

# **PROYECTO INTEGRADOR 2º DAW**



## **APLICACIÓN WEB PARA EL FESTIVAL DE CINE DE LA UNIVERSIDAD EUROPEA DE MADRID**

### **Autores:**

Julián Cruz Alonso

Herminia Edoko Sioto

Diego García Abril

### **Tutor:**

Raquel Cerdá Losa

Irene del Rincón Bello

Sara Villanueva Rosa

### **Repositorio del Proyecto:**

<https://github.com/HerminiaEdoko/FestivalCineUEM.git>

### **Ciclo Formativo:**

Desarrollo de Aplicaciones Web (DAW)

### **Centro:**

Universidad Europea de Madrid

Centro Profesional Europeo de Madrid

# ÍNDICE

1. INTRODUCCIÓN	2
2. PALABRAS CLAVE	2
3. MÓDULOS FORMATIVOS APLICADOS	3
4. HERRAMIENTAS Y LENGUAJES UTILIZADOS	7
5. FASES DEL PROYECTO	8
5.1. Modelo de Datos Utilizado	9
5.2. Análisis Funcional	12
5.3. Planificación del Desarrollo	15
5.4. Estudio de Mercado	15
5.5. Análisis de la aplicación: especificación de todos los requisitos y funcionalidades que realizará la aplicación	16
5.6. Diseño de las interfaces: sistema de diseño, mockups, prototipo interactivo	16
6. COMPONENTES DEL EQUIPO Y APORTACIÓN REALIZADA POR CADA ESTUDIANTE	16
7. PRUEBAS Y RESULTADOS	16
8. CONCLUSIONES	16
9. BIBLIOGRAFÍA	16
10. ANEXOS	16

## **1. INTRODUCCIÓN**

El Festival de Cine de la Universidad Europea de Madrid es un evento anual que promueve el talento audiovisual entre estudiantes y alumnis. Este proyecto consiste en desarrollar una aplicación web completa para gestionar todas las fases del festival: desde la difusión de información y bases del concurso, hasta la inscripción de candidaturas, revisión por parte de los organizadores y la publicación de los ganadores.

### **Objetivo del proyecto:**

Crear una plataforma web que centralice la gestión del Festival de Cine, facilitando:

- La participación de estudiantes y alumnis mediante una inscripción online.
- La administración eficiente del evento por parte de los organizadores.
- El acceso público a información relevante del festival.

### **Alcance:**

El proyecto integra desarrollo frontend (HTML, CSS, JavaScript) y backend (PHP, MySQL), implementando tres niveles de acceso diferenciados: público general, participante y organizador.

## 2. PALABRAS CLAVE

API Rest, CRUD, PHP, UX/UI, MySQL, DOM, Frontend, Backend, JavaScript, Fetch API, PDO, Validación formularios, Responsive design, RGPD, Normalización, MVC, Programación asíncrona.

Término	Definición
<b>API REST</b>	Interfaz de programación que utiliza HTTP para la comunicación cliente-servidor.
<b>AJAX/Fetch</b>	Técnica para realizar peticiones asíncronas al servidor sin recargar la página.
<b>CRUD</b>	Create, Read, Update, Delete - Operaciones básicas de gestión de datos.
<b>DOM</b>	Document Object Model - Representación estructurada del HTML.
<b>Frontend</b>	Parte visual de la aplicación con la que interactúa el usuario.
<b>Backend</b>	Parte del servidor que procesa la lógica y gestiona datos.
<b>PDO</b>	PHP Data Objects: extensión para el acceso seguro a bases de datos.
<b>Consultas Preparadas</b>	Técnica de seguridad que previene inyección SQL.
<b>UX</b>	User Experience: diseño centrado en la experiencia del usuario.
<b>Responsive</b>	Diseño web adaptable a diferentes tamaños de pantalla.
<b>RGPD</b>	Reglamento General de Protección de Datos.
<b>MVC</b>	Modelo Vista Controlador: Patrón de arquitectura de software que separa la lógica de negocio, la presentación y el control.

### **3. MÓDULOS FORMATIVOS APLICADOS**

Este proyecto integra conocimientos adquiridos a lo largo del ciclo formativo de Desarrollo de Aplicaciones Web, aplicando competencias de diversos módulos tanto de segundo como de primer curso.

