Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática

Tecnología Específica de

Ingeniería del Software

Desarrollo de un desmostrador 1D enfocado a empresas de carretera y ferrocarriles, líneas eléctricas y otros servicios lineales.

Henar Velasco Galán

Julio, 2023

Trabajo Fin de Grado

Grado en Ingeniería Informática

Tecnología Específica de

Ingeniería del Software

Autor: Henar Velasco

Tutor: José Antonio Mateo Cortes

Co-Tutor: Enrique Arias Antúnez

Julio, 2023

“Words stand only for themselves”.
Karen Spärck Jones

Declaración de Autoría

Yo, Henar Velasco Galán con DNI 48151160Q, declaro que soy el único autor del trabajo fin de grado titulado en Grado en Ingeniería Informátia y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 11 de Julio

Fdo: Henar Velasco Galán

Resumen

Los servicios lineales, que engloban carreteras, ferrocarriles, líneas eléctricas y otros sistemas de transporte, juegan un rol fundamental en la movilidad y el comercio a nivel global, al conectar regiones y permitir un flujo constante de mercancías y personas. Entre estos sistemas, el transporte lineal, como carreteras y ferrocarriles, facilita el transporte terrestre de mercancías y el desplazamiento diario de trabajadores, mientras que las líneas eléctricas son cruciales para la distribución de energía.

El reto reside en encontrar la ruta óptima y segura para cada uno de estos servicios lineales, considerando múltiples factores que influyen en su planificación. En este Trabajo de Fin de Grado (TFG), se busca desarrollar una aplicación web que demuestre, en una dimensión, la utilización de la información proporcionada por los Sistemas de Información Geográfica (SIG) y las condiciones climatológicas para el cálculo de ruta.

Este documento presenta el estudio del problema, la metodología adoptada para el desarrollo del sistema y ofrece un análisis y descripción de su funcionamiento. Asimismo, se dedica un capítulo especial al estudio de un caso específico, enfocado en el cálculo de la ruta Albacete-Murcia dentro de la red ferroviaria, proporcionando un análisis detallado. Por último, se incluyen las conclusiones obtenidas a partir de este proyecto, así como posibles líneas de trabajo futuro.

Agradecimientos

El presente Trabajo de Fin de Grado me gustaría dedicarselo con profunda gratitud y reconocimiento a mis padres, quienes con inmenso esfuerzo y dedicación me han podido dar la oportunidad de cursar y completar este grado. Asimismo, expresar mi más sincero agradecimiento a mi hermano, Jesús, por el constante respaldo y apoyo en los momentos de flaqueza y dificultad.

Destacar de manera especial a mis compañeras de estudios, Rosa Tárraga y Paloma Cuenca, quienes a lo largo de esta trayectoria académica se han mantenido presentes para ofrecerme su apoyo incondicional y ayuda en los momentos en que ha sido necesario.

Agradecida por la oportunidad y confianza depositadas en mí por José Antonio Mateo y Enrique Arias, quienes han sido una fuente inagotable de estímulo y aliento durante todo el desarrollo de este proyecto académico.

Por último, se desea expresar un sincero agradecimiento a Hermenegilda Macia y Enrique Brazález, ya que me ayudaron a fortalecer la autoestima en el ámbito de la informática y fomentar una renovada ilusión.

Índice general

Capítulo 1 Introducción 17

1.1 Motivación 18

1.2 Objetivos 20

1.3 Estructura de la memoria 21

Capítulo 2 Estado del Arte 24

2.1 Fundamentos 24

Sistemas de Información Geográfica 24

Algoritmos voraces 33

2.2 Estado del Arte 34

Herramientas similares 34

Conclusiones 38

Capítulo 3 Metodología y herramientas 41

3.1 Metodología adoptada 42

Responsabilidades 43

Dinámica 44

3.2 Plan de trabajo 45

3.3 Herramientas 46

Frameworks 49

Librerías 49

Capítulo 4 Análisis y Diseño 51

4.1 Análisis 51

4.2 Diseño 56

Modo Básico 56
Modo Intermedio 57
Modo Avanzado 59

Capítulo 5 Caso de estudio 61

5.1 Caso de estudio Albacete-Murcia 61

Control de errores 64

Capítulo 6 Conclusiones y trabajo futuro 67

6.1 Conclusiones 67

6.2 Competencias 68

6.3 Trabajo futuro 70

Bibliografía 71

Anexo I. Proceso de desarrollo 73

I.1 Sprint 1 73

I.2 Sprint 2 77

I.3 Sprint 3 80

I.4 Sprint 4 83

I.5 Sprint 5 86

I.6 Sprint 6 89

I.7 Sprint 7 92

I.8 Sprint 8 95

Índice de ilustraciones

Ilustración 1. Cambio de orden de las capa "*vías_network01_25830*" y "*raster01*". 26
Ilustración 2. Ejemplo de Datos SIG, datos geográficos y datos no geográficos. 27
Ilustración 3. Datos vectoriales 28
Ilustración 4. Valores en la celda 29
Ilustración 5. Ráster 31
Ilustración 6. Múltiples bandas en un ráster 31
Ilustración 7. Framework Scrum 42
Ilustración 8. Diagrama de Gantt orientativo 45
Ilustración 9. Distribución de sprints. 46
Ilustración 10. Arquitectura del sistema. 47
Ilustración 11. Estructura del proyecto. 53
Ilustración 12. Modelo Entidad Relación de la base de datos. 53
Ilustración 13. Diagrama de secuencia sobre la interacción de un usuario con el sistema para calcular una ruta. 55
Ilustración 14. Vista general 1Dpath 56
Ilustración 15. Vista general de modo Básico 1Dpath 57
Ilustración 16. Vista general del modo intermedio 1Dpath 58
Ilustración 17. Botón y sus opciones en el mapa en la modo intermedio. 58
Ilustración 18. Panel de información de una estación en el modo intermedio. 59
Ilustración 19. Vista general del modo avanzado 1Dpath. 60
Ilustración 20. Selección del inicio de la ruta, Albacete. 62
Ilustración 21. Selección fin de la ruta, Murcia. 62
Ilustración 22. Configuración y Resumen. 63
Ilustración 23. Resultado de la ruta Albacete-Murcia. 64
Ilustración 24. Aviso de campos requeridos. 65
Ilustración 25. Tramos implicados en el caso de estudio 66
Ilustración 26. *Sprint Backlog* del Sprint 1. 74
Ilustración 27. *Product Backlog* al terminar Sprint 1. 76

Ilustración 28. Diagrama de actividad. 76
Ilustración 29. *Sprint Backlog* del Sprint 2. 78
Ilustración 30. Primer boceto de la aplicación. 79
Ilustración 31. *Sprint Backlog* del Sprint 3. 80
Ilustración 32. Entidad resultado de la importación. 81
Ilustración 33. *Sprint Backlog* del Sprint 4. 85
Ilustración 34. *Sprint Backlog* del Sprint 5. 87
Ilustración 35. Sprint Backlog del Sprint 6. 89
Ilustración 36. Lectura fichero ráster .hdr. 90
Ilustración 37. Menú. 91
Ilustración 38. Mapa de vías en obras. 91
Ilustración 39. Mapa con estaciones. 92
Ilustración 40. *Sprint Backlog* del Sprint 7. 93
Ilustración 41. *Sprint Backlog* del Sprint 8. 97
Ilustración 42. Agrupación de estaciones. 97
Ilustración 43. Stepper. 98

Índice de tablas

Tabla 1. Comparativa entre herramientas similares. 37
Tabla 2. Información sobre el Sprint 1. 74
Tabla 3. Retrospectiva del Sprint 1. 75
Tabla 4. Información sobre el Sprint 2. 77
Tabla 5. Retrospectiva del Sprint 2. 79
Tabla 6. Información sobre el Sprint 3. 80
Tabla 7. Retrospectiva del Sprint 3. 82
Tabla 8. Información sobre el Sprint 4. 83
Tabla 9. Información sobre el Sprint 5. 87
Tabla 10. Información sobre el Sprint 6. 89
Tabla 11. Retrospectiva del Sprint 6. 92
Tabla 12. Información sobre el Sprint 7. 93
Tabla 13. Retrospectiva del Sprint 7. 94
Tabla 14. Información sobre el Sprint 8. 95

Capítulo 1

Introducción

El transporte lineal, que engloba carreteras, ferrocarriles, líneas eléctricas y otros servicios lineales, desempeña un papel vital en la movilidad y el comercio a escala mundial. Estos sistemas de transporte desempeñan un papel crucial al conectar regiones, ciudades y países, permitiendo un flujo continuo de bienes y personas. Por ejemplo, las carreteras facilitan el transporte terrestre de mercancías, el desplazamiento diario de los trabajadores y el acceso a servicios esenciales, como hospitales y escuelas. De manera similar, los ferrocarriles permiten el transporte eficiente de productos a larga distancia, favoreciendo el comercio internacional y la integración económica entre países. Asimismo, las líneas eléctricas son cruciales para la distribución de energía, asegurando el suministro eléctrico a hogares, empresas e industrias.

Encontrar la ruta óptima y segura para cada uno de estos sistemas de transporte lineal representa un reto arduo debido a una amplia variedad de factores que influyen en su planificación. La eficiencia en la movilidad y el comercio depende de la capacidad para determinar las mejores rutas, minimizando los tiempos de viaje, reduciendo los costos operativos y garantizando la seguridad de los usuarios. Además, el transporte lineal desempeña un papel clave en la reducción de la congestión del tráfico, la disminución de la contaminación ambiental y la mejora de la calidad de vida de las personas.

Por lo tanto, el uso de la información proporcionada por los sistemas de información geográfica (SIG) se presenta como una solución prometedora. Estos sistemas permiten considerar singularidades del terreno, como la altitud, la pendiente y otros factores relevantes, que influyen directamente en la planificación de trayectos eficientes y seguros. Al tomar en cuenta esta información geoespacial, es posible optimizar las rutas, evitar terrenos complicados o peligrosos, y mejorar la planificación logística en general.

Asimismo, para lograr una planificación aún más precisa, contar con datos climatológicos en tiempo real se convierte en un factor determinante. La información meteorológica actualizada proporciona una visión más clara y completa de las condiciones ambientales, como lluvias, vientos fuertes o nevadas, que pueden afectar la seguridad y la eficacia de los trayectos. Al integrar estos datos climatológicos en los sistemas de planificación, se puede tomar decisiones más informadas y adaptar las rutas según las condiciones meteorológicas específicas en tiempo real.

En conclusión, la topografía, las restricciones geográficas y las condiciones ambientales son solo algunos de los elementos cruciales a considerar para garantizar itinerario eficientes y seguros en el transporte lineal. La integración de datos analizados provenientes de sistemas de información geográfica y la incorporación de datos climatológicos en tiempo resulta ser información muy valiosa para optimizar la planificación y lograr una movilidad más eficiente, fomentando así el desarrollo económico, la conectividad y la sustentabilidad en el ámbito del transporte lineal.

1.1 Motivación

Los datos geográficos detallados sobre elementos como carreteras, ríos, edificios, entre otros, se combinan con las características del terreno y otros datos geoespaciales para el diseño y planificación de trayectorias óptimas y seguras para todo tipo de transporte lineal. Por ejemplo, elementos como las pendientes pronunciadas, que dificultan el movimiento de los trenes, deben ser considerados en la selección de la ruta más adecuada. Otros ejemplos como evitar zonas con curvas pronunciadas en la planificación de trayectos, ya que estas condiciones limitan la velocidad y la capacidad e incluso aumentan el riesgo de accidentes. Asimismo, es importante considerar factores climáticos extremos, como las lluvias torrenciales, que pueden afectar la estabilidad y

seguridad de los vehículos. Todos estos factores influyen significativamente en la selección de la ruta óptima.

El desafío que se presenta consiste en la integración de una extensa variedad de información y en la combinación de datos geográficos y geoespaciales diversos, con el objetivo de proporcionar una visión panorámica que permita a los usuarios realizar consultas informadas. A través de esta información, se busca proporcionar recomendaciones y análisis detallados que faciliten la toma de decisiones estratégicas en la planificación de itinerarios.Un buen ejemplo de la importancia de este enfoque es el impacto en la rentabilidad en una flota de camiones. Si se elige cuidadosamente una ruta, es posible ahorrar cantidades significativas de dinero.

A través de este proyecto, se desarrollará una aplicación web con un demostrador 1D, utilizando datos lineales obtenidos tanto de base de datos internas como externas, dirigidos para las empresas de infraestructuras de carreteras, ferrocarriles, líneas eléctricas y otros servicios lineales.

Recopilar, integrar, utilizar eficientemente la información geoespacial y facilitar la comprensión de la diversidad de elementos y factores que afectan a la selección de rutas, como ya se ha expuesto anteriormente, conforman una de las motivaciones de este proyecto para poder proporcionar mejores soluciones a estos clientes. Otra de las motivaciones es proporcionar una solución fácil de usar e intuitiva, que simplifique el proceso de consulta y análisis de datos. Se busca ofrecer diferentes modos de uso adaptados a los distintos usuarios, una herramienta accesible, con interfaces que muestren resultados sencillos de interpretar, proporcionando así resultados claros y concisos, que destaquen la mejor opción obtenida tras el análisis exhaustivo de los datos.

A lo largo de esta memoria, se describirá el proceso de desarrollo de la herramienta, incluyendo la recopilación y procesamiento de datos geográficos, la incorporación de información y la implementación de algoritmos de cálculo de rutas. También se presentarán los desafíos encontrados durante el proyecto y las soluciones propuestas, así como los resultados obtenidos y las posibles mejoras a considerar en trabajos futuros.

1.2 Objetivos

Como resultado de la motivación que impulsa este proyecto, se desprenden los siguientes propósitos, si bien es importante resaltar especialmente el objetivo primordial: "Desarrollar una aplicación web con un demostrador 1D con el uso de la información proporcionada por los SIG".

Para determinar el cumplimiento de este objetivo, es necesario lograr los subpropósitos que se detallan a continuación:

i. Estudio de los sistemas de información geográfica existentes para obtener un conocimiento profundo de las soluciones actuales y sus aplicaciones en el ámbito de la planificación de rutas para el trasporte ferroviario, carreteras y otros servicios lineales.

ii. Llevar a cabo un proceso de elicitación de requisitos mediante la recopilación de información directa de los clientes y expertos en el campo. Este objetivo tiene como finalidad comprender las necesidades, expectativas y preferencias de los usuarios en relación a la herramienta desarrollada. A través de entrevistas, encuestas y sesiones de trabajo colaborativas, se recopilará información valiosa para identificar los requisitos clave y las funcionalidades requeridas. De esta manera, se garantizará que la herramienta cumpla con las necesidades reales del cliente y proporcione soluciones eficientes y personalizadas para la planificación de rutas óptimas y seguras.

iii. Respecto a las funcionalidades ofertadas, la aplicación presentará tres modos: simple, intermedio y avanzado.

   a) El modo simple deberá ser sencillo, con poca interacción entre el usuario y el sistema.

   b) El modo intermedio incluirá más capacidades de interacción, incluyéndose así un mapa interactivo dónde se pueda visualizar información geográfica.

   c) El modo avanzado contará con una capacidad interactiva que permita realizar consultas de rutas en un mapa y mostrar los resultados correspondientes.

iv. La aplicación ha de ser capaz de calcular rutas óptimas en redes vectoriales conexas a través del algoritmo de Dijkstra, a partir de pesos, que se calculan en función de las difrentes bases de dtoas consideradas como de las condiciones atmosféricas, en las aristas del grafo.

1.3 Estructura de la memoria

La distribución de esta memoria es la siguiente:

Capítulo 1: Introducción.

En este capítulo se presentará un contexto relevante para el proyecto, exponiendo las motivaciones que impulsaron su realización y enumerando los objetivos que se persiguen.

Capítulo 2: Estado del arte.

En esta sección se expondrán los fundamentos necesarios para comprender el contexto y el desafío planteado por el presente proyecto. Asimismo, se identificarán herramientas similares a la presentada en el TFG, estableciendo las principales similitudes y diferencias entre ellas, con el objetivo de establecer su relación con el proyecto en cuestión.

Capítulo 3: Metodología y herramientas.

En este capítulo se realizará una descripción detallada de la metodología adoptada para la ejecución del proyecto, justificando su elección en función de las características del mismo. Además, se presentará un listado exhaustivo de las herramientas y recursos utilizados en el desarrollo del proyecto.

Capítulo 4: Análisis y Diseño.

En la sección de análisis, se presentarán diagramas que representan de manera visual la estructura y funcionalidades del proyecto, así como las diferentes partes de la aplicación. Posteriormente, en el apartado de diseño, se explicará la interfaz general del sistema, detallando los distintos modos de uso y su funcionamiento.

Capítulo 5: Caso de estudio.

En este capítulo se presentará un caso de estudio específico, focalizado en la ruta Albacete – Murcia. Se describirá paso a paso la interacción del usuario para llevar a cabo la consulta de esta ruta, incluyendo las acciones que realiza y las respuestas obtenidas

del sistema. Asimismo, se explicará el proceso que lleva a cabo el sistema para procesar la información y proporcionar la respuesta correspondiente.

Capítulo 6: Conclusiones y trabajo futuro.

En este capítulo, se presentarán las conclusiones generales del proyecto, destacando las contribuciones realizadas. Además, se mencionarán las posibles líneas de trabajo futuro, es decir, aspectos que podrían investigarse o mejorarse en futuros proyectos relacionados.

Capítulo 2

Estado del Arte

En este capítulo, abordaremos los fundamentos teóricos esenciales para comprender este proyecto, así como el estado del arte en la materia. Analizaremos los elementos clave que respaldan su desarrollo y examinaremos las tecnologías relevantes en detalle.

2.1 Fundamentos

Exploraremos los conceptos fundamentales relacionados con los Sistemas de Información Geográfica (SIG) y los algoritmos de cálculo de rutas.

Sistemas de Información Geográfica

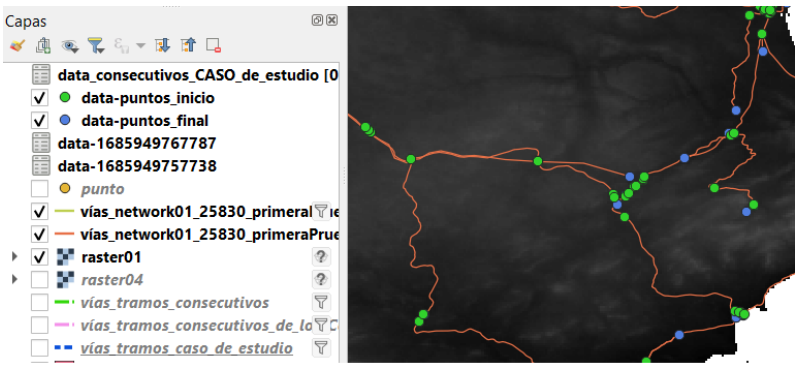
Un SIG se compone de los datos digitales, se trata de los datos geográficos que se visualizan y analizan utilizando el hardware y el software de la computadora, del hardware que son los dispositivos utilizados para almacenar datos, mostrar gráficos y procesar información y el software que son los programas que se ejecutan en el hardware del ordenador y permiten trabajar con los datos digitales. Un tipo de programa informático que forma parte del SIG se conoce como aplicación SIG. Al igual que se usa un editor de textos para redactar documentos, se puede emplear una aplicación SIG para trabajar con datos espaciales en un ordenador.

Mediante el uso de una aplicación SIG, se tiene la posibilidad de abrir mapas digitales en el ordenador, producir nueva información para agregarla al mapa y realizar análisis de la misma. La información resultante del análisis e interpretación se utilizará para la construcción de las redes de los servicios lineales y el cálculo de una ruta ideal para un contexto dado. Por lo tanto, resulta necesario comprender los datos, los diferentes tipos de datos que se pueden encontrar, así como la forma de analizarlos con la ayuda de las aplicaciones SIG, con el fin de contrastar y cerciorarse de que se están utilizando de manera adecuada. Para ello, resulta importante entender las funciones y la forma en que las aplicaciones SIG tratan los datos.

Una función habitual de las aplicaciones SIG es la visualización de capas de mapas. Estas capas se guardan como archivos en el ordenador o como registros en una base de datos. Por lo general, cada capa del mapa representa elementos o datos que se detallarán más adelante, por ejemplo, una capa de carreteras contendrá datos relacionados con el sistema vial. Cuando abrimos una capa en la Aplicación SIG, ésta se muestra en la vista de mapa. La vista de mapa presenta un gráfico que representa la capa seleccionada. Al agregar varias capas a la vista de mapa, éstas se superponen una sobre otra. Una función importante de la vista de mapa es permitir hacer zoom para ampliar o alejar, lo que permite visualizar áreas más pequeñas o más grandes respectivamente. También, se puede llevar a cabo el desplazamiento por el mapa con el objetivo de explorar diversas regiones.

Otra característica común de las aplicaciones, es la presencia de una leyenda del mapa. La leyenda del mapa proporciona una lista de las capas que se han cargado en la aplicación. A diferencia de una leyenda de mapa impresa, la leyenda del mapa o "lista de capas" se posibilita la opción de reordenar, ocultar, mostrar y agrupar las capas de manera dinámica. Para cambiar el orden de las capas, simplemente se debe hacer clic en una capa en la leyenda, mantener presionado el botón del ratón y arrastrar la capa a una nueva posición, siguiendo una mecánica parecida a la de programas como Photoshop. Un ejemplo es el que podemos observar en la Ilustración 1, antes del cambio de orden de

las capas, podíamos ver las vías, líneas rojas, en este caso, las vías del ferrocarril y después ya no los podemos ver porque están debajo de la capa raster01[1].



(a) Antes

(b) Después

Ilustración 1. Cambio de orden de las capa "*vías_network01_25830*" y "*raster01*".

