

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

SECCIÓN: -

CATEDRÁTICO: -

TUTOR ACADÉMICO: -



# ***UDRIVE***



## **PRACTICA 2**

DESARROLLO DE APLICACIÓN DESTINADA A VIAJES

## PRACTICA 2

### Objetivos

#### Generales

- Familiarizar al estudiante con el lenguaje de programación Java.
- Que el estudiante aplique los conocimientos adquiridos en el curso de Introducción a la Programación y computación 1.
- Elaborar la lógica para presentar una solución a la propuesta planteada.

#### Específicos

- Utilizar el lenguaje de programación Java como herramienta de desarrollo de software.
- Aplicación de conceptos de programación orientada a objetos.
- Construcción de aplicaciones con interfaz gráfica.
- Implementación de sentencias de control, ciclos, vectores y librerías de interfaz gráfica.
- Aplicación de conceptos de programación para crear herramientas administrativas.
- Implementación de soluciones con el uso de hilos.
- Crear respaldos de información y restablecerla con uso de serialización.

# Descripción General

### Definición:

La práctica consiste en el desarrollo de una aplicación de escritorio para un servicio de gestión de viajes en carro, que permita ingresar su ubicación actual y llegar a un destino específico en tiempo real donde se visualice el estado de su viaje desde su punto de partida hasta su punto de destino.

### Consideraciones:

- La información registrada en la aplicación debe permanecer, aunque ésta sea cerrada
- Se tendrá un máximo de 3 vehículos para viajes.
- Todos los vehículos podrán iniciar su trayectoria de entrega a la vez o de manera separada.
- Deberá existir un administrador que pueda gestionar la creación de las rutas disponibles para los viajes.
- El usuario podrá verificar el historial de los viajes completados.

## PRACTICA 2

### Aplicación

Se debe contar con las siguientes funcionalidades:

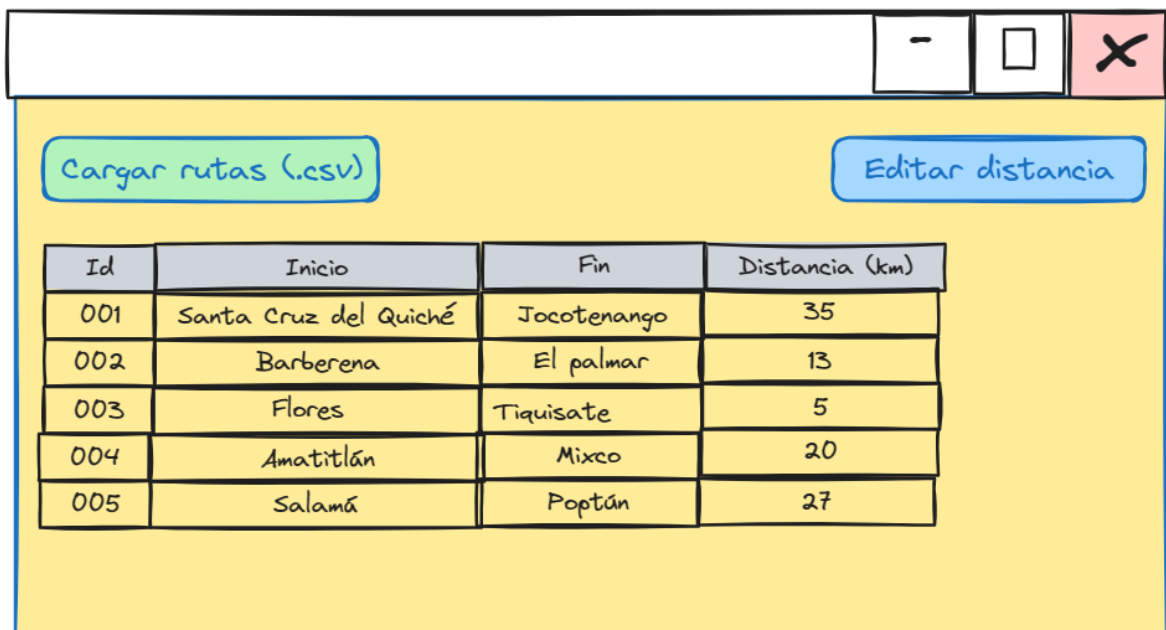
- **Menú para navegar entre pestañas**

Queda a criterio del estudiante como navegar entre las distintas pantallas que tendrá la aplicación.

