

# Toure

PROYECTO FIN DE GRADO – UNIVERSIDAD DE OVIEDO  
CARLOS MENCHACA MOSTEIRO

# *Agradecimientos*

---

A mi familia, a mis compañeros “De Chillin”, a mi pareja y a mis amigos. En especial a mi padre y madre, Jose Antonio Menchaca Fernández y María Marlen Mosteiro García, que sin ellos jamás podría haber llegado a ser la persona que soy hoy.

“Brindemos por lo que tenemos, por lo que queremos, por lo que aún no vino” *Brindis*.  
(Toteking y Easy-s, 2019).

# Resumen

---

El objetivo es crear una aplicación donde un usuario que practique senderismo pueda gestionar fácilmente su historial de rutas. Un sistema donde pueda importar sus rutas desde diferentes formatos, visualizarlas en diferentes mapas u organizarlas en listas.

Durante el desarrollo se utilizarán tecnologías extendidas dentro de la industria del desarrollo de software y no impartidas durante la carrera.

# *Palabras Clave*

---

Ruta, Mapa, Senderismo, GPS, Track, GPX, IGN

# *Abstract*

---

The main objective is to create an application through which a user who goes hiking quite often can easily manage his route history. A system through which this person can import the aforementioned routes from different formats, allowing the user to visualize them on different maps or organize them in lists.

During the development of the app, technologies that are widespread within the software development industry will be used and which have not been taught during the degree

# *Keywords*

---

Route, Map, Hiking, GPS, Track, GPX, IGN

# Índice General

<b>CAPÍTULO 1. MEMORIA DEL PROYECTO .....</b>	<b>14</b>
1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO .....	14
1.2 RESUMEN DE TODOS LOS ASPECTOS .....	15
1.2.1 <i>Introducción</i> .....	15
1.2.2 <i>Aspectos teóricos</i> .....	15
1.2.3 <i>Planificación del Proyecto y Presupuesto Iniciales</i> .....	15
1.2.4 <i>Análisis</i> .....	15
1.2.5 <i>Diseño del sistema</i> .....	15
1.2.6 <i>Implementación del sistema</i> .....	15
1.2.7 <i>Desarrollo de las pruebas</i> .....	16
1.2.8 <i>Manuales del sistema</i> .....	16
1.2.9 <i>Conclusiones y ampliaciones</i> .....	16
1.2.10 <i>Planificación del proyecto y presupuesto</i> .....	16
1.2.11 <i>Referencias bibliográficas</i> .....	16
1.2.12 <i>Apéndices</i> .....	16
<b>CAPÍTULO 2. INTRODUCCIÓN .....</b>	<b>17</b>
2.1 JUSTIFICACIÓN DEL PROYECTO .....	17
2.2 OBJETIVOS DEL PROYECTO .....	18
2.2.1 <i>Importar datos GPS y mostrarlos</i> .....	18
2.2.2 <i>Seleccionar varios tipos de mapas</i> .....	19
2.2.3 <i>Mostrar varios recorridos en un mismo mapa</i> .....	19
2.2.4 <i>Varios niveles de organización</i> .....	19
2.2.5 <i>Principios SOLID</i> .....	19
2.2.6 <i>Aprender nuevas tecnologías</i> .....	20
2.3 ESTUDIO DE LA SITUACIÓN ACTUAL .....	21
2.3.1 <i>Evaluación de Alternativas</i> .....	21
2.3.2 <i>Comparativa</i> .....	25
<b>CAPÍTULO 3. ASPECTOS TEÓRICOS .....</b>	<b>26</b>
3.1 DOMINIO DEL SISTEMA.....	¡ERROR! MARCADOR NO DEFINIDO.
3.2 ELECTRON .....	26
3.3 GOOGLEMAPS JAVASCRIPT API .....	26
3.4 NEDB.....	26
3.5 IGN Y OPENLAYERS.....	27
3.6 REACT.....	27
3.7 TYPESCRIPT.....	28
<b>CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y PRESUPUESTO INICIALES.....</b>	<b>29</b>
4.1 PLANIFICACIÓN INICIAL.....	29
4.1.1 <i>Product Backlog</i> .....	29
4.1.2 <i>Sprints</i> .....	30
4.2 PRESUPUESTO INICIAL.....	31
4.2.1 <i>Presupuesto de costes</i> .....	31
4.2.2 <i>Presupuesto del cliente</i> .....	33

<b>CAPÍTULO 5. ANÁLISIS .....</b>	<b>35</b>
5.1 REQUISITOS DEL SISTEMA .....	35
5.1.1 <i>Obtención de los Requisitos del Sistema</i> .....	35
5.1.2 <i>Identificación de Actores del Sistema</i> .....	36
5.1.3 <i>Especificación de Casos de Uso</i> .....	37
5.2 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS .....	40
5.2.1 <i>Descripción de los Subsistemas</i> .....	40
5.2.2 <i>Descripción de los Interfaces entre Subsistemas</i> .....	41
5.3 ANÁLISIS DE CASOS DE USO Y ESCENARIOS.....	42
5.3.1 <i>Caso N.1 Importar una ruta</i> .....	42
5.3.2 <i>Caso N.2 Ver datos de una ruta</i> .....	43
5.3.3 <i>Caso N.3 Eliminar una ruta del sistema</i> .....	43
5.3.4 <i>Caso N.4 Ver todas las rutas del sistema sobre un mapa</i> .....	43
5.3.5 <i>Caso N.5 Crear una lista</i> .....	44
5.3.6 <i>Caso N.6 Añadir una ruta a una lista</i> .....	44
5.3.7 <i>Caso N.7 Ver las rutas de una lista sobre un mapa</i> .....	44
5.3.8 <i>Caso N.8 Eliminar una ruta de una lista</i> .....	45
5.3.9 <i>Caso N.9 Eliminar una lista del sistema</i> .....	45
5.4 ANÁLISIS DE INTERFACES DE USUARIO .....	46
5.4.1 <i>Descripción de la Interfaz</i> .....	46
5.4.2 <i>Descripción del Comportamiento de la Interfaz</i> .....	49
5.4.3 <i>Diagrama de Navegabilidad</i> .....	50
5.5 ESPECIFICACIÓN DEL PLAN DE PRUEBAS.....	51
<b>CAPÍTULO 6. DISEÑO DEL SISTEMA.....</b>	<b>52</b>
6.1 PROBLEMAS Y NECESIDADES EN EL DISEÑO .....	52
6.1.1 <i>Comunicar archivos y mapa</i> .....	52
6.1.2 <i>Modelo y persistencia</i> .....	53
6.1.3 <i>Importar varios tipos de ficheros</i> .....	55
6.1.4 <i>Visualizar diferentes mapas</i> .....	57
6.2 DISEÑO DE LA INTERFAZ .....	59
6.2.1 <i>Barra de navegación</i> .....	59
6.2.2 <i>Gestión de rutas</i> .....	59
6.2.3 <i>Mapa general</i> .....	62
6.2.4 <i>Datos de una ruta</i> .....	65
6.3 ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	68
6.3.1 <i>Pruebas del Sistema</i> .....	68
6.3.2 <i>Pruebas de Usabilidad</i> .....	71
6.3.3 <i>Pruebas de Accesibilidad</i> .....	74
<b>CAPÍTULO 7. IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>78</b>
7.1 LENGUAJES DE PROGRAMACIÓN.....	78
7.1.1 <i>Css</i> .....	78
7.1.2 <i>Electron</i> .....	78
7.1.3 <i>GoogleMaps JavaScript API</i> .....	79
7.1.4 <i>NeDB</i> .....	80
7.1.5 <i>Openlayers</i> .....	80
7.1.6 <i>React</i> .....	80
7.1.7 <i>TypeScript</i> .....	81

7.1.8	Webpack.....	81
7.1.9	Otras dependencias.....	82
7.2	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO.....	85
7.2.1	Github.....	85
7.2.2	Npm.....	85
7.2.3	Sourcetree.....	86
7.2.4	Visual Studio Code.....	86
7.2.5	Windows 10.....	87
7.2.6	Yarn.....	87
7.3	PROBLEMAS ENCONTRADOS.....	88
7.3.1	Asincronía al mostrar datos.....	88
7.3.2	Content-Security-Policy.....	89
7.3.3	Distancia total de una ruta.....	91
7.3.4	Distinguir una ruta de una lista.....	92
7.3.5	Guardar rutas dentro de una lista.....	93
7.3.6	Mantener el centro y zoom.....	93
7.3.7	Representar un archivo KML en GoogleMaps.....	94
7.3.8	Tipo de dato devuelto por NeDB.....	94
7.3.9	Otros problemas.....	95
<b>CAPÍTULO 8. DESARROLLO DE LAS PRUEBAS.....</b>		<b>96</b>
8.1	PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA.....	96
8.1.1	Resultados.....	96
8.1.2	Cambios realizados.....	99
8.1.3	Conclusiones.....	100
8.2	PRUEBAS DE USABILIDAD.....	100
8.2.1	Resultados.....	100
8.2.2	Cambios realizados.....	102
8.2.3	Conclusiones.....	104
8.3	PRUEBAS DE ACCESIBILIDAD.....	104
8.3.1	Resultados.....	104
8.3.2	Cambios realizados.....	107
8.3.3	Conclusiones.....	108
<b>CAPÍTULO 9. MANUALES DEL SISTEMA.....</b>		<b>109</b>
9.1	MANUAL DE INSTALACIÓN.....	109
9.2	MANUAL DE EJECUCIÓN.....	110
9.3	MANUAL DE USUARIO.....	111
9.4	MANUAL DEL PROGRAMADOR.....	118
9.4.1	Inicio del sistema.....	118
9.4.2	Ayuda para modificar el código.....	118
9.4.3	Empaquetar.....	119
<b>CAPÍTULO 10. CONCLUSIONES Y AMPLIACIONES.....</b>		<b>120</b>
10.1	CONCLUSIONES.....	120
10.2	AMPLIACIONES.....	121
<b>CAPÍTULO 11. PLANIFICACIÓN DEL PROYECTO Y PRESUPUESTO FINALES.....</b>		<b>122</b>
11.1	PLANIFICACIÓN FINAL.....	122
11.1.1	Product Backlog final.....	122

---

11.1.2	<i>Sprints finales</i> .....	122
11.2	PRESUPUESTO FINAL.....	123
11.2.1	<i>Presupuesto de costes final</i> .....	124
11.2.2	<i>Presupuesto del cliente final</i> .....	126
<b>CAPÍTULO 12.</b>	<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>127</b>
12.1	LIBROS Y ARTÍCULOS.....	127
12.2	REFERENCIAS EN INTERNET .....	128
<b>CAPÍTULO 13.</b>	<b>APÉNDICES</b> .....	<b>131</b>
13.1	GLOSARIO Y DICCIONARIO DE DATOS .....	131
13.2	CONTENIDO ENTREGADO EN LOS ANEXOS.....	132
13.3	CÓDIGO FUENTE .....	133
13.3.1	<i>Contenido del código fuente</i> .....	133

# Índice de Figuras

Ilustración 2.1 Datos de una ruta .....	18
Ilustración 2.2 Interfaz de BaseCamp .....	22
Ilustración 2.3 Interfaz de Strava .....	23
Ilustración 2.4 Interfaz de Wikiloc.....	24
Ilustración 3.1 Componente de clase .....	28
Ilustración 3.2 Componente funcional .....	28
Ilustración 5.1 Diagrama con los casos de uso.....	37
Ilustración 5.2 interacción entre los subsistemas .....	41
Ilustración 5.3 Prototipo de pantalla inicial.....	46
Ilustración 5.4 Prototipo de la pantalla con el mapa general .....	47
Ilustración 5.5 Prototipo de la pantalla para gestionar las rutas .....	48
Ilustración 5.6 Prototipo de la pantalla que muestra los datos de una ruta .....	49
Ilustración 5.7 Diagrama de navegabilidad .....	50
Ilustración 6.1 Estructura del modelo .....	53
Ilustración 6.2 Datos del modelo .....	54
Ilustración 6.3 Métodos de acceso a base de datos.....	54
Ilustración 6.4 Objeto Nedb .....	55
Ilustración 6.5 Diseño del Parser .....	56
Ilustración 6.6 Ifelse en ParserManager .....	57
Ilustración 6.7 Diseño de los mapas .....	58
Ilustración 6.8 Barra de navegación .....	59
Ilustración 6.9 Pantalla de gestión .....	59
Ilustración 6.10 Interfaz para importar las rutas .....	60
Ilustración 6.11 Interfaz de gestión .....	60
Ilustración 6.12 Icono para crear una nueva lista .....	60
Ilustración 6.13 Icono para editar una lista .....	60
Ilustración 6.14 Icono para eliminar una lista .....	60
Ilustración 6.15 Modal para confirmar el borrado de una lista.....	61
Ilustración 6.16 Icono para añadir una ruta a una lista .....	61
Ilustración 6.17 Icono para visualizar una ruta.....	61
Ilustración 6.18 Icono para eliminar una ruta del sistema.....	61
Ilustración 6.19 Modal para confirmar el borrado de una ruta .....	61
Ilustración 6.20 Interfaz de gestión de una lista .....	62
Ilustración 6.21 Mapa general.....	62
Ilustración 6.22 Mapa general con rutas .....	63
Ilustración 6.23 Vista relieve del mapa general.....	63
Ilustración 6.24 Vista satélite del mapa general.....	64
Ilustración 6.25 Vista del IGN del mapa general.....	64
Ilustración 6.26 Icono para visualizar una ruta.....	65
Ilustración 6.27 Datos de una ruta .....	65
Ilustración 6.28 Vista relieve en los datos de una ruta .....	66
Ilustración 6.29 Vista satélite en los datos de una ruta .....	66
Ilustración 6.30 Vista del IGN en los datos de una ruta .....	67
Ilustración 7.1 Función para obtener todas las rutas de base de datos .....	89
Ilustración 7.2 Fallo al cargar la API de GoogleMaps .....	89

<b>Ilustración 7.3 Fallo en la CSP con cabecera .....</b>	<b>90</b>
<b>Ilustración 7.4 Solución al problema con CSP .....</b>	<b>91</b>
<b>Ilustración 7.5 Función para calcular la distancia total .....</b>	<b>92</b>
<b>Ilustración 7.6. Tipos en base de datos como constantes.....</b>	<b>92</b>
<b>Ilustración 7.7 Código para visualizar archivo KML en GoogleMaps .....</b>	<b>94</b>
<b>Ilustración 7.8 Función que transforma un documento en una Ruta .....</b>	<b>95</b>
<b>Ilustración 8.1 Error al eliminar una lista .....</b>	<b>100</b>
<b>Ilustración 8.2 Importar .....</b>	<b>103</b>
<b>Ilustración 8.3 Nombre de una ruta .....</b>	<b>103</b>
<b>Ilustración 8.4 Modal para añadir una ruta a una o varias listas, antes de las pruebas .....</b>	<b>103</b>
<b>Ilustración 8.5 Modal para añadir una ruta a una o varias listas, después de las pruebas .....</b>	<b>104</b>
<b>Ilustración 8.6 Código que muestra un icono accesible por teclado .....</b>	<b>108</b>
<b>Ilustración 8.7 Modal que notifica un formato de archivo incorrecto.....</b>	<b>108</b>
<b>Ilustración 9.1 Distribución para Windows .....</b>	<b>110</b>
<b>Ilustración 9.2 Pantalla inicial .....</b>	<b>111</b>
<b>Ilustración 9.3 Página rutas.....</b>	<b>112</b>
<b>Ilustración 9.4 Pantalla con rutas importadas .....</b>	<b>113</b>
<b>Ilustración 9.5 Pantalla editando una ruta .....</b>	<b>114</b>
<b>Ilustración 9.6 Mapa general con varias rutas importadas .....</b>	<b>115</b>
<b>Ilustración 9.7 Mapa general con rutas pintadas .....</b>	<b>116</b>
<b>Ilustración 9.8 Visualización de una ruta individual .....</b>	<b>117</b>

# Índice de Tablas

Tabla 2.1 Comparativa de alternativas.....	25
Tabla 4.1 Product Backlog inicial.....	30
Tabla 4.2 Sprint 1 inicial.....	30
Tabla 4.3 Sprint 2 inicial.....	30
Tabla 4.4 Sprint 3 inicial.....	31
Tabla 4.5 Sprint 4 inicial.....	31
Tabla 4.6 Sprint 5 inicial.....	31
4.7 Presupuesto de costes inicial.....	33
4.8 Resumen de presupuesto de costes con beneficio inicial.....	33
4.9 Presupuesto de cliente inicial.....	34
Tabla 5.1 Especificación de Caso de uso, Importar una ruta.....	37
Tabla 5.2 Especificación de Caso de uso, Ver datos de una ruta.....	38
Tabla 5.3 Especificación de Caso de uso, Eliminar una ruta del sistema.....	38
Tabla 5.4 Especificación de Caso de uso, Ver todas las rutas del sistema sobre un mapa.....	38
Tabla 5.5 Especificación de Caso de uso, Crear una lista.....	38
Tabla 5.6 Especificación de Caso de uso, Añadir una ruta a una lista.....	38
Tabla 5.7 Especificación de Caso de uso, Ver las rutas de una lista sobre un mapa.....	38
Tabla 5.8 Especificación de Caso de uso, Eliminar una ruta de una lista.....	38
Tabla 5.9 Especificación de Caso de uso, Eliminar una lista del sistema.....	39
Tabla 5.10 Caso N.1 Importar una ruta.....	43
Tabla 5.11 Caso N.2 Ver datos de una ruta.....	43
Tabla 5.12 Caso N.3 Eliminar una ruta del sistema.....	43
Tabla 5.13 Caso N.4 Ver todas las rutas del sistema sobre un mapa.....	44
Tabla 5.14 Caso N.5 Crear una lista.....	44
Tabla 5.15 Caso N.6 Añadir una ruta a una lista.....	44
Tabla 5.16 Caso N.7 Ver las rutas de una lista sobre un mapa.....	45
Tabla 5.17 Caso N.8 Eliminar una ruta de una lista.....	45
Tabla 5.18 Caso N.9 Eliminar una lista del sistema.....	45
Tabla 6.1 Pruebas del Caso N.1 Importar una ruta.....	68
Tabla 6.2 Pruebas del Caso N.2 Ver datos de una ruta.....	69
Tabla 6.3 Pruebas del Caso N.3 Eliminar una ruta del sistema.....	69
Tabla 6.4 Pruebas del Caso N.4 Ver todas las rutas del sistema sobre un mapa.....	70
Tabla 6.5 Pruebas del Caso N.5 Crear una lista.....	70
Tabla 6.6 Pruebas del Caso N.6 Añadir una ruta a una lista.....	70
Tabla 6.7 Pruebas del Caso N.7 Ver las rutas de una lista sobre un mapa.....	70
Tabla 6.8 Pruebas del Caso N.8 Eliminar una ruta de una lista.....	71
Tabla 6.9 Pruebas del Caso N.9 Eliminar una lista del sistema.....	71
Tabla 6.10 Preguntas de carácter general.....	72
Tabla 6.11 Actividades guiadas.....	72
Tabla 6.12 Preguntas cortas sobre la aplicación y observaciones.....	73
Tabla 6.13 Cuestionario para el responsable de las pruebas.....	73
Tabla 6.14 WCAG Nivel A.....	75
Tabla 6.15 WCAG Nivel AA.....	77
Tabla 8.1 Pruebas del Caso N.1 Importar una ruta.....	96
Tabla 8.2 Pruebas del Caso N.2 Ver datos de una ruta.....	97

Tabla 8.3 Pruebas del Caso N.3 Eliminar una ruta del sistema .....	97
Tabla 8.4 Pruebas del Caso N.4 Ver todas las rutas del sistema sobre un mapa .....	98
Tabla 8.5 Pruebas del Caso N.5 Crear una lista.....	98
Tabla 8.6 Pruebas del Caso N.6 Añadir una ruta a una lista.....	98
Tabla 8.7 Pruebas del Caso N.7 Ver las rutas de una lista sobre un mapa .....	99
Tabla 8.8 Pruebas del Caso N.8 Eliminar una ruta de una lista .....	99
Tabla 8.9 Pruebas del Caso N.9 Eliminar una lista del sistema .....	99
Tabla 8.10 Resultados de las preguntas de carácter general .....	101
Tabla 8.11 Resultados de las actividades guiadas .....	101
Tabla 8.12 Resultado de las preguntas cortas .....	102
Tabla 8.13 Resultado del cuestionario para el responsable de las pruebas .....	102
Tabla 8.14 WCAG Nivel A .....	106
Tabla 8.15 WCAG Nivel AA.....	107
11.1 Product Backlog final .....	122
Tabla 11.2 Sprint 1 final .....	123
Tabla 11.3 Sprint 2 final .....	123
Tabla 11.4 Sprint 3 final .....	123
Tabla 11.5 Sprint 4 final .....	123
Tabla 11.6 Sprint 5 final .....	123
11.7 Presupuesto de costes final.....	126
11.8 Resumen de presupuesto de costes con beneficio final.....	126
11.9 Presupuesto de cliente final .....	126
13.1 Contenido del proyecto.....	132
13.2 Contenido del código fuente .....	133

# Capítulo 1. Memoria del Proyecto

En este primer capítulo, se describirá brevemente el contenido y formato del presente documento.

## 1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

El sistema desarrollado es una aplicación de escritorio para Windows (extensible a Mac y Linux), que no necesita instalar ningún tipo de software adicional. El objetivo principal es proveer un entorno donde el usuario pueda guardar y gestionar sus propias rutas de senderismo. Por lo tanto, el cliente potencial hacia el que está enfocado el sistema es al usuario al que le guste practicar senderismo y guardar los recorridos que haga, para hacer un posterior análisis de ellos.

Algunos de los objetivos generales del proyecto son: importar datos GPS desde diferentes tipos de archivos, visualizar estos datos en diferentes mapas, visualizar varias rutas en un mismo mapa o aplicar tecnologías no impartidas durante la carrera.

Finalmente, el sistema aspira a competir con otros colosos del mercado como Strava o Wikiloc, proveyendo algunas funcionalidades que a estos les faltan por implementar o que, desde mi punto de vista, son mejorables.

## 1.2 Resumen de Todos los Aspectos

A continuación, se definirá de forma general la estructura del documento con una breve descripción de sus apartados.

### 1.2.1 Introducción

Una descripción más extendida de qué pretende ser el sistema. La idea tras la que nace el proyecto, los objetivos fijados y otras alternativas similares que se pueden encontrar a día de hoy (24 de mayo, 2022) en el mercado.

### 1.2.2 Aspectos teóricos

Aquí describiré las principales tecnologías usadas durante el desarrollo, profundizando en porqué he decidido usarlas y haciendo especial énfasis en aspectos teóricos de las mismas. Este capítulo pretende explicar al lector los pilares sobre los que está construido el sistema.

### 1.2.3 Planificación del Proyecto y Presupuesto Iniciales

En este capítulo detallaré la planificación y presupuesto iniciales. Ambas se estimaron al principio del proyecto.

### 1.2.4 Análisis

El Análisis contendrá los requisitos, casos de uso y otros aspectos de la documentación, a partir de la cual se elaborará posteriormente el diseño.

### 1.2.5 Diseño del sistema

En este capítulo se recogerá el diseño del proyecto previo al desarrollo. Si hiciésemos el símil con la construcción de una casa, este apartado serían los planos que se necesitan previos a la construcción de la misma.

### 1.2.6 Implementación del sistema

Se hará una breve descripción de las tecnologías usadas en el desarrollo, para que el lector que no conozca alguna de estas, pueda entenderlas rápidamente. También se explicarán algunos de los problemas encontrados durante el desarrollo

## 1.2.7 Desarrollo de las pruebas

Se mostrarán los resultados de las pruebas diseñadas en el Capítulo 6 y qué consecuencias han tenido sobre el sistema final.

## 1.2.8 Manuales del sistema

Manuales destinados al usuario que use la aplicación y al desarrollador que quiera implementar nueva funcionalidad o modificar la existente. En ellos se describe lo necesario para que ambos perfiles corran el sistema y varias ayudas con este.

## 1.2.9 Conclusiones y ampliaciones

Aquí haré una revisión global del proyecto comprobando si ha cumplido expectativas y objetivos propuestos en un principio.

## 1.2.10 Planificación del proyecto y presupuesto

En este capítulo detallaré la planificación y presupuesto finales. Estas ya no son unas estimaciones como tal, son más realistas, ya que se desarrollaron al final del proyecto.

## 1.2.11 Referencias bibliográficas

Referencias de los lugares desde los que he sacado los conocimientos y medios necesarios para desarrollar la aplicación.

## 1.2.12 Apéndices

Apartados con información extra que no encajan en otros capítulos.

## Capítulo 2. Introducción

Se hará una descripción más extendida de qué pretende ser el sistema, comparándolo con otras alternativas actuales en el mercado.

### 2.1 Justificación del Proyecto

En la Oferta del TFG y durante las primeras reuniones que tuve con mi tutor, César Acebal, la premisa del sistema era clara: crear una aplicación de escritorio para visualizar datos GPS que adaptase las mejores partes de las aplicaciones ya existentes en el mercado. Estos datos GPS serían rutas o tracks, entendiendo estas como una sucesión de coordenadas ordenadas en el tiempo y las cuales suelen obtenerse mediante un dispositivo móvil o un reloj.

Si eres un usuario al que le gusta realizar rutas de senderismo y posteriormente analizarlas, te habrás encontrado con que las soluciones existentes proveen herramientas muy útiles para visualizar y gestionar tus rutas, pero suelen faltarles alguna característica concreta que las hace incompletas. Quizás una te muestra mucha información calculada de los datos importados, pero no te deja ver varias rutas en un mismo mapa. Quizás otra te permita organizar tus rutas en diferentes niveles de organización (carpetas o listas), pero tiene una interfaz poco cuidada e intuitiva. O quizás en otra, puedas hacer todo lo anterior, pero solo te permita visualizar rutas en mapas básicos.

Esta es la idea que hace nacer el proyecto, tener un repositorio común con todo tu historial de rutas organizado, pudiendo visualizar una o varias a la vez y mostrando información detalla de las mismas en mapas diferentes.

Más allá del propio sistema a desarrollar, quise utilizar herramientas o tecnologías durante su implementación que hubiésemos usado poco o nada durante la carrera, que me llamasen la atención y que su uso estuviese extendido dentro del mercado del desarrollo de software.

## PICOS GÚA Y ESCOBÍN

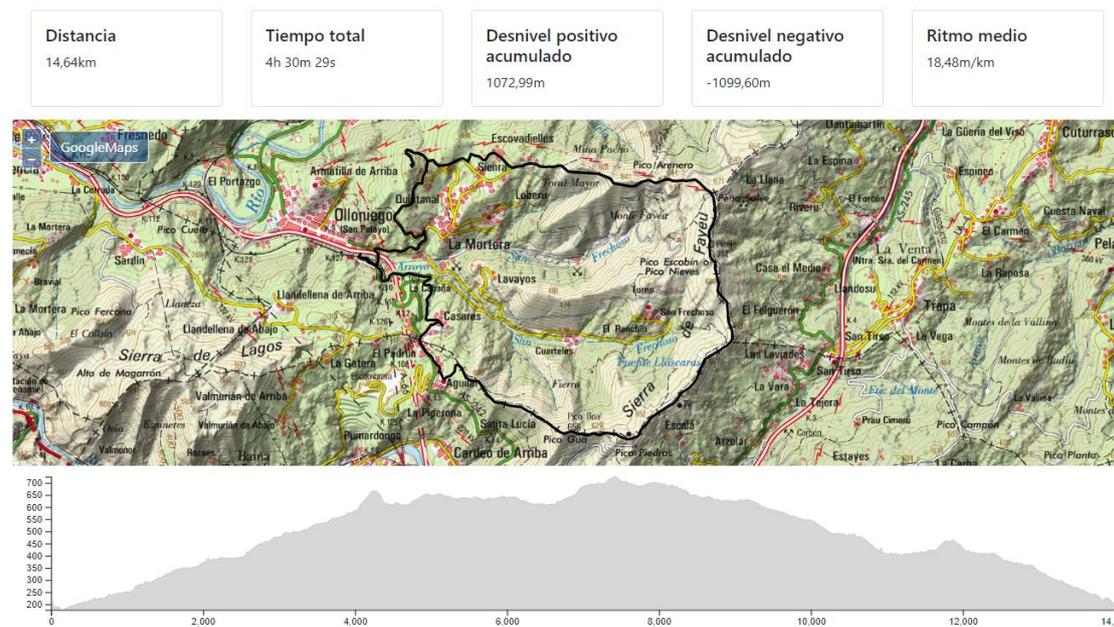


Ilustración 2.1 Datos de una ruta

## 2.2 Objetivos del Proyecto

En este apartado describiré, de forma general, los objetivos planificados. Tanto los relativos al sistema, como los relativos al proyecto, entendiendo por proyecto el TFG en su conjunto y no solo el producto final.

### 2.2.1 Importar datos GPS y mostrarlos

Este es el objetivo principal del sistema. Mi tutor hacía la siguiente descripción en la Oferta de Trabajo: *El objetivo del trabajo es crear una aplicación que permita a un usuario importar en ella todos los recorridos que desee a partir de ficheros GPS (u otros formatos equivalentes) y mostrarlos en un mapa, así como proporcionar información detallada de los mismos y realizar acciones comunes (...).*

De esta definición, obtuvimos varios subobjetivos:

- Importar datos GPS. Independientemente de la extensión del archivo en donde tengamos guardados nuestros recorridos, poder importarlos al sistema.
- Mostrar los datos representados en un mapa. Tras importar las rutas, poder visualizarlas en un mapa.
- Mostrar la información resumida de esos datos. Mostrar información básica de la ruta como distancia total, tiempo total, desnivel acumulado u otros datos.
- Realizar acciones sobre esos datos. Poder modificar la ruta añadiéndole Waypoints e, *idealmente, permitir realizar algunas operaciones sencillas de edición de los mismos,*

como eliminar ciertos segmentos, unir o dividir “tracks”, etcétera. Este es otro extracto de la Oferta de trabajo.

## 2.2.2 Seleccionar varios tipos de mapas

Otro requisito fundamental descrito en la Oferta de Trabajo como: (...) *poder seleccionar entre distintos mapas. No solo los típicos (...), sino también (y particularmente) los mapas proporcionados por el Instituto Geográfico Nacional de España.*

Dar la posibilidad al usuario de escoger entre diferentes mapas para visualizar sus rutas y, como mínimo, que uno de ellos fuese el proporcionado por el Instituto Geográfico Nacional.

## 2.2.3 Mostrar varios recorridos en un mismo mapa

Este objetivo se describía como: *En concreto, muchas (otras aplicaciones) se centran en un único recorrido, no pudiendo mostrar dos a la vez sobre el mismo mapa.*

El usuario debe poder ver dos o más rutas sobre un mismo mapa, pudiendo compararlas entre sí.

## 2.2.4 Varios niveles de organización

Los niveles de organización nacen de la necesidad de visualizar varias rutas a la vez en un mismo mapa. Esto permite almacenar varias rutas en una misma lista o carpeta, permitiendo personalizar la organización de nuestras rutas.

Un ejemplo real sería tener varias rutas en la aplicación con datos de los Picos de Europa y guardar estas rutas en una lista llamada “Rutas por Picos”. Cuando visualicemos esta lista en el mapa, podremos ver, con un solo clic, las rutas que hayamos escogido.

Este concepto es recurrente en aplicaciones donde se puedan almacenar varios objetos. Por ejemplo, Spotify es una aplicación para reproducir música vía streaming, donde podemos crear una “playlist”, guardar canciones en ella y reproducir posteriormente las canciones de esa lista. También existen aplicaciones como ToDo, en donde puedes definir varias tareas, agruparlas bajo una misma lista e ir completándolas (laborales del hogar, recados o libros que quiero leer).

## 2.2.5 Principios SOLID

Los principios SOLID (<https://profile.es/blog/principios-solid-desarrollo-software-calidad/>), definen las bases del desarrollo de un “buen” código. Como le expliqué a mi tutor en una reunión: “En vez de hacer diez funcionalidades, he preferido hacer seis y que al supuesto desarrollador que le tocase hacer la séptima, le resultase sencillo implementarla”. Alternativamente, también he tratado de seguir pautas definidas bajo el término “Clean

Code". Algunas de estas son: nombre claros y concisos, comentar lo necesario o funciones pequeñas.

Es decir, crear un código fácil de entender, fácil de modificar y fácil de escalar.

Este objetivo me lo autoimpuse, ya que cuando tuve las primeras reuniones con mi tutor, no fue una de sus peticiones. Durante el desarrollo lo hablé con él y estuvimos de acuerdo con la perspectiva tomada.

## 2.2.6 Aprender nuevas tecnologías

Este es otro objetivo externo al sistema y autoimpuesto. Como el proyecto propuesto solo estaba atado a tecnologías que trabajen con datos geoespaciales y siendo este proyecto el último punto de aprendizaje de la carrera, podía tomarme la libertad de usar tecnologías extendidas en el mercado de desarrollo web, que pudiesen ayudarme en mi futuro laboral. Más allá de ver el TFG como un fin en sí mismo, quise usarlo como un medio que me permitiese ver otras alternativas tecnológicas a las impartidas en la escuela y así, en un futuro cercano, tener más opciones que se adapten a mis ambiciones laborales.

