



**Universitat de les
Illes Balears**

Escola Politècnica Superior

Memòria del Treball de Fi de Grau

Posicionamiento de ambulancias

María Calzada González

Grau de Enginyeria Informàtica

Any acadèmic 2016-17

DNI de l'alumne: 43468598R

Treball tutelat per Beatriz Gómez Suárez
Departament de Ciències Matemàtiques e Informàtica

S'autoritza la Universitat a incloure aquest treball en el Repositori Institucional per a la seva consulta en accés obert i difusió en línia, amb finalitats exclusivament acadèmiques i d'investigació	Autor		Tutor	
	Sí	No	Sí	No
	X		X	

Paraules clau del treball:
Emergencias, Posicionamiento, Ambulancias, Ruta, Incidente

ÍNDICE GENERAL

Índice general	I
Índice de figuras	III
Índice de cuadros	V
Acrónimos	VII
Resumen	IX
1 Introducción	1
1.1. Marco conceptual	1
1.2. Motivación	2
1.3. Contexto operacional	2
1.3.1. Programas informáticos utilizados en la sala de operaciones . .	4
1.4. Objetivo	7
1.5. Planificación temporal	8
2 Tecnologías	11
2.1. Java	11
2.2. Glassfish	12
2.3. Protocolo SOAP	12
2.4. Protocolo WSLD	12
2.5. Oracle Database	12
2.6. PostgreSQL	13
2.6.1. PostGIS	13
2.7. GeoServer	13
2.7.1. GeoWebCache	14
2.8. Keytool	14
2.9. Subversion	15
2.10. Log4J	15
2.11. XML	16
2.12. Json	16
2.13. KML	16
2.14. Shapefile	16
3 Análisis	17
3.1. Principales actores	17

3.2.	Especificación de requisitos del software	17
3.2.1.	Requisitos funcionales	17
3.2.2.	Requisitos no funcionales	18
3.3.	Modelo Entidad-Relación	19
3.4.	Problemas principales	19
3.4.1.	Integración con organismos externos	19
3.4.2.	Cálculo de la ruta óptima	24
3.4.3.	Visualización de las rutas	28
4	Diseño	31
4.1.	Caso de uso	31
4.2.	Arquitectura	32
4.2.1.	Programa GeoStatus	34
4.2.2.	Cálculo de la ruta óptima y la visualización	40
4.3.	Infraestructura	42
5	Implementación y resultados	45
5.1.	Programa geoStatus	45
5.1.1.	Jerarquía de directorios	45
5.1.2.	Especificación de la implementación	47
5.1.3.	Librerías	50
5.2.	Cálculo de la ruta óptima y la visualización	51
5.2.1.	Contextualización	51
5.2.2.	Jerarquía de directorios	53
5.2.3.	Especificación de la implementación	54
6	Conclusiones	61
6.1.	Introducción	61
6.2.	Resultados	61
6.3.	Mejoras	62
A	Sistema de Coordenadas de Referencia	63
A.1.	Coordenadas geográficas	64
A.2.	Coordenadas cartesianas	64
A.3.	UTM	65
	Bibliografía	69

ÍNDICE DE FIGURAS

1.1.	Logo SEIB 112	1
1.2.	Sala de operaciones del 112 de las Islas Baleares	2
1.3.	Proceso de registro de incidencias	3
1.4.	Programas utilizados en la sala de operaciones.	4
1.5.	Computer Aided Dispatch (CAD), <i>Computer Aided Dispatch</i>	4
1.6.	Seguimiento Incidentes CAD (SIC), Seguimiento Incidentes CAD.	5
1.7.	Google Earth con una capa donde se muestran las unidades del 112.	6
1.8.	Diagrama conceptual del servicio web	8
1.9.	Diagrama de Gantt del Proyecto	9
2.1.	Comparación de las tecnologías posibles para posicionar información sobre el mapa. Obtenido en [1].	14
3.1.	Modelo Entidad-Relación	19
3.2.	Composición del mensaje EDXL	22
3.3.	Composición del mensaje Emergency Services Alerting Protocol (ESAP)	23
3.4.	Paso 1: obtener las unidades más cercanas	25
3.5.	Paso 2: obtener los puntos de referencia de la ruta	26
3.6.	Tecnologías empleadas para la visualización de las rutas	29
4.1.	Casos de uso	31
4.2.	Arquitectura del sistema	32
4.3.	Diagrama de arquitectura de GeoStatus	34
4.4.	Modelo de datos relacionado con el GeoStatus	35
4.5.	Diagrama de arquitectura del cálculo de la ruta óptima y la visualización	40
4.6.	Modelo de datos relacionado con el GeoStatus	41
4.7.	Infraestructura del proyecto	43
5.1.	Jerarquía de ficheros de geoStatus	46
5.2.	Programa SIC con el nuevo botón para obtener el posicionamiento	52
5.3.	Modal SIC para obtener el posicionamiento	52
5.4.	Jerarquía de ficheros del SIC relacionados con el proyecto	53
5.5.	Contorno de las Islas Baleares	55
5.6.	Ejemplo de visualización de un KML	56
5.7.	Ejemplo de visualización de un KML	57
5.8.	Ejemplo de visualización de las rutas, unidades e incidente.	59
A.1.	Coordenadas geográficas, [2]	64

A.2. Coordenadas cartesianas [3]	65
A.3. Proyección Mercator [4]	65
A.4. Proyección UTM [5]	66
A.5. Proyección UTM de España	67

ÍNDICE DE CUADROS

3.1.	Definición de los nodos de Emergency Data Exchange Language (EDXL) . . .	22
3.2.	Definición de los nodos de ESAP	24
4.1.	Tabla:ISEX.ISEX_PROPERTIES	36
4.2.	Tabla ISEX.ISEX_ESTADOS	36
4.3.	Tabla ISEX_XML_LOG	37
4.4.	Tabla ISEX.ISEX_LOG	37
4.5.	Tabla STORMMA.ACTIVE_UNITS	37
4.6.	Tabla STORMMA.ACTIVE_UNITS	38
4.7.	Tabla geo_limite_islas_baleares	38
4.8.	Tabla geo_radio_position_last	39
4.9.	Tabla geo_route	41
4.10.	Tabla geo_route	42

ACRÓNIMOS

API Application Programming Interface

BBDD Base de datos

CAD Computer Aided Dispatch

EAR Enterprise ARchive

EDXL Emergency Data Exchange Language

EJB Enterprise Java Bean

EPSG European Petroleum Survey Group

ESAP Emergency Services Alerting Protocol

GSM Global System for Mobile communications

Java SE Java Standard Edition

Java EE Java Platform Enterprise Edition

JSON JavaScript Object Notation

KML Keyhole Markup Language

OASIS Organization for the Advancement of Structured Information Standards

OGC Open Geospatial Consortium

OSM OpenStreetMap

SAMU Servicio de Atención Médica Urgente 061

SEIB112 Servicio de Emergencias de las Islas Baleares 112

SIC Seguimiento Incidentes CAD

SHP Shapefile

SLD Styled Layer Descriptor

SOAP Simple Object Access Protocol

SQL Structured Query Language

TFG Trabajo Final de Grado

WKB Well-Known Binary

WSDL Web Services Description Language

XML eXtensible Markup Language

RESUMEN

Actualmente, el sistema de emergencias 112 de las Islas Baleares se encarga de atender y gestionar las llamadas de urgencias y emergencias de esta Comunidad Autónoma.

En el momento que se recibe una llamada, el gestor registra la incidencia y procede a informar a los organismos o instituciones a los que dicha incidencia compete. En caso que se precise de los servicios sanitarios se le comunica la incidencia al Servicio de Atención Médica Urgente 061 (SAMU). Éste se encarga de determinar qué ambulancia o ambulancias se movilizan hacia el incidente.

Para evitar los conflictos generados entre las empresas dedicadas al transporte sanitario urgente de ámbito privado y el SAMU, el *Gobierno Balear* ha decidido que el Servicio de Emergencias de las Islas Baleares 112 (SEIB112) se encargará de regular las emergencias sanitarias.

Con este Trabajo Final de Grado (TFG) se pretende gestionar de manera más eficiente la flota de ambulancias de todos aquellos organismos que están integrados en el sistema del SEIB112, de tal forma que el 112 tenga el conocimiento de cuáles son los recursos disponibles y donde se localizan en cada momento, con la finalidad de determinar qué ambulancias son las idóneas a movilizar.

Gracias al conocimiento de la posición es posible realizar el cálculo del tiempo estimado y la ruta óptima. Este hecho supone una reducción considerable del tiempo de respuesta frente a un incidente, siendo este tiempo fundamental para la vida o la muerte de las personas implicadas.

INTRODUCCIÓN

1.1. Marco conceptual

El Servicio de Emergencias de las Islas Baleares 112 (SEIB112) [6] proporciona un servicio gratuito y de fácil acceso, que permite atender de forma personalizada y permanente todas las llamadas de emergencias a nivel europeo.

El 112 es una numeración gratuita y prefijada. Los teléfonos móviles pueden llamar al 112 siempre que haya cobertura Global System for Mobile communications (GSM) de algún operador, sea éste o no la empresa contratada por el usuario con que realiza la llamada. Esto se debe a que todas las redes atienden este tipo de llamadas gratuitamente.



Figura 1.1: Logo SEIB 112

Este servicio es gestionado individualmente por cada país. En España existe un centro por cada comunidad autónoma operando las 24 horas del día, atendiendo a todas las llamadas y redirigiéndolas a los distintos organismos dependiendo del tipo de emergencia. Cada organismo dispone de su propio sistema de información, con

la finalidad de organizar su propio personal y/o recursos. Estos organismos están incorporados en el sistema de información del 112.

La actividad operativa que corresponde al SEIB112 se desarrolla fundamentalmente en la sala de operaciones, la cual se muestra en la Figura 1.2 que aparece a continuación.



Figura 1.2: Sala de operaciones del 112 de las Islas Baleares

1.2. Motivación

La comunicación entre el SEIB112 y el resto de organismos que intervienen en la gestión de un incidente es un componente esencial para dar una respuesta óptima y eficiente al ciudadano.

La interoperabilidad es una palabra clave en la gestión de emergencias. Se define como la habilidad de dos o más sistemas de intercambiar y reutilizar la información que disponen en sus sistemas. Ésta se consigue de forma inherente en términos de conversación telefónica, no obstante, es un reto conseguirla a través de las tecnologías de la información y comunicación a que estas son heterogéneas.

1.3. Contexto operacional

El ciudadano, que por cualquier razón, llama al 112 para informar de una emergencia, obtiene la confirmación de haber contactado con este servicio y el ofrecimiento de ayuda por parte del operador. Conociendo que ya ha contactado con los servicios de emergencia se pretende tranquilizar, en la medida de lo posible, al ciudadano para que pueda aportar la información precisa con la finalidad de alertar rápidamente a los distintos organismos de emergencias.

Inmediatamente, el sistema comprueba si se encuentra registrado el teléfono desde el cual se ha hecho la llamada o bien si tiene instalada la aplicación *My112* para obtener la localización del ciudadano. De lo contrario, el operador de emergencia que atiende la llamada realiza las preguntas necesarias para localizar geográficamente el incidente,

partiendo de los datos que dispone del ciudadano. Una vez que se ha determinado la localización se centra en la obtención de los datos que permitan definir lo que ocurre.

Posteriormente, el operador finaliza la llamada y aplica el procedimiento operativo previsto para el caso concreto adaptándolo al lugar de la incidencia, incluyendo el envío telemático de las solicitudes de intervención a todos los servicios de emergencia implicados.

Por último, se realiza el seguimiento de la incidencia, anotando toda la información obtenida sobre ésta hasta su finalización.

A continuación, en la Figura 1.3, se muestra un resumen de dicha explicación de una manera visual:

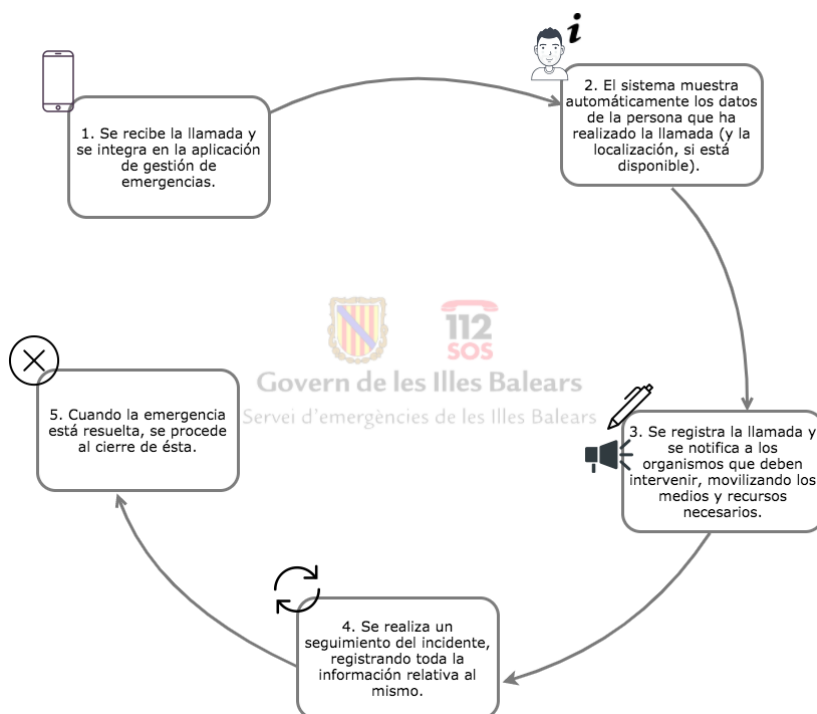


Figura 1.3: Proceso de registro de incidencias

Hoy en día, los operadores del 112 no tienen la información necesaria ni la potestad de movilizar las unidades de los distintos organismos de emergencias. Por esta razón, cuando ocurre un incidente se notifica de dicho suceso a todos los servicios que deben cubrir las necesidades de éste. Por ejemplo, cuando ocurre un accidente de coche con atrapados en la Vía de Cintura, se notifica a los bomberos de Palma, la Guardia Civil y al Servicio de Atención Médica Urgente 061 (SAMU). A partir de este momento, son los organismos los que se encargaran de movilizar sus propios recursos.

1. INTRODUCCIÓN

1.3.1. Programas informáticos utilizados en la sala de operaciones

Los programas utilizados por los operadores son principalmente tres, a continuación se procede a desarrollar sus utilidades:



Figura 1.4: Programas utilizados en la sala de operaciones.

- **Computer Aided Dispatch (CAD)**[7]: programa informático de pago que permite atender llamadas, abrir y actualizar incidentes, y gestionar los recursos críticos proporcionando interacción en tiempo real con sistemas con el mismo programa. Este programa se instaló el año 1995 en la sala de operaciones del 112. A día de hoy sólo se usa para registrar nuevos incidentes.

En la Figura 1.5 se muestra una captura de pantalla de la aplicación.

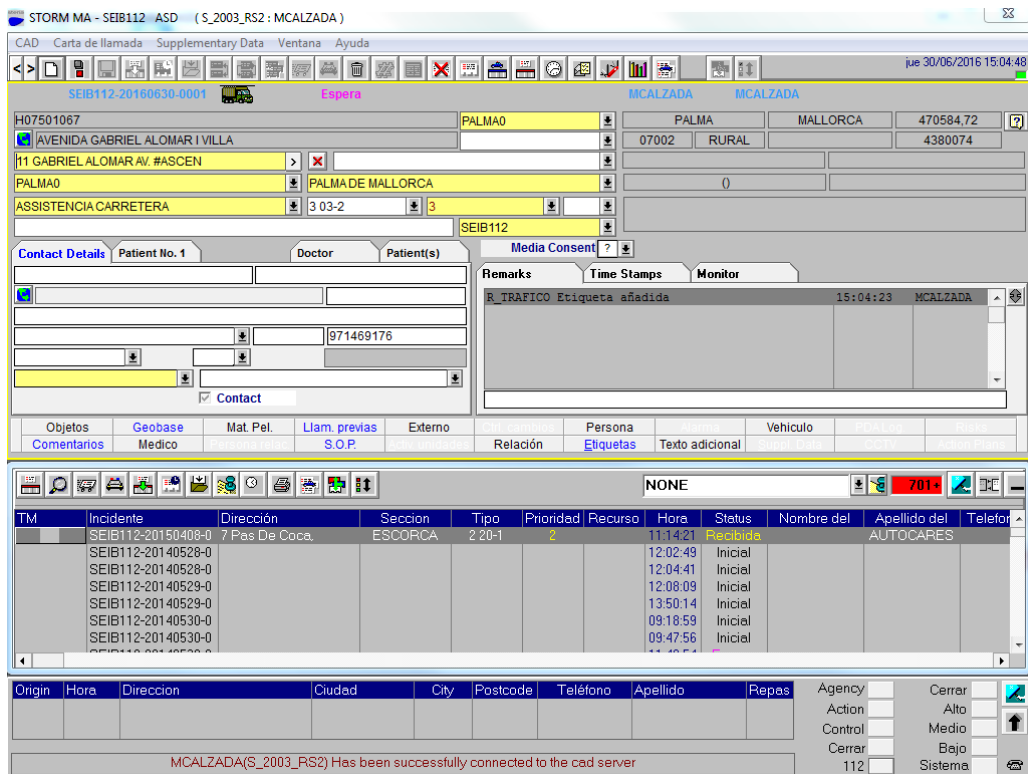


Figura 1.5: CAD, Computer Aided Dispatch.

- Seguimiento Incidentes CAD (SIC):** es una aplicación Java creada por el equipo de desarrollo del SEIB112 que se nutre de la información del CAD. Ésta fue creada con la finalidad de substituir el CAD y de crear nuevas funcionalidades especializadas para nuestra comunidad autónoma. Como ya se ha comentado antes, los operadores sólo utilizan el CAD para registrar los incidentes, el resto de gestiones se realizan a partir del SIC.

Seguidamente aparece una captura de pantalla del SIC (Figura 1.6).

