

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN
GEOGRÁFICA ACCESIBLE**

**Realizado por
DAMIÁN SERRANO THODE**

**Dirigido por
Dr. FRANCISCO VILLATORO MACHUCA**

**Departamento
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN**

MÁLAGA, DICIEMBRE 2015

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente/a Dº/Dª. _____

Secretario/a Dº/Dª. _____

Vocal Dº/Dª. _____

para juzgar el proyecto Fin de Carrera titulado:

Análisis, diseño e implementación de un Sistema de Información Geográfica Accesible

realizado por D. Damián Serrano Thode

tutorizado por Dr. Francisco Villatoro Machuca,

y, en su caso, dirigido académicamente por

Dº/Dª. _____

ACORDÓ POR _____ OTORGAR LA CALIFICACIÓN
DE _____

Y PARA QUE CONSTE, SE EXTIENDE FIRMADA POR LOS COMPARECIENTES
DEL TRIBUNAL, LA PRESENTE DILIGENCIA.

Málaga a ____ de _____ del 20__

Presidente

Secretario

Vocal

Fdo:

Fdo:

Fdo:

Agradecimientos:

A todos, por la espera

Contenidos

1. Introducción.....	1
1. Objetivos.....	1
2. Descripción de la plataforma solicitada.....	1
3. Contenidos de la memoria.....	2
2. Plataformas de uso, recursos actuales disponibles.....	3
1. Lenguajes de programación y contenedores de aplicaciones.....	3
Java.....	3
PHP.....	4
.NET.....	4
Elección del lenguaje de programación.....	5
2. Bases de datos relacionales con características geoespaciales.....	5
ESRI ArcSDE.....	6
Mysql Spatial.....	6
PostGIS.....	6
Elección del motor de base de datos.....	7
3. Motores de Sistemas de Información Geográfica.....	7
ESRI ArcGIS.....	8
GeoServer.....	8
UMN MapServer.....	8
Elección del motor de GIS.....	9
4. Sistemas de lectura de texto.....	9
Festival.....	9

Loquendo TTS.....	10
Elección del motor de TTS.....	10
3. Diseño y estructura del desarrollo.....	11
1. Recogida de requisitos generales.....	11
2. Desarrollo de la aplicación web.....	12
Servicios web.....	13
Diseño de las páginas web.....	13
Diseño de los servlets.....	17
Servicio de geocodificación.....	23
Servicio de voz.....	25
4. Descripción de los módulos.....	27
1. Servicios intermedios.....	27
Página y servlet de búsqueda.....	28
Búsqueda por dirección.....	29
Búsqueda por localidad.....	31
Búsqueda por código postal.....	31
Página y servlet de detalles.....	32
Servlet de mapas.....	33
Servlet de contenido de voz.....	34
Servlet de imágenes.....	34
Servlet de edición.....	35
2. Servicios geoespaciales.....	35
Servicio WFS.....	35
Servicio WMS.....	37

Servicio de geocodificación.....	39
3. Servicio de voz.....	40
5. Instalación.....	41
1. Instalación y configuración de PostgreSQL + PostGIS.....	41
Instalación de la cartografía.....	41
La herramienta shp2pgsql.....	42
La estructura de una base de datos geoespacial.....	43
Crear entidades geoespaciales.....	45
2. Instalación y configuración de Apache Tomcat.....	48
3. Instalación y configuración de GeoServer.....	48
4. Instalación de la aplicación.....	48
6. Conclusiones y líneas futuras.....	50
7. Glosario.....	52
8. Bibliografía.....	53

Índice de tablas

Tabla 1: Parámetros de búsqueda por dirección.....	28
Tabla 2: Parámetros de búsqueda por localidad.....	29
Tabla 3: Parámetros de búsqueda por código postal.....	29
Tabla 4: Parámetros de búsqueda para los recursos de tipo CASE.....	29
Tabla 5: Valores por defecto para la petición WMS.....	34
Tabla 6: Parámetros de una petición WMS.....	38

Índice de figuras

Figura 1: Esquema funcional de la interfaz de usuario.....	13
Figura 2: Estructura de las páginas JSP.....	14
Figura 3: Diagrama de las clases de acceso a datos de los recursos CASE.....	18
Figura 4: Diagrama de las clases de transferencia de datos.....	19
Figura 5: Diagrama de secuencia del proceso de búsqueda.....	19
Figura 6: Diagrama de secuencia del proceso de detalles.....	20
Figura 7: Diagrama de secuencia de la generación de un mapa.....	21
Figura 8: Diagrama de secuencia de la búsqueda de fotografías.....	22
Figura 9: Diagrama de secuencia de las peticiones de voz.....	23
Figura 10: Diagrama de secuencia del servicio de geocodificación.....	24
Figura 11: Esquema de clases para el proceso de geocodificación.....	25
Figura 12: Diagrama de clases del servicio de voz.....	26
Figura 13: Diagrama de clases del cliente de texto a voz.....	26
Figura 14: Esquema del sistema completo.....	27
Figura 15: Página de resultados de la búsqueda.....	30
Figura 16: Página de refinamiento de la dirección.....	31
Figura 17: Página de detalles del recurso social.....	32

1.Introducción

1 Objetivos

La presentación de la información, cuando está acompañada de las modalidades adecuadas, resulta más amigable y efectiva al usuario. Por este motivo, el objetivo principal que se aborda en este proyecto es facilitar al usuario la búsqueda de la información y ofrecer una presentación más eficaz.

Dentro del catálogo de recursos de la Fundación Andaluza de Servicios Sociales disponibles al usuario se encuentran muchos servicios cuya información sería más útil si se presentara de forma más adecuada y comprensible. Por ello, se eligió un conjunto de servicios comprendido por los centros de atención socio-educativa (CASE) o guarderías, las residencias para la tercera edad y los centros de estancias diurnas, como piloto para este proyecto, consistente en un interfaz web para la búsqueda y la presentación de la información sobre estos servicios, acompañados de un mapa que presenta su localización. Además, para los usuarios con deficiencias visuales, la información se ofrecía de forma audiodescrita mediante un sistema de lectura de texto.

Este proyecto fue desarrollado durante el año 2006, antes del auge de los servicios públicos de cartografía como Google Maps y ofrece una visión sobre las soluciones disponibles entonces. El proyecto, además, sirvió como plataforma de pruebas para lo que posteriormente vendría a ser el Sistema de Información Geográfica de la Junta de Andalucía y su Callejero Digital.

Dentro de la repercusión del proyecto más allá del ámbito de la Junta de Andalucía, este fue presentado en una comunicación en las I Jornadas de SIG Libre organizadas en Girona en marzo de 2007 [SIGTE] y también en una ponencia en la Conferencia Mundial de Software Libre celebrada en Málaga en octubre de 2008 [OSWC].

2 Descripción de la plataforma solicitada

El sistema debía de ser accesible por el usuario mediante su navegador web, por lo que se alojaría detrás de una URL de acceso a la web de la Junta de Andalucía.

Para ello se contaría con un servidor web que almacenaría las páginas del proyecto que debían de ser dinámicas para poder modificar la información disponible.

También se contaría con una base de datos relacional en la que estaría contenida toda la información de los servicios ofertados.

Además, por las características geográficas del proyecto, se debería contar con una base de datos de información geoespacial donde estarían incluidos los datos cartográficos de los mapas, los puntos de interés y las localizaciones de dichos servicios.

Y por las características de accesibilidad a los usuarios con deficiencias visuales, se debería contar también con un sistema que permitiera la lectura de la información sobre los servicios dentro del navegador del usuario.

Por último, dentro del interés de la Junta de Andalucía por el fomento del software libre, sería deseable que el proyecto funcionara con componentes de este tipo de software.

3 *Contenidos de la memoria*

En el capítulo 2 “Plataformas de uso, recursos actuales disponibles” se muestran las posibilidades técnicas y tecnológicas para la construcción de la plataforma, y se toma la decisión entre las posibilidades de cada categoría.

En el capítulo 3 “Diseño y estructura del desarrollo” se describe el proceso seguido para el desarrollo de la plataforma, junto con los diagramas necesarios para explicar su estructura.

En el capítulo 4 “Descripción de los módulos” se describe el funcionamiento de cada uno de los módulos de los que se compone la plataforma.

En el capítulo 5 “Instalación” se describe el proceso necesario para instalar y configurar los componentes de la aplicación.

En el capítulo 6 “Conclusiones y líneas futuras” se describen las conclusiones a las que se llegaron al desarrollar la plataforma y las posibles mejoras a desarrollar en una futura etapa.

2. Plataformas de uso, recursos actuales disponibles

Para el análisis de la plataforma de ejecución de la aplicación se llevó a cabo un estudio de las soluciones disponibles en el mercado, tanto de software libre como propietario, pues aunque se deseaba una plataforma compuesta por software libre, no se estaba cerrado a utilizar una herramienta de software privativo si las ventajas de su uso justificaban el coste de adquisición.

1 Lenguajes de programación y contenedores de aplicaciones

La primera decisión en el proyecto era elegir la plataforma de desarrollo. Entre las plataformas a contemplar se encontraban las siguientes:

- Java J2EE
- PHP
- .NET

Java

El lenguaje Java fue desarrollado por Sun Microsystems en 1996, y se trata de un lenguaje orientado a objetos con una amplia trayectoria en el mundo del desarrollo [DUR].

El lenguaje se basa en una máquina virtual que ejecuta un microcódigo precompilado a partir del código fuente, similar al ensamblador, y esta máquina virtual es la que hace de capa de abstracción entre el lenguaje y las funcionalidades del hardware, lo que facilita su portabilidad a varias plataformas.

Concebido inicialmente como un lenguaje para sistemas empujados, vivió una amplia expansión en el mundo empresarial y actualmente no hay área de desarrollo en la que Java no esté presente.

Entre las ventajas de este lenguaje, están que cuenta con una gran variedad de entornos de desarrollo, librerías, y aplicaciones auxiliares. Además, puesto que cuenta con un recolector de basura, libra al programador de la gestión de la memoria, descargando esa tarea en la máquina virtual de Java.

Otra de las ventajas es la experiencia en el uso del mismo por el equipo de desarrollo, ya que anteriormente había sido usado para otros proyectos y el tiempo necesario para iniciar el desarrollo sería más corto que con otros lenguajes.

Una ventaja adicional es que el lenguaje cuenta con muchas herramientas para el diseño y modelado del software, que podían ayudar a la hora de definir las funcionalidades y diseñar la arquitectura de la aplicación.

Y por último, otra ventaja de esta plataforma es que es multiplataforma y cuenta con herramientas de desarrollo, librerías y entornos de ejecución para prácticamente todos los sistemas operativos existentes en el mercado.

PHP

PHP es un lenguaje usado principalmente para el desarrollo de aplicaciones web, aunque en los últimos años ha visto un incremento en su uso en el ámbito de la administración de sistemas. Es un lenguaje orientado a objetos y fue creado por Rasmus Lerdorf en 1994.

Tiene con una amplia difusión debido a que tiene una sintaxis muy parecida a la del lenguaje de programación C y cuenta con una amplia variedad de librerías para ayudar al desarrollo e integración con otras aplicaciones.

Al igual que Java, dispone con un entorno de ejecución para prácticamente todos los sistemas operativos existentes en el mercado, por lo que no habría problema a la hora de decidir el entorno de servidores de producción.

.NET

.NET es un entorno de programación creado por Microsoft en el año 2002 [CHA]. Cuenta con dos lenguajes de programación principales, C# (c sharp) [WIL] y Visual Basic.NET, aunque como uno de los componentes del entorno es un lenguaje de ejecución común (Common Language Runtime, CLR), existen implementaciones de otros lenguajes a la plataforma .NET como Fortran (F#).

El entorno está orientado hacia una máquina virtual que ejecuta el microcódigo CIL (Common Intermediate Language, Lenguaje Común Intermedio) y también cuenta con un recolector de basura de modo que el programador no se tiene que preocupar de la gestión de la memoria.

Una de las principales desventajas de esta plataforma es que limita su ejecución a sistemas operativos Windows, por lo que no sería posible utilizar un servidor Linux en el entorno de producción para dar servicio a la aplicación. Ciertamente es que en la actualidad Microsoft ha abierto la plataforma .NET y desde hace unos años se dispone de un entorno Open Source para la ejecución de aplicaciones desarrolladas en .NET, llamado Mono, sin embargo, en las fechas en las que se desarrolló el proyecto esta posibilidad no estaba disponible.

Elección del lenguaje de programación

Contempladas las posibilidades, y contando con los argumentos expuestos para cada opción, se acabó decidiendo que el proyecto se desarrollaría en lenguaje Java, utilizando el entorno de librerías de Java J2EE.

Como motor de aplicaciones se decidió utilizar Apache Tomcat [BRI][CHO] debido a su amplia trayectoria, a la cantidad de documentación disponible y al gran soporte con el que cuenta el programa en los foros de Internet.

2 Bases de datos relacionales con características geoespaciales

Las bases de datos con características geoespaciales se utilizan para almacenar la información geoespacial relativa a la cartografía [SIL][RIG]. En este tipo de bases de datos se dispone de unos tipos de datos especiales que representan elementos espaciales como un punto, una línea o un polígono, entre otros.

Además de los tipos de datos, también ofrecen funciones para la manipulación de este tipo de información, como por ejemplo, realizar uniones de polígonos, calcular distancias entre dos puntos o averiguar si un punto está dentro de una línea o un polígono.

Estas funciones, además, se pueden utilizar para la realización de consultas, por lo que se podría construir una sentencia SQL que permitiera seleccionar un conjunto de filas de una tabla en función de una serie de condiciones geoespaciales que se pueden combinar con las condiciones relacionales.

En el enfoque del proyecto se decidió ceñirse a los estándares en la medida de lo posible, por lo que un requisito deseable es que la capa geoespacial de la base de datos fuera conforme a la especificación de OGC “*Simple Feature Access: SQL Option*” [SFS], que define un esquema SQL estándar para el almacenamiento, obtención, consulta y actualización de características a través del interfaz de llamadas SQL/CLI.

Según esta especificación, en la base de datos se define una tabla de características que almacenará las referencias a las columnas espaciales de las tablas de un esquema. Esta tabla, además del nombre de la columna, tendrá información adicional como el sistema de referencia espacial utilizado en la información geoespacial, el tipo de dato almacenado (punto, línea, polígono, etc...), la dimensión del tipo de dato (unidimensional para un punto, bidimensional para una línea, etc...), entre otros.

Por lo tanto, dentro de los requisitos se podían encontrar las siguientes opciones disponibles:

- ESRI ArcSDE.

- Mysql Spatial.
- PostGIS.

ESRI ArcSDE

Es un producto comercial de la casa ESRI que ofrece una capa de recursos geoespaciales para las bases de datos Microsoft SQL Server, IBM DB2 u Oracle.

Como ventajas ofrece el soporte que hay detrás gracias a ESRI, que es el líder mundial en herramientas comerciales de GIS.

Entre las desventajas, la principal es el coste de adquisición, al que hay que añadir el coste de la propia base de datos sobre la que funciona el programa, puesto que también son productos comerciales con un coste de licencia.

Mysql Spatial

A partir de la versión 4.1 se podía encontrar implementado dentro de MySQL un subconjunto de las especificaciones del OGC para SQL.

Una ventaja de esta opción es que se trata de un proyecto de código libre, por lo que no es necesario la adquisición de ninguna licencia para el uso del producto.

