

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA
DE CARTAGENA



Proyecto Fin de Carrera

**SUMI (Sistema de Seguimiento de
Usuarios y Monitorización de
Itinerarios)**



AUTOR: Pedro Juan García Sánchez
DIRECTOR: José Fernando Cerdán Cartagena



Autor	Pedro Juan García Sánchez
E-mail del Autor	pedro.juan.garcia@gmail.com
Director	José Fernando Cerdán Cartagena
E-mail del Director	Fernando.cerdan@upct.es
Codirector	Diego García Sánchez
Título del PFC	SUMI (Sistema de Seguimiento de Usuarios y Monitorización de Itinerarios)
<p>Resumen</p> <p>Este proyecto se basa en un servidor de monitorización de dispositivos electrónicos que recibe la información recogida por parte de las estaciones cliente. El objetivo de este servidor es procesar dicha información, generar estadísticas y presentarla de forma sencilla para su análisis.</p> <p>Para ello se ha creado un servidor que recibe archivos de los distintos clientes y mediante una aplicación en JAVA se analizan y ordenan para insertarlos en una BBDD MySQL y presentar en una interfaz los resultados obtenidos.</p> <p>Con este sistema se obtiene la posibilidad de obtener multitud de información acerca de cómo los usuarios se comportan y de ello obtener estadísticas tales como el número de personas que acuden a un estadio, a un establecimiento, a qué hora hay más o menos tránsito, el tiempo que permanecen subidos en un transporte público, etc.</p>	
Titulación	Ingeniería Técnica de Telecomunicación especialidad Telemática
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Julio - 2013

Índice

CAPITULO I: INTRODUCCIÓN.....	8
1.1. Objetivos	8
1.2. Descripción	8
1.3. Arquitectura del sistema	9
CAPITULO II: TECNOLOGÍAS UTILIZADAS	10
2.1. Introducción	10
2.2. Java	11
2.3. HSDPA, 3G Y WIFI.....	12
2.4. FTP.....	17
2.4. MYSQL	20
2.5. Excel	23
2.6. Interfaz de seguimiento GPS	24
2.7. Conclusión	25
CAPITULO III: CARACTERISTICAS DEL SISTEMA.....	26
3.1. Introducción	26
3.2. Entorno de desarrollo.....	27
3.2. Funcionamiento	30
3.3. Aplicación JAVA.....	34
3.4. Conclusión	42
CAPITULO IV: CASOS PRÁCTICOS	43
4.1. Estadio Cartagonova	43
4.2. Línea de tren	44
4.3. Worten y Mediamarkt.....	45
4.4. Coche de seguridad.....	46
CAPITULO V: CONCLUSION Y LINEAS DE FUTURO	47
5.1. Conclusión	47
5.2. Líneas de futuro	48
CAPITULO VI: MANUAL DE INSTALACIÓN	49
CAPITULO VII: BIBLIOGRAFIA	51

INDICE DE FIGURAS

FIGURA 1: Modelo FTP con un servidor.....	16
FIGURA 2: Modelo FTP con dos servidor.....	18
FIGURA 3: Ejemplo de Google Earth.....	23
FIGURA 4: Esquema del proyecto.....	28
FIGURA 5: Funcionamiento de la aplicación.....	31
FIGURA 6: Import de funciones SQL en Java.....	33
FIGURA 7: Swing de la aplicación.....	33
FIGURA 8: Clase FTP.....	34
FIGURA 9: Archivo recibido en bruto.....	35
FIGURA 10: Separador de variables con Stringtokenizer.....	35
FIGURA 11: Inserción de cadena a BBDD.....	36
FIGURA 12: Conectar a BBDD con Netbeans.....	36
FIGURA 13: Conexión en Java con MySQL.....	33
FIGURA 14: Secuencia de la cadena.....	33
FIGURA 15: Salida por pantalla del programa.	38
FIGURA 16: Tabla en phpmyadmin.....	40
FIGURA 17: Exportar tabla a Excel.....	40
FIGURA 18: Panel de control de Xampp.....	49
FIGURA 19: Login en base de datos.....	50

CAPITULO I: INTRODUCCIÓN

1.1.Objetivos

El objetivo principal de este proyecto es la implementación de un servidor de monitorización de dispositivos electrónicos que a través de varias etapas muestre informes y estadísticas por una interfaz. Todo ello a partir de las detecciones obtenidas en distintos ámbitos por las estaciones clientes.

Este sistema consta de estaciones clientes las cuales son balizas detectoras que recogen la información, la posiciona mediante GPS y las envía al servidor, el cual tiene el objetivo de ordenar, guardar y mostrar todo lo recibido por los distintos clientes.

El servidor recibe las tramas generadas en ficheros de texto enviados periódicamente por el cliente y descargados de un servidor FTP, estos archivos son los que la aplicación se encarga de procesar y almacenar en la BBDD para su posterior presentación a través de una interfaz o mostrar informes estadísticos como por ejemplo el tiempo que una persona permanece en un establecimiento o las rutas más frecuentadas en transporte público.

Obtener información anónima, procesarla y presentarla mediante una interfaz gráfica es el objetivo principal de este proyecto. Además de la utilización para estudio de comportamiento, este sistema posee una multitud de usos de monitorización de personas tales como la medición aproximada de aforo a estadios, comercios o control de empleados.

1.2.Descripción

Se ha implementado un servidor sobre *Windows* donde se reciben del cliente los datos desde internet (HSDPA, 3g, Wifi), los cuales son archivos con líneas de texto. Éstos llegan al servidor por FTP, mediante una aplicación creada en Java se lleva a cabo el procesado de los campos de cada archivo para introducirlos en una tabla SQL.

Desde la BBDD donde están las tablas se realizan todo tipo de consultas según el estudio estadístico que se desee realizar y se exportan hacia una hoja de cálculo la cual actúa de interfaz para mostrar a partir de tablas dinámicas e informes los resultados obtenidos.

Hay diferentes tipos de estudio estadístico y uno de los realizados en este proyecto consiste en probar el tiempo óptimo de detección de las balizas a través de distintos test aumentando y disminuyendo este tiempo de detección y así obtener resultados más fiables.

1.3.Arquitectura del sistema

El sistema SUMI está compuesto por un servidor de monitorización y una estación de detección instalada en el lugar donde se desea realizar el estudio.

Estaciones de detección: *Baliza detectora*

Servidor de monitorización: *PC Windows (aplicación en Java)*

- Servidor FTP.
- Servidor con BBDD MySQL.
- Aplicación Java para procesado y almacenamiento.
- Gestor de BD con interfaz gráfica.

En este proyecto se desarrolla el servidor de monitorización, el cual es un ordenador con conexión a internet que se encuentra permanentemente encendido. Éste accede periódicamente a los archivos enviados por el cliente a través del servidor FTP, guarda localmente en carpetas y procesa esos archivos de texto en la aplicación para después presentarlos en una interfaz sencilla donde se pueden hacer consultas, analizar los resultados y tener una visión más útil de la información.

CAPITULO II: TECNOLOGÍAS UTILIZADAS

2.1.Introducción

A continuación se detallan las distintas tecnologías utilizadas en este proyecto, las cuales son:

- Java: Se ha elegido este lenguaje de programación para esta aplicación por su rapidez, versatilidad y eficiencia, principalmente para este proyecto la facilidad y seguridad en la conectividad con bases de datos MySQL mediante los conectores jdbc. Con Java es muy sencillo probar los errores y ofrece mucha estabilidad.
- FTP: Protocolo elegido para la transmisión de los archivos que se van descargando desde el cliente FTP por su velocidad y por la independencia del sistema operativo de cada equipo, que en este caso la otra parte del proyecto trabaja sobre Linux y este servidor trabaja sobre Windows.
- HSDPA, 3G, Wifi: El modo por el cual esta aplicación tiene el acceso a internet para la descarga de los archivos se puede realizar a través de cualquiera de estos modos.
- MySQL: Tipo de BBDD escogida para el procesado de los datos.
- Excel: Hoja de cálculo la cual mediante tablas dinámicas implementadas se introducen los campos y se obtienen estadísticas y gráficos para analizar los datos más fácilmente.

2.2. Java

Java Database Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la BBDD a la cual se accede, utilizando el dialecto SQL del modelo de BBDD que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de BBDD. Un manejador de conexiones hacia un modelo de BBDD en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a BBDD (URL) que pueden manejar. Para utilizar una BBDD particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su BBDD, y accede a ella estableciendo una conexión; para ello provee el localizador a la BBDD y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tarea con la BBDD a la que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la BBDD, etc.

2.3. HSDPA, 3G Y WIFI

2.3.1. HSDPA

La tecnología HSDPA (High Speed Downlink Packet Access), también denominada 3.5G, 3G+ o mini 3G, es la optimización de la tecnología espectral UMTS/WCDMA, una tecnología basada en conexiones minis, de menor velocidad a el promedio de la actual 3G, incluida en las especificaciones de 3GPP Release 5 y consiste en un nuevo canal compartido en el enlace descendente (downlink) que mejora significativamente la capacidad máxima de transferencia de información pudiéndose alcanzar tasas de bajada de hasta 14 Mbps (1,8, 3,6, 7,2 y 14,4 Mbps) Aunque sin mejorar el 3G. Soporta tasas de throughput promedio cercanas a 1 Mbps.

Actualmente, también está disponible la tecnología HSUPA, con velocidades de subida de hasta 5,8 Mbps, y HSPA+ con velocidades de hasta 84 Mbps de bajada y 22 Mbps en la subida.

Es la evolución de la tercera generación (3G) de tecnología móvil, llamada 3.5G, y se considera el paso previo antes de la cuarta generación (4G), la futura integración de redes. Actualmente se está desarrollando la especificación 3.9G antes del lanzamiento de 4G.

Es totalmente compatible en sentido inverso con WCDMA y aplicaciones ricas en multimedia desarrolladas para WCDMA que funcionarán con HSDPA. La mayoría de los proveedores UMTS dan soporte a HSDPA.

HSDPA lleva a las redes WCDMA a su máximo potencial en la prestación de servicios de banda ancha, mediante un aumento en la capacidad de datos celulares, con throughput más elevado.

