



Nombre del alumno
<b>Iván Herrero Galván</b>
Fecha de entrega
<b>16 de mayo de 2025</b>
Curso
<b>Desarrollo de aplicaciones multiplataforma</b>
Tutor del Proyecto
<b>José Alberto Hernanz Albertos</b>

## Contenido

Introducción .....	2
Explicación del problema .....	2
Descripción de la solución .....	3
Planificación .....	4
Previsión de tiempos.....	4
Realidad de tiempos .....	5
Metodología.....	6
Análisis de requisitos .....	7
Requisitos funcionales.....	7
Requisitos no funcionales .....	8
Diseño.....	9
E-R base de datos para obtención de planos.....	9
E-R base de datos para obtención de dimensiones .....	11
Casos de uso.....	12
Diseño cliente-servidor .....	13
Elección del servidor y Base de Datos .....	13
Interfaz gráfica del cliente .....	13
Conexión cliente-servidor .....	13
Diseño de la interfaz .....	14
Codificación .....	15
Creación de base de datos en AWS RDS.....	15
Creación de la app con JavaFX.....	15
Pruebas.....	16
Desglose económico .....	17
Roadmap.....	19

# Introducción

## Explicación del problema

Habiendo trabajado para Viatres Ingenieros, S.L. durante un periodo de tiempo de cinco años, he podido identificar algunos puntos de mejora que ayudarían a la empresa a ser más eficientes en los tiempos de entrega de los informes de Road Survey a los clientes, por medio de la reducción del trabajo necesario por parte de los empleados para la realización de dichos informes.

La metodología de trabajo que maneja la empresa es la siguiente: un cliente establece un origen y un destino para las diferentes maquinarias eólicas u otros componentes, con el objetivo de transportar los componentes desde fabricas o puertos hasta los puntos de construcción de futuros parques eólicos o parques eólicos ya construidos u otro tipo de destinos. En la empresa con esta información se establece un itinerario que discurre desde el punto o puntos de origen proporcionados por el cliente hasta el punto o puntos de destino, también proporcionados por el cliente. Posteriormente se analiza el itinerario para identificar los puntos que puedan presentar dificultades para el paso de los transportes especiales solicitados por los clientes; recalcar que no siempre se transporta la misma maquinaria, pero el abanico de opciones tampoco es excesivamente grande. De estos puntos conflictivos identificados se obtienen una ficha, que muestra la maniobra a realizar, y un plano, el cual analiza la situación señalando objetos tales como: señales, farolas, bordillos..., que se pueden ver afectados por el paso del transporte especial.

Para la realización del plano se obtiene una imagen aérea georreferenciada de diversas fuentes de internet como puede ser Iberpix. Esta imagen se referencia a un archivo de extensión DWG, los cuales se manejan con los softwares de AutoCAD. Este programa, AutoCAD, permite la compatibilidad con otro software, AutoTurn, el cual permite al usuario crear una representación 2D del vehículo que va a transportar el componente, para posteriormente haciendo uso de AutoCAD y la imagen georreferenciada descargada previamente hacer un análisis del trazado del vehículo, el cual mostrara los elementos que se verían afectados en una situación real.

Tras finalizar el análisis de un punto conflictivo, el archivo DWG al igual que el plano en PDF obtenido se almacenan en la carpeta del estudio, lo cual es eficiente para acceder a toda la documentación de dicho estudio, pero dificulta la búsqueda de maniobras analizadas con anterioridad, existiendo situaciones en las que resulta más fácil y rápido realizar todo el proceso de nuevo que buscar el archivo DWG y el plano en PDF realizados tiempo antes.

Añadir otro punto de mejora señalado por Jorge Sardinero, el cual recalco el poder hacer uso de una herramienta que le permitiese realizar búsquedas rápidas y sencillas sobre las dimensiones máximas de los vehículos que han circulado por un punto conflictivo específico, para agilizar la resolución de dudas de los diferentes clientes con los que cuenta el estudio de ingeniería.