A continuación, se va a desarrollar el concepto Datos SIG y su relación con la información geográfica. En este ámbito, el término "dato" se utiliza como sinónimo de "información". Es importante destacar que la información utilizada en un SIG habitualmente tiene un componente geográfico. No obstante, una particularidad estos sistemas es su capacidad para asociar información, que puede ser no geográfica, con ubicaciones específicas en el mapa. De hecho, una aplicación SIG puede almacenar una gran cantidad de datos asociados con cada ubicación, algo que los mapas en papel no logran hacer eficientemente. Por ejemplo, en la **Ilustración 2**, podemos ver que las filas contienen datos geográficos, como la "x" e "y" que representan ubicaciones en el espacio. Por otro lado, la fila "*estacionescercanas*" contiene datos no geográficos, en este ejemplo, la información que contiene es sobre las estaciones cercanas al punto

[1] QGIS project. (2002). En el apartado segundo "Introducción a SIG" de la *Guía del usuario/Manual de QGIS Desktop (QGIS 3.28).* Recuperado, el 20 de junio de 2023.

seleccionado, el punto que se está marcado en rojo a la derecha de dicha ilustración, como la distancia a la que se encuentran y su id.

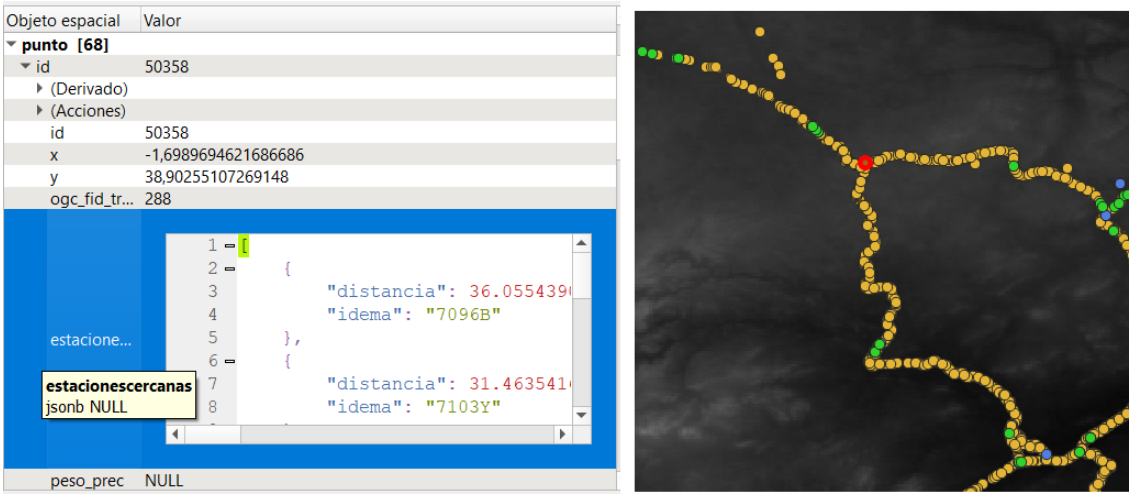


Ilustración 2. Ejemplo de Datos SIG, datos geográficos y datos no geográficos.

Dentro del proyecto en cuestión, se hacen uso de diversos tipos de datos SIG, entre los cuales se encuentran los **datos vectoriales**, que se almacenan en la memoria del ordenador como una serie de pares de coordenadas X, Y. Es por ello que es necesaria una investigación previa, estudio y comprensión de estos tipos de datos para su correcto uso en el proyecto. Estos datos son especialmente útiles para representar puntos, líneas y áreas en un espacio geográfico. En la **Ilustración 3** se pueden observar diferentes tipos de datos vectoriales. Estos datos suelen verse representados como puntos (p.ej. inicio y fin de las vías del ferrocarril), líneas (p.ej. carreteras) y polígonos (p. ej. delimitaciones de municipios). En el ámbito del presente TFG, los puntos y líneas serán integrados en los mapas interactivos de la aplicación desarrollada.

En cuanto a otros tipos de datos, se encuentran los **datos ráster**. Los datos ráster se utilizan en una aplicación SIG, por norma general, cuando queremos mostrar información que es continua en un área y no se puede dividir fácilmente en características vectoriales, un ejemplo de esto podría ser la representación de la variación de la temperatura en un área determinada, donde se utilizan datos ráster para mostrar la distribución espacial de la temperatura en toda el área (QGIS project, 2002). Los rásteres son fotografías aéreas digitales, imágenes de satélite, imágenes digitales o incluso mapas escaneados y los datos recogidos se almacenan en una cuadrícula de valores, es decir, se compone de una matriz de celdas o píxeles organizados en filas y columnas (Esri, 2021a). Y cada una de estas celdas contiene un valor que captura y puede representar

información sobre fenómenos del mundo real como interpretaciones o categorizaciones de los mismos, estos datos que se almacenan en formato ráster, los podemos diferenciar en:

- Los datos temáticos (también conocidos como discretos) representan entidades como datos de la tierra o de uso de la tierra. Un objeto discreto tiene límites conocidos y definibles, resulta sencillo establecer con exactitud el punto de inicio y finalización del objeto. Ejemplos de objetos discretos incluyen edificios, carreteras y parcelas.
- Los datos continuos representan fenómenos como la temperatura, la elevación o datos espectrales, entre ellos imágenes satelitales y fotografías aéreas. Éstos cambian de forma progresiva a medida que se van moviendo por la superficie.
- Las imágenes incluyen mapas escaneados o dibujos y fotografías de edificios.[2]

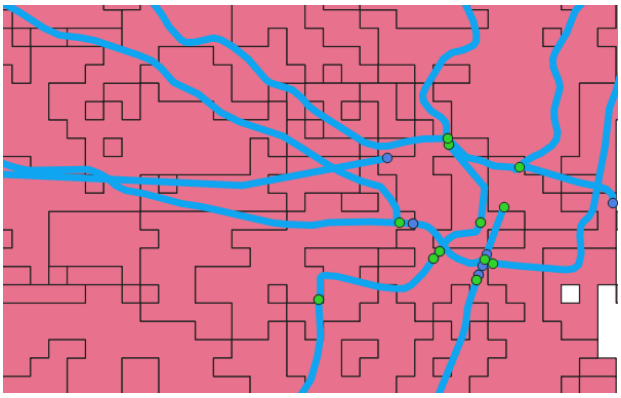


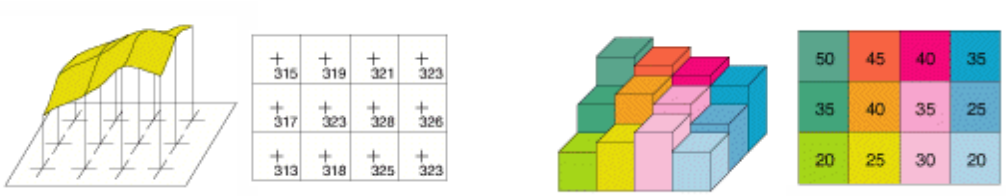
Ilustración 3. Datos vectoriales

Los rásteres temáticos y continuos se pueden visualizar en el mapa en forma de capas de datos junto con otros datos geográficos. Y como ya hemos mencionado previamente, estos valores en las celdas pueden abarcar diversas características, como categorías, magnitudes, alturas o valores espectrales Otros ejemplos relevantes son cuando una casilla puede indicar magnitudes como el porcentaje de precipitaciones, la temperatura en un determinado punto, entre otros. Estos ejemplos son destacados ya que será información que consideraremos al calcular las rutas en el contexto de este proyecto.

[2] Esta es una clasificación extraída de *¿Qué son los datos ráster?* del Environmental Systems Research Institute (2021). Recuperado, el 20 de junio de 2023, de https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm

Los rásteres de imágenes suelen emplearse como atributos en tablas, pueden ser representados gráficamente con datos geográficos y se utilizan para transmitir información adicional sobre las entidades geográficas de mapas (Esri, 2021a). Estos datos pueden visualizarse junto con información cartográfica para transmitir información adicional. Un ejemplo de utilización de rásteres de imágenes es el seguimiento en tiempo real de los vehículos de transporte público. Las empresas de transporte pueden utilizar tecnología GPS para registrar la posición de sus vehículos en tiempo real y crear datos ráster de imágenes que representen la intensidad del tráfico en diferentes zonas de la ciudad. Estos datos se pueden visualizar en un mapa utilizando SIG para mostrar las áreas donde el tráfico es más intenso, lo que permite planificar rutas adecuadas para el transporte público.

En cuanto a los valores que contienen estas celdas del ráster, para ciertos tipos de datos, como podemos analizar en la **Ilustración 4**a, el valor de la celda representa un valor medido en el punto central de la celda, un ejemplo es un ráster de elevación. Por otro lado, véase en la **Ilustración 4**b, el valor de la celda representa una muestra del cuadrado, es decir, de toda la celda.



(a) El valor se aplica al punto central de la celda (b) El valor se aplica a toda el área de la celda

Ilustración 4. Valores en la celda[3]

Además, las celdas pueden tener asignado un valor "NoData" para indicar la falta de información en esa ubicación específica (Esri, 2021a).

Pese a que la estructura de datos ráster es sencilla, su utilidad es amplia y diversa en numerosas aplicaciones. En un SIG, los datos ráster se pueden clasificar en cuatro categorías principales:

[3] Fuente: Esri. (2021). En el apartado "Características generales de datos ráster" de *¿Qué son los datos ráster?* Recuperado, el 20 de junio de 2023, de https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm

1. Rásteres en forma de mapas base. Un uso frecuente de los datos ráster en un SIG consiste en utilizarlos como fondo visual para otras capas de entidades.
2. Rásteres en forma de mapas de superficie. Los valores de elevación que se miden desde la superficie de la Tierra son el uso más habitual de los mapas de superficie, pero otros valores, como las precipitaciones, la temperatura, la concentración y la densidad de población, también pueden definir superficies que se pueden analizar espacialmente.
3. Rásteres en forma de mapas temáticos. Un enfoque habitual de análisis implica la categorización de una imagen satelital según las diferentes clases de cobertura del suelo.
4. Rásteres en forma de atributos de una entidad. Los datos ráster empleados como atributos de una entidad pueden ser imágenes digitales, documentos digitalizados o representaciones gráficas escaneadas asociadas a un objeto o lugar geográfico[4].

Cada celda en un ráster representa un área o superficie con el mismo ancho y altura, que es una fracción equivalente de toda la superficie representada por el conjunto de datos ráster, véase en la Ilustración 5, y el tamaño de éstas puede variar, ser tan grande o pequeño como sea necesario para representar la superficie del conjunto de datos ráster y las entidades dentro de ella, como un kilómetro cuadrado o incluso un centímetro cuadrado. El tamaño de la celda determina la resolución con la que se representarán los patrones o entidades en el ráster. Cuanto más pequeña sea la celda, más suave y detallado será el ráster. Sin embargo, a medida que aumenta el número de celdas, el procesamiento se vuelve más lento y se requiere más espacio de almacenamiento. Si el tamaño de la celda es demasiado grande, se puede perder información. Por ejemplo, si el tamaño de la celda es mayor que el ancho de una carretera, la carretera podría no aparecer en el conjunto de datos ráster (QGIS project, 2002).

[4] Esta es una clasificación extraída de *¿Qué son los datos ráster?* del Environmental Systems Research Institute (2021). Recuperado, el 20 de junio de 2023, de https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm

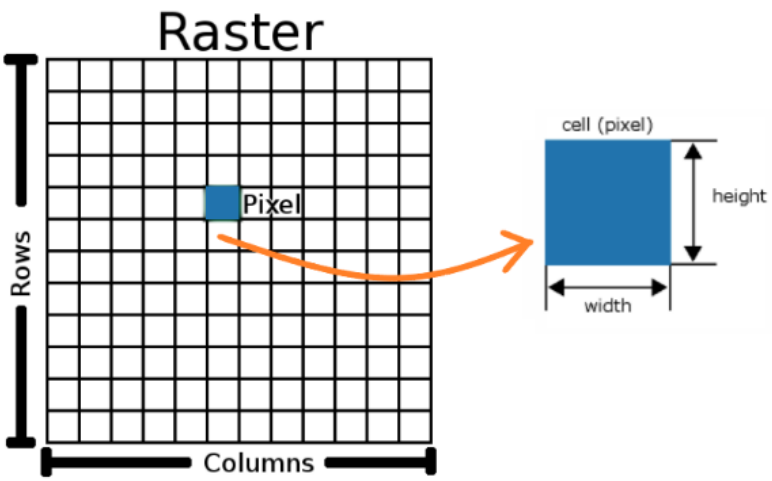


Ilustración 5. Ráster[5]

Existen rásteres con una sola banda o capa, que representa una única característica medida, mientras que otros poseen múltiples bandas. En esencia, una banda se representa mediante una matriz de valores de celdas, y un ráster con múltiples bandas contiene varias matrices de valores de celdas que se alinean espacialmente y representan la misma área geográfica. Cuando se utilizan múltiples bandas, cada celda contiene más de un valor asociado. Y cada una suele representar una sección específica del espectro electromagnético capturado por un sensor, véase en la Ilustración 6 (Esri, 2021).

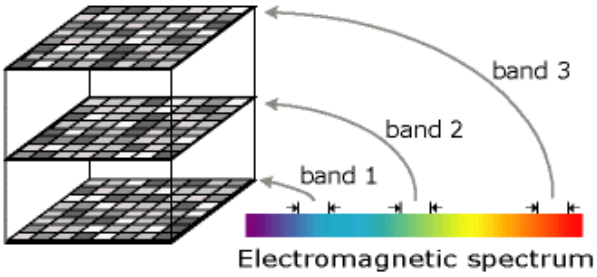


Ilustración 6. Múltiples bandas en un ráster

La ubicación de cada celda se define por su fila y columna en la matriz del conjunto de datos ráster. La matriz se representa como un sistema de coordenadas cartesianas, donde las filas son paralelas al eje x y las columnas al eje y del plano cartesiano. Los valores de fila y columna comienzan desde 0 (Esri, 2021a).

[5] Fuente: QGIS project. (2002). En el apartado "Datos Ráster" de *Una Introducción fácil a GIS*. Recuperado, el 20 de junio de 2023, de https://docs.qgis.org/3.28/es/docs/gentle_gis_introduction/index.html

Si bien se han descrito los datos SIG, el **archivo shapefile** es un formato específico que se utiliza para guardar estos datos en un SIG y se identifican por tener la extensión ".shp" cuando se visualizan desde el explorador de archivos. Un shapefile es un formato sencillo y no topológico que se utiliza para almacenar la ubicación geométrica y la información de atributos de las entidades geográficas. Las entidades geográficas de un shapefile se pueden representar por medio de puntos, líneas o polígonos (áreas) (Esri, 2021c).

Los **archivos** ráster **BIL** (Band Interleave By Line) son un formato de datos raster utilizado en los SIG para almacenar datos geográficos. La información se almacena en bandas, donde cada banda representa una capa única de datos. Este tipo de formato es utilizado principalmente para imágenes satelitales y fotografías aéreas, y se diferencia de otros formatos ráster por su estructura jerárquica de bandas. En cuanto a su estructura, los datos dentro de los archivos ráster BIL se organizan en filas y columnas, y cada píxel se almacena como un valor numérico asociado a diferentes variables. Estos archivos suelen requerir de un archivo de encabezado para ser interpretados correctamente (.hdr). Estos archivos de encabezado contienen información sobre los metadatos, el tamaño de los datos, las coordenadas de referencia y otra información importante que ayuda a los programas de SIG a interpretar los datos de la imagen. Los archivos de encabezado también contienen información sobre el número de bandas, el tamaño de píxel, la resolución y la profundidad de bit de la imagen. Esta información es procesada por los programas SIG para poder visualizar los datos en un mapa o en una imagen (Esri, 2021d).

En resumen, mientras que los archivos .shape almacenan información geográfica en formato vectorial, los archivos raster BIL son utilizados para almacenar datos geográficos en formato ráster. Ambos tipos de archivos pueden requerir archivos de encabezado para ser interpretados correctamente, y estos archivos de encabezado proporcionan información importante sobre la proyección, coordenadas y otro metadatos.

Algoritmos voraces

En el marco de este TFG, un objetivo es abordar el desafío de planificar y construir rutas óptimas para las redes de transporte, incluyendo ferrocarriles, carreteras y servicios lineales. Para lograr este objetivo, es fundamental contar con algoritmos que nos ayuden a calcular las rutas más cortaS entre dos puntos en una red, considerando diversos factores como la distancia, el tiempo o los Datos SIG, como abordamos en el apartado previo a éste.

Un algoritmo voraz toma decisiones en función de la información inmediata disponible, y busca maximizar algún criterio de optimización en cada paso individual. Funciona eligiendo el arco o la tarea que parece más prometedora en cada momento; nunca reconsidera esta decisión, sin importar la situación que pueda surgir más adelante. Los algoritmos voraces son, a menudo, simples de implementar y pueden proporcionar soluciones rápidas y eficientes en muchos casos (Brassard & Bratley, 1996, p. 189). Este es uno de los motivos que ha llevado a la selección dell algoritmo de Dijkstra, debido a su capacidad para encontrar la ruta más corta en una red ponderada.

El algoritmo de Dijkstra nos permite encontrar la ruta más corta desde un punto de origen a todos los demás puntos de la red. Crea una lista de los nodos visitados y asigna un peso a cada nodo. Luego, desde el nodo de partida, se considera cada nodo vecino y se actualiza su peso si un camino más corto es encontrado. Este proceso se repite hasta que todos los nodos hayan sido visitados. Su enfoque voraz nos permite tomar decisiones óptimas en cada etapa, minimizando la distancia recorrida en cada paso. Su facilidad de implementación y eficiencia en el tiempo de ejecución son aspectos clave que respaldan su elección (Brassard & Bratley, 1996, p.198).

Es importante tener en cuenta algunas de las limitaciones del algoritmo de Dijkstra. En redes extremadamente grandes, su aplicación puede volverse ineficiente. Requiere que las distancias entre los nodos sean no negativas, debido a su enfoque de toma de decisiones limitado, los algoritmos voraces no siempre encuentran la solución óptima global para todos los problemas. En algunas situaciones, pueden quedarse atrapados en mínimos locales o tomar decisiones que parecen buenas en el momento pero que resultan en soluciones subóptimas en general. Por lo tanto, es importante evaluar cuidadosamente si un algoritmo voraz es apropiado para un problema específico y considerar las posibles limitaciones. No obstante, en el marco actual de nuestro proyecto,

estas limitaciones no son un obstáculo y el algoritmo se adapta perfectamente a nuestras necesidades.

En definitiva, nuestro objetivo de planificar rutas óptimas para las redes de transporte requiere algoritmos efectivos, como el algoritmo de Dijkstra, que nos permitan calcular las rutas más cortas. Conscientes de las limitaciones y dispuestos a considerar otros algoritmos si los requisitos del proyecto cambian en el futuro. Por ahora, confiamos en la idoneidad del algoritmo para nuestro propósito.

2.2 Estado del Arte

En este apartado, se realizará una exposición de las principales herramientas relacionadas a la presentada en este TFG.

Herramientas similares

En el ámbito de este TFG, se estudiarán algunas alternativas, diversas herramientas y/o aplicaciones que tienen en cuenta varios factores al calcular rutas y ofrecen visualización de la misma aportando información complementaria. Se tendrá presente si éstas tienen en cuenta aspectos como el clima, la topografía y su relación con los sistemas de información geográfica, se presenta en también en forma de tabla, veáse en la **Tabla 1**, las ventajas y desventajas basadas principalmente en la consideración de datos climáticos y geográficos en tiempo real, aunque es posible que existan otros aspectos que no hayan sido mencionados. Esta información permitirá evaluar cuál de ellas tiene objetivos parecidos a los de este proyecto.

- Mapbox GL JS: esta biblioteca de JavaScript permite crear mapas interactivos con estilos personalizados y datos vectoriales. A pesar de que no ofrece funcionalidades específicas relacionadas con el clima o la topografía, puede integrarse con fuentes de datos externas para mostrar información relacionada con estas características. Además, Mapbox ofrece servicios adicionales, como la API de pronóstico del clima y datos de elevación del terreno, que se pueden combinar con Mapbox GL JS para obtener información climática y topográfica.
- Mapbox Fleet: es una solución diseñada para empresas que necesitan rastrear y gestionar flotas de vehículos en tiempo real. Permite el seguimiento de vehículos,

optimización de rutas y entrega de información en tiempo real sobre el rendimiento y la ubicación de la flota

- Google Maps: es una herramienta de búsqueda de ubicaciones que permite geolocalizar un punto concreto, calcular rutas, encontrar los lugares de interés más cercanos, visualizar mapas con imágenes de satélite, étc. Muestra información en tiempo real sobre el tráfico y las condiciones de conducción, pero hasta donde se ha llegado en la investigación, no tiene en cuenta directamente el clima al calcular la ruta. En condiciones climáticas adversas, como lluvia intensa o nieve, Google Maps puede mostrar rutas alternativas o estimaciones de tiempo de viaje más largas debido a las mismas.
- Google Maps: ofrece a los usuarios la posibilidad de crear, personalizar y compartir mapas personalizados de forma gratuita, para su uso en una amplia variedad de contextos. Los usuarios pueden agregar diversos elementos, como marcadores, líneas, formas o texto, entre otros, y también pueden colaborar en tiempo real con otros usuarios. Además, los usuarios tienen la libertad de crear itinerarios personalizados a su antojo, ya que la herramienta no proporciona automáticamente resultados a las consultas que se realicen. Por último, se puede exportar el mapa personalizado en diversos formatos para su uso en otras aplicaciones o dispositivos. La API de JavaScript de Google Maps es ampliamente utilizada para integrar mapas interactivos de Google Maps en aplicaciones web. Aunque la API en sí no proporciona funcionalidades directas relacionadas con el clima o la topografía, Google Maps ofrece servicios adicionales, como la API de pronóstico del clima y datos de elevación del terreno, que se pueden utilizar junto con la API de JavaScript para obtener información relacionada.
- Google Earth Engine: es una plataforma en línea proporcionada por Google que permite a los usuarios acceder y analizar grandes conjuntos de datos geoespaciales. Esta plataforma combina las imágenes de satélite y otras herramientas de Google Earth con herramientas de análisis avanzado para permitir la exploración y el análisis de datos geográficos en una amplia variedad de contextos.Con Google Earth Engine, los usuarios pueden acceder a una gran cantidad de datos geoespaciales, incluyendo imágenes de satélite, datos climáticos, datos de escala arbolada y otros tipos de datos geoespaciales. Además,

los usuarios pueden utilizar diversas herramientas de análisis avanzado, como la clasificación de imágenes, la detección de cambios y la medición de la vegetación. La plataforma también permite a los usuarios compartir y colaborar en proyectos con otros usuarios en línea, lo que facilita el trabajo en equipo y el intercambio de información. En resumen, Google Earth Engine es una plataforma en línea poderosa y versátil que permite a los usuarios acceder y analizar grandes conjuntos de datos geoespaciales en una amplia variedad de contextos.

