

#### TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

#### Departamento de Informática

PROYECTO

### GESTIÓN DE UNA ASOCIACIÓN DE ROL

## Manual Técnico

Autor/es: MANUEL JESÚS DONOSO PÉREZ

Curso Académico: 2024 / 2025

Índice

Contenido

[GESTIÓN DE UNA ASOCIACIÓN DE ROL 1](#_Toc198574986)

[Manual Técnico 1](#_Toc198574987)

[Introducción 3](#_Toc198574988)

[Arquitectura de la aplicación 3](#_Toc198574989)

[Frontend 3](#_Toc198574990)

[Tecnologías usadas 3](#_Toc198574991)

[Entorno de desarrollo 3](#_Toc198574992)

[Backend 3](#_Toc198574993)

[Tecnologías usadas 3](#_Toc198574994)

[Entorno de desarrollo 4](#_Toc198574995)

[Estructura de la Aplicación 4](#_Toc198574996)

[Estructura del Proyecto 5](#_Toc198574997)

[Documentación técnica 6](#_Toc198574998)

[Análisis 6](#_Toc198574999)

[Diagramas de Casos de Usos: 6](#_Toc198575000)

[Diagrama de clases 25](#_Toc198575001)

[Diagrama de entidad-relación 26](#_Toc198575002)

[Desarrollo 26](#_Toc198575003)

[Pruebas realizadas 30](#_Toc198575004)

[Proceso de despliegue 33](#_Toc198575005)

[Propuestas de mejoras 35](#_Toc198575006)

[Bibliografía 35](#_Toc198575007)

Introducción

El presente manual técnico forma parte del proyecto desarrollado por Manuel J. Donoso como culminación del módulo de Desarrollo de Aplicaciones Web (DAW). Este documento tiene como objetivo proporcionar una guía detallada sobre la instalación, configuración, estructura y funcionamiento del sistema web implementado, así como facilitar su mantenimiento y futura evolución.

El proyecto se centra en la creación de una plataforma web diseñada para la gestión de una asociación de rol, permitiendo la interacción de distintos tipos de usuarios (administradores, Moderadores y a Jugadores) a través de funcionalidades específicas según su perfil. Incluye además un sistema de foros, Sistema de alquiler, chat, y eventos.

Para acceder a una descripción completa del proyecto, incluyendo diagramas, funcionalidades, tecnologías empleadas y documentación técnica adicional, se puede consultar el siguiente enlace:  
👉 <https://github.com/ManuelJDonoso/Proyecto_daw/wiki>

Asimismo, tanto el proyecto como su repositorio en GitHub están disponibles a través de la página web oficial del autor:  
🌐 [**www.manueldonoso.es**](https://www.manueldonoso.es)

Este manual está dirigido principalmente a desarrolladores, administradores de sistemas y profesionales interesados en desplegar, analizar o extender las funcionalidades del proyecto.

Arquitectura de la aplicación

Frontend

Tecnologías usadas

**HTML5:** Utilizado para la estructura del contenido de la aplicación

**CSS3**: Para el diseño y estilización de la interfaz de usuario, asegurado una presentación visual tractiva y adaptable

**JavaScript**: Lenguaje de programación utilizado para la lógica del frontend, permitiendo la interactividad en la interface

Entorno de desarrollo

**Editor de código:** Visual Studio Code 1.98.1

**Servidor web:** Apache/2.4.62(Debian)  
**Sistema operativo:** Debian 12

**Herramientas adicionales**: Extensiones de VSCode para HTML, CSS, JavaScript, así como herramientas de depuración en navegadores

Backend

Tecnologías usadas

**Php 8.2.26:** Lenguaje de programación para lógica del servidor y la gestión de peticiones.

**MariaDB 10.11.6**: Sistema de gestión de bases de datos relacional, utilizado para el almacenamiento de datos.

**Apache/2.4.62:** Servidor web encargado de gestionar las peticiones HTTP

Entorno de desarrollo

**Sistema operativo:** Debian 12

**Servidor web:** Apache/2.4.62

**Base de datos:** MariaDB 10.11.6

**Entorno de desarrollo:** Visual Studio Code 1.98.1 con Extensiones para PHP y base de datos

Estructura de la Aplicación

La Estructura de la aplicación se organiza de la siguiente Manera

Estructura del Proyecto

**A continuación, se detalla la estructura formal del proyecto, incluyendo las carpetas y archivos principales, así como su propósito y contenido:**

**Documentación**

**Documentación del Proyecto: Esta carpeta contiene toda la documentación relacionada con el proyecto, incluyendo especificaciones técnicas, manuales de usuario, diagramas y cualquier otro material de referencia.**

* **Src**

**Proyecto: Esta carpeta contiene el código fuente principal del proyecto, incluyendo todos los archivos y directorios necesarios para el desarrollo y funcionamiento del sistema.**

* **Sql**

**Script para la Creación de la Base de Datos y Datos por Defecto: Esta carpeta contiene los scripts SQL necesarios para la creación de la base de datos y la inserción de datos predeterminados. Estos scripts son esenciales para la inicialización y configuración inicial del sistema.**

* **www**

**Carpeta que se Sube al Servidor: Esta carpeta contiene todos los archivos y recursos que deben ser subidos al servidor para el despliegue de la aplicación web. Incluye archivos HTML, CSS, JavaScript y otros recursos estáticos.**

* **Assets**

**Componentes No Específicos del Programa: Esta carpeta contiene recursos generales que no son específicos del programa, como imágenes, música, etc.**

* **Images: Imágenes utilizadas en el diseño de la aplicación.**
* **Uploads: Imágenes subidas por los usuarios a través de la interfaz de la aplicación.**
* **Config**

**Fichero de Configuración para el Acceso a la Base de Datos: Esta carpeta contiene los archivos de configuración necesarios para el acceso a la base de datos. Estos archivos incluyen credenciales, parámetros de conexión y otras configuraciones relevantes.**

* **Css**

**Fichero de Estilo para la Aplicación: Esta carpeta contiene los archivos CSS que definen el estilo visual de la aplicación web. Incluyen estilos para la interfaz de usuario, diseño de páginas y componentes visuales.**

* **Controllers**

**Elementos del Backend que Ejecutan Acciones: Esta carpeta contiene los controladores del backend, que son responsables de manejar las solicitudes del usuario, interactuar con el modelo y devolver las respuestas adecuadas.**

* **Js**

**Ficheros JavaScript que Le Dan Funcionalidad a la Aplicación Web: Esta carpeta contiene los archivos JavaScript necesarios para proporcionar funcionalidad interactiva a la aplicación web. Estos archivos pueden incluir scripts para la validación de formularios, manipulación del DOM, interacción con el backend, etc.**

* **Models**

**Modelo de Clases para el Backend: Esta carpeta contiene los modelos del backend, que representan las estructuras de datos y las lógicas de negocio del sistema. Estos modelos interactúan directamente con la base de datos y proporcionan datos a los controladores.**

* **Views**

**Vistas del Programa: Esta carpeta contiene los archivos de vista del programa, que son responsables de renderizar la interfaz de usuario. Estos archivos pueden estar escritos en HTML, CSS y JavaScript, y son utilizados por los controladores para generar la salida que se muestra al usuario.**

* **Descripción General**

**La estructura del proyecto ha sido diseñada para facilitar la organización y el mantenimiento del código, asegurando que todos los componentes y recursos estén correctamente categorizados y accesibles. Cada carpeta tiene un propósito específico, lo que permite una separación clara de responsabilidades y una mejor gestión del desarrollo y despliegue del sistema.**

Documentación técnica

Análisis

Diagramas de Casos de Usos:

3.1.1.1 Casos General  
  
Diagrama de Casos de Uso – *Crónicas de Mérida*

El siguiente diagrama de casos de uso representa las funcionalidades principales del sistema "Crónicas de Mérida", una plataforma destinada a la gestión de eventos, materiales, foros y usuarios. El sistema contempla distintos tipos de actores, cada uno con permisos y responsabilidades diferenciadas. La dirección del diagrama es de izquierda a derecha para facilitar su lectura y está organizado en paquetes que agrupan los casos de uso por funcionalidad.

Actores del sistema:

Visitante: Usuario no registrado o sin autenticación. Puede realizar acciones básicas como visualizar contenido público.

Jugador: Usuario registrado en el sistema. Hereda las funcionalidades del Visitante y añade nuevas relacionadas con la participación activa.

Moderador: Usuario con permisos adicionales sobre la gestión del foro y el seguimiento de incidencias. Hereda las funcionalidades del Jugador.

Administrador: Usuario con control total del sistema. Hereda las funcionalidades del Moderador.

Paquetes funcionales:

1. Eventos

Agrupa las funcionalidades relacionadas con los eventos disponibles en la plataforma:

*UC1:* Ver eventos disponibles

*UC2:* Ver detalles del evento

*UC3:* Inscribirse al evento

2. Material

Contiene los casos de uso para la gestión del material físico (libros, juegos, etc.):

*UC4:* Ver material disponible

*UC5:* Préstamo de material

*UC6:* Devolución de material

*UC7:* Alta de material

*UC8:* Baja de material

3. Foro

Abarca las acciones relacionadas con la interacción entre usuarios mediante el foro:

*UC9:* Ver foro

*UC10:* Crear categoría

*UC11:* Crear tema

*UC12:* Crear mensaje

*UC13:* Responder mensaje

*UC14:* Eliminar mensaje

*UC15:* Eliminar tema

*UC16:* Cerrar / Abrir tema

*UC17:* Cerrar / Abrir categoría

*UC26:* Dar "Me gusta" a mensajes

4. Sistema

Incluye los casos de uso administrativos y de gestión de usuarios e incidencias:

*UC18:* Registrar usuario

*UC19:* Modificar usuario

*UC20:* Eliminar usuario

*UC21:* Modificar rol

*UC22:* Dar de baja usuario

*UC23:* Iniciar incidencia

*UC24:* Ver incidencias

*UC25:* Modificar incidencia

Asignación de funcionalidades por actor:

Visitante:

Ver eventos disponibles (UC1)

Ver detalles del evento (UC2)

Ver material disponible (UC4)

Ver foro (UC9)

Registrar usuario (UC18)

Jugador *(además de las del Visitante)*:

Inscribirse al evento (UC3)

Préstamo de material (UC5)

Crear tema (UC11)

Crear mensaje (UC12)

Responder mensaje (UC13)

Dar de baja usuario (UC22)

Iniciar incidencia (UC23)

Dar "Me gusta" (UC26)

Moderador *(además de las del Jugador)*:

Devolución de material (UC6)

Eliminar mensaje (UC14)

Eliminar tema (UC15)

Cerrar / Abrir tema (UC16)

Cerrar / Abrir categoría (UC17)

Ver incidencias (UC24)

Modificar incidencia (UC25)

Administrador *(además de las del Moderador)*:

