



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

School of Professional & Executive Development

Trabajo final de postgrado Full-Stack Web Technologies



Conjunto de aplicaciones Web y Android para la gestión y
organización de pedidos en servicios de restauración

Marta Adell Noguera
Miguel Conde Iglesias

Barcelona, septiembre 2020

Índice

1. Introducción	3
2. Arquitectura y flujo de aplicaciones	5
3. Modelo de datos	6
4. Base de datos	6
5. Front-End	8
5.1. Propuesta de diseño	8
5.2. Aplicación web	10
5.3. Aplicación móvil	16
6. Conclusiones	21
7. Código fuente	22

1. Introducción

El desarrollo Full-Stack se refiere al manejo de cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web. Es decir, al desarrollo tanto del Front-End y Back-End, junto con lenguajes de programación y sistemas operativos necesarios para ello.

Como trabajo final del postgrado se ha realizado este proyecto, que pretende abarcar los diferentes campos que trata un desarrollador full-stack en su día a día. De la misma manera se ha intentado repartir el trabajo como

Se plantea la situación actual en la que, debido a las restricciones por el Covid-19, las aplicaciones Web se han popularizado aún más entre los clientes, ya que son una opción cómoda que permite la total individualización del producto. A modo de enfocar el trabajo a un caso realista y en vista al panorama actual, se plantea una propuesta que fácilmente podría ayudar a un posible cliente.

La propuesta final, **BarApp**, trata de un conjunto de dos aplicaciones que conectan a los usuarios (local de restauración y cliente) para facilitar el manejo de comandas y su comunicación. Para ello se usa una base de datos común que comunica ambas aplicaciones y su información.

1.1. Objetivo y requerimientos

Una vez planteada la meta del proyecto, que sería el desarrollo total de la aplicación, se plantean un seguido de requerimientos que ayudarán a definir con mayor claridad el objetivo final, observando también el límite de tiempo para su realización.

En este caso, debido al reducido número de componentes del grupo y sus respectivos perfiles (más dedicados al Front que Back-End), se ha optado por la opción de incorporar *firebase* para alimentar las aplicaciones, reduciendo así el trabajo en el campo de bases de datos y Back-End. El proyecto se ha enfocado más al desarrollo Front-End.

Por lo tanto, los requerimientos finales serán los siguientes:

- App móvil
 - Lector de QRs para hacer las comandas.
 - Consulta a los platos del bar/restaurante.
 - Escritura de comandas: selección de cantidad de platos y notas.
 - extra: disponibilidad de historial de comandas con la identificación de usuario.
- Web app
 - Identificación de usuario para acceder a los datos del restaurante.
 - Consulta y escritura de la información del bar/restaurante: número de mesas y platos.
 - Consulta de las comandas del usuario de la app móvil + escritura para mostrar el estado actual de la comanda.
- Frontend: diseño básico, minimalista y común en ambas aplicaciones.
- API → firestore con toda la información.

2. Arquitectura y flujo de aplicaciones

Se presenta una estructura simple: el bar/restaurante dispone de una aplicación web para la gestión de los platos que ofrece y las mesas que dispone para sus clientes, y dichos clientes pueden visualizar desde la aplicación móvil el contenido del bar/restaurante para posteriormente hacer el pedido, escaneando un código QR que los relaciona con la mesa desde la cual hacen el pedido.



Figura 1: Arquitectura a alto nivel

Este esquema se traslada virtualmente a lo siguiente: las aplicaciones web y móvil, que funcionan independientemente, se enlazan mediante una base de datos *firebase*.

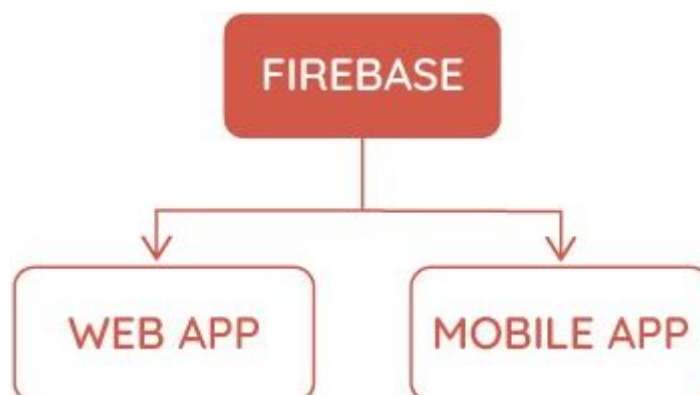


Figura 2: Arquitectura a bajo nivel

3. Modelo de datos

El modelo de datos final usado para el proyecto es el siguiente:

