

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



# **PROYECTO FIN DE CARRERA**

**DESARROLLO DE INTERFAZ PARA LA REPRESENTACIÓN  
GEOGRÁFICA DE MÉTRICAS DE CALIDAD EN OPERADORES  
DE TELEFONÍA MÓVIL**

**Diego Perera Martín**

**JUNIO 2015**



**DESARROLLO DE INTERFAZ PARA LA REPRESENTACIÓN  
GEOGRÁFICA DE MÉTRICAS DE CALIDAD EN OPERADORES  
DE TELEFONÍA MÓVIL**

**AUTOR: Diego Perera Martín**  
**TUTOR: José Alberto Rodríguez Salazar**  
**PONENTE: Jorge E. López de Vergara Méndez**

**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Junio de 2015**



## **Resumen**

Con la aparición de los *smartphones*, han surgido numerosas aplicaciones móviles destinadas a la prueba y monitorización de la calidad en redes de telefonía móvil 2G, 3G o 4G. Estas herramientas disponen de mecanismos para la obtención de métricas de calidad de red móvil medidas en la interfaz radio. Dichas métricas se construyen a través de la información de señalización, datos de referencia de los elementos de red y medidas de los niveles de señal electromagnética. Datos que son almacenados en bruto en el terminal móvil que, para su posterior estudio y entendimiento, necesitan un procesamiento aparte en un sistema que centralice los datos de los diferentes terminales de medida.

Los propósitos de este procesamiento pueden ser varios: localización de problemas en red (*troubleshooting*), detección de problemas en tiempo real (alarmas), análisis general de calidad de red, vigilancia tras trabajos en red, soporte a incidencias en atención a cliente, etc.

Para conseguir estos propósitos, estos sistemas suelen disponer de aplicativos que muestran la información de las métricas (ratios y valores en crudo) en diferentes formatos, como pueden ser: tablas, gráficos, mapas, consolas de alarmas, etc.

El objetivo del presente proyecto consiste en desarrollar una interfaz que transforme los datos obtenidos con este tipo de aplicaciones móviles en un formato que facilite la visualización geográfica de estos datos y que los represente dinámicamente de una manera útil para la monitorización de la red a nivel geográfico y para su comprensión de la forma más fácil e intuitiva posible.

A grandes rasgos, la interfaz debe ser capaz de leer los datos que ofrece una de estas aplicaciones móviles, procesarlos junto con la información de topología de red u otra información relevante y enviarlos al sistema de visualización donde el usuario dispondrá de toda la información ordenada según las características pertinentes, ya sea un rango de fechas, un operador concreto, una región geográfica, etc.

## **Palabras clave**

Representación geográfica, interfaz web, drive test, operadores de telefonía móvil, sistemas de monitorización.

## ***Abstract***

With the advent of smartphones, many mobile applications for testing and monitoring the quality of 2G, 3G or 4G mobile networks have surfaced. These tools have mechanisms for obtaining quality metrics measured on mobile radio network interface. These metrics are constructed through the signaling information, reference data of the network elements and electromagnetic signal level measurements. Raw data is stored in the mobile phone and for further study and understanding, a separate processing is required in a system that centralizes data from different phones.

There may be several purposes for this processing: network troubleshooting, problem detection in real time (alarms), general analysis of network quality, network surveillance after network interventions, supporting customer care support, etc.

To achieve these purposes, these systems usually have applications that show the user metrics (ratios and raw values) in different formats, such as: tables, graphs, maps, alarm consoles, etc.

The objective of this project is to develop an interface that transforms the data obtained with these mobile applications into a format that facilitates geographic visualization of this data and dynamically represent it in a useful way for geographical network monitoring and for as easy and intuitive as possible understanding.

Broadly speaking, the interface must be able to read the data provided by one of these mobile applications, process it, together with the network topology information or another relevant information and send it to the visualization system where the user will have access to all the information organized by some relevant features either a date range, a particular mobile phone operator, geographic region, etc.

## ***Keywords***

Geographical representation, web interface, drive test, mobile operators, monitoring systems.

## *Agradecimientos*

En primer lugar quisiera agradecerle a Fede la oportunidad que me brindó y a José Alberto la ayuda que me ofreció todo este tiempo.

Por supuesto agradecer también a Dusan su paciencia y conocimientos sin los que todo esto hubiera sido mucho más difícil.

Gracias a mi ponente Jorge Enrique.

Especial mención también a Will que tantas veces me echó un cable, a Isa y Laura que me amenizaron el proceso, a Rubén y sus reuniones, a Laura V y a Rober, a Alex, Andrés y Pamela, sin olvidar a las chicas de la oficina y su simpatía: Patri, Sole, Cris, Shira... ¡gracias!

Gracias a Javi y a Borja porque sin ellos a lo mejor no hubiera acabado este camino, a Almu y todos los apuntes que me dejó, a Álvaro y esas noches locas por Oropesa, a Óscar y todas esas horas juntos en la facultad, a Carlos y todos los partidos de fútbol que echamos, a toda la gente de las peceras y del IIC. A Alf y Pebs.

Mil gracias a Lore por ayudarme a pasar el mejor año de mi vida.

Gracias a todo aquel que alguna vez me echó una mano en alguna práctica, o preparando un examen o en lo que fuera. Gracias a Jose por hacer de teleco cuando lo suyo son los bichos ;)

Un millón de gracias a GTM (Great Teacher Mecano!) que me tutoró tantas y tantas horas de forma magistral. ¡Gracias Mec!

Gracias a Luís, por todo, y por siempre.

Y, por supuesto, muchas gracias a mi familia, sobretodo a mis padres, que siempre estuvieron ahí pasase lo que pasase. Os quiero.

Gracias.



# Índice general

<b>1.</b>	<b>Introducción</b>	<b>1</b>
1.1	Marco contextual	1
1.2	Motivación	2
1.3	Estado del arte	3
1.3.1	Introducción	3
1.3.2	Sistemas de medición radio	3
1.3.3	Aplicaciones móviles de medida	5
1.3.4	Drive test	6
1.4	Aplicación web original	7
1.5	Objetivos del proyecto	8
1.6	Organización de la memoria	9
<b>2.</b>	<b>Aplicación web existente</b>	<b>11</b>
2.1	Introducción	11
2.2	Arquitectura de la aplicación	11
2.2.1	Módulo de visualización de KPIs geolocalizados (Maps)	12
2.3	Tecnologías empleadas durante el desarrollo del sistema	13
2.3.1	Ruby on Rails	13
2.3.2	MySQL	13
2.3.3	Google Earth y ficheros KML	14
2.3.4	Git	14
2.3.5	Rails Admin	14
2.4	Base de datos existente	14
2.5	Conclusiones	16

<b>3.</b>	<b>Diseño de la base de datos</b>	<b>17</b>
3.1	Introducción a la base de datos	17
3.2	Cells	18
3.3	Lacs	18
3.4	Las principales tablas	20
3.5	El proceso de carga	22
3.6	Las migraciones	22
3.7	Conclusiones	23
<b>4.</b>	<b>Obtención de los datos</b>	<b>25</b>
4.1	Introducción	25
4.2	Uso de G-NetTrack	26
4.3	Archivo de medidas	29
4.4	Archivo de eventos	32
4.5	Conclusiones	33
<b>5.</b>	<b>Proceso de carga de datos</b>	<b>35</b>
5.1	Introducción	35
5.2	Funciones de carga	36
5.3	Conclusiones	43
<b>6.</b>	<b>La interfaz web</b>	<b>45</b>
6.1	Introducción	45
6.2	Módulo Maps	46
6.3	Funcionalidad DT	48
6.3.1	La pestaña de fecha	50
6.3.2	KPI	51
6.3.3	Events	54
6.3.4	Selección de filtros	55
6.3.5	Área de notificación	56
6.3.6	Load Layer	57
6.3.7	Import KML	58
6.4	Ventana de visualización	58
6.5	Conclusiones	60
<b>7.</b>	<b>Integración, pruebas y resultados</b>	<b>61</b>
7.1	Introducción	61
7.2	Conclusiones	63

<b>8.</b>	<b>Conclusiones y trabajo futuro</b>	<b>65</b>
8.1	Introducción	65
8.2	Conclusiones	65
8.3	Trabajo futuro	66
<b>9.</b>	<b>Manual del programador</b>	<b>69</b>
9.1	Condiciones de desarrollo	69
9.1.1	Recursos software	69
9.1.2	Recursos hardware	70
9.1.3	Requisitos de permisos y conectividad	70
<b>A -</b>	<b>Bibliografía</b>	<b>71</b>
<b>B -</b>	<b>Presupuesto</b>	<b>73</b>
<b>C -</b>	<b>Pliego de condiciones</b>	<b>75</b>



# Índice de figuras

1.1	Drive test	6
2.1	Base de datos existente	15
3.1	Ampliación esquema base de datos	19
3.2	Tabla dt_operators	21
4.1	Aplicación G-NetTrack	26
4.2	G-NetTrack en funcionamiento	27
4.3	Archivos obtenidos de una ejecución	29
4.4	Archivo de medidas	31
4.5	Archivo de eventos	33
6.1	Módulo Maps	46
6.2	Representación de celdas	47
6.3	Tipos de estudio anteriores	48
6.4	Nuevo tipo de estudio	48
6.5	DT	49
6.6	Pestaña de fecha	50
6.7	Datapicker	50
6.8	Selección de KPI	51
6.9	Tabla dt_colors	52
6.10	Tabla levels	53
6.11	Lista de eventos	54
6.12	Selección de eventos	54
6.13	Selección de filtros	55
6.14	Área de notificación	56
6.15	Simulación del proceso 'Load Layer'	57
6.16	Ejemplo capa Drive Test	59
6.17	Tooltip	60



# Glosario

KPI Key Performance Indicator  
DT Drive Test  
SQL Structured Query Language  
KML Keyhole Markup Language  
XML, eXtensible Markup Language  
GIS Geographic Information System  
BSC Base Station Controller  
RNC Radio Network Controller  
LAC Location Area Code  
GSM Global System for Mobile communications  
UMTS Universal Mobile Telecommunications System  
LTE Long-Term Evolution  
SNR Signal-to-noise ratio  
CQI Channel Quality Indication  
PSC Primary Scrambling Code  
GPRS General Packet Radio Service  
EDGE Enhanced Data Rates for GSM Evolution  
HSPA High-Speed Packet Access  
HSPA+ Evolved High-Speed Packet Access  
HSDPA High Speed Downlink Packet Access  
GPS Global Positioning System

MCC Mobile Country Code  
MNC Mobile Network Code  
Cell ID Cell Identifier  
RSCP Received Signal Code Power  
RSSI Received Signal Strength Indicator  
RSRP Reference Signal Received Power  
RSRQ Reference Signal Received Quality  
DL Download  
UL Upload  
RxQual Reception Quality  
RxLev Reception Level  
Rx Reception  
I Idle  
V Voice  
VD Voice-Data  
PINGAVG Ping Average  
PINGMIN Ping Minimum  
PINGMAX Ping Maximum  
PINGSTDEV Ping Standard Deviation  
SSID Service Set Identifier  
IRAT Inter-Radio Access Technology  
ID Identifier  
PLMN Public Land Mobile Network

# 1

## Introducción

### **1.1 Marco contextual**

En una sociedad en la que el uso de dispositivos móviles toma posiciones frente a los ordenadores, las empresas ven en los *smartphones* un camino esencial en el que conseguir presencia y a través del cual ofrecer sus servicios. Son muchos los estudios que muestran un cambio en los hábitos del consumidor que cada vez consulta información a través de su dispositivo móvil con más frecuencia.

El aumento de usuarios de la red móvil implica también un aumento de la necesidad de la medición de la calidad de esta red y conlleva una proliferación de herramientas de medición o testeo.

Fue el incremento del uso de dispositivos móviles en nuestro entorno el mayor empuje a la hora de considerar los terminales móviles como la fuente de los datos de medición de calidad de red móvil utilizada en este proyecto, mediante técnicas como por ejemplo los drive tests.

El objetivo del presente proyecto ha consistido en desarrollar una interfaz que transforma la información obtenida con estos dispositivos en un formato que facilita la visualización geográfica de estos datos y que los representa dinámicamente de una manera útil para monitorización de red a nivel geográfico, siendo los operadores móviles los potenciales usuarios del sistema.

## **1.2 Motivación**

---

La necesidad de pruebas de calidad en redes móviles es mayor que nunca. Surgen normas y requisitos nuevos, la demanda de ancho de banda por parte de los usuarios aumenta, hay que aceptar más aplicaciones y más clientes exigen más calidad. Se deben llevar a cabo multitud de pruebas rigurosas en cada etapa del ciclo de vida de la red, directamente a través de la red activa.

