

# MEMORIA DIAGRAMA DE CLASES

## Ruta:

La clase ruta se relacionarán con las clases temporizador y UsuarioRegistrado, pues un usuario podrá crear una única ruta a la vez, y a su vez, la ruta tendrá 3 temporizadores.

En esta clase tendremos dos atributos, tiempo y aviso. La primera, se utilizará para calcular el tiempo de cada temporizador, y la segunda será el aviso que enviará cada temporizador una vez finalizado su tiempo.

De igual forma, tendremos un conjunto de métodos, los cuales explicamos a continuación:

- obtenerTiempoAPIGoogle: en la cual el usuario introducirá un inicio y un destino, y la API de Google calculará el tiempo del recorrido automáticamente.
- obtenerTiempoManual: el usuario introducirá el tiempo deseado de forma manual.
- obtenerTiempo: método en el cual se elige entre introducir el tiempo automáticamente o de forma manual.
- setAviso: establecer el intervalo de tiempo que habrá entre avisos secundarios (segundo temporizador).
- getAviso: se muestra el aviso por pantalla de que el tiempo ha finalizado.

## UsuarioRegistrado:

La clase UsuarioRegistrado representa al usuario que hará uso de la aplicación. La clase tiene los siguientes atributos :

- Email : El correo con el que se registra el usuario.
- Teléfono : El teléfono del usuario, este teléfono podrá ser visible para el administrador, ya que es lo que usará para vetar o amonestar a los usuarios.
- Ruta : Todo usuario que utilice la aplicación deberá especificar una ruta.

El usuario podrá consultar y cambiar su número de teléfono mediante los métodos :

- getEmail() : Muestra el email del usuario
- setEmail() : Permite cambiar el email del usuario
- getTelefono() : Muestra el teléfono del usuario
- setTelefono() : Permite cambiar el teléfono del usuario

### **Administrador:**

La clase representa está relacionada con las clases sanción y UsuarioRegistrado y representa el rol de Administrador, cuya función es supervisar el correcto uso que realizan los usuarios de la aplicación.

Presenta los atributos mismos atributos que la clase usuario registrado previamente descritos.

Esta clase realiza dos funciones:

- amonestación, donde el administrador aplica una sanción a los usuarios que considere están realizando un uso inadecuado de la aplicación, de forma que se usa como una advertencia.
- vetar, donde el administrador dejará de permitir el uso de la aplicación a la cuenta asociada de manera inmediata.

### **Sanción:**

La clase sanción está relacionada con las clases UsuarioRegistrado y Administrador, en dicha clase se almacenan los datos del tipo de sanción que va a aplicar el usuario administrador siendo estas las mencionadas previamente (amonestación y vetar)

Presenta los atributos de fechaSancion, donde se almacena el día en que se aplica la sanción, tipoSancion, donde se muestra cuál de las dos sanciones se va a aplicar y telefonoUsuarioSancionado, dónde se asocia el tipo de sanción con el usuario al que se le va a aplicar.

Esta clase realizar las funciones de:

- amonestar, donde el sistema tramita y aplica la sanción al usuario previamente asignado.
- vetar, donde el sistema, tras tramitar la orden, denegara el acceso a la aplicación al usuario asignado para dicha orden.

## **Contactos:**

En esta clase se podrá configurar el número de teléfono del contacto al que se solicitará la ayuda. Se relaciona con la clase UsuarioRegistrado, para que el usuario pueda registrar uno o varios contactos.

Para ello tendrá un solo atributo, que será el número de teléfono introducido por el usuario.

Los métodos que maneja esta clase serán:

- getTelefono() : se mostrará el número de teléfono que se registra.
- setTelefono(telefono): el usuario añadirá el número de teléfono que podrá ser modificado posteriormente.

## **Notificación:**

Esta clase se relaciona con las clases temporizador y usuario registrado, pues el usuario podrá recibir 0 o más notificaciones y el temporizador podrá enviar 0 o más notificaciones.

Tiene un único método: mensaje. Este se encargará de enviar notificaciones al usuario informándole de las distintas funcionalidades.

## **Temporizador:**

La clase temporizador representa una cuenta atrás desde un número que le asignará el usuario, representado en la propiedad tiempo (en segundos) hasta 0.

La clase contiene los siguientes métodos :

- mandarNotificación() : Muestra un mensaje por pantalla que alerta al usuario.
- getTiempo() : Devuelve el tiempo en segundos
- setTiempo(tiempo : double) : Establece el tiempo a partir del cual comienza la cuenta atrás. Recibe como argumento el tiempo a establecer.