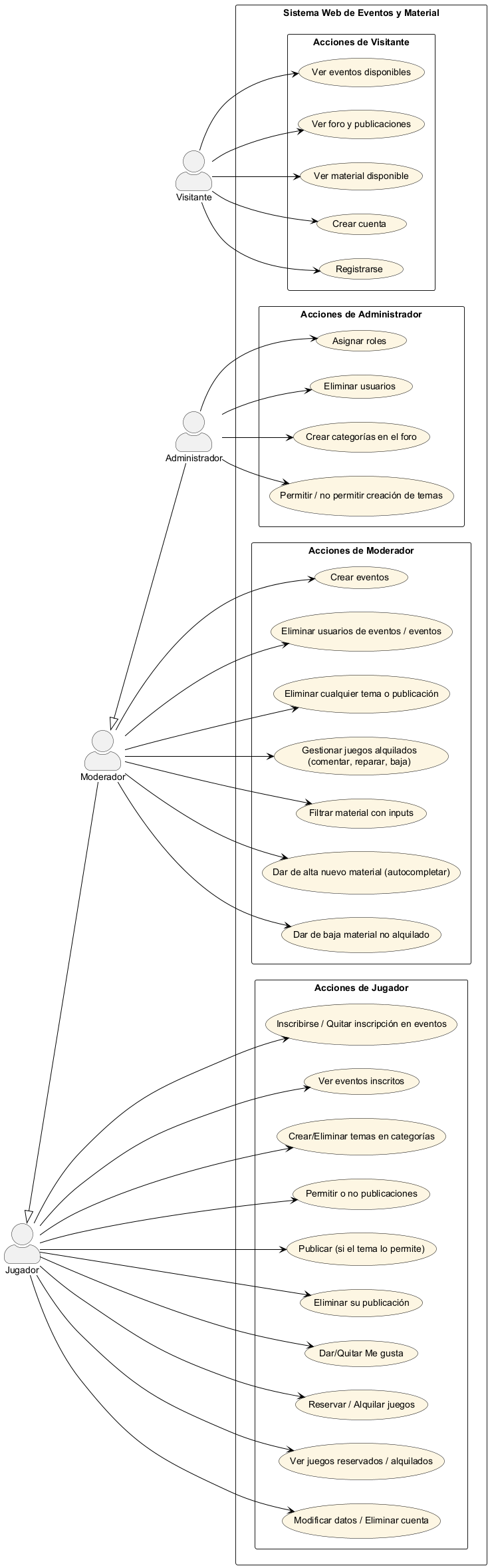
Alta de material (UC7)

Baja de material (UC8)

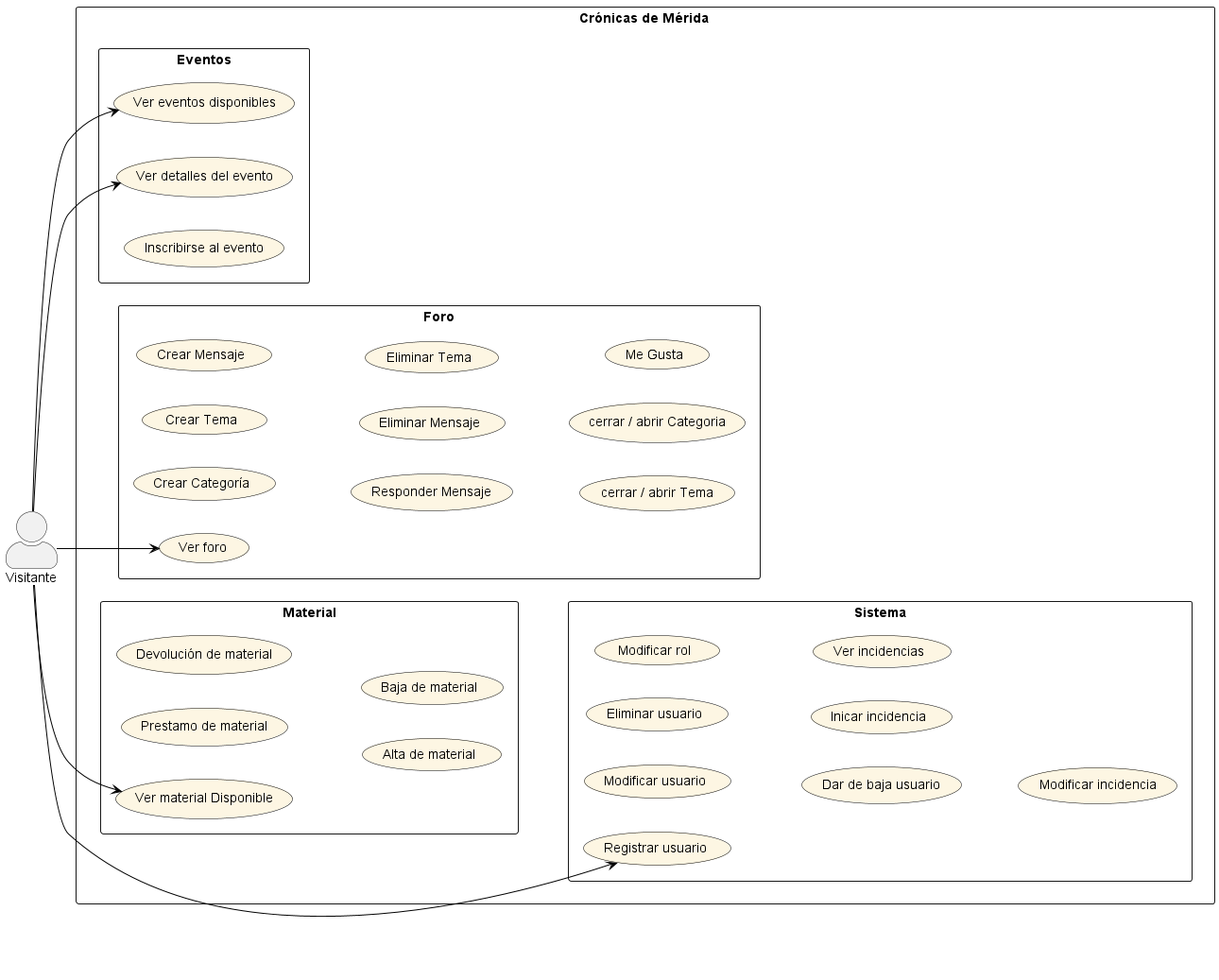
Crear categoría (UC10)

Eliminar usuario (UC20)

Modificar rol (UC21)

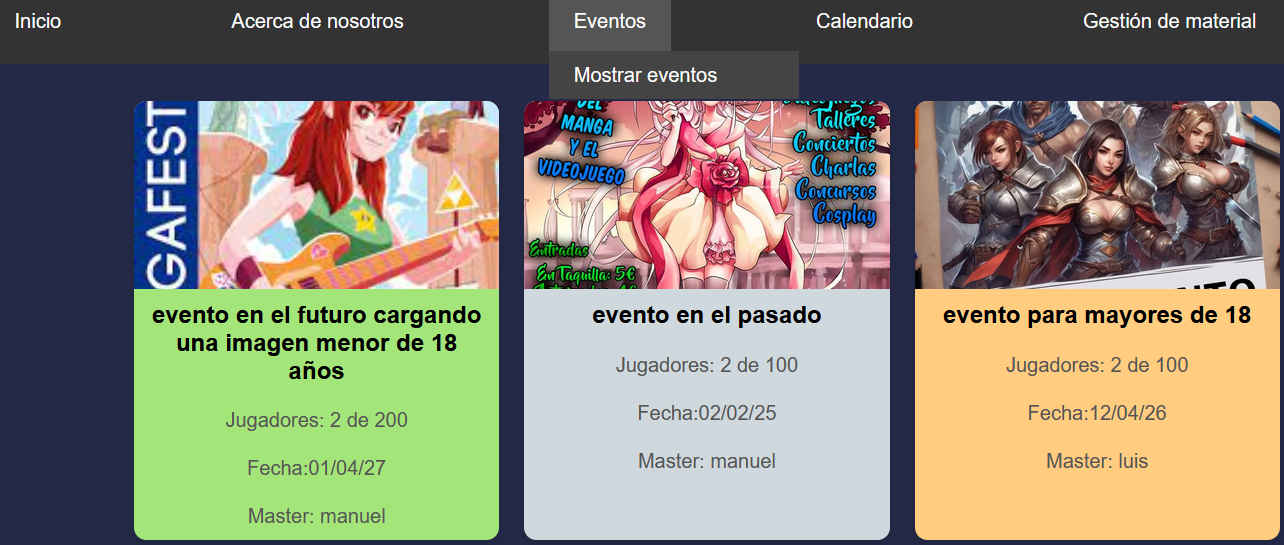
**  
Para mayor facilidad adjunto la imagen por caso:**

