

# Ingeniería de Software

## Trabajo: Desarrollo iterativo incremental con TDD

**Descripción del trabajo de evaluación: 8 puntos**

**Fecha de entrega final: 11/09/2025**

Se compone de dos partes:

1.- Identificar una lista de funcionalidades en las que podemos dividir el desarrollo de la aplicación descrita abajo en anexos. Se debe tomar en cuenta los siguientes aspectos:

- Queremos identificar funcionalidades(o rebanadas) que sean demostrables y de valor/utilidad para el usuario.
- Ninguna funcionalidad es solo un mockup de interfaz de usuario que no hace nada.
- Considera que las rebanadas o funcionalidades deberían ser pequeñas para poder desarrollar la aplicación de forma incremental.

2.- Para esa lista de funcionalidades del punto anterior, desarrollar el producto de software **de forma iterativa/incremental aplicando TDD**. Y siguiendo estos lineamientos:

- Hacer un commit y push por cada ciclo de TDD con pruebas en verde y/o funcionalidad implementada.
- Los mensajes de los commits que sean descriptivos de la funcionalidad que se implementó en ese commit (siguiendo el [estandar que definimos](#) feat, refactor, fix, etc)

**Entregables del trabajo:**

- Lista de funcionalidades para desarrollo incremental de la aplicación. Siguiendo los lineamientos descritos arriba.
- Link a la app deployada en Netlify
- Link a las gráficas del proyecto en TDDLab
- Repositorio github. La implementación o programación **siguiendo TDD (Test Driven Development)**
- Por lo menos **un commit por cada funcionalidad implementada y por cada ciclo de TDD con pruebas en verde**. Tambien es sugerido hacer push de forma regular, no solo al final.
  - Los mensajes de los commits deben ser descriptivos de la funcionalidad o modificación añadida
- (opcional) Video 3 minutos que haga una demo del programa en funcionamiento. No se necesita mostrar el código solo el programa funcionando.

**Criterios de evaluación:**

- La correcta aplicación de una perspectiva iterativa/incremental para gestionar el desarrollo de un producto de software. Entregar las funcionalidades como una lista en documento .pdf
- Correcta aplicación de TDD (Test Driven Development)
- Producto funcional resultante que incluye las funcionalidades solicitadas en el presente documento.
- Haber realizado el trabajo de forma incremental, con el debido tiempo, no es lo mismo que los commits estén todos la última noche, a que se vea un progreso de commits con debido tiempo.

## ANEXO:

### Aplicación a desarrollar:

## Cálculo de tarifas de estacionamiento

Una empresa de estacionamientos necesita un sistema que permita calcular cuánto debe pagar un cliente por su estadía.

El sistema debe considerar las siguientes **políticas comerciales**:

- **Tarifa base por hora:** Bs 10.00
- **Tarifa nocturna (22:00–06:00):** Bs 6.00 por hora o fracción
- **Tope máximo por día calendario:** Bs 50.00
- **Penalidad por pérdida de ticket:** Bs 80.00 (reemplaza al cálculo normal)
- El cobro es por **hora o fracción**, redondeando siempre hacia arriba.
- El monto final se presenta con **redondeo monetario a dos decimales**.
- El sistema debe **validar** que la hora de salida no sea anterior a la de entrada.

## Requisitos adicionales

La empresa desea que el sistema tenga una **interfaz de usuario simple** donde el usuario pueda:

1. Registrar la **hora de ingreso** y la **hora de salida** del vehículo.
2. Indicar si el ticket fue **perdido**.
3. Consultar el **monto total a pagar**.
4. Ver un **desglose por día** cuando la estadía cruza varios días (por ejemplo, cuánto se cobró en cada día antes y después de aplicar el tope).

La interfaz de usuario debe ser simple y es importante que permita ingresar los datos y mostrar claramente el resultado.