



INSTITUTO DE EDUCACIÓN SECUNDARIA CAMPANILLAS

Grado Superior Desarrollo de Aplicaciones Web (FP DUAL / Bilingüe)



FIGHT! My Hero

Memoria Proyecto DAW

Realizado por
M^a de las Mercedes Perea Roper

MÁLAGA, 11 JUNIO DE 2020



ÍNDICE

1. Presentación del proyecto	3
1.1 Introducción	3
1.2 Objetivos	3
1.3 Contexto	4
1.4 Planteamiento del problema	4
1.4 Principales tecnologías y herramientas empleadas	5
2. Descripción general	7
2.1 Perspectiva del producto	7
2.2 Funcionalidades del producto	7
2.3 Características del usuario	9
2.4 Restricciones, supuestos y dependencias	9
3. Requisitos específicos	9
3.1. Requerimientos funcionales	9
3.2 Definición de interfaces de usuario	20
3.2.1 Home	21
3.2.2 Detalle Héroe	22
3.2.3 Búsqueda de héroe/usuario	23
3.2.4 Perfil público	23
3.2.5 Perfil privado del usuario	24
3.2.6 About us	25
3.2.7 Fight Hero	25
3.2.8 Fight Team	26
3.2.9 Panel de administrador	26
3.2.10 Panel de administrador - Añadir Héroe	27
3.2.11 Panel de administrador - Editar Héroe	27
3.2.12 Login	28
3.2.13 Registro	28
3.3 Interfaces hardware	29
3.4 Interfaces software	29
3.5 Interfaces de comunicaciones	29
3.6 Estándares cumplidos	29
3.7 Seguridad	29
3.8 Facilidades de mantenimiento	30
3.9 Planificación temporal	30
4. Diseño de la base de datos	30
5. Diagrama de casos de uso	31



6. Diseño	34
7. Documentación del código de la aplicación	35
8. SCV	35
9. DESPLIEGUE	37
9.1 DESPLIEGUE LOCAL	37
9.2 DESPLIEGUE LOCAL EN APACHE	38
9.3 DESPLIEGUE REMOTO	43
9.3.1 Plataforma	43
9.3.2 Configuración	43
9.3.3 Despliegue Automatizado	45
9.3.4 Uso SSH	46
10. Conclusiones	46
11. Bibliografía / Webgrafía	47



1. Presentación del proyecto

1.1 Introducción

Este documento describe el proyecto realizado para el módulo de Proyecto de Desarrollo de Aplicaciones Web en el Instituto de Educación Secundaria Campanillas, cursando el Grado Superior de Desarrollo de Aplicaciones Web. El proyecto realizado se trata de una aplicación web que es accesible desde cualquier navegador web, y está realizado con tecnologías punteras en el entorno de la programación web actual, buscando crear una aplicación que resulte ágil, dinámica y en la cual se vea reflejado todo el conocimiento adquirido durante la formación en este grado.

1.2 Objetivos

El proyecto realizado se trata de una aplicación web que tiene como objetivo ser un entorno donde los amantes de cómics, películas del género de fantasía y ciencia ficción, y de la cultura *pop* en general puedan participar. Se pretende crear un espacio en el que los usuarios puedan buscar a sus personajes preferidos, ver información sobre ellos, almacenarlos, dar su opinión, votarlos e interactuar con ellos bajo diferentes acciones. Esta aplicación también tiene carácter de red social ya que los usuarios podrán, entre otras cosas, buscar a otros usuarios e interactuar con ellos.

En cuanto a los usuarios que pueden participar dentro de esta aplicación encontraremos tres roles principales:

- Usuarios no registrados, que únicamente podrán acceder a la información general del *Home*, realizar búsqueda de personajes y ver información sobre ellos.
- Usuarios registrados, que podrán realizar las mismas acciones que los no registrados, sumándole que pueden realizar búsquedas de otros usuarios, ver perfiles públicos y seguirlos. También podrán comentar, votar, añadir a favoritos y seguir a personajes (héroes). Por otro lado, también podrán crear su equipo de héroes y participar en las *Fight*.
- Usuario Administrador, que podrá realizar las mismas funcionalidades que los dos anteriores, pudiendo además acceder al panel de administrador, donde podrá ver estadísticas de la web y realizar



diferentes acciones, como son añadir nuevos héroes, modificar su información o eliminar usuarios de la aplicación.

1.3 Contexto

El proyecto ha sido realizado como idea surgida tras hablar con varios coleccionistas de cómics y propietarios de empresas dedicadas al sector, y viendo la oportunidad de crear un espacio donde los amantes de la cultura *pop* puedan buscar información referente a sus personajes preferidos e interactuar con ellos. Uno de los aspectos principales de esta aplicación es poder enfrentar héroes entre sí, tanto de forma individual como por equipos, ya que es muy común en estos ambientes hablar y comparar qué superhéroes son mejores que otros, quienes tienen mejores estadísticas, y quién ganaría en un combate; por lo tanto, añadiendo esta funcionalidad, el usuario no solo puede ver información sobre los héroes, sino además interactuar con ellos y dar respuestas a esos debates.

1.4 Planteamiento del problema

El problema que se nos ha planteado es la creación de un sitio web en el que los usuarios puedan participar de forma dinámica para obtener información sobre las estadísticas, habilidades, historia, etc. de sus personajes de ficción preferidos (a los que, a lo largo de este documento, se denominarán como *héroes*), pudiendo añadir comentarios sobre ellos, almacenarlos y realizar diferentes acciones, como son enfrentar a unos personajes con otros para ver cual tiene mayores estadísticas. También se busca crear un espacio en el que los usuarios puedan buscar otros usuarios e interactuar con ellos de diversas maneras. Las funcionalidades, a grandes rasgos, serían las siguientes:

- Mostrar información general de los *héroes*.
- Realizar búsquedas de usuarios y *héroes*.
- Seguir a usuarios y *héroes*.
- Añadir *héroes* a favoritos.
- Comentar *héroes*.
- Puntuar *héroes*.
- Crear tu propio equipo de *héroes*.
- Enfrentar en combate a *héroes* entre ellos o a equipos entre ellos.
- Permitir al administrador añadir *héroes*.
- Permitir al administrador modificar *héroes*.
- Permitir al administrador eliminar usuarios.



1.4 Principales tecnologías y herramientas empleadas

En cuanto a las tecnologías y herramientas utilizadas para la realización de este proyecto, se han optado por el uso de tecnologías y herramientas que se están demandando mucho en el mercado actual de este sector.

Los lenguajes de programación que se han utilizado son TypeScript y JavaScript. Más específicamente, se ha decidido que para la parte *backend* de nuestra aplicación se use Node.js, que es un entorno de ejecución de JavaScript orientado a eventos asíncronos; Node.js está diseñado para crear aplicaciones network escalables. Para trabajar en Node.js utilizaremos el framework Express.js: este framework está respaldado por una gran comunidad de desarrolladores, lo que garantiza la continua actualización de sus características principales; Express.js es un framework minimalista que permite crear aplicaciones web y APIs rápidamente, ayuda a los usuarios a configurar rutas para enviar y recibir solicitudes entre la vista *Front* y la *Base de Datos* actuando como un servidor HTTP y, así mismo, es compatible con muchos paquetes *npm*, por lo tanto tiene muchísimas posibilidades para trabajar con él.

Para la parte *frontend* de la aplicación se ha decidido utilizar Angular 8 (TypeScript): Angular es uno de los frameworks más demandados del sector actual, de código abierto y mantenido por Google; tiene una gran comunidad de desarrolladores que lo soportan y hacen que trabajar con él sea muy cómodo. Utilizando Angular CLI se permite comenzar la creación de un proyecto de forma muy rápida con pocos pasos, y la modularidad de sus componentes permite crear elementos muy reutilizables lo que permite una programación más dinámica y escalable; por todo esto y por muchas más cualidades que tiene este framework, ha sido elegido para la creación de este proyecto.

Gracias al uso de estos lenguajes se ha conseguido crear una aplicación robusta de forma rápida y sin grandes problemas durante su desarrollo.

Se ha utilizado también HTML para dar formas a las vistas de la aplicación.

Para dar estilos a los diferentes elementos se ha utilizado CSS mediante el procesador SASS, que nos permite trabajar de forma mucho más dinámica y eficiente con nuestros estilos; se han implementado diferentes animaciones y efectos conseguidos gracias a las múltiples posibilidades que nos da CSS3.



En algunos casos se ha utilizado jQuery para añadir efectos concretos en algunos apartados, como es en el caso de la integración del vídeo en el apartado *About us* de la aplicación.

Como apoyo para el diseño, se han utilizado las bibliotecas de Bootstrap y de Angular Material para la creación de algunos elementos, como son los formularios, la barra de navegación y las alertas.

Como Base de datos se ha optado por MySQL, que es libre y muy potente, y es además la base de datos que más hemos estado trabajando durante lo cursado en este grado; es una base de datos relacional que nos permite trabajar de forma muy intuitiva, creando elementos, modificándolos y eliminándolos sin ningún tipo de restricción.

Para el control de versiones se ha utilizado Git: para ello hemos utilizado el software *Sourcetree*, que es un cliente GUI para manejar repositorios Git gratuito y muy intuitivo, que permite realizar todas las acciones para trabajar con Git sin tener que depender de la consola de comandos.

Como entorno de desarrollo, hemos utilizado Visual Studio Code, ya que es un entorno gratuito con muchísimos módulos que ayudan en diferentes tareas, permitiendo escribir código de una forma cómoda y veloz. Visual Studio Code nos permite además abrir una terminal dentro de él, que nos ayudará a crear los elementos y movernos dentro del proyecto de una forma muy rápida.

Para el despliegue remoto, se ha utilizado la plataforma IBM Cloud: es una plataforma mantenida por IBM y por lo tanto es muy estable, que tiene opciones gratuitas y una gran cantidad de herramientas que permiten que el despliegue de las diferentes partes de la aplicación resulte muy intuitiva.

Para el Almacenamiento remoto de la BD se ha decidido usar Clever Cloud, que es una plataforma para desarrolladores que permite el almacenamiento de bases de datos MySQL con una interfaz elegante y cómoda, teniendo opciones de almacenamiento gratuito de gran capacidad y estable.

También se ha implementado Swagger para documentar y probar las diferentes funcionalidades de la API de Node.js que se ha construido.



Finalmente, esta aplicación, a parte de depender de los diferentes elementos que se han creado para conformar su estructura, también se encuentra conectada a la API gratuita <https://www.superheroapi.com/>, de donde se saca principalmente la mayoría de la información de los diferentes héroes que en la aplicación se muestran.

2. Descripción general

2.1 Perspectiva del producto

La aplicación web desarrollada pretende ser un medio en el que los amantes de los personajes de ficción puedan realizar búsquedas sobre ellos y realizar diferentes acciones sobre ellos, creando también un entorno donde poder interactuar con otros usuarios con las mismas aficiones.

2.2 Funcionalidades del producto

Las funcionalidades que conforman la aplicación se pueden distinguir según el tipo de usuario que se encuentre en ella, y son:

- Usuarios no registrados:
 - Autenticación: cuando el usuario no registrado intente hacer una acción que dependa de su autenticación se le informará de que debe registrarse o loguearse para acceder a ella.
 - Registro en la aplicación, mediante un formulario.
 - Visionado del *Home* y de la diversa información que se encuentra en él.
 - Búsqueda de *héroes* a través de la barra de navegación que se encuentra en la parte superior de la aplicación.
 - Visionado de la información específica de un *héroe* concreto.
- Usuarios registrados:
 - Cerrar sesión: se podrá cerrar sesión mediante un botón definido en la barra de navegación, que una vez pulsado redirigirá al *Home*.
 - Comentar *héroe*: podrá comentar, editar un comentario propio y eliminar un comentario propio dentro de la página específica del *héroe*.



- Votar *héroe*: podrá votar un *héroe* dentro de la página específica de ese personaje.
- Seguir a un *héroe*: podrá seguir o dejar de seguir a un personaje dentro de la página específica de ese personaje, donde se le mostrará un botón para realizar las diferentes acciones.
- Añadir o eliminar de favoritos un *héroe*: dentro de la página específica de ese personaje se mostrará un botón para realizar las diferentes acciones.
- Crear un equipo: el usuario registrado podrá crear un equipo con sus personajes favoritos, pudiendo añadir y editar un nombre para el equipo, y añadir o eliminar miembros del equipo en cualquier momento; este equipo podrá ser visto por otros usuarios en el perfil público del usuario.
- Buscar usuarios: en el buscador situado en la parte superior de la página podrá buscar otros usuarios y ver sus perfiles públicos.
- Seguir o dejar de seguir a otros usuarios: dentro del perfil público de cada usuario se encontrarán botones que permiten hacer las diferentes acciones.
- Editar perfil del usuario: en el apartado correspondiente de la aplicación encontrará un lugar donde poder ver sus datos y editarlos, pudiendo además cambiar de contraseña o eliminar su cuenta.
- Enfrentar héroes: dentro del apartado *Fight* podrá realizar enfrentamientos entre dos héroes que se seleccionen, y así ver quién sería el vencedor en un combate entre ambos, basándose en sus estadísticas.
- Enfrentar equipos: dentro del apartado *Fight* podrá realizar enfrentamientos entre los equipos creados por los usuarios y ver qué equipo de *héroes* sería el vencedor en un combate, basándose también en sus estadísticas.
- *Usuario Administrador*:
 - Ver estadísticas de la aplicación: se podrá consultar el número de usuarios totales, el número de personajes almacenados y el número de equipos creados.
 - Añadir un *héroe* nuevo a la base de datos: mediante el cumplimiento de un formulario se podrán añadir nuevos personajes a la aplicación.
 - Editar *héroe*: podrá editar la información de los *héroes* ya almacenados en la base de datos.
 - Ver lista de usuarios de la aplicación.



