UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFM del Máster en Ingeniería Informática

Sistema para el registro y análisis de trayectorias semánticas

Presentado por David Moreno del Hoyo
en Universidad de Burgos — 18 de septiembre de 2017
Tutores: Dr. Bruno Baruque Zanón y
Dr. Santiago Porras Alfonso

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. Dr. Bruno Baruque Zanón, profesor del departamento de Ingeniería Civil, Área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. David Moreno del Hoyo, con DNI 71305821-W, ha realizado el Trabajo final de Máster en Ingeniería Informática titulado Sistema para el registro y análisis de trayectorias semánticas.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 18 de septiembre de 2017

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
Dr. Bruno Baruque Zanón	Dr. Santiago Porras Alfonso

Resumen

El presente Trabajo de Fin de Máster aborda el complejo problema de la gestión de rutas basadas en coordenadas GPS. Cada ruta puede estar compuesta de un número realmente elevado de coordenadas lo que puede repercutir en tiempos de procesado elevados.

En este trabajo se desarrolla una plataforma web que permite realizar una limpìeza y etiquetado de las rutas llevadas a cabo por un usuario concreto, reconociendo las paradas que dicho usuario realiza durante el transcurso de la ruta y asignando semántica a dichas paradas. La citada semántica viene dada por la búsqueda de Puntos De Interés (PDI) cercanos a cada parada detectada.

Descriptores

Plataforma web, ruta, coordenada, PDI, ruta semántica.

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

Web, track, coordinate, POI, semantics.

Índice general

Índice de figuras

Introducción

1.1. Estructura del documento

La estructura del presente documento será la siguiente:

Memoria

En la parte correspondiente a la memoria se encontrarán las siguientes secciones:

1. **Introducción:** se describe el contexto en el que el proyecto ha sido llevado a cabo.
2. **Objetivos del proyecto:** detalla los objetivos marcados a la hora de realizar este proyecto. Se distingue entre objetivos generales, técnicos y personales.
3. **Conceptos teóricos:** realiza una descripción de los conceptos necesarios para la realización del proyecto.
4. **Técnicas y herramientas:** detalla las herramientas usadas para la realización de este proyecto.
5. **Aspectos relevantes del desarrollo del proyecto:** contiene los aspectos más importantes o interesantes del desarrollo del proyecto resumiendo la experiencia práctica del mismo.
6. **Estado del arte:** realiza una comparación con otros trabajos o proyectos relacionados con este.
7. **Conclusiones y líneas de trabajo futuras:** por último se incluirá un conjunto de ideas resultantes de la experiencia recabada durante el desarrollo de este proyecto.

Anexos

Los anexos quedan divididos en las siguientes secciones:

- **Anexo I Plan de proyecto software:** permite mostrar la planificación temporal, económica y legal del proyecto.
- **Anexo II Especificación de requisitos:** este anexo muestra la especificación de requisitos de la plataforma web.
- **Anexo III Especificación de diseño:** detalla la estructura de los datos, el prototipado de la aplicación y la arquitectura del sistema.
- **Anexo IV Documentación técnica de programación:** permite explicar cómo instalar el entorno virtual necesario para ejecutar el sistema.
- **Anexo V Documentación de usuario:** en el manual de usuario se detallan los aspectos fundamentales a la hora de hacer uso de la plataforma.

Objetivos del proyecto

A continuación se detallan los objetivos que se han llevado a cabo en este proyecto. Se han dividido en tres secciones, la primera trata de los objetivos generales (qué se ha querido implementar), en la segunda se tratan los objetivos técnicos (qué herramientas se han usado) y por último se abordan los objetivos personales (por qué se escogió este proyecto).

2.1. Objetivos generales

Los objetivos fundamentales planteados a la hora de realizar este trabajo son los siguientes:

- Implementación de una plataforma web que permita la gestión de todas las tareas relacionadas con la gestión de rutas compuestas por coordenadas GPS.
- La plataforma permitirá el manejo de ficheros de extensión csv (subida, procesado y borrado) con el fin de procesar las rutas de los usuarios.
- Proveerá de un proceso guiado (paso a paso) que permita elegir al usuario los valores que usará el algoritmo implementado a la hora de buscar paradas y PDIs en las rutas. Además, permitirá indicar si se desean almacenar en el sistema los resultados obtenidos.
- Para lograr el objetivo anterior se usará un diseño web adaptable (Responsive Web Design), con lo que cualquier dispositivo será válido para visualizar el portal web.

2.2. Objetivos técnicos

En esta sección se van a tratar los objetivos de carácter técnico que se han pretendido alcanzar al desarrollar este trabajo.

- Implementación de un algoritmo que permita el análisis de datos y convertirlos en información útil para el usuario.
- Desarrollo de la plataforma web de la manera más compatible con todos los navegadores.
- Uso del patrón MVC para la separación de datos, interfaz y lógica de negocio.
- Uso de ProstgreSQL como SGBD, accediendo al mismo mediante librerías JDBC.

Para llevar a cabo estos objetivos se han analizado ciertos parámetros como el mejor Sistema Operativo disponible, el SGBD que más se adapta a los requerimientos técnicos o las librerías de datos. En los siguientes apartados se aporta este análisis.

Sistema Operativo

Ubuntu es un sistema operativo basado en GNU/Linux y distribuido como software libre y de código abierto. Aunque la última versión estable es la 17.04, el sistema está basado en la 16.04 LTS que cuenta con un soporte extendido. Podría haberse usado una distribución de servidor pero para facilitar el desarrollo de la plataforma web se optó por una de escritorio que contase con interfaz gráfica.

Al ser una distribución de soporte extendido suele recibir mayor soporte y perduran un mayor tiempo. Por tanto, los administradores de sistemas suelen ser reticentes a sustituir un sistema que funciona y sobre el que han invertido muchas horas de trabajo, además de que los cambios requieren una inversión elevada de tiempo y recursos y eso, en una empresa, se traduce en dejar de ganar dinero.

Este soporte extendido implica actualizaciones del sistema por un mínimo de 5 años.

Ubuntu frente a otros Sistemas Operativos

Aunque Ubuntu fue el sistema elegido, no fue el único analizado para dar soporte a esta plataforma web. Otro de los sistemas analizados fue CentOS, usado de forma amplia en muchos entornos de servidor.

CentOS (Community ENTerprise Operating System) es una distribución que trata de ofrecer una plataforma gestionada por la comunidad, de tipo empresarial y de código abierto. Es compatible con Red Hat Enterprise Linux, distribución en la que se basó su desarrollo. Es una versión orientada a servidores y cuenta con un amplio soporte. Las versiones que se lanzan tienen un

tiempo de vida de diez años, significativamente mayor que el habitual para la distribución de escritorio, que suele variar entre los 6 meses (muchas versiones beta para realizar correcciones de la comunidad de cara a Release Candidates) y los 60 (como se ha especificado sobre las versiones Long Time Support de Ubuntu). Desde enero de 2014, CentOS pasó a ser adoptado por Red Hat, por lo que el equipo principal de desarrollo tiene mayor interacción con los equipos de Fedora y RHEL.

Actualmente, CentOS, ocupa el tercer lugar en porcentaje de uso en servidores en producción.

Aunque en principio es un sistema igualmente válido que Ubuntu, se optó por la versión 16.04 LTS debido a que otros desarrollos previos en el que se basa este TFM contaban con versiones previas de este Sistema Operativo como base.

Sistema Gestor de Bases de Datos (SGBD)

Como Sistema Gestor de Base de Datos para el sistema, se ha optado por PostgreSQL (completamente compatible con MySql). La instalación es sencilla en este sistema operativo y se obtienen resultados óptimos con cargas de datos de hasta 1TB.

Además se necesitaba hacer uso de una extensión llamada PostGIS que añade soporte para objetos geográficos, por tanto, la elección del SGBD queda limitada por esta extensión que permite la realización de análisis espaciales mediante sencillas consultas SQL.

Mapas basados en OpenLayers

Para mostrar las rutas se deseaba hacer uso de un mapas, inicialmente se trató con la API de OpenStreetMaps y, posteriormente, se ha terminado usando OpenLayers.

OpenLayers es una biblioteca de JavaScript de código abierto con la que el usuario puede disfrutar de sencilla interacción con el usuario permitiendo señalar zonas en un mapa determinado mediante una figura como puede ser un cuadrado, un triángulo, etc. También permite dibujar rutas mediante coordenadas geográficas (latitud, longitud) y añadir marcadores en lugares determinados.

Cabe destacar que OpenLayers aporta un gran juego a los mapas ya que permite añadir una infinidad de capas con mayor funcionalidad que la usada en esta plataforma web permitiendo alcanzar los objetivos planteados con un gran margen.

