UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Tour Planner BackEnd

Presentado por Ignacio Aparicio Blanco
en Universidad de Burgos — 8 de septiembre de 2019
Tutor: Bruno Baruque Zanon

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Ignacio Aparicio Blanco, con DNI 71302717A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

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, 8 de septiembre de 2019

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

Este proyecto se basa en la ampliación del proyecto de generación de rutas personalizadas en dispositivos Android “Ampliación de la Aplicación para la Generación de Rutas Turísticas Personalizadas", realizado por Alejandro Cuevas Álvarez, siendo este a su vez una ampliación del proyecto “Generación de Rutas Turísticas Personalizadas", realizado por Íñigo Vázquez Gómez y Roberto Villuela Uzquiza.

Aunque puede parecer que el tener una ruta personalizada optimiza totalmente una visita, no estamos teniendo en cuenta un factor imprescindible: el tiempo. Como dijo Alejandro Magno antes de morir: “El tiempo es el recurso más valioso que tenemos, porque es limitado". Siguiendo esta filosofía se ha decidido mejorar las rutas añadiendo una nueva funcionalidad al proyecto, que genera la ruta teniendo en cuenta los horarios de apertura y cierre de los diferentes puntos de interés, reduciendo enormemente el tiempo perdido en esperar a la apertura de los destinos y eliminando totalmente el problema de encontrarnos con un establecimiento cerrado.

Descriptores

Rutas, Backend, Servidor, GPS, GIS, Ingeniería Informática.

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

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

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

En el proyecto "Generación de Rutas Turísticas Personalizadas", realizado por Íñigo Vázquez Gómez y Roberto Villuela Uzquiza [5] se presentó una herramienta capaz de generar rutas en una ciudad introduciendo simplemente un tiempo, un origen, un destino y unas preferencias en cuanto a intereses. Esta aplicación fue ampliada un año después por el alumno Alejandro Cuevas Álvarez, bajo el título "Ampliación de la Aplicación para la Generación de Rutas Turísticas Personalizadas"[6].

En este documento se van a presentar las ampliaciones realizadas a ese último proyecto, más concretamente al *BackEnd*. Se ha querido mantener la funcionalidad del proyecto original en su totalidad, aportando a este nuevas formas de generar rutas turísticas teniendo en cuenta detalles como los horarios de los establecimientos o el tiempo que el visitante va a pasar en cada punto. De esta forma logramos obtener resultados que se adapten mejor a cada situación, mejorando enormemente la experiencia del usuario.

Para la realización de este proyecto ha sido necesario involucrarse en diferentes áreas dentro de la Informática, como pueden ser algoritmia, bases de datos y sistemas distribuidos.

Se han mantenido las decisiones tomadas en el proyecto original, por lo que se ha utilizado el lenguaje de programación orientado a objetos *Java*. Se dispone de una base de datos *PostgreSQL* [15], en la que se encuentra almacenada información como puntos de interés, sus características, relaciones entre puntos, usuarios, rutas...

Objetivos del proyecto

Los objetivos que se muestran a continuación están basados en un pilar central, que es la mejora de la aplicación ya existente, añadiendo nuevas funcionalidades que la hagan más atractiva para el usuario.

Se van a llevar a cabo las siguientes mejoras:

- Actualizar la base de datos con información referente a horarios de apertura y cierre de los distintos puntos de interés.
- Actualizar las etiquetas de los diferentes puntos de interés para poder buscar por filtros personalizados.
- Implementar un algoritmo que tenga en cuenta los horarios de apertura y cierre de los distintos puntos de interés, para añadir realismo a la aplicación y generar rutas más eficientes, ya que el modelo anterior, al no tener en cuenta estos datos, podía añadir a la ruta un punto de interés "no visitable".
- Mantenimiento del código correspondiente con la subida al servidor debido a cambios importantes que han recibido GlassFish y Maven desde la realización del anterior proyecto.

Los objetivos personales son:

- Aprender a utilizar las herramientas Maven y GlassFish para tener una base sólida en el campo de creación y uso de servidores.
- Aprender a utilizar bases de datos espaciales (GIS) con PostGIS, aprovechando para reforzar los conocimientos ya adquiridos sobre PostgreSQL.

- Aprender a utilizar el sistema de composición de textos LaTeX, así como la herramienta de desarrollo Texmaker.
- Aprender a trabajar en un equipo de desarrollo y mejorar en el mismo para tener una experiencia más parecida a la que podemos encontrarnos a la hora de realizar un proyecto real.

Conceptos teóricos

En este capítulo se describirán, como ya anticipa el título, diversos aspectos relacionados con cuestiones teóricas, que permitirán al lector una correcta comprensión del trabajo.