De la misma manera en que UMTS incrementa la eficiencia espectral en comparación con GPRS, HSDPA incrementa la eficiencia espectral en comparación con WCDMA. La eficiencia espectral y las velocidades aumentadas no sólo habilitan nuevas clases de aplicaciones, sino que además permite que la red sea utilizada simultáneamente por un número mayor de usuarios; HSDPA provee de tres a cuatro veces más capacidad que WCDMA.

En cuanto a la interfaz de las aplicaciones en tiempo real tales como videoconferencia y juegos entre múltiples jugadores, actualiza a la tecnología WCDMA al acortar la latencia de la red (se prevén menos de 100 ms), brindando así mejores tiempos de respuesta.

Alcanza sus elevadas tasas de velocidad gracias al agregado de modulación de mayor orden (Modulación de Amplitud en Cuadratura 16 – 16 QAM), codificación variable de

errores y redundancia incremental, así como la introducción de nuevas y potentes técnicas tales como programación rápida.

WD Además, HSDPA emplea un eficiente mecanismo de programación para determinar qué usuario obtendrá recursos. Están programadas varias optimizaciones para HSDPA que aumentarán aún más las capacidades de UMTS/HSDPA, comenzando con un enlace ascendente optimizado (HSUPA), receptores avanzados y antenas inteligentes/MIMO.

Finalmente, comparte sus canales de alta velocidad entre los usuarios del mismo dominio de tiempo, lo que representa el enfoque más eficiente.

2.3.2. 3G

3G es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles).

Los servicios asociados con la tercera generación proporcionan la posibilidad de transferir tanto voz y datos (una llamada telefónica o una videollamada) y datos no-voz (como la descarga de programas, intercambio de correos electrónicos, y mensajería instantánea).

Aunque esta tecnología estaba orientada a la telefonía móvil, desde hace unos años las operadoras de telefonía móvil ofrecen servicios exclusivos de conexión a Internet mediante módem USB, sin necesidad de adquirir un teléfono móvil, por lo que cualquier computadora puede disponer de acceso a Internet.

Existen otros dispositivos como algunos ultraportátiles (netbooks) que incorporan el módem integrado en el propio equipo, pero requieren de una tarjeta SIM (la que llevan los teléfonos móviles) para su uso, por lo que en este caso sí es necesario estar dado de alta con un número de teléfono.

Las redes 3G ofrecen mayor grado de seguridad en comparación con sus predecesoras 2G. Al permitir a la UE autenticar la red a la que se está conectando, el usuario puede asegurarse de que la red es la intencionada y no una imitación.

En la conferencia BlackHat 2010 un hacker demostró (con un presupuesto de US\$ 1.500) que podía obtener números celulares e incluso escuchar las llamadas de teléfonos GSM cercanos, esto era logrado haciéndose pasar por una base (antena receptora/transmisora) de la telefónica AT&T en este caso.

Las redes 3G usan el cifrado por bloques KASUMI en vez del anterior cifrador de flujo A5/1. Aun así, se han identificado algunas debilidades en el código KASUMI.

Además de la infraestructura de seguridad de las redes 3G, se ofrece seguridad de un extremo al otro cuando se accede a aplicaciones framework como IMS, aunque esto no es algo que sólo se haga en el 3G.

2.3.3. WIFI

Wi-Fi es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi, tales como: un ordenador personal, una consola de videojuegos, un smartphone o un reproductor de audio digital, pueden conectarse a Internet a través de un punto de acceso de red inalámbrica.

Dicho punto de acceso (o hotspot) tiene un alcance de unos 20 metros en interiores y al aire libre una distancia mayor. Pueden cubrir grandes áreas la superposición de múltiples puntos de acceso.

Wi-Fi es una marca de la Wi-Fi Alliance (anteriormente la WECA: Wireless Ethernet Compatibility Alliance), la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

Historia:

Esta nueva tecnología surgió por la necesidad de establecer un mecanismo de conexión inalámbrica que fuese compatible entre los distintos dispositivos. Buscando esa compatibilidad fue que en 1999 las empresas 3com, Airones, Intersil, Lucent Technologies, Nokia y Symbol Technologies se reunieron para crear la Wireless Ethernet Compatibility Alliance, o WECA, actualmente llamada Wi-Fi Alliance.

El objetivo de la misma fue designar una marca que permitiese fomentar más fácilmente la tecnología inalámbrica y asegurar la compatibilidad de equipos.

De esta forma, en abril de 2000 WECA certifica la interoperabilidad de equipos según la norma IEEE 802.11b, bajo la marca Wi-Fi. Esto quiere decir que el usuario tiene la garantía de que todos los equipos que tengan el sello Wi-Fi pueden trabajar juntos sin problemas, independientemente del fabricante de cada uno de ellos. Se puede obtener un listado completo de equipos que tienen la certificación Wi-Fi en Alliance – Certified Products.

En el año 2002 la asociación WECA estaba formada ya por casi 150 miembros en su totalidad. La familia de estándares 802.11 ha ido naturalmente evolucionando desde su creación, mejorando el rango y velocidad de la transferencia de información, entre otras cosas.

La norma IEEE 802.11 fue diseñada para sustituir el equivalente a las capas físicas y MAC de la norma 802.3 (Ethernet). Esto quiere decir que en lo único que se diferencia una red Wi-Fi de una red Ethernet es en cómo se transmiten las tramas o paquetes de datos; el resto es idéntico. Por tanto, una red local inalámbrica 802.11 es completamente compatible con todos los servicios de las redes locales (LAN) de cable 802.3 (Ethernet).

Seguridad:

Uno de los problemas a los cuales se enfrenta actualmente la tecnología Wi-Fi es la progresiva saturación del espectro radioeléctrico, debido a la masificación de usuarios, esto afecta especialmente en las conexiones de larga distancia (mayor de 100 metros).

En realidad Wi-Fi está diseñado para conectar ordenadores a la red a distancias reducidas, cualquier uso de mayor alcance está expuesto a un excesivo riesgo de interferencias.

Un muy elevado porcentaje de redes son instalados sin tener en consideración la seguridad convirtiendo así sus redes en redes abiertas (o completamente vulnerables ante el intento de acceder a ellas por terceras personas), sin proteger la información que por ellas circulan.

De hecho, la configuración por defecto de muchos dispositivos Wi-Fi es muy insegura (routers, por ejemplo) dado que a partir del identificador del dispositivo se puede conocer la clave de éste; y por tanto acceder y controlar el dispositivo se puede conseguir en sólo unos segundos.

El acceso no autorizado a un dispositivo Wi-Fi es muy peligroso para el propietario por varios motivos. El más obvio es que pueden utilizar la conexión. Pero además, accediendo al Wi-Fi se puede monitorizar y registrar toda la información que se transmite a través de él (incluyendo información personal, contraseñas...). La forma de hacerlo seguro es seguir algunos consejos:

- Cambios frecuentes de la contraseña de acceso, utilizando diversos caracteres, minúsculas, mayúsculas y números.
- Se debe modificar el SSID que viene predeterminado.
- Realizar la desactivación del broadcasting SSID y DHCP.
- Configurar los dispositivos conectados con su IP (indicar específicamente qué dispositivos están autorizados para conectarse).
- Utilización de cifrado: WPA2.

Existen varias alternativas para garantizar la seguridad de estas redes. Las más comunes son la utilización de protocolos de cifrado de datos para los estándares Wi-Fi como el WEP, el WPA, o el WPA2 que se encargan de codificar la información transmitida para proteger su confidencialidad, proporcionados por los propios dispositivos inalámbricos. La mayoría de las formas son las siguientes:

- WEP, cifra los datos en su red de forma que sólo el destinatario deseado pueda acceder a ellos. Los cifrados de 64 y 128 bits son dos niveles de seguridad WEP. WEP codifica los datos mediante una “clave” de cifrado antes de enviarlo al aire. Este tipo de cifrado no está muy recomendado debido a las grandes vulnerabilidades que presenta ya que cualquier cracker puede conseguir sacar la clave, incluso aunque esté bien configurado y la clave utilizada sea compleja.
- WPA: presenta mejoras como generación dinámica de la clave de acceso. Las claves se insertan como dígitos alfanuméricos.

- IPSEC (túneles IP) en el caso de las VPN y el conjunto de estándares IEEE 802.1X, que permite la autenticación y autorización de usuarios.
- Filtrado de MAC, de manera que sólo se permite acceso a la red a aquellos dispositivos autorizados. Es lo más recomendable si solo se va a usar con los mismos equipos, y si son pocos.
- Ocultación del punto de acceso: se puede ocultar el punto de acceso (Router) de manera que sea invisible a otros usuarios.
- El protocolo de seguridad llamado WPA2 (estándar 802.11i), que es una mejora relativa a WPA. En principio es el protocolo de seguridad más seguro para Wi-Fi en este momento. Sin embargo requieren hardware y software compatibles, ya que los antiguos no lo son.

2.4. FTP

Introducción al protocolo FTP

El protocolo FTP (*Protocolo de transferencia de archivos*) es, como su nombre lo indica, un protocolo para transferir archivos entre equipos.

La implementación del FTP se remonta a 1971 cuando se desarrolló un sistema de transferencia de archivos (descrito en [RFC 141](#)) entre equipos del Instituto Tecnológico de Massachusetts (*MIT, Massachusetts Institute of Technology*). Desde entonces, diversos documentos de RFC han mejorado el protocolo básico, pero las innovaciones más importantes se llevaron a cabo en julio de 1973.

La función del protocolo FTP

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP. Su objetivo es permitir que equipos remotos puedan compartir archivos y permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor y garantizar una transferencia de datos eficaz.

El modelo FTP

El protocolo FTP está incluido dentro del modelo cliente-servidor, es decir, un equipo envía órdenes (el cliente) y el otro espera solicitudes para llevar a cabo acciones (el servidor).

Durante una conexión FTP, se encuentran abiertos dos canales de transmisión: Un canal de comandos (canal de control) y un canal de datos.