Una desventaja importante, sin embargo, es que la implementación de la especificación del OGC está disponible para las tablas que usen el motor MyISAM, y el problema de este motor es que no implementa características básicas del modelo relacional como la integridad referencial. Además, el soporte para características espaciales en MySQL fue añadido en la versión 4.1 como un añadido implementado sobre el tipo de cadena de caracteres, por lo que no es un tipo de dato nativo geoespacial y necesitaba de una capa de traducción del dato para su almacenamiento y operación con funciones geoespaciales.

PostGIS

Es un proyecto de código libre que ofrece una librería de funciones adicional para la base de datos PostgreSQL, que implementa las características definidas por el OGC para SQL, disponible para PostgreSQL a partir de la versión 7 [MAN]. Este proyecto alcanzó su versión estable 1.0 en 2005.

Entre las ventajas, está que es un proyecto de código libre que está igualmente basado en un motor de base de datos de código libre, por lo que el coste de adquisición de una licencia es nulo. Otras ventajas son que PostgreSQL, el motor SQL sobre el que está basada la librería, tiene un soporte completo del estándar SQL92, integridad transaccional, un sistema extensible de tipos o columnas de tamaño ilimitado, entre otras.

Elección del motor de base de datos

Vistas las características de cada producto, se evaluaron las opciones y se decidió utilizar PostGIS junto con PostgreSQL para la capa de datos geoespaciales por su nulo coste de adquisición frente a la opción de ESRI, y por su mayor soporte de los estándares de SQL y OGC frente a MySQL.

3 Motores de Sistemas de Información Geográfica

Los motores de Sistemas de Información Geográfica (SIG), son el núcleo de una plataforma que trate datos de tipo geoespacial. Estas aplicaciones se encargan de procesar los datos geoespaciales almacenados en la base de datos o en los archivos de características.

Entre las funciones de esas aplicaciones se pueden encontrar las siguientes:

- Interpretar la información geoespacial contenida en la base de datos.
- Realizar operaciones geoespaciales con los datos.
- Generar cartografías a partir de los datos geoespaciales o las operaciones realizadas con ellos.
- Generar descripciones de las características a partir de los datos geoespaciales o las operaciones realizadas con ellos.
- entre otras...

Los requisitos para el proyecto eran los siguientes:

- Debía generar imágenes de la cartografía para poder mostrar un mapa con la localización del recurso.
- Debía mezclar en el mapa datos geoespaciales de distintas entidades.
- Debía permitir desplazarse en el mapa y cambiar el nivel de zoom.

Los requisitos geoespaciales no eran muy grandes, ya que era una herramienta de consumo y no se planteaba un uso profesional de las características de los sistemas de información geográfica. Por lo tanto, cualquier motor de SIG que se ajustara a la plataforma podría servir para las necesidades del proyecto.

Entre las opciones disponibles estaban las siguientes:

- ESRI ArcGIS.
- GeoServer.

ESRI ArcGIS

Es un producto comercial de la casa ESRI que ofrece las funcionalidades de motor de un sistema de información geográfica. Este programa funciona en cooperación con ArcSDE, que es la capa que ofrece el acceso a datos geoespaciales.

Como ventajas ofrece el soporte que hay detrás gracias a ESRI, que es el líder mundial en herramientas comerciales de GIS.

Entre las desventajas, la principal es el coste de adquisición, al que hay que añadir el coste de la base de datos relacional con características geoespaciales sobre la que funciona el programa, puesto que también es un producto comercial con un coste de licencia.

GeoServer

Es un programa servidor GIS de código libre con licencia GPL, mantenido por la comunidad internacional. Está basado en Java y se distribuye sobre un contenedor Jetty de serie, aunque se puede desplegar sobre cualquier contenedor Java en caso de necesitarlo, puesto que cumple con los estándares de Java para el despliegue de aplicaciones.

Como ventajas, se trata de un producto de código libre, y cuenta con una amplia trayectoria de desarrollo y una hoja de ruta de funcionalidades a incluir para las futuras versiones.

Otra ventaja es que la aplicación funciona sobre Java y, puesto que ya se había decidido utilizar Java J2EE para el desarrollo de la aplicación web, facilitaba el mantenimiento de la plataforma en su conjunto.

Una ventaja adicional, y también la más importante, es que implementa una amplia variedad de servicios geoespaciales definidos por el OGC, como Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), entre otros; y es, además, la implementación de referencia para el Open Geospatial Consortium. En este proyecto se iban a utilizar los servicios WMS y WFS, por lo que las necesidades estaban cubiertas en este sentido, y contar con el respaldo del OGC daba muchas garantías a esta aplicación.

UMN MapServer

Este es otro programa servidor GIS de código libre con licencia GPL, desarrollado en C por la Universidad de Minnesota y mantenido por la comunidad internacional.

Este programa se basa en una serie de scripts en PHP que utilizan un conjunto de herramientas de GIS como Proj.4 (librería para las conversiones de sistemas de proyección), GDAL (librería para

trabajar con formatos raster de cartografía) u OGR (librería para trabajar con formatos vectoriales de cartografía), para construir las respuestas a las peticiones de servicio.

Una desventaja de este programa era que en la fecha en la que se realizó el proyecto, el soporte a los estándares del Open Geospatial Consortium no era tan completo como el que ofrecía GeoServer.

Adicionalmente, al ser un programa basado en varios componentes, cada uno necesitaba su propia configuración y soporte, lo cual complicaba el mantenimiento del sistema servidor GIS, ya que había que estar atento a las actualizaciones de cada uno de los componentes.

Elección del motor de GIS

Dado que se prefería la utilización de programas de código libre por su nulo coste de adquisición, y que el soporte de los estándares definidos por el Open Geospatial Consortium era más amplio para GeoServer, aparte de ser la implementación de referencia y que era más sencillo de configurar al depender sólo de una aplicación, la elección se decantó por este servidor GIS.

4 Sistemas de lectura de texto

En este campo no hay tantas opciones, ya que se necesita una plataforma que permita generar la lectura del texto de forma programática, y no sirve un programa de escritorio que genere el audio mediante un interfaz de ventanas.

Las opciones consideradas fueron las siguientes:

- Festival.
- Loquendo TTS.

Festival

Festival es un producto de lectura de textos de código libre, mantenido por la comunidad internacional.

Proporciona un interfaz de comandos para la generación del audio, además de una librería para invocar a las funciones de texto a voz desde el programa.

Esta aplicación funciona bajo entornos Linux, por lo que, aunque es una opción a favor, la plataforma de producción estaba aún por determinar y no se quería cerrar que pudiera ejecutarse bajo Windows Server 2003.

Una opción en contra de este programa es que la calidad de la voz generada era muy pobre y no contaba con una pronunciación correcta de palabras medianamente complejas, además de dar un

tono de voz excesivamente metálico, por lo que para una aplicación de cara al público no alcanzaba un nivel aceptable de calidad.

Loquendo TTS

Loquendo TTS es el producto de lectura de textos de la empresa Loquendo. Entre las ventajas figuran que el software es multiplataforma, y que dispone de una amplia variedad de voces para la lectura de texto.

A través de la página web de Loquendo se podían hacer pruebas del sistema de texto a voz, por lo que se generaron varios audios de prueba y se observó que la calidad de la voz generada era muy aceptable y daba un aspecto natural.

Elección del motor de TTS

Después de evaluar los dos sistemas de texto a voz, se decidió elegir el programa de Loquendo porque la calidad de voz generada es muy alta y ofrece una librería multiplataforma que da libertad a la hora de elegir el entorno de producción.

Aunque el programa era de software privativo, las ventajas que presentaba sobrepasaban ampliamente el coste de adquisición del mismo.

3. Diseño y estructura del desarrollo

El desarrollo se estructuró siguiendo una metodología ágil [MAR][COC]. De esta forma, el sistema completo se dividió en distintos paquetes que fueron especificados y cada uno fue abordado de forma secuencial, dotando progresivamente al sistema de funcionalidades.

En este sentido, el desarrollo se estructuró de la siguiente forma:

1. Recogida de requisitos generales.
2. Desarrollo de la aplicación web.
 - a. Recogida de requisitos y especificación de una funcionalidad parcial.
 - b. Desarrollo de la funcionalidad especificada.
 - c. Pruebas y validación de la funcionalidad desarrollada.

De este modo, siguiendo la metodología de desarrollo ágil, los pasos 2.a a 2.c son repetidos para cada una de las funcionalidades a desarrollar del sistema. En el paso 1 se recoge una vista a grandes rasgos sobre las necesidades de la plataforma y servirá de directriz para el diseño de los componentes que conforman la plataforma a desarrollar.

1 Recogida de requisitos generales

En esta fase se llevaron a cabo reuniones con los responsables de los departamentos involucrados para extraer la información necesaria para diseñar una visión a grandes rasgos sobre las necesidades a cubrir con la plataforma.

En ellas se determinaron las siguientes características que debía cumplir el sistema:

- Como punto principal se determinó que la plataforma debía ser accesible a través de un navegador web, eliminando la necesidad de instalar un cliente pesado. Esto era un punto importante, ya que podía dar soporte de esta manera a todos los sistemas operativos pues el único requisito era que el usuario necesitaba un navegador web para acceder al sistema y no era necesario desarrollar una aplicación de escritorio con la dificultad adicional de tener que hacerla multiplataforma para dar soporte al mayor número de usuarios posible.
- En una fase inicial el sistema iba a proporcionar información sobre los centros de atención socioeducativa (guarderías), adscritos a la Junta de Andalucía, pero posteriormente se agregarían más tipos de entidades al sistema, como residencias para personas de la tercera

edad, por lo que la aplicación debía ser fácilmente extensible con los nuevos tipos de recursos.

- El acceso a la plataforma sería a través de un apartado de la página web de la Junta de Andalucía, por lo que sería gratuito y estaría disponible las 24 horas del día para su consulta. El público objetivo de la aplicación sería cualquier ciudadano de la comunidad autónoma que necesitara información sobre los centros de atención socioeducativa (guarderías) adscritos a la Junta de Andalucía, o cualquier otro tipo de recurso social incorporado en fases posteriores, por lo que debía de ser de fácil uso e intuitivo, teniendo en cuenta que los usuarios no necesariamente iban a tener amplios conocimientos sobre el funcionamiento de aplicaciones web.
- La aplicación debía permitir buscar los recursos sociales de diferentes formas: mediante una dirección, por la localidad o a través de un código postal.
- No sólo debía mostrar información geoespacial, sino que deberían aparecer datos adicionales sobre las entidades que componen el sistema, como información sobre los horarios de apertura, una descripción del centro, fotografías y otros datos a considerar.
- El sistema debía ser suficientemente accesible para las personas con algún tipo de discapacidad visual y debería proporcionar una descripción en audio de los resultados de la búsqueda.
- Y como característica de especial interés, que la plataforma funcionara sobre sistemas y aplicaciones de código libre, dentro de la política de apoyo al software libre de la Junta de Andalucía. Esto no era considerado como una restricción fuerte, sino más bien como una característica deseable, puesto que si se encontraba algún software privativo que ofreciera una ventaja considerable frente a uno de código libre, podría ser tenido en cuenta en la plataforma.

2 Desarrollo de la aplicación web

A la vista de los requisitos generales recogidos, se planteó una estructura de páginas y servicios web para hacer frente a las necesidades de la aplicación. Estas páginas y servicios estarían organizados según los siguientes grupos:

- Servicios web.
- Servicios geoespaciales.
- Servicios de voz.

Servicios web

En este apartado se va a explicar la estructura elegida para organizar la aplicación web.

Diseño de las páginas web

Se planteó una estructura basada en un flujo a través de una serie de páginas para proporcionar al usuario la información solicitada.

La estructura de navegación que se planteó fue la que se puede observar en la figura 1, donde se muestran los cuatro bloques principales con los que el usuario de la aplicación interactuará.

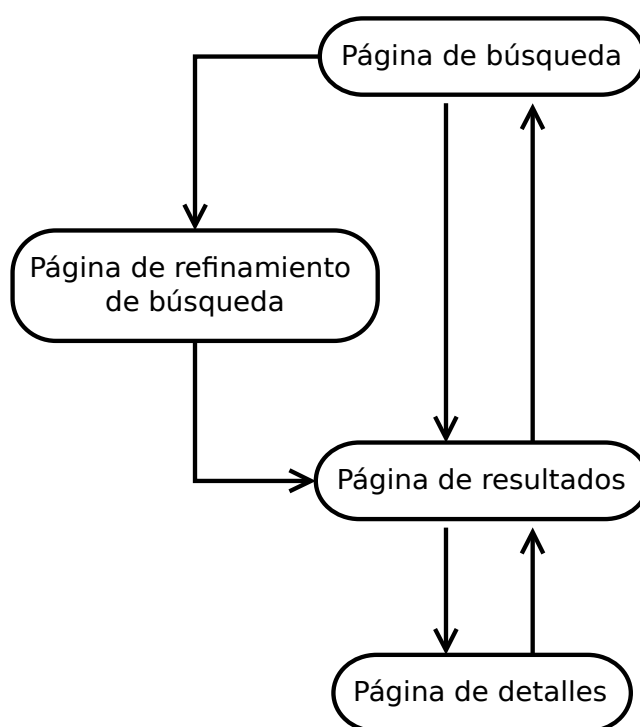


Figura 1: Esquema funcional de la interfaz de usuario

En el bloque de la pantalla de búsqueda se mostrará una información general que explique la finalidad de la aplicación y los resultados que se pueden obtener mediante su uso, además de los campos de búsqueda para encontrar los recursos sociales; en el bloque de la pantalla de refinamiento de búsqueda se solicitará al usuario más información si la búsqueda que ha realizado no da un resultado concluyente y es necesario que indique una mayor precisión; en el bloque de la pantalla de resultados se mostrará al usuario la información contenida en la base de datos que cumple los criterios de búsqueda introducidos por el mismo; y en el bloque de la pantalla de detalles, el usuario de la aplicación web podrá consultar la información detallada del recurso social solicitado, como por ejemplo el nombre completo, la dirección, los horarios de apertura o un mapa de localización de la zona.

Contando ya con la plataforma tecnológica, se decidió diseñar la estructura de esta parte de la aplicación mediante páginas JSP que invocaran servlets que realizarían las funciones necesarias de búsqueda. Estas páginas JSP utilizarían las etiquetas definidas por Java J2EE para poder incluir plantillas comunes para todas las páginas, como las cabeceras y pies de páginas, y demás contenidos compartidos.

