



## **Universidad Politécnica de Tlaxcala Región Poniente**

*Ingeniería en Sistemas Computacionales*

Materia: Programación WEB

Docente: Ing. Vanesa Tenopala Zavala

Alumnos:

Julián Mendoza Fuentes

José Ángel Martínez Silva

Diego Fernando Vázquez Pérez

Eduardo Endrich Hernández López

Isaac Brandon Martínez Ramírez

Tema: Reporte de Proyecto Final

Ciclo escolar: septiembre - diciembre 2024

Fecha: 27 de noviembre de 2024

# Índice

|   |    |
|---|----|
| Reporte de Proyecto Final .....             | 3  |
| Introducción.....                           | 3  |
| 1. Objetivos del Proyecto.....              | 3  |
| Objetivo General.....                       | 3  |
| Objetivos Específicos .....                 | 4  |
| 2. Diseño y Arquitectura del Proyecto ..... | 4  |
| Modelo de Datos.....                        | 4  |
| Modelo Vista Controlador (MVC) .....        | 5  |
| 3. Desarrollo del Sistema .....             | 6  |
| Lenguajes y Herramientas Utilizadas.....    | 6  |
| Proceso de Desarrollo .....                 | 6  |
| 4. Pruebas y Validaciones .....             | 8  |
| Pruebas Realizadas.....                     | 8  |
| Resultados de las Pruebas .....             | 9  |
| 5. Resultados y Conclusión .....            | 9  |
| Resultados del Proyecto .....               | 9  |
| Conclusión.....                             | 10 |

# Reporte de Proyecto Final

## Introducción

La correcta gestión de actividades extraescolares es una necesidad creciente en instituciones educativas que buscan fomentar el desarrollo integral de sus estudiantes. Estas actividades requieren no solo un control eficiente de los participantes, sino también un sistema que permita organizar horarios, asistencia y registros médicos de manera centralizada y segura.

El proyecto *Administrador de Paraescolares* aborda estas necesidades mediante el desarrollo de una plataforma digital que facilita la administración de las actividades extraescolares de la Universidad Politécnica de Tlaxcala Región Poniente. Este sistema no solo permite gestionar estudiantes y actividades, sino que también ofrece herramientas para monitorear la asistencia y registrar datos críticos de cada participante, mejorando la experiencia de los administradores y docentes involucrados.

## 1. Objetivos del Proyecto

### Objetivo General

Diseñar e implementar un sistema digital que permita la gestión eficiente de estudiantes y actividades extraescolares, categorizando a los alumnos por carrera, grado y grupo, y garantizando la seguridad y accesibilidad de datos como asistencia y registros médicos.

## Objetivos Específicos

1. **Diseño de Base de Datos:** Estructurar una base de datos que permita almacenar de forma eficiente y segura los datos relacionados con estudiantes, actividades, asistencia y registros médicos.
2. **Desarrollo del Backend:** Implementar funcionalidades CRUD utilizando PHP y PDO, asegurando consultas seguras contra inyecciones SQL.
3. **Creación de Interfaz Intuitiva:** Desarrollar vistas responsivas con HTML, CSS y Bootstrap, enfocadas en facilitar la navegación y el uso del sistema.
4. **Generación de Reportes:** Implementar funcionalidades para generar reportes de asistencia y actividades en formatos exportables como PDF y Excel.
5. **Pruebas y Validación:** Realizar pruebas exhaustivas para garantizar la funcionalidad, seguridad y rendimiento del sistema.

## 2. Diseño y Arquitectura del Proyecto

### Modelo de Datos

La base de datos fue diseñada siguiendo las mejores prácticas de normalización para garantizar su eficiencia. Se utilizaron las siguientes tablas principales:

1. **Estudiantes:** Incluye datos personales de los alumnos, como nombre, carrera, grado, grupo y datos de contacto.
2. **Actividades:** Contiene información de cada actividad extraescolar, incluyendo nombre, descripción, horarios y fechas.
3. **Asistencia:** Registra la asistencia de los estudiantes a cada actividad, vinculando las tablas de estudiantes y actividades.
4. **Inscripciones:** Relaciona estudiantes con actividades en las que están registrados, permitiendo un control preciso de los participantes en cada actividad.