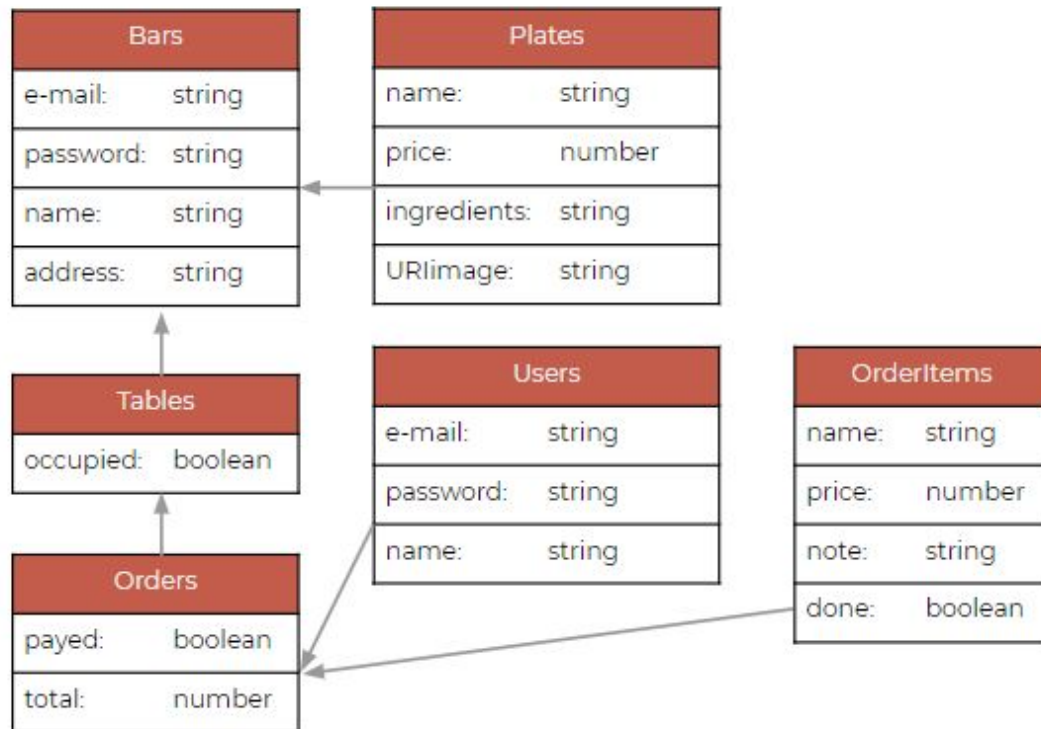


Figura 3: Modelo de datos

4. Base de datos

Para el almacenamiento y clasificación de datos se ha creado un proyecto *firebase* para conectar con las dos aplicaciones. Se han usado distintas herramientas de la plataforma.

- Firestore: base de datos del modelo
- Storage: almacenamiento de imágenes (relacionado con la base de datos con las urls de las fotos)
- Autenticación: registro de usuarios para ambas aplicaciones, también relacionado con la base de datos.

Storage

/barId	/platesimages	image ...	← URI stored in firestore as imageURI from platId
--------	---------------	--------------	--

Authentication

email	password	displayName	← stored in firestore as barId for bar users
-------	----------	-------------	---

Firestore

collection	document	fields
------------	----------	--------

/bars	/barId	address					
		name					
	/menu	/itemsList	/starters	/platId	name		
			/main	"	price		
			/desserts	"	ingredients		
			/drinks	"	imageUri		
	/orders	/orderId	payed				
			total				
			userId				
		/orderItems	/orderItemId	done			
				name			
				price			
				note			
	/tables	/tableId	occupied				
			orderId				

Figura 4: Distribuciones firebase

5. Front-End

5.1. Propuesta de diseño

Uno de los objetivos planteados al inicio del proyecto es ofrecer una interfaz simple y fácil de usar. Desde el primer momento se ha planteado el conjunto de aplicaciones con un diseño minimalista, en fondo blanco y rompiendo el monocromo a ser posible con un solo color acentuador.

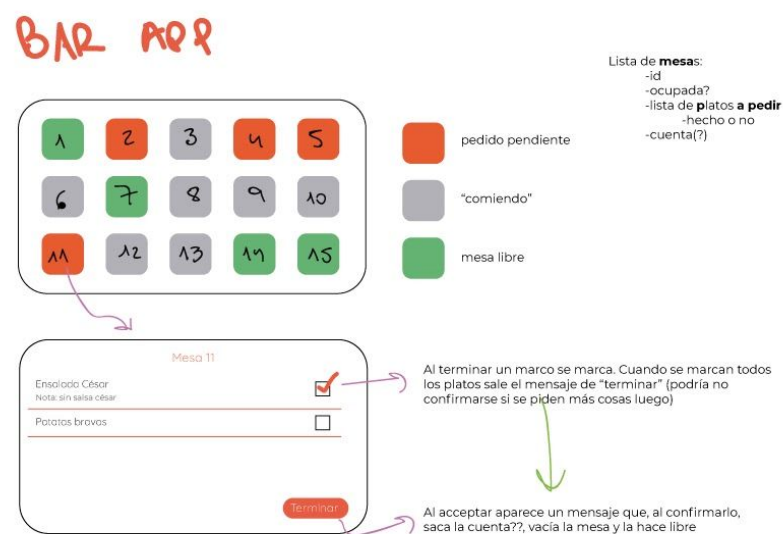


Figura 5: primer esbozo de BarApp Web

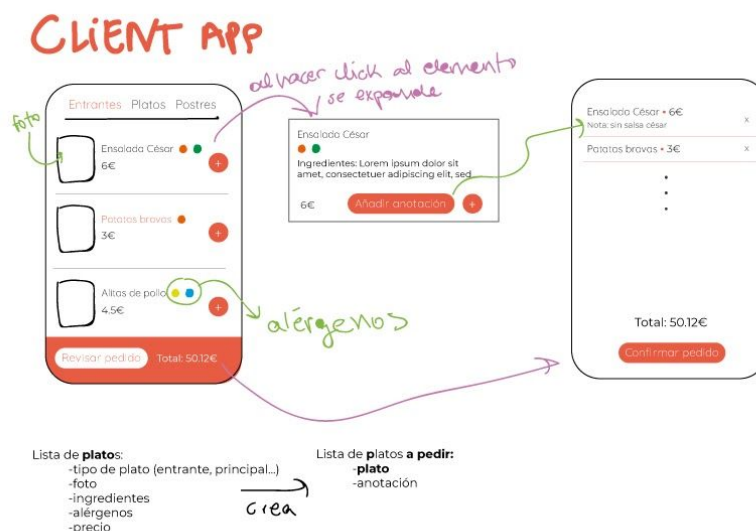


Figura 6: primer esbozo de BarApp móvil

Este paso supone un poco de dificultad, así que a pesar de haber presentado inicialmente unas variantes de layouts al final ha sido la propia experiencia a la hora de probar las aplicaciones la que ha ido moldeando y finalmente determinado su diseño final.

Se ha optado por usar una paleta de colores y distintas tipografías que identificarán la aplicación y sus elementos. Para los iconos y *buttons*, se ha optado por usar formas redondeadas para hacer el diseño más atractivo y uniforme.