3.1. The Orienteering Problem

El "problema de la orientación" u "Orienteering Problem" (OP)[1] propone una situación con diferentes puntos que se pueden visitar, cada uno con una puntuación o score asociado. El objetivo es, partiendo de un punto determinado, llegar al destino dentro de un límite de tiempo maximizando la puntuación total obtenida de visitar los diferentes puntos propuestos. Cada punto disponible podrá visitarse como máximo una sola vez.

Aunque se pueden ver ciertas similitudes con el "Problema del Viajante" ("Travelling Salesman Problem")[17] debemos considerarlo como un problema totalmente distinto, debido a que NO es necesario visitar todos los nodos disponibles, debido al límite de tiempo. Podemos ver al "Problema de la Orientación" como una combinación entre el "Problema del Viajante" y el "Problema de la Mochila"[16].

De este problema original han surgido diferentes variantes a medida que se han ido proponiendo nuevos escenarios, como pueden ser *trabajando en equipo (TOP)*, teniendo en cuenta *ventanas de tiempo (OPTW)*, *dependiente del tiempo (TDOP)* y todas las combinaciones posibles de los anteriores, por lo que podemos encontrarnos con el caso de "Problema de la orientación por equipos dependiente del tiempo y con ventanas de tiempo"(TDTOPTW).

The Orienteering Problem with Time Windows

El "problema de la orientación con ventanas de tiempo" (OPTW)[2] considera unos márgenes de tiempo para cada punto disponible para visitar. La visita a un punto solo puede comenzar dentro de sus márgenes de tiempo [Oi,Ci].

En caso de llegar antes del horario de apertura (Oi) será necesario esperar. Si por el contrario llegamos después del horario de cierre(Ci) la visita a ese punto será inviable y por tanto no podrá comenzar (no se obtendrá recompensa).

3.2. Sistema de información geográfica

Un sistema de información geográfica (GIS) [19] es un conjunto de herramientas que integra y relaciona diversos componentes que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

Funcionalidad de los GIS:

1. Almacenar información, que se puede obtener por diferentes métodos, como pueden ser: GPS, fotografía aérea, imágenes satélite, bases de datos ...
2. Visualizar datos almacenados.
3. Hacer consulta sobre los datos seleccionados. De esta forma podemos presentar la información de una forma mucho más amigable para el usuario, mediante mapas, gráficos, tablas ...
4. Hacer análisis para generar nuevas capas de información útil.

Los tipos de datos geográficos más utilizados actualmente por las GIS son:

- Datos Vectoriales, utilizados para representar fenómenos discretos, describiendo objetos geográficos a partir de vectores definidos por pares de coordenadas.

- Datos Raster, utilizdos para representar fenómenos no discretos. Divide el área a representar en una retícula de celdas y atribuye a cada una de estas un valor numérico para representar su valor temático. Cabe destacar que el origen siempre esta en la esquina superior izquierda de la retícula.

3.3. Scrum

Scrum es un marco de trabajo para desarrollo ágil de software.[18] Es una metodología muy popular en la actualidad y aunque su enfoque inicial es para proyectos de software, puede ser fácilmente adaptable a otros contextos.

Si no se siguen bien sus reglas puede ser algo complicado de implementar, pero en general es una metodología muy fácil de comprender.

Scrum Team

Scrum se compone de tres roles principales:

- **Product Owner:** Es la persona que representa al cliente y tiene la misión de conocer todas las necesidades de este. Deberá transmitir esta información tanto al *Scrum Master* como al *Development Team*.
- **Scrum Master:** Es el moderador del grupo de trabajo. Aunque esta posición puede guardar similitudes con la de un líder, el *Scrum Master* no es el encargado ni de dar órdenes ni de decidir el modo de hacer las cosas. Su función es asegurarse de que el *Development Team* entienda y trabaje en las necesidades del cliente
- **Development Team:** Son las personas capacitadas para crear la solución que el cliente necesita.

Ciclo de Vida Scrum

Inicialmente el *Product Owner* definirá un artefacto llamado *Product Backlog*, donde plasmará todas las solicitudes del cliente y será presentado al *Scrum Team* en una reunión llamada *Sprint Planning Meeting*. Como resultado de esta primera reunión se obtendrá una lista de funcionalidades tomadas del *Product Backlog* a las que se asigna el nombre de *Sprint Backlog*.