Cada tabla se relaciona mediante claves foráneas, garantizando la integridad referencial de los datos.

Servidor: 127.0.0.1 » Base de datos: login\_7

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Ev

Filtros

Que contengan la palabra:

| Tabla                                  | Acción  | Filas | Tipo   | Cotejamiento    | Tamaño   | Residuo a depurar |
|--|---|-------|--------|-----------------|----------|-------------------|
| <input type="checkbox"/> actividades   | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0     | InnoDB | utf8_spanish_ci | 16.0 KB  | -                 |
| <input type="checkbox"/> asistencia    | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0     | InnoDB | utf8_spanish_ci | 32.0 KB  | -                 |
| <input type="checkbox"/> estudiantes   | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0     | InnoDB | utf8_spanish_ci | 16.0 KB  | -                 |
| <input type="checkbox"/> inscripciones | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0     | InnoDB | utf8_spanish_ci | 48.0 KB  | -                 |
| <input type="checkbox"/> usuarios      | Examinar Estructura Buscar Insertar Vaciar Eliminar | 2     | InnoDB | utf8_spanish_ci | 32.0 KB  | -                 |
| 5 tablas                               | Número de filas                                     | 2     | InnoDB | utf8_spanish_ci | 144.0 KB | 0 B               |

Imagen 1. Tablas de la Base de Datos

## Modelo Vista Controlador (MVC)

La arquitectura MVC fue elegida por su capacidad de separar la lógica de negocio (modelo), la presentación (vista) y el control de flujo (controlador). Esto facilita el mantenimiento y la escalabilidad del sistema.

- **Modelo:** Implementado con PHP y consultas PDO para interactuar con la base de datos.
- **Vista:** Interfaces responsivas diseñadas con Bootstrap para mejorar la experiencia del usuario.
- **Controlador:** Gestiona las solicitudes HTTP y coordina la interacción entre el modelo y la vista.