## Descripción de la solución

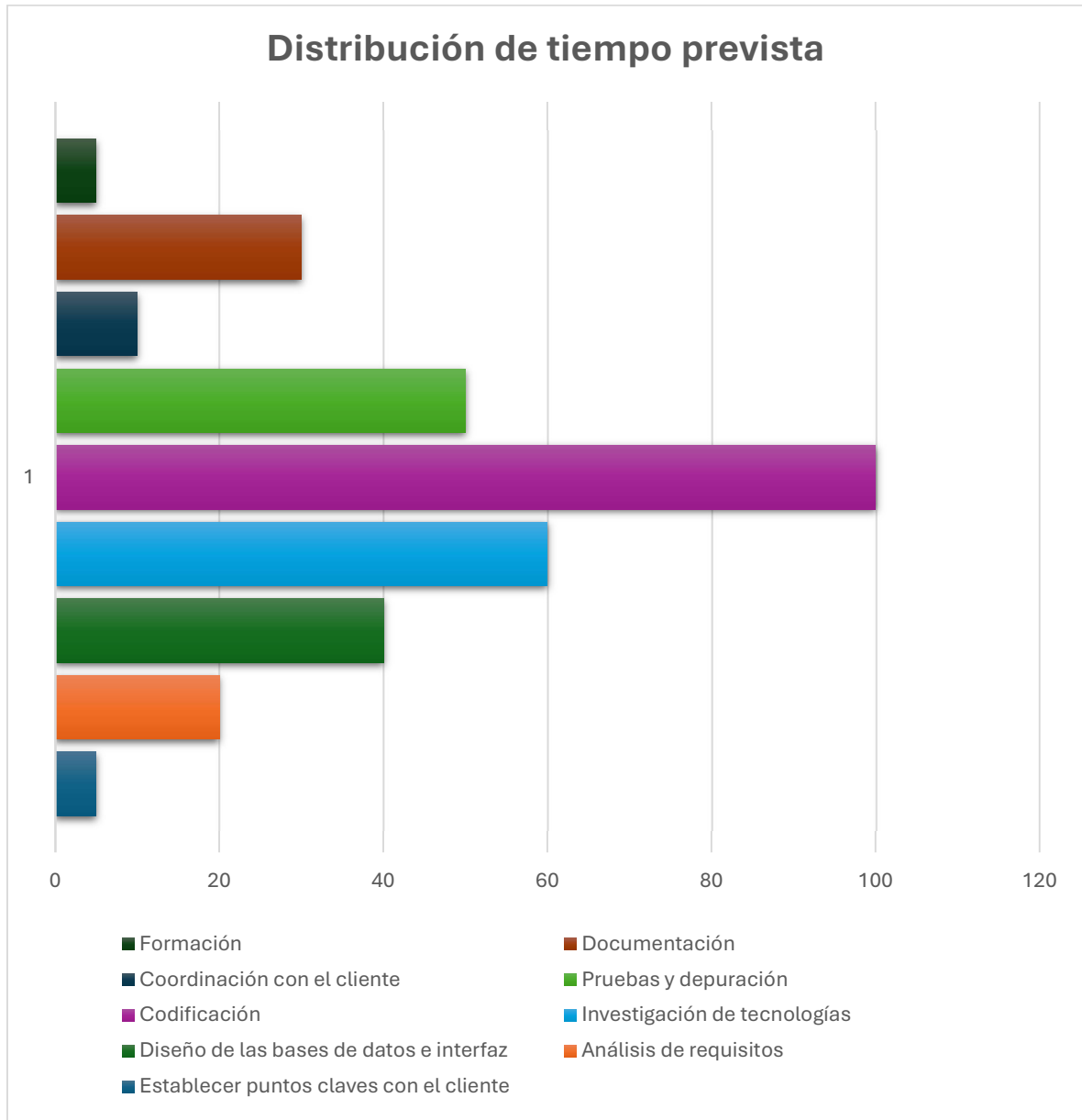
Para resolver los problemas expuestos anteriormente, se desarrollará una aplicación de escritorio, la cual contara con dos opciones: una, realizar búsquedas automáticas de los planos necesarios para completar los informes de Road Survey solicitados por los clientes y, dos, otra opción que permita obtener las dimensiones máximas de los transportes que han circulado por un punto conflictivo específico.

Para la primera opción se desarrollará una base de datos que contemple los estudios, las carreteras, los transportes y los planos, es posible que se necesiten crear más entidades para desarrollar una base de datos sólida. De esta manera se tendrá toda la información almacenada y de forma que sea fácil de localizar a través de una simple búsqueda. Se hará uso de un sistema automatizado que pueda detectar las carreteras de una ruta creada en Google Maps por parte del usuario para posteriormente realizar la búsqueda de forma automática en la base de datos, obteniendo así los estudios de los cuales se pueden reutilizar los planos y que planos se pueden reutilizar de cada estudio. De esta forma solo será necesario modificar la fecha del plano, reduciendo drásticamente el tiempo necesario para la realización de los informes de Road Survey, permitiendo a la empresa manejar un volumen mayor de trabajo, al eliminar la necesidad del trabajador de buscar los planos uno a uno o en el peor de los casos el que tenga que volver a realizar nuevamente la simulación.

Para la segunda opción se realizará una base de datos independiente de la anterior la cual estará formada por diversas entidades, tales como: carretera, transporte, punto conflictivo.... El uso de esta base de datos será incluso más sencillo que la anterior mencionada, ya que solo será necesario introducir la carreta de origen, la carretera de destino (en caso de que exista), el punto kilométrico (si fuese necesario) y la dirección (Madrid, Barcelona, Castellón...) para que automáticamente muestre las dimensiones máximas (longitud, anchura, numero de ejes...) de los transportes que han circulado por ese punto sin que fuese necesario realizar actuaciones.

