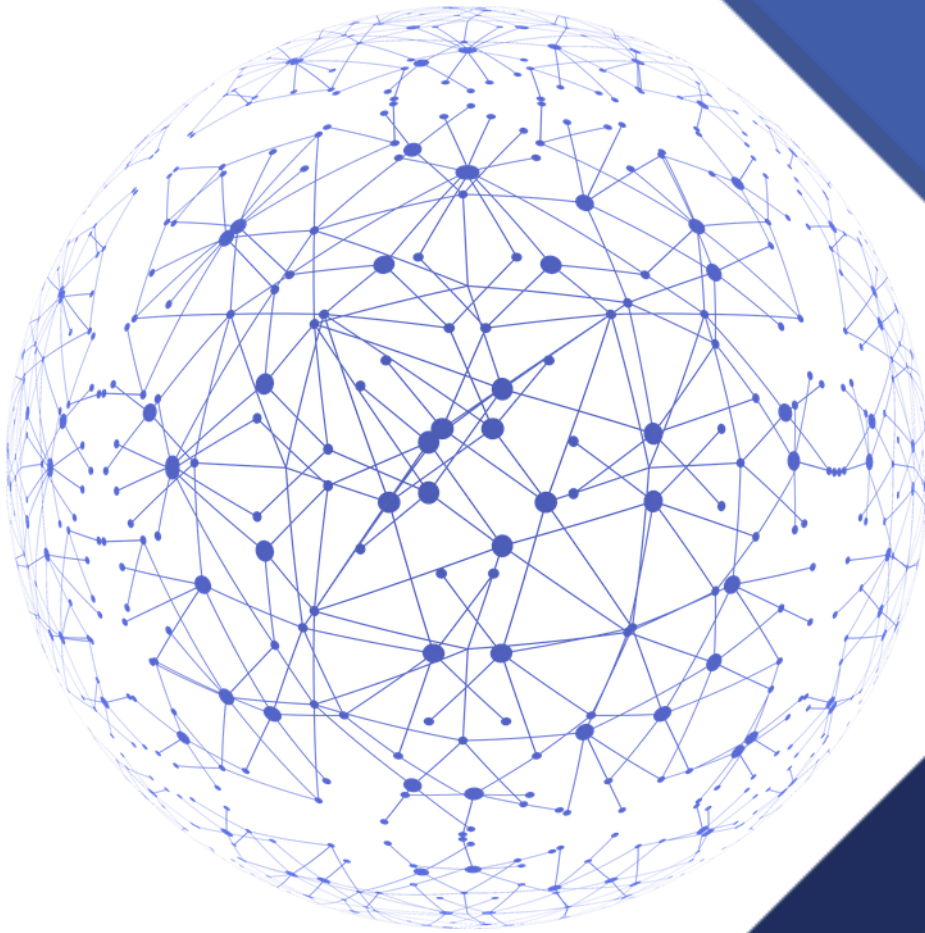


2021



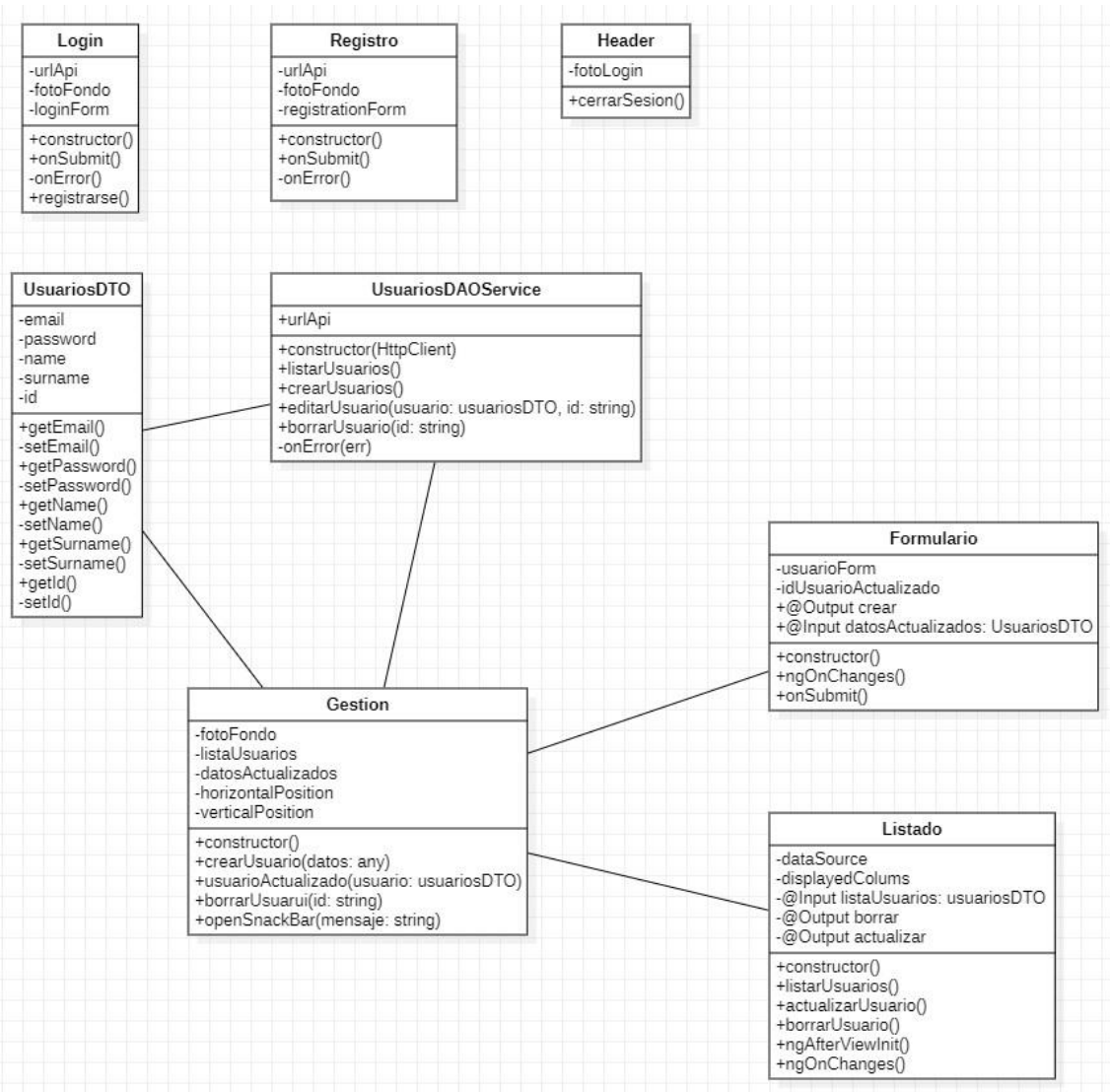
TECHNICAL TEST - HIBERUS

MEAN

Maria Isabel Fuentes Martínez

DIAGRAMA DE CLASES	1
DECISIONES TOMADAS DURANTE EL DESARROLLO.....	2
NIVEL DE ARQUITECTURA A ALTO NIVEL	2
ESCALABILIDAD	2
NOMENCLATURA.....	2
GESTIÓN DE DATOS DE SESIÓN	2
DIFICULTADES ENCONTRADAS Y SOLUCIONES IMPLEMENTADAS.....	2
DOCUMENTACIÓN	3
FUNCIONAMIENTO DE LA APLICACIÓN	3

DIAGRAMA DE CLASES



DECISIONES TOMADAS DURANTE EL DESARROLLO

NIVEL DE ARQUITECTURA A ALTO NIVEL

El patrón de diseño utilizado es el propio de Angular: Modelo Vista Controlador.

Para cumplir los requisitos de escalabilidad y modularidad, he decidido integrar un patrón Data Transfer Object (DTO) y el patrón Data Access object (DAO).

Para gestionar el login y el token he usado un mediador.

También haré uso de los pipe.