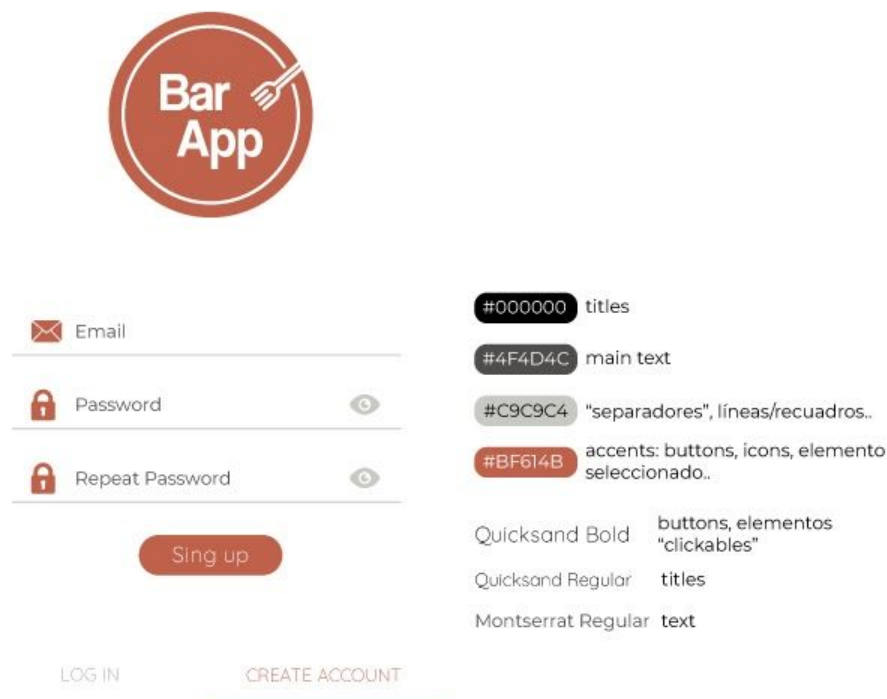


Figura 7: propuesta de diseño de BarApp móvil

Finalmente para identificar las aplicaciones se ha creado un logotipo con el nombre de ellas: BarApp. También simple y reflejando de la forma más directa su función:



Figura 8: logotipo



Figura 9: icon

5.2. Aplicación web

La aplicación web está dirigida al bar/restaurante, y tiene como funcionalidades principales el registro, acceso, listado de mesas del local, visualización del pedido de cada mesa y su código QR de acceso y visualización y edición de los platos del bar/restaurante. Además de ello, el usuario puede editar la información del nombre y dirección del local y modificar el número de mesas de las cuales dispone.

Al iniciar la aplicación, en caso de no haberse registrado ni accedido como usuario, se presenta el componente **Login**, que dispone la acción de cambiar al **Register**. Para registrarse se requieren los campos de: e-mail, contraseña, nombre de usuario (que funcionará como id del bar para la base de datos), nombre del bar, dirección y número de mesas, que definirán el bar a registrar al enviar el *form*. Para iniciar sesión se necesitan el e-mail y contraseña proporcionados en el registro.

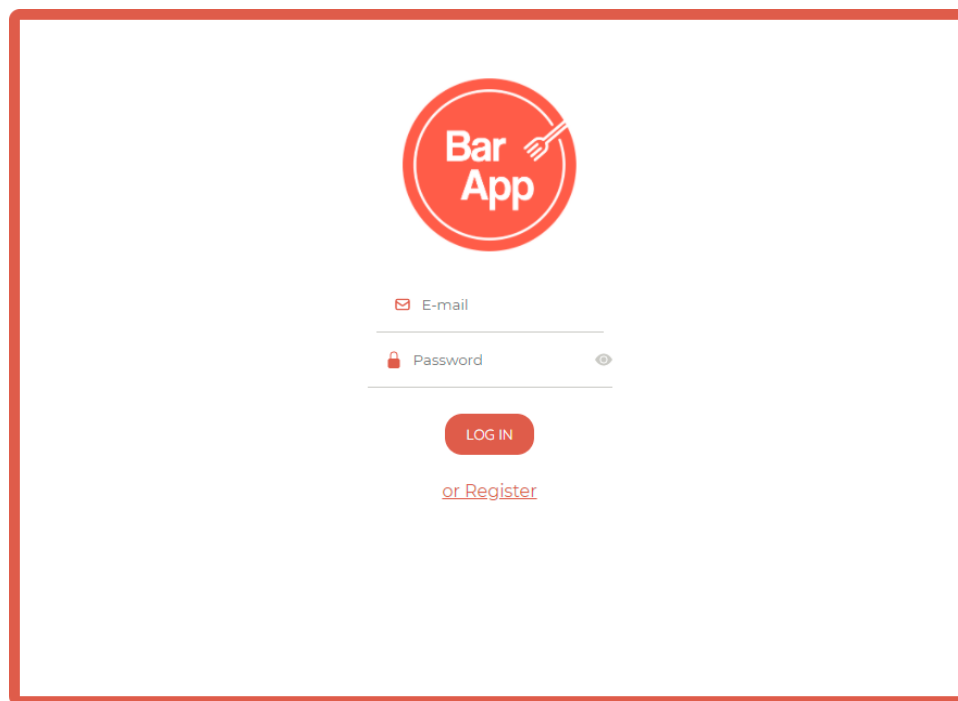
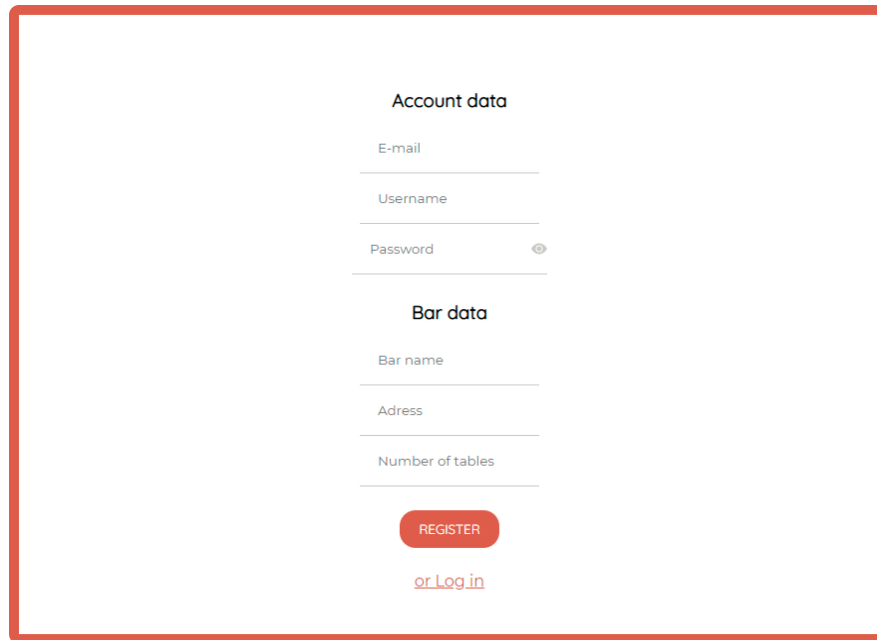