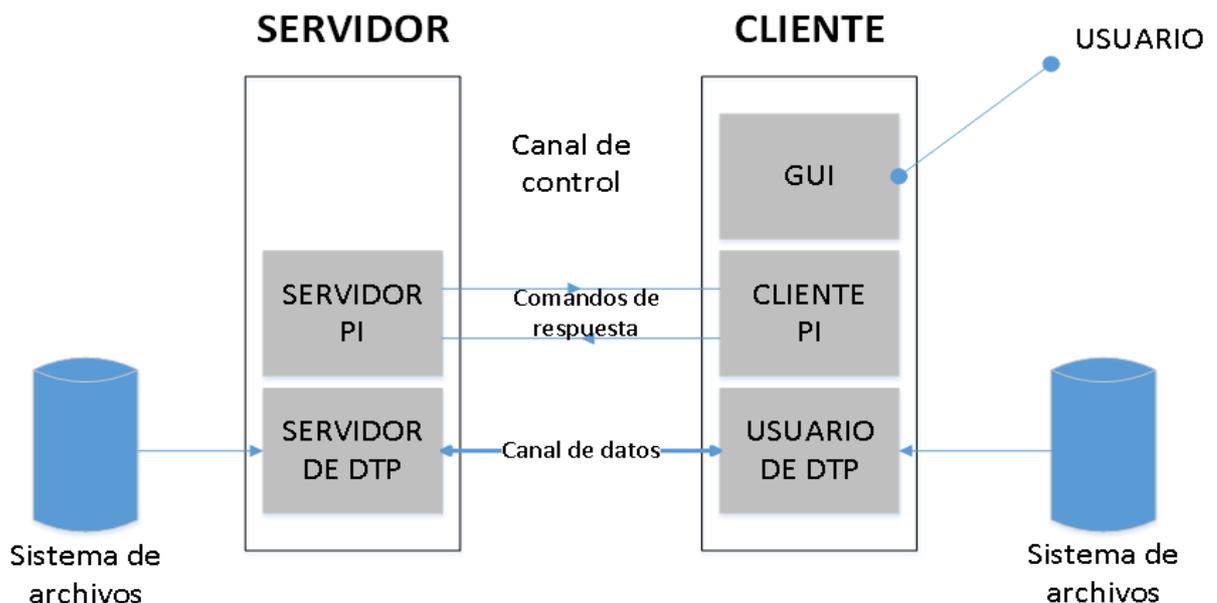


FIGURA 1: Modelo FTP con un servidor

Por lo tanto, el cliente y el servidor cuentan con dos procesos que permiten la administración de estos dos tipos de información:

DTP (*Proceso de transferencia de datos*) es el proceso encargado de establecer la conexión y de administrar el canal de datos. El DTP del lado del servidor se denomina *SERVIDOR DE DTP* y el DTP del lado del cliente se denomina *USUARIO DE DTP*.

PI (*Intérprete de protocolo*) interpreta el protocolo y permite que el DTP pueda ser controlado mediante los comandos recibidos a través del canal de control. Esto es diferente en el cliente y el servidor:

El Servidor PI es responsable de escuchar los comandos que provienen de un Usuario PI a través del canal de control en un puerto de datos, de establecer la conexión para el canal de control, de recibir los comandos FTP del Usuario PI a través de éste, de responderles y de ejecutar el Servidor DTP.

El Usuario PI es responsable de establecer la conexión con el servidor FTP, de enviar los comandos FTP, de recibir respuestas del Servidor PI y de controlar al Usuario de DTP, si fuera necesario.

Cuando un cliente FTP se conecta con un servidor FTP, el Usuario PI inicia la conexión con el servidor de acuerdo con el protocolo Telnet. El cliente envía comandos FTP al servidor, el servidor los interpreta, ejecuta su DTP y después envía una respuesta estándar. Una vez que se establece la conexión, el servidor PI proporciona el puerto por el cual se enviarán los datos al Cliente DTP.

El cliente DTP escucha el puerto especificado para los datos provenientes del servidor. Es importante tener en cuenta que, debido a que los puertos de control y de datos son canales separados, es posible enviar comandos desde un equipo y recibir datos en otro. Entonces, por ejemplo, es posible transferir datos entre dos servidores FTP mediante el paso indirecto por un cliente para enviar instrucciones de control y la transferencia de información entre dos procesos del servidor conectados en el puerto correcto.

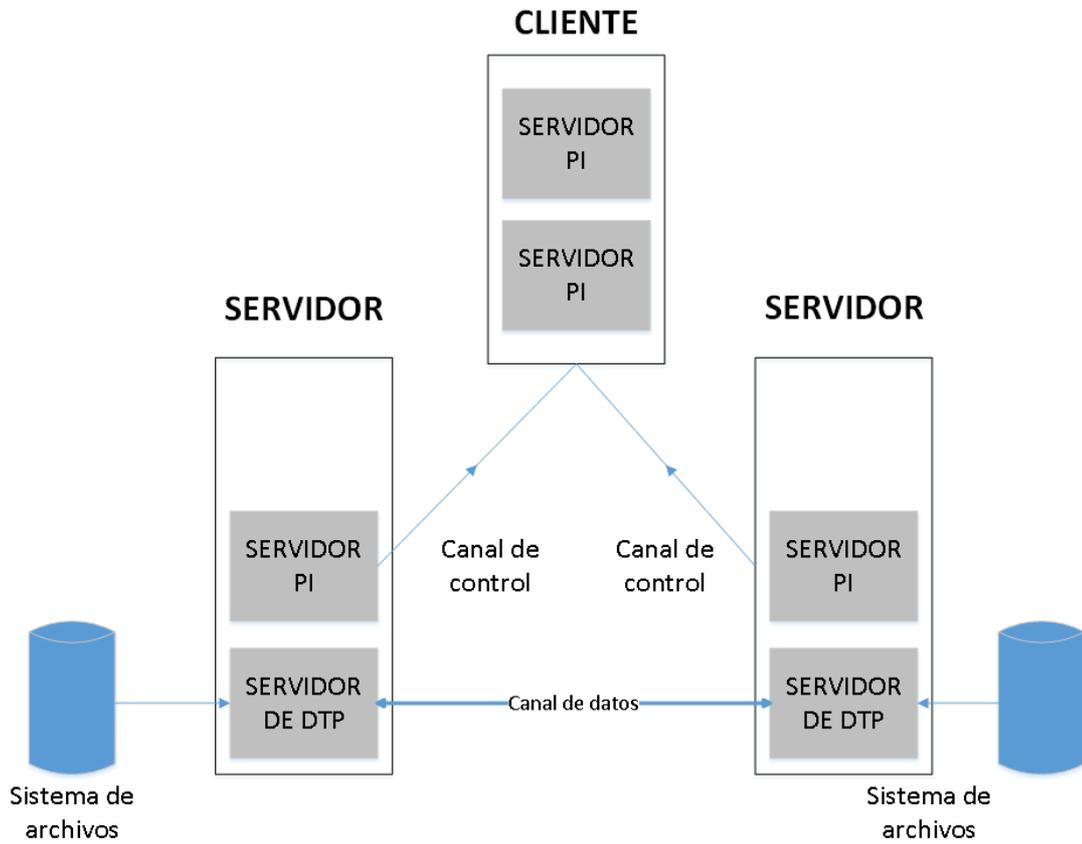


FIGURA 2: Modelo FTP con dos servidor

En esta configuración, el protocolo indica que los canales de control deben permanecer abiertos durante la transferencia de datos. De este modo, un servidor puede detener una transmisión si el canal de control es interrumpido durante la transmisión.

2.4.MYSQL

2.4.1. Introducción

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python Ruby, Gambas, REALbasic (Macy Linux), FreeBASIC, y Tcl; cada uno de estos utiliza una interfaz de programación de aplicaciones específica.

También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.

MySQL es una BBDD muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

2.4.2. Características

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones...

- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

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

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

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

2.4.3. Características adicionales

Usa GNU Automake, Autoconf, y Libtool para portabilidad

Uso de multihilos mediante hilos del kernel.

Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice

Tablas hash en memoria temporales

El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.

Completo soporte para operadores y funciones en cláusulas select y where.

Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación

Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.

Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.

Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.

En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.

MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

2.5. Excel

Microsoft Excel es una aplicación distribuida por Microsoft Office para hojas de cálculo. Este programa es desarrollado y distribuido por Microsoft, y es utilizado normalmente en tareas financieras y contables.

Microsoft comercializó originalmente un programa de hojas de cálculo llamado Multiplan en 1982, que fue muy popular en los sistemas CP/M, pero en los sistemas MS-DOS perdió popularidad frente al Lotus 1-2-3.

Microsoft publicó la primera versión de Excel para Mac en 1985, y la primera versión de Windows (numeradas 2-05 en línea con el Mac y con un paquete de tiempo de ejecución de entorno de Windows) en noviembre de 1987. Lotus fue lenta al llevar 1-2-3 para Windows y esto ayudó a Microsoft a alcanzar la posición de los principales desarrolladores de software para hoja de cálculo de PC, superado al también muy popular Quattro Pro de Borland.

Este logro solidificó a Microsoft como un competidor válido y mostró su futuro de desarrollo como desarrollador de software GUI. Microsoft impulsó su ventaja competitiva lanzando al mercado nuevas versiones de Excel, por lo general cada dos años. La versión actual para la plataforma Windows es Microsoft Excel 2013. La versión actual para Mac OS X es Microsoft Excel 2011.

2.6. Interfaz de seguimiento GPS

Para poder aprovechar la incorporación del dispositivo GPS que incorporan las balizas detectoras de las cuales se recibe la información y de un modo experimental y con el principal objetivo de obtener un mayor atractivo más que darle una nueva utilidad a la aplicación se usó una interfaz de seguimiento GPS llamada Google Earth para la visualización del posicionamiento de las balizas.

Partiendo de los archivos .gpx que proporciona la baliza y en el cual se incorpora información acerca de la posición y añade una marca de tiempo con la cual se puede construir mediante un programa un archivo kmz que son archivos de placemark para Google Earth.

Los múltiples puntos de longitud y latitud junto con el tiempo de cada uno conforman el trazo que dibuja sobre el mapa el recorrido, el cual puede ser visto gracias a la herramienta de Google que superpone sobre el mapa dichos puntos en el tiempo.