#### **Diseño de Interfaces Web**

El módulo de Diseño de Interfaces Web ha sido fundamental para la creación de una experiencia de usuario coherente y accesible. Se han aplicado principios de UX/UI en el diseño de todas las pantallas del sistema, asegurando que la navegación sea intuitiva para cada tipo de usuario. Se han creado mockups y prototipos interactivos que han permitido validar el diseño antes de la implementación. El diseño responsive garantiza que la aplicación sea completamente funcional en dispositivos móviles, tablets y ordenadores de escritorio, adaptándose automáticamente al tamaño de pantalla. Se ha desarrollado un sistema de diseño coherente con paleta de colores, tipografía y componentes reutilizables que mantienen la identidad visual del festival a lo largo de toda la aplicación.

#### **Desarrollo Web en Entorno Cliente**

En este módulo se han aplicado técnicas avanzadas de JavaScript para crear una interfaz dinámica e interactiva. Se han implementado validaciones de formularios en tiempo real que proporcionan feedback inmediato al usuario, mejorando significativamente la experiencia de uso. La manipulación del DOM permite mostrar y ocultar contenido de forma dinámica sin necesidad de recargar la página completa. Se ha utilizado la Fetch API para establecer comunicación asíncrona con el servidor, lo que permite enviar y recibir datos en formato JSON sin interrumpir la navegación del usuario. La gestión de eventos del usuario y el control del flujo de la aplicación del lado del cliente son aspectos clave que se han desarrollado siguiendo las mejores prácticas de programación.

## **Desarrollo Web en Entorno Servidor**

El desarrollo del backend se ha realizado aplicando los conocimientos del módulo de Desarrollo Web en Entorno Servidor. Se ha implementado una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC), que separa claramente las responsabilidades del sistema. Los controladores reciben las peticiones HTTP del cliente y coordinan las respuestas, los modelos encapsulan la lógica de negocio y gestionan el acceso a la base de datos mediante PDO, y las vistas generan las respuestas en formato JSON que se envían al cliente. Esta separación facilita el mantenimiento del código y permite que múltiples desarrolladores trabajen en paralelo sin interferencias.

Se ha desarrollado una API REST con PHP que procesa todas las operaciones CRUD del sistema, siguiendo los principios de arquitectura REST para garantizar un diseño escalable y mantenable. La gestión de sesiones permite mantener el estado de autenticación de los usuarios de forma segura entre peticiones. El procesamiento y validación de formularios en el servidor constituye una capa adicional de seguridad, complementando las validaciones del cliente. Se han implementado medidas de seguridad robustas, incluyendo el cifrado de contraseñas mediante el algoritmo bcrypt y el uso de consultas preparadas con PDO para prevenir inyecciones SQL.

## **Lenguajes de Marcas**

Los conocimientos de este módulo se han aplicado en la estructuración semántica del contenido mediante HTML5, utilizando las etiquetas apropiadas para mejorar la accesibilidad y el posicionamiento. El intercambio de datos entre cliente y servidor se realiza mediante JSON, formato que se ha trabajado extensamente en el módulo. La validación de formatos de datos y la correcta estructuración de la información son aspectos fundamentales que se han implementado siguiendo los estándares aprendidos.

## **Bases de Datos**

El módulo de Bases de Datos ha sido esencial para el diseño del modelo de datos del sistema. Se ha realizado un análisis exhaustivo de los requisitos de información, plasmado en un diagrama Entidad-Relación que identifica todas las entidades y sus relaciones. El modelo ha sido normalizado hasta la Tercera Forma Normal para eliminar redundancias y garantizar la integridad de los datos. Se han implementado restricciones de integridad referencial mediante claves foráneas, y se han optimizado las consultas mediante la creación de índices estratégicos. El diseño contempla el cumplimiento de la normativa RGPD, permitiendo la eliminación selectiva de datos personales cuando finaliza cada edición del festival.