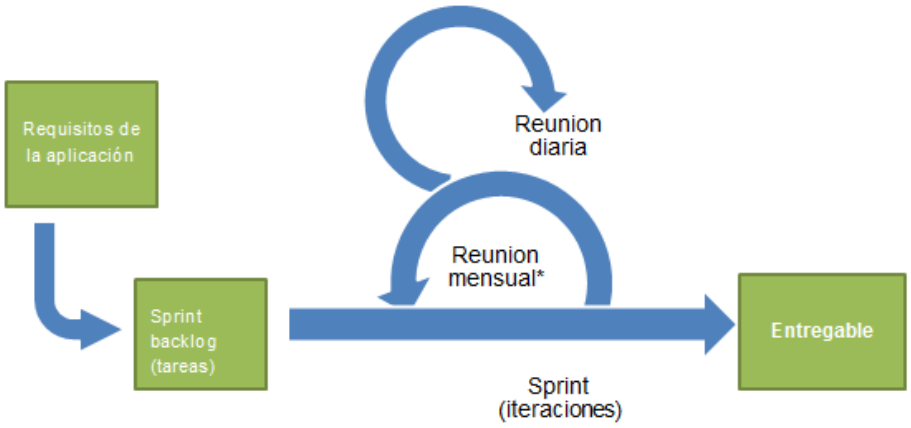

Figura 3.1: Ciclo de Vida Scrum[18]

Las funcionalidades definidas en el *Sprint Backlog* deberán completarse en un periodo de 1 a 4 semanas denominado *Sprint*.

Sprint

Es el corazón del *Scrum*, ya que responde al proceso de construcción de las necesidades del cliente, pero divididas en un módulo funcional. El tiempo de desarrollo debe ser entre 1 y 4 semanas, dependiendo de como de complejas sean las funcionalidades acordadas en el *Sprint Backlog*.

Las partes que participan en el *Sprint* son el *Scrum Master* y el *Development Team*, siendo este último el encargado de construir la necesidad que construye el *Sprint*. El primero tendrá la misión de ayudar al equipo de desarrollo.

Las reuniones

Las reuniones son uno de los elementos característicos del *Scrum*. Estas reuniones tienen lugar en cada uno de los *Sprints* y son las siguientes:

Sprint Planning Meeting

Como se ha mencionado previamente es el primer paso al comenzar un *Sprint*.Se definirá un *Sprint Backlog* con las funcionalidades a completar en el *Sprint* actual.

DailyScrum

Es una de las actividades representativas del *Scrum* y tiene como finalidad hacer un seguimiento diario del proceso de desarrollo.

Consiste en una reunión entre el *Scrum Master* y el *Development Team*, en la que se harán una serie de preguntas muy puntuales a cada pregunta dentro del equipo de desarrollo. Estas preguntas son: qué se hizo ayer, qué se va a hacer hoy, qué se va a hacer mañana y que problemas se han encontrado.

Debido a la naturaleza informativa de esta reunión, su duración deberá ser entre 5 y 15 minutos.

Sprint Review

Se trata de una de las dos reuniones que tienen lugar al finalizar un *Sprint*. En esta estarán involucrados tanto el *Scrum Master* como el *Product Owner* y *Development Team* para verificar el cumplimiento de los objetivos de ese *Sprint* y así garantizar los tiempos de entrega del producto.

Retrospectiva del Sprint

La segunda de las reuniones que tiene lugar después de cada *Sprint*. Se analizarán los resultados del *Sprint* recién acabado, para poder solventar problemáticas y mejorar el proceso para *Sprints* posteriores. Nada más acabar un Sprint se iniciará uno nuevo, volviendo a repetir el ciclo.

Técnicas y herramientas

En este capítulo se analizarán brevemente las herramientas y técnicas utilizadas en la realización del proyecto.

4.1. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto.

Como muchos otros proyectos de *código abierto*, el desarrollo de *PostgreSQL* no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales.[15]

Se ha utilizado junto a diversas extensiones para adaptarlo a las necesidades del proyecto.

4.2. PostGIS

PostGIS es una extensión de bases de datos espaciales para la base de datos relacional PostgreSQL. Agrega soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL. [13]

Una de las características más llamativas de esta extensión es que permite almacenar en una misma tabla diferentes tipos de geometría.

Si se usa junto a PgAdmin (versiones superiores a PgAdmin4 3.3) es posible utilizar el visor de geometrías integrado para obtener un visual de nuestras consultas, contando con que estas tengan una columna de geometría.

Un ejemplo de su uso en este proyecto es la obtención de diferentes puntos de interés con sus respectivos horarios a partir de unas coordenadas.

4.3. Osmosis

Osmosis es una aplicación *Java* de línea de comandos para procesamiento de datos OSM (Open Street Maps).

La herramienta consiste en componentes que se pueden encadenarse para realizar operaciones más grandes.

Por ejemplo, tiene componentes para leer/escribir bases de datos, archivos, ordenar datos...

Algunos ejemplos de uso son:

- Generar ficheros osm de una base de datos.
- Cargar ficheros osm en una base de datos.
- Realizar/aplicar sets de cambios a una base de datos local.
- Comparar dos ficheros osm y producir un set de cambios.