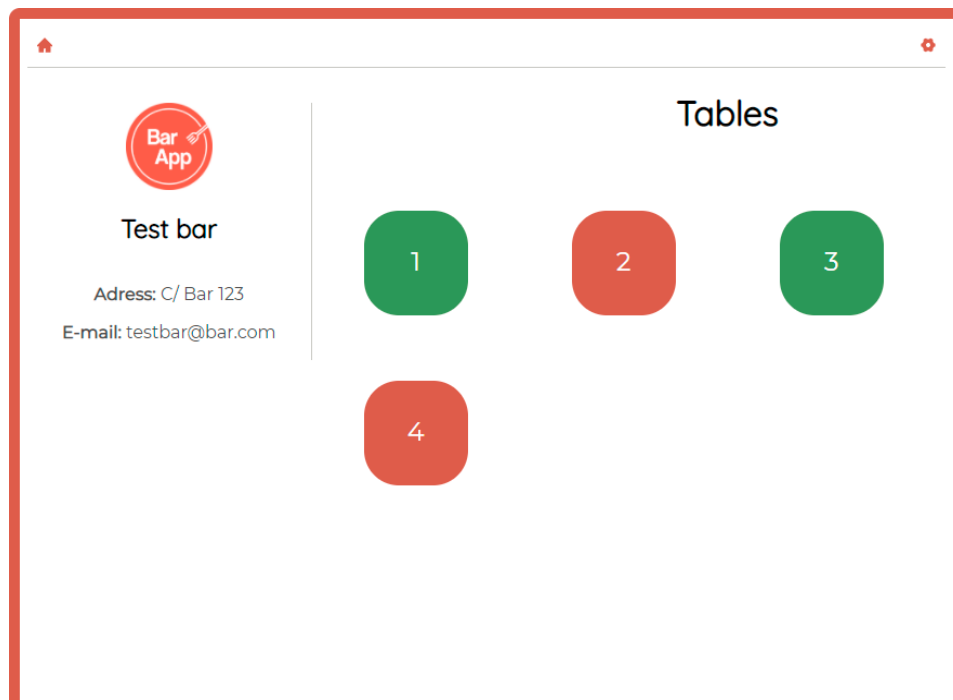
Figura 10: Inicio de sesión en la aplicación web



The registration form is divided into two sections: 'Account data' and 'Bar data'. The 'Account data' section includes input fields for 'E-mail', 'Username', and 'Password' (with a toggle for visibility). The 'Bar data' section includes input fields for 'Bar name', 'Adress', and 'Number of tables'. Below these fields is a red 'REGISTER' button and a link 'or Log in'.

Figura 11: Registro de usuario en la aplicación web

En caso de registrarse la aplicación dirigirá al usuario directamente a **MenuScreen**, donde podrá introducir todos los datos del menú del bar/restaurante. En caso de iniciar sesión, se irá directamente a **HomeScreen**.



The home screen features a header with a home icon and a settings icon. On the left, there is a 'Test bar' section with the 'Bar App' logo, the bar name 'Test bar', and contact information: 'Adress: C/ Bar 123' and 'E-mail: testbar@bar.com'. On the right, under the heading 'Tables', there are four numbered buttons: 1 (green), 2 (red), 3 (green), and 4 (red).

Figura 12: Página de inicio

Dentro de **HomeScreen** se disponen 3 componentes principales. Arriba de todo está el **Header**, presente en todo momento excepto en las pantallas de registro e inicio de sesión, proporciona de nuevo acceso a **HomeScreen** y los ajustes o **Settings**. Seguidamente, al lado izquierdo de la pantalla está el **SideMenu**, que muestra la información principal del bar/restaurante. Finalmente **TablesScreen**, a la derecha, expone en forma de cajas las distintas mesas del local. En caso de que alguna de las mesas tenga un pedido activo, esta se muestra de color rojo y es accesible por el usuario.



Figura 13: Visualización del pedido de la mesa 2

Al hacer click en alguna de las mesas ocupadas, la aplicación redirige al usuario a **OrderScreen**, que muestra el pedido activo de dicha mesa (proporcionándole el número de esta) mediante el componente **OrderItemsList** junto con el código QR de la mesa.

OrderScreen se sincroniza con el documento correspondiente al pedido establecido para la mesa seleccionada, mostrando, además de la lista de platos pedidos con su respectivo nombre (en **OrderItemsList**), precio, estado (hecho o no) y notas (en caso de incluirlas), el cálculo total del pedido y estado de pago (completado o no). En caso de completar el pedido hay una opción que limpia la mesa y vuelve a marcarse como libre (el pedido quedará guardado en el registro para futuras consultas desde la otra app).

