

# TRIVIALITY®

**Neil Vargas Calle**

**Daniel Jiménez Parreño**

**DAW 2**



## **COPYRIGHT**

MIT License

Copyright (c) 2024 Neil Vargas Calle, Daniel Jiménez Parreño

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.





## **Resumen del proyecto:**

**Triviality** es una plataforma interactiva basada en un juego de preguntas y respuestas estilo trivia que busca unir entretenimiento y aprendizaje en un entorno competitivo y educativo. El proyecto tiene como objetivo principal proporcionar una experiencia divertida y educativa en la que los usuarios puedan ampliar sus conocimientos sobre diversas áreas como ciencia, historia, deportes y entretenimiento, mientras compiten y miden su desempeño frente a otros jugadores.

El sistema integra elementos de gamificación, como perfiles personalizados, logros y clasificaciones, para incentivar la participación y mejorar la experiencia del usuario. Además, emplea tecnologías modernas como React para un frontend dinámico y MySQL como base de datos, junto con la API Open Trivia Database (OTDB) y LibreTranslate para obtener preguntas en tiempo real en diferentes idiomas.

Los principales resultados esperados incluyen una aplicación estable y atractiva que fomente el aprendizaje a través del juego, motive la interacción social y permita expandir funcionalidades en futuras iteraciones.

## **Palabras clave:**

- Trivia interactiva
- Aplicación web dinámica
- Interfaz
- Desarrollo full-stack
- Gamificación



## **Abstract:**

### **ENGLISH**

*Triviality* is an interactive platform designed as a trivia-based question-and-answer game that merges entertainment and learning in a competitive and educational environment. The project's main goal is to provide a fun and educational experience where users can expand their knowledge in various areas such as science, history, sports, and entertainment while competing and benchmarking their performance against other players.

The system integrates gamification elements, such as personalized profiles, achievements, and leaderboards, to encourage participation and enhance the user experience. It also uses modern technologies like React for a dynamic frontend and MySQL as the database, along with the Open Trivia Database (OTDB) and LibreTranslate APIs to fetch real-time questions in different languages.

The primary expected outcomes include a stable and engaging application that promotes learning through play, encourages social interaction, and allows for functionality expansion in future iterations.



## **CATALÁN**

*Triviality* és una plataforma interactiva basada en un joc de preguntes i respostes tipus trivia que combina entreteniment i aprenentatge en un entorn competitiu i educatiu. L'objectiu principal del projecte és oferir una experiència divertida i educativa en què els usuaris puguin ampliar els seus coneixements en diverses àrees com ciència, història, esports i entreteniment, mentre competeixen i comparen el seu rendiment amb altres jugadors.

El sistema integra elements de gamificació, com perfils personalitzats, assoliments i classificacions, per incentivar la participació i millorar l'experiència de l'usuari. A més, utilitza tecnologies modernes com React per a un frontend dinàmic i MySQL com a base de dades, juntament amb les API d'Open Trivia Database (OTDB) i LibreTranslate per obtenir preguntes en temps real en diferents idiomes.

Els principals resultats esperats inclouen una aplicació estable i atractiva que fomenti l'aprenentatge a través del joc, motivi la interacció social i permeti expandir funcionalitats en futures iteracions.

# Índice

1. Introducción	7
1.1 Contexto	7
1.2 Justificación	7
1.3 Objetivos	9
1.3.1 Objetivos generales:	9
1.3.2 Objetivos específicos:	9
1.5 Metodología de trabajo	10
2. Descripción del proyecto	11
2.1 Análisis de requisitos	11
2.1.1 Requisitos funcionales	11
2.1.2 Requisitos no funcionales:	11
2.2 Tecnologías	12
2.2.1 Comparativa de las tecnologías valoradas:	12
2.2.1.1 Base de datos	12
2.2.1.2 Frontend	14
2.2.1.3 BackEnd	15
2.2.2 Tecnologías escogidas	17
2.2.2.1 Base de datos	17
2.2.2.2 FrontEnd	18
2.2.2.3 BackEnd	23
2.2.2.4 APIs Externas	24
2.3 Estructura del proyecto	26
2.4 Descripción de los componentes	27
2.4.1 Login	28
2.4.2 SignUp	29
2.4.3 Home	30
2.4.4 Header	30
2.4.5 Footer	31
2.4.6 Jugar	31
2.4.7 About Us	32
2.4.8 Perfil	33
2.4.9 Trivia Game	34
2.4.9.1 Configuración Inicial (Pantalla de Preparación)	34
2.4.9.2 Interfaz de Carga (Transición)	35
2.4.9.3 Núcleo del Juego (Pantalla de Preguntas)	36
2.4.9.4 Resultados Finales (Resumen de Partida)	37
2.4.10 Clasificación	38
2.5 Definición de las funcionalidades	39
2.5.1 Sistema de Autenticación y Registro	39
2.5.2 Gestión de Perfiles de Usuario	40



2.5.3 Sistema de Juego de Trivia	41
2.5.4 Sistema de Clasificaciones	42
2.5.5 Sistema de Internacionalización	42
2.5.6 Características Técnicas	45
2.5.7 Interfaz de Usuario	45
3 Herramientas	46
3.1 TRELLO	46
3.2 GITHUB	46
3.3 VISUAL STUDIO CODE	47
3.4 DBEAVER	47
4 Conclusiones	47
4.1 Conclusiones generales del proyecto	47
4.2 Consecucion de los objetivos	48
4.3 Valoración de la metodologia i planificació	48
4.4 Visión de futuro	49
5. Anexos	49
DIAGRAMAS	50
TRÁMITES DE PUESTA EN MARCHA	57
<b>6. Bibliografia</b>	<b>60</b>
<b>7. Agradecimientos</b>	<b>61</b>



## 1. Introducción

En un mundo en constante transformación digital, las plataformas educativas han evolucionado para incorporar elementos lúdicos y fomentar el aprendizaje a través del entretenimiento. Juegos como los trivias se han convertido en herramientas poderosas para estimular el razonamiento y la adquisición de conocimientos de forma relajada y accesible.

*Triviality* se posiciona como una solución innovadora en este ámbito, ofreciendo a los usuarios una plataforma donde pueden aprender y competir de manera dinámica y divertida.

### 1.1 Contexto

El proyecto surge de la necesidad de crear una herramienta que combine aprendizaje y diversión, adaptándose a las tendencias actuales de la gamificación y la educación interactiva. *Triviality* no solo busca llenar este vacío, sino también ofrecer un entorno interactivo que motive la participación continua y el desarrollo del conocimiento general.

### 1.2 Justificación

En el contexto actual, donde la gamificación educativa es una tendencia en crecimiento (ej: Duolingo, Kahoot! tienen millones de usuarios), *Triviality* surge como respuesta a tres necesidades clave del mercado:

- Falta de plataformas híbridas (educación + entretenimiento):

Muchas herramientas educativas son rígidas o aburridas, mientras que los juegos de trivia existentes (ej: Trivia Crack) priorizan el ocio sobre el aprendizaje. *Triviality* combina ambos enfoques, ofreciendo contenido didáctico en un formato competitivo y adictivo.





- Personalización y accesibilidad:

Los usuarios demandan experiencias adaptadas a sus intereses y lenguajes. Al integrar APIs como LibreTranslate, Triviality supera la barrera idiomática, algo crítico en mercados globalizados. Además, su diseño responsivo lo hace accesible desde cualquier dispositivo.

- Oportunidad en un nicho en expansión:

El mercado de e-learning gamificado está en constante expansión. Triviality se posiciona como una alternativa escalable, con potencial para monetizarse mediante suscripciones premium o publicidad contextual (ej: preguntas patrocinadas por marcas educativas).



## 1.3 Objetivos

### 1.3.1 Objetivos generales:

- Diseñar y desarrollar una aplicación web que combine elementos de aprendizaje y entretenimiento en un formato de trivia.
- Incentivar la interacción social mediante competencias y clasificaciones globales o temáticas.

### 1.3.2 Objetivos específicos:

- Crear un sistema de perfiles personalizados, que incluyan niveles, y estadísticas detalladas.
- Implementar preguntas dinámicas y variadas en categorías como ciencia, deportes, entretenimiento, historia y tecnología (en diferentes idiomas).
- Desarrollar una interfaz intuitiva y atractiva con tecnologías modernas.
- Integrar de forma satisfactoria elementos propios de un proyecto de desarrollo web.

## 1.5 Metodología de trabajo

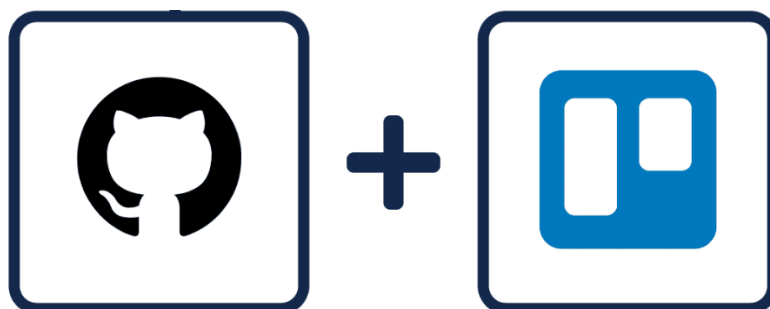
Para el desarrollo de *Triviality*, se adoptará una metodología ágil, que permita una planificación flexible:

La organización y el seguimiento del trabajo se realizarán utilizando **Trello**, una herramienta de gestión de proyectos basada en tableros, que facilita:

- La asignación de tareas a los miembros del equipo.
- La priorización de actividades mediante etiquetas y categorías.
- El seguimiento del avance de cada tarea en las diferentes fases del proyecto (por ejemplo: "Pendiente", "En progreso", "Completado").
- La visualización de todo el flujo de trabajo de manera clara e intuitiva.

Además, se integrará GitHub como repositorio para el control de versiones y colaboración en el desarrollo del código, asegurando un trabajo coordinado y evitando conflictos en los archivos del proyecto.

Con el uso combinado de Trello y GitHub, se busca maximizar la eficiencia del equipo, mantener una comunicación clara y garantizar que el proyecto se desarrolle dentro de los plazos previstos.