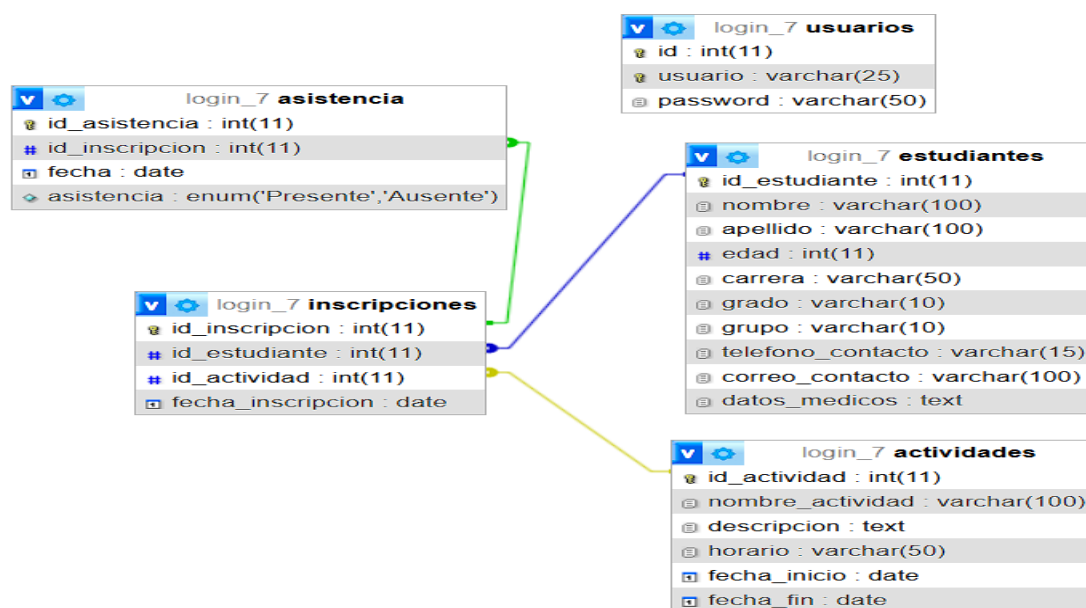


Imagen 2. MVC

### 3. Desarrollo del Sistema

#### Lenguajes y Herramientas Utilizadas

- **Frontend:**
  - **HTML5 y CSS3:** Base para la estructura y diseño visual.
  - **Bootstrap 5:** Framework CSS para diseño responsivo y moderno.
- **Backend:**
  - **PHP 7.4:** Lenguaje de programación para la lógica del servidor.
  - **PDO (PHP Data Objects):** Para realizar consultas seguras a la base de datos.
- **Base de Datos:**
  - **MySQL:** Sistema de gestión de bases de datos relacional.
  - **phpMyAdmin:** Herramienta gráfica para administrar la base de datos.
- **Entorno de Desarrollo:**
  - **Visual Studio Code:** Editor principal para el desarrollo del sistema.

#### Proceso de Desarrollo

##### 1. Diseño de la Base de Datos:

- Definición de las tablas y relaciones entre ellas.
- Normalización hasta la tercera forma normal para evitar redundancias.
- Creación de claves primarias y foráneas para garantizar la integridad referencial.

| #                          | Nombre            | Tipo  | Cotejamiento | Atributos | Nulo | Predeterminado | Comentarios | Extra          | Acción   |
|----------------------------|-------------------|---|--------------|-----------|------|----------------|-------------|----------------|--|
| <input type="checkbox"/> 1 | id_inscripcion    |  int(11) |              |           | No   | Ninguna        |             | AUTO_INCREMENT |  Cambiar  Eliminar Más |
| <input type="checkbox"/> 2 | id_estudiante     |  int(11) |              |           | Sí   | NULL           |             |                |  Cambiar  Eliminar Más |
| <input type="checkbox"/> 3 | id_actividad      |  int(11) |              |           | Sí   | NULL           |             |                |  Cambiar  Eliminar Más |
| <input type="checkbox"/> 4 | fecha_inscripcion | date  |              |           | Sí   | NULL           |             |                |  Cambiar  Eliminar Más |

Imagen 3. Vista de la tabla Inscripciones con llaves primarias y foráneas

## 2. Implementación del Backend:

- Desarrollo de CRUDs para cada tabla (Estudiantes, Actividades, Asistencia y Datos Médicos).
- Uso de consultas preparadas en PDO para proteger contra inyecciones SQL.

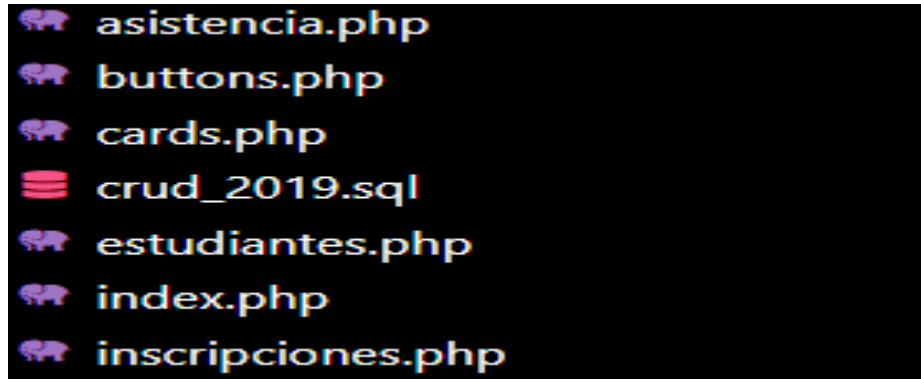


Imagen 4. Vista de archivos de CRUDs

## 3. Diseño del Frontend:

- Creación de interfaces responsivas con Bootstrap.
- Uso de tablas dinámicas para listar datos y formularios modales para agregar y editar información.

