CHEATMEAL

## SERGIO RAMÓN BAYONA



Índice de contenidos

[Escribir Una Descripción 3](#_TOC_250007)

[Investigar Quienes Serán Los Usuarios 3](#_TOC_250006)

[Crear Bocetos De La App 4](#_TOC_250005)

[ESPECIFICAR ARQUITECTURA DE LA APP 7](#_TOC_250004)

[Elegir PLATAFORMA DE DESARROLLO 8](#_TOC_250003)

[Desarrollar Una Versión Cascarón 8](#_TOC_250002)

[Desarrollar Prototipo funcional de la APP 9](#_TOC_250001)

[Especificar Diagrama de tiempos aproximado 10](#_TOC_250000)

Cálculo del presupuesto aproximado de costes…………………………………………………………

CheatMeal

# Escribir Una Descripción

CheatMeal es una aplicación que nos permite pedir la comida en un restaurante sin necesidad de tener que utilizar camareros. Podrida asemejarse a un 100 montaditos, pero sin tener que ir a la barra a pedir la comida directamente desde el móvil podemos pedir, ver imágenes del producto y pagar la cuenta. Una vez preparada la comida con una notificación o sms en su defecto la aplicación te avisara que la comida esta lista para recoger en barra. También tiene una funcionalidad de poder recoger la comida en el establecimiento a una hora concreta tipo Glovoo. Esto también hace que el restaurante pueda ganar dinero no solo de la gente que está en el restaurante sino también de la gente que recoge a domicilio.

Para el usuario es una forma muy buena de poder ir a cenar sin tener contacto con un camarero ya que en tiempos difíciles como el actual COVID no tiene que relacionarse con gente externa. Y hace una experiencia más segura en el restaurante. Lo bueno también que podemos encontrar es que si cada uno quiere pedir cosas diferentes porque tiene menos dinero o tiene un máximo permitido para gastar podrá pagar directamente su cena sabiendo el precio final en todo momento.

Los resultados de esta aplicación son diversos para los usuarios una nueva forma de ir a cenar a un sitio con una nueva experiencia de ir a cenar, para el restaurante tener menos empleados aumenta las ganancias de este y un mayor servicio ya que se puede centrar en solo tener que hacer la comida.

# Investigar Quienes Serán Los Usuarios

Esta app será utilizada principalmente por dos tipos de personas, por una parte, la gente que quiera ir a nuestros restaurantes y en segundo lugar los restaurantes para poder ver los pedidos y gestionar los platos que estén disponibles para ellos.

Los procesos que cubro son la rapidez de poder pedir, la mayor atención al cuidado del resultado final al tener más tiempo y que el cliente siempre sabe que se gasta.

### Flujos de usuario (CASOS DE USO)

Caso 1 (Personas que van al restaurante):

1. El usuario va al restaurante.
2. Abre la aplicación.
3. Hace login en la app.
4. Escanea el QR.
5. Selecciona los productos que quiere tomar.
6. El cliente paga la cena.
7. Al cliente le llega una notificación que está su cena lista.
8. El cliente recoge la cena y disfruta de ella. Caso 2 (Persona que pide para recoger):
9. El usuario va al restaurante.
10. Abre la aplicación.
11. Hace login en la app.
12. El cliente selecciona el restaurante que le apetece.
13. Selecciona los productos que quiere tomar.
14. El cliente selecciona la hora de recogida del producto.
15. El cliente paga la cena.
16. Al cliente le llega una notificación que está su cena lista.
17. El cliente recoge la cena y disfruta de ella.

Caso 3 (Personal del restaurante que mira lo que tiene que preparar):

1. El usuario va al restaurante.
2. Abre la aplicación.
3. Hace login en la app.
4. Mira las peticiones de comida
5. Se encarga de hacer la comida
6. Da el de que ya está casi listo.

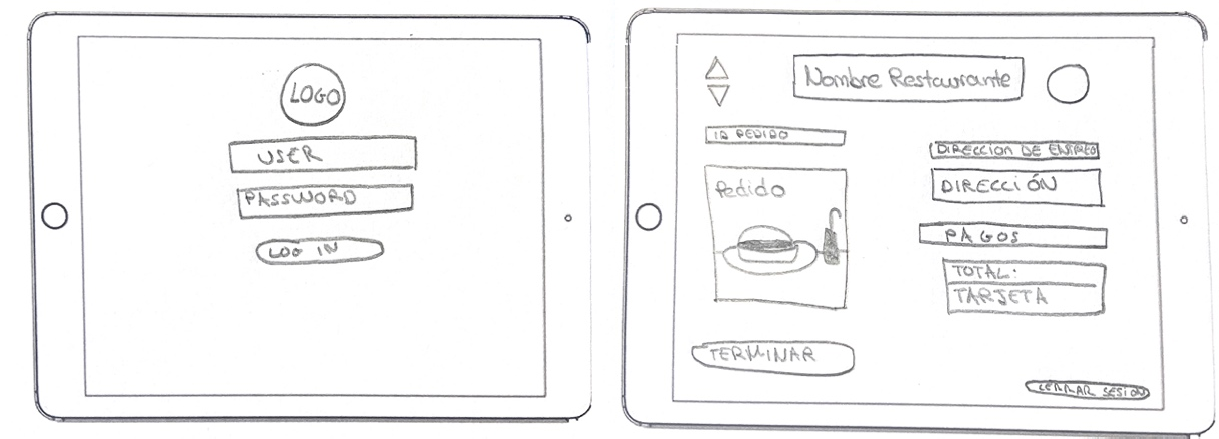


✔

1. El cocinero mira otro pedido.
2. El cocinero empieza a preparar otro pedido.
3. Vuelve al menú anterior por que ya ha terminado ese pedido.
4. Le da a finalizar.

