



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas



Ingeniería Telemática

* Ingeniería WEB *

Proyecto: Página Web

Alumnos:

Bernabé Báez Fernanda Itzel
Sánchez Zaragoza Diego

Profesor:

Polanco Montelongo Francisco Antonio

Grupo: 2TM4

Nuestra idea para la creación de nuestro proyecto, fue crear una página web para gestionar una cineteca, donde distintos usuarios puedan realizar distintas operaciones como por ejemplo, crear un perfil en la página, observar los distintos productos que se ofrecen, entre otras cosas. Este proyecto se realizó haciendo uso principalmente del lenguaje PHP.

1. Análisis de requerimientos

A continuación se mostrarán los requerimientos funcionales (RF) y los no funcionales (RNF) en formato IEEE 830, con respecto al proyecto.

1.1 Requerimientos funcionales (RF)

RF01. El sistema debe permitir a los usuarios registrarse en el evento proporcionando nombre, correo y contraseña.

RF02. El sistema debe enviar un correo de confirmación al usuario tras registrarse exitosamente.

RF03. El sistema debe permitir a los usuarios autenticarse mediante su correo y contraseña.

RF04. El sistema debe mostrar un panel de administración para usuarios autenticados con permisos de administrador.

RF05. El sistema debe permitir crear, leer, actualizar y eliminar (CRUD) los siguientes recursos:

- Usuarios
- Eventos
- Categorías
- Días y Horarios

RF06. El sistema debe permitir registrar usuarios a eventos de películas específicos.

RF07. El sistema debe gestionar el número de accesos que estén permitidos por tipo de paquete que tenga el usuario (ej. Básico, Premium).

RF08. El sistema debe generar un código QR único para cada registro confirmado.

RF09. El sistema debe permitir a los usuarios ver el calendario de eventos y filtrar por día y categoría.

1.2 Requerimientos no funcionales (RNF)

RNF01: Seguridad.

- El sistema debe almacenar las contraseñas de forma cifrada (bcrypt o equivalente).
- Las sesiones deben estar protegidas contra ataques de sesión (uso de cookies seguras).
- Se deben validar las entradas de usuario para evitar inyecciones SQL o XSS.

- El sistema debe restringir funcionalidades según el rol del usuario (autenticación y autorización).
- Debe usarse HTTPS en producción.

RNF02: Rendimiento.

- El tiempo de carga inicial del sistema no debe superar los 3 segundos en condiciones normales.
- Debe optimizarse el acceso a base de datos mediante paginación y consultas eficientes.

RNF03: Mantenibilidad.

- El sistema debe seguir una estructura modular y reutilizable.
- Deben separarse claramente los modelos, controladores y vistas.

RNF04: Escalabilidad.

- El sistema debe permitir la adición de nuevos módulos sin alterar los ya existentes.

RNF05: Usabilidad.

- El sistema debe tener una interfaz responsiva y clara para usuarios y administradores.

RNF06: Comunicación Segura (correo)

- El sistema debe utilizar **Mailtrap** en el entorno de desarrollo para interceptar y revisar correos sin enviarlos a usuarios reales.
- En producción, el sistema deberá utilizar un proveedor SMTP seguro (no definido aquí, pero Mailtrap debe cambiarse en deploy).

RNF07: Pagos Simulados

- El sistema debe usar **PayPal Developer (Sandbox)** para simular procesos de compra sin transacciones reales.
- La validación del pago es necesaria para activar ciertos accesos (por ejemplo, tipo de paquete).

2. Diseño

2.1 Arquitectura del diseño.

Presentación (Frontend): HTML, CSS, JS



Controlador (Router.php, includes/)



Modelo (models/, classes/)

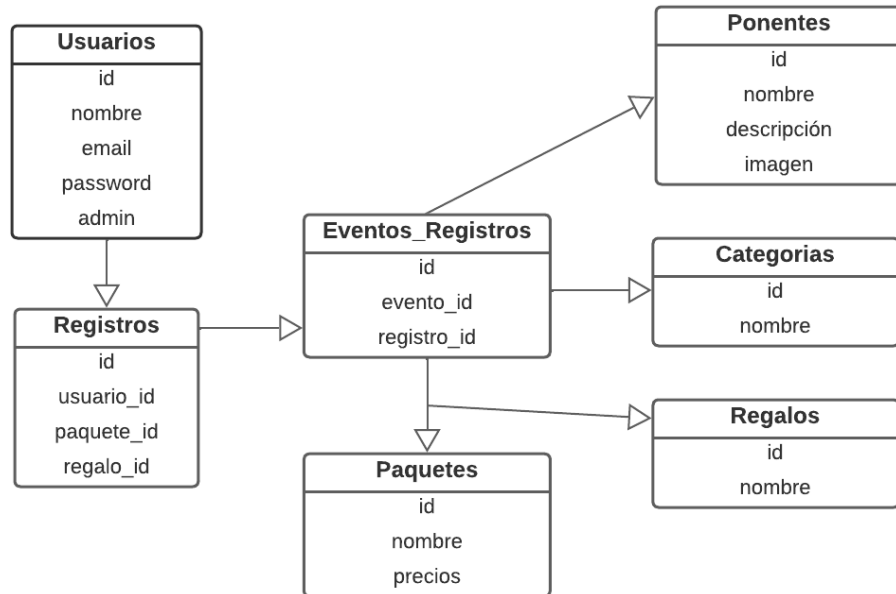


Base de datos (MySQL)

- **Modelos:** Manejan lógica de negocio y acceso a datos (usuario, evento, etc.).