Los proveedores de servicios de red, como la compañía en la que desarrollé este sistema de visualización que da soporte a varios operadores móviles, deben verificar el cumplimiento de las especificaciones técnicas, la funcionalidad, capacidad, calidad de la experiencia y calidad del servicio.

Las pruebas pasivas implican la captura y el análisis del tráfico o medidas en diversas interfaces de muchos elementos de las redes móviles. El tipo de análisis realizado en este proyecto es una prueba pasiva más, nacida de la necesidad de ver con claridad el estado del tráfico de la red y corregir los posibles errores con rapidez.

Este proyecto complementó una aplicación web ya creada, utilizada por los ingenieros de la empresa en la que estuve trabajando 12 meses como becario. Una aplicación que facilitaba la visualización de los KPIs (*Key Performance Indicator*) de la red.

El tipo de herramientas usado en la obtención de datos para el funcionamiento de esta aplicación ya existente, sondas pasivas/activas, suponía un elevado coste, por lo que se buscaron diferentes tipos de fuentes. Una de ellas implicaba el uso de aplicaciones móviles cuyas principales ventajas frente al resto de herramientas eran el precio y el alcance a nivel de usuario.

Es por ello por lo que se decide extender esta aplicación web ya creada y añadir un nuevo módulo en el que los datos analizados provienen de aplicaciones móviles, que cualquier usuario de la red móvil puede utilizar y cuyo uso en algunos casos es gratuito.

La motivación a la hora de realizar un sistema de medición de calidad es:

- Optimizar la calidad de red.
- Supervisión de la red.
- Análisis de las causas de los problemas (*Troubleshooting*).
- Análisis del impacto de incidencias.

- Tendencias de tráfico de red y evaluación de comportamiento de usuarios.
- Evaluación de experiencia de usuarios.
- Pruebas de regresión en red.

## **1.3 Estado del arte**

---

### **1.3.1 Introducción**

En este apartado se hablará de los tipos de medición y se introducirá al estado del arte actual. De esta manera se podrá tener una visión global del estado de las mediciones de calidad de experiencia existentes. También se hará una pequeña introducción a la aplicación web que este proyecto complementa.

### **1.3.2 Sistemas de medición radio**

Hay múltiples formas distintas de determinar la calidad de una red móvil, dependiendo de varios factores como pueden ser: el tipo de terminal de captura, si las medidas se toman en estático o en dinámico, el tipo de pruebas realizadas (activas o pasivas) o el objetivo específico que busca cada tipo de medición.

A continuación se hará un listado genérico de los tipos de sistemas y se especificará el modelo del método de medición empleado en este proyecto.

#### **Tipo de terminal de captura**

- Terminal de ingeniería: herramientas de testeo específicamente desarrolladas para la medición de los parámetros de rendimiento y calidad de las redes inalámbricas. Estos terminales representan un coste muy elevado pero su principal ventaja es que tienen acceso a la capa de señalización (capa 3).

- Agentes móviles de medida: aplicaciones para móviles diseñadas para la prueba y monitoreo de la calidad en redes 2G, 3G o 4G. En este caso, la información se obtiene a través de los sistemas operativos de los *smartphones* (API en el caso de Android). El coste de las pruebas se abarata enormemente lo cual supone una ventaja respecto al caso anterior pero este tipo de herramienta no tiene acceso a la capa de señalización.

### **Tipo de medidas radio**

- Medidas en estático: por ejemplo las realizadas por las sondas pasivas/activas, en las que no se precisa un cambio constante de ubicación del material de medida.
- Medidas en dinámico: este tipo de medidas requieren un vehículo móvil dotado con equipos de medición que pueden detectar y registrar una amplia variedad de parámetros del servicio móvil en una zona geográfica determinada. A las medidas en dinámico se las denomina comúnmente ‘drive tests’, descritos en detalle más adelante.

### **Tipo de medidas según realización**

- Activas: el terminal realiza tests con tráfico no real. Esto implica la inyección de tráfico de prueba en la red y el control de la circulación de este tráfico.
- Pasivas: los terminales se limitan a recoger el tráfico que circula por la red. Es más un proceso de observación, en lugar de la inyección de tráfico virtual en la red, el monitoreo pasivo se ciñe a analizar el tráfico que ya está en la red.

### **Tipo de medidas según objetivo**

- *Troubleshooting*: consiste en encontrar la fuente del posible problema mediante la constante puesta a prueba del sistema para que el fallo pueda ser resuelto.
- *Benchmarking*: está basado en comparar un servicio propio con otros que pertenezcan a otras entidades que presenten los resultados más óptimos sobre el área de interés, con el propósito de transferir el conocimiento de las mejores prácticas.
- Monitorización/supervisión de red: supone el control, análisis y estudio de la red móvil buscando componentes lentos o fallidos.

### **1.3.3 Aplicaciones móviles de medida**

La aplicación web preexistente (de la que se habla con detalle más adelante) utilizaba los datos obtenidos mediante sondas activas y pasivas, sondas que disponen de mecanismos para la obtención de métricas de calidad de red. Estas métricas se construyen a través de la información de señalización, datos de referencia de los elementos de red y medidas de los niveles de señal electromagnética. El uso de estas sondas y escáneres supone un gasto económico elevado.

Una de las ventajas del uso de aplicaciones es la reducción de costes porque pueden instalarse en cualquier *smartphone* (no hacen falta terminales especiales de ingeniería). Por la misma razón, existe la posibilidad de llegar a un espacio muestral mucho mayor (cualquier móvil es un potencial aparato de medida). Un inconveniente del uso de aplicaciones es que solamente se pueden extraer las métricas de red que proporciona el sistema operativo y eso limita la precisión de las medidas y los análisis que se pueden realizar.

Para la obtención de los datos que se procesarán en el nuevo módulo de visualización de drive test se ha utilizado la aplicación para Android 'G-NetTrack', una herramienta para la monitorización de varios parámetros de redes de telefonía móvil como GSM, UMTS y LTE que permite el seguimiento y registro de estos valores de red móvil sin necesidad de utilizar equipos especializados.

Esta aplicación se describirá con más detalle en el capítulo 4, aquí podemos ver un resumen de los parámetros más importantes que proporciona:

- Cell ID, el identificador de la celda que está prestando el servicio.
- Level, la intensidad de señal.
- QUAL, la calidad de la señal de red.
- SNR, ratio señal-ruido.
- CQI, indicador de la calidad del canal, medido sólo en LTE (4G).

### 1.3.4 Drive test

Drive test (“recorrido de prueba”) es un método de medición y evaluación de la cobertura, capacidad y calidad de servicio de una red móvil. La técnica consiste en el uso de un vehículo de motor dotado de equipos de medición de radio móvil que pueden detectar y registrar una amplia variedad de los parámetros físicos y virtuales de servicio de telefonía móvil celular en una zona geográfica determinada.

Los principales objetivos son: simular lo que un abonado de la red inalámbrica podría experimentar en un área específica, evaluación comparativa de red, optimización y solución de problemas y monitoreo de la calidad de servicio.



Figura 1.1: Drive test

El análisis de un drive test aporta una ayuda fundamental para el trabajo de cualquier profesional en el campo de las redes móviles que comprende dos fases: la obtención de los datos y el análisis de estos.

Aunque a través del análisis de KPIs se pueden identificar problemas tales como la interrupción de llamadas, los drive tests permiten un análisis más profundo en el campo y la identificación de las áreas de cobertura de cada sector, la interferencia, la evaluación de cambios en la red y otros parámetros.

La información obtenida en los drive tests se puede utilizar para ayudar en la búsqueda de problemas específicos en las fases de despliegue de nuevas redes o para reproducir los fallos denunciados por los consumidores durante la fase operativa de la red. De este modo, los datos de estas pruebas se utilizan para diagnosticar la causa de los fallos localizados de red, como las llamadas caídas, errores de traspaso entre celdas vecinas o fallos en la definición de vecindades entre celdas. Mediante este tipo de mediciones los operadores móviles pueden realizar cambios en sus redes para proporcionar una mejor cobertura y servicio a sus clientes.

Ejemplos de compañías que comercializan aplicaciones especializadas en drive testing serían:

- Ascom (TEMS).
- SwissQual.
- Nemo.

## **1.4 Aplicación web original**

Antes de este proyecto, existía una aplicación web original, un sistema de visualización gráfica de la evolución de los KPIs de toda la red del operador móvil, que se explicará con más detalle más adelante, pero cuya existencia de momento es necesario conocer para una mejor comprensión de la lectura.

Este proyecto es una extensión de esta aplicación web e incorpora las métricas procedentes de estos terminales móviles de medida.

Se decidió implementar una aplicación web en lugar de una aplicación instalable o de escritorio debido al requisito de los usuarios de acceder desde diferentes ubicaciones al mismo sistema, a la necesidad de recopilar todas las medidas de todos los usuarios en un sistema central, a que la información que manejamos puede ser accesible desde cualquier ordenador con conexión a Internet y que el sistema preexistente ya estaba definido de esta forma, siendo este proyecto un módulo adicional.

## **1.5 Objetivos del proyecto**

---

Los objetivos específicos del proyecto se resumen en los siguientes puntos:

1. Diseño de la base de datos y adaptación a la ya existente y utilizada por la aplicación web previamente descrita.
2. Obtención de los datos de calidad de la red móvil mediante el uso de una aplicación móvil de medida en sesiones de drive tests.
3. Carga de los datos de los ficheros obtenidos en los drive tests, procesado y almacenamiento en una base de datos. Implementación de un proceso de carga mediante código desarrollado en lenguaje Ruby on Rails.
4. Diseño, creación e integración de un nuevo módulo en la interfaz web que permita al usuario visualizar la información obtenida en los drive tests y permitir filtrarla por los parámetros correspondientes, ya sea por fechas, celdas específicas, regiones, etc.
5. Generación del código necesario que haga las consultas pertinentes a la base de datos y pueda devolver a la interfaz la información solicitada por el usuario.

## **1.6 Organización de la memoria**

---

En el capítulo 2, a continuación, se hablará de la aplicación web ya creada sobre la que se basa este proyecto.

En el capítulo 3 se realizará un análisis de la base de datos utilizada por el nuevo módulo de la aplicación web, así como la forma de obtener los datos que luego almacenará y sus principales tablas.

En el capítulo 4 se explica el proceso de obtención de datos, así como el uso del agente móvil y los tipos de archivos con los que se trabaja.

En el capítulo 5 se analiza el proceso de carga de datos y todas las funciones necesarias durante el mismo.

El capítulo 6 muestra la interfaz web y sus distintos elementos.

En el capítulo 7 se trata el proceso de integración del nuevo proyecto con el anterior sistema y las pruebas del nuevo módulo.

En el capítulo 8 se mostrarán las conclusiones del trabajo desarrollado así como las aportaciones futuras que podrán ser añadidas.

El capítulo 9 lista las condiciones software y hardware del sistema de desarrollo.



# 2

## Aplicación web existente

### **2.1 Introducción**

---

Este capítulo tratará de la aplicación web existente que complementa este proyecto. Una aplicación que da asistencia a varios departamentos dentro de un operador de telefonía móvil y que permite la visualización de los KPIs globales de toda la red, es decir, un sistema de visualización de datos de calidad en redes móviles.

Como se explica en el siguiente punto, el presente proyecto es una extensión de esta aplicación, cuyo diseño y desarrollo son afines a la filosofía del sistema original.

### **2.2 Arquitectura de la aplicación**

---

La aplicación consta de varios módulos, cada uno con su funcionalidad. El presente proyecto es un módulo más de esta aplicación. Un resumen del resto de módulos sería:

- Módulo de autenticación de usuarios. Es el módulo encargado de la autenticación de los clientes mediante la petición de un nombre de usuario y contraseña.

- Módulo de permisos y autorización de recursos. En este módulo se definen los permisos de los distintos usuarios que acceden a la aplicación.
- Módulo de incidencias. Módulo creado para poder acceder al registro de las incidencias de la red.
- Módulo de informes. Módulo para almacenar los informes internos del operador móvil relacionados con los KPIs de la red. Los usuarios tienen la posibilidad de visualizar y recuperar estos informes.
- Módulo del panel de mando. La idea de este módulo es representar los valores actuales de los KPIs más importantes de la red e informar al usuario acerca de la evolución de estos KPIs en el tiempo.
- Módulo de visualización de KPIs globales de la red. Este módulo sirve para enseñar al usuario un histórico semanal o mensual de la evolución de los KPIs globales de toda la red del operador móvil.
- Módulo de visualización de las peores celdas de la red. Este módulo se utiliza para mostrar al usuario un resumen con las peores celdas de la red.
- Módulo de administración (Admin). Este módulo de administración permite a los usuarios administradores el uso de varias acciones no disponibles para el resto de usuarios.

