



I E S A L B A R R E G A S

TÉCNICO SUPERIOR EN DESARROLLO DE
APLICACIONES WEB

Departamento de Informática



Rick and Morty Fan Hub
Manual Técnico

Autor/es: Rubén Escudero Calamonte
Curso Académico: 2023/2024

Índice

Contenido

1. Introducción	3
2. Arquitectura de la aplicación.....	3
2.1 Frontend	3
2.1.1 Tecnologías usadas.....	3
2.1.2 Entorno de desarrollo	3
2.1.3 Justificación de la elección	4
2.2 Backend.....	4
2.2.1 Tecnologías usadas.....	4
2.2.2 Entorno de desarrollo	5
2.2.3 Justificación de la elección	5
3. Documentación técnica.....	6
3.1 Análisis	6
3.1.1 Diagrama de casos de uso.....	6
3.1.2 Diagrama de entidad relación	7
3.1.3 Diagrama UML	7
3.2 Pruebas realizadas.....	7
4. Proceso de despliegue	10
4.1 Despliegue hosting	11
4.2 Despliegue local	12
5. Propuestas de mejoras	15
6. Bibliografía.....	16

1. Introducción

Bienvenido al Manual Técnico de la aplicación "RickAndMortyFanHub". Este documento está diseñado para proporcionar una comprensión profunda y detallada de la arquitectura, implementación y mantenimiento de esta aplicación web. Desarrollada utilizando Angular, Node.js, Express.js y MySQL, esta aplicación ofrece una plataforma interactiva para los aficionados de "Rick and Morty", permitiendo explorar información sobre personajes, episodios y mucho más.

A lo largo de este manual, abordaremos cada aspecto técnico del sistema, desde su configuración inicial hasta las pautas para su despliegue y mantenimiento. Este manual servirá como una guía indispensable tanto para los desarrolladores nuevos en el proyecto como para aquellos que buscan profundizar su comprensión de la aplicación.

Espero que este manual le proporcione los conocimientos y herramientas necesarios para trabajar de manera efectiva con "RickAndMortyFanHub", facilitando así una experiencia enriquecedora tanto para los desarrolladores como para los usuarios finales de la aplicación.

2. Arquitectura de la aplicación

2.1 Frontend

2.1.1 Tecnologías usadas

- Angular (v16.2.9): proporcionando una base sólida para el desarrollo de aplicaciones SPA.
- Bootstrap (v5.3.2): para el diseño y estilización de la interfaz de usuario.
- Ng-Bootstrap (v15.1.1) y Ngx-Bootstrap (v11.0.2): para componentes de UI basados en Bootstrap adaptados para Angular.
- FontAwesome (v4.7.0): para iconografía.
- SweetAlert2 (v11.7.32) y Ngx-SweetAlert2 (v12.2.0): para alertas personalizadas y diálogos en la interfaz de usuario.
- RxJS (v7.8.0): para la programación reactiva.

2.1.2 Entorno de desarrollo

- Visual Studio Code: Un editor de código fuente ampliamente utilizado para desarrollo en Angular y TypeScript.
- AngularCLI (v16.2.6)
- TypeScript: (v5.1.6) para el lenguaje de programación.
- Google Chrome: Para pruebas y depuración, especialmente útil con sus herramientas de desarrollador.
- GitHub: Para el control de versiones y la colaboración en el código fuente.

2.1.3 Justificación de la elección

La elección de las tecnologías y herramientas para el frontend de "RickAndMortyFanHub" se justifica por los siguientes motivos:

- Angular (v16.2.9): Seleccionado por su robustez y eficiencia en el desarrollo de aplicaciones SPA (Single Page Applications), lo que facilita una experiencia de usuario fluida y dinámica.
- Bootstrap (v5.3.2): Proporciona un marco de diseño consistente y receptivo para la interfaz de usuario, lo que facilita el desarrollo de un diseño atractivo y adaptable.
- Ng-Bootstrap (v15.1.1) y Ngx-Bootstrap (v11.0.2): Estos complementos integran las capacidades de Bootstrap en el ecosistema Angular, asegurando una mejor cohesión entre el diseño y la funcionalidad.
- Font Awesome (v4.7.0): Ofrece una amplia gama de iconos, mejorando la estética y la interactividad de la interfaz de usuario.
- SweetAlert2 (v11.7.32) y Ngx-SweetAlert2 (v12.2.0): Estas herramientas proporcionan diálogos y alertas personalizadas, mejorando la experiencia del usuario y la interactividad.
- RxJS (v7.8.0): Facilita la programación reactiva, lo que es crucial para manejar múltiples eventos y datos en tiempo real en aplicaciones web modernas.