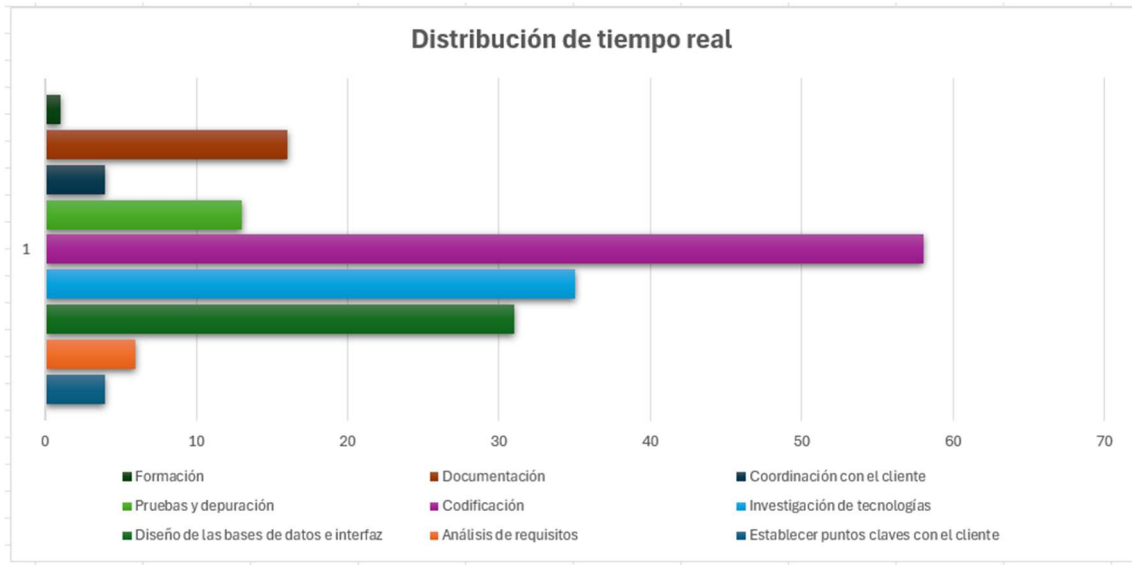
## Planificación

### Previsión de tiempos



Previsión, un total de 320 horas.

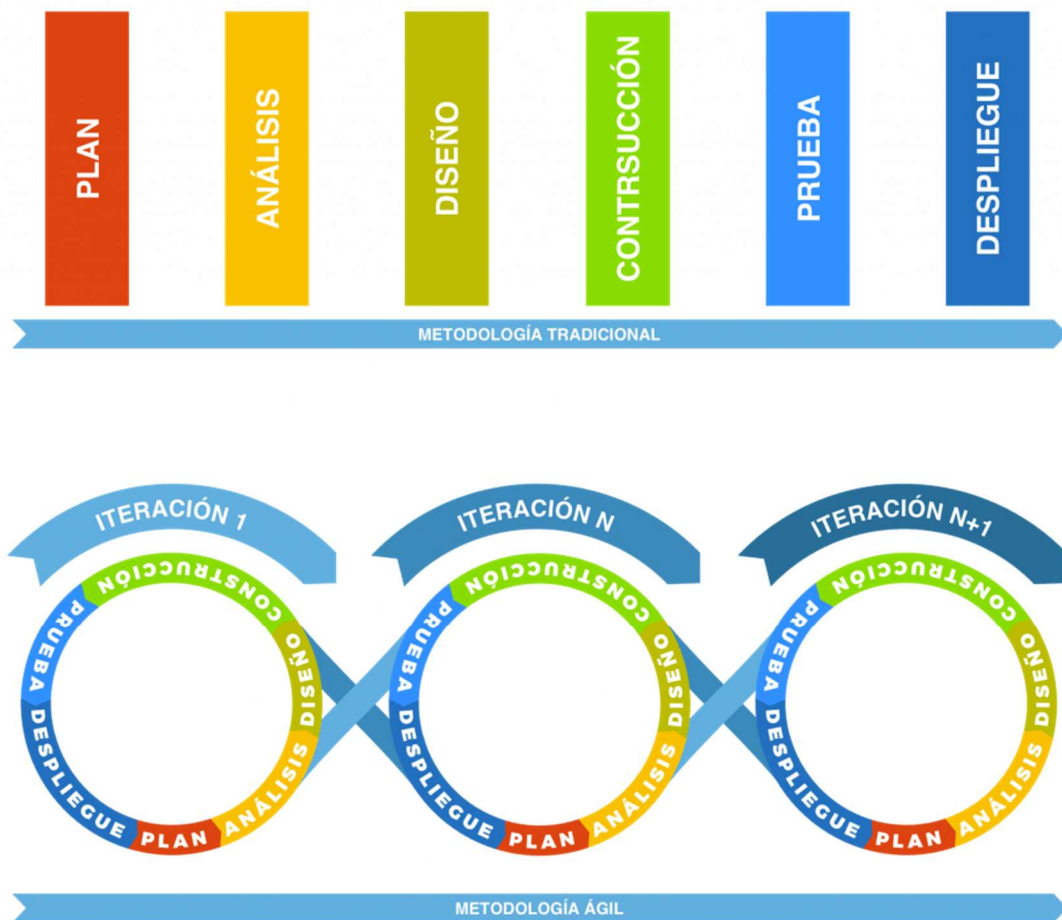
## Realidad de tiempos



Real, un total de 168 horas.

## Metodología

Para el desarrollo de esta aplicación se pretende seguir una metodología scrum, ya que se ha optado por modular la aplicación en partes más pequeñas, para posteriormente ir juntando estas partes, aunque se está abierto a cambios solicitados por el cliente en los diferentes módulos, aunque estos ya estén desarrollados.