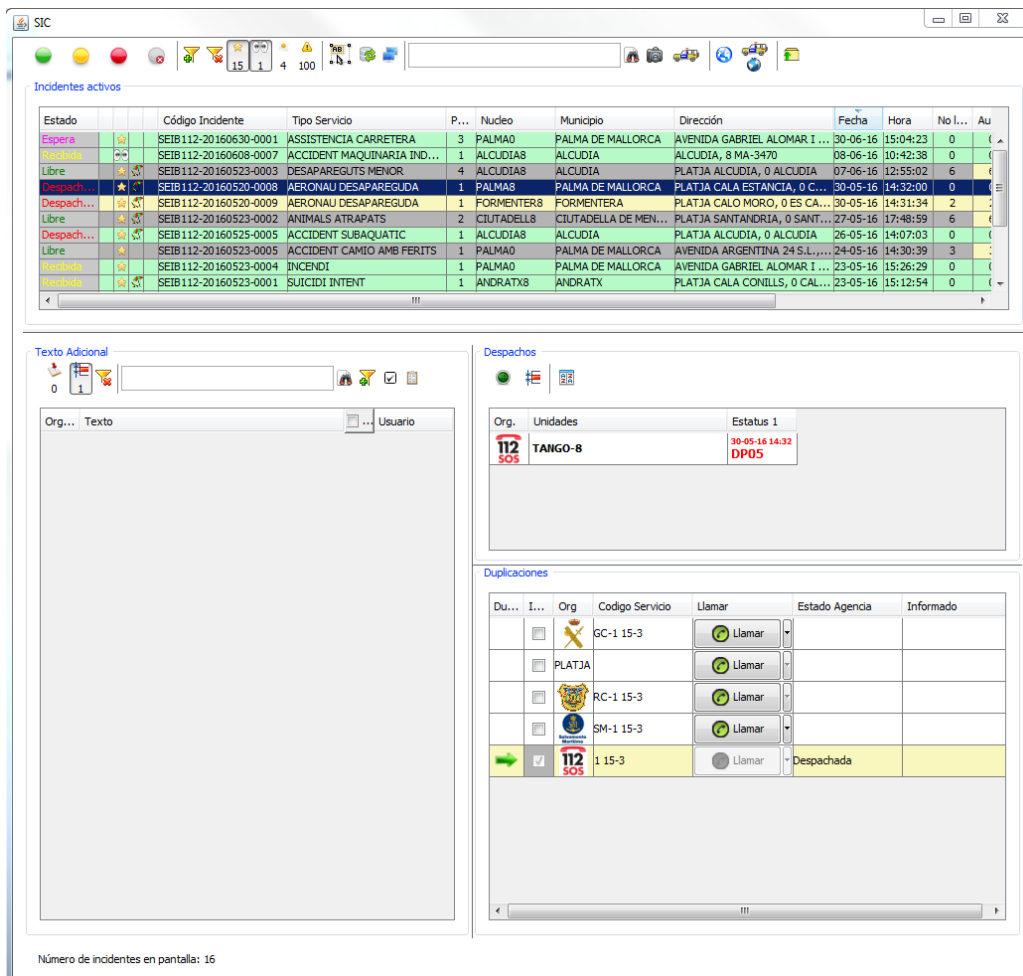


Figura 1.6: SIC, Seguimiento Incidentes CAD.

1. INTRODUCCIÓN

- **Google Earth:** es un programa informático que permite visualizar la cartografía del globo terráqueo. Posee un potente buscador por coordenadas, nombres de ciudades, pueblos, calles, edificios importantes, comercios, etc. Además permite agregar capas para representar datos geográficos en tres dimensiones. Dicho programa es utilizado para geoposicionar los incidentes y las unidades del 112. También, se empleará para mostrar la ubicación de las ambulancias y la ruta calculada hasta el incidente.

A continuación, en la Figura 1.7, se muestra un ejemplo de una capa añadida en dicho programa.

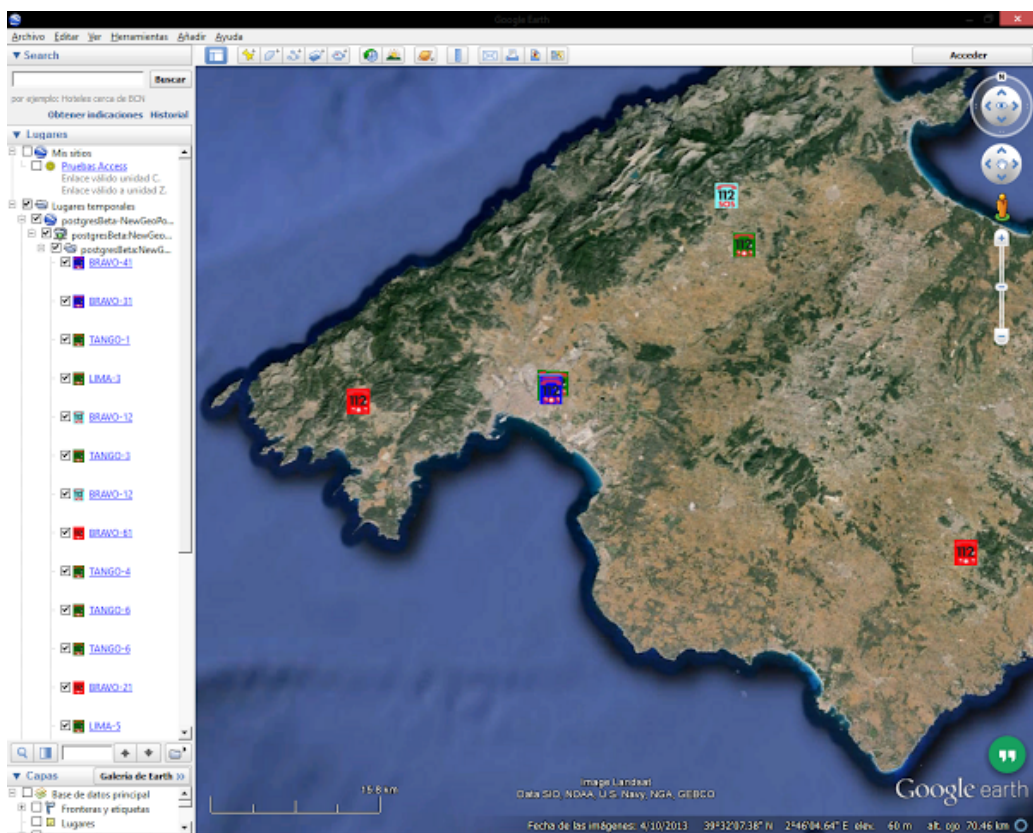


Figura 1.7: Google Earth con una capa donde se muestran las unidades del 112.

1.4. Objetivo

Desde hace años, las empresas privadas dedicadas al transporte sanitario de emergencias de las Islas Baleares disputan por los pacientes de accidentes de tráfico con la finalidad de poder facturarlos a las diferentes compañías de seguros. Así pues, cuando se les notifica de un nuevo suceso, con tal de conseguir a los heridos, acuden más unidades de las necesarias al incidente. [8]

Este proyecto tiene como objetivo solventar esta situación, por lo que se propone incorporar al sistema informático del centro de control del 112 una nueva funcionalidad que permitirá a los operadores de sala asignar a un incidente el vehículo o vehículos sanitarios que se encuentren más próximos al lugar del accidente.

Para resolver este problema, todos los vehículos sanitarios deberán informar de su posición y estado. Al conocer la posición y estado de las unidades sanitarias, es posible calcular cuál o cuáles son las unidades disponibles óptimas para acudir a un incidente. De este modo, solo acudirán al incidente únicamente las unidades necesarias.

Así pues, además de conseguir el propósito de monitorizar y gestionar la flota de vehículos sanitarios de las Islas Baleares, como consecuencia también se logra un beneficio colateral para la sociedad balear. Este beneficio es dar un mejor servicio al ciudadano que requiera de la asistencia de un recurso médico, ya que habrá más unidades disponibles, puesto que en lugar de competir por cada incidente estarán a la espera de otro incidente.

De esta manera, también se logra mejorar la interoperabilidad entre el Servicio de Emergencias de las Islas Baleares 112 (SEIB112) y el Servicio de Atención Médica Urgente 061 (SAMU) manteniendo la información actualizada de las unidades controladas por el SAMU.

Además, el hecho de que la toma de decisiones se centralice en el centro de emergencias 112 evita los favoritismos a según que empresas privadas. Puesto que la decisión únicamente se tomará en base a la rapidez en acudir al incidente.

Gracias al conocimiento preciso de cuál es el estado y posición de todas las unidades, el SIC podrá proponer a qué agencia se le debe notificar del incidente e informar de cuál es el recurso que debe movilizar.

Para facilitar la comprensión de la situación actual de las unidades, se añadirá la posibilidad de visualizar las posiciones de las distintas unidades y la ruta de las unidades más cercanas a un determinado incidente a través del Google Earth.

En la Figura 1.8 podemos ver el esquema que representa el protocolo explicado hasta ahora.

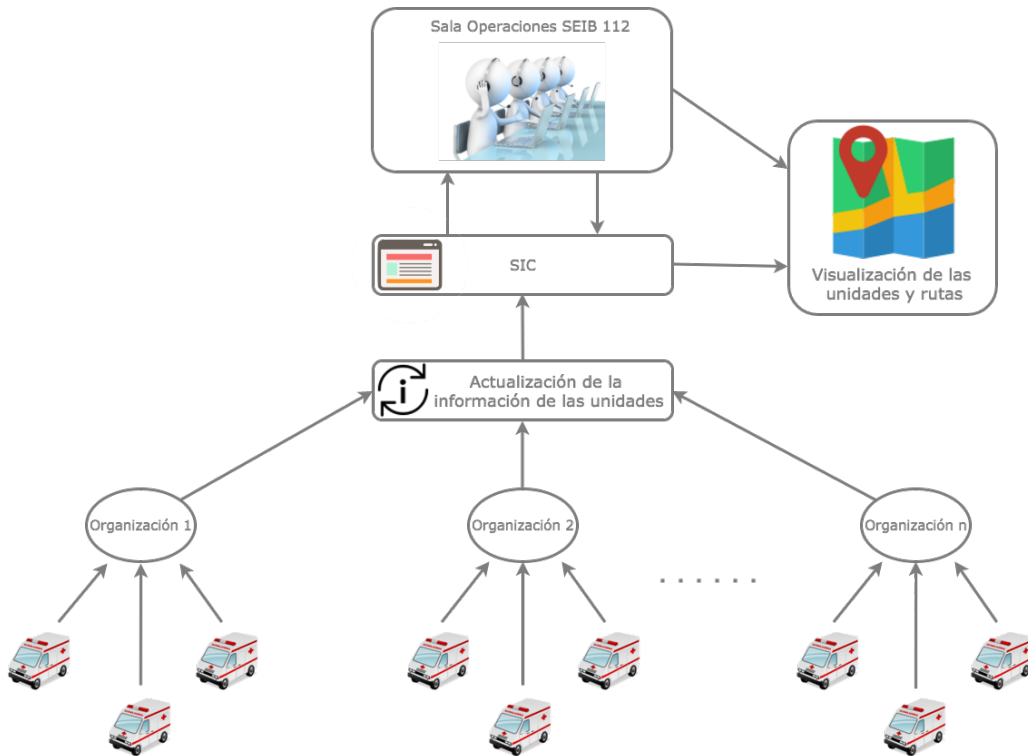


Figura 1.8: Diagrama conceptual del servicio web

1.5. Planificación temporal

En este apartado se recoge la planificación temporal del proyecto. Se ha dividido esta planificación en 6 tareas principales:

1. **Introducción al proyecto:** constituye todas aquellas tareas de investigación para poder alcanzar el perfil técnico requerido y así garantizar la finalización del proyecto. También se incluye la instalación y configuración del software necesario para poder realizar el desarrollo de éste.
2. **Análisis:** consiste en estudiar, investigar y especificar los requisitos del proyecto para poder cumplir con los objetivos que se esperan de este proyecto. Aparte, se hará un examen detallado de estos requisitos para poder incidir en la solución de forma correcta.
3. **Diseño:** una vez pasada la etapa de análisis, se deben descomponer y abstraer la solución en varios problemas para poder cumplir con los requisitos de éste. Además, es importante tener en cuenta el ecosistema donde convivirá el proyecto una vez integrado.
4. **Implementación:** se ejecutará la solución encontrada y diseñada anteriormente. Habrá que tener en cuenta los requisitos no funcionales del sistema.

5. **Integración:** se debe de facilitar y preparar la puesta en producción del proyecto.
6. **Documentación:** elaboración de documentos entregables: manual de usuario, memoria y presentación. Es importante documentar de forma exacta y precisa de manera que la documentación tenga valor.

En la Figura 1.9 se muestra el diagrama de Gantt en el cual se representan todas las actividades comentadas anteriormente con la estimación de cada una de éstas.

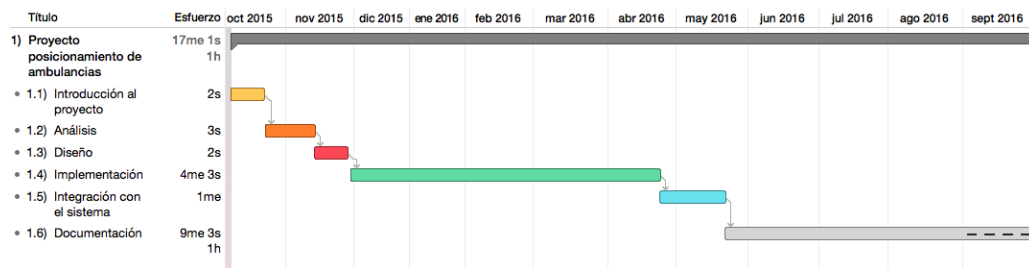


Figura 1.9: Diagrama de Gantt del Proyecto

TECNOLOGÍAS

A continuación se describen las tecnologías empleadas para la realización de este proyecto. Al tener que integrar este Trabajo Final de Grado (TFG) en el entorno del SEIB112, la mayoría de las tecnologías utilizadas para éste son heredadas.

2.1. Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y multiplataforma [9]. Con su versatilidad, eficacia y portabilidad, Java se ha convertido en uno de los lenguajes más usados en todo el mundo.

Para implementar la parte del proyecto que engloba la recepción y el tratamiento del XML se ha empleado el formato Enterprise ARchive (EAR) utilizado por Java Platform Enterprise Edition (Java EE).

Java EE es una plataforma centrada en el desarrollo de software empresarial basada en el lenguaje Java. Es una extensión de la plataforma Java Standard Edition (Java SE) que proporciona una API con funcionalidades añadidas. Java EE simplifica el desarrollo de aplicaciones y reduce la necesidad de programación y formación para programadores al crear componentes modulares reutilizables y hacer énfasis en la convención sobre la configuración.

El formato de fichero de empaquetado de módulos de Java EE es EAR, el cual empaqueta un conjunto de recursos (la aplicación web (.war), la aplicación EJB (.jar) y la aplicación java (.jar)) con el fin de facilitar el despliegue.

Los Enterprise Java Bean (EJB) proporcionan un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación

empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

2.2. Glassfish

La aplicación EAR, comentada anteriormente, se debe desplegar en un servidor de aplicaciones. La herramienta elegida para este proceso es Glassfish [10].

Glassfish es un servidor de aplicaciones de software libre que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Además, su panel de control proporciona una gran cantidad de funcionalidades de forma transparente al usuario.

2.3. Protocolo SOAP

Simple Object Access Protocol (SOAP) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

El mensaje está compuesto de tres partes: un sobre, la cabecera y el cuerpo. El sobre rodea al mensaje y contiene la cabecera y el cuerpo. La cabecera es un elemento opcional que provee información para el encaminamiento del mensaje y el cuerpo contiene datos etiquetados como XML.

2.4. Protocolo WSDL

Web Services Description Language (WSDL) es un fichero XML que describe el conjunto de métodos expuestos por un servicio web. Esta descripción incluye el número de argumentos, y tipo de cada uno de los parámetros de cada uno de los métodos, así como la descripción de los elementos que retornan. Estas descripciones son las que se utilizan para generar los objetos proxy que se usan en los entornos de desarrollo. El código local de un proxy de servicios web es el encargado tanto de construir como de recepcionar las llamadas SOAP del servicio web.

2.5. Oracle Database

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional desarrollado por Oracle Corporation [11]. Se considera como uno de los sistemas de bases de datos más completos, destacando por el soporte de transacciones, la estabilidad, la escalabilidad, y el soporte multiplataforma.

La mayoría de los datos almacenados del SEIB112 se encuentran en bases de datos Oracle.

2.6. PostgreSQL

En este proyecto se requiere visualizar en el mapa la posición de las ambulancias. PostgreSQL junto a PostGIS permite almacenar la localización de tal manera que dicha información pueda representarse en un mapa. Además, también posibilita establecer relaciones en los datos teniendo en cuenta su posición geográfica.

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos de código abierto [12]. Incluye una biblioteca de funciones estándar con cientos de funciones integradas que van desde las operaciones matemáticas básicas, operaciones con strings y soporte para almacenar y manipular datos espaciales.

Además, con PostgreSQL es posible implementar vistas materializadas. Una vista materializada es una tabla virtual, que representa el resultado de una consulta que se almacena en una tabla cacheada, la cual será actualizada de forma periódica a partir de las tablas originales. Se emplearán las vistas para visualizar los elementos deseados en un mapa.

2.6.1. PostGIS

PostGIS es una extensión que añade soporte de objetos geográficos a PostgreSQL y permite almacenar, manipular y realizar análisis mediante consultas Structured Query Language (SQL) espaciales [13]. Estos objetos geográficos, PostGIS los almacena en columnas del tipo GEOMETRY que no es directamente interpretable por el usuario. Utilizan un formato descrito por el Open Geospatial Consortium (OGC) denominado Well-Known Binary (WKB), el cual garantiza la interoperabilidad.

2.7. GeoServer

Una vez ya decidida la plataforma que se encarga de la visualización de mapas, el siguiente paso es posicionar los elementos deseados sobre el éste. Para ello, será necesario obtener la geolocalización de dichos elementos, las cuales se encuentran en una base de datos PostgreSQL externa. Por este motivo, GeoServer es la herramienta perfecta [14].

GeoServer es un servidor de mapas de código abierto. Es una aplicación web de código abierto, escrito en Java, que permite a los usuarios compartir y editar datos geoespaciales. GeoServer es capaz de obtener la información desde muchas fuentes de datos diferentes, en este caso accede a una base de datos externa PostgreSQL.

Su principal funcionalidad en este trabajo será la de servir de comunicación entre la aplicación donde se muestran los datos y la base de datos donde se almacenan.

Para poder entender el funcionamiento del GeoServer a continuación se describen las distintas partes de las que está compuesto:

- *Base de datos*: almacena las posiciones con sus descripciones de las distintas unidades.

2. TECNOLOGÍAS

- *Capas*: los datos almacenados en la Base de datos (BBDD) se pueden mostrar visualmente.
- *Estilos*: permite configurar la visualización, es decir las imágenes que deben aparecer y los colores de la fuente.

El GeoServer está en un servidor Web, por lo cual, si el usuario no tiene los permisos de red no se podrán obtener los datos externamente.

2.7.1. GeoWebCache

GeoWebCache es una aplicación web de código abierto, basada en Java, que aumenta el rendimiento de un Keyhole Markup Language (KML) mediante un pre-dibujo y almacenamiento (cacheado) de las capas de mapa servidas por GeoServer para dar una respuesta más rápida a las solicitudes de los clientes.

A continuación, se muestran las múltiples tecnologías que permiten la visualización del mapa junto con las geolocalizaciones de los elementos deseados como la posición de las ambulancias, el lugar del incidente, las líneas que indican la ruta óptima, etc.



Figura 2.1: Comparación de las tecnologías posibles para posicionar información sobre el mapa. Obtenido en [1].