- Eliminar usuarios de la aplicación.

2.3 Características del usuario

Se pueden definir dos tipos de usuarios principales: Los no registrados y los registrados; dentro de los registrados podemos diferenciar dos subtipos:

- Usuarios NO registrados: solamente tendrán acceso a datos generales de la aplicación y no a ninguna de sus funcionalidades más destacables.
- Usuarios Registrados: son los que tienen acceso a las funcionalidades más interesantes de la aplicación, diferenciando entre usuarios registrados generales y usuarios administradores:
 - Usuarios Registrados Generales: pueden realizar todas las funcionalidades de la aplicación, pero sin tener acceso a la parte de administración de la web.
 - Usuarios Administradores: pueden acceder además al apartado de administración de la web y realizar diferentes acciones consideradas sensibles para el correcto funcionamiento de la aplicación.

2.4 Restricciones, supuestos y dependencias

El origen natural de esta aplicación web hace que se requiera de un ordenador, teléfono o tablet con navegador, y de una conexión a internet para poder tener acceso a ella.

La aplicación ha sido desarrollada utilizando el navegador Google Chrome, y aunque se han ido haciendo pruebas en otros navegadores como Safari, algunos aspectos estéticos de la aplicación pueden variar según el navegador en donde se esté visionando. La aplicación ha sido a su vez desarrollada en Windows, aunque también se han realizado pruebas de compatibilidad en MacOS.

3. Requisitos específicos

3.1. Requerimientos funcionales

Paso a describir mediante una serie de tablas las diversas funciones que ofrece esta aplicación web, dispuesta según el tipo de usuario:

**Usuarios no registrados**

Apartado	Descripción
Título	Autenticación
Propósito	Acceder a apartados concretos de la app
Entrada	Email y contraseña
Proceso	Se comprueba la existencia del usuario
Salida	Permite el acceso a la aplicación o devuelve un error si no es válido.

Apartado	Descripción
Título	Registro
Propósito	Registrar al usuario dentro de la aplicación
Entrada	Alias, Email y contraseña
Proceso	Registrar al usuario en la aplicación
Salida	Si los datos son correctos se registra al usuario en la aplicación y se lleva al login; si no, se devuelve un error avisando de qué ha ocurrido.

Apartado	Descripción
Título	Buscar héroes
Propósito	Buscar héroes dentro de la aplicación
Entrada	Nombre del héroe
Proceso	Búsqueda de héroe por nombre dentro de la base de datos



Salida	Si el héroe se encuentra dentro de la aplicación se mostrará un detalle de él para que el usuario pueda acceder a su información; si no existe, se avisará de ello.
--------	---

Apartado	Descripción
Título	Consultar información de un héroe
Propósito	Ver información específica de un héroe concreto
Entrada	-
Proceso	Se muestra la información referente a un héroe específico extraída de la DB mediante la id del héroe
Salida	Información referente al héroe, datos biográficos, puntuación media dada por los usuarios, comentarios, etc...

Usuarios registrado

Apartado	Descripción
Título	Buscar usuarios
Propósito	Buscar usuarios dentro de la aplicación
Entrada	Nombre del usuario
Proceso	Busca al usuario por nombre dentro de la base de datos
Salida	Si el usuario se encuentra dentro de la aplicación se mostrará un detalle de él para que el usuario pueda acceder a su información; si no existe, se avisará de ello.



Apartado	Descripción
Título	Consultar información de un usuario
Propósito	Ver información específica de un usuario concreto
Entrada	-
Proceso	Se muestra la información referente a un usuario específico extraída de la base de datos mediante la id del usuario
Salida	Información referente al usuario, datos biográficos, número de seguidores y seguidos y sus nombres, e información de su Equipo.

Apartado	Descripción
Título	Seguir a un usuario
Propósito	Sigue a un usuario concreto
Entrada	-
Proceso	Se añade una relación como seguidor de este usuario referente al usuario que está siguiendo en la base de datos
Salida	Se podrá ver qué usuarios se están siguiendo tanto en el perfil público como en el privado del usuario.

Apartado	Descripción
Título	Dejar de seguir a un usuario
Propósito	Deja de seguir a un usuario concreto al cual se estaba siguiendo



Entrada	-
Proceso	Se elimina la relación como seguidor de este usuario referente al usuario que está siguiendo en la base de datos
Salida	Se podrá ver qué usuarios se están siguiendo tanto en el perfil público como en el perfil privado del usuario.

Apartado	Descripción
Título	Seguir a un <i>héroe</i>
Propósito	Sigue a un héroe concreto
Entrada	-
Proceso	Se añade una relación como seguidor de este usuario referente al héroe que está siguiendo en la base de datos
Salida	Se podrá ver que <i>héroes</i> se están siguiendo en el apartado <i>My Heroes</i> de la aplicación.

Apartado	Descripción
Título	Dejar de seguir a un <i>héroe</i>
Propósito	Deja de seguir a un héroe concreto al cual estaba siguiendo
Entrada	-
Proceso	Se elimina la relación como seguidor de este usuario referente al héroe que está siguiendo en la base de datos
Salida	Se podrá ver que <i>héroes</i> se están siguiendo en el apartado <i>My Heroes</i> de la aplicación.



Apartado	Descripción
Título	Añadir a favoritos a un <i>héroe</i>
Propósito	Añadir a favoritos a un héroe concreto
Entrada	-
Proceso	Se añade una relación como favorito de este usuario referente al héroe que está siguiendo en la base de datos
Salida	Se podrá ver que <i>héroes</i> están en favoritos en el apartado <i>My Heroes</i> de la aplicación.

Apartado	Descripción
Título	Eliminar un <i>héroe</i> favorito
Propósito	El usuario elimina de favoritos a un héroe al cual tenía en su lista de favoritos
Entrada	-
Proceso	Se elimina la relación como favorito de este usuario referente al héroe que está siguiendo en la base de datos
Salida	Se podrá ver que <i>héroes</i> están en favoritos en el apartado <i>My Heroes</i> de la aplicación.

Apartado	Descripción
Título	Puntuar <i>héroe</i>
Propósito	Poner una puntuación a un héroe concreto
Entrada	Puntuación



Proceso	Se añade la puntuación de ese usuario a ese héroe en la base de datos
Salida	Se podrá ver la puntuación que se ha dado a ese <i>héroe</i> desde el detalle del héroe.

Apartado	Descripción
Título	Comentar <i>héroe</i>
Propósito	Añadir un comentario a un héroe concreto
Entrada	Comentario
Proceso	Se añade el comentario de ese usuario a ese héroe en la base de datos
Salida	Se podrá ver el comentario que se ha dado a ese héroe desde el detalle del héroe

Apartado	Descripción
Título	Editar comentario <i>héroe</i>
Propósito	Editar un comentario ya hecho sobre un héroe
Entrada	Nuevo comentario
Proceso	Se añade el nuevo comentario de ese usuario a ese héroe en la base de datos
Salida	Se podrá ver el comentario que se ha dado a ese <i>héroe</i> desde el detalle del héroe.



Apartado	Descripción
Título	Eliminar comentario <i>héroe</i>
Propósito	Elimina un comentario ya hecho sobre un héroe
Entrada	-
Proceso	Elimina el comentario que el usuario ha hecho de un héroe específico de la base de datos
Salida	Se podrá ver cómo el comentario se ha eliminado desde el detalle del héroe.

Apartado	Descripción
Título	Crear un equipo
Propósito	El usuario podrá crear un único equipo para añadir sus héroes favoritos y así enfrentarlo con los equipos de otros usuarios
Entrada	Nombre del equipo
Proceso	Se añadirá un equipo para un usuario concreto en la base de datos
Salida	Si el equipo se ha creado correctamente se informará de ello al usuario.

Apartado	Descripción
Título	Modificar nombre del equipo
Propósito	El usuario podrá modificar el nombre del equipo
Entrada	Nombre del equipo
Proceso	Se actualizará el nombre del equipo en la base de datos



Salida	Si el nombre del equipo se ha modificado correctamente se informará de ello al usuario.
--------	---

Apartado	Descripción
Título	Eliminar un equipo
Propósito	El usuario podrá eliminar un equipo ya creado
Entrada	-
Proceso	Se eliminará el equipo del usuario de la base de datos
Salida	El equipo se eliminará de la página de <i>My Heroes</i> y el usuario podrá crear un equipo nuevo si así lo desea.

Apartado	Descripción
Título	Añadir miembro al equipo
Propósito	El usuario podrá añadir miembros al equipo de los héroes que tiene en favorito
Entrada	-
Proceso	Se añadirá un nuevo miembro al equipo, pudiendo tener como máximo 5 miembros.
Salida	El miembro aparecerá en la información del equipo del usuario.

Apartado	Descripción
Título	Eliminar miembro del equipo
Propósito	El usuario podrá eliminar miembros del equipo



Entrada	-
Proceso	Se eliminará el miembro seleccionado del equipo
Salida	El miembro ya no aparecerá en la lista de miembros del equipo del usuario

Apartado	Descripción
Título	Enfrentar heroes
Propósito	Dentro del apartado <i>Fight</i> de la aplicación el usuario podrá enfrentarse a heroes
Entrada	Seleccionar Hero1 y seleccionar Hero2
Proceso	Se comprobarán las estadísticas de cada héroe y se mostrará como ganador el héroe que suma mayores estadísticas.
Salida	Se mostrará la información del héroe que ha ganado el enfrentamiento.

Apartado	Descripción
Título	Enfrentar equipos
Propósito	Dentro del apartado <i>Fight</i> de la aplicación el usuario podrá buscar equipos, tanto el suyo como el de otros usuarios, para enfrentarlos
Entrada	Seleccionar Team1 y seleccionar Team2
Proceso	Se comprobarán las estadísticas de cada equipo y se mostrará como ganador el equipo que suma mayores estadísticas
Salida	Se mostrará la información del equipo que ha ganado el enfrentamiento.

**Usuario administrador**

Apartado	Descripción
Título	Eliminar usuarios
Propósito	Eliminar usuarios de dentro de la aplicación
Entrada	-
Proceso	Se elige el usuario que se quiere eliminar y se borrarán sus datos de la base de datos
Salida	Se confirmará que el usuario se ha podido eliminar correctamente, y este ya no se mostrará en la aplicación.

Apartado	Descripción
Título	Añadir <i>héroe</i> nuevo
Propósito	Añadir héroes nuevos a la aplicación
Entrada	Información relevante del héroe
Proceso	Se añadirá la información del héroe a la base de datos
Salida	Se confirmará que el héroe se ha añadido correctamente mediante un mensaje, y si ha habido algún problema también se informará sobre ello.

Apartado	Descripción
Título	Editar información de un <i>héroe</i>
Propósito	Modificar información de un héroe
Entrada	Información que se va a cambiar
Proceso	Se añadirá la información del héroe a la base de datos



Salida	Se confirmará que el héroe se ha modificado correctamente mediante un mensaje, y si ha habido algún problema también se informará de ello.
--------	--

3.2 Definición de interfaces de usuario

Se trata de una aplicación web en la que todas las funciones se prestan a través de las diferentes páginas que la componen, a las cuales se podrán acceder a través de un navegador web. Se podrán encontrar diferentes botones para realizar ciertas funciones relevantes en la aplicación. La toma de datos de los usuarios en diferentes apartados se hará mediante formularios. Se hace uso de modales para mostrar diferentes datos destacables y se mostrarán mensajes al realizar algunas de las acciones para confirmar que se han realizado con éxito o, por el contrario, han fracasado.

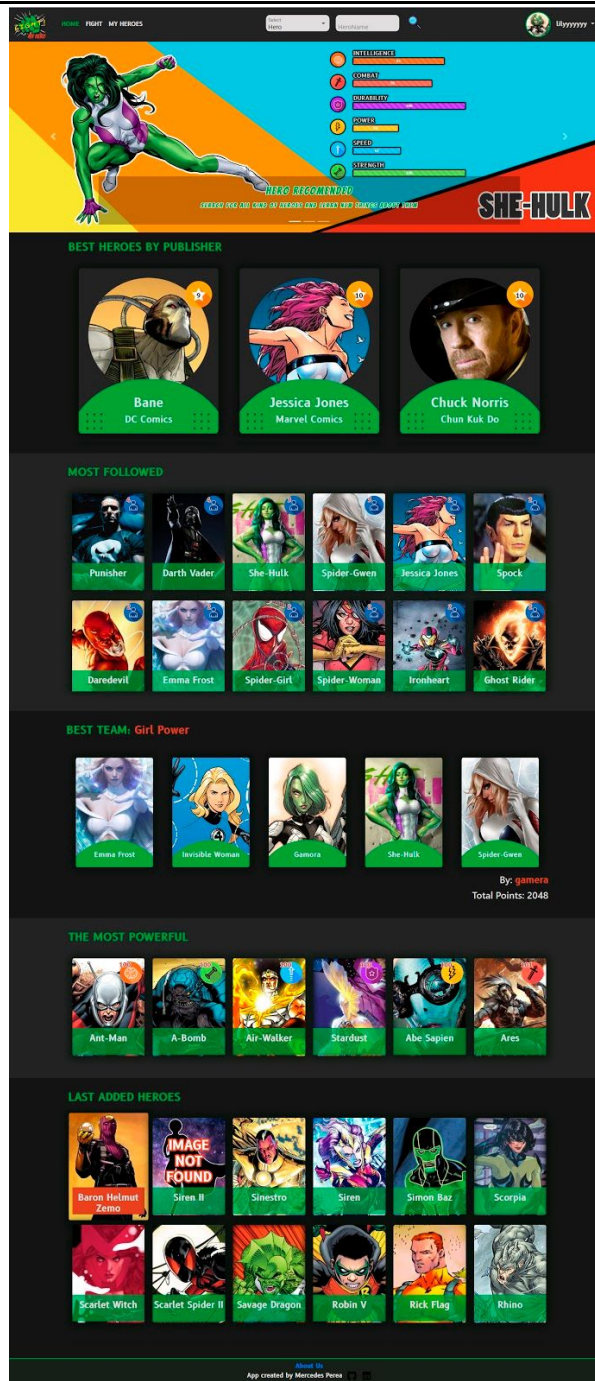
Se ha intentado realizar una interfaz llamativa e intuitiva en la que el usuario pueda tener una buena experiencia navegando en ella. Tiene un diseño *responsive* que se adapta a ordenador, tablet y móvil.