- HERE Maps API: es otra opción para crear mapas interactivos en aplicaciones web. Además de funcionalidades de visualización y enrutamiento, HERE Maps ofrece servicios de pronóstico del clima y datos de elevación del terreno. Estos servicios se pueden aprovechar en combinación con la API para obtener información sobre clima y topografía.
- OpenRouteService: es una plataforma que ofrece servicios de enrutamiento, geocodificación y análisis geoespacial. OpenRouteService es un servicio web gratuito que proporciona datos y herramientas para la planificación de rutas y la navegación, especialmente enfocado en la movilidad sostenible y el transporte urbano. Este servicio se basa en el uso de datos geoespaciales de código abierto y utiliza algoritmos de enrutamiento avanzados para encontrar las mejores rutas para diferentes tipos de modos de transporte, como caminar, montar en bicicleta, conducir o utilizar el transporte público. Entre las herramientas disponibles en OpenRouteService se encuentra la posibilidad de calcular rutas con distintas opciones, como la ruta más rápida o la más corta, y también se pueden incluir criterios adicionales, como la elevación o la accesibilidad. Además, el servicio ofrece información sobre el tráfico en tiempo real en muchas ciudades, lo que permite a los usuarios planificar sus rutas de forma más eficiente. OpenRouteService también ofrece una API gratuita que permite a los desarrolladores integrar estas funcionalidades en sus propias aplicaciones, y la plataforma permite hacer cálculos avanzados de enrutamiento basados en parámetros personalizados. Es la más cercana al objetivo de nuestro proyecto pero es cierto que aparentemente no tiene en cuenta para el cálculo de la ruta las condiciones climáticas.

– MapQuest: es una plataforma de mapas en línea que ofrece servicios de visualización, enrutamiento y geocodificación. No se menciona que tenga funcionalidades específicas relacionadas con el clima o la topografía.

Tabla 1. Comparativa entre herramientas similares.

Herramienta	Características	Ventajas	Desventajas
Mapbox GL JS	Biblioteca de JavaScript para crear mapas interactivos	Integrable con fuentes de datos externas para mostrar información climática y topográfica	No ofrece funcionalidades específicas relacionadas con el clima o la topografía
Mapbox Fleet	Solución para gestionar flotas de vehículos en tiempo real	Permite rastrear y optimizar rutas	Diseñada para empresas y no para uso individual
Google Maps	Herramienta de búsqueda de ubicaciones y cálculo de rutas	Ofrece información en tiempo real sobre tráfico y condiciones de conducción	No tiene en cuenta directamente las condiciones climáticas para calcular la ruta
Google Maps	Posibilidad de crear, personalizar y compartir mapas personalizados de forma gratuita	Gran libertad para crear itinerarios personalizados	No proporciona automáticamente resultados a las consultas
Google Earth Engine	Plataforma en línea para acceder y analizar grandes conjuntos de datos geoespaciales	Ofrece una gran cantidad de datos geoespaciales y diversas	Orientada a usuarios con conocimientos avanzados en el tema

		herramientas de análisis avanzado	
HERE Maps API	API para crear mapas interactivos	Ofrece servicios de pronóstico del clima y datos de elevación del terreno	No se mencionan desventajas
OpenRouteService	Plataforma para la planificación de rutas y navegación enfocada en la movilidad sostenible y el transporte urbano	Permite calcular rutas con distintas opciones y criterios adicionales como la elevación o accesibilidad	No tiene en cuenta directamente las condiciones climáticas para calcular la ruta
MapQuest	Plataforma de mapas en línea	Ofrece servicios de visualización, enrutamiento y geocodificación	No se mencionan características específicas sobre el clima o la topografía

Conclusiones

Las herramientas analizadas no mencionan el uso directo de datos climatológicos para calcular la ruta. Esto significa que no tienen en cuenta las condiciones meteorológicas en la determinación de la ruta óptima. Aunque las herramientas proporcionan información complementaria sobre la ruta, como el tiempo y el tráfico, ésta información se limita a mostrar el estado actual de la ruta. No ofrecen consejos adicionales ni alternativas de ruta en función de las condiciones del clima u otros factores.

Las herramientas solo ofrecen rutas alternativas en caso de retenciones de tráfico. Es decir, solo consideran desvíos o caminos alternativos si existen congestiones en la ruta original. Además, no proporcionan opciones de ruta alternativa por otros motivos, como condiciones climáticas desfavorables o eventos especiales. Por último, aunque algunas herramientas permiten seleccionar la hora de salida o la hora deseada para iniciar el viaje, no proporcionan sugerencias de hora de salida para optimizar la ruta, ni aportaciones

sobre el momento más adecuado para evitar congestiones o aprovechar condiciones de tráfico más favorables. Sin embargo, la herramienta más destable de todas, OpenRouteService, tiene en cuenta información como la elevación o la accesibilidad, y permiten elegir entre diferentes opciones de ruta, como la más rápida o la más corta. Además, en OpenRouteService se pueden seleccionar distintos criterios de enrutamiento, y se pueden personalizar rutas en función de preferencias individuales.

En resumen, las herramientas analizadas carecen de la utilización directa de datos climatológicos en el cálculo de la ruta, no ofrecen consejos o alternativas de hora de salida para optimizar la ruta y solo consideran rutas alternativas en caso de retenciones de tráfico.

Capítulo 3

Metodología y herramientas

En este capítulo, se expondrá el enfoque adoptado para estructurar el desarrollo de este proyecto, el cual se basa en el framework Scrum. Se presentará una descripción del contexto del TFG, estableciendo las responsabilidades y artefactos involucrados en el proceso. Conjuntamente, se proporcionará un listado de las herramientas utilizadas, acompañado con una breve descripción de cada una de ellas.

En el ámbito de la gestión de proyectos, las metodologías *plan-driven* o clásicas se caracterizan por la planificación exhaustiva previa a la ejecución del trabajo, a pesar de que es posible incluir procesos para la gestión de cambios, pueden resultar costosas a nivel temporal y económico. Por su parte, las metodologías ágiles se basan en una planificación a corto plazo que considera la posibilidad de cambios en cada iteración. En este contexto, la metodología adoptada, inspirada en el framework Scrum, destaca por su flexibilidad y capacidad de adaptación en la gestión de proyectos, fomentando la colaboración, la transparencia y la rápida capacidad de respuesta ante cambios imprevistos.

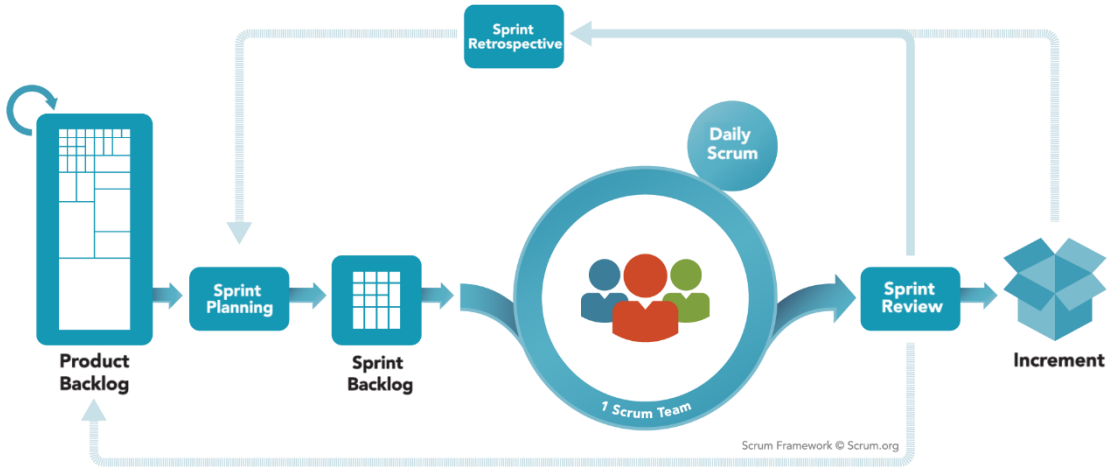


Ilustración 7. Framework Scrum[6]

3.1 Metodología adoptada

Como ya se ha mecionado previamente, se ha llevado a cabo una adaptación del framework Scrum a las particularidades del proyecto. Este proyecto ha sido llevado a cabo a lo largo de un período de cinco meses interrumpidos. Se ha estructurado en Sprints, aplicándole un enfoque iterativo con una duración aproximada de dos semanas cada uno, aunque no todos poseen la misma duración.

Previamente a la explicación de cómo se ha organizado el equipo, resulta necesario clarificar los conceptos que podemos ver en la **Ilustración 7**, la cual esquematiza el sistema de Scrum. En la gestión de este proyecto se recogen tanto los requisitos funcionales y no funcionales como elementos ó historias de usuario, las cuales a su vez se componen de tareas y se jerarquizan por orden de prioridad dentro de un conjunto llamado "*Product Backlog*" *(PB)*. El "Sprint Backlog" *(SB)* se compone de tres elementos. En primer lugar, el *Sprint Goal*, que en este caso particular ha sido denominado como "Objetivo del Sprint", representa la meta que se busca alcanzar al finalizar el Sprint, y sirve como guía para orientar las actividades y el trabajo realizado durante este periodo. Además, el *SB* incluye un conjunto específico de historias de usuario y/o tareas seleccionadas del *PB* para su desarrollo durante el sprint en curso (Schwaber &

[6] Fuente: (Scrum.org, 2020). En el apartado "What is Scrum?" de *Framework Scrum*. Recuperado, el 25 de junio de 2023, de https://www.scrum.org/learning-series/what-is-scrum

Sutherland, 2020, p. 11). Estas historias de usuario y/o tareas representan las funcionalidades o trabajos concretos que se deben completar durante este periodo de tiempo determinado. Por último, el *SB* también contempla un plan para la entrega del *Incremento* al finalizar el sprint. El *Incremento* se produce cuando un elemento del *PB* cumple con la *Definition of Done* (*DoD*) (Schwaber & Sutherland, 2020, p. 12), en este proyecto lo hemos definido como el estado de finalización de un elemento del *PB*, es decir, una vez que se han completado todas las tareas asociadas a dicho elemento y por lo tanto, se considera que se ha alcanzado un *Incremento*. Este *Incremento* representa un avance hacia el logro del Objetivo del *PB*, impulsando el progreso general del proyecto. Finalmente, el concepto de artefacto en el contexto del desarrollo ágil y en este TFG. Se define como los elementos que representan trabajo o valor y son generados en cada iteración del proceso. Estos artefactos cumplen un papel importante al mejorar la transparencia y el enfoque del equipo, y son compromisos que proporcionan información relevante y medible para evaluar el progreso realizado en el proyecto, para el *Product Backlog* es el objetivo del *PB*, para el *Sprint Backlog* es el objetivo del *SB* y para el *Incremento* es la *Definition of Done*.

Responsabilidades

Por consiguiente, una vez precisados los conceptos, se definen tres responsabilidades adaptadas y específicas dentro del equipo, siendo asumida cada una de ellas por una única persona: *Product Owner* (PO), *Scrum Master* (SM) y *Developers*:

i. Product Owner (PO), Enrique Arias, responsable de:
   a) Maximizar el valor del producto resultante del trabajo, tanto como la aplicación y la memoria.
   b) Gestionar de forma eficaz el *PB.*
   c) Comunicar los objetivos.
   d) Ordenar las tareas y/ó historias de usuario del *PB*.

ii. Scrum Master (SC), José Antonio Mateo, responsable de:
   a) Ayudar al equipo a enfocarse en crear incrementos de alto valor que cumplan con la *DoD*.
   b) Ayudar al Equipo Scrum a comprender la necesidad de elementos claros y concisos del *PB*.

iii. Developers, Henar Velasco, responsable de:

a) Crear y definir el *Sprint Backlog*.

b) Adaptar la planificación del sprint para alcanzar el objetivo del mismo.[7]

Finalmente, es relevante resaltar la contribución del cliente, Francisco Javier Tapiador, en el proyecto. Como cliente único, Francisco Javier Tapiador ha desempeñado un papel fundamental al explicar el problema de estudio, proporcionar datos y recursos necesarios, y participar en reuniones con el equipo en ciertos sprints. Su colaboración ha sido esencial para obtener información sobre el problema y establecer los requisitos necesarios para el desarrollo del proyecto.

Dinámica

La dinámica adoptada en este proyecto es iterativa y se siguen los siguientes pasos generales (Schwaber & Sutherland, 2020, p. 3):

1. El Product Owner refina y ordena el PB.
2. Todo el equipo, selecciona qué historias de usuario y/o tareas del PB formarán parte del SB. Durante el período de tiempo acordado, se enfocan en llevar a cabo las tareas y lograr los objetivos establecidos.
3. Al finalizar cada Sprint, el Equipo Scrum se reune para inspeccionar los resultados obtenidos. Se realizan ajustes y adaptaciones necesarios para el próximo Sprint, con el objetivo de mejorar el proceso y obtener mejores resultados en futuras iteraciones.
4. Este ciclo de trabajo se repite de manera continua a lo largo del proyecto, permitiendo un enfoque iterativo y una mejora progresiva en la calidad y el valor del producto.

Por otro lado, es importante destacar que los eventos específicos que ocurren en cada Sprint, considerando el propio Sprint como un evento; tales como *Daily Scrum*, *Sprint Review* y *Sprint Retrospective* (Schwaber & Sutherland, 2020, p. 7), pueden variar según las necesidades y características particulares de este proyecto. Estos eventos se adaptarán y se especificarán más detalladamente en el siguiente capítulo de esta

[7] Estas responsabilidades son extraídas del apartado “Scrum Team” de Schwaber, K., & Sutherland, J. (2021). *The Scrum Guide.* 2020.

memoria. De este modo, se garantiza que los eventos se ajusten de manera adecuada a los objetivos específicos de este proyecto en particular.

3.2 Plan de trabajo

Se plantea una planificación aproximada para el desarrollo de este TFG, acordada por todo el equipo y se encuentra representada mediante un diagrama de Gantt que se puede observar en la **Ilustración 8**. Es importante tener en cuenta que, a pesar de utilizar un framework ágil y no se debería llevar a cabo una planificación, se considera necesario llevar a cabo esta planificación orientativa en el contexto de la entrega del TFG.

1. Inicio y estudio: comprenderán los sprints enfocados a tareas de estudio del problema, investigación del campo, definición de product backlog, selección de herramientas.
2. Desarrollo: incluye los sprints que sus tareas prioritarias consisten en la elicitación de requisitos, análisis de los mismos, definición de la arquitectura e implementación.
3. Imprevistos y Memoria: una vez finalizados los sprints relacionados con las tareas de implementación, se podrá proceder a la elaboración de la memoria, teniendo en cuenta la posibilidad de que puedan surgir algunos imprevistos.
4. Depósito y Defensa: entrega de la memoria del TFG, preparación y defensa del mismo ante el tribunal.

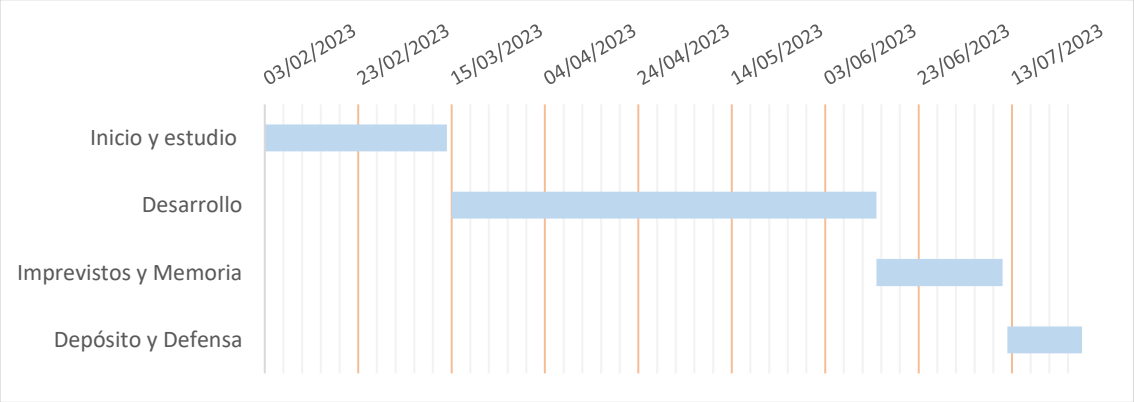


Ilustración 8. Diagrama de Gantt orientativo

Para finalizar, en la **Ilustración 9** se puede observar cómo finalmente se han distribuido los sprints en el desarrollo de este TFG. Se precisan más detalles sobre los

objetivos de cada sprint, historias de usuario y tareas desarrolladas, impedimentos surgidos y reuniones en el Anexo I, que se podrá encontrar al final de esta memoria.

Ilustración 9. Distribución de los sprints.

3.3 Herramientas

El sistema desarrollado sigue una arquitectura específica que se muestra en la **Ilustración 10**. La comprensión detallada de cada elemento dentro de la arquitectura del sistema permitirá una visión completa y clara de cómo se ha construido y qué funciones cumple cada uno de ellos. Esto servirá como base para comprender el funcionamiento del proyecto en su conjunto y facilitará su mantenimiento y futuras mejoras. A continuación, se detallarán las diferentes herramientas, tecnologías y componentes presentes en el sistema, explicando su funcionalidad y relevancia en el proyecto

El sistema consta de un servidor que utiliza Node.js. Además, se utiliza el framework Express, un módulo que nos va ayudar a crear el servidor y así construir nuestra Rest API. En el backend del sistema, se han instalado una serie de módulos que destacables en el desarrollo del servidor y en la conexión con la base de datos:

- El primer módulo instalado es Typescript, que permite utilizar un único lenguaje de programación tanto en el frontend como en el backend. Con Typescript, es posible traducir el código escrito en Typescript a JavaScript.

- El servidor está conectado a una base de datos Postgres que utiliza la extensión PostGIS. Esta extensión permite el almacenamiento y manipulación de objetos geográficos, lo que resulta fundamental para el manejo de datos espaciales en el sistema. Para establecer la conexión con la base de datos, se ha instalado el módulo pg-promise.
- Además, se ha instalado el módulo morgan, que permite visualizar las peticiones realizadas por el cliente en la consola del servidor. Esto resulta útil para el seguimiento y depuración del sistema durante el desarrollo.
- Por último, se ha utilizado el módulo cors, el cual posibilita la comunicación entre dos servidores. Dado que Angular levanta un servidor en el puerto 4200, es necesario utilizar el módulo cors para permitir la comunicación entre este servidor y la API Rest del sistema.

En el lado del frontend, se ha utilizado el framework Angular para crear la aplicación, la cual ha sido nombrada "1Dpath".

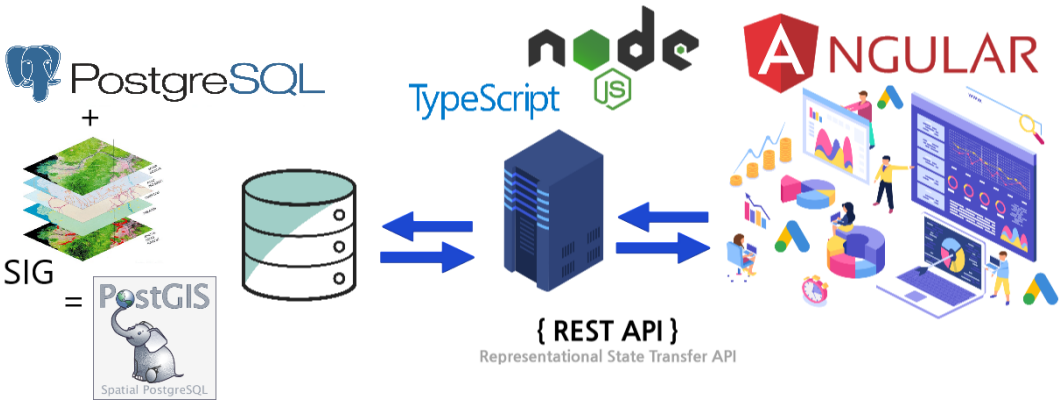


Ilustración 10. Arquitectura del sistema.

En el entorno de trabajo, se utilizan varios componentes de software que desempeñan roles específicos y contribuyen al desarrollo del proyecto. Estos componentes incluyen programas, entornos de desarrollo integrados (IDEs), bibliotecas específicas y otras herramientas esenciales que se describen a continuación:

- Microsoft Teams: la plataforma de colaboración entre el equipo que ha posibilitado la comunicación y el intercambio de documentos en el contexto de este proyecto.
- Monday.com: una plataforma de gestión de proyectos y colaboración en línea. Se ha utilizado para administrar tanto el Product Backlog como el Sprint Backlog,

facilitando la organización y priorización de los requisitos y funcionalidades del sistema, así como la selección y asignación de historias de usuario y tareas para cada sprint.