Y por último el módulo sobre el que se han hecho los desarrollos contemplados en este proyecto:

### **2.2.1 Módulo de visualización de KPIs geolocalizados (Maps)**

El último módulo de visualización del sistema sirve para visualizar KPIs sobre una componente cartográfica, es decir, se trata de un sistema de información geográfica o GIS.

Un GIS es un conjunto de herramientas que integra y relaciona diversos componentes que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

Este módulo ha sido creado para trabajar en uno de los cuatro siguientes modos de funcionamiento llamados “tipos de estudio”: tipo de estudio por celda, por nodo, por área y por drive test, donde cada uno de los modos presenta los KPIs sobre la componente cartográfica con un diferente nivel de agregación.

En este módulo se pueden visualizar las celdas, nodos o áreas filtradas en función de las diferentes dimensiones de la red (BSCs, RNCs, LACs o provincias).

El tipo de estudio por drive test es el que trata este proyecto. Mientras que el resto de módulos de la aplicación tienen otras fuentes de información (sondas activas, sondas pasivas y contadores de red). El nuevo módulo tiene su fuente en los drive tests realizados con una aplicación instalada en un terminal móvil.

## **2.3 Tecnologías empleadas durante el desarrollo del sistema**

### **2.3.1 Ruby on Rails**

El sistema preexistente fue escrito en este *framework*. Se escogió Ruby on Rails para poner en funcionamiento el servidor de la aplicación por la rapidez con la que permite desarrollar y adaptar nuevas funcionalidades. Ruby on Rails busca juntar la posibilidad de desarrollar aplicaciones con la simplicidad, teniendo que escribir menos código que con otros *frameworks* y precisando un mínimo de configuración.

### **2.3.2 MySQL**

MySQL es un sistema de gestión de bases de datos relacional, multiusuario y multihilo. Se desarrolla como software libre en un esquema de licenciamiento dual, por lo que, para evitar costes adicionales con las licencias, se eligió MySQL como sistema de gestión de bases de datos.

### **2.3.3 Google Earth y ficheros KML**

KML (Keyhole Markup Language) es un lenguaje basado en XML usado para la representación de datos geográficos en tres dimensiones. Fue desarrollado para ser manejado con Google Earth, cuyo plugin de navegador es el utilizado en el presente proyecto para la visualización de datos sobre una componente cartográfica.

### **2.3.4 Git**

Git es un software de control de versiones diseñado para garantizar la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Al tratarse este proyecto de un módulo adicional de una aplicación mayor, el número de personas programando no se limitaba solo a una. Eran varios los programadores que trabajaban al mismo tiempo sobre el código fuente, a veces incluso en la misma parte del código, así que Git jugaba un papel importante a la hora de la compatibilización y desarrollo final del código.

### **2.3.5 Rails Admin**

Rails Admin es un *gem* de Rails que proporciona una interfaz fácil de usar para la gestión de datos. Se decidió usar Gem Rails Admin como módulo de administración, tanto en su día para el desarrollo original, como posteriormente para las modificaciones contempladas en este proyecto.

Rails Admin permite la creación, modificación y eliminación de instancias de Modelos, estas representan registros de las tablas de la base de datos.

## **2.4 Base de datos existente**

---

El primer paso a la hora de empezar este proyecto fue el análisis de la base de datos existente y el diseño de las modificaciones pertinentes para la nueva funcionalidad. Como ya se ha dicho, este proyecto consiste en la ampliación de una aplicación web ya creada y funcional. Esta aplicación cuenta con una base de datos cuyo esquema está incluido en la figura adjunta de la siguiente página.



## **2.5 Conclusiones**

---

En este capítulo se ha hablado de la aplicación web ya creada sobre la cual se ha trabajado. De esta manera se tiene un conocimiento previo del tipo de aplicación web sobre el que se ha desarrollado este proyecto. Se han visto los tipos de tecnología que se han utilizado para el desarrollo del proyecto, tanto para la visualización de los datos como para su procesamiento. También se ha visto el diseño de la base de datos de este sistema, una base de datos cuyo esquema será ampliado para la integración de los nuevos datos usados en este proyecto.

En el siguiente capítulo veremos cómo estos datos son almacenados en la base de datos y se tratará sobre la estructura y las tablas principales de esta.

# 3

## Diseño de la base de datos

### **3.1 Introducción a la base de datos**

La base de datos es el integrante central del sistema de visualización. Almacena la información necesaria para la visualización de los diferentes tipos de KPIs. Contiene también información acerca de los usuarios registrados en el sistema e información relacionada con la configuración de las diferentes partes del sistema.

Toda la información recogida en los drive tests es tratada por medio de un proceso de carga implementado con el lenguaje de programación Ruby y almacenada en una base de datos MySQL.

Como se ha explicado en el punto 2.4, uno de los primeros pasos fue extender la base de datos del sistema existente (ver figura 2.1). De todas las tablas originales se usarán dos de ellas en este proyecto, que estarán relacionadas con las tablas que se diseñan y usan en el nuevo módulo. Estas dos tablas existentes son:

## **3.2 Cells**

---

En la tabla de celdas se guarda una lista de celdas disponibles en la red. Se guardan los nombres de las celdas y características de estas como la altura de la antena que cubre el área de la celda, la dirección a la que apunta la antena o la inclinación de ésta.

## **3.3 Lacs**

---

Cada estación base se encuentra dentro de un código de área de ubicación o LAC (Location Area Code).

La tabla de LACs guarda los posibles valores de LAC definidos en la red.

La nueva base de datos que almacenará la información obtenida en los drive tests tiene este esquema:

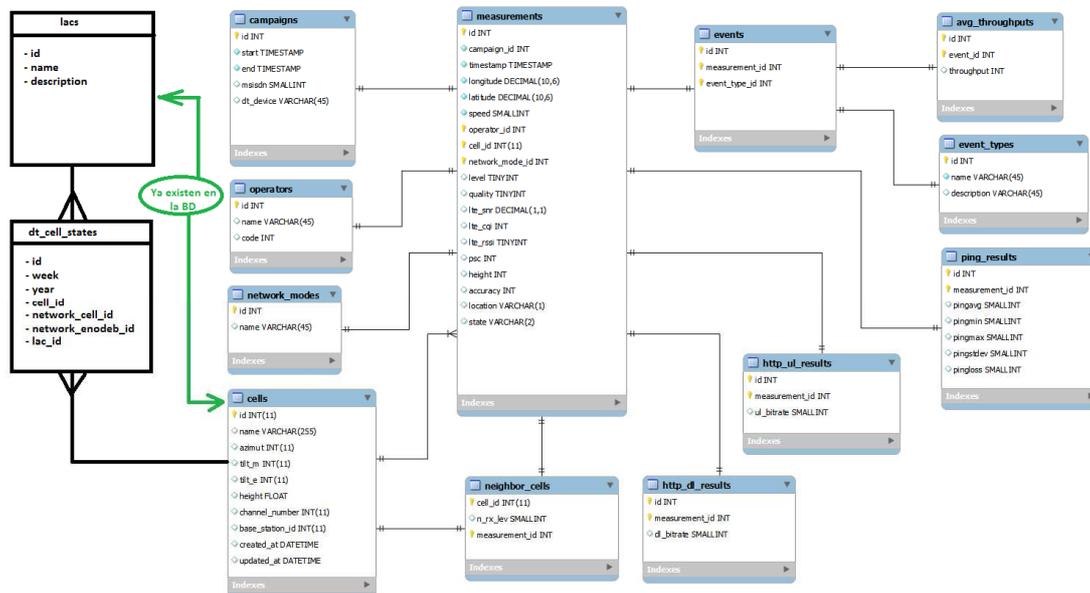


Figura 3.1: Ampliación esquema base de datos

Como puede verse, aparecen las tablas cells y lacs, esto es debido a que las muestras recogidas por la aplicación móvil de medida contienen campos que están relacionados con estas tablas.

### **3.4 Las principales tablas**

---

Las tablas más importantes de la ampliación de la base de datos se describen a continuación.

- dt\_measurements es la tabla, junto a dt\_campaigns, más importante de la base de datos. Es la tabla que guarda los valores de las medidas, sus campos son:

id: identificador, número entero que es único para cada medida e identifica unívocamente a estas. Todas las tablas contienen un identificador de este tipo, así que obviaremos este campo en adelante.

campaign\_id: es el identificador de campaña. Las medidas corresponden a campañas, una campaña está delimitada desde que se empieza a tomar medidas hasta que se detiene. En el punto 3.5 se define el concepto de campaña con más detalle y en el capítulo 4 se explica cómo realizar una campaña.

timestamp: es la fecha y la hora de cada una de las medidas en este formato: 2014.02.13\_08.05.54.

El resto son los valores de cada medida: longitud, latitud, velocidad, identificador de operador, celda y tecnología, nivel de señal, calidad, SNR (Signal to Noise Ratio), relación señal-ruido, CQI (Channel Quality Indicador) indicador de calidad del canal, RSSI (Received Signal Strength Indicador) indicador de intensidad de señal recibida de la energía total de la banda ancha en todo el canal (estos 3 últimos valores se miden sólo en LTE), PSC (Primary Scrambling Code), código primario de aleatorización, sólo para 3G, la altura del nivel del suelo, la precisión con la que se determina la ubicación y el estado actual del teléfono. Como ya se ha dicho, se hablará con más detalle de todos estos campos en el capítulo 4.

- dt\_campaigns: es la tabla que almacena las campañas que se han procesado, contiene datos sobre estas, tales como la fecha y hora del inicio y fin de campaña, que corresponden con la primera y última medidas de esta.

- dt\_operators: como los operadores de telefonía móvil siempre van a ser los mismos, hablando siempre de usuarios en el territorio español, se decide crear una tabla que contenga los operadores más comunes. Ver figura adjunta:

🔑 id	📞 name	📞 code
1	Movistar	21407
2	Yoigo	21404
3	Vodafone	21401

Figura 3.2: Tabla dt\_operators

Si el proceso de carga detecta un operador nuevo, éste es almacenado automáticamente en esta tabla.

Las medidas no guardarán el nombre y el código del operador, sino el identificador correspondiente a su operador, con lo que se consigue optimizar el espacio de almacenamiento.

- dt\_network\_modes: de forma parecida a la tabla de operadores se hace una con las tecnologías de red, GPRS, EDGE (2G), HSPA, HSPA+, HSDPA (3G), LTE (4G).

- lacs y cells, definidas anteriormente.

- dt\_neighbor\_cells: es la tabla con la información de las celdas vecinas respecto a la celda que da servicio en el momento de la medida. Se guarda el ID de la celda, su nivel de señal y el ID de la medida relacionada.

- dt\_events: como se explica más adelante, una medida puede tener un evento relacionado (test de subida/descarga, pérdida de señal, etc), estos eventos se guardan en esta tabla que contiene el ID de la medida y del tipo de evento que tiene una tabla aparte.

- dt\_event\_types: contiene el nombre del evento y una pequeña descripción.

Las tablas dt\_avg\_throughputs, dt\_ping\_results, dt\_http\_ul\_results y dl\_results almacenan los valores de las pruebas de descarga de datos que la aplicación móvil de medida permite hacer en medio de una campaña.

### **3.5 El proceso de carga**

---

La aplicación móvil de medida genera archivos en formato ‘.txt’ con la información en bruto recopilada de la monitorización de la red móvil a la que estuvo conectado el terminal durante el drive test. Estos archivos contienen la información de las campañas.

Se denomina campaña o log a todo el proceso de obtención de medidas, desde que se comienza a medir (Log Start) hasta que se finaliza (Log End). Las campañas están organizadas en filas y columnas, son tablas cuyo índice principal es el ‘timestamp’, la fecha y hora de la medida. Las columnas corresponden con los valores de las medidas, más adelante se mostrará con más detalle todos estos valores.

Una vez acaba una campaña, se genera un fichero ‘.txt’ en la memoria interna del terminal móvil. Este fichero con la información en bruto es muy poco intuitivo para el usuario, por lo que se precisa de un procesamiento.

El proceso de carga es el encargado de leer estos ficheros y almacenar en la base de datos los campos de las medidas que cumplan unas condiciones predefinidas, por ejemplo, que tengan una latitud y longitud correctas.

### **3.6 Migraciones**

---

Todas estas tablas junto con sus claves foráneas son creadas mediante migraciones en Ruby on Rails. Las migraciones nos permiten modificar nuestra base de datos de manera organizada y estructurada; además de reducir casi completamente el uso de SQL. También nos permiten manejar versiones del estado de la base de datos.