La aplicación se compone de 13 páginas diferentes.

A través de toda la aplicación será accesible una barra de navegación en la que aparecen diferentes elementos que hagan más cómoda la interacción del usuario en la web.

3.2.1 Home

Es la pantalla por defecto de la aplicación; en ella se podrán ver diferentes rankings que irán variando con el uso de la web, como por ejemplo, los héroes mejor valorados por los usuarios, lo más seguidos por los usuarios, o el equipo creado más poderoso.

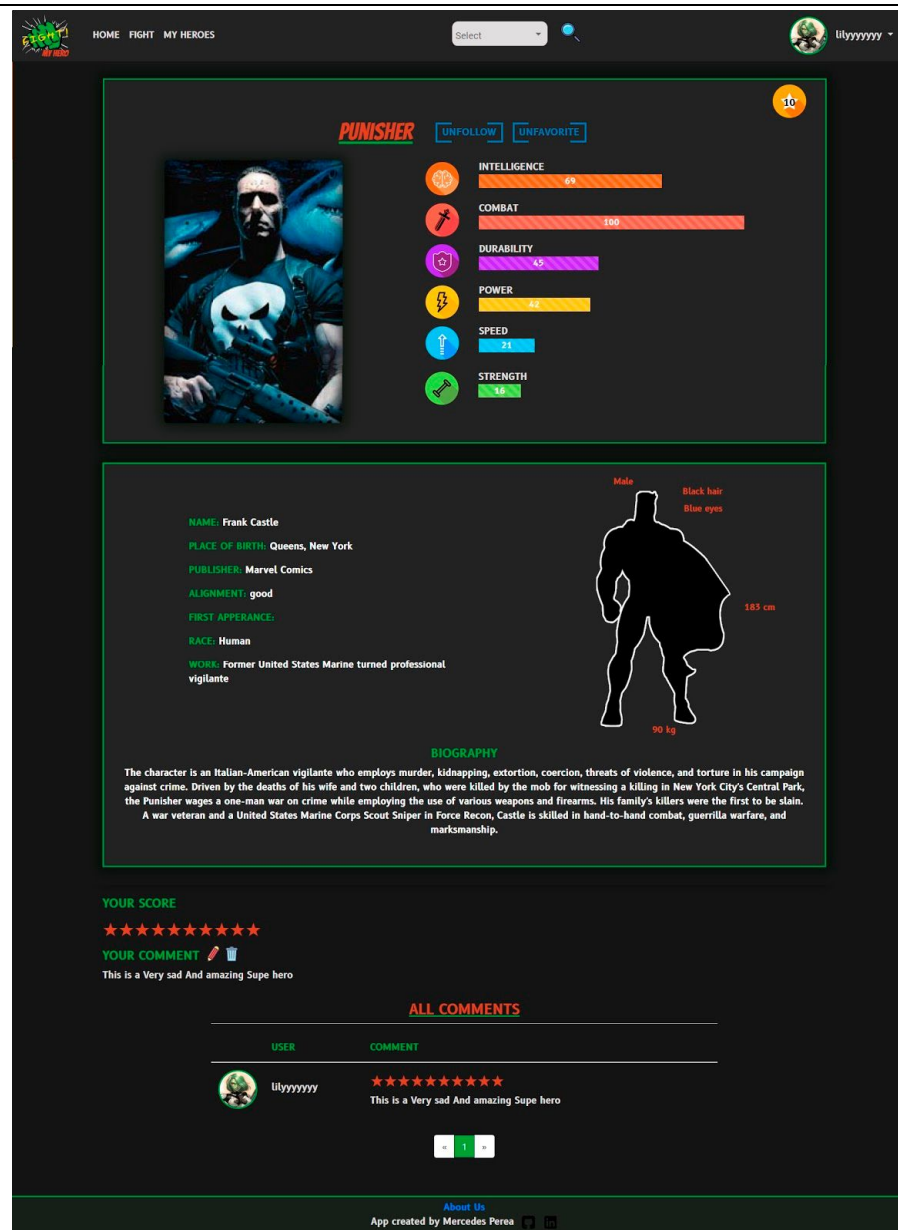


Captura de la aplicación: Home



3.2.2 Detalle Héroe

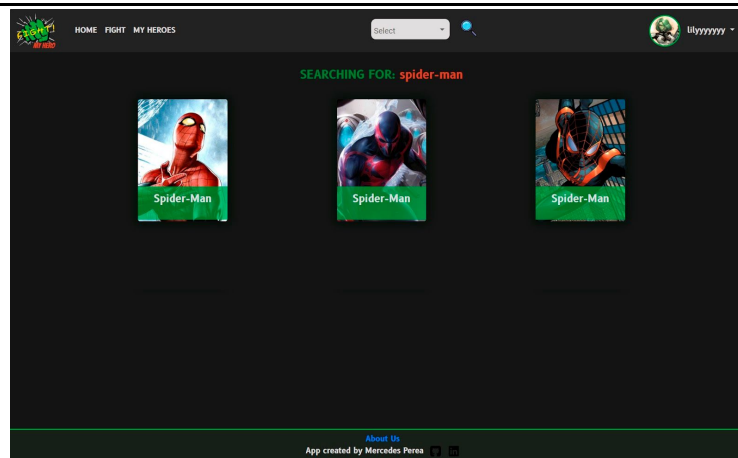
En este apartado se podrá ver información específica del héroe, y los usuarios registrados podrán realizar diferentes acciones referentes a él, como seguir, hacer favorito, comentar y puntuar.



Captura de la aplicación: **Detalle Héroe**

3.2.3 Búsqueda de héroe/usuario

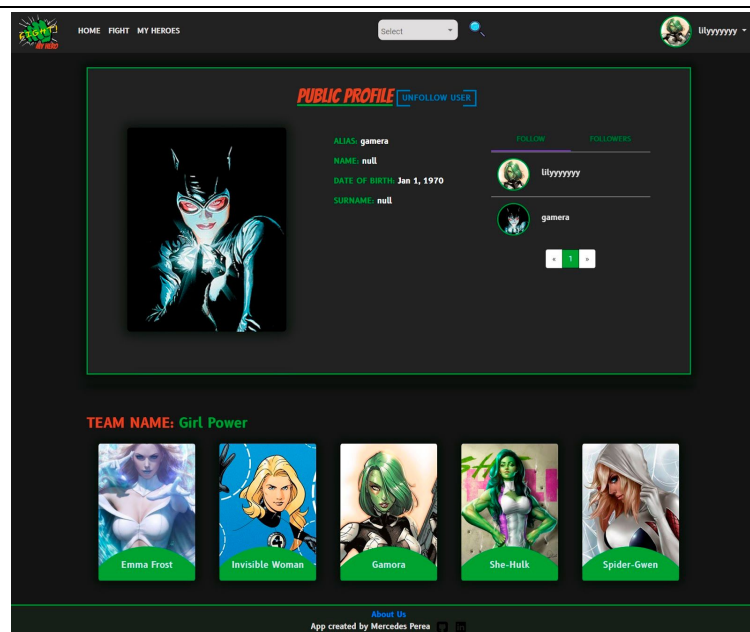
Aquí se mostrarán los resultados a las búsquedas que se realicen, tanto de héroes como de usuarios.



Captura de la aplicación: **Búsqueda de héroe**

3.2.4 Perfil público

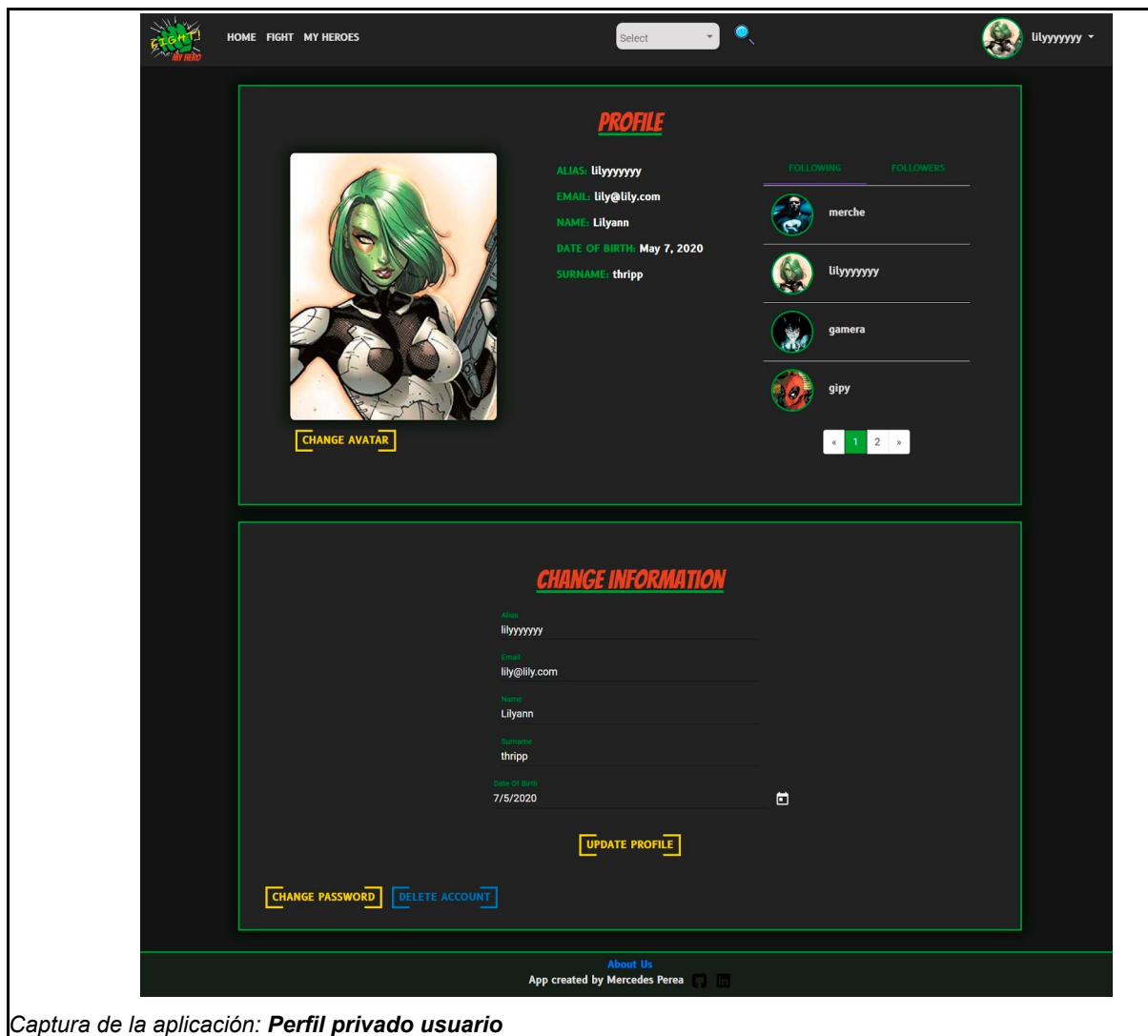
El perfil público del usuario que aparece cuando se realiza una búsqueda de él.



Captura de la aplicación: **Perfil público**

3.2.5 Perfil privado del usuario

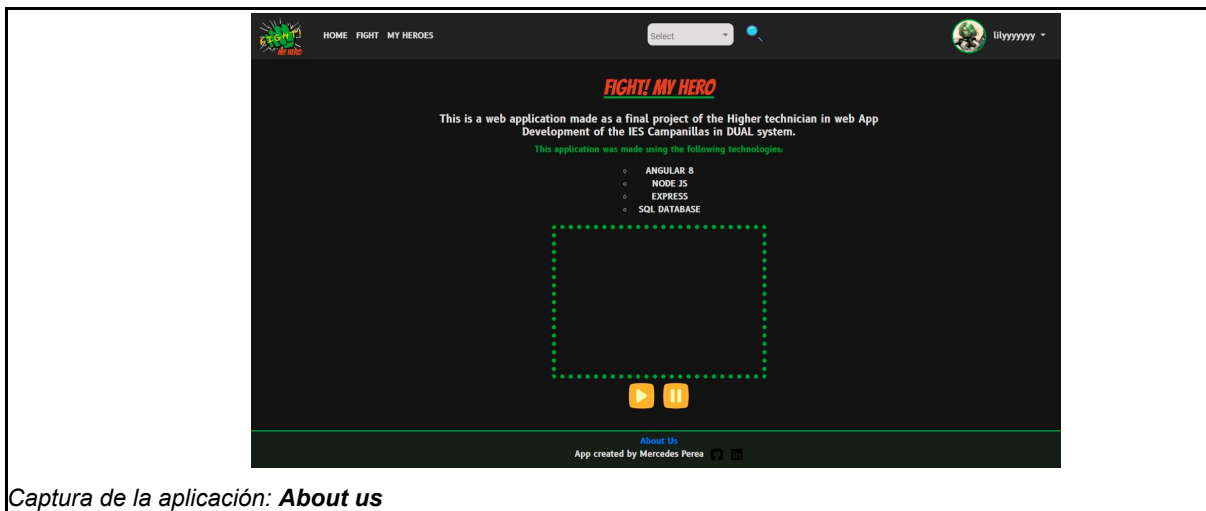
En él se podrá ver información personal del usuario y realizar diferentes acciones, como modificar información, cambiar contraseña o eliminar cuenta.



