Toshokan

TU BIBLIOTECA DE MANGA Y ANIME

Javier Valor Martínez
Grau d'Enginyeria Informàtica
Desenvolupament Web

Proyecto Final de Grado

INDICE

INTRODUCCIÓN

Este proyecto surgió con la idea de poder organizar mi propia biblioteca de series, tanto anime (películas) como manga (comics). Por ello, busqué una API externa con el contenido actualizado, para poder centrarme en como queremos mostrarlo y hacer que nuestra aplicación sea práctica y funcional.

MOTIVACIÓN

Crear una aplicación desde cero, siguiendo una planificación y unos requisitos que tu mismo te has puesto, tiene un extra de **motivación** y **compromiso**. Si además puedes **elegir la tecnología** y tratar de ponerte **nuevas metas**, es aún **más excitante.**
Es por ello que he tratado de equilibrar el desarrollo entre tecnologías que controlo y otras nuevas que me gustaría explorar, dando como resultado un producto **fresco y robusto**, aunque, mermado por el transcurso de la pandemia.

TECNOLOGÍAS

1 CLIENTE

Utilizamos **NodeJS**, con la librería de **ReatJS**, el contenido esta desarrollado en Javascript, utilizando su última versión **ES6**, todo ello se encapsula y transpila con **Webpack**

2 SERVIDOR

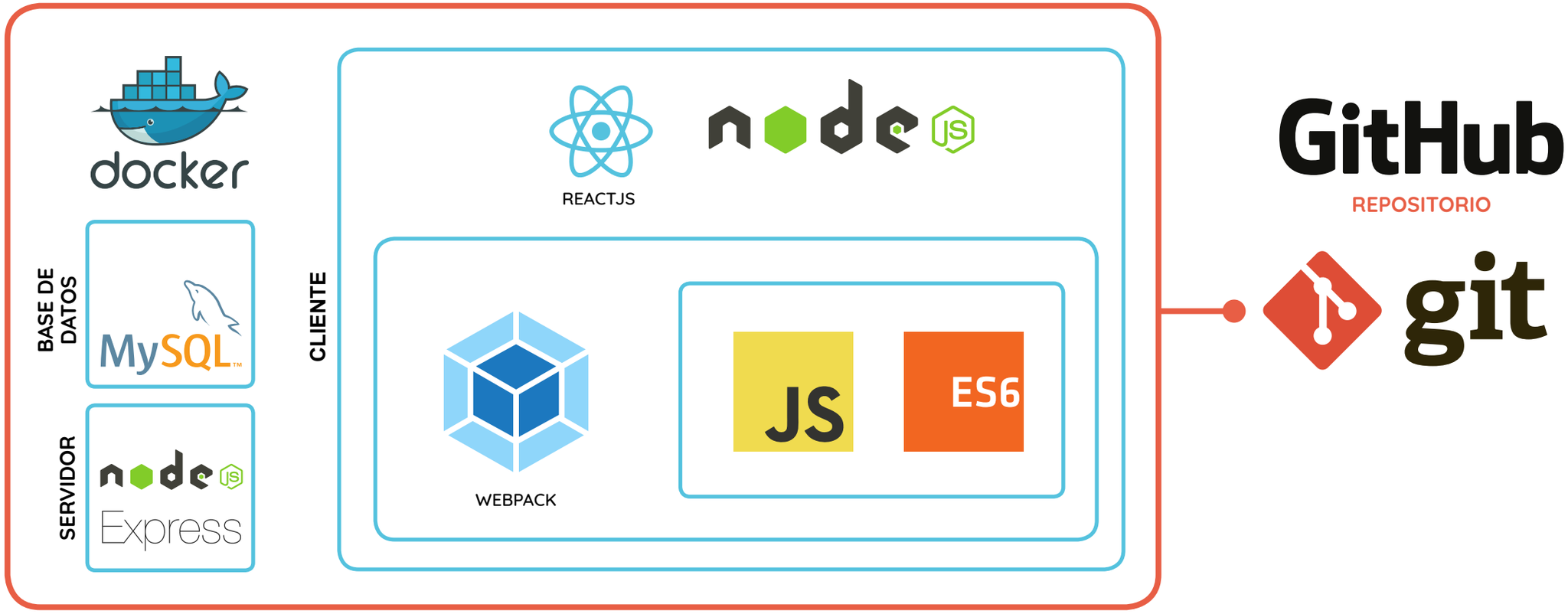
También utilizamos **NodeJS**, con el framework **Express**

3 BASE DE DATOS

Está montada con **MySQL**

4 OTROS

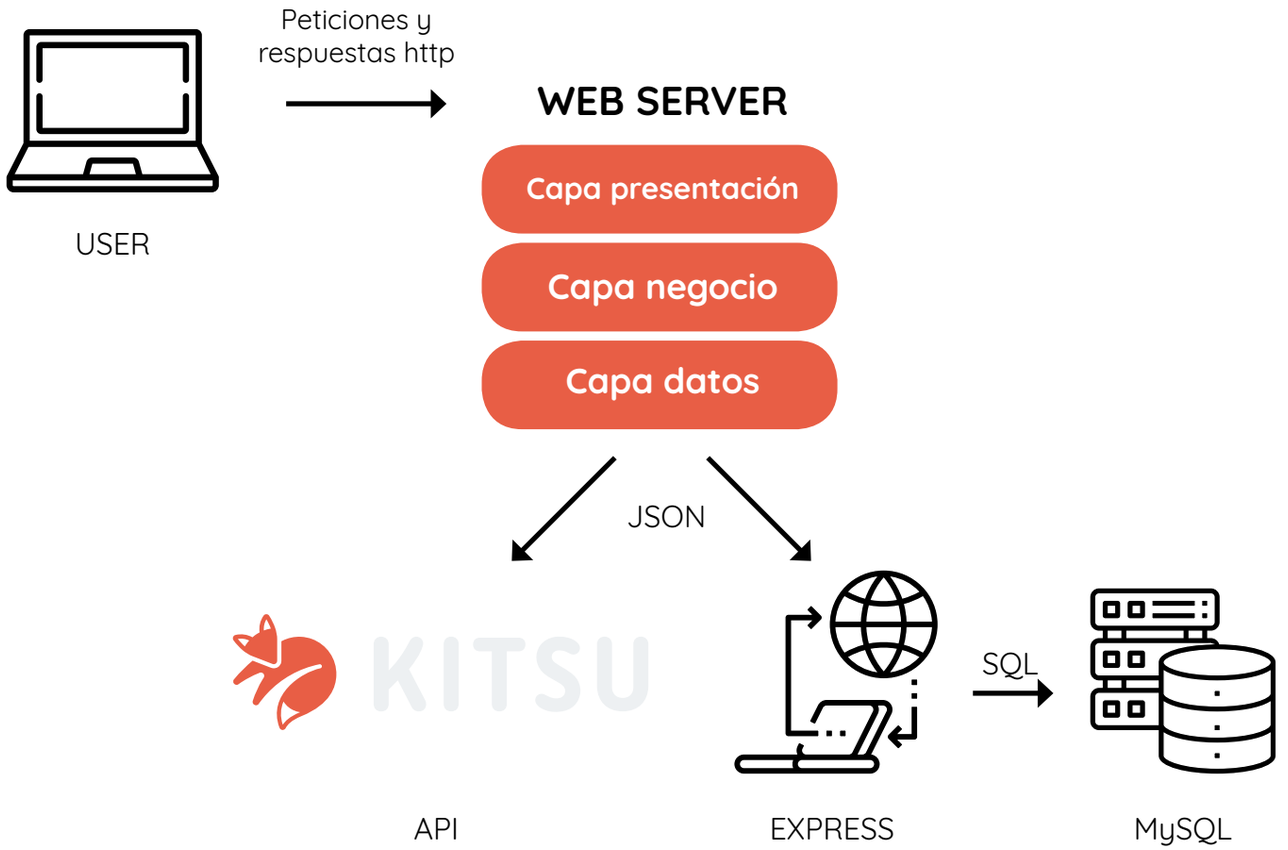
Todo esto esta encapsulado en contenedores que se despliegan gracias a Docker y subido en un repositorio de **GitHub** a través de **Git**