## **Entornos de Desarrollo**

Los conocimientos de Entornos de Desarrollo se han aplicado en la configuración y gestión del entorno de trabajo. Se ha utilizado Git como sistema de control de versiones, permitiendo el trabajo colaborativo y el seguimiento de cambios en el código fuente. El repositorio del proyecto está alojado en GitHub (<https://github.com/HerminiaEdoko/FestivalCineUEM.git>), donde se gestiona el versionado mediante commits descriptivos y ramas para el desarrollo de nuevas funcionalidades. La documentación del código sigue estándares de buenas prácticas, facilitando el mantenimiento futuro del proyecto. Se han aplicado técnicas de depuración para la detección y corrección de errores tanto en el cliente como en el servidor.

## **4. HERRAMIENTAS Y LENGUAJES UTILIZADOS**

### **Lenguajes de Programación**

- **HTML5** - Estructura de las páginas web.
- **CSS3** - Diseño visual y responsive.
- **JavaScript** - Lógica del cliente, validaciones, Fetch API.
- **PHP** - Lógica del servidor.
- **SQL (MySQL)** - Gestión de base de datos.

### **Entorno de Desarrollo**

- **Visual Studio Code** - Editor de código.
- **XAMPP** - Servidor local (Apache + MySQL + PHP).
- **Git / GitHub** - Control de versiones.

### **Herramientas de Diseño**

- **Figma** - Mockups y prototipos.
- **Draw.io / Canva** - Diagramas E-R y UML.

### **Herramientas de Planificación**

- **Trello** - Gestión de tareas y sprints.
- **Canva** - Diagrama de Gantt.

