**TriWorld**

Dibujo en blanco y negro

Descripción generada automáticamente con confianza media

**CFGS DAW**

**CURSO 2023/2024**

**ÁLVARO BOYERO PLAZA, JOSÉ DANIEL VEGAS MARTÍN**

**COLEGIO CALASANZ SALAMANCA**

**Índice**

[1. Introducción 3](#_Toc166596844)

[2. Descripción de la aplicación. 3](#_Toc166596845)

[3. Plan de empresa 3](#_Toc166596846)

[4. Tecnologías escogidas y justificación 3](#_Toc166596847)

[5. Diseño de la aplicación 3](#_Toc166596848)

[5.1. Diagramas y definición de casos de uso 3](#_Toc166596849)

[5.2. Diagramas de clase 3](#_Toc166596850)

[5.3. Modelo entidad relación 3](#_Toc166596851)

[6. Arquitectura de la aplicación 3](#_Toc166596852)

[6.1. Estructura del proyecto 4](#_Toc166596853)

[6.2. Librerías externas utilizadas 4](#_Toc166596854)

[7. Manual de despliegue 4](#_Toc166596855)

# Introducción

La aplicación consistirá en un juego de un trivial, donde la gente responderá a preguntas, pueden ser de diferentes dificultades y categorías, a mayor dificultad, mayor cantidad de puntos.

El usuario tendrá que registrarse para poder jugar al trivial (mediante una API).

También se hará uso de un ranking donde se podrá ver quienes han ganado más puntos.

# Descripción de la aplicación.

El inicio de la página será el logo de la página en la cabecera y junto a dos botones de inicio sesión y de registrarse, más abajo en mitad de la página habrá un botón de jugar junto a un selector para elegir la categoría que la persona prefiera.

Si el usuario no ha iniciado sesión o no se ha registrado e inmediatamente accede a jugar se le redirigirá al inicio de sesión o registro.

En la página de juego se irá cargando las preguntas de la temática que haya elegido el usuario.

Las preguntas cuanto más difíciles, mayor puntuación, acertar una fácil 1 punto, acertar una de dificultad media 2 puntos y por último en difícil será de 3 puntos.

El ranking estará ordenado por quien ha conseguido más puntos.

En el ranking aparecerán los nombres junto a su puesto actual más la puntuación.

El usuario podrá acceder a la información de su perfil a través de su cuenta. Podrá desactivar su cuenta si el usuario quiere. Además accediendo a su perfil mostrará el listado de partidas y el resultado de puntos que ha ido consiguiendo.

El usuario administrador podrá listar los usuarios, insertar usuarios, actualizar datos de algún usuario y eliminar usuarios.

# Plan de empresa

**TriWorld** es una plataforma en línea que ofrece un juego de trivial interactivo diseñado para entretener y educar a los usuarios de todas las edades.

Nuestro objetivo es proporcionar una experiencia divertida y estimulante que fomente el aprendizaje a través del juego, al tiempo que se promueve la competencia amistosa entre los jugadores.

**TriWorld** se diferenciará de otros juegos de trivial en línea por su interfaz intuitiva, su amplia variedad de preguntas y su enfoque en la educación lúdica.

Usaremos estrategias de promoción en redes sociales para aumentar la visibilidad y atraer usuarios. Colaboraciones con influencers y bloggers de juegos para llegar a audiencias específicas. Estrategias de SEO para mejorar el posicionamiento en los motores de búsqueda y aumentar el tráfico orgánico. Campañas de marketing de contenidos para generar interés y compromiso con la marca.

Nuestra estructura organizativa va a basarse de esta manera:

* **CEO**: Responsable de la dirección estratégica y la toma de decisiones clave.
* **Equipo de Desarrollo**: Encargado del diseño, programación y mantenimiento de la plataforma web y aplicaciones móviles.
* **Equipo de Contenidos**: Responsable de la creación y curación de las preguntas y temáticas de los juegos.
* **Equipo de Marketing**: Encargado de promocionar la plataforma, gestionar redes sociales y estrategias publicitarias.
* **Equipo de Soporte**: Atiende las consultas y problemas técnicos de los usuarios, garantizando una experiencia óptima.

El mercado de los juegos en línea ha experimentado un crecimiento significativo en los últimos años, con una demanda creciente de experiencias interactivas y socialmente conectadas. Existe una tendencia hacia el entretenimiento educativo, donde los usuarios buscan juegos que les permitan aprender mientras se divierten.

