

# INFORME FINAL:

## CLUB DE ROL

Alumno/a: CARLOS GARCÍA DE CASTRO

Asignatura: Aplicaciones Híbridas

Fecha: 14/12/2025

### 1. Explicación de la Arquitectura de la Aplicación

El proyecto es una aplicación Full Stack desarrollada con Node.js/Express para el Backend y React JS para el Frontend.

#### A. Estructura del Backend (API REST)

El servidor implementa una Arquitectura de Capas clara, similar al patrón MVC (Modelo-Vista-Controlador), separando la lógica para mayor orden.

- *Router*: Define las URLs (/api/juegos, /api/campanas, etc.) y actúa como puerta de entrada, aplicando middlewares de seguridad antes de pasar la petición al controlador.
- *Middleware*: Son filtros que verifican los datos de entrada (validación de formularios) y la seguridad (validateToken para la autorización).
- *Controller*: Recibe la petición, decide qué hacer con ella y llama a la capa de servicios. Se encarga de devolver el código HTTP correcto (200, 401, 500).
- *Service*: Contiene la lógica de negocio. Aquí es donde se realizan las operaciones con MongoDB, la encriptación y el manejo de JWT.

#### B. Estructura del Frontend (React SPA)

El Frontend es una Single Page Application (SPA) construida con React, enfocada en una buena experiencia de usuario.

- *Context API*: Usamos SessionContext para manejar el estado global de la autenticación. Si el usuario se loguea, su token y sus datos se guardan en el contexto, y persisten en el localStorage para evitar que se pierdan al recargar la página.
- *Custom Hooks*: Implementamos hooks (como useCampanas) para centralizar la lógica de obtención y manejo de datos, manteniendo los componentes limpios.
- *Estilo*: Usamos Tailwind CSS para un diseño responsivo y moderno, centrado en la usabilidad, se le dio atmósfera de “juego de rol”.

### 2. Explicación de Autenticación y Autorización con JWT

El sistema de seguridad protege todas las rutas sensibles (CRUD) usando JWT y encriptación.

#### A. Autenticación y Encriptación

1. *Encriptación (bcrypt)*: Al registrar un usuario, la contraseña se encripta inmediatamente usando bcrypt con un salt de coste 10. Solo se almacena el hash en la base de datos, garantizando la seguridad.
2. *JWT Generación*: Al iniciar sesión, el servidor verifica la contraseña encriptada. Si es correcta, genera un JSON Web Token (JWT) con una firma secreta. Este token es enviado al cliente.
3. *Recuperación*: Implementamos el flujo completo de Recuperación de Contraseña enviando un token de un solo uso por email (nodemailer) para restablecer la clave. *El antivirus dio algunos problemas, no funcionando en local con él activado.*

#### B. Autorización de Rutas (validateToken)

La autorización es el punto más importante para garantizar la privacidad de los datos.

1. *Envío del Token*: Desde el Frontend, todas las peticiones a rutas privadas adjuntan el JWT en el header Authorization con el formato Bearer <token>.
2. *Verificación (validateToken)*: Este middleware intercepta la petición y:
  - Verifica la firma del token con la clave secreta del servidor.
  - Comprueba que el token no haya expirado.
3. *Filtrado de Recursos*: Si el token es válido, decodificamos el payload y adjuntamos el ID del usuario a la sesión de Express (req.session.id = token.id).  
Esto es crucial, ya que permite a la capa de Servicios filtrar los datos, devolviendo solo las campañas y recursos que pertenecen al usuario que está logueado.

### 3. Cumplimiento Detallado del CRUD

Se implementó el CRUD completo para la entidad principal (Campañas / Proyectos) y la entidad secundaria (Juegos / Recursos).

#### A. Funcionalidad sobre Campañas (Proyectos)

Operación	Método HTTP	Detalle de Cumplimiento
Crear	POST /api/campanas	Permite crear un nuevo proyecto/campaña, asignándolo automáticamente al ID del usuario logueado.
Listar	GET /api/campanas	Devuelve <b>solo las campañas</b> creadas por el usuario autenticado (Privacidad).
Actualizar	PUT /api/campanas/:id	Permite editar todos los campos de una campaña existente.
Eliminar	DELETE /api/campanas/:id	Implementa un <b>borrado lógico</b> , marcando la campaña como inactiva en la base de datos.