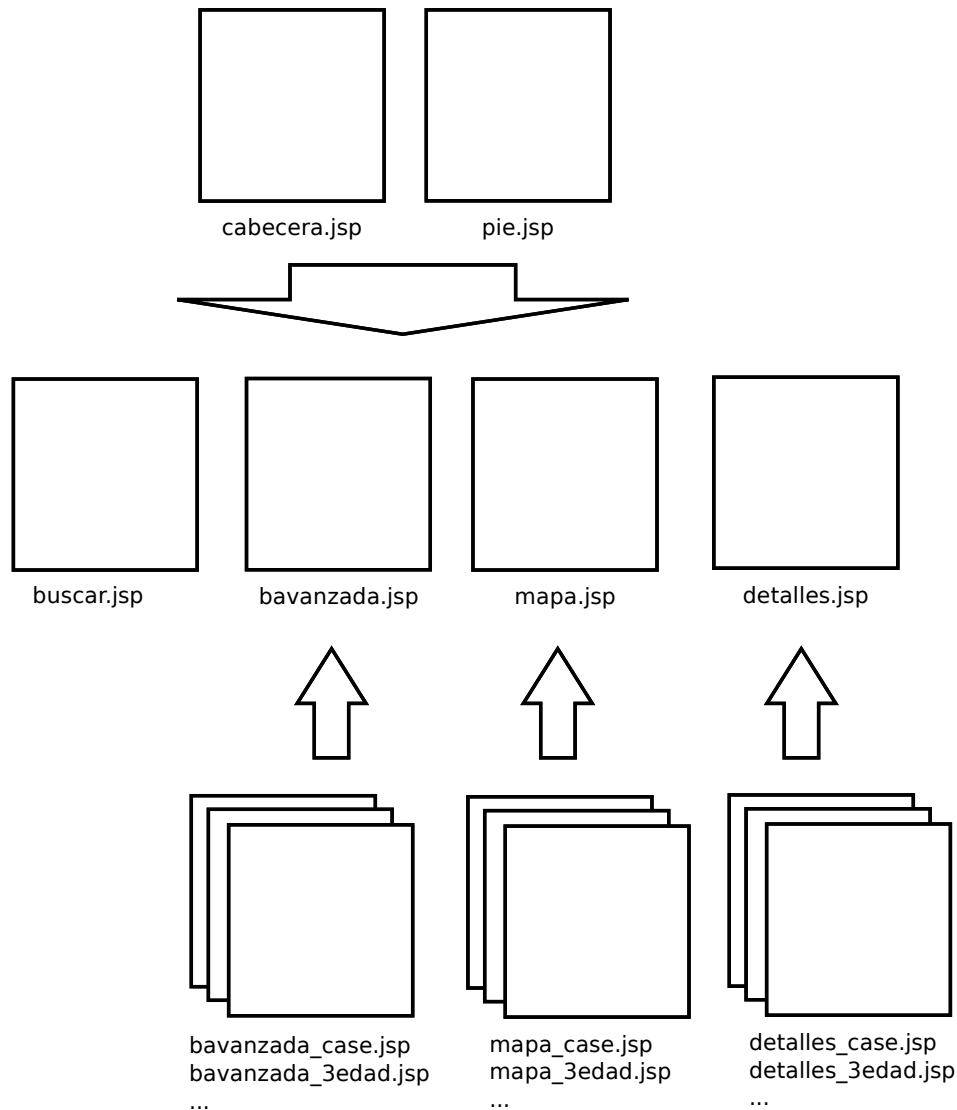


Figura 2: Estructura de las páginas JSP

Al ser necesario mostrar distintos tipos de recursos en la aplicación, se debía contar con diferentes páginas que mostraran los resultados de cada tipo con sus particularidades, de esta forma, se decidió la siguiente estructura jerárquica de páginas y plantillas que se puede ver en la figura 2.

Cada una de las páginas principales incluye mediante etiquetas `<jsp:include>` el contenido de las plantillas de cabecera o pie y de las páginas del tipo de recurso concreto sobre el que se está consultando la información. De esta forma, se puede proporcionar un aspecto homogéneo en toda la aplicación reutilizando componentes en las páginas JSP.

En los siguientes apartados se van a describir los requisitos que se recogieron para cada una de las páginas en las reuniones con el personal encargado de definir las funcionalidades de la aplicación.

Página de búsqueda

En esta página el usuario introducirá los parámetros de búsqueda para encontrar los recursos sociales deseados.

Los requisitos establecidos para esta página fueron los siguientes:

- Al usuario se le ofrecerán tres tipos de búsqueda distintos: Mediante una dirección que incluya el tipo de vía, nombre de la vía, número de la vía, localidad y provincia; mediante una localidad y provincia; o bien mediante un código postal. Cada tipo de búsqueda le mostrará al usuario los recursos sociales según las siguientes directrices:
 - Para una búsqueda mediante dirección, se deberán encontrar los recursos sociales a partir de una circunferencia con centro en la dirección introducida.
 - En la búsqueda por localidad y provincia, se deberán mostrar todos los recursos sociales existentes en la localidad introducida.
 - Para la búsqueda mediante código postal, se deberán mostrar los recursos sociales existentes en el código postal introducido.
- Los parámetros de búsqueda deben incluir el tipo de recurso social que se desea encontrar.
- En caso de seleccionar el método de búsqueda mediante dirección, si no se puede encontrar una dirección única, se le ofrecerán al usuario todas las posibles coincidencias para que seleccione la dirección correcta. Sin embargo, si la coincidencia es única o bien selecciona uno de los otros métodos de búsqueda, se dirigirá al usuario a la página de resultados.

Según los requisitos establecidos, se pueden extraer las siguientes necesidades para esta página:

- Debe ofrecer un mecanismo para traducir una dirección en un par de coordenadas.
- Debe disponer de un sistema para realizar búsquedas de los recursos sociales.

Página de refinamiento de búsqueda

En esta página se deben mostrar los resultados de la búsqueda de direcciones cuando no exista una coincidencia única para la dirección introducida.

Los requisitos para esta página son los siguientes:

- Deberá mostrar todas las coincidencias de la vía introducida como parámetro en la página de búsqueda.
- Deberá facilitar un medio para seleccionar la vía correcta de entre todas las disponibles.
- Una vez seleccionada la vía correcta deberá mostrar los resultados en la página de resultados.

Página de resultados

La página de resultados mostrará los resultados obtenidos por una búsqueda en la página de búsquedas. Estos resultados se pueden obtener bien por una búsqueda por dirección, por localidad o por código postal.

Los requisitos que se recogieron para esta página fueron los siguientes:

- Deberá mostrar toda la información pública disponible sobre el recurso social, como por ejemplo el nombre, la dirección, los horarios de apertura, la información sobre el curso lectivo si estuviera disponible (de aplicación para los CASE).
- Deberá contar además con un mapa de situación, centrado en el punto donde se encuentra el recurso social, y deberá mostrar las calles adyacentes junto con información de puntos de interés de alrededor para facilitar su ubicación. Este mapa deberá ser navegable para que el usuario pueda cambiar el nivel de zoom y desplazarse sobre el mismo.
- Deberá proporcionar información textual sobre los puntos de interés, indicando la distancia aproximada desde el punto de interés hasta el recurso social.
- Asimismo, deberá ofrecer una locución que explique los detalles del centro de forma hablada.

Página de detalles

La página de detalles mostrará los detalles para el recurso social seleccionado. Los requisitos recogidos para esta página son los siguientes:

- Deberá mostrar los datos disponibles para el recurso social como el nombre, titularidad, dirección, horario, etc...
- Deberá ofrecer también un mapa centrado en la localización del recurso social, y que muestre también puntos de referencia disponibles alrededor hasta un radio determinado.
- Deberá disponer también de una descripción en audio del texto mostrado en la página.

Diseño de los servlets

A raíz de las necesidades recogidas en la fase anterior de toma de requisitos para cada una de las páginas JSP de la aplicación web, se diseñaron una serie de servlets que acompañarían a la aplicación y responderían a las peticiones de los usuarios, usando las páginas JSP únicamente para la visualización de la información.

Los servlets serán invocados cuando se solicite alguna acción en las páginas web mencionadas anteriormente. Estos servlets se encargarán de realizar la búsqueda según los parámetros introducidos, obtener los datos del recurso social seleccionado, o recuperar las imágenes de los recursos, entre otras tareas.

La organización de servlets diseñada para la aplicación es la siguiente:

- DetailsServlet. Este servlet se encarga de recuperar los detalles del recurso social seleccionado.
- EditarServlet. Este servlet se encarga de actualizar la posición sobre el mapa del recurso social seleccionado.
- FotosServlet. Este servlet se encarga de recuperar de la base de datos las fotografías de los recursos sociales.
- MapServlet. Este servlet se encarga de obtener de los servicios geoespaciales una imagen del mapa para las coordenadas recibidas como parámetros.
- SearchServlet. Este servlet se encarga de obtener los recursos sociales disponibles en la base de datos según los parámetros de búsqueda introducidos.
- VoiceServlet. Este servlet se encarga de obtener el audio correspondiente a la lectura del texto resultado de una búsqueda o de los detalles de un recurso social.

A continuación, en los siguientes apartados se va a describir cada uno de estos servlets.

Acceso a la base de datos

Puesto que un requisito de la aplicación era que pudiera soportar distintos tipos de recursos sociales, se construyó una estructura basada en clases abstractas e interfaces para poder extender fácilmente los objetos y dar soporte a todos los tipos de recursos sociales. En la figura 3 se puede ver la estructura de clases correspondiente a la capa de acceso a base de datos para el tipo de recurso CASE.

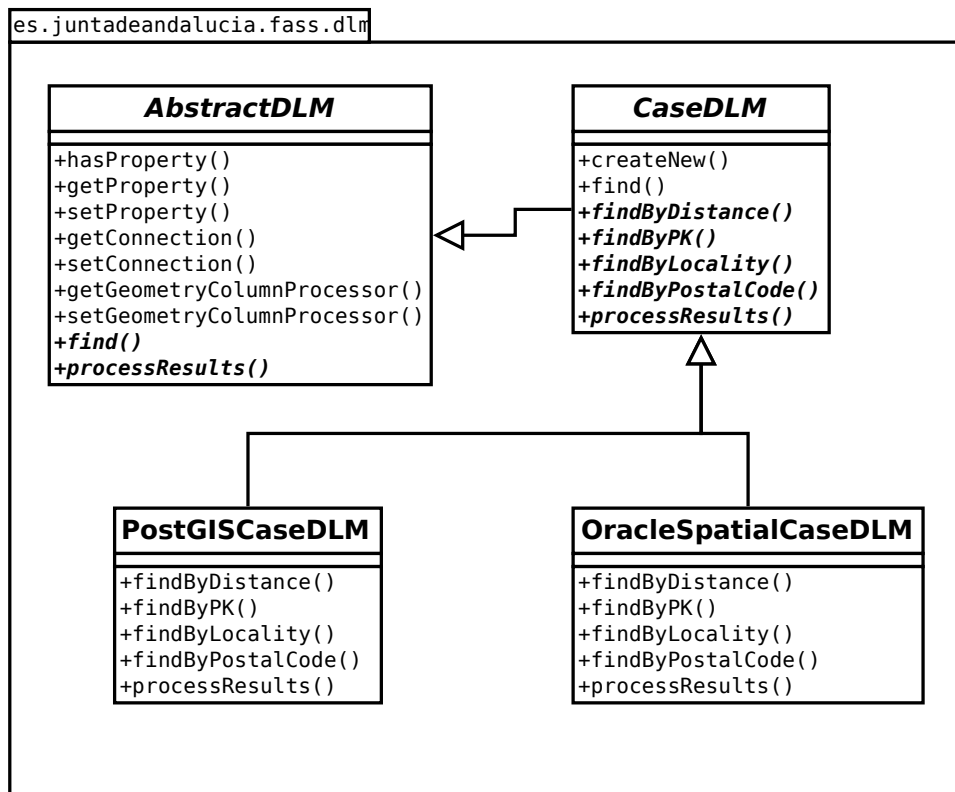


Figura 3: Diagrama de las clases de acceso a datos de los recursos CASE

Existe una clase abstracta principal común para todos los tipos de recursos llamada `AbstractDTO` que define dos métodos abstractos: un método de búsqueda y un método para procesar los resultados de la búsqueda.

La clase abstracta `CaseDLM` hereda de `AbstractDTO`, implementando el método `find`, pero definiendo otros métodos abstractos para realizar la búsqueda según el tipo de búsqueda seleccionado. Estos métodos abstractos serán implementados por las clases hijas `PostGISCaseDTO` y `OracleSpatialCaseDTO` que son las que realizan la búsqueda utilizando las funciones geoespaciales concretas definidas por cada motor de base de datos geoespacial, PostgreSQL más PostGIS u Oracle más OracleSpatial.

Transferencia de los datos

Los métodos de búsqueda devuelven los resultados incrustados en una clase llamada `SearchResult`. Esta clase abstrae los resultados de los distintos tipos de recursos en una clase opaca que maneja por igual todos los tipos de recursos sociales.

Esta clase `SearchResult` contiene internamente un `Map` de recursos donde los mismos están indexados por su tipo y el valor es una clase `List` de objetos del mismo tipo de recurso. En el

diagrama de la figura 4 se puede observar la organización de una parte de los objetos de transporte de resultados (*Data Transfer Objects*, DTO).

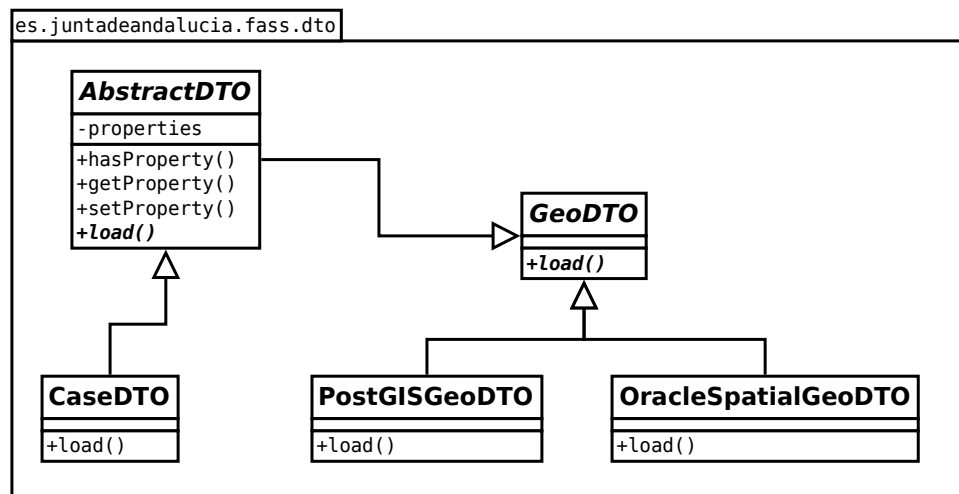


Figura 4: Diagrama de las clases de transferencia de datos

Servlet de búsqueda

El proceso de búsqueda está definido por el diagrama de secuencia de la figura 5. En éste se puede observar la organización de las clases mediante la clase abstracta `AbstractSearch` que generalizan el proceso de búsqueda independientemente del tipo de recurso solicitado.

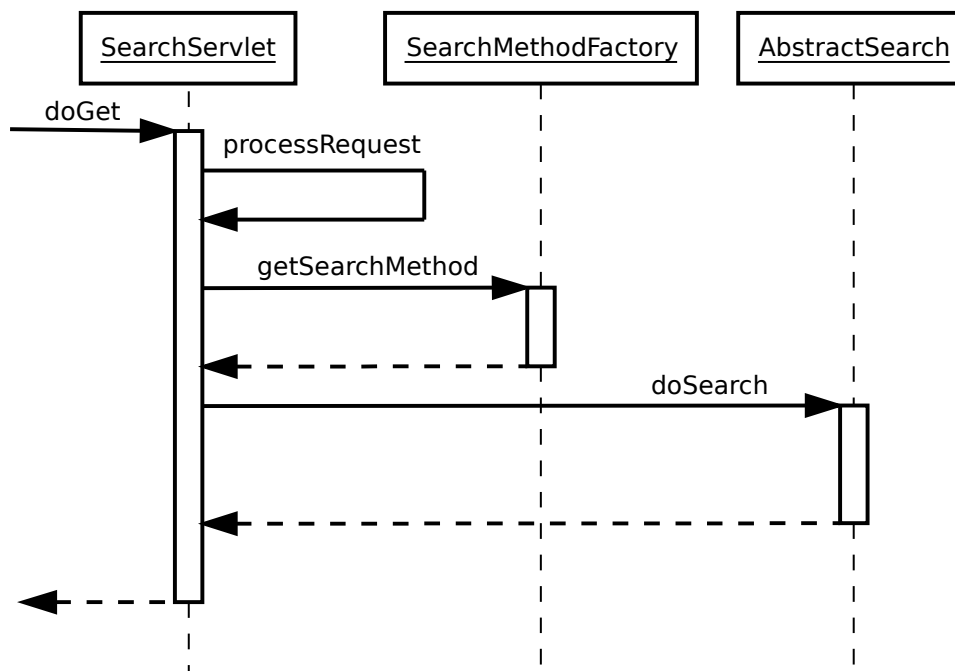


Figura 5: Diagrama de secuencia del proceso de búsqueda

Servlet de detalles

El servlet que se encarga de devolver los detalles para un recurso social sigue un proceso que está reflejado en la figura 6.