Captura de la aplicación: **Perfil privado usuario**

3.2.6 About us

Página en la que se muestra información sobre la página y en el que se muestra un vídeo sobre la aplicación.



Captura de la aplicación: **About us**

3.2.7 Fight Hero

En este apartado se podrán realizar enfrentamientos de héroe contra héroe. Para ello será necesario seleccionar los héroes que se quieren enfrentar y pulsar el botón FIGHT, situado en el centro.

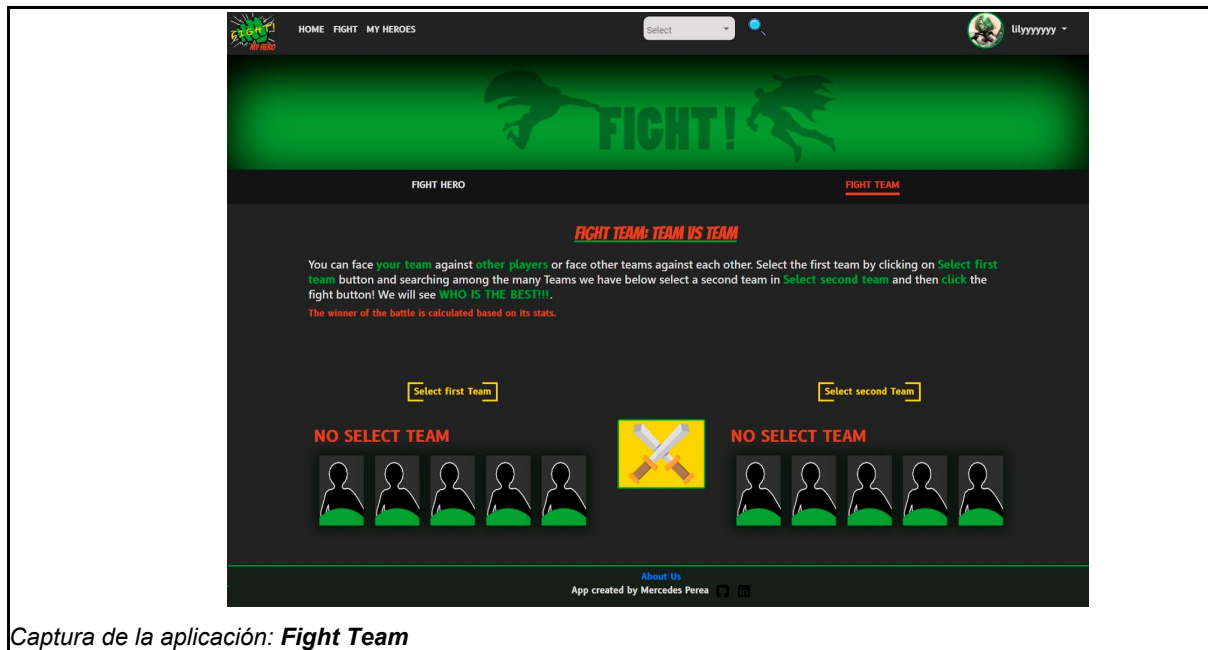


Captura de la aplicación: **Fight Hero**



3.2.8 Fight Team

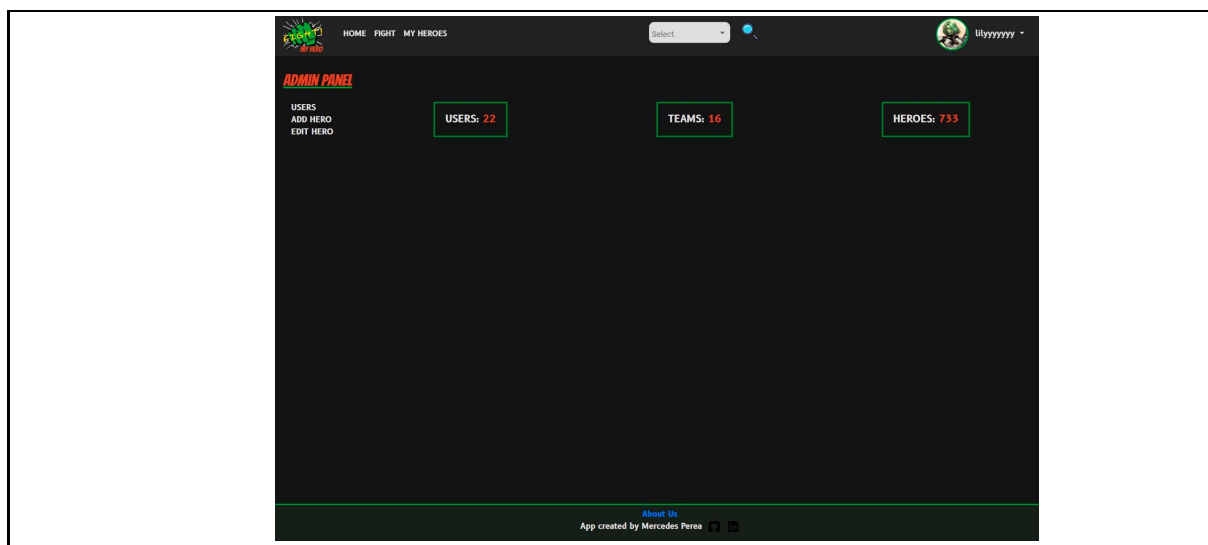
En este apartado se podrán realizar enfrentamientos de Team contra Team. Para ello, será necesario seleccionar los equipos que se quieren enfrentar y pulsar el botón FIGHT.



Captura de la aplicación: **Fight Team**

3.2.9 Panel de administrador

Panel principal de administrador donde se muestran las estadísticas de la página.





Captura de la aplicación: **Admin Panel**

3.2.10 Panel de administrador - Añadir Héroe

Apartado del panel de administrador donde añadir héroes a la base de datos de la aplicación.

The screenshot shows the 'ADMIN PANEL' interface. At the top, there's a navigation bar with 'HOME', 'FIGHT', and 'MY HEROES'. Below this, a sidebar on the left contains 'USERS', 'ADD HERO', and 'EDIT HERO'. The main area displays three statistics: 'USERS: 22', 'TEAMS: 16', and 'HEROES: 733'. The 'Add New Hero' form is the central focus, featuring a grid of input fields for various attributes: heroName, image, fullName, placeOfBirth, publisher, alignment, firstAppearance, gender, race, height, weight, eyeColor, hairColor, work, biography, intelligence, strength, speed, durability, power, and combat. Each field has a corresponding label and a value (e.g., 0). A yellow 'ADD NEW HERO' button is at the bottom left of the form. The footer includes a link to 'About Us' and the text 'App created by Mercedes Perea'.

Captura de la aplicación: **Admin Panel - Add Hero**

3.2.11 Panel de administrador - Editar Héroe

Apartado del panel de administrador donde editar información de un héroe.



Captura de la aplicación: **Admin Panel - Edit Hero**

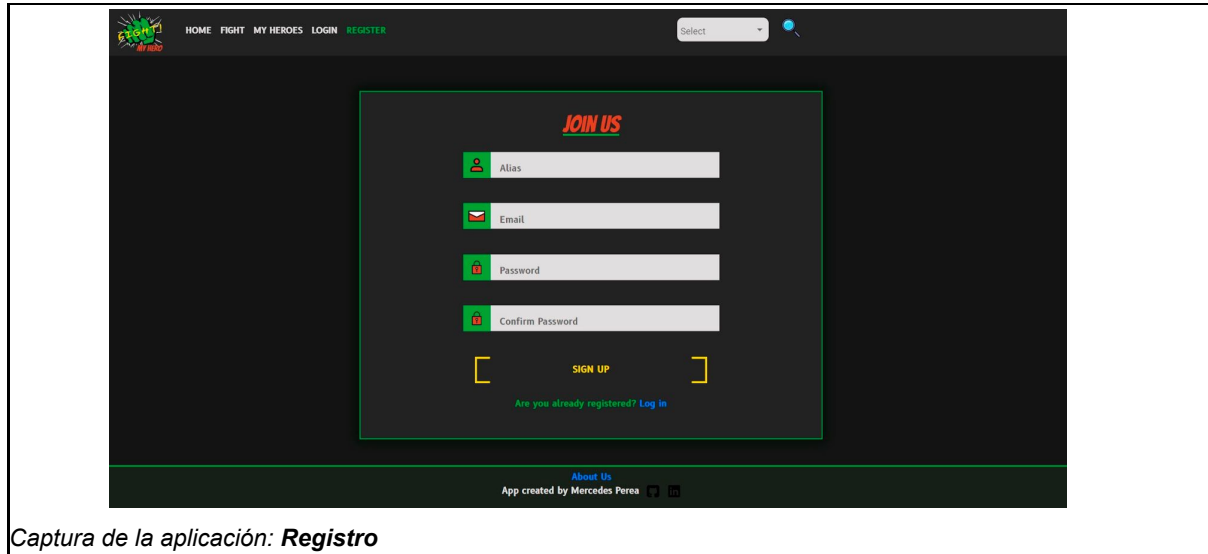
3.2.12 Login

Formulario de login de la aplicación.

Captura de la aplicación: **Login**

3.2.13 Registro

Formulario de registro a la aplicación.



3.3 Interfaces hardware

Puesto que se trata de una aplicación web, su visualización será posible a través de cualquier sistema operativo.

3.4 Interfaces software

La aplicación funciona sobre cualquier móvil, tablet u ordenador que disponga de un navegador web y de una conexión a internet.

3.5 Interfaces de comunicaciones

Las comunicaciones dentro de la aplicación se realizarán mediante el uso del protocolo HTTP mediante la conexión TCP/IP.

3.6 Estándares cumplidos

A la hora de realizar la aplicación web se han seguido los estándares generales de todo sitio web con un acceso seguro, creando diferentes perfiles que se controlan mediante el sistema de autenticación, y que mediante la obtención de un *token* se controla que no se puedan acceder a las zonas de la aplicación que no sean adecuadas para su nivel dentro de ella.



La aplicación está realizada íntegramente en inglés por ser el idioma “universal”, y para así poder hacer accesible su uso al mayor rango posible de usuarios.

3.7 Seguridad

El tema de la seguridad se ha intentado hacer de la forma más óptima posible, controlando los permisos que poseen los usuarios dentro de la aplicación tanto en la parte del *front* como en el *back*: para poder interactuar con la aplicación será necesario el registro y login del usuario vía email y contraseña. Se hace control de sesiones, generación de *tokens* y nunca se pasa a la parte frontal información sensible del usuario, como puede ser su contraseña.

3.8 Facilidades de mantenimiento

Esta aplicación tiene un mantenimiento bastante sencillo; el mantenimiento básico podrá ser llevado por el administrador de la aplicación.

Para realizar modificaciones más complejas en la base de datos o añadir nuevas funcionalidades en la aplicación será necesario que sea un programador web el que las realice.

3.9 Planificación temporal

El tiempo estimado para la realización de esta tarea inicialmente está contemplado que para este módulo de proyecto sean unas 40 horas, pero debido a la envergadura del proyecto y a las actuales circunstancias en las que nos encontramos, he decidido crear un proyecto de más duración en el que poder aplicar todos los conocimientos que he ido adquiriendo durante la realización del curso, e ir ampliando e implementando conocimientos nuevos.

Así, el total de horas de este proyecto se han calculado en unas 100 horas, contando una media de 3 horas de dedicación diarias durante un total de 33 días.

4. Diseño de la base de datos

Para la creación de la base de datos se ha optado por un modelo relacional; el diagrama entidad-relación de este proyecto sería el siguiente:

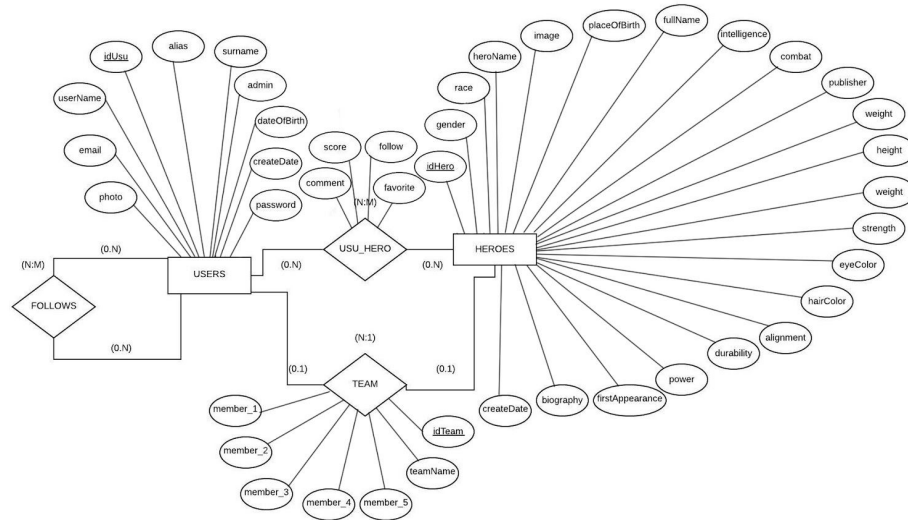


Diagrama Entidad-Relación del proyecto

En este diagrama podemos apreciar que consta de dos entidades principales: *USERS*, que contiene la información de los usuarios, y *HEROES*, la cual contiene la información referente a los héroes. Las relaciones que existen son *Usu_Hero*, la cual contendrá información referente a la relación de los usuarios con los respectivos heroes; *TEAM*, que nace de la unión de un usuario y héroes, ya que en él un usuario podrá almacenar muchos héroes diferentes, y un usuario solo podrá tener un *TEAM*; y la relación *FOLLOWS*, que nace de la idea de que un usuario puede seguir a ninguno o a muchos usuarios y a un usuario lo pueden seguir ninguno o muchos usuarios.