# Crear Bocetos De La App

La App está creada para Tablet y Smartphone, empecemos por la Tablet ya que solo se encargará de reproducir los pedidos en ella:



1. En la primera imagen tenemos el Login de la aplicación. Sus **puntos de interacción** son:

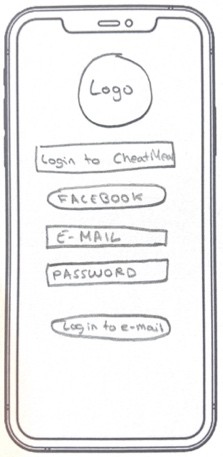
* Button que nos permite comprobar si ese user se encuentra en nuestra base de datos.

1. En la segunda imagen tenemos el pedido donde tenemos la info del pedido, la dirección de entrega, el pago, subir y bajar de pedido, finalizar pedido y cerrar sesión.

Sus **puntos de interacción** son:

* 2 Buttons de Subir y Bajar Pedido.
* 1 Button de Cerrar Sesión.
* 1 Button de estado de Pedido para enviar al usuario si está listo o no.
* 1 Button de Terminar pedido para hacer saber al cliente que está listo.

Vamos a por el Smartphone esta es una parte más complicada porque tenemos muchas más activitys. En primer lugar, el login y el register.



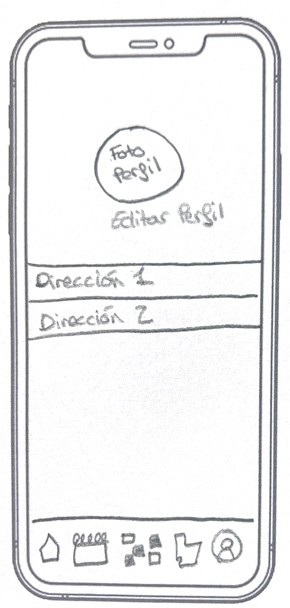
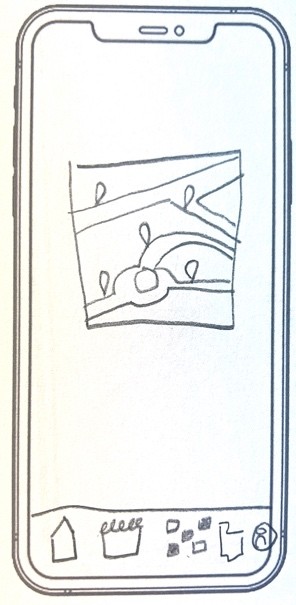
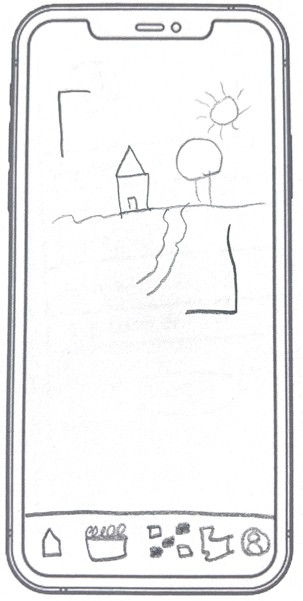
1. En la primera imagen tenemos el Login de la aplicación. Sus **puntos de interacción** son:

* Button que nos permite comprobar si ese user se encuentra en nuestra base de datos.
* Button que nos permite logearnos por Facebook.
* 2 EditText que nos permite poner el email y el password.

1. En la segunda imagen tenemos el register de la aplicación. Sus **puntos de interacción** son:

* Button que nos redirige para permitir Facebook el acceso mediante Facebook.
* Button que nos permite registrarnos.
* 2 EditText que nos permite poner el email y el password.

Ahora ya vamos al centro de la aplicación vamos a empezar con el movimiento entre la navbar inferior aquí hay 5 opciones estas son sus capturas.



1. En la primera imagen tenemos el Login de la aplicación. Sus **puntos de interacción** son:

* Los botones dependerán de la API ya que dependiendo del numero de restaurantes será lo que influya para poder pedir comida
* 5 Buttons de navegación NavBar.

1. En la segunda imagen tenemos el resumen de los pedidos tanto como métodos y estado del pedido. Esta Activity nos permite ver información entonces no tenemos interacción.

Sus **puntos de interacción** son:

* 5 Buttons de navegación NavBar.

1. En la tercera imagen solo escaneamos el código QR que nos llevará a la pagina del Restaurante Sus **puntos de interacción** son:

* 5 Buttons de navegación NavBar.

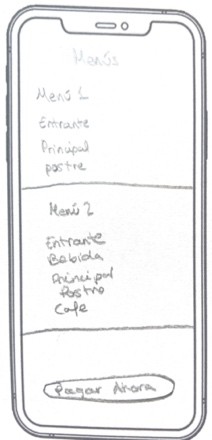
1. En la cuarta imagen veremos los restaurantes que tenemos en la zona o cerca de ti. Sus **puntos de interacción** son:

* 5 Buttons de navegación NavBar.
* 1 Mapa interactivo

1. En la quinta imagen tenemos info sobre el usuario. Sus **puntos de interacción** son:

* 5 Buttons de navegación NavBar.
* 1 Button editar user

Ahora vamos a como pedir una comida ya sea mediante el QR o clicando en un restaurante.