## B. Funcionalidad sobre Juegos (Entidades Secundarias)

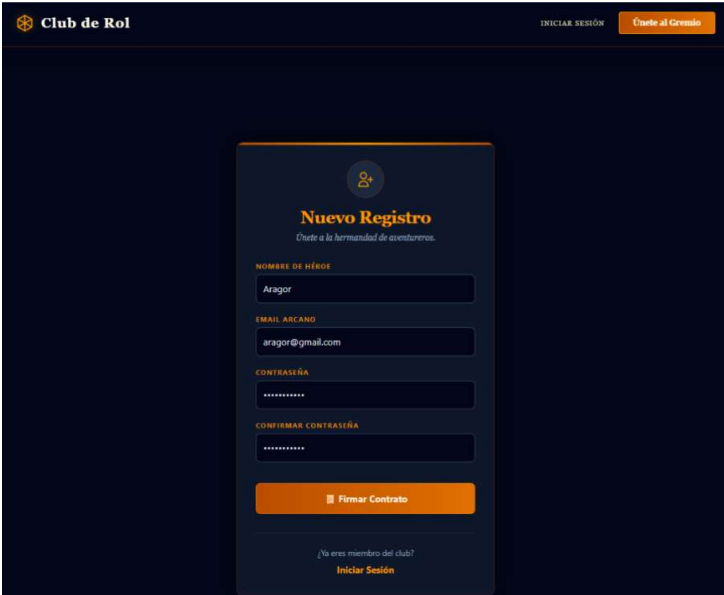
Operación	Método HTTP	Detalle de Cumplimiento
Crear	POST /api/juegos	Permite crear nuevos juegos para la biblioteca del club (requiere autenticación).
Listar	GET /api/juegos	Listado general de la biblioteca de juegos (permite filtros opcionales).
Actualizar	PATCH /api/juegos/:id	Permite modificar campos específicos del juego.
Eliminar	DELETE /api/juegos/:id	Implementa un <b>borrado lógico</b> , marcando el juego como inactivo en la base de datos.

## 4. Capturas de Pantalla de las Principales Funcionalidades

A continuación, se presentan las capturas de pantalla de las funcionalidades clave, demostrando la interfaz y la funcionalidad:


### A. Autenticación y Perfil

#### 1. Vista de Inicio de Sesión



Club de Rol

INICIAR SESIÓN [Únete al Grupo](#)