Librerías necesarias

Para poder implementar un mapa basado en OpenLayers ha de descargarse la librería correspondiente a la versión de la que se quiere hacer uso.

Estas librerías se pueden obtener desde Internet (mediante un enlace a las mismas) o pueden ser descargadas y guardadas para ser usadas de forma local. También son necesarias nociones en JavaScript/jQuery puesto que se hace uso de este lenguaje para pintar los gráficos.

Datos

Los datos necesarios para pintar una ruta sobre el mapa son obtenidos mediante llamadas Ajax a la Base de Datos. Una vez obtenidos se han de formatear de forma específica en coordenadas y transformar las coordenadas al sistema soportado por OpenLayers. En este caso, se transforman las coordenadas desde el sistema EPSG:4326 a EPSG:3857.

Nota: el *European Petroleum Survey Group* o EPSG fue una organización científica asociada a la industria petrolera en Europa. Formada por especialistas topólogos cartógrafos, entre otros, compiló y difundió el conjunto de parámetros geodésicos EPSG. Siendo los dos códigos mencionados anteriormente alguno de los sistemas de coordenadas con los que cuenta EPSG.

Distribución de la plataforma web

En este trabajo se ha desarrollado una plataforma web completa y funcional. Por tanto, uno de los objetivos es poder distribuir esta aplicación en Internet. Para ello, se podrá contratar un hosting, en el que será posible migrar el sistema de una forma sencilla.

Un valor añadido de esta plataforma es su instalación y funcionamiento sin necesidad de un acceso a Internet permanente. Para ello se cuenta con un servicio virtualizado (como es una máquina virtual) en el que se encuentra instalada la aplicación.

2.3. Objetivos personales

El presente Trabajo de Fin de Máster ha supuesto un reto personal. A la hora de comenzar el proyecto no contaba con conocimientos en extensiones como PostGis para la consulta espacial sobre una Base de Datos ni tenía destreza con tecnología como JSP. Tampoco conocía OpenLayers ni había hecho un gran uso de la funcionalidad de Open Street Maps.

Durante los meses en los que se ha trabajado sobre este proyecto, se ha realizado un esfuerzo por aprender las tecnologías necesarias así como todos los conceptos teóricos asociados a las mismas.

Es por estos motivos por los que inicialmente decidí escoger este proyecto, entendiendo que proporcionaría la oportunidad de aprender nuevas técnicas y herramientas durante todo el transcurso del desarrollo de este Trabajo de Fin de Máster.

Conceptos teóricos

Esta sección del presente documento tratará dos de los aspectos más importantes en la realización de este trabajo.

La primera parte se centrará en todos los aspectos relacionados con el análisis de datos. Cabe señalar que, para poder emplear en futuros análisis los datos de trayectorias en crudo, obtenidos de distintos usuarios se han de seguir una serie de pasos que permitan limpiarlos y transformarlos en información útil para el sistema. Esta serie de pasos será descrita en esta primera parte de la presente sección.

Por último se verán aspectos relativos al desarrollo de una plataforma web centrada en el usuario donde uno de los aspectos más importantes es la calidad del diseño.

3.1. Conceptos previos

En este apartado se verán los conceptos previos para entender los pasos posteriores aquí detallados.

Área

Toda ruta realizada puede englobarse en un área determinada. Esta área estará marcada por las coordenadas máximas y mínimas de las posiciones que componen la ruta.

Al analizar un conjunto de datos se puede obtener el área formada por todas las posiciones de las rutas que forman el conjunto.

Coordenada geográfica

Una coordenada geográfica está formada por un valor de longitud y un valor de latitud. Estos valores son obtenidos mediante los sistemas GPS o

GLONASS (basados en un sistema de satélites y receptores terrestres). El valor de la longitud estará comprendido entre -180.0º y +180.0º, situando la coordenada en el espacio horizontal del globo terrestre. El valor de la latitud sitúa al receptor en el eje vertical y puede tomar valores de entre -90.0º y +90.0º. Siendo el origen de coordenadas el valor formado por la coordenada (0,0).

Además, una coordenada puede contar con otros valores como la marca de tiempo en la que fue tomada o la altura de la misma sobre la superficie terrestre.

Ruta

Cada ruta contará con una serie de coordenadas geográficas que conforman el recorrido que, en este caso, el usuario ha realizado. En la ruta se podrán descubrir "paradas" (explicado en los siguientes apartados) sobre las que se puedan asignar Puntos De Interés.

El origen y fin de la ruta estará marcado por el orden de las coordenadas geográficas y/o por la marca temporal de las mismas.

Puntos De Interés

Un PDI o Punto De Interés (en ingles *POI*) permite marcar un lugar representativo o con características relevantes sobre un mapa. Por ejemplo, se pueden marcar lugares de culto, museos, centros comerciales, farmacias, paradas de autobús, hoteles, etc. El número de PDIs sobre un mapa es muy elevado y permite al usuario tomar conciencia de los lugares de interés que le rodean.

Cada PDI podrá contar con distinta información, el nombre propio del lugar, el tipo de lugar, horarios, localización exacta mediante una coordenada geográfica, etc. Es el tipo de lugar el que recibe el nombre de *amenity* y el que será usado como dato relevante en este proyecto.

Prueba o ejecución del algoritmo

Cada una de las ejecuciones del algoritmo será almacenada como una prueba del mismo y permitirá recuperar los datos con los que fue ejecutado. De esta forma se permite visualizar al usuario todas las ejecuciones que ha realizado y los valores de las mismas para así poder variarlos de la forma correcta para que el análisis de los ficheros de datos sea lo más satisfactorio posible.

3.2. Análisis de datos de trayectorias geo-espaciales

Como se ha descrito anteriormente, en este primer apartado de la presente sección se mostrará cómo funciona de forma teórica el algoritmo de análisis de rutas implementado. El primer paso que se ha de dar es el de la limpieza de datos o *data cleaning*. Después se procederá al reconocimiento de las rutas, posteriormente se buscarán paradas y, antes de guardar los resultados, se buscarán Puntos De Interés cercanos a las paradas detectadas. La Figura 3.1 muestra un esquema de funcionamiento del algoritmo.

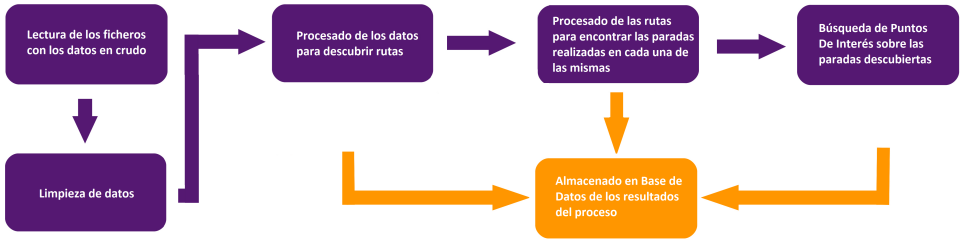

Figura 3.1: Esquema del algoritmo.

A continuación se explica cada uno de los puntos mencionados.

Limpieza de datos

Debido a fallos hardware o software, pueden capturarse rutas que no correspondan con la realidad. Este motivo implica la necesidad de un proceso previo de análisis de los datos para detectar posiciones que no son reales.

Algunos de los problemas que ocasionan que los datos deban pasar por una limpieza previa son los siguientes:

- En ocasiones el hardware GPS puede fallar almacenando una posición errónea que no coincide con el resto de posiciones de la ruta.
- La precisión de los datos también puede verse afectada por una señal pobre en una zona con mala cobertura GPS.
- Este problema también puede ser ocasionado por el software al realizar mal alguno de los cálculos necesarios.
- Posiciones inexistentes por no cumplir con las características inherentes a una coordenada GPS. Por ejemplo, su longitud o latitud sobrepasen los límites (-90.0º - +90.0º para la latitud y -180.0º - +180.0º para la longitud).

- Falta de posiciones intermedias al haberse perdido la conexión y dándose una unión entre puntos de una ruta poco realista, generando una línea recta entre dos puntos cuando en condiciones normales dicha línea debería ser sustituida por puntos adicionales.

Reconocimiento de rutas en cada fichero y asignación de área

Una vez eliminadas las posibles posiciones irreales, el siguiente paso es el reconocimiento de las rutas que existen en el fichero analizado. Dentro de un fichero pueden convivir una o varias rutas dependiendo de la actividad del usuario. Estas rutas han de poder ser encontradas y reconocidas.