Esto ha derivado en usar mayoritariamente tres tecnologías: Electron, [React](#) orientado a clases y [Typescript](#). Por separado existe mucha documentación, pero apenas he encontrado sistemas que usen las tres conjuntamente.

## 2.3 Estudio de la Situación Actual

En este capítulo, analizaré la situación actual del mercado de aplicaciones web/escritorio cuyo objetivo sea visualizar rutas sobre un mapa y presentar información de estas. Para ello, analizaré las principales aplicaciones de referencia: Garmin, Strava y Wikiloc. Por último, haré una comparación entre ellas y mi sistema.

Aunque las alternativas provean más funcionalidades de las aquí evaluadas, solo se tomarán en cuenta las respectivas a la importación, visualización y gestión de rutas propias. Por ejemplo, funcionalidades relacionadas con redes sociales o previsión meteorológica, no serán evaluadas.

Solo se evaluará el contenido gratuito.

### 2.3.1 Evaluación de Alternativas

A continuación, se presentarán las alternativas estudiadas. La estructura seguida tiene como objetivo presentar las ventajas y desventajas principales, sin adentrarse en funcionalidades concretas.

#### 2.3.1.1 Garmin basecamp

Garmin basecamp (<https://www.garmin.com/es-ES/software/basecamp/>) se define como: *Utiliza BaseCamp para planificar tu próxima aventura de senderismo, ciclismo, motociclismo, conducción por carretera o todoterreno. Puedes ver mapas, planificar rutas y marcar waypoints y tracks desde tu ordenador para después transferirlos a tu dispositivo.*<sup>1</sup>

---

<sup>1</sup> Software BaseCamp™. (2022, 25 de mayo). Garmin. <https://www.garmin.com/es-ES/software/basecamp/>

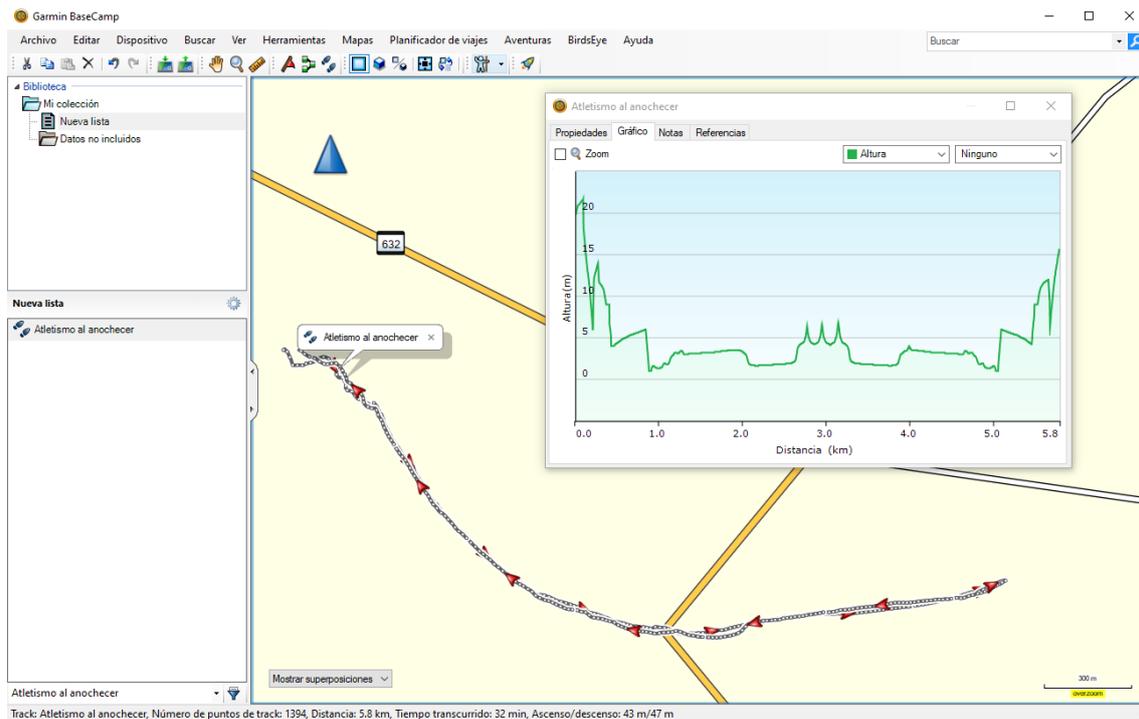


Ilustración 2.2 Interfaz de BaseCamp

### 2.3.1.1.1 Puntos fuertes

Los puntos fuertes más relevantes encontrados en BaseCamp son los siguientes:

- Permite importar rutas en formatos .GPX, .TCX, .KML y otros.
- Permite definir waypoints
- Permite ver varias rutas en un mismo mapa
- Permite hacer un seguimiento de la ruta poniendo el ratón sobre la gráfica.

### 2.3.1.1.2 Puntos débiles

Los puntos débiles más relevantes encontrados en BaseCamp son los siguientes:

- Provee un único mapa muy básico y si quieres usar otros mapas deber importarlos o comprarlos en su tienda.
- Interfaz poco intuitiva

### 2.3.1.2 Strava

Strava ([Strava | Seguimiento de carreras y entrenamientos en la red social para atletas](#)) se define como: *Strava es una red social basada en Internet y GPS enfocada a deportistas como pueden ser ciclistas y corredores y una aplicación de seguimiento GPS deportiva.*<sup>2</sup>

<sup>2</sup> Strava. (2022, 20 de mayo). Wikipedia. <https://es.wikipedia.org/wiki/Strava>

Está enfocado principalmente a ser una aplicación de entrenamiento, no de senderismo.

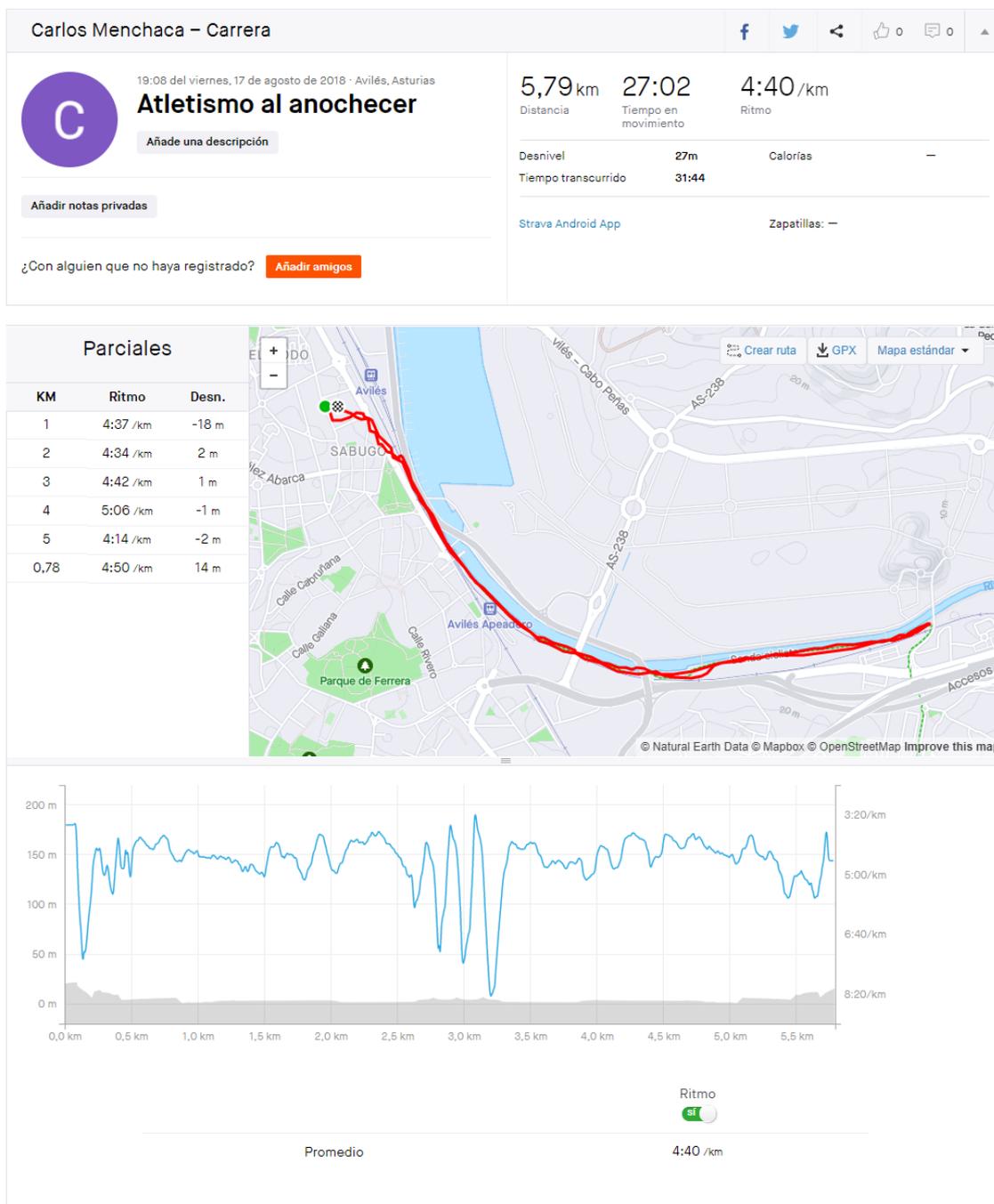


Ilustración 2.3 Interfaz de Strava

### 2.3.1.2.1 Puntos fuertes

Los puntos fuertes más relevantes encontrados en Strava son los siguientes:

- Interfaz bonita, cuidada e intuitiva.
- Permite definir un segmento dentro de la ruta.
- Permite ver ritmo, distancia y desnivel en una única gráfica.
- Permite hacer un seguimiento de la ruta poniendo el ratón sobre la gráfica.

### 2.3.1.2.2 Puntos débiles

Los puntos débiles más relevantes encontrados en Strava son los siguientes:

- No permite ver varias rutas propias a la vez sobre un mapa.
- Para importar una ruta, necesito pagar una suscripción.
- No permite definir listas.
- Solo permite visualizar OpenStreetMap y un Mapa Satélite.

### 2.3.1.3 Wikiloc

Wikiloc (<https://es.wikiloc.com>) se describe como: *Wikiloc es una aplicación que permite crear, almacenar y compartir rutas al aire libre georreferenciadas y puntos de interés de todo el mundo. Dichas rutas pueden organizarse en diferentes categorías (...)*<sup>3</sup>

The screenshot displays the Wikiloc interface for a trail titled "Senderismo al anochecer". The main map shows a route in an urban area with labels for "LOS CARBAYOS", "VALLINES", "SAN SEBASTIAN", and "LA QUINCUA PEDRO". The route is highlighted in orange. Below the map is an elevation profile showing a maximum altitude of 22m and a minimum of 1m over a distance of 5.79 km. On the right side, there is a summary box with the following data:

<b>Descarga</b>	
Autor: MrMenchaca	
1 2 1 0	
Añade compañeros de ruta	
Distancia 5,79 km	Desnivel positivo 7 m
Dificultad técnica Moderado	Desnivel negativo 7 m
Altitud máxima 22 m	Trailrank 8
Altitud mínima 1 m	Tipo de ruta Circular
Tiempo 31 minutos	
Coordenadas 1394	
Fecha de subida 20 de mayo de 2022	
Fecha de realización agosto 2018	

Ilustración 2.4 Interfaz de Wikiloc

### 2.3.1.3.1 Puntos fuertes

Los puntos fuertes más relevantes encontrados en Wikiloc son los siguientes:

- Permite visualizar las rutas en varios mapas como OpenStreetMap, Apple Maps, Mapa Ráster del IGN y otros.
- Permite importar rutas en formatos .GPX, .TCX, .KML y otros.
- Permite hacer un seguimiento de la ruta poniendo el ratón sobre la gráfica.

<sup>3</sup> ¿Qué es Wikiloc? y su importancia para las empresas de Turismo Activo. (2022, 20 de mayo). Urquiabas. <https://blog.urquiabas.com/la-importancia-de-wikiloc/>

- Permite definir listas.

### 2.3.1.3.2 Puntos débiles

Los puntos débiles más relevantes encontrados en Wikiloc son los siguientes:

- No aparece ninguna información sobre el ritmo.
- No puedes definir un segmento de la ruta.
- No permite importar una ruta ya existente en Wikiloc.
- Poco intuición para mostrar varias rutas a la vez.

## 2.3.2 Comparativa

La siguiente tabla presenta las alternativas estudiadas junto con el sistema desarrollado para poder hacerse una idea global.

Funcionalidad	BaseCamp	Strava	Wikiloc	Toure
Importar archivos .gpx	X		X	X
Importar archivos .tcx	X		X	X
Ver rutas sobre GoogleMaps				X
Ver rutas sobre Mapa Ráster del IGN			X	X
Ver rutas sobre OpenStreetMap		X	X	
Ver varias rutas sobre un mismo mapa	X		X	X
Guardar rutas en listas	X		X	X
Añadir waypoints	X			
Seleccionar un segmento		X		
Perfil de altitud	X		X	X
Seguimiento de la ruta con el perfil de altitud	X	X	X	
Datos sobre el ritmo	X	X		X

Tabla 2.1 Comparativa de alternativas

## Capítulo 3. Aspectos Teóricos

En esta sección describiré las características y ventajas que me han aportado las principales tecnologías usadas en el proyecto. Para un descripción breve y teórica de estas y otras tecnologías usadas, revisar el [Capítulo 7.2 Lenguajes de Programación](#). Igualmente, en cada apartado se hará referencia al homónimo del Capítulo 7.2

Además, se hará una explicación global del dominio del sistema para el lector que no tenga experiencia con este tipo de aplicaciones.

### 3.1 Electron

Mi tutor me propuso la idea de hacer “una aplicación local hecha con tecnologías web”. Revisando, vi que [Electron](#) encajaba con la idea que teníamos en mente por los siguientes puntos:

- No necesito un servidor externo para usar la aplicación. Electron está construido sobre [Node.js](#), que provee herramientas para desplegar un servidor en local. Por defecto provee una configuración inicial por lo que es muy sencillo desplegar la aplicación.
- Desarrollar el sistema sobre tecnologías web. El otro pilar sobre el que se construye Electron es [Chromium](#), que provee la interfaz necesaria para desarrollar mi propio sistema con Typescript, React y CSS.
- Es multiplataforma, compatible con Mac, Windows y Linux
- Actualmente es popular y tiene una gran proyección, pues lo usan otras aplicaciones como Spotify, Visual Studio Code, Skype, WhatsApp o Discord.

### 3.2 GoogleMaps Javascript API

Uno de los dos mapas usados para representar las rutas en mi sistema es GoogleMaps, el cual obtengo mediante [GoogleMaps Javascript API](#). La razón de peso que me hizo introducir este mapa fue el gran porcentaje de usuarios que están familiarizados con ella, con respecto a otros mapas interactivos similares.

### 3.3 NeDB

La elección de usar [NeDB](#), surge posteriormente a la elección de usar una [base de datos documental](#). Escogí usar una base de datos documental por los siguientes puntos:

- Es muy flexible. Esto me ayuda en el desarrollo ya que los datos que necesito almacenar sobre los modelos cambian con facilidad.

- No necesito la infraestructura que me proporcionan otros tipos de bases de datos. Al tener únicamente tres modelos y dos relaciones, no necesito hacer una inversión en definir la estructura de estas.
- Aprender una nueva tecnología. Ya que durante la carrera se explican, mayoritariamente, bases de datos relacionales, quería aprender otra alternativa.

Tras esta elección, estuve buscando una opción dentro de las bases de datos documentales que se ajustase a mis necesidades y di con NeDB. Lo que me hizo escogerla fue lo siguiente:

- Está pensada para manejar pocos datos. Esta escrita en Javascript y no tiene ningún tipo de dependencia binaria, lo que la hace muy ligera y perfecta a mi sistema.
- El lenguaje es muy similar a MongoDB, que es otra base de datos documental usada brevemente en la carrera.

## 3.4 IGN y OpenLayers

Otro de los mapas usados para representar información geoespacial es el proporcionado por el Instituto Geográfico Nacional. El IGN provee diferentes mapas en formato WMS/WMTS, con muchos tipos de información. Algunos ejemplos son: Información sísmica, Planimetrías, Redes geodésicas o La primera edición de los mapas topográficos nacionales. Todos ellos son de libre uso y se pueden consultar en el siguiente enlace: <https://www.ign.es/web/ign/portal/ide-area-nodo-ide-ign>. El que he usado en este proyecto es Cartografía Ráster, un mapa físico que principalmente muestra el relieve terrestre y también muestra datos básicos relacionados con núcleos de población (pueblos, ciudades, villas, ...).

Para poder visualizar mapas WMS/WMTS, se necesita un intérprete. Dentro de estos, escogí OpenLayers por los siguientes puntos:

- El IGN cuenta con una API llamada CNIG ([GitHub - IGN-CNIG/API-CNIG](#)), donde se usa OpenLayers para representar sus propios mapas.
- Refiriéndonos a aplicaciones que muestren datos WMS/WMTS de código libre, OpenLayers junto con Leaflet, son las opciones más populares.

## 3.5 React

La principal razón que me llevó a usar un framework front-end, es la gran proyección que este tipo de tecnologías aportan de cara a mi futuro profesional. Dentro de estos, me decanté por React por las siguientes razones:

- Se basa en encapsular funcionalidad en componentes, pudiendo reutilizarlos en diferentes vistas. Esto encaja muy bien con el objetivo que definí de basarme en los Principios SOLID durante el desarrollo.

- Es uno de los frameworks más populares, si no el más popular, dentro del mercado (Fuente Stackoverflow<sup>4</sup>). En este punto se desmarca de otro de sus competidores, Vue, con gran diferencia.
- Curva de aprendizaje. La curva mantendría una pendiente continua, no necesita un gran aprendizaje inicial como puede pasarle a Angular, otro de sus principales competidores.

Dentro de React, existen dos posibles estructuras para definir los componentes: funciones o clases. Escogí definirlos como clases, para mantener la estructura usada en el resto del código (back-end). Esta elección dio pie al problema [7.4.1.2 Mantener el centro y zoom](#). A continuación, muestro un mismo componente definido de ambas maneras.

```
export class GoogleMapsMap extends Component<MapProps, MapState>{
```

*Ilustración 3.1 Componente de clase*

```
export function GoogleMapsMap(props: MapProps): Component<MapProps, MapState>{
```

*Ilustración 3.2 Componente funcional*

## 3.6 Typescript

La elección de usar [Typescript](#), surge posteriormente a la elección de usar [Javascript](#). En primera instancia, había escogido Javascript por los siguientes puntos:

- Es uno de los lenguajes de desarrollo más usados en el ámbito web (Fuente TIOBE<sup>5</sup>).
- Estoy familiarizado con él, ya que es uno de los principales lenguajes dados en la carrera.

Tras esta elección, me di cuenta de que en la asignatura Arquitectura del Software, habíamos usado Typescript y revisé esta opción. Me decanté por ella por los siguientes puntos:

- Añade tipos estáticos y objetos basados en clases. Es decir, paso de un lenguaje débilmente tipado (Javascript), a uno fuertemente tipado (Typescript). Dado que uno de los principales objetivos del TFG era hacer un desarrollo basándome en los [Principios SOLID](#), este aspecto me facilitaba mucho esta tarea.
- Derivado del punto anterior, al hacer una comprobación más exhaustiva de los tipos, es más fácil detectar errores en una versión temprana del desarrollo.
- En primer lugar, había pensado en usar Angular en vez de React para la parte de front-end y para ello, recomiendan usar Typescript.

<sup>4</sup>Web frameworks. (2022, 16 de mayo). Stackoverflow. <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks>

<sup>5</sup> TIOBE Index for May 2022. (2022, 16 de mayo). TIOBE. <https://www.tiobe.com/tiobe-index/>

# Capítulo 4. Planificación del Proyecto y Presupuesto Iniciales

A continuación, se presentará la planificación inicial temporal y presupuestal del proyecto en su conjunto.

## 4.1 Planificación Inicial

Para planificar el proyecto, hemos decidido basarnos en metodologías ágiles, concretamente en SCRUM. Una metodología ágil se define como: *El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.*<sup>6</sup> Al ser una única persona el equipo de desarrollo y otra el cliente, hemos tenido que adaptar esta metodología a nuestro propio marco de trabajo, por lo que el lector notará que faltan ciertos conceptos.

La planificación explicada es sobre el desarrollado el sistema. Paralelamente a este desarrollo, se ha ido documentando el presente documento.

### 4.1.1 Product Backlog

El Product Backlog se define como: *El product backlog (o pila de producto) es un listado de todas las tareas que se pretenden hacer durante el desarrollo de un proyecto.*<sup>7</sup> La idea es definir unas historias de usuario genéricas y cuando se introduzcan en cada sprint, modificarlas y extenderlas con el cliente. No se ha realizado una previsión temporal de las historias por falta de conocimiento y experiencia sobre las tecnologías y conceptos usados. Adicionalmente, las tareas están ordenadas por orden de prioridad, de más prioritaria a menos prioritaria.

Se ha creado el Product Backlog inicial a partir de la Oferta del TFG y de la primera reunión con mi tutor, César Acebal. El resultado es el siguiente:

Product backlog	
1	Aprender y practicar las tecnologías empleadas en el proyecto
2	Configuración inicial de la aplicación
3	Visualizar mapa GoogleMaps e IGN
4	Visualizar varias rutas en los mapas
5	Analizar, escoger y comenzar a usar una BD
6	Conversores de los datos guardados en BD a los necesitados por los mapas (.gpx o .kml,

<sup>6</sup> Desarrollo ágil de software. (2022, 02 de junio). Wikipedia. [https://es.wikipedia.org/wiki/Desarrollo\\_ágil\\_de\\_software](https://es.wikipedia.org/wiki/Desarrollo_ágil_de_software)

<sup>7</sup> Scrum: ¿Qué es el Product Backlog?. (2022, 03 de junio). PYM. <https://programacionymas.com/blog/scrum-product-backlog>

	por ejemplo) y viceversa
7	Sistema de usuarios
8	Visualizar varios datos de una ruta
9	Modificar los datos básicos de una ruta
10	Extender la funcionalidad de modificar una ruta (Añadir waypoints o seleccionar un segmento)
11	Organización de las rutas (carpetas, listas, etiquetas...)
12	Mejoras en la interfaz
13	Finalizar la documentación

Tabla 4.1 Product Backlog inicial

## 4.1.2 Sprints

Sprint se define como: *Sprint es el nombre que va a recibir cada uno de los ciclos o iteraciones que vamos a tener dentro de un proyecto Scrum.*<sup>8</sup>

Se han planificado los siguientes Sprints:

Sprint 1	
<b>Inicio</b>	Semana del 1 al 7 de noviembre de 2021.
<b>Duración</b>	3 meses.
<b>Sprint review</b>	Semana del 31 de enero al 4 de febrero de 2022.
<b>Tareas</b>	1. Aprender y practicar las tecnologías empleadas en el proyecto. 2. Configuración inicial de la aplicación.

Tabla 4.2 Sprint 1 inicial

Sprint 2	
<b>Inicio sprint</b>	Semana del 31 de enero al 4 de febrero de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 28 de febrero al 4 de marzo de 2022.
<b>Tareas</b>	3. Visualizar mapa GoogleMaps e IGN 4. Visualizar varias rutas en los mapas 5. Analizar, escoger y comenzar a usar una BD

Tabla 4.3 Sprint 2 inicial

Sprint 3	
<b>Inicio</b>	Semana del 28 de febrero al 4 de marzo de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 4 al 8 de abril de 2022.
<b>Tareas</b>	6. Conversores de los datos guardados en BD a los necesitados por los mapas (.gpx o .kml, por ejemplo) y viceversa 7. Sistema de usuarios

<sup>8</sup> Qué es un Sprint de Scrum. (2022, 03 de junio). OpenWebinars. <https://openwebinars.net/blog/que-es-un-sprint-scrum/>

8. Visualizar varios datos de una ruta
9. Modificar los datos básicos de una ruta

Tabla 4.4 Sprint 3 inicial

Sprint 4	
<b>Inicio</b>	Semana del 4 al 8 de abril de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 2 al 6 de mayo de 2022.
<b>Tareas</b>	10. Extender la funcionalidad de modificar una ruta (Añadir waypoints o seleccionar un segmento) 11. Organización de las rutas (carpetas, listas, etiquetas...) 12. Mejoras en la interfaz

Tabla 4.5 Sprint 4 inicial

Sprint 5	
<b>Inicio</b>	Semana del 2 al 6 de mayo de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 6 al 12 de junio de 2022.
<b>Tareas</b>	13. Finalizar la documentación

Tabla 4.6 Sprint 5 inicial

## 4.2 Presupuesto Inicial

Para el presupuesto se han tomado las siguientes consideraciones:

- El desarrollo durará 7 meses y el tiempo se repartirá de la siguiente manera:
  - Sprint 1. 6 horas semanales durante 3 meses
  - Sprint 2, 3 y 4. 10 horas semanales durante 3 meses
  - Sprint 5. 14 horas semanales durante 1 mes
- El único miembro del equipo de desarrollo cobrará 12,5 €/hora
- Calculo final con un beneficio del 12%

### 4.2.1 Presupuesto de costes

A continuación, se desglosará el presupuesto de costes:

Capítulo	Concepto	Cantidad	Coste/Mes(€)	Horas	Coste Total (€)
<b>C.1</b>	<b>Costes indirectos</b>				<b>3499</b>
<b>C.1.1.</b>	<b>Licencias</b>				<b>194</b>
<b>C.1.1.1</b>	Microsoft Office	1	7		49

C.1.1.2	Microsoft Windows 10	1			145
C.1.1.3	GoogleMaps API	1			0
C.1.1.4	Visual Studio Code	1			0
<b>C.1.2.</b>	<b>Equipamiento físico</b>				<b>1100</b>
C.1.2.1	Ordenador	1			900
C.1.2.2	Material de oficina	1			200
<b>C.1.3</b>	<b>Instalaciones</b>				<b>2205</b>
C.1.3.1	Alquiler instalaciones		175		1225
C.1.3.2	Luz		40		280
C.1.3.3	Agua		10		70
C.1.3.4	Gas		30		210
C.1.3.5	Internet		60		420
<b>C.2</b>	<b>Costes directos</b>				<b>3175</b>
<b>C.2.1</b>	<b>Sprint 1</b>			<b>78</b>	<b>975</b>
C.2.1.1	Reuniones con el cliente			2	25
C.2.1.2	Análisis			10	125
C.2.1.3	Diseño			4	50
C.2.1.4	Estudio y elección de tecnologías			62	775
<b>C.2.2</b>	<b>Sprint 2</b>			<b>40</b>	<b>500</b>
C.2.2.1	Reuniones con el cliente			1	12,5
C.2.2.2	Análisis			4	50
C.2.2.3	Diseño			6	75
C.2.2.4	Estudio y elección de tecnologías			8	100
C.2.2.5	Desarrollo			19	237,5
C.2.2.6	Documentación			2	25
<b>C.2.3</b>	<b>Sprint 3</b>			<b>40</b>	<b>500</b>
C.2.3.1	Reuniones con el cliente			1	12,5

C.2.3.2	Análisis			3	37,5
C.2.3.3	Diseño			4	50
C.2.3.4	Estudio y elección de tecnologías			4	50
C.2.3.5	Desarrollo			26	325
C.2.3.6	Documentación			2	25
C.2.4	<b>Sprint 4</b>			<b>40</b>	<b>500</b>
C.2.4.1	Reuniones con el cliente			1	12,5
C.2.4.2	Análisis			2	25
C.2.4.3	Diseño			4	50
C.2.4.4	Estudio y elección de tecnologías			4	50
C.2.4.5	Desarrollo			25	312,5
C.2.4.6	Documentación			2	25
C.2.4.7	Plan de pruebas			2	25
C.2.5	<b>Sprint 5</b>			<b>56</b>	<b>700</b>
C.2.5.1	Reuniones con el cliente			4	50
C.2.5.2	Desarrollo			2	25
C.2.5.3	Documentación			40	400
C.2.5.4	Plan de pruebas			10	125

#### 4.7 Presupuesto de costes inicial

<b>Coste del proyecto</b>	<b>6674</b>
<b>Beneficio (12%)</b>	<b>800,88</b>
<b>Total</b>	<b>7474,88</b>

#### 4.8 Resumen de presupuesto de costes con beneficio inicial

### 4.2.2 Presupuesto del cliente

La siguiente tabla expone el presupuesto que se le mostrará al cliente:

Descripción	Coste
<b>Licencias</b>	217,28

<b>Oficina</b>	3701,6
<b>Análisis y estudio del proyecto</b>	1484
<b>Diseño</b>	252
<b>Desarrollo</b>	1008
<b>Pruebas</b>	168
<b>Documentación</b>	644
<b>Subtotal</b>	<b>7474,88</b>
<b>IVA (21%)</b>	<b>1569,7248</b>
<b>Total</b>	<b>9044,6048</b>

#### 4.9 Presupuesto de cliente inicial

# Capítulo 5. Análisis

Este apartado contendrá toda la especificación de requisitos y toda la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño.

## 5.1 Requisitos del Sistema

### 5.1.1 Obtención de los Requisitos del Sistema

A continuación, se expondrán los requisitos funcionales y no funcionales acordados con el cliente, en este caso mi tutor, César Acebal. Se irán describiendo los diferentes requisitos en un esquema por niveles, manteniendo la trazabilidad gracias al índice y con una descripción clara y concisa. Se ha dividido en los siguientes módulos:

#### 5.1.1.1 Gestión de las rutas

Los siguientes requisitos se refieren a la gestión de las rutas individuales en el sistema.

- RF.1 El sistema permitirá gestionar las rutas individualmente.
  - RF.1.1 Importar rutas desde diferentes formatos.
    - RF.1.1.1 Admitirá formato .gpx 1.0.
    - RF.1.1.2 Admitirá formato .tcx.
  - RF.1.2 Eliminar una ruta del sistema.
    - RF.1.2.1 Esta acción también eliminará la ruta de las listas donde se encuentre
  - RF.1.3 Mostrar todas las rutas almacenadas en el sistema en un listado.

#### 5.1.1.2 Visualización de las rutas

Los siguientes requisitos se refieren a la visualización de las rutas en el sistema y cómo se deben mostrar sus datos.

- RF.2 El sistema permitirá ver las rutas almacenadas en la aplicación.
  - RF.2.1 El sistema permitirá ver las rutas en diferentes mapas.
    - RF.2.1.1 Mapa proporcionado por GoogleMaps.
      - RF.2.1.1.1 La capa Base.
      - RF.2.1.1.2 La capa Relieve.
      - RF.2.1.1.3 La capa Satélite.
        - RF.2.1.1.3.1 Con etiquetas.
        - RF.2.1.1.3.2 Sin etiquetas.
    - RF.2.1.2 Mapa proporcionado por el Instituto Geográfico Nacional.
  - RF.2.2 El sistema tendrá varias vistas para la visualización de las rutas.

- RF.2.2.1 Vista individual de una ruta.
  - RF.2.2.1.1 La ruta se podrá ver en los mapas especificados en el RF.2.1
  - RF.2.2.1.2 Mostrará varios datos de la ruta.
    - RF.2.2.1.2.1 Distancia total en kilómetros.
    - RF.2.2.1.2.2 Tiempo total en horas, minutos y segundos.
    - RF.2.2.1.2.3 Desnivel positivo acumulado en metros.
    - RF.2.2.1.2.4 Desnivel negativo acumulado en metros.
    - RF.2.2.1.2.5 Ritmo medio en minutos por kilómetro.
  - RF.2.2.1.3 Mostrar perfil de altitud.
    - RF.2.2.1.3.1 Eje Y debe mostrar el desnivel en metros.
    - RF.2.2.1.3.2 Eje X debe mostrar la distancia total en metros.
- RF.2.2.2 Vista general de todas las rutas almacenadas en el sistema.
  - RF.2.2.2.1 Las rutas se podrán ver en los mapas especificados en el RF.2.1.
  - RF.2.2.2.2 Se podrá mostrar/ocultar una ruta individual.
  - RF.2.2.2.3 Se podrán mostrar/ocultar todas las rutas a la vez.
  - RF.2.2.2.4 Se podrán mostrar/ocultar las rutas de una lista a la vez.

### 5.1.1.3 Gestión de las listas

Los siguientes requisitos se refieren a la gestión de las listas y cómo deben interaccionar con las rutas.

- RF.3 El sistema permitirá gestionar listas de rutas
  - RF.3.1 Crear una lista.
  - RF.3.2 Eliminar una lista.
  - RF.3.3 Añadir una ruta a una o varias listas.
  - RF.3.4 Eliminar una ruta de una lista.

### 5.1.1.4 Requisitos no funcionales

- RNF.1. El sistema debe ser una aplicación de escritorio

## 5.1.2 Identificación de Actores del Sistema

Se analizarán los posibles actores que intervendrán en los diferentes procesos del sistema. Identificamos como actor a todo elemento que interactúa con el sistema desde fuera, o que es algo o alguien sobre el cual interactúa el sistema.