DISEÑO

ARQUITECTURA APLICACIÓN

La aplicación se conecta a una API externa para obtener el **contenido**, mientras que los usuarios y bibliotecas se almacenan en una **base de datos**. El cliente, a través de servicios, se conecta al **servidor**, que hace de **conector** entre cliente y base de datos. Se ha seguido un patrón de diseño **Facade** para mantener y estructurar el entorno de programación.

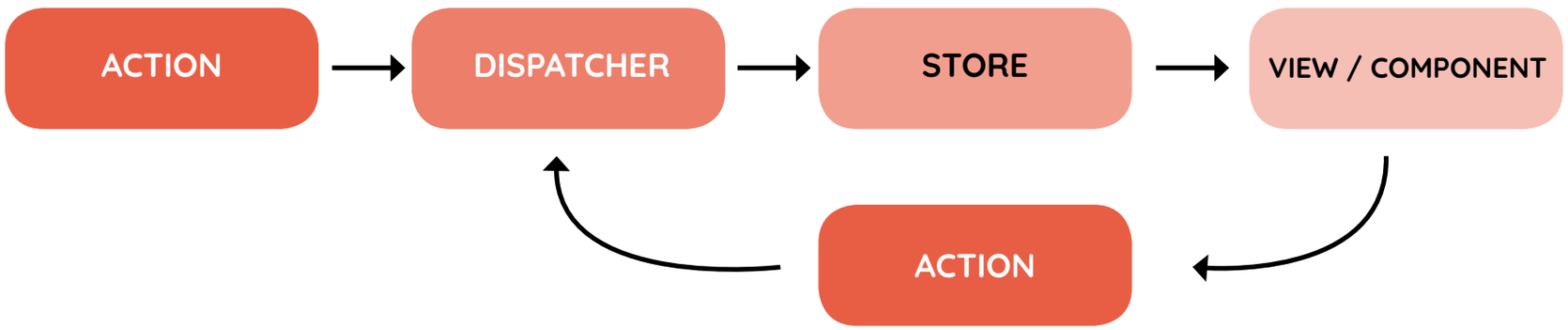


DISEÑO

ARQUITECTURA CLIENTE

La arquitectura del cliente ha dejado de ser siguiendo el patrón **MVC** (Modelo Vista Controlador) para dar paso a **Flux**, que propone una arquitectura en la que el flujo de datos es **unidireccional**.

ESQUEMA DEL PATRÓN FLUX



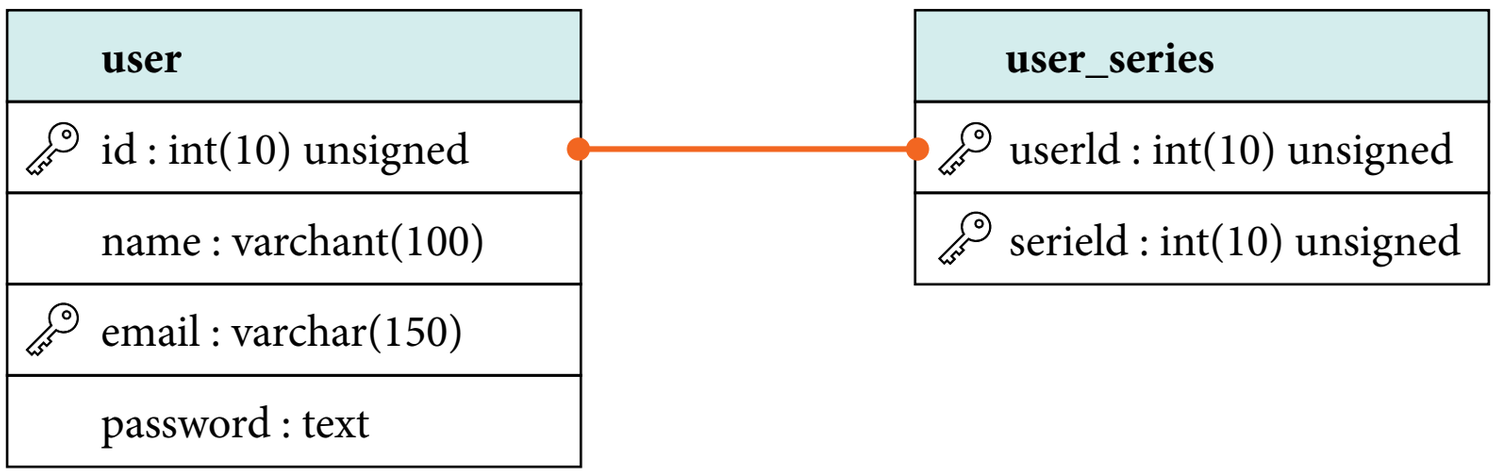
DISEÑO

ARQUITECTURA BASE DE DATOS

Hemos optado por un sistema de gestión de base de datos relacional MySQL, en nuestro caso utilizamos únicamente dos tablas **user** y **user_series**.

La tabla **user** tiene clave primaria id y el email como clave única.

