



**Universidad de las Fuerzas Armadas ESPE**  
**Ciencias de la Computación**  
**Programación Web**  
**Conjunta Parcial 2**

**Nombres:** Leonel Tipan, Mayeli ortiz

**NRC:**2257

**Fecha:**24/01/2025

**Ingeniero:** Diego Gamboa

**Caso Práctico Escenario:** Una biblioteca digital necesita mejorar su sistema de gestión de préstamos y devoluciones de libros. Actualmente, los usuarios pueden buscar libros, pero no hay una forma eficiente de reservarlos, devolverlos o verificar su disponibilidad en tiempo real. Además, el sistema no cuenta con alertas para recordar a los usuarios la fecha de devolución ni notificaciones sobre la disponibilidad de libros reservados.

Desafío para los estudiantes:

Se requiere que los estudiantes analicen y propongan una solución para mejorar el sistema de gestión de la biblioteca digital, aplicando los conceptos de JavaScript aprendidos en clase. En su análisis, deberán considerar lo siguiente:

#### **Uso de la clase Libro**

Esta clase modela la estructura básica de un libro.

- Encapsula datos importantes del libro
- Mantiene el estado de disponibilidad
- Registra información de préstamo

Datos clave

- id: Identificador único
- título, autor, género: información descriptiva
- disponible: estados de los préstamos
- fechaPréstamo: Seguimiento temporal

#### **Uso de la clase BibliotecaDigital**

Gestiona el catálogo y operaciones del libro

- **Métodos principales:**
  - buscarLibros(): Filtrado dinámico con filter()
  - prestarLibro(): Cambiar estado de disponibilidad
  - devolverLibro(): Restaurar libro al catálogo

**Beneficios:**

- Centraliza la lógica de gestión de libros
- Permite operaciones complejas de manera simple
- Mantiene la integridad de los datos del catálogo

#### **Uso de la clase SistemaNotificaciones**

Conectar modelo de datos con la interfaz de usuario

- **Metodos Principales:**

- crearNotificacion(): Generar mensajes
- mostrarNotificacion(): Renderizar en interfaz
- notificarDisponibilidad(): Alertas asíncronas
- recordatoriosDevoluciones(): Seguimiento de préstamos

**Ventajas:** Desacoplamiento de la lógica de notificación Manejo de diferentes tipos de alertas Implementación de promesas para procesos asincrónicos

### **Clase InterfazBiblioteca**

Propósito: Conectar modelo de datos con interfaz de usuario

- **Responsabilidades:**

- inicializarEventos(): Configurar interacciones
- renderizarLibros(): Actualizar vista dinámica
- Gestionar interacciones de préstamo

- **Beneficios:**

Separación de concerns

Renderización reactiva

Gestión de eventos de usuario

## **Principios de Diseño Implementados**

1. **Programación Orientada a Objetos**

- Clases con responsabilidades únicas
- Encapsulamiento de datos y comportamientos

2. **Funcional Reactivo**

- Uso de filter(), map()
- Promesas para operaciones asincrónicas

3. **Modularidad**

- Cada clase con función específica
- Fácil mantenimiento y extensión

4. **Gestión de Estado**

- Seguimiento dinámico de libros
- Control de préstamos y disponibilidad

## Consideraciones Técnicas

- **Asincronía:** setTimeout() simula latencias reales
- **Eventos:** Listeners para interacciones
- **Notificaciones:** Sistema flexible de alertas
- **Renderización:** Actualización dinámica del DOM

## Beneficios para el Usuario

- Búsqueda intuitiva de libros
- Proceso de préstamo transparente
- Notificaciones contextuales
- Interfaz reactiva y moderna

## Mejoras Potenciales

- Integración con base de datos
- Autenticación de usuarios
- Historial de préstamos más detallado
- Reservas de libros