## 2. Descripción del proyecto

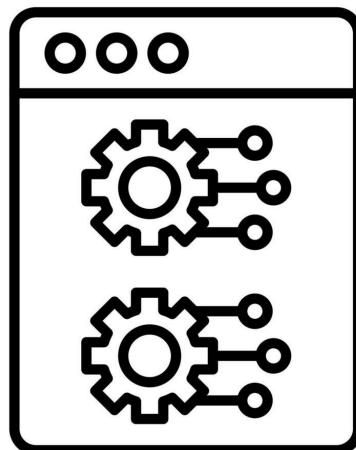
### 2.1 Análisis de requisitos

#### 2.1.1 Requisitos funcionales

- Registro y autenticación de usuarios, incluyendo cifrado de contraseñas.
- Gestión de perfiles personalizados con niveles, y estadísticas con gráficos.
- Generación de preguntas dinámicas con selección de dificultad.
- Clasificaciones globales por puntos acumulados.
- Traducción de la página y del juego.

#### 2.1.2 Requisitos no funcionales:

- Tiempo de respuesta inferior a 10 segundos para consultas básicas.
- Diseño responsivo, compatible con dispositivos móviles y escritorio.





## 2.2 Tecnologías

### 2.2.1 Comparativa de las tecnologías valoradas:

Durante el desarrollo del proyecto, se consideró el uso de las siguientes tecnologías para las tareas de gestión de bases de datos, así como para el desarrollo del frontend y backend de la aplicación.

#### 2.2.1.1 Base de datos

Característica	MySQL	PostgreSQL
Licencia	GPL	Licencia MIT
Rendimiento	Mejor en operaciones simples y lectura	Mejor en cargas complejas
Escalabilidad	Fácil de configurar (replicación)	Mejor concurrencia
Soporte SQL	SQL básico, menos estricto	Soporta estándares SQL completos
Facilidad de Uso	Sencillo, más fácil para principiantes	Más complejo, pero flexible y potente
Extensiones	Limitadas	Gran cantidad de extensiones, ideal para usos avanzados
NoSQL	Soporte básico para JSON	Excelente soporte para NoSQL (JSONB, arrays, etc.)

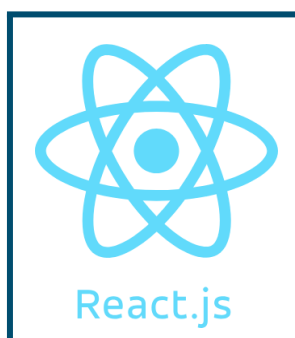
En este punto, realizando un vistazo objetivamente para este proyecto el equipo no ha tenido demasiadas dudas en cuanto a su elección. Teniendo en cuenta que no requiere demasiada exigencia en cuanto a las operaciones con este campo, se optó desde el principio por MySQL. Por su sencillez en cuanto a su uso y ejecución en operaciones, ha sido la elección más acertada. Esto debido a que mucha de la información del proyecto proviene de fuentes externas como una API.



### 2.2.1.2 Frontend

Característica	React	Angular
Licencia	MIT (muy permisiva)	MIT (muy permisiva)
Rendimiento	Excelente con Virtual DOM	Muy bueno, aunque algo más pesado en apps complejas
Arquitectura	Flexible, basada en componentes	Estructurada, basada en módulos y servicios
Lenguaje base	JavaScript + JSX	TypeScript (obligatorio)
Facilidad de uso	Intuitivo para quienes ya conocen JavaScript	Más rígido, pero con muchas funcionalidades integradas
Escalabilidad	Alta, pero depende de cómo lo organices	Muy alta, ideal para grandes equipos y empresas
Data Binding	Unidireccional	Bidireccional
Tipo	Biblioteca de UI	Framework completo

En este aspecto, la decisión ha sido más complicada ya que en su momento el conocimiento sobre ambas tecnologías era aún muy difuso. Ahora bien, recogiendo información, por su arquitectura y lo intuitivo en cuanto a concepto se decidió por React.





### 2.2.1.3 BackEnd

Característica	Laravel (PHP)	Spring Boot (Java)
Licencia	MIT	Apache 2.0
Lenguaje base	PHP, ampliamente usado y fácil de aprender	Java, robusto pero más complejo
Curva de aprendizaje	Suave, ideal para desarrolladores web	Más técnica, requiere dominio de Java y arquitectura Spring
Rendimiento	Rápido y suficiente para la mayoría de proyectos web	Alto rendimiento en sistemas complejos y concurrentes
Escalabilidad	Adecuada para proyectos medianos y grandes	Excelente para arquitecturas empresariales complejas
Productividad	Muy alta gracias a su sintaxis clara y herramientas integradas	Alta, aunque con mayor configuración inicial
Facilidad de uso	Muy amigable, centrado en el desarrollador	Robusto, pero más formal y estructurado
Despliegue	Sencillo, compatible con entornos tradicionales y modernos	Generalmente orientado a entornos empresariales (Docker, Kubernetes)

La selección del backend es una decisión crítica para el proyecto, ya que implementar un lenguaje o framework incompatible con las funcionalidades requeridas podría resultar contraproducente, llegando incluso a estancar el desarrollo por limitaciones técnicas.



Tras un análisis cuidadoso, se ha optado por Laravel (PHP) como solución óptima, principalmente por su capacidad para gestionar eficientemente la conexión con APIs externas y su robustez en el funcionamiento del sistema.





## 2.2.2 Tecnologías escogidas

### 2.2.2.1 Base de datos

#### **MySQL:**

MySQL es uno de los instrumentos más empleados a nivel global para administrar bases de datos de código abierto. La base de MySQL se encuentra en SQL (Structured Query Language), lenguaje particular creado para interactuar con bases de datos relacionales. Con él se puede consultar, alterar, suprimir o gestionar información de forma exacta.

Dentro del marco de *Triviality*, MySQL emerge como la opción perfecta gracias a su magnífica habilidad para gestionar relaciones entre datos. MySQL no solo ofrece un desempeño excelente en estas operaciones relacionales, sino que también brinda características que se aprovechan activamente en el proyecto: funciones de agregación (SUM y AVG) para calcular estadísticas de juego como puntuaciones totales y tiempos medios de respuesta, transacciones para garantizar la integridad de los datos al guardar los resultados de las partidas, y un sistema de consultas que permite gestionar eficientemente el historial de juegos y logros de los usuarios. La integración con Eloquent de Laravel simplifica estas operaciones y permite mantener un código más limpio y mantenible.



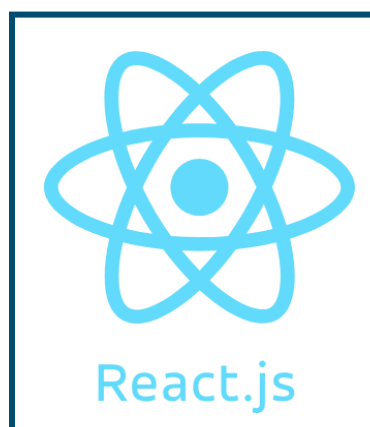
### 2.2.2.2 FrontEnd

#### **React:**

React es una librería de JavaScript creada por Meta, que facilita la creación eficaz y modular de interfaces de usuario. Su metodología basada en componentes permite segmentar una aplicación en secciones reutilizables, cada una con su propio razonamiento y estado, lo que optimiza la organización y el mantenimiento del código.

Se escoge React para *Triviality* debido a su habilidad para gestionar actualizaciones en tiempo real sin tener que recargar toda la página, un aspecto crucial en un juego enérgico como este. Su sistema de administración de estado y su apoyo para la internacionalización se ajustan perfectamente a las demandas del proyecto, tales como presentar preguntas o calificaciones.

Por tanto, en *Triviality*, gracias a esta tecnología, se ha diseñado toda la página web, destacando elementos concretos como el sistema de puntuación, la representación de preguntas y las estadísticas del juego.





## TypeScript:

TypeScript es un lenguaje de programación que extiende las capacidades de JavaScript, incorporando nuevas herramientas que ayudan a escribir código más robusto, claro y menos propenso a errores. Gracias a estas mejoras, trabajar con TypeScript facilita la creación de aplicaciones más ordenadas y fáciles de mantener.

Dado que TypeScript es un superconjunto de JavaScript, cualquier código escrito en JS funciona perfectamente dentro de un proyecto TypeScript. Sin embargo, no sucede lo mismo en sentido inverso: el código en TypeScript necesita ser transformado a JavaScript antes de poder ejecutarse en un navegador, ya que estos solo entienden JavaScript. Este proceso se conoce como “transpilación”, y permite que el código TS sea convertido automáticamente en un formato que los navegadores pueden interpretar. En el desarrollo de *Triviality*, TypeScript ha ofrecido ventajas clave al permitir una programación más estructurada y segura.



## Bibliotecas complementarias:

**React Router 6** es una librería empleada para gestionar la navegación y las rutas en una aplicación React. Facilita segmentar la interfaz en distintas



páginas o vistas sin la necesidad de recargar el navegador, reproduciendo una experiencia parecida a la de una aplicación local. Esta herramienta te permite establecer rutas tanto estáticas como dinámicas, además de rutas anidadas que facilitan una mejor organización de la aplicación. La versión 6 de React Router implementó una sintaxis más intuitiva y optimizaciones en el desempeño, lo que simplificó aún más la administración de rutas.



React Intl

**React-Intl** se centra en la internacionalización de las aplicaciones React. Facilita la traducción de textos y la estructuración de elementos como fechas, números y divisas de acuerdo a las normas del idioma y la región del usuario. Esta biblioteca resulta particularmente

beneficiosa cuando se busca proporcionar la interfaz en varios idiomas, tales como español, inglés y catalán, entre otros. Además, contribuye a la organización del código al centralizar todas las traducciones, lo que facilita su mantenimiento y actualización. Gracias a esto en *Triviality* se ha podido implementar la internacionalización de la página web en español, catalán e inglés.

**FontAwesome** es una reconocida biblioteca de iconos que ofrece una amplia gama de elementos gráficos preparados para ser utilizados. Estos





emblemas son escalables y a medida, lo que implica que puedes modificar su tamaño, color y estilo de manera sencilla con CSS. Debido a su uniformidad visual, contribuye a mantener una identidad gráfica homogénea durante toda la aplicación. En React, se incorpora a través de paquetes particulares que hacen más sencilla la incorporación de iconos directamente en los elementos. En *Triviality* se ha usado esta biblioteca para crear una interfaz de usuario más atractiva e intuitiva.



**Axios** es una herramienta muy útil para realizar peticiones HTTP, tanto para obtener como para enviar datos a un servidor o API.

Basada en promesas, Axios permite manejar las respuestas y errores de manera eficiente y clara dentro de una aplicación React. Una de sus ventajas es su capacidad para configurar interceptores, lo que permite gestionar cabeceras, autenticación o respuestas globales de forma centralizada. Es una opción muy popular por su simplicidad y versatilidad a la hora de conectar una aplicación con servicios externos.