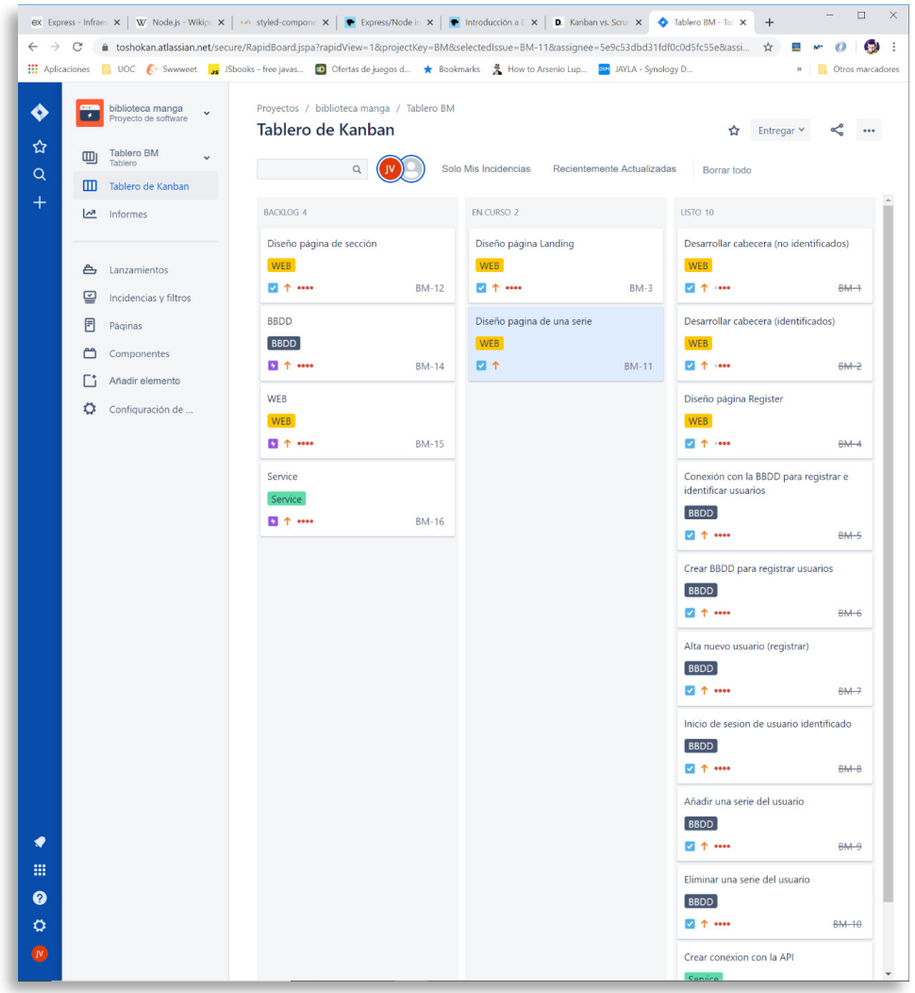
La tabla **user_series**, tiene como claves primarias serieId y userId, siendo esta clave foránea de la tabla user.



DESARROLLO

PLANIFICACIÓN

Antes de iniciar el desarrollo, desglosé las historias de usuario en **tareas**, para, mediante un tablon **Kanban**, listar todas las tareas a realizar e ir pasándolas entre las columnas in progress a **Done**.



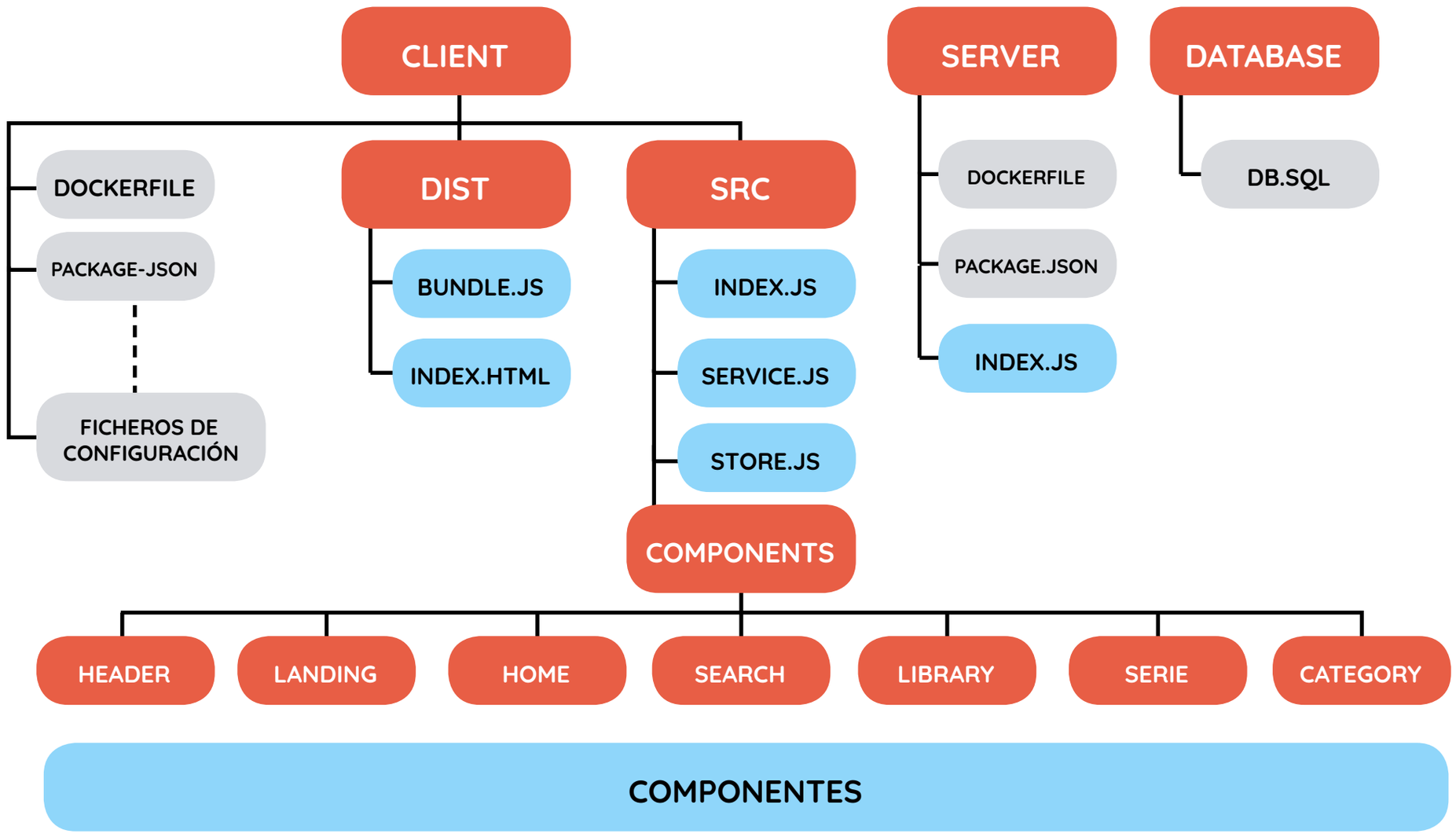
PROTOTIPO

Diseñé un prototipo para que al realizar el desarrollo, tener **más claro cual es el producto que quiero conseguir**, como tal, a sufrido cambios, pero ha sido de mucha utilidad para afrontar el desarrollo.

