

Tienda Sol

Plataforma de Comercio Electrónico

Trabajo Práctico Integrador
Desarrollo de Software
2C 2025 - Grupo 6

Alex Fiorenza
Ian Gabriel Sanna
Facundo Tomasetti
Ignacio Alejo Scarfo
Ignacio Castro Planas



Tienda Sol
ONLINE STORES

Entrega 1: Implementación del Modelo de Objetos + Configuración del Proyecto

Fecha: 11/09/2025

Entregables

1. **Implementación** del Diagrama de Clases provisto, completando cada uno de los métodos mencionados y agregando aquellos que el equipo considere necesarios.
2. **Implementación** del endpoint “Health Check”.
3. **Explicación** del Git flow definido para utilizar a lo largo del proyecto.

Tecnologías a utilizar

- Backend
 - Lenguaje de programación base: [JavaScript](#)
 - Entorno de ejecución: [Node.js](#)
 - Framework: [Express](#)
- Git como Sistema de Versionado de Código, con repositorios en Github.

Respecto a las notificaciones:

- Cada vez que se realice un pedido, es necesario enviarle una notificación al Vendedor, donde se le indique quién realizó el pedido, qué productos incluye, el total del mismo y dirección de entrega.
- Cada vez que un vendedor marque un pedido como enviado, es necesario notificar al comprador.
- Si un comprador decide cancelar un pedido, es necesario notificar al vendedor.

Justificaciones iniciales

Repositorio de implementación del TP:

<https://github.com/ddso-utn/tpa-2025-2c-grupo-6>

Se trata de un monorepo, que integra una aplicación frontend con Next.js y un backend con Express. La estructura monorepo facilita el desarrollo y la gestión de ambos proyectos en un único entorno.

Se tomó esta decisión, en primer lugar ya que Create React App está deprecado.

<https://react.dev/blog/2025/02/14/sunsetting-create-react-app>

GitFlow definido

La existencia de sólo dos ramas en GitHub, main y development, se justifica porque permite un flujo de trabajo claro y ordenado: la rama main se reserva exclusivamente para el código listo para entregar, mientras que develop concentra las tareas en curso, integrando nuevas funcionalidades y correcciones antes de ser validadas y fusionadas en main. Esta estrategia minimiza riesgos de romper la versión entregable, facilita el control de cambios y simplifica la gestión del repositorio, evitando la complejidad innecesaria que puede traer manejar múltiples ramas adicionales.

<https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>