**Chart.js** es una librería enfocada en la generación de diagramas visuales e interactivos en programas web. Facilita la exhibición de información en diversos formatos como gráficos de barras, líneas, gráficos de pastel, radar, entre otros, simplificando la interpretación visual de las



estadísticas. Pese a estar redactada en JavaScript, su integración con React es sencilla mediante bibliotecas como react-chartjs-2, que actúan como un enlace entre ambos ambientes. Es particularmente beneficioso en dashboards o paneles donde resulta crucial exponer datos numéricos de manera clara y atractiva. En *Triviality* se ha implementado esta librería para ofrecer una visualización de las estadísticas de usuario de una forma más atractiva.



## CSS Moderno:

El CSS moderno incorpora métodos y recursos sofisticados que simplifican la construcción de interfaces web dinámicas, asequibles y totalmente flexibles. Herramientas como flexbox y grid para una organización eficaz de los elementos, variables CSS que aceleran la administración de estilos, efectos transitorios y animaciones delicadas para potenciar la interacción, y diseños responsivos que se adaptan a cualquier aparato.

En *Triviality*, se utilizan estas habilidades del CSS moderno para proporcionar una experiencia visual pulida y dinámica. Además, se aplican media queries que aseguran una excelente adaptabilidad, mejorando el rendimiento tanto en ordenadores como en aparatos móviles. Estas normas posibilitan que los elementos ajusten su ubicación, dimensiones o aspecto de acuerdo al tamaño de la pantalla, garantizando que *Triviality* sea fácil de entender y disponible en cualquier plataforma.





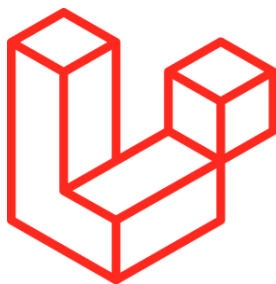
### 2.2.2.3 BackEnd

#### **Laravel:**

Laravel es un framework de PHP moderno que ofrece una base sólida y bien organizada para la creación de aplicaciones web. Su estructura fundamentada en el patrón Modelo-Vista-Controlador (MVC) simplifica la estructuración del código, distinguiendo la lógica empresarial de la capa de presentación, y proporcionando una sintaxis nítida y expresiva para labores habituales como el enrutamiento, las sesiones y la autenticación.

Se utiliza Laravel en *Triviality* por su integración natural con bases de datos mediante Eloquent ORM, que permite trabajar de forma intuitiva con modelos como usuarios, preguntas o categorías. Además, se aprovechan herramientas como migraciones y seeders para mantener la coherencia de la base de datos en distintos entornos, así como su sistema de internacionalización para soportar múltiples idiomas dentro del juego.

Gracias a Laravel, en *Triviality* se implementa una API REST segura mediante tokens JWT, se protegen rutas con middleware, y se optimiza el rendimiento con caché y colas de trabajo. Todo ello garantiza una comunicación fluida con el frontend y una experiencia de usuario rápida y consistente durante las partidas.



# Laravel





#### 2.2.2.4 APIs Externas

En *Triviality*, las APIs son fundamentales para el funcionamiento de la aplicación, actuando como puentes de comunicación entre diferentes partes del sistema. Nuestra aplicación utiliza una API interna desarrollada con Laravel que sirve como intermediario entre el frontend y las APIs externas. Esta API interna se encarga de gestionar la autenticación, procesar datos y manejar el almacenamiento en caché. Además, se integraron dos APIs externas principales: una para traducciones y otra para obtener preguntas de trivia. La API de traducción, se encarga de traducir las preguntas y respuestas a diferentes idiomas, mientras que la API de trivia proporciona el contenido principal del juego con diferentes categorías y niveles de dificultad.

El sistema de caché implementado para las traducciones y preguntas mejora significativamente el rendimiento, proporcionando respuestas rápidas a los usuarios. La seguridad está garantizada a través de una autenticación centralizada y una validación rigurosa de datos. Además, la flexibilidad del sistema nos permite integrar fácilmente nuevas APIs y adaptarnos a diferentes fuentes de datos, lo que es crucial para mantener la aplicación actualizada y escalable.

Las APIs son, en esencia, el corazón de *Triviality*, permitiendo una comunicación fluida entre componentes y servicios, y facilitando la creación de una experiencia de usuario rica y dinámica. Esta arquitectura no solo mejora la eficiencia del sistema sino que también permite una fácil expansión y mejora de las funcionalidades en el futuro.



Aquí tenemos las APIs seleccionadas para mejorar el proyecto:

**Open Trivia Database (OTDB):**

- Fuente principal de preguntas de trivia.
- Más de 4,000 preguntas verificadas.
- 3 niveles de dificultad.
- Formato estandarizado JSON.
- Sistema de tokens para evitar repeticiones.



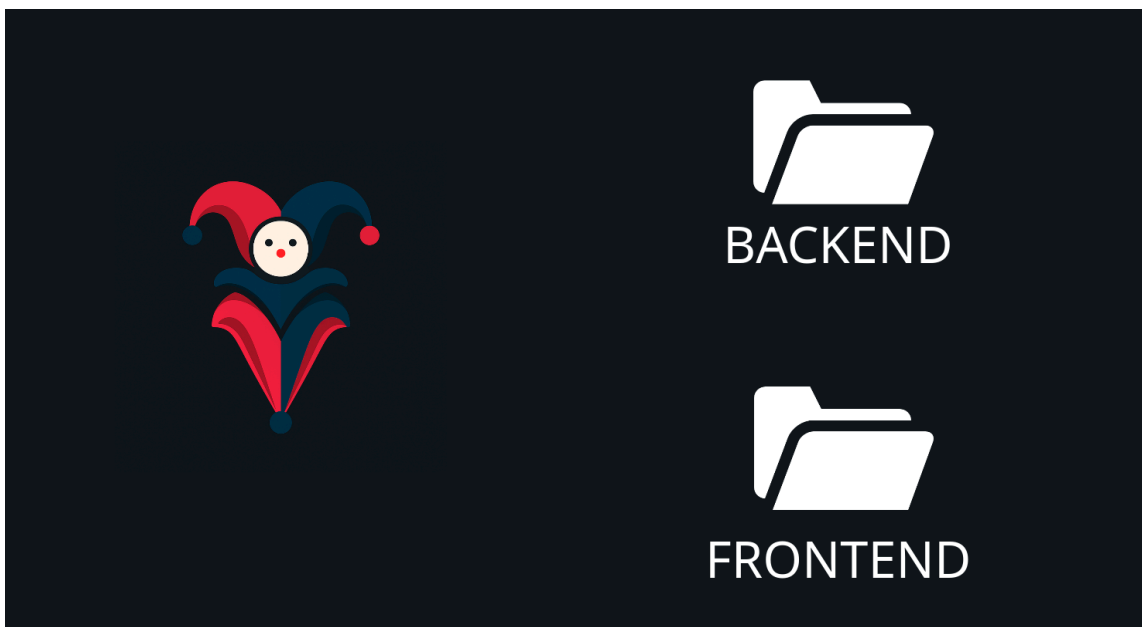
**LibreTranslate:**

- Servicio de traducción automática de código abierto.
- Soporte para múltiples idiomas.



## 2.3 Estructura del proyecto

*Triviality* es una aplicación web de preguntas y respuestas (trivia) desarrollada con una arquitectura moderna de dos partes. El backend se ha desarrollado con Laravel, un marco de trabajo PHP que gestiona toda la lógica, la verificación de usuarios y la administración de la base de datos MySQL. La codificación frontend se realiza en React con TypeScript, ofreciendo una interfaz de usuario interactiva y de respuesta. La interacción entre ambos lados se lleva a cabo mediante una API REST, en la que el backend muestra endpoints para la asignación de usuarios, la autenticación y la administración de las preguntas de trivia. El sistema emplea Laravel para la verificación basada en tokens, garantizando así la seguridad de las solicitudes. La aplicación está concebida para ser escalable y mantenible, alineándose con las prácticas más óptimas de desarrollo web actual.





### API REST:

- Endpoints estructurados para:
  - `/api/auth` (autenticación).
  - `/api/questions` (gestión de preguntas).
- Formato JSON en solicitudes y respuestas.

### Scripts Bash:

- Automatización de la instalación de dependencias.
- Verificación del entorno de desarrollo (PHP, Node.js, servicios Docker).

### Docker:

- Contenedorización opcional para LibreTranslate.

### Ventajas de la Arquitectura

- **Escalabilidad:** Separación clara entre frontend y backend.
- **Mantenibilidad:** Código modular (patrón MVC en Laravel, componentes en React).
- **Seguridad:** Tokens de API y validación estricta.

## 2.4 Descripción de los componentes

El proyecto dispone de una amplia gama de componentes, sin embargo, no es práctico ni imprescindible especificar cada uno de estos. En cambio, se decide detallar los distintos estados de la aplicación, dado que es en estos donde se agrupan los elementos más significativos y representativos del sistema:



## 2.4.1 Login

Idioma: Español

**Iniciar Sesión**

Email

Contraseña

INICIAR SESIÓN

¿No tienes cuenta? [Regístrate](#)

Este componente trata sobre la pantalla de inicio de sesión del proyecto, esta es la puerta principal donde los usuarios pueden acceder a sus cuentas existentes.

Como todo Login este es un formulario, donde se debe poner un mail y su contraseña pertinente. Por ende se aplica una obligatoriedad en estos campos, ya que sin esos datos no puedes iniciar sesión. Si hay un error sobre si no coincide el mail o la contraseña, salta la alarma de credenciales inválidas. Ahora bien, si se añade la información correctamente, se redirige automáticamente a la home.