DESARROLLO

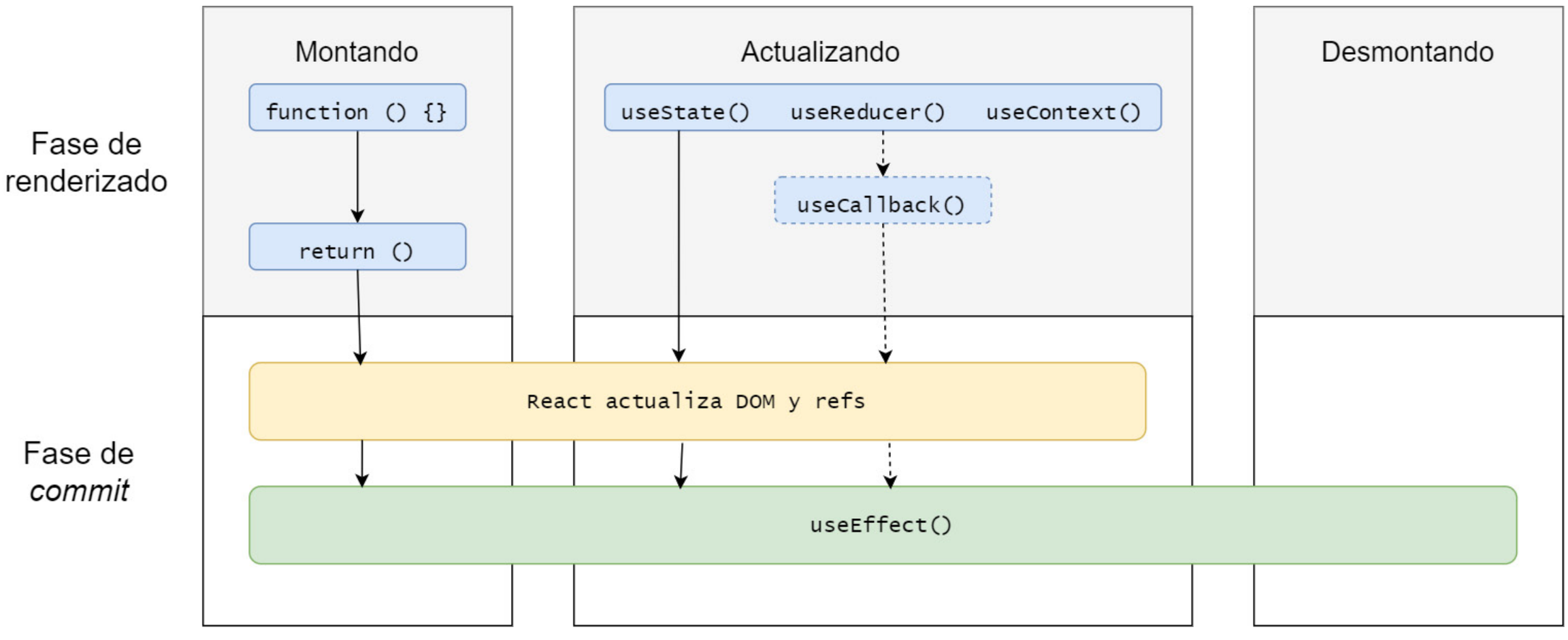
ESTRUCTURA

La estructura del proyecto por carpetas, separando **servidor**, **cliente** y **base de datos**. La parte cliente, siendo más densa, la he separado por **componentes**, con la idea de poder **reutilizarlos y desarrollarlos** de forma desacoplada del resto.



