

🌟 Descripción del Contexto

Uber es una plataforma de transporte que conecta conductores con pasajeros para ofrecer servicios de movilidad 🚗. Los conductores reciben pagos por cada viaje realizado, dependiendo de la distancia, tiempo y tarifa base establecida por la plataforma 💰.

En este contexto, los conductores necesitan llevar un registro detallado de sus viajes ✉ para analizar su desempeño, calcular ingresos y optimizar sus horarios de trabajo ⌚. Actualmente, Uber proporciona información en su aplicación, pero no permite personalizar el almacenamiento ni realizar cálculos según las necesidades del conductor 🔍.

La falta de un sistema propio de gestión de viajes puede dificultar la toma de decisiones 😞, ya que el conductor no tiene acceso inmediato a métricas clave como el promedio de ingresos por hora o la distribución geográfica de los viajes 🌐.

📈 Definición de Necesidad

💡 a. Explicación del problema o necesidad

Como conductor de Uber, es fundamental llevar un registro detallado de los viajes realizados para evaluar el rendimiento y optimizar los horarios de trabajo 🕒. Sin embargo, actualmente no se cuenta con una herramienta personalizada que permite almacenar, consultar y analizar esta información de manera eficiente 📝.






⚠ b. Consecuencias de la falta de una solución

- ❌ Dificultad para calcular el promedio de ingresos por hora de trabajo.
 - ❌ Falta de un registro histórico de viajes para análisis posterior.
 - ❌ Imposibilidad de evaluar cuáles son los horarios y localidades más rentables.
 - ❌ Dependencia de los reportes de Uber, que pueden ser limitados o no mostrar la información de manera personalizada.
-

📚 Justificación de la Solución




💻 a. Beneficios concretos de la implementación

La creación de una aplicación en Java para registrar y consultar los viajes permitirá al conductor:

-  Tener un control detallado de sus ingresos diarios.
-  Calcular automáticamente el promedio de ingresos por hora de trabajo.
-  Identificar patrones en los viajes (horarios y zonas más rentables).
-  Facilitar la toma de decisiones sobre cuándo y dónde trabajar para maximizar las ganancias.
-  Tener un respaldo de los datos sin depender de la aplicación de Uber.



b. Cuantificación de los beneficios

Si el conductor puede identificar las horas y zonas más rentables, podría optimizar su tiempo de trabajo y aumentar sus ingresos . Por ejemplo:

- Si actualmente trabaja 10 horas y gana en promedio \$18k por hora, pero identifica que en ciertos horarios puede ganar \$25k por hora, podría incrementar sus ingresos diarios en un 39% aproximadamente .
- Al reducir el tiempo en zonas poco rentables, podría hacer más viajes en menos tiempo, aumentando su eficiencia .
- Contar con un registro claro también facilita el control de ingresos .





Propuesta de Solución

a. Desarrollo de una aplicación en Java

Para abordar la necesidad identificada, se propone el desarrollo de una aplicación en **Java** que permita a los conductores de Uber registrar y consultar información sobre sus viajes . Esta aplicación brindará herramientas para gestionar datos de manera eficiente y visualizar métricas clave en tiempo real .

b. Funcionamiento de la aplicación

La aplicación permitirá a los conductores:

1. **Registrar manualmente cada viaje** ingresando localidad, valor, hora y duración.
2. **Almacenar la información dentro de la consola** para futuras consultas .
3. **Calcular métricas clave automáticamente**, como:
 - Número total de viajes en un día .
 - Ingreso total diario .
 - Promedio de ingresos por hora trabajada .
4. **visualizar datos dentro de la consola** para análisis.
5. **Interfaz simple y amigable**, con menús claros y opciones intuitivas.

c. Funcionalidades principales

- **Ingreso de datos de viajes:** Permite registrar localidad, hora, valor del viaje, etc.
- **Cálculo automático de ingresos:** Calcula totales diarios y promedio por hora.

- **Historial de viajes dentro de la consola:** Guarda y permite consultar viajes previos.
 - **Interfaz en consola:** Fácil de usar y sin necesidad de conexión a internet.
-

Modelado de Clases

Módulo de Clases

La aplicación estará organizada en diferentes clases, cada una con una responsabilidad específica:






1. **Viaje** : Representa un viaje individual realizado por el conductor.
 2. **Conductor** : Representa al usuario de la aplicación.
 3. **RegistroViajes** : Gestiona la lista de viajes registrados.
 4. **CalculadoraIngresos** : Realiza cálculos como ingresos totales y promedio por hora.
 5. **Aplicacion** : Controla la interacción del usuario con la aplicación.
-

Diagrama de Clases UML - Aplicación de Gestión de Viajes Uber

Conductor

- **Atributos:** nombre (String), registroViajes (RegistroViajes)
- **Métodos:**
 - registrarViaje(Viaje viaje)
 - consultarHistorial()

tiene un

RegistroViajes

- **Atributos:** viajes (ArrayList<Viaje>)
- **Métodos:**
 - agregarViaje(Viaje viaje)
 - obtenerTodosLosViajes()

contiene una lista de

Viaje

- **Atributos:** localidad (String), valor (double), hora (String), duracion (double)
- **Métodos:**

- getResumen()

📌 es utilizado por

📌 **CalculadoraIngresos** 💰

- **Atributos:** registroViajes (RegistroViajes)
- **Métodos:**
 - calcularTotalDiario()
 - calcularPromedioPorHora()

📌 es gestionado por

📌 **Aplicacion** 📱

- **Métodos:**
 - mostrarMenu()
 - registrarNuevoViaje()
 - consultarIngresos()