## 2.4.2 SignUp

Idioma: Español

**Registro**

Nombre

Email

Contraseña

Confirmar Contraseña

**REGISTRARSE**

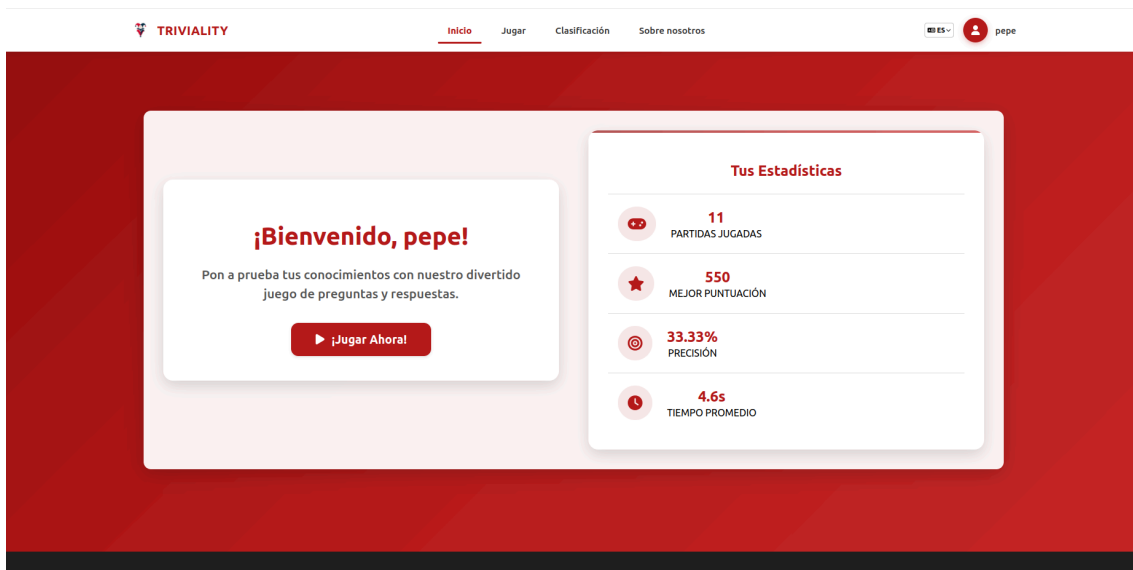
¿Ya tienes cuenta? [Iniciar Sesión](#)

Este componente es de los más importantes ya que abre la aplicación a cualquier nuevo usuario que quiera jugar y aprender con este proyecto.

Este también es un formulario donde se tienen cuatro campos, el nombre de usuario (aparece dentro), un mail, la contraseña y la confirmación. La confirmación se hace para evitar que el usuario se equivoque al guardar su contraseña y así evitar errores.

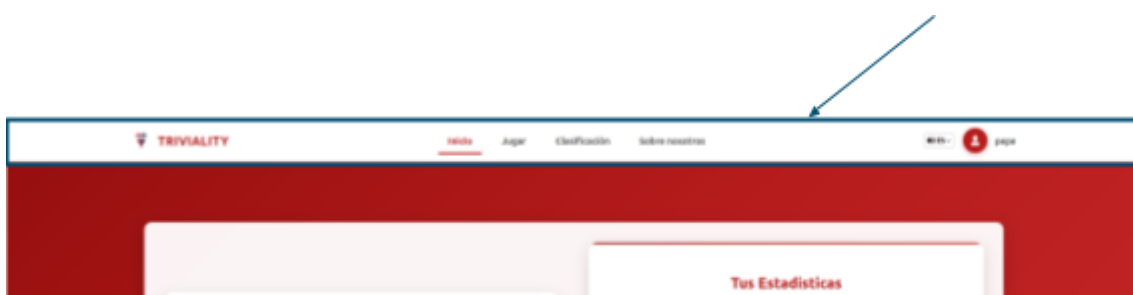
Después de añadir la información correctamente se redirige automáticamente a la home, como un nuevo usuario recién creado.

## 2.4.3 Home



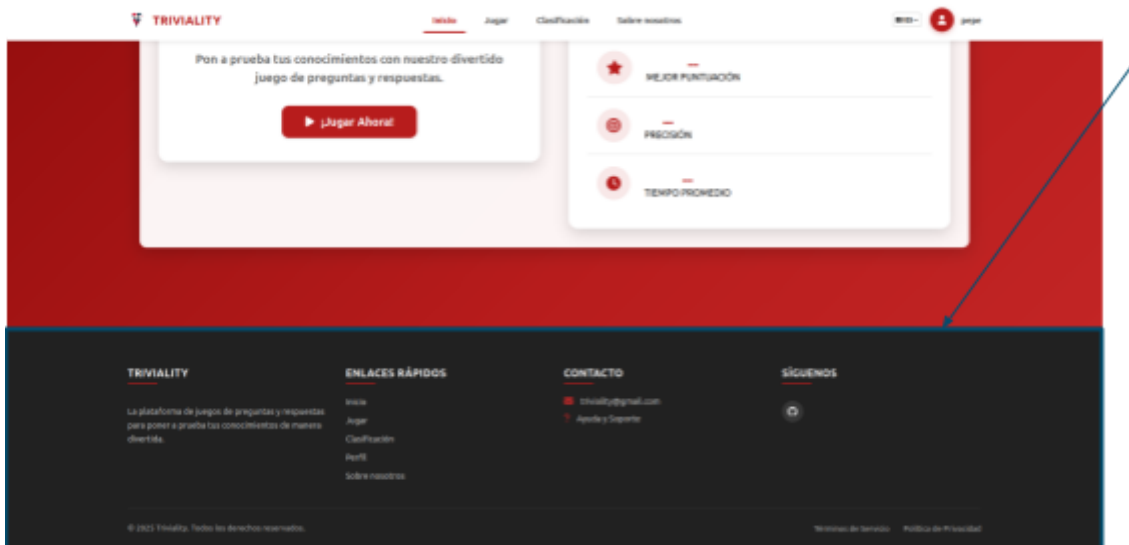
Ahora bien, después de verificar al usuario o al crear uno nuevo, se viajará a este componente. Esta parte es la cara principal del proyecto, ya que muestra la opción de jugar al juego e información rápida sobre el perfil, mostrando ciertos datos estadísticos.

## 2.4.4 Header



Otro componente que no tiene menor relevancia es el Header, el punto de conexión entre las páginas. Aparte de tener relación con los demás componentes, se tiene la opción de cambiar el idioma en el que se ve la página, además del botón para volver al inicio con el nombre de Triviality.

## 2.4.5 Footer



Aquí se tiene una parte importante del proyecto, en este área podemos añadir información sobre el equipo, como un correo de contacto o incluso más información sobre qué es *Triviality*. Además proporciona un aspecto más profesional y estético a la página.

## 2.4.6 Jugar

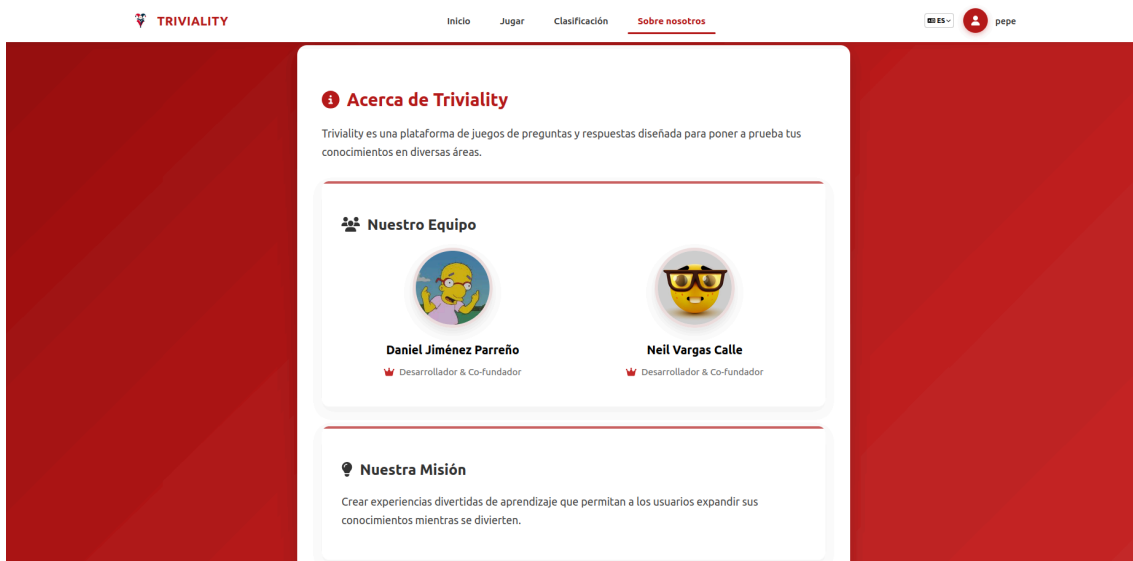






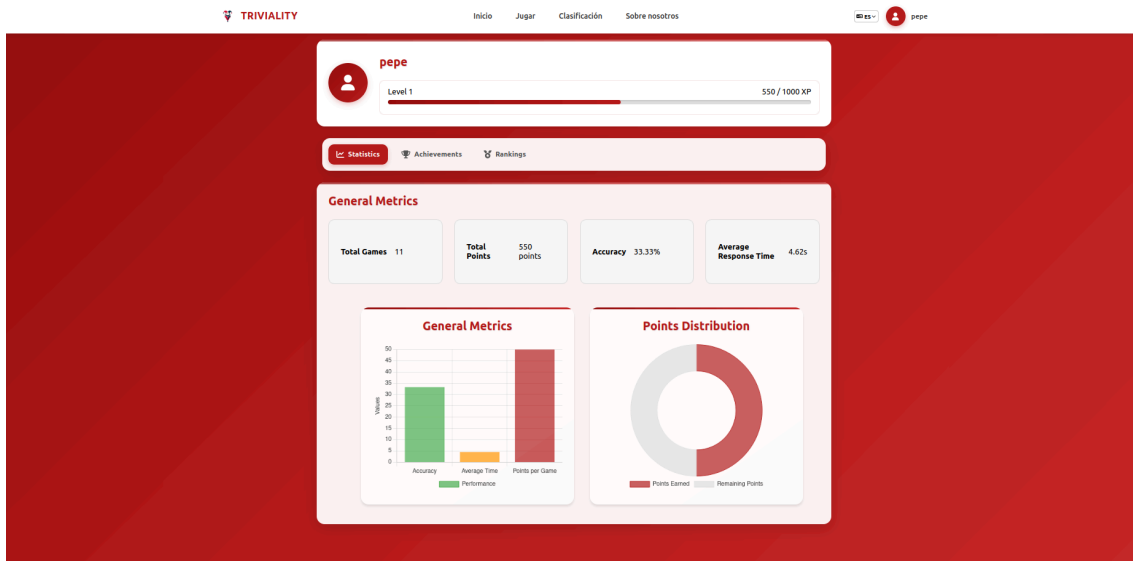
Este componente básicamente ubica la zona central de juego, exactamente en este lugar se podría decidir a qué jugar. Dado que este proyecto se centra en conseguir un juego funcional, se decidió tener un modo de juego y ver los demás como una visión a futuro.

### 2.4.7 About Us



Ahora bien, este es un componente muy influyente ya que se muestra mucha más información sobre el equipo de desarrollo. Además de mostrar más información en cuanto a *Triviality* y al proyecto en sí.

## 2.4.8 Perfil



Una de las implementaciones más innovadoras del sistema fue la creación de un perfil de usuario interactivo. Este perfil recopila y muestra datos estadísticos relevantes sobre el rendimiento del jugador, incluyendo:

- Nivel de experiencia y rango correspondiente
- Número total de partidas jugadas
- Puntos acumulados
- Porcentaje de respuestas correctas
- Tiempo medio de respuesta


Para presentar esta información de forma clara y visualmente atractiva, se implementó Chart.js, que genera gráficos dinámicos que permiten a los usuarios analizar fácilmente su progreso y rendimiento en el juego.