- Github Desktop: facilita la gestión y colaboración en proyectos alojados en GitHub, Git para el control de versiones en el contexto del desarrollo de software y GitHub como plataforma de alojamiento de repositorios.
- Visual Studio Code: como editor de código fuente desarrollado por Microsoft, se ha utilizado para desarrollar la aplicación requerida para este TFG.
- QGIS: es un software SIG libre y de código abierto que permite la captura, visualización, análisis y gestión de datos georreferenciados. En este proyecto se ha utilizado como apoyo a la comprensión y análisis del contenido de los ficheros proporcionados por el cliente como ya se comentará en el Capítulo 4.
- Draw.io: es una herramienta de diagramación en línea que permite crear diagramas y gráficos de manera sencilla. Una herramienta, utilizada en este proyecto, para el modelado UML (lenguaje de modelado unificado) que permite crear diagramas de clase, diagramas de secuencia, diagramas de actividad, entre otros.
- PostMan: esta herramienta de colaboración de desarrollo de API permite crear, probar, documentar y compartir API. En este proyecto, una de sus aplicaciones claves fue validar las solicitudes al servidor y garantizar la obtención de los resultados deseados.
- PgAdmin y PostGIS: es un software de gestión y desarrollo de bases de datos que se integra con el sistema de gestión de bases de datos PostgreSQL. Además, ha sido utilizada de manera conjunta con PostGIS, una extensión que añade soporte para datos geoespaciales al mencionado sistema de gestión de bases de datos PostgreSQL.
- Invision y Figma: dentro del marco del proyecto, se utilizaron estas dos herramientas de prototipado que han permitido crear y compartir prototipos interactivos para la aplicación web y que además han posibilitado la integración a otras herramientas de diseño.

Frameworks

Angular junto con sus complementos Angular Material y Material Design, es el framework utilizado para el desarrollo en el frontend de esta aplicación. En cuanto a los IDEs, se utiliza Angular CLI (Command Line Interface) como entorno de desarrollo para la construcción y gestión del proyecto Angular. Angular CLI proporciona una serie de comandos y utilidades que simplifican el proceso de desarrollo, como la generación de componentes, servicios y módulos

Express.js: es el framework de backend utilizado que ha permitido estructurar y desarrollar el servidor de la aplicación web y APIs.

Librerías

Leaflet: es una biblioteca de JavaScript de código abierto que se ha utilizado en la aplicación para implementar e integrar paneles con mapas interactivos y basados en la web. Esta biblioteca proporciona diversas funciones y características que permiten a los usuarios personalizar y trabajar con los mapas.

Material Symbols de Google Fonts es una biblioteca de iconos de código abierto que ha sido utilizada con el objetivo de mantener una estética y diseño coherentes en la aplicación, en línea con el diseño proporcionado por Material Design.

Capítulo 4

Análisis y Diseño

En este capítulo se introduce el sistema desarrollado. Como se comentaba en el capítulo de introducción, esta aplicación permitirá una mejor toma de decisiones a la hora de elegir una ruta. Se divide en dos secciones principales. En la primera sección, titulada "Análisis", se presentarán y explicarán diversos diagramas que tienen como objetivo representar y comunicar diferentes aspectos del sistema y sus procesos. Eso nos permitirá comprender mejor la estructura y el funcionamiento del sistema. En la segunda sección, denominada "Diseño", se expondrá la interfaz del sistema, se mostrará cómo se ha diseñado la interfaz de usuario para ofrecer una experiencia intuitiva y efectiva a los usuarios. Se describirán los elementos visuales, las funcionalidades y la disposición de los componentes de la interfaz.

4.1 Análisis

En primer lugar, se proporcionará una descripción detallada de la estructura general del proyecto, enfocada en lograr un mejor mantenimiento del código y promover la modularidad y escalabilidad. Se explicará cómo se ha organizado y distribuido de manera estratégica las carpetas y directorios del proyecto para facilitar la administración y promover una arquitectura flexible. Posteriormente, se presentará un diagrama de clases de la base de datos, el cual representa la estructura y las relaciones entre las entidades principales del sistema. Por último, se mostrará un diagrama de secuencia, el

cual ilustra la interacción y el flujo de eventos entre los diferentes elementos de la aplicación. Este diagrama permitirá comprender cómo se ejecutan las acciones y se comunican los distintos componentes durante la ejecución de un escenario específico.

A continuación, se procede a explicar cómo se ha estructura el proyecto, tal y como muestra la **Ilustración 11**:

- Carpeta *Server*: se encuentra todo lo relativo a la configuración del servidor. Entre los archivos que se encuentran aquí se halla el archivo "index.ts" y el archivo "database.ts", este último se encarga de gestionar la conexión a la base de datos. Los archivos que definen las rutas se encuentran en la carpeta "routes". En la carpeta "controllers" se encuentran los archivos donde se define la acción a ejecutar cuando se hace una petición a alguna de las rutas.
- Carpeta *Cliente*: aquí es donde se encuentra todo lo referente al frontend. La carpeta app, contiene dos grandes secciones. La primera sección, la carpeta "core", se ubican los modelos, los servicios y la carpeta "shared" es dónde se encuentran los elementos compartidos, como contenedores y componentes que se utilizan en todo el proyecto. En la segunda sección, la carpeta *public*, se encuentra la parte accesible a todos los usuarios de la aplicación. Se ha realizado esta distinción en caso de que en un futuro se desee crear una sección privada donde se deban tener en cuenta distintos permisos para diferentes usuarios. Actualmente, en la etapa actual del proyecto no se requiere dicha carpeta privada.

Ahora se procederá a explicar el diagrama entidad-relación de la base de datos, el cual se encuentra representado en la **Ilustración 12**. Este diagrama representa las entidades y las relaciones entre ellas en el contexto del problema del ferrocarril. El diagrama consta de tres entidades principales. La primera entidad es "tramo", que representa un segmento de las vías del ferrocarril. La segunda es "punto", la cual representa cualquier punto que forme parte de un tramo. Además, la entidad "punto" cuenta con una clave foránea llamada "estacionescercanas", la cual se relaciona con "estaciones_aemet". Esta última entidad representa las estaciones de AEMET que recogen datos climatológicos.

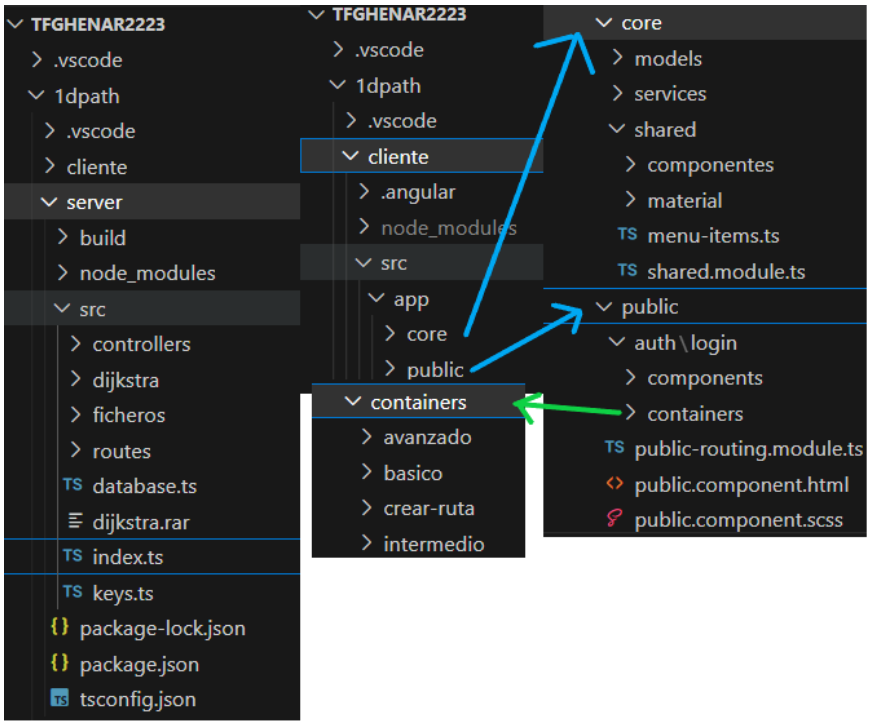


Ilustración 11. Estructura del proyecto.

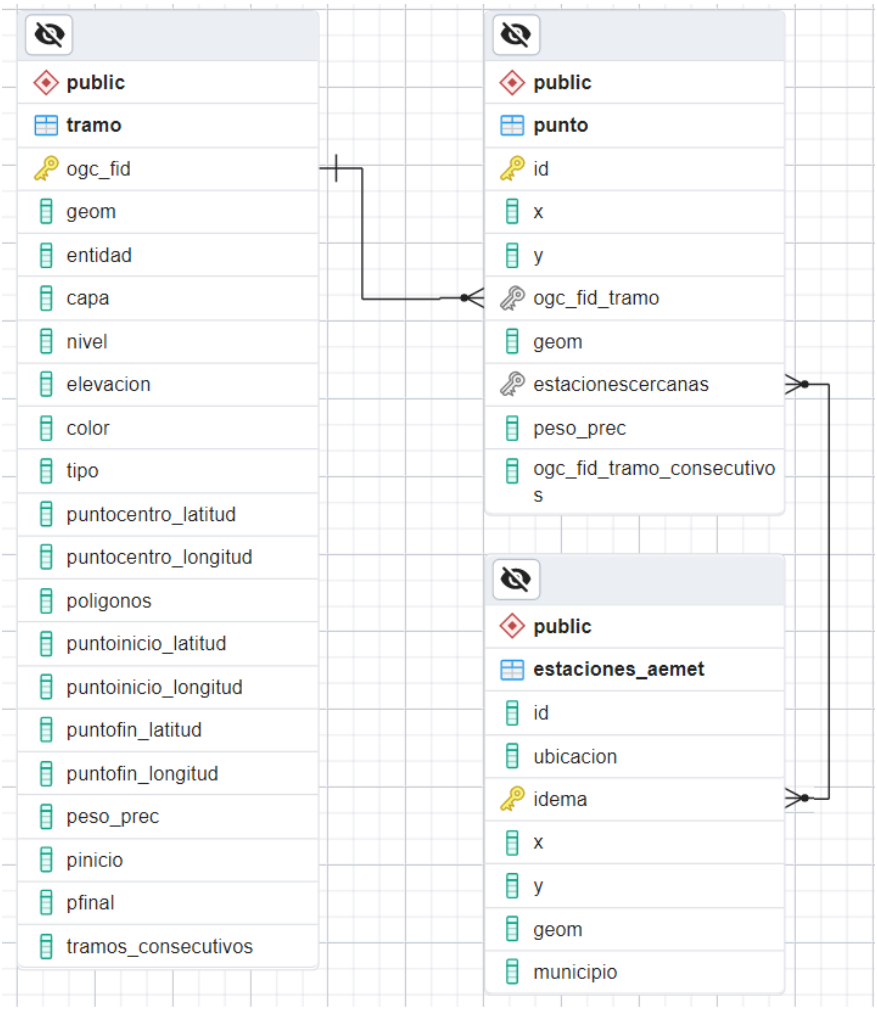


Ilustración 12. Modelo Entidad Relación de la base de datos.

La relación entre las entidades en el contexto del sistema se establece de la siguiente manera: un tramo puede tener múltiples puntos asociados, mientras que un punto pertenece únicamente a un tramo. Es importante destacar que, en términos prácticos, puntos con la misma geometría pertenecen a tramos diferentes. Sin embargo, debido a que la clave primaria de la tabla es el ID, la relación entre tramo y punto se representa como se muestra en el diagrama entidad-relación. En cuanto a la relación entre la tabla "punto" y "estaciones_aemet", se trata de una relación de *muchos a* muchos y así se refleja en el diagrama. Sin embargo, en este caso no se ha creado una tabla intermedia en la base de datos fruto de dicha relación y por eso mismo no se representa en el diagrama. En su lugar, se ha utilizado una columna denominada "estacionescercanas" en la entidad "punto", que es de tipo JSONB. En esta columna se almacena un listado de las estaciones de AEMET asociadas a un punto, incluyendo su IDEMA (identificador de la estación) y la distancia a la que se encuentra, sin la necesidad de crear más tablas.

Como se comentó anteriormente, el modo avanzado de la aplicación ofrece a los usuarios una destacada funcionalidad para calcular rutas entre estaciones de inicio y fin. La interacción del usuario con esta función se encuentra representada en el diagrama de secuencia de la **Ilustración 13**. La interacción se divide en tres pasos principales. En el primer paso, el usuario selecciona la estación de inicio, lo cual provoca que el sistema actualice el mapa y guarde la información correspondiente a dicha estación. En el segundo paso, el usuario realiza una selección para la estación de fin de la ruta y el sistema responde de una forma similar al primer paso. Luego, en el tercer paso, se presentan dos paneles, como se describe detalladamente en el apartado de diseño de este mismo capítulo. El primer panel permite al usuario configurar y ponderar los parámetros relevantes para el cálculo de la ruta (si lo desea), mientras que el segundo panel muestra un resumen de esta información. Una vez completados estos pasos, el usuario debe pulsar el botón "calcular ruta" para obtener el resultado deseado. El sistema responderá actualizando el mapa con la solución correspondiente. Es importante destacar que el usuario tiene la flexibilidad de repetir esta interacción tantas veces como desee.

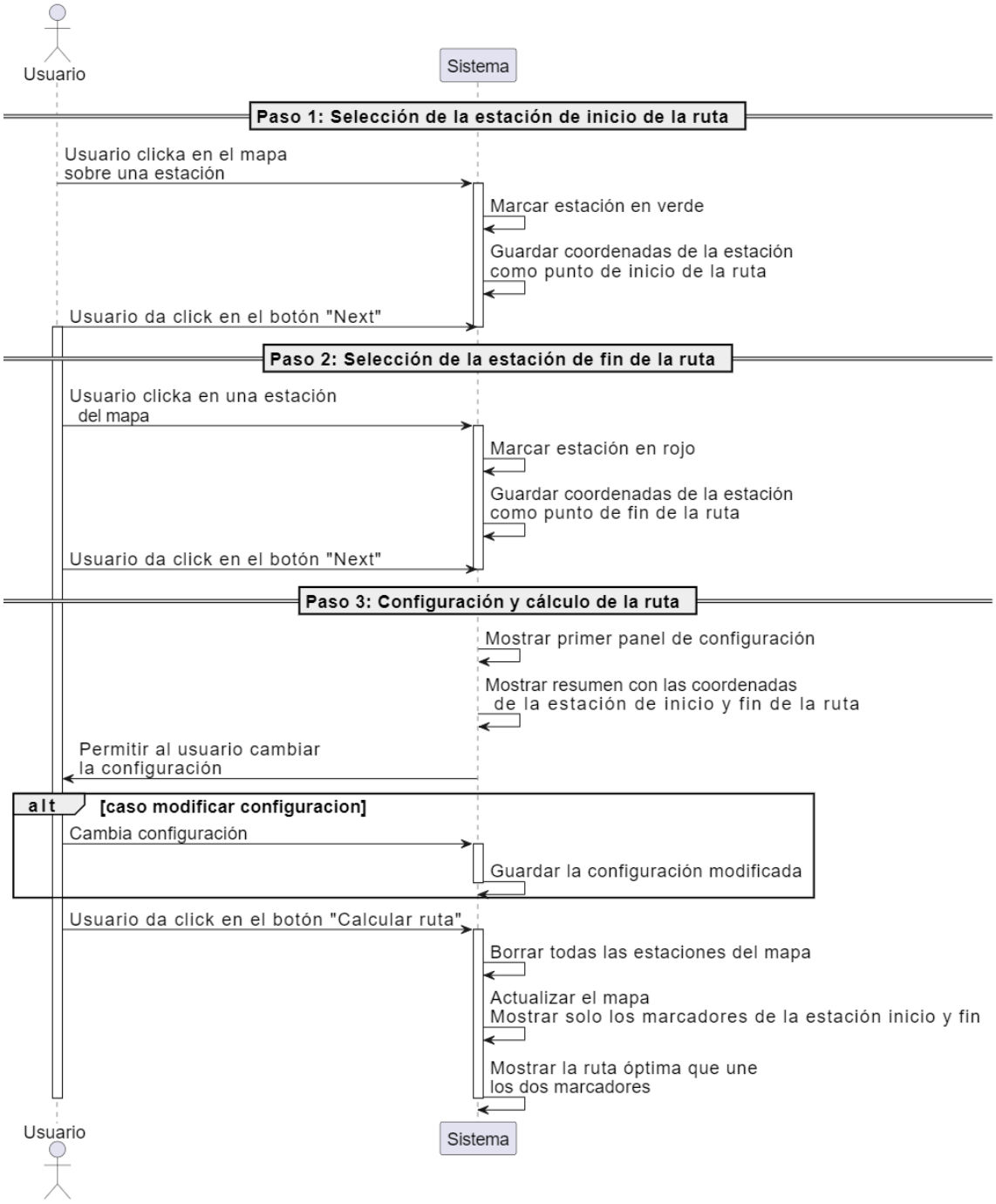


Ilustración 13. Diagrama de secuencia sobre la interacción de un usuario con el sistema para calcular una ruta.

4.2 Diseño

Esta sección se centra en describir y detallar la interfaz de la aplicación desarrollada. Se proporcionará una visión completa de cómo se ha diseñado la interfaz para una experiencia de usuario intuitiva y eficiente.

Nótese que la aplicación actualmente se encuentra en un entorno local y no ha sido desplegada en un servidor público. Por lo tanto, para acceder a la aplicación, el usuario deberá contar con un dispositivo que tenga la aplicación alojada localmente. Al ingresar a la aplicación, el usuario se encontrará con dos barras visibles en la interfaz. Una de ellas, situada en posición horizontal, es el toolbar, señalado en la **Ilustración 14** con el número 1, el cual consta de un botón que permite abrir o cerrar la barra o panel vertical, posibilita al usuario de ocultar o mostrar el menú principal. Mientras que la otra es una barra vertical que corresponde al menú principal, se puede apreciar en la **Ilustración 14** señalizado con el número 2, dispone de tres opciones que reflejan los distintos modos disponibles en el sistema: modo básico, intermedio y avanzado. Cada uno de estos modos ofrece un conjunto de funcionalidades y características específicas que desgranaremos a continuación.

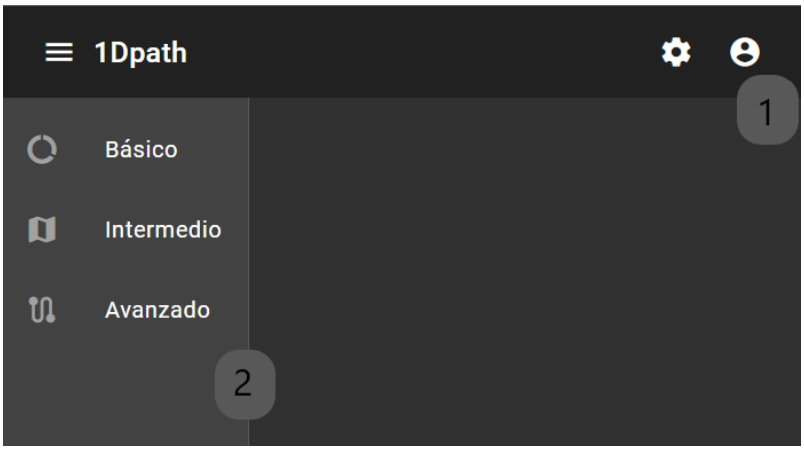


Ilustración 14. Vista general 1Dpath

Modo Básico

En el Modo Básico, como propuesta preliminar sujeta a validación por parte del cliente, se presenta una interfaz de poca interacción que consta de cuatro componentes principales (véase en la **Ilustración 15**). Estos componentes proporcionan información relevante sobre las condiciones climáticas y el estado de la red en cuestión. El primer bloque de la interfaz muestra información relacionada con las condiciones climáticas. El

segundo bloque está compuesto por un mapa interactivo que muestra las alertas e incidentes presentes en distintas zonas de la red mediante marcadores. El tercer bloque se subdivide en dos secciones. A la izquierda, se encuentra un gráfico que presenta estadísticas detalladas sobre el estado de las vías, en este caso, del ferrocarril. El gráfico permite identificar visualmente la proporción de vías que se encuentran en estado de normalidad y aquellas que están en proceso de obras o mantenimiento. En la parte derecha del tercer bloque, se muestra un mapa complementario en el cual también se marcan las vías, utilizando los mismos colores que en el gráfico. Esto facilita la correlación entre la representación visual y las estadísticas presentadas, permitiendo al usuario tener una visión más clara y completa del estado de las vías en tiempo real.

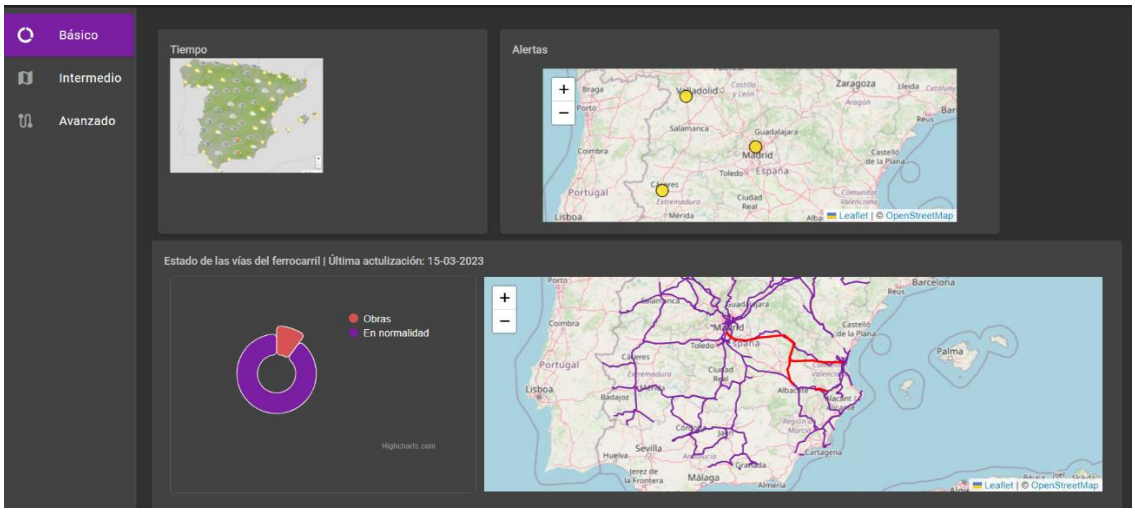


Ilustración 15. Vista general de modo Básico 1Dpath

Modo Intermedio

En el Modo Intermedio, se incorporan más capacidades de interacción en comparación con la vista anterior, ya que incluye un mapa interactivo para visualizar información geográfica relevante. La interfaz se divide en dos bloques, como se muestra en la Ilustración 16. En el bloque de la derecha, se encuentra un mapa interactivo que permite al usuario explorar y obtener información complementaria a la que se presenta en el modo básico. Para acceder a esta información adicional, se debe hacer clic en el icono ubicado en la esquina superior derecha del mapa, tal como se indica en la Ilustración 17. Al hacer clic en dicho icono, se desplegará un menú que permitirá cambiar el mapa base y ajustar la visibilidad de las estaciones disponibles, en este caso, las estaciones del ferrocarril, como se muestra en la misma ilustración. Cuando se hace clic

en una estación específica en el mapa, se abre un *popup* en el mapa con la coordenadas de la misma y se mostrará en el bloque de la izquierda un gráfico con el que se puede interaccionar con información detallada sobre esa estación, tal como se ilustra en el ejemplo de la **Ilustración 18** En este ejemplo se muestran las condiciones climatológicas. Si no se clica en niguna estación, por defecto, se mostrará instrucciones de cómo obtener información sobre las estaciones como se puede observar también en la misma ilustración.