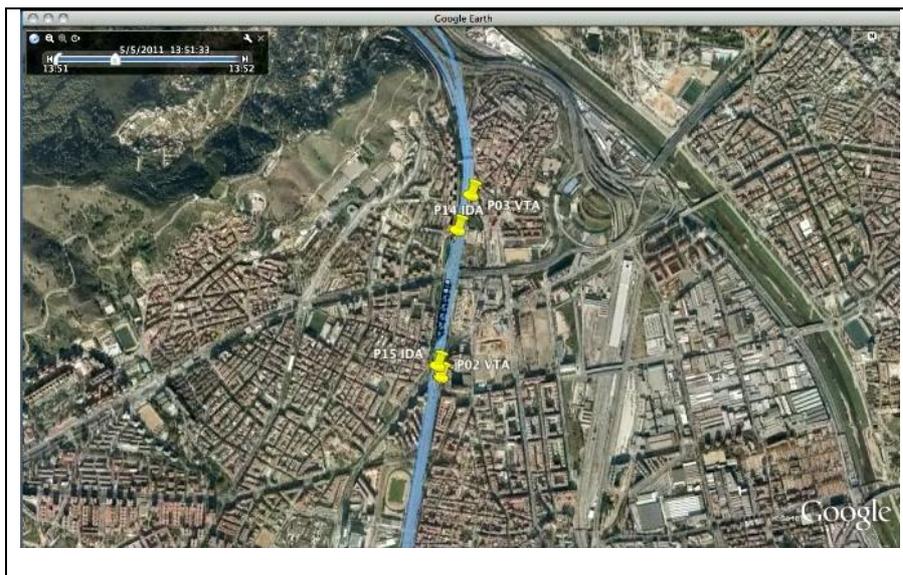


FIGURA 3: Ejemplo de Google Earth

El resultado final es una línea formada por muchos puntos que se asemeja al histórico juego de la serpiente y mediante el cual se consigue una vista muy atractiva de los resultados obtenidos, con múltiples vistas para el análisis y mejora de los valores de captura y localización para optimizar la aplicación.

2.7. Conclusión

En este capítulo principalmente se detallan las tecnologías y programas usados para la creación del sistema, desde el punto de vista de la utilidad de cada uno de ellos y del potencial que tienen han sido escogidos para cada función.

En la introducción de este capítulo se explica la elección de cada herramienta para cada parte del sistema, y por los resultados obtenidos se puede decir que se tiene la suerte de tener un gran abanico de posibilidad de elegir entre muchas aplicaciones pero con éstas se han logrado los objetivos fijados.

El objetivo en esta parte ha sido dar una visión general de cada una de las tecnologías y la razón de su elección, sin adentrarse en mucho más que en su funcionalidad, lo cual es el objetivo del siguiente capítulo.

CAPITULO III: CARACTERISTICAS DEL SISTEMA

3.1 Introducción

Este capítulo tiene como objetivo explicar detalladamente el sistema, primero desde el punto de vista de los diferentes programas para su desarrollo, los cuales han sido necesarios hasta terminar con una breve guía de uso.

Los programas escogidos para el diseño del sistema han sido los siguientes:

El entorno de programación Netbeans por ser el más familiar para programar en Java, con unas herramientas muy útiles en la conexión de base de datos, los conectores jdbc para trabajar directamente desde Java a la BBDD elegido por su facilidad de uso y por la amplia documentación encontrada.

La BBDD en MYSQL como anteriormente se ha explicado y por último para mostrar todos los resultados la hoja de cálculo de Microsoft Excel que con las tablas dinámicas y los gráficos se obtienen una visión clara y sencilla de los datos en principio en bruto recibidos de las balizas.

A continuación se explica el funcionamiento de una manera general de los pasos que se realizan en cada programa, desde la recepción de los datos por FTP, como el procesamiento de los archivos en JAVA como el envío a la BBDD y finalmente la presentación en Excel.

Por último se detalla principalmente el programa en JAVA paso a paso, con el código del mismo, es aquí donde más dificultades se han encontrado para poder conseguir los objetivos deseados en un principio.

Con la guía de uso para la instalación y la puesta en funcionamiento del sistema y una breve conclusión finaliza este capítulo.

3.2 Entorno de desarrollo

3.1.1. NETBEANS

Se ha usado para este servidor el entorno de desarrollo NetBeans para la implementación de la aplicación, programada en java con uso de conexiones a BBDD, aunque se podría haber usado cualquier otro IDE como Eclipse, elegí este por la familiaridad que tenía por haberlo usado anteriormente en asignaturas.

El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para diferentes lenguajes de programación. Existe además un número importante de módulos para extender el IDE NetBeans.

El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Ha sido elegido por su facilidad de uso y su integración ya que contienen todos los elementos necesarios directamente integrados, consiguiendo una comodidad muy favorable para la programación, además es un IDE creado por Sun para programar en Java lo que aporta mucha documentación concisa.

Da la posibilidad de crear aplicaciones para todas sus plataformas, ya que desde la misma página web de Sun se pueden descargar múltiples herramientas con diversos módulos que se integran con la misma aplicación con mucha sencillez.

3.1.2. JDBC

En la tarea de conectar el código Java con la BBDD MySql he utilizado los conectores JDBC, que es una interfaz de programación de aplicaciones desarrollado por Sun como Netbeans que permite la ejecución de operaciones sobre bases de datos, a la que se accede con parámetros de conexión específicos que se delimitan dentro del mismo Netbeans.

En el momento que ya está realizada esta conexión, se pueden realizar cualquier tipo de tareas sobre la BBDD a la que se tenga permiso: consulta, actualización, creación, modificación y borrado de las tablas que previamente han sido creadas para cumplir con los campos que requiere el programa.

3.1.3. SWING

La interfaz gráfica de la aplicación se ha desarrollado a partir de la librería `javax.swing` que es estándar y pertenece a las JFC (Java Foundation Classes), es una biblioteca gráfica desde su versión 1.2 y fue creada a partir de “`java.awt`” es de fácil manejo y permite una interfaz adaptada a cada sistema operativo sin cambiar el código.

Swing permite perfectamente añadir o quitar labels y textbox en tiempo de ejecución, además existen librerías o frameworks para facilitar la creación de ventanas e incluso añaden componentes nuevos.

3.1.4. EXCEL y TABLAS DINÁMICAS

En que consiste esta hoja de cálculo está explicado en el segundo capítulo pero para esta parte hay una particularidad, que son la posibilidad de crear tablas dinámicas que es la razón fundamental de elegir este programa para la presentación de los resultados.

Una tabla dinámica sirve para obtener informes resumidos a partir de los datos que tienes en una hoja de cálculo. A diferencia de la herramienta Subtotales, la tabla dinámica aparece en una hoja de cálculo distinta a la que tiene los datos, eso sí en el mismo libro.

La posibilidad de crear informes a partir de estas tablas es un valor añadido a esta herramienta, con un informe de tabla dinámica puede resumir, analizar, explorar y presentar un resumen de los datos de la hoja de cálculo o un origen de datos externos.

Un informe de tabla dinámica es especialmente útil cuando tiene una larga lista de cifras para sumar y los datos agregados o subtotales podrían servir para mirar los datos desde perspectivas diferentes y comparar las cifras de datos similares.

También es posible en Excel hacer gráficos dinámicos, a partir de estas tablas obteniendo un resultado más global y vistoso. Una vez creadas las tablas dinámicas, se pueden modificar y los gráficos se modifican conforme a las tablas.