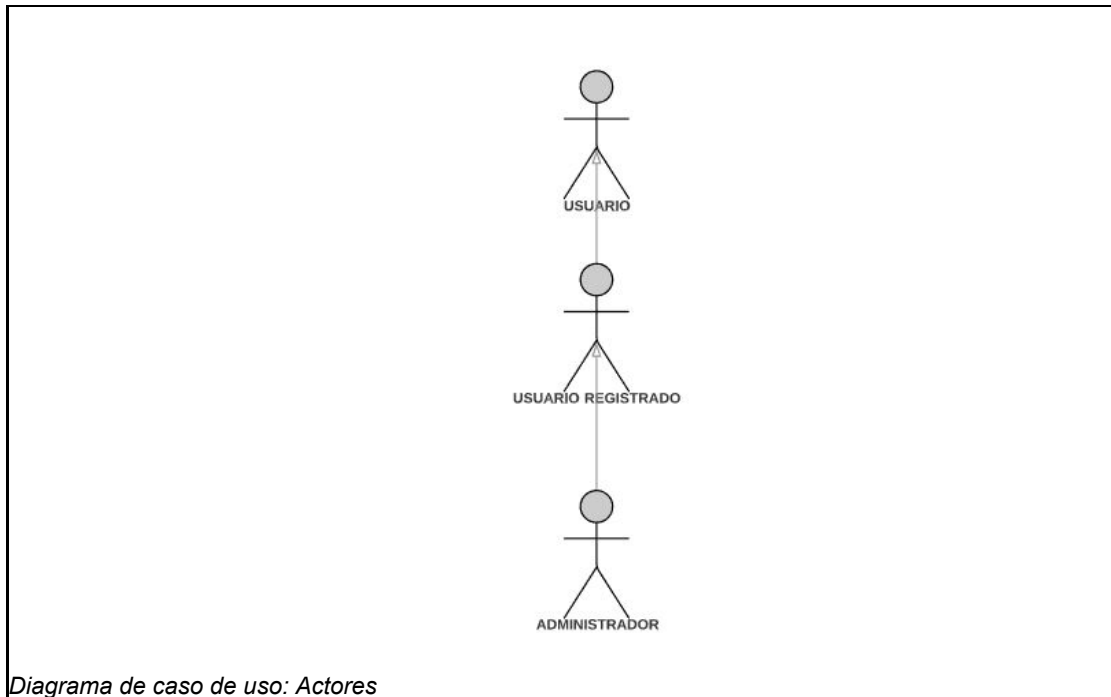
5. Diagrama de casos de uso

A continuación se muestran unos diagramas de casos de uso que nos permiten diferenciar los diferentes actores que interactúan en nuestra aplicación, cómo se relacionan y las acciones que les son disponibles según sus privilegios.

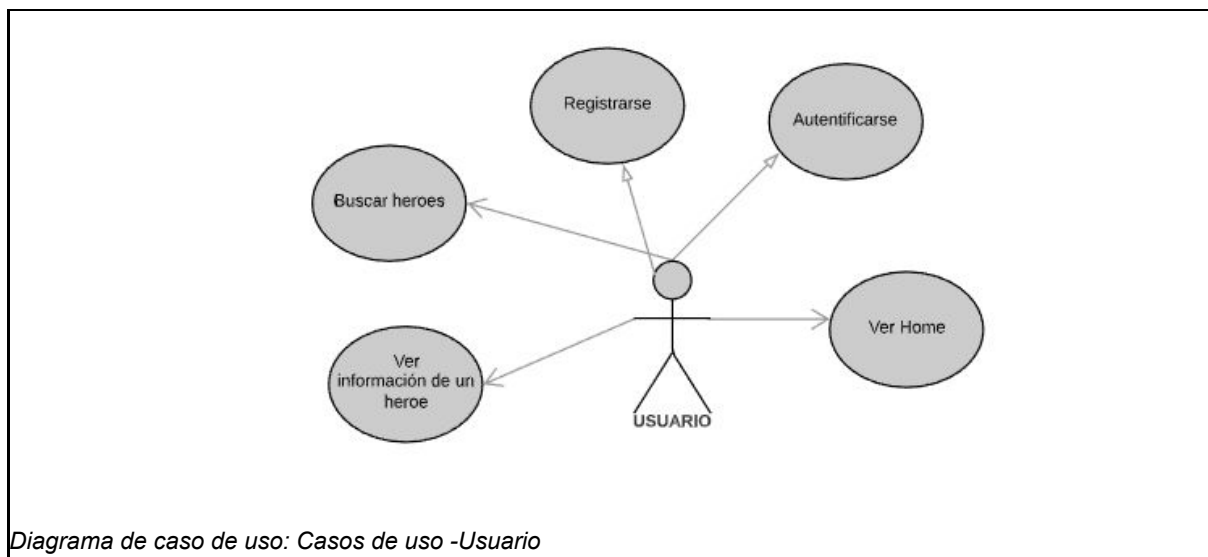
Este tipo de diagrama funciona como una forma simplificada de explicar las funciones y requisitos básicos de la aplicación.

Dentro de nuestro diagrama diferenciamos a:

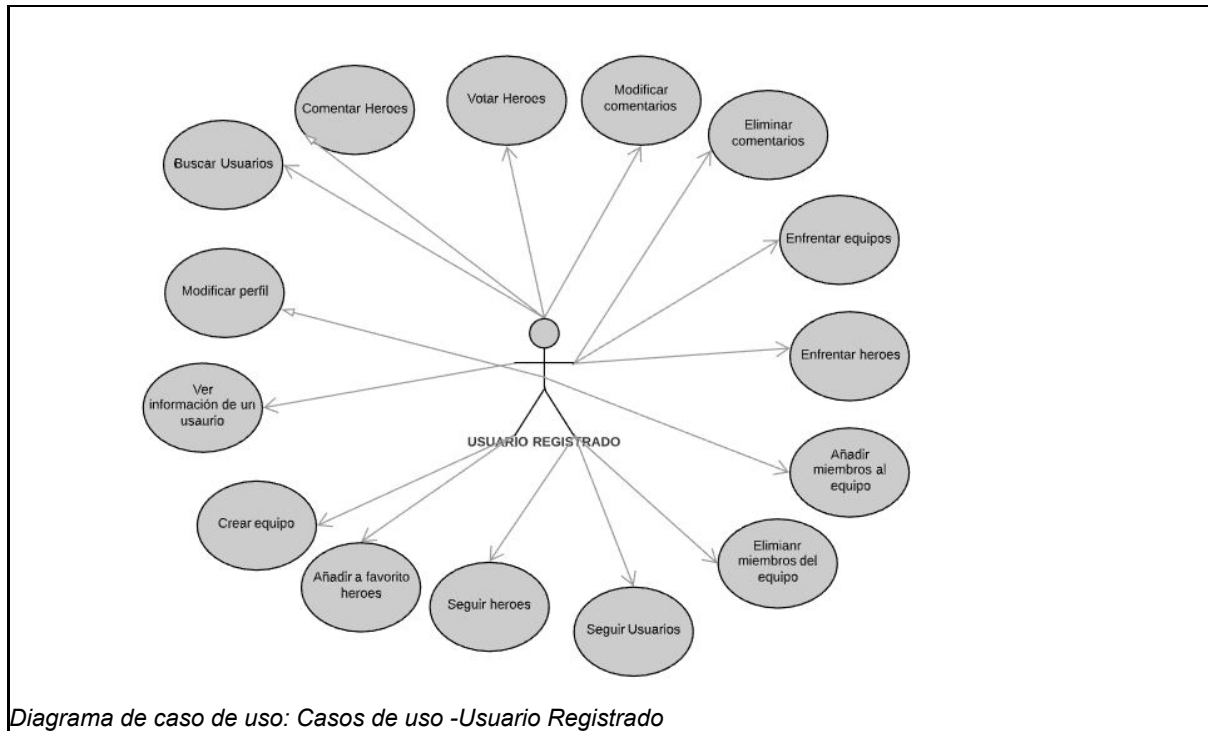
- Actores (usuarios del sistema): Usuarios generales, usuarios registrados y administradores.



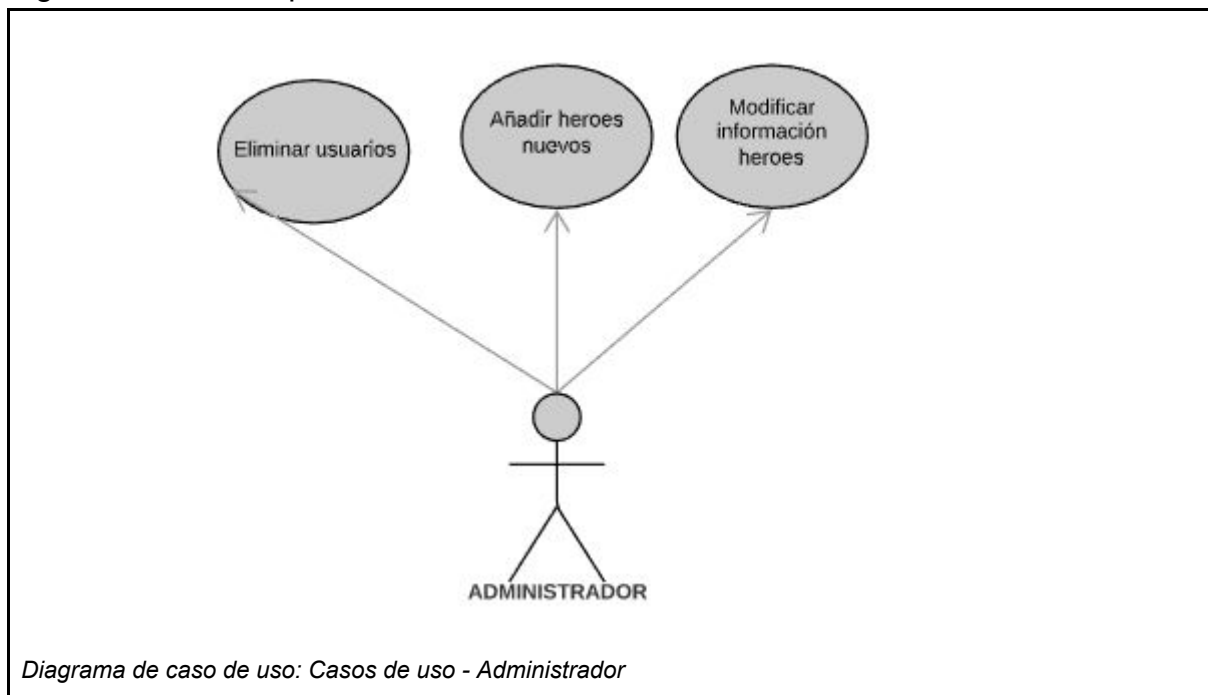
- Casos de uso: donde se describen la forma de cómo utiliza cada Actor las distintas funcionalidades de la aplicación.



Este Usuario no se ha registrado en la aplicación por lo tanto sus acciones serán muy limitadas.



El usuario registrado podrá optar por múltiples funcionalidades que hacen que el registro merezca la pena.

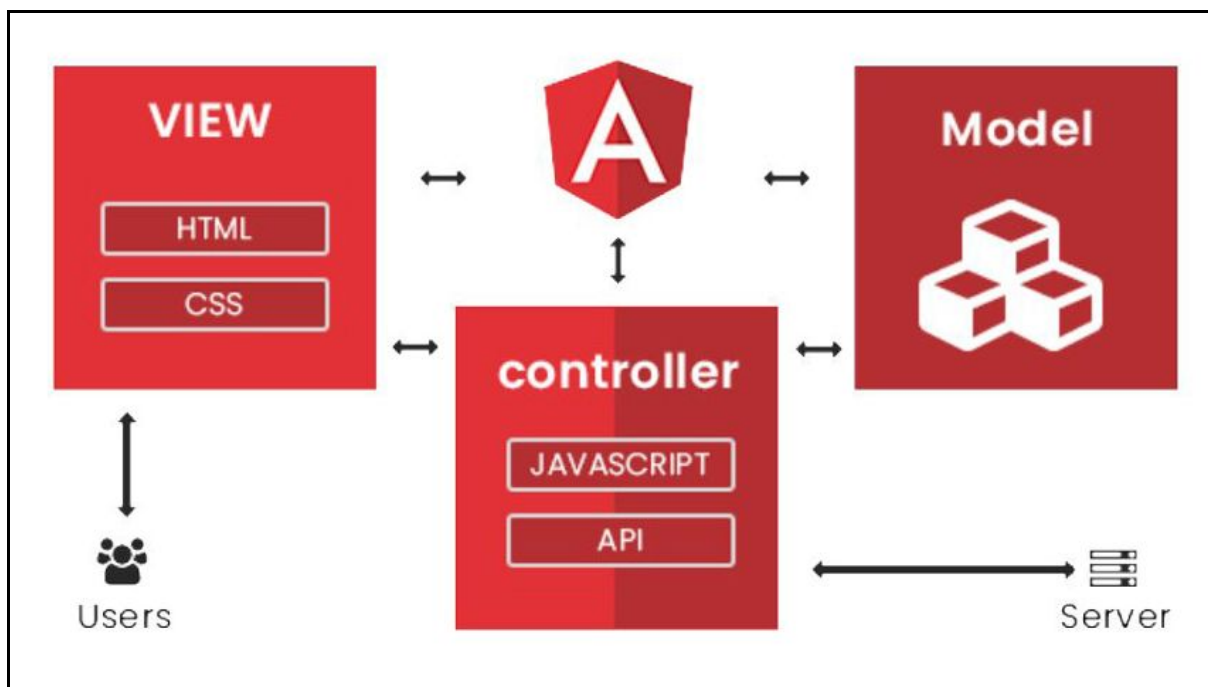


El Administrador podrá realizar funcionalidades de carácter sensible dentro de la aplicación.