Un ejemplo de migración sería:

```
class CreateDtOperators < ActiveRecord::Migration
  def up
    create_table :dt_operators do |t|
      t.string :name,      :limit => 45
      t.integer :code,     :limit => 11
    end

    add_index :dt_operators, :name, :unique => true
    add_index :dt_operators, :code, :unique => true
  end

  def down
    drop_table :dt_operators
  end
end
```

Mediante la consola de Ruby on Rails y el uso del comando: rake db:migrate, se ejecutarán todas las migraciones de este tipo en las que primero se define la creación de la tabla y después su eliminación, en el caso de usar el comando contrario: rake db:rollback.

### **3.7 Conclusiones**

---

En este capítulo se ha hecho una introducción a la base de datos y se ha entrado en detalle en su estructura y en sus principales tablas.

En el siguiente capítulo se hablará sobre cómo se obtienen estos datos y los tipos de archivos que se manejan.



# 4

## Obtención de los datos

### **4.1 Introducción**

---

La idea de este proyecto es conseguir un estudio global de la red móvil al menor coste posible, y estos son los aspectos que se tuvieron en cuenta:

- Cada *smartphone* es considerado como una potencial fuente de información.
- Percepción real de usuario (conseguir reproducir lo que los usuarios experimentan en su día a día).
- Información de red geolocalizada.
- Con un coste mínimo, varios usuarios “activos” capturarían información de buena parte de la red.

Por todo ello, de entre todas las aplicaciones similares en el mercado, se decidió usar G-NetTrack. Se optó por ella porque es gratuita y los parámetros recogidos eran bastante amplios. Como ya se ha explicado, todo este proyecto gira en torno a la información obtenida con esta aplicación móvil. Cuyas características son:

- “User friendly” app. Uso sencillo e intuitivo.
- App gratuita orientada a drive test. Disponible en “Google play”.
- Soporta 2G, 3G y 4G.
- Información de GPS (coordenadas, altitud, velocidad).
- Información de red (MCC, MNC, Cell ID, Node, LAC, etc).
- Parámetros radio (nivel de señal, nivel de calidad, SINR/CQI, DL-bitrate, UL-bitrate).
- Eventos (establecimientos, caídas, handover, etc).
- Configuración de secuencias de pruebas de voz y datos.
- Ofrece KMLs de los distintos KPIs medidos.
- Log de medidas en ficheros ‘.txt’.

## 4.2 Uso de G-NetTrack

El uso de esta aplicación es muy sencillo. Cualquier usuario de Android con una versión 2.2 o superior puede descargarla y empezar a obtener información de la red móvil de la que está haciendo uso.

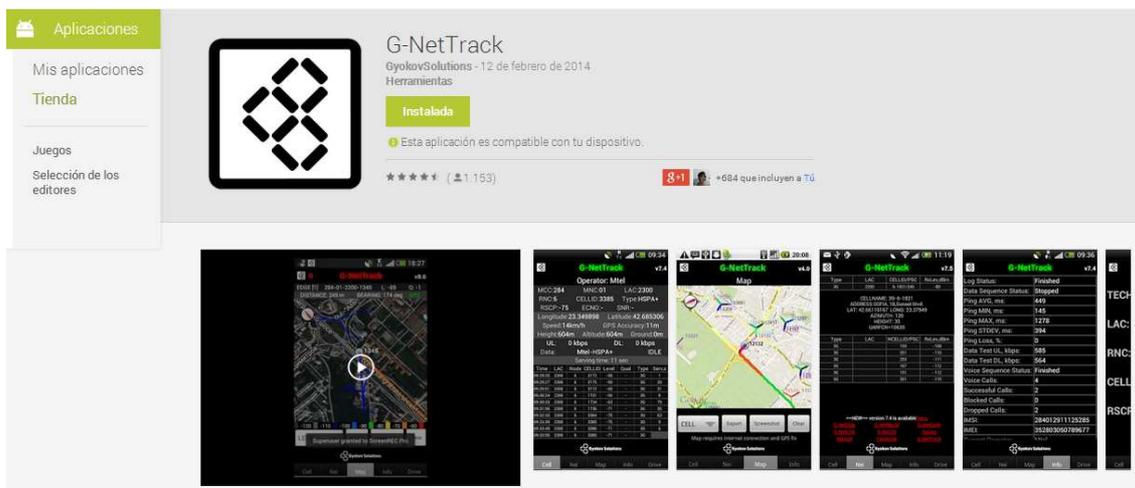


Figura 4.1: Aplicación G-NetTrack

En la figura adjunta se puede ver la página de Google play desde la cual descargarse esta aplicación en su última versión v9.0:

<https://play.google.com/store/apps/details?id=com.gyokovsolutions.gnettrack>

Una vez instalada la aplicación y desde el momento en que la abrimos, ya podemos ver en la pantalla principal información de nuestra red, tanto si estamos conectados a una red móvil o WiFi.



Figura 4.2: G-NetTrack en funcionamiento

Los datos obtenidos dependerán de la marca y modelo del terminal móvil que usemos ya que G-NetTrack obtiene más o menos tipos de datos en función de la compatibilidad con el terminal y de las capacidades de este.

La lista de compatibilidad está disponible en la página web del fabricante:

<http://www.gyokovsolutions.com/survey/surveyresults.php>

G-NetTrack dispone de varias opciones, pero la que nos interesa para la realización de los drive tests es la opción de Start Log.

Al iniciar el Log o campaña la aplicación crea automáticamente un directorio en la carpeta "G-NetTrack\_Logs" en el almacenamiento principal del teléfono con un nombre de este estilo: NOMBRE\_OPERADOR\_año.mes.día\_hora.minuto.segundo (por ejemplo: VODAFONE\_2014.04.08\_12.00.00), que contiene el nombre del operador móvil al cual se está conectando al comenzar la campaña y se crean unos archivos de texto y KMLs.

Los KMLs no los usaremos, nos centraremos en los archivos de texto. Estos archivos se irán actualizando y creciendo a medida que la campaña siga estando activa, hasta que se detenga con la opción End Log.

La propia naturaleza de los drive tests implica estar en movimiento en el momento en que se obtienen las medidas, por lo que G-NetTrack permite ajustar la toma de medidas al usuario, tanto en tiempo como en espacio, se puede configurar cada cuántos segundos y/o metros se quiere obtener una muestra nueva. Como en nuestro caso precisamos de la mayor cantidad de información posible lo hemos establecido a 1 medida / segundo. Una vez se haya realizado la ruta establecida y se finaliza el Log o campaña, la aplicación acaba de editar todos los archivos de texto que se habían creado al iniciar el drive test. Si pasamos estos archivos a un ordenador la carpeta creada tendría un aspecto similar al de la figura adjunta:

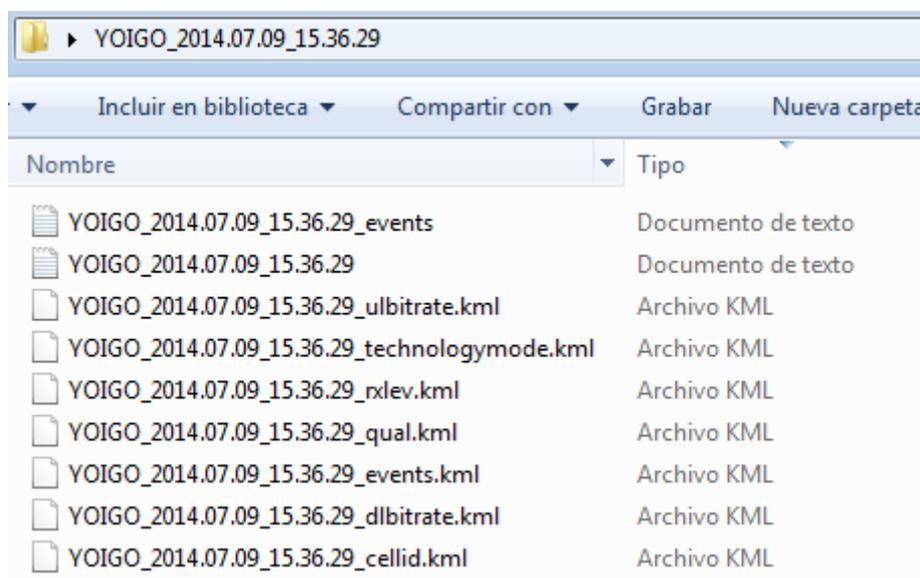


Figura 4.3: Archivos obtenidos de una ejecución

Donde se puede ver que se han creado varios archivos KML, que obviaremos, y dos archivos de texto: el de medidas y el de eventos.

### **4.3 Archivo de medidas**

El archivo con las medidas contiene información en forma de tabla (filas y columnas) cuyas columnas son los campos de las medidas y cada una de las filas es una medida, excepto la primera, que contiene los nombres de los campos, que son:

- . Timestamp: año.mes.día\_hora.minuto.segundo
- . Longitude: longitud.
- . Latitude: latitud.

- . Speed: en km/h.
- . Operatorname: nombre del operador.
- . Operator: código del operador (Movistar: 21407, Vodafone: 21401, Orange: 21403, Yoigo: 21404).
- . CGI: Common Gateway Interface, (Operator + LAC + Cell ID).
- . Cellname: nombre de la celda.
- . Node: nodo de la celda.
- . CellID: el ID de la celda actual.
- . LAC: Location Area Code de la celda.
- . NetworkTech: tecnología - 2G, 3G o 4G.
- . NetworkMode: GPRS, EDGE (2G), HSPA, HSPA+, HSDPA (3G), LTE (4G).
- . Level: la intensidad de la señal actual en dBm. Para las distintas tecnologías la leyenda es diferente - RXLEV en 2G, RSCP en 3G y RSRP en 4G.
- . Qual: la calidad de la señal de la red. Para las distintas tecnologías la leyenda es diferente - RXQUAL en 2G, Ecno en 3G y RSRQ en 4G.
- . SNR: Signal to Noise Ratio, relación señal-ruido. Se mide sólo en LTE.
- . CQI: Channel Quality Indicator, indicador de calidad del canal. Se mide sólo en LTE.
- . RSSI: Received Signal Strength Indicator, indicador de intensidad de señal recibida de la energía total de la banda ancha en todo el canal. Se mide sólo en LTE.
- . DL\_bitrate: tasa de bits de transferencia de datos de bajada.
- . UL\_bitrate: tasa de bits de transferencia de datos de subida.
- . PSC: Primary Scrambling Code, código primario de aleatorización, sólo para 3G.
- . Altitude: la altitud medida del GPS.
- . Height: la altura del nivel del suelo, es ajustable en la configuración.
- . Accuracy: la precisión con la que se determina la ubicación. La ubicación puede estar basada en GPS o red (usando el servicio de localización de Google - usando las celdas vecinas o redes WiFi).
- . Location: N (Network) o G (GPS), de dónde se obtiene la localización.
- . NLAC1-2..., NCellid1-2..., NRxLev1-2... (Celdas vecinas): LAC, CellID y RxLevel, identificador de celda, y nivel de señal de las 6 celdas vecinas.

- . State: estado actual del teléfono, I (Idle, inactivo), V (Voice, llamada de voz activa), D (Data, transferencia de datos activa) o VD (Voz y Datos simultáneos).
- . PINGAVG, PINGMIN, PINGMAX, PINGSTDEV, PINGLOSS, TESTDOWNLINK, TESTUPLINK: al iniciar una secuencia de datos se muestran los datos sobre el PING y al acabar los bitrates medios del DL y UL.
- . DataConnection\_Type: M de Mobile (2G, 3G o 4G) o WiFi.
- . DataConnection\_Info: información sobre la conexión de datos, SSID del WiFi o número del tipo de conexión de datos, 2 para EDGE, 8 para HSDPA, 10 para HSPA, 13 para LTE, 15 para HSPA+, etc.
- . Layer: 3 capas (1, 2, 3) una por tecnología (2G/3G/4G), permite crear capas por tecnologías y mostrar distintos colores por cada capa.
- . Filemark: para añadir comentarios mientras se toman medidas.