## 2.4.9 Trivia Game

### 2.4.9.1 Configuración Inicial (Pantalla de Preparación)

Al acceder al juego, los jugadores encuentran una interfaz de personalización donde pueden:

- Definir preferencias de juego:
  - Idioma para las preguntas (opciones disponibles: español, catalán, inglés)
  - Nivel de complejidad (básico, intermedio, avanzado)
  - Área temática (opcional: ciencias, cultura, deportes, etc.)
  - Cantidad de preguntas por partida
- La interfaz utiliza controles modernos con:
  - Selectores desplegables intuitivos
  - Diseño visualmente atractivo y responsive
- Acción principal: botón "Comenzar Juego" para iniciar la partida

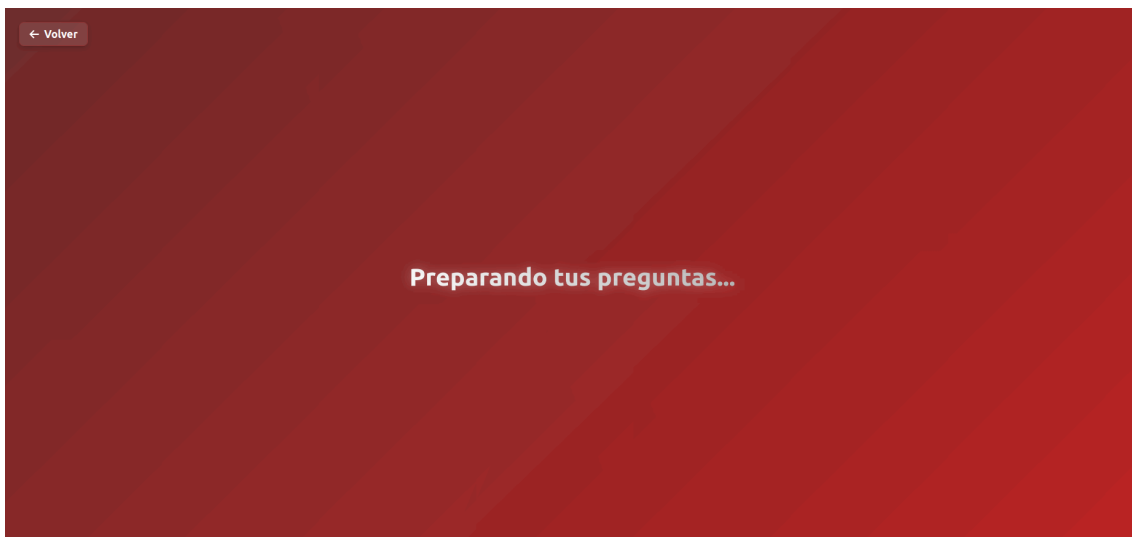




#### 2.4.9.2 Interfaz de Carga (Transición)

Durante la preparación del juego:

- Procesos en segundo plano:
  - Conexión con la API de preguntas
  - Traducción automática cuando corresponde
- Elementos de interacción:
  - Animaciones fluidas de carga
  - Mensajes contextuales que mantienen al usuario informado
- Objetivo: crear expectativa mientras se cargan los recursos

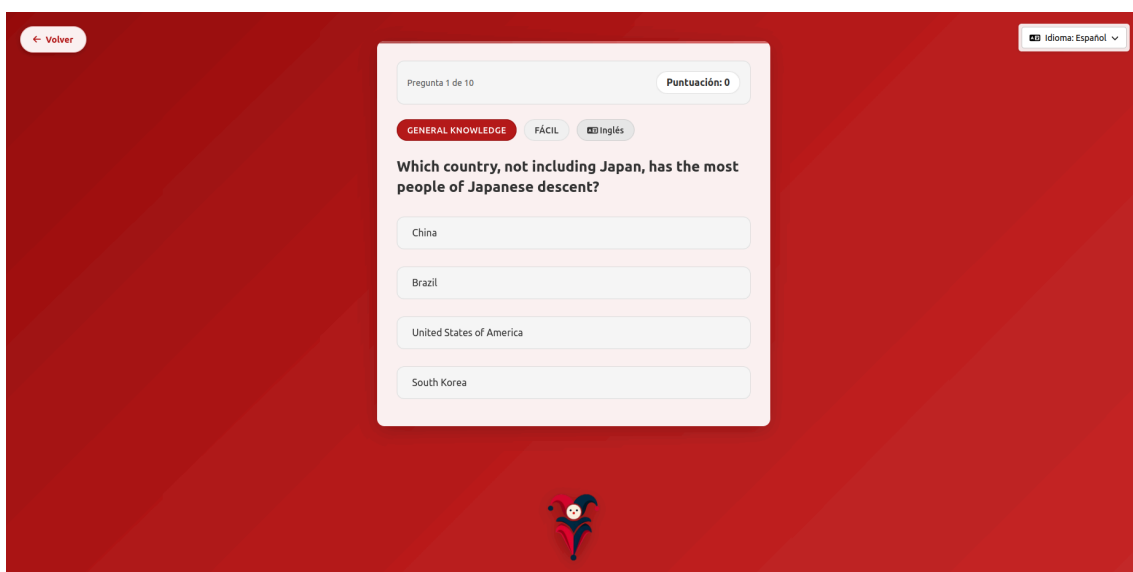




### 2.4.9.3 Núcleo del Juego (Pantalla de Preguntas)

Interfaz principal con:

- Estructura clara:
  - Enunciado destacado en zona superior
  - Cuatro alternativas de respuesta
  - Indicador de progreso (ej: "2/10 preguntas")
- Dinámica de juego:
  - Selección de respuesta
  - Retroalimentación visual inmediata
  - Transición automática a siguiente pregunta





#### 2.4.9.4 Resultados Finales (Resumen de Partida)

Al completar todas las preguntas:

- Datos mostrados:
  - Puntuación global
  - Efectividad (aciertos/errores)
  - Velocidad promedio de respuesta
  - Reconocimientos especiales
- Opciones posteriores:
  - Reiniciar partida
  - Modificar configuración
  - Consultar perfil de jugador
- Propósito: ofrecer análisis de desempeño y motivar a mejorar





## 2.4.10 Clasificación



La aplicación incluye un ranking global que muestra a todos los jugadores ordenados según su puntuación acumulada. Esta funcionalidad no solo mejora la experiencia visual (UI) con un diseño intuitivo y atractivo, sino que también fomenta la competitividad entre los usuarios. Al poder comparar sus resultados con los de otros jugadores, los participantes se sienten motivados a mejorar su desempeño y, en consecuencia, interactúan más frecuentemente con la aplicación, aumentando así su engagement y tiempo de uso.



## 2.5 Definición de las funcionalidades

Ahora bien, la aplicación web contiene varias funcionalidades ya mencionadas, en este punto se hablará mucho más sobre estas.

### 2.5.1 Sistema de Autenticación y Registro

En Triviality, el sistema de registro y autenticación es un componente esencial que asegura la protección y privacidad de nuestros usuarios. El procedimiento se inicia con el registro, en el que los usuarios recién incorporados añaden su información personal, la cual es verificada meticulosamente para garantizar su integridad y unicidad. Una vez verificadas, las claves son encriptadas a través de algoritmos seguros antes de ser guardadas en la base de datos, salvaguardando de esta manera la información delicada de los clientes.

El sistema de registro incorpora tokens JWT (Tokens Web JSON), los cuales ofrecen una capa extra de protección. Cuando un usuario accede, el sistema crea un token exclusivo que guarda datos encriptados acerca de la sesión del usuario. Este token se emplea para validar las solicitudes subsiguientes, facilitando al usuario el acceso a las funcionalidades resguardadas de la aplicación sin la necesidad de repetir la introducción de sus credenciales.

Las sesiones persistentes permiten a los usuarios mantener su sesión activa incluso después de cerrar el navegador, siempre que el token sea válido. Esto mejora la experiencia del usuario mientras mantiene la seguridad del sistema. Todo este sistema de autenticación está integrado con Laravel Sanctum, que proporciona una capa adicional de seguridad y facilita la gestión de tokens y sesiones.





### 2.5.2 Gestión de Perfiles de Usuario

- Sistema de niveles basado en experiencia
- Estadísticas de juego (partidas jugadas, aciertos, etc.)

En *Triviality*, la gestión de perfiles de usuario es un sistema dinámico y motivador que enriquece la experiencia de juego. Cada usuario comienza en el nivel 1 y va progresando a través de un sistema de niveles basado en la experiencia acumulada. La experiencia se gana mediante la participación en partidas, donde cada respuesta correcta y cada partida completada contribuyen a aumentar los puntos de experiencia del usuario. Este sistema de progresión está diseñado para mantener a los jugadores comprometidos y motivados, ofreciendo una sensación tangible de avance y logro.

El perfil de usuario incluye un conjunto completo de estadísticas que reflejan el rendimiento del jugador. Estas estadísticas abarcan desde el número total de partidas jugadas hasta el porcentaje de aciertos, pasando por el tiempo promedio de respuesta y las categorías favoritas. Todas estas métricas se actualizan en tiempo real, permitiendo a los usuarios ver su progreso y áreas de mejora. El sistema también mantiene un historial detallado de las partidas, incluyendo las preguntas respondidas, las categorías jugadas y los resultados obtenidos, lo que permite a los usuarios analizar su desempeño y establecer objetivos de mejora.

Esta gestión de perfiles no solo sirve como herramienta de seguimiento personal, sino que también fomenta la competencia amistosa entre usuarios, ya que las estadísticas son visibles en las tablas de clasificación. El sistema está diseñado para ser intuitivo y fácil de entender, presentando la información de manera clara y atractiva, lo que contribuye a mantener el interés y la participación de los usuarios en la plataforma.



### 2.5.3 Sistema de Juego de Trivia

- Generación dinámica de preguntas
- Categorías temáticas
- Diferentes niveles de dificultad
- Sistema de puntuación
- Feedback inmediato de respuestas

El sistema de juego de Triviality está diseñado para ofrecer una experiencia dinámica y envolvente a través de su sistema de trivia. El corazón del juego es la generación dinámica de preguntas, que se obtienen de una API externa y se procesan a través del backend. Este sistema asegura que cada partida sea única, con preguntas que se seleccionan y mezclan aleatoriamente, manteniendo el interés y el desafío para los jugadores.