4.4. osm2po

Se trata de una aplicación *Java* de línea de comandos que funciona tanto como conversor como *routing engine*[11].

Es capaz de generar ficheros *sql* para *PostGIS* y puede trabajar con grandes cantidades de datos.

En las nuevas versiones se ha añadido un pequeño simulador web con el que probar las rutas generadas por la herramienta, de forma que podamos saber fácilmente si el fichero *sql* generado tiene rutas válidas o no.

4.5. osm2pgsql

osm2pgsql[10] es una aplicación *Java* de línea de comandos que convierte datos *OpenStreetMaps* a bases de datos *PostGIS*.

Aunque se encuentra disponible para *Linux, Mac OS X y Windows* es recomendable su uso en distribuciones Ubuntu, ya que cuenta con mayor documentación y mantenimiento.

4.6. Maven

Apache Maven[14] es una herramienta de gestión y compresión de proyectos software.

Basándose en el concepto de un *modelo de objetos de proyecto (POM)*, Maven puede gestionar la compilación, los informes y la documentación de un proyecto a partir de una información central.

Aunque actualmente la mayor parte de IDEs tienen la posibilidad de integrar *Maven*, también existe la posibilidad de instalar la herramienta y ejecutarla por línea de comandos, lo cual puede resultar útil en proyectos avanzados.

El proceso de construcción de software sigue unas etapas muy diferenciadas, que facilitan enormemente la labor del programador.

Cabe destacar que todas las operaciones realizadas son llevadas a cabo por diferentes *plugins*, y *Maven* permite descargar de un repositorio central otros que se adapten a las necesidades del proyecto.

4.7. GlassFish

GlassFish[23] es un servidor de aplicaciones de software libre desarrollado por *Sun Microsystems*, compañía adquirida por *Oracle Corporation*, que implementa las tecnologías definidas en la plataforma *Java EE* y permite ejecutar aplicaciones que siguen esta especificación.

GlassFish está basado en el código fuente donado por *Sun* y *Oracle Corporation*; este último proporcionó el módulo de persistencia *TopLink*. *GlassFish* tiene como base al servidor *Sun Java System Application Server* de *Oracle Corporation*, un deriado de *Apache Tomcat*, y que usa un componente adicional llamado *Grizzly* que usa *Java NIO* para escalabilidad y velocidad.

4.8. Eclipse

Eclipse[22] es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama “Aplicaciones de Cliente Enriquecido”, opuesto a las aplicaciones “Cliente-liviano” basadas en navegadores.

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés *IDE*), como el *IDE* de *Java* llamado

J*ava Development Toolkit (JDT)* y el compilador *(ECJ)* que se entrega como parte de *Eclipse* (y que son usados también para desarrollar el mismo *Eclipse*).

Eclipse fue liberado originalmente bajo la *Common Public License*, pero después fue re-licenciado bajo la *Eclipse Public License*. La *Free Software Foundation* ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de *GNU (GNU GPL)*.

4.9. Git

Git[21] es un software de control de versiones diseñado por *Linus Torvalds*, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Al principio, *Git* se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o *front end* como *Cogito* o *StGIT*.Sin embargo, *Git* se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena.

En cuanto a derechos de autor, *Git* es un software libre distribuible bajo los términos de la versión 2 de la *Licencia Pública General de GNU*.

4.10. Texmaker

Texmaker[24] es un editor gratuito distribuido bajo la licencia *GPL* para escribir documentos de texto, multiplataforma, que integra muchas herramientas necesarias para desarrollar documentos con *LaTeX*, en una sola aplicación. *Texmaker* incluye soporte Unicode, corrección ortográfica, auto-completado, plegado de código y un visor incorporado en pdf con soporte de *synctex* y el modo de visualización continua.

Para que *Texmaker* pueda funcionar es necesario haber instalado *TeX* previamente: *TeX Live*, *MiKTeX* o *proTeXt*.

4.11. AstahUML

Astah[20], anteriormente conocida como *JUDE*, es una herramienta de modelado UML creada por la compañía japonesa *Change Vision*. *JUDE*

recibió el premio "Producto de software del año 2006", establecido por la *Agencia de Promoción de Tecnologías de la Información* en Japón.

Cabe destacar que *Astah Community* está interrumpido desde el 26-Septiembre-2018. Algunas alternativas son: *Astah UML, Astah Professional o Astah Viewer*.

4.12. Pencil Project

Pencil Project[12] es una herramienta de creación de prototipos GUI. Es de código abierto y está disponible para todas las plataformas.

4.13. Hamachi

LogMeIn Hamachi[4] es un servicio de host VPN que permite crear conexiones LAN de forma segura. Se ha utilizado para realizar la conexión entre las máqinas virtuales cliente y servidor para poder simular de forma más realista el producto final.