2.8. Keytool

Keytool [15] es un pequeño programa auxiliar distribuido entre los binarios del entorno de desarrollo de Java que sirve para gestionar claves criptográficas. Esta herramienta permite generar fácilmente pares de claves para algoritmos de criptografía asimétrica.

Se empleará para reconocer si el organismo está autorizado por el SEIB112.

2.9. Subversion

El uso de un control de versiones es imprescindible para cualquier desarrollador, el cual permite gestionar los diversos cambios que se realizan sobre los elementos de un producto. Para este proyecto se ha empleado el cliente de versiones TortoiseSVN, puesto que es el que utiliza el equipo técnico del SEIB112.

Subversion es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Esto implica que el usuario puede recuperar versiones antiguas de los archivos y examinar la historia de cómo y cuándo cambiaron los datos, y quién hizo el cambio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. Además, permite trabajar de forma colaborativa, es decir, cada usuario modifica la copia local y cuando éste decide compartir los cambios, el sistema automáticamente intenta combinar las diversas modificaciones.

2.10. Log4J

Se emplea la librería Log4j (Log For Java) para centralizar todos los mensajes de información del proyecto en archivos logs. Estos ficheros almacenan todas las trazas o errores a lo largo del funcionamiento de los distintos servicios. Este framework permite configurar las trazas de salidas en un formato concreto y por distintos niveles de prioridad:

- *Fatal*: se utilizan para mensajes críticos de la aplicación.
- *Error*: se utilizan en los mensajes de errores producidos por ejemplo por algún parámetro de entrada o inconsistencia de algunos datos de entrada.
- *Warn*: se emplea para avisar de alguna alerta sobre determinados eventos o funcionamientos que no afectan el funcionamiento normal de la aplicación.
- *Info*: se utilizan para mostrar mensajes informativos.
- *Debug*: sirve para mostrar los mensajes de depuración en las fases de pruebas o pre-producción.
- *Trace*: su funcionamiento es similar que DEBUG, pero con un mayor nivel de detalle.

Una vez desplegada la aplicación, el contenedor redirige todas las trazas o informaciones necesarias para registrar tanto el uso como los parámetros de entrada, estados de ejecución, etc., al log para alguna evaluación o depuración a posteriori.

2.11. XML

eXtensible Markup Language (XML), es un meta-lenguaje que permite definir lenguajes de marcas utilizado para estructurar, almacenar e intercambiar datos en forma legible. Éste se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

2.12. Json

JavaScript Object Notation (JSON) es un formato ligero para intercambiar datos de manera sencilla. Para los humanos resulta de fácil lectura y escritura y para las máquinas es simple de interpretar y de generar.

2.13. KML

KML es un lenguaje de marcado basado en XML que se utiliza para mostrar datos geográficos, anotaciones e imágenes en un navegador terrestre, como *Google Earth*, *Google Maps*.

2.14. Shapefile

Un shapefile es un formato de almacenamiento de datos vectoriales para almacenar la ubicación, la forma y los atributos de las entidades geográficas.

CAPÍTULO 3

ANÁLISIS

En el siguiente capítulo se detallan los requisitos como base para el diseño del proyecto, se define el Modelo entidad-relación y también se investigan los problemas principales del proyecto.

3.1. Principales actores

A continuación se presentan los actores que van a intervenir en el proyecto:

- **Organismo / Agencia:** cuerpo operativo que en algún momento es susceptible de ser movilizadado para intervenir en la resolución eficaz de la emergencia.
- **Unidad:** en el ámbito de este proyecto un recurso son todos aquellos vehículos gestionados por un organismo.
- **SEIB112:** organismo cuya misión es favorecer la coordinación entre servicios y organismos intervinientes en una emergencia.

3.2. Especificación de requisitos del software

A continuación se muestran de forma enumerada una breve descripción de los requisitos establecidos para el diseño y desarrollo del proyecto.

3.2.1. Requisitos funcionales

RF-1: *El estado o la posición de la unidad deben estar especificados en el XML.*

RF-2: *El sistema debe establecer un tiempo mínimo de actualización de la unidad: en caso de no cumplir este tiempo mínimo no se tendrá en cuenta para realizar el*

cálculo de unidades cercanas al incidente, puesto que se deduce que los datos no están actualizados y por tanto que el recurso no está activo.

RF-3: *El sistema siempre tiene que determinar qué unidades son las más cercanas.*

RF-4: *El Identificador del XML recibido debe ser único.*

RF-5: *El sistema debe de registrar las constantes actualizaciones de las unidades.*

3.2.2. Requisitos no funcionales

Usabilidad

RNF-1 *La nueva funcionalidad del programa SIC debe constar de una interfaz sencilla e intuitiva, de tal forma que su uso no suponga un impedimento o esfuerzo al usuario a la hora de utilizar la aplicación.*

RNF-2 *El número de pasos hasta saber cuál es la ambulancia idónea deben ser 3.*

RNF-3 *El usuario debe reconocer fácilmente cuál es la ambulancia más cercana al incidente: para ello, debe ser fácil visualizar qué ambulancias están disponibles, cuya manera más fácil es distinguir los estados con diferentes colores. Además, también se debe mostrar el tiempo o la distancia estimada desde cada ambulancia mostrada hasta un determinado incidente.*

RNF-4 *Para facilitar la decisión sólo se deben de mostrar 5 rutas para cada incidente: el cálculo de la ruta se limitará por las cinco ambulancias que se encuentren más cerca del incidente.*

Rendimiento

RNF-5 *El sistema deberá tener un tiempo máximo de respuesta de 6 segundos: este requisito se debe cumplir tanto en el servicio web como en la nueva funcionalidad del SIC. Es decir, la respuesta del XML y el cálculo de las unidades más cercanas a un incidente, deberá ser menor a la definida anteriormente.*

Seguridad

RNF-6 *El sistema deberá ser capaz de evitar ataques de inyección de SQL sistemáticos.*

RNF-7 *Se deberá de utilizar la firma digital para la validación de los datos: una vez recibido el XML se debe identificar la procedencia de éste; en caso de no reconocerla el XML se rechazará. Sólo se aceptarán como válidos los XML emitidos por el SEIB112 y por una serie limitada de organismos, que corresponden al conjunto de cuerpos operativos que participan en el modelo de interoperabilidad. Cualquier otro mensaje generado por cualquier otra entidad quedará automáticamente descartado.*

Organizacionales

El SEIB112 ha implementado anteriormente una serie de integraciones con servicios externos, de estas experiencias anteriores vienen dados los siguientes requisitos para la parte de la integración de los datos de posicionamiento y estado de las unidades.

RNF-8 La integración se realizará mediante un XML y el protocolo Emergency Data Exchange Language (EDXL) [16].

RNF-9 La comunicación con los organismos será mediante el protocolo SOAP.

Por otra parte, existe el siguiente requisito que afecta a la totalidad del proyecto:

RNF-10 *El lenguaje que se utilizará será Java*: es necesario realizarlo en este lenguaje ya que las herramientas en las que se va a integrar están desarrolladas en Java.

3.3. Modelo Entidad-Relación

El modelo de datos surge a partir del análisis de la información necesaria para realizar el cálculo de la ruta más corta y de la integración con los organismos de emergencias sanitarias.

La representación del modelo de datos se realiza a través del siguiente diagrama entidad relación en el que, para facilitar su lectura, no se detallan los atributos de las entidades.

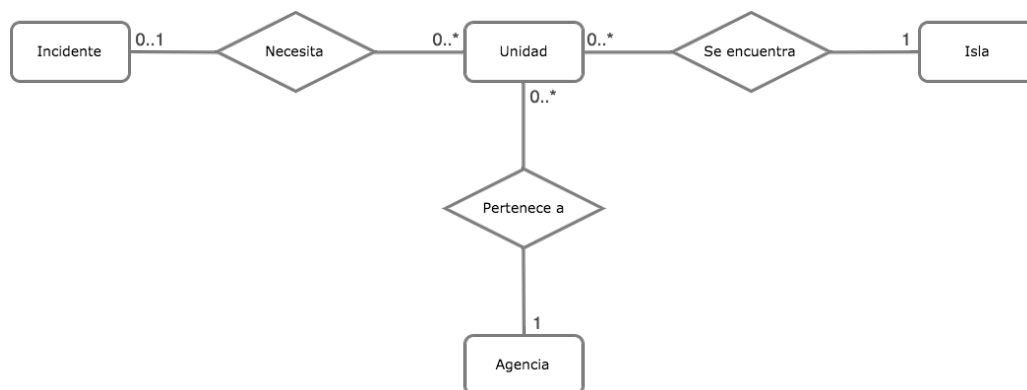


Figura 3.1: Modelo Entidad-Relación

3.4. Problemas principales

3.4.1. Integración con organismos externos

Como se ha comentado en el RNF-8, para realizar el intercambio entre los datos de los diferentes organismos y el SEIB112 se empleará el protocolo EDXL.

3. ANÁLISIS

EDXL es un protocolo basado en un conjunto de mensajes estándar XML que facilitan el intercambio de información relativa a emergencias entre los diferentes actores que participan en su gestión y resolución. Este protocolo fue desarrollado por la organización Organization for the Advancement of Structured Information Standards (OASIS) y su objetivo es mejorar la velocidad y calidad en el intercambio de información entre diferentes organizaciones.

OASIS, es un consorcio que impulsa el desarrollo, la convergencia y la adopción de estándares abiertos para la sociedad mundial de la información. Dentro de OASIS se distingue una unidad que está especializada en la publicación de estándares específicos para el intercambio de información de emergencias. Se trata de la organización OASIS *Emergency Management TC*.

Estructura del mensaje:

- **EDXL:** contiene la información necesaria por las tareas de encaminamiento y distribución de la mensajería. El principal consumidor de esta parte es la pasarela, que interpretará la información para encaminar y transformar el mensaje según el destino.
- **Emergency Services Alerting Protocol (ESAP):** contiene la información de datos de emergencias que son necesarios para la gestión de la emergencia. El protocolo ESAP está encapsulado dentro de la familia de protocolos EDXL de OASIS. Ha sido creada por Telefónica para mejorar el intercambio de información entre centros coordinadores de emergencias y organismos que disponen de recursos para resolverlas.

A continuación se muestra un ejemplo de cómo son los campos principales de un XML enviado por el cuerpo operativo hacia el 112.

```
<?xml version="1.0" encoding="ISO-8859-15" standalone="no"?>
<EDXLDistribution xmlns="urn:oasis:names:tc:emergency:EDXL:DE:1.01" ←
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ←
  xsi:schemaLocation="urn:oasis:names:tc:emergency:EDXL:DE:1.01 EDXL-ED-1.01.xsd ←
  urn:oasis:names:tc:emergency:ESAP ESAP_v3.08.xsd">
  <distributionID>MCALZADA228</distributionID>
  <senderID>IB@COP092.COM</senderID>
  <dateTimeSent>2014-12-11T07:41:31.727Z</dateTimeSent>
  <distributionStatus>Actual</distributionStatus>
  <distributionType>Report</distributionType>
  <combinedConfidentiality>CLASSIFIED AND SENSITIVE</combinedConfidentiality>
  <contentObject>
  <xmlContent>
  <embeddedXMLContent xmlns:other="urn:oasis:names:tc:emergency:ESAP">
  <other:esap xmlns="urn:oasis:names:tc:emergency:ESAP" schemaVersion="3.07">
  <msgIncidentUpdate>
  <header>
  <origin>092PALMA</origin>
  <destination>112Balears</destination>
  <incidentID/>
  <incidentIDService/>
  <date>11/12/2014 08:41:31</date>
  </header>
  <resources>
  <resource>
```



```

<name>Ambulancia X</name>
<status>B</status>
<ISSI>1234</ISSI>
<code>ECO-610</code>
<annotation/>
<coordinates>
  <utm_X>2.650286</utm_X>
  <utm_Y>39.564605</utm_Y>
  <timeZone> </timeZone>
  <datum> </datum>
</coordinates>
</resource>
</resources>
</msgIncidentUpdate>
</other:esap>
</embeddedXMLContent>
</xmlContent>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod ↵
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2001/04/xmenc#sha256"/>
      <DigestValue>NmWZ0skpqCch07WndghKHBZTZ2M5otDJocH3v8Mpd8w=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>
    e6D3VcJQABxE/6G/SblHmByP62QcUjHFGmqbg4i8FX9eE5IPZY5GBxtC2Xem00AHTSHqVPf72YWm
    LUV1UCDWgQ5nWl5RdCoe4hbKlMqEIHJnnkbHBcim5TLuTIaravz7k0ugmRV6noL2mPmD5lW9+jb4
    8s/KiILBj31BoJMuiRpbTkqsjIc15uVksSwsB3xwBx33ZVn9zix30VysRclwfjmrR3Yz9f5aSbjV
    VrOnLoLyLtm0Y5WnVx89cVTcT0Xe09EjRo0a8Roq78D0oGNVlVvVvAQjwaSSPhJ0soj pneUN8udIQ
    QCiB13D0jMmip/motMcWU6hQzE4SIwy0SCpWg==
  </SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509Certificate>
        MIIDezCCAmOgAwIBAgIESTiwDjANBgkqhkiG9w0BAQsFAADBuMQswCQYDVQQGEwJFUzERMA8GA1UE
        CBMIQmFsZWZyZXNxdjAMBGNVBAcTBVBhbG1hMQ4wDAYDVQQKEwUxMTJpYjETMBEGA1UECmKRGVz
        YXJyb2xsbzEXMBUGA1UEAwwOTWFyYwFhIENhbHphZGZGEwHhcNMTYwMjE5MDg1ODM0WhcNMTYwNTE5
        MDg1ODM0WjBuMQswCQYDVQQGEwJFUzERMA8GA1UECBMIQmFsZWZyZXNxdjAMBGNVBAcTBVBhbG1h
        MQ4wDAYDVQQKEwUxMTJpYjETMBEGA1UECmKRGVzYXJyb2xsbzEXMBUGA1UEAwwOTWFyYwFhIENh
        bHphZGZGEwEiMAOGCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCpA19ND6PzMRU20nWUTBBagjoJ
        2R0KL/8IhK1oK6IXVCJ6E+KudKxVZmm10Xr0BYjxkGduOMZT0QzSDwq15w/6bcqXKGQV0ub0A+Y0
        UzNhQjYMFZehvspFCL6JY0KjX2IN3cMxibgMDu/S+uQvqj8RsY1X95A0E3pPJDbjxYKQmMVTddSW
        L67bTm0YngdzFtBHyM8cvY9HSVnzUxtcEz0mJd8FBWPNW6X3j4Miaql1vLiLy6dWm4MKohCTDTdQd
        q6yhgh8Lnfbc//2KoFhv1iM5ptbmSbro7A4KON0Dbrhn2Q20+gAytSxIzaCX4UJJ6u8DdQr7VtZ
        vp6pRzw2hH1hAgMBAAGjITAfMBOGA1UdDgQWBBSM5CXRb1uRxrDqvFX7bLe7yX0WUzANBgkqhkiG
        9w0BAQsFAAOCQAQEAEBJJ/yux4o6QHysfbGkETQjVhrUArson0x4/am3jHihSIXFZMMPOz0M366Iv
        iirct13jnvU+Clpf8hI7AJtW+2QTHvINOM0sAXcmEuWSN4LW9e1NNy0r1Wqg3Dr/165UabuHQVx
        S7WbemCnM6hB+Y0XMUFQTYDisS5dazExI+g52zLacmBIRzvhjFoX3NUuClufEm0oyo1iIGk2CP/i
        G1c0kIjOP+JJ1aAi3wuqIAbksQ7xj19v5yxbHHOEDNZd1214dXU1nvD2T4CK00o1IzI/dhQnf6Jz
        N4nPnXPB2arrd/jufg9ppEkKbjcWRX25Y3L4j3MwUjUabJsGeH6NQQ==
      </X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>
</contentObject>
</EDXLDistribution>

```

Una vez que los nodos han sido estudiados, se han definido qué valores deberán de

contener y cuáles son necesarios para la correcta distribución del mensaje.

La estructura general del mensaje se muestra en las próximas dos figuras, EDXL y ESAP. En **negrita** se marcan los nodos que OASIS considera obligatorios y en verde los que se ha considerado necesarios para la correcta integración de los datos. Además, se añade una tabla correspondiente con cada imagen que define brevemente cada nodo.

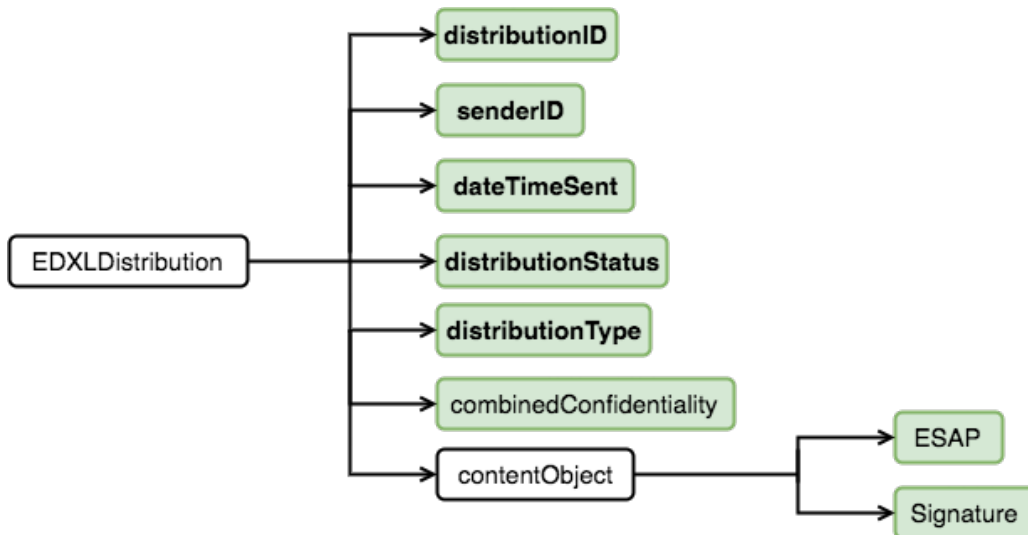


Figura 3.2: Composición del mensaje EDXL

EDXL	
<i>distributionID</i>	Identificación única del mensaje
<i>senderID</i>	Identificación del origen
<i>dateTimeSent</i>	Día y hora del envío
<i>distributionStatus</i>	Categorización de la información
<i>distributionType</i>	Función del mensaje
<i>combinedConfidentiality</i>	Confidencialidad de la distribución
<i>contentObject</i>	Información de datos de emergencias

Tabla 3.1: Definición de los nodos de EDXL

En cuanto a la cabecera de distribución EDXL, se han definido el valor de los siguientes nodos como predefinidos. Si no se reciben estos valores se rechazará el XML. Estos nodos son:

- **DistributionStatus:** tiene que estar definido como *Actual*.
- **DistributionType:** debe estar como *Report*.
- **CombinedConfidentiality:** debe recibirse como *CLASSIFIED AND SENSITIVE*.

Los mensajes que incorpora el protocolo ESAP especifican el detalle necesario de la emergencia como su tipología, su localización, anotaciones relevantes, información de quién alerta, datos de los cuerpos operativos implicados, su estado, etc.