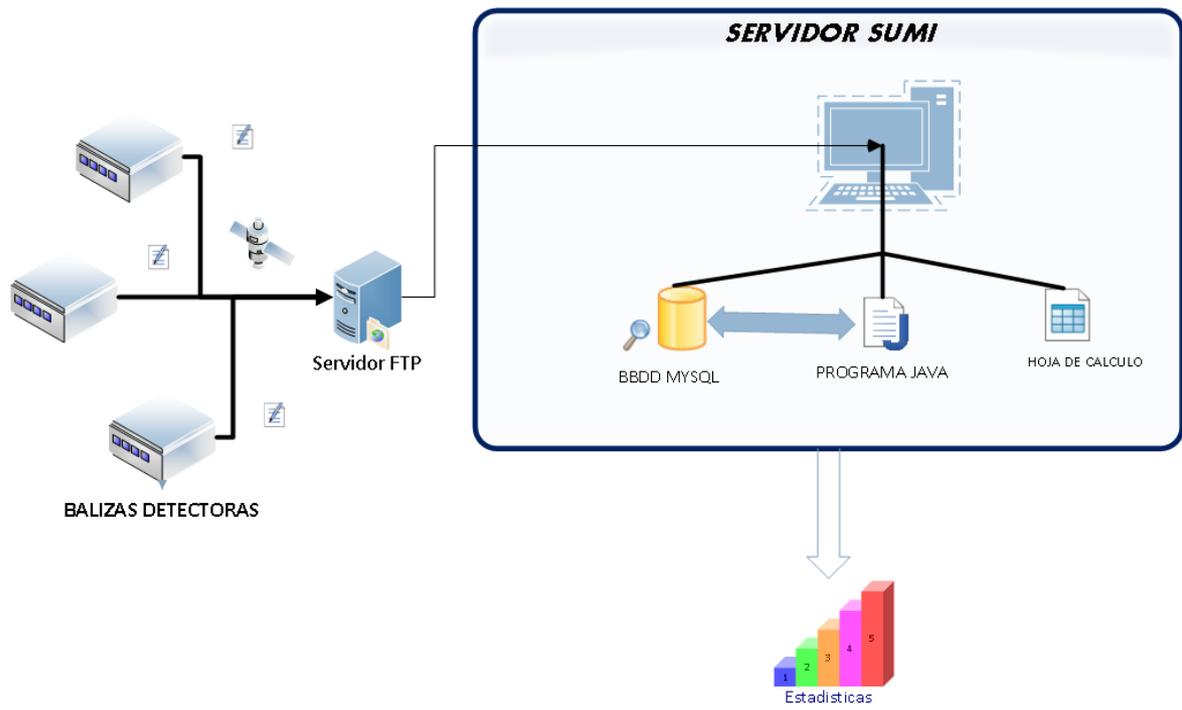


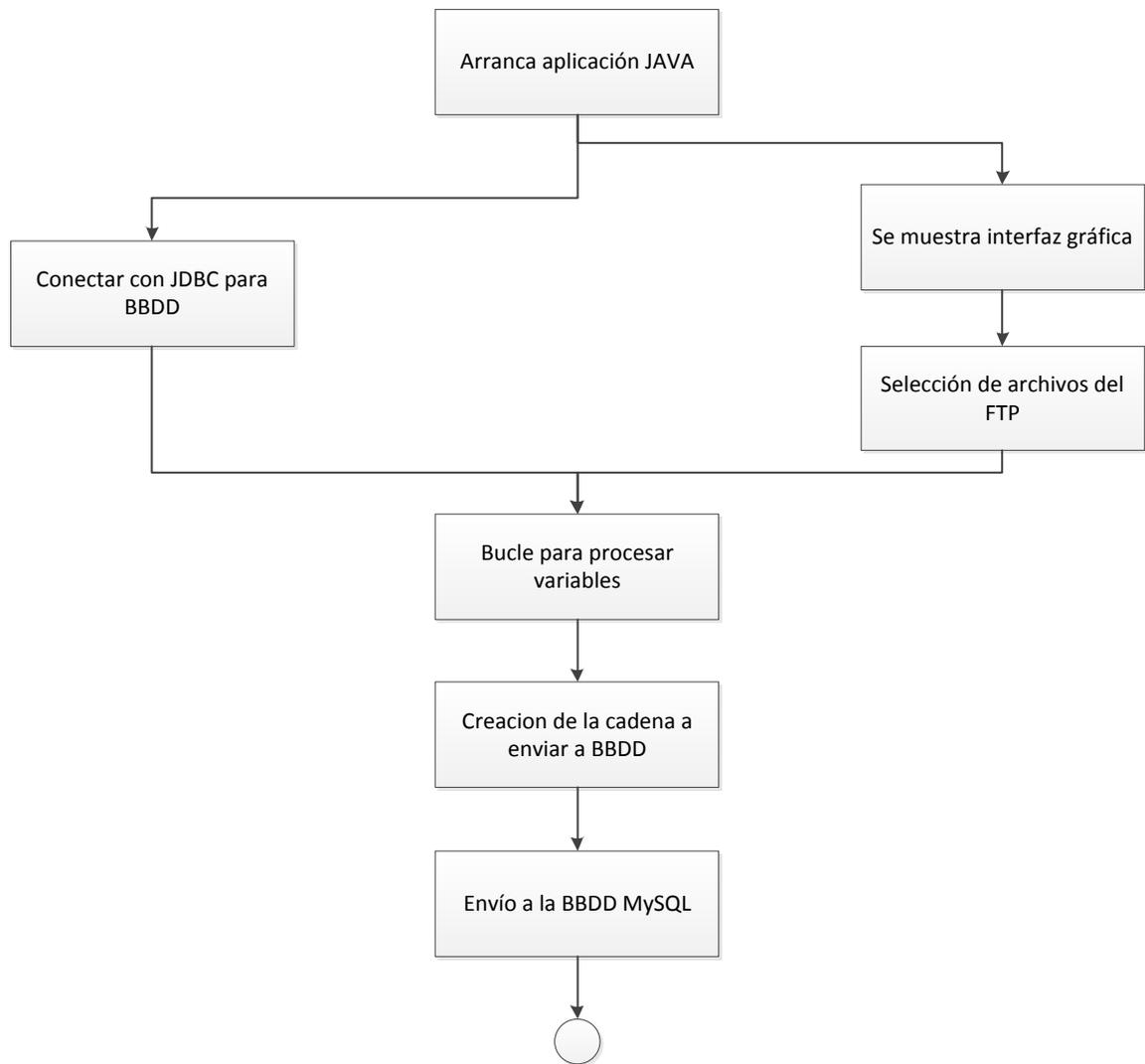
FIGURA 4: Esquema del proyecto

3.2. Funcionamiento

La aplicación Java del servidor arranca y queda permanentemente activa, ésta se conecta una BBDD (MySQL) mediante conectores JDBC, entonces se muestra una interfaz gráfica de usuario, en ésta se decide que archivos del servidor FTP procesar, los cuales serán almacenados en carpetas locales, la elección de los archivos es debido a que este servidor está preparado para recibir múltiples tramas de distintos clientes a la vez.

Una vez seleccionados los archivos comienza un bucle para recoger en variables cada campo de cada archivo y de este modo ir fabricando las cadenas que se envían a la BBDD previamente conectada.

Cuando se reciben todas las detecciones se guardan en la “tabla de detecciones”, se calcula el número total de detecciones que la baliza detectora realizó durante el tramo anterior a partir de los números del atributo “Id detección”.



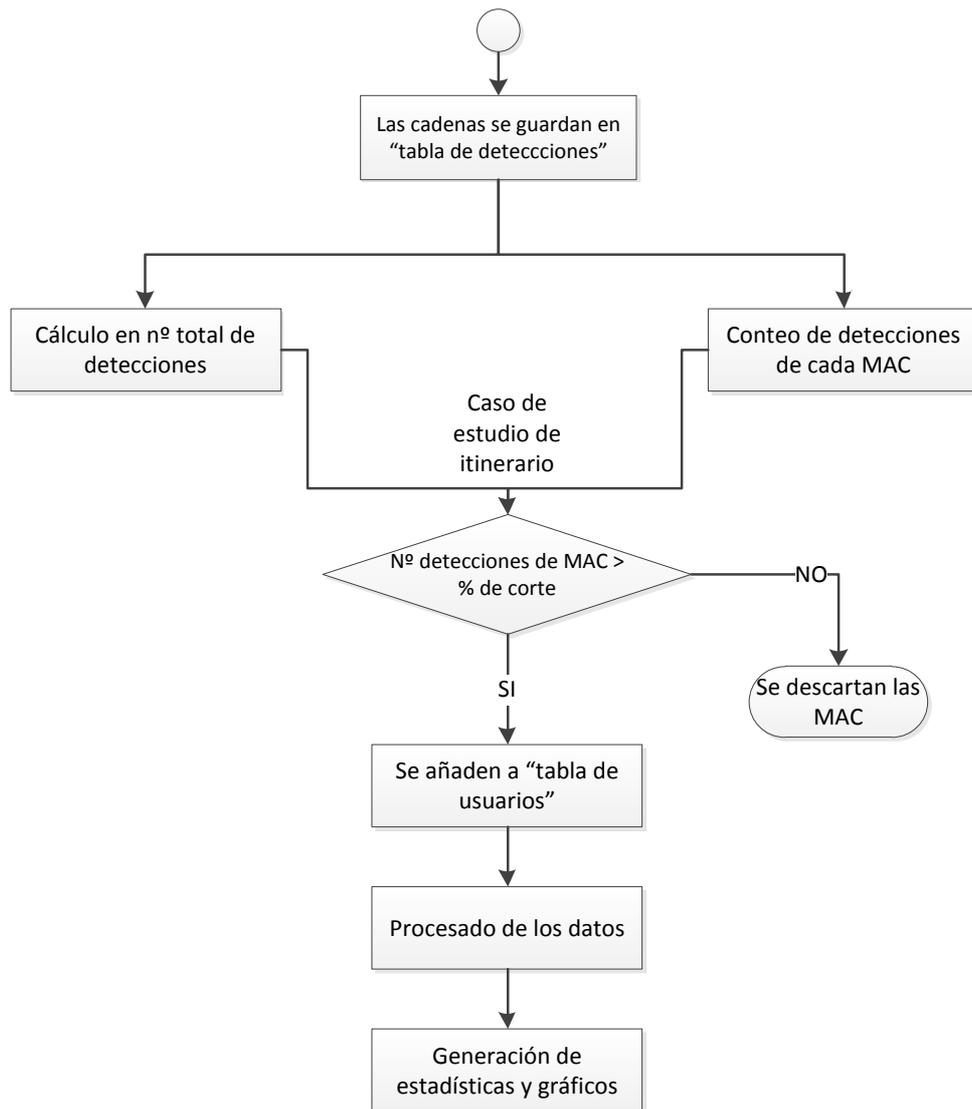


FIGURA 5: Funcionamiento aplicación

A continuación se cuenta para cada MAC el número de veces que ha sido detectada, es decir, se cuentan las entradas con esa MAC que pertenezcan al mismo lugar, de modo que se sabrá el número de veces que ese dispositivo fue detectado durante todo el tiempo de escucha.

Para el caso de hacer el control de un itinerario, si este número es inferior a un porcentaje de corte, estas entradas se considerarán correspondientes a dispositivos externos al vehículo que han sido detectados en algún momento puntual y no deben ser tenidas en cuenta, por ejemplo otro vehículo que se para cerca en el mismo semáforo.

Las direcciones MAC que superen el porcentaje de corte serán incorporadas en otra tabla llamada "tabla de usuarios", descartando así las detecciones que no pertenecen a usuarios que nos interesan. De esta nueva tabla ya se procesan los datos con un programa que permite gestionar bases de datos a través de una interfaz.

Con este programa podremos ver los datos que nos interesan para el estudio estadístico y también gráficos creados a partir de la “tabla de usuarios”.

Es muy interesante por último destacar el uso de la interfaz de seguimiento GPS (Google Earth) para la visualización de recorridos, creando un archivo kml que se ejecuta en el programa. El resultado es un video en el que se ve el itinerario del dispositivo sobre el mapa en relación a los puntos por donde ha sido detectado y la hora de los mismos.

3.3.Aplicación JAVA

En primer lugar y como requisito necesario para la programación se invocan las librerías para conectar con las base de datos mediante el `import java.sql.*` para poder trabajar desde Java con la BBDD SQL en su posterior conexión.

```
import java.awt.*;  
import java.awt.List;  
import java.awt.event.*;  
import javax.swing.*;  
import java.io.*;  
import java.util.*;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import java.sql.*; //Importar las funciones SQL
```

FIGURA 6: Import de funciones SQL en Java

Se crea una clase Servidor que será donde se declaren las principales variables para la aplicación, a continuación se configura la interfaz gráfica de usuario para la posterior elección de los archivos que se desean procesar. En esta interfaz se configura el formato y los botones que la forman, ésta hereda las propiedades de JFrame.