Un ejemplo de un archivo de medidas obtenido en una ejecución de G-NetTrack es:

Timestamp	Longitude	Latitude	Speed	Operatorname	Operator	CGI	Cellname
2015.05.25_20.29.33	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.34	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.34	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.35	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.36	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.37	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.38	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.40	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.41	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.43	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.44	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.45	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.46	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.48	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.49	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.50	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.51	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.52	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.53	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.55	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.56	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.57	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.29.59	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.30.00	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.30.01	-3.713041	40.479188	0	movistar	21407	214070286239872	
2015.05.25_20.30.02	-3.713041	40.479188	0	movistar	21407	214070286239872	

Figura 4.4: Archivo de medidas

## **4.4 Archivo de eventos**

---

La creación de una campaña genera también un archivo de texto con los eventos que tuvieron lugar durante el drive test. Estos eventos pueden ser:

- TEST DOWNLINK: test de descarga de datos, G-NetTrack permite la opción de iniciar tests de pruebas de datos, que incluyen una prueba de descarga, otra de subida y una de ping.
- TEST UPLINK: test de subida.
- CALL START: inicio de una llamada.
- CALL END: fin de llamada.
- BLOCKED CALL: llamada bloqueada.
- DROPPED CALL: llamada caída.
- CELL RESELECTION: reelección de celda.
- HANDOVER VOICE: cambio de celda en un servicio de voz.
- HANDOVER DATA: cambio de celda en un servicio de datos.
- COVERAGE LOSS: pérdida de cobertura.
- NETWORK FOUND: nueva red encontrada.
- IRAT HANDOVER DATA: cambio de tecnología en un proceso de datos, puede ser cualquiera de las 6 combinaciones posibles entre 2G, 3G y 4G (4G3G, 4G2G, 2G4G, 2G3G, 3G4G, 3G2G).
- IRAT HANDOVER VOICE: igual que el evento anterior pero en medio de un proceso de voz.
- IRAT CELL RESELECTION: cambio de una celda a otra de distinta tecnología. Las posibles combinaciones son las mismas que en los dos casos anteriores.

En la figura de la siguiente página podemos ver cómo sería un ejemplo real de una campaña con eventos asociados.

Timestamp	Longitude	Latitude	Node	CellID	LAC	NetworkTech	Level	EventType	State
2014.02.14_08.21.52	-3.699.554	40.406.506	166	553	21601	3G	-101	HANDOVER DATA	D
2014.02.14_08.22.48	-3.695.529	40.406.890	166	6163	21601	3G	-81	CELL RESELECTION	I
2014.02.14_08.22.53	-3.695.529	40.406.890	166	532	21601	3G	-83	CELL RESELECTION	I
2014.02.14_08.23.02	-3.691.166	40.409.418	166	6163	21601	3G	-83	CELL RESELECTION	I
2014.02.14_08.23.06	-3.691.166	40.409.418	166	551	21601	3G	-83	CELL RESELECTION	I
2014.02.14_08.23.46	-3.692.459	40.408.544	166	532	21601	3G	-77	HANDOVER DATA	D
2014.02.14_08.25.03	-3.691.166	40.409.418	166	6163	21601	3G	-77	CELL RESELECTION	I
2014.02.14_08.25.23	-3.689.910	40.409.287	166	532	21601	3G	-73	HANDOVER DATA	D
2014.02.14_08.27.23	-3.692.459	40.408.544	166	551	21601	3G	-65	HANDOVER DATA	D
2014.02.14_08.28.22	-3.692.459	40.408.544	166	6163	21601	3G	-71	HANDOVER DATA	D
2014.02.14_08.28.50	-3.692.459	40.408.544	166	532	21601	3G	-67	HANDOVER DATA	D
2014.02.14_08.28.59	-3.691.166	40.409.418	166	6163	21601	3G	-73	HANDOVER DATA	D
2014.02.14_08.29.06	-3.691.166	40.409.418	166	6161	21601	3G	-71	HANDOVER DATA	D
2014.02.14_08.29.28	-3.691.166	40.409.418	166	943	21601	3G	-69	HANDOVER DATA	D
2014.02.14_08.30.04	-3.691.166	40.409.418	166	4722	21601	3G	-63	HANDOVER DATA	D
2014.02.14_08.32.29	-3.693.088	40.415.375	166	6182	21601	3G	-71	HANDOVER DATA	D
2014.02.14_08.34.29	-3.691.398	40.422.064	166	4552	21601	3G	-89	HANDOVER DATA	D
2014.02.14_08.35.00	-3.693.135	40.424.187	166	4492	21601	3G	-95	CELL RESELECTION	I
2014.02.14_08.35.05	-3.693.135	40.424.187	166	4552	21601	3G	-89	CELL RESELECTION	I
2014.02.14_08.36.26	-3.693.135	40.424.187	166	4492	21601	3G	-91	HANDOVER DATA	D
2014.02.14_08.36.42	-3.693.135	40.424.187	166	6182	21601	3G	-87	HANDOVER DATA	D
2014.02.14_08.42.57	-3.691.398	40.422.064	166	4552	21601	3G	-93	CELL RESELECTION	I
2014.02.14_08.46.31	-3.693.135	40.424.187	166	6182	21601	3G	-77	HANDOVER DATA	D
2014.02.14_08.50.05	-3.691.398	40.422.064	166	6183	21601	3G	-65	CELL RESELECTION	I
2014.02.14_08.50.09	-3.691.398	40.422.064	166	6182	21601	3G	-65	CELL RESELECTION	I
2014.02.14_08.50.16	-3.691.398	40.422.064	166	6183	21601	3G	-63	CELL RESELECTION	I
2014.02.14_08.50.24	-3.691.398	40.422.064	166	4493	21601	3G	-63	HANDOVER DATA	D

Figura 4.5: Archivo de eventos

## 4.5 Conclusiones

---

En este capítulo se ha explicado el uso de G-NetTrack, la aplicación móvil que obtiene los parámetros de calidad de las redes móviles. Se ha mostrado su aspecto y se han analizado los archivos y tipos de datos que se obtienen con su uso.

El siguiente capítulo habla del tratamiento de los datos obtenidos en los drive test y del proceso de carga a la base de datos.



# 5

## Proceso de carga de datos

### **5.1 Introducción**

---

Como ya se ha explicado, los archivos creados por G-NetTrack a la hora de realizar las campañas en los drive tests contienen los datos recopilados mediante este proceso, unos archivos a modo de tabla donde cada columna es un campo de información y cada fila una medida tomada.

Este formato es muy poco práctico ya que no permite ningún tratamiento directo si se quiere procesar la información, así que el segundo paso después de crear la base de datos es generar un código o script que lea estos archivos y los almacene en la base de datos anteriormente creada.

## **5.2 Funciones de carga**

---

Este proceso está escrito en Ruby on Rails. Todas las funciones de dicho código se explican a continuación:

.Función `create_campaign`: creamos una campaña usando el modelo `dt_campaign`, modelo creado en una migración (un modelo representa una entidad dentro de la aplicación, normalmente esa entidad viene asociada a una tabla de la base de datos y una instancia del modelo representa una fila de la tabla):

- `start` y `end`, comienzo y final de la campaña, se inicializan a 0, después se actualizarán.
- `campaign_document` y `events_document` cogen los nombres de los archivos generados en G-NetTrack, el de las medidas y el de eventos respectivamente.
- Se devuelve el ID de la campaña creada.

El propio modelo de `dt_campaign`, después de guardarse la campaña, llama a la función (`after_save :process_documents`) que a su vez llama a la función `load_file_data` que es donde se procesa toda la carga:

```
def DtLoadTxt.create_campaign(data_file = nil, events_file = nil)
  c = DtCampaign.create!(
    :start => 0,
    :end => 0,
    :campaign_document => data_file,
    :events_document => events_file,
  )
  return c.id
end
```

.Función `load_dt_data`: coge los nombres de los archivos y llama a la función `create_campaign` pasándoselos como parámetros:

```
def DtLoadTxt.load_dt_data(data_file_name, events_file_name)
  ActiveRecord::Base.transaction do
    file1 = File.open(data_file_name)
    file2 = File.open(events_file_name)
    DtLoadTxt.create_campaign(file1, file2)
  end
end
```

.Función `load_file_data`: función principal de carga:

- Llamamos a `init_hashes`, función que crea los hashes (también conocidos como arrays asociativos, son mapas o diccionarios parecidos a los arrays, son una colección indexada de referencias a objetos) con los contenidos disponibles en la base de datos de las tablas: `dt_operators`, `dt_network_modes` y `dt_event_types`:

```
def DtLoadTxt.init_hashes
  @table_operators = Hash.new ;nil
  op = DtOperator.all
  op.each do |o|
    @table_operators.store(o.name,o.id) ;nil
  end

  @table_network_modes = Hash.new ;nil
  nm = DtNetworkMode.all
  nm.each do |n|
    @table_network_modes.store(n.name,n.id) ;nil
  end
end
```

```
@table_event_types = Hash.new ;nil
ev = DtEventType.all
ev.each do |e|
  @table_event_types.store(e.name,e.id) ;nil
end
end
```

- Creamos @m y @times\_events, hashes globales (por el @) que contienen los datos de los archivos de medidas y de eventos a cargar respectivamente:

```
@m=[]
i=0
f1 = File.open(data_file_name, "r")
f1.each_line do |line|
  data = line.split(/\n/)
  j=0
  d=[]
  while j<55 #numero de columnas del txt de medidas
    d[j] = [data.map{|d| d.split(/\t/)}.flatten][0][j]
    j=j+1
  end
  @m[i]=d
  i=i+1
end

# creamos hash que tendra los eventos
@times_events=[]
i=0
f2 = File.open(events_file_name, "r")
f2.each_line do |line|
  data = line.split(/\n/)
```

```
j=0
de=[]
while j<14 #numero de columnas del txt de eventos
  de[j] = [data.map{|d| d.split(/\t/).flatten}[0][j]
  j=j+1
end
@times_events[2*i]="(#{de[0]})" #solo nos interesa el timestamp
@times_events[2*i+1]= de[12]    #y el tipo de evento
i=i+1
end
```

- Se carga @cell\_states que contiene los datos correspondientes a la tabla dt\_cell\_states cuyo año y semana coinciden con los de la primera medida de la campaña a cargar.

- Inicializamos variables: contadores, arrays que tendrán los datos:

#carga de hash con los datos de dt\_cell\_states con año y semana correspondiente a la primera línea del txt medidas

```
sql_cs = "select l.name, network_cell_id, network_enodeb_id, cell_id from
dt_cell_states c left outer join lacs l on c.lac_id=l.id where c.year = year("#{ @m[1][0]})
AND c.week = week("#{ @m[1][0]}",3);"
```

```
result_cs = ActiveRecord::Base.connection.select_all(sql_cs)
```

```
@cell_states = Hash.new
```

```
i_cs=0
```

```
while (i_cs<result_cs.size)
```

```
  #hash con: LAC/CELL/eNODEb = CELLid
```

```
@cell_states["#{result_cs[i_cs]['name']}/#{result_cs[i_cs]['network_cell_id']}/#{result_
cs[i_cs]['network_enodeb_id']}"] = result_cs[i_cs]['cell_id']
```

```
  i_cs=i_cs+1
```

```
end
```

Entramos en el bucle principal que va línea a línea del hash @m (con las medidas) y leyendo sus valores correspondientes en variables, convirtiendo a integer (entero) o a float (flotante) las necesarias, con las que se operará en los siguientes métodos:

```
while (i<@m.size && @m[i][0]!=nil).
```

El primer IF es el de las celdas vecinas, estos campos no tienen por qué tener valores, depende del modelo de móvil con el que se tomaron las medidas, si este campo tiene datos se entra en el bucle donde almacenan los valores de las tablas vecinas en el array *neighbors*. Las celdas vecinas pueden llegar a ser hasta 6, si hay datos disponibles están en orden: ncellid1, ncellid2..., hasta ncellid6:

```
if(ncellid1 != 0)
```

El segundo IF es el del test de PINGS, si alguno de los campos correspondientes a los pings (average, valor mínimo, valor máximo, desviación estándar o pérdida) tiene algún valor distinto de NULL, se guardan todos en el array pings, junto al timestamp correspondiente a la línea de la medida:

```
if ((pingavg != "" || pingmin != "" || pingmax != "" || pingsdev != "" || pingloss != "") && i<@m.size)
```

Este IF incluye un bucle, ya que si hay pings significa que estamos en el inicio de un Data Test (que consiste en una prueba de Ping-Upload-Download) por lo que si detectamos datos en los campos de ping, aparte de guardarlos entramos en un bucle while que irá guardando los valores de la columna de testup (en la que G-NetTrack guarda los valores bitrates de subida) hasta que detecte que el timestamp de la actual o de la siguiente línea es nulo o hasta que el contador del bucle principal sea mayor que el número de líneas del txt (@m.size) porque puede ocurrir que una campaña acabe justo en medio de un Data Test. Como justo al acabar un ping empieza una prueba de subida, que nosotros consideramos como un evento, guardamos el timestamp y el valor de uplink en un array que luego se guardará como evento:

```
while(testuplink_b == "" && timestampc != "" && i_b+1 < @m.size)
```

El siguiente bucle es parecido al anterior, sólo que este busca cambios en la columna de testuplink, lo que significa que la aplicación ha acabado con el test de subida y que va a comenzar uno de bajada.

```
if (testuplink != "" && i<@m.size)
```

Con los siguientes bucles nos aseguramos de que los campos que estén vacíos o con un guión sean nulos en vez de tener un valor. Por ejemplo, con el campo 'qual':

```
if (qual == "-" || qual == "")
    qual = "NULL"
end
```

Entramos en el apartado de funciones, usadas durante el proceso de carga:

`get_operator_id_or_create`: primero compara que los nombres de los operadores están incluidos en `.operatorname`, el campo que guarda estos valores, para almacenarlos de forma correcta (por ejemplo: `Vodafone_es -> Vodafone`). Después busca una coincidencia en el hash `@table_operators`, que se ha cargado antes con los nombres de los operadores guardados en la base de datos y sus ID's, y si la hay, rescata el ID correspondiente, si no, introduce este nuevo operador con su código y después coge su ID, que será el `operator_id`.

