



Selfo – Self Ordering App

Nombre Estudiante: Luis Hong Wu Wang
Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: Pau Dominkovics Coll
Profesor/a responsable de la asignatura: Carles Garrigues Olivella

Fecha de Entrega



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

© Luis Hong Wu Wang

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Selfo</i>
Nombre del autor:	<i>Luis Hong Wu Wang</i>
Nombre del consultor/a:	<i>Pau Dominkovics Coll</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2019
Titulación::	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Máximo 3 palabras clave, validadas por el director del trabajo (dadas por los estudiantes o en base a listados, tesauros, etc.)</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Selfo es una plataforma con la finalidad de mejorar la experiencia de un cliente a la hora de hacer su pedido en un restaurante. Incluye una app móvil multiplataforma que el usuario podrá utilizar en los restaurantes y tomar la nota del pedido de manera totalmente independiente sin necesidad de interactuar con un camarero.</p> <p>Esta app actúa como un camarero electrónico, donde el cliente una vez dentro del restaurante podrá ver toda la carta del menú mediante el simple hecho de un escaneo de código QR. En todo momento el cliente podrá ver el estado de la comanda e incluso pedir la cuenta y pagar mediante la app al finalizar el servicio.</p> <p>Selfo también ayudará al restaurante a mejorar el volumen de negocio y la gestión de éste. Gracias a la plataforma, ofrecerá un servicio de más calidad que impactará directamente a la satisfacción del cliente.</p> <p>Para realizar esta plataforma se desarrollará una aplicación con el framework multiplataforma "Ionic" integrado con servidor y base de datos en "Firebase". Teniendo como resultado final una aplicación funcional para Android, iOS y Web.</p> <p>Con este proyecto se espera aprender e iniciarse en el mundo de las aplicaciones y en todo el proceso que conlleva en obtener el producto final.</p>	

Abstract (in English, 250 words or less):

Selfo is a platform with the purpose of improving the experience of a customer when placing the order in a restaurant. It includes a multiplatform mobile app that the user can use in restaurants and take the note of the order completely independent without the interaction with a waiter or waitress.

This app acts as an electronic waiter, where the customer once inside the restaurant can see the entire menu by the simple act of scanning a QR code. At any time, the customer would be able to see the status of the command and even ask for the bill and pay through the app at the end of the service.

Selfo will also help the restaurant improve its volume of business and management. Thanks to the platform, it will offer a higher quality service that will directly affect to customer satisfaction.

To build this platform, an application will be developed with the "Ionic" multiplatform framework integrated with a server and database in "Firebase". Having as result a functional application for Android, iOS and Web platforms.

With this project, It is hoped to learn and to get initiated in the world of applications and in the whole process that involves in obtaining the final product.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo	3
1.5 Breve resumen de productos obtenidos.....	4
1.6 Breve descripción de los otros capítulos de la memoria	4
2. Análisis: Usuarios y contexto de uso.....	5
2.1 Metodología en los análisis de usuario	5
2.2 Ficha de los perfiles seleccionados.....	6
2.3 Necesidades de los usuarios	8
2.4 Benchmarking	8
3. Diseño conceptual.....	9
3.1 Escenarios de uso.....	9
3.2 Flujos de interacción	10
3.3 Prototipo de baja fidelidad.....	13
3.4 Prototipo alta fidelidad.....	18
4. Evaluación.....	23
4.1 Casos de uso	24
4.2 Diseño de la arquitectura	30
5. Desarrollo.....	33
5.1 Herramientas para el desarrollo.....	33
5.2 Estructura del proyecto	35
5.3 Pruebas.....	36
5.4 Estado actual del desarrollo.....	37
6. Conclusiones.....	38
7. Glosario.....	39
8. Bibliografía	40
7. Anexos	41

Lista de figuras

Fig. 1. Planificación en fechas.....	3
Fig. 2. Diagrama de Gantt	4
Fig. 3. Diagrama de clases.....	28
Fig. 4. Diagrama de bases de datos	29
Fig. 5. Diagrama de arquitectura.....	30

1. Introducción

1.1 Contexto y justificación del Trabajo

Para el trabajo final de máster de Desarrollo de Aplicaciones Móviles, se intentará desarrollar una idea que tenía mucho tiempo en mente. La idea principal es ayudar a mejorar la experiencia de un comensal cuando va a un restaurante.

Hoy en día estamos inmersos en la era digital en que prácticamente se utiliza la tecnología en cualquier ámbito con el fin de mejorar el servicio que se ofrece y obtener un valor añadido que lo diferencie del resto.

Intentaré juntar los conocimientos sobre el negocio de la hostelería y la experiencia en el desarrollo de software con el fin de crear una aplicación para ayudar tanto al cliente como al hostelero.