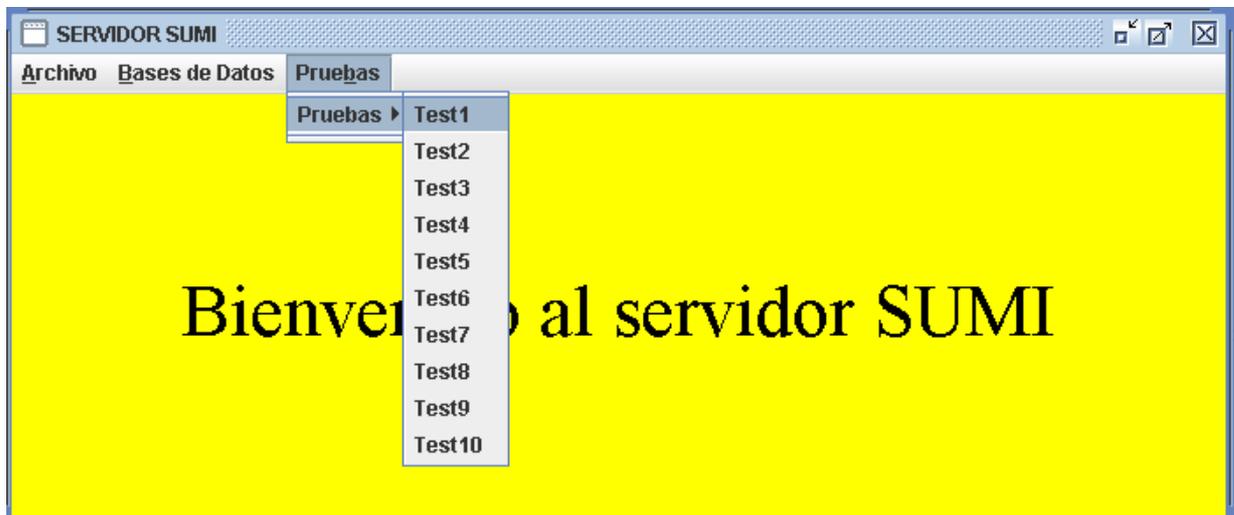


FIGURA 7: Swing de la aplicación

Se declara el main del programa creando un nuevo objeto Servidor llamado aplicación sobre el cual se desarrolla todo el proceso.

La clase ManejadorEventos proporciona la capacidad de seleccionar los archivos de las pruebas de campo recibidas por parte de las balizas.

Los archivos han sido obtenidos mediante la clase clienteftp creando una conexión con el servidor de datos, una vez procesados los archivos se generan cadenas con los ficheros de texto en bruto.

La descarga del servidor FTP al ordenador se realiza para tener una copia de seguridad y para trabajar en local, lo que aporta mayor seguridad de los archivos ante un error en la conexión.

```
package clienteftp;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.URL;
import java.net.URLConnection;
public class Main {

    public static void main(String[] args) throws IOException {
        List lista= new ArrayList();
        clienteftp clie=new clienteftp("192.168.1.3","Usuario1", "");
        lista=clie.listar();//asignamos a lista lo q nos devuelve el metodo
        for (Object lis : lista) { //recorremos la lista
            System.out.println(lis.toString());
        }
        System.out.println(" \n CAMBIAMOS DE DIRECTORIO Y LISTAMOS \n");
        clie.setDirectorio("C:/FTP");//cambiamos el directorio
        lista=clie.listar(); //asignamos a lista lo q nos devuelve el metodo
        for (Object lis : lista) { //recorremos la lista
            System.out.println(lis.toString());
        }
    }
}
```

FIGURA 8: Clase FTP

Una vez seleccionada la carpeta, se muestran por pantalla el nombre de los archivos que contiene y se pasa a procesar los campos de cada archivo.

Así es como se reciben los archivos por parte de las balizas detectoras, lo que se denominan los archivos en bruto:

```
1 PRUEBA_0001_CARTAGONOVA_2011-03-24_17-10.txt
2 0234
3 0002 A0:07:98:10:76:BE 5a0204 GT-S5600
4 0001 00:24:03:B3:24:98 5a0204 Titef
5 0001 E0:A6:70:5D:04:1E 5a0204 Antonio
6 0001 00:26:CC:99:FD:C1 5a0204 Nokia_2730_classic
7 0001 3C:F7:2A:D2:0F:FE 5a0204 [Desconocido]
8 0001 00:23:B4:99:CC:05 5a0204 Nokia_3120_classic
9 0001 5C:57:C8:5C:2A:29 5a020c Fdavid
10 0001 20:D6:07:12:9D:7F 5a0204 [Desconocido]
11 0001 00:22:FC:79:EB:1B 520204 Nokia_jam
12 0002 CC:55:AD:8E:2B:85 7a020c BlackBerry_9800
13 0001 00:25:E7:BA:0D:AF 5a0204 [Desconocido]
14 0001 18:14:56:1E:17:12 5a020c Gny
15 0001 00:23:D6:56:D6:5C 520204 Padrino+SGH-M310
16 0003 00:26:69:21:54:7E 5a0204 Mateo
17 0001 40:5F:BE:C3:F0:92 7a020c BlackBerry_9780
18 0001 EC:9B:5B:90:EB:26 520204 [Desconocido]
19 0001 00:26:5D:46:32:E8 5a0204 Katy
```

FIGURA 9: Archivo recibido en bruto

A través de múltiples variables que se declaran, se guardan en cada una los campos que son necesarios mediante la clase StringTokenizer que ayuda a dividir una cadena en sub-cadenas o tokens en base a otra cadena que tienen un delimitador que en este caso es un espacio en blanco.

```
st = new StringTokenizer(sCadena, "\t");
do{
    id++;
    num_det=st.nextToken();
    mac=st.nextToken();
    tipo=st.nextToken();
    nombreBT=st.nextToken();
} while(st.hasMoreTokens());
```

FIGURA 10: Separador de variables con StringTokenizer

Una vez separados los campos de cada cadena se extrae del nombre del fichero la fecha y la hora, que previamente la baliza creó y que no vienen incluidas en las líneas de cada archivo.

Se crea un bucle para recorrer todos los ficheros e ir completando todas las variables y a continuación se declara la cadena para introducirla posteriormente en la BBDD.

```
// Aquí se encuentran guardadas en variables los datos de cada archivo

// Ahora se pasa a la Base de Datos

CADENA = "INSERT INTO
"+nombreTabla+"(id,num_det,mac,tipo,nombreBT,fecha,hora)
VALUES("+id+", "+num_det+", "+(String)mac+", "+(String)tipo+", "+(String)nombreB
T+", "+(String)fecha+", "+(String)hora+"");
```

FIGURA 11: Inserción de la cadena a BBDD

La aplicación establece la conexión a la BBDD mediante los conectores jdbc.

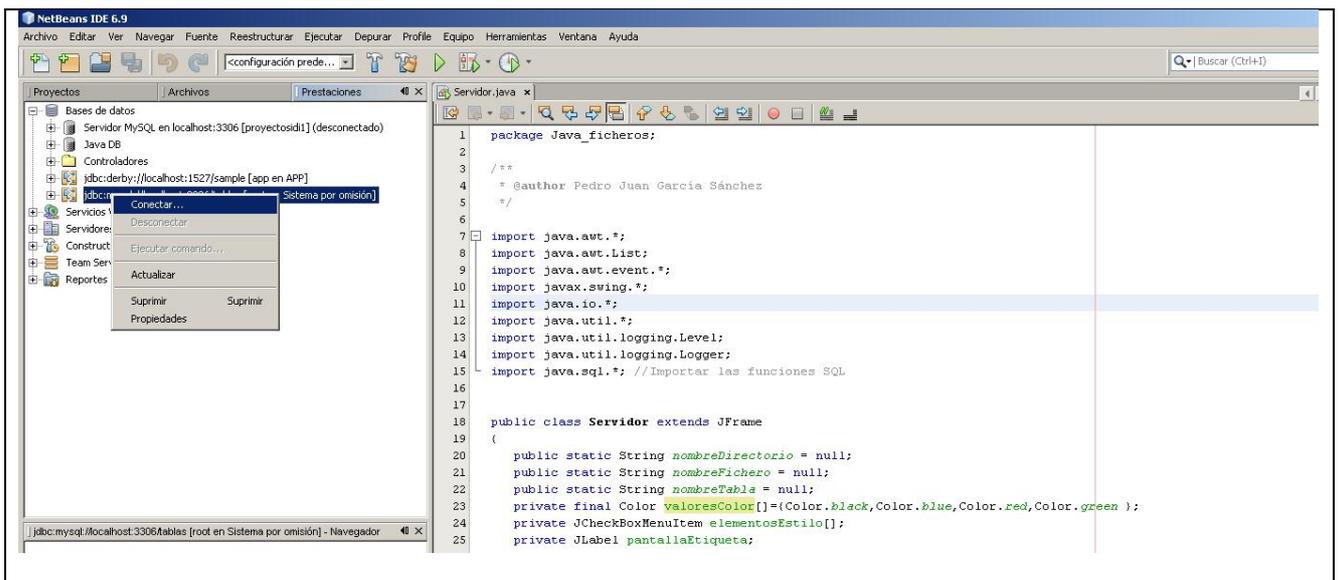


FIGURA 12: Conectar a BBDD en Netbeans

Posteriormente se procede a introducir la cadena con los datos ya clasificados para su análisis, mediante la clase executeUpdate que ejecuta una instrucción SQL determinada en este caso INSERT.

```

try
{ //Registramos el JDBC Driver a utilizar

    Class.forName("com.mysql.jdbc.Driver");
    //Establecemos la conexión con la base de datos
    System.out.println("antes de connection");
    Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost:3306/tablas","root","
    System.out.println("despues de connection");
    //Creamos un objeto Statement para generar nuestra instrucción SQL
    Statement sqlCadena = conexion.createStatement();
    /* Se imprime por pantalla la cadena que se va a enviar a la base de datos
    y se ejecuta executeUpdate que ejecuta la instrucción SQL determinada por CADENA */
    System.out.println(CADENA);
    sqlCadena.executeUpdate(CADENA);
    System.out.println("Cadena introducida correctamente");
    sqlCadena.close();
}catch (ClassNotFoundException ex) {
    Logger.getLogger(listaArchivos.class.getName()).log(Level.SEVERE, null, ex);
}
    catch(SQLException e){} //Captura posible error de SQL