DESARROLLO

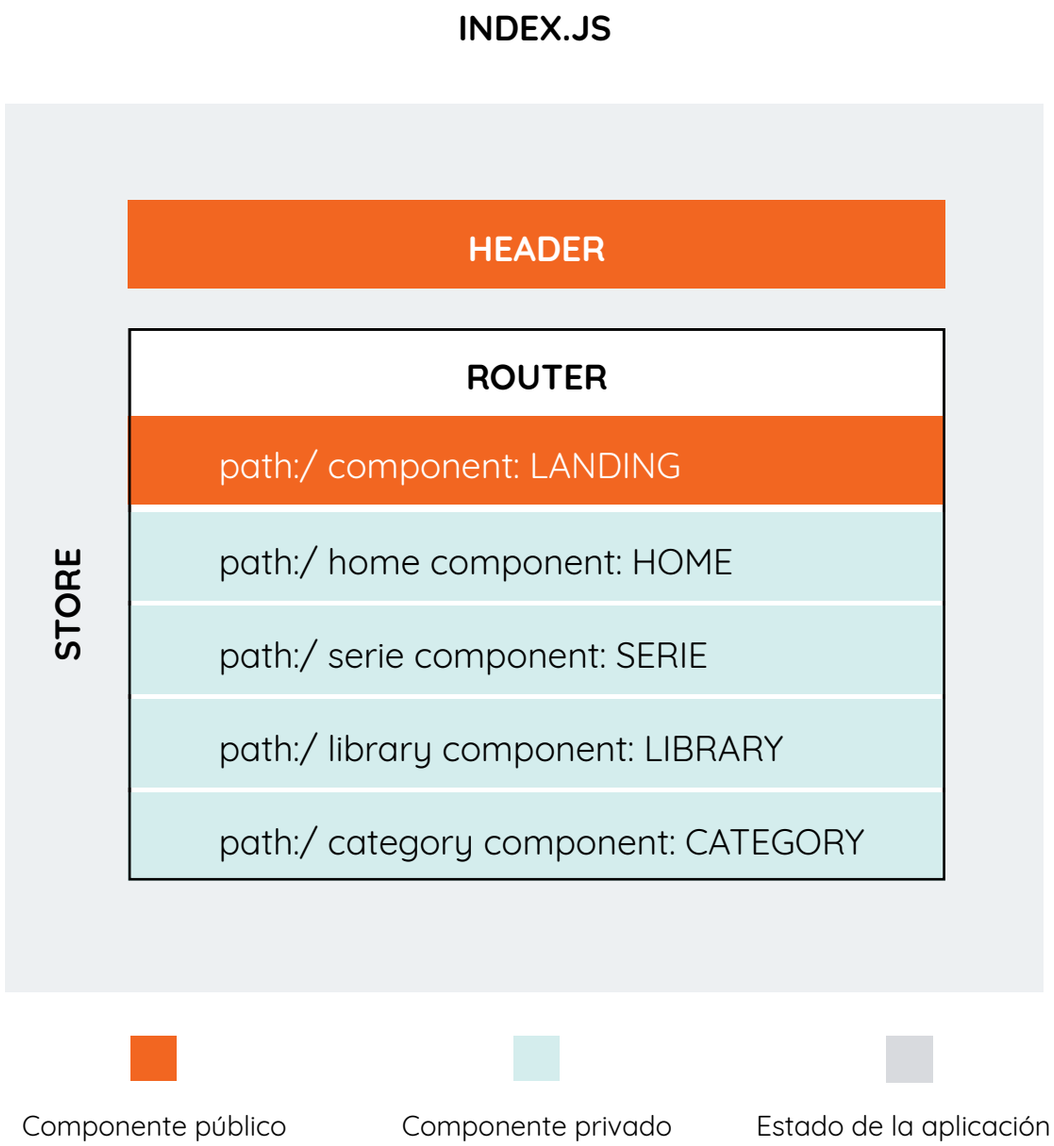
ARQUITECTURA DE COMPONENTES



DESARROLLO

PUNTO DE ENTRADA

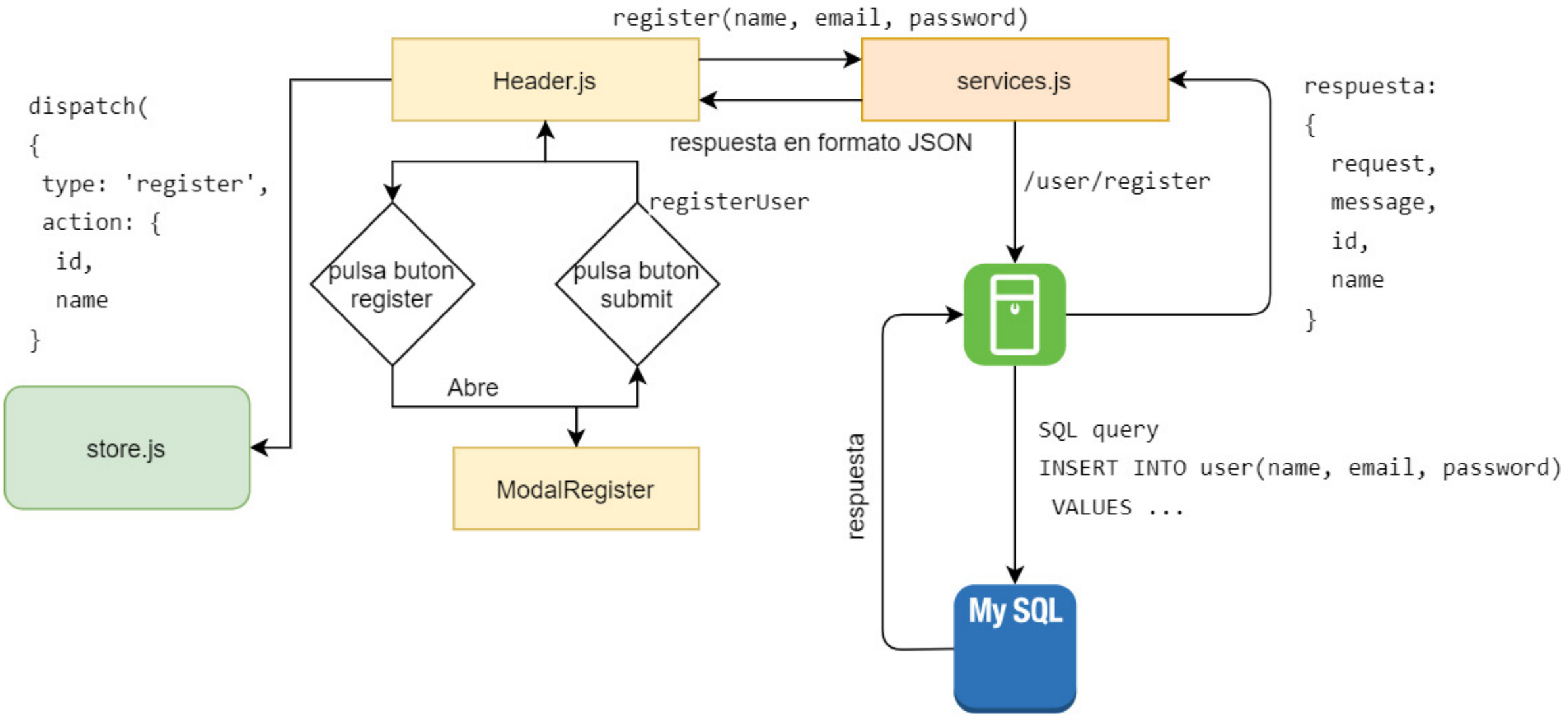
El punto de entrada de la aplicación, **index.js**, es el que la añade al **DOM** e importa los diferentes componentes que hacen la función de **vistas**. Para que todo esto esté **conectado** utilizamos el paquete **Router**, que básicamente toma las rutas y según el path **muestra** un componente u otro, en nuestro caso comprueba si se trata de una ruta **privada** o de una **pública**. Y por último, todos estos componentes están bajo el paraguas del **StateProvides** que conecta la aplicación con el estado global de la misma.