`.get_network_mode_id_or_create`: usa la misma lógica que la función anterior pero para Network Mode solo que no procesa el nombre antes de buscar una coincidencia con el hash cargado de la tabla `dt_network_modes` de la base de datos.

`.get_cell_id`: se usa para asignar el ID de celda a cada medida, compara los valores de Network Cell ID y LAC (para 2G/3G) o Network Enode ID (para 4G), con los del hash `@cell_states` previamente cargado, y si hay coincidencia se queda con el Cell ID, el ID de la tabla de Cells (no confundir con Network Cell ID). Si no hay coincidencia le asigna un valor NULL.

`.compare_timestamps`: esta función es necesaria ya que se observó que los timestamps de los eventos no siempre coincidían con los de medidas, por lo que si se quería asignar un evento a una medida (dándole el mismo ID) se debía normalizar antes su timestamp. La función usa como parámetro el timestamp de cada medida, la va comparando con los tiempos de `@times_events`, si el timestamp de la medida es igual o mayor que el del evento se le asocia este timestamp junto con el tipo de evento en el array `time_and_event`. Para que luego ese timestamp no vuelva a ser asignado se le cambia el valor y se le da el de la hora actual, para así asegurarse que ningún timestamp de una medida realizada en el pasado sea mayor o igual, ya que es la hora del momento en el que se hace la carga. Para que cada timestamp no recorra del todo el hash de eventos se usa un contador global, que se va recuperando cada vez que se compara.

Después de estas funciones se va preparando el insert masivo mediante el comando push en el array 'inserts', siempre y cuando las medidas cumplan unas condiciones, longitud y latitud distintas de 0, level mayor que -200 y precisión menor que 50. También se prepara el insert masivo para eventos.

Una vez acabado el bucle general de lectura de los dos ficheros se comienzan a insertar los datos procesados, hay una condición general que es inserts.size != 0, si no hay datos preparados no se hace nada.

El primer insert es el más importante, es en el que se cargan las medidas principales en la tabla dt\_measurements. Después se actualiza la entrada de la tabla dt\_campaign con las horas inicio y fin de la campaña.

El siguiente paso es coger el ID de todas las medidas que se acaban de insertar, ya que se usarán para relacionar el resto de datos (pings, eventos...) con la tabla dt\_measurements. Se guarda el ID y su timestamp en la variable result, que será muy utilizado de aquí en adelante.

El siguiente insert es el de los eventos, pero antes hay que cambiar el timestamp por el ID de la medida con la misma hora. Para ello se usa la función give\_timestamp\_event\_id, que va comparando los timestamps de result y de los eventos, si hay coincidencia se lleva el ID. Esta función llama a su vez a get\_event\_type\_id, que coge el ID de cada tipo de evento.

Los dos siguientes son eventos especiales, los de Test Up y Downlink, porque para la aplicación no son eventos pero nosotros los tratamos como tal. El procedimiento es el mismo para los dos:

- Primero se rescata el ID correspondiente al evento.
- Después se entra en un bucle que va recorriendo el array con los datos a almacenar.  
En este caso no se puede hacer un insert masivo y hay que hacer un insert por cada línea del array ya que cada vez que se inserta un dato en la tabla de la base de datos se incrementa en 1 el contador de ID que hay que rescatar para guardarlo a su vez en la tabla dt\_avg\_throughputs.
- De forma parecida a la anterior se procede a almacenar los datos de las pruebas de ping, de celdas vecinas, pruebas de descarga y de subida.

## **5.3 Conclusiones**

---

En este capítulo se ha analizado el proceso de la carga de datos, un proceso escrito mediante el uso del *framework* Ruby on Rails, y se han descrito las principales funciones y su cometido en la carga final.

En el siguiente capítulo se muestra la aplicación web final que es la encargada de leer la información desde la base de datos y mostrarla al cliente de forma intuitiva aplicando los filtros indicados por el usuario.



# 6

## La interfaz web

### **6.1 Introducción**

---

El presente proyecto es una ampliación de una aplicación web, un sistema de visualización cuya fuente de datos está basada en contadores, sondas activas y sondas pasivas, capaz de mostrar:

- Los KPIs globales de toda una red.
- Los KPIs filtrados por diferentes dimensiones de la red móvil.
- Los KPIs con diferentes niveles de agregación temporal: nivel semanal y mensual.
- Los KPIs geolocalizados.

## 6.2 Módulo Maps

---

Como se ya dicho antes, esta aplicación consta de varios módulos. Este proyecto es una ampliación del módulo de 'Maps'.

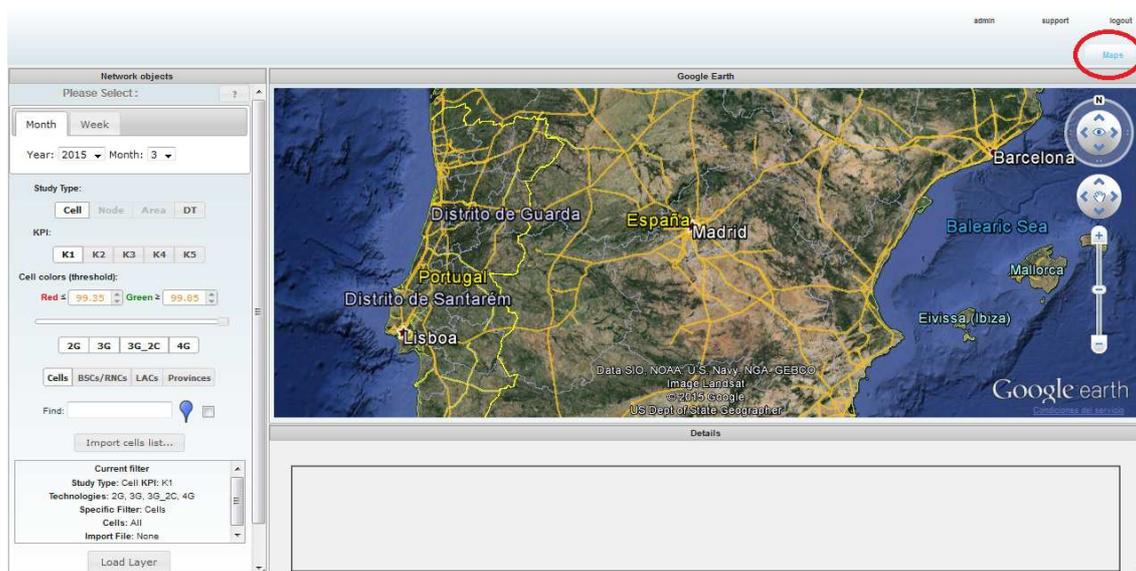


Figura 6.1: Módulo Maps

Este módulo sirve para obtener los datos de los KPIs a nivel de celda, nodo o área y mostrar la información geográficamente referenciada. Se implementa la funcionalidad para visualizar las capas de drive tests.

El presente proyecto consiste en la agregación de una nueva funcionalidad o tipo de estudio al sistema. Los tipos de estudio existentes son:

- Cell: estudio por celdas. La aplicación nos permite ver los KPIs de las celdas de la red. Cada una de las celdas es una de las porciones de 60 grados que puede verse en la figura 6.2. La dirección a la que apunta el aumento del área de la celda representa la dirección de la celda o azimut. Un ejemplo de una representación geográfica de celdas sería:



Figura 6.2: Representación de celdas

- Node: estudio por nodos. Representación de performance por nodos de red a través de códigos de colores (controladores radio: RNCs en 3G y BSCs en 2G) los nodos se mostrarán posicionados geográficamente representando en colores los KPIs. Los umbrales del código de colores se podrán modificar por parte del usuario antes de la generación de la representación.

- Área: se trata de una representación de performance por área geográfica, mostrando los diferentes KPIs a nivel zonal con opciones similares a las de los apartados anteriores. Se puede seleccionar el periodo de estudio, el KPI de análisis, los umbrales para los colores y una o varias portadoras (seleccionando varias portadoras el área reflejará el performance agrupado de esas portadoras).

Estos son los 3 tipos de estudio anteriores que obtienen los datos de fuentes distintas a las usadas en este proyecto:

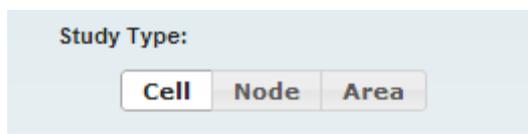


Figura 6.3: Tipos de estudio anteriores

Una vez finalizado este proyecto, este es el aspecto que tendrá esta parte de la aplicación:

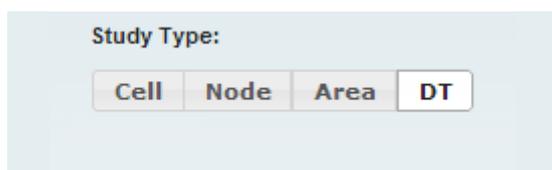


Figura 6.4: Nuevo tipo de estudio

## **6.3 Funcionalidad DT**

---

Como se puede observar se ha añadido una nueva función DT (Drive Test), un nuevo módulo dentro de la pestaña de Maps, obligando a rediseñar el panel de control y la interfaz web, intentando siempre reutilizar lo existente. El nuevo módulo tendría estas opciones:



Figura 6.5: DT

El objetivo de este nuevo módulo es permitir al usuario visualizar gráficamente los datos recopilados mediante los drive tests (usando G-NetTrack). La aplicación web generará un KML con toda la información dentro de unos parámetros definidos por el usuario, ya sean filtros o condiciones específicas, y lo mostrará en el mapa.

A continuación se procede a enumerar y describir cada una de las estas funciones (botones) del módulo DT.

### 6.3.1 La pestaña de fecha

El primer filtro para la búsqueda de datos a representar es la fecha. Se puede filtrar por mes, semana o día. El sistema sólo mostrará dinámicamente las fechas con datos disponibles, para ello hará una consulta a la base de datos en cuanto se cargue este módulo.

Para las pestañas de mes y semana se usan dos listas desplegables con los años y los meses disponibles en el primer caso y los años y las semanas disponibles en la segunda pestaña. Este es el aspecto de las listas:

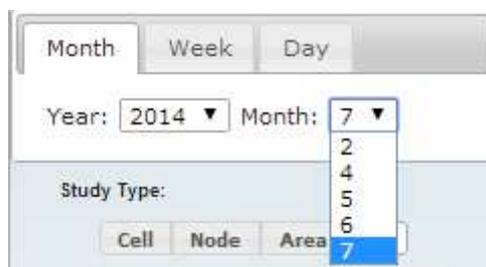


Figura 6.6: Pestaña de fecha

Para la nueva pestaña de día se decidió usar un calendario implementado con jQuery (*Datepicker*), y luciría así:

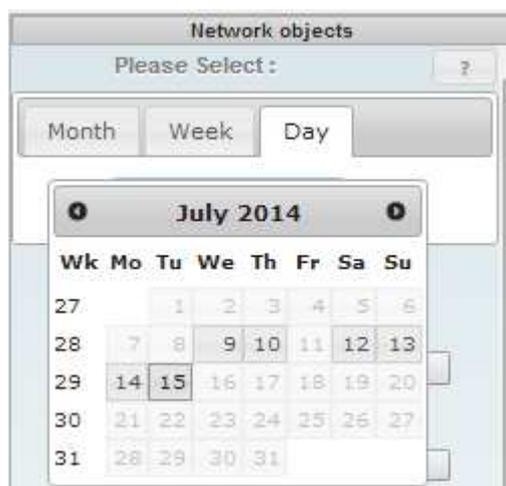


Figura 6.7: Datapicker

Como se puede observar, sólo algunos días están sombreados, estos son los días disponibles en la base de datos, los días en los que se realizaron campañas y por lo tanto disponen de datos para representar.