4.14. Haguchi

Haguichi[3] es una herramienta de código abierto desarrollada en los lenguajes *Vala* y *GTK+*, que nos ofrece una interfaz gráfica para Hamachi en *Linux*.

Aunque es cierto que Hamachi permite una total configuración en sistemas operativos Linux a partir de comandos en el terminal, esta herramienta facilita mucho el trabajo, simplificando la gestión de nuestras redes.

Aspectos relevantes del desarrollo del proyecto

5.1. Algoritmos para el cálculo de rutas

Una de las decisiones más importantes del proyecto es la de elegir que algoritmo implementar, ya que es el pilar central de la ampliación que se quiere realizar.

El problema se trata de una variante del algoritmo base, que es el *problema de la orientación* (OP), detallado en la sección *Conceptos teóricos*, concretamente la variante *problema de la orientación con ventanas de tiempo* ($OPTW$).

Las bases del algoritmo son sencillas: se quiere visitar una ciudad y se han de seleccionar que puntos de interés (ordenados) crean la ruta más óptima. No suele darse en caso en que se seleccionen todos los puntos disponibles, pero esto es por las restricciones de tiempo.

La diferencia principal con el algoritmo base es que el tiempo de viaje entre puntos ya no es el único tiempo a tener en cuenta. A cada punto de interés se le asigna una ventana de tiempo en la que será obligatorio llegar para que el punto sea visitable. Se puede pensar que la solución a este problema es tan sencilla como añadir una condición al proceso de selección, pero la verdadera complicación es cómo reducir lo máximo posible los tiempos de cómputo.

La decisión de que algoritmo de los disponibles implementar se basó en su extensibilidad, es decir, debido a que OPTW no es la única variación del algoritmo base, se eligió una solución que permitiese, con no demasiados

cambios, resolver alguna de dichas variaciones en futuros proyectos. Dicho algoritmo el el basado en *colonia de hormigas*.

Una de las desventajas que se encontraron con ese algoritmo fue la falta de tablas en las que se mostrasen comparaciones de los tiempos de cómputo, aunque ya se adelantaba que eran altos. Es por eso que se decidió implementar un segundo algoritmo, a poder ser basado en una resolución lo más diferente posible, para así poder comparar como de eficiente era cada uno. Este segundo algoritmo está basado en *búsquedas iterativas*.

Los dos algoritmos solución elegidos fueron:

- Iterated local search for OPTW
- Ant Colony System

Las implementaciones de las dos soluciones se encuentran explicadas detalladamente en el *Anexo IV: 2.Algoritmos para el cálculo de rutas*.

5.2. Obtención y tratamiento de datos

Obtención de horarios

Se evaluaron dos posibles fuentes de datos:

- Google Maps
- OpenStreetMaps

Lo primero que se investigó fue el formato de los datos en cada una de las opciones, centrándonos sobre todo en los horarios de los establecimientos ya que eran los nuevos datos a implementar.

Por un lado *Google Maps* es uno de los servicios más utilizados cuando se requieren datos de geolocalización. Cuenta con una API[7] relativamente sencilla de usar, ofrece unos datos en un único formato y que (normalmente) han sido verificados, lo cual la hace una elección muy llamativa.

Cada punto cuenta con 3 categorías y cada una contiene diferentes campos. Dichas categorías y datos son:

Basic

- address_component

- adr_address
- formatted_address
- geometry
- icon
- name
- permanently_closed
- photo
- place_id
- plus_code
- type
- url
- utc_offset
- vicinity

Contact

- formatted_phone_number
- international_phone_number
- opening_hours
- website

Atmosphere

- price_level
- rating
- review
- user_rattings_total

Este servicio no solo ofrecía los datos necesarios, sino que aportaba otros extra que podían ser utilizados para hacer la aplicación más realista, permitiendo por ejemplo descartar ciertos puntos de interés en función del precio.

Al ir a implementar la API se encontró un obstáculo insalvable, las licencias de Google Maps. Si bien es cierto que la restricción del número de elementos por consulta (100 elementos) ha existido desde el nacimiento de esta API, a partir del 16 de Julio de 2018 es necesario aportar los datos de facturación en la cuenta con la que se utiliza dicho servicio, ya que en esta fecha se estableció un coste por el uso de los datos.

Los algoritmos utilizados necesitan una gran cantidad de datos, ya que para disminuir el tiempo de cómputo se utiliza una *matriz de distancias*, en

la que se cargan los datos de localización de cada punto para facilitar la obtención del espacio entre estos.

Aunque era posible solventar la restricción de peticiones disminuyendo considerablemente los datos disponibles para los algoritmos, no había opción con el pago por el servicio, por lo que se tuvo que descartar la opción de utilizar Google Maps.

