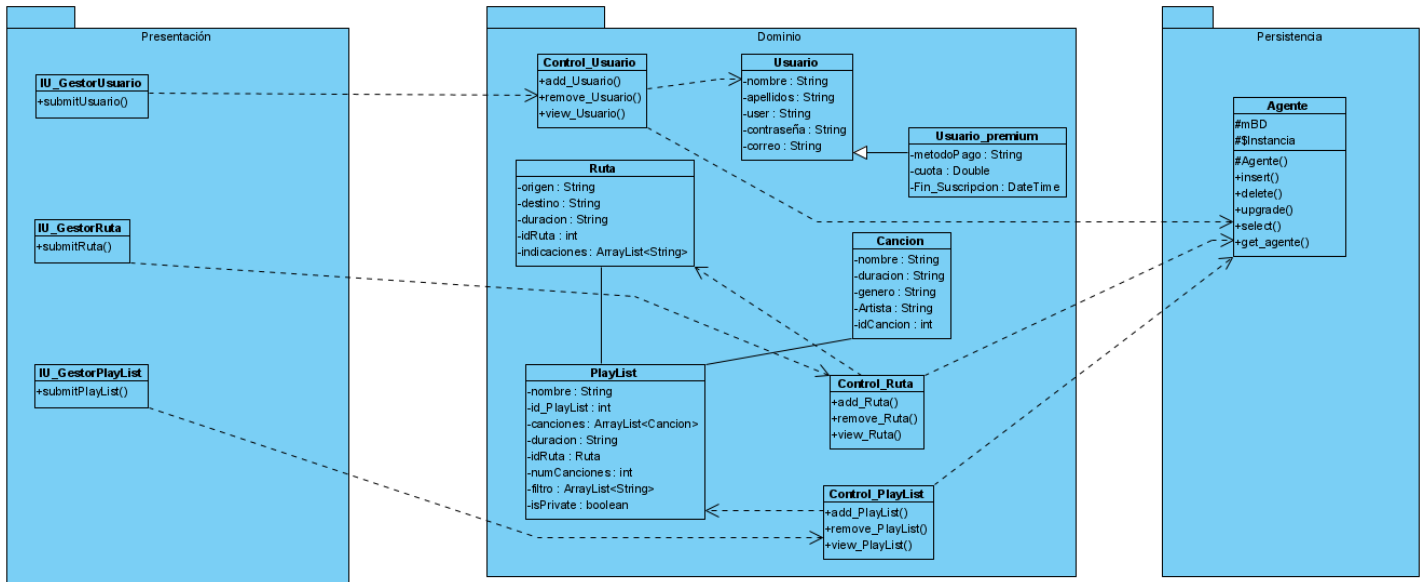


# SPRINT 2: MUMAPS

1. Arquitectura global detallada
2. Tecnologías a usar
3. Interfaces y estructuras de datos
4. Prototipos/Test tecnologías individuales
5. Mockup de GUI y user experience

## 1. Arquitectura global detallada



Como podéis observar en el diagrama de clases de nuestro proyecto podemos ver que hemos creado cuatro clases: Ruta, Usuario, Playlist y Canción, cada una con sus respectivos atributos y las clases de control que albergan los diferentes métodos de las clases mencionadas.

Las clases de control `Control_Ruta`, `Control_PlayList` y `Control_Usuario` están conectadas mediante una asociación de dependencia con un Agente que es nuestra base de datos, es decir que guardaremos los datos de los usuarios, las playlist y las rutas.

## 2. Tecnologías a usar.

- **Visual Paradigm:** Hemos usado esta herramienta para el diseño de la arquitectura global detallada mediante la creación de los diagramas de casos de uso y diagramas de clase.
- **GitHub:** Utilizamos esta herramienta para controlar el versionado de nuestro proyecto.
- **Visual Studio Code:** Hemos utilizado este compilador para poder programar todo lo relacionado con nuestro proyecto.
- **API seleccionado:** Para poder realizar el programa del GPS hemos tenido que utilizar el api público de HereWeGo.
- **Lenguaje de programación:** En nuestro caso en la parte de “backend” hemos decidido programar nuestro proyecto en Python. En la parte “frontend” vamos a utilizar WPF y C#.
- **API Spotify:** Para poder crear las playlist de canciones y compartirlas es necesario utilizar el api público que nos ofrece Spotify.
- **GoodNotes:** Hemos utilizado esta herramienta para el prototipo de la interfaz de nuestro proyecto.

### 3. Interfaces y estructuras de datos.

Como se ha comentado en la arquitectura global detallada (punto 2), haremos uso de programación orientada a objetos (POO). Respecto a las estructuras de datos, tendremos:

- **ArrayList para el guardado de canciones:** un objeto playlist está formada por un conjunto de objetos de canciones.
- **ArrayList de indicaciones:** la API REST de HereWeGo, nos devuelve un conjunto de información una vez se introduce un origen y destino. Esta información puede servir como indicaciones de viaje externas al sistema GPS, por lo que es conveniente guardarlas en una estructura de datos.