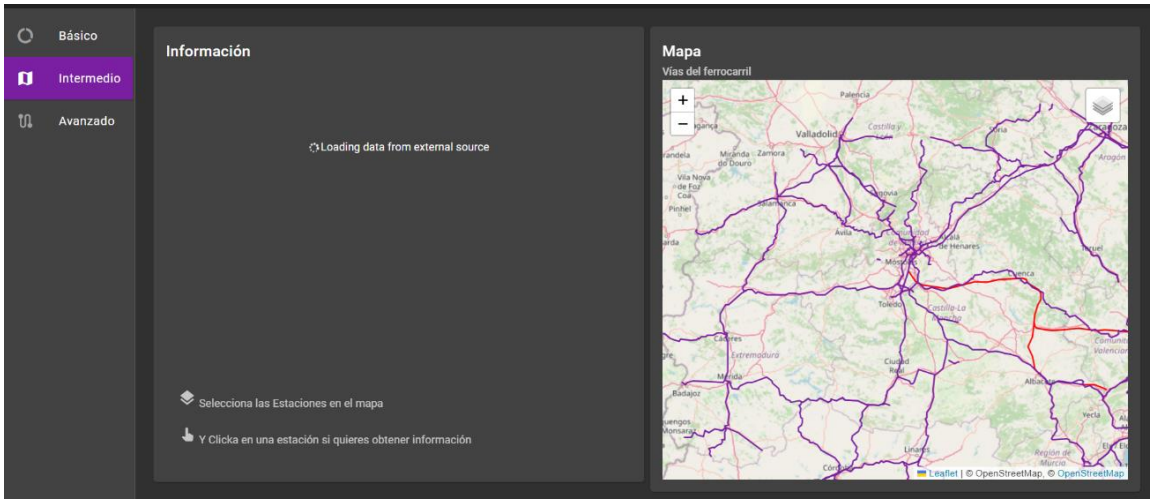


Ilustración 16. Vista general del modo intermedio 1Dpath

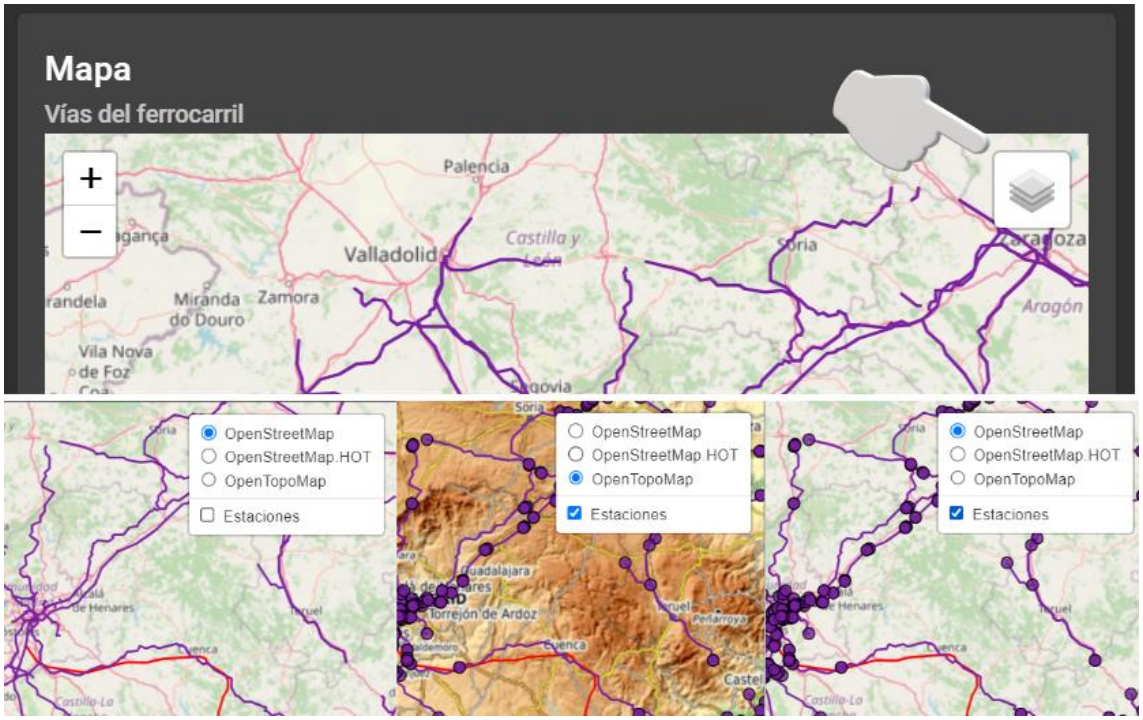


Ilustración 17. Botón y sus opciones en el mapa en la modo intermedio.

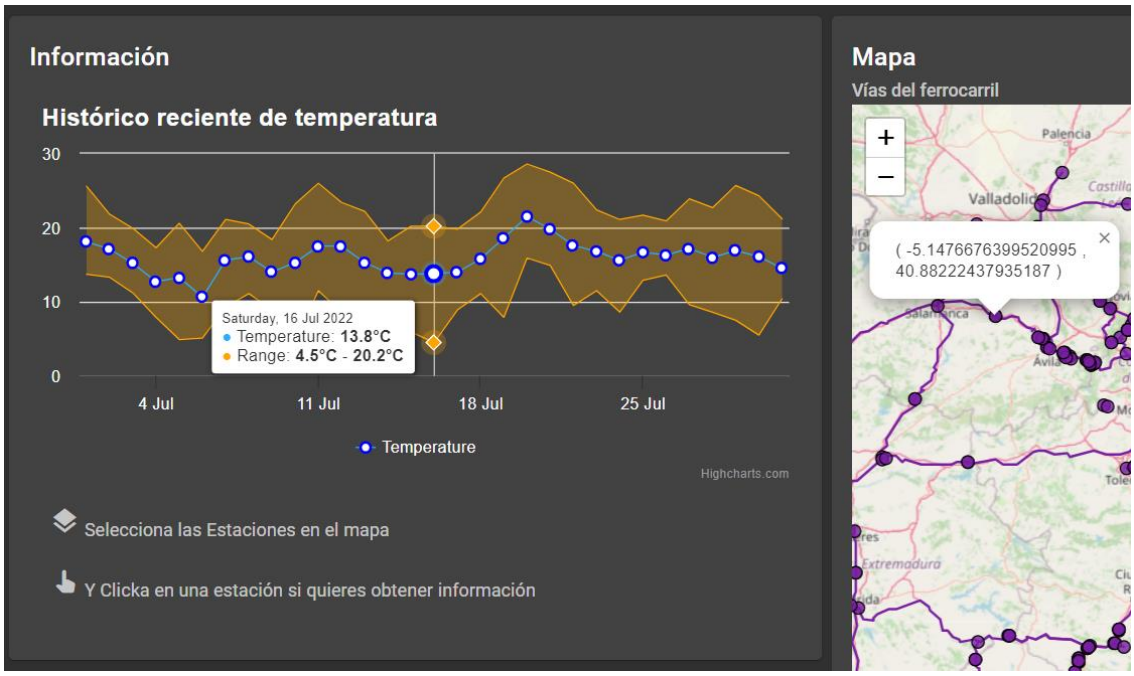


Ilustración 18. Panel de información de una estación en el modo intermedio.

Modo Avanzado

El Modo Avanzado ofrece una capacidad interactiva que permite realizar consultas de rutas en un mapa y mostrar los resultados correspondientes. En la vista general del Modo Avanzado, se presentan dos paneles. En el panel dos, veáse en la **Ilustración 19**, se encuentra un mapa interactivo que, en este ejemplo, muestra las vías del ferrocarril. En dicho mapa, se señalan las estaciones de manera predeterminada. Para facilitar la comprensión del mapa, se incluye una leyenda, como se muestra en la **Ilustración 19** ampliada. En la leyenda, se indica que los iconos de color morado representan estaciones individuales, mientras que los iconos en forma de círculo indican agrupaciones de varias estaciones. Al hacer clic en un icono de agrupación, el mapa se amplía automáticamente y se realiza un zoom para mostrar de manera más detallada esa zona específica y las estaciones individuales que la componen. En el bloque 1, veáse en la **Ilustración 19**, se encuentra el paso a paso para encontrar la ruta entre dos puntos. Este proceso detallado se desglosará en el caso de estudio específico. Por último, también lo podemos observar en la **Ilustración 19**, específicamente señalado con el número 4, que el usuario tiene la capacidad de ajustar el tamaño del bloque 2 arrastrándolo hacia una posición más amplia. Esto le permite al usuario personalizar el tamaño del bloque donde se encuentra insertado el mapa, según sus preferencias, y se puede identificar fácilmente en la esquina inferior derecha del bloque 2 debido a un tono de color más claro.

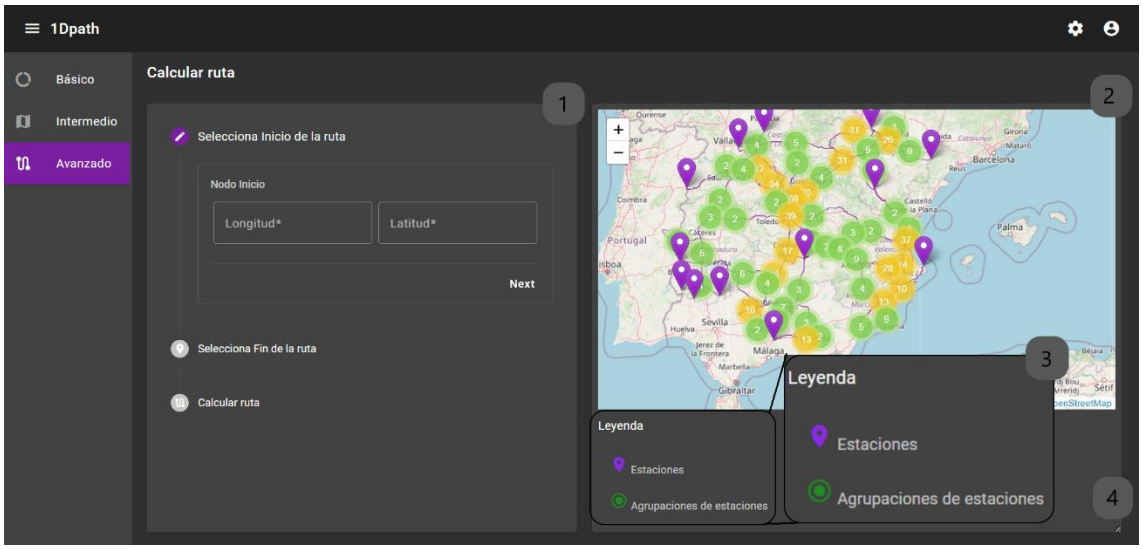


Ilustración 19. Vista general del modo avanzado 1Dpath.

Capítulo 5

Caso de estudio

Este capítulo se dedica a la explicación del caso de estudio Albacete-Murcia, el cual se enfoca en el análisis detallado de la ruta que conecta estas dos ubicaciones geográficas específicas. Se explorarán las características y particularidades de este caso de estudio, teniendo en cuenta aspectos como las vías disponibles para el cáculo de la ruta.

5.1 Caso de estudio Albacete-Murcia

El caso de estudio Albacete-Murcia tiene como propósito verificar si la aplicación cumple con el objetivo de calcular la ruta utilizando el algoritmo de Dijkstra, teniendo en cuenta, en este ejemplo, únicamente la temperatura. Se proporcionan valores representativos de temperatura asociados a los puntos que forman parte de los diferentes tramos.

En el panel de la derecha, donde se encuentra el mapa, el usuario debe seguir una serie de pasos. En primer lugar, seleccionará la estación de inicio de la ruta, en este ejemplo Albacete, la cual se marcará en verde. Automáticamente, se abrirá un cuadro emergente en el mapa que muestra las coordenadas de dicha estación, y se actualizará el panel de la izquierda, como se muestra en la **Ilustración 20**. Después de hacer clic en "Next", el usuario pasará al segundo paso, que consiste en seleccionar la estación final de la ruta, en este caso de estudio es Murcia. Esta estación se marcará en rojo y se obtendrá

la misma respuesta que en el caso de la estación de inicio, mostrando el cuadro emergente y actualizando el panel de la izquierda, tal como se ilustra en la **Ilustración 21**.

En el último paso, se pueden observar dos secciones. La primera sección corresponde a la configuración, donde se encuentran los parámetros que el algoritmo tendrá en cuenta para el cálculo de la ruta. El usuario puede ajustar los valores de ponderación para determinar en qué medida se deben considerar dichos parámetros en el cálculo de la ruta. La sección inferior muestra un resumen de las estaciones seleccionadas como podemos observar en la **Ilustración 22**.

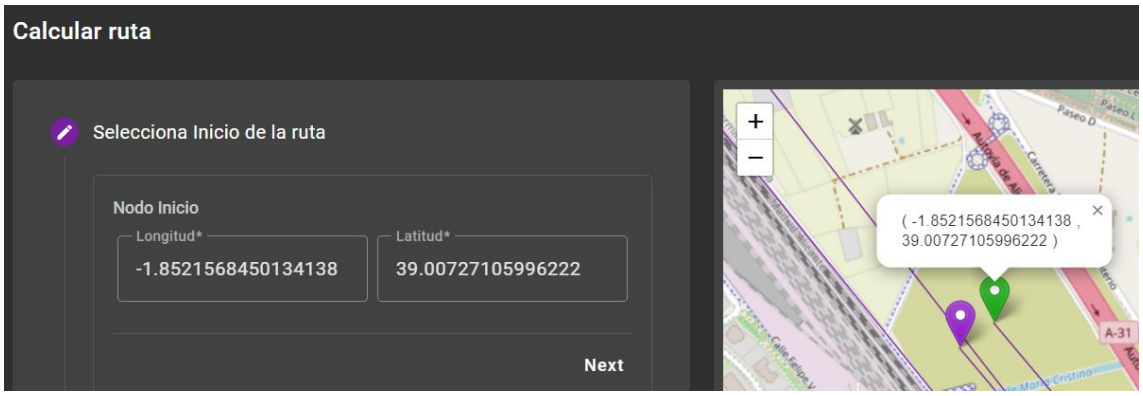


Ilustración 20. Selección del inicio de la ruta, Albacete.

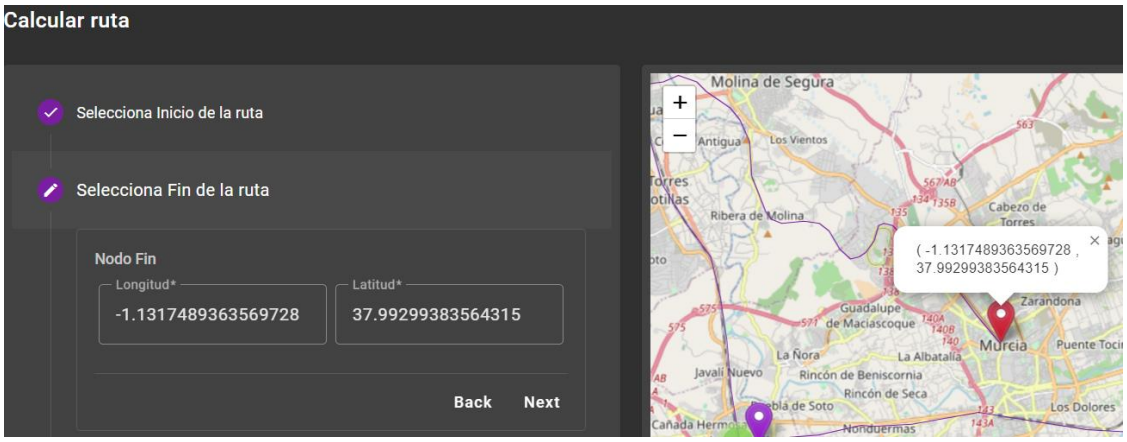


Ilustración 21. Selección fin de la ruta, Murcia.

Al finalizar, se ofrecen varias opciones. Si el usuario no está de acuerdo con los datos ingresados, puede hacer clic en el botón "Reset" para borrar todos los campos. Por otro lado, si los datos son correctos, se puede calcular la ruta pulsando el botón correspondiente (el botón de "Calcular ruta") el mapa mostrará la ruta óptima, como se puede apreciar en la **Ilustración 23**. La ruta se compone de tramos de diferentes colores, los cuales indican bifurcaciones y decisiones tomadas durante el trayecto.

En relación a la sección de configuración, es importante destacar que actualmente no está funcional y requiere ser validada por el cliente. En el segundo paso, donde se selecciona la estación final de la ruta, así como en el último paso de configuración y resumen, el usuario puede regresar al paso anterior y realizar cambios, es decir, modificar la estación inicial y final. Estos cambios se reflejarán en el mapa ya que se actualizarán los marcadores.

Es importante recordar que el alcance, en cuanto al cálculo de rutas, en este TFG, se limita a este caso de estudio, en específico, a calcular la ruta entre Albacete y Murcia. En consecuencia, al pulsar el botón "Calcular ruta", en lo referente a la interfaz se mostrará la ruta dibujada y que se puede observar en la **Ilustración 23**. En el lado del servidor, se determina la ruta óptima. Los tramos que se encuentran representados en la **Ilustración 25** mediante una línea discontinua azul, en concreto los puntos que pertenecen a estos tramos, son los que se presentan al algoritmo para su procesamiento y búsqueda de la ruta más eficiente, teniendo en cuenta en exclusiva esos puntos.

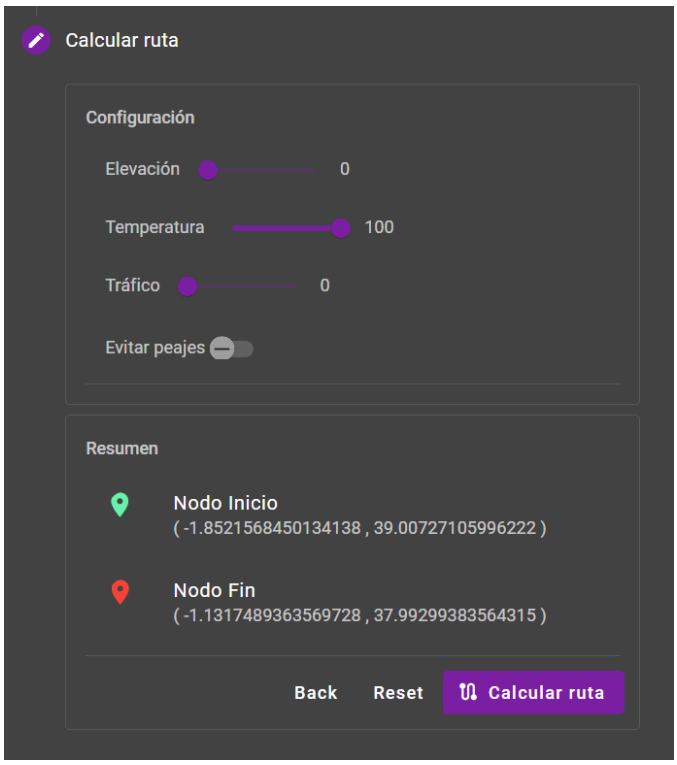


Ilustración 22. Configuración y Resumen.

Control de errores

El sistema cuenta con un control para verificar si el usuario no ha seleccionado las estaciones de inicio o fin de la ruta. En caso de que estos campos no estén completados y el usuario decidiera pulsar a “Next”, se mostrará un mensaje indicando que son campos requeridos, esto se puede ver en la **Ilustración 24**, el cambio de color de las letras a rojo y la sustitución de los iconos por signos de admiración. Si aun así el usuario pulsara el botón de “Calcular ruta”, el sistema no hace ningún cálculo y se le seguiría notificación que los campos son requeridos.

Por último, si el usuario hace clic en los campos de entrada de texto y luego no selecciona una estación en el mapa, el sistema también lo notificará marcando en rojo el campo de entrada correspondiente, indicando así que no se ha seleccionado una estación. Se ha deshabilitado el campo de entrada de coordenadas para evitar que el usuario tenga que introducir manualmente las coordenadas, lo cual sería un proceso tedioso. En su lugar, se indica al usuario que la forma de seleccionar las estaciones de inicio y fin de la ruta es interactuando directamente con el mapa.