<https://www.eniun.com/metodologia-scrum-desarrollo-software-agil/>

# Análisis de requisitos

## Requisitos funcionales

1. Inicio de sesión mediante nombre de usuario asociado a un correo y contraseña.
2. Posibilidad de mantener la sesión iniciada en un dispositivo por parte de los usuarios.
3. Dos posibles roles de usuarios: administrador y empleado
  - 3.1. Administrador
    - 3.1.1. Podrá crear y eliminar usuarios de rol empleado.
    - 3.1.2. Podrá realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre la base de datos. Los borrados se llevarán a cabo de forma lógica.
    - 3.1.3. Tendrá la capacidad de acceder a un historial de creación y modificación de registros de la base de datos en la cual se mostrará el empleado que ha realizado dicha acción.
  - 3.2. Empleado
    - 3.2.1. Podrá crear, modificar y leer registros de la base de datos, pero en ningún caso eliminarlos.
    - 3.2.2. Las modificaciones y creaciones realizadas por usuarios de este rol se guardarán.
4. El sistema contará con dos bases de datos.
  - 4.1. Una base de datos permitirá la búsqueda de planos y de estudios usando como parámetros las carreteras que crean el punto crítico o el punto kilométrico y la nomenclatura del transporte.
  - 4.2. Otra base de datos permitirá realizar consultas rápidas sobre las dimensiones máximas de un transporte que ha circulado por un punto crítico sin la necesidad de realizar actuaciones. Recibirá como parámetros el nombre de las carreteras que forman el punto crítico o el kilómetro en caso de ser necesario. Estos parámetros se introducirán de forma manual por parte del usuario.
5. Los usuarios podrán realizar búsquedas en la base de datos de dos formas:
  - 5.1. Mediante la subida de un archivo PDF. Este archivo contiene las carreteras por las que circulará el transporte y mediante la programación necesaria, se extraerá la información del documento para realizar las búsquedas necesarias en la base de datos de forma automática.
  - 5.2. Señalando el punto de origen y de destino mediante coordenadas obtenidas de Google Maps.
  - 5.3. Además, para ambas búsquedas, el usuario tendría que indicar el nombre del transporte.



## Requisitos no funcionales

1. El sistema debe de ser capaz de soportar las consultas simultaneas de 8 usuarios sin afectar al rendimiento.
2. Las búsquedas en la base de datos deben de poderse realizar en menos de 3 segundos cada una. (En caso de insertar un PDF como fuente de datos se entiende como una única búsqueda cada consulta que se realiza en la base de datos obteniendo los parámetros del PDF. Desde un archivo PDF se realizan más de una consulta de forma automática.)
3. El sistema debe de garantizar operatividad casi de forma ininterrumpida con tiempos mínimos de inactividad.
4. Se debe de garantizar una seguridad correcta para evitar robo de datos, modificación de registros de la base de datos de forma malintencionada por parte de personas externas a la aplicación o evitar inyecciones SQL.
5. La interfaz debe de ser sencilla e intuitiva.
6. El tiempo de aprendizaje de uso de aplicación debe ser inferior a 40 minutos.

## Diseño

### E-R base de datos para obtención de planos

El objetivo de la siguiente base de datos es el poder obtener los estudios o planos que contengan el análisis de los enlaces requeridos con un tipo de vehículo indicado.

Para ello es necesario almacenar los estudios, los cuales estarán formados por varios planos, estos planos pueden pertenecer a varios estudios. Los estudios tendrán un código que los identifique, este código tendrá el siguiente formato, YYMMDD(nº del estudio del día, un ejemplo para el quinto estudio solicitado el 25 de febrero de 2025 sería: 25022505).

Los planos serán la representación de un único enlace, en los cuales se hará el análisis de un único transporte. El código de los planos estará formado por el código del enlace + el código del transporte. Ejemplo enlace A-7 con la A-3 en dirección norte de un transporte con nomenclatura V150: A-7\_ENL\_A-3\_N->N\_V150.

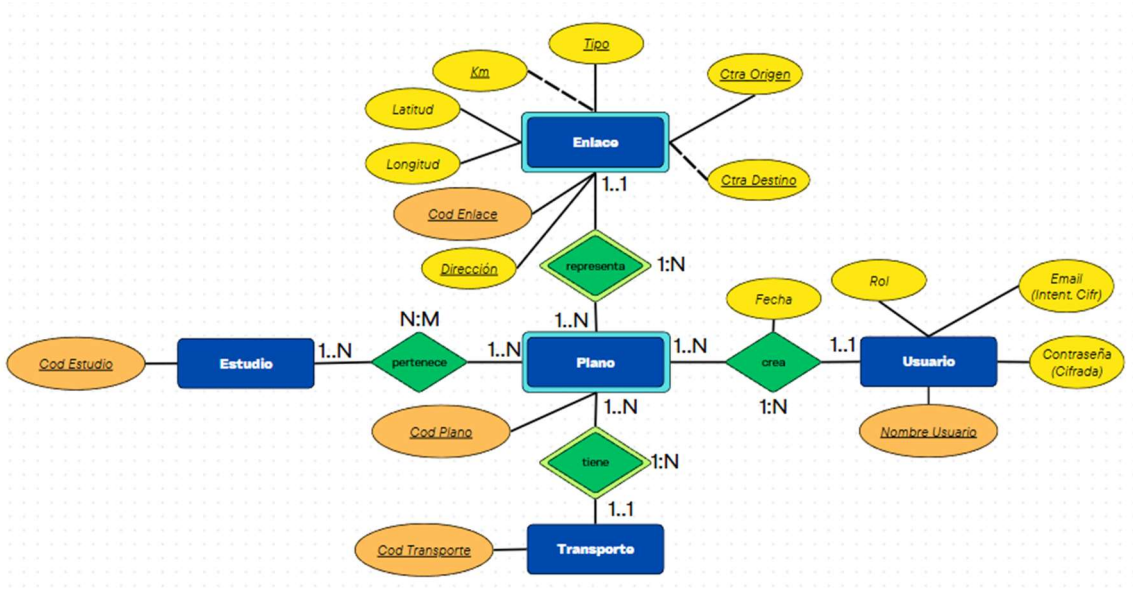
De los transportes solo se guardará su código identificativo el cual es proporcionado por el transportista.