### 6.3.2 KPI

Un KPI (Key Performance Indicator) conocido como ‘Indicador Clave de Rendimiento’, es una medida del nivel del desempeño de un proceso; el valor del indicador está directamente relacionado con un objetivo fijado de antemano.

En este módulo debemos elegir un KPI de una lista a representar, una lista que contiene alguno de los campos de las medidas que G-NetTrack capturaba a la hora de hacer las campañas. La lista es la siguiente:

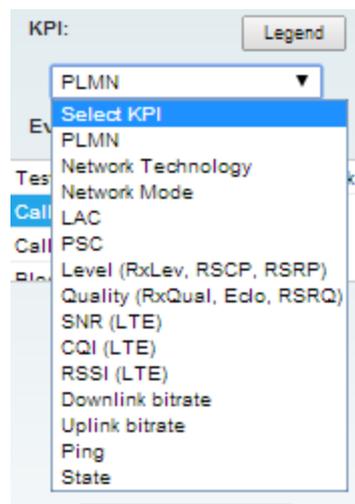


Figura 6.8: Selección de KPI

- PLMN: Public Land Mobile Network, es el KPI que debemos seleccionar si queremos que se nos muestren las medidas diferenciadas por operador móvil. Como se vio anteriormente, en la base de datos las medidas tienen un campo que es `dt_operator_id`, que es un identificador que apunta a la tabla `dt_operators` que contiene el nombre y el código de todos los operadores que se han detectado dentro de las medidas hasta el momento.

El KML generado pintará cada medida con un color por cada operador. Estos colores están predefinidos, pudiendo ser cambiados por el usuario desde el módulo de administración ('admin').

Aquí se puede ver un extracto de la tabla dt\_colors que es la tabla que contiene el código RGB de los colores y desde el cual se cargarán los estilos de los archivos KMLs, pero todo esto se explicará con más detalle.

PLMN	Yoigo	41003F	Purple
PLMN	Movistar	0080FF	Blue
PLMN	Vodafone	FF0000	Red
PLMN	Orange	FF8000	Orange
PLMN	Rest	808080	Grey

Figura 6.9: Tabla dt\_colors

- Network Technology: tecnología utilizada, distingue las medidas que estaban conectadas a una celda 2G, 3G o 4G.
- Network Mode: GPRS, EDGE (2G), HSPA, HSPA+, HSDPA (3G), LTE (4G).
- LAC: organiza las medidas según su LAC (Location Area Code).
- PSC: organiza las medidas por su PSC (Primary Scrambling Code) el código primario de aleatorización de la celda servidora.
- Level: a cada punto a representar, cada medida, se le asigna un color (un estilo en el KML) dependiendo del nivel de señal que tenga. Estos valores, o umbrales, han sido prefijados y, al igual que los colores, pueden ser modificados por el usuario. Como puede observarse en la figura, cada tecnología tiene un trato especial, ya que no sería lo mismo tener un cierto nivel si la celda que proporciona el servicio es 2G, 3G o 4G.

2G	level	0000	-199	-100	FF0000	Red
2G	level	0000	-100	-85	FFFF00	Yellow
2G	level	0000	-85	1000	00CC00	Green
3G	level	0000	-199	-100	FF0000	Red
3G	level	0000	-100	-85	FFFF00	Yellow
3G	level	0000	-85	1000	00CC00	Green
4G	level	0000	-199	-100	FF0000	Red
4G	level	0000	-100	-85	FFFF00	Yellow
4G	level	0000	-85	1000	00CC00	Green

Figura 6.10: Tabla levels

Los colores elegidos fueron rojo, amarillo y verde, de menor a mayor nivel de señal, para que fuera lo más intuitivo posible.

- Quality: la calidad de señal de la red. Al igual que con Level, este KPI tiene 3 colores posibles fijados por unos umbrales ya definidos.

- Lo mismo ocurre con: SNR (Signal to Noise Ratio), relación señal-ruido, CQI (Channel Quality Indicator) indicador de calidad del canal, RSSI (Received Signal Strength Indicator) indicador de intensidad de señal recibida de la energía total de la banda ancha en todo el canal.

- Downlink bitrate y Uplink bitrate: son los KPIs que muestran los datos de las pruebas de descarga y de subida respectivamente, están limitados por umbrales, cada uno con un color definido. Rojo para las velocidades bajas, amarillo para las medias y verde para las buenas.

- Ping: es el KPI para las pruebas de ping, funciona de manera idéntica a los dos KPIs anteriores.

- State: este identificador muestra el estado del teléfono en el momento en el que se tomó la medida, VD (Voz y Datos), D (Datos), V (Voz) o I (Idle, sin uso).

### 6.3.3. Events

Como ya se ha explicado, durante un drive test pueden ocurrir distintos eventos, por ejemplo reelección de celdas, eventos de secuencia de voz, inicio de llamada, fin de llamada, bloqueo de llamadas, llamadas caídas, etc.

Por defecto los KMLs generados no muestran ninguno de estos eventos, es el usuario el que tiene que elegir uno a uno qué eventos quiere que se representen junto a las medidas que serán generadas.

Lista de eventos:

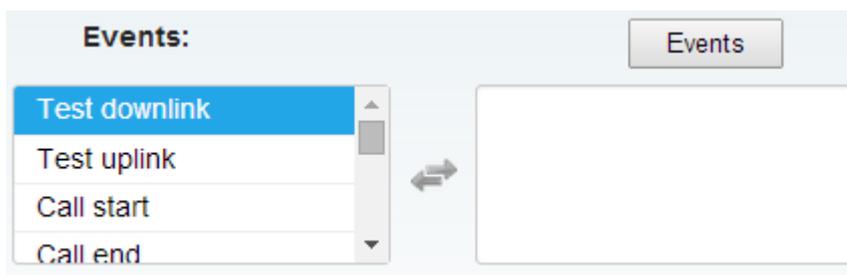


Figura 6.11: Lista de eventos

Cada vez que se seleccione un evento de la lista de la izquierda pasará a la lista de la derecha que es la lista que contiene los eventos a representar.

Por ejemplo, la siguiente situación representaría los eventos ‘Test downlink’ y ‘Test uplink’.

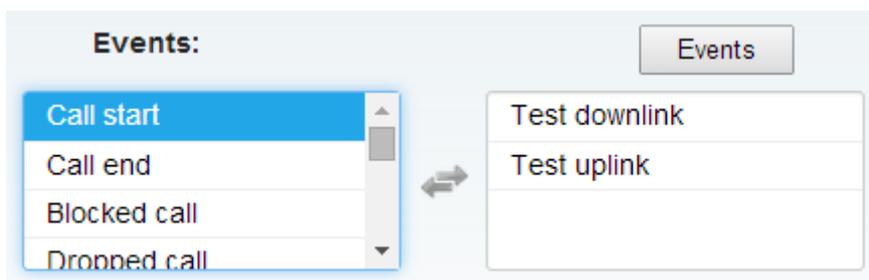


Figura 6.12: Selección de eventos

### 6.3.4 Selección de filtros

La aplicación web da la opción al usuario de filtrar las muestras de las métricas (KPI) y los eventos por distintos parámetros, esa es la finalidad de los siguientes botones:

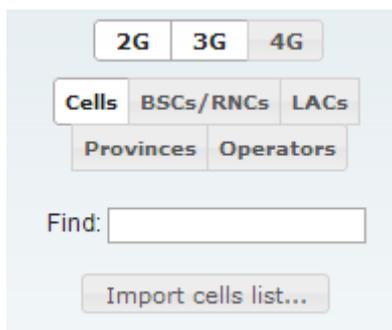


Figura 6.13: Selección de filtros

La primera fila muestra los filtros por tecnología, sólo las tecnologías marcadas serán mostradas en el KML, es decir, si sólo tenemos el botón de 4G activado, se mostrarán únicamente medidas cuyas celdas a las que estaban conectadas eran LTE.

La segunda fila de filtros es:

- “Cells”, serviría para representar únicamente las celdas cuyos nombres se introduzcan en la caja de texto (separados por comas).
- “BSCs/RNCs” permite filtrar todas las celdas que pertenezcan a uno o varios de estos controladores.
- Seleccionando “LACs” es posible introducir uno o varios Location Area Codes por los que filtrar las celdas.
- “Provinces” daría la opción de filtrar las celdas pertenecientes a provincias concretas.
- “Operators” permite filtrar por operador móvil y mostrar sólo las medidas de uno o varios operadores.

Cada uno de estos filtros se complementa con la caja de texto con el título “Find:” donde el usuario escribirá el nombre de celda, BSC/RNC, LAC, provincia u operador por el que quiera filtrar. Este cuadro contiene una función ‘autocompletar’ que facilita al usuario la escritura del nombre en cuestión.

También se puede importar un archivo de texto que contenga los nombres de las celdas, BSCs/RNCs, LACs, provincias u operadores a filtrar mediante el botón “Import (filtro seleccionado) list...”

### 6.3.5 Área de notificación

El siguiente cuadro muestra un resumen del KML que vamos a generar. Por ejemplo:

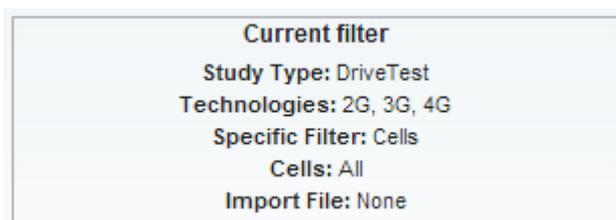


Figura 6.14: Área de notificación

En esta situación tendríamos:

- Study Type: Drive Test. Tenemos el tipo de estudio por drive test.
- Technologies: 2G, 3G, 4G. Haríamos un filtro por estas tecnologías.
- Specific Filter: Cells. Tenemos seleccionado el tipo de filtro por celdas.
- Cells: All. Como no se ha escrito el nombre de ninguna celda por defecto se cogen todas ellas.
- Import File: None. No se ha cargado ningún archivo de texto con nombres de celdas.

### 6.3.6 Load Layer

Este botón contiene la función principal de la página: generar el KML con los datos y filtros seleccionados por el usuario.

El funcionamiento es ir de arriba a abajo en el módulo seleccionando los datos a representar y los filtros a aplicar, y una vez acabado presionar el botón de “Load Layer”.

Este botón funciona también como área de notificación:

- Una vez pulsado se mostrará la etiqueta “Querying”, es decir que se está haciendo la *query* (consulta) al servidor.
- Después aparecerá un porcentaje que irá incrementando que indica el proceso de generación del KML.
- Por último aparecerá la etiqueta “Downloading” seguido de un porcentaje que indica la descarga de este KML un archivo que ocupará más o menos dependiendo de los datos solicitados.
- Una vez cargado el KML este botón vuelve a su aspecto inicial “Load Layer” esperando a la generación de nuevas capas.

Una simulación del proceso sería:



Figura 6.15: Simulación del proceso ‘Load Layer’

### **6.3.7 Import KML**

Este botón sirve para mostrar en el mapa un fichero KML local que el usuario pueda tener ya generado, ya sea de sesiones anteriores, o ajeno a esta aplicación, procesarlo y visualizarlo sobre el mapa de Google Earth.

Se puede usar también para subir al panel de mapas una nueva capa que se superpondría a la generada con la generada por aplicación.

## **6.4 Ventana de visualización**

---

Una vez creado, generado y cargado el KML este estará visible en la ventana de visualización. Para la visualización de datos sobre una componente cartográfica, se decide usar el plugin de navegador de Google Earth. El motivo principal por el que se optó por utilizar el plugin de Google Earth, antes que otros métodos de visualización geográfica, es por el uso generalizado de los ficheros con formato KML y por usarse el plugin ya en el sistema original.

Un archivo KML se utiliza para mostrar datos geográficos en un navegador terrestre, como Google Earth. KML utiliza una estructura basada en etiquetas con atributos y elementos anidados y está basado en el estándar XML. Un archivo KML especifica un conjunto de características geográfica en dos o tres dimensiones (imágenes, polígonos, modelos 3D, descripciones textuales, etc).

Un ejemplo de una capa de drive test cargada sobre la ventana de visualización sería:



Figura 6.16: Ejemplo capa Drive Test

Cada ‘bola’ representa una medida con unos valores determinados. Puede observarse cómo las bolas están alineadas formando una ruta. Todas estas medidas fueron tomadas desde un coche en movimiento por lo que es normal que las rutas estén sobre carreteras.