La segunda opción disponible era utilizar el proyecto colaborativo *OpenStreetMaps*[8], que contaba con la ventaja de tener los datos abiertos al publico, ya que es la comunidad quien se encarga de aportar dichos datos. Para ello el único requisito es estar registrado.

Este último punto plantea otro inconveniente, dado que cualquiera puede añadir datos nuevos (o corregir los antiguos), era posible encontrarnos con ciertas incongruencias en los datos, sobre todo en los referentes a horarios de apertura y cierre de los establecimientos.

Otros inconvenientes de utilizar OpenStreetMaps eran:

- Falta de una API
- Documentación oficial escasa del uso de sus herramientas
- Necesidad de tener los datos previamente volcados en nuestro servidor para poder utilizarlos
- Necesidad de una base de datos GIS

Dado que no había otra alternativa, la elección estaba tomada sin importar los inconvenientes.

Cabe destacar el gran trabajo que hicieron los anteriores contribuyentes de este proyecto: Íñigo Vázquez Gómez, Roberto Villuela Uzquiza y Alejandro Cuevas Álvarez. Su trabajo facilitó mucho la comprensión y el uso de esta herramienta

Volcado de datos, Osmosis vs Osm2pgsql

Se investigó que herramientas existían para el volcado de los datos presentes en los ficheros descargados de *OpenStreetMaps* a nuestra base de datos geoespacial. Se valoraron dos alternativas:

- Osmosis
- Osm2pgsql

Una de las ventajas de Osmosis es que los datos se vuelcan de una forma muy particular, creando relaciones de las que podíamos beneficiarnos como ya habían hecho en años anteriores el resto de contribuyentes de esta aplicación.

Una vez volcados los datos se reparó en que no solo no se había importado la etiqueta opening hours, sino que no se había importado ninguna etiqueta que no fuese de las principales, entendiendo por principales:

- ID
- Nombre
- Localización
- Tipo de nodo/vía
- Relaciones entre nodos

En vista del resultado se pasó a utilizar la herramienta *Osm2pgsql* ya que consta de un archivo de estilo, llamado *default style*, en el que se pueden indicar que etiquetas de todas las existentes en OpenStreetMaps queremos importar. El volcado se realizó con éxito, pero el formato de las tablas creadas en *PostgreSQL* no eran adecuadas para el uso que se les quería dar.

Finalmente, se decidió que, debido a que cada herramienta aportaba utilidades imprescindibles, se utilizarían ambas herramientas, beneficiándonos así de sus puntos fuertes y solventando las debilidades. La solución final se encuentra explicada detalladamente en el *Anexo RELLENAR ANEXO*.

Cabe destacar que en caso de querer volcar una gran cantidad de datos, como por ejemplo, de toda España, existe la posibilidad de modificar la cantidad de memoria de nuestro PC que permitimos utilizar a Osm2pgsql. Si dicha cantidad no es suficiente, a mitad del volcado se quedará sin espacio y se nos notificará con un error que no se han podido volcar los datos. Por el contrario, si ofrecemos demasiada memoria nuestro ordenador dejará de responder y tendremos que forzar un apagado del sistema manteniendo pulsado el botón de encendido.

Se recomienda que, en caso de necesitar utilizar ficheros tan pesados, es mejor obtener cortes de dicho fichero, por ejemplo, por comunidades autónomas o por provincias y realizar el volcado de uno en uno, para así evitar errores como los mencionados.

Tratamiento de horarios

Tras el volcado de datos se pasó a comprobar la calidad de la información.

La etiqueta *opening_hours* cuenta con una sintaxis bastante compleja, que bien utilizada es capaz de aportar una gran información. Un ejemplo de dicha sintaxis es el siguiente:

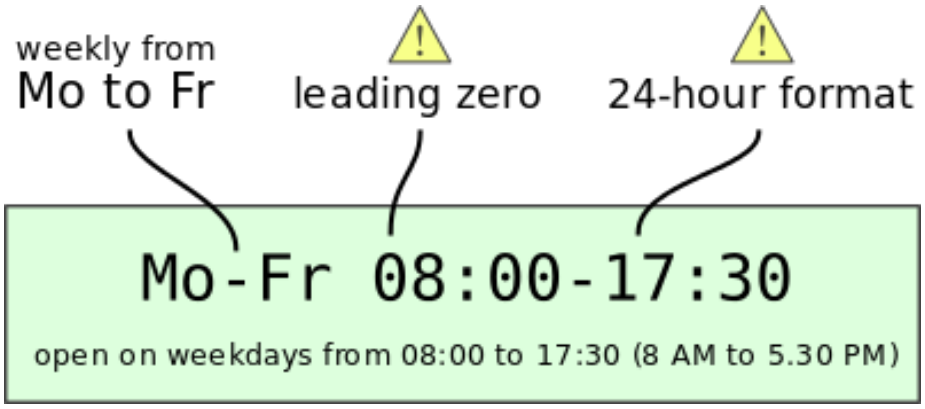

Figura 5.2: Ejemplo de sintaxis de la etiqueta opening hours [9]

Como se ha comentado con anterioridad, OpenStreetMaps permite a sus usuarios subir sus propios datos, lo que causa que algunos de ellos contengan una sintaxis errónea.