DESARROLLO

COMPONENTES

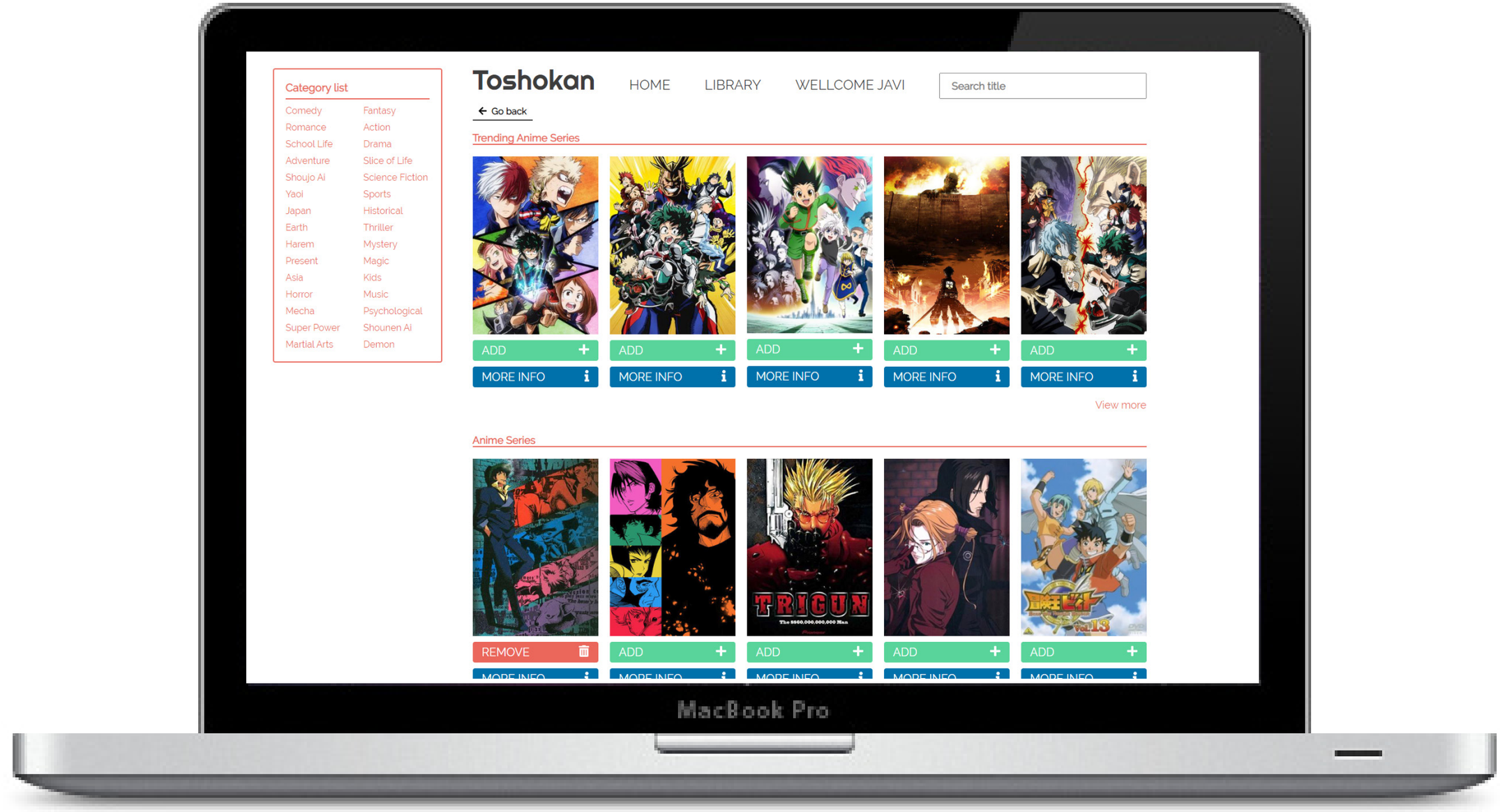
Antes hemos hablado del **ciclo de vida** de un componente, ahora quiero enseñar una pequeña parte de cómo funciona concretamente el **registro de usuarios**.



RESULTADOS

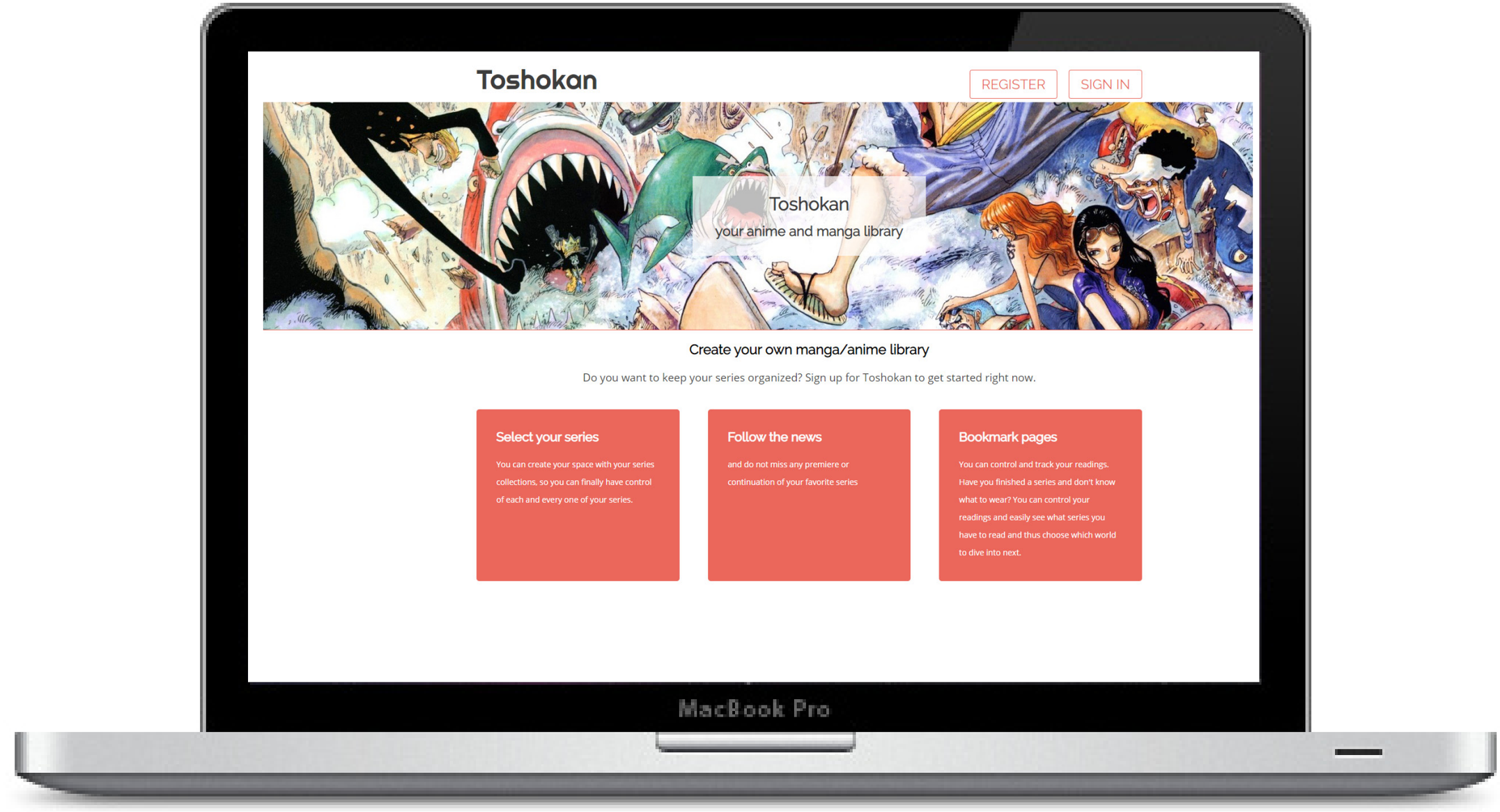
INICIO

Aplicación que muestra una **página de bienvenida**, que además permite registrar usuarios (y logearse). Una vez pasado este trámite, el usuario **accede a la página principal**, donde encontramos diferentes maneras de **acceder** a las series. Sugerencias iniciales, por **categorías** o por el **buscador**. En todas ellas aparece el botón de **añadir** (o eliminar) serie de la colección.



