

# TP N.º 1 Paradigmas III:

Avances del integrados

**CARRERA:** Ingeniería en Sistemas de Información

**MATERIA:** Paradigmas y Lenguajes de Programación III

**COMISIÓN:** A

**PROFESOR:** Mgter . Ing. . Agustín Encina

**ESTUDIANTE:** Matias Nahuel Mantaras

# Descripción del Proyecto Integrador

Premium Store es un proyecto de e-commerce desarrollado como trabajo integrador para la materia Paradigmas y Lenguajes de Programación III. Se trata de una aplicación web que simula una tienda en línea completa, implementando las mejores prácticas de desarrollo frontend.

---

## Arquitectura del Programa

Para armar este proyecto de e-commerce, he estructurado el programa en tres componentes principales que trabajarán en conjunto:

### 1. Frontend (Interfaz de Usuario) - IMPLEMENTADO

El frontend es la parte de la aplicación que los usuarios verán y con la que interactuarán directamente a través de un navegador. La principal tarea es asegurar que la información se muestre de forma clara y atractiva.

#### Componentes Implementados:

##### • Página de Login y Registro

- Punto de entrada al sistema
- Formulario sencillo para correo electrónico y contraseña
- Al hacer clic en "Ingresar", envía información al backend para validación
- Redirección automática tras autenticación exitosa

##### • Catálogo de Productos

- Página principal tras autenticación
- Muestra todos los productos disponibles
- Obtiene lista de productos del backend
- Cada producto se muestra con nombre, precio e imagen
- Disponible en dos formatos: tarjetas y tabla

##### • Página del Producto y Carrito de Compras

- Vista detallada al seleccionar un producto
- Botón "Comprar" que simula adición al carrito
- Preparación automática para el proceso de pago

- Gestión visual del carrito con contador

#### • **Página de Pago**

- Formulario para datos de tarjeta de prueba
- Simulación completa del proceso de compra
- Validación de campos requeridos
- Integración con el flujo de carrito

#### • **Mensaje de Confirmación**

- Pantalla final con "Compra Finalizada"
- Confirmación visual del proceso exitoso
- Opción de continuar navegando

#### **Tecnologías Frontend:**

- HTML5 para la estructura semántica
- CSS3 para el diseño visual y animaciones
- JavaScript para la interactividad y validaciones
- Tailwind CSS como framework de utilidades
- Font Awesome para iconografía
- SVG para imágenes vectoriales optimizadas

## **2. Backend (Lógica de Negocio) - PENDIENTE**

*Será implementado en futuras iteraciones del curso*

- Validación de credenciales de usuarios
- Gestión de base de datos de productos
- Procesamiento de pagos simulados
- API REST para comunicación con frontend
- Autenticación y autorización con JWT
- Middleware de seguridad

## **3. Base de Datos - PENDIENTE**

*Será implementada en futuras iteraciones del curso*

- Almacenamiento de usuarios registrados
- Catálogo completo de productos
- Historial de compras y transacciones
- Gestión de sesiones de usuario
- Logs de actividad del sistema

---

# Análisis de Requerimientos Inicial

## Requerimientos Funcionales

### RF001 - Autenticación de Usuario

- **Descripción:** El sistema debe permitir a los usuarios registrarse e iniciar sesión con email y contraseña
- **Prioridad:** Alta
- **Estado:** ☒ Implementado (Frontend)
- **Criterios de Aceptación:**
  - Formulario de login funcional
  - Validación de campos requeridos
  - Redirección tras login exitoso

### RF002 - Visualización de Catálogo


- **Descripción:** El sistema debe mostrar una lista completa de productos disponibles con información básica
- **Prioridad:** Alta
- **Estado:** ☒ Implementado
- **Criterios de Aceptación:**
  - Lista de productos con imagen, nombre y precio
  - Dos formatos de visualización (tarjetas y tabla)
  - Navegación fluida entre productos

### RF003 - Detalle de Producto


- **Descripción:** El sistema debe mostrar información detallada de cada producto individual
- **Prioridad:** Alta
- **Estado:** ☒ Implementado
- **Criterios de Aceptación:**
  - Página individual por producto
  - Información completa (descripción, características, precio)
  - Botón de compra funcional

### RF004 - Gestión de Carrito

- **Descripción:** El sistema debe permitir agregar productos al carrito de compras
- **Prioridad:** Media


- **Estado:**  Implementado (Simulación)
- **Criterios de Aceptación:**
  - Adición de productos al carrito
  - Contador visual de productos
  - Persistencia durante la sesión

## RF005 - Proceso de Pago


- **Descripción:** El sistema debe procesar pagos con datos de tarjeta de prueba
- **Prioridad:** Media
- **Estado:**  Implementado (Simulación)
- **Criterios de Aceptación:**
  - Formulario de pago funcional
  - Validación de datos de tarjeta
  - Confirmación de compra exitosa

## Requerimientos No Funcionales

### RNF001 - Usabilidad

- **Descripción:** La interfaz debe ser intuitiva y fácil de usar
- **Criterios:**
  - Diseño responsive para móviles y desktop
  - Navegación clara e intuitiva
  - Tiempo de aprendizaje mínimo
- **Estado:**  Implementado

### RNF002 - Rendimiento

- **Descripción:** El sistema debe tener tiempos de respuesta óptimos
- **Criterios:**
  - Tiempo de carga de páginas menor a 3 segundos
  - Imágenes optimizadas en formato SVG
  - Código CSS y JS minificado
- **Estado:**  Implementado

### RNF003 - Compatibilidad

- **Descripción:** El sistema debe funcionar en diferentes navegadores
- **Criterios:**
  - Compatible con Chrome, Firefox, Safari, Edge
  - Soporte para versiones recientes de navegadores
  - Degradación elegante en navegadores antiguos