1. En la imagen vemos los menús/platos para pedir que solamente clicando se añadirán a la cesta y dando al botón de pagar se pagarán

Sus **puntos de interacción** son:

* 5 Buttons de navegación NavBar.
* Buttons dependiendo de los platos del Restaurante
* Button Pagar ahora que nos permite realizar el pedido.

# ESPECIFICAR ARQUITECTURA DE LA APP

Los datos de almacenarán:

* Bases de datos local propia
* Bases de datos de un servidor propio
* APIs propias.
* La tarjeta SD del dispositivo
* Sistema de preferencias de Android
* La memoria del dispositivo Sincronización de datos:
* Las operaciones van a ser varias la de crear datos, guardar datos, deletear datos, actualizar datos.

o Todos los datos se sincronizan automáticamente así que el usuario no tiene que preocuparse por eso

* Los registros serán online y no estarán puestos en cola ya que el registro es escalonado.
* Si el programa no tiene conexión a internet no funcionará ya que todo se centra en una base de datos online.
* El pronostico es que sea de una media de unos 400 por restaurante. (Eso dependerá de lo que el empresario estime).
* La sincronización es automática ya que depende del servidor y se actualizará cuando el cliente entre a la app.
* El limite de accesos dependerá de la grandeza que alcance la salida de prueba.
* En caso de cambios en el servidor no se enviará información a la app.

# Elegir PLATAFORMA DE DESARROLLO

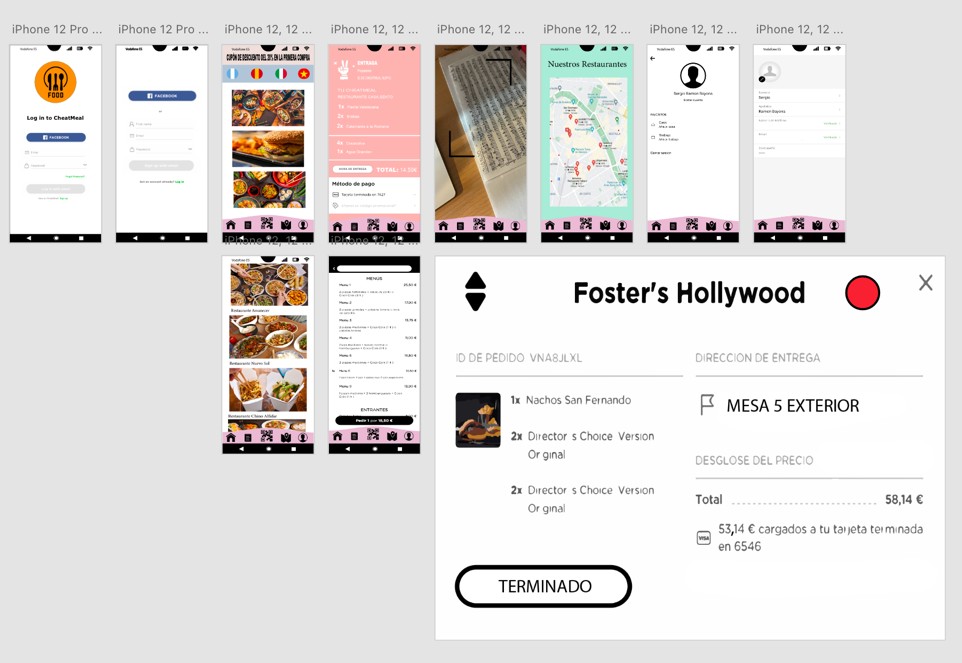
Los idiomas soportados son el español e inglés.

# Desarrollar Una Versión Cascarón

Crear y configurar un proyecto Android

* Añadir librerías
* Definir estructura de paquetes java
* Supervisar los procesos de compilación Iniciar un punto de versionamiento con GitHub
* Crear cuenta GitHub
* Crear repositorio
* Control de versiones y texto plano
* Gestionar Conflictos Implementar Login
* Crear Actividad de Login
* Crear fragmento login
* Definir parámetros del usuario Implementar forma de pago
* Insertar Plugin de pago
* Definir formas de pago Implementar forma de escanear QR
* Crear Actividad QR
* Crear fragmento QR
* Definir los parámetros QR Implementación de QR
* Crar Actividad Mapa
* Crear Fragmento de Mapa
* Definir Parametros de Mapa

