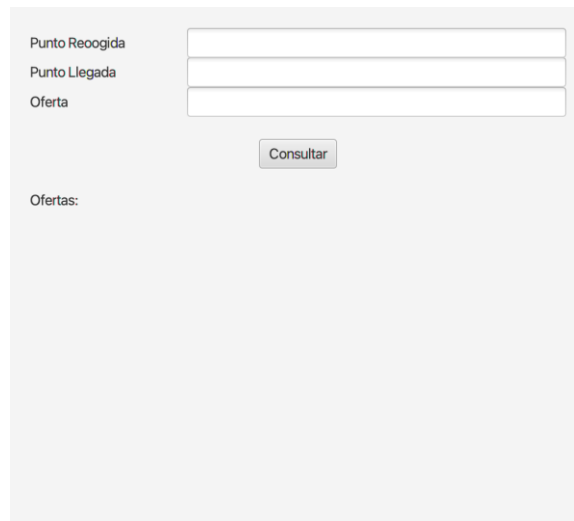


ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
PROGRAMACIÓN ORIENTADA A OBJETOS
SEGUNDA EVALUACIÓN
2021 - 1T

Desarrollo (60 puntos)

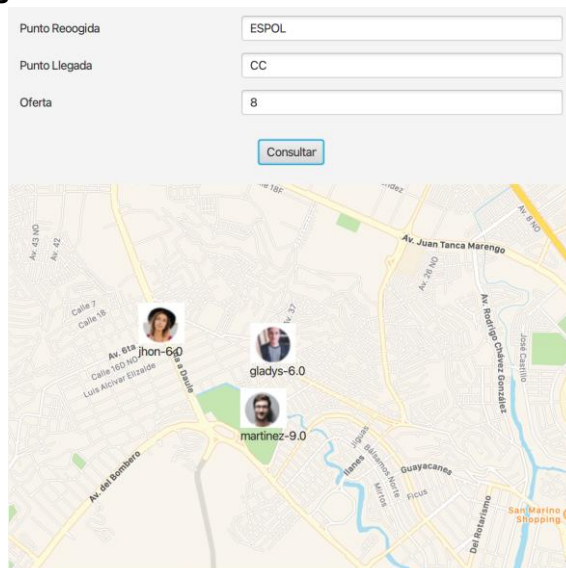
Desarrolle una aplicación que permita a los usuarios solicitar una carrera desde un **Punto de Recogida** hacia un **Punto de Llegada** y ofrecer un valor monetario (Oferta) por la carrera.



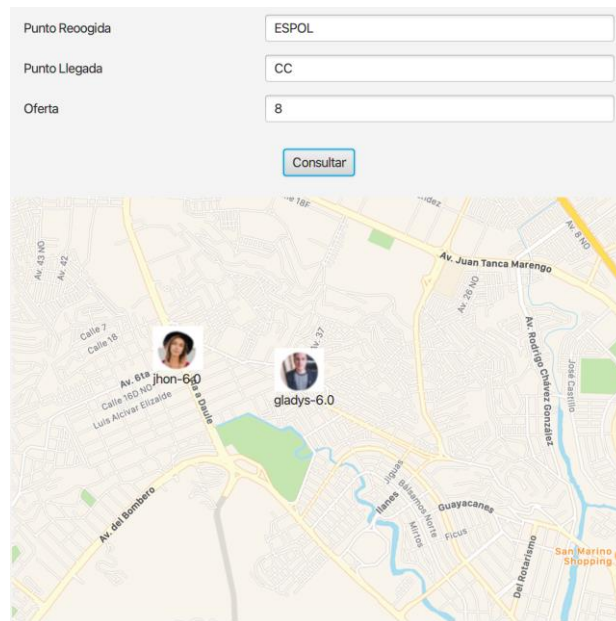
Formulario de solicitud de carrera:

- Punto Recogida:
- Punto Llegada:
- Oferta:
- Botón: Consultar
- Ofertas:

Cuando el usuario ha ingresado la información en el formulario y da click en el botón **Consultar**, La aplicación muestra los conductores más cercanos que están a una distancia de 9 km al **Punto de Recogida**.