### Nuevo Registro

*Únete a la hermandad de aventureros.*

NOMBRE DE HÉROE

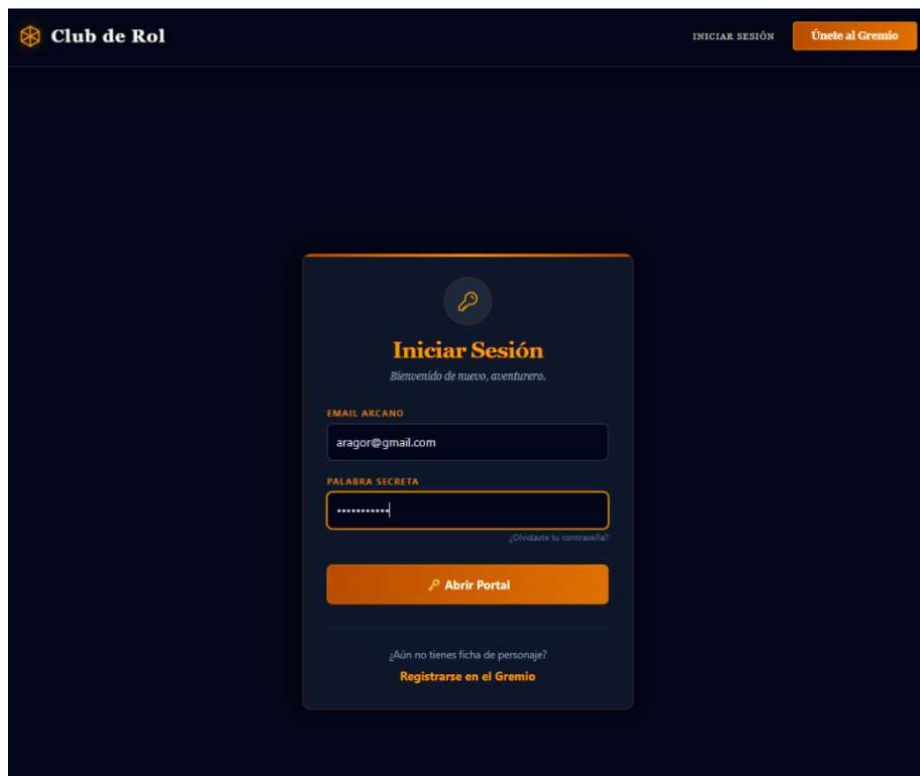
EMAIL ARCANO

CONTRASEÑA

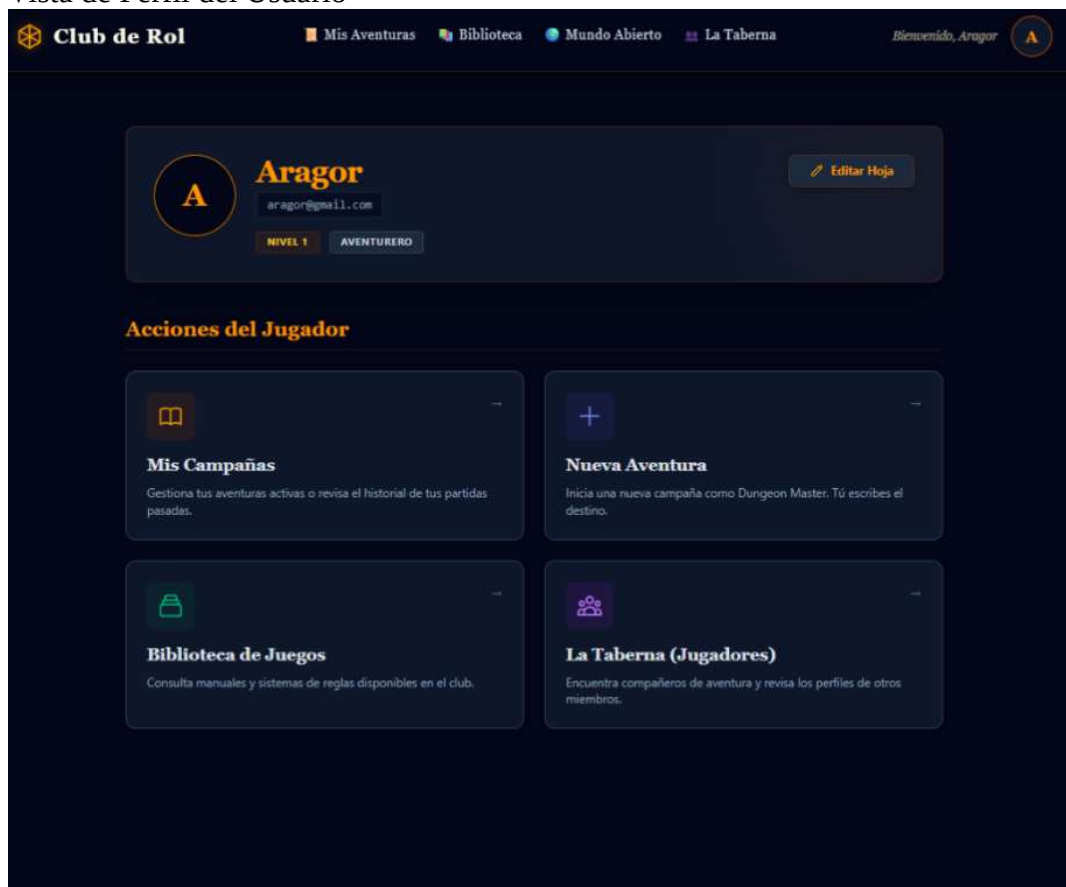
CONFIRMAR CONTRASEÑA

¿Ya eres miembro del club?

