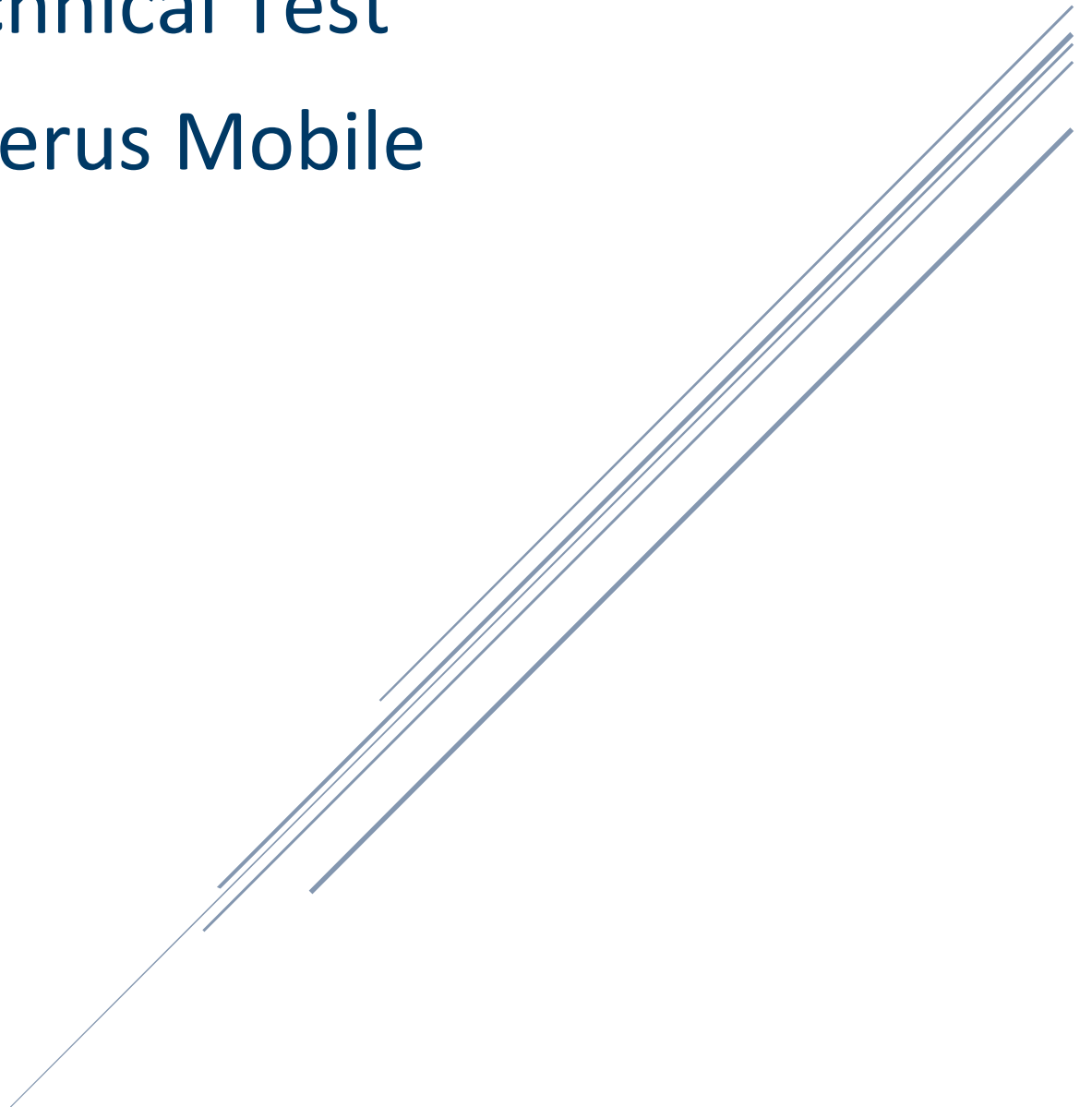




Technical Test

Hiberus Mobile



Regina Rodríguez Cárdenas

Contenidos

1. Tecnologías y herramientas utilizadas	2
2. Estimación y planificación	3
3. Problemas encontrados	5
4. Mejoras.....	5

1. Tecnologías y herramientas utilizadas

Para la realización de esta aplicación web he usado Angular 14.1. La decisión de este framework ha sido meramente por gusto y practicidad, ya que es el único de los tres que hemos aprendido con Typescript. El código que se escribe en Typescript tiene las ventajas que hereda de Javascript y una fiabilidad en el código, el cual te hace identificar los errores antes y te evita cometerlos gracias al tipado fuerte. Por otro lado, la división de componentes en este framework personalmente me resulta ordenado y visualmente agradable, habiendo una separación distinguible entre el template, la hoja de estilos, y el archivo ts.