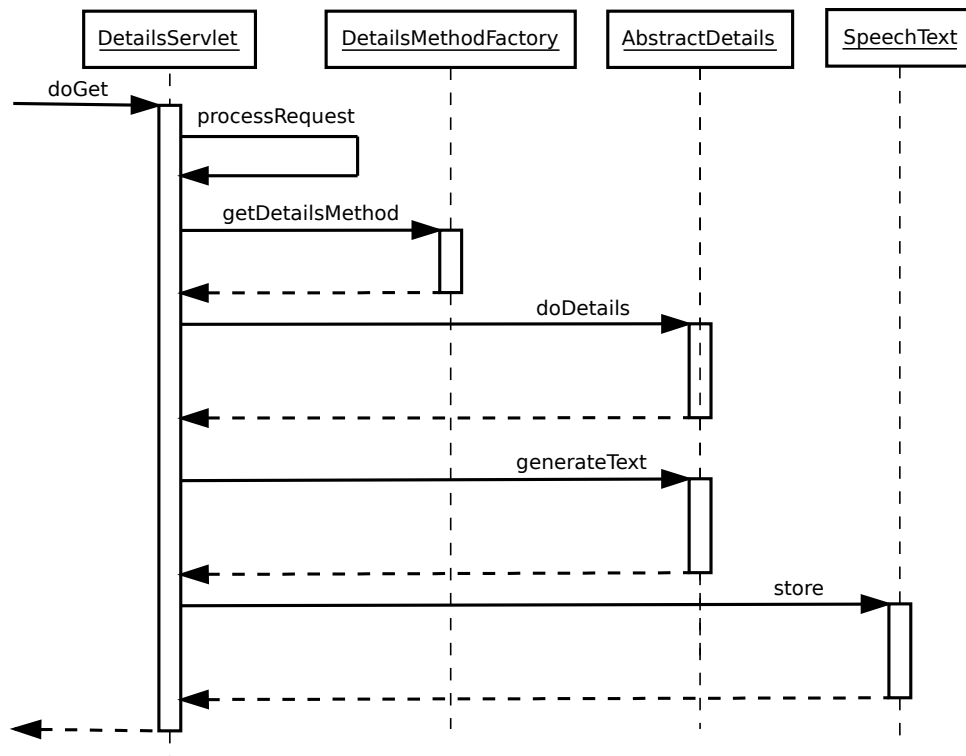


Figura 6: Diagrama de secuencia del proceso de detalles

Se siguió la misma estructura que para el servlet de búsqueda, en la cual una clase factoría devuelve la implementación de la clase abstracta AbstractDetails que se encarga de gestionar el proceso de carga de datos para devolver los resultados de la página según el tipo de recurso social sobre el que se desea buscar. Adicionalmente, en esta fase se genera el texto que desde la página se leerá mediante el sistema de texto a voz.

Servlet de mapas

Este servlet se encarga de generar una imagen de un mapa con las coordenadas recibidas en la petición.

Este servlet recibe como parámetros una extensión definida por un par de coordenadas (x1, y1), (x1, y2) que delimitan un rectángulo por sus esquinas superior izquierda e inferior derecha, respectivamente. Esta será el área visualizada en la imagen del mapa.

Con estos datos más otra información extraída de un archivo de configuración que define una serie de parámetros fijos se construye una petición al servicio WMS siguiendo el estándar del Open Geospatial Consortium para peticiones WMS del tipo GetMap.

Este servidor WMS devolverá una imagen del mapa la cual será devuelta al usuario que inició la petición, para ser mostrada en la página de resultados de la búsqueda o en la página de detalles de un recurso social.

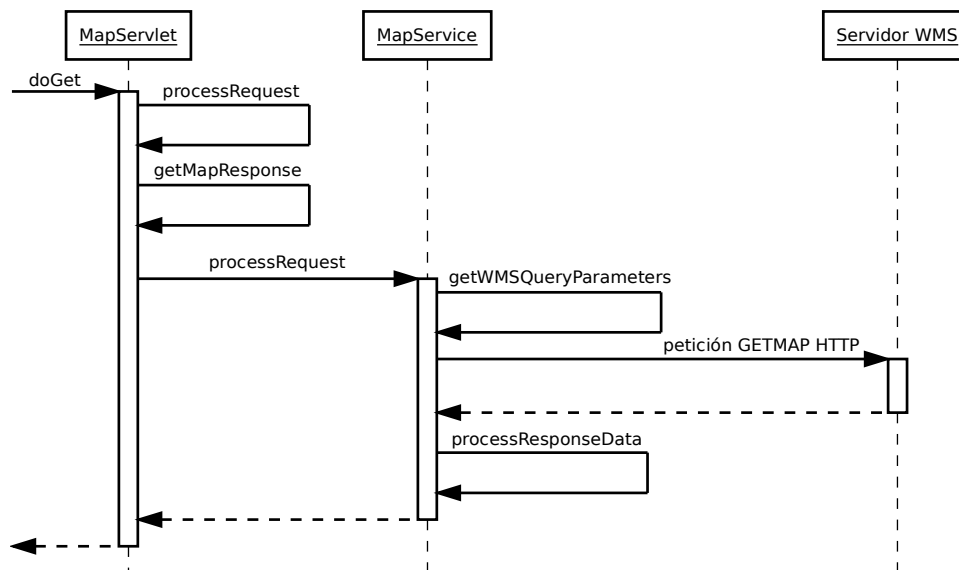


Figura 7: Diagrama de secuencia de la generación de un mapa

Servlet de fotos

Adicionalmente a la información textual, geográfica o de voz, se mostraría en la aplicación las fotografías disponibles sobre el recurso social solicitado.

Por ejemplo, para los recursos CASE se trataría de fotos del propio centro, las instalaciones de las aulas o los patios de juego infantiles.

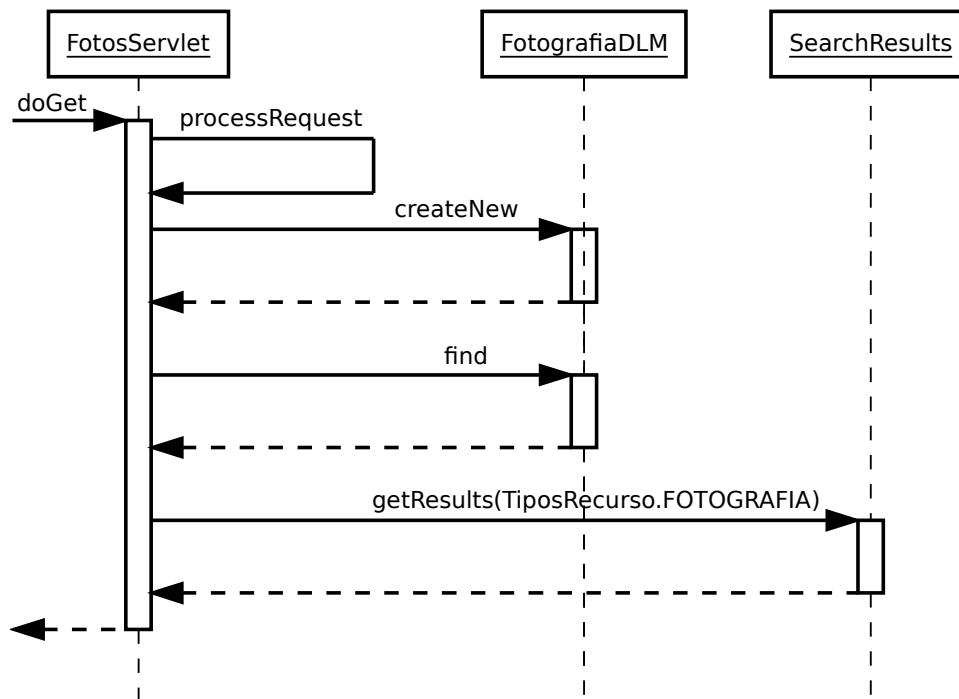


Figura 8: Diagrama de secuencia de la búsqueda de fotografías

Este servlet recibiría una petición que contendría la clave principal del recurso social, además del tipo de recurso, y con esta información, extraería las fotografías de la base de datos y las devolvería al usuario para ser mostradas en la página web de detalles.

Servlet de voz

Este servlet se encarga de devolver la lectura a voz en formato MP3 del texto generado en una fase anterior. El diagrama de secuencia del proceso de este servlet se puede observar en la figura 9.

La clase `AbstractVoiceClient` abstrae la implementación del motor de texto a voz en un interfaz común que facilita el desarrollo pudiendo implementar un cliente para nuevo motor de texto a voz si fuera necesario.

El servlet recibiría la clave generada en el servlet `DetailsServlet` la cual le permitiría recuperar un texto ya generado. Este texto sería enviado al motor de texto a voz siguiendo un interfaz común definido por la clase abstracta mencionada anteriormente y recibiría como respuesta un audio en formato RAW, el cual sería enviado a un codificador a MP3 para poder ser reproducido en la página web mediante un reproductor de audio en formato Adobe Flash incrustado en la página.

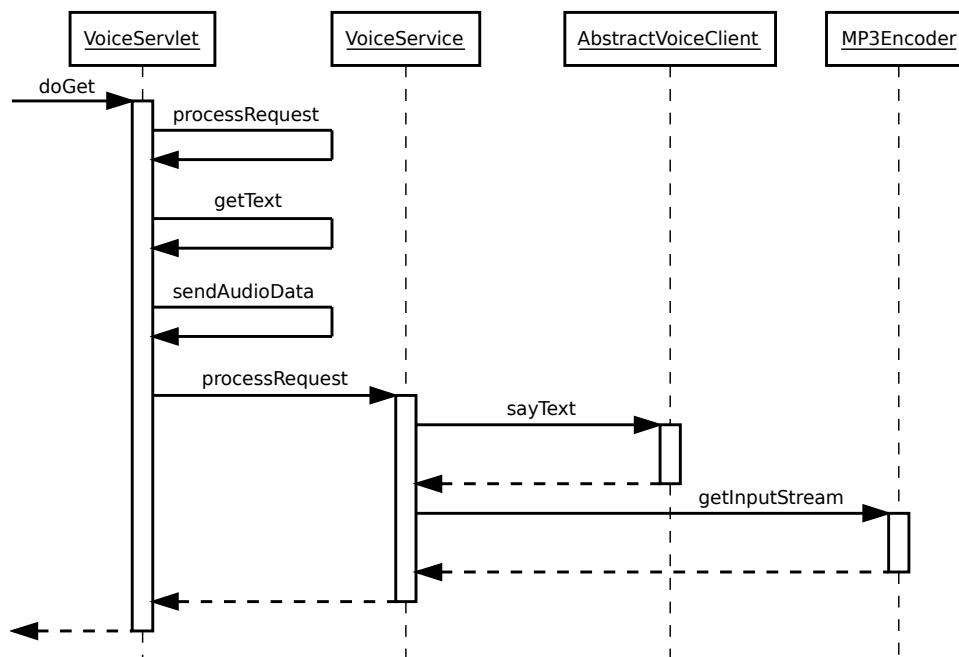


Figura 9: Diagrama de secuencia de las peticiones de voz

Servicio de geocodificación

Aparte de los servicios geoespaciales ofrecidos por GeoServer siguiendo el estándar definido por el Open Geospatial Consortium, fue necesario desarrollar un servicio que transformara una dirección a un par de coordenadas, llamado servicio de geocodificación.

Este servicio recibiría unos datos textuales sobre una dirección, en concreto el tipo de vía, nombre de vía, número de vía, localidad y provincia, e intentaría obtener un par de coordenadas para esta dirección, que posteriormente se utilizarían como punto de partida para obtener información de características geoespaciales o una imagen de un mapa.

Es por ello que este servicio es de vital importancia ya que un tercio de la funcionalidad de la aplicación depende de su correcto funcionamiento. Lamentablemente, en el enfoque del proyecto no se contaba con tiempo ni los recursos suficientes para desarrollar un servicio de geocodificación, puesto que este es un campo muy extenso de investigación en las áreas de inteligencia artificial, procesamiento del lenguaje natural o análisis de datos.

Además, las implementaciones existentes de otros servicios de geocodificación son muy dependientes del conjunto de datos y no es sencillo o factible generalizar su funcionamiento, por lo que tampoco se podían realizar adaptaciones de otras aplicaciones existentes.

Debido a ello, se diseñó un servicio de geocodificación simple que tomara los datos anteriormente mencionados e intentara localizar la dirección mediante comparación de subcadenas. De esta forma, en la mayoría de situaciones se podría encontrar la vía indicada sin muchas complicaciones ya que

las coincidencias en la nomenclatura de las calles no era tan habitual, y en los casos en los que surgieran resultados no concluyentes, se mostrarían al usuario las posibilidades para que eligiera la dirección deseada.

En la figura 10 se puede observar el diagrama de secuencia de llamadas entre las clases involucradas en el proceso de geocodificación.

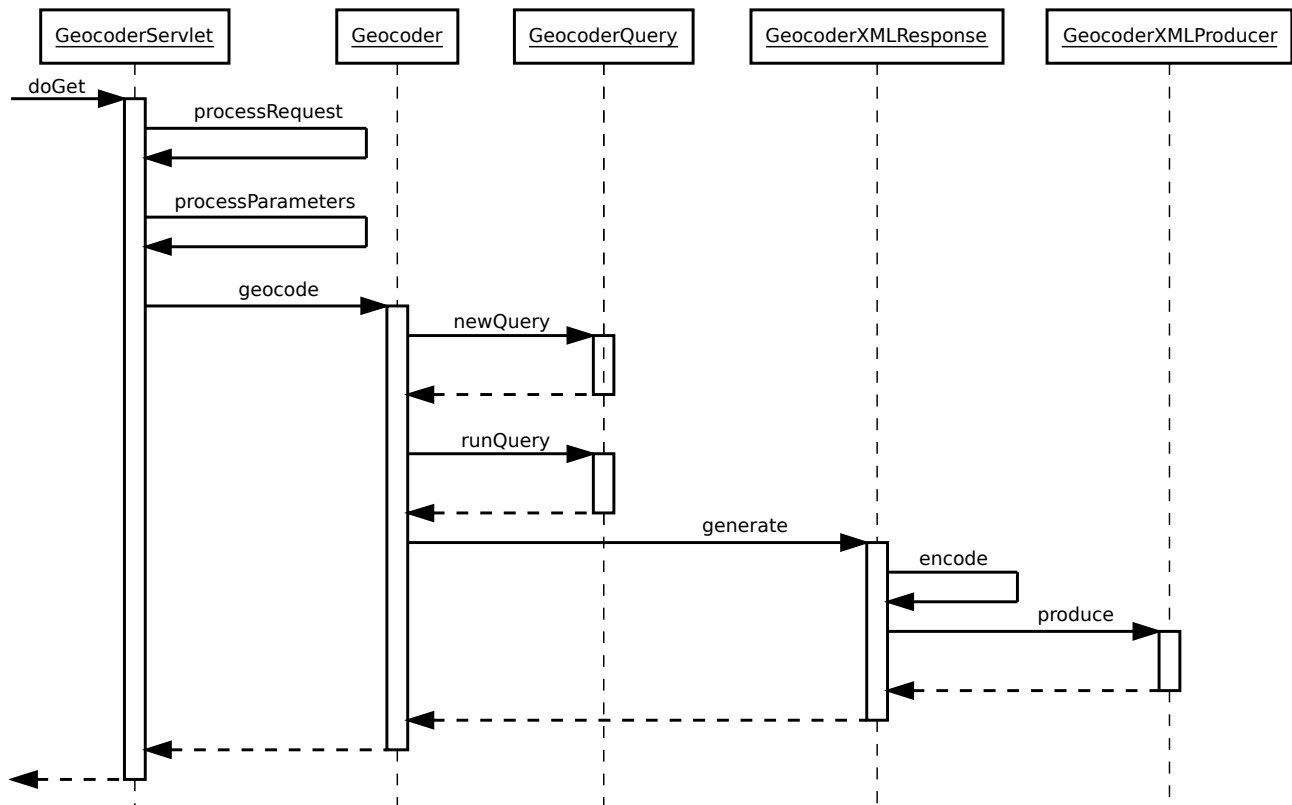


Figura 10: Diagrama de secuencia del servicio de geocodificación

En el servicio de geocodificación la información geoespacial de los viales se encontraba en la base de datos PostgreSQL con las extensiones geoespaciales de PostGIS, sin embargo, se diseñó una estructura de clases abstractas que permitiera intercambiar el motor de base de datos geoespacial para el servicio de geocodificación. De esta forma, si se trasladaran los datos geoespaciales de los viales a otro motor de base de datos, sería necesario con implementar esta clase abstracta según la estructura de la base de datos en concreto.