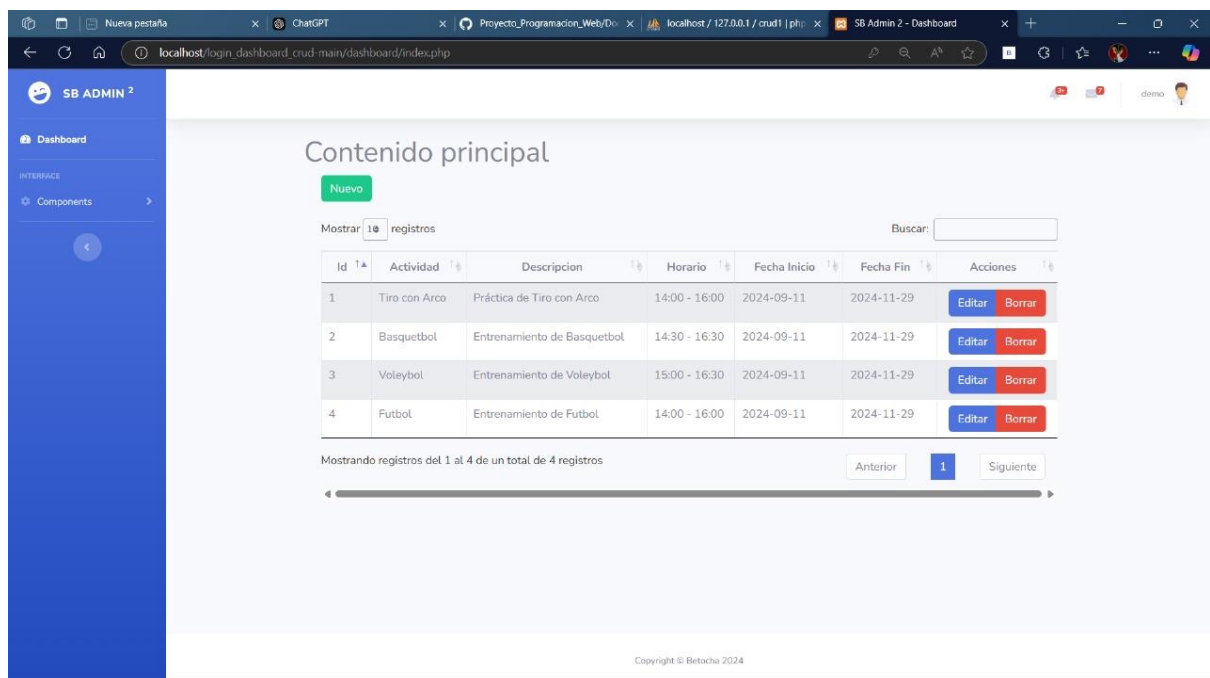


Imagen 5. Tabla de Actividades

#### 4. Seguridad:

- Implementación de autenticación con usuario y contraseña para limitar el acceso al sistema.
- Restricción de operaciones según el rol del usuario (administrador o maestro).



Imagen 6. Login con usuario y contraseña

## 4. Pruebas y Validaciones

### Pruebas Realizadas

#### 1. Pruebas de Funcionalidad:

- CRUDs: Verificación de creación, lectura, edición y eliminación de registros en todas las tablas.
- Reportes: Generación y exportación de reportes en diferentes formatos.

#### 2. Pruebas de Seguridad:

- Simulación de inyecciones SQL para garantizar la protección de la base de datos.
- Validación de accesos no autorizados mediante autenticación y roles.

#### 3. Pruebas de Responsividad:

- Evaluación en dispositivos móviles y de escritorio para asegurar una experiencia de usuario consistente.

#### 4. Pruebas de Rendimiento:



- Evaluación del tiempo de respuesta del sistema con múltiples usuarios concurrentes.

