

# **PROYECTO INTEGRADO**

## **VTES-APP**

**Curso 2023/2024**

**IES Sotero Hernández**



## **INDICE**

<b>1 - Presentación</b>	<b><u>pag.2</u></b>
- <u>Introducción</u>	<u>pag.2</u>
- <u>Perfiles</u>	<u>pag.2</u>
- <u>Épica de la aplicación</u>	<u>pag.2</u>
- <u>Tecnologías y herramientas</u>	<u>pag.3</u>
- <u>Planificación</u>	<u>pag.3</u>
<b>2 - Diseño de la aplicación</b>	<b><u>pag.4</u></b>
- <u>Arquitectura del sistema</u>	<u>pag.5</u>
- <u>Entidades</u>	<u>pag.5</u>
- <u>Historial de Usuarios</u>	<u>pag.5</u>

## **INTRODUCCIÓN**

La aplicación consistirá en un conjunto de herramientas para la gestión de un juego de cartas y su comunidad. Constará de:

- Construcción de mazos con múltiples campos para filtrado de cartas.
- Pagina para diseñar cartas personalizadas.
- Creación y gestión de eventos para organizar los torneos.
- Foros de discusión.
- Visualizado de mazos propios y de la comunidad.
- Sistema de registro de Usuarios.
- Zona de Administrador para gestionar permisos.

<https://github.com/whejhe/App-Vtes.git>

<https://front-vtes.vercel.app/>

### **Perfiles de Usuarios:**

- La aplicación tendrá diferentes roles para permitir el acceso a las paginas o funcionalidades. Admin, user, invitado, colaborador.
- Los invitados tendrán solo acceso de lectura a la aplicación sin posibilidad de editar.
- Los usuarios registrados podrán crear cartas y mazos personalizados y participar en foros aunque no crearlos.
- Los colaboradores tendrán los mismos permisos que los usuarios y también podrán crear eventos y nuevos foros de discusión.
- Y el rol de Administrador permitirá todo lo anterior y la posibilidad de eliminar o bloquear usuarios.

### Épica de la Aplicación:

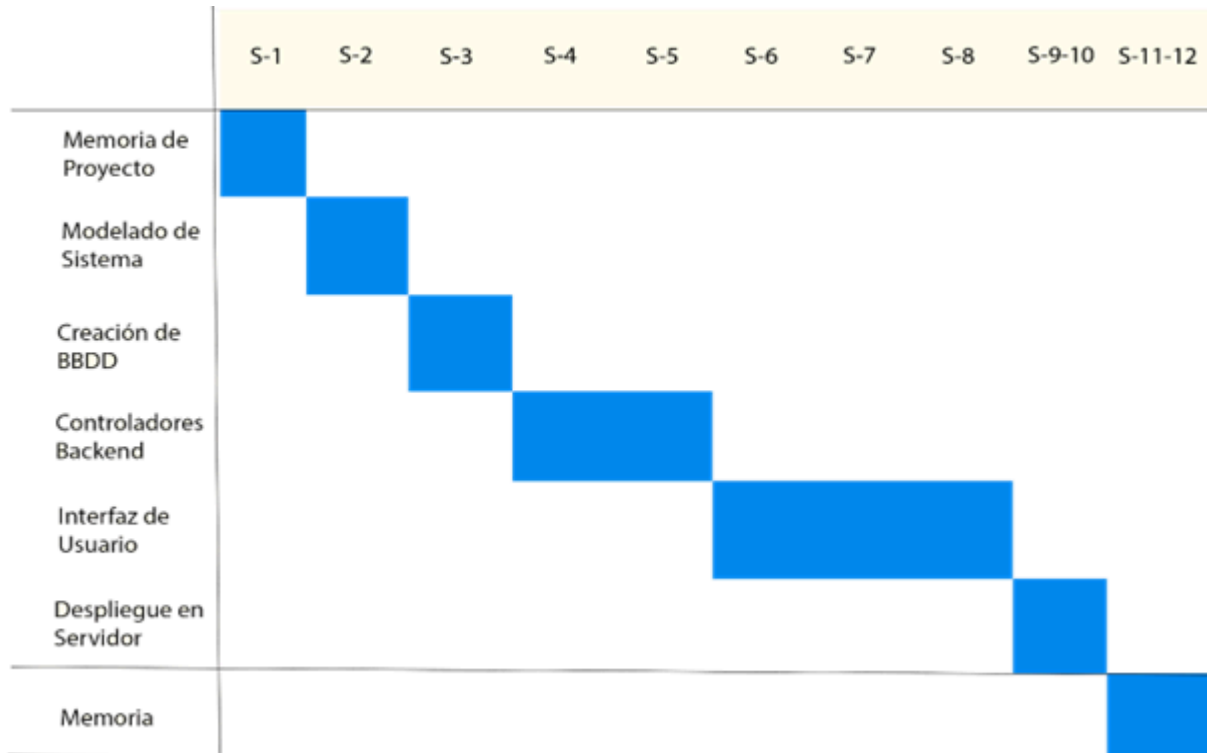
- La aplicación permitirá a los usuarios registrados la creación de nuevos mazos y la impresión de las cartas. Los mazos podrán ser públicos para todos los usuarios o no.
- La aplicación permitirá el diseño personalizados de cartas que podrán almacenarse como publicas o privadas en el perfil del usuario.
- La aplicación contendrá diferentes sistemas de filtrado para tipos de carta, dependiendo de si son cartas de personajes o de librería(cartas que pueden jugar los personajes), y otros filtrados más específicos por diferentes campos: nombre, tipo, title, disciplinas...
- Creación de eventos para gestionar partidas, con un panel personalizado para cada evento que muestre los participantes, las mesas asignadas a cada participante, la posición y los puntos obtenidos en el torneo.
- Listado de mazos y cartas personalizadas (publicas)
- (Si me da tiempo) Foros de discusión para diversos temas del juego.
- La aplicación deberá manejar los errores, mostrando mensajes personalizados para cada caso, implementando dichos mensajes en los modelos.
- La aplicación permitirá a los administradores eliminar o modificar permisos de otros usuarios, en caso de eliminar los usuarios se eliminaran todos los campos asociados.
- Los usuarios podrán eliminar mazos de sus librerías, eliminándolo a su vez de la lista publica en caso de existir.