ESCALABILIDAD

Para sacar el máximo partido a los módulos y componentes de Angular, he creado 3 módulos principales: login, Shell y usuarios. El login contendrá los formularios de inicio y de registro, Shell contiene los componentes footer y header, mientras que usuario va a tener toda la lógica de la aplicación una vez que el usuario ha sido logeado.

NOMENCLATURA

He usado la guía de Google: <https://google.github.io/styleguide/jsguide.html>.

GESTIÓN DE DATOS DE SESIÓN

He usado el Sesión Storage de HTML5.

DIFICULTADES ENCONTRADAS Y SOLUCIONES

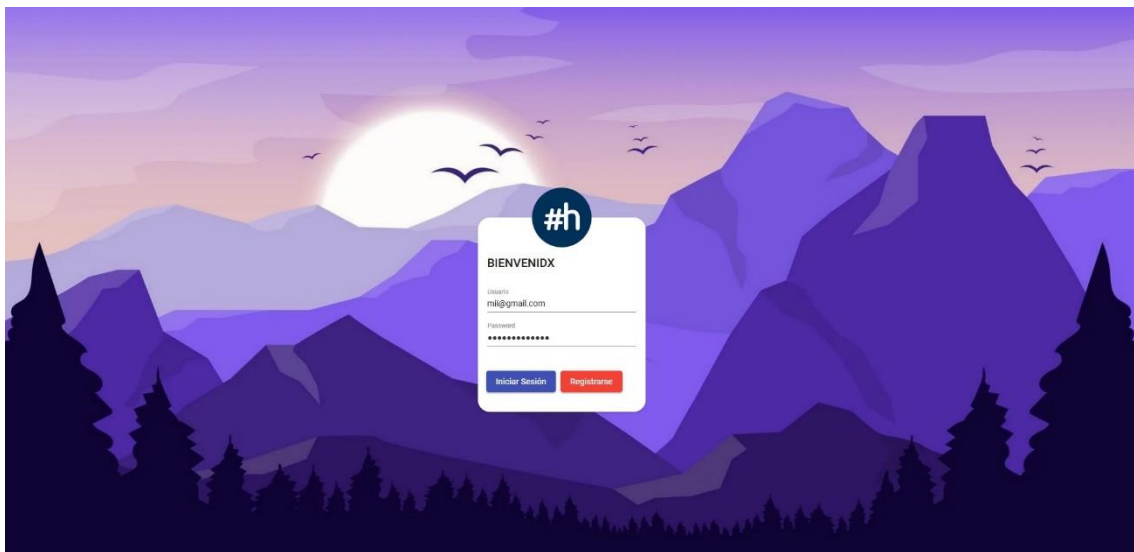
IMPLEMENTADAS

Una de las dificultades ha sido que, cuando editábamos, después debería dejarnos insertar de nuevo sin tener que recargar la página. Para solucionarlo bastó con inicializar los datos del usuario a nulo.