Las oportunidades de negocio en el sector de los juegos en línea son prometedoras, con un amplio mercado objetivo que incluye desde jóvenes estudiantes hasta adultos interesados en actividades de ocio intelectual. Además, la posibilidad de ofrecer versiones premium, publicidad integrada y asociaciones con marcas puede generar ingresos adicionales.

TriWorld es un proyecto innovador que combina la diversión de los juegos de trivia con la accesibilidad de una plataforma en línea. Nuestra propuesta de valor radica en la variedad de categorías de preguntas, la interactividad de los desafíos y la posibilidad de competir con amigos y familiares en tiempo real.

Características de Triworld, podemos destacar las siguientes:

* **Interfaz de Usuario Intuitiva**: Diseño atractivo y fácil de usar para una experiencia de usuario óptima.
* **Amplia Variedad de Categorías**: Diversidad de temas para adaptarse a los intereses de diferentes usuarios.
* **Modos de Juego Interactivos:** Opciones para jugar individualmente tranquilamente o de manera competitiva mediante el uso del ranking. Puedes jugar como tú prefieras.
* **Integración Social**: Funcionalidades para compartir resultados, retar a amigos y participar en rankings globales.

TriWorld se compromete a cumplir con todas las regulaciones legales y fiscales aplicables en su jurisdicción, incluyendo el cumplimiento de normativas de protección de datos, derechos de autor y comercio electrónico.

Para la financiación inicial del proyecto TriWorld, se buscarán inversores interesados en apoyar iniciativas innovadoras en el sector de los juegos en línea. Además, se explorarán posibles subvenciones y programas de financiamiento para empresas emergentes en tecnología y entretenimiento.

El guion de trabajo para la elaboración del proyecto TriWorld incluye las siguientes etapas:

* **Investigación de Mercado**: En esta etapa, realizaremos un estudio exhaustivo del mercado de los juegos en línea y específicamente del segmento de juegos de trivia. Analizaremos tendencias actuales, demanda del mercado, competencia, perfiles de usuarios y oportunidades de crecimiento. Esta fase o etapa ya ha sido realizada, y nuestro mayor competidor sería la empresa creadora de Preguntados.
* **Desarrollo de la Plataforma:** Con base en los hallazgos de la investigación de mercado, procederemos al desarrollo de la plataforma TriWorld. Esta fase incluirá la creación de la arquitectura de software, diseño de la interfaz de usuario, programación de funcionalidades clave y configuración de servidores. Nos enfocaremos en garantizar que la plataforma sea robusta, escalable, segura y fácil de usar. Podemos contar de que esta etapa también ya ha sido realizada.
* **Creación de Contenidos**: Nosotros usaremos una API gratuita, Triworld cargará las preguntas de la API las cuales nunca se van a repetir ya que hay múltiples categorías y dentro de esas categorías miles de preguntas.
* **Pruebas y Mejoras**: Una vez completado el desarrollo inicial de la plataforma y los contenidos, procederemos a realizar pruebas exhaustivas. Identificaremos y corregiremos errores, optimizaremos el rendimiento y ajustaremos la experiencia del usuario según los comentarios recibidos durante las pruebas.
* **Lanzamiento y Promoción**: Con la plataforma y los contenidos probados y optimizados, procederemos al lanzamiento oficial de TriWorld. Durante esta fase, implementaremos estrategias de promoción y marketing para generar conciencia y atraer a nuevos usuarios. Utilizaremos una combinación de tácticas de marketing digital, promociones con influencers y anuncios también en otros juegos.
* **Soporte y Mantenimiento**: Una vez lanzada la plataforma, nos comprometeremos a brindar un soporte continuo a nuestros usuarios. Esto incluirá la atención de consultas, solución de problemas técnicos, actualizaciones periódicas de contenido y mantenimiento de la plataforma para garantizar su funcionamiento óptimo.

Por último, se establecerá un sistema de evaluación continua para monitorear el rendimiento de la plataforma, recopilar retroalimentación de los usuarios y realizar ajustes según sea necesario para garantizar la calidad y satisfacción del cliente. La opinión del usuario es importantísima para nosotros y valoraremos cada propuesta.