RESULTADOS

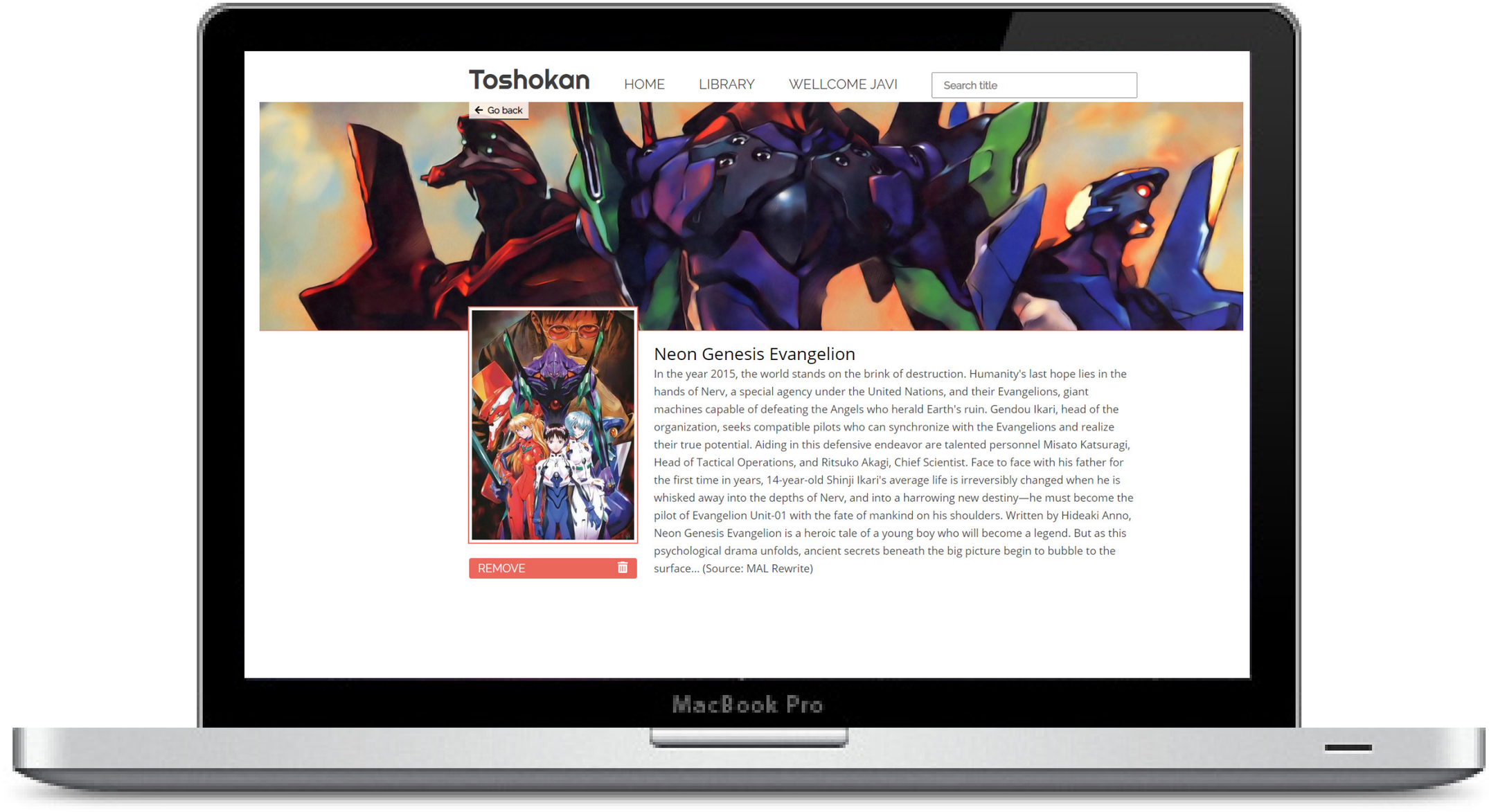
INICIO



RESULTADOS

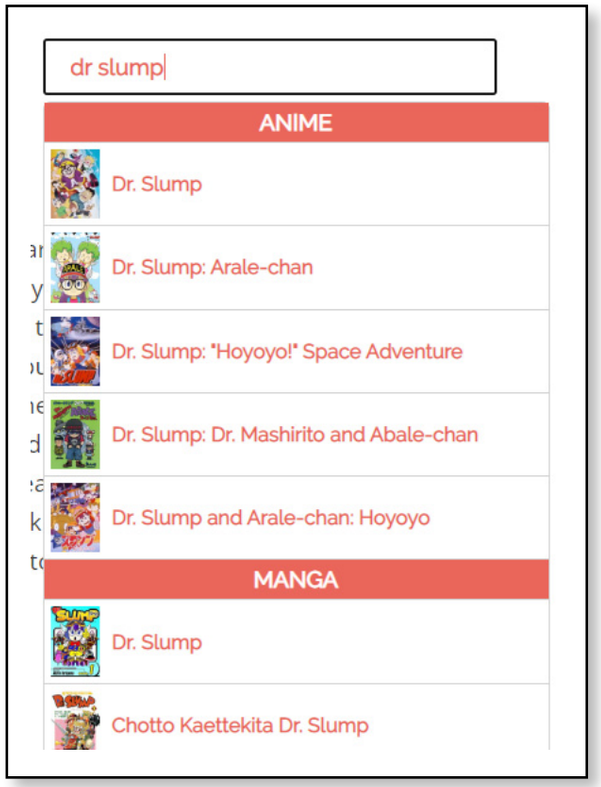
CONTENIDOS

Los usuarios pueden acceder a las diferentes series para **encontrar información detallada** de la misma. Así como entrar por **categorías**, con su descripción, **número** de series que contiene y algunas de las series mas **recientes y valoradas** por los usuarios.



