

Arquitectura del Sistema Web de Gestión Bibliotecaria FISI

Versión Final - Sistema Implementado

1. Vista General de la Arquitectura

Arquitectura en 3 Capas (3-Tier Architecture)

<p>CAPA DE PRESENTACIÓN (Fronted – Interfaz de Usuario) HTML5 + CSS3 + JavaScript + Bootstrap</p>
<p>CAPA DE LÓGICA DE NEGOCIO (Backend – Procesamiento) PHP 8+ con PDO + MVC</p>
<p>CAPA DE DATOS (Base de Datos MySQL) 9 Tablas + Vistas + Triggers</p>

2. Estado Actual del Sistema (Implementado)

Datos Reales Cargados:

- **6 Libros únicos** del catálogo oficial FISI
- **27 Ejemplares físicos** con códigos únicos
- **8 Categorías** basadas en clasificación bibliotecaria
- **2 Usuarios** del sistema (admin + bibliotecario)
- **100% Integridad** de datos verificada

Origen de los Datos:

- Basado en análisis de **3,373 registros** del catálogo oficial
 - Libros reales: "Lógica Matemática", "Psicología Revolucionaria", "Principios de Economía"
 - Clasificaciones auténticas: BC135.C78, BF176.A88, HB171.5.M22
-

3. Estructura de Directorios del Proyecto

biblioteca-fisi/

```
|
|
|— index.php          # Página principal del sistema
|— prueba_conexion.php # IMPLEMENTADO - Verificación BD
|
|
|— config/
|   |— database.php    # Configuración de conexión BD
|   └— config.php      # Configuraciones generales
|
|
|— models/             # Capa de Datos (Modelos)
|   |— Usuario.php     # Modelo para tabla 'usuarios'
|   |— Libro.php       # Modelo para tabla 'libros'
|   |— Prestamo.php    # Modelo para tabla 'prestamos'
|   |— EjemplarLibro.php # Modelo para tabla 'ejemplares_libros'
|   └— BaseDatos.php   # Clase de conexión PDO
|
|
|— controllers/        # Capa de Lógica (Controladores)
|   |— AutenticacionController.php
|   |— LibroController.php
|   |— PrestamoController.php
|   |— UsuarioController.php
|   └— EstadisticasController.php
|
|
|— views/              # Capa de Presentación (Vistas)
|   |— layouts/
|       |— header.php
|       |— footer.php
|       └— sidebar.php
|   |— auth/
```

```

|   |   |— login.php
|   |   |— registro.php
|   |— catalogo/
|   |   |— busqueda.php
|   |   |— detalles.php
|   |— prestamos/
|   |   |— lista.php
|   |   |— crear.php
|   |   |— devolver.php
|   |— admin/
|   |   |— dashboard.php
|   |   |— libros.php
|   |   |— usuarios.php
|   |   |— estadisticas.php
|   |— usuario/
|   |   |— dashboard.php
|   |   |— perfil.php
|
|— assets/          # Recursos estáticos
|   |— css/
|   |   |— bootstrap.min.css
|   |   |— custom.css
|   |— js/
|   |   |— bootstrap.min.js
|   |   |— main.js
|   |— img/
|   |   |— logo-fisi.png
|
|— includes/        # Archivos auxiliares
|   |— functions.php  # Funciones globales en español

```

```
|      └── session.php      # Manejo de sesiones
|
|
└── sql/
    ├── base_datos.sql      # IMPLEMENTADO - Script principal
    └── datos_ejemplo.sql   # IMPLEMENTADO - Libros del catálogo
```

4. Base de Datos - Estado Actual

Tablas Implementadas (9 tablas):

Tabla	Registros	Descripción
usuarios	2	Admin y bibliotecario
categorías	8	Clasificación bibliotecaria (BC, BF, HB, etc.)
libros	6	Títulos únicos del catálogo real
ejemplares_libros	27	Copias físicas con códigos únicos
prestamos	0	Préstamos (tabla lista para usar)
historial prestamos	0	Historial de acciones
reservas	0	Sistema de reservas
multas	0	Control de multas
libro_categorias	6	Relaciones libro-categoría

Características Avanzadas:

- **Triggers automáticos** para actualizar estados de ejemplares
 - **Vistas optimizadas** (vista_prestamos_activos, vista_disponibilidad_libros)
 - **Índices estratégicos** para búsquedas rápidas
 - **Integridad referencial** completa
-

5. Mapeo de Historias de Usuario

HU-01: Consulta de Catálogo

Estado: Base de datos lista

- **Tablas:** libros, ejemplares_libros, vista_disponibilidad_libros
- **Datos reales:** 6 libros con 27 ejemplares disponibles
- **Controlador:** LibroController.php
- **Vista:** views/catalogo/busqueda.php

HU-02: Préstamos y Devoluciones

Estado: Estructura completa

- **Tablas:** prestamos, historial_prestamos, multas
- **Funcionalidad:** Lista para implementar lógica de préstamos
- **Controlador:** PrestamoController.php
- **Vista:** views/prestamos/

HU-03: Gestión de Usuarios (Login)

Estado: Usuarios creados

- **Tablas:** usuarios (2 usuarios listos)
- **Roles:** Estudiante, Docente, Bibliotecario, Administrador
- **Controlador:** AutenticacionController.php
- **Vista:** views/auth/login.php

HU-04: Estadísticas Generales

Estado: Datos disponibles para reportes

- **Fuente:** Todas las tablas con datos reales
- **Métricas disponibles:** Libros por categoría, ejemplares por estado
- **Controlador:** EstadisticasController.php

HU-05: Gestión de Inventario

Estado: Catálogo base cargado

- **Funcionalidad:** CRUD de libros y ejemplares
- **Importación:** Lista para catálogo completo (3,373 registros)
- **Controlador:** LibroController.php

6. Tecnologías y Justificación

Stack Tecnológico Elegido:

- **PHP 8+:** Lenguaje maduro, amplia documentación, compatible con XAMPP
- **MySQL 8+:** Base de datos robusta, soporte completo para español (UTF-8)
- **Apache:** Servidor web estable incluido en XAMPP
- **Bootstrap 5:** Framework CSS para desarrollo rápido y responsivo
- **PDO:** Conexión segura a base de datos, previene inyección SQL

Entorno de Desarrollo:

- **XAMPP:** Entorno local completo y fácil de configurar

- **phpMyAdmin:** FUNCIONANDO - Administración visual de BD
 - **Git:** Control de versiones para trabajo en equipo
 - **Visual Studio Code:** Editor con extensiones para PHP
-

7. Patrones de Diseño Implementados

MVC (Modelo-Vista-Controlador):

- **Modelo:** Interacción con tablas específicas (usuarios, libros, etc.)
- **Vista:** Interfaz de usuario en HTML/CSS/JS
- **Controlador:** Lógica de negocio en PHP

Singleton para Conexión BD:

- Una sola instancia de conexión PDO por request
- Optimización de recursos y conexiones

Repository Pattern:

- Cada tabla tiene su modelo correspondiente
 - Separación clara entre lógica de negocio y acceso a datos
-

8. Seguridad Implementada

Medidas de Seguridad:

- **Passwords hasheados** con password_hash() PHP
- **Conexión PDO** con prepared statements
- **Charset UTF-8** para soporte completo de español
- **Validación de integridad** automática con triggers

Próximas implementaciones:

- Validación de sesiones
 - Control de acceso por roles
 - Sanitización de inputs de usuario
 - Logs de actividad del sistema
-

9. Escalabilidad y Mantenimiento

Características de Escalabilidad:

- **Arquitectura modular** permite agregar funcionalidades
- **Base de datos normalizada** evita redundancia
- **Vistas optimizadas** para consultas complejas
- **Índices estratégicos** para rendimiento

Capacidad Probada:

- **Maneja 27 ejemplares** actualmente
 - **Preparado para 3,373 registros** del catálogo completo
 - **Soporte para múltiples roles** de usuario
 - **Estructura extensible** para nuevas funcionalidades
-

10. Demostración del Sistema

URL de Prueba:

http://localhost/biblioteca-fisi/prueba_conexion.php

Evidencia de Funcionamiento:

- Conexión exitosa a base de datos
- 6 libros únicos cargados
- 27 ejemplares disponibles
- 8 categorías bibliotecarias
- 2 usuarios del sistema
- 100% integridad de datos