Como el tiempo era limitado, he sacrificado la parte estética por la funcional, por lo que instalé Bootstrap para, mientras pintaba o enviaba los datos, ir aplicando una serie de estilos homogéneos y responsivos. Aparte, añadí la librería Bootswatch para añadir estilos diferentes, cogiendo el modelo vapor, que se mueve entre unas tonalidades rosas, moradas y azules, estilo vaporwave, una estética que en lo personal me encanta.

A la hora de guardar los tokens, escogí sessionStorage sobre localStorage porque al abrir en otra pestaña el proyecto no se solapaban las sesiones y era más fácil hacer pruebas. Entiendo que de cara a un cliente es más eficaz localStorage porque no se pierden los valores, aunque por otro lado y a priori, sessionStorage parece más seguro que su contraparte.

Por último, realicé el despliegue en Vercel, pudiéndose acceder desde el enlace <https://wikiusers.vercel.app>. Usé esta plataforma porque te actualiza a tiempo real cada commit de Github que ejecutas, con lo cual se iba actualizando diariamente. Sin embargo, al no detectarse Swagger como una plataforma segura por tener protocolo http, no se puede acceder a los datos.

2. Estimación y planificación

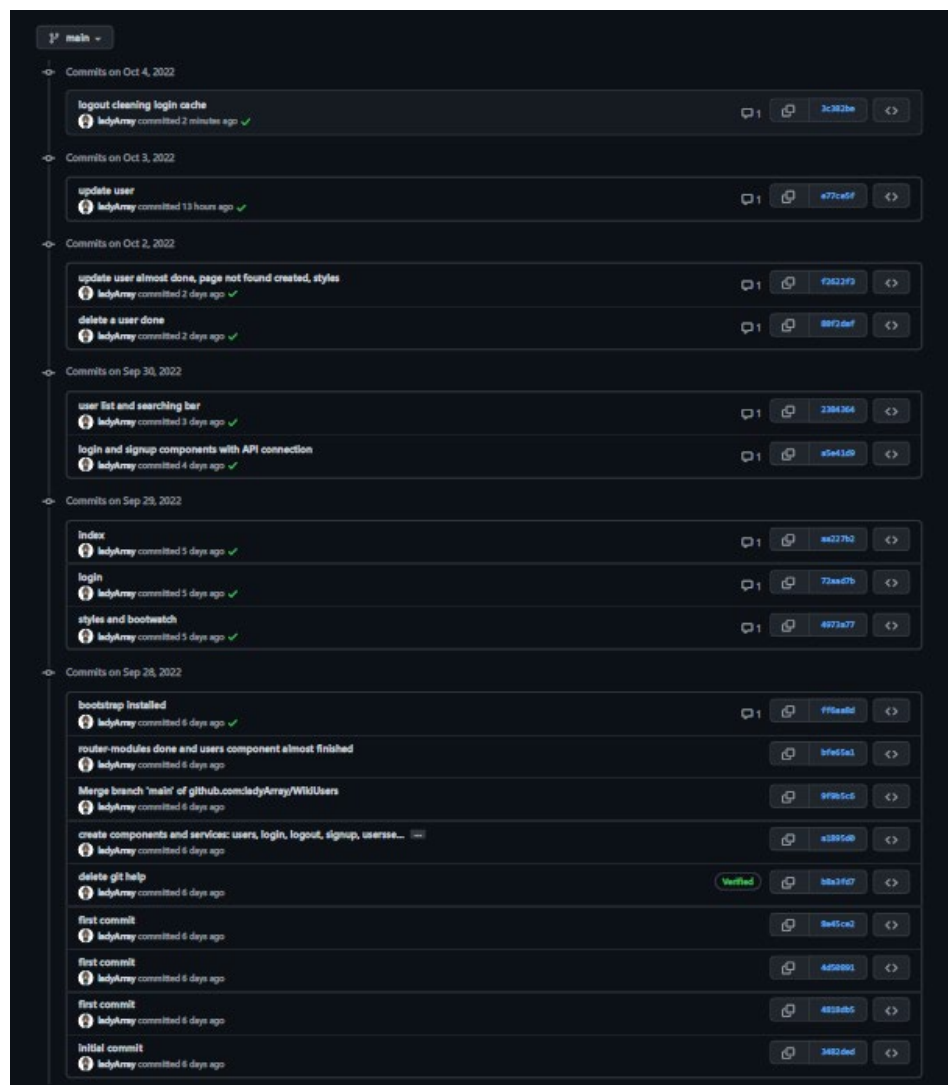
El principal objetivo es una aplicación web con 4 componentes (login, signup, logout y users). Para un trabajo eficiente en un tiempo limitado, los he dividido en objetivos específicos, y cada día llegaba a cumplirlos o los dejaba como prioritarios al día siguiente. Para representar las tareas previstas, así como su control y seguimiento en cada una de las etapas y días, he elaborado un diagrama de Gantt.