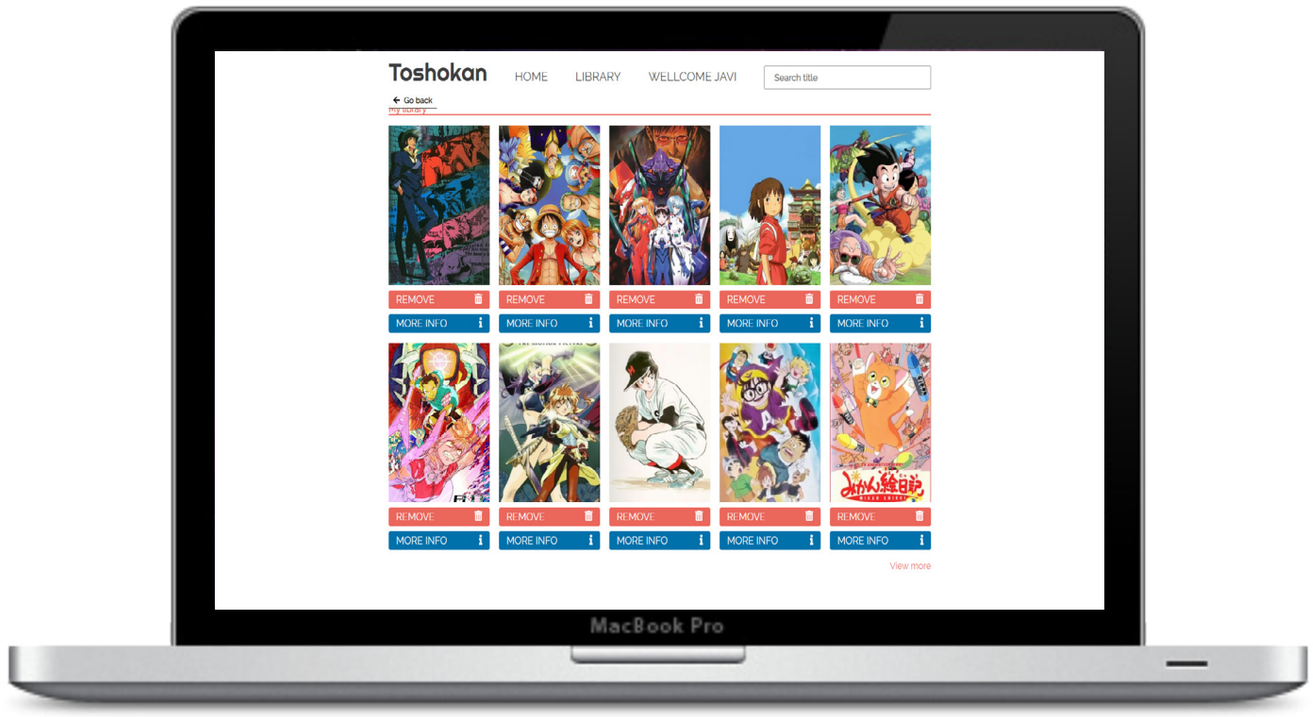
RESULTADOS
BUSCADOR

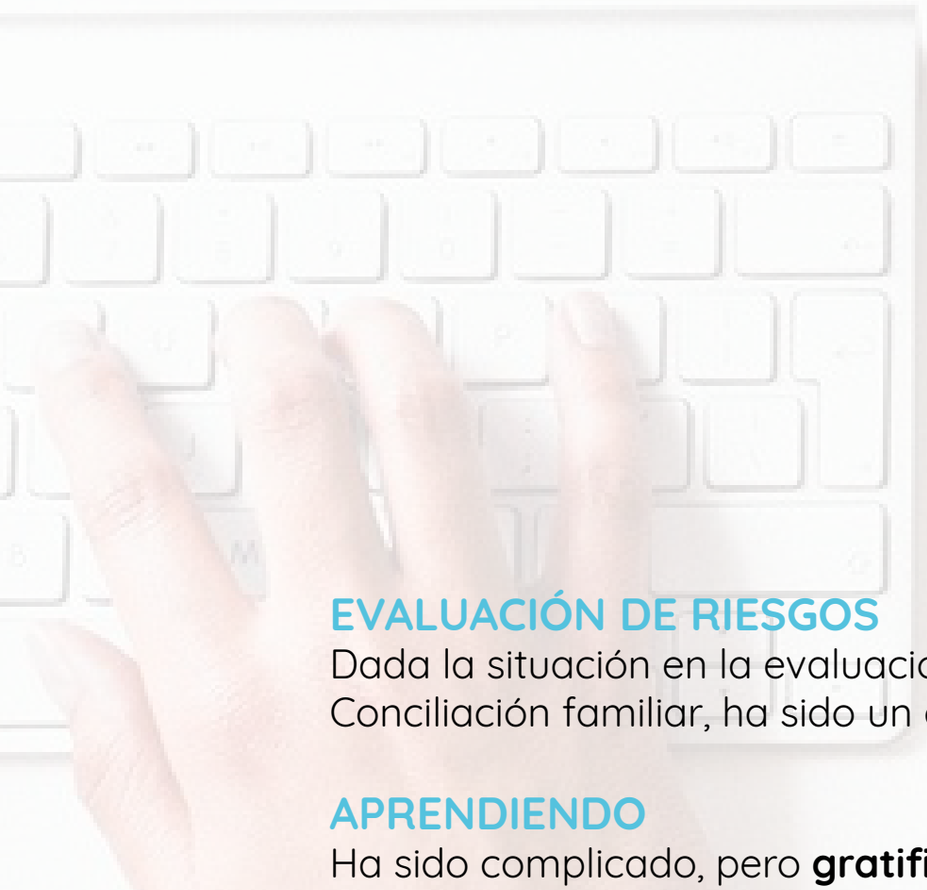
Para poder acceder a cualquier serie que el usuario desee, se ha **implementado un buscador** en tiempo real, que te da **sugerencias** mientras el usuario escribe.



RESULTADOS
BIBLIOTECA

Por último, los usuarios pueden **acceder a su biblioteca** para ver sus series.





CONCLUSIONES

EVALUACIÓN DE RIESGOS

Dada la situación en la evaluación de riesgos nos hemos quedado cortos. Confinamiento – Trabajo – Conciliación familiar, ha sido un coctel que ha **lastrado** los objetivos del proyecto.

APRENDIENDO

Ha sido complicado, pero **gratificante** ver como el proyecto se ha ido **gestando**, **planificando** y **desarrollando**. En el proceso he podido **aprender** muchas cosas.

OBJETIVOS

No hemos podido llegar a todos los objetivos, ha habido **grandes cambios** en la metodología del desarrollo que nos hemos visto obligados a realizar para poder entregar un **producto final**, como es el caso del uso de **TDD (Test-Driven Development)** o el uso de **Sonarqube** para la evaluación y calidad del código.

FUTURO

El proyecto esta creado de forma **escalable**, y esto, deja mucho margen a **futuras actualizaciones**, como añadir **paginación** para listar más series, un **buscador avanzado**, o mejoras en la **gestión** y el **orden** de la biblioteca.