En cuanto al entorno de desarrollo, se han elegido herramientas que optimizan el proceso de desarrollo, pruebas y colaboración, como Visual Studio Code, Angular CLI, TypeScript, Google Chrome para pruebas y depuración, GitHub para control de versiones y colaboración, y Karma con Jasmine para pruebas automatizadas. Estas herramientas proporcionan un entorno de desarrollo integral y cohesivo, esencial para mantener la calidad y eficiencia del desarrollo del proyecto.

2.2 Backend

2.2.1 Tecnologías usadas

- Node.js (v18.13.0) y Express.js (v4.18.2): para el servidor y el manejo de rutas y APIs.
- MySQL (v2.18.2): para la gestión de la base de datos.
- Bcrypt (v5.1.1): para la encriptación de contraseñas.
- Body-Parser (v1.20.2): para el procesamiento de datos de solicitudes entrantes en el middleware del servidor Express.
- Cors (v2.8.5): para habilitar CORS (Cross-Origin Resource Sharing).
- Jsonwebtoken (v9.0.2) y Jwt-Decode (v4.0.0): para la generación y decodificación de tokens JWT.

2.2.2 Entorno de desarrollo

El entorno de desarrollo del backend está integrado en el mismo ecosistema de

Node.js y Express, utilizando las mismas herramientas y dependencias mencionadas. Además, se incluyen las siguientes herramientas para fortalecer y facilitar el proceso de desarrollo:

- Visual Studio Code: Un editor de código ampliamente utilizado que proporciona soporte para Node.js, JavaScript y otras tecnologías relacionadas, facilitando el desarrollo y la depuración del código.
- phpMyAdmin (v5.2.1): Una herramienta de administración de bases de datos MySQL que permite una gestión más fácil y visual de las bases de datos, utilizada en conjunto con MySQL para manejar las operaciones de la base de datos de manera eficiente.

2.2.3 Justificación de la elección

La elección de las tecnologías para el backend de "RickAndMortyFanHub" se justifica de la siguiente manera:

- Node.js (v18.13.0) y Express.js (v4.18.2): Estas tecnologías son ideales para el desarrollo de aplicaciones web en tiempo real debido a su rendimiento y flexibilidad. Node.js permite un manejo eficiente de operaciones de I/O no bloqueantes, mientras que Express.js ofrece un marco ligero para construir APIs y manejar rutas de manera eficiente.
- MySQL (v8.0.28): Un sistema de gestión de bases de datos relacional robusto y ampliamente utilizado, ideal para manejar datos estructurados y relaciones entre datos.
- Bcrypt (v5.1.1): Proporciona una manera segura de encriptar contraseñas, esencial para mantener la seguridad de la aplicación.
- Body-Parser (v1.20.2): Facilita la interpretación de datos entrantes en el servidor Express, permitiendo una fácil manipulación de los datos de solicitud.
- Cors (v2.8.5): Esencial para permitir solicitudes de origen cruzado de manera segura, lo que es importante en aplicaciones modernas con frontend y backend separados.
- Jsonwebtoken (v9.0.2) y Jwt-Decode (v4.0.0): Estos módulos son fundamentales para la implementación de la autenticación y autorización basada en tokens JWT, asegurando un acceso seguro a los recursos del servidor.

En cuanto al entorno de desarrollo, se han seleccionado Visual Studio Code y phpMyAdmin por su eficiencia y facilidad de uso en el desarrollo y la gestión de bases de datos, respectivamente. Visual Studio Code proporciona un entorno de desarrollo integrado, mientras que phpMyAdmin ofrece una interfaz gráfica para la administración de MySQL.