Existen varios tipos de mensajes ESAP, sin embargo, en el ámbito de este proyecto el único nodo que se necesita es *msgIncidentUpdate*, que servirá para comunicar los cambios que puedan sufrir los recursos de un organismo.

Este tipo nodo es el que empleará el organismo para poder notificar cambios sucedidos. El uso habitual de este nodo es para la actualización de los datos de un incidente. Sin embargo, en el proyecto, se utilizará para la actualización de los datos de los recursos sin que exista referencia alguna a un incidente.

Este mensaje tiene la siguiente estructura:

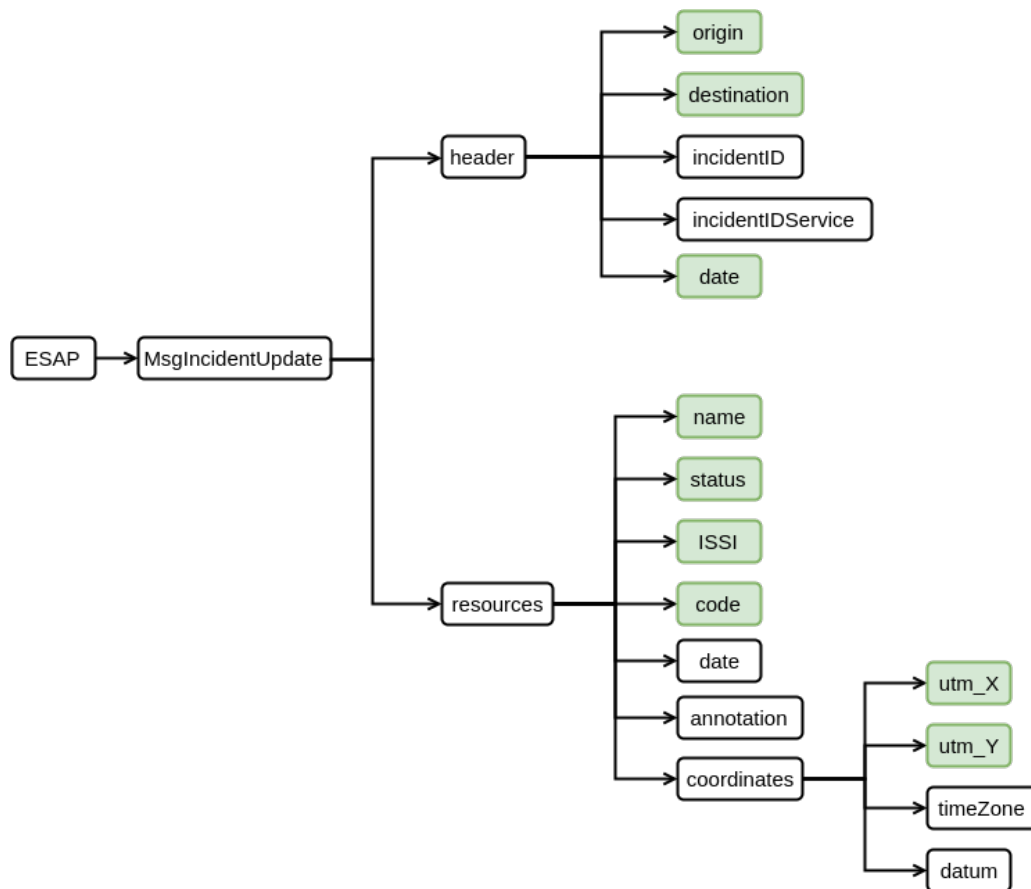


Figura 3.3: Composición del mensaje ESAP

3. ANÁLISIS

ESAP	
origin	Entidad que origina el mensaje
destination	Entidad destino final del mensaje
incidentID	vacio, no existe incidente
incidentIDService	vacio, no existe incidente
date	Fecha y hora del mensaje
name	Nombre del recurso
status	Estado del recurso
ISSI	Identificador de la radio del recurso
code	Código del recurso
date	Fecha y hora del cambio de estado del recurso
annotation	Anotación introducida en el cambio de estado
utm_X	Coordenada UTM X de la localización del recurso
utm_Y	Coordenada UTM X de la localización del recurso
timeZone	Uso de las coordenadas
datum	Elipsoide de referencia de las coordenadas

Tabla 3.2: Definición de los nodos de ESAP

3.4.2. Cálculo de la ruta óptima

Uno de los objetivos de este proyecto es determinar cual es la unidad preferible para cada incidente. Para ello se ha planteado las siguientes alternativas con diferentes costes:

- **OpenStreetMap (OSM) y pgRouting[17]:** gracias a OSM es posible tener un Shapefile (SHP) con todas las carreteras de las Islas Baleares. A partir de éste y con una consulta de pgRouting se puede calcular la ruta más cercana entre dos puntos, teniendo en cuenta la estructura de las carreteras de una manera totalmente gratuita. Sin embargo, OSM no almacena las direcciones de las calles, por lo que el resultado obtenido en la consulta podía ser totalmente erróneo. Se investigó la manera de obtener dichas direcciones por otras fuentes pero fue imposible.
- **API Google Maps:** a través de las APIs de Google Maps es posible obtener la ruta óptima, teniendo en cuenta el sentido de las calles. Adicionalmente, esta aplicación también tiene en cuenta el tráfico en vivo, por lo que, de este modo, se tiene la certeza de elegir el camino correcto. No obstante, esta solución tiene un coste asociado, una vez superado un determinado límite, la empresa te cobra por cada consulta.

El hecho de no conocer el sentido de las carreteras puede provocar que la ruta calculada sea errónea debido a que la ruta en un sentido puede ser suficientemente mayor que en el sentido contrario, siendo otra unidad la óptima.

Puesto que es necesario conocer con exactitud cual es la unidad óptima en cada caso, se descarta la primera solución.

Para conocer cual es la unidad que se encuentra más cerca de un incidente se deben seguir los pasos siguientes:

1. Distancia al incidente:

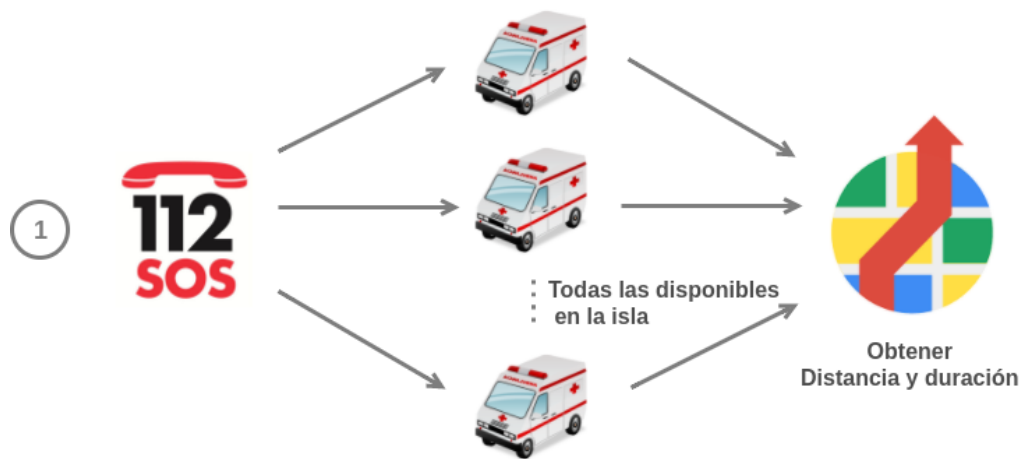


Figura 3.4: Paso 1: obtener las unidades más cercanas

El primer paso es conocer la distancia y la duración de todas las unidades disponibles en la isla donde sucede el incidente. Para ello, se accederá a la *Distance Matrix API* [18] pasándole la latitud y longitud de cada una de las ambulancias y la del incidente. Para cada petición se puede enviar un máximo de 25 puntos origen, así pues, habrá ocasiones que se deba hacer más de una petición.

A continuación se muestra un ejemplo de uso de *Distance Matrix API*:

- La petición: https://maps.googleapis.com/maps/api/distancematrix/json?origins=39.558734,2.624371|39.569425,2.641172|39.543247,2.386299&destinations=39.563732,2.619361&key=seib_key
- La respuesta:

```
{
  "destination_addresses" : [
    "Carrer Castell de Bellver, 07014 Palma, Illes Balears, España"
  ],
  "origin_addresses" : [
    "Avinguda de Joan Miró, 89-91, 07015 Palma, Illes Balears, España",
    "Avinguda de Gabriel Roca, 07012 Palma, Illes Balears, España",
    "Av. Almirante Riera Alemany, 15a, 07157 Port d'Andratx, Illes ↔
    Balears, España"
  ],
  "rows" : [
    {
      "elements" : [
        {
          "distance" : {
            "text" : "3,3 km",
            "value" : 3281
          },
          "duration" : {
```

```
        "text" : "12 min",
        "value" : 731
      },
      "status" : "OK"
    }
  ]
},
{
  "elements" : [
    {
      "distance" : {
        "text" : "3,4 km",
        "value" : 3359
      },
      "duration" : {
        "text" : "12 min",
        "value" : 748
      },
      "status" : "OK"
    }
  ]
},
{
  "elements" : [
    {
      "distance" : {
        "text" : "34,1 km",
        "value" : 34149
      },
      "duration" : {
        "text" : "40 min",
        "value" : 2415
      },
      "status" : "OK"
    }
  ]
}
],
"status" : "OK"
}
```

2. Obtener las rutas:

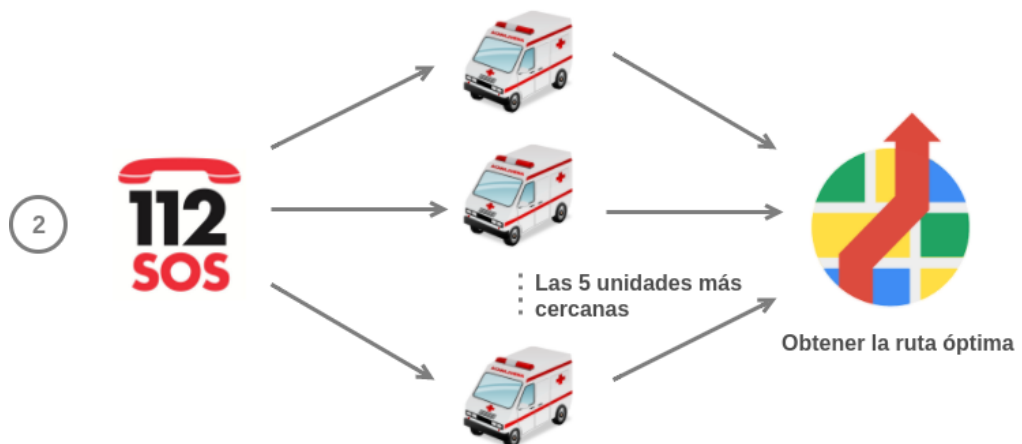


Figura 3.5: Paso 2: obtener los puntos de referencia de la ruta

Una vez obtenidas la distancia y la duración de todas las unidades disponibles, se debe hacer otra consulta para obtener la ruta de de las 5 unidades más cercanas. En este caso, se tiene que realizar una petición por unidad. Se empleará el servicio web *Directions API*[19] para conocer las distintas rutas. Dicho servicio devuelve los puntos claves y las intersecciones de la ruta.

- La petición: https://maps.googleapis.com/maps/api/directions/json?origin=39.558734,2.624371&destination=39.563732,2.619361&key=seib_key
- La respuesta:

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "EKBBdmluZ3VkYSBkZSBKb2FuIE1pczOzLCA4OS05",
      "types" : [ "street_address" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIjy1NXI3WSlxR9CMmhmLocvE",
      "types" : [ "route" ]
    }
  ],
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : 39.5692896,
          "lng" : 2.6288119
        },
        "southwest" : {
          "lat" : 39.5587349,
          "lng" : 2.6150322
        }
      },
      "copyrights" : "Datos de mapas 2016 Google, Inst. Geogr. Nacional",
      "legs" : [
        {
          "distance" : {
            "text" : "3,3 km",
            "value" : 3281
          },
          "duration" : {
            "text" : "12 min",
            "value" : 731
          },
          "end_address" : "Carrer Castell de Bellver, 07014 Palma, ←  
Illes Balears",
          "end_location" : {
            "lat" : 39.5636408,
            "lng" : 2.6199732
          },
          "start_address" : "Avinguda de Joan Miró, 89-91, 07015 Palma, ←  
Illes Balears",
          "start_location" : {
            "lat" : 39.5587349,
            "lng" : 2.6243797
          },
          "steps" : [
            {
              "distance" : {
```