El título de la ventana muestra un resumen del KML cargado.

En este en concreto podemos observar que se trata de un KPI: PLMN en el mes de febrero del 2014 y que muestra las tecnologías 2G, 3G y 4G.

### **Tooltip:**

Cada bola o medida dispone de una información distinta a las demás. Si se quiere consultar un resumen de esta información basta con pinchar sobre la medida en cuestión, y automáticamente el sistema genera un cuadro de información con los valores que pueden observarse en la siguiente figura:



Figura 6.17: Tooltip

- El ID de la medida.
- El día en que fue tomada.
- La hora exacta.
- El nombre de la celda que le estaba dando servicio.
- La tecnología utilizada.
- El nombre del operador móvil.

## **6.5 Conclusiones**

---

En este capítulo se ha mostrado el aspecto final de la aplicación, tanto su interfaz como cada uno de sus módulos y opciones. También se enseña un ejemplo de ejecución del programa y el aspecto final de una capa KML cargada sobre la ventana de mapa.

El uso de la aplicación es intuitivo para cualquier usuario que esté familiarizado con las tecnologías relacionadas con las redes móviles, este capítulo también podría servir de manual de uso de la aplicación.

# 7

## Integración, pruebas y resultados

### **7.1 Introducción**

---

Como en cualquier proyecto de software, las pruebas de funcionamiento de esta aplicación consistieron en:

- Pruebas unitarias de cada uno de los componentes tras ser desarrollado. Esto incluía:

.Pruebas del proceso de carga: una vez obtenidos los datos había que tratarlos y cargarlos a la base de datos, este proceso realizado en con Ruby on Rails era muy propenso a dar error ya que se realizaban múltiples operaciones con datos de diferentes clases y al finalizar había que hacer una o varias inserciones (*inserts*) de la información a la base de datos. Estos datos provenían de las campañas realizadas mediante la aplicación móvil de medida y estas campañas implicaban una dificultad extra, ya que al tratarse de las pruebas, se precisaba disponer de todos los escenarios posibles, y esto se traducía en tener que forzar al terminal móvil a varias situaciones, como el cambio de tecnología, pérdida de cobertura, inicio de pruebas de descarga y de subida, etc. También era conveniente disponer de datos del mayor número de redes posible por lo que cuanto más se desplazaba el terminal tomando medidas, mejor, lo que implicaba tener que desplazarse a menudo.

.Pruebas de la ampliación de la interfaz web: como ya se ha dicho este módulo era una extensión de una aplicación ya diseñada por lo que la estructura debía ser igual a la existente. Se optó por ir aplicando cada funcionalidad nueva de una en una y probando su funcionamiento al acabar la integración.

.Pruebas de las *queries*: las consultas a la base de datos o *queries*, son una parte fundamental del proceso ya que son las encargadas de seleccionar la información solicitada por el cliente a la base de datos, incluyendo los filtros aplicados (fecha, tipo de tecnología, etc). Las pruebas en este apartado consistían en realizar infinidad de consultas y comprobar si cada una de ellas era correcta, muchas veces realizando las consultas a la base de datos a parte y viendo si los resultados eran afines.

- Pruebas de integración sobre el sistema existente: como este proyecto era una ampliación de otro, ambos debían ser compatibles, y esto se traducía a que las capas o KMLs generados por el nuevo módulo debían añadir información a las generadas por los otros módulos del sistema.

- Pruebas de testeo de interfaz: este proyecto fue diseñado en base a las necesidades de un cliente, un operador móvil. La interfaz debía ser no solo intuitiva, sino también vistosa, elegante y de fácil manejo. Los principales problemas en este apartado fueron de diseño, tales como la creación y colocación de nuevos botones, las nuevas listas, cuadros y funcionalidades.

- Pruebas de testeo de sistema, las pruebas finales incluían todo este proceso:

.Se realizaba una nueva campaña de drive test, intentando obtener la mayor información posible (cuanto más duraba la campaña, más datos se conseguían), también se forzaba a la aplicación a realizar una serie de eventos, procurando anotar la hora exacta de ellos para su posterior comprobación.

.Se almacenaban los datos obtenidos en la base de datos, comprobando en las tablas que la inserción había sido correcta.

.Se accedía a la aplicación y se ingresaba en el sistema. Una vez en el módulo de 'DT' se seleccionaba el día en que la campaña había sido ejecutada, y se cargaba el KPI correspondiente así como los eventos que querían visualizarse. Una vez cargada la capa se comprobaba visualmente que la información mostrada en el mapa correspondía con la campaña realizada.

## **7.2 Conclusiones**

---

Como conclusión a este capítulo remarcar la importancia a la hora de hacer las pruebas ya que cada una ellas simulaba una situación real y se intentó siempre abarcar todos y cada uno de los escenarios posibles, consiguiendo así reparar cualquier tipo de fallo o error que pudiera encontrarse el cliente a la hora del uso real de la aplicación.

Recalcar la dificultad a la hora de conseguir forzar a la aplicación móvil a experimentar todas las situaciones imaginables para comprobar a posteriori que el sistema las adaptaba y las mostraba correctamente.



# 8

## Conclusiones y trabajo futuro

### **8.1 Introducción**

---

En este apartado se hará un análisis del trabajo desarrollado expuesto en esta memoria y se llevarán a cabo las conclusiones finales del documento. Como añadido, se analizarán aquellos puntos que necesitan un trabajo adicional en el subapartado de “Trabajo Futuro”.

### **8.2 Conclusiones**

---

El objetivo del presente PFC era modificar una aplicación web existente, desarrollando una interfaz que transformara los datos obtenidos mediante aplicaciones móviles de medida en un formato que facilite la visualización geográfica de estos datos y que los representara dinámicamente de una manera útil para la monitorización de red a nivel geográfico.

A grandes rasgos, la interfaz debe ser capaz de leer los datos que ofrezcan estas aplicaciones móviles de medida, procesarlos junto con la información geográfica y otra información relevante y enviarlos al sistema de visualización.

El proyecto se ha dividido en tres partes fácilmente diferenciables: diseño y creación de la base de datos, obtención y pretratamiento de los datos mediante el agente móvil para cargarlos en la base de datos y diseño de la interfaz web, que incluye las ‘queries’ que la aplicación hará contra la base de datos.

### **8.3 Trabajo futuro**

---

Algunas de las ideas pendientes para este proyecto serían:

- Creación de filtros por sesiones y usuarios, a día de hoy todos los drive tests realizados sólo pueden distinguirse por la fecha (timestamp) sin diferencias entre quién los realizó ni en qué sesión fueron ejecutados.
- Datos estadísticos o cálculo de ‘*Binning*’, realizar “medias aritméticas” de todas las muestras que haya en una cuadrícula de  $n \times n$  metros.
- Conexión de las muestras con la celda que está sirviendo en cada momento. Sería muy interesante saber a qué celda está conectada cada muestra para poder observar qué celda está fallando u ofreciendo mejor servicio.
- Combinación de datos del agente móvil (drive test) con los de otras fuentes (sondas pasivas).

- Creación de un agente móvil propio. Actualmente la aplicación G-NetTrack sólo está disponible para el sistema operativo Android. Una idea futura sería poder realizar drive tests en cualquier terminal móvil (iOS, Windows Phone...), usando una aplicación de medida propia.
- En el caso en que el número de agentes móviles se disparara entre los usuarios de la red y el uso de la herramienta se multiplicase, se podría evaluar la utilización de bases de datos no relacionales NoSQL (Mongo, Cassandra, etc) para optimizar el rendimiento de la base de datos.
- Adaptación del sistema a las medidas obtenidas mediante otras aplicaciones de agentes móviles.



# 9

## Manual del programador

### **9.1 Condiciones de desarrollo**

Estas serían las especificaciones o requisitos básicos que debe tener la máquina de producción que acoja la aplicación.

También se necesita un disco duro externo para almacenar las campañas drive test creadas desde el cual el sistema las cargará periódicamente añadiéndolas a la base de datos.

#### **9.1.1 Requisitos software**

Los requisitos del servidor que almacena la aplicación web son los siguientes:

- Sistema operativo: Ubuntu server, Debian o CentOS.
- Servidor de aplicaciones Apache 2 (mínimo versión 2.2) con Phusion Passenger 4.0.x.
- Rails versión 3.2.x.
- Ruby versión 1.9.3-p385.
- Node.js.
- RubyGems: 1.8.24.
- MySQL en su versión 5.5.31.
- Git, versión 1.7.10.4 o superior.

En cuanto a los navegadores web de los clientes que se conectan a la aplicación, actualmente está soportado el Google Chrome y Firefox en sus últimas versiones. No está garantizado un correcto funcionamiento del sistema frente a otros navegadores web (Internet Explorer, Safari, Opera, versiones anteriores de Chrome o Firefox), por lo que se desaconseja el uso de estos.

### **9.1.2 Requisitos hardware**

La máquina que alberga actualmente el servidor de aplicaciones tiene las siguientes especificaciones:

- Procesador de dos núcleos Intel Xeon 5160 a 3Ghz.
- 8GB de memoria RAM DDR2 a 667Mhz.
- 2 discos duros de 72GB y 10000rpm dispuestos en RAID1 que almacenan el sistema operativo (Debian 7) y el servidor de aplicaciones.
- 5 discos duros de 146GB y 10000rpm dispuestos en RAID5 con uno adicional en hot-spares en los que se almacena la aplicación, los ficheros subidos a la aplicación y la base de datos del sistema.

### **9.1.3 Requisitos de permisos y conectividad**

Para poder gestionar correctamente la aplicación y el despliegue de los posibles parches o futuras versiones, será necesario lo siguiente:

- Permisos de lectura y escritura (incluyendo la creación de tablas) sobre la base de datos MySQL.
- Conectividad mediante SSH al servidor (será necesario abrir un puerto para este propósito).
- Permisos para poder leer y escribir en la carpeta en la que estará almacenado el repositorio Git.
- Permisos para poder leer y escribir en la carpeta en la que estará desplegada la aplicación.
- Apertura del puerto 443 para el acceso de los usuarios a la aplicación mediante el protocolo HTTPS.

## **A - Bibliografía**

---

- [1] “UMTS Networks, Architecture, Mobility and Services”, Heikki Kaaranen, Ari Ahtiainen, John Wiley & Sons, LTd, pp 129-179.
- [2] “UMTS Performance Measurement, A practical guide to KPIs for the UTRAN Environment”, Ralf Kreher. John Wiley & Sons, LTd, cap 2, pp. 61-180.
- [3] “Signaling in Telecommunication networks”, John G. Van Bosse, Fabrizio U. Devetak. John Wiley & Sons, LTd, pp 327-386.
- [4] “Understanding LTE and its Performance”, Tara Ali-Yahiya, Springer, pp 105-125.
- [5] “G-NetTrack Pro Manual” [Online]. Available: <http://www.gyokovsolutions.com/>
- [6] “HTML/CSS, JavaScript, SQL tutorials”. Available: <http://www.w3schools.com/>
- [7] “Android Telephony” [Online]. Available: <http://developer.android.com/>
- [8] “Google earth plugin” [Online]. Available: <https://www.google.com/earth/>
- [9] Imagen 1.1, fuente: <http://blog.pucp.edu.pe/>
- [10] Imagen 4.1, fuente: <https://play.google.com/>
- [11] “GIS” [Online]. Available: [http://en.wikipedia.org/wiki/Geographic\\_information\\_system](http://en.wikipedia.org/wiki/Geographic_information_system)
- [12] “Rails Guides” [Online]. Available: <http://guides.rubyonrails.org/>
- [13] “QoE” [Online]. Available: <http://www.exfo.com/>



## **B - Presupuesto**

---

- 1) Ejecución Material
  - Compra de ordenador personal (Software incluido)..... 850 €
  - Material de oficina..... 50 €
  - Total de ejecución material..... 900 €
  
- 2) Honorarios Proyecto
  - 12 meses a 500€/ mes ..... 6000 €
  
- 3) Material fungible
  - Gastos de impresión..... 30 €
  - Encuadernación..... 95 €
  
- 4) Subtotal del presupuesto
  - Subtotal Presupuesto..... 7025 €
  
- 5) I.V.A. aplicable
  - 21% Subtotal Presupuesto ..... 1475,25 €
  
- 6) Total presupuesto
  - Total Presupuesto..... 8500,25 €

Madrid, Junio de 2015

El Ingeniero Jefe de Proyecto

Fdo.: Diego Perera Martín  
Ingeniero de Telecomunicación



## **C - Pliego de condiciones**

---

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un desarrollo de interfaz para la representación geográfica de métricas de calidad en operadores de telefonía móvil. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### **Condiciones generales**

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

## **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.