### 5.1.2.1 Usuario

En nuestro sistema existe un único actor y es el usuario que utiliza el sistema. Este usuario tiene total acceso a toda la funcionalidad de la aplicación.

### 5.1.3 Especificación de Casos de Uso

A continuación, se presentará un diagrama con los casos de uso que puede llevar a cabo el único actor, ósea un usuario, dentro del sistema. Posteriormente, se describirán de forma general los casos de uso. Para un desarrollo en profundidad de estos, véase el [Capítulo 5.5 Análisis de Casos de Uso y Escenarios](#)

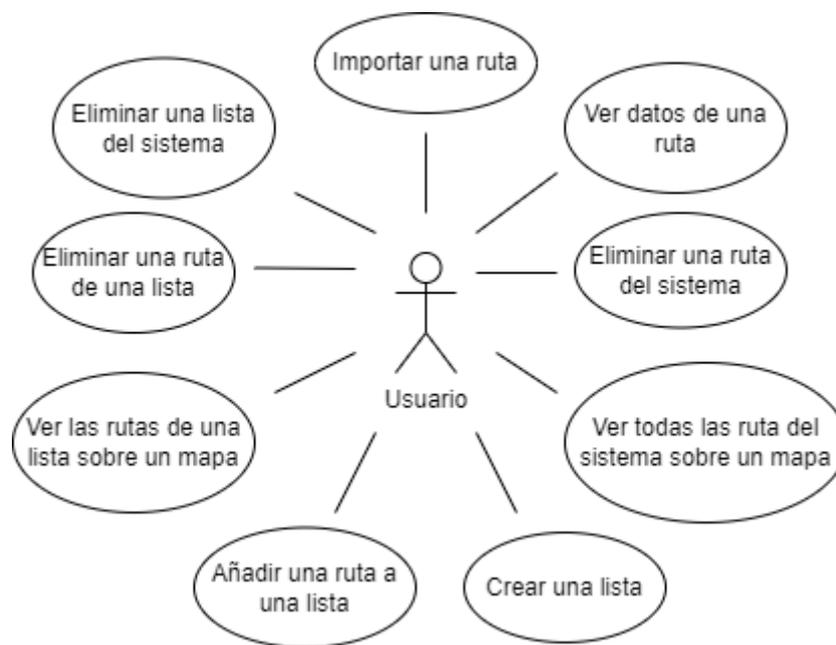


Ilustración 5.1 Diagrama con los casos de uso

<b>Nombre Del Caso de Uso</b>
Importar una ruta
<b>Descripción</b>
Un usuario desea importar en el sistema una ruta que tenga almacenada en su computadora.

Tabla 5.1 Especificación de Caso de uso, Importar una ruta

<b>Nombre Del Caso de Uso</b>
Ver datos de una ruta
<b>Descripción</b>

Un usuario desea ver datos de una de sus rutas importadas

- Datos generales (Distancia, tiempo, ritmo...)
- Su ruta en el mapa
- Perfil de altitud

*Tabla 5.2 Especificación de Caso de uso, Ver datos de una ruta*

**Nombre Del Caso de Uso**

Eliminar una ruta del sistema

**Descripción**

Un usuario desea eliminar una de sus rutas de todo el sistema.

*Tabla 5.3 Especificación de Caso de uso, Eliminar una ruta del sistema*

**Nombre Del Caso de Uso**

Ver todas las rutas del sistema sobre un mapa

**Descripción**

Un usuario desea ver sobre un mismo mapa, una o más rutas a la vez.

*Tabla 5.4 Especificación de Caso de uso, Ver todas las rutas del sistema sobre un mapa*

**Nombre Del Caso de Uso**

Crear una lista

**Descripción**

Un usuario desea crear una lista donde guardar sus rutas.

*Tabla 5.5 Especificación de Caso de uso, Crear una lista*

**Nombre Del Caso de Uso**

Añadir una ruta a una lista

**Descripción**

Un usuario desea añadir una ruta a una o varias listas.

*Tabla 5.6 Especificación de Caso de uso, Añadir una ruta a una lista*

**Nombre Del Caso de Uso**

Ver las rutas de una lista sobre un mapa

**Descripción**

Un usuario desea poder ver sobre un mismo mapa, todas las rutas de una lista.

*Tabla 5.7 Especificación de Caso de uso, Ver las rutas de una lista sobre un mapa*

**Nombre Del Caso de Uso**

Eliminar una ruta de una lista

**Descripción**

Un usuario desea eliminar una ruta de una lista, sin que la ruta se elimine del sistema.

*Tabla 5.8 Especificación de Caso de uso, Eliminar una ruta de una lista*

<b>Nombre Del Caso de Uso</b>
Eliminar una lista del sistema
<b>Descripción</b>
Un usuario desea eliminar una lista del sistema, sin que se eliminen las rutas contenidas dentro de esta.

*Tabla 5.9 Especificación de Caso de uso, Eliminar una lista del sistema*

## 5.2 Identificación de los Subsistemas en la Fase de Análisis

El objetivo de esta sección es analizar el sistema, descomponiéndolo en sistemas más pequeños (subsistemas) que faciliten su posterior análisis.

### 5.2.1 Descripción de los Subsistemas

A continuación, analizaré los diferentes subsistemas de la aplicación. Un subsistema engloba conceptos con funcionalidad común, independientemente de cómo se haga el diseño y desarrollo posterior.

#### 5.2.1.1 Subsistema Base de datos

Dado que necesito que las rutas importadas en el sistema persistan en el tiempo, para ello usamos una base de datos. La base de datos usada será NeDB, una base de datos documental embebida. Se encargará de almacenar los datos de las rutas. Al ser una base de datos embebida, estará integrada dentro de mi sistema y el acceso a ella será mediante un objeto.

Por tanto, lo más probable es que sea una clase con métodos estáticos para poder acceder a ella desde cualquier parte del sistema.

#### 5.2.1.2 Subsistema Parsers

Necesito una o varias entidades que lean los archivos que el usuario quiere importar y los interprete. A estas entidades las llamaré Parser. Un Parser es el encargado de convertir el archivo con la ruta almacenada en la computadora, a datos del modelo.

Una primera idea sería declarar una interfaz común y que de ella herede un Parser por cada extensión. Separando y aislando así la funcionalidad propia de cada extensión.

#### 5.2.1.3 Subsistema Modelo

El modelo se encargará de definir una estructura de datos para unificar las diferentes extensiones y poder tratarlas a todas por igual. Los datos del modelo son los que posteriormente se usarán en todo el sistema. Hará falta algún tipo de conversión para que los mapas puedan leerlos.

#### 5.2.1.4 Subsistema Vista

Este subsistema se estructurará con React y será el encargado de mostrar la Interfaz, es decir, las pantallas que verá y usará el usuario.

## 5.2.2 Descripción de los Interfaces entre Subsistemas

En este apartado mostraré mediante un diagrama cómo se comunicarán entre ellos los cuatro subsistemas. Se trata de tener una visión general de la estructura del proyecto y de cómo se espera que interactuarán entre sí. El diagrama es el siguiente:

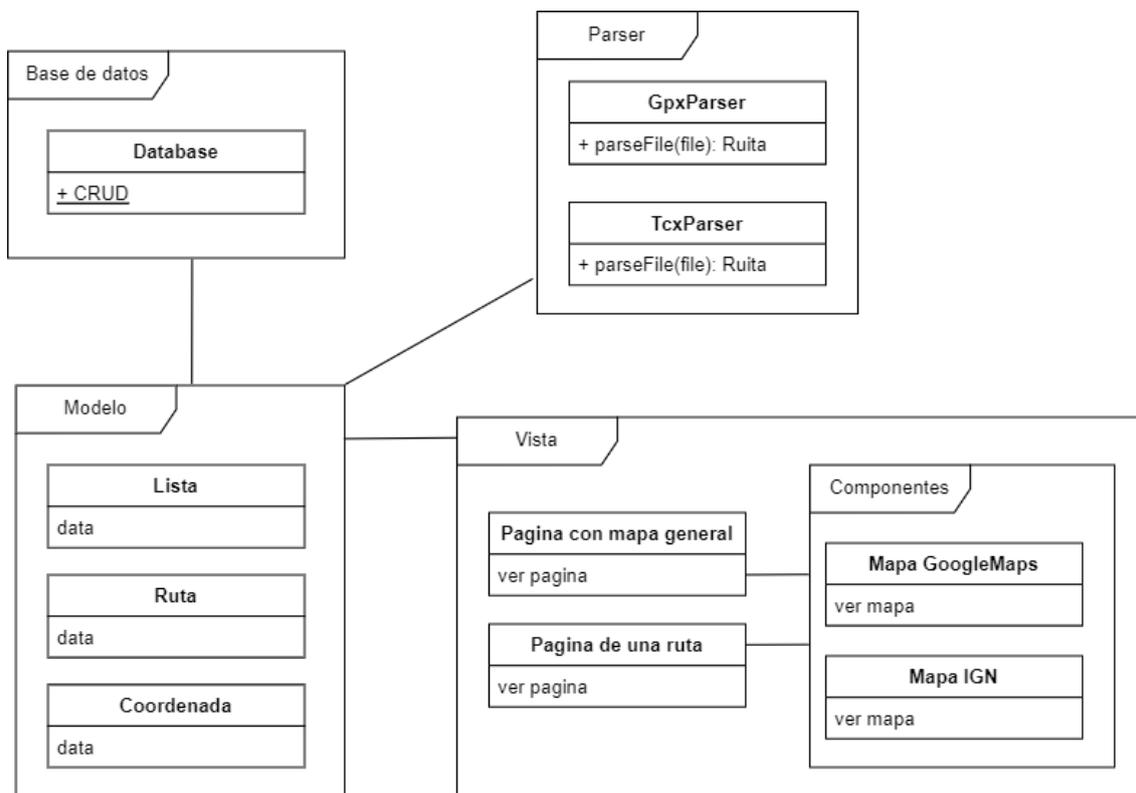


Ilustración 5.2 interacción entre los subsistemas

Las líneas indican que existe una relación entre los subsistemas.

## 5.3 Análisis de Casos de Uso y Escenarios

En este capítulo se desarrollarán los casos de uso presentados en el [Capítulo 5.2.3 Especificación de Casos de Uso](#). Estos se han obtenido a partir de los [Requisitos del Sistema](#) y trata de abarcar todos ellos. Además, dentro de los casos de uso, podremos tener diferentes escenarios, es decir, diferentes formas de lograr el objetivo descrito o fallos que puedan surgir durante la realización de este.

En el desarrollo de los siguientes casos de uso, no se determinará qué actor los realiza, ya que en mi sistema solo existe un único actor posible, un usuario.

La estructura seguida para su explicación es la siguiente:

- **Descripción:** Explica de forma general qué se quiere conseguir.
- **Requisitos:** Los Requisitos del Sistema que se ven involucrados.
- **Precondiciones:** Qué se necesita para llevar a cabo el caso de uso.
- **Postcondiciones:** Las modificaciones que sufre el sistema tras finalizar el caso de uso.
- **Flujo primario:** Una lista de eventos o acciones definidos en orden temporal explicando las diferentes fases por las que pasa la ejecución principal del caso de uso.
- **Rutas alternativas:** Una serie de movimientos que pueden causar resultados alternativos derivados del flujo de eventos. Estos resultados pueden ser correctos o erróneos.
- **Notas:** Observaciones a tener en cuenta durante el caso de uso.

### 5.3.1 Caso N.1 Importar una ruta

Importar una ruta	
Descripción	Un usuario desea importar en el sistema una ruta que tenga almacenada en su computadora.
Requisitos	RF.1.1 y RF.1.3.
Precondiciones	- Tener en la computadora archivos .gpx o .tcx
Postcondiciones	- Aparece en el listado "Todas las rutas" la ruta importada. - Se añadirá a base de datos la ruta importada.
Flujo primario	1. El usuario se va a la pantalla Rutas. 2. El usuario selecciona el botón "Seleccionar desde explorador de archivos". 3. El usuario selecciona la ruta que desea importar.
Rutas alternativas	1. Ruta alternativa 1. 1.1 El usuario se va a la pantalla Rutas. 1.2 El usuario arrastra la ruta al dropbox. 2. Ruta alternativa 2. 2.1 El usuario se va a la pantalla Rutas. 2.2 El usuario intenta importar un archivo con una extensión no permitida. 2.3 Aparece un modal notificando al usuario el error.

Notas

Tabla 5.10 Caso N.1 Importar una ruta

### 5.3.2 Caso N.2 Ver datos de una ruta

Ver datos de una ruta	
Descripción	Un usuario desea ver datos de una de sus rutas importadas.
Requisitos	RF.2.2.1 y RF.2.1.1
Precondiciones	- Tener, al menos, una ruta en el sistema.
Postcondiciones	- Acabar en la pantalla de visualizar una ruta concreta
Flujo primario	1. El usuario va a la pantalla Rutas. 2. El usuario selecciona el icono con un ojo de la ruta que desee ver.
Rutas alternativas	1. Ruta alternativa 1. 1.1 El usuario va a la pantalla Mapa. 1.2 El usuario despliega el listado con todas las rutas. 1.3 El usuario selecciona el icono con un ojo de la ruta que desee ver.
Notas	- En la pantalla de visualizar una ruta, el usuario puede cambiar entre los diferentes mapas.

Tabla 5.11 Caso N.2 Ver datos de una ruta

### 5.3.3 Caso N.3 Eliminar una ruta del sistema

Eliminar una ruta del sistema	
Descripción	Un usuario desea eliminar una de sus rutas de todo el sistema.
Requisitos	RF.1.2
Precondiciones	- Tener, al menos, una ruta en el sistema.
Postcondiciones	- La ruta se borrará de base de datos. - La ruta desaparecerá del listado "Todas las rutas".
Flujo primario	1. El usuario va a la pantalla Rutas. 2. El usuario selecciona el icono con una papelera de la ruta que desee eliminar. 3. El usuario debe confirmar la acción.
Rutas alternativas	
Notas	

Tabla 5.12 Caso N.3 Eliminar una ruta del sistema

### 5.3.4 Caso N.4 Ver todas las rutas del sistema sobre un mapa

Ver todas las rutas del sistema sobre un mapa	
Descripción	Un usuario desea ver sobre un mismo mapa, una o más rutas a la vez.
Requisitos	RF.2.1 y RF.2.2.2
Precondiciones	- Tener una o más rutas en el sistema.
Postcondiciones	- Aparecen las o las rutas pintadas en el mapa.

<b>Flujo primario</b>	<ol style="list-style-type: none"> <li>1. El usuario va a la pantalla Mapa.</li> <li>2. El usuario despliega el listado con todas las rutas.</li> <li>3. El usuario selecciona la o las rutas que desee ver.</li> </ol>
<b>Rutas alternativas</b>	
<b>Notas</b>	- El usuario puede cambiar entre los diferentes mapas.

Tabla 5.13 Caso N.4 Ver todas las rutas del sistema sobre un mapa

### 5.3.5 Caso N.5 Crear una lista

<b>Crear una lista</b>	
<b>Descripción</b>	Un usuario desea crear una lista donde guardar sus rutas.
<b>Requisitos</b>	RF.3.1
<b>Precondiciones</b>	
<b>Postcondiciones</b>	<ul style="list-style-type: none"> <li>- Se crea una nueva lista en base de datos.</li> <li>- Aparece la nueva lista en el listado de Listas.</li> </ul>
<b>Flujo primario</b>	<ol style="list-style-type: none"> <li>1. El usuario va a la pantalla Rutas.</li> <li>2. El usuario pulsa sobre el icono justo a la derecha de "Listas"</li> <li>3. El usuario introduce el nombre de la lista y pulsa en el botón Crear.</li> </ol>
<b>Rutas alternativas</b>	
<b>Notas</b>	

Tabla 5.14 Caso N.5 Crear una lista

### 5.3.6 Caso N.6 Añadir una ruta a una lista

<b>Añadir una ruta a una lista</b>	
<b>Descripción</b>	Un usuario desea añadir una ruta a una o varias listas.
<b>Requisitos</b>	RF.3.3
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Tener, al menos, una lista creada.</li> <li>- Tener, al menos, una ruta en el sistema.</li> </ul>
<b>Postcondiciones</b>	- Se añadirá la "relación" a base de datos.
<b>Flujo primario</b>	<ol style="list-style-type: none"> <li>1. El usuario va a la pantalla Rutas.</li> <li>2. El usuario pulsa sobre un icono que acompañe a la ruta (probablemente sea un +)</li> <li>3. El usuario selecciona la o las listas a las que quiere añadir la ruta y pulsa el botón Añadir.</li> </ol>
<b>Rutas alternativas</b>	
<b>Notas</b>	

Tabla 5.15 Caso N.6 Añadir una ruta a una lista

### 5.3.7 Caso N.7 Ver las rutas de una lista sobre un mapa

<b>Ver las rutas de una lista sobre un mapa</b>	
<b>Descripción</b>	Un usuario desea poder ver sobre un mismo mapa, todas las rutas de

	una lista.
<b>Requisitos</b>	RF.2.2.2.4 y RF.2.1
<b>Precondiciones</b>	- Tener una lista en el sistema con una o varias rutas dentro.
<b>Postcondiciones</b>	- Aparecen las o las rutas pintadas en el mapa.
<b>Flujo primario</b>	1. El usuario va a la pantalla Mapa. 2. El usuario pulsa sobre la lista con las rutas que desea ver.
<b>Rutas alternativas</b>	1. Ruta alternativa 1. 1.1 El usuario va a la pantalla Mapas. 1.2 El usuario pulsa en cada ruta individual que esté dentro de la lista deseada.
<b>Notas</b>	- El usuario puede cambiar entre los diferentes mapas.

*Tabla 5.16 Caso N.7 Ver las rutas de una lista sobre un mapa*

### 5.3.8 Caso N.8 Eliminar una ruta de una lista

Eliminar una ruta de una lista	
<b>Descripción</b>	Un usuario desea eliminar una ruta de una lista, sin que la ruta se elimine del sistema.
<b>Requisitos</b>	RF.3.4
<b>Precondiciones</b>	- Tener una lista con una o varias rutas dentro.
<b>Postcondiciones</b>	- Se elimina la "relación" de base de datos. - Desaparece del listado con las rutas de la lista
<b>Flujo primario</b>	1. El usuario va la pantalla Rutas. 2. El usuario va a la lista deseada (probablemente pulsando un icono). 3. El usuario selecciona el icono con una papelera de la ruta que desee eliminar de la lista.
<b>Rutas alternativas</b>	
<b>Notas</b>	- Esta acción no elimina la lista del sistema

*Tabla 5.17 Caso N.8 Eliminar una ruta de una lista*

### 5.3.9 Caso N.9 Eliminar una lista del sistema

Eliminar una lista del sistema	
<b>Descripción</b>	Un usuario desea eliminar una lista del sistema, sin que se eliminen las rutas contenidas dentro de esta.
<b>Requisitos</b>	RF.3.2
<b>Precondiciones</b>	- Tener una lista en el sistema
<b>Postcondiciones</b>	- Se elimina la lista de base de datos
<b>Flujo primario</b>	1. El usuario va a la pantalla Rutas. 2. El usuario pulsa el icono con una papelera de la lista que desee eliminar del sistema. 3. El usuario debe confirmar la acción.
<b>Rutas alternativas</b>	
<b>Notas</b>	

*Tabla 5.18 Caso N.9 Eliminar una lista del sistema*

## 5.4 Análisis de Interfaces de Usuario

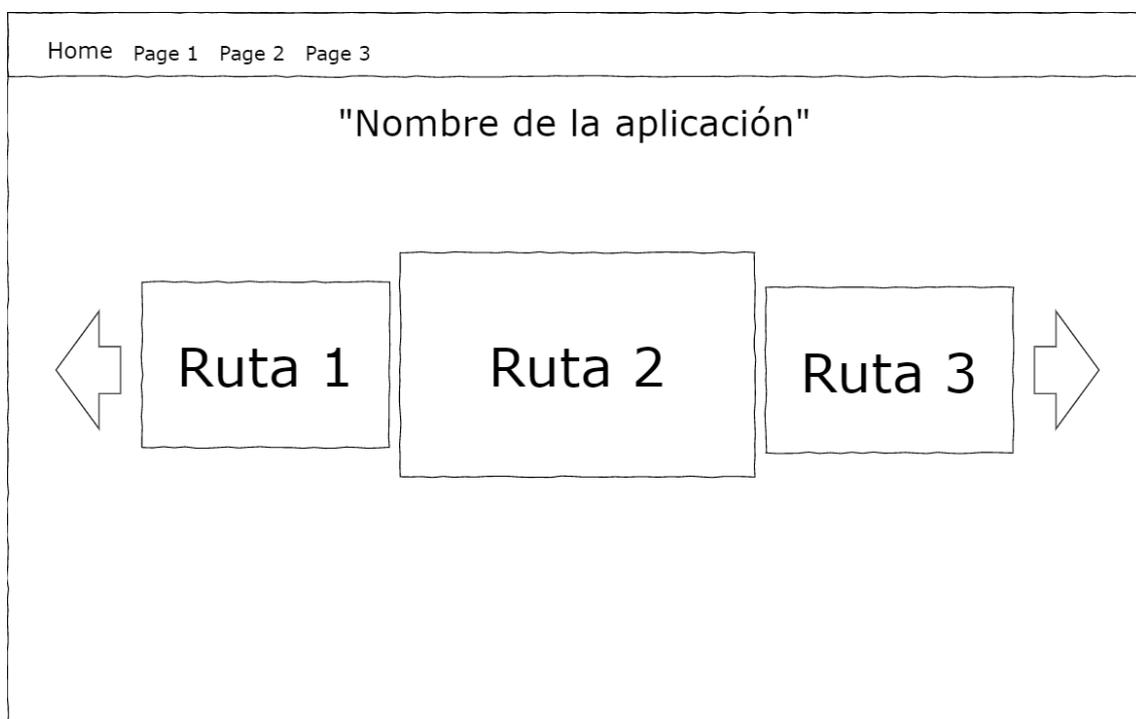
A continuación, se mostrará el resultado del análisis de la interfaz. Diseños e interacción entre ellos.

### 5.4.1 Descripción de la Interfaz

En esta sección se mostrarán los diseños iniciales de las diferentes pantallas.

En todas las pantallas, en la parte superior, aparece un prototipo de la barra de navegación.

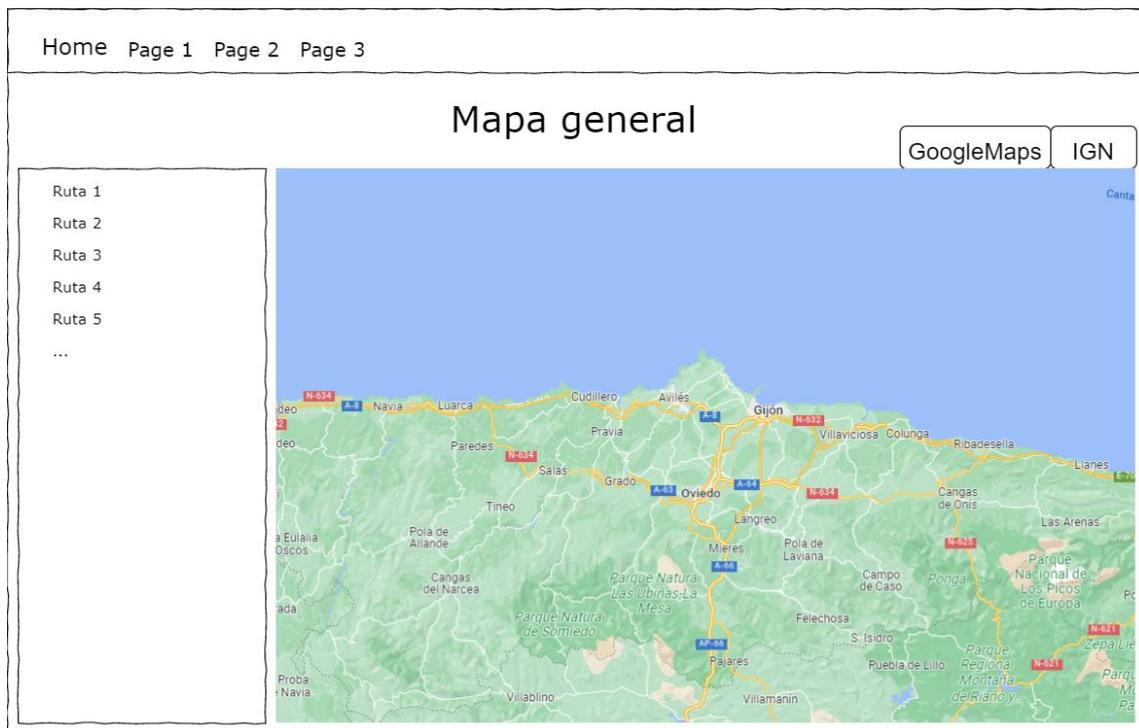
#### 5.4.1.1 Pantalla inicial



*Ilustración 5.3 Prototipo de pantalla inicial*

La pantalla inicial será una pantalla de "bienvenida", donde el usuario podrá ver y seleccionar sus rutas recientes mediante un elemento llamado "Carousel".

### 5.4.1.2 Mapa general



*Ilustración 5.4 Prototipo de la pantalla con el mapa general*

En la siguiente pantalla, el usuario podrá visualizar una o varias rutas tanto en el mapa de GoogleMaps como del IGN. Las rutas se pintarán o despintarán del mapa, en función de las que escoja en el listado izquierdo. También podrá cambiar de mapa con los botones “GoogleMaps” o “IGN”.

### 5.4.1.3 Gestión



*Ilustración 5.5 Prototipo de la pantalla para gestionar las rutas*

Esta pantalla se usa para gestionar las rutas y listas del sistema. El usuario puede importar sus rutas arrastrándolas al cuadro habilitado para ello. También puede gestionar sus listas y sus rutas. Cómo mostrar las acciones que pueda hacer sobre ellas aún están por definir.

### 5.4.1.4 Ruta individual

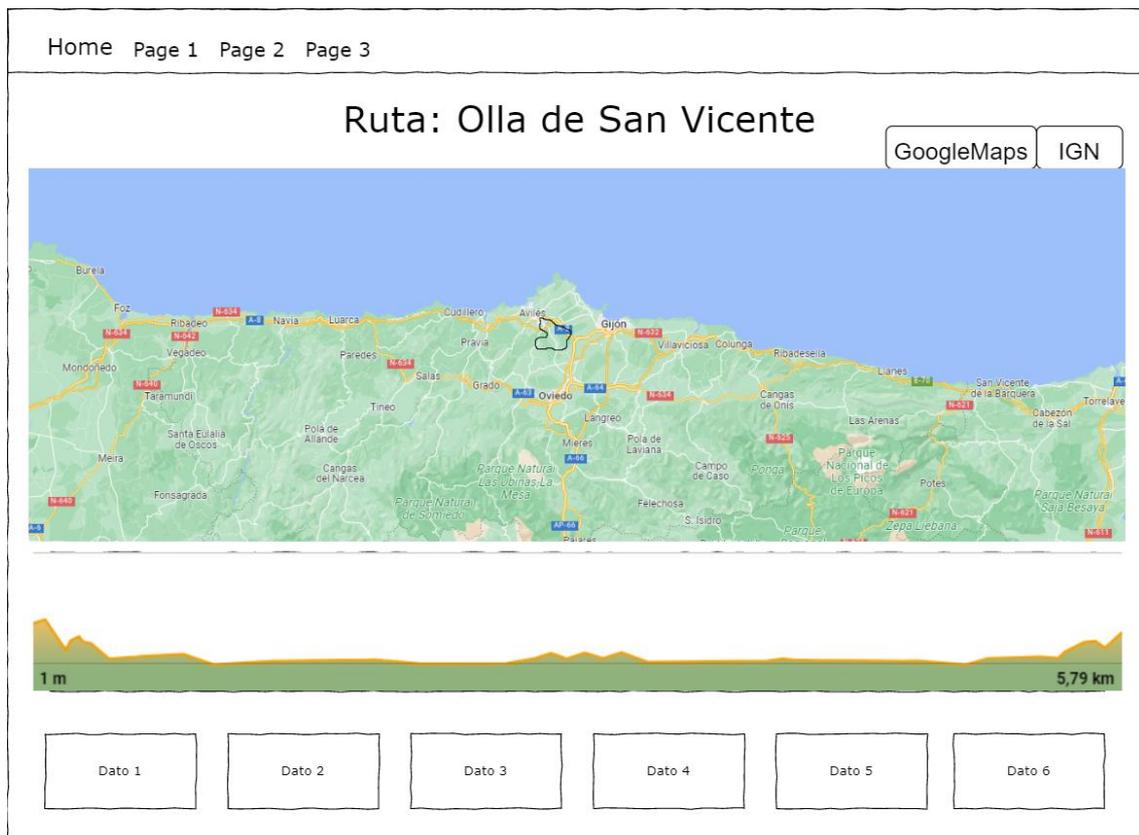


Ilustración 5.6 Prototipo de la pantalla que muestra los datos de una ruta

En esta última pantalla, el usuario podrá ver todos los datos de una ruta en concreto. Datos como distancia total, tiempo o desnivel, se mostrarán en tarjetas independientes. Adicionalmente, se mostrará el mapa y un perfil de altitud.

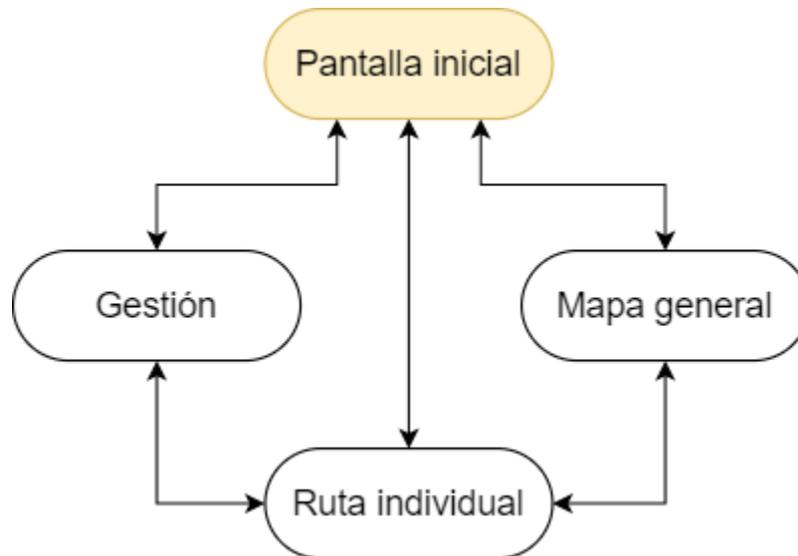
## 5.4.2 Descripción del Comportamiento de la Interfaz

En la fase de análisis no surgieron ayudas al usuario, explicando el comportamiento de la aplicación. La idea inicial era crear una buena interfaz, de tal forma que el usuario no necesitase de ayudas externas para saber qué está pasando en cada momento.

Como se habrá notado, esta sección la estoy escribiendo en una fase posterior al análisis, dándome cuenta del error que cometí. Tras pasar las pruebas desarrolladas en el [Capítulo 8. Desarrollo de las pruebas](#), he notado que los usuarios necesitaban ayudas concretas en ciertos momentos, a pesar de que las acciones que pueden hacer son reducidas. Por ejemplo, si intenta importar un fichero inválido, mostrar un modal con el error. Otro ejemplo es, cuando intenta añadir una ruta a una lista, añadir un texto de ayuda. Estos cambios se detallan en el Capítulo mencionado con anterioridad.

### 5.4.3 Diagrama de Navegabilidad

El diagrama de navegabilidad de este sistema es muy sencillo al contar con solo cuatro páginas y una barra de navegación. Básicamente, todas las pantallas son accesibles desde cualquier pantalla. Igualmente, el diagrama sería el siguiente:



*Ilustración 5.7 Diagrama de navegabilidad*

## 5.5 Especificación del Plan de Pruebas

En esta sección explicaré, de forma general, el plan de pruebas que se llevará a cabo. El plan de pruebas tiene como objetivo descubrir errores de todo tipo, para así corregirlos en la fase de implementación.

Para ello he usado los siguientes tipos de pruebas:

- **Pruebas del Sistema:** Las pruebas del sistema permiten probar el conjunto de todo el sistema, verificando así que todos sus requisitos se cumplan. Se realizará una extracción de casos de prueba a raíz de los casos de uso diseñados con anterioridad. Las pruebas se llevarán a cabo manualmente.
- **Pruebas de Usabilidad:** Las pruebas de usabilidad están enfocadas a conocer la satisfacción con el producto final de diferentes tipos de clientes. Para ello, se usarán varios cuestionarios, tratando de recoger la máxima cantidad de información posible. También se tomará una muestra diversa, con potenciales clientes y usuarios medios que tengan poco conocimiento del dominio.
- **Pruebas de Accesibilidad:** Las pruebas de accesibilidad son un subgrupo de las pruebas de usabilidad en las que los usuarios objetivo tienen discapacidades que afectan a la forma de utilizar el sistema. Aunque nuestro sistema sea una aplicación de escritorio y no web, hace uso de tecnologías web, por lo que aplicaré herramientas pensadas para probar páginas web.