**Actores del sistema:**

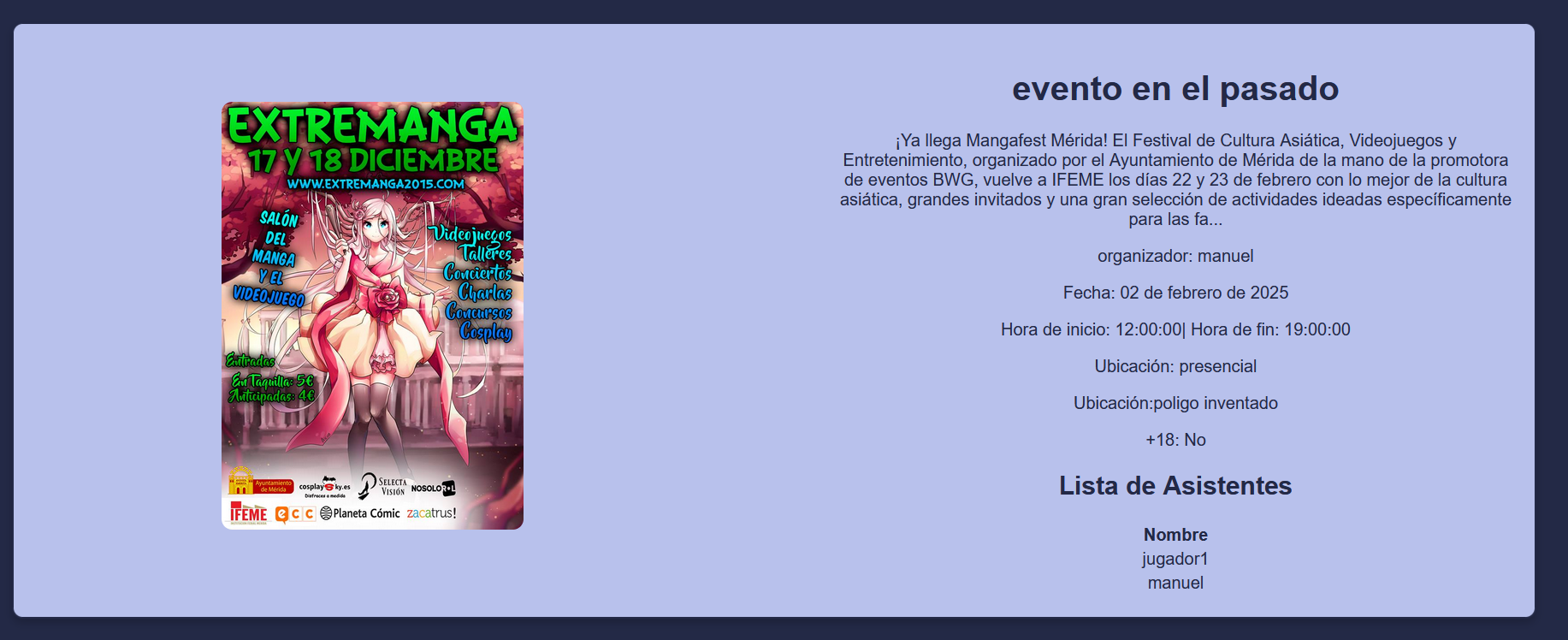
**Visitante: **

**Solo tiene permiso de lectura sobre los eventos:**

**los eventos disponibles:**

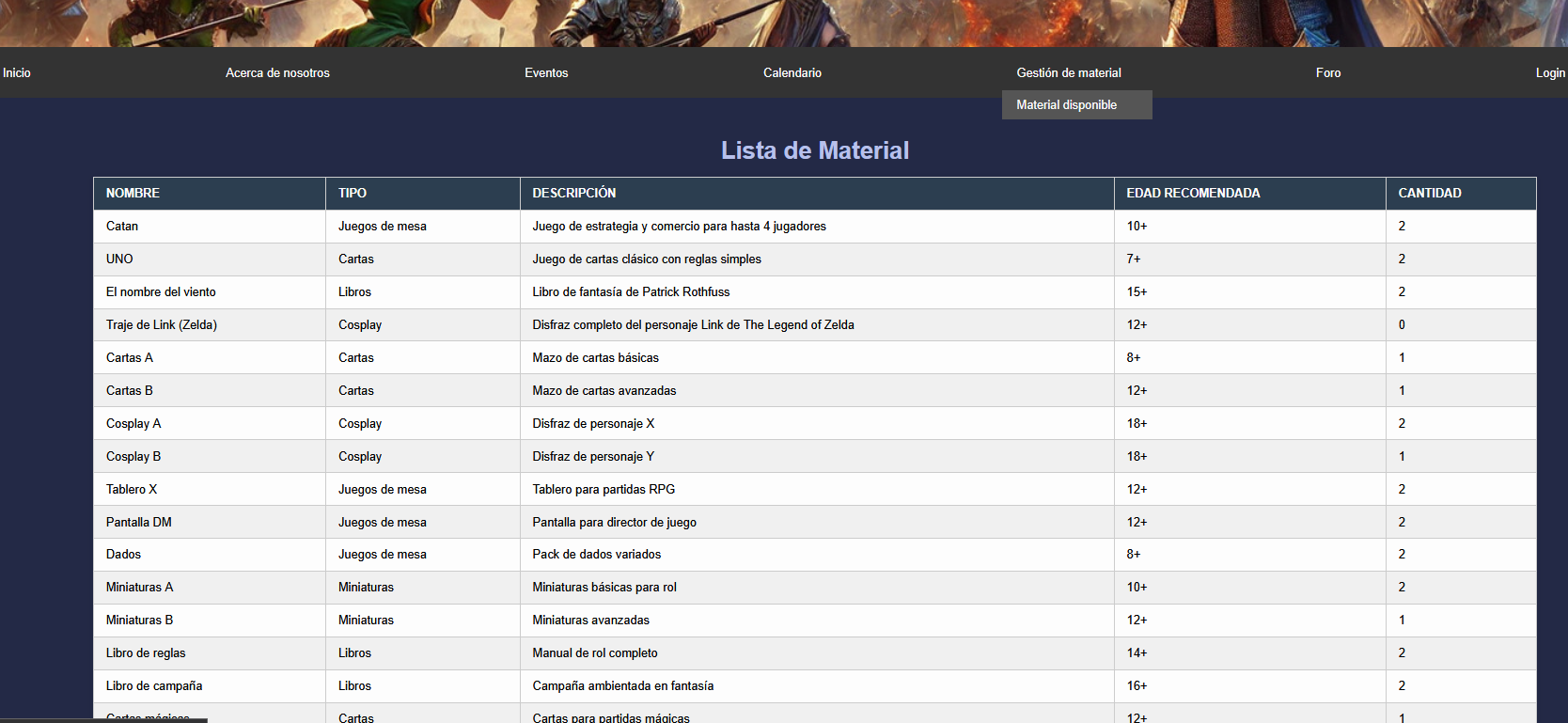


**El evento:**

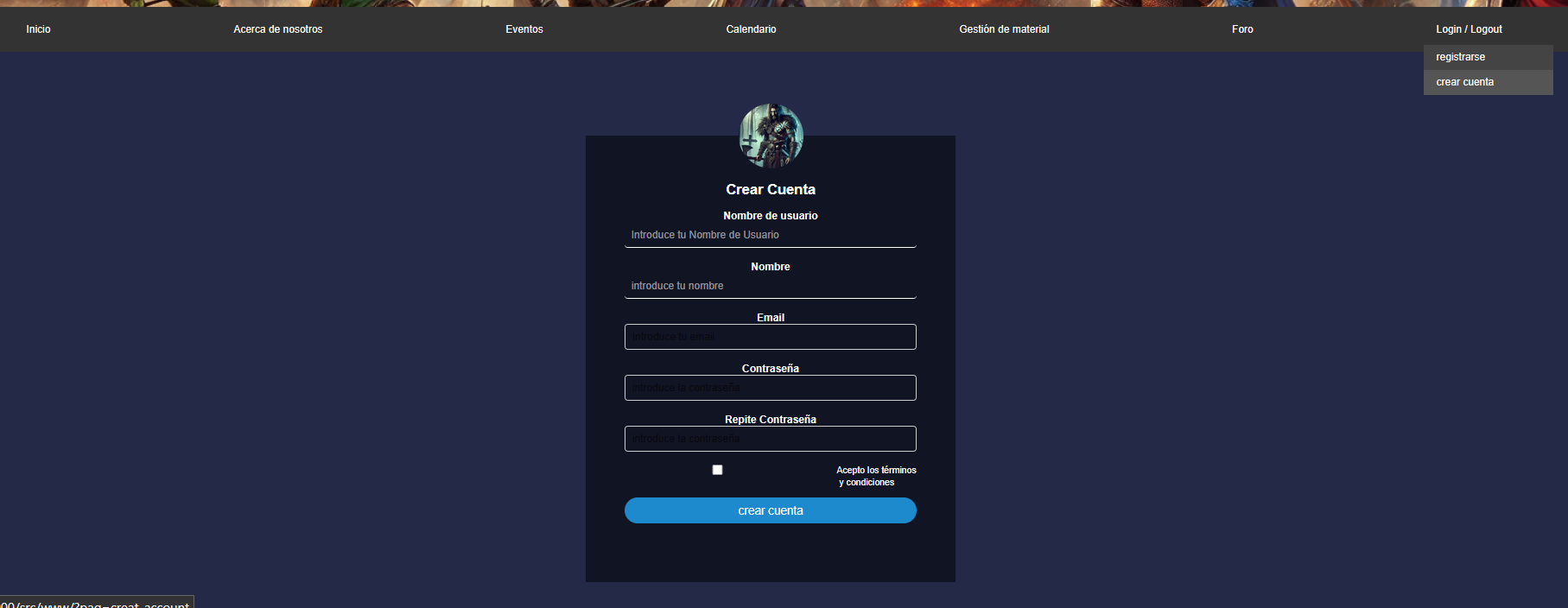


**Puede ver el foro y sus publicaciones:**

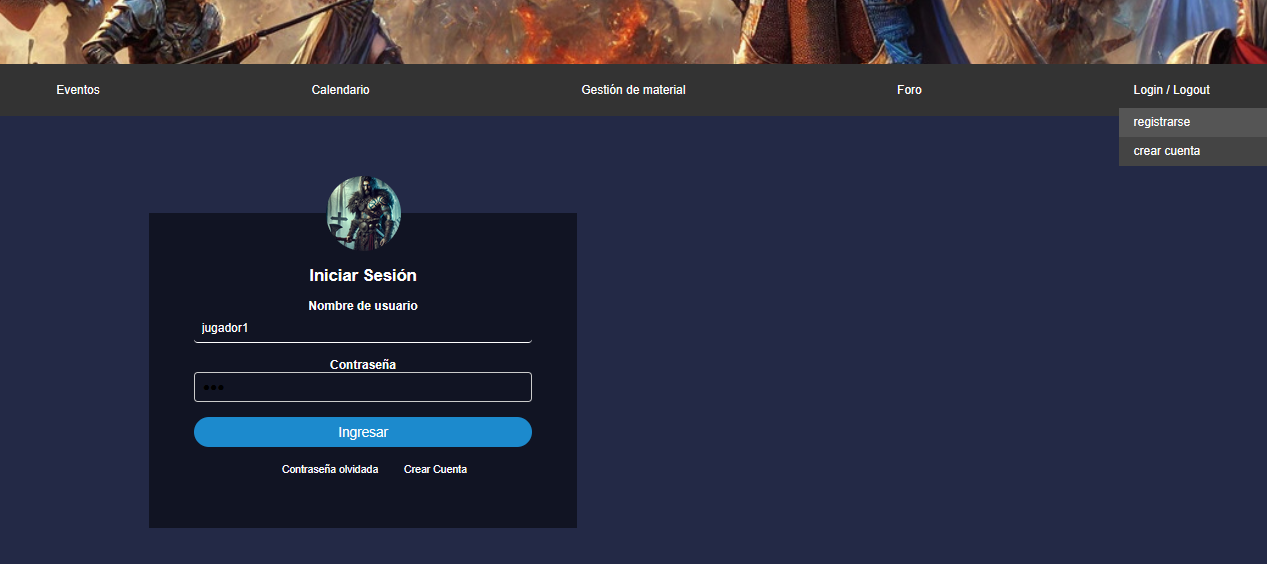


**Puede ver el material disponible para alquilar.**

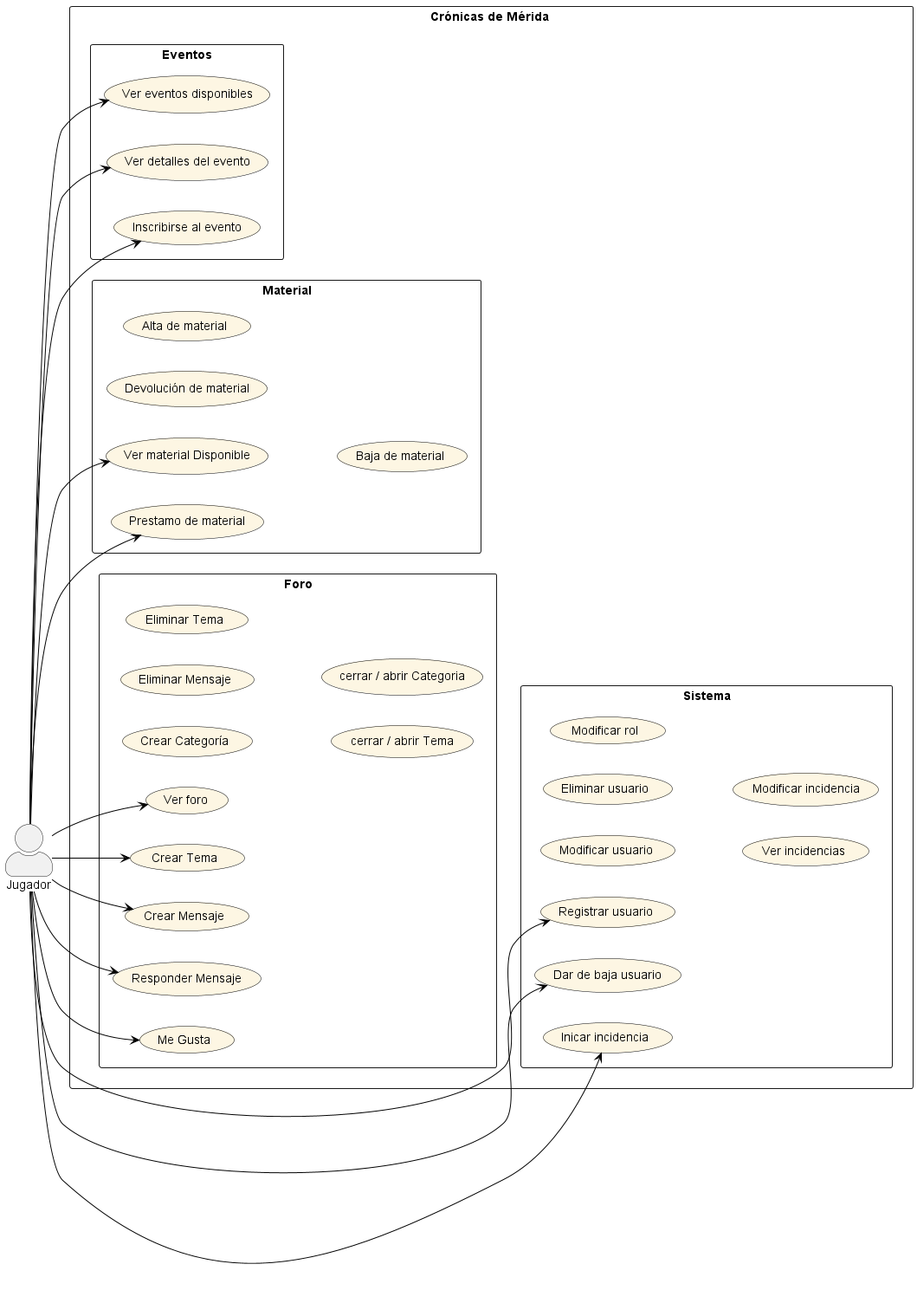
**Puede crear una cuenta**



**Puede registrarse:**



**Jugador:**

****

**Inscribirse en eventos o quitarse de los eventos suscritos.**



**Ver los eventos en lo que está inscrito**

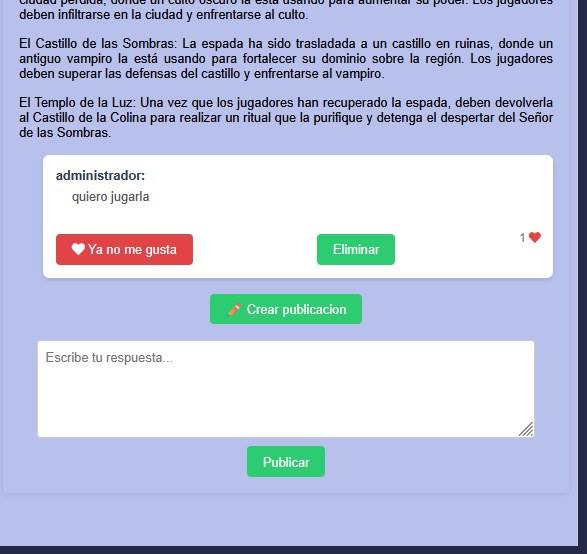