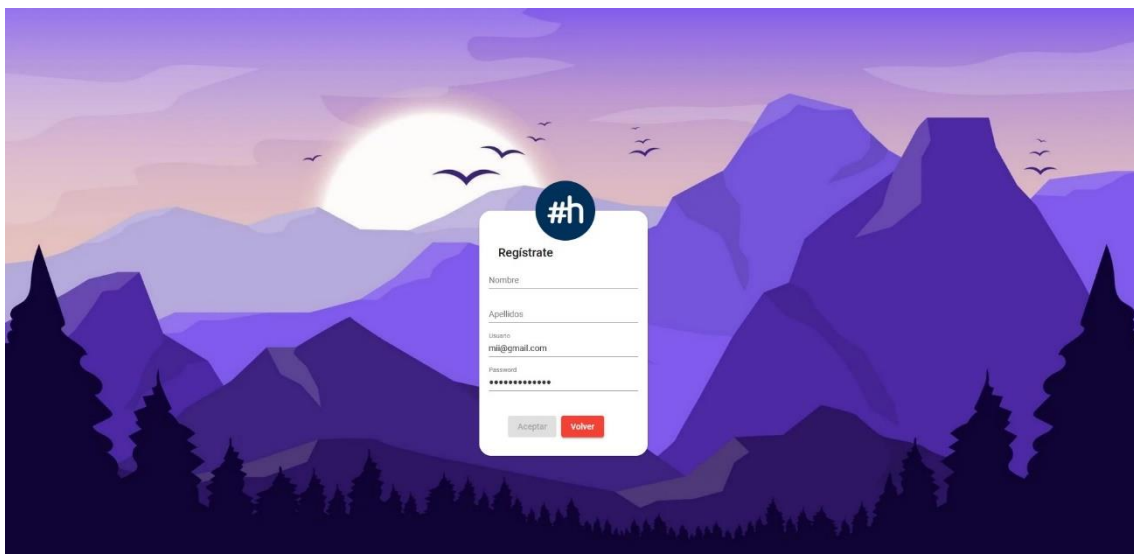
DOCUMENTACIÓN

FUNCIONAMIENTO DE LA APLICACIÓN

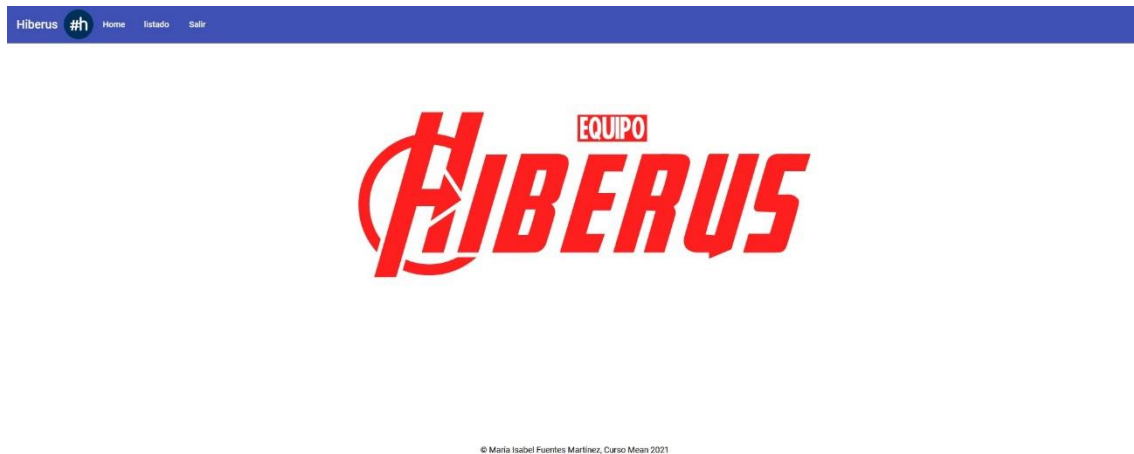
Cuando entramos a la aplicación, nos recibe un módulo de login que se va a encargar de logearnos o por el contrario, permitir que nos registremos. El usuario puede decidir entre iniciar sesión, o registrarse.



El formulario de registro es idéntico que al anterior solo que tiene nuevos campos para poder registrarse con éxito. Tenemos las validaciones necesarias para que, si tenemos algún error en los campos, no permitir que el usuario se registre. Además, hasta que todos los campos no estén rellenos, no se habilitará el botón de aceptar.



Una vez hayamos entrado a la aplicación, nos recibirá una página de Home con el header de la aplicación y el footer, ambos estarán presentes en todas las páginas que mostremos al usuario una vez logueado.



El header contiene la barra de navegación, cuyas operaciones son llevarnos de nuevo al home, mostrar el listado de usuarios y, permitirnos salir de la aplicación. Cuando salimos de la aplicación eliminamos el token, y en el servicio Auth-interceptor (método intercept) no permitimos entrar si no tenemos token. De esta forma, si el usuario hace click en salir y después pulsa retroceso en el navegador, no se le permitirá entrar.

Va a ser la opción de listar la que nos muestre la tabla con todas las peticiones a la API de forma asíncrona: añadir, editar y borrar.