3. Documentación técnica

3.1 Análisis

3.1.1 Diagrama de casos de uso

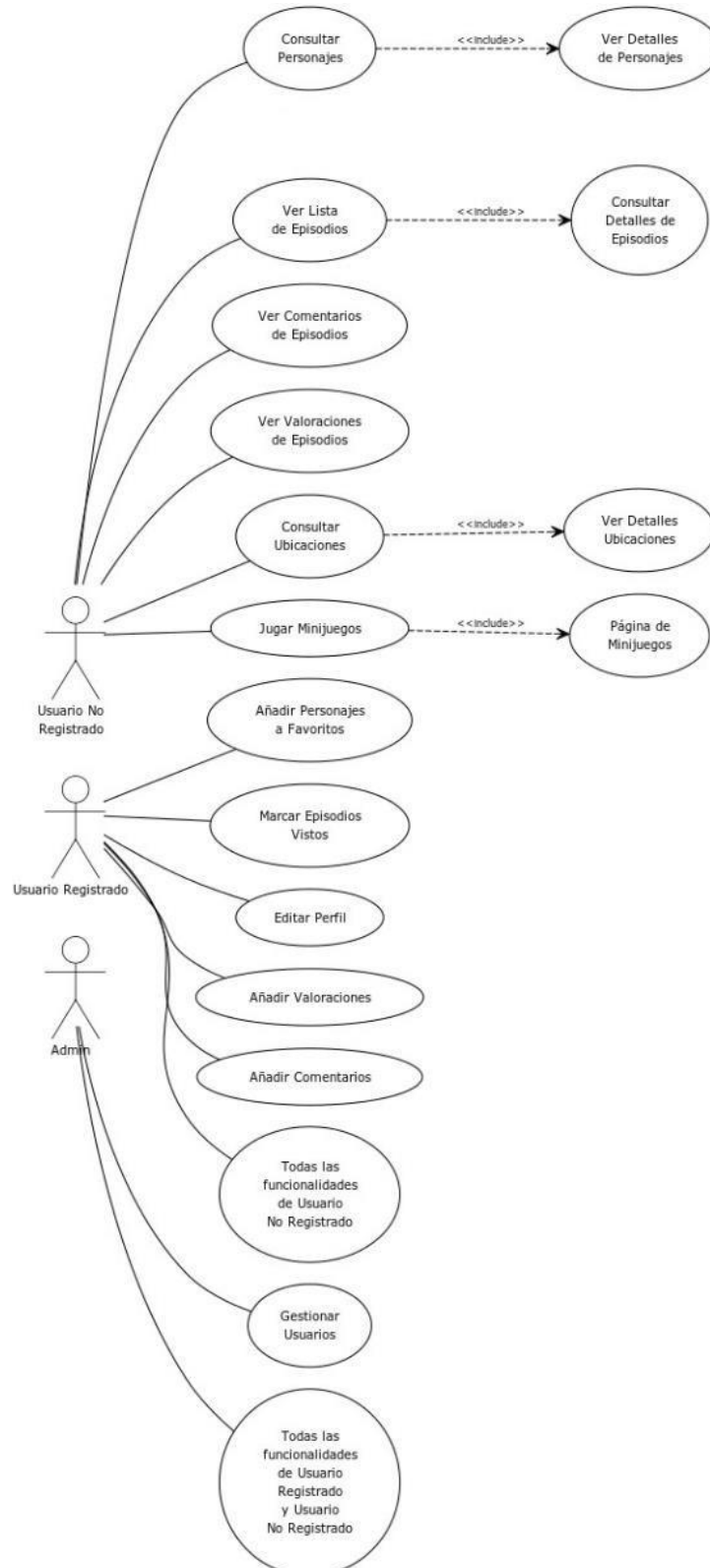


Diagrama de casos de uso.

3.1.2 Diagrama de entidad relación

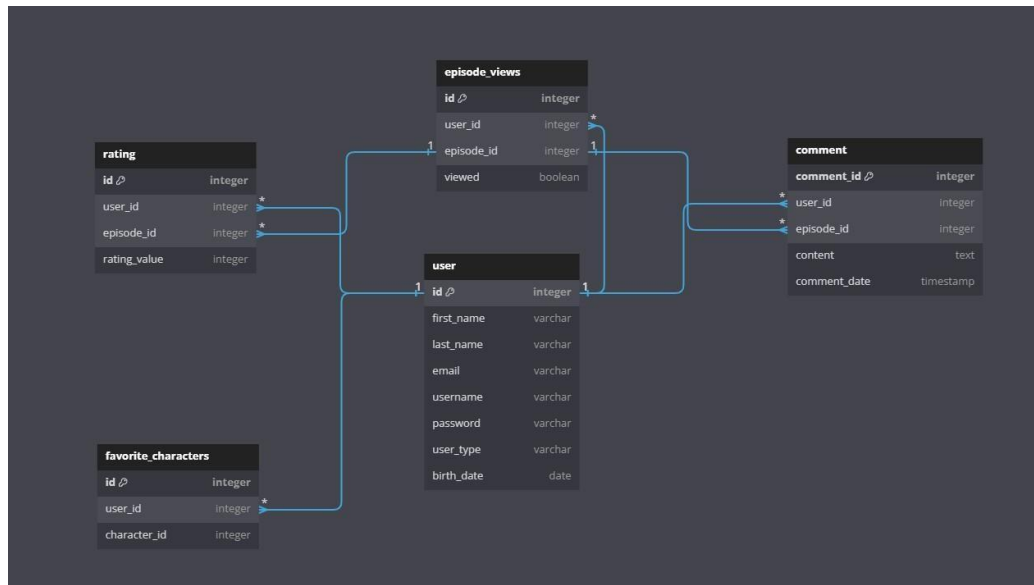


Diagrama E-R.