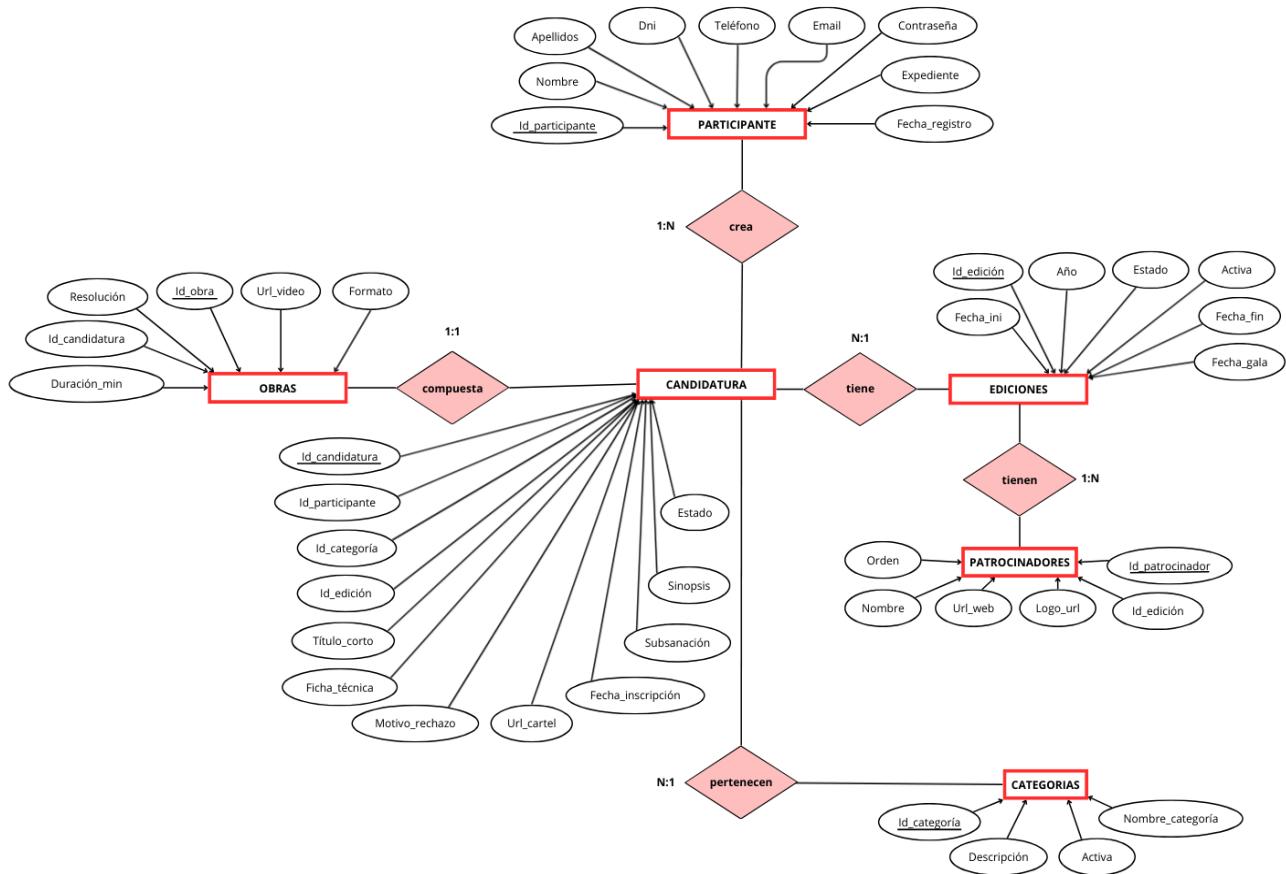
### **Navegadores**

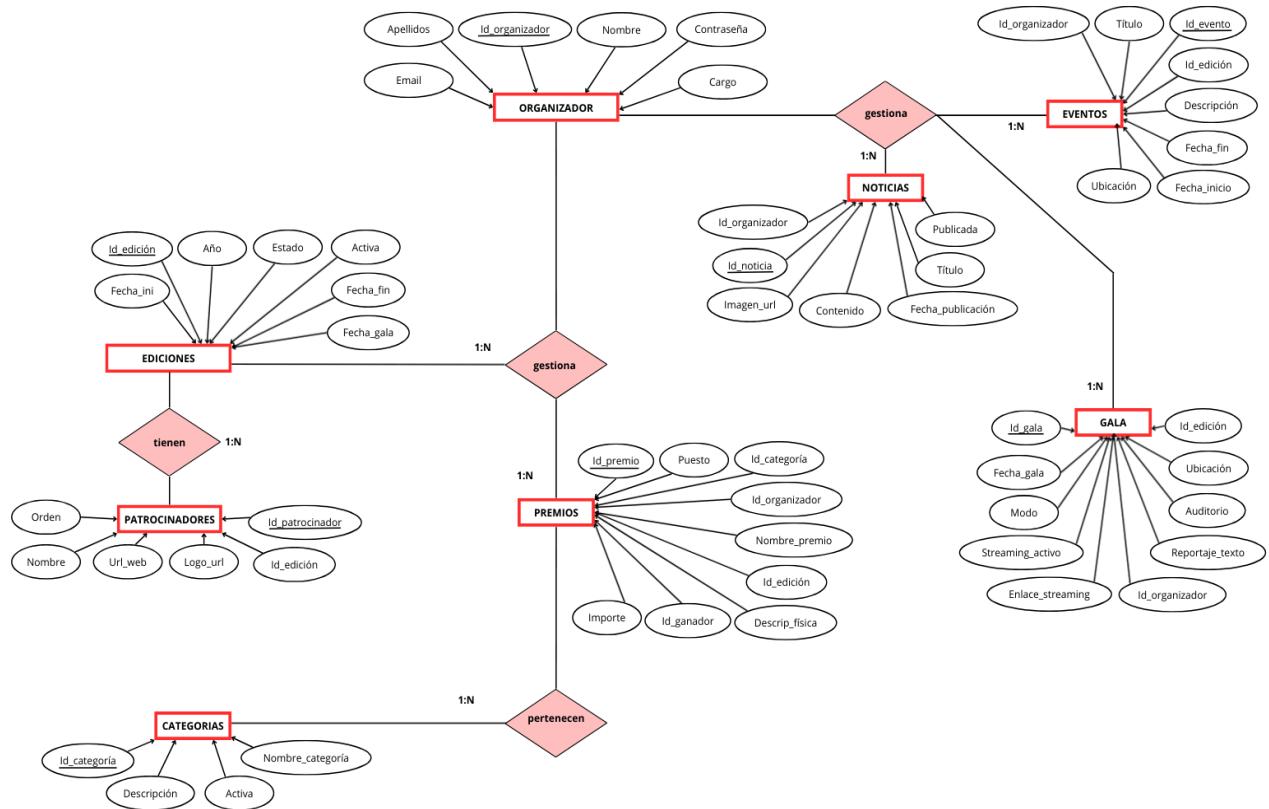
- Chrome, Firefox, Safari y Edge.