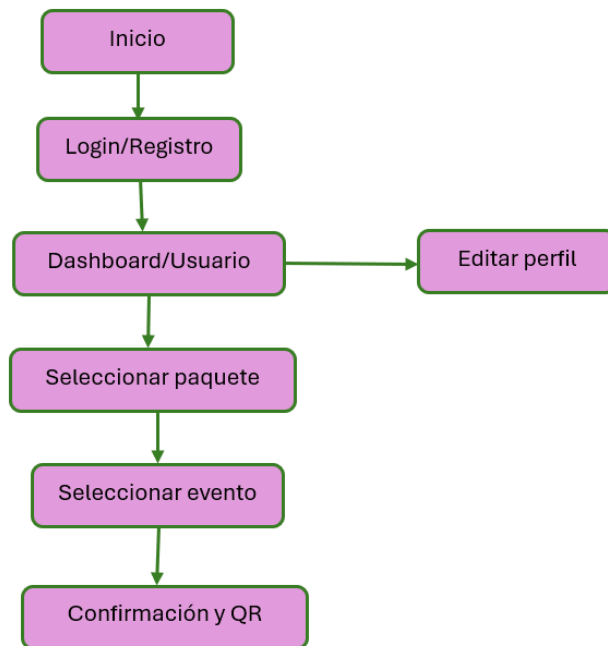
- **Vistas:** HTML creado, junto con los distintos estilos y scripts necesarios.
- **Controlador/Enrutador:** **Router.php** y scripts en **includes/** que reciben peticiones y llaman al modelo correspondiente.

2.2 Modelado UML

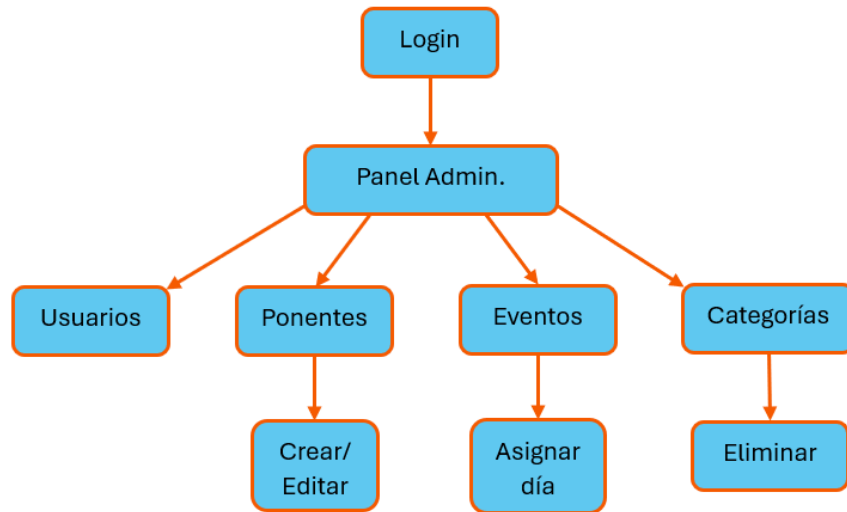


2.3 Maquetado y diagrama de navegación

- Usuario



- Administrador



2.4 Modelo de datos

Resumen de entidades:

- **usuarios:** id, nombre, email, password, admin
- **eventos:** id, nombre, categoria_id, ponente_id, dia_id, hora_id
- **ponentes:** id, nombre, descripcion, imagen, redes sociales
- **paquetes:** id, nombre, precio, beneficios
- **regalos:** id, nombre
- **registros:** id, usuario_id, paquete_id, regalo_id
- **eventos_registros:** evento_id, registro_id

3. Construcción

3.1 Frameworks utilizados

Backend:

- PHP con arquitectura MVC personalizada.
- Composer para manejo de dependencias y autoloader.
- Mailtrap para pruebas de envío de correos en entorno de desarrollo.
- PayPal Developer (Sandbox) para pruebas de integración de pagos.

Frontend:

- HTML5, SASS, JS Vanilla.
- Gulp para tareas automatizadas (compilación de estilos, etc.).

Base de Datos:

- MySQL (basado en estructura de modelos).