Los planos serán creados por los usuarios de la aplicación, cuyos datos se guardarán en la base de datos. Además, cada usuario tendrá un rol, el cual le proporcionará diferentes permisos dentro de la base de datos. Las contraseñas de los usuarios se almacenarán también como atributo de estos, pero se almacenará el hash de la contraseña y no el texto plano, con el correo electrónico se hará lo mismo para mejorar la seguridad de los usuarios. Del correo se verificará que tenga un formato correcto. Como futuro desarrollo se encuentra la verificación de que el correo introducido exista.

Cada enlace representará un punto crítico, los cuales pueden ser de diferente tipo, tales como glorietas, gálidos, cruces, curvas o salidas los enlaces tendrán un único tipo. Los tipos se identificarán con un código que estará formado por 3 caracteres (Enlace – ENL, glorieta – GLO, gálido – GAL, cruce – CRU, salida - SAL), en caso de ser necesario se podría ampliar el número de tipos. Los puntos críticos siempre tienen una carretera de origen, pero no siempre una de destino, en estos casos será necesario obligatoriamente indicar el punto kilométrico en el que se encuentra (en caso de que indique carretera de destino este atributo obligatoriamente estará vacío). El código de un enlace estará formado por el código de la carretera de origen + el código del tipo + código de la carretera de destino si tiene (en caso contrario y no contar con carretera de destino, este código se sustituirá por el punto kilométrico) + la dirección. Cogemos el ejemplo anterior, enlace A-7 con la A-3 en dirección

norte, el código sería A-7\_ENL\_A-3\_N->N, pero si no tenemos una carretera de destino como puede ser una glorieta o un galibo el código sería de la siguiente manera: glorieta de la N-120 en el kilómetro 306,050 con dirección Valladolid: N-120\_GLO\_306+050\_O->O. Cada enlace contará también con dos atributos que representen las coordenadas en las que se encuentra el punto crítico: latitud y longitud.

Para el código de identificación de las carreteras se usarán los códigos establecidos por el Ministerio de Fomento y los diferentes organismos territoriales. Un sitio web del que sacar estos datos de forma correcta es la página: [SIGCAR. Sistema de Información Geográfico de Carreteras \(beta\)](#).

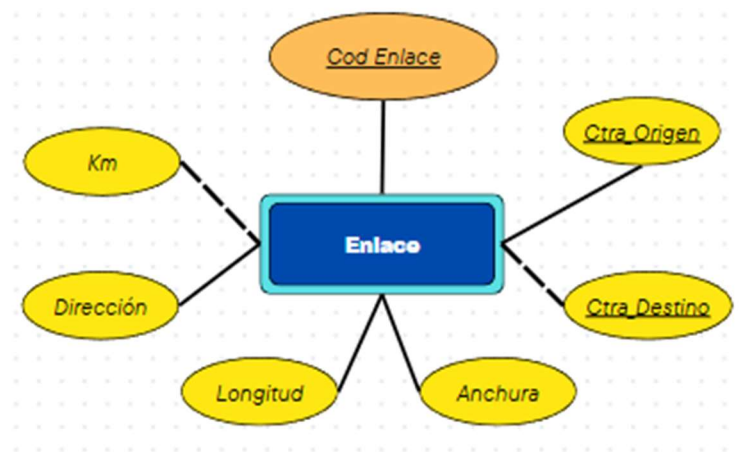


## E-R base de datos para obtención de dimensiones

El objetivo de esta base de datos es la obtención rápida de información sobre las dimensiones máximas de un transporte que ha circulado por un punto conflictivo sin que hayan sido necesarias operaciones de retiradas de señales viales o elementos de protección.

Para ello se reutilizará el elemento Enlace de la anterior base de datos, aunque en enlace se añadirán nuevos atributos, los cuales serán: longitud y anchura; y se eliminarán otros que en este contexto no son necesarios.

De esta forma lo que se busca es que, introduciendo las carreteras de origen y destino o el punto kilométrico en su defecto, se obtengan las dimensiones máximas de circulación sin la necesidad de realizar actuaciones.