# Tecnologías escogidas y justificación

* **Spring Boot**: Spring Boot es una opción popular para el desarrollo de aplicaciones web y servicios RESTful por varias razones:
  + Por su facilidad a la hora de configurarse.
  + Productividad mejorada: Spring Boot proporciona un conjunto de herramientas y funcionalidades que ayudan a los desarrolladores a ser más productivos.
  + Spring Boot se basa en el marco de trabajo Spring, lo que significa que hereda todas las características y beneficios de Spring
  + Spring Boot es respaldado por una comunidad activa de desarrolladores. Esto significa que hay una gran cantidad de recursos disponibles, incluida la documentación detallada, tutoriales, ejemplos de código y ayuda en línea.
* **MySQL**: Es un sistema de gestión de BBDD. Este es uno de los más usados y también cuenta con una comunidad activa, por tanto es de gran utilidad ya que podremos encontrar una gran cantidad de recursos.
* **Bootstrap**: Bootstrap es un marco de trabajo de front-end que proporciona herramientas y componentes CSS y JavaScript preestilizados para el desarrollo rápido y fácil de sitios web y aplicaciones web responsivas. Ofrece una amplia gama de componentes listos para usar, como botones, formularios, barras de navegación, etc.

* **Vue**: Nos ha sido útil usar vue sobre todo por las vistas y componentes que hemos ido creando y por tener su característica principal de html, css y javascript en un mismo archivo.
* **Open Trivia DB**: API gratuita la cual nos da una amplia gama de opciones para elegir la categoría, cuántas preguntas en total y en qué dificultad queremos estas preguntas. Esta API la hemos usado en la parte del front-end.

# Diseño de la aplicación

## Diagramas y definición de casos de uso

Diagrama

Descripción generada automáticamente

## Diagramas de clase

Diagrama

Descripción generada automáticamente

## Modelo entidad relación

Diagrama

Descripción generada automáticamente

# Arquitectura de la aplicación

## Estructura del proyecto

TriWorld cuenta con una base de datos propia, su parte front-end y su parte back-end.

Nuestra base de datos contiene todos los usuarios, dificultades, categorías, contraseñas, etc. En cuanto al frontend, usamos vue la cual maneja vistas y componentes, maneja también una api para cargar las preguntas y para acabar hablaremos del back-end la cual es una API-REST que se encarga de hacer un CRUD de los usuarios, roles, categorías y del ranking.

En la parte front se encuentra dividida de esta forma:  
Hemos usado Vue la cual tenemos vistas y varios componentes los cuales vamos a comentar ahora.

En cuanto a las **vistas** tenemos:

* **HomeView**: Contiene la página principal de TriWorld, el botón de iniciar partida, el menú de navegación para ir a inicio, ranking y noticias, acceso a la parte de administración y para crear tu cuenta, en el footer las redes sociales, y políticas de protección de datos y cookies y el aviso legal.
* **RankingView**: Contiene la puntuación de los usuarios ordenado de mayor a menor gracias a la puntuación, en caso de empate a puntos, quien lo haya hecho a mayor dificultad estará por delante. Cabe añadir que cada usuario tiene su perfil.
* **NoticiasView**: En esta página se encuentra el apartado de noticias relacionadas con TriWorld.
* **CuentaView**: En esta página nos encontramos la vista del perfil de cada usuario. Aquí los usuarios podrán añadir su descripción y comprobar el número de partidas y los puntos, también pueden añadir su foto de perfil.
* En la carpeta usuarios nos encontraremos con:
  + **AdminsView**: Esta vista se encuentra el listado de usuarios, en este apartado se podrá añadir, actualizar y eliminar los usuarios que el usuario administrador desee.
  + **InsertarUsuarioView**: En esta vista se accede una vez pulses el botón de añadir usuario, será un formulario donde se deberá introducir los datos que se quiera insertar.
  + **ActualizarUsuarioView**: En esta vista se accede una vez le des al botón de editar, se recoge el id del usuario y habrá un formulario para actualizar los datos del usuario.
* **PreguntasView**: En esta página nos encontraremos con las preguntas una vez elegida ya la dificultad, en la vista aparecerá también la puntuación que llevemos. Una vez terminadas las preguntas aparecerá un pequeño formulario donde el mensaje será distinto dependiendo de la puntuación.

Y en cuanto a los **componentes** nos encontraremos con lo siguiente:

* **SeleccionDificultad**: Este componente va ante de la vista de las preguntas, cuando pulsamos el botón de iniciar partida aparecerá este componente el cual tendrá las dificultades, dependiendo de cual elijamos de las dificultades en PreguntasView las preguntas tendrán esta dificultad.
* **AvisoLegal**: Este componente lo único que contiene es el aviso legal de la web la cual informamos a los usuarios una serie de datos relacionados con el titular y el funcionamiento de una página web.
* **PoliticaProteccionDatos**: En esta vista encontraremos la protección de datos donde informamos el uso de los datos que nos ha aportado el usuario.
* **PoliticaCookies**: En esta vista encontraremos la política de cookies que usamos donde informamos del distinto tipo de cookies que usamos.