3.1.3 Diagrama UML

El desarrollo de aplicaciones Angular, al igual que muchos otros frameworks modernos de desarrollo web, sigue un enfoque principalmente basado en componentes y servicios. Mientras que los diagramas UML (Lenguaje Unificado de Modelado) son excelentes para visualizar y planificar la estructura y relaciones en sistemas orientados a objetos, su aplicación puede ser menos directa en arquitecturas basadas en componentes y servicios, como Angular. Es importante notar que un diagrama UML tradicional puede no capturar completamente todas las características específicas de un proyecto Angular, pero voy a intentar proporcionar una visión general de la arquitectura y el diseño de la aplicación.

Debido al gran tamaño de la imagen se facilita un enlace al [diagrama](#).

3.2 Pruebas realizadas

Prueba para CharacterService:

- Componente: CharacterService - getCharacterList()
- Datos de Entrada: N/A (No se requieren datos para este método).
- Resultado de Salida Esperado: Lista de personajes con sus respectivos datos.
- Herramienta de Pruebas: Postman para enviar la solicitud GET al endpoint correspondiente y verificar la lista de personajes.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



Prueba para EpisodeService:

- Componente: EpisodeService - getEpisodeList()
- Datos de Entrada: N/A (No se requieren datos para este método).
- Resultado de Salida Esperado: Lista de episodios con sus respectivos datos.
- Herramienta de Pruebas: Postman para enviar la solicitud GET al endpoint correspondiente y verificar la lista de episodios.

Prueba para LocationService:

- Componente: LocationService - getAllLocations()
- Datos de Entrada: N/A (No se requieren datos para este método).
- Resultado de Salida Esperado: Lista de ubicaciones con sus respectivos datos.
- Herramienta de Pruebas: Postman para enviar la solicitud GET al endpoint correspondiente y verificar la lista de ubicaciones.

Prueba para UserService:

- Componente: UserService - getUserById(id: number)
- Datos de Entrada: id = 1
- Resultado de Salida Esperado: Datos del usuario con el ID proporcionado.
- Herramienta de Pruebas: Postman para enviar la solicitud GET al endpoint correspondiente y verificar los datos del usuario.

Prueba para AuthenticationService (Guard):

- Componente: Guard - canActivate()
- Datos de Entrada: Usuario intenta acceder a una ruta protegida.
- Resultado de Salida Esperado: Respuesta booleana que indica si el usuario puede activar la ruta o no.
- Herramienta de Pruebas: Postman para simular una solicitud a la ruta protegida y verificar si se permite el acceso.

Prueba para CharacterListComponent:

- Componente: CharacterListComponent - onSelectCharacter(character: Character)
- Datos de Entrada: Simulación de selección de un personaje con id.
- Resultado de Salida Esperado: Detalles del personaje con id = 101 mostrados en la interfaz.
- Herramienta de Pruebas: Postman para simular la solicitud y verificar que los detalles del personaje se reciben correctamente.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



Prueba para EpisodeListComponent:

- Componente: EpisodeListComponent - onSelectEpisode(episode:Episode)
- Datos de Entrada: Simulación de selección de un episodio con id =25.
- Resultado de Salida Esperado: Detalles del episodio con id = 25mostrados en la interfaz.
- Herramienta de Pruebas: Postman para simular la solicitud y verificar que los detalles del episodio se reciben correctamente.

Prueba para LocationListComponent:

- Componente: LocationListComponent - onSelectLocation(location:Location)
- Datos de Entrada: Simulación de selección de una ubicación con id = 12.
- Resultado de Salida Esperado: Detalles de la ubicación con id = 12mostrados en la interfaz.
- Herramienta de Pruebas: Postman para simular la solicitud y verificar que los detalles de la ubicación se reciben correctamente.

Prueba para RegisterComponent:

- Componente: RegisterComponent - onSubmit(formGroup:FormGroup)
- Datos de Entrada: Simulación de envío de formulario de registro con datos de un nuevo usuario.
- Resultado de Salida Esperado: Confirmación de registro exitoso.
- Herramienta de Pruebas: Postman para enviar la solicitud de registro y verificar la respuesta de confirmación.