# Desarrollar Prototipo funcional de la APP

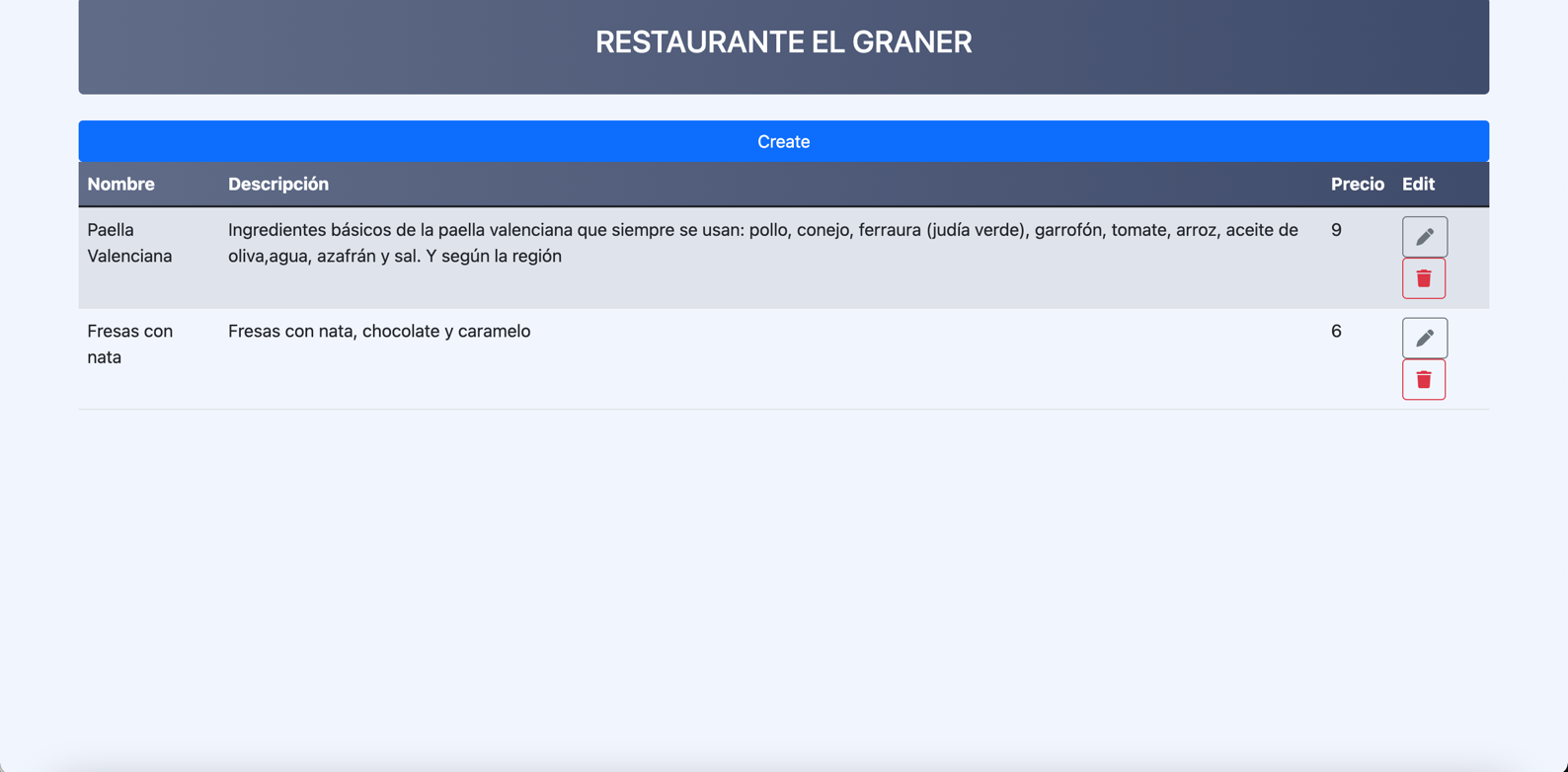


Enlace a Adobe XD: https://drive.google.com/file/d/1eGHz6GgEwb9Q5jErZn59JR\_veeVgHetk/view?usp=sharing

Actualización de la APP

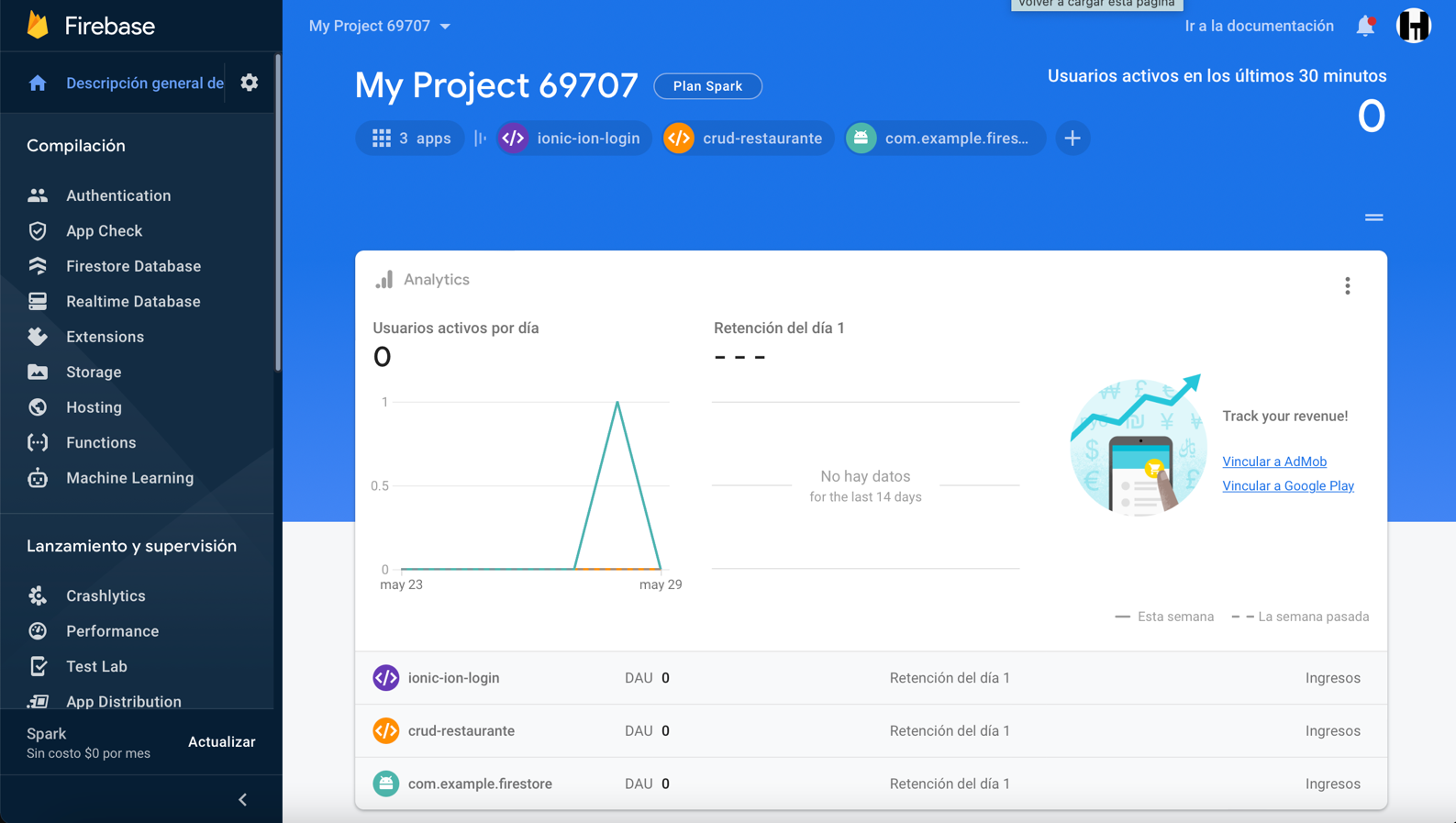
La app se ha basado en 3 aplicaciones un backEnd de control de carta y frontEnd ionic en los cuales el cliente Usuario y cliente Empresa aquí las actualizaciones.