**Puede crear en las categorías temas, eliminar los temas que has creado, permitir o no publicaciones.**

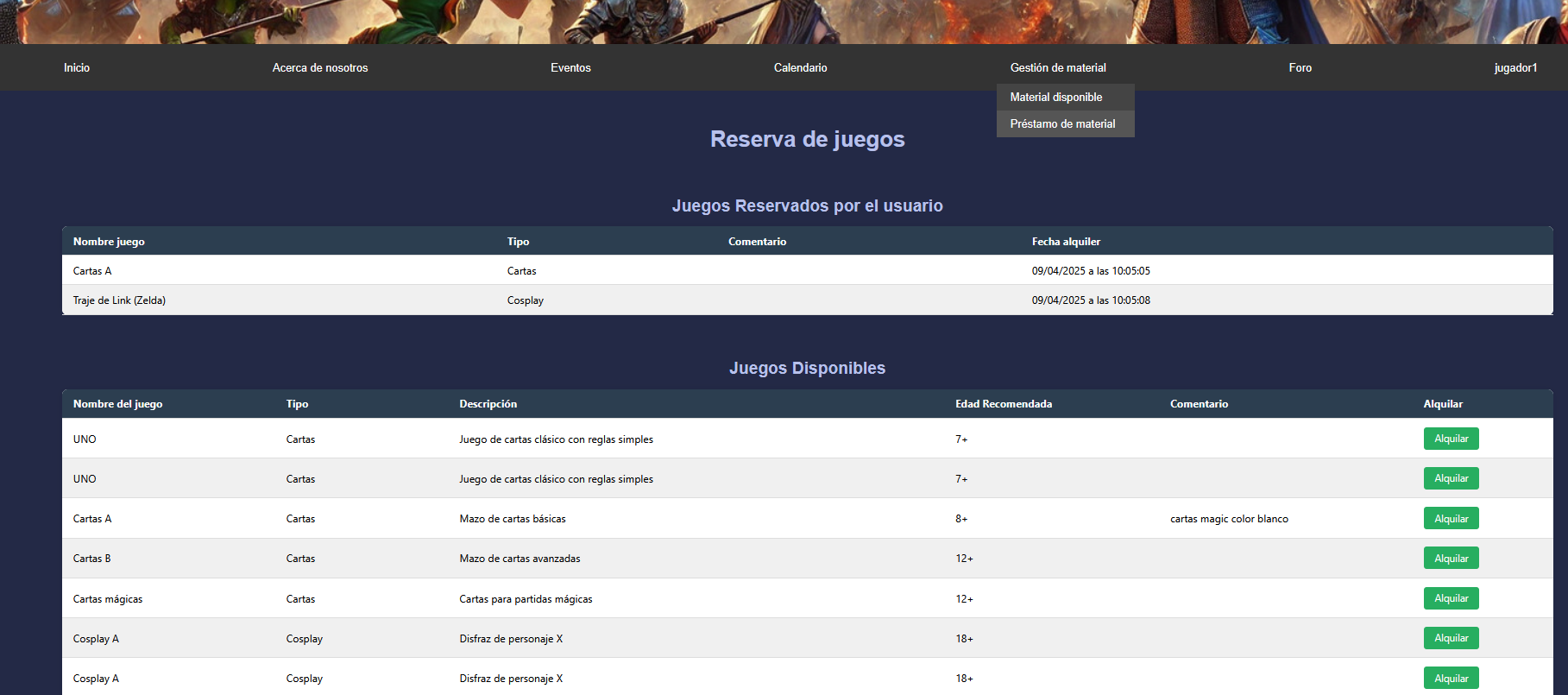


**Puede publicar, si lo permite el tema, eliminar su publicación, además puede dar me gusta a las publicaciones. Solo se permite una por publicación y usuario, si se pulsa una segunda vez el “me gusta” se elimina**

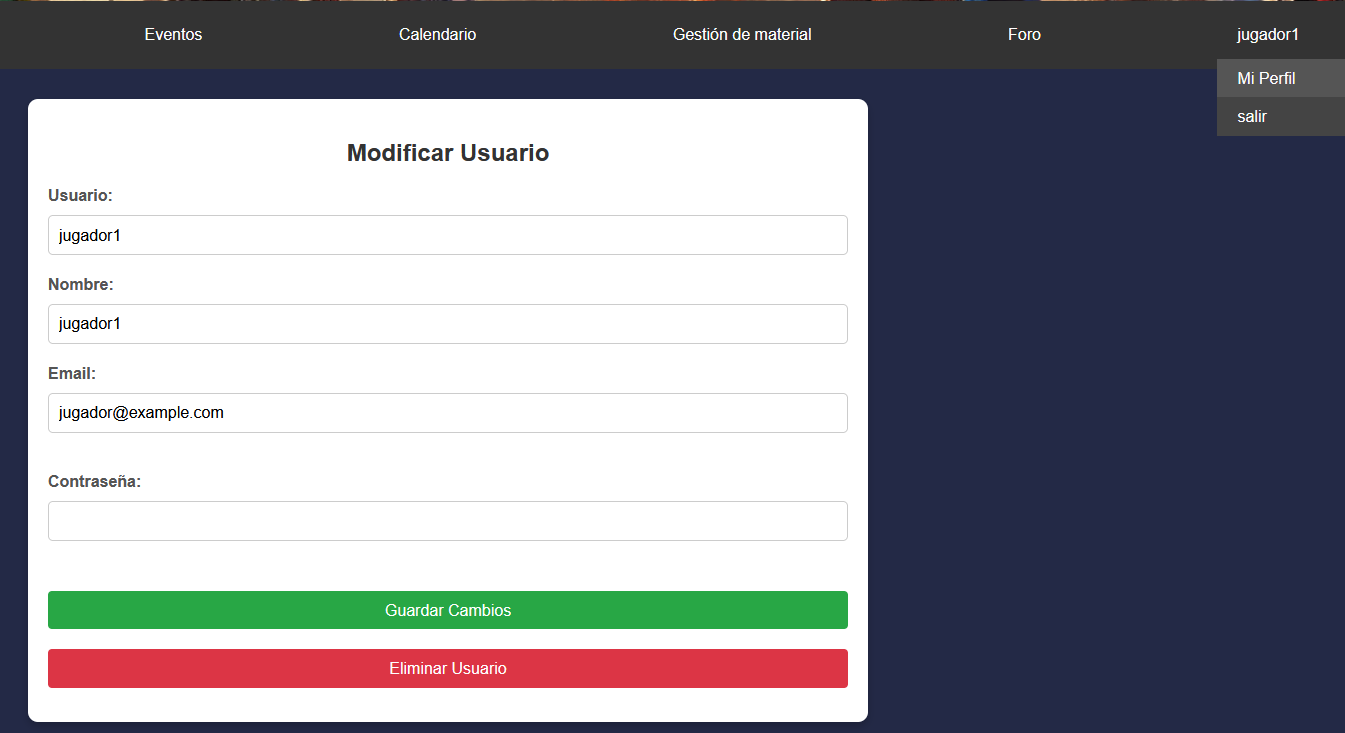


****

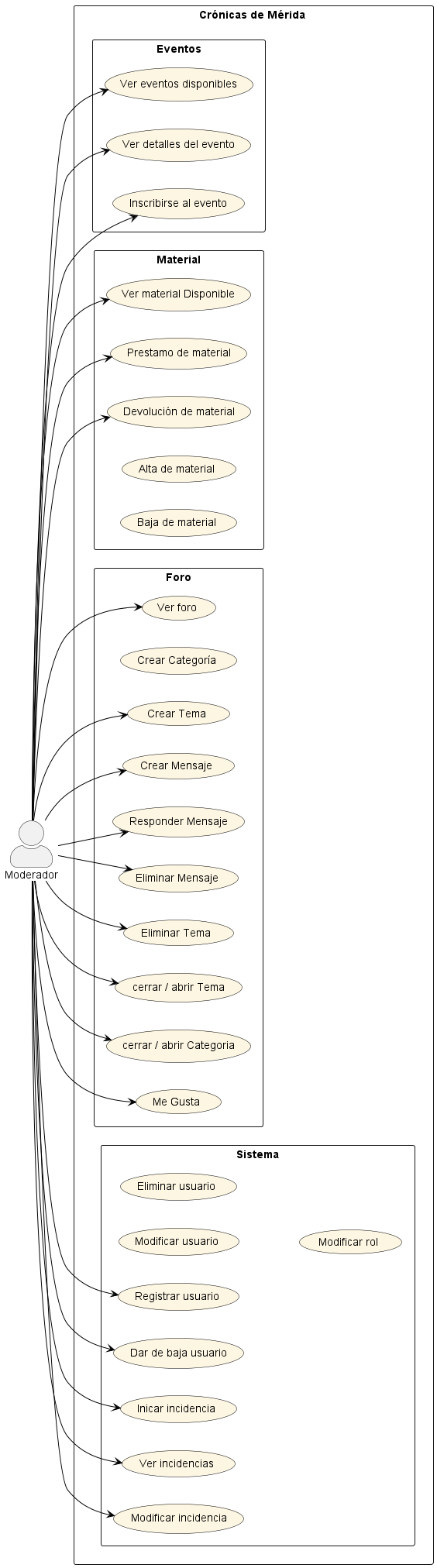
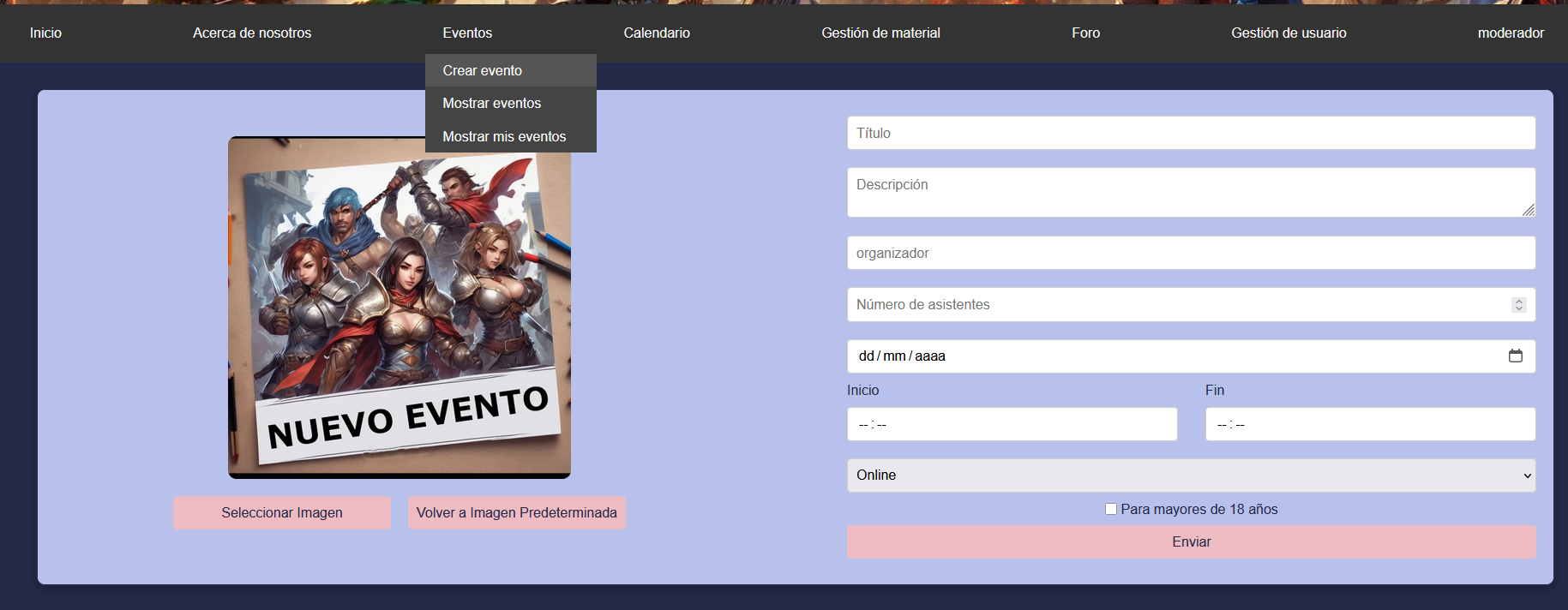
**Reserva/alquilar de juegos: el usuario puede reservar/alquilar juegos y ver los juegos que tiene reservados/alquilados.**