Prueba para LoginComponent:

- Componente: LoginComponent - onLogin(formGroup:FormGroup)
- Datos de Entrada: Simulación de envío de formulario de inicio de sesión con credenciales de usuario.
- Resultado de Salida Esperado: Confirmación de inicio de sesión exitoso y redirección a la página principal.
- Herramienta de Pruebas: Postman para enviar la solicitud de inicio de sesión y verificar la respuesta y redirección.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



Prueba para EditProfileComponent:

- Componente: EditProfileComponent - onSubmit(formGroup:FormGroup)
- Datos de Entrada: Simulación de envío de formulario de edición de perfil con datos actualizados de usuario.
- Resultado de Salida Esperado: Confirmación de actualización de perfil exitoso.
- Herramienta de Pruebas: Postman para enviar la solicitud de actualización de perfil y verificar la respuesta de confirmación.

Prueba para EditProfileComponent:

- Componente: EditProfileComponent - onSubmit(formGroup:FormGroup)
- Datos de Entrada: Simulación de envío de formulario de edición de perfil con datos actualizados de usuario.
- Resultado de Salida Esperado: Confirmación de actualización de perfil exitoso.
- Herramienta de Pruebas: Postman para enviar la solicitud de actualización de perfil y verificar la respuesta de confirmación.

Pruebas Blackbox UI/UAT :

- Pruebas funcionales: Pruebas para verificar que todas las características de la app, como la creación de perfiles de usuario, la navegación entre episodios y personajes, etc. funcionan exactamente como se espera según sus especificaciones.
- Pruebas de usabilidad: Se ha evaluado la experiencia del usuario interactuando con la app. Por ejemplo, se ha comprobado si la selección de un personaje o episodio es intuitiva y si la información se muestra de manera clara y accesible.
- Pruebas de compatibilidad: Dado que los usuarios pueden acceder a la aplicación desde diversos dispositivos y navegadores, se han realizado pruebas para asegurar que la app se vea y funcione bien en cada uno de estos entornos, manteniendo un rendimiento y diseño consistentes.
- Pruebas de regresión: Tras cualquier actualización o cambio en la app, se han efectuado pruebas para confirmar que las nuevas funcionalidades o correcciones no han afectado negativamente a las ya existentes, garantizando que la app sigue siendo estable y fiable.

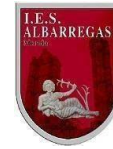
4. Proceso de despliegue

La aplicación se encuentra desplegada en el dominio <https://rick-and-morty-fan-hub.vercel.app/home>

Se ha creado un usuario con permisos de administrador para las pruebas: Usuario: Test
Contraseña: Test123.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



4.1 Despliegue hosting

Se ha realizado un despliegue de los diferentes componentes del proyecto de manera eficiente y escalable. La base de datos ha sido desplegada en Hostinger, aprovechando su robusta infraestructura y su facilidad de manejo. Esto asegura una conexión segura y un rendimiento óptimo para el almacenamiento y manejo de datos.

Tablero principal - Hosting - rick-and-morty-fan-hub.online

rick-and-morty-fan-hub.online
Sitio web creado el 2023-11-16 [Mejorar plan](#)

Premium Web Hosting
Activo
[Ver detalles](#)

Dominio
Activo
[Administrar](#)

Email gratuito
Configuración pendiente
[Configurar](#)

Copias de seguridad diarias
Inhabilitado
[Administrar](#)

Puntaje de rendimiento
100
Dispositivo de escritorio
Último escaneo el 2023-11-21
[Ejecutar prueba de velocidad](#)

Administrador de archivos

Bases de datos

Autoinstalador

Tu sitio web funciona correctamente
No se ha encontrado ningún problema

Administración - Hosting - rick-and-morty-fan-hub.online - Bases de datos - Administración

+ Crear nueva base de datos MySQL y usuario de base de datos

Nombre de la base de datos MySQL: Nombre de la base de datos

Nombre de usuario MySQL: Nombre de usuario

Contraseña: [Ojo](#)

[✓ Crear](#)

Lista de bases de datos y usuarios actuales de MySQL

Base de Datos MySQL	Usuario MySQL	Creado en	Sitio web	
u496649649_RickAndM orty 1 MB	u496649649_admin	2023-11-16	rick-and-morty-fan- hub.online	Ingresar a phpMyAdmin ⋮