```
        "text" : "0,7 km",
        "value" : 728
    },
    "duration" : {
        "text" : "2 min",
        "value" : 116
    },
    "end_location" : {
        "lat" : 39.5650294,
        "lng" : 2.6263654
    },
    "start_location" : {
        "lat" : 39.5587349,
        "lng" : 2.6243797
    },
    "travel_mode" : "DRIVING"
},
{
    "distance" : {
        "text" : "66 m",
        "value" : 66
    },
    "duration" : {
        "text" : "1 min",
        "value" : 13
    },
    "end_location" : {
        "lat" : 39.5651626,
        "lng" : 2.6271087
    },
    "start_location" : {
        "lat" : 39.5650294,
        "lng" : 2.6263654
    },
    "travel_mode" : "DRIVING"
},
],
"traffic_speed_entry" : [],
"via_waypoint" : []
}
],
"summary" : "Avinguda de Joan Miró y Carrer Castell de Bellver",
"warnings" : [],
"waypoint_order" : []
}
},
"status" : "OK"
}
```

3.4.3. Visualización de las rutas

En el Capítulo 2 se han expuesto las tecnologías usadas en el proyecto. Algunas de éstas se utilizan para representar las rutas sobre un mapa.

A continuación, se muestran las tecnologías que permiten disponer de una aplicación escritorio que proporciona los mapas necesarios junto con las geolocalizaciones de los elementos deseados como la posición de las ambulancias, el lugar del incidente y las líneas que indican la ruta óptima.

Así pues, para poder visualizar lo comentado, será necesario tener instalado en el

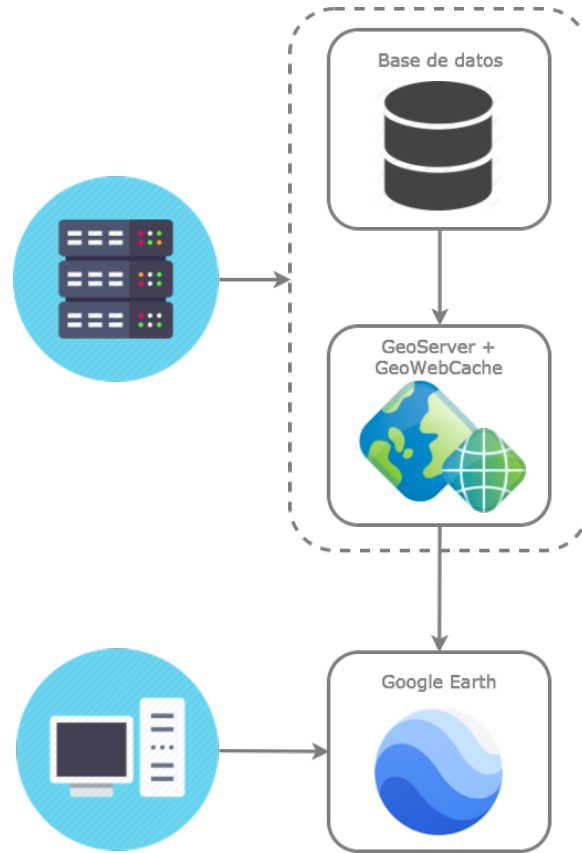


Figura 3.6: Tecnologías empleadas para la visualización de las rutas

ordenador la aplicación de *Google Earth*. Por otra parte, al servidor que contiene los datos geográficos se deberá instalar *GeoServer*.

Los datos que contienen dicho servidor son sensibles, por lo que, deben estar protegidos y ser únicamente visible para los operadores de guardia. Para ello, la solución optada es otorgar permisos de red a los ordenadores de la sala de operaciones.

DISEÑO

Esta sección contiene el caso de uso de un operador, la explicación de la arquitectura del sistema y el diseño de la base de datos. Además de definir el modo en el cual se realizarán las características más importantes del programa.

4.1. Caso de uso

La siguiente imagen define cuál es el comportamiento de un operador:

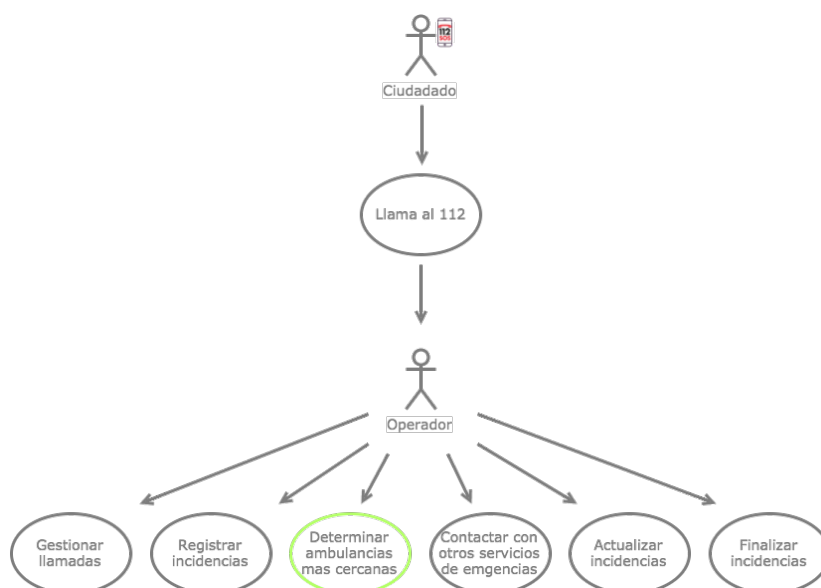


Figura 4.1: Casos de uso

Todas las circunferencias con el borde de color gris ya se encuentran implementadas. Este trabajo se centra en la circunferencia con el borde verde, como ya se ha comentado anteriormente, el objetivo de este TFG es determinar cuál es la ambulancia idónea para cada incidente que se produce en las Islas Baleares.

4.2. Arquitectura

En la Figura 4.2 se muestra la arquitectura del problema. Posteriormente, se descompone en dos vistas parciales de la arquitectura, donde se entra más en detalle.

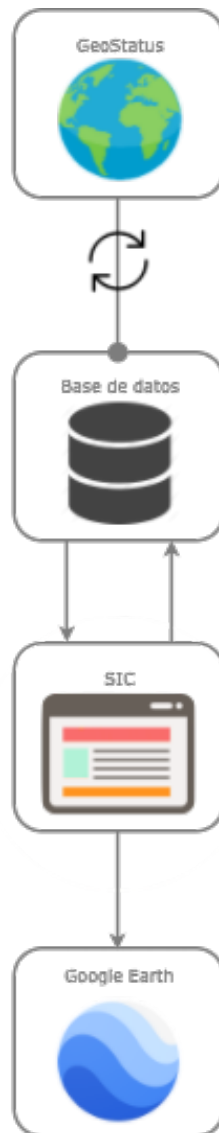


Figura 4.2: Arquitectura del sistema

A continuación, se definen las bases de datos que mantienen relación con el proyecto:

- **STORMMA:** se encarga de almacenar toda la información referente a las diferentes unidades sea cual sea su organismo. Entre los datos que almacena destacan el organismo al que pertenece, el estado de la unidad, la localización, última actualización de datos o el identificador de la unidad entre otros.

El programa encargado de la gestión de incidencias, explicado en el capítulo 1, se nutre de los datos aportados por la base de datos STORMMA para administrar las diferentes incidencias.

- **TIB:** una de sus tablas se emplea para almacenar la isla de las Islas Baleares donde se ubican las unidades.
- **ISEX:** almacena toda la información referente a la gestión del archivo XML. Una de las funciones de esta base de datos es evitar la duplicidad de datos, comprobar la veracidad de los datos o registrar el estado del programa GeoStatus.
- **GEOGPS:** se caracteriza por ser una base de datos PostgreSQL, este tipo de base de datos facilita la gestión de los datos de geolocalización. Esta base de datos permite ubicar en qué isla se encuentra una unidad, realizar el cálculo de la distancia entre dos puntos de interés para el servicio, guardar la información obtenida a través de la API de google junto con el almacenaje de la información más relevante de las distintas unidades. El *GeoServer* utiliza la información proporcionada por la base de datos *GEOGPS* para ser capaz de situar la unidad en el mapa de Google Earth. Mediante esta localización las unidades se posicionan en tiempo real sobre el mapa y se les indica cuál es la ruta adecuada para llegar a su destino.

4.2.1. Programa GeoStatus

GeoStatus es el nombre del servicio web que se encarga de actualizar la posición y/o el estado de las distintas unidades recibidas a través de los XML siguiendo el protocolo EDXL. A continuación, en la Figura 4.3 se muestra el diagrama de la arquitectura de manera abstracta.

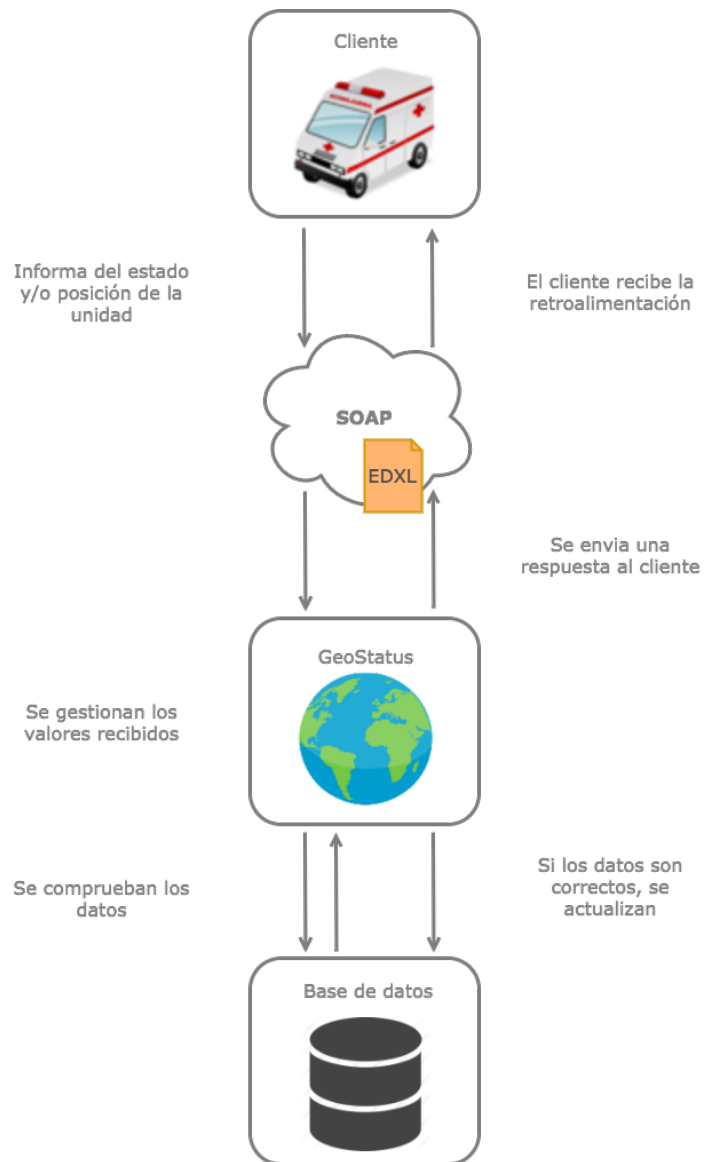


Figura 4.3: Diagrama de arquitectura de GeoStatus

Modelo de datos

A continuación, se definen las tablas que se utilizan en el GeoStatus. Además, se añade la Figura 4.4 para facilitar la comprensión.

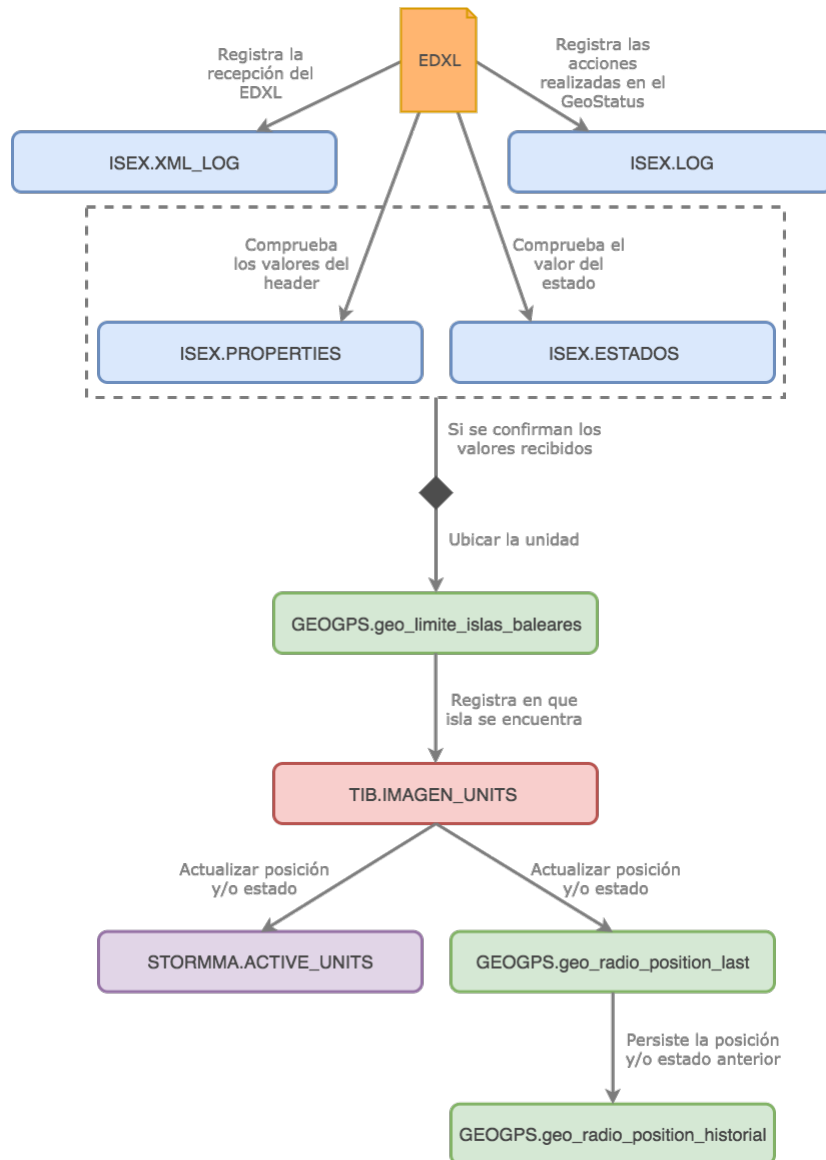


Figura 4.4: Modelo de datos relacionado con el GeoStatus

ISEX

Las tablas que se muestran a continuación pertenecen a la base de datos *ISEX*.

La siguiente tabla se utiliza para comprobar que los datos del archivo XML son los correctos. Para llevar a cabo este proceso se comprueba que los datos extraídos del XML coincidan con lo que alberga dicha tabla. Algunos campos de esta tabla actúan de mapeador mediante consultas SQL, convirtiendo los parámetros de entidad recibida

4. DISEÑO

por un sistema de emergencias externo en parámetros conocidos por nuestro sistema. El resto de campos se emplean solamente para validar.

Por ejemplo, para obtener el valor utilizado por el SEIB112 que hace referencia al campo *ORIGIN* del XML, se deberá de buscar que el *VALUE* de *PROPERTY_NAME* coincida y una vez que se tiene la instancia *INST_X* a la cual hace referencia, obtener el *ORIGIN_CAD_AGENCY*. Éste último campo es el valor empleado por nuestro sistema.

En la Tabla 4.1 se ilustra un ejemplo de los parámetros que contiene esta tabla para cada uno de los diferentes organismos.

ISEX.ISEX_PROPERTIES		
INSTANCE_NAME	PROPERTY_NAME	VALUE
INST_061	ORIGIN	Parámetro de entrada del XML
INST_061	ORIGIN_CAD_AGENCY	Valor utilizado por el sistema
INST_061	DESTINATION	Parámetro de entrada del XML
INST_061	DESTINATION_CAD_AGENCY	Valor utilizado por el sistema
INST_061	EDXL_SENDER_ID	Parámetro a validez del XML
INST_061	WORKSTATION	Parámetro a validez del XML
INST_061	CLAVE_PUBLICA_FIRMA	Clave pública del XML
INST_061	DATOS_USUARIO_FIRMA	Usuario del envío XML

Tabla 4.1: Tabla:ISEX.ISEX_PROPERTIES

La Tabla 4.2 ISEX_ESTADOS, es parecida a la anterior, opera como nexo de codificación entre el 112 y el resto de organizaciones. En esta tabla se almacenan los diferentes posibles valores recibidos a través del XML correlacionados al estado utilizado en nuestro sistema. Así pues, previamente se han definido las codificaciones entre los diferentes organismos y el SEIB112.

ISEX.ISEX_ESTADOS	
AGENCY_CODE	Nombre de la organización
ID_ESTADO	Estado recibido por la organización
ID_ESTADO_RELACIONADO	Estado relacionado empleado en nuestro sistema

Tabla 4.2: Tabla ISEX.ISEX_ESTADOS

La Tabla 4.3, ISEX_XML_LOG se utiliza para registrar cada XML recibido. Además, mediante esta tabla será posible conocer si a ya se ha obtenido un XML con un determinado *DISTRIBUTION_ID*. En este caso, el campo *TIPO_XML* siempre deberá ser *informe*. Además, el hecho de que sean las agencias las que envían el XML el campo *AGENCY_DEST* debería ser SEIB112.

ISEX_XML_LOG	
DISTRIBUTION_ID	Clave primaria del XML
XML_BODY	Cuerpo delXML recibido
AGENCY	Agencia de origen el XML
AGENCY_DEST	Agencia de destino del XML
TIPO_XML	Objetivo del XML
FECHA	Fecha en la que se ha recibido el XML

Tabla 4.3: Tabla ISEX_XML_LOG

La tabla Tabla 4.4 ISEX_LOG se emplea para documentar la sucesión de hechos e informar de los errores que han tenido lugar durante la ejecución del programa *GeoStatus*.

ISEX.ISEX_LOG	
FECHA	Fecha en la que se ha recibido el mensaje
HORA	Hora en la que se ha recibido el mensaje
ID_INCIDENTE	Identificador del incidente
ACCION	Referente al suceso que se quiere registrar
TEXTO	Información sobre la acción

Tabla 4.4: Tabla ISEX.ISEX_LOG

STORMMA

La tabla que se muestra a continuación pertenecen a la base de datos STORMMA.

La Tabla 4.5 STORMMA.ACTIVE_UNITS es la encargada de almacenar la información de cada unidad, visualizada en SIC. Los parámetros que definen una cierta unidad son los que aparecen en dicha tabla.

STORMMA.ACTIVE_UNITS	
AGENCY_CODE	Clave primaria nombre del organismo
UNIT	Clave primaria característica de la unidad
S_STATUS	Estado en el que se encuentra la unidad
MOD_DATE	Fecha de la última actualización
MOD_TIME	Hora de la última actualización
ISR_NO	Número de incidente asignado
LONGITUDE	Punto del eje x donde se encuentra la unidad
LATITUDE	Punto del eje y donde se encuentra la unidad

Tabla 4.5: Tabla STORMMA.ACTIVE_UNITS

TIB

La siguiente tabla se encuentra en la base de datos TIB.

La Tabla 4.6 TIB.IMAGEN_UNITS almacena la isla en la que se encuentra cada unidad. Esta tabla, también se necesita para la obtención de la ruta óptima.

TIB.IMAGEN_UNITS	
UNIT	Clave primaria característica de la unidad
ISLA	Nombre de una isla donde se encuentra la unidad

Tabla 4.6: Tabla STORMMA.ACTIVE_UNITS

GEOGPS

Las tablas que se muestran a continuación pertenecen a la base de datos GEOGPS.

La Tabla 4.7 geo_limite_islas_baleares contiene el contorno de cada isla, de este modo, es posible conocer si una localización determinada se encuentra en una de ellas y en cuál.

geo_limite_islas_baleares	
isla	Nombre de una Isla Balear
geom	Perímetro de la Isla Balear

Tabla 4.7: Tabla geo_limite_islas_baleares

La Tabla 4.8 geo_radio_position_last mantiene actualizados los datos de cada unidad. Esta tabla se utiliza para visualizar mediante el *GeoServer* la posición en tiempo real de todas las unidades.

La tabla geo_radio_position_historial es la encargada de almacenar los datos antiguos de posición además de otros datos de interés. Contiene los mismos campos que la Tabla 4.8. De esta manera el acceso a la posición en tiempo real de las unidades es mucho más rápido.

Capas de abstracción

Durante el desarrollo del programa *GeoStatus* y su integración con el SIC se deberá de tener en cuenta tres niveles distintos de arquitectura. Estos niveles son para tener un mayor control y seguridad en el sistema.

Los niveles son los siguientes:

- **Primer nivel:** grupo de funciones de alto nivel, serán las únicas que se podrán consultar desde el exterior. Éste sólo se efectuará en el programa SIC.

geo_radio_position_last	
id_radio_position	Clave primaria
issi	Número identificativo de la radio que pertenece a la unidad
longitude	Punto del eje x donde se encuentra la unidad
latitude	Punto del eje y donde se encuentra la unidad
altitude	Punto del eje z donde se encuentra la unidad
unit	Nombre identificativo de la unidad
unit_agency	Nombre de la agencia que pertenece la unidad
personnel_id	Código del personal de la unidad
hora	Fecha de la última actualización
punto	Conversión de la longitud, latitud y altitud en un punto
isr	Código del incidente
estado	Estado de la unidad respecto al incidente

Tabla 4.8: Tabla geo_radio_position_last

- **Segundo nivel:** tiene como objetivo administrar y gestionar las instrucciones de base de datos.
- **Tercer nivel:** la conexión con la base de datos para ejecutar las instrucciones en la base de datos.

4.2.2. Cálculo de la ruta óptima y la visualización

La Figura 4.5 muestra el diseño de la interacción del SIC con el resto de elementos que hace posible la visualización de las rutas.

En el momento que un operador consulta las rutas de las unidades cercanas, se realiza una serie de peticiones a las APIs de *Google Maps*. Una vez obtenida toda la información necesaria, se pasa un KML con las rutas obtenidas al *GIS Manager* y éste se encarga de definir la visualización a mostrar.

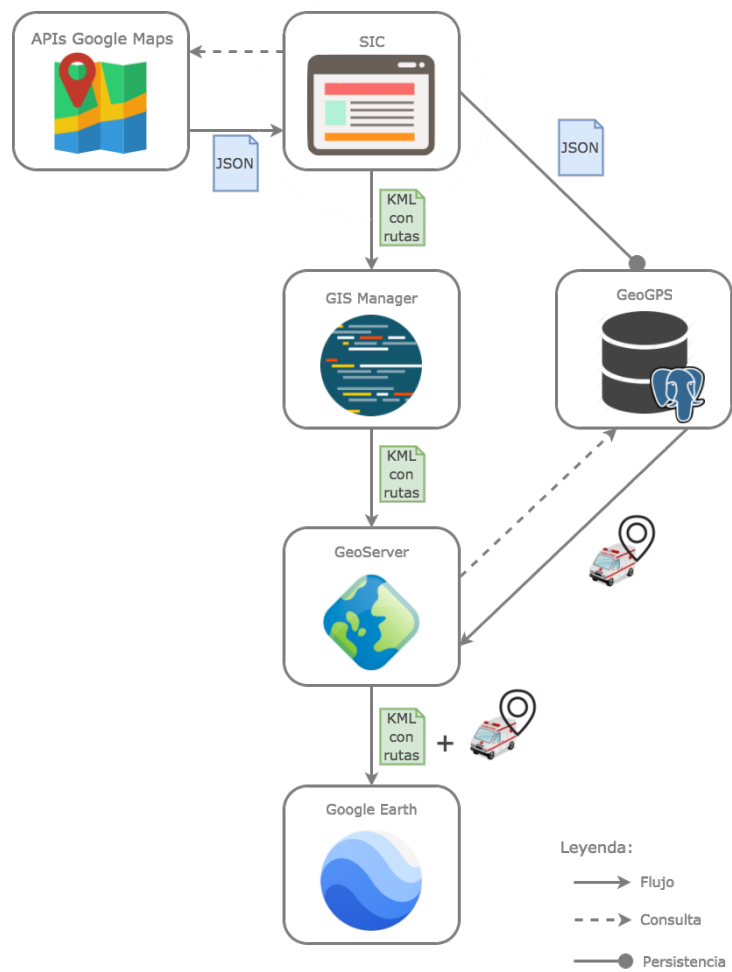


Figura 4.5: Diagrama de arquitectura del cálculo de la ruta óptima y la visualización

Modelo de datos

Del mismo modo que anteriormente, a continuación se definen las tablas que se utilizan para el cálculo de la ruta óptima. Además, se añade la Figura 4.6 para facilitar la comprensión.

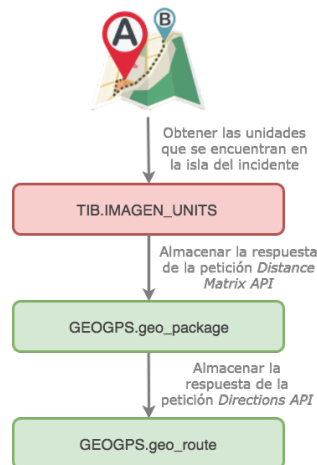


Figura 4.6: Modelo de datos relacionado con el GeoStatus

Las dos siguientes tablas se almacenan, como ya se ha comentado en el RNF-8, con la finalidad de tener un registro de los cálculos obtenidos. La Tabla 4.9 almacena el JSON recibido junto a incidente relacionado.

geo_package	
id_geo_package	Clave primaria autoincremental
personnel_id	Id de la persona que realizar la consulta
fecha	Fecha en la que se realiza la petición a la API
hora	Hora en la que se realiza la petición
isr	Incidente al cual se le solicita calcular las rutas
json	Cuerpo del JSON recibido

Tabla 4.9: Tabla geo_route

El objetivo de la Tabla 4.10 es guardar la distancia y el tiempo obtenidos para poder mostrarlos en el SIC, además del archivo JSON proporcionado por la API de *Google Maps*.

geo_route	
id_geo_route	Clave primaria autoincremental
id_geo_package	Clave foránea de la tabla geo_package
unit	Nombre identificativo de la unidad
unit_agency	Nombre de la agencia que pertenece la unidad
issi	Número identificativo de la radio que pertenece a la unidad
distancia	Distancia entre la unidad y el incidente relacionado
tiempo	Tiempo entre la unidad y el incidente relacionado
fecha	Fecha en la que se realiza la petición a la API
hora	Hora en la que se realiza la petición
linea	Línea de la ruta entre el punto origen y destino
json	Cuerpo del JSON recibido

Tabla 4.10: Tabla geo_route

4.3. Infraestructura

A continuación se muestra la Figura 4.7 donde se muestra el diagrama de la infraestructura del entorno del SEIB112 relacionada con este proyecto.

El archivo XML retransmitido por los organismos de emergencias se recibe a través del servicio web. Este servicio web se encarga de actualizar los datos recibidos por el organismo de emergencia correspondiente en las diferentes bases de datos.

En el SEIB112 existen diferentes sistemas de gestión de bases de datos. Por un lado, la información referente a la geolocalización se almacenará en una base de datos PostgreSQL. Por otro lado, el resto de datos se guardaran en una base de datos Oracle. En total, se va a trabajar con cuatro bases de datos encargadas de almacenar la información característica de la unidad del organismo.

Al tener los datos actualizados, el SIC es capaz de mostrar de manera ordenada las unidades más cercanas al incidente.

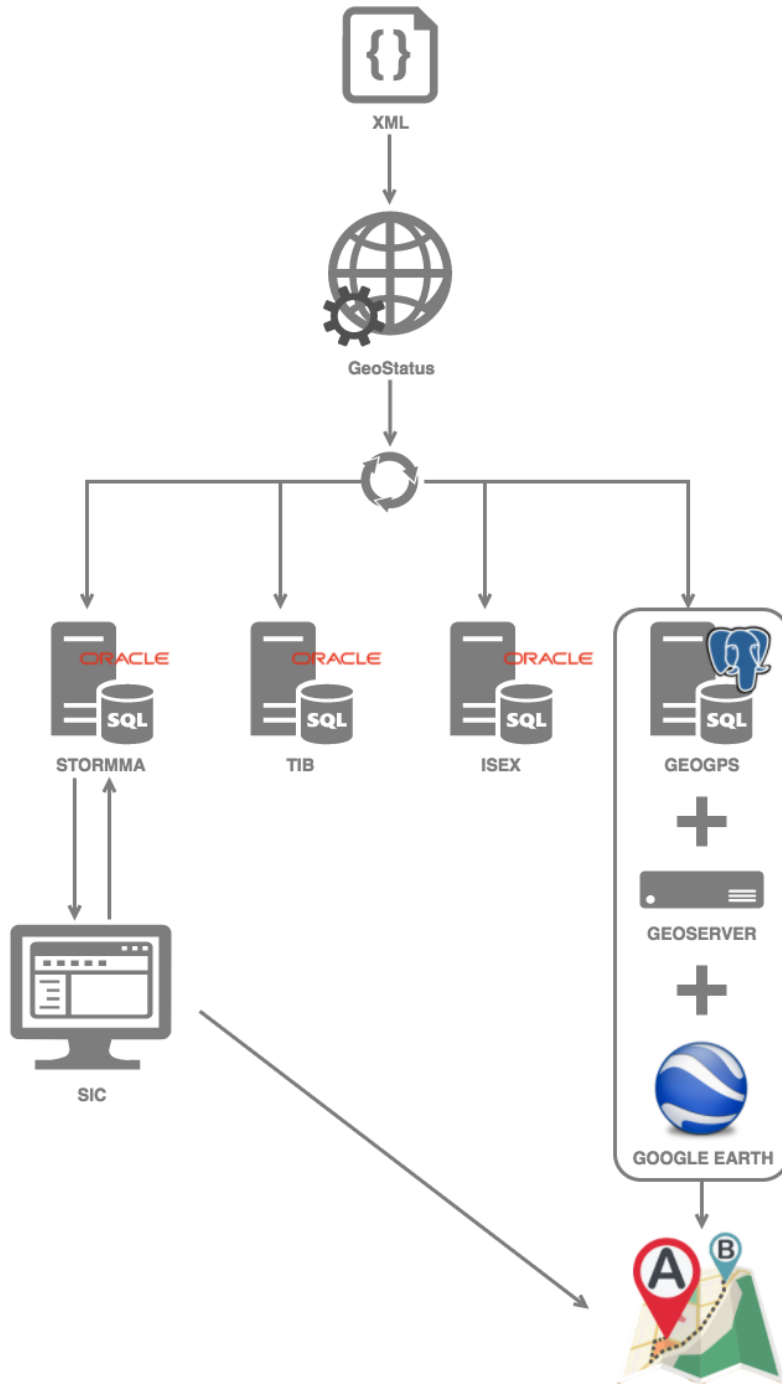


Figura 4.7: Infraestructura del proyecto

IMPLEMENTACIÓN Y RESULTADOS

En este quinto capítulo se aborda la fase de desarrollo del proyecto junto a los resultados obtenidos. En el mismo, se explican algunos detalles relevantes así como las diferentes dificultades que se han encontrado en el proceso. Esta fase consta de la implementación del diseño realizado en el Capítulo 4, el cual está basado en el catálogo de requisitos del Capítulo 3.

A lo largo del capítulo se tratarán los siguientes temas:

- La jerarquía de directorios del proyecto *GeoStatus* y SIC.
- Aspectos relevantes de la implementación del *GeoStatus*.
- Características significativas de la nueva funcionalidad añadida en el proyecto SIC.

5.1. Programa *geoStatus*

5.1.1. Jerarquía de directorios

En la siguiente Figura 5.1 se muestran las carpetas y ficheros que componen el proyecto *geoStatus*.

La carpeta *GeoStatus-ejb*, contiene toda la lógica del servicio web, mientras que la carpeta *GeoStatus-war* se encarga de realizar la comunicación con el EJB y el exterior.

Posteriormente se detalla cada fichero mostrando partes del código o el pseudocódigo.

5. IMPLEMENTACIÓN Y RESULTADOS

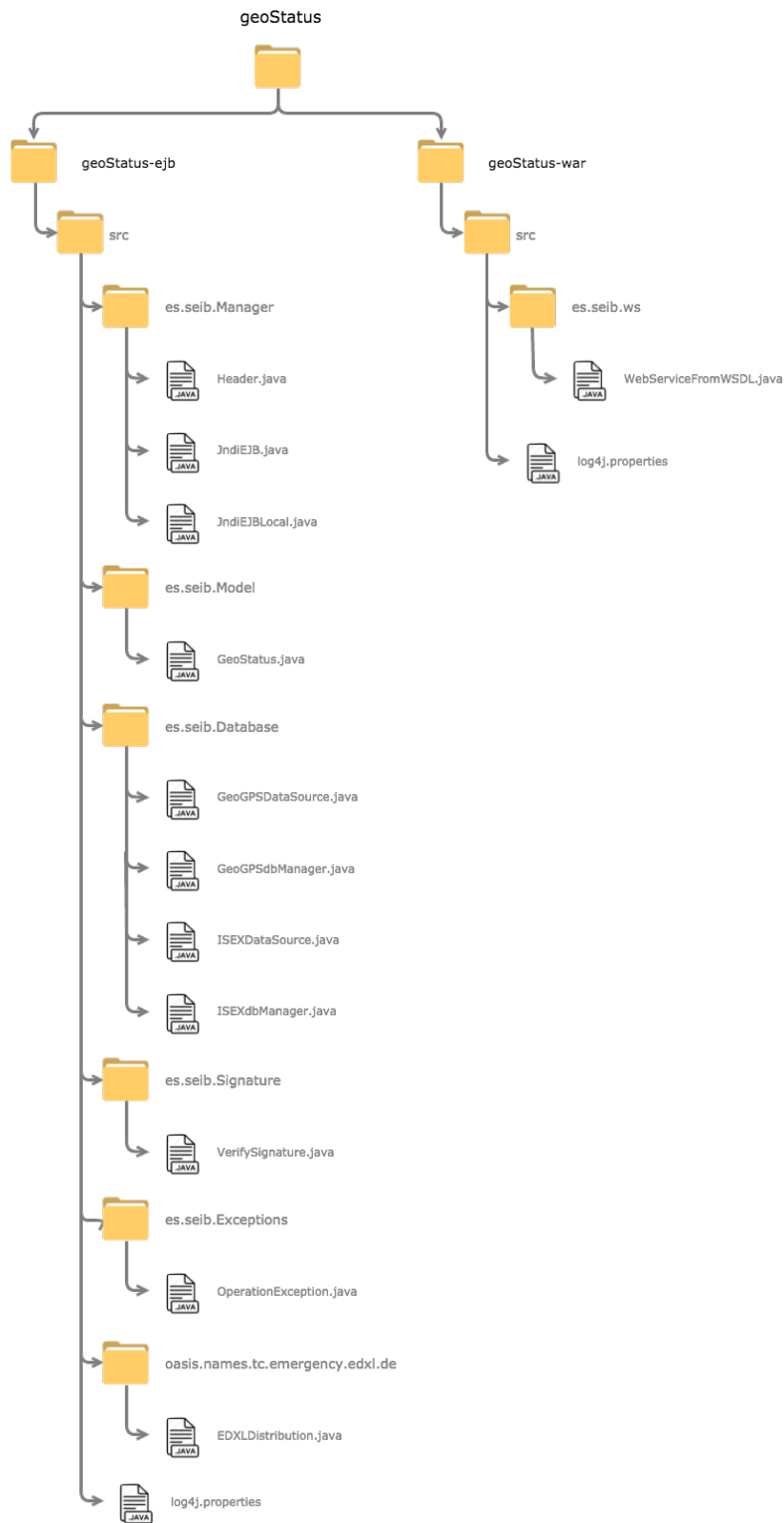


Figura 5.1: Jerarquía de ficheros de geoStatus

En la siguiente lista se describe la funcionalidad de cada paquete del proyecto:

- *Manager*: verifica la validez del objeto *EDXLDistribution* recibido y lo gestiona realizando llamadas a los ficheros *dbManager* de la carpeta *Database*.
- *Model*: *GeoStatus* contiene la representación de la información necesaria sobre una unidad. Se encarga de gestionar todos los accesos a dicha información, tanto consultas como actualizaciones.
- *Database*: se encarga de insertar, actualizar y eliminar ciertas filas de las bases de datos introducidas en la Figura 4.7.
- *Signature*: manipular instancias de la clase *XMLSignature* para poder verificar firmas.
- *Exceptions*: contiene la excepción utilizada para reportar errores.
- *oasis.names.tc.emergency.edxl.de*: contiene una clases para cada etiqueta del XML. Con el fin de reducir el tamaño de la imagen, se ha optado por solo mostrar la clase que más se emplea, *EDXLDistribution*.
- *WebService*: permite realizar el acceso al *Web Service GeoStatus-ejb*.
- *log4j.properties*: son los ficheros de configuración de la librería *Log4J*, con la cual es posible registrar mensajes de información de lo que está sucediendo en la aplicación.

5.1.2. Especificación de la implementación

Una vez explicada la estructura se procede a explicar cada fichero:

- *Header*: se encarga del realizar el tratamiento de la cabecera del XML.