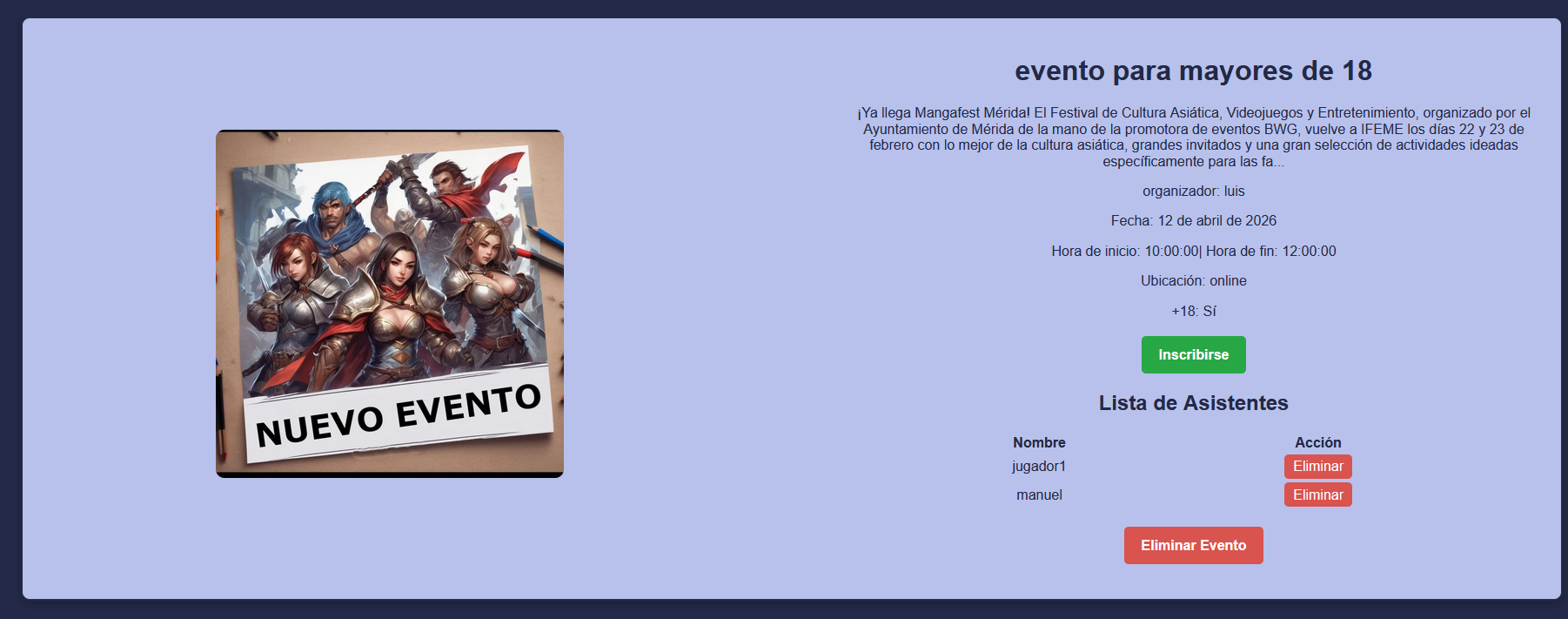
****

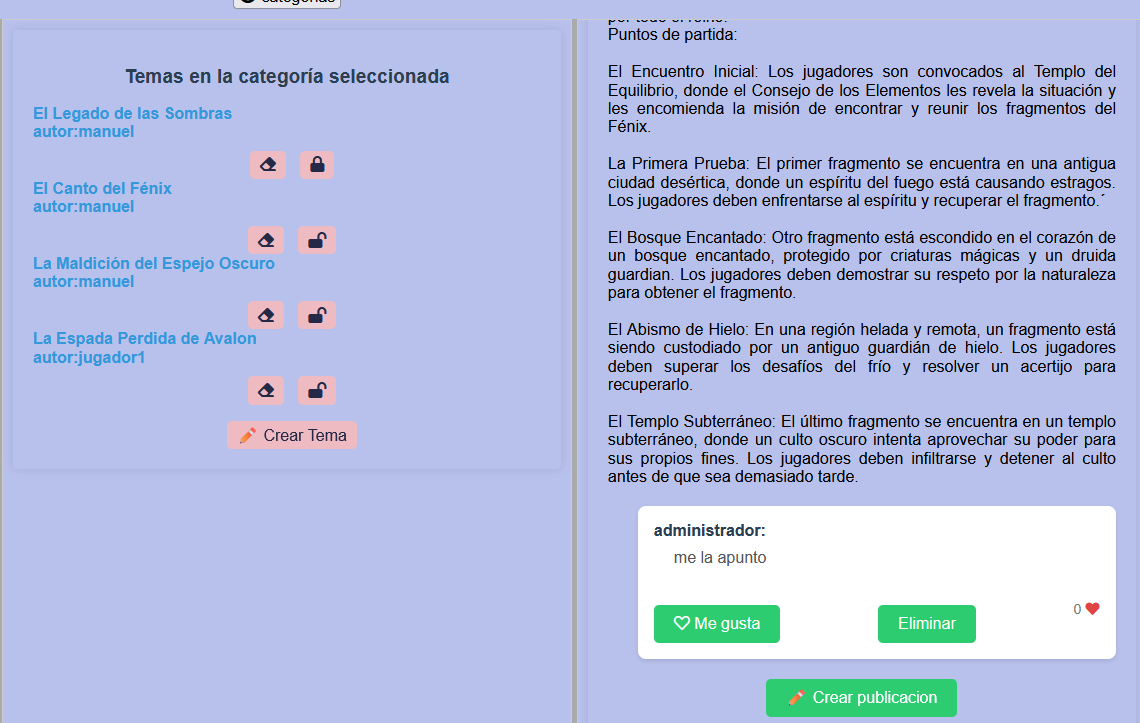
**Puedes modificar tus datos o eliminar cuenta.**