[Iniciar Sesión](#)

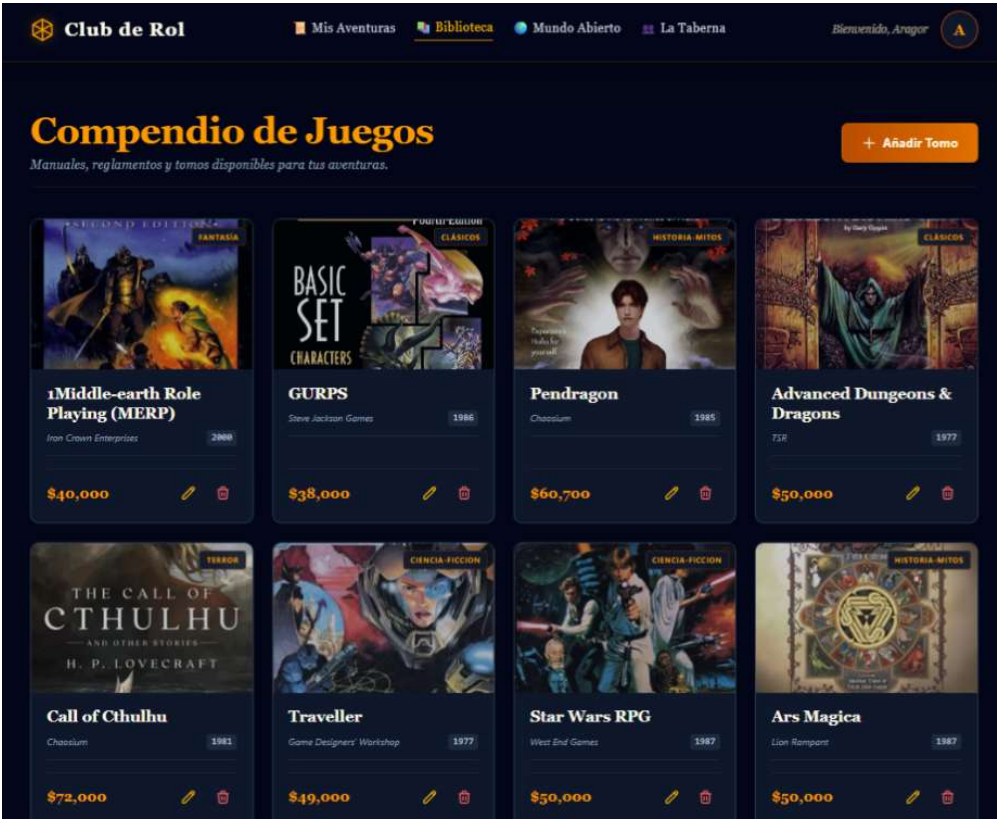


## 2. Vista de Perfil del Usuario

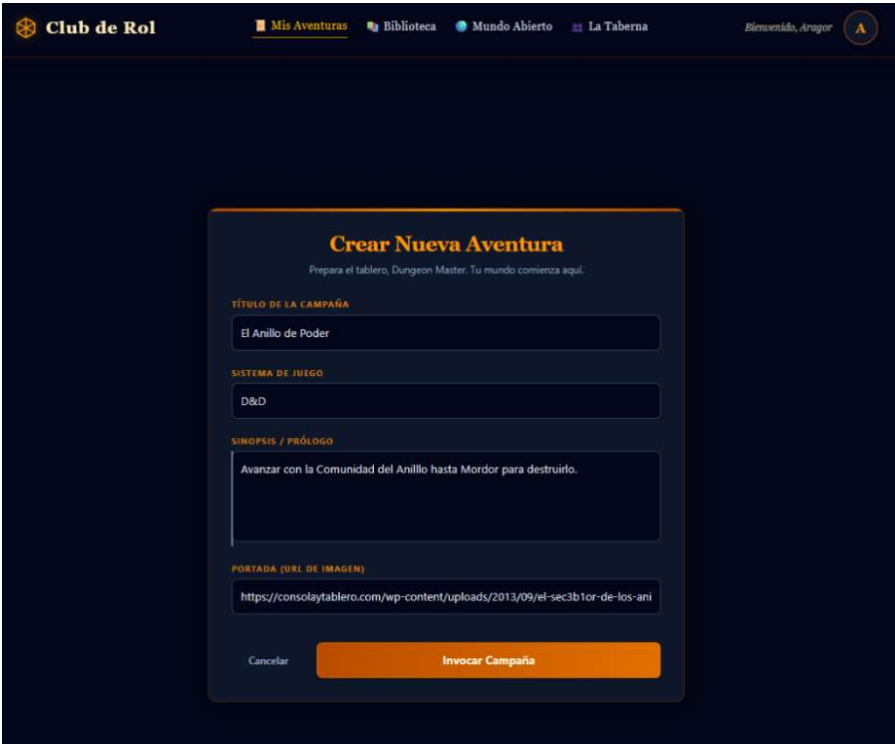