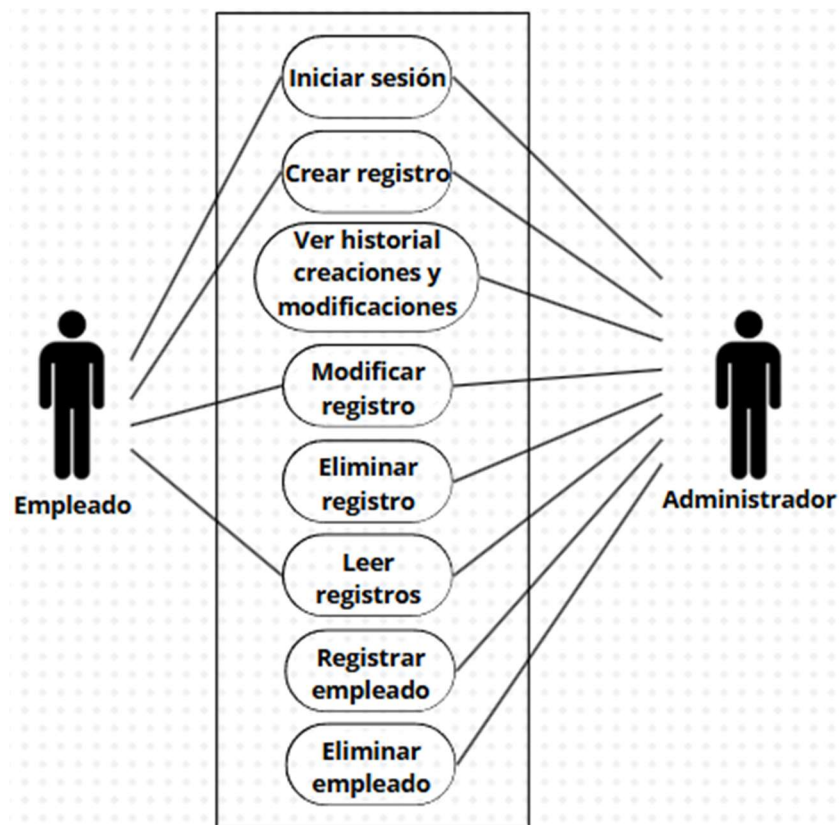
## Casos de uso

Para los casos de uso nos encontramos ante 3 actores principales: empleado, administrador y gestor del software.

El empleado será el rol con menor número de privilegios, contado solo con la capacidad de crear, leer y modificar registros de la base de datos, sin tener la capacidad de realizar eliminaciones. Se podría considerar la posibilidad de que pudiese solicitar la eliminación de un registro al administrador, que en última estancia decidiría si esa eliminación es oportuna o no, esto es necesario confirmarlo con el cliente.

El administrador será el rol con mayor cantidad de permisos destinado a un usuario de la aplicación. Podrá realizar las mismas acciones que el empleado, además, podrá eliminar registros de forma lógica en la base de datos, registrar a un usuario con el rol de empleado y ver un historial de modificaciones y creaciones de registros en la base de datos siempre y cuando estos se hayan hecho por parte de un usuario.

El gestor del software se entiende como la persona externa a la empresa cliente que se encargara del mantenimiento de la aplicación, así como de sus bases de datos, pudiendo realizar modificaciones de estas si el cliente así lo solicita. Puede realizar todas las acciones que realiza el administrador ya que se contempla la posibilidad de que el administrador pueda delegar sus ocupaciones con el software al gestor.



## Diseño cliente-servidor

Para el desarrollo del sistema, se ha optado por una estructura cliente-servidor, en la cual la base de datos estará alojada en un servidor remoto, y los clientes accederán a ella mediante una aplicación de escritorio con interfaz gráfica.

### Elección del servidor y Base de Datos

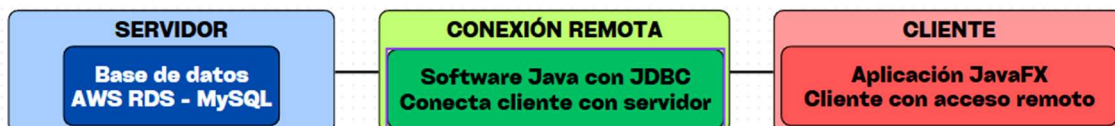
Tras investigar diferentes servicios de base de datos en la nube se ha optado por Amazon Web Services Relational Database Service, que nos permite hacer uso de sus servicios, por unos 8€ al mes, los cuales nos ofrecen 20gb de almacenamiento, lo cual es suficiente para el tipo de datos que vamos a almacenar. Además, podemos ampliar el almacenamiento en caso de que fuese necesario, siendo esta la plataforma la más económica, fiable y sencilla encontrada. Para crear las bases de datos se hará por medio de comandos desde el CMD, ya que nos permite cargar scripts completos. Esto es un punto importante, ya que permitirá agilizar el proceso de creación de la base de datos al tener conocimientos en MySQL.

### Interfaz gráfica del cliente

Para el cliente se opta por el uso de JavaFX para el desarrollo de una interfaz gráfica, ya que los usuarios de este software no tienen los conocimientos informáticos necesarios para manejar una base de datos de forma directa.