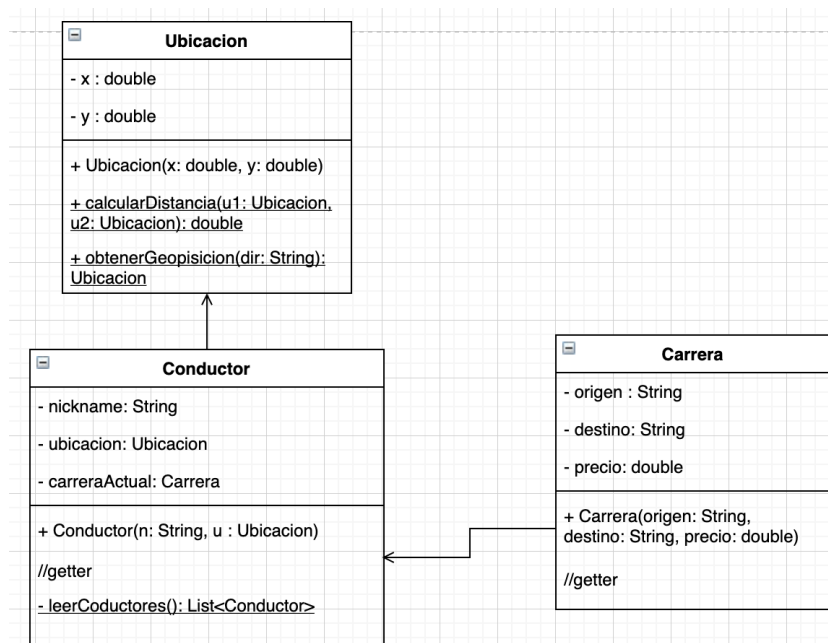
El panel que muestra los conductores **se actualiza cada 40 segundos** para mostrar la última información disponible de los conductores. Por ejemplo, luego de transcurrido los 40 segundos, el conductor Martinez ya no se muestra porque ya no se encuentra a una distancia de 9 kilómetros con respecto al **Punto de Recogida**.



En el panel no se muestra a martinez porque ya se movió y ya no está a 9 km

A usted se le da un proyecto base. Considere lo siguiente:

- UML del modelo



La clase **Ubicacion** representa una coordenada x,y en pixeles que se usará para ubicar los conductores en el mapa.

- El método estático **obtenerGeoposicion(dir: String)**, que recibe una dirección y retorna un objeto de tipo Ubicación correspondiente a la dirección ingresada. **<YA ESTA IMPLEMENTADO>**
- El método estático **calcularDistancia(u1: Ubicacion, u2: Ubicacion)**, retorna la distancia en kilómetros entre dos objetos de tipo ubicación. **<YA ESTA IMPLEMENTADO>**

- La información de los conductores se encuentra en el archivo conductores.txt que está en la carpeta datos en el directorio raíz del proyecto. Cada línea del archivo tiene el siguiente formato:

nickname, ubix, ubiy

Realice lo siguiente:

PARTE 1

- En la clase **Conductor**, implemente el método **leerConductores** que lee la información de los conductores desde el archivo conductores.txt y retorne una List<Conductor>.

Tenga en cuenta que debe crear un objeto de tipo Ubicación para las coordenadas “x” y “y” que tiene el conductor.

Considere lo siguiente:

- El método lanza una excepción de tipo IllegalArgumentException si una línea no está bien formada (es decir no tiene tres elementos)
- El método lanza excepción de tipo IOException

PARTE 2

- a. En la clase PrimaryController implemente un manejador para la acción del botón “Consultar” para que cuando se dé click suceda lo siguiente:
 - i. Se cargan en el panel **panelMapa**, los conductores con su foto, nombre y distancia a la que se encuentra dentro de un rango de 9 kilómetros. El archivo de la foto del conductor corresponde al **nickname** más la extensión **png**.

Nota: Recuerde que tiene el método **obtenerGeoposicion** y el método **calcularDistancia** que le servirá para obtener el valor en kilómetros.

- ii. En caso de presentarse un error al momento de leer el archivo de conductores muestre un mensaje de error al usuario con lo sucedido.
- iii. **Cada 40 segundos** el programa debe leer nuevamente el archivo de los conductores y reflejar ese cambio en el **panelMapa**.
- iv. Al dar clic a un conductor, este se selecciona para hacer la carrera, se debe crear un objeto de tipo **Carrera** con la información de la carrera y se fija este objeto en la variable carreraActual del conductor seleccionado. Luego, se presenta un mensaje que se ha realizado el proceso exitosamente (Alert). Finalmente, se debe detener la actualización del mapa que ocurría cada 40 segundos, se debe quitar los conductores que están en el mapa y se debe limpiar las cajas de texto.