6. Diseño

Para el diseño de esta aplicación se ha usado un patrón de diseño tipo modelo vista-controlador adaptado a las necesidades de los diferentes lenguajes y frameworks que en ella se utilizan, y teniendo en cuenta las características propias de cada uno de los frameworks.





Esquema MVC en Angular

Aquí se puede ver un ejemplo de cómo se puede entender el patrón MVC desde el framework de Angular.

La vista representa la parte visual de la aplicación: es la parte donde el usuario va a interactuar, también conocida como la interfaz de usuario, y se compone por la información que se envía al cliente y los mecanismos de interacción con él.

El modelo contiene una representación de los datos que maneja el sistema, su lógica y mecanismos.

El controlador actúa como intermediario entre el Modelo y la vista, gestiona el flujo de la información entre ellos y hará las transformaciones necesarias para que se adapten a las necesidades de cada uno.

7. Documentación del código de la aplicación

Para la documentación de las diferentes partes de la aplicación se han utilizado, respectivamente, JSDoc para documentar el código de Node.js, y Compudoc para documentar Angular; ambas son herramientas que permiten generar documentación automática de manera ágil, dando buenos resultados; es especialmente destacable Compudoc para Angular, ya que es capaz de extraer muchísima información referente a la aplicación puesto que está especialmente diseñado para este framework, y en ella se dan detalles que resultan muy útiles para el entendimiento del código generado.

8. SCV

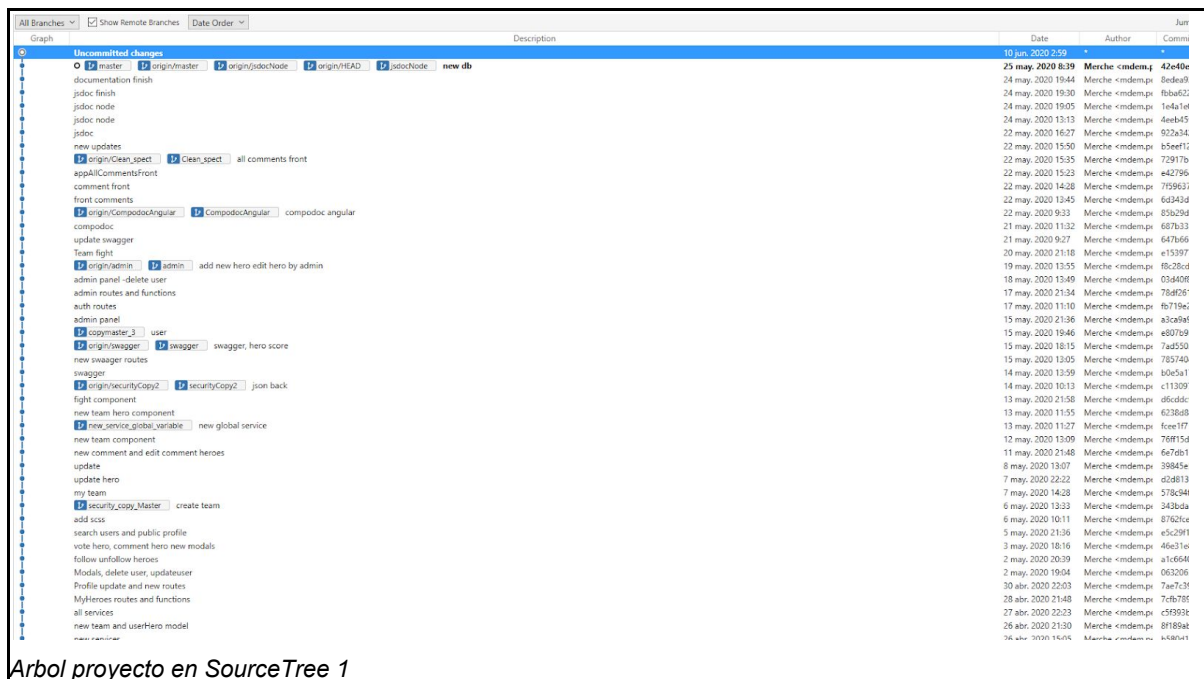
Como se ha comentado con anterioridad en el inicio de este documento, se ha utilizado Git para el control de versiones, utilizando como cliente GUI el software Sourcetree.

Durante el proceso de desarrollo de esta aplicación se han utilizado además dos repositorios: uno para trabajar antes de llevar a producción en Github, y otro hospedado en IBM Cloud para trabajar con el proyecto ya desplegado.



Este cambio de repositorios se debe a que el que se encuentra en IBM Cloud permite conseguir un despliegue con Integración Continua (CI) de nuestra aplicación con solo hacer *push* a la rama master del proyecto, pudiendo tener así siempre de forma rápida y eficaz el proyecto desplegado lo más actualizado posible.

En ambos repositorios se han ido generando diferentes ramas según han ido surgiendo necesidades diferentes durante la creación del proyecto, haciendo subidas prácticamente diariamente.



Arbol proyecto en SourceTree 1

En esta imagen se puede ver el trabajo realizado en Github, el primer repositorio en el que se estuvo trabajando y donde se hacían subidas de *back* y *front* juntas.



Memoria Proyecto DAW

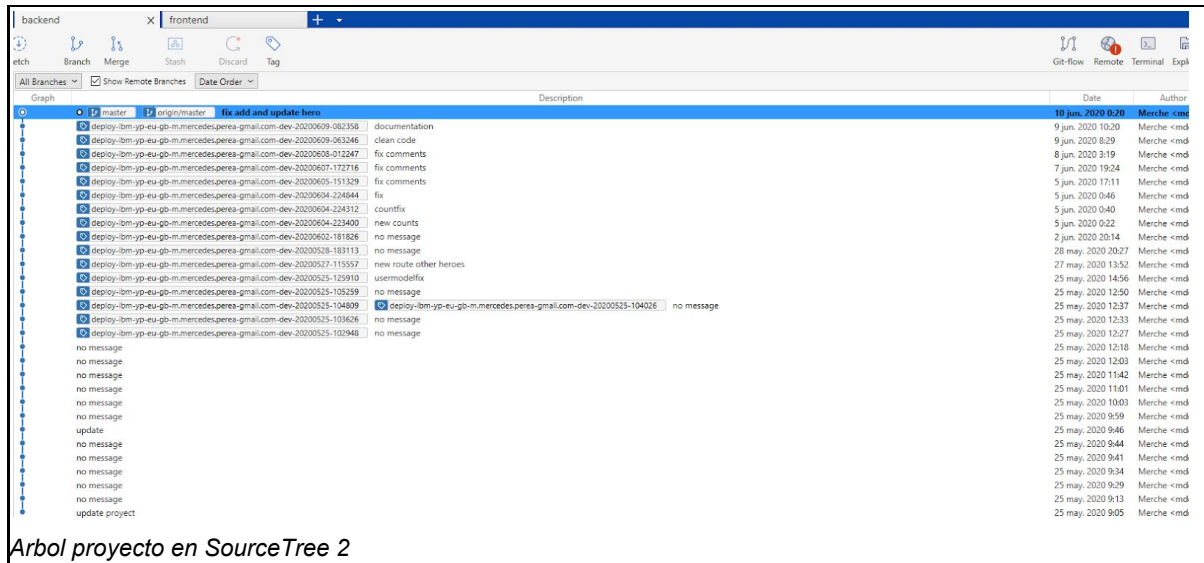


Imagen donde se ve el seguimiento y uso de ramas en la parte del *back* una vez desplegado en IBM Cloud.

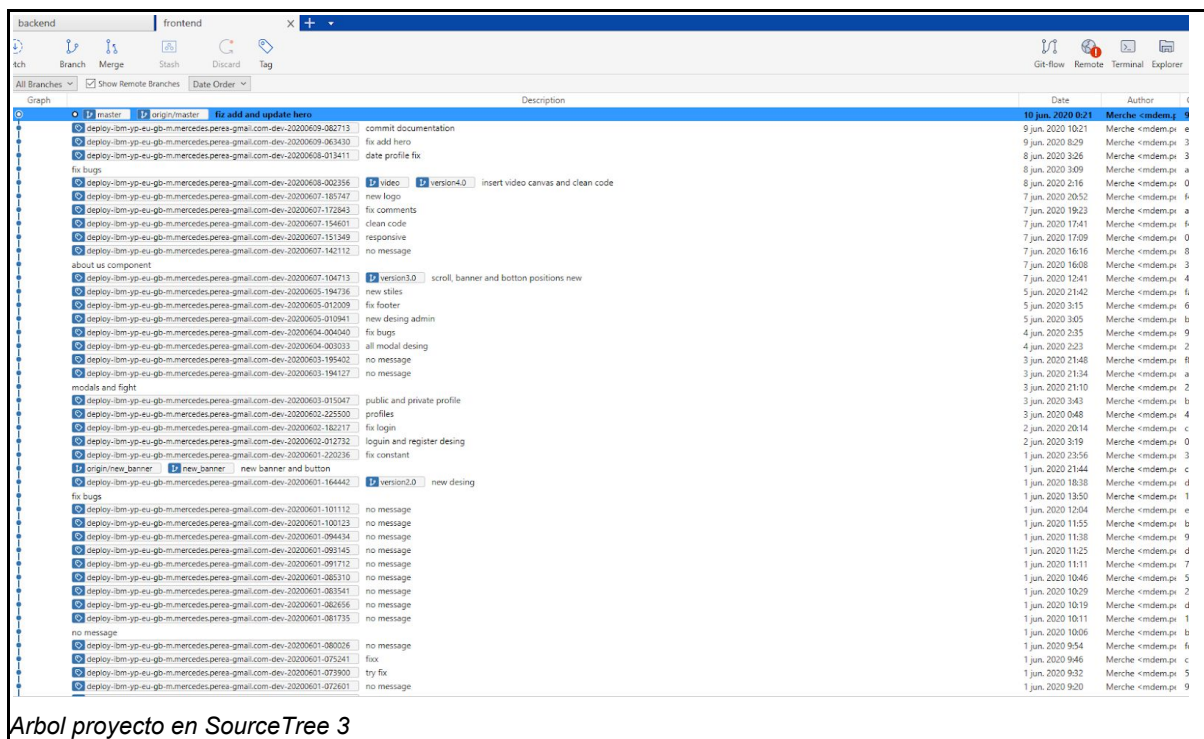


Imagen donde se ve el seguimiento y uso de ramas en la parte del *frontal* una vez desplegado en IBM Cloud.



9. DESPLIEGUE

9.1 DESPLIEGUE LOCAL

Para el despliegue local de nuestra aplicación solo es necesario tener instalado Node.js en nuestro equipo: Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

Para poder trabajar localmente con nuestro proyecto Angular basta con ir a la terminal y utilizar el comando “*npm i*” para descargarse todos los *node_modules* (*npm folders*) necesarios para el funcionamiento de la aplicación, y seguidamente usar el comando “*ng serve*” para compilar la aplicación: esto logrará que de forma automática, y según hemos configurado, se despliegue la aplicación localmente. Una vez compilado, basta con ir a nuestro a la dirección *http://localhost:4000* y podremos ver nuestra aplicación Angular desplegada localmente.

Para desplegar la parte de Node.js es necesario también ejecutar el comando “*npm i*”, para así instalar todas las dependencias de *node_modules*; hemos configurado su puerto de salida para que aparezca en la dirección *http://localhost:3000*. Al utilizar el comando “*node nombreDeLaApp.js*”, se compilará de forma automática y se mostrará en el puerto configurado; en este caso, se ha configurado el archivo *package.json* para que con solo usar el comando “*npm run start*” o “*npm run dev*” se realice la compilación.

Para el despliegue local de la base de datos se ha utilizado phpMyAdmin con XAMPP.

9.2 DESPLIEGUE LOCAL EN APACHE

Para el despliegue local en Apache, requerimiento de la asignatura Despliegue de Aplicaciones Web, hemos utilizado el paquete de software libre XAMPP en una máquina Windows: XAMPP consiste principalmente en un sistema de gestión de bases de datos MySQL, un servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. Utilizaremos sus propiedades como gestor de base de datos y servidor Apache.



Comenzaré explicando las configuraciones necesarias que hacen falta para desplegar la parte Angular en el Apache de XAMPP; basta con seguir unos pocos pasos:

- Modificar el archivo *index.html*: Se deberá de cambiar el atributo `<base href="/">` por `<base href="."/>` para que así coja bien el sistema de rutas de la aplicación.

Captura de cambio producido en el archivo *index.html* del proyecto Angular

- Usar el comando `"ng build --prod"` en nuestra terminal donde se encuentre nuestro proyecto Angular; este comando genera una versión para un ambiente de producción.

Este comando nos generará una carpeta denominada *dist* donde se encontrará nuestro proyecto generado para producción.