```

logger.info("Recibido un nuevo xml con DistributionID: " + DistributionID + ".");

try {

    if (DistributionID no esta registrado en nuestra bbdd) {
        String XML = se convierte el objeto en XML;
        GeoStatus gs = se inserta los parametros del XML en el objeto GeoStatus

    if (el origen y el destino estan registrados) {
        gs.setDestination(destino)

        if (el senderId esta registrado) {
            comprobar firma

            if (la firma es valida) {
                gs.setWorkstation(workstation)
                gs.setPersonalID(peronsal_id)
                gs.setLoginName(login_name)

                //Si es el tipo de mensaje que se ha definido
                if (DistributionStatus == "Actual"
                    && DistributionType == "Report"
                    && CombinedConfidentiality == "CLASSIFIED AND SENSITIVE") {

```

5. IMPLEMENTACIÓN Y RESULTADOS

```
        if (la unidad pertenece a la agencia pasada) {
            gs.setIsr(isr)
            insertar XML en nuestra bbdd
            logger.info("El header del XML es correcto.");
            return gs;
        }
    }
}
}
}
} catch () {
    reportar error
    return null;
}
```

- *JndiEJB*: es el programa principal del proyecto, a continuación se muestra el pseudocódigo.

```
GeoStatus gs = se realiza el tratamiento del header mediante la clase Header

if (gs != null) {
    if (gs contiene un estado valido) {
        actualizar estado
        if (gs contiene una posicion valida) {
            actualizar posicion
        }
    } else if (si el estado esta vacio y existe informacion de la posicion) {
        actualizar posicion sin cambiar de estado
    } else {
        reportar error, los parametros del estado o posicion son incorrectos
    }
} else {
    reportar error, se ha producido un error en el tratamiento del header
}

return ok;
```

- *JndiEJBLocal*: este fichero es necesario para indicar que no se implementa una interface de negocios para el cliente.
- *GeoStatus*: entidad que contiene los siguientes atributos junto a sus correspondientes getters y setters.

```
private final String HourFormat = "yyyyMMdd";
private final String TimeFormat = "HHmms";

private String distributionID = " ";
private String senderID = " ";

private String dateTimeSentHour = "yyyyMMdd";
private String dateTimeSentTime = "HHmms";

private String origin = " ";
private String destination = " ";

private String dateHour = "yyyyMMdd";
```

```

private String dateTime = "HHmmss";

private String status = " ";
private String ISSI = " ";
private String codeUnit = " ";

private Double utmX = 0.0;
private Double utmY = 0.0;
private Double coorGeoX = 0.0;
private Double coorGeoY = 0.0;
private String timeZone = " ";
private String datum = " ";
private String island = " ";

private String isr = " ";

private String loginName = " ";
private String workstation = " ";
private int personalID = 0;

```

- *GeoGPSDataSource*: realiza las peticiones a la base de datos PostgreSQL y las mapea en objetos que posteriormente se utilizan en las clase de más alto nivel (*GeoGPSdbManager*).

```

function getIncidentIsland (Connection c, Longitude, Latitude) {

    PreparedStatement statement = c.prepareStatement(
        "SELECT isla FROM geo_limite_islas_baleares "
        + "WHERE ST_Contains(geom, ST_SetSRID(ST_Point( ?, ?), 25831))");

    statement.setDouble(1, Longitude);
    statement.setDouble(2, Latitude);

    ResultSet query = statement.executeQuery();
    boolean next = query.next();

    if (query == null || !next) {
        return null;
    } else {
        return query.getString("isla");
    }
}

```

- *GeoGPSdbManager*: es la encargada de abstraer las transacciones a la base de datos de PostgreSQL.

```

function updatePositionGeoGPS (GeoStatus gs) {

    c = Recuperamos el pool de conexiones de __GeoGPS
    if (ya se encuentra registrado el gs.issi) { //consulta mediante GeoGPSDataSource
        eliminar la fila con dicho ISSI de geo_radio_position_last //mediante ←
        GeoGPSDataSource
    }
    Actualizar las tablas geo_radio_position_last y geo_radio_position_historial ←
    //mediante GeoGPSDataSource

    c.close() //cierra la conexion
}

```

5. IMPLEMENTACIÓN Y RESULTADOS

- *ISEXDataSource*: realiza las peticiones a la base de datos Oracle, concretamente a la tabla ISEX y las mapea en objetos que posteriormente se utilizan en las clase de más alto nivel (*ISEXdbManager*).

Las consultas y actualizaciones sobre la base de datos STORMMA y TIB se realizan desde el proyecto SIC, las cuales se comentarán en la próxima sección.