BackEnd ->

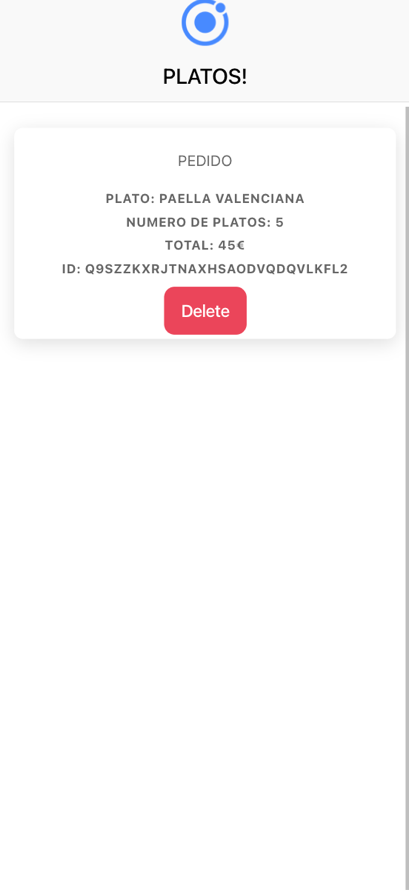
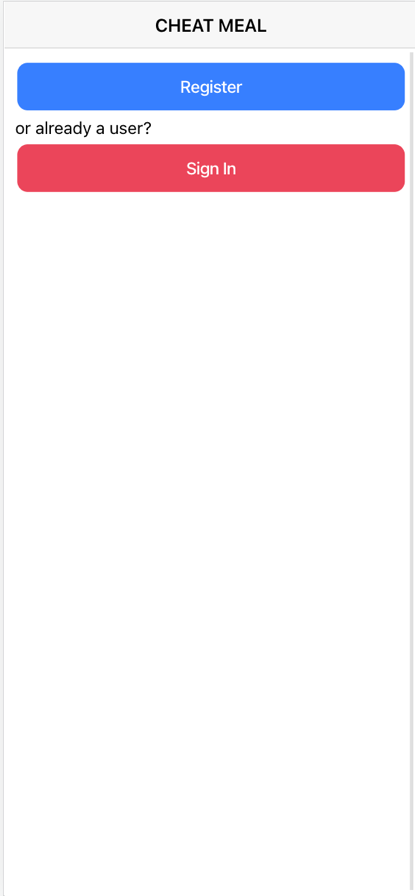
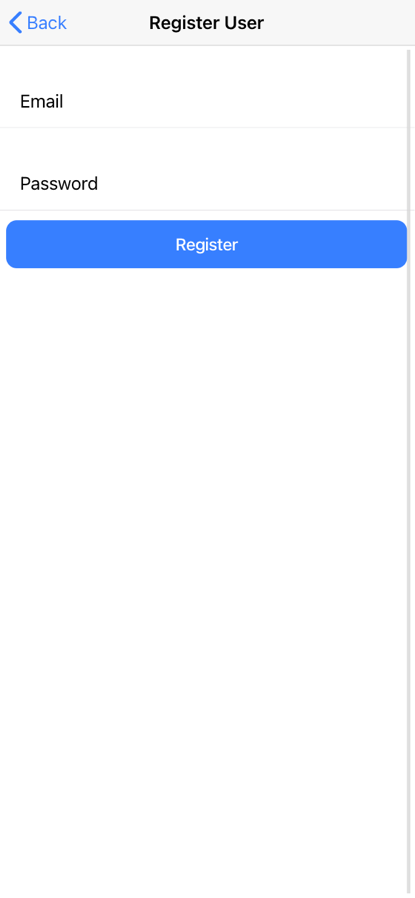
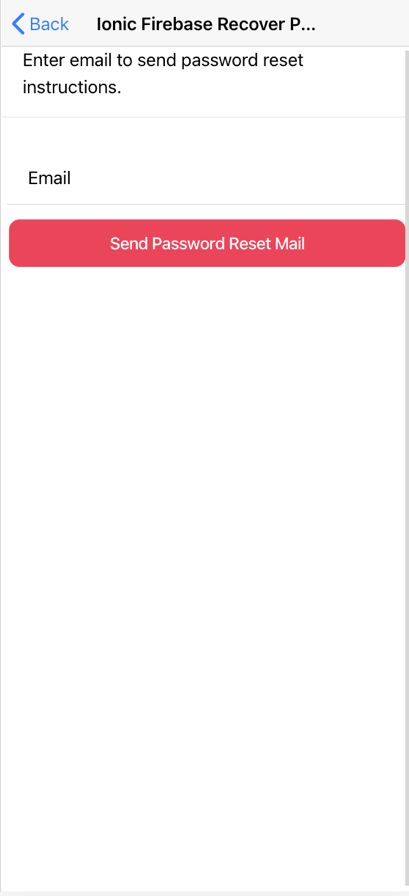