En este caso, el algoritmo trata de descubrir estas rutas mediante el cálculo de la mediana temporal aportado por cada marca de tiempo de las posiciones. Una vez obtenida dicha mediana, esta será comparada con las diferencias temporales entre todas las posiciones y, en el caso de encontrarse un gran salto temporal, se calculará el salto espacial entre las posiciones tan alejadas temporalmente. Si estas posiciones, además de estar separadas temporalmente, sobrepasan el umbral espacial marcado al algoritmo, las rutas se considerarán distintas y serán separadas.

Este cálculo se realizará con las rutas restantes y se obtendrá el total de rutas existentes en los ficheros analizados.

Una vez reconocida cada una de las rutas y, dependiendo de las opciones seleccionadas por el usuario, se asignará un área a dichas rutas.

Dependiendo de la opción, se seguirá uno de estos dos caminos:

- Asignación de una área existente: se asignarán todas las rutas analizadas en la ejecución actual a un área ya existente en la Base de Datos. Esta opción es válida si se conoce cuál es el ámbito de las rutas analizadas y se tiene la certeza de que sus coordenadas cumplen con los límites del área seleccionada.
- Creación de una nueva área: el sistema creará un nuevo área con el nombre y descripción asignadas por el usuario. Las coordenadas se calcularán durante el análisis de las rutas.

Segmentación de la trayectoria

El siguiente paso consiste en dividir la trayectoria seguida en “movimiento” y “paradas”. Un periodo de movimiento es el tiempo en el que la persona se encuentra moviéndose entre dos ubicaciones mientras que un periodo de parada es el tiempo en el que, por ejemplo, la persona se encuentra viendo un lugar de interés.

Estas paradas son detectadas una vez calculada la mediana como se ha comentado en los puntos anteriores. En el caso de que existan posiciones cercanas espacialmente pero separadas en el tiempo. Estas posiciones formarán una parada. Entendiendo como parada la entrada en un lugar de interés, la llegada y salida del puesto de trabajo de la persona que porta el dispositivo, etc.

De esta forma se obtienen paradas dentro de la ruta, contemplando el resto de posiciones como movimiento dentro de la misma.

Enriquecimiento semántico

El último proceso realizable es el correspondiente a la asignación de semántica a la ruta. Los datos obtenidos por el usuario solo incluyen posiciones geográficas tomadas en un instante determinado, es decir, solo se tienen datos como la latitud, longitud y el espacio temporal en el que han sido tomados. En esta etapa se incluye información adicional a estos datos.

Este proceso toma las paradas detectadas en la ruta y las contrasta con las posiciones de los Puntos De Interés almacenados en el sistema. Para ello se toma como límite el radio que marca la distancia máxima al que buscar un PDI.

Un PDI obtenido de Open Street Maps puede contener diversa información, dicha información es aportada por la persona que registra dicho dato dentro de los sistemas OSM. Esta información incluye de forma obligatoria la posición espacial del Punto De Interés pero deja al usuario la posibilidad de añadir más información. Generalmente también se cuenta con un nombre en el idioma nativo de la persona que registra el PDI pero en otras ocasiones también existe un nombre genérico en inglés. Adicionalmente, un nodo suele incluir el tipo al que pertenece, este tipo recibe el nombre de *amenity*. Una *amenity* es una generalización de lugar al que representa el PDI. Por ejemplo, pueden existir museos, lugares de culto, gasolineras, centros comerciales, etc. De esta forma se puede conocer el nombre propio del lugar pero también qué tipo de lugar representa.

Una vez se han obtenido los PDIs cercanos a las paradas, todos los datos serán actualizados en la Base de Datos, permitiendo al usuario recuperarlos en el momento que desee, mostrando los resultados en la página de resultados de la plataforma.

Técnicas y herramientas

Esta sección presenta un breve resumen de las técnicas y herramientas que han sido usadas para llevar a cabo este trabajo.

4.1. Técnicas

Modelo-Vista-Controlador

Se ha hecho uso del patrón MVC para el desarrollo del proyecto. Este patrón permite la separación de la lógica de negocio de los datos y las vistas. Como su nombre indica, sus componentes quedan divididos en:

- **Modelo:** representa los datos y la lógica de negocio.
- **Vista:** presenta la información del modelo.
- **Controlador:** controla las entradas del usuario y selecciona la vista.

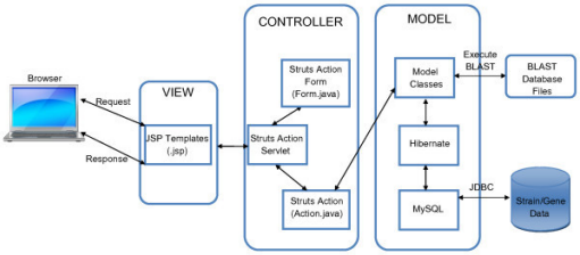

Figura 4.2: Modelo-Vista-Controlador.

En el Anexo referente al Manual de Programación se detalla la estructura del proyecto según este patrón.

Diseño Web Centrado en el Usuario

A la hora de realizar una página web es fundamental tener en cuenta al usuario final de la misma, por tanto, siempre se ha de tener presente que la calidad de diseño es algo fundamental. El diseño de la plataforma modelará la interacción entre el usuario y la aplicación. De esta forma, un buen diseño, facilitará la consecución de los objetivos del usuario, mientras que un diseño pobre hará que el usuario busque otro medio por el que conseguir sus objetivos.

Durante el desarrollo de este proyecto se ha tenido siempre en cuenta el Diseño Centrado en el Usuario. En este apartado se darán algunas nociones sobre este aspecto tan importante a la hora de que un sitio web sea accesible y usable por todo tipo de usuarios.

En esta sección se va a tratar el diseño centrado en el usuario, siguiendo el trabajo presentado en el trabajo realizado por Yusef Hassan-Montero, Francisco J. Martín Fernández y Ghzala Iazza.

Usabilidad y accesibilidad

La usabilidad es un anglicismo que quiere decir que algo es de fácil uso. Aunque hay multitud de definiciones, se puede dar la ofrecida por la Organización Internacional de Estandarización (ISO). La ISO define usabilidad como: "Grado de eficacia, eficiencia y satisfacción con la que usuarios específicos pueden lograr objetivos específicos en contextos de uso específicos". Mediante esta definición se pueden apreciar dos tipos de atributos diferenciados.

- Atributos cuantificables de forma objetiva: como pueden ser la eficacia o el número de errores que comete un usuario al realizar una tarea y la eficiencia o tiempo que ha necesitado el usuario para llevar a cabo una tarea en concreto.
- Atributos cuantificables de forma subjetiva: como puede ser la satisfacción de uso de la plataforma web por parte de un usuario concreto.

Aunque no siempre se va a poder dotar a la aplicación o plataforma de una usabilidad universal, es decir, que sea usable para todos los posibles tipos de usuarios, a la hora de diseñar dicha plataforma, se ha de tener en cuenta quiénes serán los usuarios potenciales y qué necesidades y conocimientos tienen.

Otro concepto ligado de forma íntima a la usabilidad es la accesibilidad. Esto es, el diseño ha de posibilitar el acceso a la información a la mayor cantidad de usuarios posibles. No sólo consiste en facilitar el acceso a personas invidentes, sino que también se ha de prever el acceso desde conexiones lentas, hardware y software no actualizados, etc. En este punto se puede observar una

contradicción ya que un diseño usable puede que deba centrarse en un tipo de usuario en concreto, mientras que la accesibilidad trata de universalizar la posibilidad de acceso de los usuarios al sitio web.

En este proyecto se hace especial énfasis en la usabilidad del sitio, haciendo que los menús y opciones sean lo más sencillos y claros posibles.

Arquitectura de información

Aunque un usuario típico pensará que la interfaz es la aplicación puesto que es la parte de dicha aplicación con la que interactúa, se debe entender que la usabilidad de una aplicación depende en gran medida de su arquitectura, es decir, depende del componente no visible del diseño. Por este motivo, es importante una elaboración óptima del backend (lógica de la aplicación) para que el uso del portal web sea lo más sencillo e intuitivo posible.

La Arquitectura de la Información (AI) se puede definir como "el arte y la ciencia de organizar espacios de información con el fin de ayudar a los usuarios a satisfacer sus necesidades de información". La actividad de organizar la información, incluye la estructuración, la clasificación y el rotulado de los contenidos de la aplicación.

Recuperación de la información

El principal objetivo de un correcto diseño de la AI es que permita al usuario recuperar la información de la forma más sencilla posible. El objetivo de este portal web es permitir al usuario analizar rutas y visualizar los resultados sobre mapas permitiendo ver toda la información disponible mediante el uso de dichos mapas y tablas. De esta forma, un usuario en concreto, podrá comprender de mejor forma los datos obtenidos de un determinado análisis.