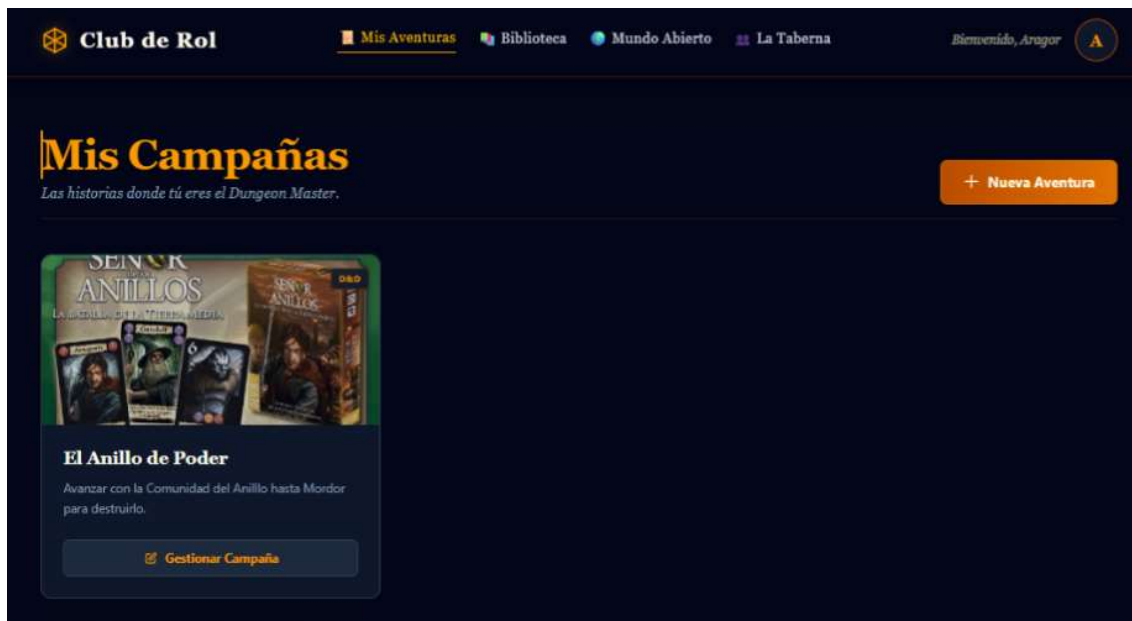
B. Gestión de Proyectos y Recursos

3. Biblioteca de Juegos (Listado)