Aquí podemos en contrar la típica api rest que edita crea y elimina datos en y el Empresario poder Actualizar la carta del restaurante.

Base de Datos utilizada ->

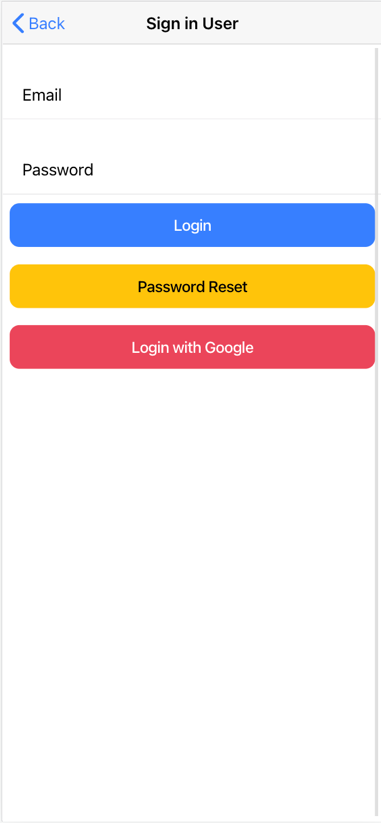
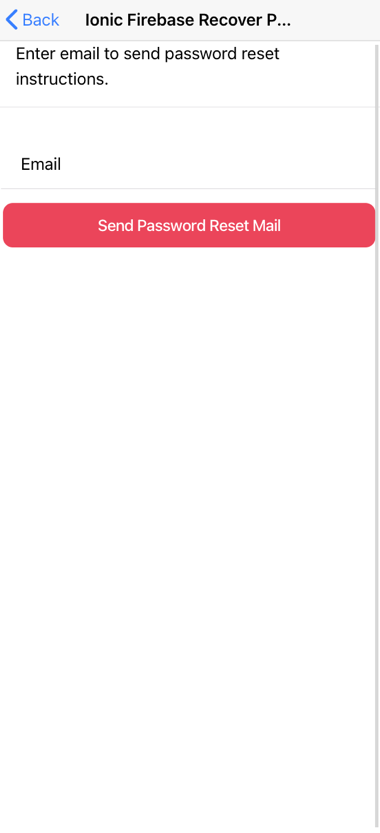
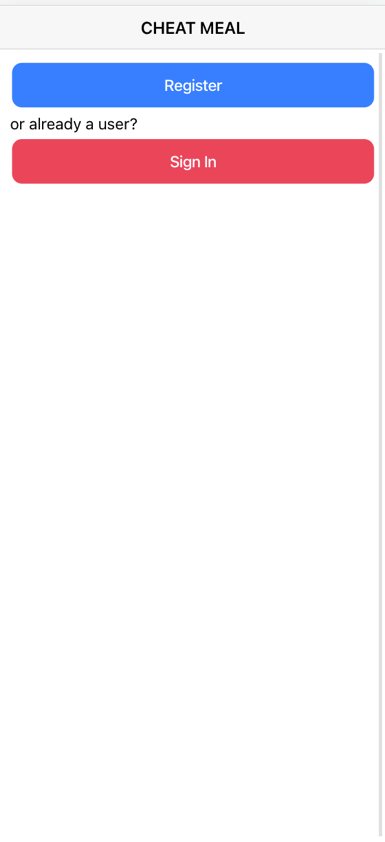


He elegido esta BD por la sincronización entre diferentes tipos de código. -> Firestore ha sido el elegido.