- **Estado:**  Implementado
- 

## Casos de Uso Básicos

### CU001 - Iniciar Sesión

- **Actor:** Usuario no autenticado
- **Precondición:** El usuario accede a la página de login
- **Flujo Principal:**
  - El usuario accede a la página de login
  - Ingresa su email en el campo correspondiente
  - Ingresa su contraseña en el campo correspondiente
  - Hace clic en el botón "Ingresar"
  - El sistema valida las credenciales (simulación)
  - El sistema redirige al catálogo de productos
- **Flujo Alternativo:**
  - Si las credenciales son inválidas, mostrar mensaje de error
  - Si faltan campos, mostrar validación correspondiente
- **Postcondición:** Usuario autenticado y redirigido al catálogo

### CU002 - Visualizar Catálogo

- **Actor:** Usuario autenticado
- **Precondición:** Usuario ha iniciado sesión exitosamente
- **Flujo Principal:**
  1. El sistema carga automáticamente el catálogo de productos
  2. El usuario puede ver la lista completa de productos
  3. Cada producto muestra imagen, nombre y precio
  4. El usuario puede alternar entre vista de tarjetas y tabla
  5. El usuario puede seleccionar un producto para ver detalles
- **Postcondición:** Catálogo mostrado correctamente con todos los productos

### CU003 - Comprar Producto

- **Actor:** Usuario autenticado
- **Precondición:** Usuario está visualizando un producto específico
- **Flujo Principal:**
  - El usuario revisa los detalles del producto
  - Hace clic en el botón "Comprar" o "Agregar al Carrito"
  - El producto se agrega al carrito (actualización del contador)

- El usuario procede al proceso de pago
- Ingresa datos de tarjeta de prueba en el formulario
- Confirma la compra haciendo clic en "Finalizar Compra"
- El sistema procesa la información (simulación)
- Se muestra el mensaje "Compra Finalizada"
- **Flujo Alternativo:**
  - Si faltan datos en el formulario de pago, mostrar validaciones
  - Si hay error en el proceso, mostrar mensaje correspondiente
- **Postcondición:** Compra simulada exitosamente y confirmación mostrada

## CU004 - Navegar entre Páginas

- **Actor:** Usuario (autenticado o no)
  - **Precondición:** Usuario está en cualquier página del sitio
  - **Flujo Principal:**
    1. El usuario identifica el menú de navegación en el header
    2. Selecciona la página deseada del menú
    3. El sistema carga la nueva página solicitada
    4. El usuario puede continuar navegando libremente
  - **Postcondición:** Usuario navega exitosamente entre las diferentes secciones
- 

## Diseño de la Aplicación

### Flujo de Usuario Principal

**Acceso → Login → Catálogo → Producto → Carrito → Pago → Confirmación**

### Estructura de Navegación

- **Inicio:** Página principal con productos destacados y hero section
- **Productos:** Vista en tarjetas del catálogo completo
- **Catálogo:** Vista en tabla con información detallada
- **Producto:** Página individual con detalles completos
- **Comprar:** Proceso de compra y gestión del carrito
- **Login:** Autenticación y registro de usuarios

### Paleta de Colores

- **Primario:** Púrpura (#8B5CF6) - Botones principales y enlaces
- **Secundario:** Gris (#6B7280) - Texto secundario y bordes

- **Acentos:** Verde (#10B981) para éxito, Rojo (#EF4444) para errores
- **Fondo:** Gris claro (#F9FAFB) - Fondo general de la aplicación
- **Blanco:** (FFFFFF) - Tarjetas y contenedores principales

## Principios de Diseño Aplicados

- **Minimalismo:** Interfaz limpia sin elementos innecesarios
  - **Consistencia:** Patrones de diseño uniformes en toda la aplicación
  - **Accesibilidad:** Contraste adecuado y navegación por teclado
  - **Responsive:** Adaptación fluida a diferentes tamaños de pantalla
  - **Usabilidad:** Flujos intuitivos y retroalimentación clara
- 

## Estado Actual del Proyecto

### **Completado (TP1 - Primera Instancia)**

- Estructura completa del frontend
- Todas las páginas HTML implementadas
- Estilos CSS responsivos y modernos
- JavaScript para interactividad básica
- Navegación funcional entre páginas
- Simulación completa del flujo de compra
- Imágenes SVG optimizadas
- Arquitectura de carpetas escalable

### **Pendiente (Futuras Iteraciones)**

- Implementación del backend con Node.js o Python
- Base de datos para persistencia de datos
- Autenticación real con JWT
- API REST para comunicación frontend-backend
- Validaciones del lado del servidor
- Sistema de pagos real (integración con Stripe/PayPal)
- Panel de administración
- Sistema de usuarios y perfiles



# Conclusiones del TP1

Este primer trabajo práctico ha permitido establecer una base sólida para el proyecto Premium Store, implementando exitosamente todo el frontend de la aplicación. Se han aplicado las mejores prácticas de desarrollo web moderno, creando una interfaz atractiva, funcional y escalable.

La arquitectura implementada facilita la futura integración con backend y base de datos, manteniendo una separación clara de responsabilidades y un código limpio y mantenible.

El proyecto demuestra competencia en:

- Desarrollo frontend moderno
- Diseño responsive
- Experiencia de usuario (UX)
- Arquitectura de software
- Análisis de requerimientos
- Documentación técnica

---

**Fecha de Entrega:** [Fecha del TP1]

**Versión:** 1.0 - Primera Instancia

**Próxima Iteración:** Implementación de Backend y Base de Datos

---

*Desarrollado como proyecto integrador - Paradigmas y Lenguajes de Programación III*