```
function unitPertainAgency(OriginAgencyCode, unit) {
    try {
        query = "Select AGENCY_CODE "
            + "from STORMMA.ACTIVE_UNITS where UNIT = ?";

        query.parametro(1, unit);

        return query == OriginAgencyCode;
    } catch (excepcion) {
        notificar la excepcion
    }
}
```

- *ISEXdbManager*: clase que gestiona y administra la tabla ISEX.
- *VerifySignature*: clase que comprueba la veracidad de firma recibida [20].
- *OperationException*: excepción propia.
- *WebServiceFromWSDL*: es el puerto de comunicaciones empleado para realizar las llamadas desde los organismos externos.

5.1.3. Librerías

Puesto que el SEIB112 lleva años trabajando con proyectos relacionados a éste, existen una serie de librerías que permiten que la carga de trabajo invertido en el programa *GeoStatus* sea menor.

A continuación se presentan las librerías empleadas en dicho programa:

- *Librería log4j*: permite registrar una determinada transacción en tiempo de ejecución.

La configuración empleada para el servicio web es la siguiente:

```
log4j.appender.archivo.File=./logs/GeoStatus-ejb.log
log4j.appender.archivo.append=true
log4j.appender.archivo.maxNumberOfDays=30
log4j.appender.archivo.encoding=ISO-8859-15
log4j.appender.archivo.DatePattern='yyyy-MM-dd
log4j.appender.archivo.layout=org.apache.log4j.PatternLayout
log4j.appender.archivo.layout.ConversionPattern=%d{ABSOLUTE} %p %c{1}:%L - %n%n
log4j.rootLogger=ALL, LOGFILE
```

- Mediante la primera línea de código se pretende dirigir los mensajes hacia el archivo *GeoStatus-ejb.log*.
- La configuración *append* indica que no reemplazará el fichero una vez finalizada la ejecución.
- Con la tercera línea de código delimitaremos el tamaño del fichero siendo 30 el máximo número de días. Una vez pasados dichos días se creará otro fichero con el mismo nombre pero añadiendo un identificador al final de éste.
- En cuanto a la ISO-8859-15 es un estándar de codificación de caracteres que incluye los caracteres acentuados.
- Como tipo de diseño de los mensajes se ha utilizado *org.apache.log4j.PatternLayout*.
- La variable *ConversionPattern* define como se mostrará cada fila de log.
 - La letra *d* indica la fecha y la hora de la salida del registro.
 - La letra *p* es la prioridad de la salida de registro de información, a saber, la depuración, INFO, WARN, ERROR, FATAL.
 - La letra *c* indica el nombre del fichero.
 - La letra *p* indica la línea del código.
 - La letra *m* es registro de información a anotar.
 - La letra *n* añade un salto de línea.
- *GeoConverter*: se emplea para realizar conversiones entre coordenadas cartesianas y polares y viceversa.
- *DateUtils*: proporciona una cadena de caracteres con la hora actual de la forma (hhmmss) o bien una cadena con el día de hoy (yyyymmdd).
- *XML*: esta librería se usa para captar los datos del servicio web para posteriormente ser manipulados en el objeto *geoStatus*. Esta librería es la única que no es propia del SEIB112.

5.2. Cálculo de la ruta óptima y la visualización

5.2.1. Contextualización

Una vez que ya se tiene la posición y el estado actualizado de todas las unidades disponibles es posible calcular cuál es la indicada para acceder a una localización determinada. Esta información se muestra en el proyecto SIC, el programa que se encarga de gestionar las incidencias. Para ello, se ha añadido un nuevo botón en la ventana principal, el cual, se encuentra rodeado en la Figura 5.2.

Una vez que se pulsa en el botón, se muestra la siguiente ventana (Figura 5.3), donde aparece la información de todas las unidades activas. Si en la ventana principal se ha seleccionado un incidente, en esta información aparece la distancia y el tiempo que existe entre la unidad y la ubicación del incidente.

5. IMPLEMENTACIÓN Y RESULTADOS

Incidentes activos

Estado	Código Incidente	Tipo Servicio	P...	Núcleo	Municipio	Dirección	Fecha	Hora	No l...	Aut...	Asignado	Llamante	Teléfono
Libre	SEIB 112-20160530-0001	ASISTENCIA CARRETERA	3	PALMAO	PALMA DE MALLORCA	AVENIDA GABRIEL ALOMAR I...	20-06-16	15:04:23	0	0	MCALZADA		971469176
Libre	SEIB 112-20160608-0007	ACCIDENT MAQUINARIA IND...	1	ALCLODIAS	ALCLODIA	ALCLODIA, 8 MA-3470	08-06-16	10:42:38	0	0	MONTALV		
Libre	SEIB 112-20160523-0003	DESAPAREGUTS MENOR	4	ALCLODIAS	ALCLODIA	PLATJA ALCLODIA, 0 ALCLODIA	07-06-16	12:55:02	6	6	MCALZADA	666689419	
Despach...	SEIB 112-20160520-0008	AERONAU DESAPAREGUDA	1	PALMAS	PALMA DE MALLORCA	PLATJA CALA ESTANCIA, 0 C...	30-05-16	14:32:00	0	0	MCALZADA	675618510	
Despach...	SEIB 112-20160520-0009	AERONAU DESAPAREGUDA	1	FORMENTER8	FORMENTERA	PLATJA CALO MORO, 0 ES CA...	30-05-16	14:31:34	2	2	MCALZADA		
Libre	SEIB 112-20160523-0002	ANIMALS ATIPATS	2	ELITFADELL8	ELITFADELLA DE MEN...	PLATJA SANTANDREA, 0 SANI...	27-05-16	17:48:59	6	6	MCALZADA		
Despach...	SEIB 112-20160525-0005	ACCIDENT SUBAQUATIC	1	ALCLODIAS	ALCLODIA	PLATJA ALCLODIA, 0 ALCLODIA	26-05-16	14:07:03	0	0	MCALZADA	666689419	
Libre	SEIB 112-20160523-0005	ACCIDENT CAMIO AMB FERITS	1	PALMAO	PALMA DE MALLORCA	AVENIDA ARGENTINA 24 S.L...	24-05-16	14:30:39	3	3	MCALZADA		
Despach...	SEIB 112-20160523-0004	INCENDI	1	PALMAO	PALMA DE MALLORCA	AVENIDA GABRIEL ALOMAR I...	23-05-16	15:26:29	0	0	MCALZADA	971469176	
Despach...	SEIB 112-20160523-0001	SUICIDI INTENT	1	ANDRATX8	ANDRATX	PLATJA CALA CONELLS, 0 CAL...	23-05-16	15:12:54	0	0	MCALZADA		

Figura 5.2: Programa SIC con el nuevo botón para obtener el posicionamiento

Tabla Unidades Activas

Lista Unidades Activas

I...	Unidad	Tipo de U...	Des...	Fecha	Hora	Incidente
	SIERRA-1	UOP	DESP...	26-05-16	11:32:08	SEIB 112-2016052
	TANGO-1	UOP	DISP...	21-06-16	11:27:21	
	TANGO-3	UOP	DESP...	17-05-16	09:39:02	SEIB 112-2014021
	TANGO-6	UOP	DISP...	16-05-16	12:04:01	
	TANGO-7	UOP	DESP...	30-05-16	14:31:34	SEIB 112-2016052
	TANGO-8	UOP	DESP...	30-05-16	14:32:00	SEIB 112-2016052
	BRAVO-21	UOP	DISP...	21-11-14	11:33:11	
	BRAVO-31	UOP	EN EL...	18-03-16	09:58:23	SEIB 112-2015040
	BRAVO-41	UOP	EN R...	18-03-16	09:58:33	SEIB 112-2015041
	BRAVO-61	UOP	DESP...	21-06-16	12:15:57	SEIB 112-2016061
	BRAVO-62	UOP	DISP...	31-10-14	12:24:48	
	BRAVO-71	UOP	DESP...	26-05-16	14:07:03	SEIB 112-2016052
	BRAVO-81	UOP	DISP...	18-03-16	09:59:46	
	BRAVO-82	UOP	DISP...	19-11-15	09:42:42	
	LIMA-2	UOP	DISP...	05-02-14	17:45:04	
	LIMA-3	UOP	DISP...	03-01-14	15:23:19	
	LIMA-5	UOP	DISP...	23-08-13	01:01:06	
	UCA DGE	UOP	DISP...	23-07-15	09:47:20	
	TANGO-4	UOP	DESP...	21-06-16	11:26:17	SEIB 112-2016050
	CAP-UOP	UOP	DISP...	09-07-15	11:41:08	
	BRAVO-12	UOP	DISP...	26-01-15	11:48:51	
	TANGO-5	UOP	DISP...	28-10-15	12:03:33	
	BRAVO-13	UOP	EN S...	29-01-16	11:56:19	

Figura 5.3: Modal SIC para obtener el posicionamiento

A parte de esta nueva funcionalidad también se han añadido los métodos necesarios

para realizar la gestión del cambio de estado de las unidades.

5.2.2. Jerarquía de directorios

En la imagen Figura 5.4 se muestran las carpetas y ficheros que se han modificado o creado para este proyecto.

Posteriormente se detalla cada fichero.

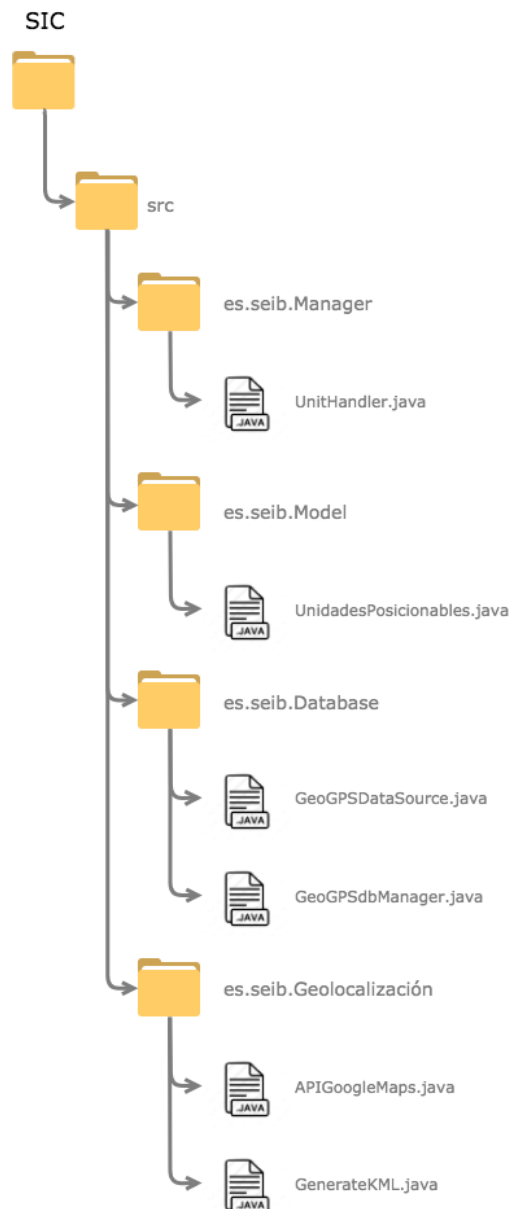


Figura 5.4: Jerarquía de ficheros del SIC relacionados con el proyecto

La funcionalidad de los paquetes del proyecto es la misma que en la Subsección 5.1.1, añadiendo la carpeta Geolocalización. La función de esta carpeta es gestionar

la parte de la obtención de rutas y pintado.

5.2.3. Especificación de la implementación

A continuación se realiza una explicación de cada fichero:

- *UnitHandler*: se encarga de la funcionalidad de la gestión del botón comentado en la Figura 5.2.
- *UnidadesPosicionables*: entidad que contiene los siguientes atributos junto a sus correspondientes getters y setters.

```
String unit;  
String agencyUnit;  
String fecha;  
String hora;  
Double longitude;  
Double latitude;  
String km;  
Double minutos;  
String issi;  
int personnel_id;  
int id_geo_package;
```

- *APIGoogleMaps*: realiza las peticiones Application Programming Interface (API) a *Google Maps* siguiendo las indicaciones de la Subsección 3.4.2 y la Subsección 4.2.2 y las parsea al objeto *UnidadesPosicionables*. Posteriormente realiza el guardado de las respuestas. Se accederá a los ficheros de esta carpeta en el momento que el usuario pulse el botón para conocer cuales son las ambulancias óptimas para un incidente, se realizan una serie de acciones.
- *GenerateKML*: convierte los puntos definidos en el objeto *UnidadesPosicionables* a KML.

Los ficheros *GeoGPSDataSource* y *GeoGPSdbManager* tienen la misma función que en el proyecto *GeoStatus* (Página 49).

Se ha decidido no se incluir código debido a que no es suficientemente relevante para el proyecto y carecen de complejidad. A continuación, se aclaran los aspectos destacados para conseguir el objetivo:

Conocer la isla donde se encuentra la unidad

Como ya se ha comentado antes, la obtención de la ruta óptima puede llegar a generar un coste si se excede del límite diario. Por ello, es necesario limitar el número de peticiones realizadas. Para acortar dichas consultas, se ha creado una tabla nueva en la base de datos TIB donde se almacena en qué isla se encuentra la unidad, de tal manera que sólo se realizará la petición con las unidades disponibles de la isla en cuestión.

Para conocer en qué isla se encuentra la unidad también se ha añadido una tabla en GEOGPS que contiene el perímetro de las islas, llamada *geo_limite_islas_baleares*.

Mediante la función *ST_Contains* de *PostGIS* [21] es posible conocer si el punto en el que se encuentra de la unidad se ubica dentro de uno de los perímetros registrados en la tabla *geo_limite_islas_baleares*. A continuación se muestra la consulta SQL para conocer en qué isla se encuentra el punto (latitud, longitud). Posteriormente este dato se almacena en la tabla *geo_radio_position_last* y *TIB.IMAGEN_UNITS*.

```
"  
SELECT isla FROM geo_limite_islas_baleares  
WHERE ST_Contains(geom, ST_SetSRID(ST_Point(latitude, longitud), 25831))");  
"
```

Dicha tabla se ha importado desde un Shapefile, el cual se visualiza a continuación.

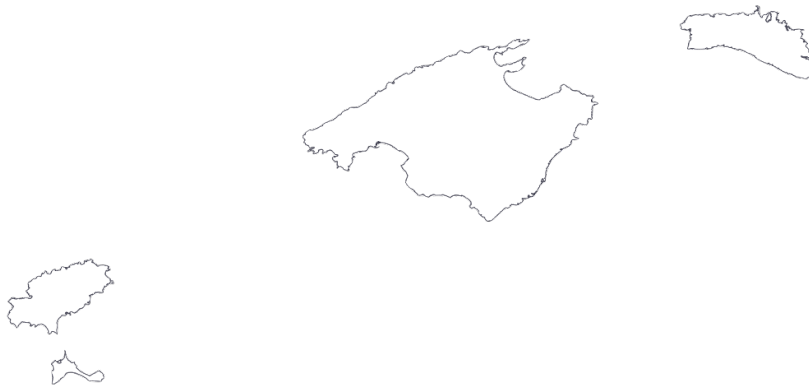


Figura 5.5: Contorno de las Islas Baleares

Visualizar las mejores rutas

Una vez que ya se tienen los puntos por donde debe circular la unidad, es posible generar un KML para situar en el mapa cada uno de los puntos recibidos. A continuación se muestra un ejemplo de KML sobre Palma.

Para conseguir que los puntos se muestren como una única línea se realiza mediante la etiqueta *<LineString>*.

```
<?xml version="1.0" encoding="UTF-8"?>  
<kml xmlns="http://earth.google.com/kml/2.2">  
  <Document>  
    <name>Java Generated KML Document</name>  
    <Placemark>  
      <name>Rute line</name>  
      <description>Unit X</description>  
      <Style>  
        <LineStyle>  
          <color>ffff0000</color>
```

5. IMPLEMENTACIÓN Y RESULTADOS

```
<width>4.0</width>
</LineStyle>
</Style>
<MultiGeometry>
  <LineString>
    <coordinates>
      2.6574603,39.5673229,0 2.6575276,39.5674559,0
      2.6568278,39.5643377,0 2.6811946,39.5643655,0
      2.6862712,39.5634674,0 2.6863722,39.5666203,0
      2.6861401,39.5702062,0 2.7099647,39.5683194,0
      2.9108193,39.5664146,0 2.9712433,39.5619126,0
      3.0688959,39.5704761,0 3.1204307,39.5752491,0
      3.1204307,39.5752491,0
    </coordinates>
  </LineString>
</MultiGeometry>
</Placemark>
</Document>
</kml>
```

A continuación, en la Figura 5.6 se muestra este KML sobre *Google Earth*.



Figura 5.6: Ejemplo de visualización de un KML

Este proceso se debe de repetir 4 veces más, para poder visualizar las 5 rutas óptimas.

Por lo tanto, de momento, ya se puede visualizar las rutas pero no es posible reconocer a qué unidad pertenece cada ruta. Para ello, es necesario el servidor de datos espaciales *GeoServer*. Este servidor tiene acceso a la base de datos *GeoGPS*, donde se encuentra el estado actual de cada una de las unidades. De esta manera, es posible visualizar los puntos iniciales de cada ruta.

A estos *puntos iniciales*, es decir, la localización de las unidades, se le debe de crear un estilo gráfico mediante un fichero Styled Layer Descriptor (SLD) donde se ha cambiado el punto inicial por una imagen de una ambulancia y el punto final, el incidente, se muestra mediante una rosa de los vientos, tal y como se muestra en Figura 5.7.



Figura 5.7: Ejemplo de visualización de un KML

Además de indicar ambos extremos de la ruta también se ha definido mostrar la distancia y el tiempo estimados de ésta. A continuación aparece una de las 5 reglas de estilos definidas para las rutas[22]. Cada una de estas reglas pinta en el mapa el tiempo y distancia de las unidades óptimas. En este caso, se muestra el estilo creado para los valores de la ruta KML azul. De este modo, los valores tiempo y distancia de una unidad se pintan del mismo color que la ruta, tal y como se muestra en la Figura 5.8.

Para ello, se ha creado la propiedad *type* donde se le define un valor a cada una de las 5 unidades. En este caso, cuando la propiedad *type* es igual a 0 se pintarán la información de color azul. Este valor se ha creado a propósito para poder pintar cada ruta de un color diferente. El resto de reglas son idénticas a ésta cambiando la comparación de *type* por otro valor para asignarle otro color.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/sld ↵
    StyledLayerDescriptor.xsd"
  xmlns="http://www.opengis.net/sld"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<!-- a Named Layer is the basic building block of an SLD document -->
<NamedLayer>
  <Name>default_line</Name>
  <UserStyle>
    <Title>Default Line</Title>
    <Abstract>A sample style that draws a line</Abstract>
    <FeatureTypeStyle>
```

5. IMPLEMENTACIÓN Y RESULTADOS

```
<Rule>
  <Name>rule1</Name>
  <Title>Blue Line</Title>
  <Abstract>A solid blue line with a 1 pixel width</Abstract>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>tipo</ogc:PropertyName>
      <ogc:Literal>0</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <TextSymbolizer>
    <Label>
      <ogc:PropertyName>tiempo</ogc:PropertyName> min,
      <ogc:PropertyName>distancia</ogc:PropertyName> km
    </Label>
    <Font>
      <CssParameter name="font-size">14</CssParameter>
      <CssParameter name="font-weight">bold</CssParameter>
    </Font>
    <Fill><CssParameter name="fill">#1E90FF</CssParameter></Fill>
  </TextSymbolizer>
  <LineSymbolizer>
    <Stroke>
      <CssParameter name="stroke">#1E90FF</CssParameter>
      <CssParameter name="stroke-width">4</CssParameter>
      <CssParameter name="stroke-dasharray">5 5</CssParameter>
    </Stroke>
  </LineSymbolizer>
</Rule>
<Rule>
  ...
</Rule>
</FeatureTypeStyle>
</UserStyle>
</NamedLayer>
</StyledLayerDescriptor>
```

El servidor *GeoServer* solamente puede obtener los datos a partir de *Shapefiles* o base de datos. Sin embargo, la propiedad *type*, comentada anteriormente, no está definida en ninguna tabla de la base de datos *geoGPS*.

Para conseguir este campo, se ha creado una vista llamada *GeoRutaGoogle* sobre las tablas *geo_package* y *geo_route*. Una vista es una estructura lógica que permite visualizar un grupo de datos que provienen de una o varias tablas. Así pues, con sólo una consulta a esta vista es posible obtener los datos necesarios para la visualización, es decir, la posición, distancia, duración y *type* de una unidad.

La vista *GeoRutaGoogle* es la siguiente:

```
CREATE OR REPLACE VIEW public."GeoRutaGoogle" AS
SELECT row_number() OVER (ORDER BY g1.id_geo_route) AS row_number,
       g1.id_geo_route,
       g1.issi,
       g1.unit,
       g1.unit_agency,
       to_timestamp((g1.fecha || ' '::text) || g1.hora, 'YYYYMMDD HH24MISS'::text) AS fecha,
       to_number(g1.fecha || g1.hora, '9999999999999999'::text) AS fechanumerica,
       g1.distancia,
       g1.tiempo,
```

5.2. Cálculo de la ruta óptima y la visualización

```
g1.linea,  
g2.isr,  
g1.id_geo_route % 5::numeric AS tipo  
FROM geo_route g1  
JOIN geo_package g2 ON g2.id_geo_package = g1.id_geo_package  
ORDER BY g1.unit, g1.id_geo_route;  
  
ALTER TABLE public."GeoRutaGoogle"  
OWNER TO postgres;
```

Así pues, la Figura 5.8 muestra un ejemplo del resultado final de la visualización es:



Figura 5.8: Ejemplo de visualización de las rutas, unidades e incidente.