En la figura 11 se pueden observar las relaciones entre las clases involucradas en el proceso de geocodificación.

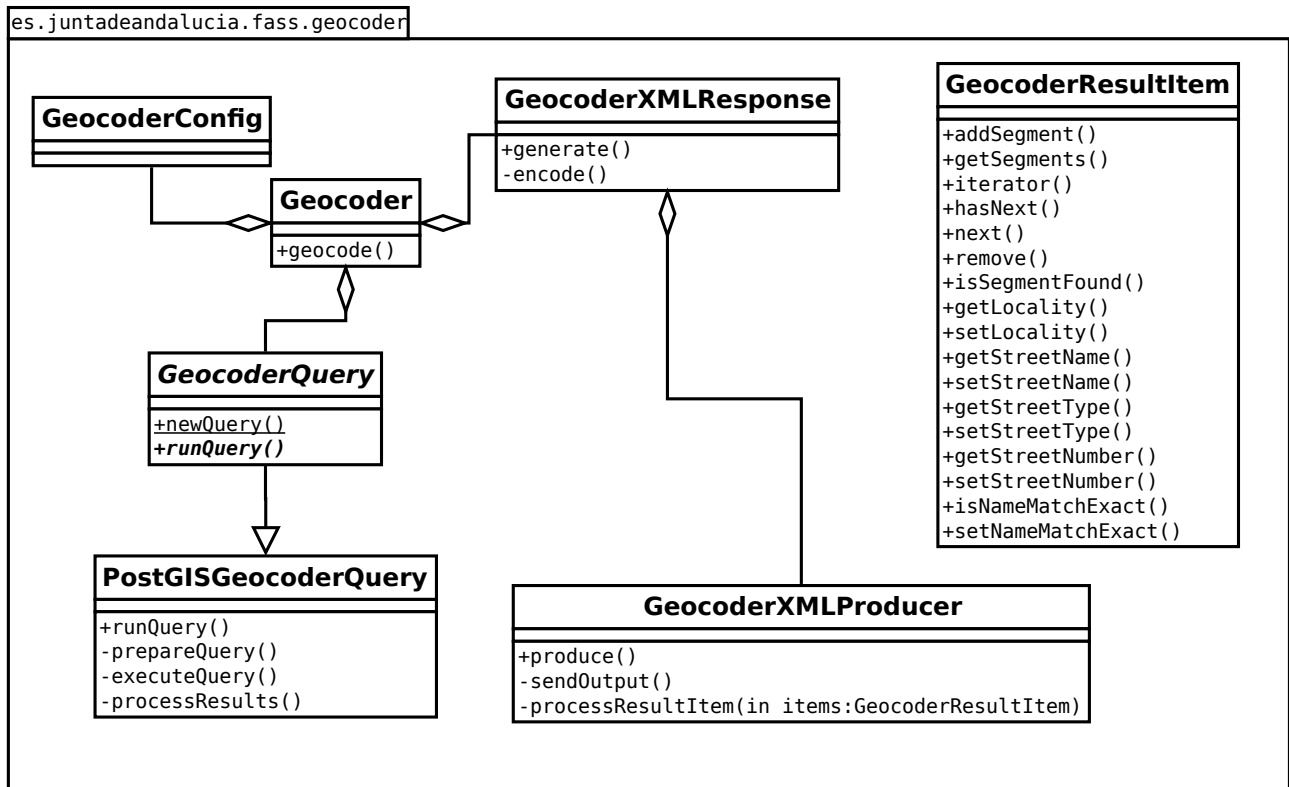


Figura 11: Esquema de clases para el proceso de geocodificación

Servicio de voz

La función del servicio de voz era la de transformar un texto de entrada en un *stream* de audio para posteriormente ser reproducido en la página web. Puesto que para la página se quería un cierto nivel de accesibilidad, se tomó la decisión de proporcionar una lectura en audio de un texto generado a partir de los resultados obtenidos en la búsqueda de la página web.

Este servicio se diseñó alrededor de una estructura de clases que permitiera cambiar el motor de voz a texto si fuera necesario, sólo requiriendo desarrollar un cliente para cada motor de texto a voz siguiendo una clase abstracta, como se puede ver en el diagrama de clases de las figuras 12 y 13.

Este servicio, además de la generación del audio a partir del texto, se encargaba de codificar este audio en formato MP3, puesto que en este formato comprimido ocupaba menos espacio y hacía más rápida la descarga, permitiendo que la reproducción se iniciara mucho antes que si el audio estuviera en otro formato como WAV.

es.juntadeandalucia.fass.gis.services

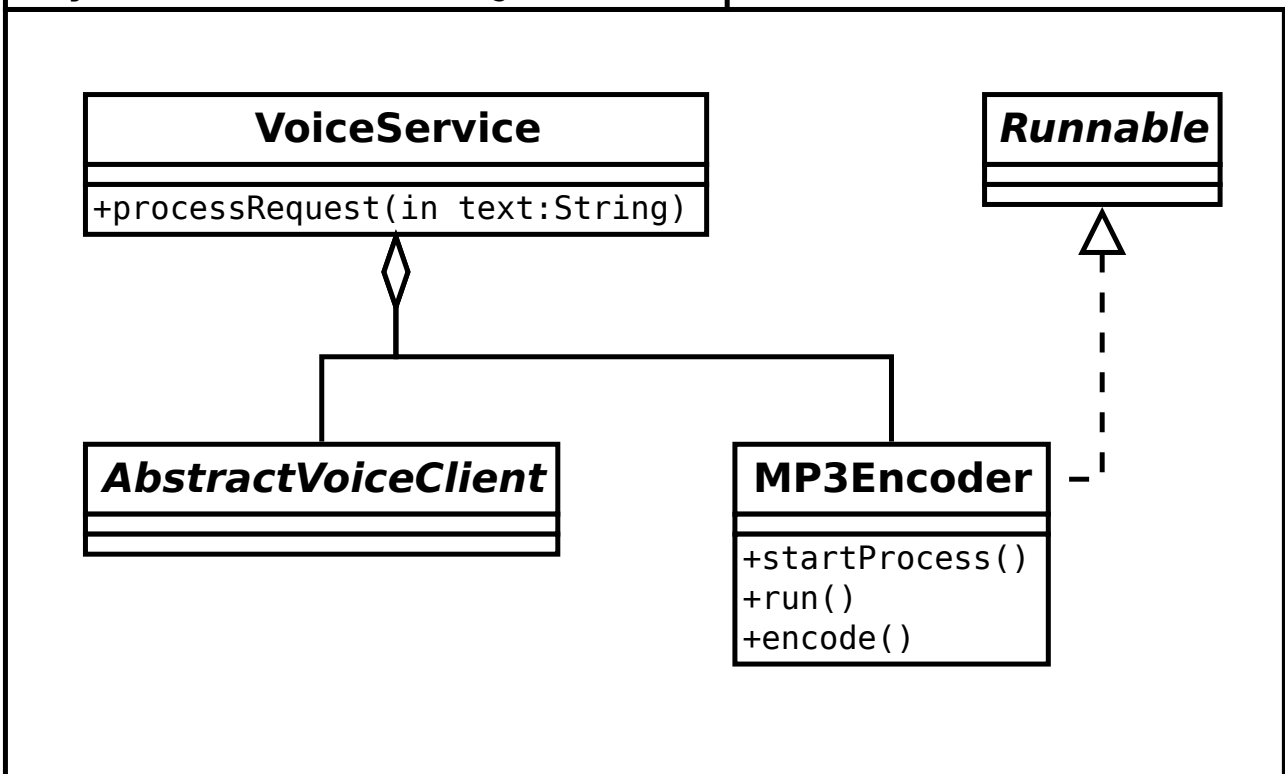


Figura 12: Diagrama de clases del servicio de voz

es.juntadeandalucia.fass.tts

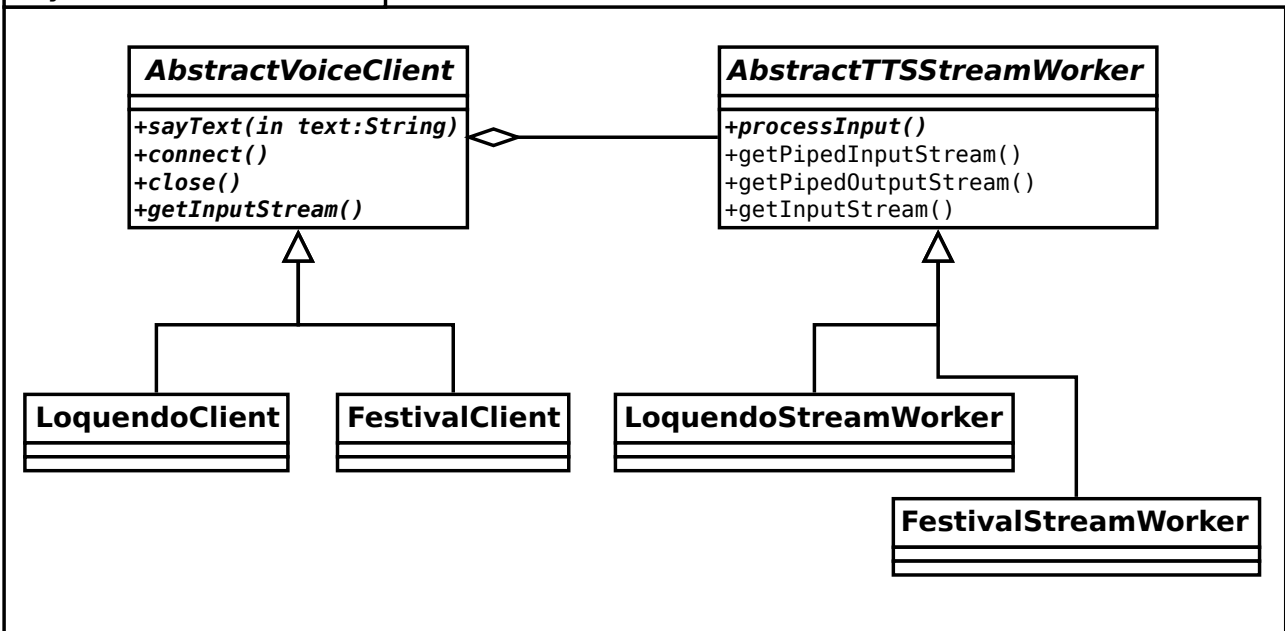


Figura 13: Diagrama de clases del cliente de texto a voz

4.Descripción de los módulos

A continuación se van a describir cada uno de los componentes de la aplicación desde el punto de vista funcional.

En la figura 14 se puede observar cómo están relacionados entre sí en el flujo de las peticiones web desde el navegador del usuario.

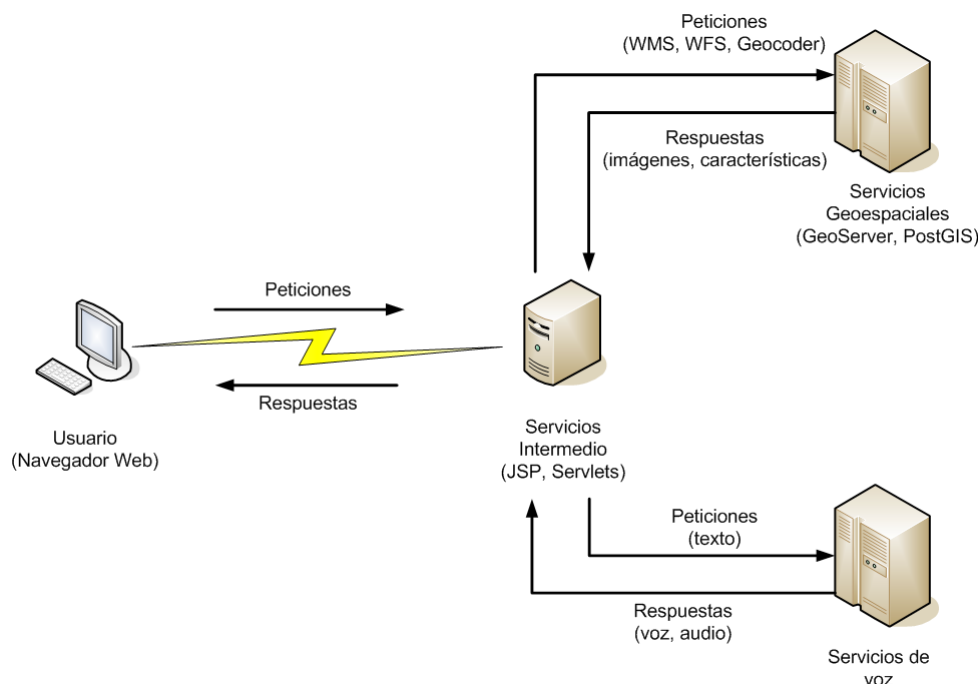


Figura 14: Esquema del sistema completo

1 Servicios intermedios

Los servicios intermedios son los servlets o páginas JSP alojados en el servidor Tomcat de la aplicación. Estos servicios responderán con el contenido generado por el servlet invocado o bien con la página JSP que resulta de la redirección de la solicitud.

Estos servlets o páginas responden a cada una de las funcionalidades independientes de la aplicación web, que son las que se especificaron en la parte de toma de requisitos y diseño, a saber:

- Página y servlet de búsqueda.
- Página y servlet de detalles.
- Servlet de características geoespaciales.
- Servlet de mapas.

- Servlet de contenido de voz.
- Servlet de imágenes.
- Servlet de edición.

A continuación se va a desarrollar la funcionalidad y la estructura de cada uno de ellos.