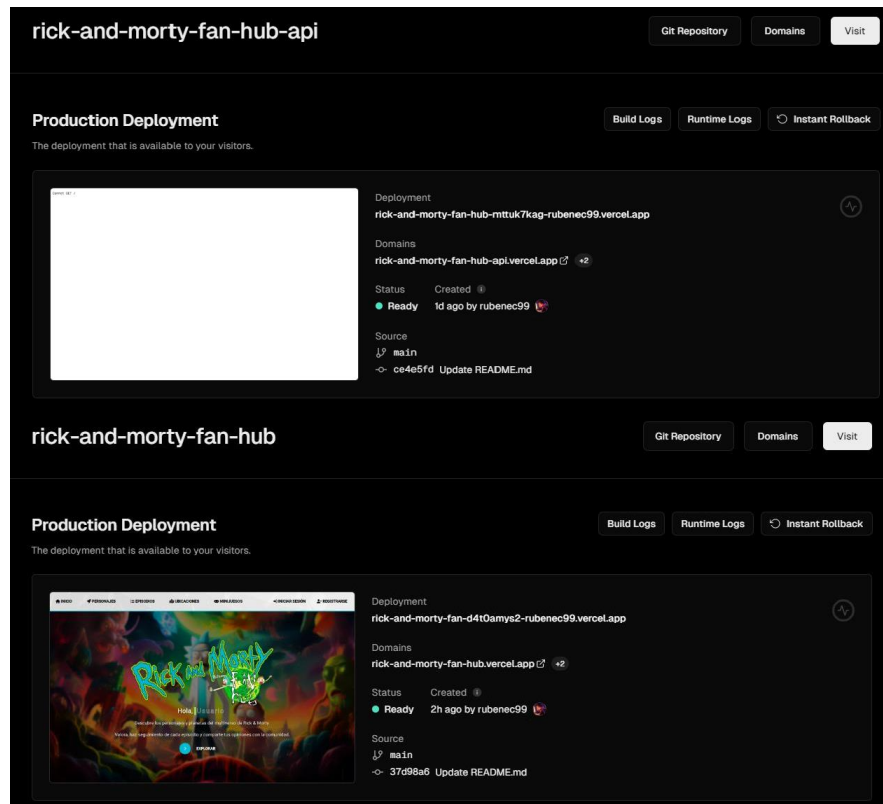
Panel de control de Hostinger.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



Por otro lado, tanto el backend como el frontend del proyecto han sido desplegados en Vercel. Vercel proporciona un entorno ideal para aplicaciones desarrolladas con tecnologías como Node.js y Angular, ofreciendo una integración continua, un despliegue automático y una alta disponibilidad, lo que resulta en una experiencia de usuario fluida y profesional.



Panel de control de Vercel.

4.2 Despliegue local

Siguiendo estos pasos podrás obtener una copia del proyecto RickAndMortyFanHub y configurar la base de datos utilizando phpMyAdmin con XAMPP:

Configuración del Entorno de Desarrollo:

Antes de clonar los repositorios, asegúrese de tener un entorno de desarrollo adecuado.

1. Instalación de Node.js y Angular CLI:

- Comience por asegurarse de que [Node.js](#) está instalado. Puede descargar la última versión estable desde el sitio oficial de Node.js.
- Una vez instalado Node.js, abra la terminal y ejecute `npm install -g @angular/cli` para instalar Angular CLI globalmente.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



2. Instalación de Visual Studio Code:

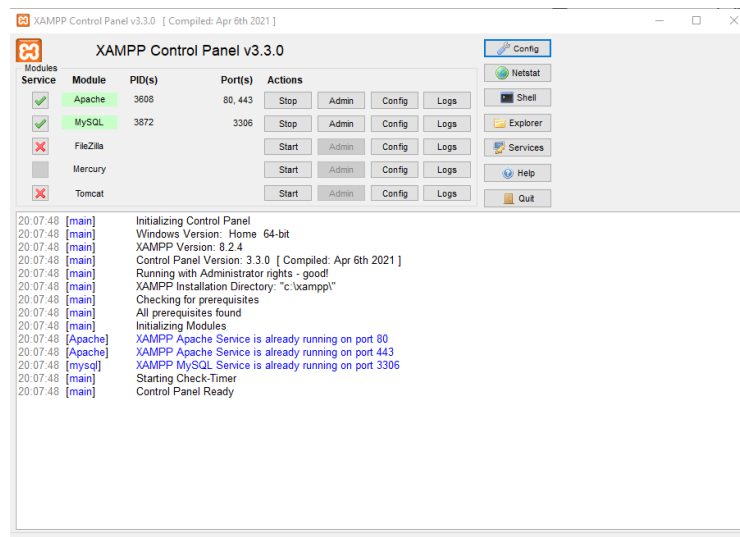
- Descargue [Visual Studio Code](#) desde su página oficial, e instale el editor siguiendo las instrucciones para su sistema operativo.