## 5. FASES DEL PROYECTO

### 5.1. Modelo de Datos Utilizado

#### Diagrama Entidad-Relación





La estructura del sistema se organiza en torno a ediciones, que representan cada año del festival. Los organizadores crean ediciones y gestionan todo su contenido: noticias, eventos, premios y gala. Los participantes se registran en una edición específica y presentan candidaturas, cada una compuesta por una obra (cortometraje). Las candidaturas se clasifican en categorías (Alumnos, Alumni o Carrera Profesional) que determinan los premios disponibles.

Esta estructura permite eliminar datos por edición al finalizar el festival, cumpliendo con RGPD mientras se conservan estadísticas y cortometrajes ganadores.

## **Modelo Relacional**

A partir del diagrama E-R se ha desarrollado el siguiente modelo relacional normalizado:

**ORGANIZADOR:** (ID\_organizador\*, Nombre, Apellidos, Email, Contraseña, Cargo)

- PK: ID\_organizador

**EDICIONES:** (ID\_edición\*, ID\_organizador\*, Nombre, Año, Estado, Fecha\_inicio, Fecha\_fin, Fecha\_creación)

- PK: ID\_edición
- FK: ID\_organizador → ORGANIZADOR

**PARTICIPANTES:** (ID\_participante\*, ID\_edición\*, Nombre, Apellidos, Email, Contraseña, Teléfono, Dni, Expediente, Fecha\_registro)

- PK: ID\_participante
- FK: ID\_edición → EDICIONES

**CATEGORÍAS:** (ID\_categoria\*, Nombre, Descripción)

- PK: ID\_categoria

**CANDIDATURAS:** (ID\_candidatura\*, ID\_participante\*, ID\_edición\*, ID\_categoria\*, Estado, Motivo\_rechazo, Fecha\_creacion, Fecha\_revisión, Fecha\_modificación)

- PK: ID\_candidatura
- FK: ID\_participante, ID\_edición, ID\_categoria

**OBRAS:** (ID\_obra\*, ID\_candidatura\*, Título, Sinopsis, Ficha\_técnica, Cartel, Vídeo, Duración)

- PK: ID\_obra
- FK: ID\_candidatura (relación 1:1)

**NOTICIAS:** (ID\_noticia\*, ID\_organizador\*, ID\_edición\*, Título, Contenido, Fecha\_publicacion, Visible)

- FK: ID\_organizador, ID\_edición

**EVENTOS:** (ID\_evento\*, ID\_organizador\*, ID\_edición\*, Título, Descripción, Fecha, Hora, Lugar)

- FK: ID\_organizador, ID\_edición

**GALA:** (ID\_gala\*, ID\_edición\*, Fecha, Hora, Lugar, Tipo, Descripción)

- FK: ID\_edición

**PATROCINADORES:** (ID\_patrocinador\*, ID\_edición\*, Nombre, Logo, URL)

- FK: ID\_edición

**PREMIOS:** (ID\_premio\*, ID\_edición\*, ID\_categoria\*, ID obra\*, Nombre, Descripción)

- FK: ID\_edición, ID\_categoria, ID obra

## Normalización

El modelo alcanza la Tercera Forma Normal (3FN):

- **1FN:** Todos los atributos son atómicos y cada tabla cuenta con clave primaria.
- **2FN:** Se han eliminado dependencias parciales mediante la separación de candidatura y obras.
- **3FN:** Se han eliminado dependencias transitivas utilizando referencias mediante claves foráneas.