El juego ofrece diferentes niveles de dificultad, desde principiante hasta experto, lo que permite a los jugadores progresar gradualmente y enfrentarse a desafíos apropiados para su nivel de conocimiento. El sistema de puntuación es sofisticado y tiene en cuenta varios factores: la dificultad de la pregunta, la velocidad de respuesta y la precisión. Las respuestas correctas otorgan puntos, mientras que las incorrectas pueden resultar en penalizaciones, creando un sistema de juego equilibrado y desafiante.

Un aspecto crucial del sistema es el feedback inmediato que proporciona a los jugadores. Después de cada respuesta, el usuario recibe información instantánea sobre si su respuesta fue correcta o incorrecta. Este feedback no solo mejora la experiencia de aprendizaje sino que también mantiene a los jugadores comprometidos y motivados para mejorar su rendimiento en las siguientes preguntas.



#### 2.5.4 Sistema de Clasificaciones

- Tabla de clasificación global
- Estadísticas de rendimiento
- Comparativa con otros jugadores

El sistema de clasificaciones de Triviality es un elemento motivador clave que fomenta la competencia sana y el compromiso de los usuarios. La tabla de clasificación global funciona como un escaparate dinámico que muestra los mejores jugadores de la plataforma, ordenados por su puntuación total acumulada. Esta clasificación se actualiza en tiempo real, reflejando instantáneamente los nuevos logros y puntuaciones de los jugadores, lo que mantiene la competitividad y el interés de la comunidad.

#### 2.5.5 Sistema de Internacionalización

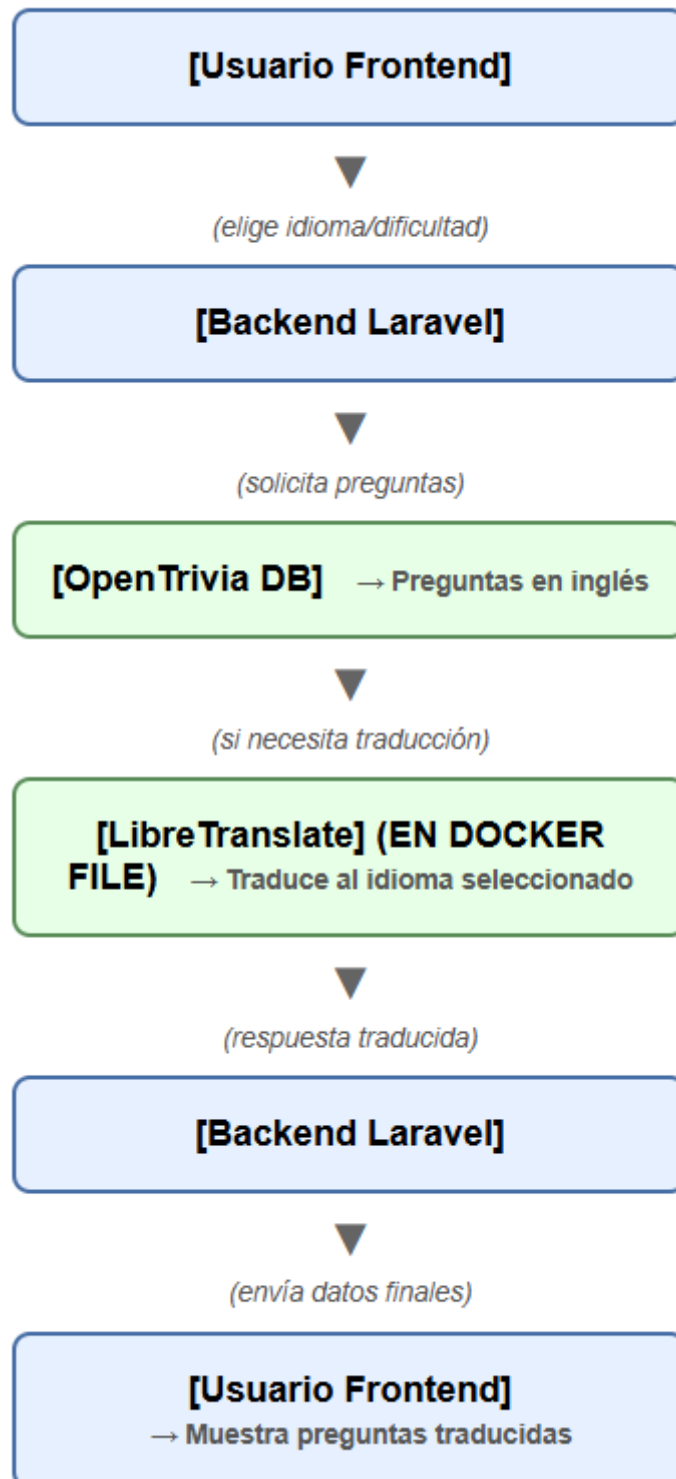
El sistema de internacionalización de Triviality es un componente sofisticado que opera en dos niveles distintos para proporcionar una experiencia más globalizada. En el primer nivel, tenemos la internacionalización de la interfaz de usuario, que permite a los usuarios navegar por la aplicación en tres idiomas principales: español, catalán e inglés. React-intl (librería ya comentada) se integra perfectamente con el sistema de componentes React, permitiéndonos definir y gestionar las traducciones de manera cómoda con archivos JSON.

El segundo nivel, más complejo, es el sistema de traducción de preguntas. Cuando recibimos preguntas de la API externa de trivia, estas llegan inicialmente en inglés. Cuando un usuario interactúa con la interfaz de React, las peticiones se envían al backend de Laravel, donde se validan las credenciales y permisos. El backend, a su vez, se comunica con las APIs externas, procesa los datos recibidos y los almacena en caché para optimizar el rendimiento. Finalmente, los datos procesados se envían de vuelta al frontend, actualizando la interfaz de usuario y manejando cualquier error que pueda surgir. El sistema de traducción automática, se encarga de traducir estas



preguntas a un mundo de idiomas añadidos. Este proceso es inteligente y eficiente, ya que incluye un sistema de caché que almacena las traducciones más comunes para evitar llamadas innecesarias al servicio de traducción.

Este sistema de internacionalización asegura que tanto la experiencia de usuario como el contenido del juego sean accesibles y disfrutables para una audiencia global, eliminando barreras lingüísticas y permitiendo que jugadores de diferentes países disfruten de Triviality en su idioma preferido.





### 2.5.6 Características Técnicas

Las características técnicas de Triviality están diseñadas para garantizar un rendimiento óptimo y una experiencia de usuario fluida. La arquitectura API RESTful es el pilar de nuestra comunicación entre el frontend y el backend, proporcionando endpoints bien definidos y estandarizados para todas las operaciones. Además el sistema de caché de preguntas es una característica crucial para el rendimiento de la aplicación. Este sistema reduce significativamente la carga en los servidores y mejora los tiempos de respuesta, especialmente importante durante picos de tráfico o cuando múltiples usuarios están jugando simultáneamente.

Otro componente crucial del sistema de administración es la mejora de las consultas a la base de información. Estas mejoras son especialmente significativas en labores como la elaboración de tablas de clasificación o la recuperación de estadísticas de usuario, donde el rendimiento es esencial para mantener una experiencia de usuario sin inconvenientes.

### 2.5.7 Interfaz de Usuario

La interfaz de Triviality se ha diseñado con un énfasis en el usuario, dando prioridad a la accesibilidad y a una experiencia de juego suave. La característica del diseño adaptable es esencial, puesta en marcha a través de un sistema de grid versátil y media queries que facilita que la aplicación se ajuste de manera ideal a cualquier dispositivo, desde teléfonos inteligentes hasta pantallas de escritorio. Esta capacidad de adaptación garantiza que los usuarios disfruten de Triviality con la misma experiencia de alta calidad sin importar el dispositivo que empleen. A la vez que se añadieron unas pocas animaciones para lograr una percepción de fluidez que mejora la navegación de nuestra aplicación.

### 3 Herramientas

Para optimizar el flujo de trabajo y garantizar una mayor productividad durante el desarrollo del proyecto, se han empleado las siguientes herramientas clave:

- Trello
- Github
- Visual Studio Code
- DBeaver

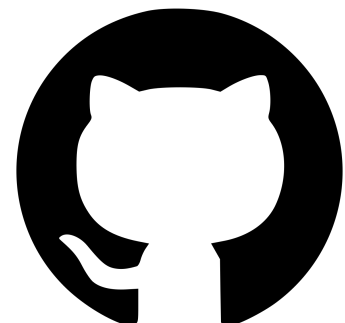
#### 3.1 TRELLO

Trello es una herramienta de gestión de proyectos que utiliza tarjetas y tableros virtuales para organizar tareas de manera visual. Su interfaz intuitiva y flexible lo hace ideal para equipos de desarrollo, permitiendo planificar, asignar y hacer seguimiento de las actividades del proyecto.



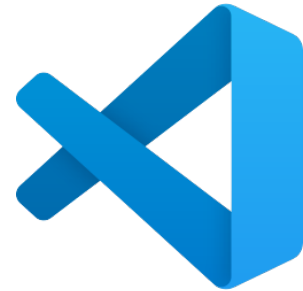
#### 3.2 GITHUB

GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Su interfaz clara y potente lo convierte en una herramienta esencial para equipos de programación, permitiendo gestionar el código fuente, revisar cambios y coordinar el trabajo entre múltiples desarrolladores.



### 3.3 VISUAL STUDIO CODE

Visual Studio Code es un editor de código ligero pero potente, con soporte para múltiples lenguajes de programación. Su ecosistema de extensiones y características integradas lo hacen ideal para el desarrollo ágil, ofreciendo herramientas avanzadas para escribir, depurar y mantener el código de manera eficiente.



### 3.4 DBEAVER

DBeaver es una herramienta universal para administración y desarrollo de bases de datos. Su interfaz intuitiva y soporte para múltiples sistemas gestores de bases de datos lo convierten en una solución versátil, permitiendo ejecutar consultas, diseñar esquemas y analizar datos de manera visual y organizada.



## 4 Conclusiones

### 4.1 Conclusiones generales del proyecto

Triviality ha demostrado ser un proyecto exitoso que combina tecnología moderna con una experiencia de usuario atractiva. La arquitectura elegida, utilizando Laravel en el backend y React en el frontend, ha permitido crear una aplicación robusta.





## 4.2 Consecucion de los objetivos

Triviality ha logrado cumplir con todos los objetivos establecidos al principio.