```

FIGURA 13: Conexión en Java con MySQL

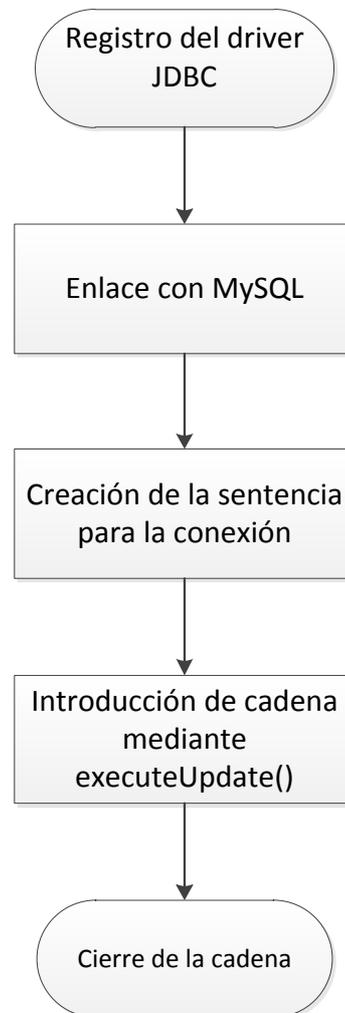
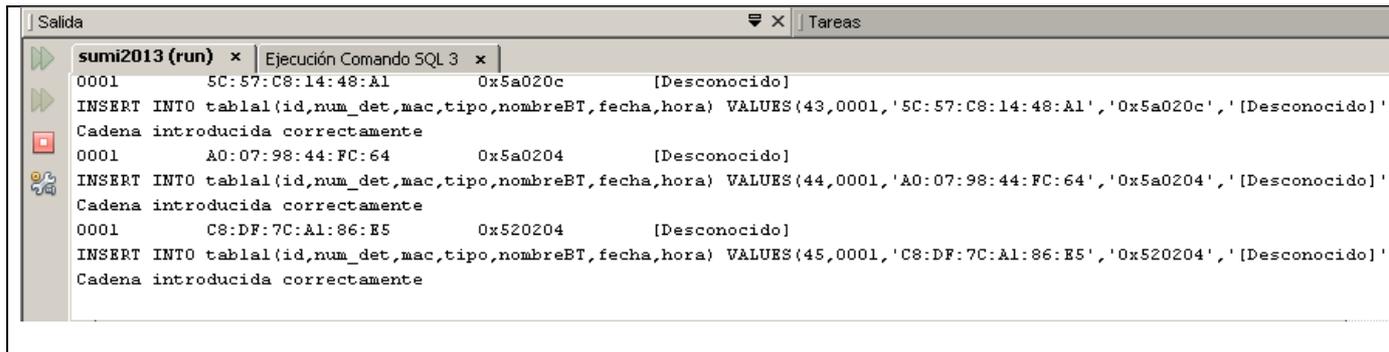


FIGURA 14: Secuencia de la cadena

Tras este proceso se puede observar en tiempo real a través de Netbeans que los registros que han sido introducidos en la BBDD se han realizado correctamente, esto fue muy útil en las diferentes fases de depuración.

Para poder comprobar que las cadenas enviadas eran correctamente introducidas a la base de datos, se colocaron sentencias de comprobación que validaban las sentencias sql.

En esta parte aparecieron varios problemas, como que la cadena construida no podía ser insertada por no tener los campos que requería la base de datos, o que algún campo se excedía de la longitud fijada.



```
sumi2013 (run) x | Ejecución Comando SQL 3 x
0001      5C:57:C8:14:48:A1      0x5a020c      [Desconocido]
INSERT INTO tablal(id,num_det,mac,tipo,nombreBT,fecha,hora) VALUES(43,0001,'5C:57:C8:14:48:A1','0x5a020c','[Desconocido]'
Cadena introducida correctamente
0001      A0:07:98:44:FC:64      0x5a0204      [Desconocido]
INSERT INTO tablal(id,num_det,mac,tipo,nombreBT,fecha,hora) VALUES(44,0001,'A0:07:98:44:FC:64','0x5a0204','[Desconocido]'
Cadena introducida correctamente
0001      C8:DF:7C:A1:86:E5      0x520204      [Desconocido]
INSERT INTO tablal(id,num_det,mac,tipo,nombreBT,fecha,hora) VALUES(45,0001,'C8:DF:7C:A1:86:E5','0x520204','[Desconocido]'
Cadena introducida correctamente
```

FIGURA 15: Salida por pantalla del programa

Además de las clases nombradas hay una clase aparte importante que es getFiles la cual devuelve una lista con todos los archivos que se encuentran dentro de la carpeta indicada.

Una vez concluida esta primera etapa, se procede a extraer estos datos para procesarlos, mediante el uso de sentencias sql a través del gestor phpmyadmin.

En primer lugar se selecciona la tabla que se quiera analizar, y mediante la exportación de al formato CSV, se obtiene el fichero para Excel con el que se procede a analizar cada caso de estudio de una forma más eficiente y sencilla.

Mostrando registros 0 - 29 (~59 total) , La consulta tardó 0.0032 seg)

```
SELECT *
FROM `tabla1`
LIMIT 0 , 30
```

Mostrar todo > >> | Mostrar : Fila de inicio: 30 | Número de filas: 30 | Cabec

Ordenar según la clave: Ninguna

+ Opciones

	id	num_det	mac	tipo	nombreBT	fecha	hora
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	2	38:E7:D8:23:D5:DC	7a020c	HTC_HD2	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	1	00:22:66:40:86:7F	520204	[Desconocido]	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	1	A8:7E:33:6D:79:C1	5a020c	Roca66	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	4	2	44:A4:2D:77:C2:C4	5a0204	Vintisinco	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	5	2	9C:18:74:DF:82:D6	5a0204	Nokia_5130c-2	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	6	2	00:1D:28:8B:D4:E3	5a0204	Miguel	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	7	1	00:26:CC:09:F1:2B	5a0204	[Desconocido]	2011-05-15	16-06
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	8	2	4C:54:99:1B:5D:E0	58020c	miguel	2011-05-15	16-06

FIGURA 16: Tabla en phpmyadmin

Método de exportación:

Rápido - mostrar sólo el mínimo de opciones de configuración

Personalizado - mostrar todas las opciones de configuración posibles

Filas:

Volcar algunas filas

Número de filas:

Fila con la que comenzar:

Volcar todas las filas

Salida:

Esta opción para exportar a Excel la tabla de la base de datos

CSV para datos de MS Excel

usar esto para exportaciones futuras

Compresión:

FIGURA 17: Exportar tabla a Excel

En segundo lugar y una vez las tablas ya están exportadas a la hoja de cálculo se crea una tabla dinámica por cada caso de estudio, en la cual se pueden ir aplicando diferentes filtros.

Gracias a estos filtros es posible un resumen de la información acerca de una franja horaria de detecciones, un tipo de dispositivo específico, ordenar por número de veces detectados los dispositivos, etc.

Estas tablas dinámicas permiten modificar los métodos de filtrado de los datos, para mostrar los resultados obtenidos de una forma muy intuitiva y fácil de cara al usuario final.

Por último y a partir de las tablas dinámicas creadas y con los filtros activados según cada caso se crean los informes de resultados procesados, construyendo las vistas a base de gráficos o códigos de colores para mostrar un resultado final más sencillo y rápido de analizar y con el cual se consigue un modo mucho más atractivo de mostrar toda la información que observando innumerables filas de una BBDD.

3.4. Conclusión

En este capítulo se ha explicado con detalle todo el desarrollo de la aplicación desde el punto de vista de entorno de desarrollo, programación y ejecución.

Para cada elemento del sistema se ha detallado sus fases de desarrollo, depuración y validación. En esta fase es donde se concentra el núcleo de procesamiento paso a paso necesario para obtener resultados viables para la práctica.

Es en esta parte del proyecto es donde se ha encontrado mayor dificultad y problemas, pero también donde los resultados han sido más visibles y se ha aprendido más por ser la fase con mayor dificultad.

CAPITULO IV: CASOS PRÁCTICOS

4.1. Estadio Cartagonova

Este estudio fue realizado para comparar la afluencia de aficionados al estadio Cartagonova desde distintos puntos del estadio, con el fin de conseguir un resultado estadístico del uso de la tecnología bluetooth.

El estudio consistió en colocar balizas detectoras en las puertas del estadio en dos partidos y mediante las capturas obtenidas y el número de aficionados que acudieron al estadio poder dar un dato contrastado del uso de esta tecnología.

El fin es poder usar esta estadística para extrapolarla a otros campos y así poder estimar el número de personas que se encuentran en un sitio, a partir de este estudio el cual es posible cuantificar casi exactamente las que entran al estadio y que en otros casos es imposible.

Determinar este estudio de comportamiento es uno de los primeros objetivos fijados del proyecto, el cual se observó que podría ser muy útil para muchos más casos.

En ambos partidos de fútbol en los que se realizaron las pruebas hubo una entrada de 6.500 y 5.800 respectivamente, en el primero se capturaron alrededor de 1700 y en el segundo 1300.

El tanto por ciento obtenido es del 26,15% en el primero y 22,41% en el segundo, lo que en media se obtiene un 24,28%, este dato muestra que aproximadamente un cuarto de los usuarios llevan conectado el bluetooth en sus dispositivos móviles.

4.2. Línea de tren

En este caso de estudio las balizas detectoras fueron colocadas dentro de líneas de tren para observar el comportamiento de los usuarios, donde se suben y se bajan, cuanto tiempo permanecen dentro, las paradas menos frecuentadas, etc.

Las balizas fueron colocadas estratégicamente para abarcar todo el tren y con los datos recogidos, el servidor procesa por cada parada previamente posicionada los dispositivos que hay nuevos, los que siguen en el tren o los que ya no se encuentran.

A partir de todos estos datos se obtiene una información muy útil para la empresa puesto que sabiendo cual puede ser el tiempo medio de estancia de un pasajero en el interior del tren, las paradas más o menos transitadas, es posible optimizar la línea para que sea más eficaz.

En vista de los resultados en las pruebas realizadas, se puede decir que en estos casos no existía ninguna parada que fuese prescindible pero si hay alguna que es usada por muchas más personas que suben y bajan con respecto a las otras.