Visibilidad

El punto anterior está estrechamente ligado con la visibilidad de la información, es decir, que cada elemento de información pueda ser encontrado. Un usuario que acceda a este portal web, encontrará varias opciones en el menú principal de la aplicación como: "Tratamiento de ficheros", "Visualización de resultados" o "Gestión". Quedando claro el tipo de información que se encontrará en cada página visitada. Si el usuario accede a la página de tratamiento de ficheros, encontrará diversas opciones de gestión de los mismos. Por su parte, si accede a la página de visualización de resultados, podrá encontrar dos secciones diferenciadas. En la primera sección (parte izquierda de la pantalla) podrá elegir el área o áreas sobre las que filtrar las rutas y en la segunda (parte derecha) las rutas encontrada que cumplan con los criterios indicados.

Diseño web centrado en el usuario

Para poder afirmar que una aplicación cumple con los niveles de usabilidad requeridos, el diseñador necesita de una serie de técnicas, métodos y procedimientos diseñados para tal fin.

Como se ha comentado desde el comienzo de esta memoria, el trabajo se ha centrado siempre en el usuario, siguiendo la metodología conocida como Diseño Centrado en el Usuario (User-Centered Design) adaptándose a las necesidades y características de una plataforma web.

El Diseño Web Centrado en el Usuario se caracteriza por asumir que todo el proceso de diseño y desarrollo de la plataforma ha de estar conducido por el usuario final y las necesidades y objetivos del mismo.

Centrar el diseño en el usuario, implica involucrar al mismo desde el comienzo. Se ha de conocer al usuario, lo que necesita, cuál será el uso que den a la aplicación, cómo es la experiencia de uso de la misma, etc.

El DiseñoWeb Centrado en el Usuario, usado en este proyecto, se divide en varios pasos o etapas. Como se ve en la Figura 4.3, las fases de diseño, prototipado y evaluación son cíclicas. Quiere decir que cada diseño deberá ser evaluado de forma constante.

A continuación, se procede a explicar cada una de las fases del Diseño Web Centrado en el Usuario.

Planificación: En este momento se han de identificar todos los objetivos, necesidades y requerimientos de la plataforma web. Se trata de encontrar un equilibrio entre las necesidades del usuario y lo que un programador puede ofrecer. En este sentido no suele ser fácil obtener toda la información necesaria por parte del usuario. Será importante estudiar a los usuarios potenciales para que el sitio se adapte de la mejor forma posible a sus necesidades.

La etapa de planificación, se basará por tanto, en la recogida de toda la información (lo más detallada posible) con el objetivo de poseer una base sólida sobre la que poder tomar decisiones en las siguientes etapas.

Diseño: En esta etapa, se tomarán todas las decisiones de diseño en base a la información obtenida en el paso anterior. La fase de diseño se puede dividir en los siguientes puntos:

- **Modelado del usuario:** se tratará de conocer las necesidades de los potenciales usuarios de la aplicación.
- **Diseño conceptual:** tratará sobre el diseño de contenidos del portal web.

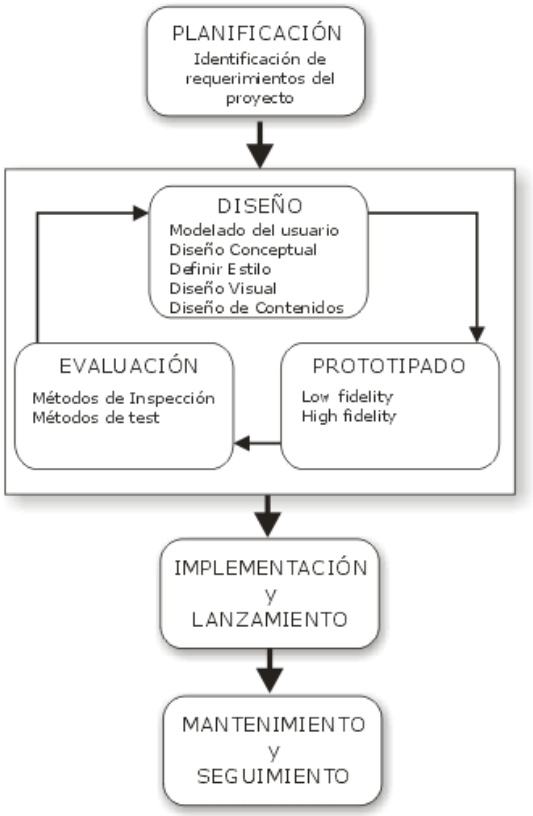

Figura 4.3: Esquema del diseño web centrado en el usuario.

- **Definir el estilo:** se definirá el estilo de todo el sitio web.
- **Diseño visual:** trata de definir el número de colores que serán usados, el tipo de letra, etc.
- **Diseño de contenidos:** en este punto se valorará el contenido que cada sección mostrará al usuario.

Prototipado: Uno de los mayores problemas a la hora de evaluar la usabilidad del sitio web, es que todavía no se encuentra implementado. La solución, pasa por el uso de prototipos, es decir, modelos de la interfaz de la página, que ayudarán a comprobar la usabilidad. Este prototipado puede ser clasificado en:

- **Prototipado horizontal:** reproducen el aspecto visual del sitio, pero no tienen por qué representar la funcionalidad del mismo.

- **Prototipado vertical:** se reproduce sólo una parte del diseño del sitio, pero además incluye toda la funcionalidad final del mismo.
- **Prototipado de alta fidelidad:** este prototipo será idéntico al sitio web ya terminado.
- **Prototipado de baja fidelidad:** este prototipo podrá ser muy distinto del diseño del sitio web una vez finalizado.

Para ahorrar costes y agilizar los primeros pasos, es posible que los primeros diseños estén realizados en papel.

Evaluación: Existen numerosos métodos para comprobar la usabilidad del sitio web, además de poder ser realizada sobre distintas representaciones de la aplicación.

Implementación y lanzamiento: Lo más recomendable a la hora de desarrollar un sitio web, es el uso de estándares, como pueden ser, HTML 5 o CSS 3. Así se asegura una compatibilidad y escalabilidad futura. Otro aspecto importante, es que la hoja de estilos (CSS) esté separada del propio contenido HTML del sitio.

En esta etapa es fundamental supervisar que todo funcione como se había especificado, si algo no funciona, no será usable.

En el momento en el que el sitio está acabado y su funcionalidad testada, es posible ponerlo en producción y presentarlo al usuario. Si se cree conveniente es posible diseñar una serie de páginas de ayuda al usuario novel, que le guíen y ayuden en los primeros pasos sobre el sitio web.

Mantenimiento y seguimiento: Una plataforma web está en constante desarrollo, sus contenidos y audiencia pueden cambiar con el tiempo, el diseño puede cambiar.

Pero es fundamental que ese diseño no varíe de forma drástica de un día para otro siendo estos cambios muy sutiles.

Seguimiento de la usabilidad de la plataforma web

Por último, cabe destacar, que no todos los problemas pueden haber sido descubiertos en las etapas previas. Es importante obtener un feedback (mediante formularios o contacto con el usuario) de la experiencia que el usuario tiene con la web.

4.2. Herramientas

Las herramientas usadas durante la implementación del proyecto se muestran en los siguientes apartados de esta sección.

NetBeans

El IDE elegido para el desarrollo del trabajo ha sido NetBeans en su última versión disponible a la hora de finalizar este proyecto(8.2).

Este entorno ha sido elegido principalmente por haber sido usado en proyectos anteriores y poseer un conocimiento más avanzado del mismo que de otros entornos como puede ser Eclipse.

Atom ha sido usado como editor secundario para manejo de ficheros csv entre otros.

Glassfish

Para desplegar la aplicación web se cuenta con un servidor Glassfih en su versión 4.1.

Open Street Maps

Open Street Maps (OSM) es un proyecto colaborativo que permite crear y editar mapas. Estos mapas son de uso libre y gratuito para todos los usuarios, siendo solo los usuarios registrados los que pueden crear dichos mapas.

La creación de los mapas se lleva a cabo mediante la recogida de coordenadas geográficas usando dispositivos móviles. Estos dispositivos móviles obtendrán su ubicación gracias a sistemas GPS o Glonass o al uso de wifi o a la ubicación proporcionada por el operador de telefonía en caso de que el dispositivo cuente con una tarjeta SIM.

Los mapas son distribuidos bajo licencia ODbL (Licencia Abierta de Bases de Datos).

