

Sistema de Gestión para una Plataforma de Alquiler de Autos



Objetivos

Objetivo General

El trabajo práctico tiene por objetivo garantizar que cada alumno/a, a través de la investigación y experimentación, transite por el proceso de diseñar e implementar un sistema complejo.

Objetivos particulares

Principal

Que los alumnos/as adquieran, mediante el desarrollo del trabajo práctico, una mejor comprensión de los temas que se analizan en la materia desde una perspectiva teórica.

Trabajo en equipo

Que los alumnos/as comprendan la importancia del trabajo en equipo para llevar adelante un proyecto. Los alumnos/as deberán conformar equipos de 5 integrantes.

Técnicas y herramientas

Que los alumnos/as comiencen a utilizar, o incrementen sus conocimientos sobre, técnicas y herramientas de desarrollo de software actualmente utilizadas en la industria.

Forma de entrega y evaluación

El trabajo práctico debe estar versionado en un repositorio de código (github).

El trabajo práctico será evaluado tanto en forma grupal como en forma individual.

Evaluación grupal

Cada equipo debe realizar una presentación grupal del trabajo, en el cuál se evaluará el código y documentación de la solución.

Evaluación individual

Para la evaluación individual, serán tenidos en cuenta los aportes de cada integrante del equipo dentro del repositorio. En caso de que alguno de los integrantes del grupo no demuestre haberse comprometido con el trabajo grupal, dicho alumno/a deberá implementar funcionalidades extras al trabajo y luego se realizará un coloquio en el que deberá defender la solución propuesta.

Introducción

"DriveHub", la nueva plataforma de alquiler de autos, necesita un sistema para gestionar su flota de vehículos y las reservas de sus clientes. El sistema debe ser capaz de manejar diferentes tipos de vehículos y calcular los costos de alquiler de manera flexible.

Etapa 1: Diseño e Implementación del Sistema de Alquiler Básico

Requerimientos Funcionales

El sistema debe ser capaz de:

- **Gestionar Vehículos:** Existen tres categorías principales: **Compacto, Sedán y SUV**. Cada vehículo tiene un número de matrícula, un estado (Disponible, En Alquiler, En Mantenimiento, Necesita Limpieza) y una lógica de tarifa específica.
- **Gestión de Clientes y Reservas:** Un cliente puede crear una reserva para un vehículo específico, indicando la fecha de inicio y fin. El sistema debe validar la disponibilidad del vehículo para las fechas solicitadas.
- **Cálculo de Tarifas:**
 - **Compacto:** Tarifa base de **\$30** por día. Aplica un cargo de **\$0.15** por cada kilómetro recorrido si se superan los 100 km por día de alquiler.
 - **Sedán:** Tarifa base de **\$50** por día. Aplica un cargo de **\$0.20** por cada kilómetro recorrido, sin límite diario.
 - **SUV:** Tarifa base de **\$80** por día. Aplica un cargo fijo adicional de **\$15** por día por concepto de seguro y un cargo de **\$0.25** por cada kilómetro recorrido si se superan los 500km en total durante el período de alquiler.
- **Gestión de Kilometraje:** cada vez que un cliente entrega el vehículo, se debe tomar nota del kilometraje de manera tal que se pueda calcular los cargos adicionales correspondientes.
- **Mantenimiento de Vehículos:** El sistema debe poder registrar el costo y la fecha de los mantenimientos de cada vehículo.

¿Qué se pide?

1. Diagrama de clase de la solución propuesta.
2. Diagramas de secuencia de 2 o más requerimientos a definir por el equipo.
3. Desarrollar una aplicación que permita emular la situación planteada. Proveer el código y las pruebas unitarias para verificar:
 1. Creación de distintos tipos de vehículos

2. Lógica de cálculo de tarifas y recargos en distintos escenarios.
3. Costo total de una reserva. Plantear distintos escenarios.

Requerimientos adicionales:

- Desarrollar las pruebas unitarias para cada método realizado.
 - El porcentaje de código cubierto por pruebas unitarias debe superar el 80%.
 - Agregar archivo README.md al proyecto describiendo el proyecto, agregar instrucciones para su compilación y funcionamiento.
 - Generar documentación de código utilizando JSDoc.
-



continuará... 😊