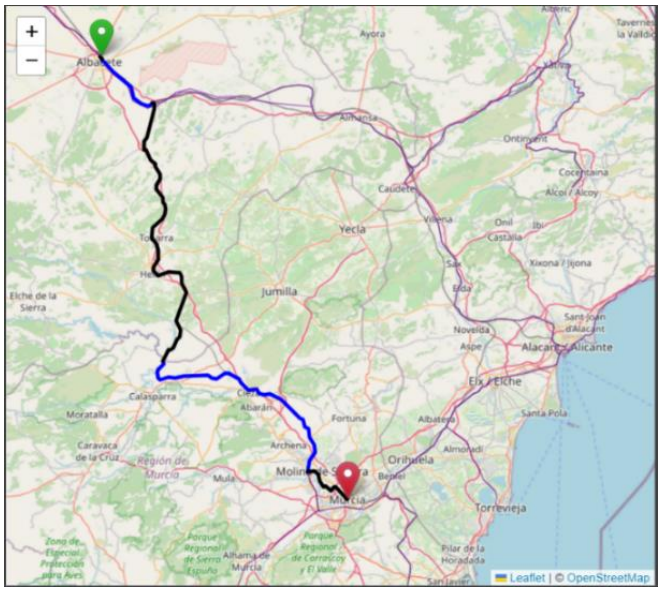


Ilustración 23. Resultado de la ruta Albacete-Murcia.

Ilustración 24. Aviso de campos requeridos.

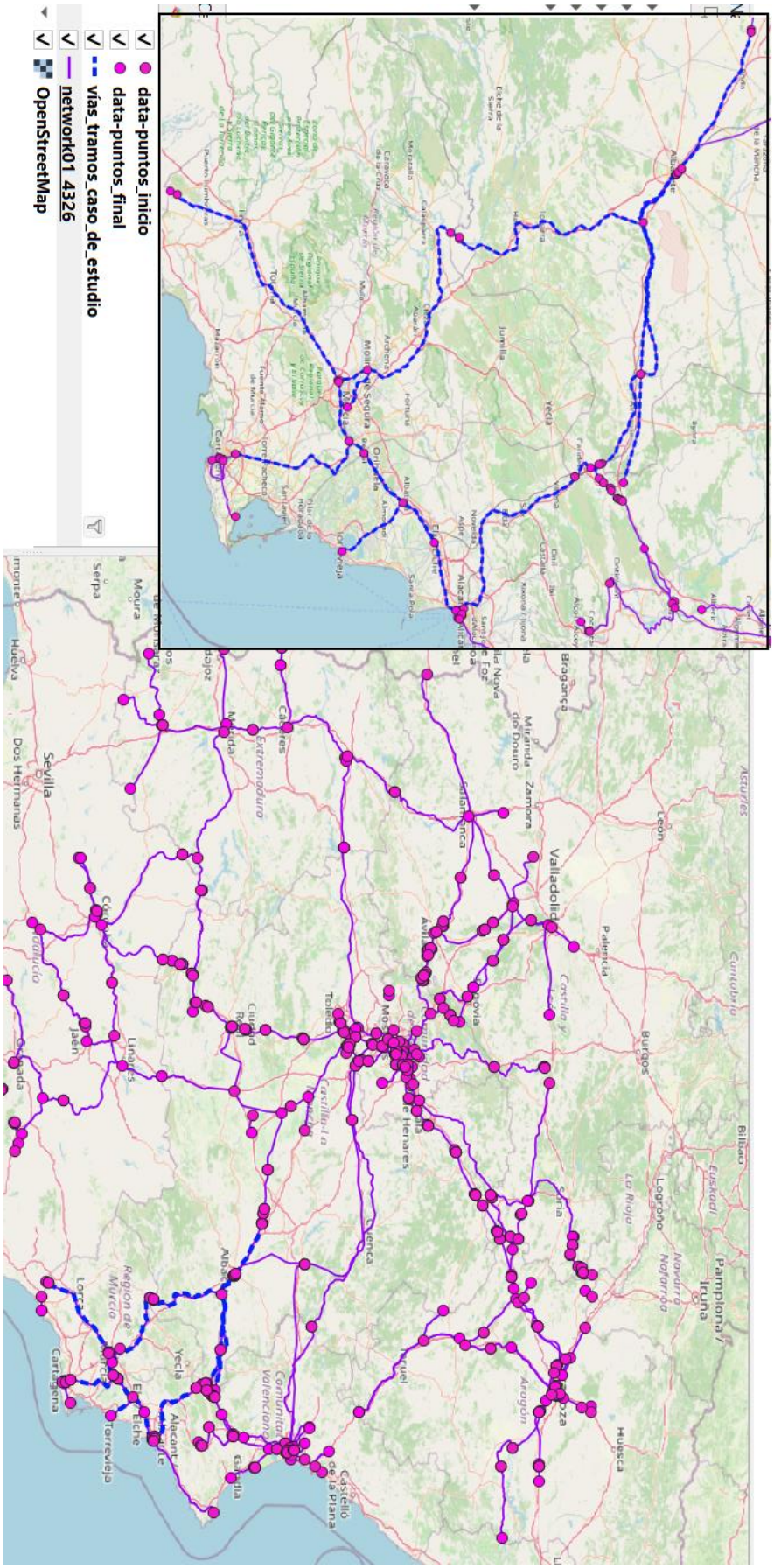


Ilustración 25. Tramos implicados en el caso de estudio

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo, se presentarán las conclusiones derivadas de la realización de este Trabajo de Fin de Grado, así como los conocimientos adquiridos durante el desarrollo de la aplicación. Además, se abordarán las posibles mejoras que podrían ser implementadas en el sistema, considerando la continuidad del desarrollo a través del proyecto de investigación: "*PROGRAMA Precipitation Data Systems de referencia PDC2022-133834-C21 financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea "NextGeneration EU/PRTR con referencia PDC2022-133834-C21".*

6.1 Conclusiones

En el marco de este trabajo, se lograron cumplir los objetivos establecidos, demostrando un progreso a través del proceso iterativo llevado a cabo. A lo largo del desarrollo de este proyecto, han surgido múltiples desafíos relacionados con el estudio del problema y la recopilación de requisitos. Sin embargo, se lograron superar estos obstáculos mediante la adaptación de los sprints en términos de tiempo y número de eventos, lo que permitió resolver los impedimentos surgidos durante la etapa inicial. En cuanto a los sprints orientados a la implementación, también se presentaron desafíos significativos. Entre ellos, se encontraron dificultades para construir un grafo conexo, así como el reto de comenzar desde cero, lo cual implicó la construcción de la arquitectura del sistema asegurando que todos los elementos estuvieran interconectados de manera

adecuada. A pesar de estas dificultades, se logró alcanzar el objetivo de implementar el algoritmo de Dijkstra y calcular la ruta en el caso de estudio específico entre Albacete y Murcia.

Alcanzar los objetivos de esteTFG demuestra la capacidad del equipo para superar impedimentos y cumplir con los requerimientos funcionales del proyecto.

6.2 Competencias

Durante el proceso de desarrollo de este TFG, se pudo poner en práctica una serie de competencias adquiridas a lo largo de la formación académica. Estas competencias incluyen la capacidad para evaluar las necesidades del cliente y especificar los requisitos del software, resolver problemas de integración utilizando estrategias y estándares disponibles, y diseñar, desarrollar, implementar, verificar y documentar soluciones software basadas en conocimientos teóricos y técnicas actuales. Estas compentencias son las siguientes:

- [IS2] *Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones*.
- [IS3] *Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles*.
- [IS4] *Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales* (Escuela Superior de Ingeniería Informática, 2023).

Además, se adquirieron otras competencias durante el propio proceso del TFG, tales como:

- [CO6] *Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos*. Esta competencia se ha fortalecido a través del análisis y diseño de algoritmos aplicados en el proyecto, evaluando su adecuación y complejidad para resolver los problemas planteados.

- [CO8] *Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados*. Esta competencia se ha adquirido principalmente en el diseño e implementación de la arquitectura del sistema. Durante el desarrollo del proyecto, se ha demostrado la habilidad para analizar, diseñar y construir aplicaciones con un enfoque en la robustez, seguridad y eficiencia. Se han seleccionado los paradigmas y lenguajes de programación más adecuados para asegurar la calidad del sistema.
- [CO12] *Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos*. Esta competencia se ha adquirido fundamentalmente ya que durante todo el proceso de implementación se han llevado a cabo buenas prácticas, teniendo en cuenta la modularidad y la escalabilidad. El manejo adecuado de las bases de datos ha sido fundamental en el desarrollo del proyecto, comprendiendo y aplicando las características y funcionalidades que permiten un uso adecuado de ellas. Además, se ha tenido en cuenta el diseño, análisis e implementación de aplicaciones basadas en bases de datos, siguiendo buenas prácticas que garantizan la modularidad y escalabilidad del sistema.
- [CO14] *Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.* Se ha adquirido la competencia en programación paralela, concurrente, distribuida y de tiempo real, comprendiendo los principios y técnicas fundamentales para su implementación.
- [CO16] *Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software*. Esta competencia se ha adquirido durante el desarrollo del proyecto,ya que se ha aplicado una metodología ágil adaptada a las particularidades del proyecto.

6.3 Trabajo futuro

El presente TFG ha abordado desafíos significativos en su desarrollo, los cuales han dejado abiertas diversas líneas de trabajo futuro. Debido al alcance complejo del proyecto, la definición de distintos tipos de escenarios y los criterios necesarios para la consecutividad entre los puntos y los tramos no pudieron ser establecidos. Esta limitación representa una oportunidad para investigaciones futuras que permitan abordar este desafío de manera más completa y precisa. Además, se destaca la importancia de mantener reuniones con el cliente con el objetivo de definir los arquetipos de usuario final que podrían utilizar la aplicación. Estas reuniones proporcionan una valiosa información para llevar a cabo pruebas de testing que contribuyan al progreso y mejora continua de la aplicación.

Cabe mencionar que este TFG forma parte de un proyecto de investigación más amplio, lo que proporciona mayores posibilidades para la consecución de líneas de trabajo futuro. En este sentido, se proponen las siguientes:

i. Añadir una interfaz que contenga un modelo 2D, lo cual permitirá mejorar la visualización y comprensión de la información en la aplicación.
ii. Incorporar una interfaz que contenga un modelo 3D, lo que proporcionará una representación más realista y enriquecedora de los datos.

Estas líneas de trabajo futuro se presentan como oportunidades de gran valor para ampliar y mejorar el proyecto, en términos de funcionalidad y experiencia de usuario. Esto permitirá obtener resultados más sólidos y avanzados en el ámbito del proyecto.

El código desarrollado y ésta misma memoria se puede encontrar en el siguiente enlace:
https://github.com/HenarVGalan/TFGhenar2223/commits?author=HenarVGalan

Bibliografía

Esri. (2021a). *Bandas de ráster*. Recuperado, el 20 de junio de 2023, de: https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/raster-bands.htm

Esri. (2021b). *¿Qué son los datos ráster?* Recuperado, el 20 de junio de 2023, de: https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm

Esri. (2021c). *¿Qué es un shapefile?* Recuperado, el 20 de junio de 2023, de: https://desktop.arcgis.com/es/arcmap/latest/manage-data/shapefiles/what-is-a-shapefile.htm

Esri. (2021d). *Archivos ráster BIL, BIP y BSQ*. Recuperado, el 20 de junio de 2023, de: https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/bil-bip-and-bsq-raster-files.htm

QGIS project. (2002). *Una Introducción fácil a GIS*. Recuperado, el 20 de junio de 2023, de: https://docs.qgis.org/3.28/es/docs/gentle_gis_introduction/index.html

Brassard, G., & Bratley, P. (1996). *Fundamentals of algorithmics*. Prentice-Hall, Inc..

Schwaber, K., & Sutherland, J. (2021). *The Scrum Guide*. 2020.

Scrum.org (2020). *Framework Scrum*. Recuperado, el 25 de junio de 2023, de: https://www.scrum.org/learning-series/what-is-scrum

PostGIS Project (2023). *PostGIS 3.3.4dev Manual*. Recuperado, el 25 de junio de 2023, de: http://postgis.net

Esri. (s.f). *Tipos de datos de PostgreSQL admitidos en ArcGIS*. Recuperado, el 10 de julio de 2023, de: https://pro.arcgis.com/es/pro-

app/latest/help/data/geodatabases/manage-postgresql/data-types-postgresql.htm

Escuela Superior de Ingeniería Informática. (2023). *Objetivos y Competencias específicas*. Recuperado, el 20 de junio de 2023, de: https://www.esiiab.uclm.es/grado/objetivos.php

Anexo I. Proceso de desarrollo

En este anexo se presenta el proceso de desarrollo diseñado para alcanzar los objetivos establecidos en este Trabajo de Fin de Grado. A lo largo de este anexo, se describirá de manera detallada qué historias de usuario y tareas se han llevado a cabo en cada uno de los sprints. En la explicación de cada sprint, se incluirá en primer lugar una breve descripción sobre la planificación del sprint, que en el contexto de este proyecto consiste en establecer la fecha de inicio y finalización, su objetivo específico, así como la selección de historias de usuario y tareas elegidas por el equipo para ser completadas durante el sprint. Una vez planificado el sprint, se procederá a describir el desarrollo del mismo, destacando las tareas más relevantes y cualquier reunión o evento significativo que haya tenido lugar. Por último, se abordará la revisión del sprint y la reunión retrospectiva, donde se evaluarán los resultados obtenidos y se reflexionará sobre posibles mejoras para el próximo sprint.

I.1 Sprint 1

El primer sprint se establece como la fase inicial de puesta en marcha, donde los tutores plantean el tema del TFG y proporcionan el contexto necesario para el desarrollo del proyecto. Durante este sprint, se establecen las bases y se define la dirección a seguir en el desarrollo del trabajo.

I.1.1. Planificación del Sprint 1

En la planificación del sprint, se establecen las fechas de inicio y finalización del Sprint 1, así como su objetivo específico, como se detalla en la **Tabla 2** correspondiente.

Además, se presentan las historias de usuario que serán desarrolladas durante este primer sprint, como se puede observar en la **Ilustración 26**.

Tabla 2. Información sobre el Sprint 1.

Fecha Inicio	Fecha Fin
03/02/2023	14/02/2023
Objetivo del Sprint 1	
Estudio de conceptos y herramientas pertinentes en el ámbito del problema.	

Sprint 1

Historia de usuario	Id. de eleme...	T.Estimado	Prioridad
Planificación	1225428418	3600 s	Medium
Estudio de conceptos 3	1225428417	3600 s	Medium
Estudio de herramientas	1225636663	14.400 s	Medium
Prueba con leaflet 2	1225579771	3000 s	Low
Elección de herramientas	1225610501	1800 s	Low
Recogida de requisitos	1225611603	7200 s	Low
Retrospectiva	1225428419	3600 s	Medium

Ilustración 26. *Sprint Backlog* del Sprint 1.

I.1.2. Desarrollo del Sprint 1

El sprint se inicia con una reunión conjunta entre el equipo y el cliente, donde este último expone sus expectativas respecto a la aplicación. Durante la reunión, el cliente transmite los siguientes puntos clave:

- Desea que la aplicación cuente con un mapa que refleje puntos geográficos, los cuales podrán ser seleccionados como inicio y fin de una ruta. Estos puntos son de naturaleza geográfica, no topológica.
- Requiere que la aplicación pueda manejar archivos ráster y que estos sean considerados en el cálculo de la ruta.
- Expresa la necesidad de contar con tres pantallas con diferentes interacciones, denominadas "modo básico", "modo intermedio" y "modo avanzado", sin proporcionar más detalles al respecto.

La reunión concluye con el acuerdo de que el cliente proporcionará recursos a través de correo electrónico para iniciar el estudio y análisis correspondientes.

Posteriormente, el equipo realiza una reunión para planificar el sprint, estableciendo su objetivo principal: familiarizarse con las herramientas SIG, los tipos de datos y los tecnicismos relacionados. En los días siguientes, se aborda la historia de usuario titulada "Estudio de conceptos", se divide en tareas específicas, las cuales incluyen el estudio de los conceptos relacionados con los datos ráster, el algoritmo Dijkstra y los SIG.

Durante el desarrollo de este sprint, el 6 de febrero, se realiza una reunión de *Daily Scrum* con la asistencia de José Antonio Mateo y Henar Velasco. En esta reunión, se establecen como tareas prioritarias para los próximos días, aquellas que se centran en la ejecución de un pequeño proyecto de prueba que incluye la inserción de un mapa de ejemplo utilizando Leaflet, con el objetivo de familiarizarse con esta tecnología. Al finalizar el sprint, se realiza una recopilación de requisitos basada en la reunión con el cliente, lo cual resulta en un diagrama de actividad, veáse en la Ilustración 28, que representa la interacción del usuario con los principales elementos solicitados por el cliente.

I.1.3. Revisión del Sprint 1

Se presenta el proyecto de prueba en el que se pone en práctica la inserción de un mapa con Leaflet. Y por último se actualiza el *Product Backlog*, veáse en la Ilustración 27.

I.1.4. Retrospectiva del Sprint 1

La reunión retrospectiva se resume en la Tabla 3.

Tabla 3. Retrospectiva del Sprint 1.

¿Qué estuvo bien?	¿Qué se puede mejorar?	Acciones para mejorar
Se entienden los conceptos teóricos.	Problemas con la comprensión de QGIS.	Añadir tareas al *PB* dedicadas a practicar con QGIS.

Product Backlog

Historia de usuario	Id. de eleme...	T.Estimado	Prioridad	Estado
Crear proyecto 4	1226003966	28.800 s	High	En cola

Subelemento	Id. de eleme...	Prioridad	Estado	T.Estimac
Servidor	1226004713	High	En cola	
API Rest	1226004756	High	En cola	
BBDD	1226004793	High	En cola	
Cliente	1226005626			
+ Agregar Subelemento				

Pruebas con QGIS	1226005866	7200 s	Low	En cola

Ilustración 27. *Product Backlog* al terminar Sprint 1.

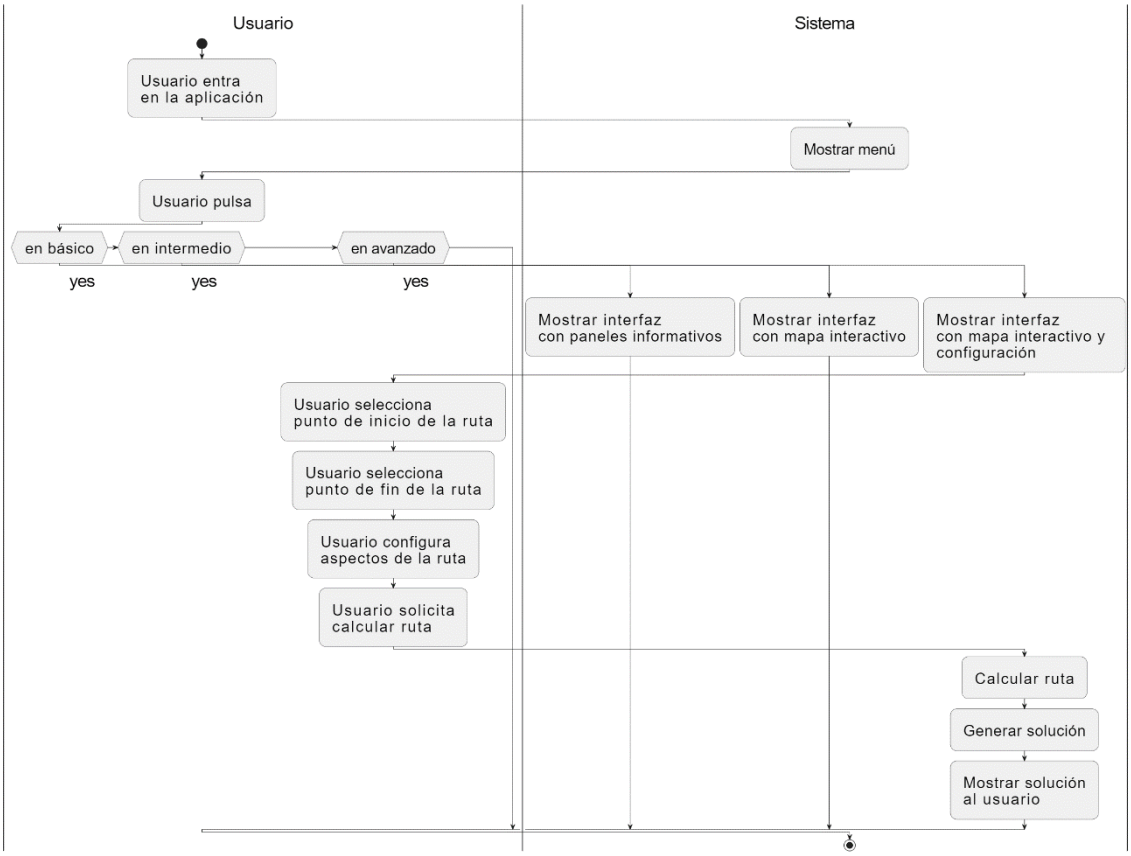


Ilustración 28. Diagrama de actividad.

I.2 Sprint 2

En el segundo sprint, se aborda de manera más sustancial el trabajo con los ficheros suministrados por el cliente, los cuales contienen datos relevantes sobre el dominio del problema. Durante el desarrollo de este sprint, se detallarán los tipos de datos proporcionados y se identificarán los problemas surgidos en relación a estos ficheros.

I.2.1. Planificación del Sprint 2

El sprint actual, según se indica en la **Tabla 4**, presenta una duración más prolongada y tiene como objetivo principal llevar a cabo un exhaustivo estudio y tratamiento de los ficheros suministrados. Para ello, las tareas asignadas para este sprint las podemos ver en la **Ilustración 29**.

Tabla 4. Información sobre el Sprint 2.

Fecha Inicio	Fecha Fin
06/03/2023	14/04/2023
Objetivo del Sprint 2	
Estudio y análisis de los ficheros.	

