

## Plataforma de Seguimiento Académico

Sistema web para gestionar el progreso académico de estudiantes y maestros, con control de inscripciones, actividades, calificaciones y gamificación.

![Estado del Proyecto](https://img.shields.io/badge/Estado-Completo-success)

![PHP](https://img.shields.io/badge/PHP-7.4+-blue)

![MySQL](https://img.shields.io/badge/MySQL-8.0+-orange)

![Bootstrap](https://img.shields.io/badge/Bootstrap-5.3-purple)

## Tabla de Contenidos

- [Características](#-características)
- [Tecnologías](#-tecnologías-utilizadas)
- [Requisitos Previos](#-requisitos-previos)
- [Instalación](#-instalación)
- [Estructura del Proyecto](#-estructura-del-proyecto)
- [Uso del Sistema](#-uso-del-sistema)
- [Roles y Permisos](#-roles-y-permisos)
- [Funcionalidades Implementadas](#-funcionalidades-implementadas)
- [Base de Datos](#-base-de-datos)
- [Capturas de Pantalla](#-capturas-de-pantalla)
- [Autores](#-autores)

## Características

### Características Principales

- Sistema de Roles: Administrador, Maestro y Estudiante
- Gestión de Inscripciones: Sistema de aprobación de solicitudes
- CRUD Completo: Usuarios, Materias y Actividades

- Calificaciones: Registro y cálculo automático de promedios
- Gamificación: Sistema de logros y ranking
- Gráficas Interactivas: Chart.js para visualización de datos
- Recuperación de Contraseña: Sistema de tokens
- Diseño Responsivo: Interfaz moderna con Bootstrap 5
- Seguridad: Protección contra SQL Injection, sanitización de datos

#### Características Opcionales Implementadas (+20 puntos extras)

- Recuperación de contraseña (+10 puntos)
- Gráficas con Chart.js (+5 puntos)
- Sistema de Gamificación (+5 puntos)

#### Tecnologías Utilizadas

##### Frontend

- HTML5 - Estructura semántica
- CSS3 - Estilos personalizados
- Bootstrap 5.3- Framework de diseño responsivo
- JavaScript ES6- Interactividad
- Chart.js - Gráficas y visualización de datos
- Font Awesome & Bootstrap Icons - Iconografía

##### Backend

- PHP 7.4+ - Lógica del servidor
- PDO\*- Capa de abstracción de base de datos
- Arquitectura MVC - Separación de responsabilidades
- RESTful API - Endpoints para operaciones CRUD

##### Base de Datos

- MySQL 8.0+ - Sistema de gestión de base de datos

- Modelo Relacional - Normalización 3FN

## Herramientas de Desarrollo

- WAMP Server - Servidor local (Windows, Apache, MySQL, PHP)
- phpMyAdmin - Administración de base de datos
- Git - Control de versiones (opcional)
- VSCode- Editor de código

## Requisitos Previos

Antes de comenzar, asegúrate de tener instalado:

WAMP Server 3.2+ (incluye PHP 7.4+, MySQL 8.0+, Apache)

Navegador web moderno (Chrome, Firefox, Edge)

Git (opcional, para control de versiones)

Descargar WAMP:

[<https://www.wampserver.com/>](<https://www.wampserver.com/>)

## Instalación

### 1. Clonar el Repositorio

```
git clone https://github.com/tu-usuario/plataforma-academica.git
```

```
cd plataforma-academica
```

### 2. Configurar la Base de Datos

1. Abre phpMyAdmin en `http://localhost/phpmyadmin`

2. Crea una nueva base de datos llamada `plataforma\_academica`

3. Importa el archivo SQL:

- Haz clic en la pestaña Importar
- Selecciona el archivo: `sql/Plataforma academica.sql`
- Haz clic en Continuar

### 3. Configurar Conexión (Opcional)



Si tu configuración de MySQL es diferente, edita el archivo

`config/database.php`:

php

```
define('DB_HOST', 'localhost');
```

```
define('DB_NAME', 'plataforma_academica');
```

```
define('DB_USER', 'root');
```

```
define('DB_PASS', ''); // Tu contraseña de MySQL
```