Página y servlet de búsqueda

La página de búsqueda ofrece al usuario la posibilidad de buscar recursos sociales de distintos tipos según unos parámetros determinados. Estas búsquedas se pueden hacer de tres formas distintas, y cada una de ellas ofrece una serie de parámetros:

- **Búsqueda por dirección.** Este tipo de búsqueda ofrece los siguientes parámetros:

Parámetro	Descripción
tipo_vía	El tipo de vía sobre el que se quiere hacer la búsqueda
nombre_vía	El nombre de la vía sobre el que se quieren buscar los recursos sociales
numero_vía	El número de la vía sobre el que se quiere buscar
provincia	La provincia en la que se quieren buscar recursos sociales
localidad	La localidad sobre la que se quiere buscar
distancia	Radio máximo dentro del cual se van a buscar recursos sociales

Tabla 1: Parámetros de búsqueda por dirección

- **Búsqueda por localidades.** La búsqueda por localidades ofrece únicamente los siguientes parámetros:

Parámetro	Descripción
provincia	La provincia sobre la que se desean buscar los recursos sociales
localidad	La localidad en la que se desea buscar

Tabla 2: Parámetros de búsqueda por localidad

- **Búsqueda por código postal.** La búsqueda por código postal ofrece el siguiente parámetro de búsqueda:

Parámetro	Descripción
codigo_postal	El código postal sobre el que se desean buscar los recursos sociales

Tabla 3: Parámetros de búsqueda por código postal

Además de los parámetros generales, cada tipo de recurso puede tener sus propios parámetros según sus características según el diseño de la aplicación, aunque son únicamente los recursos del tipo CASE los que tienen definidos parámetros propios:

Parámetro	Descripción
servcome	Si el CASE debe tener servicio de comedor, o no
servludo	Si el CASE debe tener servicio de ludoteca, o no

Tabla 4: Parámetros de búsqueda para los recursos de tipo CASE

Una vez que el usuario ha confirmado el formulario para realizar la búsqueda, estos parámetros llegan al servlet de búsqueda el cual determinará primero el tipo de búsqueda que se desea realizar, y según el tipo de búsqueda invocará unos servicios adicionales o no.

Búsqueda por dirección

Este es el tipo de búsqueda más complejo, pues para poder realizar la búsqueda en la base de datos geoespacial, primero debe obtener una coordenada a partir de la cual realizar la búsqueda de recursos sociales en el radio determinado en los parámetros. Para este proceso, el servlet de búsqueda utilizará el servicio de geocodificación, cuya finalidad es precisamente esa, transformar una dirección en un par de coordenadas. En el apartado del servicio de geocodificación se explicará con más detalle su funcionamiento.

Una vez traducida la dirección a un par de coordenadas, se buscarán en la base de datos los centros que cumplen con las condiciones de localización en un círculo determinado por el punto obtenido y el radio determinado en los parámetros de búsqueda.

Si no se encontrara ninguno, se mostraría una página informando que para el área indicada no se ha obtenido ningún resultado. Sin embargo, si hubiera alguna coincidencia, se mostraría la página de resultados con la información de los recursos sociales encontrados, incluyendo un mapa centrado en la posición obtenida que mostraría los recursos y con un nivel de zoom que mostrara la extensión total que incluyera a estos. Esta página, además, haría una lectura a voz de los resultados obtenidos, para ayudar a las personas con deficiencias visuales. Ver la figura 15 para un ejemplo de página de resultados.

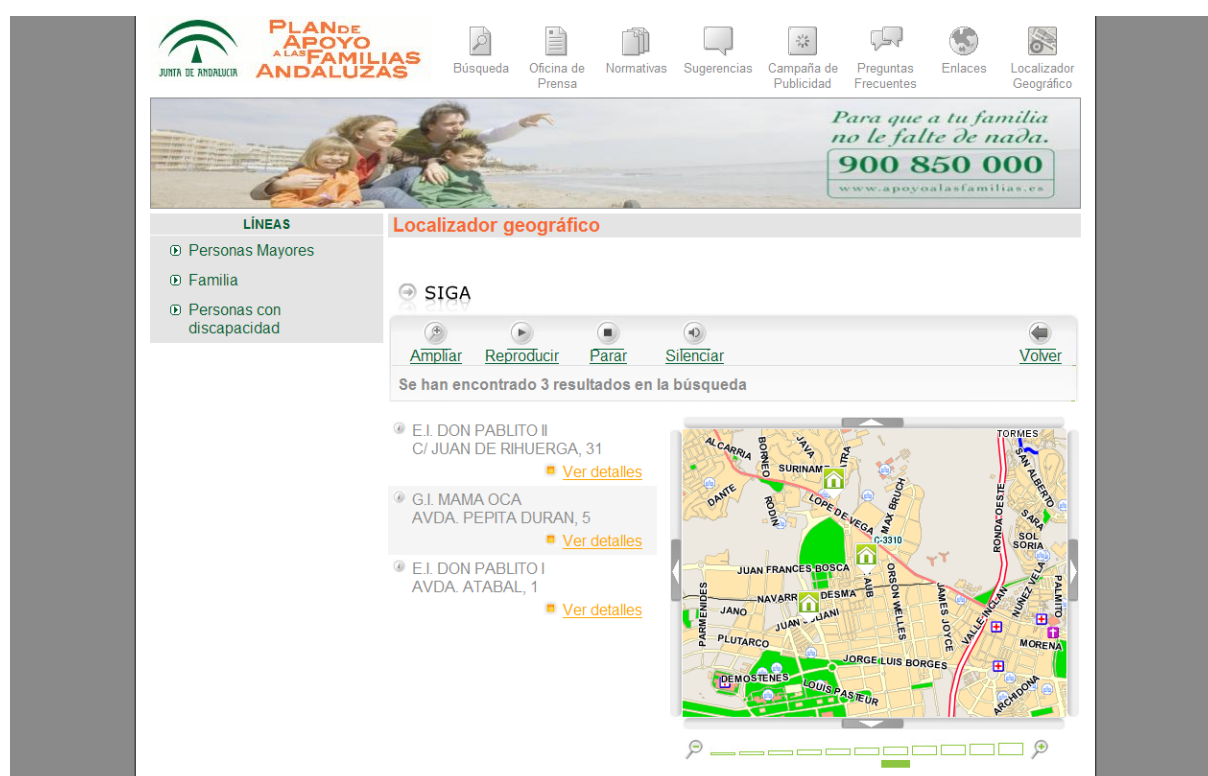


Figura 15: Página de resultados de la búsqueda

Si la dirección no se pudiera traducir de forma inequívoca, se mostraría una página informando de las posibles coincidencias para que el usuario elija la dirección exacta sobre la que desea realizar la búsqueda de recursos sociales, como se puede observar en la figura 16 para un ejemplo de página de refinamiento de dirección.

En dicha imagen se puede ver el formulario que presenta la aplicación cuando se pide localizar la dirección y únicamente se ha introducido “Juan” como nombre de la vía. Se mostrarían todas las posibles coincidencias y se deja en manos del usuario seleccionar cuál es la vía que se deseaba localizar de entre todas las posibilidades encontradas.

Búsqueda por localidad

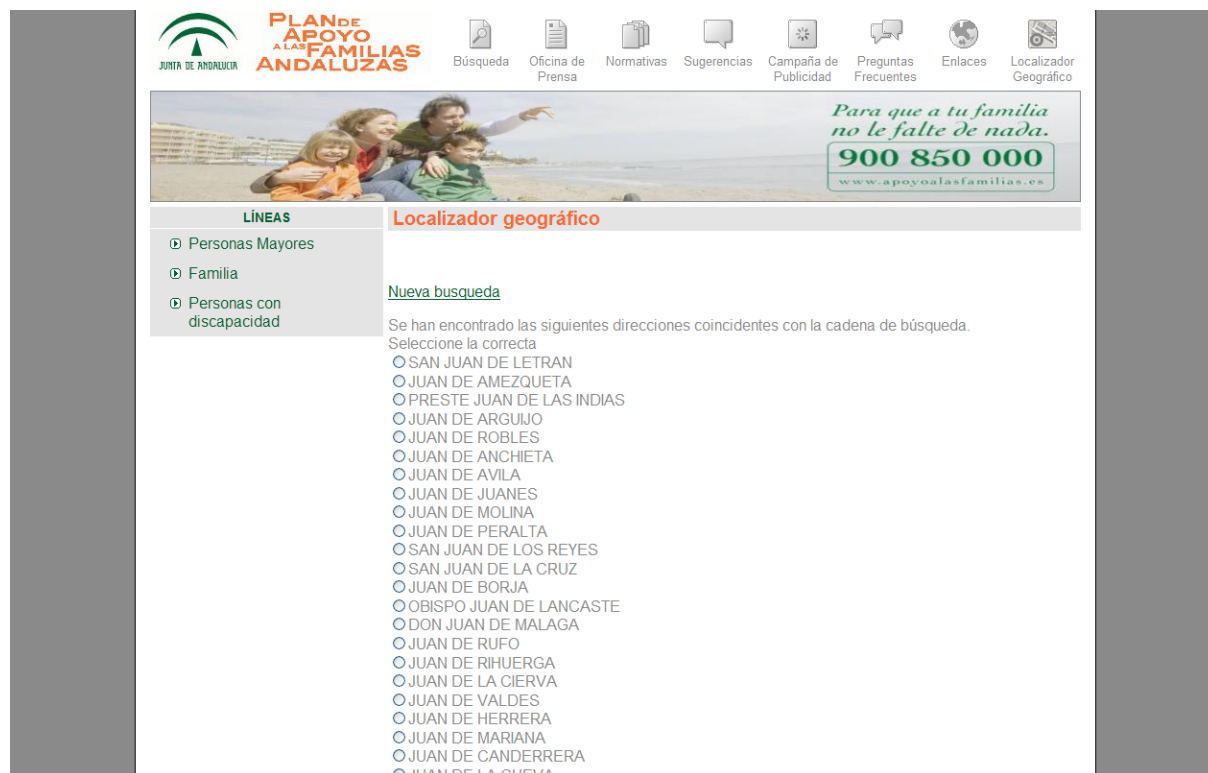


Figura 16: Página de refinamiento de la dirección

Este caso de búsqueda es más sencillo, pues el usuario sólo debe seleccionar en primer lugar la provincia sobre la que desea realizar la búsqueda de los recursos sociales, y a continuación la localidad, de un desplegable que muestra las existentes en la provincia seleccionada.

El servlet de búsqueda localizará los centros que estén situados en la localidad solicitada y mostrará los resultados en la página de resultados, junto con el mapa y la lectura de las coincidencias. Ver la figura 15 para un ejemplo de página de resultados.

En caso de no existir ninguna coincidencia, se mostraría una página informando que no se han encontrado los recursos sociales solicitados en la localidad introducida.

Búsqueda por código postal

Este último caso es el más sencillo de todos, pues el usuario únicamente debe introducir el código postal sobre el cual realizar la búsqueda de los recursos sociales.

El servlet de búsqueda buscará los centros que pertenezcan al código postal introducido y mostrará la página de resultados junto con el mapa y la lectura de los resultados, como se puede ver en la figura 15.

Página y servlet de detalles

El servlet de detalles recibe la petición para mostrar los detalles de un recurso social.

Este servlet buscará en la base de datos el recurso social solicitado, mediante los parámetros incluidos en la URL que definen la clave del recurso social y el tipo de recurso que se desea visualizar.

Una vez recuperado de la base de datos, genera el texto correspondiente a la descripción de este tipo de recurso y lo almacena para su posterior conversión a voz, y por último, redirige la petición a la página JSP de detalles, que compondrá la vista que se presentará al usuario, como se puede ver en la figura 17.

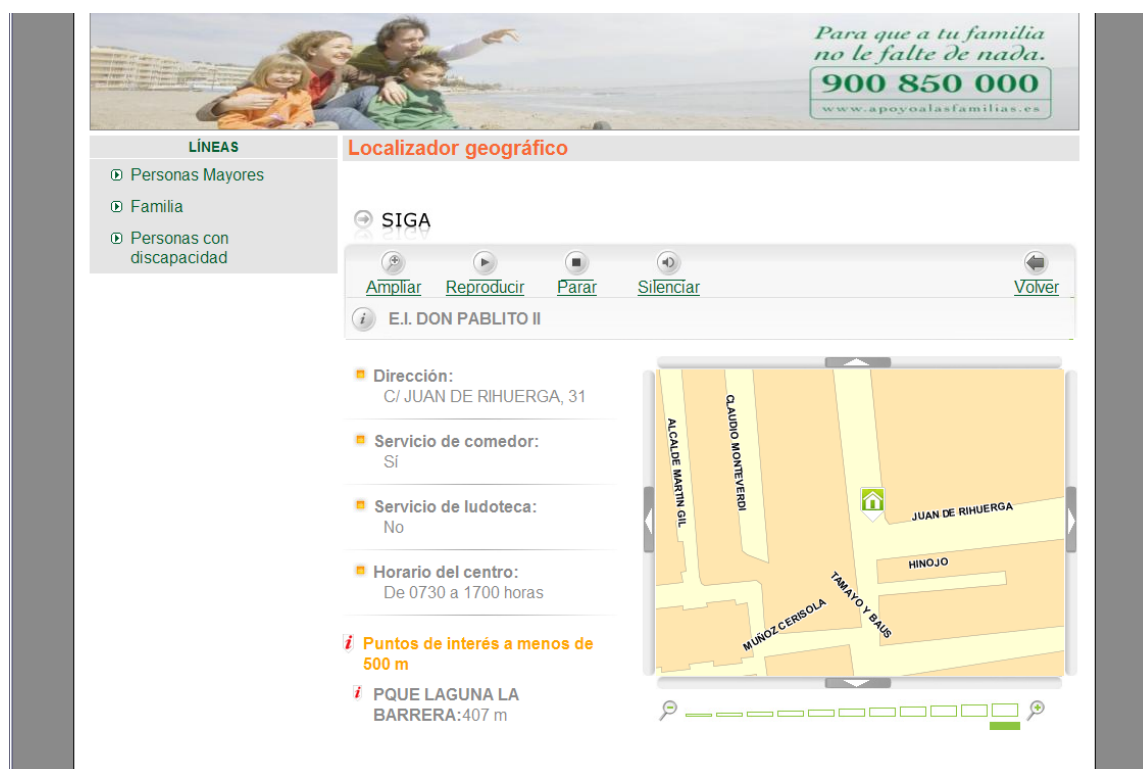


Figura 17: Página de detalles del recurso social

En esta página se presenta al usuario la información almacenada en la base de datos sobre el recurso, junto con un mapa que muestra la localización del mismo el cual permite la navegación mediante la barra de zoom y los botones direccionales existentes en los bordes del mapa, y una descripción mediante voz de la misma información ofrecida de forma textual, que se puede controlar mediante unos botones disponibles en la barra superior que permiten reproducir el audio si este ha terminado o ha sido detenido, parar la reproducción del audio o bien silenciar el audio para que no vuelva a sonar en ninguna página. Además, se muestra también un listado de puntos de interés situados hasta 500m de la posición del recurso social, para facilitar la referencia de la localización.

Servlet de mapas

Este servlet se encarga de devolver un mapa como imagen a partir de las coordenadas recibidas como parámetros.

El mapa generado tendrá la extensión delimitada por las coordenadas (x1, y1), (x2, y2) que definen las esquinas superior izquierda e inferior derecha, respectivamente del rectángulo de visualización.

Una vez extraídas las coordenadas de la solicitud, contacta con el servicio WMS de GeoServer a través de una petición GET construyendo una URL conforme a la especificación del Open Geospatial Consortium para peticiones WMS, según el formato indicado más abajo en el apartado de los servicios geoespaciales. Los parámetros que debe recibir son los siguientes:

- REQUEST
- SRS
- VERSION
- FORMAT
- WIDTH
- HEIGHT
- LAYERS
- STYLES
- BBOX

En la configuración de la aplicación se especifican los valores fijos de este servicio en un archivo XML, que son:

Parámetro	Valor
REQUEST	GET
SRS	EPSG:4326
VERSION	1.0
FORMAT	image/png
WIDTH	400
HEIGHT	300
LAYERS	Lista de capas separada por comas
STYLES	Lista de estilos para cada una de las capas, separados por comas

Tabla 5: Valores por defecto para la petición WMS

Una vez recibida la imagen desde el servicio WMS, devuelve estos datos al cliente en la respuesta a la petición HTTP.