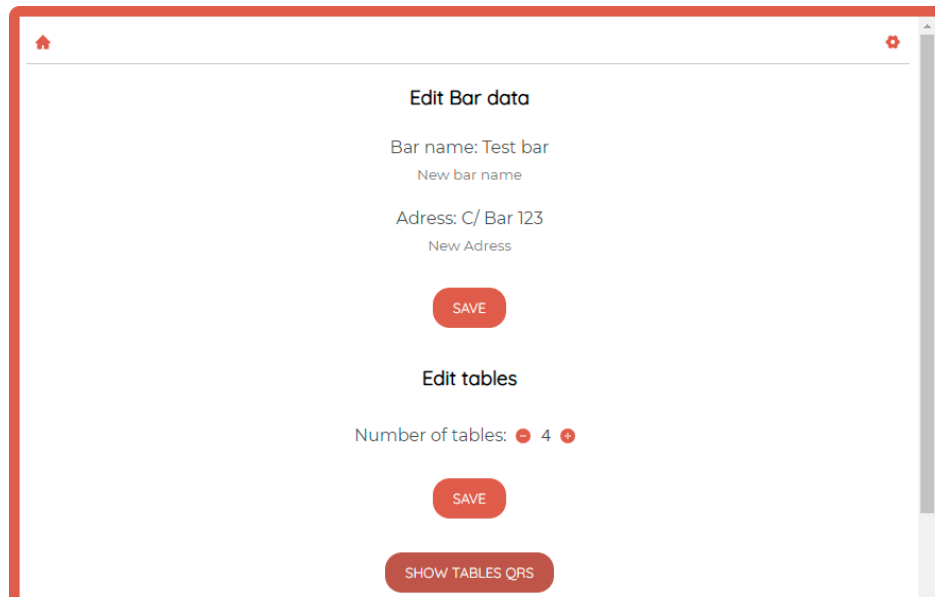


Figura 14: Página de ajustes: Información de usuario y mesas

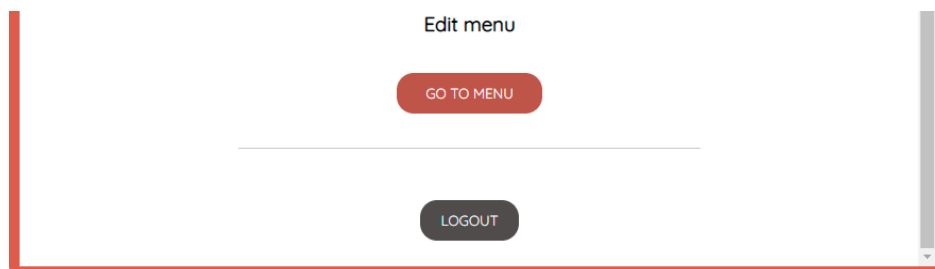


Figura 15: Página de ajustes 2: menú y logout

Volviendo a las funciones del **Header**, la ruedecita de la derecha lleva a **SettingsScreen**, que contiene los componentes **UserEditor**, con un *form* que permite cambiar la información de usuario (nombre y dirección), y **TablesEditor**, con otro para modificar el número de mesas y visualizar sus códigos QR. Además de estos componentes, desde

esta página se puede acceder al menú de platos del bar/restaurante y modificarlo (redireccionamiento a **MenuScreen**) o, finalmente, cerrar sesión.

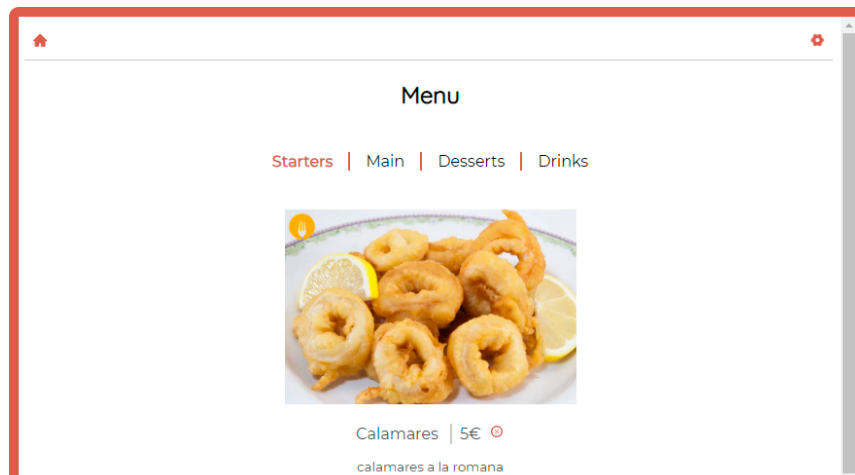


Figura 16: Carta del bar/restaurante

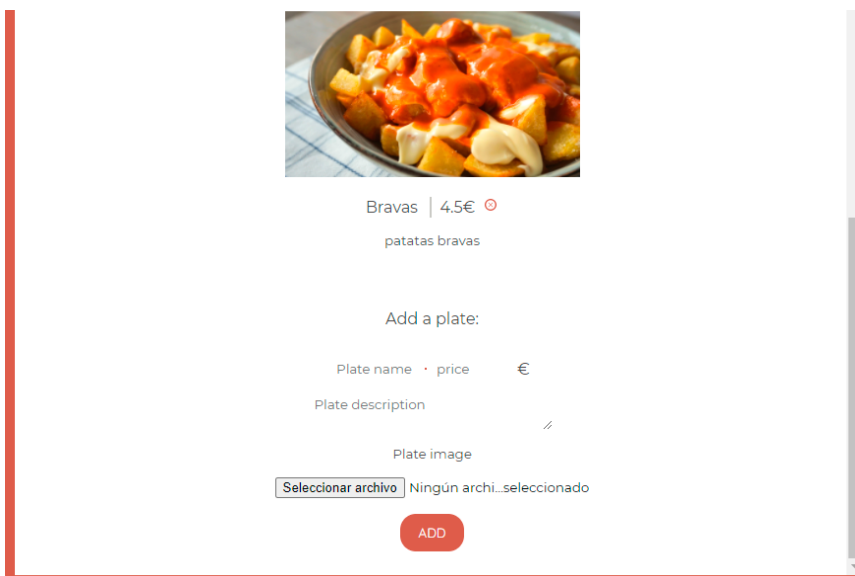


Figura 17: Carta del bar/restaurante 2: form

Finalmente en esta página se muestra la carta completa del local, categorizados por tipo de plato. Debajo de todo se dispone **PlateForm**, que consiste de un *form* para añadir más platos a la carta (dentro de cada categoría). Los campos a rellenar son: el nombre, precio (obligatorios), definición e imagen (opcionales). Una vez añadido el plato se sincroniza automáticamente con la lista **PlatesLists** que muestra la lista actualizada.