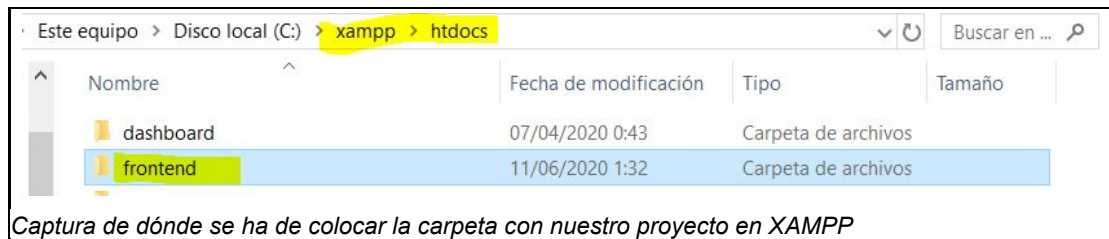
```
PS C:\Users\Merche\Documents\Merchicosas\projects\FightMyHeroIBMCLOUD\frontend> ng build --prod
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {2} polyfills-es2015.5b10b8fd823b6392f1fd.js (polyfills) 36.2 kB [initial] [rendered]
chunk {3} polyfills-es5.8e50a9832860f7cf804a.js (polyfills-es5) 127 kB [initial] [rendered]
chunk {1} main-es2015.5fed80abb976940beb36.js (main) 1.45 MB [initial] [rendered]
chunk {1} main-es5.5fed80abb976940beb36.js (main) 1.58 MB [initial] [rendered]
chunk {0} runtime-es2015.c5fa8325f89fc516600b.js (runtime) 1.45 kB [entry] [rendered]
chunk {0} runtime-es5.c5fa8325f89fc516600b.js (runtime) 1.45 kB [entry] [rendered]
chunk {4} styles.49d1bcacf9a03e37296a.css (styles) 210 kB [initial] [rendered]
chunk {scripts} scripts.9e7d3e9eed653021f88d.js (scripts) 145 kB [entry] [rendered]
Date: 2020-06-10T23:44:58.543Z - Hash: 8ec6d52eeeab3d80ced3 - Time: 76711ms
```

Captura del proceso sucedido tras utilizar el comando `ng build --prod` en nuestro proyecto Angular

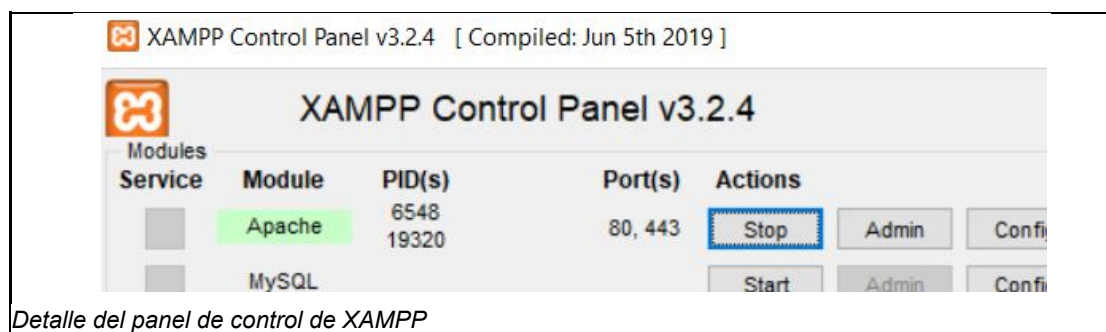
- Copiamos la carpeta que se encuentra dentro del directorio *dist* que acabamos de generar, la cual tendrá el nombre que le hayamos dado a nuestra aplicación (en este caso *frontend*) y la llevamos a XAMPP: para ello,

copiamos esta carpeta en el directorio *htdocs* de nuestra instalación de XAMPP.

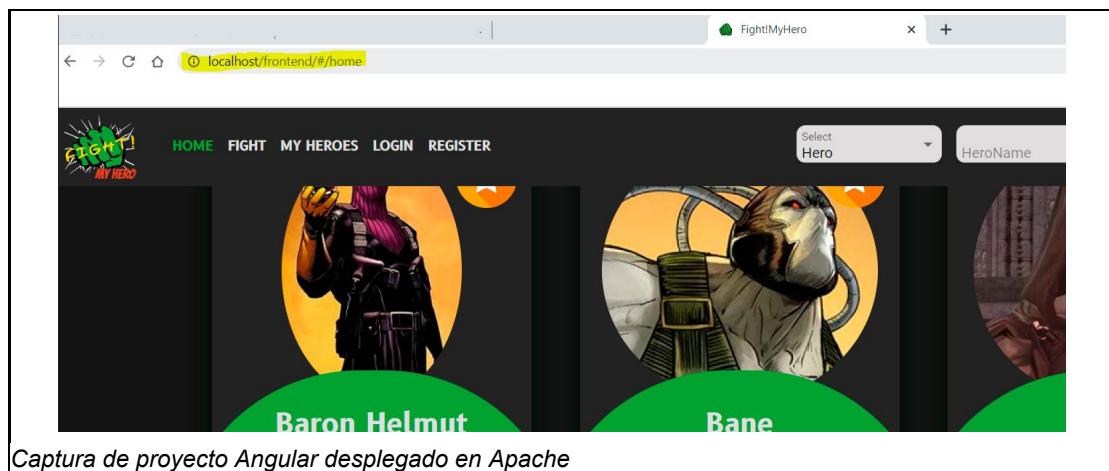


Captura de dónde se ha de colocar la carpeta con nuestro proyecto en XAMPP

- Ahora solo basta con arrancar nuestro servidor apache desde XAMPP e ir en nuestro navegador a `localhost/nombreDeNuestraAplicación` para poder ver nuestra aplicación Angular desplegada en Apache correctamente.



Detalle del panel de control de XAMPP

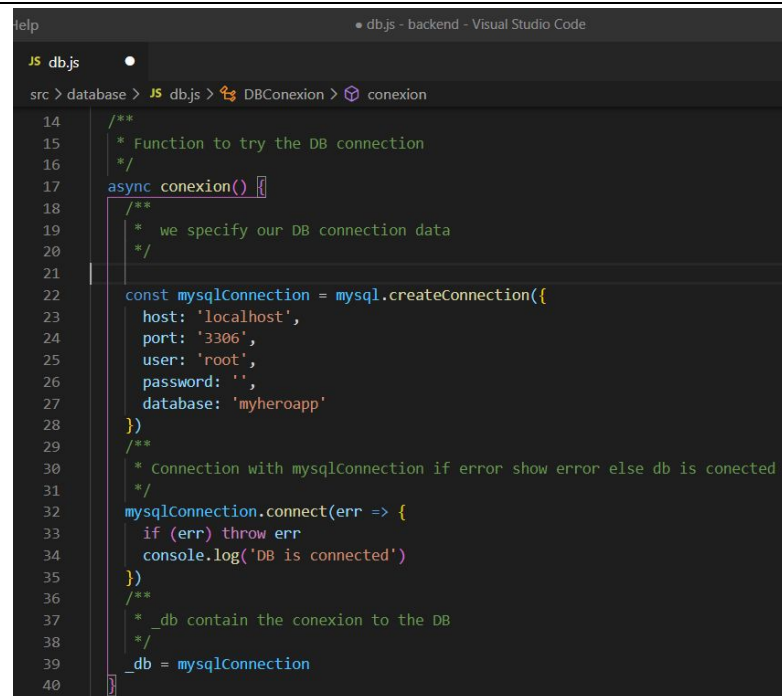


Captura de proyecto Angular desplegado en Apache

Para el despliegue de la Base de datos hemos utilizado el gestor de base de datos que nos trae XAMPP: simplemente con arrancarlo desde el panel de XAMPP e ir a la sección admin se nos abrirá en nuestro navegador un panel de phpMyAdmin en el que podremos insertar nuestro script de la base de datos y conectarnos a ella.



Nos conectaremos a nuestra base de datos servida por XAMPP desde Node.js mediante la función que se muestra en la captura inferior, que hemos creado en el archivo *db.js* de nuestra aplicación Node.js, y la cual será llamada al arrancar nuestra aplicación.

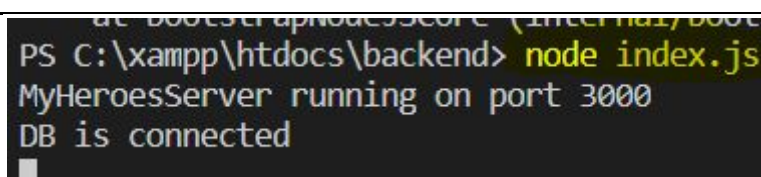


```
14  /**
15   * Function to try the DB connection
16   */
17  async conexion() {
18    /**
19     * we specify our DB connection data
20     */
21
22    const mysqlConnection = mysql.createConnection({
23      host: 'localhost',
24      port: '3306',
25      user: 'root',
26      password: '',
27      database: 'myheroapp'
28    })
29    /**
30     * Connection with mysqlConnection if error show error else db is connected
31     */
32    mysqlConnection.connect(err => {
33      if (err) throw err
34      console.log('DB is connected')
35    })
36    /**
37     * _db contain the conexion to the DB
38     */
39    _db = mysqlConnection
40  }
```

Captura que detalla la función de la conexión a la base de datos desde nuestra aplicación de Node.js

Ahora bien, para que Apache pueda servir nuestra aplicación de Node.js y que todo se pueda acceder desde nuestro servidor Apache son necesarios los siguientes pasos:

- Primero, ejecutamos localmente nuestra aplicación de Node.js mediante el comando “*node index.js*”; de este modo, nuestro servidor Node estará corriendo en <http://localhost:3000/>.



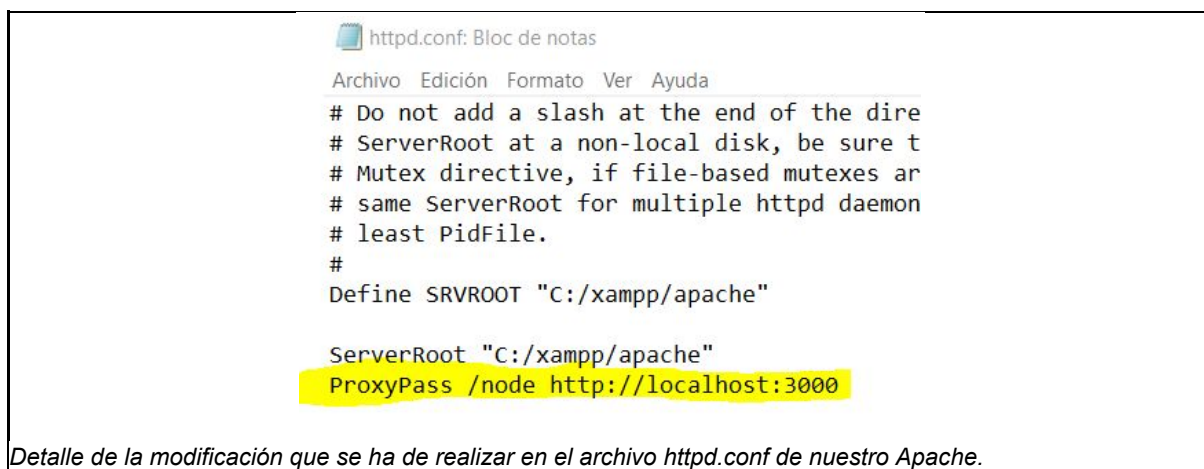
```
PS C:\xampp\htdocs\backend> node index.js
MyHeroesServer running on port 3000
DB is connected
```

Detalle del resultado de arrancar nuestra aplicación Node.js

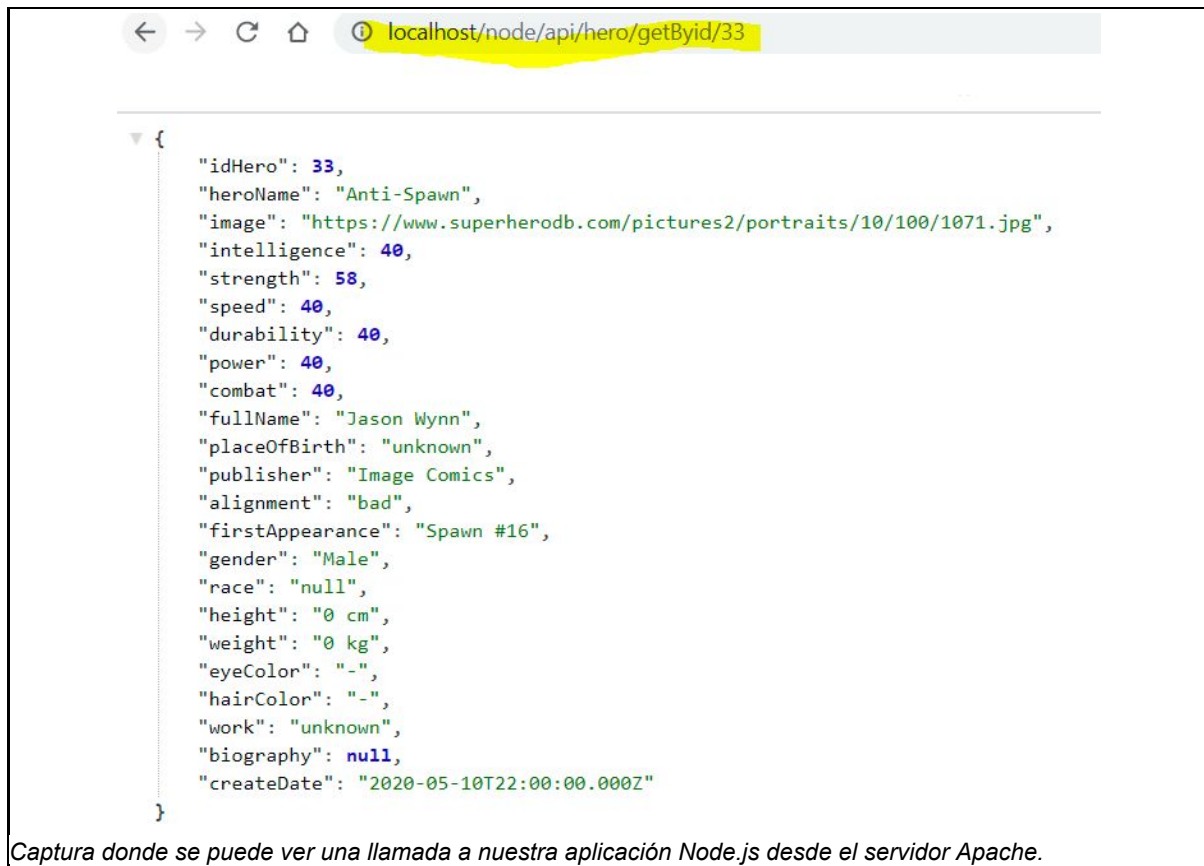
- A continuación, vamos a nuestro panel de XAMPP y le damos a configurar nuestro Apache.