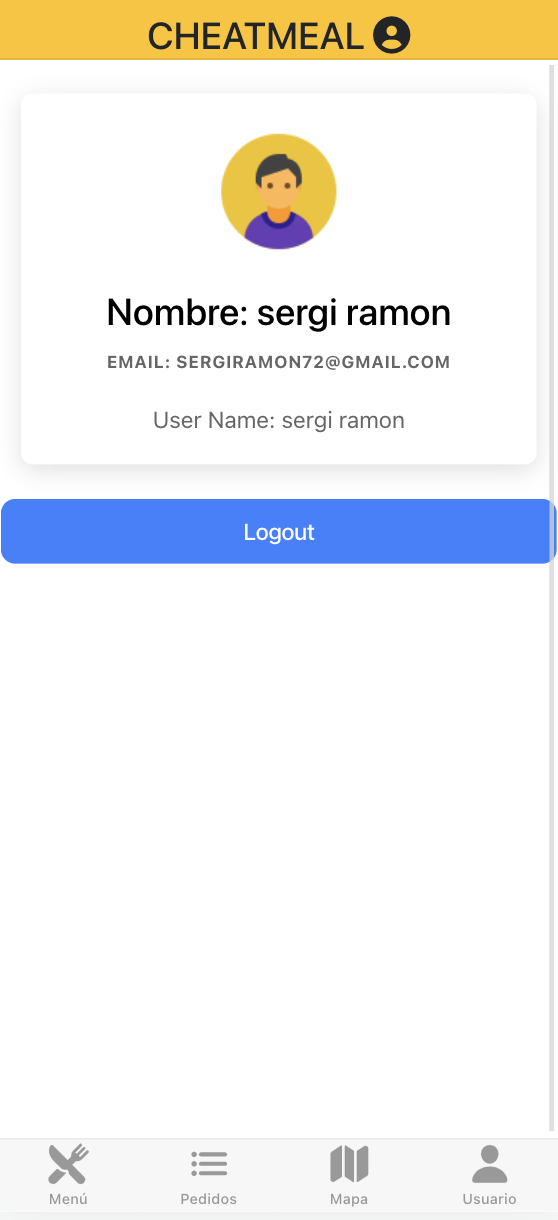
Front-End-Cliente-Empresa:

Tenemos un login para saber que usuario está trabajando. Y un botón eliminar para poder quitar el plato para que no haya quivocaciones.

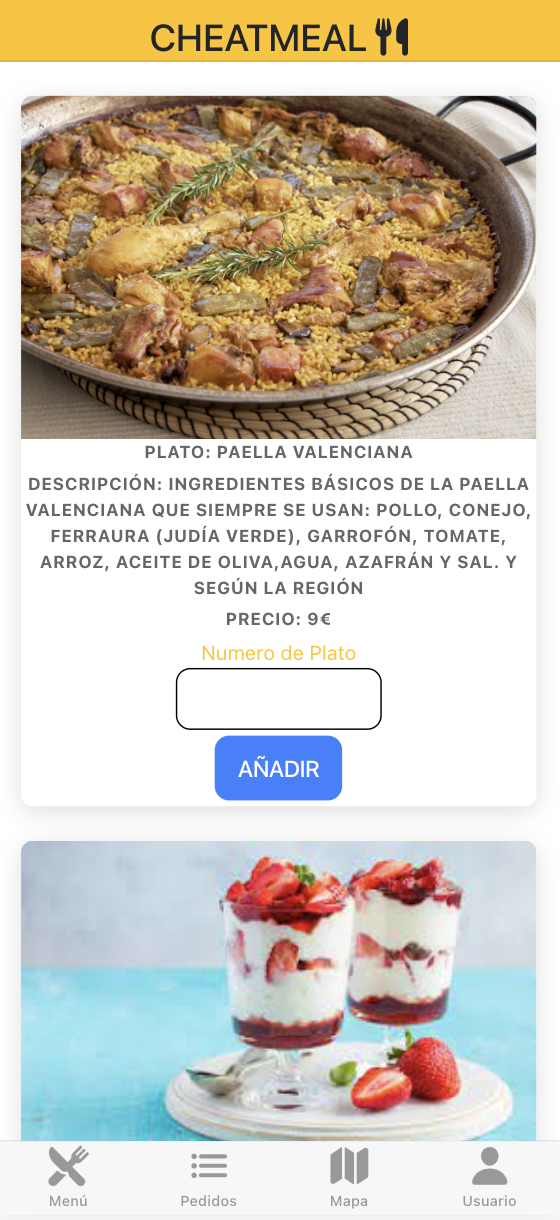
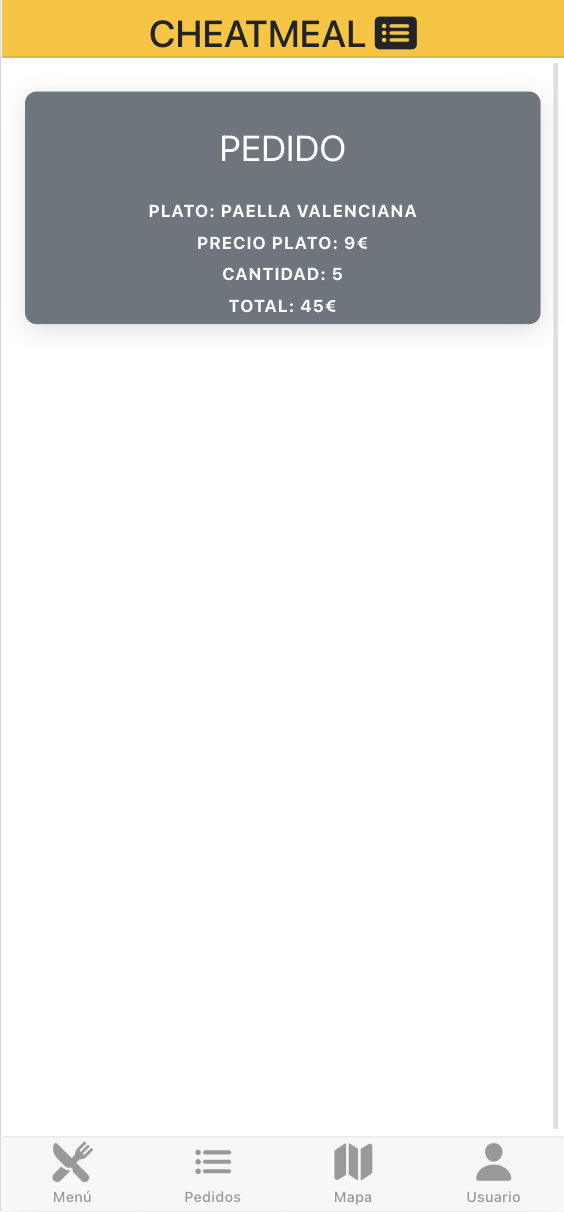
FrontEnd-Ionic-Ciente ->