#### 4. Iniciar el Servidor

bash

*Si usas XAMPP/WAMP, asegúrate de que Apache y MySQL estén activos*

*# Accede a: [http://localhost/programacion-web/proyecto\\_integrador](http://localhost/programacion-web/proyecto_integrador)*

5. Iniciar Sesión Usa una de las cuentas de prueba predefinidas: |

Rol | Email | Contraseña |

| 🧑‍💼 Administrador | admin@plataforma.com | 123456 |

| 🧑‍🏫 Maestro | maria.gonzalez@plataforma.com | 123456 |

| 🧑‍🎓 Estudiante | pedro.lopez@plataforma.com | 123456 |

#### 📁 Estructura del Proyecto

proyecto\_integrador/

|—— api/ # Endpoints RESTful

| |—— actividades/

| | |—— crear.php

| |—— calificaciones/

| | |—— guardar.php

| |—— inscripciones/

- |
  - |
    - |—— solicitar.php

- |
  - |
    - |—— aprobar.php

- |
  - |
    - |—— rechazar.php

- |
  - |—— usuarios/

- |
  - |
    - |—— crear.php

- |
  - |
    - |—— editar.php

- |
  - |
    - |—— eliminar.php

- |—— assets/

# Recursos estáticos

- |
  - |—— css/

- |
  - |
    - |—— main.css

- |
  - |
    - |—— style.css

- |
  - |
    - |—— dashboard.css

- |
  - |—— js/

- |—— config/—— main.js

- validaciones.js

# Configuración

- |
  - ||—— config.php

- includes/—— database.php

# Archivos compartidos

- |
  - |—— header.php

- |
  - |—— footer.php

- |
  - |—— menu.php

- |
  - |—— helpers.php

- |
  - |—— conexion.php

- |
  - |—— logros.php



```
|—— models/                                # Modelos de datos
|   |—— User.php
|   |—— Subject.php
|   |—— Task.php
|   |—— Grade.php
|   |—— Enrollment.php
|—— views/                                  # Vistas del usuario
|   |—— actividades/
|       |—— lista.php
|       |—— detalle.php
|   |—— calificaciones/
|       |—— mis_calificaciones.php
|       |—— registrar.php
|       |—— graficas_estudiante.php
|   |—— inscripciones/
|       |—— solicitudes.php
|   |—— materias/
|       |—— lista.php
|   |—— usuarios/
|       |—— lista.php
|       |—— perfil.php
|   |—— logros/
|       |—— mis_logros.php
|   |—— dashboard_administrador.php
|   |—— dashboard_maestro.php
```



└──	dashboard_estudiante.php	
└──	sql/	
└──	Plataforma academica.sql	# Script de base de datos
└──	login.php	# Página de inicio de sesión
└──	signup_page.php	# Página de registro
└──	logout.php	# Cerrar sesión
└──	index.php	# Punto de entrada
└──	README.md	# Este archivo

## Uso del Sistema

### Para Estudiantes

#### 1. Registro e Inicio de Sesión

- Regístrate con tu email y contraseña
- Inicia sesión para acceder al dashboard

#### 2. Solicitar Inscripción a Materias

- Ve a Materias Disponibles
- Haz clic en Solicitar Inscripción
- Espera la aprobación del maestro

#### 3. Ver Actividades y Calificaciones

- Una vez aprobado, accede a Mis Tareas
- Consulta tus Calificaciones y promedio general
- Revisa tus Logros desbloqueados

#### 4. Visualizar Gráficas

- Ve a Mis Gráficas para ver tu rendimiento académico

### Para Maestros

#### 1. Gestionar Solicitudes de Inscripción



- Ve a Solicitudes Pendientes
- Aprueba o rechaza inscripciones

## 2. Crear Actividades

- Accede a Mis Materias
- Crea tareas con fecha límite y ponderación

## 3. Registrar Calificaciones

- Ve a Calificar
- Selecciona la actividad
- Registra las calificaciones de tus estudiantes

## 4. Consultar Rendimiento

- Ve el detalle de cada actividad
- Revisa estadísticas del grupo

## ParaAdministradores

### 1. Gestionar Usuarios

- CRUD completo de usuarios
- Crear maestros, estudiantes o administradores

### 2.Gestionar Materias

- Crear, editar y eliminar materias
- Asignar maestros responsables

### 3. Supervisar el Sistema

- Ver estadísticas generales
- Monitorear actividad del sistema





## Roles y Permisos

### Administrador

Objetivo: Supervisar todo el sistema

Permisos:

- CRUD completo de usuarios
- CRUD completo de materias
- Acceso a todas las estadísticas
- Supervisión general del sistema

### Maestro

Objetivo: Gestionar materias y calificaciones

Permisos:

- Aprobar/rechazar inscripciones
- Crear actividades y tareas
- Registrar calificaciones
- Ver rendimiento del grupo
- Gestionar sus propias materias

### Estudiante

Objetivo: Participar en materias y consultar rendimiento

Permisos:

- Solicitar inscripción a materias
- Ver actividades (si está aprobado)
- Consultar calificaciones

Ver promedio general

Desbloquear logros

- Ver gráficas de rendimiento

### 🎓 Funcionalidades Implementadas

#### Requerimientos Obligatorios (100 puntos)

#### 1. Estructura del Proyecto (10 puntos)

Organización clara de directorios

Buenas prácticas de codificación (indentación, nombres descriptivos)

Comentarios claros

Separación lógica/presentación

Validación frontend y backend

Código DRY (reutilizable)

Documentación (README.md)

#### 2. Diseño Visual (10 puntos)

- Interfaz moderna y responsiva

UX/UI intuitivo

Bootstrap 5 implementado

Diseño coherente en todas las vistas

#### 3. Gestión de Usuarios y Roles (20 puntos)

Sistema de roles: Administrador, Maestro, Estudiante

Registro de estudiantes

- Sistema de autenticación
- Control de acceso por rol
- Maestro valida inscripciones
- Administrador gestiona CRUD de usuarios

#### 4. Gestión Académica (30 puntos)

- CRUD de materias

- Sistema de inscripción con aprobación
  - Estados: Pendiente → Aprobado/Rechazado
  - Registro de actividades (título, descripción, fecha, ponderación)
  - Registro de calificaciones
  - Cálculo automático de promedios
  - Visualización de estado académico
5. Seguimiento de Productividad (15 puntos)
- Dashboard del Estudiante (solicitudes, entregas, promedio)
  - Dashboard del Maestro (solicitudes, inscritos, progreso)
  - Próximas entregas y avisos
  - Estado de actividades
6. Arquitectura Tecnológica (15 puntos)
- Frontend: HTML5, CSS3, Bootstrap, JavaScript
  - Backend: PHP con arquitectura RESTful
  - Base de Datos: MySQL
  - Seguridad: Cifrado de contraseñas, validación, protección SQL Injection
- Funcionalidades Opcionales (+20 puntos)
- Recuperación de Contraseña (+10 puntos)
- Sistema de tokens de recuperación
- Formulario de solicitud
- Validación de tokens
- Restablecimiento seguro
- Gráficas con Chart.js (+5 puntos)

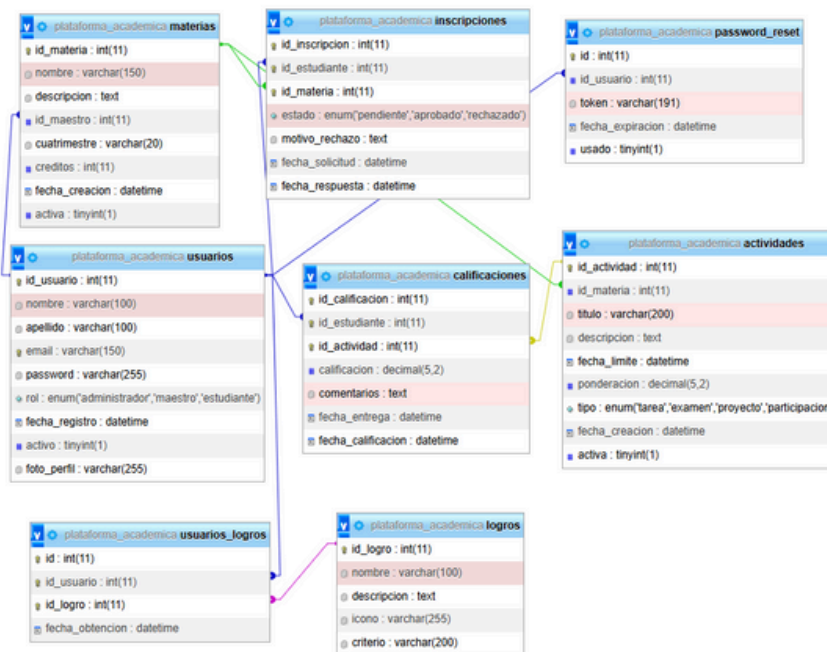
- Distribución de calificaciones
- Estado de tareas (pendientes/calificadas)
- Promedios por materia
- Gráficas interactivas