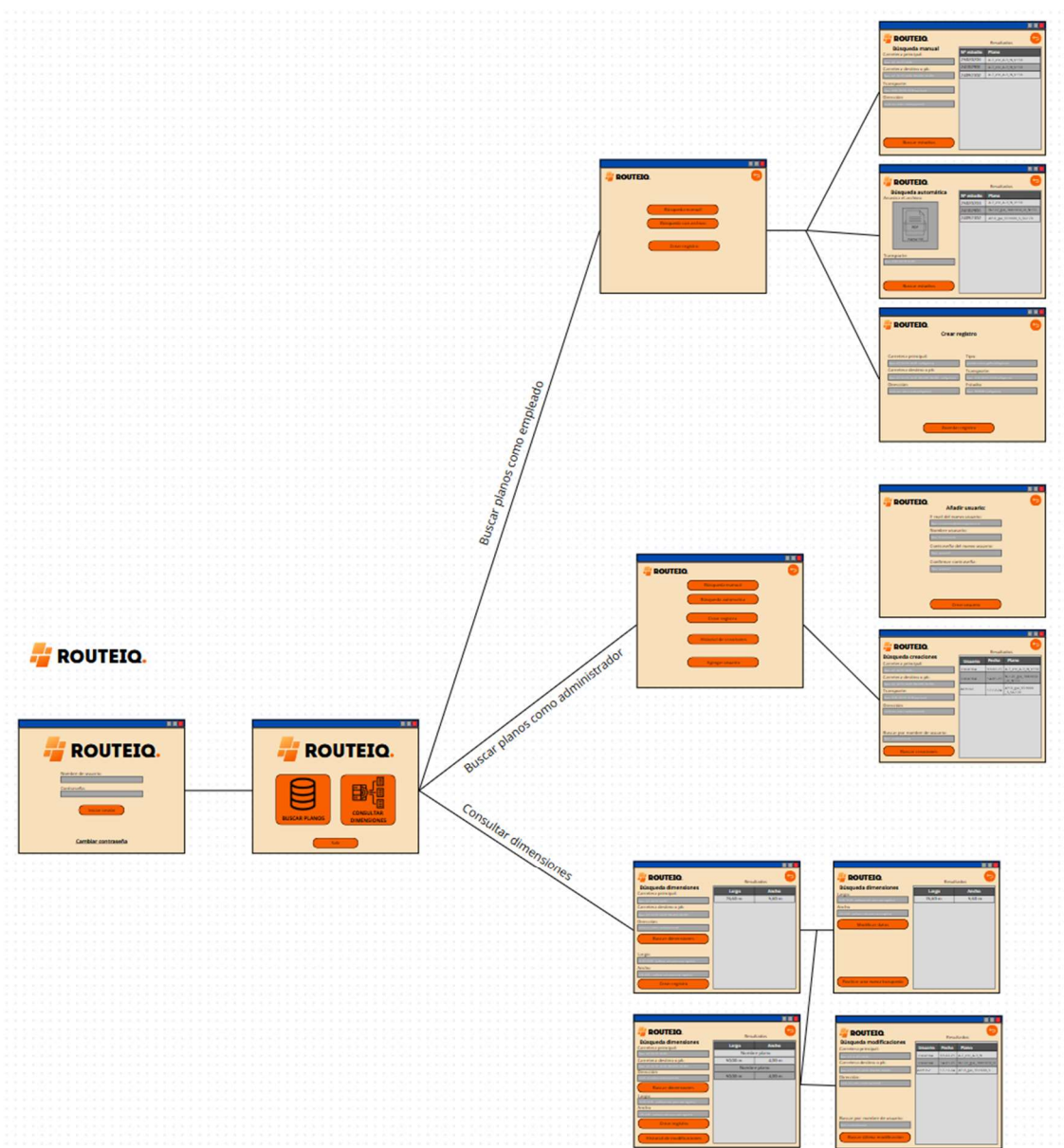
### Conexión cliente-servidor

Para la conexión cliente-servidor se desarrollará un programa en Java, el cual permitirá la conexión del cliente con el servidor en el que se aloja la base de datos mediante el uso de un JDBC.



## Diseño de la interfaz

Para el diseño de la interfaz se ha optado por un diseño simple, que no necesite de un aprendizaje constante.





## Codificación

### Creación de base de datos en AWS RDS

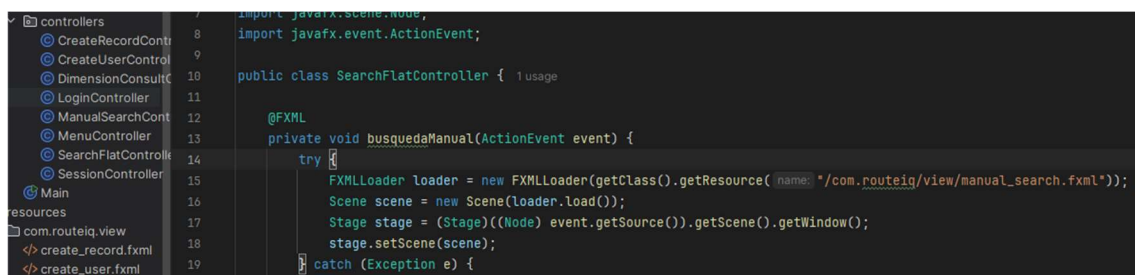
Para comenzar con la creación de la aplicación, lo primero es el desarrollo de la base de datos. Para este proceso, lo primero es crear una cuenta en AWS. Tras ello accedemos a las diferentes funciones del servicio y procedemos a crear nuestra base de datos en RDS. Una vez creada la base de datos cargamos desde CMD los scripts que podemos programar en un IDE como Visual Studio Code.

```
-- 5) Tabla de Enlace
CREATE TABLE Enlace (
  km DECIMAL(6,3),
  coordenadas VARCHAR(30),
  tipo ENUM('GLO', 'ENL', 'GAL', 'CUR', 'SAL') NOT NULL,
  -- GLO -> glorieta // ENL -> enlace // GAL -> galibo // SAL -> salida
  direccion ENUM('NORTE -> NORTE', 'NORTE -> ESTE', 'NORTE -> OESTE', 'NORTE -> SUR',
    'ESTE -> NORTE', 'ESTE -> ESTE', 'ESTE -> OESTE', 'ESTE -> SUR',
    'OESTE -> NORTE', 'OESTE -> ESTE', 'OESTE -> OESTE', 'OESTE -> SUR',
    'SUR -> NORTE', 'SUR -> ESTE', 'SUR -> OESTE', 'SUR -> SUR') NOT NULL,
  codCarreteraOrigen VARCHAR(8) NOT NULL,
  codCarreteraDestino VARCHAR(8),