## Capítulo 6. Diseño del Sistema

A continuación, se recogerá el diseño creado previo al desarrollo. La “guía” a seguir por el desarrollador (yo mismo).

### 6.1 Problemas y necesidades en el diseño

En este capítulo, explicaré problemas que surgen a la hora de intentar implementar el sistema. Problemas globales, a los que trato de dar la mejor solución posible, mediante un buen diseño. Con estas explicaciones, el lector podrá hacerse una idea global sobre el diseño y cuáles han sido los incentivos que me han llevado a hacerlo así.

Los diagramas adjuntados en cada apartado no son diagramas UML en su estricta definición, ni aparecen todos los datos necesarios. Se usan como mero apoyo en las explicaciones de cada apartado, representando en ellos lo necesario para entender la solución dada.

Durante este capítulo nombraré mucho la palabra entidad, entendiéndose como “algo” que agrupa funcionalidad. Esto pueden ser funciones, clases, clases abstractas, interfaces...

#### 6.1.1 Comunicar archivos y mapa

Al analizar el sistema, vemos que surge el problema de “comunicar” entre sí los diferentes archivos importados y los diferentes mapas para visualizar los datos de dichos archivos. Este apartado explica la solución propuesta para comunicar estos “módulos”, no al problema de tener varias extensiones de fichero, ni al problema de tener varios tipos de mapas.

La solución tomada en este sistema es común en la mayoría de sistemas que necesiten tratar con datos: crear un modelo que represente el dominio del sistema. En mi caso, crear las entidades necesarias que den forma a las rutas importadas desde los archivos.

El modelo diseñado es el siguiente:

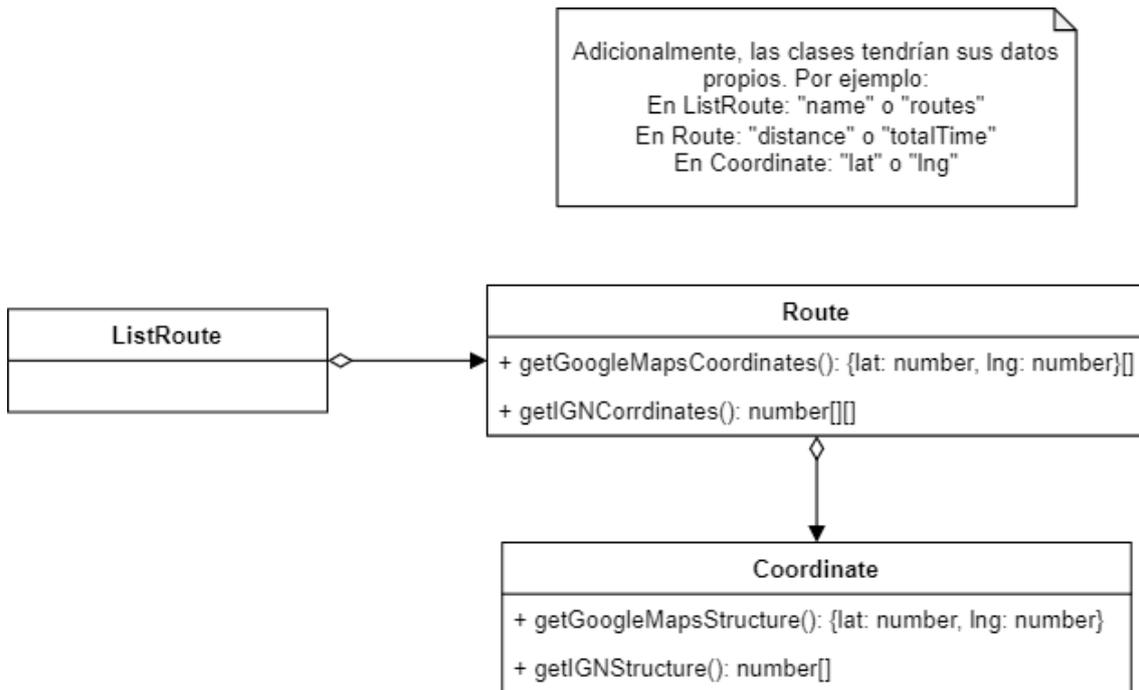


Ilustración 6.1 Estructura del modelo

Con esta solución, los archivos y los mapas tienen un punto de unión gracias a las funciones definidas en el modelo. Los archivos importados a través de los [Parsers](#) se transforman en datos del modelo y los datos del modelo son leídos por los mapas.

Para finalizar, este es el diseño que da forma a una parte del subsistema explicado en el [Capítulo 5.2.1.3 Subsistema Modelo](#)

## 6.1.2 Modelo y persistencia

Una vez definida la estructura del [modelo](#), necesitamos que los datos sean persistentes en el tiempo. Para este problema, la solución es directa: usar una base de datos. La base de datos elegida es [NeDB](#) y las razones de su elección se explican en el [Capítulo 3.3 NeDB](#).

A continuación, mostraré el diagrama que representa la estructura del modelo, con los datos almacenados en cada entidad. Cabe destacar que los datos no se “tratan” al guardarlos y obtenerlos en base de datos, por lo que el diagrama mostrado a continuación es válido tanto para base de datos como para el modelo. El diagrama es el siguiente:

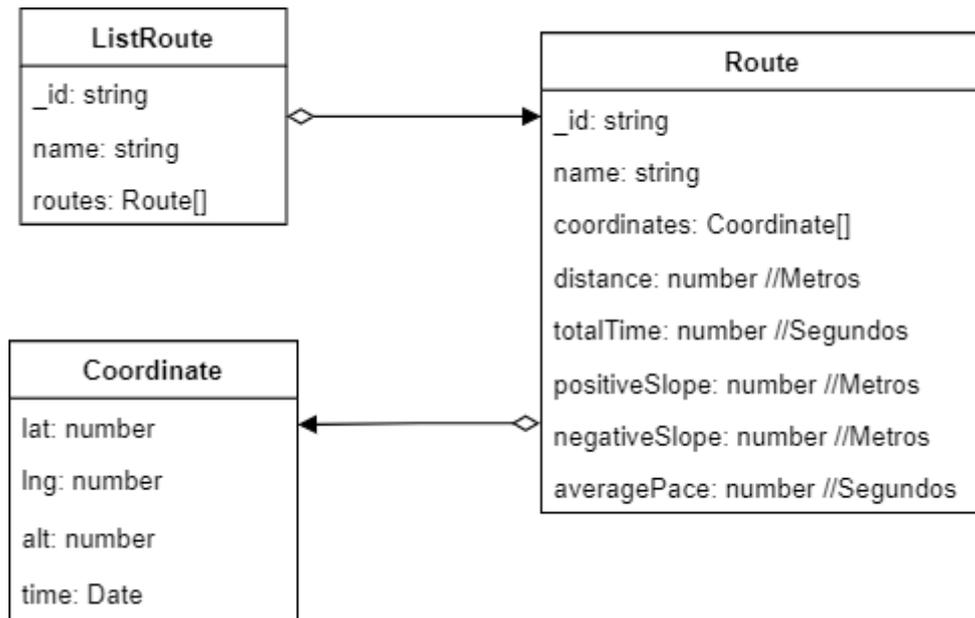


Ilustración 6.2 Datos del modelo

Una vez definida la estructura y los datos que se van a almacenar en base de datos, hay que diseñar la conexión con base de datos. NeDB provee funciones para gestionar esta conexión, por lo que decidí darles usos y agruparlas en una clase llamada Database. El diseño, que consta de una única clase, es el siguiente:

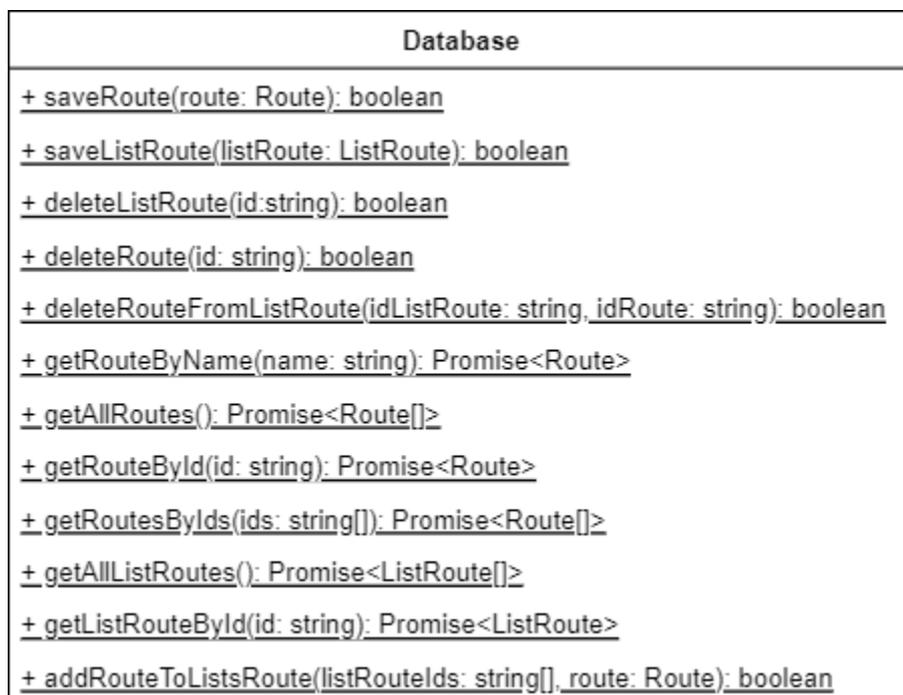


Ilustración 6.3 Métodos de acceso a base de datos

NeDB provee la clase Nedb para poder invocar a sus propias funciones. Basándome en el patrón Singleton, definí un método para obtener dicho objeto y no crear objetos innecesariamente en cada llamada a una función:

```
private static db: Nedb;

public static getInstance(){
    if(!this.db)
        this.db = new Nedb({filename: path.join('.', 'database.db'), autoload: true});
    return this.db;
}
```

*Ilustración 6.4 Objeto Nedb*

Cabe destacar cómo se transforman los datos de documentos a datos del modelo. Esto se desarrolla en el [Capítulo 7.3.8 Tipo de dato devuelto por NeDB](#).

Por último, este diseño está enlazado con el subsistema [5.2.1.1 Subsistema Base de datos](#) y el subsistema [5.2.1.3 Subsistema Modelo](#)

### 6.1.3 Importar varios tipos de ficheros

La funcionalidad raíz de la cual surge esta necesidad de diseño se puede resumir clara y concisamente: importar diferentes archivos. Como se ha explicado reiteradas veces, este es uno de los principales objetivos del proyecto, dar la posibilidad al usuario de importar diferentes extensiones dentro del sistema. Proveyendo inicialmente la funcionalidad para importar dos extensiones, pero llevar a cabo un diseño que fuese fácilmente escalable.

Debía plantear un diseño que cumpliese las siguientes propiedades:

- Para añadir una nueva extensión, debería modificar las menores entidades posibles. Este punto está fuertemente respaldado por el principio Open/Closed de SOLID, el cual dice que: “Una entidad de software debería estar abierta a la extensión, pero cerrada a la modificación.”. Es decir, si en un futuro quisiese que el sistema también admitiese archivos .kml, debería modificar las menores entidades existentes posibles y no debería modificar entidades que se encarguen de tratar las extensiones .gpx o .tcx. La responsabilidad de tratar un archivo .gpx, la responsabilidad de tratar un archivo .tcx y la responsabilidad de tratar un archivo .kml deberían ser independientes entre sí. De aquí deducimos que necesitaremos una entidad por cada extensión.
- Separar la funcionalidad común, de la individual de cada extensión. Existe una funcionalidad común: leer el archivo y existe una funcionalidad individual de cada extensión: interpretar el archivo leído.
- Necesitaré una entidad que, dependiendo de la extensión del archivo devuelva la funcionalidad correspondiente al mismo.

Partiendo de estos puntos, el diseño final resultante es el siguiente:

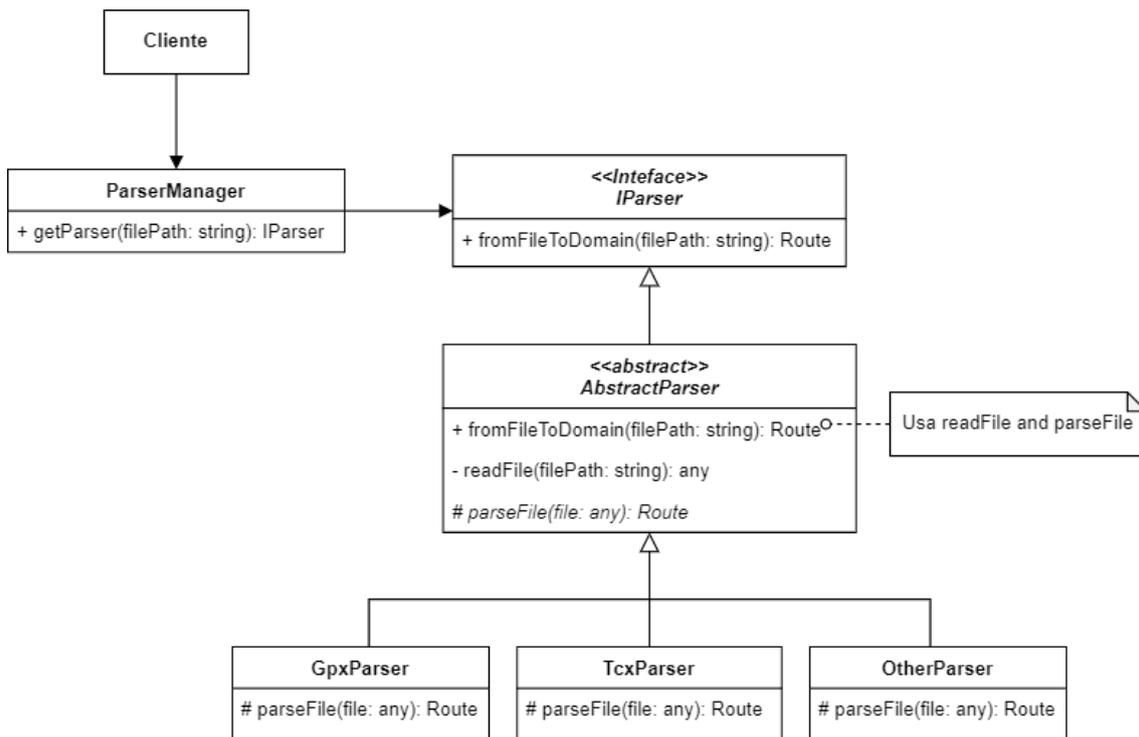


Ilustración 6.5 Diseño del Parser

Este diseño se basa en el patrón “Template method”, aunque está parcialmente adaptado. Los participantes son los siguientes:

- **IParser.** Declara el método para leer un archivo y transformarlo (`fromFileToDomain`).
- **AbstractParser.** Implementa la funcionalidad de leer fichero (`readFile`) y declara la operación de transformar el fichero leído (`parseFile`), que redefinirán los **ConcreteParsers**.
- **ConcreteParsers.** En el sistema actual serían **GpxParser** y **TcxParser**. Si quisiésemos añadir una tercera extensión, añadiríamos una clase similar (**OtherParser**). Estos implementan la funcionalidad de transformar el fichero leído (`parseFile`).
- **ParserManager.** A partir de la extensión del archivo leído, devuelve un **ConcreteParser** u otro.
- **Cliente.** Entidad externa que hace uso del Parser, en este caso nuestro sistema.

Si repasamos los objetivos del diseño descritos anteriormente, veremos que se cumplen:

- “Para añadir una nueva extensión, debería modificar las menores entidades posibles”. Para comprobar que se cumple, veamos qué debería hacer un nuevo desarrollador si quisiese añadir una tercera extensión al sistema, por ejemplo, KML:
  - Se añadiría una nueva clase, `KmlParser`, que heredase de `AbstractParser`, implementado el método `parseFile`.
  - Se añadiría un nuevo caso en el `ParserManager` que devolviese la nueva extensión.

```
//Return Parser
if (extension.toLocaleLowerCase().localeCompare("gpx") == 0){
    return new GpxParser();
}
else if (extension.toLocaleLowerCase().localeCompare("tcx") == 0){
    return new TcxParser();
}
else if (extension.toLocaleLowerCase().localeCompare("kml") == 0){
    return new KmlParser();
}
else {
    return null;
}
```

Ilustración 6.6 Ifelse en ParserManager

- “Separar la funcionalidad común, de la individual de cada extensión”. Aquí es donde entra el patrón “Template method”. Se define un método “plantilla” (fromFileToDomain) el cual implementa la funcionalidad común, en este caso leer el fichero (readFile) y declara un método común, dejando a las subclases (ConcreteParsers) que implementen la funcionalidad propia de cada extensión (parseFile).
- “Necesitaré una entidad que, dependiendo de la extensión del archivo devuelva la funcionalidad correspondiente al mismo”. Este es el ParserManager, que, dependiendo del archivo leído, devuelve un Parser u otro. Alternativamente al ParserManager, mi tutor me propuso la alternativa de definir un diccionario en la entrada de la aplicación para no tener estos “ifelses” anidados. Preferí dejar el ParserManager para tener el código más localizado, en el paquete “parsers”.

Como en anteriores capítulos, este diseño surgido a partir de la necesidad de importar diferentes archivos es el subsistema descrito en el [Capítulo 5.2.1.2 Subsistema Parsers](#).

## 6.1.4 Visualizar diferentes mapas

Este diseño nace a partir de la necesidad de representar las rutas en varios tipos de mapa diferentes. La idea es que cuando el usuario necesite cambiar de mapa, la aplicación sustituya un mapa por otro.

Esto se consigue fácilmente gracias a [React](#). Se puede aislar la funcionalidad de cada mapa en componentes, haciendo que se puedan intercambiar entre sí de una forma muy sencilla. React consigue esto basando su diseño en los patrones Decorator y Composite. El diseño resultante es el siguiente:

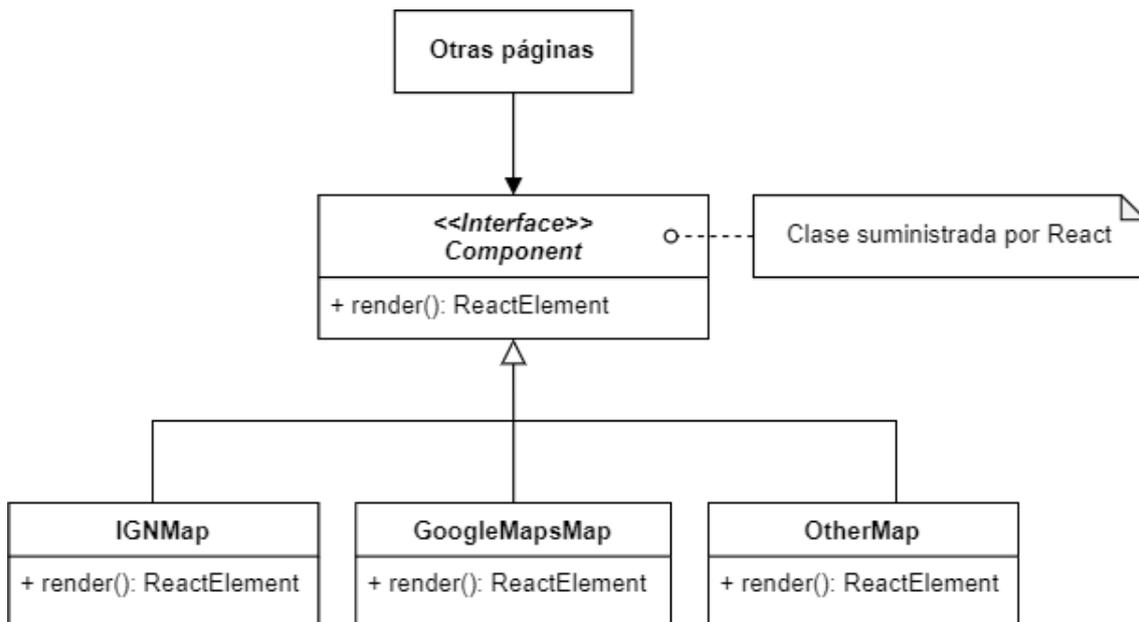


Ilustración 6.7 Diseño de los mapas

Los participantes mostrados en el diseño son los siguientes:

- **Component.** Es la interfaz de React que provee el método render. Este método devuelve uno o varios elementos “html” (realmente no son html si no un supertipo que usa React, ReactElement) que puede ser usado por otros **Component**.
- **ConcreteComponent.** En el diseño mostrado serían **IGNMap**, **GoogleMapsMap** y otro posible mapa a implementar que sería **OtherMap**. Implementan la funcionalidad propia necesaria para visualizar cada mapa.
- **Otras páginas.** Estas son las páginas que harían uso de estos mapas. En nuestro sistema son los componentes SeeRoute.tsx y AllRoutes.tsx.

Cabe indicar que este intercambio de componentes, además de utilizarlo en los mapas, se utiliza reiteradas veces en la aplicación ya que es la base de React. Por ejemplo, se ha utilizado en los componentes de los listados, para cambiar fácilmente los listados con todas las rutas o con las rutas de las listas creadas.

Si se quisiese añadir otro mapa, habría que crear la nueva clase que heredase de Component e implementando la funcionalidad propia. Además, habría que añadirla en las páginas donde se les de uso a los mapas.

Finalmente, este diseño está contenido dentro del subsistema descrito en el [Capítulo 5.2.1.4 Subsistema Vista](#).

## 6.2 Diseño de la Interfaz

En esta sección mostraré los diseños de las pantallas finales con las que el usuario puede interactuar.

Tras el desarrollo del sistema y un análisis adicional con mi tutor, la interfaz ha sufrido algunos cambios. Adicionalmente, estos diseños pueden sufrir pequeños cambios tras la revisión de las pruebas definidas. Los diseños son los siguientes:

### 6.2.1 Barra de navegación



Ilustración 6.8 Barra de navegación

La barra de navegación se coloca en la parte superior de la aplicación y está presente en todas las pantallas.

### 6.2.2 Gestión de rutas



Ilustración 6.9 Pantalla de gestión

Dentro de esta pantalla podemos distinguir dos partes:

### 6.2.2.1 Importar



*Ilustración 6.10 Interfaz para importar las rutas*

Aquí podemos hacer dos acciones: arrastrar una ruta hasta el Dropbox o seleccionar una ruta desplegando el explorador de archivos.

### 6.2.2.2 Gestionar



*Ilustración 6.11 Interfaz de gestión*

Dentro de esta sección podemos realizar varias acciones, en función de los siguientes iconos:

- Crear una nueva lista:



*Ilustración 6.12 Icono para crear una nueva lista*

- Editar una lista existente:



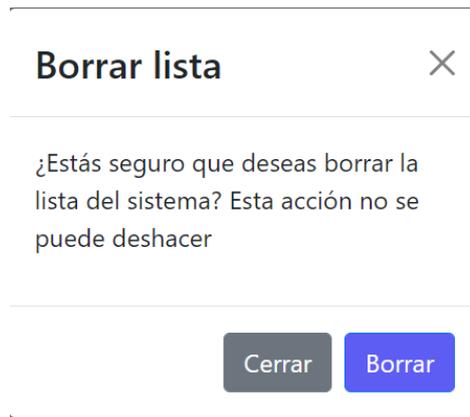
*Ilustración 6.13 Icono para editar una lista*

- Eliminar una lista existente:



*Ilustración 6.14 Icono para eliminar una lista*

Adicionalmente, el usuario necesitará confirmar la acción.



*Ilustración 6.15 Modal para confirmar el borrado de una lista*

- Añadir una ruta a una lista:



*Ilustración 6.16 Icono para añadir una ruta a una lista*

- Ir a la página de visualizar una ruta específica



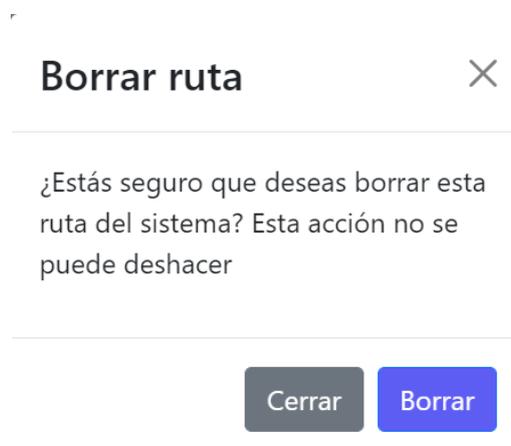
*Ilustración 6.17 Icono para visualizar una ruta*

- Eliminar una ruta del sistema



*Ilustración 6.18 Icono para eliminar una ruta del sistema*

Adicionalmente, el usuario necesitará confirmar la acción.



*Ilustración 6.19 Modal para confirmar el borrado de una ruta*

Además, si editamos una ruta, la parte derecha cambiará y mostrará las rutas contenidas dentro de la lista:

### Gestionar

Listas	Senderismo
Todas las rutas	[1/5/2018] - PICOS GÚA Y ESCOBÍN
Entrenamiento	[12/10/2018] - [Valle del Lago - Somiedo - Principado de Asturias] Ruta del Lago del Valle
Senderismo	[29/6/2018] - Camino Primitivo Oviedo - Santiago de Compostela - Link Epi...

Ilustración 6.20 Interfaz de gestión de una lista

## 6.2.3 Mapa general

La vista inicial que se le presentará al usuario es la siguiente:

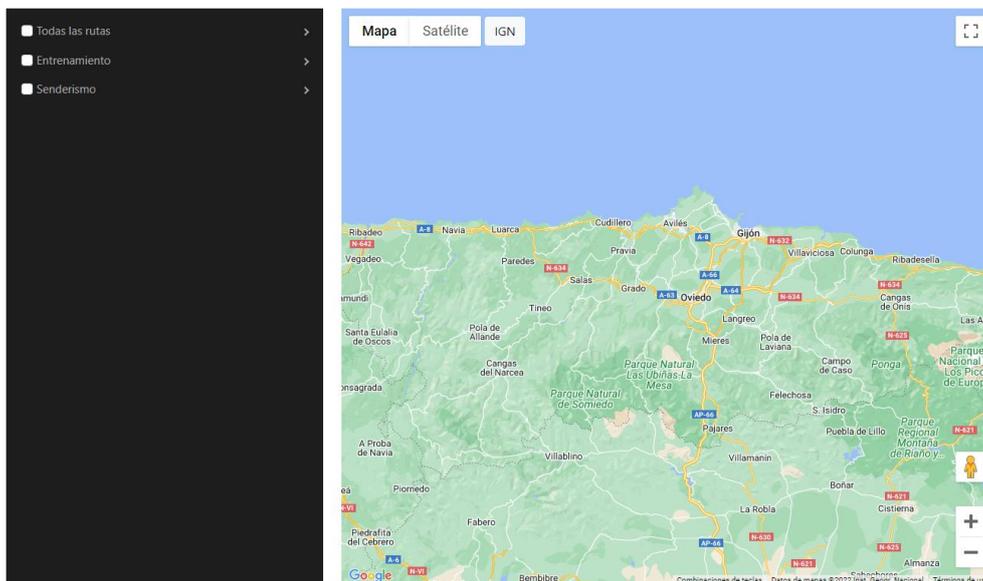


Ilustración 6.21 Mapa general

El usuario puede seleccionar una o varias rutas de la lista y se le pintaran en el mapa:

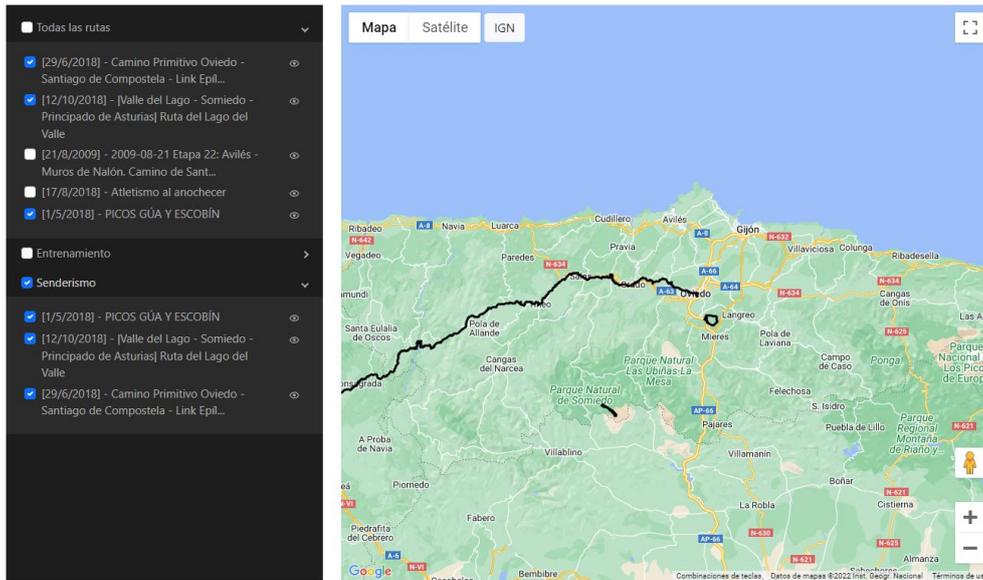


Ilustración 6.22 Mapa general con rutas

Tenemos disponible otras tres vistas del mapa:

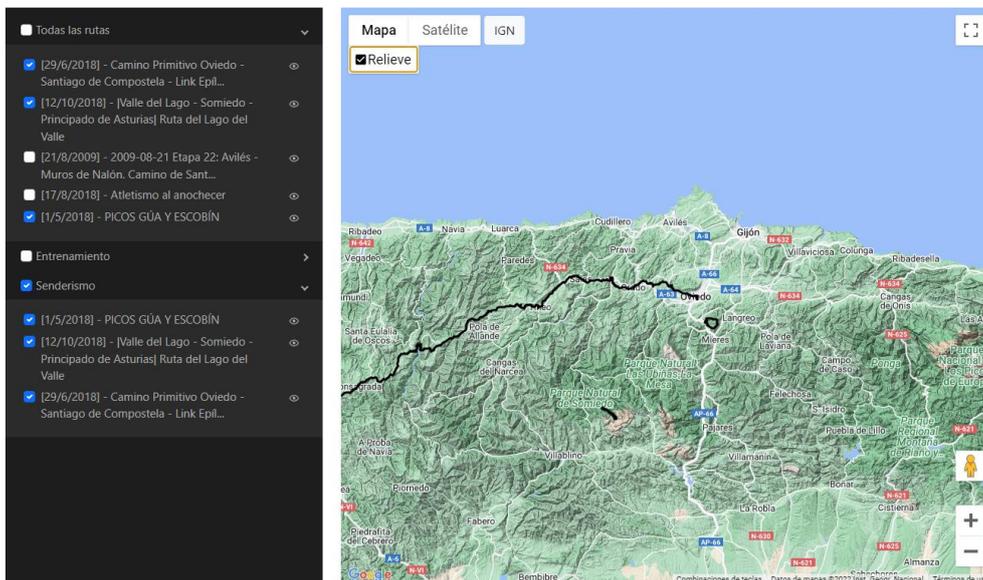


Ilustración 6.23 Vista relieve del mapa general

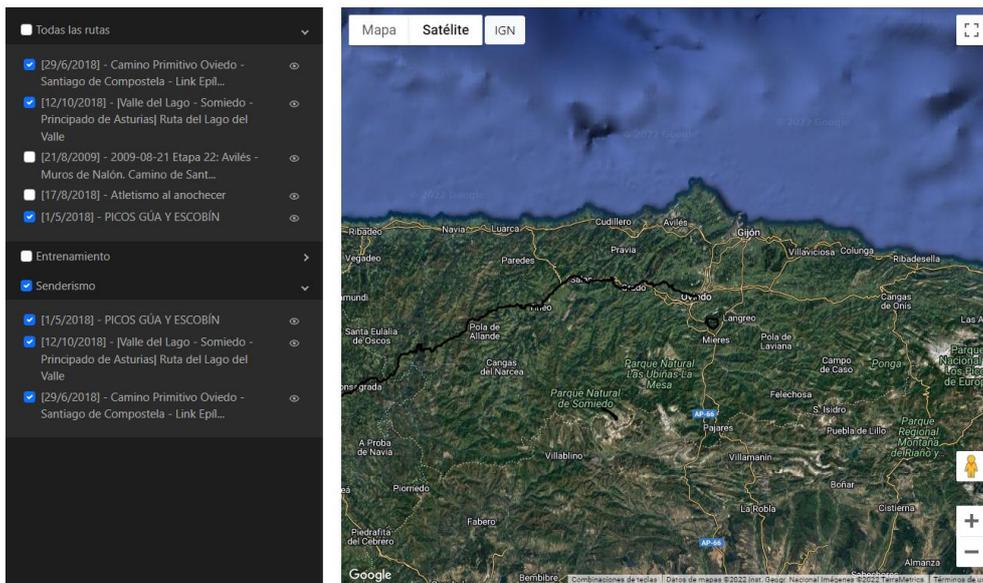


Ilustración 6.24 Vista satélite del mapa general

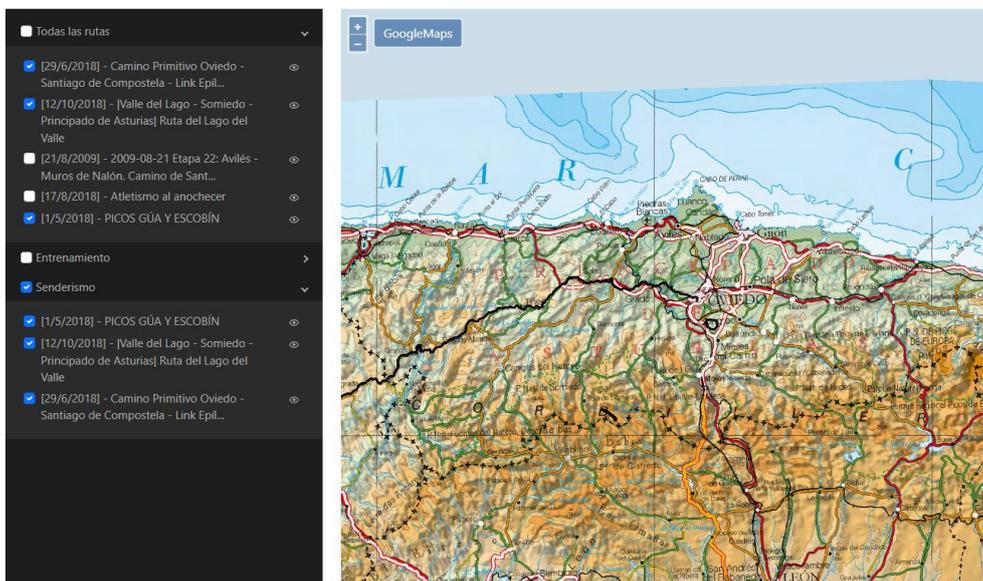


Ilustración 6.25 Vista del IGN del mapa general

Podremos ver los datos de una ruta específica si pulsamos en el siguiente icono:



Ilustración 6.26 Icono para visualizar una ruta

## 6.2.4 Datos de una ruta

La vista inicial que verá el usuario es la siguiente:



Ilustración 6.27 Datos de una ruta

Dentro de esta vista, como en el apartado anterior, puede cambiar los mapas a visualizar:

### [1/5/2018] - PICOS GÚA Y ESCOBÍN



Ilustración 6.28 Vista relieve en los datos de una ruta

### [1/5/2018] - PICOS GÚA Y ESCOBÍN



Ilustración 6.29 Vista satélite en los datos de una ruta

[1/5/2018] - PICOS GÚA Y ESCOBÍN

<b>Distancia</b> 14,64km	<b>Tiempo total</b> 4h 30m 29s	<b>Desnivel positivo acumulado</b> 1072,99m	<b>Desnivel negativo acumulado</b> -1099,60m	<b>Ritmo medio</b> 18,48m/km
-----------------------------	-----------------------------------	--	---	---------------------------------

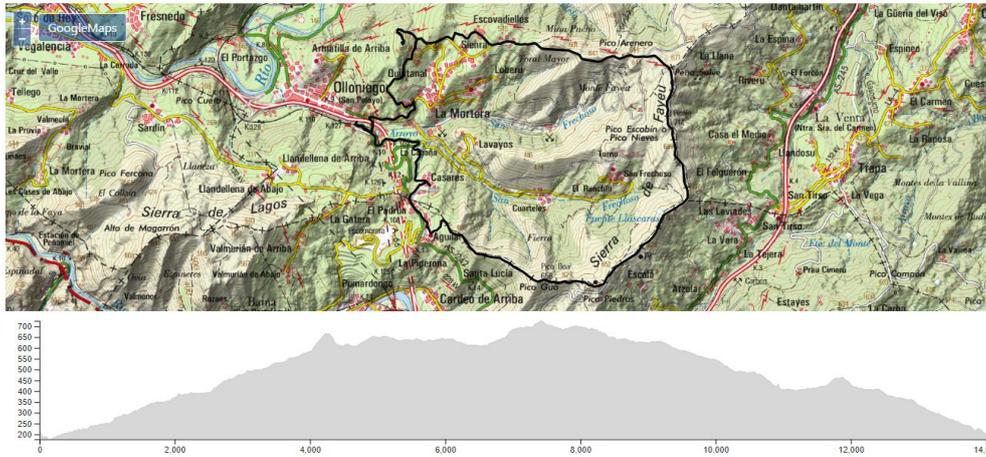


Ilustración 6.30 Vista del IGN en los datos de una ruta

## 6.3 Especificación Técnica del Plan de Pruebas

Tras desarrollar una versión estable de la aplicación, se ha definido qué pruebas se aplicarán sobre esta. Como se ha explicado previamente, se han definido tres tipos de pruebas: Pruebas del Sistema, Pruebas de Usabilidad y Pruebas de Aceptación. Se han aplicado en el orden descrito el mismo orden descrito. A continuación, se muestra el diseño de todas estas pruebas:

### 6.3.1 Pruebas del Sistema

Mediante las pruebas de sistema trataremos de probar toda la funcionalidad descrita en los escenarios presentados en el capítulo [5.5 Análisis de Casos de Uso y Escenarios](#). Las pruebas se llevarán a cabo manualmente.

La estructura seguida es la siguiente:

- **Prueba:** una descripción de cómo debe llevarse a cabo la prueba.
- **Resultado Esperado:** qué debe suceder en el sistema una vez finalizada la prueba.
- **Resultado\*:** esta columna solo se verá en el desarrollo de las pruebas y muestra el resultado obtenido al realizar la prueba.

Las pruebas obtenidas agrupadas por caso de uso son las siguientes:

Caso N.1 Importar una ruta	
Prueba	Resultado Esperado
Importar una ruta desde el explorador de archivos.	Se añade la ruta a base de datos y al listado general.
Prueba	Resultado Esperado
Importar una ruta arrastrándola al dropbox.	Se añade la ruta a base de datos.
Prueba	Resultado Esperado
Intentar importar una ruta con formato incorrecto.	Se notifica al usuario que el formato es incorrecto.

Tabla 6.1 Pruebas del Caso N.1 Importar una ruta

Caso N.2 Ver datos de una ruta	
Prueba	Resultado Esperado
Importo una ruta y desde la página Rutas, navego hasta la página que muestra los datos de la ruta	Veo los siguientes datos de la ruta importada: <ul style="list-style-type: none"> <li>- Nombre (con fecha)</li> <li>- Distancia</li> <li>- Tiempo total</li> <li>- Desnivel positivo acumulado</li> <li>- Desnivel negativo acumulado</li> <li>- Ritmo medio</li> <li>- Mapa con la ruta pintada</li> <li>- Perfil de altitud</li> </ul>
Prueba	Resultado Esperado
Importo una ruta y desde la página Mapa, navego hasta la página que muestra los	Veo los siguientes datos de la ruta importada: <ul style="list-style-type: none"> <li>- Nombre (con fecha)</li> </ul>

datos de la ruta	<ul style="list-style-type: none"> <li>- Distancia</li> <li>- Tiempo total</li> <li>- Desnivel positivo acumulado</li> <li>- Desnivel negativo acumulado</li> <li>- Ritmo medio</li> <li>- Mapa con la ruta pintada</li> <li>- Perfil de altitud</li> </ul>
<b>Prueba</b>	<b>Resultado Esperado</b>
Visualizando mi ruta, cambio a la vista del IGN.	Se cambia a la vista del IGN y veo mi ruta pintada sobre la vista del IGN.
<b>Prueba</b>	<b>Resultado Esperado</b>
Visualizando mi ruta, cambio a la vista Satélite.	Se cambia la vista a Satélite y veo mi ruta pintada sobre la vista Satélite.
<b>Prueba</b>	<b>Resultado Esperado</b>
Visualizando mi ruta, cambio a la vista Relieve.	Se cambia la vista a Relieve y veo mi ruta pintada sobre la vista Relieve.

*Tabla 6.2 Pruebas del Caso N.2 Ver datos de una ruta*

Caso N.3 Eliminar una ruta del sistema	
Prueba	Resultado Esperado
Eliminar una ruta importada pulsando el icono de la papelera.	Tras confirmar la acción, se elimina de base de datos y del listado general.
<b>Prueba</b>	<b>Resultado Esperado</b>
Cancela la acción	Cancela la eliminación y no pasa nada.

*Tabla 6.3 Pruebas del Caso N.3 Eliminar una ruta del sistema*

Caso N.4 Ver todas las rutas del sistema sobre un mapa	
Prueba	Resultado Esperado
En la pantalla Mapa, ver todas las rutas importadas pulsando el checkbox "Todas las rutas".	Se pintan todas las rutas sobre el mapa.
<b>Prueba</b>	<b>Resultado Esperado</b>
Deseleccionar el checkbox "Todas las rutas".	Se despintan todas las rutas del mapa.
<b>Prueba</b>	<b>Resultado Esperado</b>
En la pantalla Mapa, ver todas las rutas importadas pulsando uno por uno el checkbox de cada ruta.	<ul style="list-style-type: none"> <li>- Se pintan todas las rutas sobre el mapa.</li> <li>- El checkbox de "Todas las rutas" se selecciona solo.</li> </ul>
<b>Prueba</b>	<b>Resultado Esperado</b>
Deseleccionar el checkbox de una ruta.	<ul style="list-style-type: none"> <li>- Se despinta la ruta del mapa.</li> <li>- El checkbox de "Todas las rutas" se deselecciona solo.</li> </ul>
<b>Prueba</b>	<b>Resultado Esperado</b>
Visualizando las rutas, cambio a la vista del IGN.	Se cambia a la vista del IGN y veo las rutas pintadas sobre la vista del IGN.
<b>Prueba</b>	<b>Resultado Esperado</b>
Visualizando las rutas, cambio a la vista Satélite.	Se cambia la vista a Satélite y veo las rutas pintadas sobre la vista Satélite.
<b>Prueba</b>	<b>Resultado Esperado</b>

Visualizando las rutas, cambio a la vista Relieve.	Se cambia la vista a Relieve y veo las rutas pintadas sobre la vista Relieve.
--	---

*Tabla 6.4 Pruebas del Caso N.4 Ver todas las rutas del sistema sobre un mapa*

Caso N.5 Crear una lista	
Prueba	Resultado Esperado
En la pantalla Rutas, crear una nueva lista introduciendo el nombre.	Se añade una nueva lista a la base de datos y a listado con las listas.
Prueba	Resultado Esperado
Cancela la acción.	Cancela la creación y no pasa nada.

*Tabla 6.5 Pruebas del Caso N.5 Crear una lista*

Caso N.6 Añadir una ruta a una lista	
Prueba	Resultado Esperado
En la pantalla rutas, se selecciona el icono + de una de las rutas, se selecciona una lista y se confirma la acción.	Se añade la ruta a la lista seleccionada en base de datos y en la propia lista.
Prueba	Resultado Esperado
En la pantalla rutas, se selecciona el icono + de una de las rutas, se selecciona varias listas y se confirma la acción.	Se añade la ruta a las listas seleccionadas en base de datos y en la propia lista.
Prueba	Resultado Esperado
En la pantalla rutas, se selecciona el icono + de una de las rutas, NO se selecciona ninguna lista y se confirma la acción.	- No se produce ningún cambio - Se notifica al usuario.
Prueba	Resultado Esperado
Cancela la acción.	No sucede nada.

*Tabla 6.6 Pruebas del Caso N.6 Añadir una ruta a una lista*

Caso N.7 Ver las rutas de una lista sobre un mapa	
Prueba	Resultado Esperado
En la pantalla Mapa, ver todas las rutas de una lista pulsando el checkbox de la lista.	- Se pintan todas las rutas de la lista sobre el mapa. - Además se seleccionarán las rutas en los otros lugares del listado en las que aparezcan ("Todas las rutas" u otras listas).
Prueba	Resultado Esperado
Deseleccionar el checkbox de la lista	- Se despintan las rutas del mapa. - Además se deseleccionarán las rutas en los otros lugares del listado en las que aparezcan ("Todas las rutas" u otras listas).

*Tabla 6.7 Pruebas del Caso N.7 Ver las rutas de una lista sobre un mapa*

Caso N.8 Eliminar una ruta de una lista	
Prueba	Resultado Esperado
En la pantalla Rutas, seleccionamos una lista	- Se elimina la "relación" base de datos.

y eliminamos una ruta de ella con el icono papelera.	- Se elimina del listado de la lista.
--	---------------------------------------

*Tabla 6.8 Pruebas del Caso N.8 Eliminar una ruta de una lista*

Caso N.9 Eliminar una lista del sistema	
Prueba	Resultado Esperado
En la pantalla Rutas, en el listado general y eliminamos una lista con el icono papelera.	- Se elimina de base de datos. - Se elimina del listado de listas.
Prueba	Resultado Esperado
En la pantalla Rutas, dentro una lista, la eliminamos con el icono papelera.	- Se elimina de base de datos. - Se elimina del listado de listas. - Se redirige al listado con todas las rutas

*Tabla 6.9 Pruebas del Caso N.9 Eliminar una lista del sistema*

## 6.3.2 Pruebas de Usabilidad

Las pruebas de usabilidad están enfocadas a conocer la satisfacción con el producto final de diferentes tipos de clientes. Para ello, se usarán varios cuestionarios, tratando de recoger la máxima cantidad de información posible. También se tomará una muestra diversa, con potenciales clientes y usuarios medios que tengan poco conocimiento del dominio.

A continuación, se mostrarán los distintos diseños creados para realizar las pruebas

### 6.3.2.1 Preguntas de carácter general

El siguiente cuestionario recogerá información general, con el objetivo de conocer el perfil del cliente. El cuestionario contendrá las siguientes preguntas:

<b>¿Usa un ordenador frecuentemente?</b>
<ol style="list-style-type: none"> <li>1. Todos los días de la semana</li> <li>2. Entre 4 y 6 días por semana</li> <li>3. Entre 1 y 3 días a la semana</li> <li>4. Nunca o casi nunca</li> </ol>
<b>¿Qué tipo de actividades realiza con el ordenador principalmente?</b>
<ol style="list-style-type: none"> <li>1. Por trabajo</li> <li>2. Por ocio (videojuegos, películas, series, videos...)</li> <li>3. Busco información en internet</li> <li>4. Para usar una o varias redes sociales</li> </ol>
<b>¿Qué busca principalmente al navegar por la web?</b>
<ol style="list-style-type: none"> <li>1. Que sea fácil de usar</li> <li>2. Que sea intuitivo</li> </ol>

<ol style="list-style-type: none"> <li>3. Que sea rápido</li> <li>4. Que tenga todas las funciones necesarias</li> </ol>
<p>¿Ha usado alguna vez software de gestión de rutas (Strava, Wikiloc, Polar FlowSync, Garmin Connect ...)?</p>
<ol style="list-style-type: none"> <li>1. Sí, lo uso varias veces por semana</li> <li>2. Sí, lo uso ocasionalmente</li> <li>3. Sí, lo he probado, pero no lo uso</li> <li>4. No, nunca</li> </ol>
<p>¿Qué busca principalmente en el tipo de programas descritos anteriormente?</p>
<ol style="list-style-type: none"> <li>1. Que pueda ver los máximos datos posibles de una ruta</li> <li>2. Que tenga un fuerte factor social</li> <li>3. Que pueda ver varios datos de mis rutas a la vez sobre un mismo mapa</li> <li>4. Que sea muy versátil (Importar diferentes archivos o visualizar las rutas en diferentes mapas)</li> </ol>

Tabla 6.10 Preguntas de carácter general

### 6.3.2.2 Actividades guiadas

Varias actividades que el usuario puedes hacer en la aplicación, tratando de abarcar la máxima funcionalidad posible. Al final de cada actividad, se tomará nota del resultado y se preguntará si tiene alguna observación que darnos.

ID	Prueba	Tiempo	Observaciones
1	Importar una ruta		
2	Borrar una ruta del sistema		
3	Crear una lista		
4	Añadir una ruta a una lista		
5	Ver las rutas de una lista		
6	Borrar una ruta de una lista		
7	Borrar una lista		
8	Ver las rutas de una lista en el mapa general		
9	Ver los datos de una ruta en concreto		
10	Cambiar de mapa		

Tabla 6.11 Actividades guiadas

### 6.3.2.3 Preguntas Cortas sobre la Aplicación y Observaciones

Cuando el usuario haya hecho uso de la aplicación, se le harán algunas preguntas cortas sobre la facilidad de uso y el correcto comportamiento del sistema. Además, se tomará nota de posibles observaciones finales que nos quiera dar.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la				

aplicación?				
¿Existe ayuda para las funciones en caso de que tenga dudas?				
¿Le resulta sencillo el uso de la aplicación?				
<b>Funcionalidad</b>	<b>Siempre</b>	<b>Frecuentemente</b>	<b>Ocasionalmente</b>	<b>Nunca</b>
¿Funciona cada tarea como espera?				
¿El tiempo de respuesta de la aplicación es muy grande?				
¿Ha echado en falta funcionalidad?				
<b>Calidad del Interfaz</b>				
<b>Aspectos gráficos</b>	<b>Muy Adecuado</b>	<b>Adecuado</b>	<b>Poco Adecuado</b>	<b>Nada Adecuado</b>
El tipo y tamaño de letra es				
Los iconos e imágenes usados son				
Los colores empleados son				
El tamaño de los mapas es				
<b>Diseño de la Interfaz</b>		<b>Si</b>	<b>No</b>	<b>A veces</b>
¿Le resulta fácil de usar?				
¿El diseño de las pantallas es claro y atractivo?				
¿Cree que el programa está bien estructurado?				
<b>Observaciones</b>				

Tabla 6.12 Preguntas cortas sobre la aplicación y observaciones

### 6.3.2.4 Cuestionario para el Responsable de las Pruebas

En el siguiente cuestionario recogeremos los resultados y observaciones desde el punto de vista del desarrollador. Mientras se realicen las pruebas, iremos tomando nota de errores, posibles confusiones o comportamientos anómalos del usuario que debamos tener en cuenta.

Aspecto Observado	Notas
Errores leves cometidos	
Errores graves cometidos	
Confusiones	
¿No ha podido llevar a cabo alguna funcionalidad?	
Posibles mejoras	
Observaciones generales	

Tabla 6.13 Cuestionario para el responsable de las pruebas

### 6.3.3 Pruebas de Accesibilidad

Aunque nuestro sistema sea una aplicación de escritorio y no web, hace uso de tecnologías web, por lo que aplicaré herramientas pensadas para probar páginas web. No puedo usar herramientas automatizadas, así que usaré la checklist de WCAG.<sup>9</sup>

La checklist se ha obtenido de la página <https://www.digitala11y.com/wcag-checklist/>. Solo se aplicarán los niveles A y AA, si cumpliesen todas las pruebas de ambos niveles, añadiríamos el último nivel AAA. La checklist es la siguiente:

#### 6.3.3.1 Nivel A

Prueba	Cumple	No cumple	N/A
<b>1.1.1 Contenido que no es texto:</b> Proporcionar alternativas de texto para contenido que no sea de texto			
<b>1.2.1 Solo audio y solo vídeo (pregrabado):</b> Proporcione una alternativa al contenido solo de vídeo y solo de audio			
<b>1.2.2 Subtítulos (pregrabados):</b> Proporcione subtítulos para vídeos con audio			
<b>1.2.3 Descripción de audio o alternativa multimedia (pregrabado):</b> El vídeo con audio tiene una segunda alternativa			
<b>1.3.1 Información y relaciones:</b> El contenido, la estructura y las relaciones pueden determinarse lógicamente			
<b>1.3.2 Secuencia significativa:</b> Presente el contenido en un orden significativo			
<b>1.3.3 Características sensoriales:</b> Las instrucciones no se basan únicamente en las características sensoriales.			
<b>1.4.1 Uso del color:</b> No utilice una presentación que dependa únicamente del color			
<b>1.4.2 Control de audio:</b> No reproduzca audio automáticamente			
<b>2.1.1 Teclado:</b> Accesible solo con teclado			
<b>2.1.2 Sin bloqueo de teclado:</b> No impida el uso de teclado a los usuarios			
<b>2.1.4 Accesos directos:</b> Proporcionar un mecanismo para configurar los métodos abreviados de teclado			
<b>2.2.1 Temporización ajustable:</b> Los límites de tiempo tienen controles de usuario			
<b>2.2.2 Pausa, Detener, Ocultar:</b> Proporcionar controles de usuario para mover contenido			
<b>2.3.1 Tres destellos o menos:</b> Ningún contenido parpadea más de tres veces por segundo			
<b>2.4.1 Bloques de derivación:</b> Ofrecer a los usuarios la posibilidad de saltarse bloques de contenido repetidos.			

<sup>9</sup> Introducción a las Pautas de Accesibilidad para el Contenido Web (WCAG). (2022, 26 de mayo). W3C. <https://www.w3.org/WAI/standards-guidelines/wcag/es>

<b>2.4.2 Página titulada:</b> Utilice títulos de página útiles y claros			
<b>2.4.3 Orden de enfoque:</b> Orden lógico			
<b>2.4.4 Propósito del enlace (en contexto):</b> El propósito de cada enlace se desprende claramente de su contexto			
<b>2.5.1 Gestos de puntero:</b> todas las funciones que utilizan gestos multipunto o basados en rutas para hacer una operación se pueden hacer con un solo puntero sin gestos, a menos que sea esencial.			
<b>2.5.2 Cancelación del puntero:</b> Para ayudar a evitar la activación involuntaria de los controles, evite la activación no esencial del evento descendente (por ejemplo, onmousedown) al hacer clic, tocar o presionar la pantalla durante mucho tiempo.			
<b>2.5.3 Etiqueta en Nombre:</b> Para los componentes de la interfaz de usuario con etiquetas que incluyen texto o imágenes de texto, el nombre contiene el texto que se presenta visualmente.			
<b>2.5.4 Motion Actuation:</b> La funcionalidad que se activa moviendo el dispositivo (como agitar o desplazar un dispositivo móvil) o por el movimiento del usuario (como saludar a una cámara) se puede desactivar y la funcionalidad equivalente se proporciona a través de controles estándar como botones			
<b>3.1.1 Idioma de la página:</b> La página tiene un idioma asignado			
<b>3.2.1 Enfoque:</b> Los elementos no cambian cuando reciben enfoque			
<b>3.2.2 En la entrada:</b> Los elementos no cambian cuando reciben la entrada			
<b>3.3.1 Identificación de errores:</b> Identificar claramente los errores de entrada			
<b>3.3.2 Etiquetas o Instrucciones:</b> Etiquetar elementos y dar instrucciones			
<b>4.1.1 Análisis:</b> No hay errores de código importantes			
<b>4.1.2 Nombre, Rol, Valor:</b> Construir todos los elementos pensando en la accesibilidad			

Tabla 6.14 WCAG Nivel A

### 6.3.3.2 Nivel AA

Prueba	Cumple	No cumple	N/A
<b>1.2.4 Subtítulos (en vivo):</b> Los vídeos en directo tienen subtítulos			
<b>1.2.5 Descripción de audio (pregrabada):</b> Los usuarios tienen acceso a la descripción de audio para el contenido de vídeo			
<b>1.3.4 Orientación:</b> El contenido no restringe su vista y operación a una sola orientación de visualización, como vertical u horizontal, a menos que una orientación de			

visualización específica sea esencial			
<b>1.3.5 Identificar propósito de entrada:</b> Los campos de entrada que recopilan ciertos tipos de información de usuario tienen definido un atributo de autocompletar adecuado			
<b>1.4.3 Contraste (mínimo):</b> La relación de contraste entre texto y fondo es de al menos 4,5:1			
<b>1.4.4 Cambiar el tamaño del texto:</b> El texto se puede redimensionar al 200% sin pérdida de contenido o función			
<b>1.4.5 Imágenes de texto:</b> No utilizar imágenes para el texto			
<b>1.4.10 Reflujo:</b> No se pierde de contenido o funcionalidad y se evita el desplazamiento horizontal cuando el contenido se presenta con un ancho de 320 píxeles			
<b>1.4.11 Contraste no de texto:</b> Existe una relación de contraste de al menos 3:1 para diferenciar objetos gráficos (como iconos y componentes de gráficos o gráficos) y componentes de interfaz personalizados por el autor (como botones, controles de formulario y enfoque indicadores/esquemas)			
<b>1.4.12 Espaciado de texto:</b> No se produce pérdida de contenido o funcionalidad cuando el usuario adapta la altura de la línea de texto/espaciado a 1,5 veces el tamaño de la fuente, el espaciado de párrafo a 2 veces el tamaño de la fuente, el espaciado de palabras a .16 veces el tamaño de la fuente y el espaciado de letras a .12 veces el tamaño de la fuente			
<b>1.4.13 Contenido en Hover o Focus:</b> Cuando se presenta contenido adicional en el hover del cursor o en el focus del teclado: el contenido que aparece se puede descartar (generalmente a través de la tecla Esc) sin mover el puntero o el foco del teclado, a menos que el contenido presente un error de entrada o no oculte ni interfiera con el contenido de otra página; el puntero se puede mover al nuevo contenido sin que el contenido desaparezca; el nuevo contenido debe permanecer visible hasta que el puntero o el foco del teclado se aleje del control desencadenador, se descarte el nuevo contenido o el nuevo contenido ya no sea relevante			
<b>2.4.5 Múltiples maneras:</b> Ofrecer varias maneras de encontrar páginas			
<b>2.4.6 Encabezados y etiquetas:</b> Utilice encabezados y etiquetas transparentes			
<b>2.4.7 Enfoque visible:</b> Asegúrese de que el foco del teclado sea visible y claro			
<b>3.1.2 Idioma de las piezas:</b> Indique a los usuarios cuándo cambia el idioma de una página			
<b>3.2.3 Navegación consistente:</b> Utilice los menús de forma coherente			
<b>3.2.4 Identificación consistente:</b> Utilice iconos y botones de forma consistente			
<b>3.3.3 Sugerencia de error:</b> Sugerir correcciones cuando los			

usuarios cometen errores			
<b>3.3.4 Prevención de errores (Legal, Financiero, Datos):</b> Reducir el riesgo de errores de entrada de datos confidenciales			
<b>4.1.3 Mensajes de estado:</b> Los mensajes de estado se programar considerando roles o propiedades del usuario y presentarse al mismo mediante tecnologías de asistencia sin recibir el foco.			

Tabla 6.15 WCAG Nivel AA

# Capítulo 7. Implementación del Sistema

En los siguientes capítulos se explicará qué tecnologías se han usado en el desarrollo y algunos de los problemas encontrados durante el desarrollo.

## 7.1 Lenguajes de Programación

A continuación, se describirán los lenguajes de programación usados y los principales módulos que dan forma al sistema.

### 7.1.1 Css

Css (<https://developer.mozilla.org/es/docs/Web/CSS>) se define como: *Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios languages basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en el habla o en otros medios.*<sup>10</sup>

Es mantenido por World Wide Web Consortium (W3C) y el lanzamiento inicial fue el 17 de diciembre de 1996. La versión usada es: CSS3.

### 7.1.2 Electron

Electron (<https://www.electronjs.org>) se define a sí mismo como: *Electron es un framework para crear aplicaciones de escritorio usando JavaScript, HTML y CSS. Incrustando Chromium y Node.js dentro del mismo, Electron le permite mantener una base de código JavaScript y crear aplicaciones multiplataforma que funcionan en Windows, macOS y Linux, - no requiere experiencia en desarrollo nativo.*<sup>11</sup>

El creador fue Cheng Zhao, actualmente sigue el desarrollo GitHub Inc. y la fecha de lanzamiento fue el de 15 de julio de 2012. La versión usada en el proyecto es: 16.0.4.

#### 7.1.2.1 Node.js

Node.js (<https://nodejs.org/es/>) se define a sí mismo como: *Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. En el siguiente ejemplo de "hola mundo", pueden atenderse*

<sup>10</sup> CSS. (2022, 14 de mayo). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/CSS>

<sup>11</sup> Introducción | Electron. (2022, 14 de mayo). Electron. <https://www.electronjs.org/es/docs/latest>

muchas conexiones simultáneamente. Por cada conexión, se activa la devolución de llamada o callback, pero si no hay trabajo que hacer, Node.js se dormirá.<sup>12</sup>

El creador es Ryan Lienhart Dahl, actualmente sigue el desarrollo Joyent y la fecha de lanzamiento fue el 27 de mayo de 2009. La versión usada en este proyecto es: 16.13.0.

### 7.1.2.2 Chromium

Chromium (<https://www.chromium.org/getting-involved/download-chromium/>) se define a sí mismo como: *Chromium es una base de código abierto para desarrollar un navegador web, mantenida por diversas compañías que posteriormente usan el código fuente para crear su propia versión de navegador con características adicionales.*<sup>13</sup>

Lo desarrolla The Chromium Projects y la fecha de lanzamiento fue el 3 de septiembre de 2008. La versión usada en este proyecto es: 96.0.4664.55.

## 7.1.3 GoogleMaps JavaScript API

GoogleMaps JavaScript API (<https://developers.google.com/maps/documentation/javascript>) se define a sí mismo como: *La API JavaScript de Maps permite personalizar los mapas con su propio contenido e imágenes para mostrarlos en páginas web y dispositivos móviles. La API JavaScript de Maps presenta cuatro tipos de mapas básicos (mapa de carreteras, satélite, híbrido y terreno) que se pueden modificar mediante capas y estilos, controles y eventos, y diversos servicios y bibliotecas.*<sup>14</sup>

Para poder usar esta tecnología, fue necesaria la dependencia [react-wrapper](#).

Es desarrollada por Google.

### 7.1.3.1 React-wraper

React-wrapper se define como: *This wrapper uses @googlemaps/js-api-loader to load the Google Maps JavaScript API*<sup>15</sup>

Actualmente lo mantiene un equipo de desarrolladores libres en Github. La versión usada en este proyecto es: 1.1.18.

---

<sup>12</sup> Acerca | Node.js. (2022, 14 de mayo). Nodejs. <https://nodejs.org/es/about/>

<sup>13</sup> Chromium (navegador). (2022, 14 de mayo). Wikipedia. [https://es.wikipedia.org/wiki/Chromium\\_\(navegador\)](https://es.wikipedia.org/wiki/Chromium_(navegador))

<sup>14</sup> Overview. (2022, 16 de mayo). Google Maps Platform. <https://developers.google.com/maps/documentation/javascript/overview>

<sup>15</sup> React-wrapper. (2022, 16 de mayo). Github. <https://github.com/googlemaps/react-wrapper>

## 7.1.4 NeDB

Para entender NeDB (<https://github.com/louischatriot/nedb>), es necesario saber qué es una base de datos documental.

NeDB se describe como: *Hace unos días nos encontrábamos analizando la mejor solución para guardar una pequeña cantidad de datos (...) Tras estudiar varias opciones, descubrimos NeDB, una base de datos escrita completamente en Javascript, sin ningún tipo de dependencia binaria, y que expone un interfaz de programación muy similar a MongoDB.*<sup>16</sup>

Fue mantenida por un equipo de desarrolladores libres en Github, pero desde julio de 2022 han dejado de darle soporte al proyecto. La versión usada en este proyecto es: 1.8.12.

### 7.1.4.1 Base de datos documental

Una base de datos documental se define como: *Una base de datos documental es una de las principales variantes de las bases de datos no relacionales o NoSQL. (...) La información no está contenida en tablas. Por el contrario, están pensadas para el almacenamiento de datos semiestructurados, los cuáles se organizan en documentos con valores asignados*<sup>17</sup>

## 7.1.5 Openlayers

Openlayers (<https://openlayers.org>) se define como: *OpenLayers es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web.*<sup>18</sup>

Actualmente es mantenido por un equipo de desarrolladores libres en Github. La versión usada en este proyecto es: 6.12.0.

## 7.1.6 React

React (<https://es.reactjs.org>) se define como: *React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. (...) React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo.*<sup>19</sup>

El creador Jordan Walke, actualmente sigue el desarrollo Facebook and community y la fecha de lanzamiento fue el 29 de mayo de 2013. La versión usada en este proyecto es: 17.0.2.

---

<sup>16</sup> Introducción a NeDB: una base de datos javascript embebida. (200, 16 de mayo). todoJS. <https://www.todojs.com/introduccion-a-nedb-una-base-de-datos-javascript-embebida/>

<sup>17</sup> Bases de datos documentales. ¿Qué es? Tipos y ejemplos. (2022, 17 de mayo). Ayudaley. <https://ayudaleyprotecciondatos.es/bases-de-datos/documentales/>

<sup>18</sup> Openlayers. (2022, 16 de mayo). Wikipedia. <https://es.wikipedia.org/wiki/OpenLayers>

<sup>19</sup> React. (2022, 14 de mayo). Wikipedia. <https://es.wikipedia.org/wiki/React>

## 7.1.7 TypeScript

Para entender TypeScript (<https://www.typescriptlang.org>), primero es necesario entender Javascript.

TypeScript se define a sí mismo como: *TypeScript ofrece todas las características de JavaScript, y una capa adicional sobre éstas: El sistema de tipos de TypeScript. Por ejemplo, JavaScript proporciona tipos primitivos como "string" y "number", pero no comprueba que las hayas asignado de forma consistente. TypeScript lo hace. Esto significa que tu código JavaScript existente, también funciona con TypeScript. El principal beneficio de TypeScript es que puede resaltar comportamientos inesperados en tu código, reduciendo la posibilidad de bugs.*<sup>20</sup>

Lo desarrolla Microsoft y la fecha de lanzamiento fue el 1 de octubre de 2012. La versión usada es: 4.6.3.