### Tecnologías y Herramientas:

- **Node js con Express para el Backend.**
- **Base de Datos con MongoDB Atlas**
- **Front con Angular y Bootstrap**
- **Pruebas de Backend con Postman.**
- **Uso de Api externa para traer la info de las cartas:**

<https://static.krcg.org/data/vtes.json>

## Planificación:



## DISEÑO DE LA APLICACIÓN

### ARQUITECTURA DEL SISTEMA

#### Arquitectura física:

Amazon Web Service (AWS) que es uno de los proveedores líderes en servicios de computación basados en la nube, ya que permite seleccionar el sistema operativo, lenguaje de programación, base de datos, y diferentes servicios que voy a necesitar para tener acceso al entorno virtual y así poder cargar el software y los servicios que voy a usar en mi aplicación.

## **Arquitectura Lógica:**

Modelo-Vista Controlador (MVC), es con la que he trabajado y me siento más cómodo, separar las responsabilidades facilitando el mantenimiento y la escalabilidad.

Modelos:

- Cards, CustomCards, deck, event-users, event, foro, image, library, message, post, user.
- En los modelos determinare la estructura y los campos de las tablas para la bbdd.

Controlladores:

- Contienen la lógica para manejar las diferentes campos, como el registro de usuarios, añadir o eliminar eventos o almacenar las imágenes personalizadas de los usuarios.

Routes:

- Contendrá las rutas de mi aplicación a las diferentes paginas o campos y los middlewares en caso de necesitarlos para la autenticación de los usuarios o el almacenamiento de las imágenes con el uso de Multer.

Middlewares:

- Contendrá lógica adicional para reutilizar en diferentes rutas.

Vistas:

- Se realizaran con Angular y mantendrá una organización de carpetas similar al backend.

## **ENTIDADES**

- User: Detalles de los usuarios (id, email,password...)
- Cards: Información sobre cada una de las cartas
- Deck: Conjunto de cartas con el usuario asociado
- customCards: Información sobre cartas personalizadas
- event: Descripcion, fecha de nuevos eventos
- evento-users: Usuarios asociados al evento

- library: Listado con los mazos(decks) de cada usuario
- foro: Información de los diferentes hilos de conversación
- post: Mensajes de los foros

## **HISTORIAL DE USUARIOS**

Los usuarios podrán tener diferentes roles que determinarán sus permisos.

Admin, Colaborador, usuario y invitado.

Invitado:

- La aplicación requiere de registro para visualizarla así que tendré un usuario con el rol de Invitado que solo pueda visualizar.

Usuarios:

- Los usuarios podrán visualizar cartas, construir mazos y guardarlos en su librería, diseñar cartas personalizadas y almacenarlas tanto de forma privada como para todos los usuarios, podrán visualizar los eventos que se han realizado y participar en los foros públicos o a los que hayan sido invitados. También podrán reportar el uso indebido de la aplicación por parte de otros usuarios a los colaboradores o al administrador.

Colaboradores:

- Los colaboradores también podrán crear eventos y nuevos temas de discusión en los foros.
- Bloquear usuarios.
- Eliminar mensajes o cartas personalizadas.

Admin:

- Los administradores podrán eliminar cualquier usuario.
- Cambiar permisos.