5.2.1. Incorporación de *firebase* en la aplicación

Tal y como se ha mencionado en el apartado de base de datos, se ha utilizado la autenticación de *firebase* para la gestión de usuarios. En esta aplicación se ha creado el *hook* **useAuth** para manejarlo todo más fácilmente. Este *hook* contiene los métodos básicos de la autenticación de *firebase*, de manera que facilita la gestión del usuario y el estado de este en todo momento.

En los componentes **Login** y **Register** se utilizan los métodos correspondientes para iniciar sesión y crear un usuario, y en el resto de la aplicación únicamente se consultan los datos del usuario conectado.

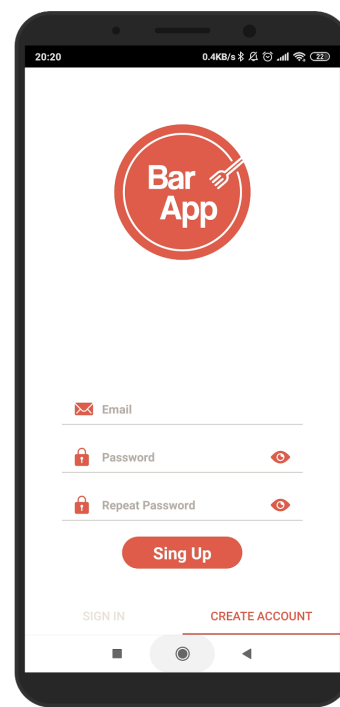
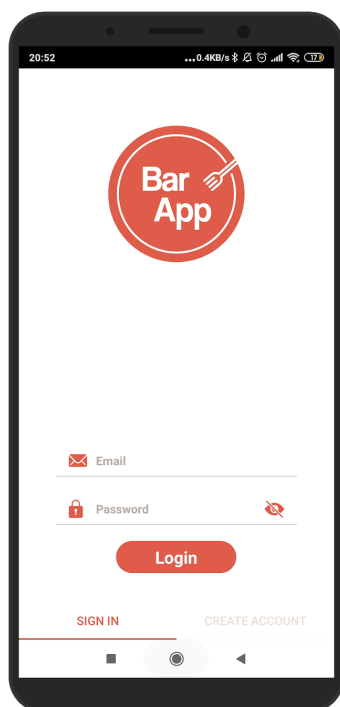
Para la conexión entre el usuario autenticado y la colección de datos se ha usado el campo **displayName** del usuario como **barId** de la colección de *firestore*, definiéndolo como el **username** del usuario bar/restaurante. Siendo el elemento principal que identifica la colección de datos, se ha programado la aplicación para no poder ni repetir dicho username ni cambiarlo una vez registrado; de esta manera se evitan posibles desvinculaciones entre las funciones de *firebase*.

Si bien la colección de *firestore* contiene todos los datos del usuario, esta función no ofrece la opción de almacenar archivos, así que se ha optado por usar también el *storage* de *firebase* para guardar las imágenes. Para cada usuario bar/restaurante, se crea una carpeta nombrada con el **username** proporcionado en la aplicación para distinguirlo, y dentro se almacenan las imágenes de los platos del bar/restaurante. Paralelamente, en *firstore* se referencian los url de dichas imágenes de cada uno de los elementos.

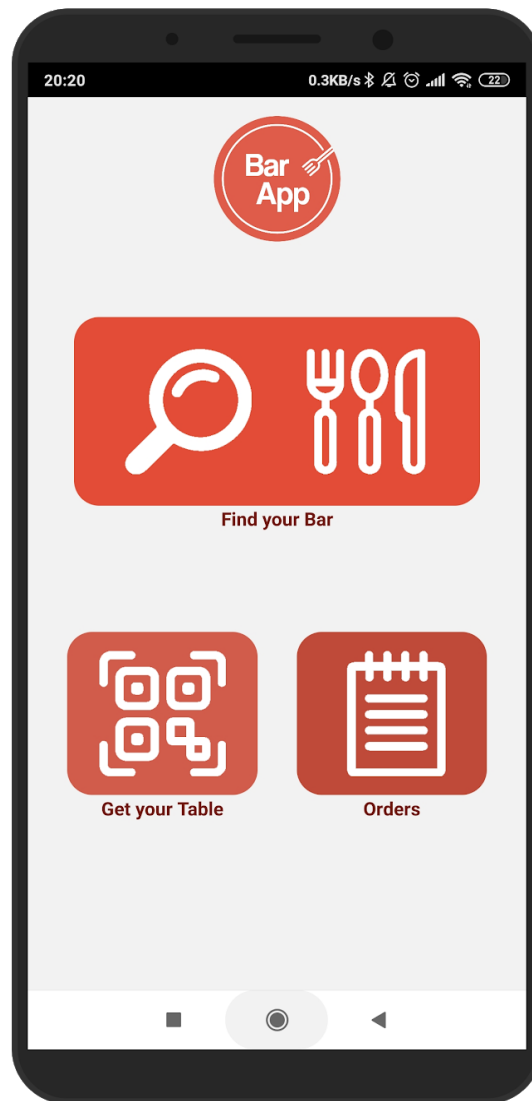
5.3. Aplicación móvil

La aplicación móvil está dirigida a los clientes, y tiene como funcionalidades principales el registro, acceso, listado de bares registrados en la plataforma, escanear código QR de las mesas para poder realizar pedidos en los bares, y visualización y selección de los platos del bar/restaurante y revisar el histórico de las comandas que ha realizado en los diferentes bares a través de la app de BarApp. Dicha app está creada con el Framework de React Native.