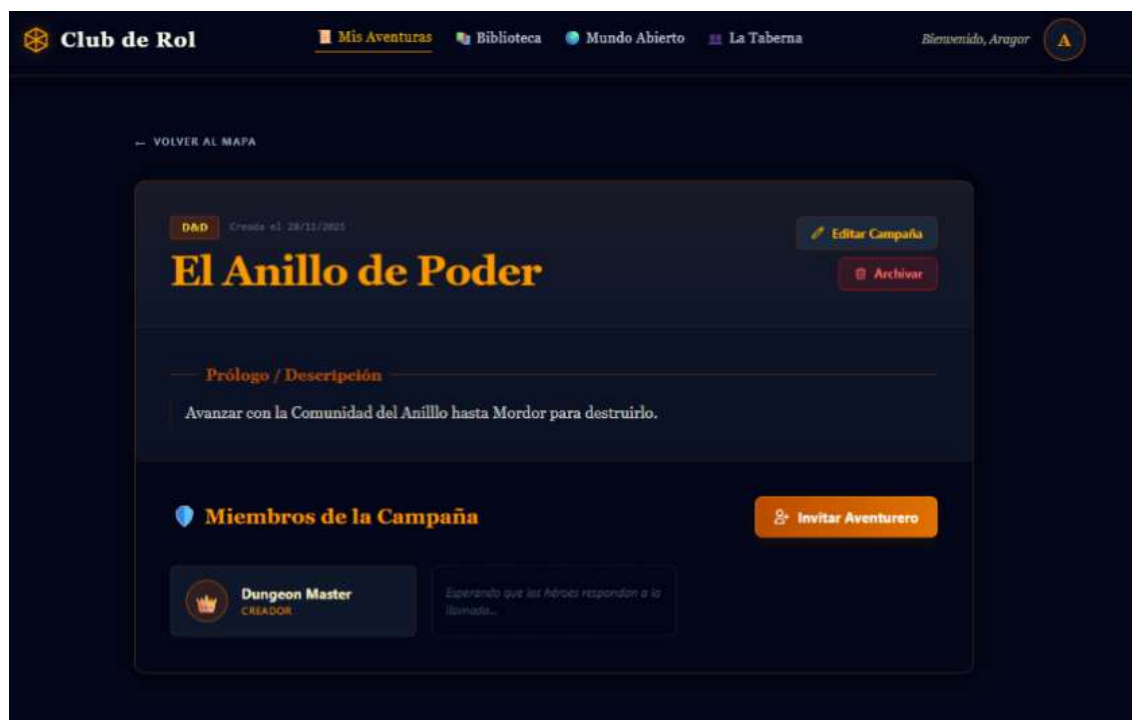


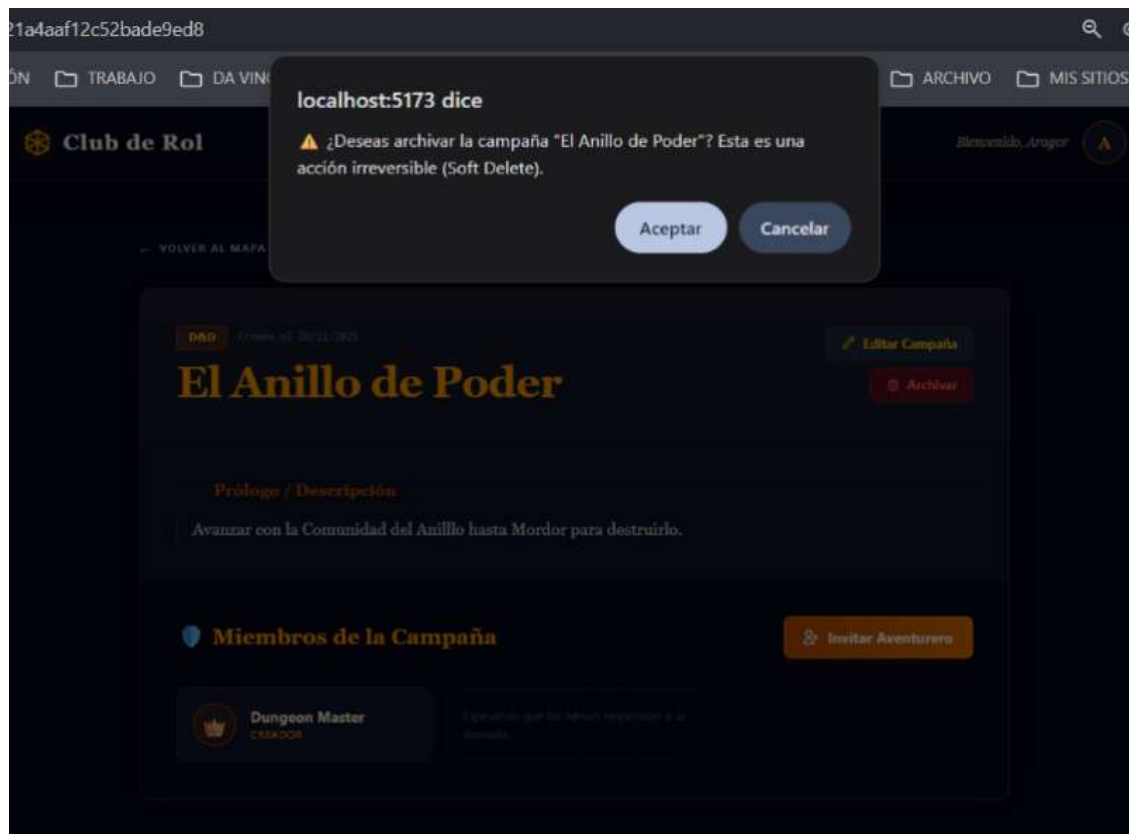
4. Detalle de Campaña (Proyecto)