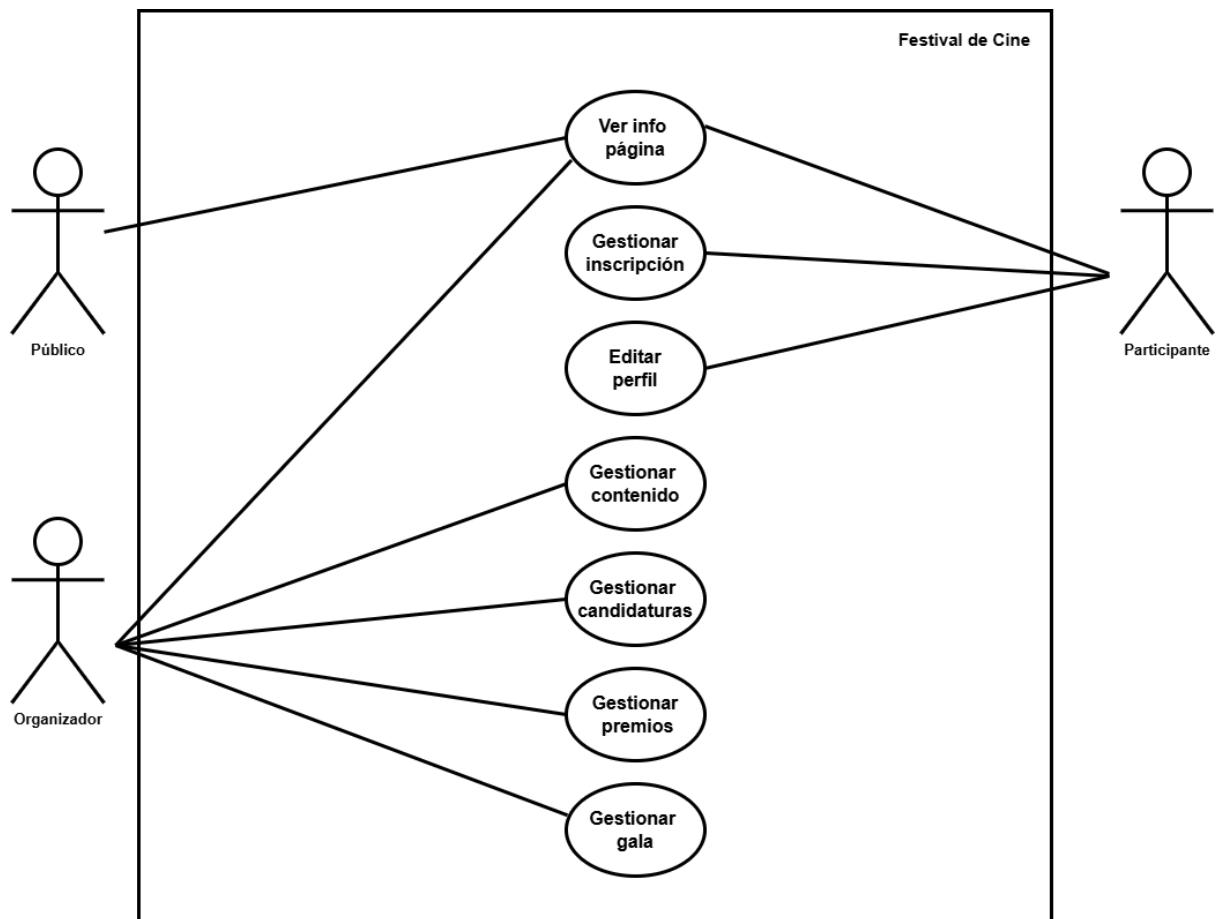
En cuanto a la gestión de ediciones: se ha optado por mantener todas las ediciones en una única tabla, diferenciándolas por su estado (activa o cerrada), lo que facilita consultas históricas y la limpieza de datos conforme al reglamento general de protección de datos.

Además, los participantes, candidaturas, eventos y premios se vinculan a una edición específica, permitiendo la eliminación en cascada cuando finaliza el festival.

Y por último, se ha optado por la separación de las tablas candidaturas y obras. Aunque mantienen una relación 1:1, la separación optimiza las consultas administrativas sin necesidad de cargar archivos multimedia pesados.

## 5.2. Análisis Funcional

### Diagrama de Casos de Uso



### **Actores identificados y funcionalidades:**

El sistema contempla tres tipos de actores con niveles de acceso diferenciados.

- **Público General**: Representan a cualquier visitante que accede a la web del festival sin necesidad de registro. Pueden consultar información pública del festival (noticias, premios, eventos, gala) y tienen la opción de registrarse como participante.

- **Participantes:** Corresponde a estudiantes actuales o alumnis que desean presentar su cortometraje. Tras registrarse, pueden inscribir una candidatura con el cortometraje, sinopsis, ficha técnica y cartel.