- **Carga de rutas**

Acá se podrá registrar los viajes que la plataforma ofrece, estos se cargarán por medio de un archivo CSV, este csv se deberá buscar con JFileChooser, el csv tendrá la siguiente estructura y posteriormente se mostrarán los datos en una tabla como se muestra a continuación:

*Estructura csv: Inicio,Fin,Distancia*



Id	Inicio	Fin	Distancia (km)
001	Santa Cruz del Quiché	Jocotenango	35
002	Barberena	El palmar	13
003	Flores	Tiquisate	5
004	Amatitlán	Mixco	20
005	Salamá	Poptún	27

#### Ejemplo archivo entrada

```
inicio,fin,distancia
Santa Cruz del Quiche,Jocotenango,35
Barberena,El palmar,13
Flores,Tiquisate,5
Amatitlán,Mixco,20
Salamá,Poptún,27
```

## PRACTICA 2

- **Editar distancia**

Esta opción permitirá editar la distancia de un registro en específico, la ventana de edición debe tener lo siguiente:



Editar distancia ruta

Ingrese el Id de la ruta:

1

Ingrese la nueva distancia:

40

Cancelar Aceptar

Id	Inicio	Fin	Distancia (km)
001	Santa Cruz del Quiché	Jocotenango	<u>40</u>
002	Barberena	El palmar	13

- **Generar viaje**

Esta pantalla permitirá generar un viaje, seleccionando de primero un punto inicial, luego un punto final y por último un tipo de transporte. **Tomar en cuenta lo siguiente:**

- Los valores solamente pueden ser seleccionados por comboBox, no se permite ingresar texto.
- Se debe validar si los 3 pilotos que se tienen en el sistema ya tienen un viaje asignado, en caso alguno esté disponible este atenderá el viaje que se genere, de lo contrario el sistema deberá deshabilitar el botón de generar un viaje y se deberá mostrar un label con texto de color rojo indicando que no hay pilotos disponibles por el momento, este label queda a discreción del estudiante dónde colocarlo. Si hay pilotos disponibles este label no debe mostrarse en pantalla.

## PRACTICA 2

- Los valores que se muestren en los comboBox de **Punto Inicial** y **Punto final** deben ser solamente los que han sido cargados al sistema por medio del archivo csv.
- Tomar en cuenta que un punto final, puede ser un punto inicial y uno inicial puede ser uno final, ya que la distancia será la misma. Por lo tanto en los comboBox de **Punto Inicial** y **Punto final** sus opciones pueden ser tanto puntos iniciales como finales.
- El sistema contará con un total de nueve transportes los cuales se listan a continuación:
  - Motocicleta 1
  - Motocicleta 2
  - Motocicleta 3
  - Vehículo estándar 1
  - Vehículo estándar 2
  - Vehículo estándar 3
  - Vehículo premium 1
  - Vehículo premium 2
  - Vehículo premium 3

Generar viaje

Seleccionar punto inicial: Poptún ▼

Seleccionar punto final: Salamá ▼

Seleccionar tipo de transporte: Vehículo premium 2 ▼ \*

Generar viaje

## PRACTICA 2

### Inicio de viajes

Se verán todos los viajes que actualmente están preparados para iniciarse. Se podrán iniciar los viajes individualmente, o bien, en conjunto. El usuario al llegar a su punto de destino puede decidir en retornar al punto de donde partió, para realizar esto, el usuario le indica a la plataforma que quiere regresar, por lo cual, el administrador de la plataforma inicia manualmente el viaje de retorno.

El recorrido será relativo a la distancia, se tendrá un mismo punto de destino, pero el tiempo para completar el recorrido será proporcional a la distancia. Todos los vehículos tienen la misma velocidad. Si el vehículo está haciendo un viaje de regreso, se debe mostrar visualmente.



Los vehículos deben mostrar en todo momento la cantidad de combustible con la que cuentan actualmente, en dado caso el vehículo se llegara a quedar sin este, se deberá desplegar un botón que permita recargar al vehículo de combustible hasta llenarse.

La distribución de gasto y capacidad del tanque de combustible es la siguiente:

Tipo de transporte	Gasto de combustible (gal/km)	Capacidad del tanque (gal)
Motocicleta	0.1	6
Vehículo Estándar	0.3	10
Vehículo Premium	0.45	12

## PRACTICA 2

The diagram shows a road network with three vehicles and their routes:

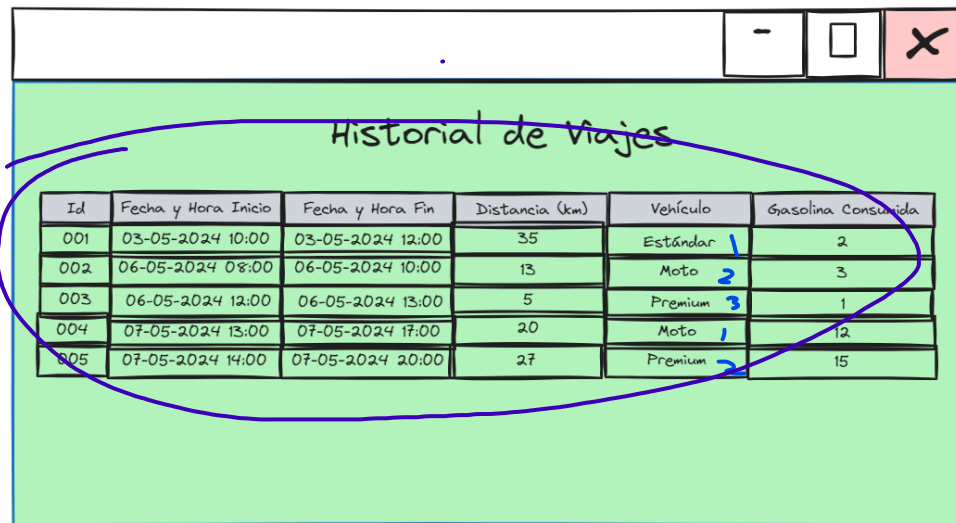
- Motocicleta 1:**
  - Distance: 35km
  - Destination: Jocotenango
  - Route: Starts at 'Inicio Santa Cruz del Quiché', travels right, then turns left and continues right.
  - Buttons: 'Iniciar' (crossed out), 'Recargar' (crossed out).
- Vehículo Estándar 1:**
  - Distance: 13km
  - Destination: El palmar
  - Route: Starts at 'Inicio Barberena', travels right.
  - Buttons: 'Iniciar', 'Recargar'.
- Vehículo Premium 1:**
  - Distance: 20km
  - Destination: Mixco
  - Route: Starts at 'Inicio Amatitlán', travels right.
  - Buttons: 'Iniciar', 'Recargar'.

At the top right, there are three buttons: a minus sign (-), a square, and a crossed-out X.



## PRACTICA 2

### Historial de pedidos



Id	Fecha y Hora Inicio	Fecha y Hora Fin	Distancia (km)	Vehículo	Gasolina Consumida
001	03-05-2024 10:00	03-05-2024 12:00	35	Estándar 1	2
002	06-05-2024 08:00	06-05-2024 10:00	13	Moto 2	3
003	06-05-2024 12:00	06-05-2024 13:00	5	Premium 3	1
004	07-05-2024 13:00	07-05-2024 17:00	20	Moto 1	12
005	07-05-2024 14:00	07-05-2024 20:00	27	Premium 2	15

Se podrá ver el historial de los viajes que se han realizado, se detallará el ID de la ruta, fecha y hora tanto de inicio como de fin, distancia recorrida, el vehículo utilizado, la distancia de trayectoria y el total de gasolina consumida por el vehículo durante el viaje.

### Serialización

Los registros del historial de viajes y el estado de los viajes debe persistir, aunque la aplicación sea cerrada, por lo que se solicita utilizar serialización para conservar los datos de ejecución, y que al iniciar de nuevo la aplicación los datos sigan en el mismo estado de la ejecución anterior. Se verificará que los archivos en los que se guarde esta información sean archivos binarios, **no se permite almacenar estos datos en formato csv, json o cualquier otro tipo.** Deben guardarse los objetos serializados en archivos binarios.

## **PRACTICA 2**

### **Requerimientos**

#### **Documentación:**

- Manual técnico en formato PDF
  - Descripción general de la solución
  - Diagrama de clases
  - Diagramas de flujo sobre los métodos principales
- Manual de usuario en formato PDF
  - Descripción general del programa
  - Requerimientos mínimos del sistema
  - Descripción de los pasos para el uso de las opciones más importantes

#### **Restricciones:**

- La aplicación debe ser desarrollada en el lenguaje de programación Java.
- No se permite utilizar código copiado o bajado de Internet.
- El IDE a utilizar queda a discreción del estudiante (se recomienda el uso de NetBeans)
- Copias obtendrán una nota de 0 y reportará a la Escuela de Ciencias y Sistemas.

## PRACTICA 2

### HABILIDADES A EVALUAR

- Habilidad para analizar y sintetizar información
- Habilidad para resolver problemas.
- Interacción con objetos en interfaz gráfica.
- Manejo de propiedades en componentes gráficos.
- Uso de hilos.
- Uso de serialización para la persistencia de la información.

### Entrega

- FECHA DE ENTREGA: 21/03/2024 antes de las 23:59 (No se aceptarán entregas a partir de esa hora) **NO HABRA PRORROGA.**
- Adjuntar todos los entregables solicitados dentro de un repositorio en GitHub con el nombre de IPC1\_Practica2\_carnet, por ejemplo:  
  
IPC1\_Practica2\_202300000.
- Subir link del repositorio en la tarea asignada en UEDi.
- Se deberá generar el ejecutable de su aplicación (.jar), se calificará la práctica únicamente con este archivo compilado.
- Agregar de colaborador al repositorio al auxiliar correspondiente a su sección:
  - Sección A: rodrialeipc
  - Sección B: RMorales202010033
  - Sección C: DavidMaldo02
  - Sección D: Ennox2
  - Sección E: rodriporon
  - Sección F: Ayeser-Cristian
  - Sección G: fedestm