Como se puede comprobar, el hecho de sólo poder obtener las intersecciones de las carreteras, a veces puede generar confusiones, como es el caso de la unidad *ECO-300*, con la ruta de color azul. Esto se debe a que la ruta que debería seguir dicha unidad es continuar por *Avenida de Gabriel Roca*, hasta el cruce con avenidas, casi donde se encuentra el punto final de la ruta. Sin embargo, la mayoría de veces, es factible comprender la ruta. Por ejemplo, el resto de rutas siguen las carreteras con bastante exactitud.

Librerías

Para facilitar las tareas comentadas anteriormente, se han empleado las siguientes librerías:

- *Json-simple*: Facilita el tratamiento de los datos de un Json. El siguiente fragmento de código muestra el modo de empleo de la misma.

5. IMPLEMENTACIÓN Y RESULTADOS

```
public static JSONObject getDistance(JSONObject json, String element) {
    JSONObject rows = (JSONObject) ((JSONArray) (json.get("routes"))).get(0);
    JSONObject elements = (JSONObject) ((JSONArray) (rows.get("legs"))).get(0);
    return (JSONObject) elements.get(element);
}
```

```
IniLat = Double.valueOf(getDistance(json, "start_location").get("lat").toString());
```

- *GisManager*: *GisManager* es una librería, creada por el SEIB112, que simplifica la construcción de un KML para realizar consultas al *geoServer* y una vez obtenidos los datos abre el *Google Earth* y le añade dicha consulta como una capa.

CONCLUSIONES

6.1. Introducción

En la gestión de un incidente donde hay vidas humanas en juego es muy importante tener la máxima información posible de los recursos. Por este motivo, se ha planteado este proyecto, con el objetivo de poder desplegar con la mayor eficacia y eficiencia dichos recursos. Así pues, se ha conseguido gestionar de manera más eficiente la flota de ambulancias de todos aquellos organismos que están integrados en el sistema del SEIB112, de tal forma que el 112 tenga el conocimiento de cuáles son los recursos disponibles y dónde se localizan en cada momento, para que los operadores y supervisores de sala puedan gestionar y coordinar los incidentes con mayor rapidez y exactitud. De este modo, la toma de decisiones se centraliza en el centro de emergencias 112 evitando favoritismos.

6.2. Resultados

Como ya se ha comentado, los objetivos del proyecto son: el tratamiento de los datos recibidos por los organismos y el cálculo de la unidad óptima con el correspondiente mapa interactivo.

Los requisitos que hacen referencia al primer objetivo, como asegurarse de que el estado o la posición están definidos o la utilización de la firma digital, se han hecho. Así pues, el primer objetivo se ha cumplido satisfactoriamente.

Respecto al segundo objetivo, también se han cumplido todos los requisitos definidos en la Sección 3.2. No obstante, las capas del GeoServer pueden llegar a causar alguna complicación para determinar cuál es la unidad más cercana. Además, en el momento que se desee realizar la integración con otros organismos, el precio de las consultas API de *Google Maps* podría encarecer el proyecto.

6.3. Mejoras

Quedan como trabajos futuros, el mejorar las capas del *GeoServer*. Por ejemplo, cambiar la capa de estilo para que cada unidad tenga una imagen distinta, de esta forma, sería mucho más fácil reconocerlas. También se podría cambiar la vista SQL para tener más información en las descripciones.

Además, aún quedan muchos otros proyectos para mejorar la eficiencia del servicio del 112. Por ejemplo, este proyecto podría orientarse a integrar los vehículos de policía prácticamente del mismo modo que se ha realizado con las ambulancias. Por ello, se ha intentado realizar el proyecto lo más genérico posible, y desacoplado al sector médico.

También se podría informar al conductor de la unidad en cuestión de cuál es la ruta que debe de tomar, de este modo, será posible llegar más rápido al destino deseado.

Otro trabajo que también se podría realizar sería comprimir el historial de los datos geográficos, es decir, reducir el volumen de datos de la tabla `geo_radio_position_historical`, mediante la división de la tabla en diferentes tablas para lograr una menor repetición de los datos. Este desarrollo sería muy interesante, ya que permitiría ahorrarse una gran cantidad de datos, lo cual conlleva a unas consultas y accesos más rápidos a dicha tabla.



SISTEMA DE COORDENADAS DE REFERENCIA

Durante la realización de este TFG ha sido necesario profundizar en los sistemas de coordenadas de referencia. Las coordenadas recibidas por parte del SAMU son coordenadas geográficas, debido a que la medición se ha hecho a partir de un GPS. Sin embargo, las aplicaciones de la sala de operaciones emplean el sistema UTM.

Un sistema de coordenadas es un conjunto de valores que permiten definir unívocamente la posición de cualquier punto de un espacio geométrico con respecto de un punto de origen. El conjunto de ejes, puntos o planos que confluyen en el origen y a partir de los cuales se calculan las coordenadas forman lo que se denomina sistema de referencia.

La decisión sobre el sistema de proyección cartográfica y el sistema de coordenadas de referencia a usar, depende de la extensión regional de la zona que se desea trabajar, del análisis que se quiere hacer y, a menudo de la disponibilidad de datos.

European Petroleum Survey Group (EPSG) fue una organización científica vinculada a la industria del petróleo europea. Estaba formada por especialistas que trabajaban en el campo de la geodesia, la topografía y la cartografía aplicadas en relación con la exploración petrolífera. EPSG compiló y difundió el conjunto de parámetros geodésicos EPSG, una base de datos ampliamente usada que contiene elipsoides, datums, sistemas de coordenadas, proyecciones cartográficas, etc.

Los sistemas de coordenadas más utilizados para representar la superficie de la tierra son:

A.1. Coordenadas geográficas

Utiliza una superficie esférica tridimensional para definir las localizaciones sobre la superficie terrestre. Se basan en el concepto de longitud y latitud para poder determinar la posición de un punto de la superficie terrestre.

El sistema de referencia utilizado son el ecuador terrestre como la línea de latitud cero y el meridiano de Greenwich es la línea cero de longitud. (Ver Ilustración A.1)

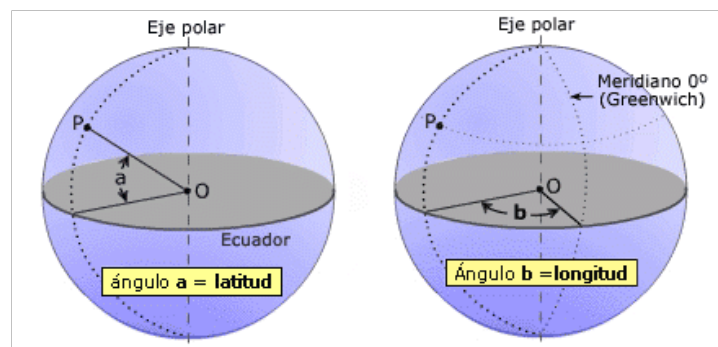


Figura A.1: Coordenadas geográficas, [2]

Es el sistema de referencia que usa la red de satélites GPS.

Se recibirán los datos de las posiciones de las unidades en este sistema, mas concretamente como EPGS:4326, también conocido como WGS84 Lat Lon World Geodetic System 84 que significa Sistema Geodésico Mundial de 1984.

A.2. Coordenadas cartesianas

Una posición se define en un espacio tridimensional por las coordenadas(x, y, z), como se puede observar en la siguiente figura.

El sistema de referencia viene definido como:

- El eje Z pasa por el centro de la tierra y por los polos.
- El eje X pasa por el centro de la Tierra y por el meridiano principal de Greenwich.
- El eje Y forma un ángulo de 90° con los otros dos ejes.

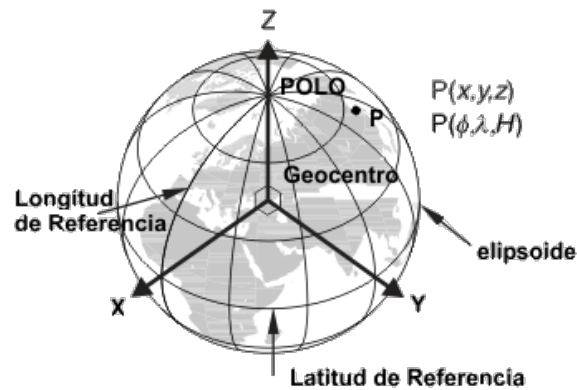


Figura A.2: Coordenadas cartesianas [3]

A.3. UTM

Es el sistema de proyección más usado. Este sistema se basa en el sistema definido por Mercator. La proyección de la superficie terrestre, como se puede observar en la siguiente ilustración se realiza sobre un cilindro.

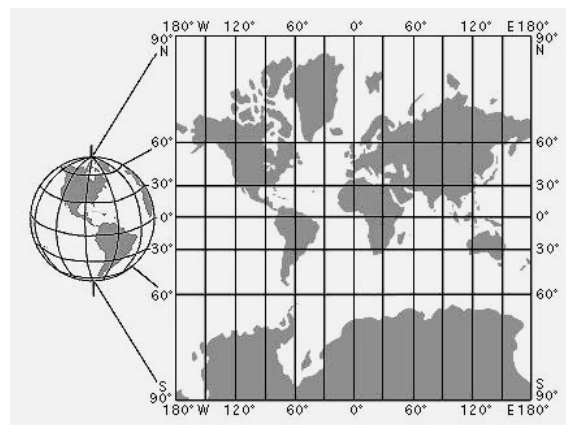


Figura A.3: Proyección Mercator [4]

Esta proyección es ideal para zonas longitudinales y estrechas, donde la distorsión con respecto a la realidad es pequeña. A medida que la superficie a representar aumenta en longitud y latitud, las deformaciones también crecen, llegando incluso a deformar totalmente la realidad.

Para minimizar estos efectos se ha llegado a los siguientes convenios:

- La proyección UTM toma como base la proyección de Mercator, sin embargo la posición del cilindro es transversal al eje de la tierra, puesto que las deformaciones en latitud son menos pronunciadas debido a la geometría elipsoide de la tierra.

A. SISTEMA DE COORDENADAS DE REFERENCIA

- Se divide la tierra en 60 husos, cada huso viene definido por dos meridianos separados por una longitud de 60. Para generar cada huso se emplea un cilindro distinto, siendo cada uno de los cilindros empleados tangente al meridiano central de cada huso. De esta manera se consigue que la superficie a proyectar tenga la mínima deformación posible en longitud, obteniendo una relación entre distancias reales y proyectadas desde 0 a 1,003 %.

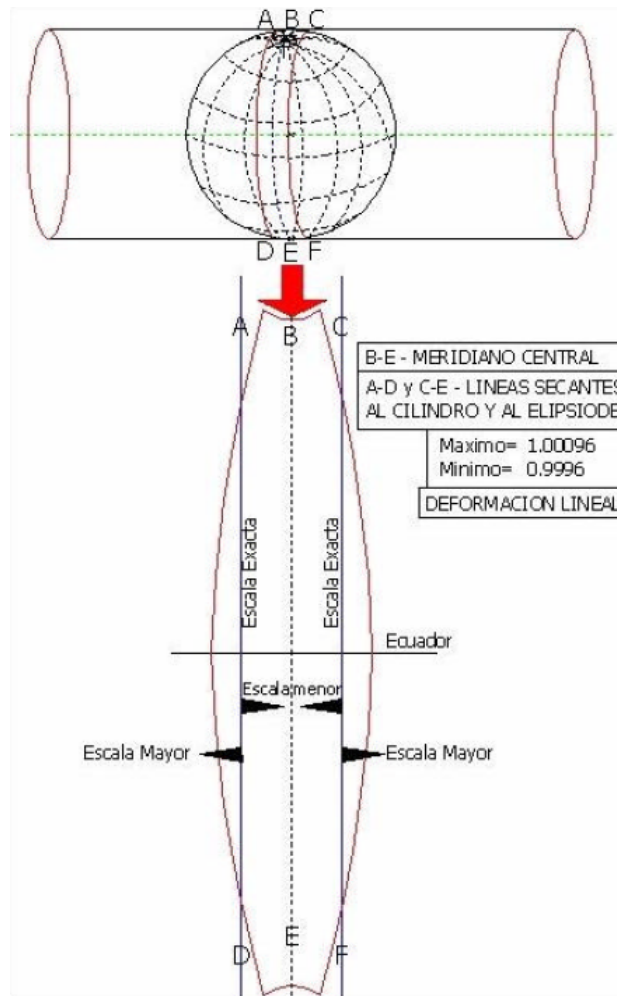


Figura A.4: Proyección UTM [5]

Ventajas que de esta proyección frente a los otros sistemas de proyección:

- Conserva los ángulos.
- No distorsiona las superficies en grandes magnitudes (por debajo de los 80° de latitud).
- Es un sistema empleado universalmente.
- Es un sistema que designa un punto o zona de manera concreta y fácil de localizar.

A pesar de recibir las ubicaciones en EPSG:4326, éstas se convertirán en otro tipo para poderlos mostrar. Mediante este sistema es como se almacena en la base de datos PostGreSQL. Este tipo es EPSG:25831, conocido como UTM ETRS89 31N, el sistema de Referencia Terrestre Europeo 1989 referido al huso 31. Como se puede observar en la siguiente imagen, las Islas Baleares pertenecen sólo al huso 31.

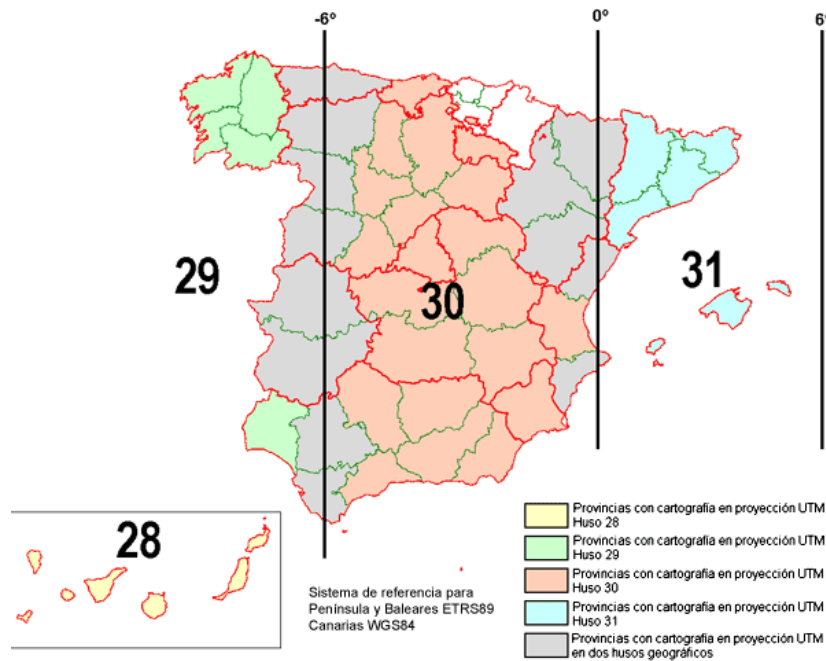


Figura A.5: Proyección UTM de España

BIBLIOGRAFÍA

- [1] “Opegeo suite.” [Online]. Available: http://www.geosamperu.com/paginas/opengeo_suite.html (document), 2.1
- [2] “Sistema de coordenadas geográficas.” [Online]. Available: <http://www.aristasur.com/contenido/sistema-de-coordenadas-geograficas-longitud-y-latitud> (document), A.1
- [3] “Sistema de coordenadas cartesianas.” [Online]. Available: <http://www.unal.edu.co/siamac/sig/geoide.html> (document), A.2
- [4] “La proyección de mercator.” [Online]. Available: <http://detopografia.blogspot.com.es/2012/12/la-proyeccion-de-mercator.html> (document), A.3
- [5] “Proyección universal transversa de mercator.” [Online]. Available: http://www.um.es/geograf/sigmur/temariohtml/node8_ct.html (document), A.4
- [6] “Servicio de emergencias de las islas baleares.” [Online]. Available: <https://www.caib.es/sites/112/es/portada-8673/> 1.1
- [7] “El uso del cad en las islas baleares,” Julio 2008. [Online]. Available: <http://www.computing.es/mercado-ti/casos-exito/1029127046401/seib-112-respuesta-rapida-localizada.1.html> 1.3.1
- [8] J. Bastida, “El 112 regulará las emergencias sanitarias para acabar con la ‘guerra de ambulancias,’” <https://ultimahora.es/sucesos/ultimas/2015/10/26/165167/112-regulara-emergencias-sanitarias-para-acabar-guerra-ambulancias.html>, Octubre 2015. 1.4
- [9] “Java,” java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. [Online]. Available: https://www.java.com/es/about/whatis_java.jsp 2.1
- [10] “Glassfish,” is the Open Source Java EE Reference Implementation. [Online]. Available: <https://javaee.github.io/glassfish/> 2.2
- [11] “Oracle database,” an object-relational database management system. [Online]. Available: <https://www.oracle.com/es/database/index.html> 2.5
- [12] “Postgresql,” an object-relational database management system. [Online]. Available: <https://www.postgresql.org/> 2.6

- [13] “Postgis,” is a spatial database extender for PostgreSQL object-relational database. [Online]. Available: <http://postgis.net/> 2.6.1
- [14] “Geoserver,” an open source server for sharing geospatial data. 2.7
- [15] “Manejo de certificados digitales con keytool,” Septiembre 2005. [Online]. Available: <https://www.adictosaltrabajo.com/tutoriales/security-ssl-keytool/> 2.8
- [16] “Edxl,” suite of XML-based messaging standards that facilitate emergency information. [Online]. Available: <https://www.oasis-open.org/committees/emergency/RNF-8>
- [17] “pgrouting,” is an extension of PostGIS and PostgreSQL geospatial database and adds routing and other network analysis functionality. [Online]. Available: <http://pgrouting.org/> 3.4.2
- [18] “Distance matrix api,” calcula el tiempo y la distancia de viaje. [Online]. Available: <https://developers.google.com/maps/documentation/distance-matrix/?hl=es> 1
- [19] “Directions api,” encuentra indicaciones para ir de una ubicación a otra. [Online]. Available: <https://developers.google.com/maps/documentation/directions/?hl=es> 2
- [20] “Firma digital sobre xml,” validar y firmar digitalmente un documento. [Online]. Available: <https://www.adictosaltrabajo.com/tutoriales/xml-signature/> 5.1.2
- [21] “St_contains,” patial Relationships and Measurements. [Online]. Available: http://postgis.net/docs/manual-1.4/ST_Contains.html 5.2.3
- [22] “Aprende a crear estilos en geoserver,” Septiembre 2015. [Online]. Available: <https://mappinggis.com/2013/01/aprende-a-crear-estilos-en-geoserver/> 5.2.3