Servlet de contenido de voz

Este servlet se encarga de generar el audio para un texto determinado. En los parámetros de la petición se incluirá la clave temporal del texto generado en una petición a un servlet anterior que se desea convertir a voz.

El proceso de este servlet será el siguiente:

1. Extraer el texto del archivo temporal generado en la llamada al servlet anterior.
2. Enviar el texto al servicio de voz de Loquendo, el cual devolverá el audio en formato WAV
3. Procesar el audio en formato WAV para convertirlo a MP3
4. Devolver en la respuesta el audio convertido a MP3

El motivo por el cual el servlet devuelve el audio en formato MP3 en vez de WAV es porque el primero ocupa menos ancho de banda en la transferencia de los datos, ahorrando así tiempo de espera al usuario a la hora de usar la aplicación web.

Servlet de imágenes

Este servlet se encarga de devolver al usuario las imágenes solicitadas de los recursos sociales.

Dichas imágenes se encuentran almacenadas en la base de datos referenciadas por la clave primaria del recurso social, por lo que para extraerlas es necesario recibir dicha clave en la solicitud, además del tipo de recurso en cuestión.

En la base de datos se encuentran las fotos de los recursos en dos tamaños, uno en tamaño completo y otro en miniatura para una tira de imágenes. Cuando el usuario hace click en la página de detalles del recurso social en una miniatura, se muestra la foto a tamaño completo.

Servlet de edición

Este servlet permite, a través de una página especial que solicita un nombre de usuario y una contraseña, actualizar la posición de los recursos sociales si este estuviera mal ubicado.

Este servlet recibirá de la página la clave primaria del recurso, el tipo de recurso y la nueva posición para el mismo, buscará el recurso en la base de datos y actualizará la posición.

El motivo de la existencia de esta página, como se ha explicado anteriormente, es porque en la fase de importación de los datos se ha realizado una geocodificación automática de las direcciones que puede no haber sido muy precisa y haber dejado algunos recursos mal ubicados, o no haber podido encontrar una ubicación para los recursos si el proceso de geocodificación ha fallado. De esta forma, desde los servicios provinciales de la Junta de Andalucía, se utilizaría esta página para que cada provincia corrigiera la ubicación de sus recursos sociales.

2 Servicios geoespaciales

Los servicios geoespaciales son los correspondientes a GeoServer, alojados en su propio contenedor de aplicaciones. Estos son los servicios Web Map Service (WMS) y Web Features Service (WFS). El primero se encargará de proporcionar una imagen de la cartografía para la página web, mientras que el segundo extraerá las características especificadas en los parámetros de la petición.

Adicionalmente se desarrolló un servicio que consistía en la normalización de las direcciones introducidas por los usuarios. Este servicio es el servicio de geocodificación, y fue necesario desarrollarlo ya que en el momento no había ningún servicio estandarizado que realizara esta transformación y limpieza de los datos introducidos por el usuario.

Servicio WFS

Web Feature Service es un protocolo estándar para proporcionar un interfaz de peticiones para obtener información sobre características geoespaciales de las capas de datos configuradas en el servidor.

Las operaciones que ofrece el servicio WFS son las siguientes:

- GetCapabilities: Esta función devuelve las capacidades del servidor WFS.
- DescribeFeatureType: Esta función devuelve el esquema XML que permite a un cliente WFS procesar los resultados de las peticiones a la función GetFeature.
- GetFeature: Esta función devuelve las características geoespaciales que coinciden con los parámetros de búsqueda especificados.

Los parámetros de las peticiones a un servicio WFS se pueden especificar de dos formas:

- Como parámetros codificados en una petición HTTP GET en pares de clave y valor.
- Como parámetros en formato XML, introducidos en el cuerpo de una petición HTTP POST o bien una petición SOAP.

En este proyecto solamente se ha usado la función GetFeature para obtener información sobre puntos de interés en el mapa para una extensión determinada en la vista de detalles del mapa.

Estos puntos de interés se incluían en los detalles del recurso social para facilitar su ubicación mediante estas referencias.

Adicionalmente, en la vista del mapa de resultados de búsqueda de muestra un mapa que contiene la localización de todos los resultados, y las posiciones de los recursos sociales encontrados se obtenían mediante una llamada a la función GetFeature, con los parámetros necesarios.

El formato en el cual se realizan las peticiones al servicio WFS en este proyecto ha sido mediante peticiones HTTP POST con los parámetros de búsqueda en formato XML en el cuerpo de la petición.

Los parámetros de búsqueda se definen según un esquema XML definido por el Open Geospatial Consortium, y permite criterios de búsqueda como los siguientes:

- Localización de la geometría según unas condiciones determinadas como inclusión en un área, intersección entre áreas, etc...
- Comparaciones entre los atributos de las entidades con valores determinados (igualdad, mayor, menor, etc...).

En este proyecto se han usado únicamente los criterios de búsqueda de inclusión de la geometría en un círculo determinado por un punto y un radio, y comparación de atributos de las entidades, como el código postal o la localidad.

Servicio WMS

Web Map Service es un protocolo estándar para proporcionar imágenes de mapas, desarrollado por el Open Geospatial Consortium.

Este protocolo cuenta con varias funciones para obtener información:

- **GetCapabilities:** Esta función devuelve las capacidades que ofrece el servidor de WMS. Entre estas capacidades están los formatos de imagen soportados por el servidor y las capas configuradas, informado de la máxima extensión, el sistema de referencia de coordenadas, la dirección de los datos y si la capa es transparente o no, para cada una de ellas.
- **GetMap:** Esta función devuelve una imagen de un mapa. Los parámetros para la petición están explicados más abajo.
- **GetFeatureInfo:** (Opcional) Esta función devuelve información sobre una coordenada en una capa del mapa, si esta capa permite peticiones de información.
- **DescribeLayer:** (Opcional) Esta función devuelve los tipos de características de la capa o capas especificados.
- **GetLegendGraphic:** (Opcional) Devuelve una imagen de la leyenda del mapa.

De todas las funcionalidades que ofrece el protocolo WMS, en este proyecto se va a utilizar únicamente la que genera imágenes de mapas. Para ello, hay que hacer una petición web al servidor WMS mediante una URL con unos parámetros determinados, que se explican a continuación:

Parámetro	Descripción
REQUEST	Valor fijo GETMAP que determina que es una petición para obtener un mapa
SRS	El sistema de referencia espacial utilizado en la cartografía. En concreto, el sistema usado es el EPSG:4326
VERSION	La versión del estándar WMS que se está usando, que en este caso es la 1.0
FORMAT	El formato MIME en el que se quiere recibir la imagen del mapa
WIDTH	El ancho en pixels que tendrá la imagen del mapa
HEIGHT	El alto en pixels que tendrá la imagen del mapa
LAYERS	Las capas que se desean incluir en el mapa, como una lista de nombres separados por comas
STYLES	Los estilos que se deben aplicar a cada una de las capas, como una lista de nombres separados por comas
BBOX	El rectángulo que contendrá la vista sobre el mapa, como una lista de coordenadas separadas por comas: x1,y1,x2,y2, siendo (x1,y1) la coordenada de la esquina superior izquierda y (x2,y2) la coordenada de la esquina inferior derecha

Tabla 6: Parámetros de una petición WMS

De estos parámetros, como se ha explicado antes, los hay que tienen un valor fijo pues definen características generales de la aplicación, como por ejemplo los parámetros del tamaño de la imagen generada o el formato de la misma.

Sin embargo, otros como el parámetro que define la vista a mostrar irá cambiando para cada recurso social y también con la navegación del usuario a través del mapa.

Servicio de geocodificación

El proceso de geocodificación consiste en traducir una información textual como una dirección en un par de coordenadas y consta de dos fases, la fase de normalización, donde se limpia la cadena de entrada, y la fase de geocodificación, donde se obtiene una coordenada para la dirección introducida.

Este es uno de los procesos más difíciles en la fase de normalización puesto que implica el reconocimiento del lenguaje natural, ya que la dirección de entrada es muy raro que venga según un esquema determinado y la mayoría de las veces los componentes de una dirección como el tipo de vía, nombre y número, la localidad y el código postal, se ordenan sin ningún criterio normalizado.

Los pasos para poder geocodificar correctamente una dirección consisten en los siguientes:

1. Normalización de cada uno de los componentes: Los componentes de una dirección pueden venir en cualquier formato, por lo que el primer paso es homogeneizar cada uno de ellos. Por ejemplo, los tipos de vía pueden seguir un código determinado, y lo primero es estandarizar su presentación. Así, para el tipo de vía introducido “avenida”, en el sistema de normalización se puede almacenar como “AVDA”.
2. Eliminación de duplicados: Una dirección puede ser dada de forma redundante, como por ejemplo “calle avenida de los guindos”, lo cual proporciona dos tipos de vía, por lo que el sistema de normalización deberá eliminar la información duplicada según lo que esté definido en la base de datos. En este caso, la información duplicada es “calle” y la dirección quedaría como “avenida de los guindos”.
3. Búsqueda de las coincidencias: Una vez limpiada la dirección de entrada, hay que realizar el proceso de búsqueda de las coincidencias con la información almacenada en la base de datos. El procedimiento habitual para esta fase es utilizar un modelo de cadenas de Markov y obtener una puntuación para cada uno de los resultados, de forma que se pueda establecer un orden sobre la idoneidad de cada uno de los resultados, y si alguno obtuviera una puntuación superior a un nivel mínimo de confianza, poder tratarla como un resultado correcto.
4. Interpolación del número de la vía: Una vez que se ha encontrado un resultados que ofrezca un nivel de confianza adecuado, el último paso es interpolar la posición según los tramos que tenga definidos en la base de datos. Las capas de viales se organizan en formato de líneas, de forma que cada tramo tiene las coordenadas de inicio y fin y el rango de números que incluye en ese tramo, tanto para el lado par como para el lado impar de la vía. De esta forma, hay que buscar el tramo que incluye el número solicitado en la dirección de entrada e

interpolando en la línea entre la coordenada de inicio y la coordenada de fin para obtener la coordenada exacta donde se encuentra el número de la vía solicitado.

Puesto que el solo desarrollo de un sistema de normalización de direcciones conforma un proyecto en si mismo, y dado que el tiempo disponible para la realización de esta aplicación era limitado, se decidió realizar un servicio de geocodificación básico que simplemente comparara textualmente la dirección introducida con los datos disponibles en la base de datos. De esta forma, si la coincidencia era exacta se podía tener la certeza de que era el resultado esperado, y si había varias coincidencias, se presentaba al usuario una lista para que seleccionara el valor correcto.

Una vez normalizada la dirección, se realizaba la interpolación del tramo y se obtenía la coordenada exacta que se estaba buscando para el número de vía introducido.

3 Servicio de voz

El servicio de voz corresponde a la lectura a voz de un texto determinado. Este servicio recibe un texto como entrada y como salida se proporciona el audio en MP3 correspondiente a la lectura de ese texto.

El sistema se configuró para poder usar un sistema de texto a voz mediante un interfaz común, como se ha visto en la parte que explica el desarrollo de la aplicación. De esta forma, si en un futuro surgiera un nuevo sistema de texto a voz, implementando este interfaz se podría sustituir el sistema actual.

El servicio de voz recibe un texto como entrada y lo envía al sistema de texto a voz configurado en la aplicación. Este sistema devuelve un audio en formato RAW de 16 bits, el cual es convertido a MP3 usando el compresor de audio LAME.

Cuando el audio se ha convertido a formato MP3, se devuelve a la aplicación como un array de bytes para su posterior reproducción a través del *plug-in* de Adobe Flash.

5.Instalación

1 Instalación y configuración de PostgreSQL + PostGIS

La versión de PostgreSQL utilizada en esta plataforma fue la 8.1, y la versión de PostGIS fue la 1.3.6. Puesto que la plataforma se ejecutaría sobre el sistema operativo Windows Server 2003, se descargó de la página web de PostgreSQL el instalador para Windows de la versión 8.1 y se procedió a su instalación (actualmente no disponible).

Posteriormente se descargó de la página web de PostGIS el instalador para Windows de la versión 1.3.6 [PGIS] y se instaló, lo cual añadió a la instalación de PostgreSQL las librerías dinámicas necesarias para trabajar con los tipos y funciones geoespaciales en la base de datos.

Instalación de la cartografía

Una vez dispuesto el software de la base de datos, lo siguiente fue crear la base de datos que iba a almacenar la información de la aplicación. Puesto que aún no se había profundizado en el diseño de la aplicación no se podían crear las tablas de las entidades, pero como sí se disponía de la cartografía, esta sí podía importarse en la base de datos.

La cartografía fue proporcionada por el Instituto Cartográfico de Andalucía, el cual tenía una recopilación de mapas vectoriales a diferentes escalas de la comunidad autónoma. En concreto los mapas utilizados en esta plataforma fueron los siguientes:

- Cartografía general de las localidades. Consistente en una capa de polígonos que definen la silueta de cada una de las localidades de las provincias andaluzas, que además se encuentran etiquetados con el código INE, el nombre y la cantidad de población, entre otros metadatos.
- Cartografía general de carreteras. Compuesta por una capa de líneas o multilíneas que definen cada una de las carreteras de la comunidad. Estas líneas se encuentran etiquetadas con el nombre de la vía y el tipo de vía, entre otros metadatos.
- Cartografía de parcelas urbanas. Compuesta por polígonos que definen las parcelas y manzanas urbanas de cada uno de los núcleos de población andaluces. Incluyen además polígonos de interés de zonas urbanas, como masas de agua o de bosque o parque. Estos polígonos se encuentran etiquetados con metadatos, entre otros con el tipo de polígono (parcela urbana, zona verde, zona hídrica) y el código INE de la población a la que pertenecen. Esto será de importancia a la hora de crear la representación gráfica del mapa, pues permitirá utilizar un color determinado para el polígono, eligiendo el color crema para

las parcelas urbanas, el color verde para las zonas verdes y el color azul para las zonas hídricas, por ejemplo.

- Cartografía de viales urbanos. Compuesta por las líneas o multilíneas de las calles, plazas y demás viales urbanos. Entre otros, están etiquetados con los metadatos de código INE de la población a la que pertenecen y el nombre de la vía.
- Cartografía de localidades. Compuesta por los puntos que determinan la posición de las localidades dentro de Andalucía. Estos puntos se encuentran etiquetados con los metadatos del código INE de la localidad, su nombre y la cantidad de población, entre otros.

La cartografía se entregó en formato Shapefile, este formato es un formato propietario de ESRI y se considera un estándar *de facto* en el campo de la cartografía, del que se dispone de la documentación técnica proporcionada abiertamente por ESRI. El formato Shapefile consiste realmente en un archivo de datos que contiene la información de las coordenadas y los metadatos, y un archivo de índice que contiene las referencias a cada una de las entidades geoespaciales. La estructura del archivo de datos es la de un archivo dBase y este se puede consultar con cualquier programa que abra este tipo de archivos, como por ejemplo Microsoft Access u OpenOffice Base, pudiendo visualizar la estructura y el contenido de los datos y los metadatos asociados a las entidades geoespaciales.

La herramienta shp2pgsql

La importación de la cartografía en la base de datos de PostgreSQL se realizó con la herramienta *shp2pgsql* proporcionada dentro del paquete de PostGIS. Esta herramienta recibe como entrada un archivo Shapefile y produce como salida la serie de comandos de SQL necesarios para crear la tabla para el tipo de entidad geoespacial, y para importar la información geoespacial dentro de la misma.