### [1.3 Objetivos](#)

## 4.3 Valoración de la metodologia i planificació

Para el desarrollo de Triviality se adoptó una metodología ágil simplificada, que demostró ser altamente efectiva pese a su enfoque práctico. Si bien existían opciones más complejas (como Scrum a gran escala), la elección de un sistema ágil ligero, con una distribución clara de tareas permitió:

- Máxima flexibilidad para adaptar requisitos según feedback temprano
- Eficiencia en la colaboración gracias a la asignación rotativa de roles
- Entrega constante de funcionalidades completas cada iteración

Esta aproximación resultó ideal para un proyecto de naturaleza lúdica como Triviality, donde:

- La prioridad era mantener un desarrollo fluido más que seguir procesos rígidos
- El equipo pequeño (2 personas) se beneficiaba de comunicación directa
- Los objetivos evolucionaban orgánicamente con las pruebas de usuario



#### 4.4 Visión de futuro

Ahora bien, se cree que la aplicación Triviality tiene un techo muy grande. Al momento de pensar sobre cómo avanzar en el proyecto y en momentos de desarrollo, hubieron muchas ideas de nuevas implementaciones y funcionalidades que podrían subir la calidad del proyecto a pasos agigantados.

Se podrían ampliar en diferentes campos como por ejemplo:

- Implementación de nuevos modos de juego (online, modo de juego temporizado, eventos temporales, etc.)
- Más opciones de personalización de usuarios (avatares, sistema de amistad, etc.)
- En cuanto a contenido se podría implementar nuevas preguntas, diferentes categorías, etc.

#### 5. Anexos

##### **Repositorio del proyecto:**

<https://github.com/DanieelJP/triviality-project>

##### **Plan de prevención de riesgos 2025:**

(Realizado pese a no estar bajo el nombre de la SL *Triviality*)

[Documento de prevención de riesgos laborales - Daniel Jimenez - Neil Vargas](#)

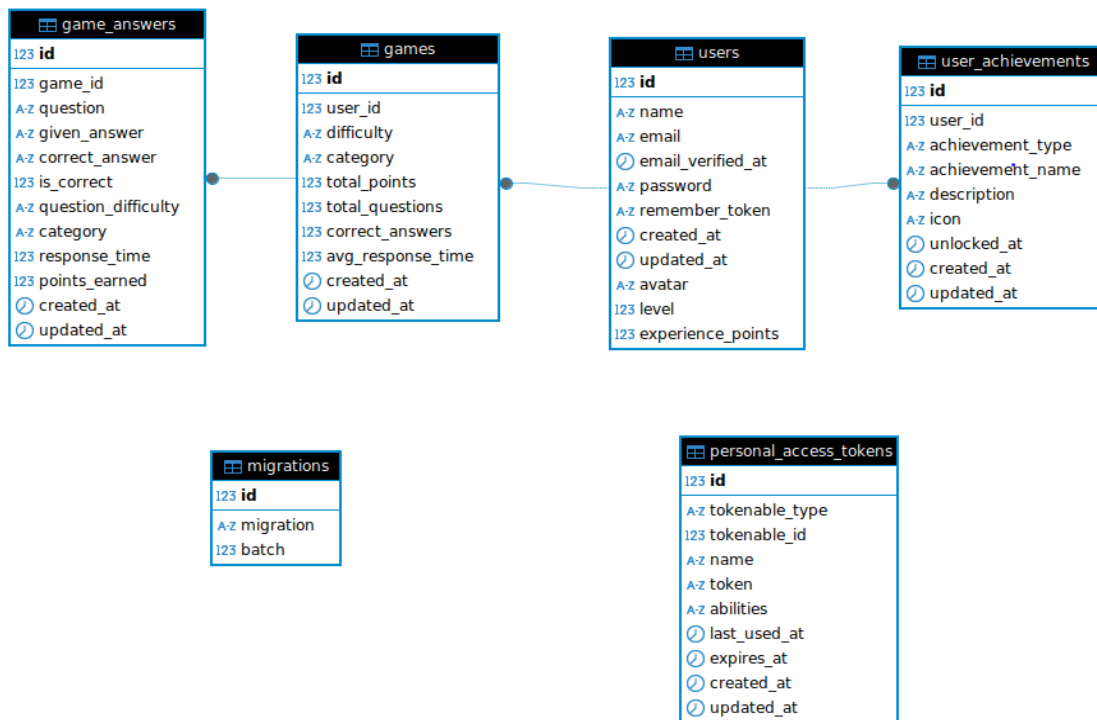
##### **Documento de las funcionalidades (primer esbozo)**

[https://docs.google.com/document/d/1SMh9R7wtd4ok8PTplwRT66FDpibSrQsQF0brbnzi6bQ/edit?usp=drive\\_link](https://docs.google.com/document/d/1SMh9R7wtd4ok8PTplwRT66FDpibSrQsQF0brbnzi6bQ/edit?usp=drive_link)



## DIAGRAMAS

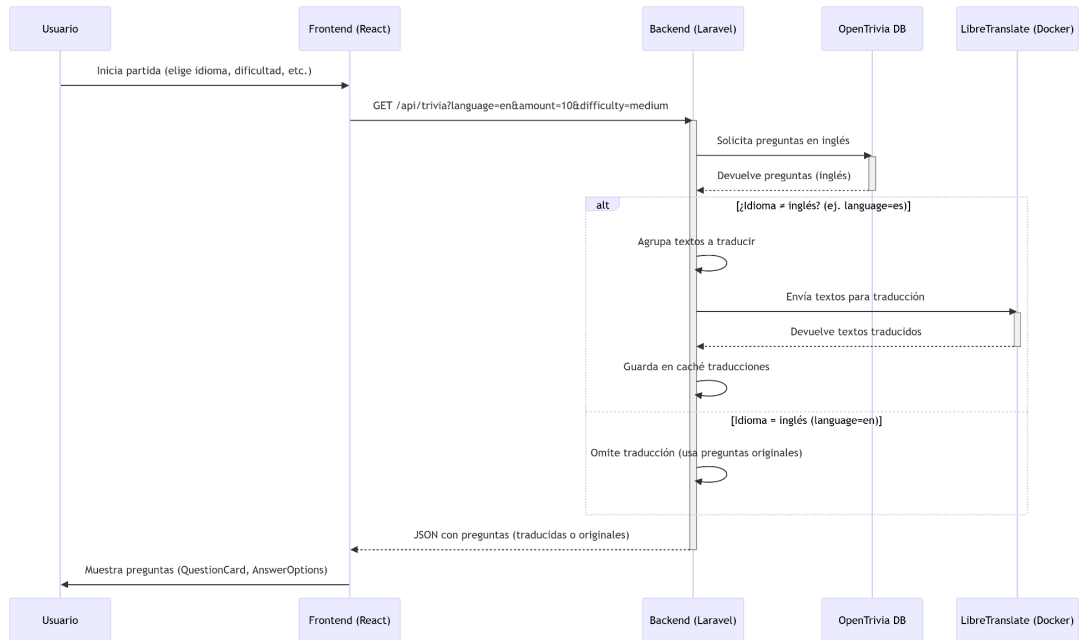
### DIAGRAMA DE CLASES DE LA BASE DE DATOS:



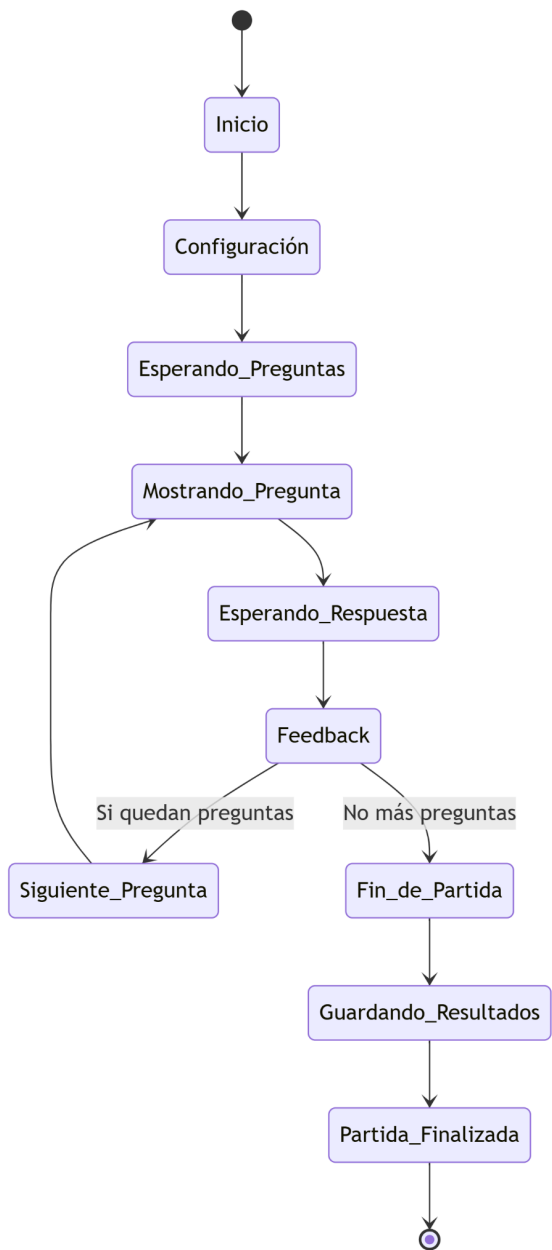
Caracterizado por su simplicidad gracias a la integración de APIs.



## DIAGRAMA DE FLUJO DE RECOGIDA Y TRADUCCIÓN DE PREGUNTAS



## DIAGRAMA DE ESTADOS DE VIDA DE UNA PARTIDA





## TRÁMITES DE CONSTITUCIÓN

Decidido que “*Triviality*” será una S.L. se necesitarán una serie de trámites de constitución para el inicio de la actividad de los socios (para que la empresa adquiera personalidad jurídica). Son estos siguientes pasos:

### **Certificación de denominación social negativa en el Registro Mercantil Central**

La denominación social es el nombre que identifica a una empresa a la hora de su actividad comercial. Este proceso se lleva a cabo para asegurar que el nombre elegido para la sociedad sea único y no lo utilice otra empresa. Para ello se registra y se asegura que el nombre “*Triviality*” es único y nos pertenece.

Las empresas comerciales realizan este trámite en el Registro Mercantil Central, mientras que las cooperativas lo hacen ante el Registro de Cooperativas. Nosotros lo realizaremos en el Registro Mercantil Central al tratarse de una empresa comercial.

La solicitud puede ser presentada por uno de los tres socios fundadores personalmente, por correo o en línea a través del sitio web del Registro Mercantil Central. En el formulario oficial, se deben indicar hasta tres nombres posibles en orden de preferencia, junto con la forma legal de la empresa. Nuestra empresa escoge el nombre “*Triviality*” como primera opción junto a otras dos opciones, pero debido a que la primera opción no está escogida por otra entidad nos definimos oficialmente como “*Triviality*”.