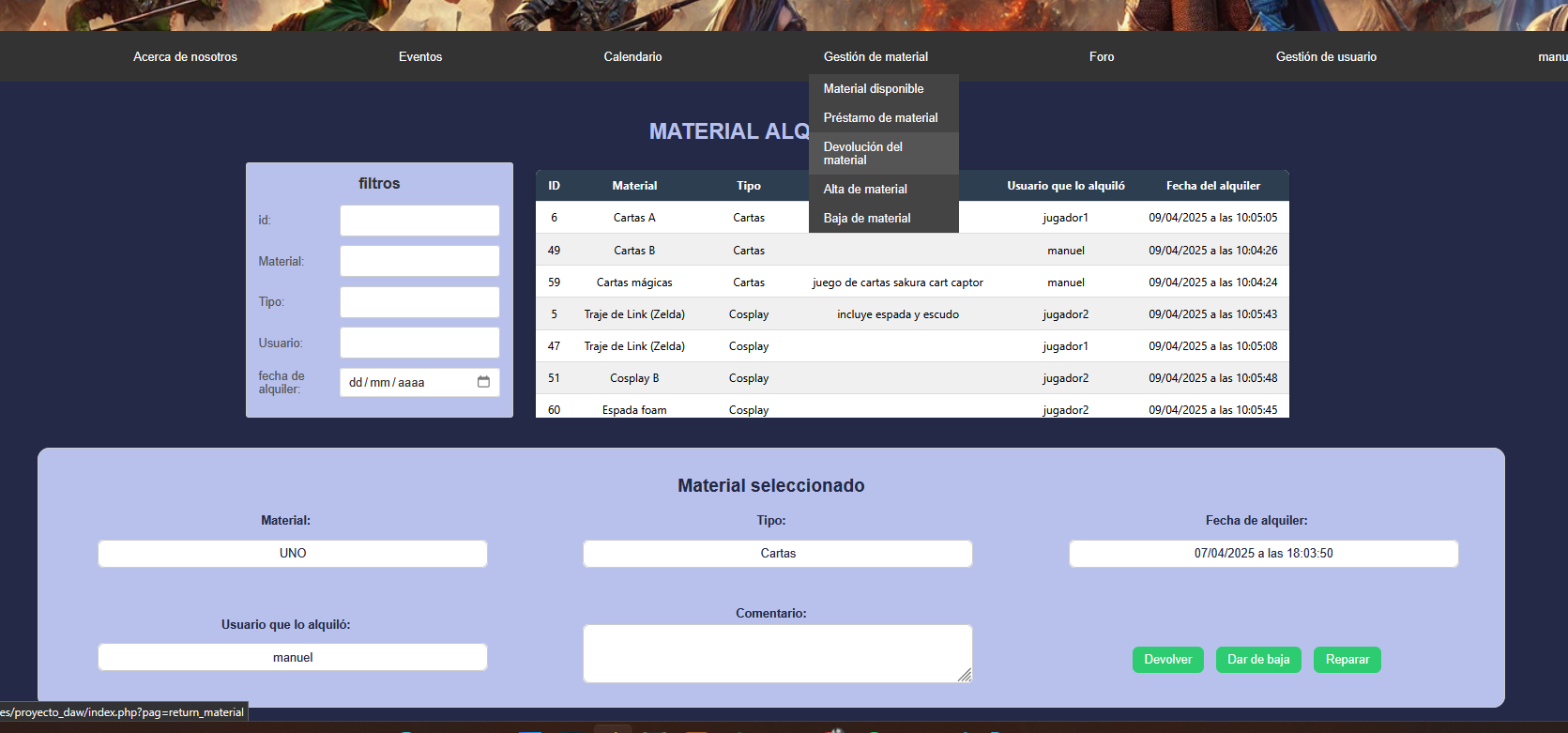


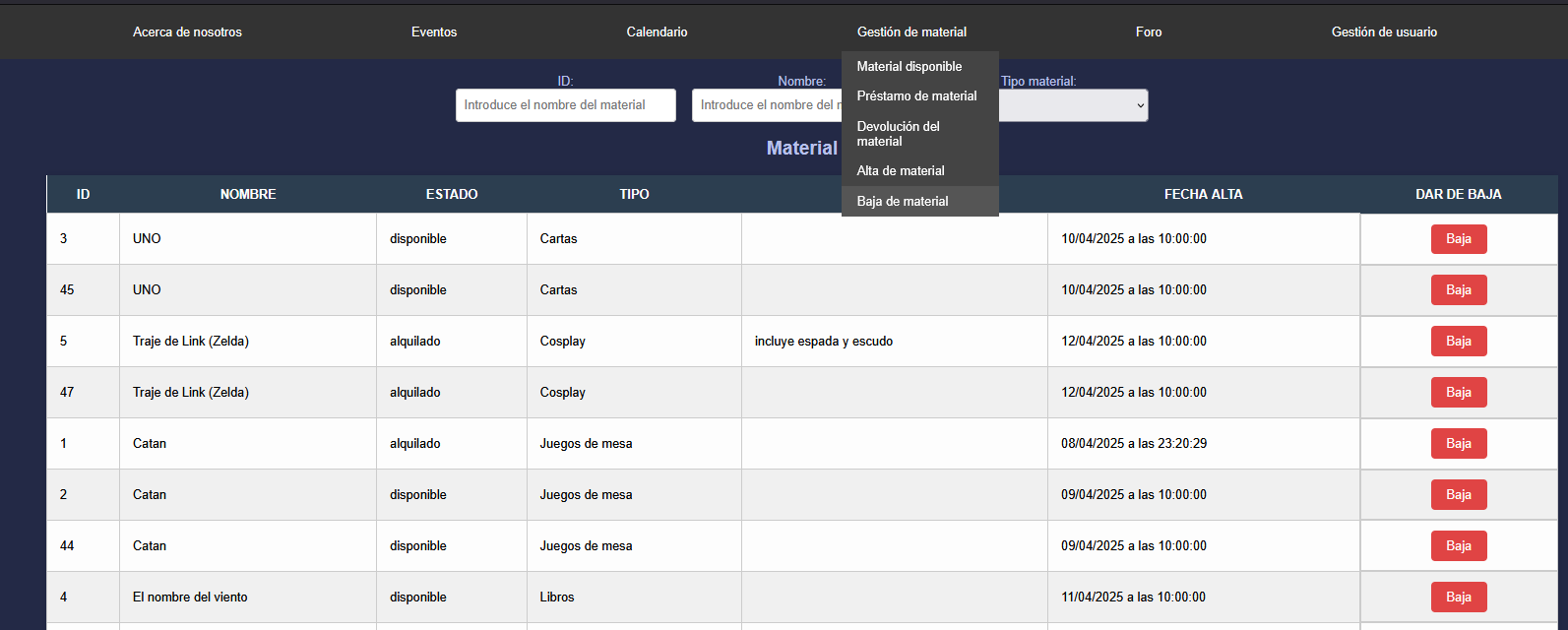
**moderador:**

**  
tienes todas las funcionalidades de Jugador.  
  
Puede crear eventos.**

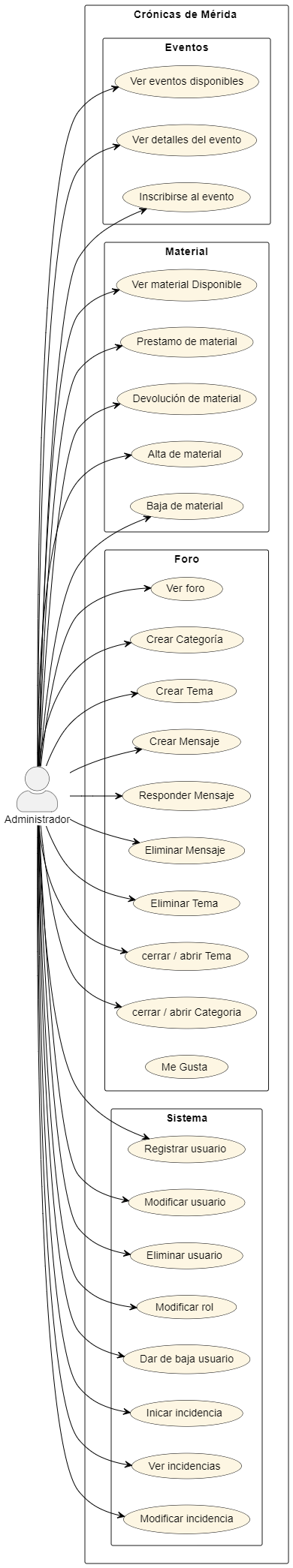
**Puede eliminar los usuarios de un evento o eliminar el evento completo** **En el foro puede eliminar cualquier Tema o publicación**



**puede recibir los juegos alquilados, comentarlo, mandarlo a reparar, o darlo de baja.  
puede filtrar el material con los inputs correspondientes**

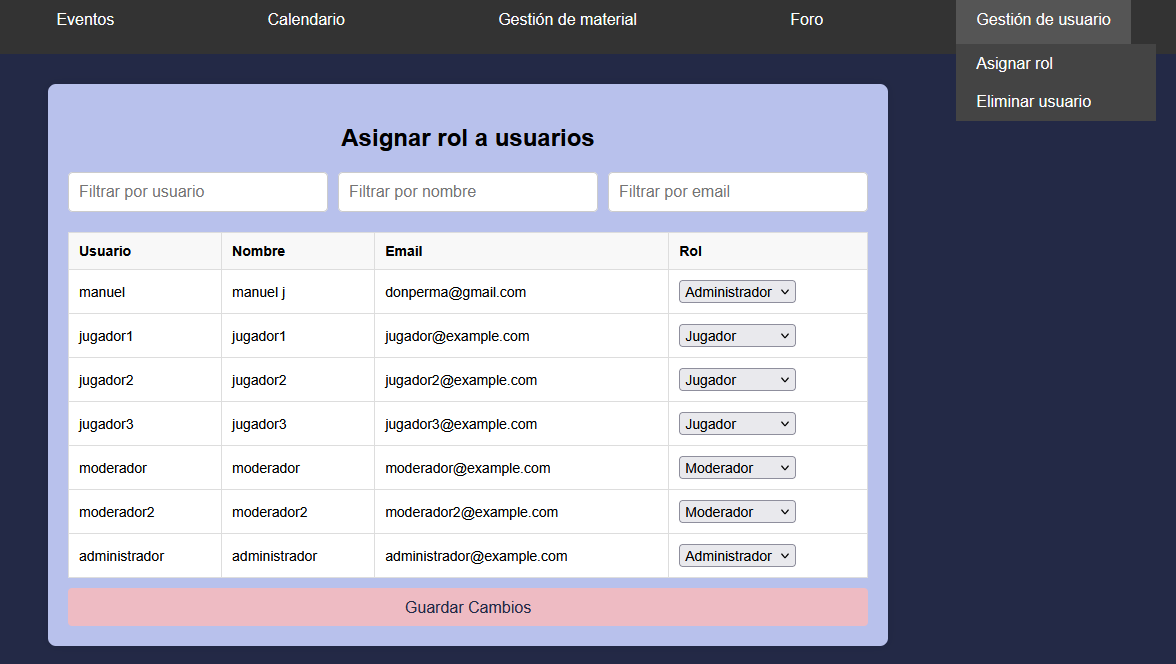
**Puede dar de alta a material nuevo: al empezar a escribir el nombre aparecerán en una tabla alado los materiales que existen en el sistema, en caso de aparecer el nuestro, se puede clicar sobre la fila del material y se autocompletara los campos menos el de comentario. Una vez dado de alta se guardará también en el sistema.** **Puede dar de baja material que no está alquilado**

**Administrador:**

****

**Tiene todas las funcionalidades de moderador.**

**Puede asignar roles**



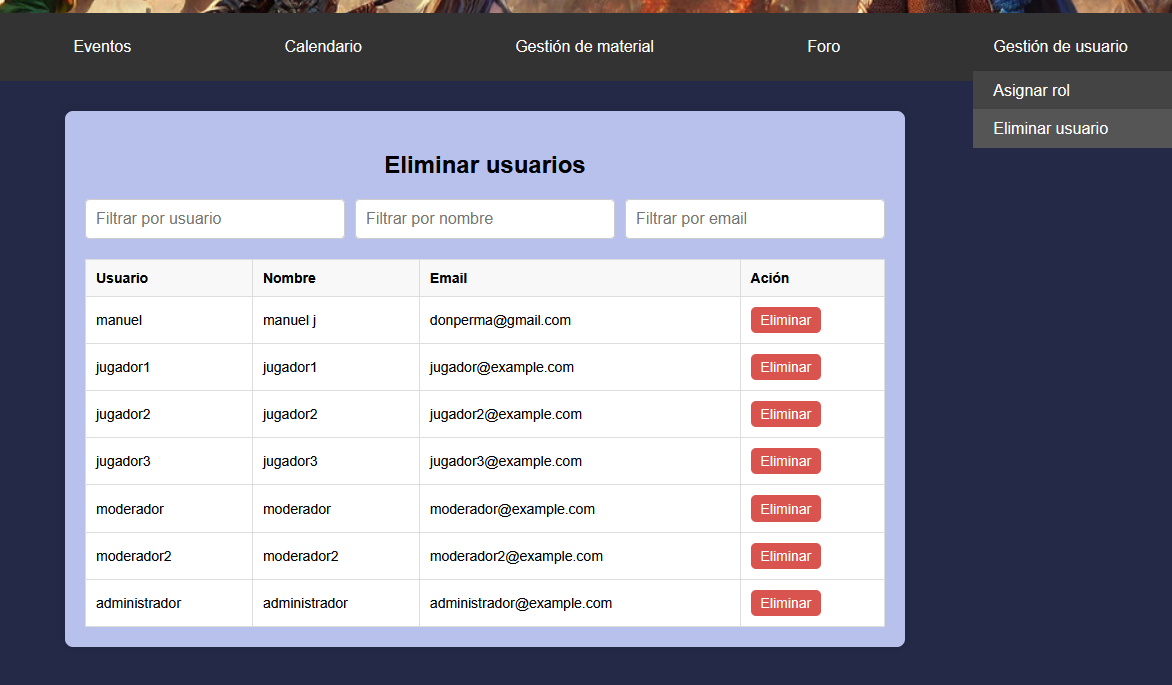
**Puede eliminar usuarios:** **En el foro. Puede crear categorías. Y permitir o no creación de temas dentro de estas.**

Diagrama de clases

**Usuario** (Clase base)

id: int

nombre: String

email: String

contraseña: String

editarPerfil()

eliminarCuenta()

**Administrador** (Hereda de Usuario)

gestionarUsuarios()

asignarRoles()

**Moderador** (Hereda de Usuario)

eliminarUsuarios()

**Jugador** (Hereda de Usuario)

inscribirseEvento()

**Evento**

id: int

nombre: String

fecha: Date

crearEvento()

eliminarEvento()