El sistema les permite consultar el estado de su candidatura (en revisión, aceptada o rechazada) y, en caso de rechazo, tienen la opción de subsanar explicando las correcciones realizadas. Pueden editar su perfil, aunque el archivo del cortometraje no puede modificarse una vez subido.

- **Organizador:** Dispone de control total sobre el sistema. Gestiona todo el contenido público (noticias, eventos, patrocinadores, gala), revisa las candidaturas y decide su aceptación o rechazo con justificación. Define los premios por categoría, asigna ganadores y configura la gala en dos modos: pre-evento (con el programa detallado) y post-evento (con el reportaje fotográfico y audiovisual).

## [APARTADOS PARA ENTREGAS FUTURAS]

### **5.3. Planificación del Desarrollo (\*\*\*\*\*HACER TAMBIÉN DIAGRAMA DE GANTT)**

El proyecto se ha desarrollado siguiendo la metodología ágil Scrum, que permite entregas incrementales de valor y adaptación continua a los requisitos. Se han organizado el trabajo en sprints de dos semanas, cada uno culminando con una entrega funcional evaluable.

La planificación se estructura en cuatro entregas principales:

**Sprint 1 - Entrega 1 (4 de enero):** Se ha realizado el análisis y diseño de la base de datos, creando el diagrama Entidad-Relación y el modelo relacional normalizado. Se ha implementado el script de creación automática de la base de datos con datos de prueba. Se ha desarrollado el sistema de login funcional con cifrado de contraseñas y gestión de sesiones. Se ha elaborado la memoria inicial del proyecto con los apartados de portada, introducción, palabras clave, módulos formativos y herramientas utilizadas.

**Sprint 2 - Entrega 2 (19 de enero):** Se desarrollará la sección completa de organizadores, implementando las operaciones CRUD para noticias, eventos, premios y patrocinadores. Se configurará el sistema de gestión de la gala en sus dos modos (pre-evento y post-evento). Se implementará la funcionalidad de revisión de candidaturas con los estados de aceptación y rechazo. Se actualizará la memoria con los apartados correspondientes al desarrollo realizado.

**Sprint 3 - Entrega 3 (26 de enero):** Se implementará la sección completa de participantes, incluyendo el formulario de inscripción de candidaturas con validación de archivos multimedia. Se desarrollará el sistema de consulta de estado de candidatura y el mecanismo de subsanación para candidaturas rechazadas. Se creará el calendario funcional que mostrará los eventos del festival. Se completará la gestión de candidaturas desde el panel de organizadores.

**Sprint 4 - Entrega 4 (8 de febrero):** Se realizarán pruebas funcionales exhaustivas de todos los puntos de entrada del sistema, verificando la correcta comunicación entre cliente y servidor. Se corregirán todos los errores detectados y se optimizará el rendimiento de la aplicación. Se completará la documentación final del proyecto, incluyendo conclusiones, bibliografía y anexos con los manuales técnico y de usuario.

El seguimiento del proyecto se realiza mediante un tablero Kanban digital que permite visualizar el estado de cada tarea, identificar bloqueos y mantener al equipo sincronizado en todo momento. El código fuente del proyecto está disponible en el repositorio público de GitHub: <https://github.com/HerminiaEdoko/FestivalCineUEM.git>

#### **5.4. Estudio de Mercado**

Se ha realizado un análisis de plataformas similares como FilmFreeway y Festhome, identificando que la solución desarrollada aporta valor diferencial mediante la personalización completa para el contexto universitario, la integración con el sistema de validación mediante número de expediente académico, y el sistema de subsanación de candidaturas rechazadas. A diferencia de las plataformas comerciales, esta solución no tiene costes de licencia y garantiza el cumplimiento de RGPD mediante la gestión por ediciones, permitiendo la eliminación controlada de datos personales al finalizar cada año del festival.

#### **5.5. Análisis de la aplicación: especificación de todos los requisitos y funcionalidades que realizará la aplicación**

#### **5.6. Diseño de las interfaces: sistema de diseño, mockups, prototipo interactivo**

### **6. COMPONENTES DEL EQUIPO Y APORTACIÓN REALIZADA POR CADA ESTUDIANTE**

### **7. PRUEBAS Y RESULTADOS**

### **8. CONCLUSIONES**

### **9. BIBLIOGRAFÍA**

### **10. ANEXOS**