El certificado se otorga para el primer nombre disponible (“*Triviality*” en nuestro caso), y la denominación se reserva a nombre del solicitante durante seis meses. Después comienzan a correr una serie de plazos determinados.



Dentro de los siguientes tres meses, se debe formalizar la escritura pública de constitución; si no se hace, la certificación caduca. En los seis meses siguientes, se debe registrar el nombre en el Registro Mercantil; de lo contrario, la reserva del nombre se cancela y haría falta volver a realizar la solicitud de nuevo. Eso se evitará si hacemos un buen uso del tiempo y gestionamos los trámites de forma efectiva.

### **Justificación de aportaciones dinerarias o no dinerarias**

La aportación del capital de los socios deberá justificarse mediante certificación bancaria para respaldar la contribución de cada miembro al fondo de la compañía. Es un requisito fundamental para la constitución de la empresa que los miembros aporten un fondo mínimo, el cual debe estar comprometido y pagado, total o parcialmente, antes de proceder con la escritura oficial. Por ello, es crucial cumplir con este procedimiento.

Las aportaciones no dinerarias se constan en las escrituras de la constitución.

### **Escritura pública de constitución ante notario**

En esta parte los tres socios fundadores de la empresa tendrán que realizar la firma de la estructura pública, que establece la constitución de la empresa “*Triviality*”. Esto se hará ante la presencia de un notario. En este documento los socios declaran su voluntad de crear la sociedad, posibilitando que adquiera personalidad jurídica propia.

Delante del notario, para poder realizar la firma, los tres socios deben presentar la siguiente documentación:

- Certificación de la no existencia de la denominación de la empresa (vista en el primer paso).
- Documentos de identificación (DNI) de los fundadores.
- Comprobante bancario que confirme el depósito realizado a nombre de la empresa, correspondiente al capital social (visto en el segundo paso).



- Documentación con los estatutos de la empresa.

### **Solicitud del Número de Identificación Fiscal**

El Número de Identificación Fiscal (NIF) es la identificación oficial de la empresa que debe emplear en todas sus transacciones. Es imprescindible para la creación de “*Triviality*”.

Una vez que la sociedad ha sido establecida de manera legal, tiene un plazo de 30 días para requerir el NIF a la autoridad fiscal correspondiente. Este número sirve para identificar a la sociedad en cuestiones tributarias.

El NIF es fundamental para poder hacer algunas acciones financieras y económicas, como por ejemplo la apertura de cuentas bancarias o la compra y venta de bienes. Es importante que todas las personas y entidades sujetas a obligaciones fiscales cuenten con un NIF válido y actualizado.

El NIF tiene como función identificar a los empresarios individuales y a las entidades jurídicas para asuntos relacionados con la fiscalidad, el empleo y la administración. Los socios pueden solicitarlo o, si lo prefieren, el notario que certifica la escritura de constitución puede pedirlo el mismo día de la firma. Los tres socios de “*Triviality*” deben firmar el formulario.





## **Liquidación del Impuesto sobre Transmisiones Patrimoniales y Actos Jurídicos Documentados**

Existe una exención de impuesto para la hora de constituir una sociedad; pero pese a eso, sigue siendo necesario realizar la autoliquidación ya que estas siguen estando sujetas a pesar de estar exentas del pago. La liquidación se lleva a cabo presentando el formulario oficial junto con la escritura de constitución o el contrato de constitución de la sociedad. También hay que entregar una copia del NIF provisional.

Este tributo grava las transacciones comerciales realizadas por las sociedades, incluida su constitución:

- Al establecer una sociedad, es obligatorio abonar este impuesto en un plazo de 30 días hábiles después de la firma de la escritura.
- El pago se realiza en la oficina territorial correspondiente a la comunidad donde esté ubicada la empresa.

## **Inscripción en el Registro Mercantil**

También es imprescindible registrar la escritura pública de constitución en el Registro Mercantil.

El Registro Mercantil tiene como objetivo divulgar las condiciones legales y comerciales, de modo que, al llevarse a cabo la inscripción, estas puedan ser accesibles para todas las partes interesadas.

Una empresa no adquiere su entidad legal hasta que no se registra en el Registro Mercantil.

La solicitud de inscripción debe realizarse en el Registro Mercantil correspondiente a la provincia dentro del plazo de un mes posterior a la fecha de otorgamiento de la escritura pública.

Los documentos que hay que presentar en el Registro Mercantil son:

- Primera copia de la escritura de constitución.
- Certificación de no coincidencia de denominación.
- Justificante de haber realizado la liquidación del ITPAJD (Impuesto sobre Transmisiones Patrimoniales y Actos Jurídicos Documentados).



- NIF provisional de la sociedad.

Todos estos documentos se han pedido y se tienen en posesión como se ha visto previamente.

## **TRÁMITES DE PUESTA EN MARCHA**

Una vez la empresa ha adquirido personalidad jurídica se deben realizar los trámites que permitirán comenzar su actividad, conocidos como trámites de puesta en marcha. Son los siguientes:

### **Trámites fiscales**

Obtención del Número de identificación fiscal: Una vez la sociedad esté inscrita, hay que presentar ante la Agencia Estatal de la Administración Tributaria, la solicitud del NIF definitivo, adjuntando el resguardo del modelo 036, la copia de la escritura de constitución de la sociedad inscrita en el Registro Mercantil y la hoja de inscripción en el Registro Mercantil. Todo esto se deberá realizar en el plazo de 1 mes desde la inscripción de la sociedad en el Registro Mercantil.

Alta en el impuesto de actividades económicas: Este impuesto es un tributo de carácter local que grava el ejercicio de actividades empresariales, profesionales o artísticas. El impuesto es obligatorio para todas las sociedades que deban presentar tantas altas como actividades diferentes se vayan a ejercer.

Declaración censal: Presentar una declaración censal informando sobre el inicio de la actividad empresarial.



### **Trámites laborales**

**Alta en el régimen de autónomos:** El Régimen Especial de Trabajadores Autónomos (RETA) es obligatorio para personas físicas que trabajan por cuenta propia y socios de sociedades civiles y mercantiles. El alta en el RETA es única, incluso si el trabajador realiza varias actividades como autónomo. Debe realizarse en los 30 días naturales siguientes al inicio de la actividad, pero los efectos retroactivos comienzan desde el primer día del mes en que se inicia la actividad.

**Inscripción de la empresa en la Seguridad Social:** Cualquier empresa que inicia una actividad debe inscribirse en la Seguridad Social para poder contratar a personal. Al inscribirse, la empresa obtiene su número patronal, que es válido para todo el territorio español. Al solicitar la inscripción hay que indicar la entidad gestora o colaboradora que cubrirá el riesgo de accidentes de trabajo y enfermedades profesionales.

### **Trámites en la Inspección de Trabajo**

**Diligenciar el Libro de Visitas:** Este se usa para que los inspectores de trabajo anoten las diligencias practicadas al visitar los centros de trabajo.

**Obtención del calendario laboral:** La empresa debe exponer el calendario laboral en un lugar visible para todos los empleados.



## **ASPECTOS FISCALES**

Los aspectos fiscales de una empresa se refieren a todos los temas relacionados con los impuestos y obligaciones tributarias que la empresa debe cumplir. Estos aspectos son fundamentales para garantizar el cumplimiento legal y la estabilidad financiera de la empresa. Aquí hay algunos aspectos fiscales clave:

- Registro y Cumplimiento Fiscal
- Impuesto sobre la Renta de la Empresa
- Impuestos a las Ventas o al Valor Agregado (IVA)
- Retenciones e Ingresos a Cuenta
- Impuestos Locales
- Declaraciones Trimestrales o Anuales
- Auditorías Fiscales

Es importante que la empresa cuente con asesoramiento contable y fiscal profesional para garantizar el cumplimiento adecuado de todas las obligaciones fiscales y aprovechar las oportunidades para optimizar la carga tributaria.



## 6. Bibliografía

- Open Trivia DB. (s. f.). [https://opentdb.com/api\\_config.php](https://opentdb.com/api_config.php)
- Swagger UI. (s. f.). <https://es.libretranslate.com/docs/>
- LibreTranslate - API de traducción automática gratuita y de código abierto. (s. f.). <https://es.libretranslate.com/>
- W3Schools.com. (s. f.). <https://www.w3schools.com/css/>
- Documentation for Visual Studio Code. (2021, 3 noviembre). <https://code.visualstudio.com/docs>
- About | DBEaver Community. (s. f.). <https://dbeaver.io/about/>
- MySQL :: MySQL Documentation. (s. f.). <https://dev.mysql.com/doc/>
- Cabrera, A. (2024, 5 diciembre). Utiliza Laravel como Backend y React.js como Frontend para Crear Interfaces de Usuarios. 10Code Software Design. <https://10code.es/en/laravel-react/>
- Laravel + React js: Cómo crear el proyecto e integrar React. (s. f.). Frontendtools. <https://frontendtools.net/v/exec?u=Laravel-%2B-React-js%3A-C%C3%B3mo-crear-el-proyecto-e-integrar-React>
- GitHub.com Documentación de ayuda. (s. f.). GitHub Docs. <https://docs.github.com/es>
- Trello Guides: Help getting started with Trello | Trello. (s. f.). <https://trello.com/guide>
- React Router home. (s. f.). React Router. <https://reactrouter.com/home>
- Overview | Format.JS. (2025, 24 mayo). <https://formatjs.github.io/docs/react-intl/>
- Font awesome docs. (s. f.). Font Awesome Docs. <https://docs.fontawesome.com/>
- Empezando | Axios Docs. (s. f.). <https://axios-http.com/es/docs/intro>
- Chart.js | chart.js. (s. f.). <https://www.chartjs.org/docs/latest/>
- PHP: Hypertext Preprocessor. (s. f.). <https://www.php.net/docs.php>
- Documentation - TypeScript for JavaScript Programmers. (s. f.). <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>



## 7. Agradecimientos

Se desea expresar el más sincero agradecimiento a los tutores del proyecto: David Tomás, Yago Morales, David García, Rubén Arroyo y Marta Martínez.

Asimismo, se extiende el reconocimiento a todo el claustro docente que ha acompañado a lo largo del curso, por transmitir conocimientos especializados y fomentar el pensamiento crítico y la innovación entre el alumnado.

Por último, se agradece a los creadores de OpenTriviaDB y LibreTranslate por ofrecer a la comunidad sus soluciones tecnológicas, las cuales han sido componentes esenciales para la implementación práctica del proyecto. Sin su contribución previa, este resultado no habría sido viable.

