

	<p align="center">Programación Visual Analista Programador Universitario Facultad de Ingeniería</p>	<p align="center">Trabajo Práctico Integrador</p>
---	--	---

Trabajo Práctico integrador grupal de presentación obligatoria.

El presente Trabajo Práctico Integrador tiene como finalidad la aplicación y consolidación de los conocimientos adquiridos durante la cursada de Programación Visual. A lo largo de este proyecto, se abordará el desarrollo progresivo de una aplicación web funcional, utilizando React, Vite, JavaScript, React Router DOM, y todos los temas vistos en la materia.

Este trabajo no solo busca integrar los diversos conceptos teóricos y herramientas prácticas exploradas en la materia, sino también simular un proceso de desarrollo iterativo. La aplicación irá evolucionando y sumando funcionalidades conforme a las pautas y entregas que se detallarán progresivamente. El objetivo último es demostrar la capacidad de diseñar, implementar y gestionar una solución de software frontend robusta, aplicando buenas prácticas de desarrollo y aprovechando las capacidades del ecosistema JavaScript actual para crear experiencias de usuario interactivas y eficientes.



Es importante aclarar que la metodología de evaluación para este Trabajo Práctico Integrador se realizará de forma continua y progresiva, alineada con el carácter evolutivo del desarrollo del proyecto.

Las evaluaciones se realizarán de manera periódica, con una frecuencia semanal, y estarán relacionadas con los avances presentados de acuerdo a los enunciados y requerimientos que el profesor a cargo de la comisión exija, por ejemplo, defensa oral con cámara y micrófono activo.

El Proyecto

Desarrollar una SPA utilizando React + Vite, que permita implementar el manejo de estado global mediante Context API o Redux, con el fin de centralizar y compartir el estado de la aplicación entre diferentes componentes y páginas.

La aplicación gestiona datos de productos a través de las siguientes funcionalidades: crear, editar, eliminar y visualizar información detallada de cada producto.

La aplicación debe estar estructurada en componentes funcionales y manejar correctamente los eventos en React y la navegación entre vistas.

Configuración del proyecto

Crear el proyecto utilizando Vite con plantilla de React.

Utilizar JSX para estructurar la interfaz.

Usar React Router Dom

Usar framework de interfaces de usuario

Si fuera necesario intalar Redux.

Funcionalidades que debe incluir la Aplicación

Página de Inicio (Home): Mostrar un listado de productos en formato de tarjetas (cards).
Cada producto tendrá:

- Id
 - Imagen representativa.
 - Nombre.
 - Precio.
 - Descripción
 - Categoría
-
- Botón para ver más detalles.
 - Ícono o checkbox para marcar como favorito.

Funcionalidad de Favoritos:

Los productos pueden ser marcados/desmarcados como favoritos.

El estado de los productos favoritos se almacenará en un estado global, accesible desde toda la aplicación.

Este manejo de estado global se realizará utilizando React Context API o Redux, según lo indique el profesor.

Página de Favoritos:

Mostrar únicamente los productos que el usuario ha marcado como favoritos.

Página de Detalle del Producto:

Accesible desde el botón “Ver más detalles” de cada card.

Mostrar información ampliada del producto (descripción, categoría, stock, etc.).

Permitir desmarcar el producto (favorito) desde esta vista, lo cual actualizará automáticamente el estado global.

Formulario de Creación y Edición:

Debe haber una vista o componente reutilizable que permita:

Crear un nuevo producto desde cero (botón accesible desde la Home o el menú principal).

Editar un producto existente, con los campos precargados según el ID del producto.

Consumo de API

La aplicación deberá consumir productos desde una API REST externa para alimentar el listado principal de productos.

Se utilizará la API pública:

<https://fakestoreapi.com/products>

El consumo debe realizarse mediante Fetch o Axios, según lo especifique el profesor.

Al cargar la aplicación, se realiza una petición GET para traer los productos.

Los productos obtenidos deben guardarse en el estado global para que estén disponibles en toda la app.

Toda la lógica de favoritos, edición o eliminación se aplicará sobre estos datos.

Versionamiento

Subir el proyecto a un repositorio en GitHub, el nombre del repositorio debe ser **pv_tp_integrador_grupo999**, donde 999 es el número del grupo.

Todos los cambios deben subirse a través de ramas.

En el archivo README.md incluir la lista de integrantes del grupo con sus respectivos nombres de usuarios github.

Añadir la descripción del proyecto.