## 5. Formulario de Edición





# ANEXO FINAL: AMPLIACIÓN DE FUNCIONALIDADES

## 1. Introducción


El presente anexo detalla las funcionalidades extendidas implementadas para la entrega final del proyecto "Club de Rol". Estas mejoras se centran en la **interacción social entre usuarios** (unirse a campañas, perfiles públicos), la **infraestructura** y el perfeccionamiento de los **flujos de seguridad**.

## 2. Interacción Social y Gestión de Usuarios

Para evolucionar la aplicación de una herramienta de gestión personal a una plataforma comunitaria, se implementaron nuevos endpoints y lógicas de controlador que permiten la interacción entre distintas cuentas de usuario.

Operación	Método HTTP	Detalle de Implementación
Unirse a Campaña	POST /api/cam panas/u nirse	Permite a un jugador autenticado crear una campaña e invitar a otros jugadores. El sistema valida la solicitud y actualiza la relación en la base de datos, añadiendo el ID del usuario a la lista de participantes de la aventura.
Ver Perfil Público	GET /api/juga dores/:id	Implementación de una vista segura de los perfiles de otros jugadores. El backend filtra la respuesta (Projection) para entregar solo datos no sensibles (Avatar, Nombre, Nivel), protegiendo el email y la contraseña del usuario consultado.



 Club de Rol

Mis AventurasBibliotecaMundo AbiertoLa Taberna

Bienvenido, GokuVence

G

VOLVER AL MAPA


13/12/2025


Hero 1


Aventura de HeroQuest


partidita de HeroQuest


Miembros (6)


 Dungeon Master  
NARRADOR


 Carlos

 Pepe

 Aragorn

 Master Carlos  
carlos@rol.com


 Lopera  
lopera2023@gmail.com

 Club de Rol

Mis AventurasBibliotecaMundo AbiertoLa Taberna

Bienvenido, GokuVence

G

 La Taberna

Conoce a los miembros del club y revisa sus hojas de personaje.

C  
Carlos

P  
Pepe

A  
Aragorn

P  
Pepe

M  
Master Carlos  
carlos@rol.com

L  
Lopera  
lopera2023@gmail.com

P  
Pepe  
ulises21@ulises21.com

G  
GokuVence  
garciaecastroprofesional@gmail.co...

D

N

U

D

← VOLVER A LA TABERNA



**Sauron**  
AVENTURERO DEL CLUB

CUERVO MENSAJERO (EMAIL)  
sauron@gmail.com

### Estadísticas de Rol

1

NIVEL DE CUENTA

?

CAMPAÑAS ACTIVAS

### Historia

"Este aventurero aún está escribiendo su leyenda. Sus hazañas son desconocidas para los bardos locales..."

~ Miembro oficial del Club de Rol ~