### 7.1.7.1 Javascript

Tal como describe la web oficial de Javascript (<https://developer.mozilla.org/es/docs/Web/JavaScript>): *JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional). Lee más en acerca de JavaScript.*<sup>21</sup>

Lo desarrolla ECMAScript y la fecha de lanzamiento fue el 4 de diciembre de 1995.

## 7.1.8 Webpack

Webpack (<https://webpack.js.org>) se define como: *Más allá de su descripción oficial, podríamos definir Webpack como una herramienta Open Source utilizada por los desarrolladores para empaquetar y exportar todos los ficheros necesarios para que un proyecto funcione con todas sus dependencias frontend. Entre otras funcionalidades, con Webpack podemos obtener un archivo JavaScript que contenga la totalidad del código necesario en un proyecto minimizado y transpilado, incluyendo sus dependencias, o bien un archivo con el CSS tras haber pasado los preprocesadores necesarios. En este artículo, te explicamos un poco más*

<sup>20</sup> TypeScript para programadores Javascript. (2022, 14 de mayo). TypeScript. <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>

<sup>21</sup> JavaScript. (2022, 14 de mayo). MDN Web Docs. <https://developer.mozilla.org/es/>

el funcionamiento de Webpack y cómo instalar esta útil herramienta para Desarrollo Web. Fue creada por Tobias Koppers.<sup>22</sup>

La funcionalidad principal que he usado en el proyecto es empaquetar todo el código fuente (archivos Typescript, html o css) e imágenes en una sola carpeta con el archivo principal "index.js".

El creador es Tobias Koppers y actualmente se sigue desarrollando en Github por un equipo de desarrolladores libres. La versión usada en el desarrollo es: 6.0.0-beta.61.

## 7.1.9 Otras dependencias

### 7.1.9.1 react-bootstrap

Para entender React-bootstrap (<https://react-bootstrap.github.io>), primero es necesario entender Bootstrap.

Se define como: *React-Bootstrap sustituye a Bootstrap. Cada componente ha sido construido desde cero como un verdadero componente React, sin dependencias innecesarias (...) ha evolucionado y crecido junto a React, por lo que es una excelente opción como base de la interfaz de usuario.*<sup>23</sup>

Es mantenido por un equipo de desarrolladores libres. La versión usada en este proyecto es la 2.4.0

#### 7.1.9.1.1 Bootstrap

Bootstrap (<https://getbootstrap.com>) se define como: *Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.*<sup>24</sup>

Es desarrollado por Twitter y su lanzamiento inicial fue el 19 de agosto de 2011. La versión utilizada en este proyecto es: 5.1.

### 7.1.9.2 react-router

React-router (<https://v5.reactrouter.com/web/guides/quick-start>) es una dependencia de React que se define como: *React Router es una librería de enrutamiento ligera y con todas las*

<sup>22</sup> ¿Qué es Webpack, cómo funciona y cómo instalarlo?. (2022, 14 de mayo). Arsys. <https://www.arsys.es/blog/programacion/webpack-instalacion-funcionamiento>

<sup>23</sup> React Bootstrap. (2022, 14 de mayo). React Bootstrap. <https://react-bootstrap.github.io>

<sup>24</sup> Bootstrap. (2022, 14 de mayo). Wikipedia. [https://es.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))

funciones para la biblioteca de JavaScript React. React Router se ejecuta en todos los lugares donde se ejecuta React; en la web, en el servidor (usando node.js) y en React Native.<sup>25</sup>

Es mantenido por un equipo de desarrolladores libres en Github. La versión usada en este proyecto es la: 6.2.1.

### 7.1.9.3 react-pro-sidebar

React-pro-sidebar (<https://github.com/azouaoui-med/react-pro-sidebar>) es una dependencia de React que se define como: *React Sidebar es una librería personalizable y adaptativa con menús desplegados y un número ilimitado de submenús anidados.*<sup>26</sup>

Es mantenido por un equipo de desarrolladores libres en Github. La versión usada en este proyecto es la 0.7.1.

### 7.1.9.4 react-icons

React-icons (<https://react-icons.github.io/react-icons/>) es una dependencia de React que se define como: *Incluya iconos populares en sus proyectos React fácilmente con react-icons, que utiliza importaciones ES6 que le permiten incluir sólo los iconos que su proyecto está utilizando.*<sup>27</sup>

Es mantenido por un equipo de desarrolladores libres. La versión usada en este proyecto es: 4.3.1.

### 7.1.9.5 d3

D3 (<https://d3js.org>) se define como: *D3.js es una librería de JavaScript para manipular documentos basados en datos. D3 le ayuda a dar vida a los datos utilizando HTML, SVG y CSS.*<sup>28</sup>

Los creadores son Mike Bostock, Jeffrey Heer, Vadim Ogievetsky, actualmente lo sigue manteniendo un equipo de desarrolladores libres y el primer lanzamiento fue el 18 de febrero de 2011. La usada en este proyecto es: 7.4.4.

### 7.1.9.6 fast-xml-parser

fast-xml-parser (<https://github.com/NaturalIntelligence/fast-xml-parser>) se define como: *Validar XML, Parsear XML a JS Object, o Construir XML desde JS Object sin librerías basadas en C/C++ y sin callback.*<sup>29</sup>

---

<sup>25</sup> React-router. (2022, 15 de mayo). Github. <https://github.com/remix-run/react-router>

<sup>26</sup> React Pro Sidebar. (2022, 15 de mayo). Npm. <https://www.npmjs.com/package/react-pro-sidebar>

<sup>27</sup> React-icons. (2022, 15 de mayo). React-icons. <https://react-icons.github.io/react-icons/>

<sup>28</sup> Data-Driven Documents. (2022, 15 de mayo). Data-Driven Documents. <https://d3js.org>

<sup>29</sup> Fast-xml-parser. (2022, 15 de mayo). Npm. <https://www.npmjs.com/package/fast-xml-parser>

Actualmente lo mantiene un equipo de desarrolladores libres. La versión usada en este proyecto es: 4.0.7.

### 7.1.9.7 *electron-packager*

*electron-packager* (<https://github.com/electron/electron-packager>) se define como: *Electron Packager is a command line tool and Node.js library that bundles Electron-based application source code with a renamed Electron executable and supporting files into folders ready for distribution.*<sup>30</sup>

Actualmente lo mantiene un equipo de desarrolladores libres en Gitub. La versión usada en este proyecto es: 15.5.1

---

<sup>30</sup> Electron-packager. (2022, 15 de mayo). Github. <https://github.com/electron/electron-packager>

## 7.2 Herramientas y Programas Usados para el Desarrollo

A continuación, se hará una descripción de todas las herramientas de desarrollo, sistemas adicionales existentes, complementos y otros productos software que he necesitado para la implementación del sistema.

### 7.2.1 Github

Github (<https://github.com>) se define como: *Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarla la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.*<sup>31</sup>

Las ventajas que ofrece a mi proyecto son las siguientes:

- Registro de los cambios en el código de la aplicación, pudiendo revisar una versión antigua si lo necesitase.
- Una copia del proyecto se aloja en los servidores de Github, por lo que puedo compartir el progreso entre ordenadores o si tuviese algún problema con mi ordenador, no se pierda la totalidad del proyecto.
- Posibilidad de compartir el código con otras personas que quieran usar el sistema o desarrollar nuevas funcionalidades.

Me he decantado por Github entre las opciones de software de control de versiones, ya que es la que hemos usado en la carrera mayoritariamente y con la que estoy más familiarizado.

Lo desarrolla Microsoft y su lanzamiento fue el 19 de octubre de 2007.

### 7.2.2 Npm

Npm (<https://www.npmjs.com>) se define como: *De sus siglas NPM (Node Package Manager) es un gestor de paquetes desarrollado en su totalidad bajo el lenguaje JavaScript por Isaac Schlueter, a través del cual podemos obtener cualquier librería con tan solo una sencilla línea de código, lo cual nos permitirá agregar dependencias de forma simple, distribuir paquetes y administrar eficazmente tanto los módulos como el proyecto a desarrollar en general.*<sup>32</sup>

Este gestor es el que he usado principalmente para instalar las dependencias necesarias ya que es el más usado en la actualidad.

---

<sup>31</sup> Qué es Github y qué es lo que le ofrece a los desarrolladores. (2022, 14 de mayo). Xataka. <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>

<sup>32</sup> Qué es NPM y para qué sirve. (2022, 14 de mayo). OpenWebinars. <https://openwebinars.net>

El creador es Isaac Schlueter, es desarrollada por Github y la fecha de lanzamiento fue 27 de enero de 2014. La versión usada en este proyecto es: 8.5.0.

### 7.2.3 Sourcetree

Sourcetree (<https://www.sourcetreeapp.com>) se define como: *Source Tree es un potente GUI (Graphical User Interface – Interfaz Gráfica de Usuario) para gestionar todos tus repositorios ya sean Git o Mercurial. Con Source Tree podemos crear, clonar, hacer commit, push, pull, merge y algunas cosas más de una forma bastante fácil.*<sup>33</sup>

Sourcetree es un producto de Atlassian, fundada en 2002 por Scott Farquhar y Mike Cannon-Brookes. La versión usada en este proyecto es: 3.4.8.

### 7.2.4 Visual Studio Code

Visual Studio Code (<https://code.visualstudio.com>) o VSC se define como: *Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.*<sup>34</sup>

Por defecto soporta Typescript, pero tuve que instalar otras extensiones para acomodarlo a las necesidades de mi proyecto. Las más relevantes son:

- **ES7+ React/Redux/React-Native snippets:** Permite definir trozos de Código rápidamente mediante un comando. <https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets>.
- **GitLens — Git supercharged:** Proporciona herramientas relacionadas con git. Las que más he usado es poder ver los últimos cambios del documento. <https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens>.
- **Better Comments:** Define una estructura para los comentarios de clases o funciones. <https://marketplace.visualstudio.com/items?itemName=aaron-bond.better-comments>.

Me he decantado por VSC por ser de código abierto (aunque la descarga oficial está bajo software privativo) y por estar desarrollado en Typescript y Electron, tecnologías principales en mi proyecto.

<sup>33</sup> SOURCE TREE. (2022, 14 de mayo). Jadcode. <https://jadcode.wordpress.com/2014/06/09/source-tree/>

<sup>34</sup> Visual Studio Code. (2022, 14 de mayo). Wikipedia. [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code)

Lo desarrolla Microsoft y fue lanzado el 29 de abril de 2015. La versión usada en este proyecto es: 1.67.0.

## 7.2.5 Windows 10

Windows se define como: *Windows 10 es un sistema operativo desarrollado por Microsoft como parte de la familia de sistemas operativos Windows NT.*<sup>35</sup>

Lo desarrolla Microsoft y su fecha de lanzamiento fue 29 de julio de 2015. La versión usada en el desarrollo de este proyecto es: Windows\_NT x64 10.0.19044.

## 7.2.6 Yarn

Yarn (<https://yarnpkg.com>) se define como: *YARN es un gestor dependencias de JavaScript, que está enfocado en la velocidad y la seguridad, y a diferencia de otros gestores como NPM, YARN es muy rápido y muy fácil de usar. (...) Este feedback que ofrece YARN son muy fáciles de leer y con colores, por lo que sin duda es un sistema de gestión dependencias muy interesante.*<sup>36</sup>

He usado este gestor para descargar y actualizar las dependencias entre diferentes ordenadores.

Lo desarrolla Facebook y su fecha de lanzamiento fue el 18 de junio de 2016. La versión usada en el desarrollo es: 1.22.18.

---

<sup>35</sup> Windows 10. (2022, 14 de mayo). Wikipedia. [https://es.wikipedia.org/wiki/Windows\\_10](https://es.wikipedia.org/wiki/Windows_10)

<sup>36</sup> Qué es Yarn. (2022, 14 de mayo). OpenWebinars. <https://openwebinars.net/blog/que-es-yarn/>

## 7.3 Problemas Encontrados

En esta sección describiré algunos de los problemas que he ido encontrando durante el desarrollo, la solución tomada (si la tiene) y otras posibles soluciones (si existiesen). Dentro de todos los problemas que han ido surgiendo, he escogido los que me ha costado más tiempo solucionar o que considere interesantes para futuros programadores que traten de implementar un sistema con unas tecnologías o características similares.

Dentro de este TFG, considero esta sección como una de las principales, ya que existía poca documentación con la combinación de tecnologías escogida.

### 7.3.1 Asincronía al mostrar datos

El primer problema, fue recurrente durante el desarrollo del sistema y es la asincronía al hacer una consulta a base de datos.

NeDB tiene una función llamada “find”, para devolver valores de base de datos. Para controlar la asincronía, uno de los parámetros pasados a esta función, es una llamada callback con otra función. Esto decir, cuando “find” acabe de buscar los valores que necesito, quiero que se ejecute la función pasada como parámetro.

Este flujo de ejecución atentaba contra el diseño que estaba creando. Por ejemplo, si quería mostrar una lista con todas las rutas, debía llamar al método find de NeDB y pasarle una función como callback que crease los componentes de React. Esto no me gustaba ya que estaba haciendo que, desde base de datos, se llamase a una función de UX y considero que debería ser al revés.

La solución que tomé fue envolver la llamada find en una Promesa y devolverla en una función superior. Una Promesa es un tipo de objeto de Javascript, utilizado para tratar con la asincronía. De esta forma, desde el “nivel UX”, podía llamar al “nivel de base de datos” sin “romper” el diseño

```

/**
 * Get all ListRoutes saved in DB.
 *
 * This method is async, so in order to use it, it will be necessary to resolve returned
 * Promise (await or then()).
 *
 * @return Promise<ListRoute[]>
 */
public static async getAllListRoutes(): Promise<ListRoute[]>{
    const db = this.getInstance();

    return await new Promise((resolve, reject) => {
        db.find({type: this.TYPE_LIST_ROUTE}, (err: Error, result: any[]) => {
            if (err) reject(err);
            const listRoutes: ListRoute[] = [];
            // console.log(result);
            result.forEach(function(document: any){
                listRoutes.push(Database.fromDbToListRoute(document));
            });
            resolve(listRoutes);
        });
    });
}

```

Ilustración 7.1 Función para obtener todas las rutas de base de datos

Además de este problema descrito, tuve otros tantos con la asincronía, pero fueron por desconocimiento de cómo controlar el flujo de ejecución correctamente. Cuanto más avanzaba en el desarrollo, iba entendiéndolo más e iba solucionando las fallas más rápidamente.

### 7.3.2 Content-Security-Policy

Este probablemente sea el problema que más tiempo me llevase subsanar. En una fase temprana de desarrollo, intentando mostrar el mapa de GoogleMaps proporcionado por [GoogleMaps Javascript API](#), me aparecía el siguiente error en consola:

```

✖ Refused to load the script 'https://maps.googleapis.com/maps/api/js?callback_index.esm.js?49ce:225
=googleMapsCallback&key=AIzaSyDBsrGdH36Y11o4Vx55Ew-01N LmL-5G6s' because it violates the following
Content Security Policy directive: "script-src 'self' 'unsafe-eval' 'unsafe-inline' data:". Note
that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.

```

Ilustración 7.2 Fallo al cargar la API de GoogleMaps

Content-Security-Policy (CSP), es un estándar para evitar diferentes ataques malignos contra el sistema. Yo no lo había configurado, así que, por defecto, alguna tecnología debía estar haciéndolo. Intenté configurar una CSP válida en mi HTML con la etiqueta <meta> (lo recomendado) y aunque me aparecía en la cabecera, no lo estaba cogiendo:

The screenshot shows the browser's developer tools. The top panel displays the HTML source code, highlighting the Content Security Policy (CSP) meta tag. The console panel below shows several error messages related to the Google Maps script failing to load due to a CSP violation. The error messages are as follows:

- Warning: <https://electronjs.org/docs/tutorial/security>. This warning will not show up once the app is packaged.
- Refused to load the script 'https://maps.googleapis.com/maps/api/js?callback index.esm.js?49ce:225 = \_googleMapsCallback&key=AIzaSyDBsrGdH36Y11o4Vx55Ew-01N LmL-5G6s' because it violates the following Content Security Policy directive: "script-src 'self' 'unsafe-eval' 'unsafe-inline' data:". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.
- Failed to load Google Maps script, retrying in 2 ms.
- Refused to load the script 'https://maps.googleapis.com/maps/api/js?callback index.esm.js?49ce:225 = \_googleMapsCallback&key=AIzaSyDBsrGdH36Y11o4Vx55Ew-01N LmL-5G6s' because it violates the following Content Security Policy directive: "script-src 'self' 'unsafe-eval' 'unsafe-inline' data:". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.
- Failed to load Google Maps script, retrying in 8 ms.
- Refused to load the script 'https://maps.googleapis.com/maps/api/js?callback index.esm.js?49ce:225 = \_googleMapsCallback&key=AIzaSyDBsrGdH36Y11o4Vx55Ew-01N LmL-5G6s' because it violates the following Content Security Policy directive: "script-src 'self' 'unsafe-eval' 'unsafe-inline' data:". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.
- Failed to load Google Maps script, retrying in 24 ms.
- Refused to load the script 'https://maps.googleapis.com/maps/api/js?callback index.esm.js?49ce:225 = \_googleMapsCallback&key=AIzaSyDBsrGdH36Y11o4Vx55Ew-01N LmL-5G6s' because it violates the following Content Security Policy directive: "script-src 'self' 'unsafe-eval' 'unsafe-inline' data:". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.

Ilustración 7.3 Fallo en la CSP con cabecera

Tras investigar, leí que [Webpack](#) configura CSP por defecto, aunque no se le indique. Mediante un plugin intenté configurarla, sin éxito. No entendía de donde podía provenir el error así que abrí una entrada en [StackOverflow](https://stackoverflow.com/q/70761741), sin respuesta (probablemente por mi mala explicación del problema): <https://stackoverflow.com/q/70761741>.

Al final, acabé en esta entrada de Stackoverflow: <https://stackoverflow.com/a/70137354> y pude solucionar el problema, definiendo en el “package.json” de la aplicación una entrada llamada “devContentSecurityPolicy” con mi CSP personalizada.

```
"plugins": [  
  [  
    "@electron-forge/plugin-webpack",  
    {  
      "mainConfig": "./webpack.main.config.js",  
      "devContentSecurityPolicy": "script-src 'self' 'unsafe-eval' https://*.googleapis.com:*",  
      "renderer": {  
        "config": "./webpack.renderer.config.js",  
        "entryPoints": [  
          {  
            "html": "./src/front/index.html",  
            "js": "./src/renderer.ts",  
            "name": "main_window"  
          }  
        ]  
      }  
    }  
  ]  
]
```

*Ilustración 7.4 Solución al problema con CSP*

Esta parada en el desarrollo, me obligó a entender más profundamente las tecnologías que estaba usando, para intentar comprender de donde podía venir el error. Aunque me llevase mucho tiempo, “ayudó” a solventar rápidamente futuros problemas.

### 7.3.3 Distancia total de una ruta

La premisa es que, los ficheros TCX proporcionan un campo con la distancia total de la ruta, pero GPX no. Incluso si ambas extensiones lo hiciesen, deberíamos tener en cuenta que el valor pueda venir vacío o que futuras extensiones a incorporar en el sistema, no lo admitan.

La conclusión es que este valor, si no viene definido en el archivo, debe ser calculado. La primera solución que nos viene a la cabeza es calcular la distancia desde la coordenada A a la coordenada B y repetir este proceso para toda la secuencia de coordenadas. Esto falla porque la tierra tiene forma elipsoide y el resultado presentaría un porcentaje de error considerable.

Tras buscar cómo calcularlo teniendo en cuenta lo explicado previamente, encontré esta solución: <https://stackoverflow.com/a/570048>. En él se provee una formula donde, se toma la tierra como una esfera (la diferencia con un elipsoide para este cálculo es mínima) y se suman las sumas parciales entre coordenadas. Dado que la formula está expresada en lenguaje SQL, tuve que traducirlo a Typescript. Tras probar el cálculo y comparar mis resultados con los datos obtenidos en otras webs (Strava y Wikiloc), las diferencias son nulas o inapreciables.

```
/**
 * This function is translated from this:
 * https://stackoverflow.com/a/570048.
 */
private setDistance(): void{
    const DtoR = 0.017453293;
    const R = 6371000; //Meters

    let totalSum = 0;
    for (let i=0; i < this.getCoordinates().length - 1; i++){
        const rLat1 = this.getCoordinates()[i].getLat() * DtoR;
        const rLng1 = this.getCoordinates()[i].getLng() * DtoR;
        const rLat2 = this.getCoordinates()[i+1].getLat() * DtoR;
        const rLng2 = this.getCoordinates()[i+1].getLng() * DtoR;

        const dLat = rLat1 - rLat2;
        const dLng = rLng1 - rLng2;

        const a = Math.pow(Math.sin(dLat/2), 2) + Math.cos(rLat1) * Math.cos(rLat2) * Math.pow(Math.sin(dLng/2), 2);
        const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a))
        const d = R * c;

        totalSum += d;
    }
    this.distance = totalSum;
}
```

Ilustración 7.5 Función para calcular la distancia total

También proponen otras soluciones más precisas basándose en fórmulas de Vincenty.

### 7.3.4 Distinguir una ruta de una lista

Al incorporar listas a la base de datos, surgió el problema de distinguir una ruta de una lista. En una base de datos documental, una lista y una ruta se almacenan como documentos y no hay forma de distinguirlos. Por ejemplo, cuando quería obtener todas las rutas para mostrarlas en el mapa, la consulta me devolvía rutas y listas y al intentar transformarlas a mi propio dominio, el sistema fallaba.

La solución tomada fue añadir un campo extra que indicasen de qué tipo son. Este campo solo se guarda en base de datos y no se pasa a las clases de dominio. Creé dos constantes en la clase para no tener problema si en el futuro decidía cambiar los nombres o añadir más clases.

```
static readonly TYPE_ROUTE = 0;
static readonly TYPE_LIST_ROUTE = 1;
```

Ilustración 7.6. Tipos en base de datos como constantes

Otra posible solución sería, al obtener las rutas en base de datos, comprobar los campos que tiene el modelo, con los campos del objeto que estoy consultando y si coinciden devuelvo el valor. Aunque posible, esta solución no es viable ya que cuando queramos añadir un nuevo campo a un modelo, tenemos que modificar todas las consultas a base de datos sobre ese modelo.

### 7.3.5 Guardar rutas dentro de una lista

Este problema nace al tratar de mostrar las rutas almacenadas dentro de una lista. Concretamente, al obtener de base de datos estas rutas.

En una primera versión, dentro de una lista guardaba solo los ids de las rutas que contenía. Así, con una doble consulta anidada: una para obtener los ids de las rutas y otro para obtener el resto de los datos de la ruta con ese id, obtendría lo que necesitaba. Esta parecía la opción más lógica, pero al anidar la consulta, fallaba a causa de la asincronía.

La solución que implementé fue, no solo guardar los ids, si no guardar también los datos que necesitase mostrar. Es decir, necesitaba hacer un listado con las rutas de una lista. Pues además de almacenar el id, guardé también el nombre en base de datos. Posteriormente, cuando necesitase mostrar los datos de una ruta de uno de esos listados, haría una consulta independiente a NeDB. Este cambio fue muy sencillo de hacer con una base de datos documental, pero que probablemente no hubiese surgido si hubiese usado una base de datos relacional.

### 7.3.6 Mantener el centro y zoom

Este es un problema al que he decidido no darle solución y, aunque pueda parecer que no tiene sentido, tiene su explicación. El problema se da cuando, en la pantalla de visualización de todas las rutas, hemos hecho zoom en una parte del mapa y añadimos una nueva ruta a visualizar o cambiamos de mapa. Esto hará que la página se refresque (esto no es exactamente así, pero nos vale como un símil), el mapa vuelva a su estado inicial, perdamos tanto el zoom como el centro y, por consiguiente, lo que estábamos viendo.

La solución a este problema es usar los webhooks que provee [React](#), concretamente el webhook `useEffect()`. Un webhook sigue el esquema: si pasa X, entonces hay que realizar la acción Y. Aplicado a nuestro caso sería: si cambia el zoom o centro del mapa, entonces guardo el nuevo valor. De esta forma cuando añada nuevas rutas o cambie de mapa, puedo refrescar la página con el centro y zoom previo y sin que el cambio se note.

El problema con esta solución es que los webhooks no se pueden usar en componentes de clases, solo pueden ser usados en componentes funcionales. Esto se explica más detalladamente aquí: <https://es.reactjs.org/docs/hooks-rules.html>.

En este punto tenía dos opciones: refactorizar los componentes de clases creados para los mapas y convertirlos en funcionales o envolver mis componentes de clase con un componente funcional. La primera opción era muy costosa en tiempo y además atentaba contra la idea de seguir un único estándar de crear todos los componentes como clases. La segunda, aunque válida, me parecía poco elegante y volvía a atentar contra lo descrito con anterioridad.

Tras comentarle esto a mi tutor y sopesar ambas soluciones, decidimos que no era una funcionalidad tan necesaria como para cambiar la estructura del proyecto.

### 7.3.7 Representar un archivo KML en GoogleMaps

Este problema me obligó a cambiar la estructura del sistema y viene dado porque la API de GoogleMaps no permite representar sobre su mapa archivos KML almacenados en local.

GoogleMaps provee funciones muy potentes para visualizar archivos KML. Sabiendo esto, la idea inicial que tenía era convertir los archivos que se importasen en la aplicación (.gpx, .tcx o .kml) y convertirlos a .kml, para mostrarlos fácilmente desde GoogleMaps. El problema es que GoogleMaps solo deja mostrar archivos KML que estén alojadas en algún servidor.

```
/**
 * setState is async, so in the second param, we place the functionality that needs the state
 * updated to work. In this case we need the map to be loaded for the kml to be displayed
 */
this.setState({
  map: map
}, () => {
  const kml = new google.maps.KmlLayer({
    url: "https://googlearchive.github.io/js-v2-samples/ggeoxml/cta.kml",
    map: this.state.map
  })
});
```

*Ilustración 7.7 Código para visualizar archivo KML en GoogleMaps*

Esto me llevó a cambiar la estrategia tomada. Decidí transformar los ficheros en clases de mi propio dominio (Lista, Ruta o Coordenada), para después volver a transformarlos en "google.maps.Data.LineString", otro formato que admite la API de GoogleMaps.

Otra posible solución hubiese sido alojar esos ficheros en el servidor local que monta [Node.js](#).

### 7.3.8 Tipo de dato devuelto por NeDB

Como he explicado en apartados previos del documento, una base de datos documental guarda documentos y no sabe de qué clase proviene, ni si el tipo de objeto almacenado está correctamente definido. El problema vino cuando quería mostrar una ruta en el mapa, estando esta almacenada en base de datos.

La solución fue crear un método en mi clase propia Database, que transformase un documento, en el tipo esperado. Por ejemplo, obtenía de base de datos un documento con la ruta que quería mostrar y transformo, campo a campo, el documento en un objeto de tipo Ruta.

```
/**
 * Create a new Route from DB data (document)
 *
 * Whether a route object can be returned directly, I don't know.
 * If an attribute is added to Route, must parse here too
 *
 * @param document Document from database
 * @return Route created
 */
public static fromDbToRoute(document: any): Route {
  const id = document["_id"];
  const name = document["name"];
  const distance = document["distance"];
  const totalTime = document["totalTime"];
  const positiveSlope = document["positiveSlope"];
  const negativeSlope = document["negativeSlope"];
  const averagePace = document["averagePace"];
  const coordinates: Coordinate[] = [];
  document["coordinates"]?.forEach((coord: any) => {
    coordinates.push(new Coordinate(Number(coord["lat"]), Number(coord["lng"]), Number(coord["alt"]), new Date(coord["time"])));
  });

  const route = new Route(name, coordinates, id, Number(distance), Number(totalTime),
    Number(positiveSlope), Number(negativeSlope), Number(averagePace));
  return route;
}
```

*Ilustración 7.8 Función que transforma un documento en una Ruta*

Podría parecer que otra posible solución es castear el documento devuelto a tipo Ruta. Cuando traté de implementar esta solución, a pesar de que parecía que hacía el cast correctamente, fallaba cuando intentaba mostrar los datos. Desconozco la razón exacta de porqué esto pasaba.

Puede que otras bases de datos documentales más avanzadas como MongoDB, provean herramientas para poder saltarte este paso, pero NeDB no lo hace.

### 7.3.9 Otros problemas

Otros problemas menores encontrados durante el desarrollo aun sin solución se describen en Github, en la página "issues": <https://github.com/MrMenchaca/Toure/issues>

## Capítulo 8. Desarrollo de las Pruebas

En este capítulo mostraré el resultado de pasar los tres tipos de prueba definidos y cómo han afectado al sistema final. La estructura para analizar los resultados de cada tipo de prueba es la siguiente:

- Resultados. Muestro los resultados
- Cambios realizados. Fallos o mejoras detectadas en los resultados y que se han implementado en el sistema.
- Conclusiones. Revisión de qué tal han ido las pruebas.

### 8.1 Pruebas de Integración y del Sistema

Ejecutamos las pruebas funcionales ya diseñadas anteriormente y anotamos el resultado que obtenemos, comparándolo con el que especificamos anteriormente.