**Inscripción**

id: int

jugador: Jugador

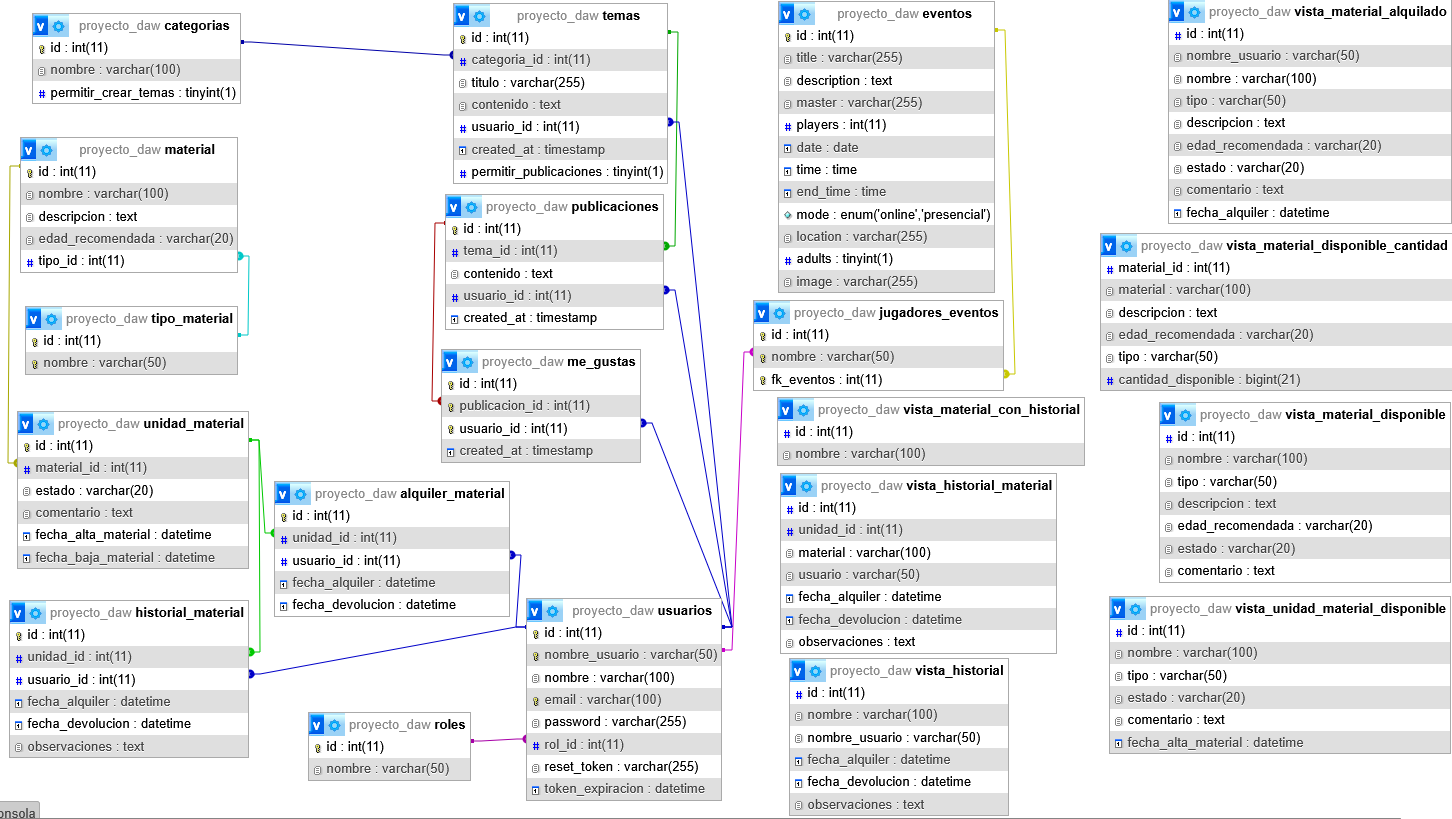
evento: Evento

inscribirJugador()

eliminarJugador()

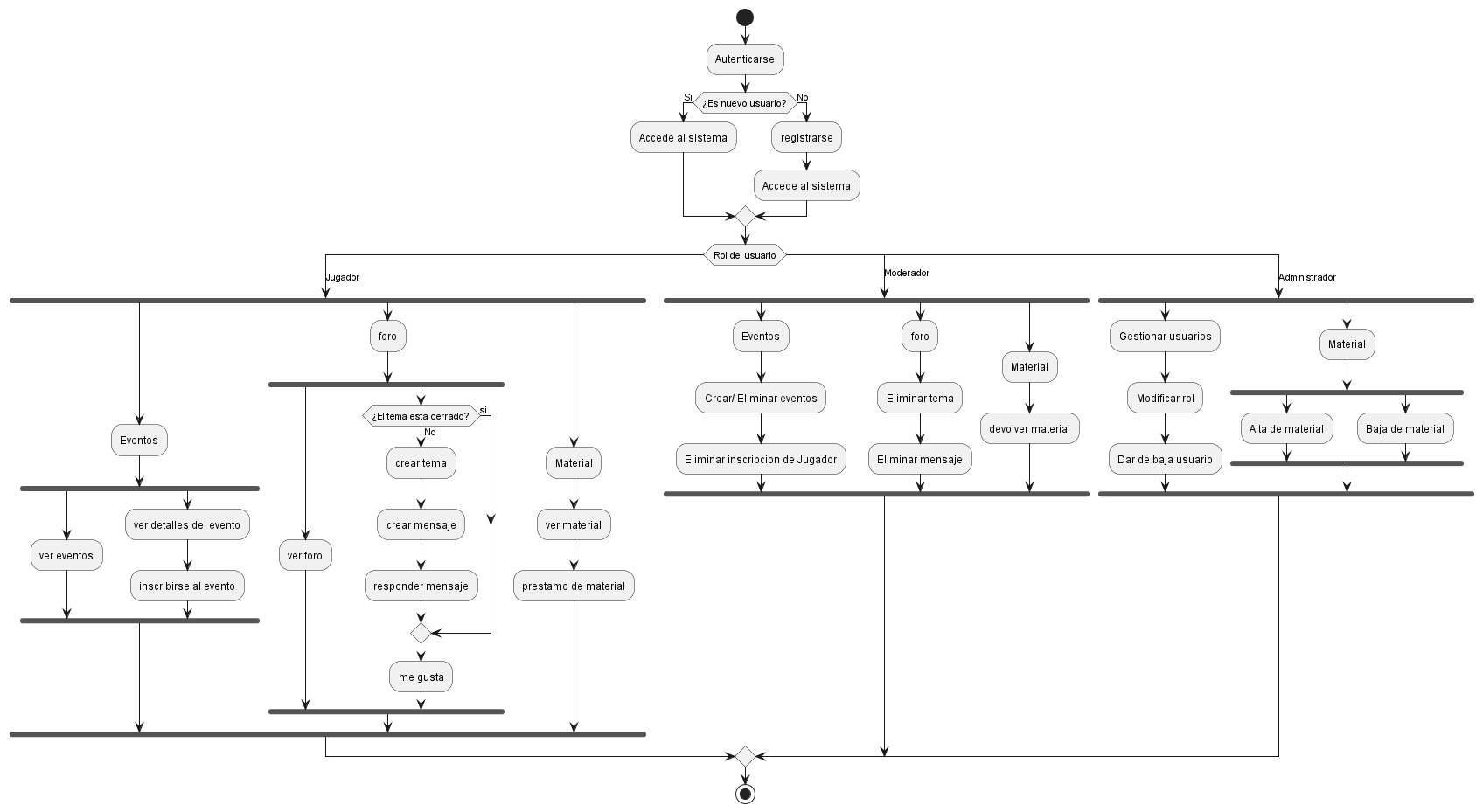
Estas clases y sus relaciones reflejan la estructura lógica de la aplicación.