3. Instalación de XAMPP:

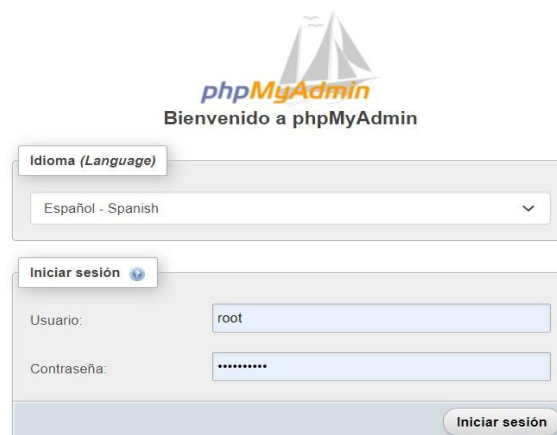
- Obtenga [XAMPP](#) desde Apache Friends y complete su instalación.

Configuración de la base de datos con phpMyAdmin y XAMPP:

- Abra XAMPP e inicie los servicios de Apache y MySQL. Acceda a phpMyAdmin a través de <http://localhost/phpmyadmin> o eligiendo la opción admin en XAMPP, utilizando 'root' como usuario y dejando el campo de la contraseña en blanco, en caso de no acceder, en la contraseña utilizaremos “password”, puesto que es la que utiliza por defecto phpMyAdmin. En mi caso la contraseña es “admin”, ya que es la que he establecido. Es conveniente recordar la contraseña si se modifica, puesto que la necesitaremos más tarde.



Panel de control de XAMPP.



Pantalla de inicio de sesión de phpMyAdmin.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



- En el panel de phpMyAdmin, cree una nueva base de datos con el nombre rickandmortyfanhubdb y seleccione utf8_spanish_ci como el conjunto de caracteres.

Bases de datos

Pantalla de creación de base de datos de phpMyAdmin.

- Con la base de datos seleccionada en phpMyAdmin tendrás que dirigirte a la página importar y seleccionar [este archivo .sql](#). Con esto tendremos una inserción instantánea de las tablas que componen la bbdd con algunos registros.

Importando al servidor actual

Pantalla de importación de phpMyAdmin.

Configuración del proyecto RickAndMortyFanHub:

1. Inicio de Visual Studio Code:

- Abra Visual Studio Code y seleccione 'Archivo' > 'Abrir Carpeta...' para crear una nueva carpeta donde residirá el proyecto.

2. Clonación del repositorio:

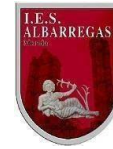
- Dirígete a [este repositorio](#) de GitHub para obtener el Frontend de la aplicación.
- Dirígete a [este repositorio](#) de GitHub para obtener el Backend de la aplicación.
- Introduce estas dos carpetas dentro de la creada anteriormente en Visual Studio Code o el IDE que hayas decidido usar.

3. Instalación de dependencias:

- Dentro de la terminal del IDE, navegue a la carpeta del backend y ejecute el comando `npm install` para instalar las dependencias necesarias.
- Repita el proceso para el frontend, asegurándose de estar en la carpeta correcta antes de ejecutar el comando de instalación.

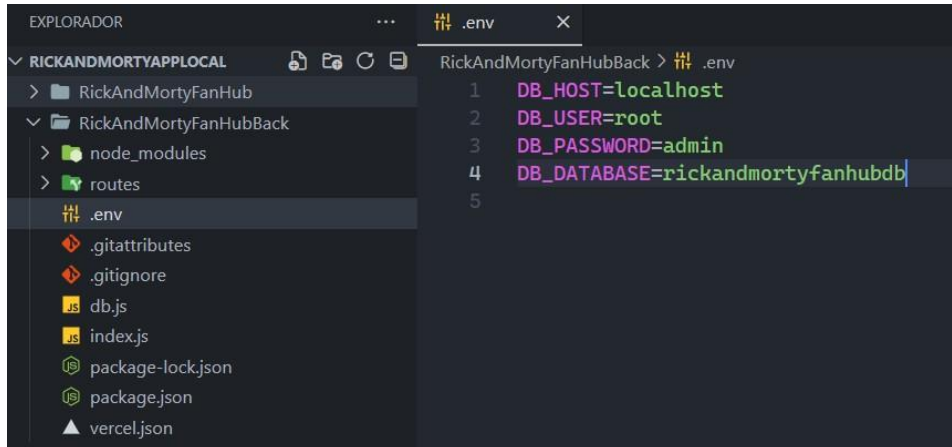
Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



4. Creación del .env:

- Dentro del IDE, navegue a la carpeta del backend y cree un archivo llamado .env, en el que introduciremos las variables de entorno para la base de datos. Esto se hace por seguridad, puesto que a GitHub no se sube este archivo.



```
1 DB_HOST=localhost
2 DB_USER=root
3 DB_PASSWORD=admin
4 DB_DATABASE=rickandmortyfanhubdb
5
```

Archivo .env de un proyecto backend.