Al iniciar la aplicación, en caso de no haberse registrado ni accedido como usuario, se presenta el componente **Login**, que dispone de una pestaña para cambiar a **Sing Up**. Para registrarse se requieren los campos de: e-mail (que funcionará como id del propio usuario para la base de datos), contraseña. Para iniciar sesión se necesitan el e-mail y contraseña proporcionados en el registro. Tanto el **Login** como el **Sing Up** se validan contra el servicio de firebase auth.

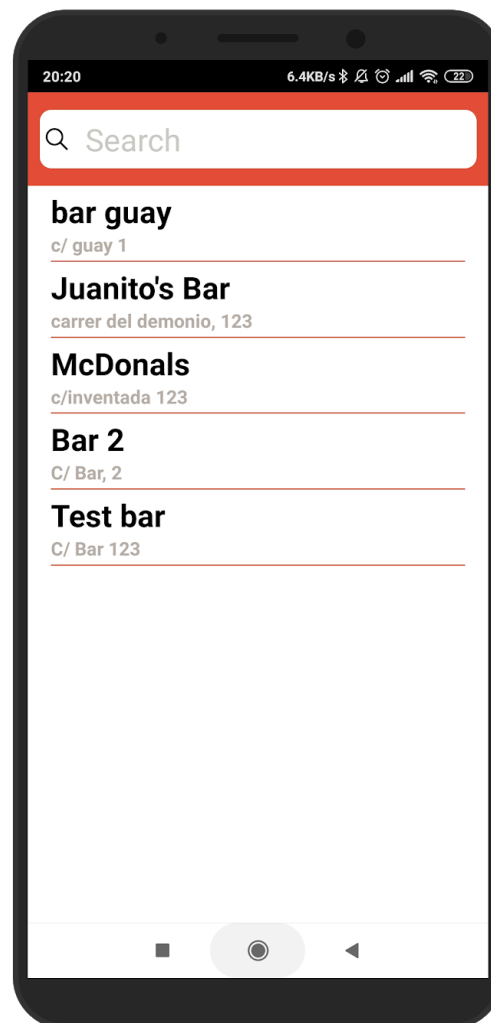


Al acceder mediante el **Login**, pasaremos a la pantalla de la interfaz, donde podremos gestionar la aplicación.

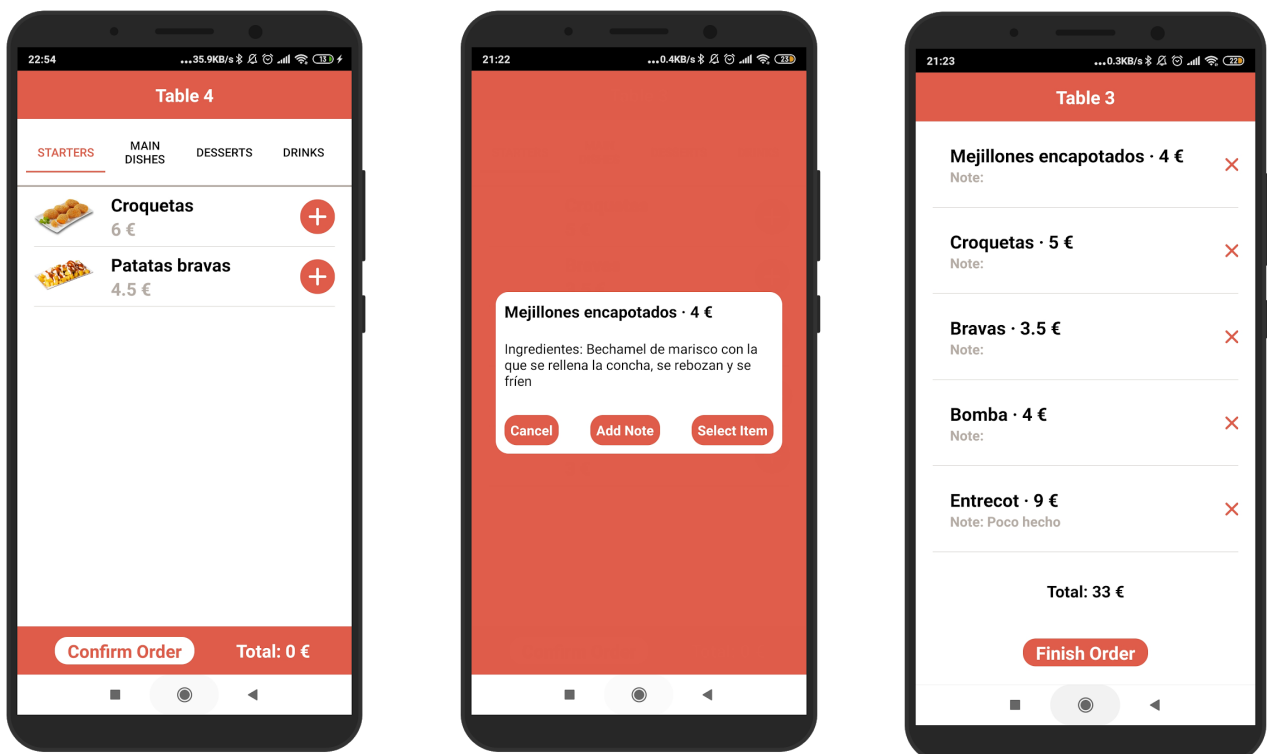


En esta pantalla encontraremos 3 botones con diferentes funciones para cada uno de ellos, **Find your Bar**, **Get your Table** y **Orders**. Explicamos dichas funciones en las siguientes páginas con una imagen que acompaña la explicación.