Aquí tenemos un login muy normal en el cual nos registramos, nos logueamos y reseteamos pasword.

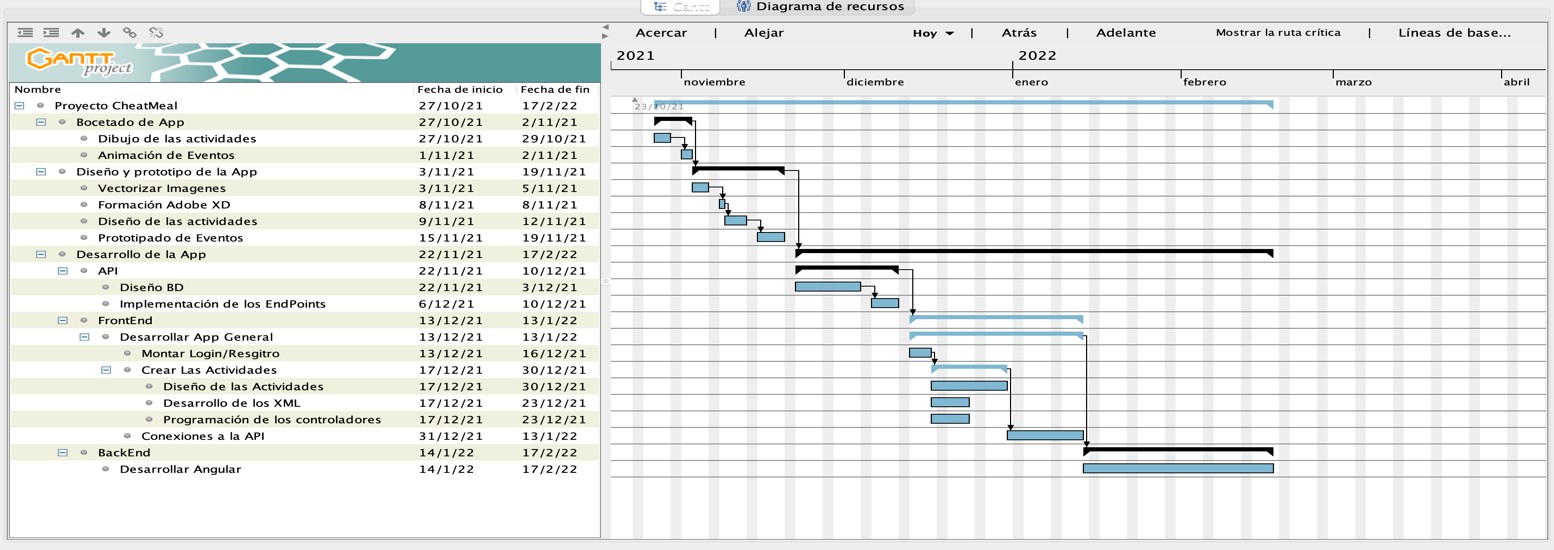


Estas son las dos pantallas en las cuales podemos ver la localización del restaurante y el des loguears de useruario así como los datos estas pestañas son para dar experiencia de usuario.



Esta es la zona que es la esencia del programa tenemos un formulario que envía. Tenemos un formulario en el cual añadimos los productos que queremos y luego un botón que envia a la base de datos. En la segunda page tenemos unas cards que nos muestra todo lo que han pagado.

# Especificar Diagrama de tiempos aproximado

Diagrama de Gant:

Enlace al gant: https://drive.google.com/file/d/1cmyT97q6bFUx5f0Mtp7Cvq65PtIJ73ng/view?usp=sharing

# PRESUPUESTO:

MARGENDE BENEDICIOS

DIRECTOS

SUELDO

ESPACIO

PRESUPUESTO

EQUIPOS

INDIRECTOS

REUNIONES CLIENTE

GASTOS VARIOS

LICENCIAS

PROGRAMADOR JUNIOR -> Costo en bruto 1900€/Brutos costo en neto para el trabajador 1200€. Coste trimestral ->1900€ x 2,5 meses = 3850€ al trimestre ya que este proyecto solo tiene un trabajador.

Espacio de Trabajo -> 667€ costo estimado del espacio de trabajo x 2,5 meses = 1667,5€ Equipos -> MacBook Pro 14 = 2000€ /2.5 meses = 800€

Internet -> 100€ mes x 2.5 meses = 250€

Total = 6567,5€ a falta de añadir gastos varios, reuniones y licencias + la ganancia empresarial. Reuniones -> 6567,5€ x 0,1 = 656,75€

Gastos Varios (Incluye licencias) -> 6567,5€ x 0,25 = 1641,88€ 2298,63€ Total gastos indirectos.

**TOTAL:**