5. Puesta en marcha de los servidores:

- En la terminal del backend, ejecute `node index.js` para iniciar el servidor de backend.
- Para el frontend, abra una nueva terminal en la carpeta correspondiente y ejecute `ng serve`. Esto iniciará el servidor de desarrollo de Angular.

Verificación final:

Una vez que ambos servidores estén funcionando, abra un navegador y vaya a <http://localhost:4200/> para ver la aplicación en funcionamiento.

Siguiendo estos pasos meticulosamente, podrá configurar tanto el entorno de desarrollo como la aplicación RickAndMortyFanHub, preparándola para pruebas y desarrollo adicional.

5. Propuestas de mejoras

1. Optimización para dispositivos móviles

- Mejorar la optimización general de la aplicación en dispositivos móviles.
- Utilizar técnicas de carga diferida (lazy loading) para mejorar los tiempos de carga en dispositivos con conexiones más lentas.

2. Funciones interactivas

- Añadir un sistema de recomendaciones personalizadas basado en las preferencias y el historial de visualización del usuario.
- Implementar funciones sociales como compartir en redes sociales.

3. Optimización del rendimiento

- Realizar más pruebas de rendimiento y optimizar consultas para un acceso más rápido a los datos.
- Implementar técnicas de optimización como el uso de Service Workers para mejorar la velocidad de carga y funcionamiento offline.

4. Seguridad

- Reforzar la seguridad en el backend, incluyendo la protección contra ataques comunes como inyección SQL o XSS.

5. Accesibilidad

- Mejorar la accesibilidad de la aplicación siguiendo las pautas WCAG para garantizar que sea usable por personas con diversas capacidades.

6. Internacionalización

- Implementar soporte multilinguaje para atraer a una audiencia global.

7. Puntuaciones y clasificaciones

- Añadir un sistema de puntuaciones y clasificaciones en el apartado de minijuegos, para aumentar la competencia y el compromiso de los usuarios.

8. Funcionalidades administrativas ampliadas (edición completa de usuarios)

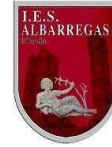
- Ampliar las capacidades administrativas para incluir la posibilidad de editar completamente los perfiles de usuario, proporcionando un control más exhaustivo sobre la gestión de usuarios.

6. Bibliografía

- Angular. (s. f.). <https://angular.io/docs>
- Animista. (s. f.). <https://animista.net/>
- Bootstrap. (s. f.). <https://getbootstrap.com/>
- GitHub. (s. f.). <https://github.com/>
- MySQL: MySQL Documentation. (s. f.). <https://dev.mysql.com/doc/>
- NPM: @angular-devkit/core. (s. f.). npm. <https://www.npmjs.com/package/@angular-devkit/core>
- NPM: @angular-devkit/core. (s. f.). npm. <https://www.npmjs.com/package/@angular-devkit/core>
- NPM: @auth0/angular-jwt. (s. f.). npm. <https://www.npmjs.com/package/%40auth0/angular-jwt>
- NPM: BCrypt. (s. f.). npm. <https://www.npmjs.com/package/bcrypt>
- NPM: Body-parser. (s. f.). npm. <https://www.npmjs.com/package/body-parser>
- NPM: Cors. (s. f.). npm. <https://www.npmjs.com/package/cors>

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Rick And Morty Fan Hub



- NPM: Dotenv. (s. f.). npm. <https://www.npmjs.com/package/dotenv>
- NPM: Express. (s. f.). npm. <https://www.npmjs.com/package/express>
- NPM: Jsonwebtoken. (s. f.). npm. <https://www.npmjs.com/package/jsonwebtoken>
- NPM: JWT-decode. (s. f.). npm. <https://www.npmjs.com/package/jwt-decode>
- NPM: MySQL. (s. f.). npm. <https://www.npmjs.com/package/mysql>
- NPM: @ng-bootstrap/ng-bootstrap. (s. f.). npm. <https://www.npmjs.com/package/%40ng-bootstrap/ng-bootstrap>
- NPM: NGX-Bootstrap. (s. f.). npm. <https://www.npmjs.com/package/ngx-bootstrap>
- NPM: NGX-order-pipe. (s. f.). npm. <https://www.npmjs.com/package/ngx-order-pipe>
- NPM: @sweetalert2/ngx-sweetalert2. (s. f.). npm. <https://www.npmjs.com/package/%40sweetalert2/ngx-sweetalert2>
- NPM: TSLib. (s. f.). npm. <https://www.npmjs.com/package/tslib>