I.2.2. Desarrollo del Sprint 2

El objetivo principal del sprint, como se mencionó previamente, consiste en el estudio de los ficheros proporcionados por el cliente. Estos ficheros se encuentran organizados en dos carpetas distintas. En una de ellas se encuentra el fichero denominado "network01.shp", mientras que en la otra carpeta se encuentran los ficheros ráster con extensiones .bil y .hdr. La primera tarea llevada a cabo fue la lectura de estos ficheros, lo cual presentó un desafío inicial al intentar leer los ficheros de extensión .bil y .hdr utilizando la herramienta ArcGis. Se detectó que estos datos carecían de un sistema de referencia adecuado. Aunque se intentó forzar ambos ficheros a utilizar el mismo sistema de referencia, no se logró ubicarlos en el mismo lugar, lo que resultó en una superposición incorrecta de los datos. Durante el proceso de estudio, se identificó que el fichero "network01" contenía datos sobre las vías ferroviarias en España, mientras que los ficheros ráster proporcionaban información sobre la elevación del terreno. Se logró superponer correctamente ambos ficheros, utilizando la herramienta QGis y se seleccionó el sistema de referencia "EPSG:4326 WGS 84".

Sprint 2

Historia de usuario	Id. de eleme...	T.Estimado	Prioridad
Planificación	1225993225	3600 s	Low
Lectura de ficheros	1226038214	7200 s	Critical
Recogida de requisitos 2	1226429923	7200 s	Low

Subelemento	Id. de eleme...	Prioridad	Estado
Estudio de los recursos	1226430089	High	
Boceto de un prototipo	1226430888	Low	
+ Agregar Subelemento			

Retrospectiva	1225993311	3600 s	Low

Ilustración 29. *Sprint Backlog* del Sprint 2.

Además de los ficheros, el cliente proporcionó un documento explicativo que sirve como base de trabajo para realizar un primer boceto del proyecto. En este documento se describen los siguientes puntos relevantes:

i. El usuario selecciona dos nodos del fichero "network01.shp", que es un fichero variable pero siempre ubicado en el mismo directorio. La selección de nodos puede realizarse haciendo clic en dos puntos en un mapa o mediante un desplegable que muestra el código de cada nodo.

ii. El algoritmo "modelo1D" calcula la ruta más rápida entre esos dos nodos teniendo en cuenta los siguientes factores:

   a) La distancia entre los nodos según el fichero "network01.shp". Este factor tiene un peso predefinido (vs. punto 3) de w00.

   b) La integral de camino entre los puntos según los valores del fichero "raster01.bil", considerando el valor mínimo de los caminos posibles (los que se encuentran en "network01.shp", ya que solo se puede transitar por ellos). Este factor se pondera con el valor w01 en el algoritmo "modelo1D".

   c) La integral de camino entre los puntos según los valores del fichero "raster02.bil", también ponderada con el valor w02 en el algoritmo "modelo1D".

d) Este proceso se repite para todos los ficheros con extensión .bil que se encuentren en el directorio "raster". Pueden existir varios ficheros, incluso más de diez.

iii. Inicialmente, se asignan los siguientes valores a los pesos:

a) w00 = 1.0, w01 = w02 = w03 = ... $wx_1x_2 = 0.0$. A este esquema se le denomina "prueba 0".

b) w00 = 0.5, w01 = w02 = w03 = ... $wx_1x_2 = 0.125$. A este esquema se le denomina "prueba 1".

La definición de los pesos se realiza en un fichero de texto llamado "weights.txt" que se ubica en el directorio "./1D_demo".

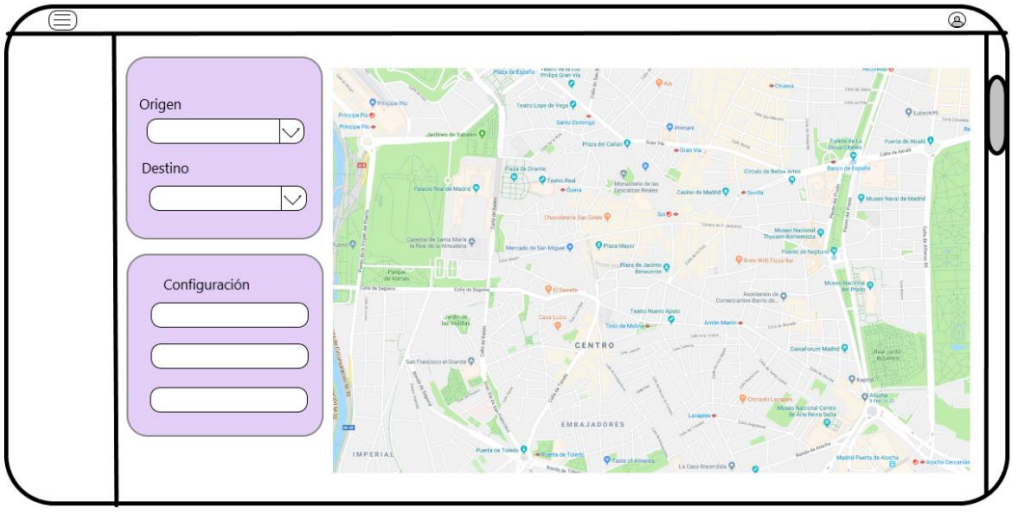


Ilustración 30. Primer boceto de la aplicación.

I.2.3. Revisión del Sprint 2

Primera representación a modo de boceto del Modo Avanzando, donde se podrá hacer la consulta por parte del usuario de una ruta, vease en la **Ilustración 30**.

I.2.4. Retrospectiva del Sprint 2

En la **Tabla 5** se puede ver plasmada de forma resumida la retrospectiva de este sprint.

Tabla 5. Retrospectiva del Sprint 2.

¿Qué estuvo bien?	¿Qué se puede mejorar?	Acciones para mejorar
Se consigue la lectura de los ficheros.	Prototipo.	Crear un prototipo más completo.

I.3 Sprint 3

En este sprint, se inicia la construcción de la arquitectura del sistema. El equipo toma la decisión de agregar nuevas historias de usuario al *Product Backlog*, las cuales serán seleccionadas para su implementación durante este sprint.

I.3.1. Planificación del Sprint 3

Con el fin de alcanzar el objetivo establecido para este sprint, como se indica en la Tabla 6, se realizarán tareas que se centran en la instalación y configuración de las herramientas adecuadas. Además, se otorgará especial importancia a las tareas relacionadas con el prototipado, ya que se busca validar posteriormente con el cliente.

Tabla 6. Información sobre el Sprint 3.

Fecha Inicio	Fecha Fin
22/03/2023	31/0/2023
Objetivo del Sprint 2	
Desarrollo de un prototipo de la interfaz Modo Avanzado y puesta en marcha de la base de datos .	

Sprint 3

	Historia de usuario		T.Estimado	Prioridad
	Planificación		3600 s	Low
	Instalar Pgadmin / PostGis 1		3600 s	Low
	Crear bbdd 2		3600 s	Low
	Estudio de herramientas de prototipado		3600 s	Low
	Primer Prototipo v0	1	18.000 s	Low
	Hacer formulario para una entrevista		1800 s	Low
	Retrospectiva		1800 s	Low

Ilustración 31. *Sprint Backlog* del Sprint 3.

I.3.2. Desarrollo del Sprint 3

Durante el desarrollo de este sprint, se han realizado las tareas de instalación y configuración de pgAdmin, estableciendo así una base de datos en PostgreSQL junto con la extensión PostGIS, la cual permite el almacenamiento, consulta y manipulación de datos geoespaciales dentro de un entorno de base de datos relacional. En este contexto,

se ha importado el fichero que contiene los datos sobre las vías ferroviarias mencionadas en el sprint anterior. Al importar dicho fichero a la base de datos, se ha creado una entidad con el mismo nombre del fichero, veáse en la **Ilustración 32**. Al analizar estos datos en la base de datos, se ha observado que las vías se encuentran fragmentadas en tramos, tal y como se interpretó previamente en QGis.

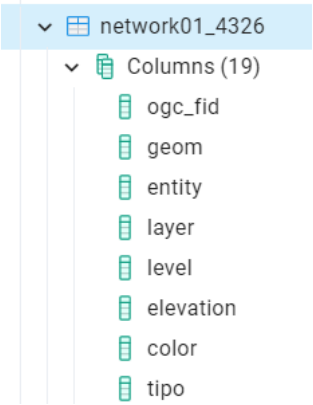


Ilustración 32. Entidad resultado de la importación.

Con el propósito de asegurar una terminología unificada, se ha preparado un formulario en forma de entrevista para el cliente. Esto permitirá realizar consultas sobre la descripción de las columnas presentes en la tabla, de modo que todo el equipo, incluido el cliente, utilicen un glosario común y compartan las mismas definiciones.

Además, se ha llevado a cabo una búsqueda exhaustiva de herramientas de prototipado con el objetivo de encontrar la más adecuada para reflejar la interacción del usuario con el sistema. Se consideró importante que el prototipo capturara el estilo y diseño que se implementaría, para posteriormente validar estas interfaces con el cliente. Como resultado de esta búsqueda, se optó por utilizar la herramienta Figma para el desarrollo del prototipo.

Durante el transcurso de este sprint, se presenta un breve paréntesis debido a las vacaciones de Semana Santa. El sprint se reanuda el día 12 de abril, con la celebración de la reunión diaria de Scrum. Los asistentes a esta reunión son José Antonio Mateo y Henar Velasco, y su objetivo principal es la presentación del prototipo desarrollado.

El día 13, se lleva a cabo una reunión de todo el equipo, encabezada por Enrique Arias. Durante esta reunión, se reorganizan las tareas y se establece como prioridad la siguiente tarea: realizar una operación sobre el fichero ráster de elevación, la cual consiste en agrupar las celdas que poseen el mismo valor. Como resultado de esta

operación, se generan polígonos más pequeños. Una vez completada esta tarea, se procederá a realizar otra operación entre el fichero de las vías del ferrocarril y esta capa de polígonos. La operación consiste en determinar la intersección entre ambas capas, con el fin de identificar qué tramo pasa por qué polígonos. Esta información se almacenará en una columna de la tabla que almacena los tramos, con el propósito de registrar los polígonos por los cuales transcurre cada tramo y el valor correspondiente del ráster, que en este caso representa la elevación del terreno.

I.3.3. Revisión del Sprint 3

Tras una evaluación del formulario, en forma de entrevista para el cliente, propuesto, el equipo ha llegado a la conclusión de que contiene un número escaso de preguntas. Por lo tanto, se ha decidido prorrogar el envío del formulario al cliente con el fin de acumular más preguntas por el equipo. Esta medida permitirá obtener una mayor claridad en cuanto a la descripción de las columnas y evitar la sobresaturación del cliente con diferentes demandas. Por lo tanto, esta tarea vuelve al *PB*. Cuando esta tarea sea completada, el formulario se enviará al cliente para su revisión y respuesta.

Finalmente, el equipo procede a revisar detalladamente el prototipo desarrollado[8] y toma la decisión de presentarlo al cliente durante el próximo sprint. Se considera que el prototipo cumple con los requisitos establecidos y se encuentra en un estado apropiado para ser presentado.

I.3.4. Retrospectiva del Sprint 3

En la Tabla 7 se puede observar la retrospectiva de este sprint de forma resumida.

Tabla 7. Retrospectiva del Sprint 3.

¿Qué estuvo bien?	¿Qué se puede mejorar?	Acciones para mejorar
Prototipo.	Entrevista al cliente.	Preparar entrevista para la presentación del prototipo.

[8] La versión del prototipo presentado en la revisión del Sprint 3 se puede encontrar en: https://www.figma.com/proto/yl3CO0HEpxJOejK3S9G6C/1dpath_v0?node-id=1351-286&scaling=scale-down&page-id=1351%3A280&starting-point-node-id=1351%3A286&show-proto-sidebar=1

I.4 Sprint 4

Este sprint se distingue por el incremento del porcentaje de tareas dedicadas a la implementación de código. En los sprints anteriores se recopilaron requisitos y se llevó a cabo un estudio exhaustivo del problema y su contexto. Sin embargo, en este sprint se da inicio a la fase de implementación del código, tal como se detallará en la planificación y desarrollo correspondiente a este periodo.

I.4.1. Planificación del Sprint 4

El sprint acordado tendrá una duración de 12 días y su principal objetivo será presentar el prototipo desarrollado en el sprint anterior al cliente. Para lograr esto, se preparará una serie de preguntas que se utilizarán durante la presentación del prototipo. Estas preguntas servirán para confirmar y garantizar que los requisitos se estén capturando de manera precisa.

La segunda parte de este sprint se enfocará en la implementación de una arquitectura sencilla pero completa, como se muestra en la Ilustración 33. Se pretende que esta arquitectura incluya un servidor, un frontend básico, una conexión a la base de datos y, por último, una API Rest. El objetivo es establecer todos los componentes necesarios para el funcionamiento del sistema.

Tabla 8. Información sobre el Sprint 4.

Fecha Inicio	Fecha Fin
17/04/2023	28/04/2023
Objetivo del Sprint 4	
Validar Prototipo v0 y construcción de la arquitectura del proyecto.	

I.4.2. Desarrollo del Sprint 4

El 18 de abril tuvo lugar la presentación del prototipo ante el cliente, y se tomó la precaución de prepararlo de forma adecuada en caso de que el cliente expresara su interés en obtenerlo en lugar de simplemente verlo durante la presentación. Dado que se tenía la preocupación de que el cliente no se sintiera cómodo interactuando con Figma, se preparó un video interactivo en InVision que mostraba las pantallas creadas en Figma. El prototipo presentado se encuentra disponible para su consulta en el siguiente enlace:

- Prototipo en Figma: https://www.figma.com/proto/yl3CO0HEpjxJOejK3S9G6C/1dpath_v0?type=design&node-id=1351-286&t=OIlBJUvyNAkkUdCu-1&scaling=scale-down&page-id=1351%3A280&starting-point-node-id=1351%3A286&show-proto-sidebar=1&mode=design
- Prototipo en InVision: https://henarprototype.invisionapp.com/overview/1dpath-clgcl6yx103ey0188fr5nc3rg/screens?share=&sortBy=1&sortOrder=1&viewLayout=2

El objetivo principal de este prototipo es reflejar cómo un usuario puede determinar la ruta entre un punto de inicio y uno de destino. Durante la reunión, el cliente realizó las siguientes observaciones: expresó su deseo de poder configurar las capas, manifestó que no desea que el mapa muestre todos los marcadores para evitar una densidad excesiva, y enfatizó la importancia de la interacción "zoom in, zoom out" para desplazarse por el mapa y visualizar las diferentes estaciones. La reunión concluyó con un feedback positivo por parte del cliente, aunque sin proporcionar detalles adicionales ni realizar más solicitudes.

Después de la presentación, el equipo se reunió con el objetivo de reorganizar las historias de usuario para este sprint y seleccionar las más prioritarias. Como resultado de esta reunión, se acordó añadir una tarea al *PB* que consiste en determinar cómo se puede identificar, a partir de los datos disponibles, si un tramo es consecutivo de otro.

Durante el resto del sprint, se llevó a cabo la construcción de la arquitectura. Para ello, se crearon carpetas separadas para el frontend, donde se alojaría todo lo relacionado con la interfaz de usuario, y una carpeta para el servidor, que contendría los archivos relacionados con la construcción e implementación del servidor utilizando el framework Express.js. Antes de llevar a cabo la implementación de la arquitectura, se realizó un breve estudio sobre el concepto de geometría, el cual adquiere especial relevancia a partir de este punto. En PostGIS, una geometría es un tipo de dato que representa objetos espaciales en un sistema de coordenadas. Estos objetos pueden ser puntos, líneas o polígonos, y se utilizan para almacenar y manipular datos geoespaciales en una base de datos PostgreSQL. PostGIS admite varios tipos de geometría que se clasifican en tres categorías principales: puntos, líneas y polígonos. Además de estos tipos

básicos, también se ofrecen tipos más complejos, como conjuntos de puntos, líneas y polígonos, permitiendo representar colecciones de objetos geométricos en una única entidad (Esri, s.f).

Una vez finalizado el estudio de este concepto, se procedió a la implementación, cuyo objetivo era establecer la conexión entre la base de datos y el servidor. Se creó una pequeña API Rest que incluía consultas a la base de datos para obtener los identificadores y geometrías de todos los tramos. Se desarrolló un método para obtener el punto central de cada tramo y se almacenó en la tabla "tramo" previamente conocida como "network". A partir de este punto central, se obtuvo la geometría. Además, se implementaron métodos para convertir estas geometrías a coordenadas y para obtener el punto inicial y final de cada tramo. Todos estos métodos se encuentran accesibles a través de peticiones a la API. El equipo decidió adoptar esta estructura pensando en posibles necesidades futuras en el frontend.

Para finalizar el desarrollo de este sprint, se retomó la creación de un formulario en el que se consultó al cliente, llamado Glorario, con el objetivo de obtener información sobre si era posible utilizar alguna función establecida por QGIS para determinar si un tramo era consecutivo de otro. Esta información era crítica para la construcción del grafo necesario para aplicar el algoritmo de Dijkstra, ya que conocer la secuencia de los nodos era fundamental.

Sprint 4

Historia de usuario	Id. de eleme...	T.Estimado
Planificación	1226556096	3600 s
Preparar Entrevista.	1226556110	1800 s
Retrospectiva	1226556098	1800 s
Crear proyecto 5	1226003966	28.800 s

Subelemento	Id. de eleme...	Prioridad	Estado
Servidor	1226004713	High	En cola
API Rest	1226004756	High	En cola
BBDD	1226004793	High	En cola
Cliente	1226005626	High	
Sistema organizativo de carpetas	1226556291		

Ilustración 33. *Sprint Backlog* del Sprint 4.

I.4.3. Revisión del Sprint 4

Construcción finalizada de la arquitectura del proyecto. Se ha establecido la conexión con la base de datos y se han realizado consultas para obtener los datos necesarios. En particular, se ha extraído información del fichero "network01" que contiene las vías del ferrocarril. Se ha calculado el punto central de cada tramo y se ha almacenado en una tabla dedicada llamada "PUNTO". Esta tabla contiene información relevante para la creación de rutas, como identificadores, geometría, coordenadas y valores asociados al ráster.

I.4.4. Retrospectiva del Sprint 4

Problemáticas encontradas: durante el desarrollo, se encontró una problemática relacionada con la función "ST_GeneratePoints" que no admite la geometría proporcionada (PostGIS Project, 2023, p. 364). Se realizaron intentos de transformar la geometría a un formato válido, pero el error persistió. Para abordar esta cuestión, resulta necesario investigar cómo se dibuja cada tramo y determinar si es posible subdividirlos. En caso de ser factible, se explorará la posibilidad de obtener puntos adicionales a partir de estos subtramos. Una posible vía de solución: se propone utilizar la función "ST_Points" como una alternativa para abordar esta problemática (PostGIS Project, 2023, p. 140). Esta función nos permitirá obtener la geometría de múltiples puntos que forman parte de cada tramo, incluyendo los puntos de inicio y fin. El objetivo principal es almacenar, para cada punto obtenido, la información correspondiente a las coordenadas, si se trata de un punto inicial y/o final, y el valor del raster asociado (dn raster), en este problema el atributo dn contien el valor de la elevación del terreno.

I.5 Sprint 5

El objetivo de este sprint es encontrar soluciones a los problemas identificados durante el sprint anterior.

I.5.1. Planificación del Sprint 5

Las historias de usuario que serán abordadas durante este sprint se encuentran detalladas en la **Ilustración 34**. En la **Tabla 9** se especifican las fechas de inicio y

finalización del sprint, así como el objetivo principal que se busca alcanzar durante este período de trabajo.

Tabla 9. Información sobre el Sprint 5.

Fecha Inicio	Fecha Fin
1/05/2023	10/05/2023
Objetivo del Sprint 5	
Estudio de los tramos y búsqueda de los consecutivos.	

Sprint 5

Historia de usuario	Id. de eleme...	T.Estimado	Prioridad
Planificación	1226756865	1800 s	Low
Retrospectiva	1226756914	1800 s	Low
Revisión del modelo	1226757801	1800 s	Low
Revisión de las respuestas del form	1226759200	3600 s	Medium
Estudio de la topografía 1	1226760240	14.400 s	Medium

Subelemento	Id. de eleme...	Prioridad	Estado	T.Estimado
Definir "consecutivo"	1226760855			

Ilustración 34. *Sprint Backlog* del Sprint 5.

I.5.2. Desarrollo del Sprint 5

El desarrollo de este sprint comienza con el análisis de las respuestas proporcionadas por Francisco Javier Tapiador en el formulario enviado durante el sprint anterior. Durante este análisis, se destacan varios aspectos importantes. En primer lugar, se establece un glosario común para unificar la terminología utilizada. Además, Francisco Javier Tapiador confirma que no existe una operación o herramienta en QGIS que facilite la identificación de puntos consecutivos. Por último, se comunica un requerimiento específico: para el cálculo de la ruta, no se desea vectorizar el raster, es decir, agrupar las celdas del raster que tienen el mismo valor. En su lugar, se prefiere trabajar directamente con el raster.

En consecuencia, el equipo se reúne y aborda los siguientes temas. Primero, se marca como historia de usuario crítica para el próximo sprint la lectura del raster y el procesamiento directo de los datos sin utilizar QGIS para su preprocesamiento. También se priorizan las tareas relacionadas con el estudio de la topografía como críticas para este sprint. Como parte del desarrollo, se realiza un estudio de los tramos y se llevan a cabo

pruebas utilizando operaciones como “ST_INTERSECT”[9] para identificar tramos cuyo punto final coincide con el inicio de otro, con el objetivo de encontrar tramos consecutivos. El resultado de estas pruebas muestra que son muy pocos los tramos que cumplen con esta condición. Cerca del final del sprint, se lleva a cabo una reunión, día 9 de mayo, conjunta entre todo el equipo y el cliente. El propósito de esta reunión es comunicar al cliente la problemática identificada. Se informa al cliente que, para aplicar el algoritmo de Dijkstra, es necesario contar con un grafo conexo. La retroalimentación recibida es que los tramos sueltos no representan un inconveniente y no es necesario unirlos con el resto, sino que deben mostrarse individualmente. Además, el cliente propone marcar nodos como punto de inicio y fin, aunque la ruta resultante pueda ser una "línea recta" o no lleve a ningún destino. Destaca que los datos proporcionados son de prueba y en un escenario real no se presentará esa situación. Asimismo, el cliente reitera la indicación de no utilizar una capa vectorizada y obtener los datos directamente del raster.