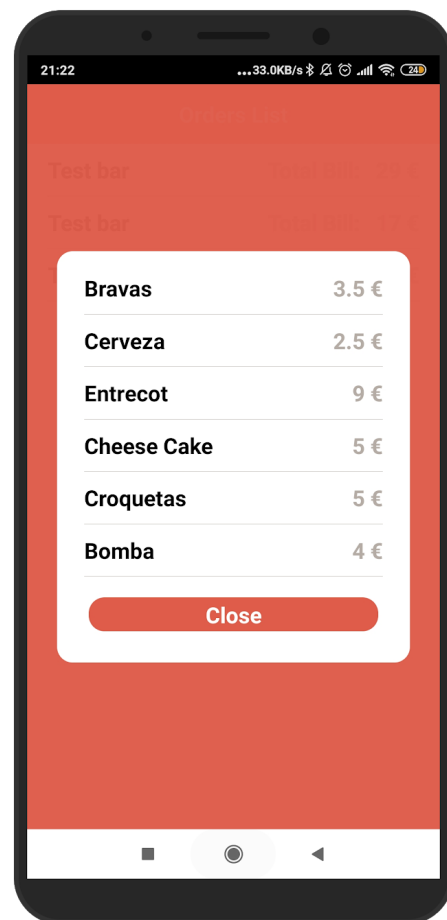
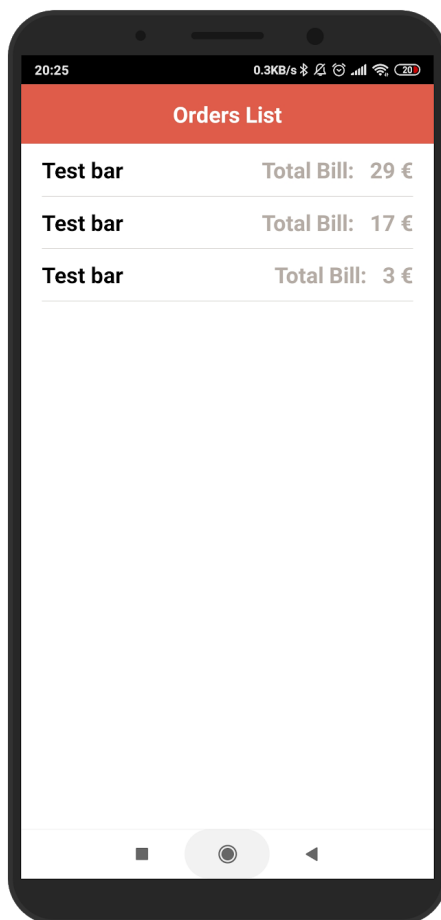
Al pulsar el botón de **Find your Bar**, accederemos a la pantalla donde nos aparecerá todos los bares/restaurantes registrados en la plataforma de BarApp. El componente de la pantalla es un FlatList de React Native. Además, tiene un Input Text dónde nos permitirá hacer una búsqueda por nombre de cualquier bar/restaurante que esté registrado en la plataforma. Esta funcionalidad de Find your Bar, estaba destinada para hacer reservas de mesa en cualquier bar, pero por complicaciones técnicas no hemos acabado de desarrollar dicha funcionalidad.



Al pulsar el botón de **Get your Table**, se nos activará la cámara del smartphone para poder escanear los códigos QR de las mesas de los bares/restaurantes y así poder acceder a la pantalla del menú de dicho bar. Si tratamos de escanear un código QR que no sea perteneciente a la plataforma, saltará un mensaje explicando que dicho código QR no es válido para proceder al menú del bar. En la pantalla del menú, nos aparecerán las pestañas “Starters”, “Main Dish”, “Desserts” y “Drinks”. En cada una de las pestañas aparecerá los platos que dispone el bar para poder seleccionar. En cada plato hay un botón de selección del plato, y al pulsarlo aparece un “Modal” donde podrás añadir una nota para el cocinero y seleccionar el plato. Después de seleccionar algún plato podremos pulsar el botón de “Confirm Order” que nos llevará a la pantalla con el resumen de los platos seleccionados. Allí podremos ver un resumen de lo seleccionado y borrar alguno de los platos seleccionados.



Por último, al pulsar el botón de **Orders** de la pantalla interfaz de la aplicación, accederemos a una pantalla donde nos aparecerán las comandas que el usuario de la aplicación ha realizado en los diferentes bares de la plataforma. Allí se podrá ver el bar y el precio total de la comanda. Si pulsamos encima de cualquier comanda, nos aparecerá un “Modal” detallando los platos pedidos en la comanda que hayamos pulsado.



6. Conclusiones

Habiendo terminado las aplicaciones y tirando la vista atrás al inicio del proyecto se puede empezar a sacar un seguido de conclusiones. La primera se refiere al reto que ha presentado realizar un trabajo Full-Stack al 100% teniendo en mente a su vez plantearlo como una aplicación profesional. Se puede confirmar que el proyecto ha provocado que el grupo no sólo repase el contenido estudiado en el postgrado, sino que también haya indagado un poco más en el contenido y practicado.

En referencia a ello, cabe recalcar que se ha estudiado con más profundidad *firebase*, una herramienta que sin duda es uno de los puntos fuertes que soportan el trabajo. Tratar con una herramienta con la que no se ha trabajado demasiado anteriormente ha supuesto pequeños retos para el grupo, ya que no estaba acostumbrado a ello. Sin embargo ahora se puede afirmar que se domina bastante.

A pesar de haber desarrollado el trabajo con un grupo únicamente de dos personas, se puede decir que la organización ha sido bastante buena. Desde el principio se han marcado los puntos a tratar por cada miembro del grupo y han habido reuniones para ponerse al día y resolver dudas y problemas. Se destaca como algo positivo ya que, lo que al principio parecía un proyecto demasiado grande, se ha terminado resolviendo con eficiencia.

Finalmente se puede decir que el proyecto es un conjunto de aplicaciones para la gestión y organización de un local de restauración. Es decir, que se ha cumplido el objetivo principal planteado a inicio de proyecto.

7. Código fuente

El código fuente de ambas aplicaciones está disponible en el repositorio de GitHub creado para el proyecto:

<https://github.com/mikebcn83/barApp>