Los datos se almacenan siguiendo el datum WGS84 (World Geodetic System 1984, constituido por parejas latitud-longitud) usando la proyección de Mercator (definida en 1569 por Gerardus Mercator). La información primitiva consta de:

- Nodos: son posiciones geográficas concretas.
- Vías: una lista ordenada de nodos constituye una vía. Esta puede ser una polilínea o un polígono.

- Relaciones: son grupos de vías, nodos y relaciones que pueden ser agrupadas ya que contienen propiedades comunes.
- Etiquetas: son usadas para almacenar metadatos sobre los objetos del mapa, constan de una pareja clave-valor.

PostgreSQL 9.6

PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) relacional, orientado a objetos y libre, distribuído bajo licencia PostgreSQL License. Actualmente se encuentra en su versión 9.6 siendo la usada en este trabajo.

PostGIS

PostGIS es una Base de Datos espacial que expande las posibilidades de PostgreSQL añadiendo soporte a objetos geográficos permitiendo consultas de localización. PostGIS ha sido diseñado y desarrollado por la empresa Refraction Research siendo distribuido bajo licencia GNU (GPLv2).

QGIS

Para los pasos previos de este trabajo se ha hecho uso del programa QGIS. QGIS es un sistema de información geográfico libre y de código abeirto. Cuenta con una infinidad de opciones, algunas de las más relevantes son:

- Conexión con PostgreSQL: permite mantener una conexión contra la Base de Datos que contiene los datos relativos a los análisis de las rutas. De esta forma es posible pintar sobre un mapa todos y cada uno de los puntos de cada ruta analizada. Es posible adaptar las consultas SQL para obtener la información deseada antes de ser pintada.
- Uso de diferentes APIs de mapas: permite usar mapas de Open Street Maps, Google Maps, etc.
- Descarga de datos OSM: permite la descarga de datos en formato osm indicando el área deseado.

Osmosis

Osmosis es una herramienta desarrollada en Java y que funciona por línea de comandos. Permite procesar ficheros OSM con un gran rendimiento debido a que está diseñada para tratar con ficheros de extensión considerable (un fichero osm conteniendo información completa de un país puede superar los 10 GiB de datos).

Esta herramienta permite desde cargar datos sobre una Base de Datos hasta extraer un subconjunto de ellos a partir de un fichero de tipo osm. Además, Osmosis puede formar parte de otras aplicaciones Java incluyéndose como una librería.

Osmconvert

Esta herramienta permite realizar distintas conversiones entre ficheros de distinta extensión. En el caso de este proyecto, la herramienta permite al usuario transformar ficheros de extensión pbf a ficheros osm con los que trabajar posteriormente.

Aunque es muy sencilla de usar, se facilita un script preparado para extraer ficheros pbf.

Osmfilter

Como su nombre indica, la herramienta Osmfilter permite filtrar ficheros osm mediante sus etiquetas. En este trabajo ha sido usada para extraer ficheros de tipo XML a partir de ficheros de extensión osm manteniendo únicamente los Puntos De Interés del área englobada en el fichero original.

Además, se provee al usuario de un sencillo script para hacer uso de esta herramienta a la hora de incluir nuevos Puntos De Interés sobre el sistema.

LaTeX

Se ha usado LaTeXpara la relaización de la documentación del presente Trabajo de Fin de Máster. El contenido ha sido editado con TexMaker en su versión 4.5.

Lenguajes de programación, marcado y/o SQL

HTML 5

HyperText Markup Lenguage 5 es la última versión del estándar HTML publicado a finales del año 2014 siendo sucesor de la versión 4.01. Incluye nuevas etiquetas dejando otras obsoletas. Es un lenguaje de marcado destinado al desarrollo de páginas web. Es un lenguaje estándar que cualquier navegador web será capaz de reconocer e interpretar.

Java

Java es un lenguaje de programación orientado a objeto concurrente y de propósito general. Java mantiene un número de dependencias reducido para permitir la ejecución del programa desarrollado en el mayor número de dispositivos posibles sin ser recompilado.

La primera revisión de este lenguaje fue desarrollada por James Gosling en la ya desaparecida Sun Microsystems (actualmente absorbida por Oracle), siendo cada aplicación Java compilada y ejecutada por una máquina virtual Java (JVM).

Aunque este lenguaje apareció en 1995, no fue hasta 2012 cuando se convirtió en uno de los lenguajes más usados, especialmente en aplicaciones cliente-servidor. La versión estable actual es la 8.

Servlet

Un servlet es un componente ubicado en el servidor Java EE encargado de dar respuesta a las peticiones de los clientes. Cada servlet amplía las capacidades del servidor en el que se encuentra pudiendo generar contenido dinámico como pueden hacer PHP o ASP.NET.

El ciclo de vida de un servlet es el siguiente:

- Inicialización: el proceso de inicialización (llamada al método init) ha de hacerse antes de que el servlet pueda interaccionar con los clientes.
- Interacción: una vez inicializado, el servlet, puede atender a las peticiones de los clientes.
- Destrucción: para destruir un servlet se ha de llamar a su método destroy. Este método será llamado cuando el servidor sea cerrado o el administrador del sistema lo requiera. Para inicializar el servlet destruido se ha de llamar al método init de nuevo.

JSP

Java Server Pages o JSP es una tecnología de vistas que corre en el lado del servidor permitiendo escribir plantillas en lenguajes como HTML, JavaScript o CSS. Una página JSP puede contener segmentos de código implementados en Java que permitan modelar dicha página de forma dinámica.

Una de las ventajas de JSP es su base en Java, permitiendo crear clases de acceso a datos (DAO), lógica de negocio, etc.

Una página JSP permite sintaxis como la siguiente:

- Directivas: permiten incluir otros ficheros, clases Java para el manejo de objetos, etc.
- Declaraciones: permiten declarar variables, funciones y datos estáticos.
- Expresiones: permiten evaluar expresiones Java.

- Etiquetas: simplifican el código aportando funcionalidad añadida. Existen dos grandes grupos de etiquetas: las que proporcionan funcionalidad a la página (como jsp:include) y las que permiten manipular componentes JavaBean (como jsp:useBean).

CSS 3

Cascading Style Sheet (CSS) es un lenguaje usado para definir la presentación visual de un documento escrito en HTML y permitiendo separar la estructura de un documento de su presentación. El estilo puede ser incorporado "en línea", mediante una hoja interna o mediante el uso de una hoja de estilo externa.

El principal problema que persiste con el paso de las versiones es el centrado vertical ya que no es sencillo y requiere de más reglas que el centrado horizontal (siendo este último casi automático).

Bootstrap 3

Esta plataforma web se ha diseñado pensando en el usuario final de la misma, por tanto, se han tenido en cuenta las pautas marcadas por un diseño adaptable (Responsive Web Design). Este objetivo ha sido posible gracias al uso de un framework como Bootstrap. Este framework permite diseñar el sitio para todos los tamaños de pantalla actuales.

Bootstrap implementa CSS para una gran cantidad de clases, lo que hará que el diseño de la plataforma web sea más sencillo. Unas de las clases más importantes son las que permiten adaptar el contenido mostrado al tamaño de pantalla desde la que la web ha sido accedida.

En Bootstrap el contenido se agrupa en filas (row) y en columnas (col) hasta un máximo de 12 columnas. Por tanto, en las líneas superiores, se crea una fila (row) y el div interno especifica las columnas para un tamaño de pantalla xs o md (estos tamaños ya están predefinidos en Bootstrap). De esta forma tan sencilla se puede agrupar el contenido de ese "div" mediante el CSS que ya contiene el framework.

Bootstrap también contiene gran cantidad de estilos para menús, botones, listas, además de colores recibiendo el nombre de componentes. Se pueden encontrar iconos que se podrán usar en botones, en texto, menús de usuario simples o con accesos desplegables, mensajes de alerta con colores llamativos para el usuario, barras de progreso, etc.

JavaScript y jQuery

JavaScript (JS) es un lenguaje interpretado, débilmente tipado y dinámico, que cumple el estándar ECMAScript siendo la última versión estable la

ECMAScript 2016. En el desarrollo de una página web, se usará en el lado del cliente, permitiendo la generación de páginas dinámicas o la mejora de la interfaz de usuario. En la actualidad, la gran mayoría de los navegadores son capaces de interpretar este lenguaje. JavaScript está provisto de una implementación del DOM (Document Object Model) para poder interactuar con la página web.

