Universidad de la Cuenca del Plata
Autorización Definitiva. Decreto Poder Ejecutivo Nacional N°091/2006

# Trabajo Practico

CARRERA: Ingeniería en sistemas de información

MATERIA: Paradigmas y Lenguajes de Programación III

COMISIÓN: “U” (única) “A”

PROFESOR: Mgter. Ing. Agustín Encina

ESTUDIANTES: Benitez Lucia.

FECHA: 02-09-2025

## Introducción

El proyecto Ahorra+ surge como una propuesta de aplicación web destinada a la gestión y organización de finanzas personales, con el objetivo de brindar a los usuarios herramientas simples y visuales para controlar ingresos, egresos y metas de ahorro.

Asimismo, el propósito inicial de este trabajo fue aplicar buenas prácticas en el uso de HTML y CSS, además de incorporar una metodología de desarrollo ordenada, que facilite la escalabilidad y mantenibilidad del sistema a futuro.

En esta primera etapa se trabajó en el diseño de la arquitectura inicial, la estructura del proyecto y la integración de tecnologías base, priorizando la modularidad y la claridad en la organización del código.

# Desarrollo

## Estado Actual del Proyecto

Actualmente, el proyecto se encuentra en una fase inicial de desarrollo, donde se implementaron principalmente la estructura y el diseño de los componentes. Se construyó una arquitectura organizada en capas, con carpetas para components, pages, shared, utils y constants.

Se desarrollaron páginas base

- Dashboard

- FixedExpenses

- SavingsGoal

- VariableIncome

### Incorporaciones

Se incorporaron componentes de interfaz reutilizables, como tablas, formularios de gastos y selectores de mes.

Los datos se simulan mediante archivos JSON, lo cual permite probar comportamientos sin necesidad de contar aún con una base de datos real.

Se implementó una funcionalidad condicional básica: cuando el JSON define que

el ingreso es de tipo *variable*, en el Dashboard aparece un botón que redirige a una página exclusiva para usuarios con ingresos variables.

En este punto, las funcionalidades son mínimas y se limitan a algunos cálculos simples en JavaScript y a la visualización de la estructura de la interfaz.

**Tecnologías Utilizadas y Justificación**

El stack tecnológico fue seleccionado considerando la eficiencia en el desarrollo, la escalabilidad futura y la facilidad de mantenimiento.

- React

  Se utilizó para el desarrollo de la interfaz basada en componentes reutilizables, lo cual permite una organización clara y un mantenimiento más sencillo.

- Tailwind CSS

  Se incorporó para el diseño visual, ya que facilita el desarrollo de interfaces responsive y modernas con menor cantidad de código CSS personalizado.

- JavaScript, HTML y CSS

  Constituyen la base del desarrollo web, utilizados para la lógica principal, estructura y estilos complementarios.

- Vite

  Elegido como compilador y empaquetador por su rendimiento superior en

comparación con herramientas tradicionales como Webpack. Esto agiliza los tiempos de desarrollo y mejora la experiencia del programador.

- Recharts

  Librería especializada en visualización de datos que se usará para representar gráficamente ingresos, gastos y metas de ahorro, permitiendo un análisis más intuitivo de la información.

- Lucide Icons

  Implementados para mejorar la estética del proyecto mediante íconos simples y modernos.

- Archivos JSON

  Utilizados para la simulación de datos en ausencia de una base de datos real, lo cual permite verificar la estructura y las reglas de negocio de forma preliminar.

- Recursos PNG

  Imágenes utilizadas para reforzar la identidad visual del sistema.

**Avances Técnicos**

- Se implementó un sistema de navegación básica entre páginas.

- Se estructuraron componentes de dashboard como tablas de gastos, formularios de ingreso fijo y variable, y resúmenes de totales.

- Se realizaron cálculos básicos en JavaScript para pruebas iniciales de sumatorias y diferencias.

Se desarrolló un primer control de acceso simulado, basado en los datos del JSON, que diferencia entre usuarios con ingresos fijos y variables.

## Futuras Implementaciones

Se plantean como próximos pasos los siguientes objetivos:

- Gestión de estado global con Redux: implementar Redux Toolkit para un manejo más eficiente de los estados de la aplicación.

- Arquitectura Limpia (Clean Architecture): reorganizar el proyecto en capas de dominio, aplicación, infraestructura y presentación, con el fin de aumentar la mantenibilidad y escalabilidad.

- Persistencia de datos: reemplazar los JSON por una base de datos y una API que permita registrar información real de los usuarios.

**Funcionalidades avanzadas**

- Registro y autenticación de usuarios.

- Configuración personalizada de metas de ahorro.

- Visualización de estadísticas en tiempo real.

- Expansión del uso de gráficos: mayor explotación de Recharts para ofrecer una visión detallada de los movimientos financieros.

## Conclusión

El proyecto Ahorra+ presenta un avance significativo en términos de diseño y arquitectura inicial. Aunque las funcionalidades implementadas aún son básicas, la estructura creada y la selección de tecnologías aseguran una base sólida para futuras etapas de desarrollo.

El siguiente desafío radica en la incorporación de Redux para la gestión de estados, la implementación de Clean Architecture y el desarrollo de una API que permita transformar este prototipo en una herramienta funcional para la gestión de finanzas personales.