#### 8.1.1 Resultados

Caso N.1 Importar una ruta		
Prueba	Resultado Esperado	Resultado
Importar una ruta desde el explorador de archivos.	Se añade la ruta a base de datos y al listado general.	Éxito
Prueba	Resultado Esperado	Resultado
Importar una ruta arrastrándola al dropbox.	Se añade la ruta a base de datos.	Éxito
Prueba	Resultado Esperado	Resultado
Intentar importar una ruta con formato incorrecto.	Se notifica al usuario que el formato es incorrecto.	Éxito

Tabla 8.1 Pruebas del Caso N.1 Importar una ruta

Caso N.2 Ver datos de una ruta		
Prueba	Resultado Esperado	Resultado
Importo una ruta y desde la página Rutas, navego hasta la página que muestra los datos de la ruta	Veo los siguientes datos de la ruta importada: - Nombre (con fecha) - Distancia - Tiempo total - Desnivel positivo acumulado - Desnivel negativo acumulado - Ritmo medio - Mapa con la ruta pintada - Perfil de altitud	Éxito
Prueba	Resultado Esperado	Resultado
Importo una ruta y desde la página Mapa, navego hasta la	Veo los siguientes datos de la ruta importada:	Éxito

página que muestra los datos de la ruta	<ul style="list-style-type: none"> <li>- Nombre (con fecha)</li> <li>- Distancia</li> <li>- Tiempo total</li> <li>- Desnivel positivo acumulado</li> <li>- Desnivel negativo acumulado</li> <li>- Ritmo medio</li> <li>- Mapa con la ruta pintada</li> <li>- Perfil de altitud</li> </ul>	
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Visualizando mi ruta, cambio a la vista del IGN.	Se cambia a la vista del IGN y veo mi ruta pintada sobre la vista del IGN.	Éxito
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Visualizando mi ruta, cambio a la vista Satélite.	Se cambia la vista a Satélite y veo mi ruta pintada sobre la vista Satélite.	Éxito
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Visualizando mi ruta, cambio a la vista Relieve.	Se cambia la vista a Relieve y veo mi ruta pintada sobre la vista Relieve.	Éxito

Tabla 8.2 Pruebas del Caso N.2 Ver datos de una ruta

Caso N.3 Eliminar una ruta del sistema		
Prueba	Resultado Esperado	Resultado
Eliminar una ruta importada pulsando el icono de la papelera.	Tras confirmar la acción, se elimina de base de datos y del listado general.	Éxito
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Cancela la acción	Cancela la eliminación y no pasa nada.	Éxito

Tabla 8.3 Pruebas del Caso N.3 Eliminar una ruta del sistema

Caso N.4 Ver todas las rutas del sistema sobre un mapa		
Prueba	Resultado Esperado	Resultado
En la pantalla Mapa, ver todas las rutas importadas pulsando el checkbox "Todas las rutas".	Se pintan todas las rutas sobre el mapa.	Éxito
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Deseleccionar el checkbox "Todas las rutas".	Se despintan todas las rutas del mapa.	Éxito
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
En la pantalla Mapa, ver todas las rutas importadas pulsando uno por uno el checkbox de cada ruta.	<ul style="list-style-type: none"> <li>- Se pintan todas las rutas sobre el mapa.</li> <li>- El checkbox de "Todas las rutas" se selecciona solo.</li> </ul>	Éxito
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Deseleccionar el checkbox de una ruta.	<ul style="list-style-type: none"> <li>- Se despinta la ruta del mapa.</li> <li>- El checkbox de "Todas las</li> </ul>	Éxito

<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
Visualizando las rutas, cambio a la vista del IGN.	Se cambia a la vista del IGN y veo las rutas pintadas sobre la vista del IGN.	Éxito
Visualizando las rutas, cambio a la vista Satélite.	Se cambia la vista a Satélite y veo las rutas pintadas sobre la vista Satélite.	Éxito
Visualizando las rutas, cambio a la vista Relieve.	Se cambia la vista a Relieve y veo las rutas pintadas sobre la vista Relieve.	Éxito

*Tabla 8.4 Pruebas del Caso N.4 Ver todas las rutas del sistema sobre un mapa*

<b>Caso N.5 Crear una lista</b>		
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
En la pantalla Rutas, crear una nueva lista introduciendo el nombre.	Se añade una nueva lista a la base de datos y a listado con las listas.	Éxito
Cancela la acción.	Cancela la creación y no pasa nada.	Éxito

*Tabla 8.5 Pruebas del Caso N.5 Crear una lista*

<b>Caso N.6 Añadir una ruta a una lista</b>		
<b>Prueba</b>	<b>Resultado Esperado</b>	<b>Resultado</b>
En la pantalla rutas, se selecciona el icono + de una de las rutas, se selecciona una lista y se confirma la acción.	Se añade la ruta a la lista seleccionada en base de datos y en la propia lista.	Éxito
En la pantalla rutas, se selecciona el icono + de una de las rutas, se selecciona varias listas y se confirma la acción.	Se añade la ruta a las listas seleccionadas en base de datos y en la propia lista.	Éxito
En la pantalla rutas, se selecciona el icono + de una de las rutas, NO se selecciona ninguna lista y se confirma la acción.	- No se produce ningún cambio - Se notifica al usuario.	Éxito
Cancela la acción.	No sucede nada.	Éxito

*Tabla 8.6 Pruebas del Caso N.6 Añadir una ruta a una lista*

Caso N.7 Ver las rutas de una lista sobre un mapa		
Prueba	Resultado Esperado	Resultado
En la pantalla Mapa, ver todas las rutas de una lista pulsando el checkbox de la lista.	<ul style="list-style-type: none"> <li>- Se pintan todas las rutas de la lista sobre el mapa.</li> <li>- Además se seleccionarán las rutas en los otros lugares del listado en las que aparezcan ("Todas las rutas" u otras listas).</li> </ul>	Éxito
Prueba	Resultado Esperado	Resultado
Deseleccionar el checkbox de la lista	<ul style="list-style-type: none"> <li>- Se despintan las rutas del mapa.</li> <li>- Además se deseleccionarán las rutas en los otros lugares del listado en las que aparezcan ("Todas las rutas" u otras listas).</li> </ul>	Éxito

Tabla 8.7 Pruebas del Caso N.7 Ver las rutas de una lista sobre un mapa

Caso N.8 Eliminar una ruta de una lista		
Prueba	Resultado Esperado	Resultado
En la pantalla Rutas, seleccionamos una lista y eliminamos una ruta de ella con el icono papelera.	<ul style="list-style-type: none"> <li>- Se elimina la "relación" base de datos.</li> <li>- Se elimina del listado de la lista.</li> </ul>	Éxito

Tabla 8.8 Pruebas del Caso N.8 Eliminar una ruta de una lista

Caso N.9 Eliminar una lista del sistema		
Prueba	Resultado Esperado	Resultado
En la pantalla Rutas, en el listado general y eliminamos una lista con el icono papelera.	<ul style="list-style-type: none"> <li>- Se elimina de base de datos.</li> <li>- Se elimina del listado de listas.</li> </ul>	Éxito
Prueba	Resultado Esperado	Resultado
En la pantalla Rutas, dentro una lista, la eliminamos con el icono papelera.	<ul style="list-style-type: none"> <li>- Se elimina de base de datos.</li> <li>- Se elimina del listado de listas.</li> <li>- Se redirige al listado con todas las rutas</li> </ul>	Fallo, se queda en la lista eliminada

Tabla 8.9 Pruebas del Caso N.9 Eliminar una lista del sistema

## 8.1.2 Cambios realizados

Se ha detectado un único fallo que se expone a continuación:

### 8.1.2.1 Tras borrar una lista, redirigir

El problema residía en que, si estoy editando una lista y borro la lista que estoy editando, en vez de redirigirme a la lista por defecto con todas las rutas, se quedaba en la lista eliminada.



Ilustración 8.1 Error al eliminar una lista

Tras subsanar el error, redirige correctamente al eliminar la lista.

## 8.1.3 Conclusiones

Las pruebas han sido muy satisfactorias ya que se ha encontrado un único fallo, el cual hemos podido subsanar. Cabe apuntar que el sistema creado no da pie a muchos errores ya que hay una única entrada de usuario (importar una ruta) y las acciones que puede hacer dentro de la aplicación son muy limitadas.

## 8.2 Pruebas de Usabilidad

Para realizar las pruebas de usabilidad se ha tomado una muestra con cinco usuarios. Durante las pruebas se han dado las indicaciones justas y necesarias para que los usuarios supiesen qué tenían que hacer, interfiriendo así lo menos posible en el resultado final.

### 8.2.1 Resultados

Los resultados obtenidos son los siguientes

Pregunta	Respuestas
<b>¿Usa un ordenador frecuentemente?</b>	
1. Todos los días de la semana.	3
2. Entre 4 y 6 días por semana.	1
3. Entre 1 y 3 días por semana.	1
4. Nunca o casi nunca.	0
<b>¿Qué tipos de actividades realiza con el ordenador principalmente?</b>	
1. Por trabajo.	3
2. Por ocio (videojuegos, películas, series, vídeos...).	1

3. Para buscar información.	1
4. Para usar una o varias redes sociales	0
<b>¿Qué busca principalmente al navegar por la web?</b>	
1. Que sea fácil de usar.	0
2. Que sea intuitivo.	1
3. Que sea rápido.	1
4. Que tenga todas las funcionalidades necesarias.	3
<b>¿Ha usado alguna vez software de gestión de rutas (Strava, Wikiloc, Polar FlowSync, Garmin Connect...)</b>	
1. Sí, lo uso varias veces por semana.	1
2. Sí, lo uso ocasionalmente.	1
3. Sí, lo he probado, pero no lo uso.	0
4. No, nunca.	3
<b>¿Qué busca principalmente en el tipo de programas descritos anteriormente?</b>	
1. Que pueda ver los máximos datos posibles de una ruta.	2
2. Que tenga un fuerte factor social.	0
3. Que pueda ver varios datos de mis rutas a la vez sobre un mismo mapa.	1
4. Que sea muy versátil (Importar diferentes archivos o visualizar las rutas en diferentes mapas).	2

Tabla 8.10 Resultados de las preguntas de carácter general

ID	Prueba	Tiempo	Observaciones
1	Importar una ruta	50s	
2	Borrar una ruta del sistema	14,75s	
3	Crear una lista	17,5s	
4	Añadir una ruta a una lista	31s	- Añadir nombre de la columna - Mensaje de se ha añadido - Añadir una ruta desde una lista
5	Ver las rutas de una lista	9,5s	- Cambiar el icono ya que el lápiz puede llevar a confusión
6	Borrar una ruta de una lista	11,25s	- Indicar algo cuando una lista está vacía
7	Borrar una lista	10,25s	
8	Ver las rutas de una lista en el mapa general	14,75s	
9	Ver los datos de una ruta en concreto	15,25s	- Cambiar el icono
10	Cambiar de mapa	12,75s	

Tabla 8.11 Resultados de las actividades guiadas

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	3	2		
¿Existe ayuda para las funciones en caso de que tenga dudas?		1	1	3
¿Le resulta sencillo el uso de la aplicación?	4	1		
<b>Funcionalidad</b>	<b>Siempre</b>	<b>Frecuentemente</b>	<b>Ocasionalmente</b>	<b>Nunca</b>
¿Funciona cada tarea como espera?	1	4		

¿El tiempo de respuesta de la aplicación es muy grande?				5
¿Ha echado en falta funcionalidad?		1	1	3
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es	1	4		
Los iconos e imágenes usados son		5		
Los colores empleados son		5		
El tamaño de los mapas es	3		2	
Diseño de la Interfaz	Si		No	A veces
¿Le resulta fácil de usar?	3			2
¿El diseño de las pantallas es claro y atractivo?	4			1
¿Cree que el programa está bien estructurado?	4			1
Observaciones				
<ul style="list-style-type: none"> <li>- Indicar el formato de los ficheros que se pueden importar.</li> <li>- Añadir una página de ayuda.</li> <li>- Añadir fecha.</li> <li>- Dar más feedback con lo que está pasando.</li> <li>- Breadcrumbs.</li> <li>- Iconos más grandes.</li> <li>- Información sobre el mapa.</li> </ul>				

Tabla 8.12 Resultado de las preguntas cortas

Aspecto Observado	Notas
Errores leves cometidos	
Errores graves cometidos	
Confusiones	<ul style="list-style-type: none"> <li>- Intentó añadir una ruta a una lista desde la propia lista</li> <li>- Pensó que había añadido la ruta sin seleccionar ningún checkbox</li> </ul>
¿No ha podido llevar a cabo alguna funcionalidad?	
Posibles mejoras	Cambiar iconos en forma de ojo
Observaciones generales	

Tabla 8.13 Resultado del cuestionario para el responsable de las pruebas

## 8.2.2 Cambios realizados

Tras una recopilación y estudio de las observaciones, confusiones y errores cometidos, he decido hacer los siguientes cambios en la aplicación:

### 8.2.2.1 Indicar el formato de los ficheros que se pueden importar

Esta fue una observación dada por un usuario y es mostrar las extensiones que admite el sistema. Decidí mostrar las extensiones admitidas en el texto del Dropbox.

Tras incorporarlo, el resultado es el siguiente:



Ilustración 8.2 Importar

### 8.2.2.2 Mostrar la fecha de la ruta

En ningún lado se mostraba la fecha de realización de la ruta. Para implementar este cambio, decidí anidar al principio del nombre de la ruta, la fecha de realización. De esta forma, también me quitaba el problema de que un usuario pudiese importar rutas con nombres vacíos.

Tras incorporarlo, el resultado es el siguiente:



Ilustración 8.3 Nombre de una ruta

### 8.2.2.3 Añadir información al añadir una ruta una lista

Cuando queríamos añadir una ruta a una o varias listas, nos aparecía el siguiente modal:

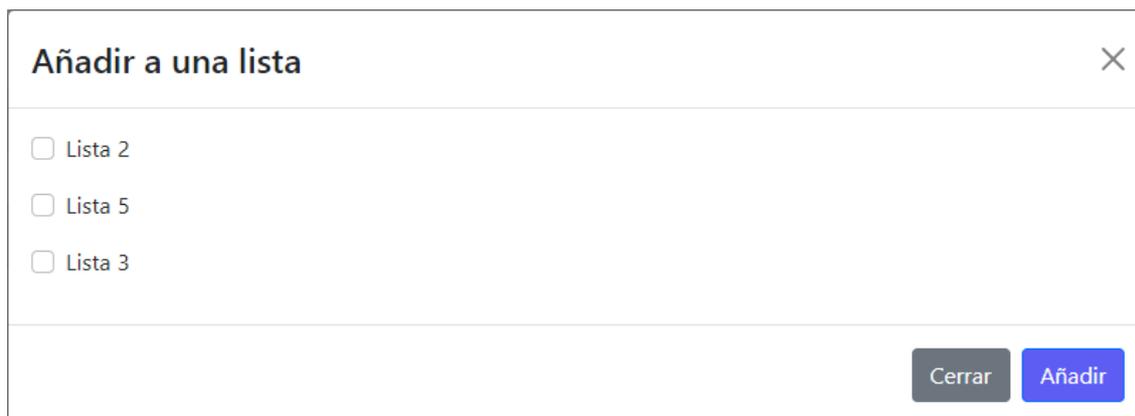


Ilustración 8.4 Modal para añadir una ruta a una o varias listas, antes de las pruebas

Durante una prueba, un usuario pulsó directamente el botón añadir sin seleccionar ninguna lista a la que añadir la ruta seleccionada.

Decidí guiar más al usuario añadiendo el texto: “Debe seleccionar una o varias listas a las que añadir la ruta” y si no hay ninguna lista seleccionada, el botón de Añadir está deshabilitado. Para que esté habilitado, debe haber al menos una lista seleccionada. El resultado es el siguiente:

*Ilustración 8.5 Modal para añadir una ruta a una o varias listas, después de las pruebas*

### 8.2.3 Conclusiones

El resultado de las pruebas ha sido muy satisfactorio ya que todos los testers estaban satisfechos con la aplicación. Además, han sido muy participativos, aportando muchas observaciones e ideas. Estas observaciones; además de errores y confusiones que he ido anotando, a corto plazo se han transformado en mejoras y en un futuro se pueden implementar como futuras funcionalidades.

## 8.3 Pruebas de Accesibilidad

### 8.3.1 Resultados

Los resultados de comprobar la checklist son los siguientes:

#### 8.3.1.1 Nivel A

Prueba	Cumple	No cumple	N/A
<b>1.1.1 Contenido que no es texto:</b> Proporcionar alternativas de texto para contenido que no sea de texto			
<b>1.2.1 Solo audio y solo vídeo (pregrabado):</b> Proporcione una alternativa al contenido solo de vídeo y solo de audio			
<b>1.2.2 Subtítulos (pregrabados):</b> Proporcione subtítulos para vídeos con audio			
<b>1.2.3 Descripción de audio o alternativa multimedia (pregrabado):</b> El vídeo con audio tiene una segunda alternativa			

<b>1.3.1 Información y relaciones:</b> El contenido, la estructura y las relaciones pueden determinarse lógicamente			
<b>1.3.2 Secuencia significativa:</b> Presente el contenido en un orden significativo			
<b>1.3.3 Características sensoriales:</b> Las instrucciones no se basan únicamente en las características sensoriales.			
<b>1.4.1 Uso del color:</b> No utilice una presentación que dependa únicamente del color			
<b>1.4.2 Control de audio:</b> No reproduzca audio automáticamente			
<b>2.1.1 Teclado:</b> Accesible solo con teclado			
<b>2.1.2 Sin bloqueo de teclado:</b> No impida el uso de teclado a los usuarios			
<b>2.1.4 Accesos directos:</b> Proporcionar un mecanismo para configurar los métodos abreviados de teclado			
<b>2.2.1 Temporización ajustable:</b> Los límites de tiempo tienen controles de usuario			
<b>2.2.2 Pausa, Detener, Ocultar:</b> Proporcionar controles de usuario para mover contenido			
<b>2.3.1 Tres destellos o menos:</b> Ningún contenido parpadea más de tres veces por segundo			
<b>2.4.1 Bloques de derivación:</b> Ofrecer a los usuarios la posibilidad de saltarse bloques de contenido repetidos.			
<b>2.4.2 Página titulada:</b> Utilice títulos de página útiles y claros			
<b>2.4.3 Orden de enfoque:</b> Orden lógico			
<b>2.4.4 Propósito del enlace (en contexto):</b> El propósito de cada enlace se desprende claramente de su contexto			
<b>2.5.1 Gestos de puntero:</b> todas las funciones que utilizan gestos multipunto o basados en rutas para hacer una operación se pueden hacer con un solo puntero sin gestos, a menos que sea esencial.			
<b>2.5.2 Cancelación del puntero:</b> Para ayudar a evitar la activación involuntaria de los controles, evite la activación no esencial del evento descendente (por ejemplo, onmousedown) al hacer clic, tocar o presionar la pantalla durante mucho tiempo.			
<b>2.5.3 Etiqueta en Nombre:</b> Para los componentes de la interfaz de usuario con etiquetas que incluyen texto o imágenes de texto, el nombre contiene el texto que se presenta visualmente.			
<b>2.5.4 Motion Actuation:</b> La funcionalidad que se activa moviendo el dispositivo (como agitar o desplazar un dispositivo móvil) o por el movimiento del usuario (como saludar a una cámara) se puede desactivar y la funcionalidad equivalente se proporciona a través de controles estándar como botones			
<b>3.1.1 Idioma de la página:</b> La página tiene un idioma asignado			
<b>3.2.1 Enfoque:</b> Los elementos no cambian cuando reciben enfoque			
<b>3.2.2 En la entrada:</b> Los elementos no cambian cuando			

reciben la entrada			
<b>3.3.1 Identificación de errores:</b> Identificar claramente los errores de entrada			
<b>3.3.2 Etiquetas o Instrucciones:</b> Etiquetar elementos que necesiten de una entrada de usuario			
<b>4.1.1 Análisis:</b> No hay errores de código importantes			
<b>4.1.2 Nombre, Rol, Valor:</b> Construir todos los elementos pensando en la accesibilidad			

Tabla 8.14 WCAG Nivel A

### 8.3.1.2 Nivel AA

Prueba	Cumple	No cumple	N/A
<b>1.2.4 Subtítulos (en vivo):</b> Los vídeos en directo tienen subtítulos			
<b>1.2.5 Descripción de audio (pregrabada):</b> los usuarios tienen acceso a la descripción de audio para el contenido de vídeo			
<b>1.3.4 Orientación:</b> El contenido no restringe su vista y operación a una sola orientación de visualización, como vertical u horizontal, a menos que una orientación de visualización específica sea esencial			
<b>1.3.5 Identificar propósito de entrada:</b> los campos de entrada que recopilan ciertos tipos de información de usuario tienen definido un atributo de autocompletar adecuado			
<b>1.4.3 Contraste (mínimo):</b> La relación de contraste entre texto y fondo es de al menos 4,5:1			
<b>1.4.4 Cambiar el tamaño del texto:</b> El texto se puede redimensionar al 200% sin pérdida de contenido o función			
<b>1.4.5 Imágenes de texto:</b> No utilizar imágenes para el texto			
<b>1.4.10 Reflujo:</b> No se pierde de contenido o funcionalidad y se evita el desplazamiento horizontal cuando el contenido se presenta con un ancho de 320 píxeles			
<b>1.4.11 Contraste no de texto:</b> Existe una relación de contraste de al menos 3:1 para diferenciar objetos gráficos (como iconos y componentes de gráficos o gráficos) y componentes de interfaz personalizados por el autor (como botones, controles de formulario y enfoque indicadores/esquemas)			
<b>1.4.12 Espaciado de texto:</b> No se produce pérdida de contenido o funcionalidad cuando el usuario adapta la altura de la línea de texto/espaciado a 1,5 veces el tamaño de la fuente, el espaciado de párrafo a 2 veces el tamaño de la fuente, el espaciado de palabras a .16 veces el tamaño de la fuente y el espaciado de letras a .12 veces el tamaño de la fuente			
<b>1.4.13 Contenido en Hover o Focus:</b> Cuando se presenta contenido adicional en el hover del cursor o en el focus del			

teclado: el contenido que aparece se puede descartar (generalmente a través de la tecla Esc) sin mover el puntero o el foco del teclado, a menos que el contenido presente un error de entrada o no oculte ni interfiera con el contenido de otra página; el puntero se puede mover al nuevo contenido sin que el contenido desaparezca; el nuevo contenido debe permanecer visible hasta que el puntero o el foco del teclado se aleje del control desencadenador, se descarte el nuevo contenido o el nuevo contenido ya no sea relevante			
<b>2.4.5 Múltiples maneras:</b> Ofrecer varias maneras de encontrar páginas			
<b>2.4.6 Encabezados y etiquetas:</b> Utilice encabezados y etiquetas transparentes			
<b>2.4.7 Enfoque visible:</b> Asegúrese de que el foco del teclado sea visible y claro			
<b>3.1.2 Idioma de las piezas:</b> Indique a los usuarios cuándo cambia el idioma de una página			
<b>3.2.3 Navegación consistente:</b> Utilice los menús de forma coherente			
<b>3.2.4 Identificación consistente:</b> Utilice iconos y botones de forma consistente			
<b>3.3.3 Sugerencia de error:</b> Sugerir correcciones cuando los usuarios cometen errores			
<b>3.3.4 Prevención de errores (Legal, Financiero, Datos):</b> Reducir el riesgo de errores de entrada de datos confidenciales			
<b>4.1.3 Mensajes de estado:</b> los mensajes de estado se programar considerando roles o propiedades del usuario y presentarse al mismo mediante tecnologías de asistencia sin recibir el foco.			

Tabla 8.15 WCAG Nivel AA

### 8.3.2 Cambios realizados

Tras una primera revisión, he visto que mi aplicación no cumple los requisitos del nivel A. Para que los cumpla, he corregido dos fallos:

#### 8.3.2.1.1 Teclado

El error venía porque no permitía que el usuario pudiese navegar por toda la aplicación solo con el teclado.

Los únicos elementos no accesibles eran los iconos que, al no definirlos como botones, no proveían esta funcionalidad por defecto. Para que pudiesen ser accesibles por teclado tuve que acomodarlos de la siguiente forma:

```
<MdEdit
  onClick={() => this.props.changeList(null)}
  tabIndex={0}
  onPress={(e) => {if(e.key === 'Enter') this.props.changeList(null)}}/>
```

Ilustración 8.6 Código que muestra un icono accesible por teclado

Tras esta revisión, ahora toda la aplicación es accesible por teclado.

### 8.3.2.2 3.3.1 Identificación de errores.

Aquí el problema era que, si el usuario intentaba importar un formato de fichero no válido, no se notificaba que hubiese un error, simplemente no lo importaba.

Como solución, he implementado un modal que notifica este error

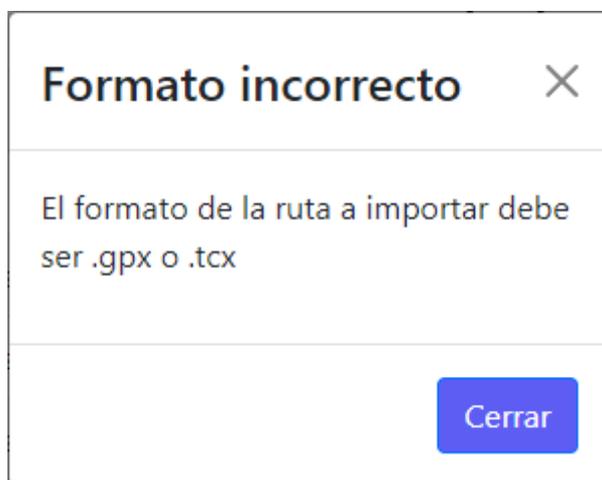


Ilustración 8.7 Modal que notifica un formato de archivo incorrecto

## 8.3.3 Conclusiones

Tras las correcciones, puedo concluir que mi aplicación cumple con el nivel A de accesibilidad del WCAG.

Muchos de las revisiones no aplican en mi sistema ya que, como expliqué en el diseño, mi sistema es una aplicación de escritorio que usa tecnologías web, pero la checklist usada está pensada para una página web.

Paralelamente, también sacamos en claro que aún queda mucho margen de mejora. Podríamos corregir los errores: 1.4.13 Contenido en Hover o Focus, 2.4.5 Múltiples maneras, 3.3.3 Sugerencia de error y 4.1.3 Mensajes de estado, para que la aplicación cumpliera el nivel AA. Tras esto, podríamos revisar el nivel AAA y subsanar los errores que fuesen surgiendo.

## Capítulo 9. Manuales del Sistema

Este capítulo está dedicado en dar a conocer a cualquier tipo de lector (usuario o desarrollador), qué debe hacer para arrancar el sistema y algunos consejos si quisiese modificarlo.

### 9.1 Manual de Instalación

No es necesario ningún tipo de instalación para usar el sistema. Todos los paquetes necesarios están contenidos en el zip con el proyecto.

## 9.2 Manual de Ejecución

El sistema se entregará en un .zip, con una distribución para ejecutarse en un sistema Windows. Si quisiese correrse en otro sistema operativo, habría que empaquetarlo para ese sistema operativo. Esto se describe en el [Capítulo 9.4.3 Empaquetar](#). Lo que nos encontraremos dentro de la carpeta al descomprimirlo es algo similar a esto:

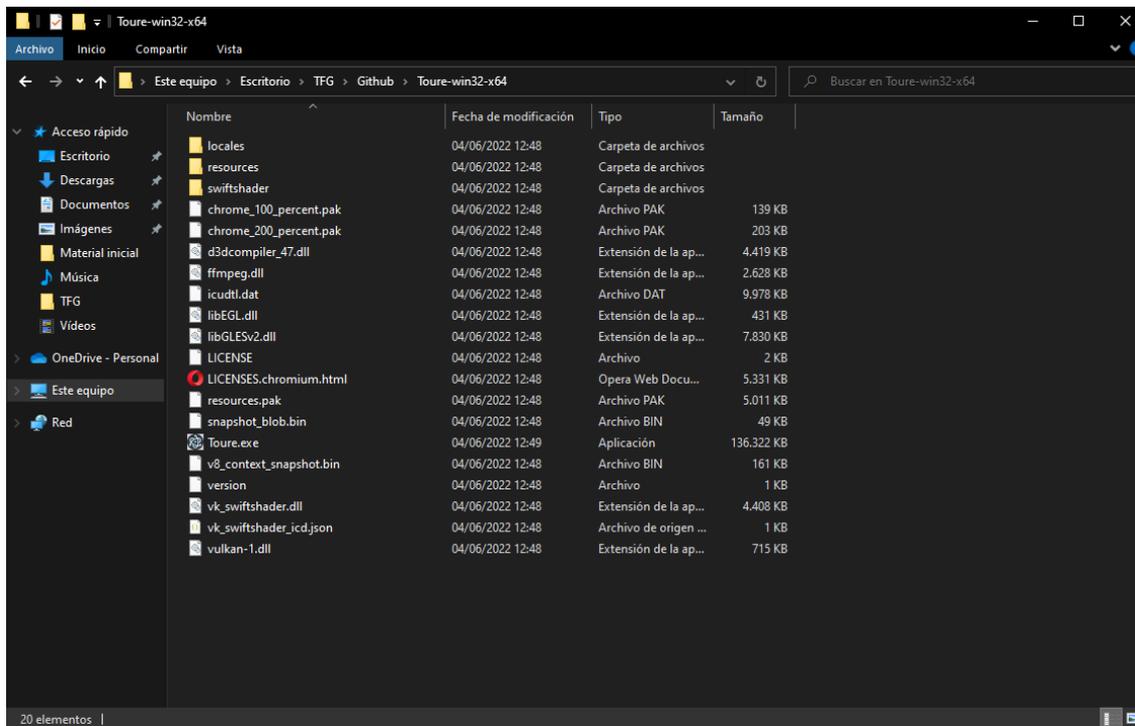


Ilustración 9.1 Distribución para Windows

Para correrlo, simplemente deberemos hacer doble clic en el ejecutable correspondiente, en este caso "Toure.exe" y se abrirá la aplicación.

## 9.3 Manual de Usuario

En este apartado, explicaré, a modo general, cómo funciona cada pantalla de la aplicación. Esto lo hago por si el usuario tuviese alguna duda sobre el funcionamiento.

La pantalla inicial es la siguiente:

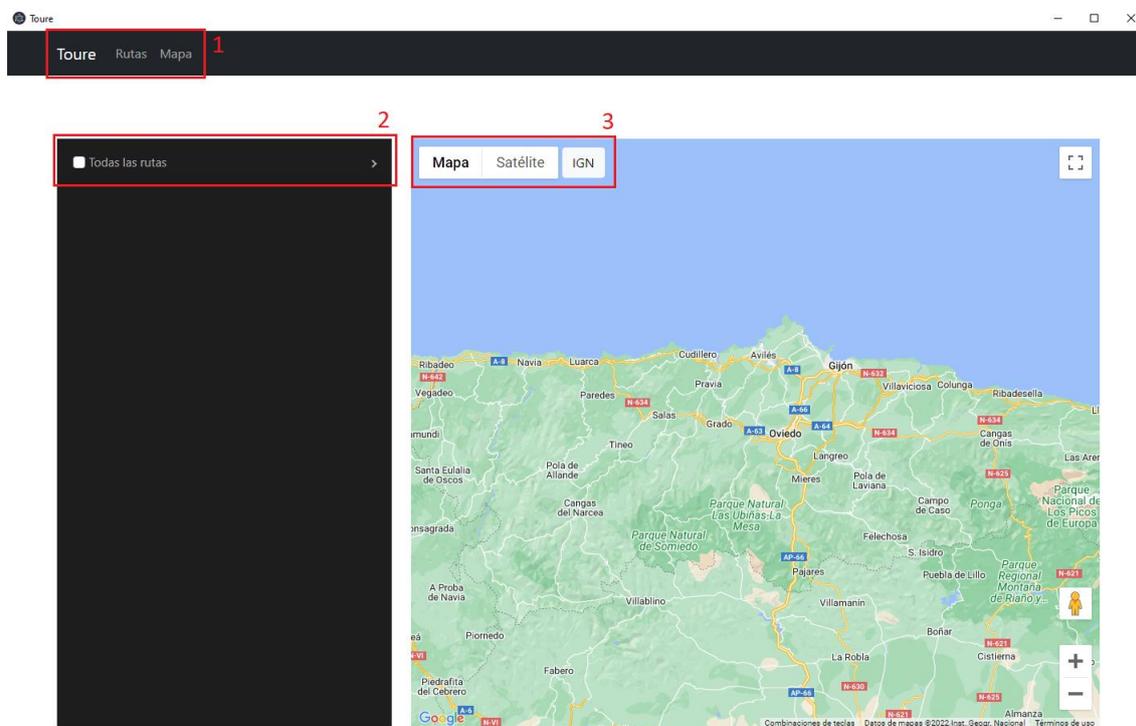


Ilustración 9.2 Pantalla inicial

Dentro de esta pantalla distinguimos las siguientes secciones:

1. Desde la barra de navegación podemos viajar dentro de la aplicación:
  - a. Toure y Mapa es la página actual, el mapa general
  - b. Rutas es la página usada para gestionar las rutas
2. Cuando hayamos importado alguna ruta o creado alguna lista aparecerán aquí
3. Con estos botones podemos cambiar entre los diferentes mapas

Si viajamos hasta la página “Rutas”, nos aparecerá la siguiente pantalla:



*Ilustración 9.3 Página rutas*

El primer paso para comenzar a usar la aplicación será importar nuestras propias rutas. En el zip se adjuntan algunas rutas para probar la aplicación. Para importarlas usaremos la sección de Importar, marcada con un 1 en la imagen. Arrastramos las rutas o las seleccionamos desde el explorador del archivo.

El resultado tras importar las rutas será algo similar a esto:



Ilustración 9.4 Pantalla con rutas importadas

Ahora podremos crear una lista y añadir varias rutas a ella:

1. Con este icono podremos crear una nueva lista escribiendo el nombre que queramos darle.
2. Con este icono podremos añadir la ruta a la nueva lista creada. Si tuviésemos más listas creadas, podríamos añadir la ruta a varias listas a la vez.
3. Con este icono podremos ir a visualizar la ruta a la página dedicada para ello.
4. Con este icono podremos borrar la ruta del sistema confirmando la acción.
5. Con un icono similar a este, pero en la ruta recién creada, podremos ver y editar la lista creada y sus rutas.

Tras crear una nueva lista y añadir varias rutas a ella, vamos a ver qué podemos hacer con ella:



Ilustración 9.5 Pantalla editando una ruta

En esta pantalla podremos realizar las siguientes acciones:

1. Al igual que en la lista con todas las rutas, podemos visualizar la ruta en la pantalla dedicado para ello
2. Con este icono podremos eliminar la ruta de la lista actual
3. Con este icono podremos eliminar la lista del sistema

Si volvemos a la pantalla inicial, con el mapa general, debería verse algo similar a los siguiente:

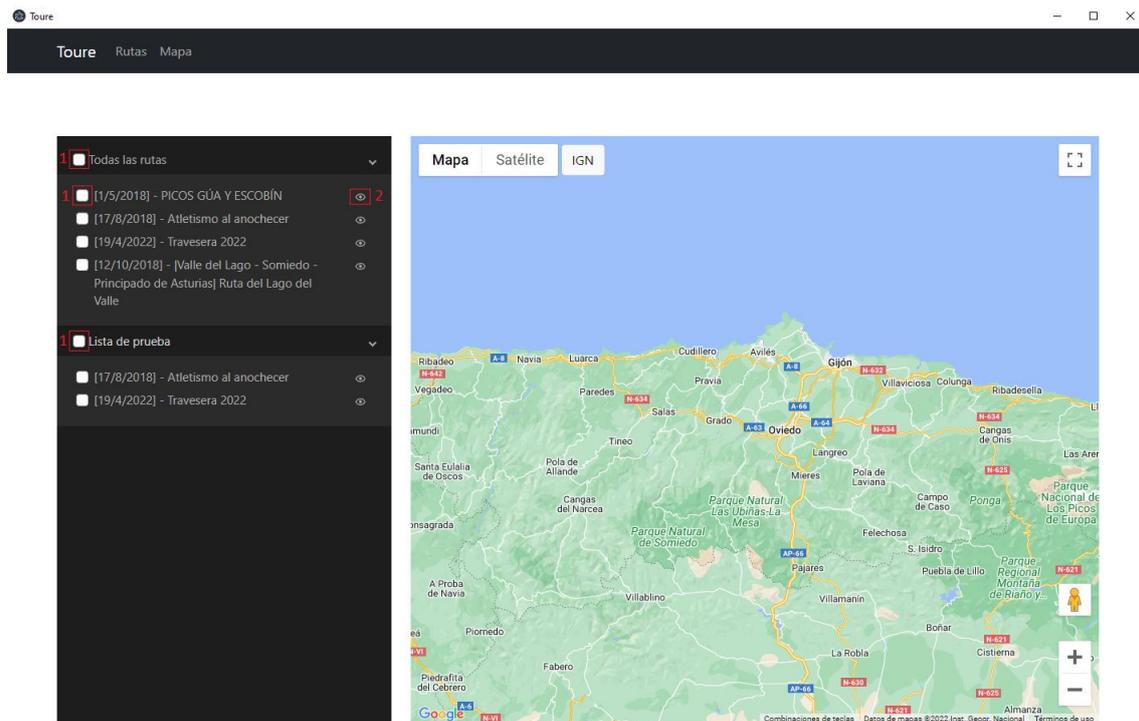


Ilustración 9.6 Mapa general con varias rutas importadas

Podremos visualizar las diferentes rutas:

1. En función del checkbox que seleccionemos, nos pintará sobre el mapa todas las rutas, las rutas de una lista o una ruta individual.
2. Este icono es similar a los mostrados en la página de Rutas y se usa para visualizar la ruta individual en la página dedicado para ello.