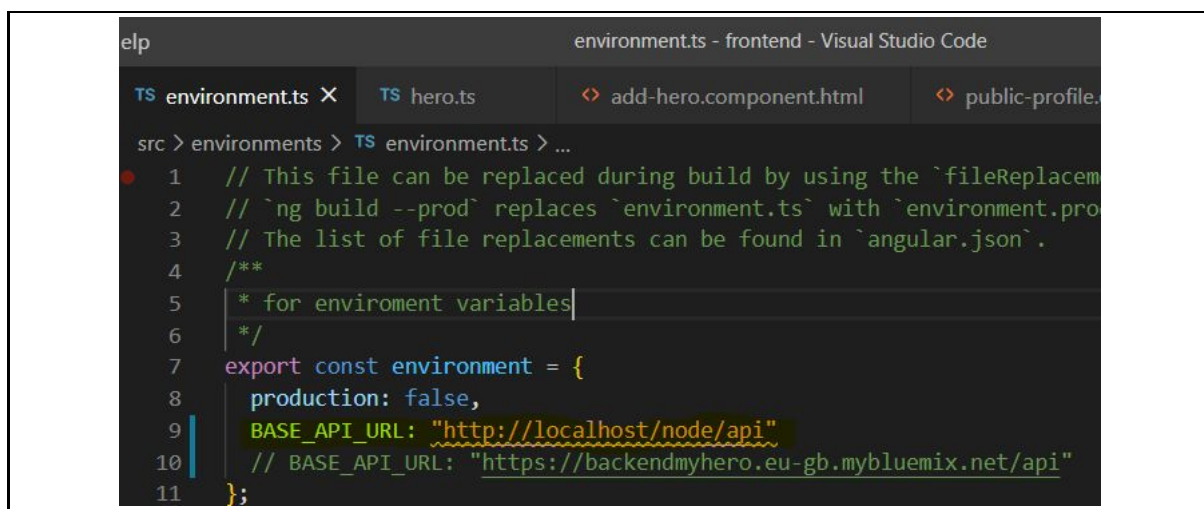
- Una vez dentro de este archivo, tenemos que agregar al documento la línea `"ProxyPass /node http://localhost:3000"`; esto nos permitirá canalizar las solicitudes URL que queramos que nos lleven a nuestra aplicación de Node.js.



- También debemos asegurarnos de que las siguientes líneas de este archivo no están comentadas; si lo están, debemos descomentarlas para permitir al proxy y a los submódulos volver a enrutar peticiones http:
 - `LoadModule proxy_module modules/mod_proxy.so`
 - `LoadModule proxy_http_module modules/mod_proxy_http.so`
- Una vez realizados estos cambios y guardados, volvemos a arrancar nuestro servidor Apache desde el panel de XAMPP y ya podemos acceder a nuestra aplicación node desde Apache.



- Por último, para conectarnos a nuestro Node.js desplegado en Apache, será necesario apuntar a su dirección desde nuestro proyecto Angular: para ello, en nuestro archivo de variables de entorno hemos añadido la dirección de nuestro Node.js de Apache, para que nuestro proyecto Angular sepa a dónde realizar sus peticiones.





Detalle que muestra el archivo `environment.ts` donde se encuentra la conexión a nuestro `Node.js` desde `Angular`

9.3 DESPLIEGUE REMOTO

9.3.1 Plataforma

La plataforma elegida para el despliegue de la aplicación, tanto de la parte realizada en `Node.js` como de la parte `Angular`, ha sido IBM Cloud, que como ya se ha mencionado anteriormente en este documento se ha elegido por ser la plataforma con opciones gratuitas más estable, cómoda de usar y que permite además el uso de diferentes herramientas que hacen que trabajar con el despliegue resulte muy dinámico.

9.3.2 Configuración

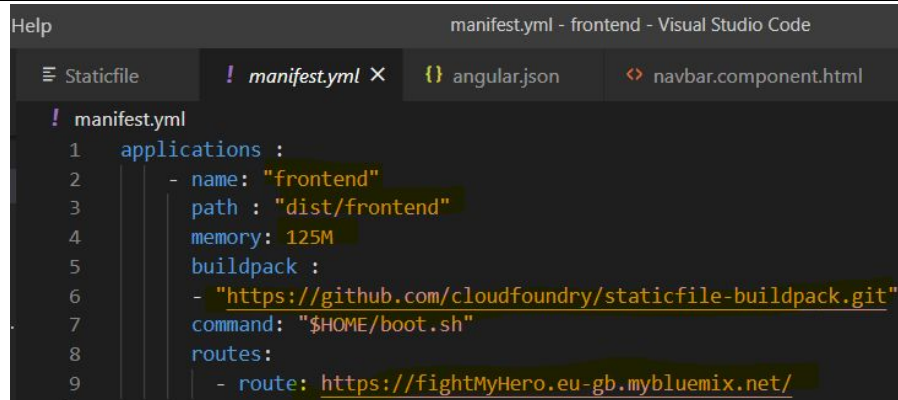
Tras la configuración necesaria que se ha de hacer para los despliegues de `Node.js` y de `Angular`, en la propia plataforma se tuvieron que crear archivos de configuración en cada parte del proyecto para su correcto despliegue en esta plataforma.

En `Angular` fue necesario añadir en la rama principal de nuestro proyecto los siguientes documentos:

- **manifest.yml:** Cuando Cloud Foundry despliega la aplicación buscará el archivo *manifest.yml* en la raíz del directorio, en el cual se especifican los parámetros a seguir durante el despliegue de la misma.

Estos parámetros serán los siguientes:

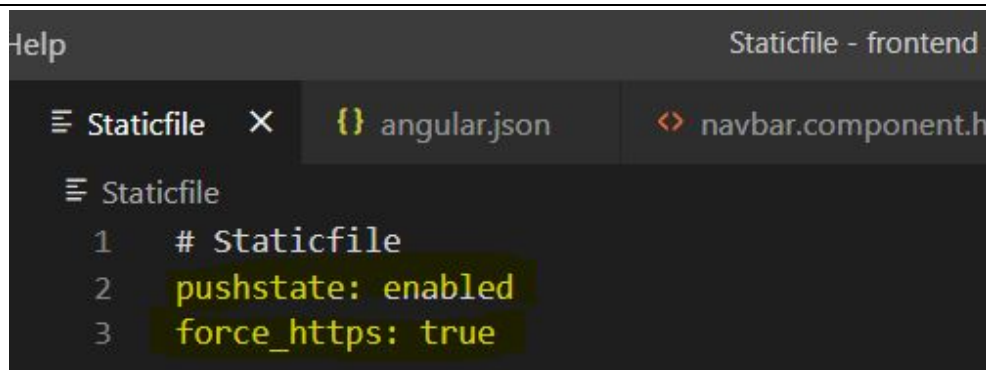
- *name*: nombre de la aplicación.
- *path*: indica dónde se encuentra el proyecto una vez se ha compilado para el despliegue.
- *memory*: uso de la memoria que va a tener la app.
- *buildpacks*: pack de compilación que va a usar para el despliegue.
- *command*: comando para lanzar la app.
- *route*: ruta web donde se encuentra la aplicación desplegada.



```
! manifest.yml
1  applications :
2    - name: "frontend"
3      path : "dist/frontend"
4      memory: 125M
5      buildpack :
6        - "https://github.com/cloudfoundry/staticfile-buildpack.git"
7      command: "$HOME/boot.sh"
8      routes:
9        - route: https://fightMyHero.eu-gb.mybluemix.net/
```

Captura del archivo manifest.yml en Angular.

- **Staticfile:** En este archivo le indicamos que mantenga las URL visibles para poder acceder a las rutas sin problemas y evitar el error *404 nginx* que nos daría la web si no se configura este archivo.
 - *pushstate: enabled:* mantiene limpias las URL visibles del navegador para aplicaciones JavaScript del lado del cliente que sirven múltiples rutas.
 - *force_https: true:* fuerza que todas las solicitudes se envíen a través de HTTPS.



```
Staticfile
1  # Staticfile
2  pushstate: enabled
3  force_https: true
```

Captura Archivo Staticfile ANGULAR

Para que este archivo sea reconocido dentro de nuestra aplicación, a la hora de desplegar será necesario añadir a nuestro archivo *angular.json* la ruta necesaria para que lo encuentre dentro de “assets”.



```
"assets": [
  {
    "glob": "favicon.ico",
    "input": "src",
    "output": "/"
  },
  {
    "glob": "assets",
    "input": "src",
    "output": "/"
  },
  {
    "glob": "Staticfile",
    "input": "",
    "output": "/"
  }
],
```

Captura del archivo angular.json en Angular.

En el proyecto de Node.js, por su parte, solo fue necesario añadir un archivo *manifest.yml*, en el que se especifica el nombre de la aplicación y la memoria que ocupará.

```
! manifest.yml × {} package.json JS index.js
! manifest.yml
1 applications:
2   - name: "backendMyHero"
3     memory: 128M
```

Captura del archivo manifest.yml de Node.js

9.3.3 Despliegue Automatizado

Para conseguir un despliegue automatizado desde IBM Cloud, basta con añadir a la configuración que se realiza para el despliegue la opción que nos brinda las herramientas de IBM Cloud; así, cada vez que se haga una subida a la rama master del repositorio en compilación del proyecto comenzará una nueva etapa automática.

- Desencadenante de etapa
- ☐ Ejecutar trabajos al ejecutar esta etapa manualmente
 - ☒ Ejecutar trabajos automáticamente para sucesos de Git en la rama elegida
 - ☒ Cuando se envía una confirmación por push
 - ☐ Cuando se abre o se actualiza una solicitud de fusión
 - ☐ Cuando se cierra una solicitud de fusión

Captura pantalla configuración despliegue en IBM CLOUD



9.3.4 Uso SSH

Para trabajar con un repositorio privado dentro de IBM Cloud ha sido necesario la creación y uso de una clave SSH que nos permita identificarnos; esta clave se ha generado directamente desde el software Sourcetree, y se ha añadido a la configuración de las herramientas de IBM Cloud.

10. Conclusiones

Paso a comentar mi valoración personal y conclusiones sobre este proyecto: la realización de este proyecto ha sido un logro alcanzado tras haber podido combinar en él todos los conocimientos adquiridos durante estos dos años de formación no solo en el instituto sino en la formación DUAL en la empresa, en mi caso VIEWNEXT, donde nos han ayudado y fomentado en el uso y conocimiento de las últimas tecnologías en el mercado y las nuevas tendencias, fomentando los ideales de trabajo organizado, duro y con capacidad de perseverancia para conseguir los objetivos deseados.

Estoy segura de que los conocimientos que he plasmado en este proyecto y que he ido consolidando y ampliando mediante su realización me serán muy útiles en mi futuro laboral.

A pesar de las múltiples mejoras y funcionalidades que se le podrían añadir a este proyecto, estoy bastante orgullosa del resultado final, más teniendo en cuenta que en algunos momentos me he sentido desorientada o perdida y que esto se ha visto agravado por la situación y bajo las circunstancias actuales en la que se ha desarrollado el proyecto; he de decir que siempre he conseguido resolver los problemas, ayudándome sobre todo de la lectura de documentación y de las diferentes web donde se puede encontrar el apoyo de la comunidad de desarrolladores.



11. Bibliografía / Webgrafía

Diferentes webs que me han resultado útiles para la resolución o la búsqueda de información para realizar esta aplicación:

- <https://www.superheroapi.com/>
API usada para la aplicación
- <https://material.angular.io/>
Web de Angular Material, usada para el diseño
- <https://getbootstrap.com/>
Web de Bootstrap, usada para el diseño
- <https://iconscout.com/icons/star?specialscout=true>
Web con iconos y diversos materiales usados en la app
- <https://dmitripavlutin.com/javascript-classes-complete-guide/>
Web con información sobre JavaScript
- <https://compodoc.app/>
Web con la documentación de la tecnología usada para documentar nuestro Angular, Compodoc
- <https://www.npmjs.com/package/jsdoc>
Web con la documentación de la tecnología usada para documentar nuestro Node.js
- <https://blog.cloudboost.io/adding-swagger-to-existing-node-js-project-92a6624b855b>
Información sobre cómo añadir Swagger a nuestro proyecto Node ya existente
- <http://www.e-nsx.com/2018/02/05/documentacion-automatica-nodejs-jsdoc/>
Guía sobre cómo documentar automáticamente con JSDoc
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
Web para uso de flexbox
- <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
Explicación simple de MVC



- <https://cloud.ibm.com>
Web de IBM Cloud
- <https://nodejs.org/es/download/>
Web de Node.js, con su documentación
- <https://angular.io/guide/setup-local>
Despliegue local de Angular de la documentación de Angular
- <https://angular5col.blogspot.com/2017/11/angular-xampp.html>
Guía de despliegue de Angular en Apache
- https://www.youtube.com/watch?v=_IPZjDL7iIU
Video explicativo de cómo servir una app Node.js desde Apache
- <https://stackoverflow.com/es/q/2583513>
Información de cómo servir una app Node.js desde Apache