Nota: El sábado 1 de octubre no aparecerá representado ya que ese día no avancé con el proyecto.

TAREA	INICIO	FIN	DÍAS	X	J	V	S	S	L	M
Inicio del proyecto	28/9	28/9	1							
Creación del proyecto y del repositorio	28/9	28/9	1							
Creación de los componentes principales: users, login, logout, signup.	28/9	28/9	1							
Creación de los servicios: UserService, LoginService.	28/9	28/9	1							
Instalación de Bootstrap.	28/9	28/9	1							
Enrutamiento de los componentes.	28/9	28/9	1							
Creación del header y footer, únicas partes fijas en la aplicación.	29/9	29/9	1							
Instalación de Bootswatch y adaptación de estilos en el .css de cada componente.	29/9	29/9	1							
Creación del formulario de login y signup en el html.	29/9	29/9	1							
Investigación del proceso de autenticación con JWT y sessionStorage para cubrir las dudas en la reunión del Viernes 30.	29/9	30/9	2							
Conseguir la conexión de la API con el login y el signup.	30/9	30/9	1							
Usando el token de sesión, acceder al componente del listado de usuarios y pintar el listado (función search).	30/9	30/9	1							
Conseguir la funcionalidad de borrar un usuario.	2/10	2/10	1							
Conseguir la funcionalidad de actualizar un usuario.	2/10	3/10	2							

Implantar el logout con su correspondiente borrado de caché.	3/10	3/10	1							
Pruebas de funcionamiento.	3/10	3/10	1							
Redacción de la documentación.	28/9	4/10	6							
Implementación y envío	4/10		1							

En esta imagen podemos ver un listado de los commits realizados en Github a lo largo de los días:



3. Problemas encontrados

A lo largo del ejercicio, he ido encontrando una serie de dificultades que me han atrasado el proceso:

- a) Desconocimiento de la API en su funcionamiento: Al haber hecho todas las tareas a lo largo del curso con Firebase mayormente, ha habido un trabajo de investigación detrás que ha conllevado tiempo, especialmente con sessionStorage, el objeto que almacena los tokens y que es necesario para el acceso a la parte de usuarios.
- b) Solapamiento de usuarios en la API: Al acceder mis compañeros y yo a todos los datos sin ningún tipo de restricción, nos hemos editado y borrado los datos entre nosotros, lo que ralentizaba las pruebas al no poder iniciar sesión y, por consiguiente, estar creando nuevos usuarios.

4. Mejoras

Como añadido, quería documentar un listado de mejoras que podrían aplicarse en este proyecto en iteraciones posteriores:

- 1. Estética: Un evento onClick que haga que el listado de los usuarios sea editable directamente o poder borrarlo sin necesidad de copiar y pegar el id, haría una interfaz más amigable e intuitiva y ahorraría tiempo al usuario.
- 2. Accesibilidad: Si este fuera un producto como tal, se tendría que escoger una paleta de colores que pudiera ser vista nítidamente por todos los usuarios, y se colocaría aria-label en todas las etiquetas correctamente.
- 3. Manejo de errores: Los errores como 404, 401 o 409 podrían ser tomados como excepción y devolver como alert o en forma de mensaje una explicación de cada uno de ellos.