Si seleccionamos el checkbox de la lista creada, el resultado debería ser similar a los siguiente:

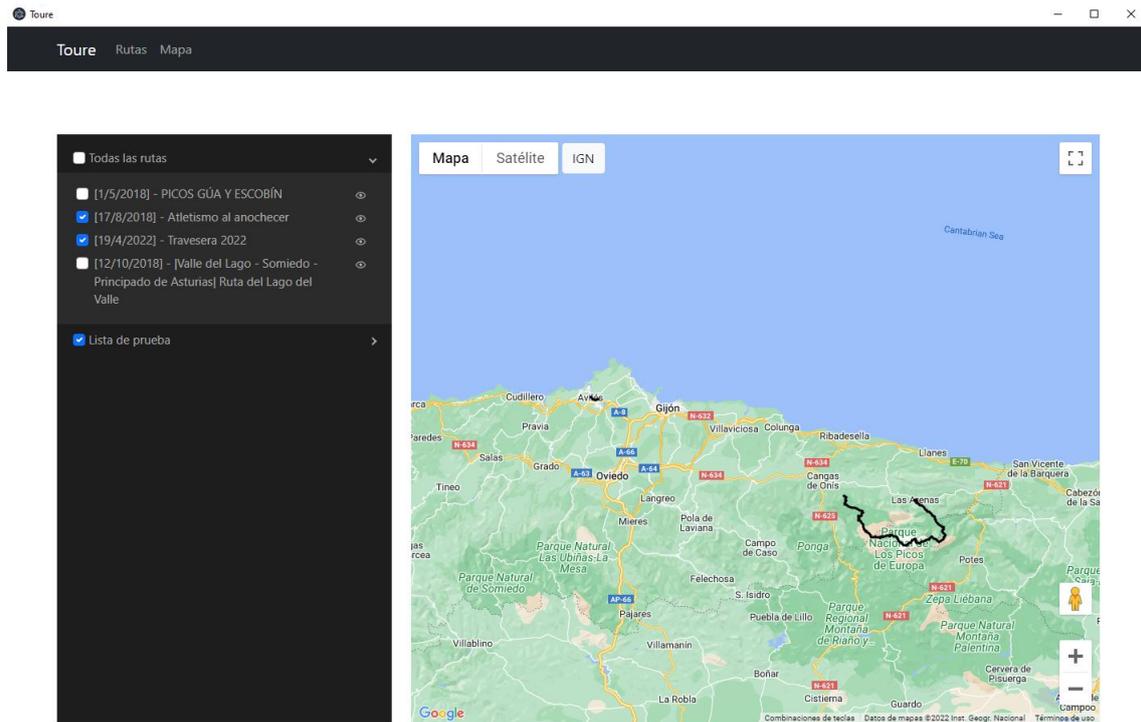


Ilustración 9.7 Mapa general con rutas pintadas

Si viajamos hasta la página para inspeccionar una ruta individual (con los iconos con un ojo), la pantalla mostrada debería ser similar a la siguiente:

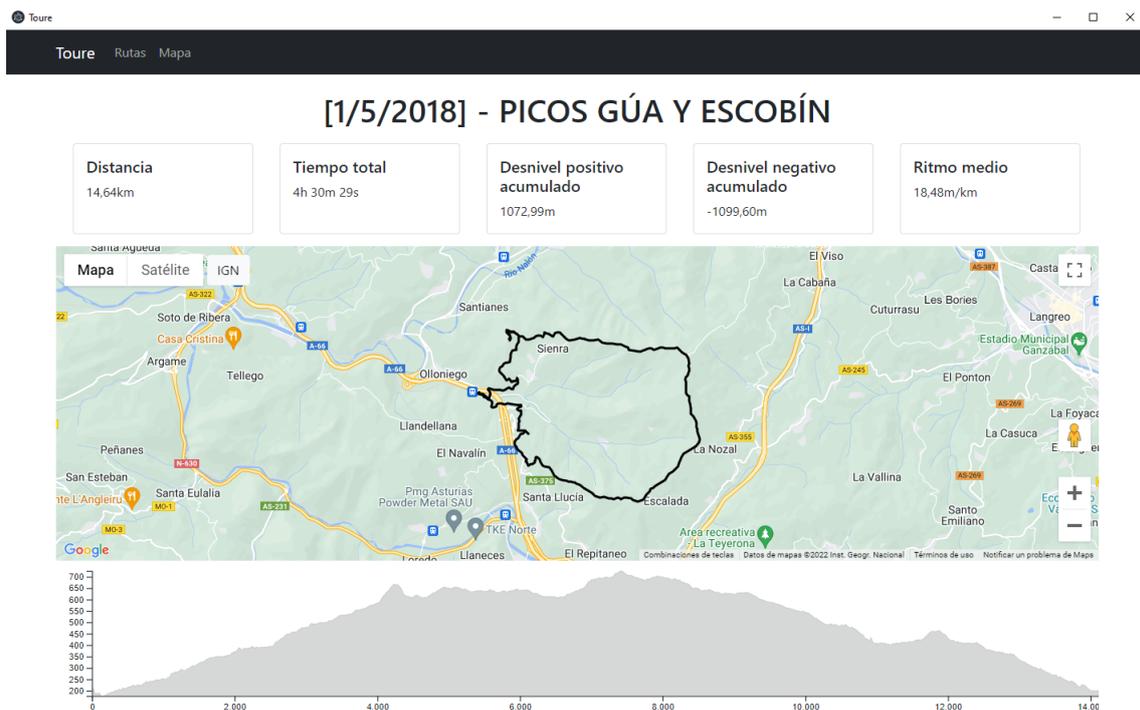


Ilustración 9.8 Visualización de una ruta individual

En esta pantalla no se podrá realizar ninguna acción y los datos mostrados son todos los que el usuario puede ver.

## 9.4 Manual del Programador

A continuación, detallaré qué debe hacer un nuevo programador si quiere desarrollar nueva funcionalidad o modificar la ya existente. Además, proveeré ideas de futuras funcionalidades que quedaron en el tintero.

### 9.4.1 Inicio del sistema

La aplicación entregada junto con esta documentación está destinada al usuario y se utiliza únicamente para probarla. Si queremos seguir desarrollando el sistema, necesitamos el código fuente. Para ello, debemos seguir los siguientes pasos:

1. Clonar el proyecto de Github desde el siguiente enlace: <https://github.com/MrMenchaca/Toure.git>. Se hace una explicación más detalla en el [Capítulo 13.4 Código fuente](#).
2. Una vez clonado, necesitamos el gestor de paquetes npm. Para ello deberemos descargar Node.js (<https://nodejs.org/es/download/>). Una vez descargado Node, desde una terminal ejecutamos el comando “npm install”.
3. Adicionalmente, puede que haga falta descargar Yarn (<https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable>) y ejecutar el comando “yarn install”.

Con esto, ya estaremos listos para modificar el sistema. Adicionalmente, el comando que inicia el sistema es “npm start”.

### 9.4.2 Ayuda para modificar el código

En esta sección, describiré posibles modificaciones que pueden surgir y qué se debería hacer, en términos generales, para llevarlas a cabo.

#### 9.4.2.1 Nueva clase del modelo

Para añadir una nueva clase al modelo, es necesario tocar en tres sitios:

- Añadir la propia clase, tomando de referencia: “Coordinate.ts”, “Route.ts” o “List.ts”.
- Añadir los métodos que se necesiten en base de datos.
- Añadir el método en base de datos que convierta un documento, en la nueva clase del modelo.

#### 9.4.2.2 Añadir un nuevo dato a un modelo

Para añadir un nuevo dato a una clase del modelo ya existente, será necesario tocar en tres sitios:

- Añadir el nuevo atributo a la clase del modelo, con sus “getters” y “setters” correspondientes
- Añadir el atributo a los métodos en base de datos. Principalmente el método que guarde la clase y el método que transforma el documento devuelto a la clase del modelo.
- Si el dato se obtiene directamente de un archivo, añadirlo a las clases parsers, para que se parsee correctamente.

#### 9.4.2.3 Añadir un nuevo Parser

Aquí simplemente habría que crear una nueva clase que extendiese de AbstractParser, como hace “GpxParser” o “TcxParser” y añadir el ifelse en ParserManager. Esto se ejemplifica y detalla en el [Capítulo 6.1.3 Importar varios tipos de ficheros](#).

#### 9.4.2.4 Añadir un nuevo Mapa

Para añadir un nuevo Mapa habría que crear un nuevo componente de React similar a “IGNMap.tsx” o “GoogleMapsMap.tsx” y crear los métodos las clases de modelo necesarios que transforme los datos de nuestro dominio en lo que lee el nuevo mapa. Esto se detalla en el [Capítulo 6.1.4 Visualizar diferentes mapas](#).

### 9.4.3 Empaquetar

Para empaquetar el sistema correctamente, es necesario correr dos comandos:

1. El primer comando es “npm run make”
2. El segundo comando “npx electron-packager .”

Por defecto, el segundo comando empaquetará la aplicación para Windows x64. Si queremos otro sistema operativo, este se puede cambiar gracias al parámetro `-platform`. Este parámetro y otros se desarrollan en la página oficial del proyecto: [electron-packager](https://github.com/electron/electron-packager) (<https://github.com/electron/electron-packager>).

# Capítulo 10. Conclusiones y

## Ampliaciones

En este capítulo haré una revisión global del proyecto comprobando si ha cumplido expectativas y objetivos propuestos en un principio.

### 10.1 Conclusiones

Haciendo un cómputo global, se ha elaborado un sistema bastante completo y versátil para ver y gestionar rutas de senderismo. Un entorno donde el usuario puede hacer una ruta y analizarla tiempo después, viendo diferentes datos o comparándola con otras rutas hechas con anterioridad.

Hay que destacar que algunos de los objetivos iniciales no se han cumplido y de haberlo hecho, el resultado final podría haber mirado “cara a cara” a otras grandes aplicaciones del mercado como pueden ser Strava o Wikiloc. A pesar de esto, creo que compararse con estas aplicaciones es muy ambicioso, ya que estas son las aplicaciones de referencia en nuestro mercado.

Más allá del resultado del proyecto, el haber usado nuevas tecnologías me ha brindado una experiencia muy positiva y a pesar de que retrasase el proyecto al principio de su desarrollo, ha hecho que descubra las virtudes de nuevas tecnologías como React o Typescript.

Además, y para finalizar, considero haber creado la aplicación sobre un buen diseño, consiguiendo así que sea fácil de extender y modificar por posibles futuros desarrolladores.

## 10.2 Ampliaciones

A continuación, nombraré algunas posibles mejoras que pueden implementarse en un futuro.

- Añadir waypoints. Poder añadir uno varios waypoints a las rutas importadas con información que introduzca el usuario.
- Poder hacer clic en una ruta del mapa general y que te lleve a la página de la ruta individual. Esta funcionalidad se intentó implementar, pero no se pudo llevar a cabo por falta de tiempo. Parece ser que GoogleMaps y OpenLayers proveen métodos para recibir un clic sobre el mapa.
- Hacer un seguimiento en el perfil de altitud y mapa, similar a cómo hacen otras aplicaciones como Strava o Wikiloc. Esta funcionalidad, al igual que la anterior, se intentó implementar, pero no se pudo llevar a cabo por falta de tiempo. Quizás pueda servir de ayuda el siguiente enlace: <https://ahrjarrett.com/posts/2019-02-11-draw-an-interactive-elevation-chart-with-d3-and-react>
- Seleccionar un segmento. La idea es que el usuario pudiese clicar en dos puntos del mapa o del perfil de elevación y que el sistema mostrase información de ese segmento. Alternativamente, podría crearse una nueva ruta con ese segmento.
- Borrar varias rutas a la vez. Añadir un botón para poder borrar varias rutas la vez (una idea es hacerlo mediante checkboxes)
- Start point y end point. Mostrar en el mapa el punto de inicio de la ruta y el punto final.
- Mejoras en la interfaz.

Estos elementos mejorarían el sistema actual, pero no se han implementado, principalmente, por falta de tiempo. Algunas son conceptos generales y otros más concretos que se han intentado implementar, pero no se ha podido llevar a cabo su desarrollo.

# Capítulo 11. Planificación del Proyecto y Presupuesto finales

A continuación, se presentará la planificación final temporal y presupuestal del proyecto en su conjunto, mostrando los cambios realizados.

## 11.1 Planificación Final

La distribución temporal se ha ajustado al plan inicial, pero ha cambiado el contenido del Product Backlog y por consiguiente de los Sprints.

### 11.1.1 Product Backlog final

El Product Backlog ha ido evolucionando con el proyecto, principalmente a raíz de las reuniones tenidas con el cliente, en este caso mi tutor César Acebal. El resultado es el siguiente:

Product backlog	
1	Aprender y practicar las tecnologías empleadas en el proyecto
2	Configuración inicial de la aplicación
3	Visualizar mapa GoogleMaps
4	Visualizar mapa IGN
5	Visualizar varias rutas en ambos mapas
6	Crear el modelo de datos
7	Convertir los datos almacenados en ficheros .tcx y .gpx a datos del modelo
8	Leer y almacenar datos del modelo en BD
9	Visualizar datos de una ruta almacenados en BD
10	Organización de las rutas en listas
11	Mejoras en la interfaz
12	Empaquetar la aplicación
13	Plan de pruebas
14	Finalizar la documentación

### 11.1 Product Backlog final

### 11.1.2 Sprints finales

Los Sprint finales son los siguientes:

Sprint 1	
<b>Inicio</b>	Semana del 1 al 7 de noviembre de 2021.
<b>Duración</b>	3 meses.
<b>Sprint review</b>	Semana del 31 de enero al 4 de febrero de 2022.

<b>Tareas</b>	<ol style="list-style-type: none"> <li>1. Aprender y practicar las tecnologías empleadas en el proyecto.</li> <li>2. Configuración inicial de la aplicación.</li> </ol>
---------------	---

Tabla 11.2 Sprint 1 final

Sprint 2	
<b>Inicio sprint</b>	Semana del 31 de enero al 4 de febrero de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 28 de febrero al 4 de marzo de 2022.
<b>Tareas</b>	<ol style="list-style-type: none"> <li>3. Visualizar mapa GoogleMaps</li> <li>4. Visualizar mapa IGN</li> <li>5. Visualizar varias rutas en los mapas</li> </ol>

Tabla 11.3 Sprint 2 final

Sprint 3	
<b>Inicio</b>	Semana del 28 de febrero al 4 de marzo de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 4 al 8 de abril de 2022.
<b>Tareas</b>	<ol style="list-style-type: none"> <li>6. Crear el modelo de datos</li> <li>7. Convertir los datos almacenados en ficheros .tcx y .gpx a datos del modelo</li> </ol>

Tabla 11.4 Sprint 3 final

Sprint 4	
<b>Inicio</b>	Semana del 4 al 8 de abril de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 2 al 6 de mayo de 2022.
<b>Tareas</b>	<ol style="list-style-type: none"> <li>8. Leer y almacenar datos del modelo en BD</li> <li>9. Visualizar datos de una ruta almacenados en BD</li> <li>10. Organización de las rutas en listas</li> </ol>

Tabla 11.5 Sprint 4 final

Sprint 5	
<b>Inicio</b>	Semana del 2 al 6 de mayo de 2022.
<b>Duración</b>	1 mes.
<b>Sprint review</b>	Semana del 6 al 12 de junio de 2022.
<b>Tareas</b>	<ol style="list-style-type: none"> <li>11. Mejoras en la interfaz</li> <li>12. Empaquetar la aplicación</li> <li>13. Plan de pruebas</li> <li>14. Finalizar la documentación</li> </ol>

Tabla 11.6 Sprint 5 final

## 11.2 Presupuesto Final

Para el presupuesto final se han mantenido las consideraciones tomadas en el presupuesto inicial:

- El desarrollo durará 7 meses y el tiempo se repartirá de la siguiente manera:
  - Sprint 1. 6 horas semanales durante 3 meses
  - Sprint 2, 3 y 4. 10 horas semanales durante 3 meses
  - Sprint 5. 14 horas semanales durante 1 mes
- El único miembro del equipo de desarrollo cobrará 12,5 €/hora
- Calculo con un beneficio del 12%

## 11.2.1 Presupuesto de costes final

A continuación, se desglosará el presupuesto de costes:

Capítulo	Concepto	Cantidad	Coste/Mes(€)	Horas	Coste Total (€)
<b>C.1</b>	<b>Costes indirectos</b>				<b>3023</b>
<b>C.1.1.</b>	<b>Licencias</b>				<b>194</b>
C.1.1.1	Microsoft Office	1	7		49
C.1.1.2	Microsoft Windows 10	1			145
C.1.1.3	GoogleMaps API	1			0
C.1.1.4	Visual Studio Code	1			0
C.1.1.	Github	1			0
C.1.1.4	Sourcetree	1			0
<b>C.1.2.</b>	<b>Equipamiento físico</b>				<b>1100</b>
C.1.2.1	Ordenador	1			900
C.1.2.2	Material de oficina	1			200
<b>C.1.3</b>	<b>Instalaciones</b>				<b>1729</b>
C.1.3.1	Alquiler instalaciones		175		1225
C.1.3.2	Luz		20		280
C.1.3.3	Agua		2		70
C.1.3.4	Gas		30		210
C.1.3.5	Internet		60		420
<b>C.2</b>	<b>Costes directos</b>				<b>3175</b>
<b>C.2.1</b>	<b>Sprint 1</b>			<b>78</b>	<b>975</b>

C.2.1.1	Reuniones con el cliente			2	25
C.2.1.2	Análisis			10	125
C.2.1.3	Diseño			4	50
C.2.1.4	Estudio y elección de tecnologías			62	775
C.2.2	<b>Sprint 2</b>			<b>40</b>	<b>500</b>
C.2.2.1	Reuniones con el cliente			1	12,5
C.2.2.2	Análisis			3	37,5
C.2.2.3	Diseño			4	50
C.2.2.4	Estudio y elección de tecnologías			12	150
C.2.2.5	Desarrollo			18	225
C.2.2.6	Documentación			2	25
C.2.3	<b>Sprint 3</b>			<b>40</b>	<b>500</b>
C.2.3.1	Reuniones con el cliente			1	12,5
C.2.3.2	Análisis			2	25
C.2.3.3	Diseño			5	62,5
C.2.3.4	Estudio y elección de tecnologías			2	25
C.2.3.5	Desarrollo			28	350
C.2.3.6	Documentación			2	25
C.2.4	<b>Sprint 4</b>			<b>40</b>	<b>500</b>
C.2.4.1	Reuniones con el cliente			2	25
C.2.4.2	Análisis			2	25
C.2.4.3	Diseño			5	62,5
C.2.4.4	Estudio y elección de tecnologías			3	37,5
C.2.4.5	Desarrollo			24	300
C.2.4.6	Documentación			2	25
C.2.4.7	Plan de pruebas			2	25

<b>C.2.5</b>	<b>Sprint 5</b>			<b>56</b>	<b>700</b>
C.2.5.1	Reuniones con el cliente			4	50
C.2.5.2	Desarrollo			8	100
C.2.5.3	Documentación			34	425
C.2.5.4	Plan de pruebas			10	125

### 11.7 Presupuesto de costes final

<b>Coste del proyecto</b>	<b>6198</b>
<b>Beneficio (12%)</b>	<b>743,76</b>
<b>Total</b>	<b>6941,76</b>

### 11.8 Resumen de presupuesto de costes con beneficio final

## 11.2.2 Presupuesto del cliente final

La siguiente tabla expone el presupuesto que se le mostrará al cliente

Descripción	Coste
<b>Licencias</b>	217,28
<b>Oficina</b>	3168,48
<b>Análisis y estudio del proyecto</b>	1484
<b>Diseño</b>	252
<b>Desarrollo</b>	1092
<b>Pruebas</b>	168
<b>Documentación</b>	560
<b>Subtotal</b>	<b>6941,76</b>
<b>IVA (21%)</b>	<b>1427,77</b>
<b>Total</b>	<b>8399,53</b>

### 11.9 Presupuesto de cliente final

# Capítulo 12. Referencias Bibliográficas

## 12.1 Libros y Artículos

Libros y artículos usados de alguna forma durante el desarrollo del proyecto o su documentación.

El formato seguido es: *[Autor/es]. (Año de publicación). Título*

- *[J. M. Redondo]. (2007). Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo.*
- *[E. Gamma, R. Helm, R. Johnson, J. Vlissides]. (1994). Design Pattern: Elements of Reusable Object-Oriented Software.*
- *[R. C. Martin]. (2008). Clean Code: A Handbook of Agile Software Craftsmanship.*

## 12.2 Referencias en Internet

Páginas Web consultadas para cualquier aspecto relacionado con el desarrollo del sistema o su documentación.

El formato seguido es: *Autor, Título, Página web, Enlace*

- GPX. Wikipedia. <https://es.wikipedia.org/wiki/GPX>
- GPX: the GPS Exchange Format. Topografix. <https://www.topografix.com/gpx.asp>
- Free GPS POI files. GPS Data Team. <https://www.gps-data-team.com/poi/>
- Infraestructura de Datos Espaciales. Instituto Geográfico Nacional. <https://www.ign.es/web/ign/portal/ide-area-nodo-ide-ign>
- Visualizar servicios de visualización teselados. Instituto Geográfico Nacional. <https://www.ign.es/web/resources/docs/IGNCniq/IDE-API-Google.pdf>
- API CNIG. Github. <https://github.com/IGN-CNIG/API-CNIG/wiki/API-REST>
- API-CNIG. Infraestructura de Datos Espaciales de España. <https://plataforma.idee.es/cniq-api>
- César Alberca. TypeScript VS JavaScript. Adictos al trabajo. <https://www.adictosaltrabajo.com/2018/08/01/typescript-vs-javascript/>
- Home. Typescript. <https://www.typescriptlang.org>
- DefinetlyTyped. Github. <https://github.com/DefinitelyTyped/DefinitelyTyped>
- Home. Electron. <https://www.electronjs.org>
- Introduction. Electron. <https://www.electronjs.org/docs/latest>
- Community. Electron. <https://www.electronjs.org/community#boilerplates>
- Shalitha Suranga. Why you should use an Electron alternative. LogRocket. <https://blog.logrocket.com/why-use-electron-alternative/>
- Home. Neutralinojs. <https://neutralino.js.org>
- Saul Lopez. Electron-forge: como empaquetar una aplicación de Electron. Itelisoft. <https://itelisoft.com/electron-forge-como-empaquetar-una-aplicacion-de-electron/>
- Package and distribute your application. Electron. <https://www.electronjs.org/es/docs/latest/tutorial/quick-start#package-and-distribute-your-application>
- Dawid Karczewski. What Are The Best Frontend Frameworks To Use In 2022?. Ideamotive. <https://www.ideamotive.co/blog/best-frontend-frameworks>
- Dave Bush. TypeScript and Electron The Right Way. Medium. <https://davembush.medium.com/typescript-and-electron-the-right-way-141c2e15e4e1>
- Quick start. Electron. <https://www.electronjs.org/docs/latest/tutorial/quick-start>
- The Strava API. Strava. <https://developers.strava.com>
- Qué nos mueve. Wikiloc. <https://es.wikiloc.com/wikiloc/about-us.do>
- AllTrails. Github. <https://github.com/alltrails>
- Tutorial: Introducción a React. React. <https://es.reactjs.org/tutorial/tutorial.html>
- React with Typescript. Electron forge. <https://www.electronforge.io/guides/framework-integration/react-with-typescript>

- Keaton Laney. *Defining Props in React Function Component with Typescript.* Pluralsight. <https://www.electronforge.io/guides/framework-integration/react-with-typescript>
- Fazt Code. *React & Typescript con Hooks | Ejemplo Práctico* . Youtube. <https://www.youtube.com/watch?v=HyaT88zs2EU>
- Overview. Google Maps Platform. <https://developers.google.com/maps/documentation/javascript/overview>
- Typescript and Google Maps. Google Maps Platform. <https://developers.google.com/maps/documentation/javascript/using-typescript>
- Data Layer. Google Maps Platform. <https://developers.google.com/maps/documentation/javascript/datalayer>
- Adding a Map and Markers to a React Application. Google Maps Platform. <https://developers.google.com/maps/documentation/javascript/react-map>
- Fazt Code. *React Route Dom V6 – Tutorial Práctico desde Cero.* Youtube. <https://www.youtube.com/watch?v=7xRVnmWcTE8&t=1573s>
- Carlos Azaustre – *Aprende JavaScript. APRENDE REACT BÁSICO en 30 MINUTOS* - Tutorial de React.js Desde Cero. ¿Cómo funcionan las Promises y Async/Await en JavaScript? [2022]. Youtube. <https://www.youtube.com/watch?v=EMk6nom1aS4>
- Carlos Azaustre – *Aprende JavaScript.* Youtube. <https://www.youtube.com/watch?v=rKK1q7nFt7M>
- Carlos Azaustre – *Aprende JavaScript. ESTE ES EL JAVASCRIPT QUE NECESITAS PARA APRENDER REACT.* Youtube. <https://www.youtube.com/watch?v=sKsNO-quEAw>
- Codigofacilito. *7.- Curso React - Uso de datos en components Props y State.* Youtube. <https://www.youtube.com/watch?v=u2-FOMMSKGc&t=12s>
- Eincode. *Electron with React JS under 40 min!*. Youtube. <https://www.youtube.com/watch?v=VCl8li22mrA&t=1888s>
- Fazt code. *React Cards | React & Bootstrap.* Youtube. <https://www.youtube.com/watch?v=Y6qiH81r1WI>
- Tech club Tajamar. *React Desktop con Electron js.* Youtube. <https://www.youtube.com/watch?v=a1UM-q-6lV0&t=49s>
- Pablo Almunia. *Introducción a NeDB: una base de datos javascript embebida.* todoJS. <https://www.todojs.com/introduccion-a-nedb-una-base-de-datos-javascript-embebida/>
- NeDB. *Database of Databases.* <https://dbdb.io/db/nedb>
- NeDB. Github. <https://github.com/louischatriot/nedb>
- Home. *Webpack.* <https://webpack.js.org>
- Christopher Townsend. *Electron Content Security Policy error when connecting to my api.* StackOverflow. <https://stackoverflow.com/a/70137354>
- Introduction. *React Bootstrap.* <https://react-bootstrap.github.io/getting-started/introduction/>
- Home. *React icons.* <https://react-icons.github.io/react-icons/>
- Deepak710agarwal. *Differences between Functional Components and Class Components in React.* GeeksforGeeks. <https://www.geeksforgeeks.org/differences-between-functional-components-and-class-components-in-react/>
- Estado y ciclo de vida. *React.* <https://es.reactjs.org/docs/state-and-lifecycle.html>

- Componentes y propiedades. React. <https://es.reactjs.org/docs/components-and-props.html>
- Amanda Fawcett. [Top 10 VSCode extensions to use in 2022](https://www.educative.io/blog/top-vscode-extensions). Educative. <https://www.educative.io/blog/top-vscode-extensions>
- React. Typescript. <https://www.typescriptlang.org/docs/handbook/react.html>
- Jose Manuel Lucas. [Introducción a TypeScript con ReactJS](https://softwarecrafters.io/typescript/introduccion-typescript-react). Software Crafters. <https://softwarecrafters.io/typescript/introduccion-typescript-react>
- José Luis Chacón. [TypeScript: qué es, diferencias con JavaScript y por qué aprenderlo](https://profile.es/blog/que-es-typescript-vs-javascript/). Profile. <https://profile.es/blog/que-es-typescript-vs-javascript/>
- Home. Openlayers. <https://openlayers.org>
- Examples. Openlayers. <https://openlayers.org/en/latest/examples/>
- Matthew Brown. [How to Use OpenLayers Maps in React](https://medium.com/swlh/how-to-incorporate-openlayers-maps-into-react-65b411985744). Medium. <https://medium.com/swlh/how-to-incorporate-openlayers-maps-into-react-65b411985744>
- @types/openlayers. Npm. <https://www.npmjs.com/package/@types/openlayers>

# Capítulo 13. Apéndices

## 13.1 Glosario y Diccionario de Datos

Por orden alfabético, todos los términos que se consideren importantes en la aplicación con una descripción breve de su significado dentro de la aplicación.

- **Extensión:** Son unos caracteres anexados a la derecha del título de un archivo, que se utiliza para que los programas sepan cómo deben leer dicho archivo.
- **Garmin Training Center:** Software desarrollado por Garmin en donde el usuario puede visualizar y administrar información obtenida a través de dispositivos de esta misma empresa.
- **Gpx:** Formato basado en XML, usado para intercambiar datos GPS. Es el formato estándar en aplicaciones que representan rutas o recorridos. Es un formato abierto, por lo que cualquier persona puede adaptarlo a sus necesidades.
- **IGN:** Siglas del Instituto Geográfico Nacional. Es una Dirección General cuyas principales funciones y competencias están relacionadas con la estructura orgánica básica del Ministerio de Transportes, Movilidad y Agenda Urbana.
- **Importar:** Leer los datos del archivo que contienen la rutas.
- **OGC:** Siglas de Open Geospatial Consortium traducido como Consorcio Geoespacial Abierto. Es una organización internacional con más de 500 miembros comprometida con la creación de estándares abiertos e interoperables sobre datos geoespaciales.
- **Strava:** Página web o aplicación móvil para visualizar, administrar y compartir datos geoespaciales.
- **Tcx:** Formato basado en XML usado para intercambiar datos relacionados con deporte (gimnasio, pádel, rutas, etc...). Lo usa principalmente el software Garmin Training Center.
- **Waypoint.** Es una coordenada para ubicar puntos de referencia tridimensionales ubicados en mapas.
- **Wikiloc:** Página web para visualizar, administrar y compartir rutas al aire libre a pie, en bicicleta u otras actividades con datos geoespaciales.
- **Wms:** Siglas de Web Map Service traducido como Servicio Web Mapa. Mapas que representan datos espaciales creados de forma dinámica en formato imagen (PNG, GIF o JPEG). Es un estándar internacional definido por OGC.

## 13.2 Contenido Entregado en los Anexos

Se entrega un archivo con nombre TFG\_v1.0.zip. Una vez se descomprima, la jerarquía de contenidos es la siguiente:

Ruta	Contenido
<code>./Atletismo_al_anocheceer.gpx</code>	Ejemplo de ruta .gpx
<code>./picos-gua-y-escobin.tcx</code>	Ejemplo de ruta .tcx
<code>./-valle-del-lago-somiedo-principado-de-asturias-ruta-del-lago.tcx</code>	Ejemplo de ruta .tcx
<code>./Toure-win32-x64</code>	Contiene los ficheros necesarios para ejecutar el sistema
<code>./Toure-win32-x64/Toure.exe</code>	Ejecutable del sistema

### 13.1 Contenido del proyecto

## 13.3 Código Fuente

El código fuente se encuentra en Github: <https://github.com/MrMenchaca/Toure>. Para descargarlo y probarlo, simplemente habría que seguir los pasos indicados en el Capítulo [9.4.1 Inicio del sistema](#)

### 13.3.1 Contenido del código fuente

La distribución de ficheros en el código fuente es la siguiente:

Ruta	Contenido
<code>./images</code>	Imágenes que se usan en el sistema
<code>./src</code>	Código fuente
<code>./src/back</code>	Carpeta con el código referente a backend
<code>./src/back/database/Database.ts</code>	Código para manejar la base de datos
<code>./src/back/domain/*</code>	Clases del dominio
<code>./src/back/parsers/*</code>	Clases con los parsers para poder importar los archivos con extensión <code>.tcx</code> y <code>.gpx</code>
<code>./src/front</code>	Carpeta con el código referente a frontend
<code>./src/front/pages/components/*</code>	Componentes que usan las páginas y otros componentes
<code>./src/front/pages/*</code>	Páginas del proyecto
<code>./src/front/AppStyle.css</code>	Archivo <code>.css</code> con los estilos
<code>./src/front/index.html</code>	Archivo de entrada de React
<code>./src/main.ts</code>	Archivo de entrada de Electron
<code>./src/renderer.ts</code>	Archivo de entrada de Webpack
<code>./database.db</code>	Base de datos creada una vez se inicie el proyecto
<code>./*</code>	Archivos de configuración y gestión de dependencias

#### 13.2 Contenido del código fuente