En conclusión, para cualquier empresa de trenes puede ser útil esta información siempre que se realice un estudio completo para poder valorar todas las líneas, ya que cada una es única y no se puede extrapolar el uso de una a otra, este estudio completo puede abaratar costes a dicha empresa.

4.3. Worten y Mediamarkt

La afluencia de clientes a los establecimientos de Worten y Mediamarkt son de un mismo perfil y buscan productos similares, y por esta razón se ha realizado el caso práctico para comparar a que tienda acude más gente y que comportamiento tienen.

Este estudio comienza situando en la puerta de los dos establecimientos una baliza detectora, a la misma hora y durante el mismo tiempo para así procesar los datos en igualdad de condiciones.

De esta forma se puede observar que los clientes de Mediamarkt permanecen dentro del local durante más tiempo que en Worten en un 20% aproximadamente, también se observó que las detecciones en Worten eran mayores cerca del mediodía mientras que en Mediamarkt tenían un flujo regular.

Una curiosidad que había era el poder comprobar que tanto por ciento de clientes hacían una comparativa de precios entre los dos, al estar tan cerca en este caso en Cartagena, y fue una sorpresa el comprobar que prácticamente el 98% no echa un vistazo en la competencia para comparar.

En conclusión, el número de detecciones fue en un 30% mayor en Mediamarkt, con mayor tiempo dentro del local, lo cual no es un dato definitivo al hacer las pruebas en dos sábados pero se puede decir que tiene mayor afluencia Mediamarkt que Worten.

4.4.Coche de seguridad

Este caso de coche de seguridad se basa en el estudio de la ruta que realiza un guarda de seguridad con su coche por una zona y con una ruta establecida, en este caso el estudio se realizó en base al coche de seguridad del polígono residencial Santa Ana.

Antes de colocar la baliza detectora dentro del vehículo, lo cual no pudo realizarse al no tener permiso por parte de la empresa responsable, se establecieron unos puntos gps por los cuales el coche debe de pasar, y con un coche particular se empezaron las pruebas.

Mediante la herramienta Google Earth se pusieron unos puntos en el mapa y la baliza iba mandando su posición gps y esta se contrasta con la ruta que debe de realizar el guarda de seguridad. Mediante los archivos kmz se puede ver una ruta como se explica en el apartado 2.6.

Recopilando todos los datos se puede observar si se ha completado la ruta según lo previsto, cuantas paradas ha hecho el coche, si ha habido alguna incidencia o cuantas veces en un turno de trabajo puede completar la vigilancia.

CAPITULO V: CONCLUSION Y LINEAS DE FUTURO

5.1. Conclusión

Se ha desarrollado este proyecto con el objetivo de cumplir los puntos iniciales propuestos, la creación de un servidor de monitorización de dispositivos electrónicos que muestre informes y estadísticas a través de una interfaz.

Este servidor con la misión de recibir, tratar y mostrar datos responde a la necesidad inicial de transformar los archivos en bruto en informes y gráficos útiles para analizar distintos casos prácticos.

Durante la creación del servidor se ha descubierto el potencial de Java con MySQL y la multitud de posibilidades que nos da, así como el abanico de resultados que ofrece exportar a Excel las tablas e ir construyendo tablas dinámicas y gráficos.

Los objetivos propuestos han sido cumplidos y aunque la función del servidor de cara al usuario es casi transparente ha planteado multitud de retos y de complicaciones que con constancia han sido resueltos, con el fin de que el uso del mismo sea lo más práctico posible.

El servicio que ofrece este servidor al usuario va a permitirle poder analizar con mayor rigor distintos casos de estudio que requiera y con esta versatilidad ha sido creado, para conseguir de diferentes ámbitos resultados simples y útiles.

5.2. Líneas de futuro

Este proyecto tiene la capacidad de mostrar datos difíciles de analizar de manera simplificada, pero este objetivo principal se plantea como punto de partida.

Como una modificación del servidor para que no sólo sea una estación receptora e interfaz de informes y gráficos, se plantean varias líneas de investigación futuras como éstas:

- Generación de alarmas automáticas:

Se plantea que a partir de unos valores de tiempo o cantidad de detecciones de dispositivos, el sistema sea capaz de generar alarmas para avisar de que algo no funciona como es espera.

Un caso muy claro sería que el guarda de seguridad que tiene que recorrer un perímetro para su vigilancia no pasase por una determinada zona en mucho tiempo o hiciese paradas más largas de lo esperado.

- Detección de variación de flujo:

Un problema que puede surgir en el proceso de detección de dispositivos es que no exista una continuidad, es decir, que se repita una MAC repetidas veces en un lugar donde solamente debe pasar una vez.

Un ejemplo puede ser en la puerta de un estadio, donde se valida la entrada, que se repitan detecciones significaría que hay un bloqueo u otro tipo de problema.

El caso de atascos de tráfico puede ser realmente útil y económico ya que si colocadas balizas en rotondas o puntos conflictivos de tráfico, ocurre que no hay una variación de dispositivos significa que hay atasco o algún accidente.

Mediante este sistema se ahorraría en cámaras de seguridad de mucho mayor coste y también se podrían optimizar los semáforos para menor bloqueo del tráfico.

- Análisis de patrones de comportamiento:

Teniendo los resultados de paso de gente a un establecimiento o el tiempo que se encuentra un usuario dentro se pueden obtener patrones de comportamiento.

Como línea futura se puede crear una aplicación que tenga la capacidad de predicción en base a este comportamiento, así conseguir anticiparse a los hábitos de los consumidores.

CAPITULO VI: MANUAL DE INSTALACIÓN

Para el funcionamiento del servidor son necesarios varios programas con sus configuraciones que son los siguientes:

- Servidor Apache: Se ha optado por instalar Xampp que es una forma fácil de instalar una distribución de Apache que contiene MySQL y es simple de instalar. Se obtiene descargándolo de su página y tras descomprimir el archivo, se instala y ejecuta.

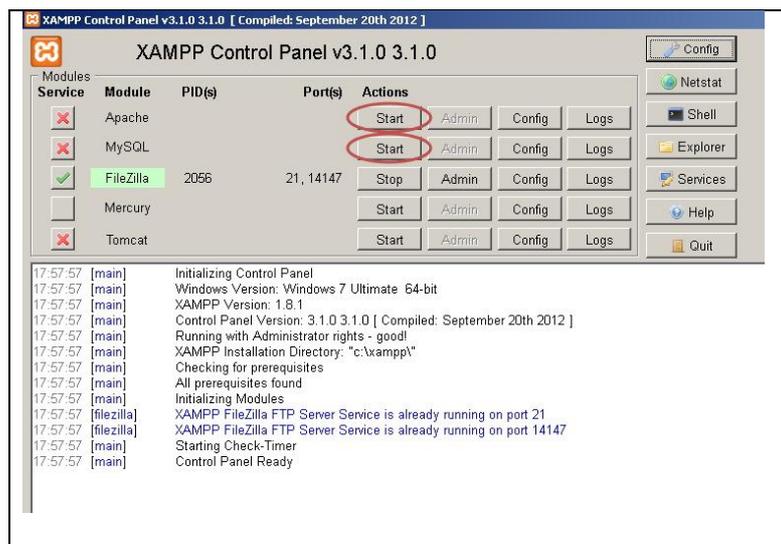


FIGURA 18: Panel de control de Xampp

- Entorno de desarrollo JAVA: Netbeans es la plataforma para el desarrollo de la aplicación, que viene con el Java Development Kit (JDK). También es necesario el driver o conector a la base de datos MySQL y que se copia dentro de la carpeta del JDK.

La configuración de Netbeans consiste en incluir un nuevo controlador de bases de datos MySQL en la pestaña Runtime. Dentro de la sección Tiempo de ejecución > Bases de datos > Drivers hay que hacer clic con el botón derecho sobre MySQL y seleccionar Conectar usando... para que aparezca la pantalla de configuración de la conexión. En la sección Nombre selecciona MySQL (Conector/J Driver) en la sección Database URL se cambia por la dirección del servidor de bases de datos. Posteriormente se escribe el nombre de usuario y contraseña para que Netbeans la recuerde durante la conexión.

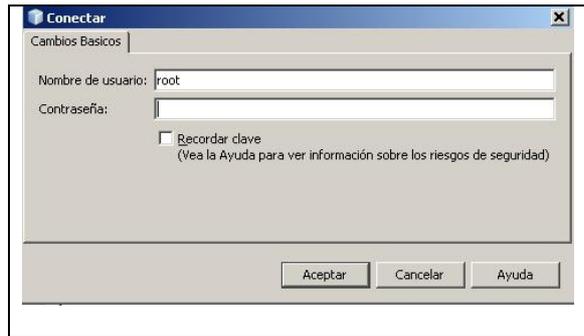


FIGURA 19: Login en base de datos

- La implementación del código se realiza creando un nuevo proyecto e incorporando las clases, la compilación la realiza Netbeans automáticamente.
- El servidor FTP para recibir los archivos se crea en Java incluido en el código del servidor, y en el proceso de ejecución funciona todo el programa.
- La tabla que se genera en MySQL se exporta a Excel mediante phpmyadmin que viene en el paquete de Xampp y en la hoja de cálculo se consiguen los informes y gráficos deseados.

CAPITULO VII: BIBLIOGRAFIA

<http://es.wikipedia.org>

<http://www.mityc.es/es-ES/guia/Paginas/QueesycomoseusaunarchivoKML.aspx>

http://earth.google.com/intl/es/outreach/tutorial_kmz.html

http://earth.google.com/intl/es/outreach/tutorial_promote.html

<http://es.wikipedia.org/wiki/KML>

<https://developers.google.com/kml/documentation/topicsinkml?hl=es>

<http://www.chuidiang.com/java/mysql/EjemploJava.php>

<http://es.kioskea.net/contents/internet/ftp.php3>

<http://www.compujuy.com.ar/postx.php?id=54>

<http://www.oracle.com/es/products/mysql/index.html>

http://hojasdecalculo.about.com/od/Excel_avanzado/ss/Paso-A-Paso-C-Omo-Hacer-Una-Tabla-Din-Amica.htm

<http://es.wikipedia.org/wiki/MySQL>

<https://netbeans.org/>

http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java

<http://monillo007.blogspot.com/2007/11/conectar-una-base-de-datos-en-mysql-con.html>

<http://www.apachefriends.org/es/xampp.html>