCES BLUETOOTH LOW ENERGY WIRELESS TECHNOLOGY, THE NEXT GENERATION OF BLUETOOTH WIRELESS TECHNOLOGY.

Disponible en: www.bluetooth.com. [Fecha consulta: Enero 2010]

[3] Seguridad Mobile.

Disponible en: <http://www.seguridadmobile.com/bluetooth/especificacion-bluetooth/arquitectura-de-protocolo/index.html>. [Fecha consulta: Octubre 2009]

[4] Bluetooth.

Disponible en: www.bluetooth.com. [Fecha consulta: Octubre 2009]

[5] Definición de Marketing.

Disponible en: www.marketing-free.com. [Fecha consulta: Octubre 2009]

[6] El Marketing de proximidad.

Disponible en: <http://www.eltiempo.com/blogs/tecnomovil/2009/11/el-marketing-de-proximidad.php>. [Fecha consulta: Noviembre 2009]

[7] JfreeCarth Samples.

Disponible en: <http://www.jfree.org/jfreechart/samples.html>. [Fecha consulta: Diciembre 2009]

[8] Eclipse Forms Independent Organization.

Disponible en: <http://www.eclipse.org/org/press-release/feb2004foundationpr.php>.

[Fecha consulta: Diciembre 2009]