I.5.3. Reunión de final del Sprint 5

En el sprint 5, si bien no se llevan a cabo eventos de revisión y retrospectiva, se programa una reunión para finalizar el sprint. Durante esta reunión, en la que participa todo el equipo, se toman decisiones importantes con respecto al progreso del proyecto y se reordena el *Product Backlog*.

En vista de la complejidad añadida al intentar determinar qué puntos son consecutivos entre sí, se decide posponer esta tarea y abordarla en un momento posterior. En cambio, se priorizará la implementación del frontend como una de las principales tareas a desarrollar en este punto del proyecto. Además, se establece la necesidad de abordar la lectura de los ficheros raster como una tarea relevante para el próximo sprint. Esto implica trabajar en la comprensión y extracción de información de estos archivos, lo cual es considerado de suma importancia para el avance del proyecto.

[9] Esta función compara dos geometrías y devuelve *true* si se cruzan. Las geometrías se intersecan si tienen algún punto en común.

I.6 Sprint 6

Este sprint se caracteriza por tener un enfoque significativo en la implementación del frontend, siendo este el principal objetivo de un gran porcentaje de las tareas asignadas.

I.6.1. Planificación del Sprint 6

Todas las tareas de este sprint van dirigidas a cumplir el objetivo del mismo, todas ellas con una prioridad media-alta. Esta información queda recogida en la **Tabla 10** y en la **Ilustración 35**.

Tabla 10. Información sobre el Sprint 6.

Fecha Inicio	Fecha Fin
11/05/2023	22/05/2023
Objetivo del Sprint 5	
Lectura de los ficheros ráster y creación de un mapa genérico en el frontend.	

Sprint 6

Historia de usuario	Id. de eleme...	T.Estimado
Planificación	1226806358	1800 s
Lectura de ficheros Ráster 2	1226809797	10.800 s
Usabilidad 1	1226810090	3600 s
Menú 1	1226810537	3600 s
Mapa general 2	1226810867	7200 s

Subelemento	Id. de eleme...	Prioridad	Estado
Servicio Punto	1226811325	Medium	
Controlador	1226811869	Medium	

Ilustración 35. Sprint Backlog del Sprint 6.

I.6.2. Desarrollo del Sprint 6

En la primera etapa de este sprint, se lleva a cabo una historia de usuario relacionada con la lectura de un archivo raster con el objetivo de obtener los datos contenidos en cada celda. En este caso, el raster utilizado contiene información sobre la elevación. Para realizar la lectura del archivo raster, se utiliza la librería gdal, que permite

abrir el archivo .hdr que contiene la información necesaria para interpretar los datos del archivo .bil. En la **Ilustración 36** se pueden observar los datos extraídos del archivo .hdr. Con esta información, se procede a recorrer el archivo raster .bil celda por celda para identificar los datos contenidos en cada una.

El resto del desarrollo del sprint se centra en tareas de implementación en el frontend, que se detallan a continuación:

i. Se crea un menú (sidenav) mediante la creación de una interfaz y una constante que define la información a mostrar en dicho menú. Esto permite una mayor escalabilidad y modularidad del sistema, veáse en la Ilustración 37.

ii. Se implementa un mapa utilizando la librería Leaflet, en el cual se muestran los puntos de inicio y fin de cada tramo, veáse en la Ilustración 39. Para obtener estos puntos, se realiza una petición a la API utilizando un servicio denominado "punto", se crea este servicio para poder reutilizar y sea modulable.

iii. Se desarrolla un panel emergente que se despliega al hacer clic en el icono de ajustes en el menú horizontal.

iv. En el backend, se agrega un método al controlador de tramos para filtrar las vías en obras y las que no lo están.

v. En el frontend, se agrega un mapa que muestra las vías en obras y las vías en estado normal para las vías del ferrocarril, veáse en la **Ilustración 38**.

```
const NCOLS_width = dataset.rasterSize.x;
const NROWS_height = dataset.rasterSize.y;
const NBANDS_bandsCount = dataset.bands.count();
const NBITS_dataTypeString = dataset.bands.get(1).dataType;
//const NBITS_dataType = parseInt(NBITS_dataTypeString);
const NBITS_dataType: number = 16;

const geoTransform = dataset.geoTransform;
const ULXMAP = geoTransform[0];
const XDIM = geoTransform[1];
const YDIM = -geoTransform[5];
const ULYMAP = geoTransform[3];
// Obtener información de la banda
const band = dataset.bands.get(1);
const nodataValue = band.noDataValue;
const buffer = Buffer.alloc(NCOLS_width * NROWS_height * NBITS_dataType / 8);
```

Ilustración 36. Lectura fichero ráster .hdr

Durante el desarrollo del sprint, se lleva a cabo una reunión de todo el equipo el día 17 de mayo. Durante esta reunión, se decide cambiar el enfoque en la búsqueda de puntos consecutivos, ya que no se encuentran dichos puntos de manera directa. En su lugar, se propone explorar la idea de seleccionar un punto de inicio y un punto de fin en

forma de rectángulo, y los tramos contenidos dentro de este rectángulo serán considerados para el cálculo de la ruta.

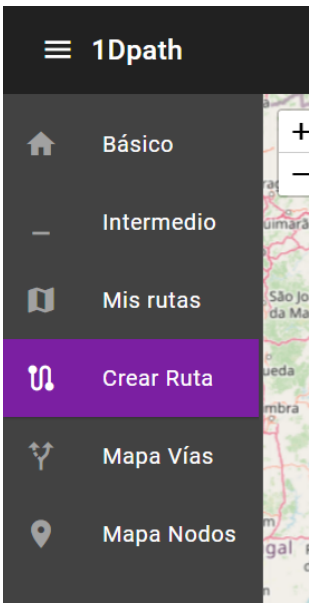


Ilustración 37. Menú.

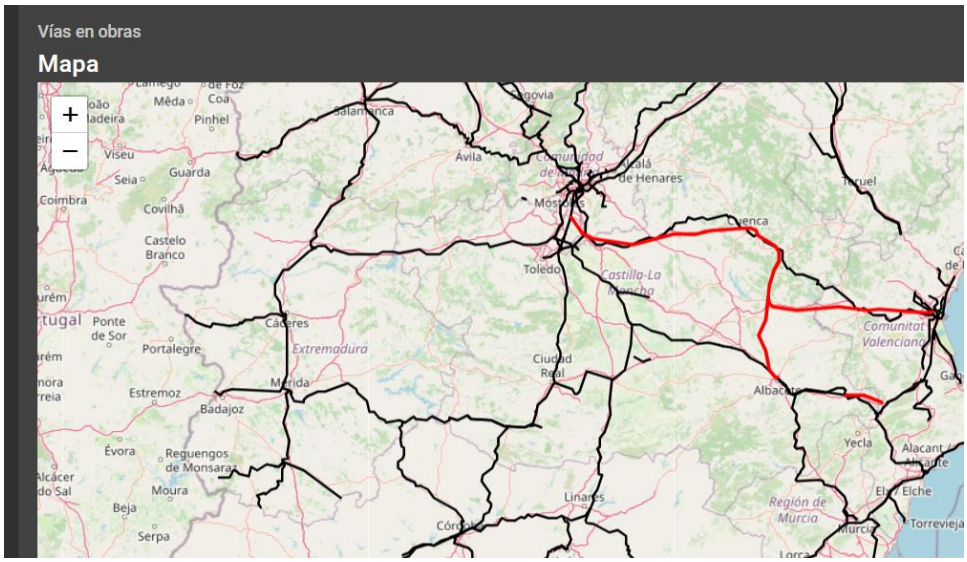


Ilustración 38. Mapa de vías en obras.

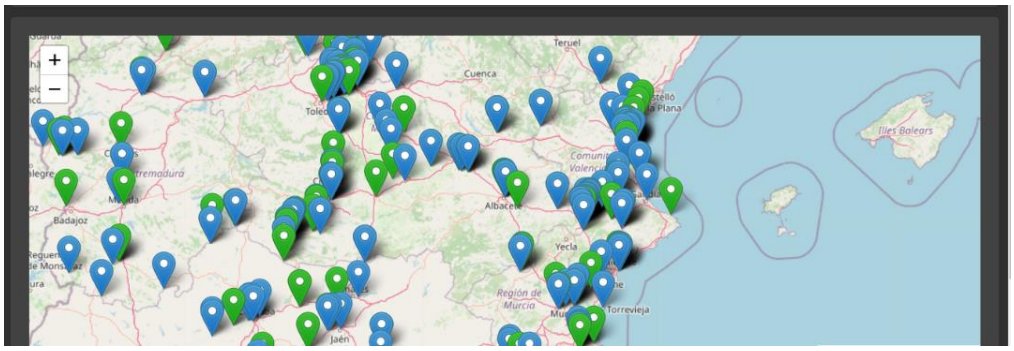


Ilustración 39. Mapa con estaciones.

I.6.3. Revisión del Sprint 6

Se presentan todos los avances explicados en el desarrollo del sprint.

I.6.4. Retrospectiva del Sprint 6.

Se hacen notables los avances en el frontend y se detecta código duplicado. La retropectiva de este sprint queda plasmada en la **Tabla 11**.

Tabla 11. Retrospectiva del Sprint 6.

¿Qué estuvo bien?	¿Qué se puede mejorar?	Acciones para mejorar
Interfaz	Hay mucho código repetido al diferenciar entre punto de inicio y fin de tramo.	Refactorizar.

I.7 Sprint 7

Este sprint se enfoca en la incorporación de datos climatológicos en el cálculo de la ruta. Para lograr esto, se toma la decisión de utilizar los datos proporcionados por la API de la Agencia Estatal de Meteorología (AEMET).

I.7.1. Planificación del Sprint 7

El objetivo de este sprint se centra en la asignación de pesos a cada uno de los puntos que conforman los tramos de las vías ferroviarias. Con este fin, se han desglosado las historias de usuario pertinentes, las cuales se detallan en la **Ilustración 40**. El propósito es lograr la asociación adecuada de pesos a los puntos para facilitar el cálculo de rutas de manera eficiente.

Tabla 12. Información sobre el Sprint 7.

Fecha Inicio	Fecha Fin
23/05/2023	02/06/2023
Objetivo del Sprint 7	
Cálculo de pesos.	

Sprint 7 6 Historia de usuarios / 1 subelemento

Historia de usuario	Id. de eleme...	T.Estimado	Prioridad
Planificación	1226890815	1800 s	Low
Retrospectiva	1226890889	1800 s	Low
Petición a api AEMET 1	1226891408	28.800 s	High
Estaciones aemet cercanas a un punto	1226892701	1800 s	High
Interpolar pesos	1226894204	3600 s	Medium
Calcular peso del tramo	1226894420	1800 s	Medium

Subelemento	Id. de eleme...	Prioridad	Estado	T.Estimado
Controlador	1226893905			
+ Agregar Subelemento				

Ilustración 40. *Sprint Backlog* del Sprint 7.

I.7.2. Desarrollo del Sprint 7

Reuniones que se dieron durante el desarrollo de este sprint fueron las siguientes:

- **Reunión - 26/05/2023** *(Daily scrum)*. Asistentes: Enrique Arias, José Antonio Mateo y Henar Velasco. El motivo de la reunión consiste es establecer los siguientes objetivos:
  i. Se plantea encontrar las estaciones cercanas a los puntos de cada tramo.
  ii. Se requiere una API key de AEMET.
  iii. Existe la problemática de que los puntos extremos de los tramos no coinciden, por lo que se debe determinar los puntos finales de cada tramo y establecer los puntos de inicio más cercanos como consecutivos.
- **Reunión - 01/06/2023** *(Daily scrum)*. Asistentes: José Antonio Mateo y Henar Velasco. El motivo de la reunión consiste en presentar los siguientes avances:
  i. Se obtienen de cada punto las estaciones AEMET más cercanas.

ii. Se consigue acceso a la API y se controlan errores (errores como problemas de conexión y la API no devuelve datos).

iii. En relación a los datos recogidos por las estaciones cercanas a cada punto, se ha aplicado un proceso de interpolación para obtener un valor representativo. Este valor interpolado se asigna posteriormente al punto correspondiente. A continuación, se realiza una suma de los pesos de todos los puntos que forman parte de un tramo y este valor resultante se asigna al tramo en cuestión.

iv. Se crea un controlador que, pasándole el id de una estación AEMET, devuelve datos recogidos por esa estación. La api de la AEMET utilizada se encuentra en la sección "valores climatológicos" del siguiente enlace: https://opendata.aemet.es/dist/index.html?#/valores-climatologicos/Climatolog%C3%ADas%20diarias.

I.7.3. Revisión del Sprint 7

Se presentan todos los avances explicados en el desarrollo del sprint.

I.7.4. Retrospectiva del Sprint 7

En esta reunión retrospectiva el equipo reflexiona sobre los impedimentos surgidos y debido a las fechas en las que se encuentra el proyecto se toman decisiones como se puede ver en la **Tabla 13** para dar solución a estos problemas.

Tabla 13. Retrospectiva del Sprint 7.

¿Qué estuvo bien?	¿Qué se puede mejorar?	Acciones para mejorar
Implementación de los métodos que asignan pesos a los puntos.	Problema de falta de recolección de datos de la API. No se obtienen datos en tiempo real de la API. Sin embargo, sí se recopilan datos de la estación (API observacion-convencional) correspondientes a las últimas 24 horas.	Generación aleatoria de valores para la precipitación en una estación, ya que no hay un valor predeterminado.

I.8 Sprint 8

Este último sprint se distingue por la adopción de decisiones y la implementación de acciones destinadas a resolver los impedimentos surgidos. Estas acciones consideran las fechas del proyecto y se enfocan en lograr un producto mínimo que cumpla con los objetivos establecidos para este trabajo.

I.8.1. Planificación del Sprint 8

La planificación de este sprint se encuentra documentada tanto en la Tabla 14 como en la Ilustración 41.

Tabla 14. Información sobre el Sprint 8.

Fecha Inicio	Fecha Fin
5/06/2023	14/06/2023
Objetivo del Sprint 8	
Construir el algoritmo Dijkstra y caso de estudio Albacete-Murcia.	

I.8.2. Desarrollo del Sprint 8

El desarrollo de este sprint se caracteriza por tres aspectos principales:

i. Se centra en la construcción del algoritmo Dijkstra.

ii. Lo cual implica la creación de un grafo que representa el caso de estudio Albacete-Murcia. Este proceso requiere un trabajo minucioso en el cual se identifican las geometrías de los puntos de inicio y fin de cada tramo, detallado en el capítulo de análisis y diseño de la memoria. La construcción del grafo se realiza de forma manual, dado que aún no se han definido los distintos tipos de escenarios y criterios para establecer la consecutividad entre los puntos y los tramos, debido a la complejidad del problema.

iii. Se lleva a cabo la actualización del frontend, con énfasis en la simplificación del menú y la mejora de la usabilidad en los modos básico e intermedio. Se añaden mapas y gráficos informativos, y en el modo avanzado se implementa una agrupación de estaciones para evitar la saturación del mapa, como se muestra en la imagen adjunta.

Las reuniones que se han llevado a cabo en el desarrollo de este sprint son las siguientes:

- **Reunión - 09/06/2023** *(Daily scrum)*. Asistentes: Enrique Arias, José Antonio Mateo y Henar Velasco. Motivo:
  - El equipo se puso día sobre en qué punto se encontra el frontend.
  - Y conocer en que punto se encuentra el desarrollo del algoritmo de Dijkstra.

En consecuencia, se marcó como tarea a desarrollar en este sprint retomar el frontend y el desarrollo del algoritmo de Dijkstra. Se mencionó que había un *stepper* [10]que no funcionaba correctamente y que necesitaba ser refactorizado para mejorar su usabilidad, como se muestra en la **Ilustración 43**. Además, se señaló que había un problema con el grafo [11]creado porque estaba dirigido en una sola dirección y se quería que se pudiera ir en ambas direcciones. Es decir, si el punto a tiene un punto consecutivo que es el punto b, faltaría por expresar en el grafo que para el punto b un punto consecutivo es el punto a.

- **Reunión - 09/06/2023** *(Daily scrum)* en la que se discute la demostración del algoritmo de Dijkstra. Avances expuestos en la reunión:
  - Se encuentra una solución para el grafo y se incluye la capacidad de ir en ambas direcciones.
  - Se termina con la interfaz del Modo Básico e Intermedio del proyecto.
- **Reunión - 12/06/2023** (*Daily scrum*), se discutió la demostración del proyecto y se señaló que se habían encontrado algunos problemas con la visualización, se muestran los tramo al completo en lugar de sólo una parte en particular, la que forma parte de la ruta. Se agregó una tarea al sprint para solucionar este problema.

I.8.3. Revisión del Sprint 6

Se muestran los avances:

i. En el mapa del Modo Avanzado: se agrupan las estaciones para que el mapa no se vea saturado, y se agrupan y desagrupan haciendo *zoom in-zoom out* permitiendo una mejora de usabilidad, veáse en **Ilustración 42**.

---

[10] *Stepper*: es un componente de Angular personalizable que visualiza el contenido como un proceso y muestra su progreso.

[11] Grafo: se construye un grafo que representa el problema analizado en el estudio. Este grafo está compuesto por los puntos correspondientes a los tramos involucrados en el caso de estudio.

ii. Se presenta la aplicación al completo, tal y como se expone en el capítulo 4 Análisis y Diseño.

Sprint 8

Historia de usuario	Id. de eleme...	T.Estimado	Prioridad
Planificación	1227001455	1800 s	Low
Retrospectiva	1227001516	1800 s	Low
Implementación Dijkstra	1227022107	28.800 s	Critical
Crear caso de estudio 1	1227024186	100.800 s	Critical
Refactorizar	1227027515	3600 s	Low
Stepper 1	1227027877	28.800 s	Low

Subelemento	Id. de eleme...	Prioridad	Estado	T.Estimad
Campos requeridos	1227027975	Low		

Mostrar resultado 1	1227028576	0 s	Low

Subelemento	Id. de eleme...	Prioridad	Estado	T.Estimad
quitar de la interfaz sobrantes	1227045239			

Modo Básico	1227028763	25.200 s	Medium
Menú	1227030732	3600 s	Low
Modo Intermedio	1227031968	25.200 s	Medium
Modo Avanzado 2	1227032734	18.000 s	Low

Subelemento	Id. de eleme...	Prioridad	Estado	T.Estimad
Mapa cluster estaciones	1227033531			
Vista stepper vertical	1227033665			

Ilustración 41. *Sprint Backlog* del Sprint 8.

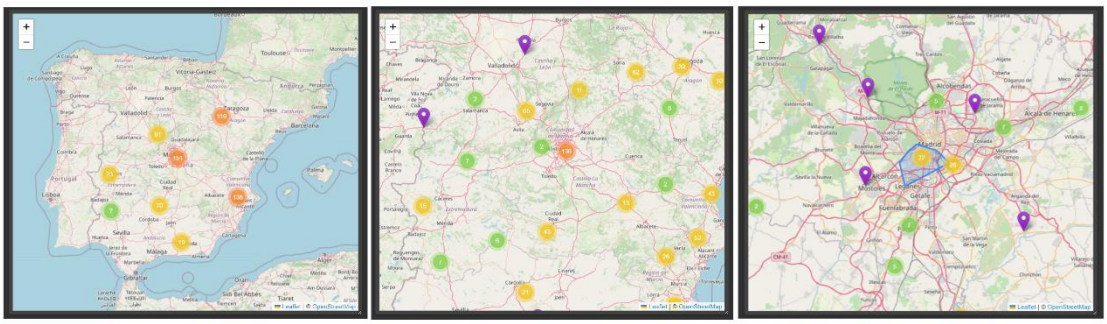


Ilustración 42. Agrupación de estaciones.

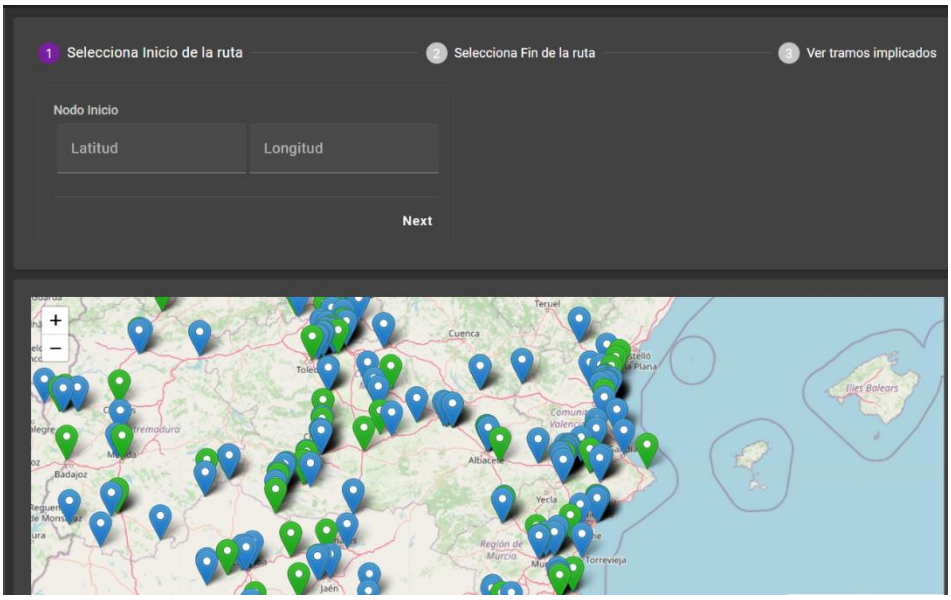


Ilustración 43. Stepper.