Aunque se pueda pensar que Java y JavaScript están relacionados, en realidad, no es así. Inicialmente JavaScript fue diseñado con una sintaxis similar a C aunque adoptando nombres y convenciones similares a las de Java.

jQuery es una biblioteca de JavaScript (presentada en enero de 2006) que pretende simplificar la interacción con los documentos HTML, la manipulación del DOM o el manejo de eventos. Además de permitir una interacción con la técnica AJAX. En la misma web que la usada para JavaScript se pueden encontrar gran variedad de ejemplos y referencias para el uso de jQuery

En jQuery, se usa la función $ para interaccionar con las páginas HTML. Recibe como parámetro el nombre de una etiqueta HTML o una expresión en CSS, devolviendo todos los nodos del DOM que concuerdan con esa expresión. Si queremos que esto funcione en Drupal, tenemos que dar un paso más, y es que debemos englobar esta función dentro de otra.

AJAX

Asynchronous JavaScript And XML, conocida como AJAX, es una técnica de desarrollo web que permite la implementación de aplicaciones interactivas. Las aplicaciones se ejecutarán en el navegador del cliente y, por detrás, se mantiene una comunicación asíncrona con el servidor. Así, es posible, que la página sea actualizada sin necesidad de realizar una recarga completa de la misma. Con AJAX, se mejora la interactividad, velocidad y usabilidad de estas aplicaciones. AJAX no es una tecnología en sí sino un término que engloba a un conjunto de las mismas que trabajan de forma conjunta. HTML y CSS: para la presentación de la información. Document Object Model (DOM): para mostrar la información e interaccionar con la misma. JSON: para intercambiar datos. XMLHttpRequest: para poder mantener una comunicación asíncrona. JavaScript: para unir estas tecnologías. En la siguiente página web podemos encontrar mucha información relativa tanto a JavaScript como a AJAX en particular.

Máquina virtual y sistema operativo

Se ha optado por instalar el Sistema Operativo Ubuntu 16.04 LTS en una máquina virtual bajo el entorno de Virtual Box. De esta forma, se permite exportar e importar la máquina pudiendo ser instalada en cualquier ordenador.

Además, este planteamiento permite hacer uso de la plataforma web de forma offline y sin tener que estar disponible en un servidor.

Además se ha hecho uso de diversos navegadores web como son Chrome, Firefox o Konqueror, para poder comprobar el correcto funcionamiento del sitio, además de la correcta visualización del contenido de la hoja de estilo. Es un punto importante ya que algunos navegadores, en versiones antiguas, puede que no soporten todas las etiquetas del estándar HTML5 o todo los correspondientes estilos de CSS3.

Aspectos relevantes del desarrollo del proyecto

A la hora de realizar un proyecto de este tipo siempre surgen problemas o cuestiones que se han de ir resolviendo en ese momento, esta sección trata los puntos más relevantes.

5.1. Concepto inicial del proyecto

El proyecto planteado, en su forma incial, trataba de la implementación de una aplicación de escritorio en lenguaje Python o Java permitiendo asignar semántica a las rutas analizadas.

Para ello, la aplicación debería poder leer y procesar ficheros de texto plano o csv. Todos los datos obtenidos serían almacenados en una Base de Datos PostgreSQL puesto que cuenta con una extensión llamada PostGIS que permite el análisis y tratamiento de posiciones geográficas añadiendo funcionalidad adicional al SGBD mencionado.

Puesto que la aplicación debía ser de fácil uso, se planteó una interfaz basada en Swing para poder mostrar los datos analizados así como mapas con las rutas procesadas.

Los primeros pasos orientados al diseño de la interfaz mediante Swing dejaron claro que no era la más adecuada ni sencilla para trabajar con mapas. Este es uno de los aspectos que animó a abandonar esta idea inicial por un desarrollo web cuya interfaz permite un manejo mucho más sencillo de todo tipo de mapas y librerías.

5.2. Conocimientos previos

A la hora de realizar este trabajo se contaban con algunos conocimientos previos de programación web obtenidos a lo largo de los distintos cursos universitarios así como del desarrollo del Trabajo de Fin de Grado (basado en el CMS Drupal).

Aunque en esta ocasión el lenguaje elegido fue Java y se completó el desarrollo haciendo uso de páginas JSP y servlets. Por tanto, se tuvo que realizar un esfuerzo para aprender el manejo y programación mediante este lenguaje.

Adicionalmente, no se contaban con conocimientos sobre el funcionamiento de los sistemas de obtención de coordenadas ni de cómo podían ser tratados posteriormente.

Coordenadas geográficas

Debido a los distintos tipos de toma y representación de coordenadas geográficas es posible que una misma coordenada representada sobre distintos sistemas de mapas de lugar a errores de representación de varios centenares de metros. Este comportamiento puede observarse al comparar coordenadas geográficas presentadas por distintos motores de mapas. Por ejemplo, una coordenada localizada en China mostrada un mapa basado en OpenStreetMaps no coincidirá con la misma coordenada mostrada por los mapas de Google.

Esto es así por el uso de distintos sistemas de coordenadas usados por ambos motores (EPSG:3857 en Google Maps frente a EPSG:4326 en OpenStreetMaps).

Al comienzo del desarrollo del trabajo, no se conocían los diferentes sistemas de coordenadas dando lugar a confusión y errores en los cálculos sobre las rutas.

La Figura 5.4 muestra la gran cantidad de sistemas de coordenadas disponibles a la hora de visualizar un fichero de puntos GPS o a la hora de obtener datos desde PostgreSQL.

5.3. Errores propios del hardware

Toda la tecnología GPS relacionada con el hardware cuenta con ciertas limitaciones, tanto en potencia de cálculo como en la toma de las posiciones en sí mismas.

Estas limitaciones favorecen la aparición de errores en las coordenadas medidas por cualquier hardware. Estos errores pueden hacer que la ruta tomada como verdadera contenga pequeños fallos en cada una de las coordenadas de

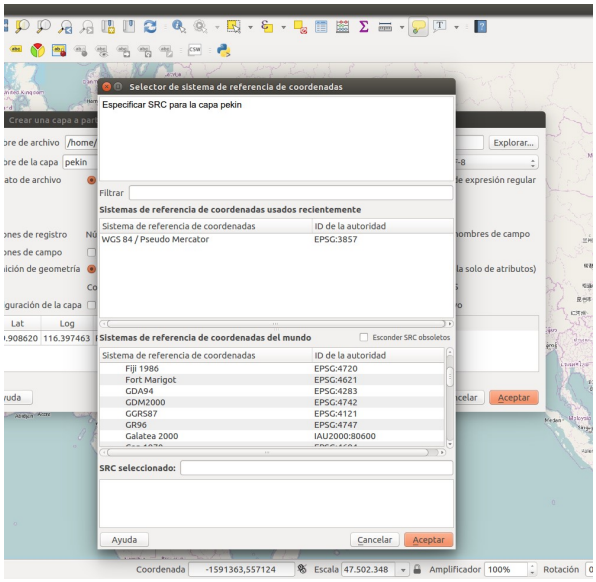

Figura 5.4: Gran cantidad de sistemas de coordenadas en QGIS.

las que está compuesta. Por este motivo, puede que una ruta inicialmente correcta indique un espacio recorrido mayor al realmente realizado o, incluso, marque puntos por los que no se han pasado.

Este echo puede ser fácilmente verificado atendiendo a las restricciones inherentes del hardware y a la forma en la que el software es implementado. Dejando un dispositivo GPS activo en un punto fijo, se verá al cabo de un cierto espacio de tiempo que el dispositivo parece moverse de forma autónoma. Este movimiento viene dado por los errores y/o fallos mencionados anteriormente.

Por tanto, es posible afirmar, que los datos con los que cada usuario cuenta tendrán un cierto error. Esto hace que los algoritmos de análisis y muestra de resultados deban tolerar estas inexactitudes. La Figura 5.5 muestra el movimiento virtual efectuado por dispositivo GPS completamente parado.

Estas restricciones han ocasionado que algunos conjuntos de datos no cuenten con la calidad suficiente para su uso en este trabajo así como que sus errores sean tan cuantiosos que las rutas no sean verosímiles.

5.4. Obtención de datos

Uno de las primeras cuestiones a resolver fue la búsqueda de datos que permitiesen dar comienzo al desarrollo del proyecto. Aunque existen gran variedad de *datasets* en diversas páginas web dedicadas a su alojamiento, es cierto que no siempre cumplían con los requisitos necesarios para proceder a su análisis.