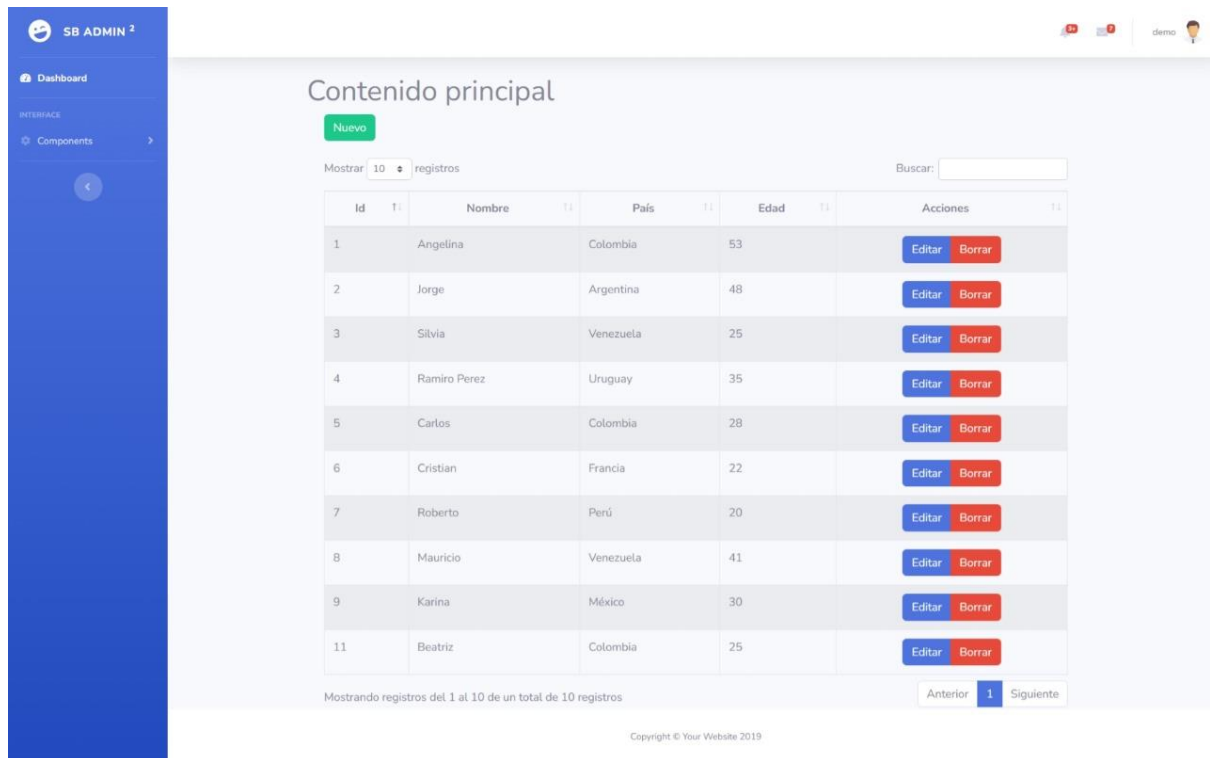


Imagen 7. Vista de la pagina principal

## Resultados de las Pruebas

- **Funcionalidad:** 100% de las operaciones CRUD funcionan correctamente.
- **Seguridad:** El sistema es resistente a ataques básicos gracias al uso de consultas preparadas y validaciones del lado del servidor.
- **Rendimiento:** Las consultas a la base de datos son rápidas, incluso con un volumen significativo de registros.

## 5. Resultados y Conclusión

### Resultados del Proyecto

El proyecto cumplió con los objetivos establecidos, proporcionando un sistema robusto y funcional para la gestión de actividades extraescolares. Las principales ventajas del sistema incluyen:

- **Organización:** Simplificación en el manejo de alumnos, actividades y asistencia.
- **Eficiencia:** Reducción del tiempo dedicado a tareas administrativas manuales.
- **Acceso Seguro:** Protección de datos sensibles mediante autenticación y cifrado.

## Conclusión

El proyecto *Administrador de Paraescolares* es un ejemplo claro de cómo la tecnología puede resolver problemas prácticos en el entorno educativo. Este sistema ofrece una solución integral para la gestión de actividades extraescolares, proporcionando herramientas que mejoran la eficiencia administrativa, reducen los errores humanos y aumentan la seguridad de los datos.

El sistema fue desarrollado siguiendo buenas prácticas, estas incluyen:

- **Separación de Código:** Cada archivo tiene una función clara, como `conexion.php` para la base de datos o `main.js` para operaciones dinámicas.
- **Diseño Modular:** Los módulos CRUD están organizados por tabla, facilitando su actualización o ampliación.
- **Uso de AJAX:** Para enviar y recibir datos del servidor sin recargar la página, mejorando la experiencia del usuario.