Después de comentar con el tutor del proyecto las posibles soluciones, se llegó a dos ideas:

- Gestión de los datos desde la base de datos
- Gestión de los datos desde los algoritmos

Las comprobaciones de los datos mostraban que una gran cantidad de puntos de interés no contaban con un horario, pero no todos los usuarios habían optado por no introducir ningún valor, ciertos establecimientos tenían en su horario "", lo cual no era detectado ni como nulo ni como espacio. Esto dio varios dolores de cabeza a la hora de probar los algoritmos hasta que se descubrió que esta era la causa.

Se decidió por tanto practicar las dos soluciones a la par, nada más introducir los datos se usaría un script SQL que buscaría aquellos que contuviesen tanto nulo como "z sustituiría esos valores por unos por defecto.

Por otro lado, se estudiaron todos los posibles tipos de sintaxis y se creó un método dentro del algoritmo que extrajese los horarios de apertura y cierre a partir de un String obtenido de la base de datos.

Estas soluciones se encuentran explicadas mas detalladamente en el *Anexo RELLENAR ANEXO*

5.3. Conocimientos adquiridos en la carrera

Conocimientos sobre inteligencia artificial

Ha sido de gran ayuda para este proyecto haber cursado durante la carrera la asignatura *Computación Neuronal y Evolutiva*, ya que tuvimos nuestro primer acercamiento a este tipo de algoritmos mediante un desafio Hash Code, así como las asignaturas *Sistemas Inteligentes*, *Algoritmia*, la asignatura *Advanced Logical and Functional Programming* cursada en la *West University of Timisoara* etc. Gracias a esto la comprensión de artículos referentes a la resolución e implementación del problema de la orientación ha sido mucho mas llevadera.

Conocimientos sobre bases de datos

De vital importancia ya que ha sido necesario crear una base de datos GIS y modificarla mediante SQL para que fuese más eficiente. Gracias a haber cursado las asignaturas de *Bases de Datos y Aplicaciones de Bases de Datos* se ha partido de una posición más avanzada y ha sido posible tanto la creación de la base de datos como el manejo del álgebra relacional.

Trabajo de investigación

No se ha impartido en ninguna asignatura en concreto, pero ha estado presente en cada uno de los trabajos que se han desarrollado en el grado.

La rapidez con la que se buscaban y comprendían nuevos conocimientos, se verificaban las fuentes de los datos y se solventaban los diversos problemas y errores que han aparecido en el desarrollo de este proyecto ha demostrado que esta capacidad es una de las más importantes que se han adquirido durante la carrera.

Trabajos relacionados

6.1. TripAdvisor (IOS/Android)

Sin duda una de las aplicaciones para planificar viajes más populares. Nos permite seleccionar una ciudad y descubrir puntos de interés de diferentes categorías, como pueden ser:

- Sitios más fotografiados
- Lugares más emblematicos
- Restaurantes según su precio
- Establecimientos especializados
- Hostales y hoteles

Permite subir fotos y vídeos a la aplicación para que el resto de usuarios puedan verlos y ofrece diferente información sobre el lugar recopilada de periódicos como el ABC, guías de viajes etc.

Una de las opciones más interesantes que presenta esta aplicación es la posibilidad de acceder a sus foros, donde los usuarios realizan preguntas sobre qué lugares recomiendan en la ciudad a la que van a viajar, que comer etc. De esta forma se logra una comunidad conectada y da la sensación de trato personal.

Cabe destacar que al instalar la aplicación nos da la opción de registrarnos utilizando una cuenta de *Gmail* o *Facebook*, pero no es obligatorio. En caso de no registrarnos:

- No se nos dejará acceder al buzón, donde recibimos notificaciones de nuestro muro de viajes y chats privados con otros usuarios.
- No se nos dejara acceder a la sección de viajes.

El uso de la aplicación es extremadamente sencillo, basta con dar un nombre a un viaje, que en realidad es un repositorio donde podremos guardar lugares para visitar más adelante.

No genera rutas automáticamente, sino que funciona más bien como planificador de viajes.

Conclusiones y Líneas de trabajo futuras

En este apartado se tratarán las diferentes conclusiones que han surgido durante la realización de este proyecto así como posibles lineas que seguir en un futuro para mejorar la funcionalidad de la aplicación.