Procedamos con la parte del **back-end**: Se trata de una arquitectura REST, su función es escuchar las peticiones que envía el cliente y responderle con los datos de la petición. Estará estructurado de esta forma:

* **ProyectoTrivialApplication**: Esta clase es la clase principal donde se inicia el back-end.

La anterior mencionada sería la clase main, ahora iremos con la parte de configuración, la cual tenemos la clase que trata los errores del CORS.

* **CorsConfig**: Da permisos al front para que reciba peticiones del back-end y trata los errores que pueda ocasionar el CORS.

De momento eso sería la parte de configuración, ahora pasamos con los controllers:

* **UserRestController**: Responde a las llamadas para obtener, crear, actualizar y borrar datos de la tabla de usuario.
* **RankingRestController**: Responde a las llamadas para obtener, crear, actualizar y borrar datos de la tabla de ranking.
* **RolesRestController**: Responde a las llamadas para obtener, crear, actualizar y borrar datos de la tabla de roles de los usuarios.
* **CategoriasRestController**: Responde a las llamadas para obtener, crear, actualizar y borrar datos de la tabla de categorías.

Estos son los DTOS:

* **CategoriasDTO**.
* **RankingDTO.**
* **RolesDTO.**
* **UsersDTO.**

A continuación las **entities**:

* **CategoriaEntity.**
* **RankingEntity.**
* **RolEntity.**
* **UserEntity.**

A continuación los **repositories**:

* **ICategoriasRepository.**
* **IRankingRepository.**
* **IRolesRepository.**
* **IUsersRepository.**

Ya por último por comentar de todo esto sería la carpeta de **resources**, donde hemos añadido ahí la url de la base de datos con el usuario y contraseña que debe usar y otras herramientas aparte para el buen funcionamiento de la aplicación.

## Librerías externas utilizadas

Las librerías que hemos usado han sido las siguientes:

* Bootstrap: Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.
* Axios: Axios es una biblioteca JavaScript que simplifica la realización de solicitudes HTTP desde el navegador o desde Node.js. Proporciona una interfaz fácil de usar para enviar solicitudes AJAX y manejar respuestas de manera eficiente.

# Manual de despliegue

Despliegue usando **Docker**:

Para desplegarlo con Docker, necesitaremos estas 3 imágenes:

* **alvaroboyero/imgbbdd .**
* **alvaroboyero/imgback .**
* **alvaroboyero/imgfront .**

Con estas tres imágenes, las bajaremos haciendo pull de ellas:

* **docker pull alvaroboyero/imgbbdd:latest**
* **docker pull alvaroboyero/imgback:latest**
* **docker pull alvaroboyero/imgfront:latest.**

Una vez bajadas las imágenes, ejecutaremos el siguiente comando:

* **docker -compose -up –build**

Con esto ejecutaremos las 3 imágenes y a su vez arrancará el contenedor. El back-end escuchará el puerto 8080 y el front-end el puerto 8081.

Despliegue usando **Github**:

Suponiendo que nuestro repositorio de Github ya ha sido clonado en el dispositivo, desplegaremos la aplicación de manera local.  
Lo primero y más importante es tener la base de datos ya preparada con el script de la BBDD. Una vez tengamos esto, abrir la aplicación de Eclipse Workspace para abrir nuestro proyecto del back-end e iniciarlo. Se iniciará en el puerto 8080. Iniciado así escuchará las peticiones que le haga el cliente.

Y por parte del front-end, abriremos el Visual Studio Code o cualquier editor de código que sea parecido a Visual (IntelliJ IDEA por ejemplo), abrimos la terminal del editor de código y tendremos que instalar las dependencias del proyecto, para ello usar este comando. **npm install** y ejecutamos la parte del front-end con **npm run serve**.

En este caso al estar ocupado el puerto 8080, se asigna automáticamente el puerto 8081. En caso de que queramos acceder a la página web, en la terminal se nos mostrará con cuales IPs podemos abrir la página.

Si por algún casual no sale en la terminal, escribir en la barra de navegador **localhost:8081**