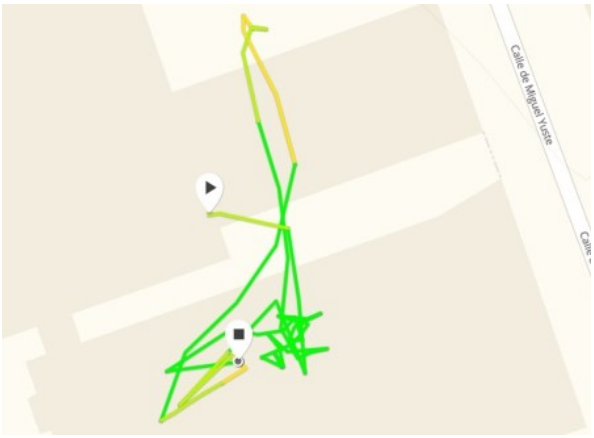

Figura 5.5: Movimiento

Algunos conjuntos de datos contaban con un buen número de ellos pero no contaban con ciertas características como una marca temporal para cada coordenada geográfica. Otros contaban con un buen número de características pero, en conjunto, resultaba un tamaño demasiado pequeño como para ser analizado.

Finalmente se encontró un conjunto de datos de Microsoft que, de forma general, satisfacía los requisitos que se habían marcado como mínimos siendo el usado para llevar a cabo las pruebas del sistema.

En este momento también comenzaron a surgir algunos de los problemas como los que se indican en apartados inferiores.

5.5. Tamaño de los datos de análisis

Como ya ha sido comentado con anterioridad, el proyecto ha sido basado en una máquina virtual. Inicialmente se otorgó un tamaño estándar y fijo para el correcto desempeño de la distribución elegida. Pronto se tuvo constancia de que este tamaño no era suficiente. Aunque ya se habían instalado y configurado todos los componentes necesarios para el proyecto (PostgreSQL, PostGIS, NetBeans, etc.) no hubo otra opción que volver a crear una nueva máquina, esta vez con un tamaño mucho mayor y de espacio dinámico. Una vez reconfigurado todo el software necesario se pudo continuar con el normal desarrollo del trabajo.

No obstante y, como se describe a continuación, no solo crecía el tamaño en disco de todo el sistema, sino que también la Base de Datos aumentaba su volumen.

Antes de comenzar con las pruebas del sistema se cargó la Base de Datos

con un completo conjunto de datos del mapa de China (los datos usados para las pruebas estaban basados en su gran mayoría en la ciudad de Pekín) viendo que el tamaño de las tablas era tan grande que ralentizaba cualquier intento de análisis de rutas.

En este momento se estaba llegando al límite de recursos de hardware que se podían asignar a la máquina virtual (llegando a contar con hasta 8 hilos, 10 GiB de RAM y un tamaño superior a las 100 GiB de espacio en disco). Se debía considerar otra aproximación que lograse una mejor experiencia y un mejor rendimiento.

Pronto se tuvo claro que no era necesaria tan elevada cantidad de datos y, que realmente, solo se usarían datos pertenecientes a la capa de Puntos De Interés. Es en este punto en el que se buscan herramientas para extraer estos PDIs de los ficheros de tipo osm descargados inicialmente. En esta búsqueda se encuentran las herramientas mencionadas en la sección anterior (Técnicas y herramientas).

De esta forma se logra disminuir el tamaño de unas tablas excesivamente grandes logrando una mejora del rendimiento global del sistema. Una vez iniciadas las pruebas del análisis de rutas del sistema se comprueba que el rendimiento ha experimentado una mejora real y se decide optar por mantener la estructura y conjunto de datos usados hasta el momento.

Como cabe esperar si finalmente la plataforma es puesta en producción, se hará uso de un servidor apropiado y con capacidades suficientes para que todo el sistema se comporte de forma correcta quedando disipado este problema .

5.6. Tiempos de ejecución

Ligado a los problemas mostrados en los apartados anteriores, los tiempos de ejecución eran tan elevados que, inicialmente, se necesitaban varias horas para obtener los primeros resultados. En estos momentos el hardware había llegado a su límite y no era posible una mayor asignación de recursos.

Una vez reducido el tamaño de los datos a consultar el rendimiento aumentó de forma considerable y ahora se pueden obtener los resultados esperados en pocos minutos.

5.7. Diseño del algoritmo

Durante los primeros pasos del proyecto se diseñó un algoritmo que incluía todos los pasos del mismo de forma secuencial. Es decir, todos los pasos debían ser ejecutados una vez se lanzase.

Este comportamiento no es el ideal puesto que el usuario puede tener la oportunidad de elegir los pasos que desea realizar sobre los datos que contienen

sus ficheros. Por ejemplo, puede desear que sus datos no se mantengan en el sistema, que una ejecución busque PDIs pero otra solo paradas, etc.

Por tanto se decidió separar el algoritmo en diferentes clases que permitiesen una ejecución del algoritmo más personalizada.

Este echo hizo que el desarrollo se viese ralentizado puesto que el cambio de la implementación del algoritmo implicó más tiempo del planeado inicialmente. Además, en este momento, se llevó a cabo la migración hacia la plataforma web, lo que implicó cambios adicionales sobre el código.

Estado del arte

En este apartado se va a aportar una visión más amplia del "estado del arte" mediante el estudio de otras herramientas similares.

6.1. Weka-STPM

WEKA for Moving Object Data Analysis and Mining o Weka-STPM es una extensión para Weka que permite soportar de forma automática el procesado de trayectorias o rutas para añadir semántica a las mismas. Está desarrollada por la Doctora Vania Bogorny, profesora del Departamento de Informática y Estadística en la Universidad Federal de Santa Catarina (Brasil).

Inicialmente parece una herramienta clave que podía haber facilitado el análisis de las rutas puesto que permitía la lectura y escritura en Bases de Datos PostgreSQL y soportaba la extensión PostGIS, pero tras algunas pruebas se vio que no se obtenían los resultados esperados.

Además, se puede ver por su código, que no está completamente terminada, encontrándose algunos errores en su ejecución y opciones no disponibles. Por ejemplo, aunque se lograba una conexión contra la Base de Datos, no se lograba que la aplicación insertase ningún tipo de resultado en la misma.

Su interfaz (Figura 6.6) basada en Swing tampoco ayuda a comprender su funcionamiento puesto que hay botonoes que quedan ocultos, opciones que no se visualizan y desplegables que desaparecen detrás del resto de menús. Ante los problemas encontrados no se puede afirmar que esta herramienta pueda ser de gran utilidad en la actualidad pero sí es posible garantizar que si su desarrollo finaliza supondrá un gran paso adelante en el análisis de cualquier tipo de trayectoria.

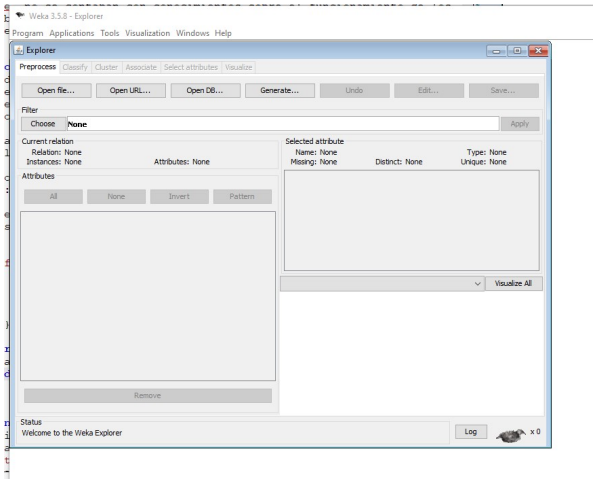

Figura 6.6: Interfaz sencilla de weka.

6.2. CARTO

Carto es una plataforma de cómputo en la nube SaaS (Software as a Service) que provee de un sistema de información geográfica o GIS y herramientas de mapeo de información para navegadores web.

Está basada en la extensión PostGIS de ProstreSQL y es de código abierto. Su *front-end* está basado en JavaScript mientas que su *back-end* está basado en Node.js.

El aspecto que muestran los mapas creados mediante CARTO puede ser visto en la Figura 6.7.

CARTO Builder

CARTO Builder es una aplicación web en la que los usuarios pueden procesar sus datos, realizar anaálisis y diseñar mapas propios. CARTO Builder está pensado y diseñado para todas las personas noveles que desean hacer uso de estas herramientas sin contar con conocimientos previos ni avanzados en sistemas de información geográficos. No obstante se provee de acceso a consultas SQL que pueden ser personalizadas si se cuenta con mayores conocimientos.

CARTO Engine

CARTO Engine provee al usuario de distintas APIs y librerías que permiten la construcción de mapas completamente personalizados.

- SQL: se pueden hacer uso de todas las sentencias SQL soportadas en PostgreSQL.