7.1. Conclusiones

Durante la realización de este proyecto se ha aprendido a utilizar repositorios como OpenStreetMaps para obtener datos geográficos, a crear y utilizar bases de datos geoespaciales, a trabajar con proyectos Java bajo la herramienta de gestión Maven, a implementar y comprender diferentes algoritmos para resolver el problema de la orientación, a trabajar con el servidor de aplicaciones Glassfish y a tratar con los diversos problemas que genera el trabajar en la ampliación de un proyecto anterior.

Como valoración personal, con este proyecto se han conseguido reforzar diferentes conocimientos adquiridos a lo largo de toda la carrera, muchos de los cuales no se habían puesto en práctica.

7.2. Líneas de trabajo futuras

En este apartado se comentarán las posibles mejoras que se podrían aplicar a este proyecto:

- Ampliar la generación de rutas, haciendo al algoritmo dependiente del tiempo, es decir, que la velocidad de ruta varíe en función de

la hora del día (Time Dependent Orienteering Problem). Los algoritmos implementados en este proyecto están preparados para dicha ampliación.

- Crear una página web que iguale la funcionalidad de la aplicación.
- Conectar la aplicación con las redes sociales, para que permita compartir y publicar las rutas generadas.
- Conectar la aplicación con el GPS del dispositivo, para que pueda guiarnos hacia el siguiente punto de interés.
- Actualizar el cliente a una versión de Android más actual.
- Crear una versión del cliente para otros sistema móviles como pueden ser IOS o Windows Phone.
- Incluir realidad aumentada en la aplicación, para poder ver los diferentes puntos de interés sin necesidad de hacer la ruta personalmente.
- Conectar la aplicación a diferentes funciones del dispositivo para, en función de la distancia, el tiempo recorrido y los pasos dados, proveer al usuario información sobre su salud física.
- Un sistema de amigos con el que se puedan ver las últimas rutas de estos.

Bibliografía

[1] Pieter Vansteenwegen Aldy Gunawan, Hoong Chuin Lau. Orienteering problem: A survey of recent variants, solution approaches and applications. 2016.

[2] Pieter Vansteenwegen Aldy Gunawan, Hoong Chuin Lau. Orienteering problem: A survey of recent variants, solution approaches and applications. page 4, 2016.

[3] Haguichi. Haguichi, 2019. [Internet; descargado 01-septiembre-2019].

[4] LogMeIn Hamachi. Logmein hamachi, 2019. [Internet; descargado 01-septiembre-2019].

[5] Roberto Villuela Uzquiza Íñigo Vázquez Gómez. Generación de rutas turísticas personalizadas. 2013.

[6] Alejandro Cuevas Álvarez. Ampliación de la aplicación para la generación de rutas turísticas personalizadas. 2014.

[7] Google Maps. Google maps platform - documentation, 2019. [Internet; descargdo 03-septiembre-2019].

[8] Open Street Maps. Open street maps, 2019. [Internet; descargado 03-septiembre-2019].

[9] OpenStreetMaps. Openstreetmaps - opening hours, 2019. [Internet; descargado 03-Septiembre-2019].

[10] OpenStreetMaps. Osm2pgsql, 2019. [Internet; descargado 28-junio-2019].

[11] osm2po. osm2po-core, 2019. [Internet; descargado 28-junio-2019].

[12] Pencil Project. Pencil project*, 2019. [Internet; descargado 28-junio-2019].

[13] PostGIS Project. Postgis 3.0.0alpha3dev manual, 2019. [Internet; descargado 27-junio-2019].

[14] Apache Maven Proyect. Apache maven proyect, 2019. [Internet; descargado 28-junio-2019].

[15] Wikipedia. Postgresql — wikipedia ; la enciclopedia libre, 2018. [Internet; descargado 10-mayo-2019].

[16] Wikipedia. Problema de la mochila — wikipedia ; la enciclopedia libre, 2018. [Internet; descargado 26-junio-2019].

[17] Wikipedia. Problema del viajante — wikipedia ; la enciclopedia libre, 2018. [Internet; descargado 26-junio-2019].

[18] Wikipedia. Scrum (desarrollo de software), 2018. [Internet; descargado 26-junio-2019].

[19] Wikipedia. Sistema de información geográfica — wikipedia , la enciclopedia libre, 2018. [Internet; descargado 10-mayo-2019].

[20] Wikipedia. Astah*, 2019. [Internet; descargado 28-junio-2019].

[21] Wikipedia. Eclipse (software), 2019. [Internet; descargado 28-junio-2019].

[22] Wikipedia. Git, 2019. [Internet; descargado 28-junio-2019].

[23] Wikipedia. Glassfish, 2019. [Internet; descargado 28-junio-2019].

[24] Wikipedia. Texmaker, 2019. [Internet; descargado 28-junio-2019].