```

### Creación de la app con JavaFX

Para desarrollar la aplicación de escritorio se hará uso de JavaFX y de SQL para mandar las diferentes ordenes a la base de datos almacenadas en los servicios de Amazon. Para esta parte se hará uso de inteligencia artificial para agilizar el trabajo de la creación de las interfaces, el trabajo de código que involucra la creación, modificación, lectura y borrado de datos se hará de forma manual en su completo procedimiento para evitar funciones no deseadas. El IDE para usar en este punto del desarrollo será IntelliJ IDEA.



```
import javafx.scene.Node;
import javafx.event.ActionEvent;

public class SearchFlatController {
    @FXML
    private void busquedaManual(ActionEvent event) {
        try {
            FXMLLoader loader = new FXMLLoader(getClass().getResource("/com.routeiq/view/manual_search.fxml"));
            Scene scene = new Scene(loader.load());
            Stage stage = (Stage)((Node) event.getSource()).getScene().getWindow();
            stage.setScene(scene);
        } catch (Exception e) {
            //
        }
    }
}
```



## Pruebas

Las pruebas se van realizando cada vez que un modulo del proyecto se finaliza, de tal forma que se verifica la correcta funcionalidad de este y que este esta disponible para implementar en la aplicación. Estas pruebas que se realizan son de caja negra, debido a la falta de tiempo.

Si el modulo supera las pruebas, este se implementa en la aplicación, la cual se despliega de forma inmediata para detectar posibles problemas existentes con otros módulos implementados con anterioridad.

## Desglose económico

Para el calculo del coste de la aplicación, se hará un recuento de las horas que se han dedicado a cada apartado necesario para conseguir una aplicación funcional. Estas horas se encuentran divididas en diferentes apartados, los cuales, cada uno, corresponde a un perfil diferente, por lo tanto, el precio por hora de cada apartado podría variar.

En el apartado de mantener comunicación con el cliente para aclarar los requisitos que debe cumplir la aplicación, encontramos que se han empleado 14 horas, las cuales corresponden a un perfil de analista. El salario por hora de un perfil como este es de 15€. (210 euros en total).

En el apartado de diseñar la base de datos y la interfaz, encontramos que se han empleado 31 horas, las cuales corresponden a un perfil de arquitecto de software y de diseñador UX. El salario por hora medio de estos dos perfiles como estos es de 30€. (930 euros en total).

En el apartado de investigar las diferentes tecnologías para el desarrollo del proyecto, encontramos que se han empleado 35 horas, las cuales corresponden a un perfil de arquitecto de software. El salario por hora medio de este perfil es de 50€. (1750 euros en total).

En el apartado codificación, encontramos que se han empleado 58 horas, las cuales corresponden a un perfil de desarrollador de software. El salario por hora medio de este perfil es de 13€. (754 euros en total).

En el apartado de prueba y depuración, encontramos que se han empleado 13 horas, las cuales corresponden a un perfil de desarrollador de software. El salario por hora medio de este perfil es de 13€. (169 euros en total).

En el apartado de documentación, encontramos que se han empleado 16 horas, las cuales corresponden a todos los perfiles que podemos encontrar a lo largo del desarrollo de la aplicación, ya que cada uno realiza su contribución en este apartado. El salario por hora medio de este apartado se ha estimado en 13€. (208 euros en total).

Finalmente, en el apartado de formación se han empleado 4 horas, las cuales serán encomendada a un perfil de consultor. El salario por hora medio de este apartado es de 30€. (120 euros en total).

Todos los salarios han sido obtenidos haciendo uso de la inteligencia artificial, haciendo que esta analizase las diferentes ofertas de trabajo que se pueden encontrar en los diferentes portales de empleo disponibles y que en función de los salarios ofrecidos sacara la media.

Estos cálculos hacen un coste la aplicación de 4141 euros (IVA no incluido).

Este precio incluye soporte correctivo durante los primeros 12 meses, ya que se entiende que habrá que ajustar puntos menores de la misma para mejorar la experiencia del cliente.

A partir del mes 13, el uso de la aplicación requiere una suscripción mensual de 100 €, que cubre acceso, soporte básico y mantenimiento evolutivo.

El coste de alojamiento en AWS no está incluido y será facturado según consumo real. En caso de impago, el acceso a la base de datos será suspendido hasta regularización del servicio con una recarga equivalente a dos meses de pago por la gestión de la reactivación.

## Roadmap

Este apartado presenta la hoja de ruta prevista para la evolución de la aplicación. Aunque la versión actual cumple con los requisitos funcionales definidos inicialmente, se contemplan futuras mejoras y funcionalidades adicionales que permitirán optimizar la experiencia del usuario, incrementar el rendimiento y facilitar el mantenimiento a largo plazo.

Las fases descritas a continuación no tienen fechas estrictamente cerradas, ya que su implementación dependerá de la demanda del cliente, los recursos disponibles y la prioridad de cada mejora. El objetivo es mantener un desarrollo sostenible, adaptado a las necesidades reales del proyecto.

- Establecer un sistema que compruebe si el correo introducido en el registro de usuarios existe y está disponible, para evitar errores o cuentas inválidas.
- Añadir la opción de recuperar la contraseña a través del correo electrónico, una vez implementada la verificación de este.
- Permitir al usuario mantener la sesión iniciada en el dispositivo por un tiempo determinado, evitando tener que iniciar sesión cada vez que abre la aplicación.
- Desarrollar una funcionalidad que permita analizar un PDF con un itinerario y detectar automáticamente los estudios que contienen planos relacionados.