Diagrama de entidad-relación



Desarrollo

**diagrama de secuencias**

****

**Registro de Usuario**

**Usuario -> Sistema: Solicitar Registro**

**Sistema -> Vista: Mostrar Formulario de Registro**

**Usuario -> Vista: Completar Formulario**

**Vista -> Controlador: Enviar Datos del Formulario**

**Controlador -> Modelo: Validar Datos**

**Modelo -> Base de Datos: Guardar Usuario**

**Base de Datos -> Modelo: Confirmar Registro**

**Modelo -> Controlador: Confirmar Registro**

**Controlador -> Vista: Mostrar Confirmación de Registro**

**Vista -> Usuario: Confirmar Registro Exitoso**

**2. Inicio de Sesión**

**Usuario -> Sistema: Solicitar Inicio de Sesión**

**Sistema -> Vista: Mostrar Formulario de Inicio de Sesión**

**Usuario -> Vista: Completar Formulario**

**Vista -> Controlador: Enviar Datos del Formulario**

**Controlador -> Modelo: Validar Credenciales**

**Modelo -> Base de Datos: Verificar Usuario**

**Base de Datos -> Modelo: Confirmar Credenciales**

**Modelo -> Controlador: Confirmar Sesión**

**Controlador -> Vista: Mostrar Confirmación de Sesión**

**Vista -> Usuario: Confirmar Sesión Exitosa**

**3. Creación de Evento (Administrador)**

**Administrador -> Sistema: Solicitar Creación de Evento**

**Sistema -> Vista: Mostrar Formulario de Creación de Evento**

**Administrador -> Vista: Completar Formulario**

**Vista -> Controlador: Enviar Datos del Formulario**

**Controlador -> Modelo: Validar Datos**

**Modelo -> Base de Datos: Guardar Evento**

**Base de Datos -> Modelo: Confirmar Creación**

**Modelo -> Controlador: Confirmar Creación**

**Controlador -> Vista: Mostrar Confirmación de Creación**

**Vista -> Administrador: Confirmar Creación Exitosa**

**4. Inscripción en Evento (Jugador)**

**Jugador -> Sistema: Solicitar Inscripción en Evento**

**Sistema -> Vista: Mostrar Formulario de Inscripción**

**Jugador -> Vista: Completar Formulario**

**Vista -> Controlador: Enviar Datos del Formulario**

**Controlador -> Modelo: Validar Datos**

**Modelo -> Base de Datos: Guardar Inscripción**

**Base de Datos -> Modelo: Confirmar Inscripción**

**Modelo -> Controlador: Confirmar Inscripción**

**Controlador -> Vista: Mostrar Confirmación de Inscripción**

**Vista -> Jugador: Confirmar Inscripción Exitosa**

**5. Publicación en el Foro (Jugador)**

**Jugador -> Sistema: Solicitar Publicación en el Foro**

**Sistema -> Vista: Mostrar Formulario de Publicación**

**Jugador -> Vista: Completar Formulario**

**Vista -> Controlador: Enviar Datos del Formulario**

**Controlador -> Modelo: Validar Datos**

**Modelo -> Base de Datos: Guardar Publicación**

**Base de Datos -> Modelo: Confirmar Publicación**

**Modelo -> Controlador: Confirmar Publicación**

**Controlador -> Vista: Mostrar Confirmación de Publicación**

**Vista -> Jugador: Confirmar Publicación Exitosa**

**Descripción de los Diagramas**

**Registro de Usuario:**

**El usuario solicita el registro y completa el formulario.**

**El controlador valida los datos y los envía al modelo.**

**El modelo interactúa con la base de datos para guardar el usuario.**

**Se confirma el registro y se notifica al usuario.**

**Inicio de Sesión:**

**El usuario solicita el inicio de sesión y completa el formulario.**

**El controlador valida los datos y los envía al modelo.**

**El modelo verifica las credenciales en la base de datos.**

**Se confirma la sesión y se notifica al usuario.**

**Creación de Evento (Administrador):**

**El administrador solicita la creación de un evento y completa el formulario.**

**El controlador valida los datos y los envía al modelo.**

**El modelo interactúa con la base de datos para guardar el evento.**

**Se confirma la creación y se notifica al administrador.**

**Inscripción en Evento (Jugador):**

**El jugador solicita la inscripción en un evento y completa el formulario.**

**El controlador valida los datos y los envía al modelo.**

**El modelo interactúa con la base de datos para guardar la inscripción.**

**Se confirma la inscripción y se notifica al jugador.**

**Publicación en el Foro (Jugador):**

**El jugador solicita la publicación en el foro y completa el formulario.**

**El controlador valida los datos y los envía al modelo.**

**El modelo interactúa con la base de datos para guardar la publicación.**

**Se confirma la publicación y se notifica al jugador.**

**Herramientas y Tecnologías Utilizadas**

**Frontend:**

**HTML5: Utilizado para la estructura del contenido de la aplicación.**

**CSS3: Para el diseño y estilización de la interfaz de usuario.**

**JavaScript: Lenguaje de programación utilizado para la lógica del frontend.**

**Backend:**

**PHP 8.2.26: Lenguaje de programación para la lógica del servidor y la gestión de peticiones.**

**MariaDB 10.11.6: Sistema de gestión de bases de datos relacional.**

**Apache/2.4.62: Servidor web encargado de gestionar las peticiones HTTP.**

**Entorno de Desarrollo**

**Sistema Operativo: Debian 12**

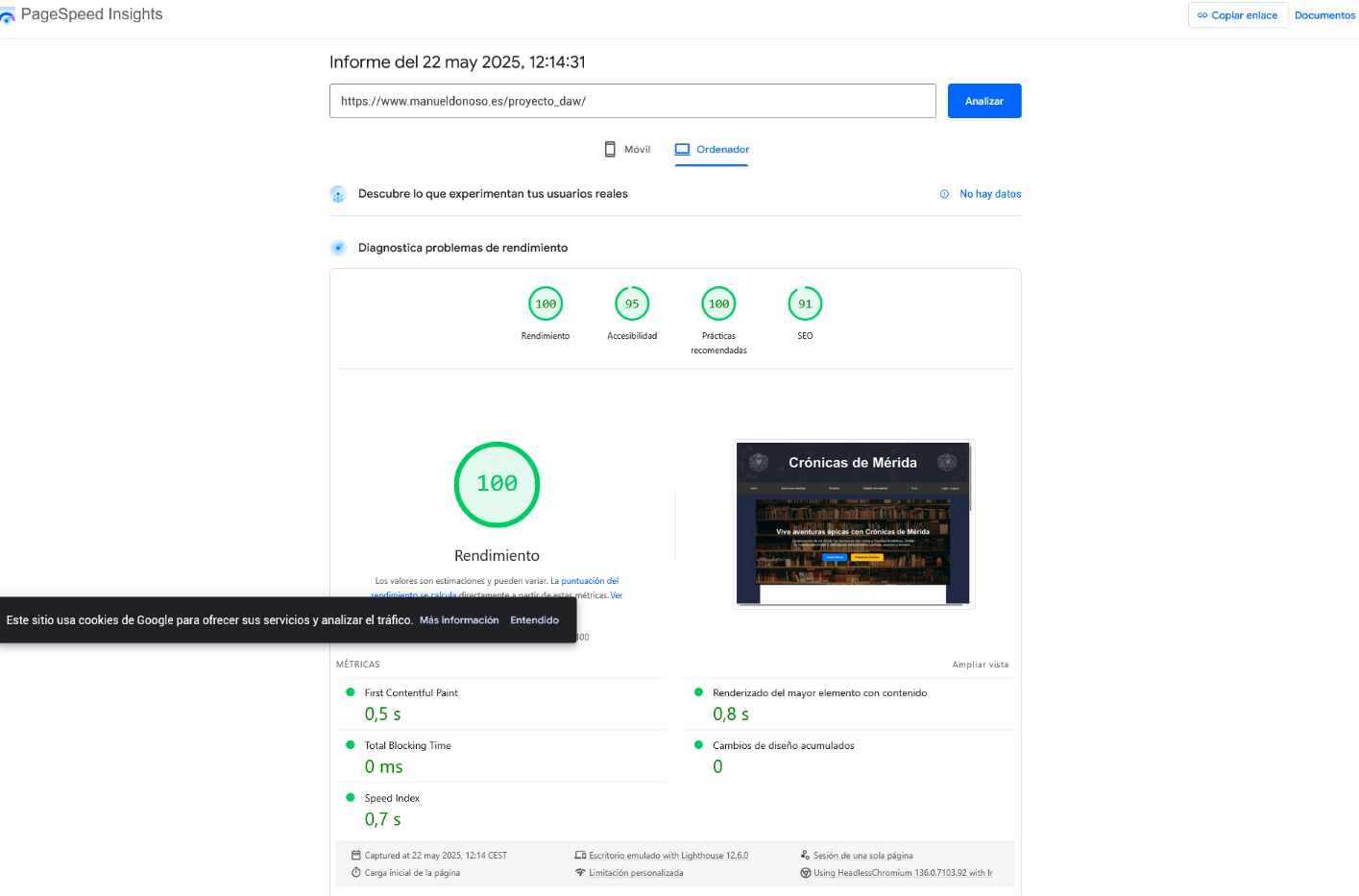
**Editor de Código: Visual Studio Code 1.98.1**

**Herramientas Adicionales: Extensiones de VSCode para HTML, CSS, JavaScript, PHP y base de datos.**

Pruebas realizadas

|  |  |
| --- | --- |
| Prueba | Resultado |
| Login | Satisfactorio |
| Crear evento | Satisfactorio |
| Inscribirse en evento | Satisfactorio |
| Eliminar a un jugador de un evento | Satisfactorio |
| Eliminar evento | Satisfactorio |
| Crear categoría en el foro | Satisfactorio |
| Eliminar Categoría en el foro | Satisfactorio |
| Crear tema en el foro | Satisfactorio |
| Eliminar Tema en el foro | Satisfactorio |
| Crear Publicación en el foro | Satisfactorio |
| Eliminar Publicación en el foro | Satisfactorio |
| Cambio de rol de usuario | Satisfactorio |
| Dar de baja a cuenta | Satisfactorio |
| Cambiar contraseña | Satisfactorio |
| Enviar email contraseña olvidada | Satisfactorio |
| Dar de alta Material | Satisfactorio |
| Dar de baja a material | Satisfactorio |
| Devolución de material | Satisfactorio |
| Crear una incidencia | Satisfactorio |
| Resolver una incidencia | Satisfactorio |
| Enviar un email de contacto | Satisfactorio |

Test realizado por un tercero.



1. SEO (91%)

Qué mide: Evalúa cómo de bien está optimizado el sitio para los motores de búsqueda (Google, Bing, etc.).

Aspectos clave:

Metaetiquetas (título, descripción).

Estructura de encabezados (H1, H2, etc.).

URLs amigables.

Uso de atributos alt en imágenes.

Enlaces internos/externos optimizados.

2. Prácticas recomendadas (100%)

Qué mide: Verifica que el sitio siga buenas prácticas de desarrollo web y seguridad.

Aspectos clave:

Uso de HTTPS.

Eliminación de librerías obsoletas (ej: jQuery viejo).

Configuración correcta de CORS.

Errores de JavaScript.

3. Accesibilidad (95%)

Qué mide: Evalúa si el sitio es usable para personas con discapacidades (ej: lectores de pantalla).

Aspectos clave:

Contraste de colores adecuado.

Etiquetas ARIA para elementos interactivos.

Estructura semántica (uso correcto de HTML5).

4. Rendimiento (100%)

Qué mide: Velocidad y eficiencia de carga del sitio.

Aspectos clave:

Tiempo de carga (First Contentful Paint, Largest Contentful Paint).

Optimización de imágenes y recursos.

Uso de caché.

Minimización de CSS/JS.

Proceso de despliegue

**Preparación del Entorno de Despliegue**

**1.1. Instalación de Apache**

**Asegúrese de que el servidor Apache esté instalado y configurado correctamente en el servidor. Puede instalar Apache utilizando el gestor de paquetes de su sistema operativo. Por ejemplo, en un sistema basado en Debian (como Debian), puede utilizar el siguiente comando:**

**sudo apt-get update**

**sudo apt-get install apache2**

**1.2. Verificación del Servidor Apache**

**Verifique que el servidor Apache esté en ejecución y que pueda acceder a él a través de un navegador web. Puede hacer esto visitando http://[su\_dirección\_ip] o http://localhost si está trabajando en un entorno local.**

**2. Preparación del Código Fuente**

**2.1. Organización del Código Fuente**

**Asegúrese de que su código fuente esté organizado en una estructura de carpetas que facilite el despliegue. 2.2. Copia del Código Fuente al Servidor**

**Copie el código fuente de su aplicación web al directorio raíz de Apache. Por defecto, este directorio es /var/www/html/ en la mayoría de las distribuciones de Linux. Puede utilizar el comando scp o cualquier otro método de transferencia de archivos para copiar los archivos al servidor.**

**3. Configuración del Servidor Apache**

**3.1. Configuración del Virtual Host**

**Configure un Virtual Host en Apache para manejar las solicitudes a su aplicación web. Edite el archivo de configuración de Apache (generalmente /etc/apache2/sites-available/000-default.conf) y agregue la siguiente configuración:**

**3.2. Habilitar Módulos Necesarios**

**Asegúrese de que los módulos necesarios estén habilitados en Apache. Por ejemplo, si su aplicación utiliza PHP, debe habilitar el módulo php:**

**sudo a2enmod php**

**3.3. Reiniciar Apache**

**Reinicie el servidor Apache para aplicar los cambios de configuración:**

**sudo systemctl restart apache2**

**4. Configuración de la Base de Datos**

**4.1. Instalación de MySQL**

**Si su aplicación utiliza una base de datos, instale MySQL en el servidor:**

**sudo apt-get install mysql-server**

**4.2. Creación de la Base de Datos y Usuario**

**Cree la base de datos y el usuario necesario para su aplicación:**

**4.3. Importar el Script de la Base de Datos**

Importe el script de la base de datos (generalmente un archivo .sql)

5. Verificación y Pruebas

5.1. Acceso a la Aplicación Web

Acceda a su aplicación web a través de un navegador web utilizando la dirección IP del servidor o el nombre de dominio configurado:

http://[su\_dirección\_ip]

5.2. Realizar Pruebas

Realice pruebas exhaustivas para asegurarse de que la aplicación web funcione correctamente. Verifique la funcionalidad de todas las páginas, formularios y scripts.

6. Mantenimiento y Seguridad

*6.1. Actualizaciones y Parches*

Mantenga el servidor actualizado con las últimas versiones de Apache, PHP y cualquier otro software utilizado. Utilice el gestor de paquetes de su sistema para aplicar actualizaciones:

sudo apt-get update

sudo apt-get upgrade

6.2. Seguridad

Implemente medidas de seguridad adicionales, como firewalls, actualizaciones de seguridad y configuraciones de acceso restringido. Asegúrese de que los archivos de configuración y los datos sensibles estén protegidos.

Propuestas de mejoras

* Autentificación Social:
  + Integrar login con Google y Facebook
* Notificaciones en tiempo real
  + Implementación webSockets para alerta instantáneas
* Migración a Laravel
  + Mejorar robustez y escalabilidad del sistema

Bibliografía

- Portal de Administración Electrónica - Métrica v3. <https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html>

- ISO/IEC/IEEE 29119 Software Testing Standard. <https://in2test.lsi.uniovi.es/gt26/>

- Abraham Silversatz - "Database Systems Concepts".  
- Php.net - Manual de PHP. <https://www.php.net/manual/es/index.php>

- Mysql.com - Manual de Referencia de Mysql. <https://dev.mysql.com/doc/refman/8.0/en/>