### 4. Prototipos/Test tecnologías individuales.

Tras una búsqueda exhaustiva observamos que, para obtener una ruta o trayecto de viaje, no podíamos introducir ni origen ni destino de forma textual en el GET pertinente. En este método, debíamos introducir coordenadas (latitud y longitud).

Sin embargo, la API REST nos ofrecía la posibilidad de utilizar un método que pudiéramos geolocalizar dicho punto. Por tanto, primero geolocalizamos el origen y destino para obtener sus coordenadas con este método y a continuación accedíamos al método de creación de rutas con los valores obtenidos.

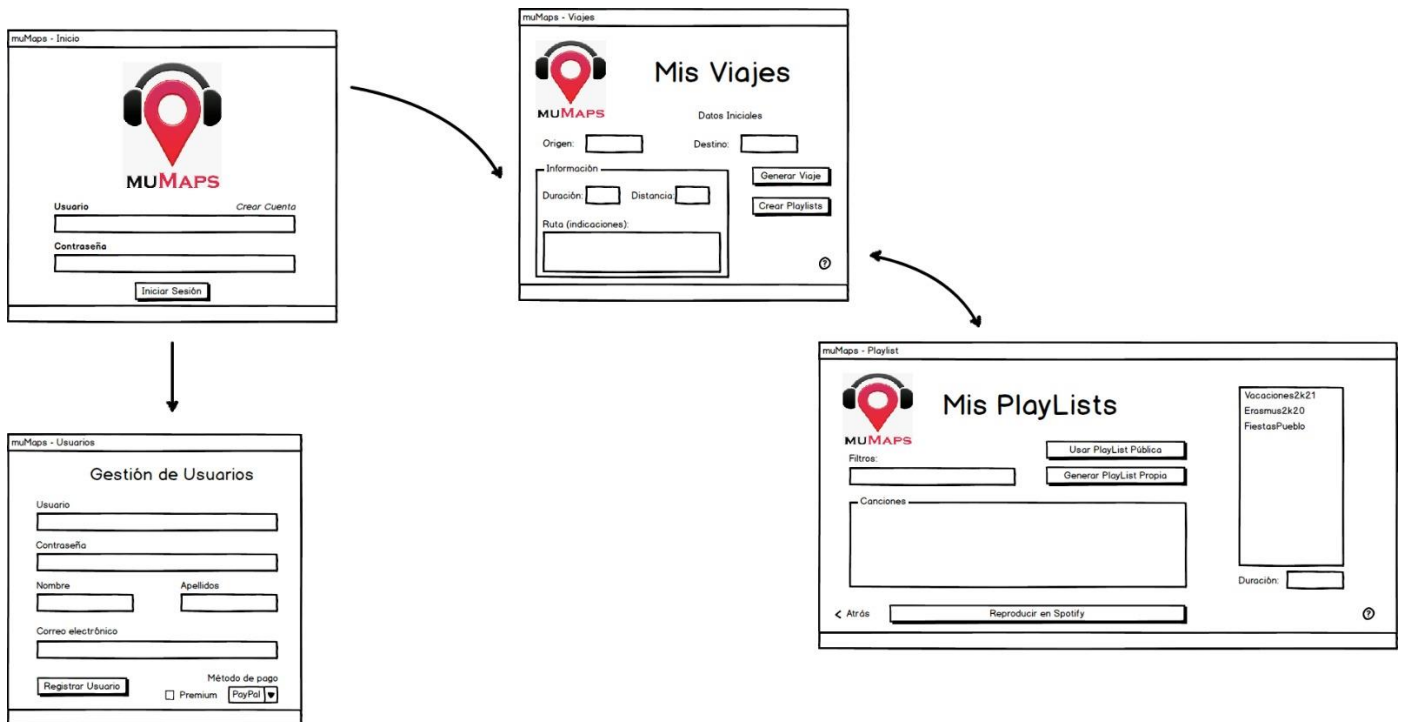
Entonces, test individuales que pueden realizarse sobre la API REST de HereWeGo y que consecuentemente, podremos testear y probar en nuestra aplicación son:

- Conectarse de forma correcta a 'Here WeGo'.
- Obtener coordenadas de un punto suministrado.
- Obtener trayecto de viaje a partir de un punto origen y destino.
- Recopilar información sobre el trayecto generado, como son las indicaciones de viaje, duración u otros.

Respecto a las pruebas que se realizarán sobre la parte de generación de playlists (parte de API de Spotify):

- Conectarse de forma correcta a 'Spotify'.
- Obtener canciones utilizando los filtros indicados.
- Obtener canciones similares a autores, álbumes u otros indicados.
- Recopilar información sobre las canciones introducidas en la playlist.
- Generar playlist con la duración indicada.
- Mostrar playlists ya existentes.

## 5. Mockup de GUI y user experience.



Aquí se pueden observar los diferentes prototipos de ventanas que tendrá nuestra aplicación.

La ventana principal para el login del usuario, la de registro, la ventana de mis viajes en la que podremos obtener información acerca de la ruta y por último la ventana de la playList.