Además permite proporcionar otros parámetros que ofrecen funcionalidades adicionales como cambiar el sistema de referencia espacial al realizar la conversión o indicar el nombre de la columna que almacena la geometría.

Una vez convertidos todos los archivos Shapefile de cada uno de los tipos de entidad geoespacial, estos se importaron en la base de datos de PostgreSQL ejecutando las secuencias de comandos SQL generadas por la herramienta.

Más adelante en el capítulo se verá cómo se manipula una tabla con tipos de datos geoespaciales y qué procedimientos hay que seguir para su creación.

La estructura de una base de datos geoespacial

Una base de datos de PostgreSQL que tiene características geoespaciales usando PostGIS consta de las siguientes partes:

- Tipos de datos geoespaciales para representar las entidades:
 - POINT: Un punto en el plano o el espacio.
 - LINESTRING: Una línea en el plano o el espacio.
 - POLYGON: Un polígono en el plano o el espacio.
 - MULTIPOINT: Un conjunto de puntos.
 - MULTILINE: Un conjunto de líneas.
 - MULTIPOLYGON: Un conjunto de polígonos.
 - GEOMETRYCOLLECTION: Una colección de geometrías de distintos tipos.
- Una tabla de sistemas de referencia geoespaciales. las proyecciones disponibles para las entidades geoespaciales y sus definiciones y su nombre es SPATIAL_REF_SYS. La definición de esta tabla es la siguiente:

```
CREATE TABLE spatial_ref_sys (  
  srid          INTEGER NOT NULL PRIMARY KEY,  
  auth_name     VARCHAR(256),  
  auth_srid     INTEGER,  
  srtext        VARCHAR(2048),  
  proj4text     VARCHAR(2048)  
)
```

- La tabla de entidades geoespaciales. esta tabla contendrá, además de la columna de geometría, las columnas adicionales para los metadatos asociados a cada una de las entidades. Esta tabla es usada para poder obtener de una forma rápida cuáles son las columnas de tipo geométrico de un esquema de base de datos y poder preparar las estructuras de datos necesarias en el programa que las vaya a utilizar, al obtener el sistema de referencia espacial y las dimensiones de las mismas. Programas que utilicen esta tabla son, por ejemplo, GeoServer y MapServer. El nombre de esta tabla es GEOMETRY_COLUMNS y su definición es la siguiente:

```
CREATE TABLE geometry_columns (
    f_table_catalog    VARCHAR(256) NOT NULL,
    f_table_schema     VARCHAR(256) NOT NULL,
    f_table_name       VARCHAR(256) NOT NULL,
    f_geometry_column  VARCHAR(256) NOT NULL,
    coord_dimension    INTEGER NOT NULL,
    srid               INTEGER NOT NULL,
    type               VARCHAR(30) NOT NULL
)
```

- Funciones para crear entidades geoespaciales, a partir de su representación en texto o binaria, generando una geometría genérica o de un tipo concreto como punto, línea, polígono, etc.
- Funciones para describir geometrías, como por ejemplo:
 - Comprobar si una geometría está cerrada, en el caso de una línea.
 - Obtener el último punto de una línea.
 - Calcular el borde de una geometría.
 - Obtener las dimensiones de una geometría.
 - Comprobar si una geometría es válida.
 - entre otras...
- Funciones para las operaciones geoespaciales sobre las entidades, como por ejemplo:
 - Rotar geometrías.
 - “Aplanar” geometrías, esto es, reducir las dimensiones de las entidades.
 - Aumentar las dimensiones de una geometría.
 - Escalar una geometría.
 - Invertir el orden de los vértices.
 - Trasladar una geometría.

- Cambiar el sistema de referencia de la geometría, representando sus coordenadas en un nuevo sistema de referencia espacial.
- entre otras...
- Funciones para expresar relaciones entre entidades geoespaciales o calcular medidas de las geometrías, como por ejemplo:
 - Si una geometría está contenida dentro de otra.
 - Obtener el centroide de una geometría.
 - Calcular la longitud de una geometría.
 - Obtener la distancia entre dos geometrías.
 - entre otras...
- Funciones para procesar geometrías, como por ejemplo:
 - Envolvente convexa, que devuelve el polígono convexo mínimo que contiene a las geometrías indicadas.
 - Crear una nueva geometría a partir de un conjunto de geometrías.
 - Calcular los puntos que están a una distancia determinada de una geometría.
 - Crear una unión de geometrías.
 - Crear una diferencia de geometrías.
 - Simplificar una geometría.
 - entre otras...

Para más información sobre las diferentes funciones y tipos de geometrías ofrecidos por PostGIS se puede consultar el manual de PostGIS en el cual están descritas todas las características del mismo.

Crear entidades geoespaciales

Para crear entidades geoespaciales PostGIS ofrece dos formas, bien usando las funciones proporcionadas por la librería que manejan de forma automática toda la metainformación sobre la entidad geométrica, o bien hacer de forma manual todo el proceso. A continuación vamos a ver las dos formas de crear este tipo de entidad.

En ambos casos partiremos del siguiente esquema inicial de una tabla, donde tendremos una columna para la clave primaria, y otra columna de texto que describe a la entidad:

```
CREATE TABLE ejemplo (  
    id integer not null,  
    nombre varchar(50) not null  
)
```

El objetivo que se quiere conseguir es añadir una columna de puntos, de esta forma, esta tabla puede representar, por ejemplo, las ciudades de un país, las gasolineras en una zona, o cualquier otra distribución de puntos en un área.

Procedimiento automático

Una vez creada la tabla con la definición mostrada anteriormente, para añadir una columna de un tipo geoespacial usando el procedimiento automático de PostGIS, simplemente hay que ejecutar la siguiente instrucción que invoca a la función que añade una columna de tipo geoespacial a una tabla, dados su nombre, dimensión, sistema espacial de referencia y tipo de dato:

```
AddGeometryColumn(  
    'ejemplo',      -- Nombre de la tabla  
    'geometria',    -- Nombre de la columna  
    32730,          -- Sistema espacial de referencia  
    'POINT',        -- Tipo de geometría  
    2               -- Dimensiones de la geometría  
)
```

Se puede comprobar que la tabla contiene ahora una nueva columna si se consulta el esquema de la tabla:

```
CREATE TABLE ejemplo (  
    id integer not null,  
    nombre varchar(50) not null,  
    geometria geometry  
)
```

Sin embargo, este método automático no es posible utilizarlo en todas las ocasiones, como se verá a continuación.

Procedimiento manual

El procedimiento manual es recomendable usarlo en determinados casos, como por ejemplo cuando una tabla se ha creado a partir de una copia en lote de los datos de otra tabla, o cuando una vista va a contener un tipo geométrico en una de sus columnas.

Para añadir esta columna a la tabla creada anteriormente, se ejecuta el siguiente comando SQL, que añade una columna a la tabla con el tipo geométrico:

```
ALTER TABLE ejemplo add column geometria geometry;
```

Y posteriormente se registra esta nueva columna en la tabla de columnas geométricas del esquema de la base de datos:

```
INSERT INTO geometry_columns (  
    f_table_schema,  
    f_table_name,  
    f_geometry_column,  
    coord_dimension,  
    srid,  
    type  
) VALUES (  
    'public',  
    'ejemplo',  
    'geometria',  
    2,  
    32730,  
    'POINT'  
);
```

Se puede observar que el método automático proporcionado por las funciones auxiliares de PostGIS es mucho más cómodo que tener que ejecutar estas sentencias manualmente siempre y cuando, claro está, que no se trate de alguna de las situaciones mencionadas anteriormente en las que no es posible utilizar estas funciones auxiliares.

2 Instalación y configuración de Apache Tomcat

La versión de Apache Tomcat utilizada en esta plataforma fue la 5.5.20. Como aún no se conocían detalles técnicos de la aplicación, la instalación llevada a cabo fue la que ofrece el programa por defecto, y más adelante, cuando la aplicación fuera tomando forma, se llevarían a cabo personalizaciones en la configuración del contenedor de aplicaciones.

En primer lugar fue necesario descargar la aplicación, disponible a través de la página de descargas del proyecto Apache Tomcat (actualmente es necesario dirigirse al apartado de histórico de descargas, sin embargo, entonces era la versión actual) [APA].

La distribución descargada fue la que viene en formato ZIP, en concreto el archivo *apache-tomcat-5.5.20.zip*, el cual fue posteriormente descomprimido y el resultado fue colocado en su ubicación definitiva.

3 Instalación y configuración de GeoServer

La versión de GeoServer utilizada en esta plataforma fue la 1.4. El desarrollo de la plataforma coincidió con el desarrollo de esta versión de GeoServer, por lo que durante el ciclo de entregas de GeoServer se estuvieron probando sucesivas versiones candidatas y problemas que se habían encontrado se resolvieron según iba alcanzando la madurez esta versión.

Para obtener el programa fue necesario dirigirse a la página de descargas de GeoServer [GEO] y descargar el archivo *geoserver-1.4.1-war.zip*, el cual es la versión 1.4.1 de GeoServer en formato WAR, que es el formato de aplicaciones web empaquetadas para un contenedor de aplicaciones web Java como Apache Tomcat.

A continuación, hay que desplegar el archivo WAR dentro del contenedor de aplicaciones, y para ello hay que copiar el archivo WAR a la ruta *webapps*, que se encuentra dentro del directorio de Apache Tomcat que se creó en el paso anterior. Una vez hecho esto, al arrancar el servidor de aplicaciones web, desplegará automáticamente la aplicación del servidor GIS.

4 Instalación de la aplicación

Para instalar la aplicación es necesario compilar el proyecto de Eclipse incluido en la documentación del proyecto. Para ello hay que abrir el proyecto con dicho software y generar la aplicación como un paquete en formato WAR, que es el formato empaquetado para aplicaciones web desarrolladas según el estándar J2EE.

Una vez generado el archivo WAR, hay que copiarlo a la ruta *webapps* que se encuentra dentro de la instalación de Apache Tomcat realizada en el paso anterior, y una vez que arranque el servidor de aplicaciones, se desplegará dentro del contenedor y se podrá iniciar su uso.

6. Conclusiones y líneas futuras

Aplicaciones SIG

El ámbito de las aplicaciones SIG, en aquel entonces, presentaba un futuro prometedor, pues las aplicaciones estaban ya lo suficientemente avanzadas y no se disponía aún de una plataforma web lo suficientemente potente y amigable al usuario como las existentes en la actualidad, véanse Google Maps o Bing Maps.

Este proyecto sirvió como inspiración para el desarrollo de la plataforma SIG de la Junta de Andalucía, el cual usaba una arquitectura similar a la expuesta en este.

Proyecto SIGA

Este proyecto no llegó a ser utilizado por el público, debido a motivos de agenda de la Consejería, pese a que se encontraba desarrollado en su totalidad y estaba desplegado en el entorno de producción, esperando la inauguración oficial y la actualización de la página de la Fundación con el enlace a la herramienta.

De haber sido puesto al público, los usuarios hubieran contado con una herramienta moderna y dinámica para la consulta de la información ofrecida por la misma, que en una primera fase se limitaba a los recursos de centros de atención socio-educativa (guarderías), pero en fases posteriores se pensaban incluir más tipos de recursos, como centros para la tercera edad, por ejemplo.

También en otra fase posterior se contemplaba la migración de la información geoespacial de los recursos de PostgreSQL a Oracle, ya que los datos ofrecidos a través de la aplicación procedían de una exportación periódica y automatizada de la base de datos original donde se encontraban esos datos a la base de datos que utilizaba este programa, como ya se ha descrito más arriba. De esta forma se podría simplificar el mantenimiento de esta aplicación al eliminar ese paso, y los datos originales se ampliarían incluyendo la información geoespacial de los mismos gracias a la extensión Oracle Spatial, que fue adquirida por la Junta de Andalucía cerca de la finalización del desarrollo de la aplicación, y permite el almacenamiento y la operación con tipos geoespaciales de forma nativa dentro de la base de datos Oracle. Sin embargo, la información geoespacial relativa a la cartografía como los polígonos de los parcelados, las líneas de las carreteras o los puntos de las poblaciones, se seguiría manteniendo en la base de datos de PostgreSQL.

Una de las mejoras que se podrían haber realizado en la aplicación de haber continuado el proyecto era el uso de un visualizador basado en sectores que permita un desplazamiento más natural, ya que el visualizador actual tiene que generar la imagen del mapa en cada desplazamiento del usuario. De

esta otra forma, el visualizador se va descargando el mapa por sectores y el usuario ve siempre una imagen continua de la cartografía, ya que el visualizador va cacheando los sectores circundantes que se encuentran fuera del rectángulo de la vista. Para ello existe un estándar del OGC llamado Tile Map Service (TMS), que permite esta funcionalidad y además se encuentra implementada actualmente en el servidor GIS GeoServer, y existen visores web de la cartografía que permiten este tipo de navegación por el mapa, como OpenLayers.

7. Glosario

A continuación se explican las abreviaturas utilizadas en el documento:

CLR	Common Language Runtime
GIS	Geographic Information System
OGC	Open Geospatial Consortium
SFS	Simple Features for SQL
SIG	Sistema de Información Geográfica
SQL	Structured Query Language
TMS	Tile Map Service
TTS	Text to speech
URL	Uniform Resource Locator
WFS	Web Features Service
WKB	Well Known Binary
WKT	Well Known Text
WMS	Web Map Service

8. Bibliografía

- **[APA] Página de descarga de la versión de Apache Tomcat 5.5.20**
<http://archive.apache.org/dist/tomcat/tomcat-5/v5.5.20/bin/>
- **[BRI] Tomcat: the definitive guide**
Jason Brittain and Ian F. Darwin
ISBN: 9780596101060
- **[CHA] Understanding .NET**
David Chappell
ISBN: 0321194047
- **[CHO] Professional Apache Tomcat 5**
Vivek Chopra
ISBN: 9780764559020
- **[COC] Agile software development: the cooperative game**
Alistair Cockburn
ISBN: 9780321482754
- **[DUR] Programación orientada a objetos con Java**
Francisco Durán, Francisco Gutiérrez, Ernesto Pimentel.
ISBN: 9788497325721
- **[GEO] Página de descarga de la versión de Geoserver 1.4.1**
<http://sourceforge.net/projects/geoserver/files/GeoServer/1.4.1/>
- **[MAN] Manual de PostGIS versión 1.3**
<http://postgis.refractory.net/documentation/manual-1.3/>
- **[MAR] Agile software development : principles, patterns, and practices**
Robert Cecil Martin
ISBN: 0135974445

- **[OSWC] Open Source World Conference 2008**
<http://malaga08.opensourceworldconference.com/>
- **[PGIS] Página de descarga de PostGIS 1.3.6**
<http://download.osgeo.org/postgis/windows/pg81/>
- **[RIG] Spatial Databases: With Application to GIS**
 Rigaux, Philippe; Voisard, Agnès; Scholl, Michel O.
 ISBN: 9781558605886
- **[SFS] Open Geospatial Consortium Simple Features for SQL**
<http://www.opengeospatial.org/standards/sfs>
- **[SIGTE] I Jornadas de SIG Libre**
<http://www.sigte.udg.edu/jornadassiglibre2007/jornadas.php?id=11>
- **[SIL] Fundamentos de bases de datos**
 Abraham Silberschatz, Henry F. Korth, S. Sudarshan
 ISBN: 8448146441
- **[WIL] Microsoft Visual C# .NET**
 Mickey Williams
 ISBN: 0735612900