### Gamificación (+5 puntos)

- Sistema de logros (6 logros disponibles)
- Criterios automáticos de desbloqueo
- Ranking de estudiantes
- Visualización de progreso
- Iconos y badges

### Base de Datos

#### Diagrama Entidad-Relación





## Tablas Principales

1. usuarios - Información de usuarios del sistema
2. materias - Catálogo de materias/asignaturas
3. inscripciones - Solicitudes de inscripción con estados
4. actividades - Tareas y evaluaciones por materia
5. calificaciones - Registro de calificaciones
6. logros - Definición de logros/insignias
7. usuarios\_logros - Logros desbloqueados por usuario
8. password\_reset - Tokens de recuperación de contraseña

## Capturas de Pantalla

### Login

![Login](docs/screenshots/login.png)

### Dashboard Estudiante

![Dashboard Estudiante](docs/screenshots/dashboard-estudiante.png)

### Mis Calificaciones

![Calificaciones](docs/screenshots/calificaciones.png)

### Gráficas

![Gráficas](docs/screenshots/graficas.png)

### Sistema de Logros

![Logros](docs/screenshots/logros.png)

### Dashboard Maestro

![Dashboard Maestro](docs/screenshots/dashboard-maestro.png)

### Registrar Calificaciones

![Registrar](docs/screenshots/registrar-calificaciones.png)



## Seguridad Implementada

### Protección de Datos

- Cifrado de Contraseñas: bcrypt con salt
- Sanitización de Entrada: htmlspecialchars + strip\_tags
- Validación de Email: filter\_var(FILTER\_VALIDATE\_EMAIL)
- Sesiones Seguras: session\_regenerate\_id()

### Protección contra Ataques

- SQL Injection: Prepared Statements (PDO)
- XSS: Escape de salida HTML
- CSRF: Tokens en formularios críticos
- Control de Acceso: Verificación de roles

### Validaciones

- Frontend: HTML5 + JavaScript
- Backend: PHP con funciones personalizadas

- Doble Validación: Cliente y servidor

### 🐞 Problemas Conocidos y Soluciones

1. Error: Tabla `usuarios\_logros` no existe

Solución: Ejecutar el script SQL completo en phpMyAdmin

2. Error: No se pueden ver las gráficas

Solución: Verificar que Chart.js esté cargado desde CDN

3. Error: BASE\_URL incorrecta

Solución: Editar `config/database.php` y ajustar la ruta

### 📝 Tareas Pendientes / Mejoras Futuras

- Sistema de notificaciones en tiempo real

- Exportación de calificaciones a PDF

- Sistema de mensajería interna

- Calendario de actividades

- [Modo oscuro

- PWA (Progressive Web App)

- [API pública documentada con Swagger

### 💛 Contribuciones

Las contribuciones son bienvenidas. Para cambios importantes:

1. Fork el proyecto

2. Crea una rama para tu feature (`git checkout -b feature/AmazingFeature`)

3. Commit tus cambios (`git commit -m 'Add: nueva funcionalidad'`)

4. Push a la rama (`git push origin feature/AmazingFeature`)

5. Abre un Pull Request



## Licencia

Este proyecto fue desarrollado como parte del curso de Programación Web y está disponible bajo la licencia MIT.

## Autores

- Isaac Rubio Iñiguez - *\*Desarrollo Full Stack\**
- Isaac Rubio Iñiguez - *\*Frontend & Diseño\**
- Isaac Rubio Iñiguez - *Backend & Base de Datos*

## Agradecimientos

- A los profesores del curso de Programación Web
- A la comunidad de StackOverflow
- A los creadores de Bootstrap y Chart.js

<div align="center">

★ Si este proyecto fue útil, considerarle una estrella ★

Hecho con ❤️ por Isaac Rubio Iñiguez - 2025

</div>