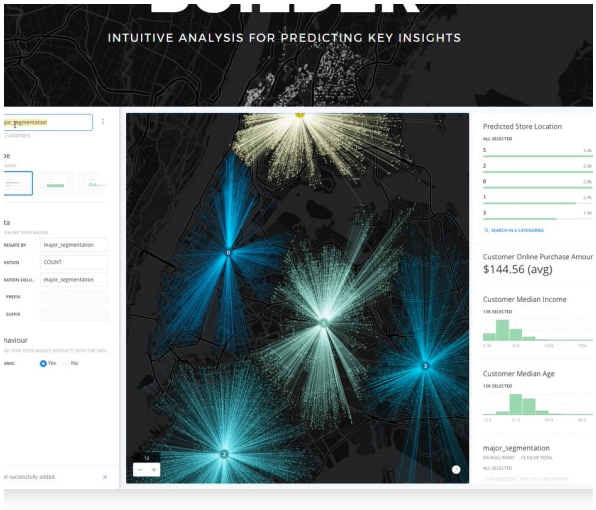

Figura 6.7: Aspecto final de los mapas en CARTO.

- Mapas: permite obtener distintas plantillas de mapas en base a los requerimientos del usuario. De esta forma, cada usuario puede diseñar un mapa propio que usar en una aplicación web.
- Servicio de datos: permite desarrollar una mayor funcionalidad como el **geocoding**.

Aunque como se puede apreciar, su gran variedad de características permite una amplia capacidad de análisis de datos pudiendo ser una herramienta potente y de interesante uso, no se adapta al completo a los requerimientos iniciales de este proyecto, por lo que se decidió no hacer uso de la misma.

6.3. Google Trips

Quizá la primera aproximación a esta aplicación pueda parecer que es la que más se adapta al espíritu de este proyecto. Google Trips (Figura 6.8) permite organizar viajes o rutas en pocos minutos.

También facilita obtener sugerencias de sitios no visitados que guardan relación con los visitados anteriormente y que hayan gustado a su usuario, preparar rutas diarias, organizar reservas según un calendario en Gmail, etc.

Pero el aspecto que esta aplicación móvil no realiza es el análisis y asignación de semántica a rutas ya realizadas, aspecto que diferencia la plataforma desarrollada con esta aplicación.

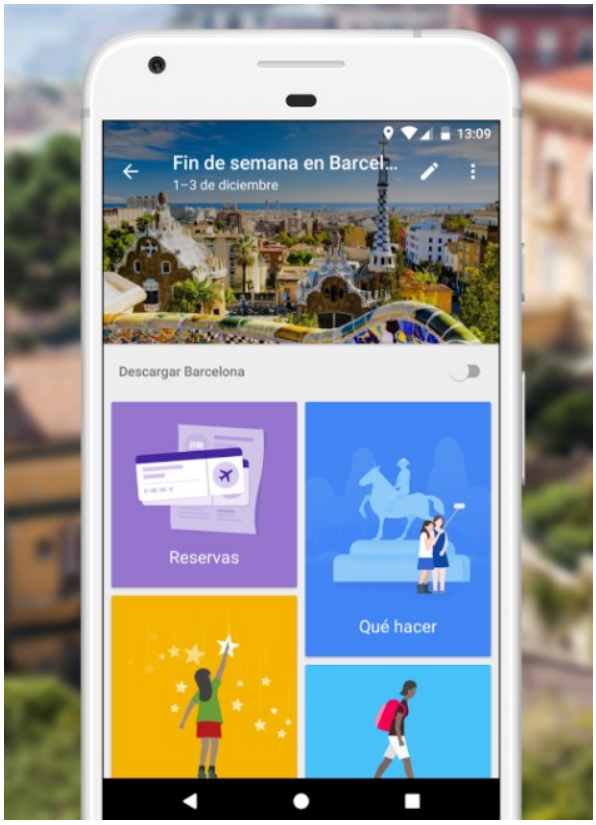

Figura 6.8: Aspecto de la interfaz de la aplicación Google Trips.

6.4. Conclusiones sobre el estado del arte

Como se ha visto en los apartados anteriores, existe una variedad interesante de plataformas y/o aplicaciones desarrolladas con el afán del análisis y/o gestión de rutas basadas en coordenadas GPS. Si bien existen más plataformas similares a las mencionadas, en esta sección se han mostrado las juzgadas como más interesantes.

Comparando sus características se puede ver que estas plataformas no permiten asignar semántica a trayectorias basadas en coordenadas geográficas. Por tanto, se puede afirmar que por los ejemplos mostrados en esta sección de la documentación, la aplicación web aquí mostrada añade nuevas funcionalidades a los sistemas existentes.

Por último, cabe destacar que, la plataforma desarrollada está completamente basada en tecnologías y datos libres de los que cualquier usuario puede hacer uso.

Conclusiones y Líneas de trabajo futuras

En este apartado se pueden ver, en primer lugar, las dificultades encontradas a la hora de realizar este proyecto, en segundo lugar, las conclusiones obtenidas y, por último, las líneas de trabajo futuras.

7.1. Conclusiones

Se puede afirmar que el presente Trabajo de Fin de Máster ha supuesto un pequeño reto personal en diversos aspectos. Quizá el desconocimiento inicial de cualquier tipo de tecnología relacionada con el análisis de rutas generadas a partir de posiciones GPS ha podido influir en un desarrollo más lento de lo esperado inicialmente. Además, algunos de los desarrollos existentes (como las extensiones para weka) no han sido de gran ayuda debido a que su funcionamiento no ha sido el esperado o que ni tan siquiera funcionaban correctamente.

Además, la escasez de tiempo material por diversos motivos ha sido un gran factor en contra a la hora de desarrollar este trabajo.

Ya en el aspecto centrado en el desarrollo se puede afirmar que la implementación de esta aplicación ha merecido y merece el esfuerzo invertido en el mismo. Como se ha afirmado anteriormente, existen desarrollos anteriores que tratan este problema pero que no llegan a funcionar de la mejor forma posible.

Si se profundiza en el uso de estas aplicaciones, en el mejor de los casos, cuentan con un diseño de interfaz muy pobre y, que, además, obligan al usuario a tener conocimientos previos sobre la aplicación ya que no resultan todo lo intuitivas que deberían ser.

En este último aspecto se ha puesto una especial atención al basar el desarrollo *front-end* en los principios documentados por Hassan-Montero en su *paper*: Diseño Web Centrado en el Usuario.

Por último y no menos importante, cabe señalar que el esfuerzo invertido en la plataforma web desarrollada permite una sencilla ampliación de la misma añadiendo nueva funcionalidad como los aspectos comentados en las líneas de trabajo futuras mostradas en el siguiente apartado.

7.2. Lineas de trabajo futuras

Como se ha comentado en apartados anteriores de esta documentación, no ha sido posible el completo desarrollo planteado durante el comienzo del presente Trabajo de Fin de Máster. Por tanto, las líneas de futuro parecen claras en cuanto a la posibilidad de continuar con el planteamiento inicial. Las ideas principales de dichas líneas se pueden resumir en:

- **Generación de árboles de predicción:** los árboles de predicción son el aspecto más importante de este proyecto que no ha podido ser llevado a cabo. A partir de la semántica asignada a las rutas y del análisis de un número elevado de las mismas, se haría posible la predicción de un siguiente Punto De Interés dentro de una hipotética ruta todavía no realizada. El desarrollo de este árbol posibilitaría la apertura de nuevos caminos teniendo en mente un futuro uso de este Trabajo como base para un siguiente. Dichos árboles podrían permitir a un hipotético usuario obtener una recomendación de una ruta turística sobre una ciudad no visitada con anterioridad teniendo en cuenta los tipos de lugares visitados en rutas anteriores.

- **Exportación o importación de datos:** la implementación de un sistema de exportación permitiría obtener un conjunto de datos a partir de la Base de Datos del sistema. Este aspecto sería realmente útil para la migración de los datos sin tener que implicar conocimientos técnicos sobre el manejo de una Base de Datos como es PostgreSQL. De esta forma, cualquier usuario sería capaz de exportar los datos de las rutas analizadas. Otro aspecto importante es el desarrollo de un sistema de importación de datos para restaurar los datos anteriormente comentados.

- **Sistema de usuarios:** aunque inicialmente no ha sido uno de los aspectos planteados, el desarrollo de un sistema de usuarios podría permitir el análisis de rutas de forma privada y/o pública. Es decir, cada usuario podría decidir si comparte el análisis de una o varias de sus rutas con el resto de usuarios. Adicionalmente daría soporte a lo comentado en el

primer punto de esta lista. Es decir, la predicción de rutas para cada uno de los usuarios del sistema teniendo en cuenta sus gustos y/o prioridades durante la visita de una nueva localidad.

Bibliografía

[1] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[2] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].