Hoy en día existen infinitas aplicaciones relacionadas con el mundo de la restauración, alguna de ellas realmente muy logradas y que han tenido una penetración y aceptación en la sociedad impresionante. Esto quiere decir, crear una aplicación parecida para hacer competencia directa a estas aplicaciones ya consolidadas sería una tarea muy complicada y el porcentaje de éxito muy reducido.

Dicho esto, la idea que se plantea para este proyecto es un servicio complementario para todas estas aplicaciones. Un camarero electrónico, dónde el propio cliente, una vez sentado en el restaurante, podrá tomar nota de lo que desea pedir a través de su terminal o dispositivo. La ventaja de esta aplicación es tener total libertad para pedir y no tener que esperar a que un camarero venga a tomarte nota. Dentro de esta aplicación el usuario también podrá pedir la cuenta e incluso pagar a través de las pasarelas de pago integradas.

Las ventajas de esta aplicación es aprovechar la penetración de los dispositivos móviles y utilizarlo para que el comensal de un restaurante tenga más libertad a la hora de obtener un servicio. El cliente, tendrá un trato mínimo con los camareros del establecimiento y podrá en cualquier momento pedir lo que crea conveniente y pagar la cuenta a su antojo.

1.2 Objetivos del Trabajo

El objetivo principal del trabajo es realizar una aplicación sencilla y muy funcional apto para todos los públicos. Los requisitos funcionales de esta aplicación son:

- Poder ver la carta detallada del restaurante.
- Tomar nota de todo lo que se quiera pedir en el restaurante.
- Tener un resumen de lo que se ha pedido antes de aceptar la comanda.
- Poder añadir o cualquier cosa en todo momento.
- Llamar al camarero.
- Pedir la cuenta.
- Pagar la cuenta.

En cambio, los requisitos no funcionales son:

- La aplicación será multiplataforma, lo que quiere decir que se podrá utilizar tanto en dispositivos iOS como Android incluso se podrá acceder mediante navegador web. Siendo esta parte la menos utilizada por los comensales.
- Diseño sencillo y minimalista. Esto implica que la curva de aprendizaje para utilizar la aplicación es reducida.
- Plataforma, quiere decir que cualquier restaurante puede adherirse para que sus clientes puedan utilizarlo. El objetivo es ofrecer un servicio que beneficie a ambas partes implicadas.

1.3 Enfoque y método seguido

El enfoque de este proyecto es totalmente nuevo, es un producto que muchos restaurantes ya tienen implementado sobre todo los de comida rápida. Sin embargo, en territorio español existen pocas plataformas que ofrezca un servicio personalizado para a medida para cada cliente.

Al tratarse de una plataforma, el método beneficia a ambas partes mejorando negocio y servicio ofrecido. Es un método bastante extendido ya que existes varias plataformas en el mundo de la hostelería.

Esta nueva app, dará soporte a muchas otras apps del mundo de la hostelería que te permitía buscar restaurantes, ver valoraciones, opiniones, reservar e incluso ver fotos del sitio. Como en las otras plataformas el establecimiento o restaurante deberá adherirse a nuestra plataforma para que los usuarios puedan disfrutar del servicio.

Las ventajas para el comensal o cliente están bastante bien definidas. Sin embargo, para el restaurante también tiene muchas ventajas. Por ejemplo, ayudará a disminuir la carga de trabajo del camarero y así podrá ofrecer un mejor servicio, se tendrá control a tiempo real de todos los pedidos o mesas del restaurante, incluso ver qué platos faltan por servir a cada mesa.

Para desarrollar el proyecto se utilizará el framework de desarrollo multiplataforma llamado ionic. Usando tecnologías de HTML5, CSS y JavaScript. Ionic te da facilidades en crear aplicaciones para todo tipo de dispositivos de manera rápida con una interfaz amigable y sobretodo es de código abierto. Para la parte servidor y base de datos, estará hospedada en Firebase, ya que permite aplicaciones multiplataforma con librerías adaptadas para todas las plataformas y ofrece características de autenticación e informe de fallos que irán muy bien para ahorrar tiempo en el desarrollo.

1.4 Planificación del Trabajo

El trabajo a realizar será siguiendo las pautas de las prácticas de evaluación continua (PEC). Cada PEC tiene unas tareas a realizar y se centra en una característica importante del proyecto.

- PEC 1: Plan de trabajo.
- PEC 2: Diseño de la app.
- PEC 3: Desarrollo e implementación.
- Entrega final.

Para realizar el trabajo se ha calculado un único recurso y que va a trabajar 4h por jornada. Lo que no significa es que se vayan a hacer en ese día en concreto. Se va a aprovechar días de fin de semana.

1	<input type="checkbox"/> PEC 1 - Plan de trabajo	16 days	19/09/18 8:00	10/10/18 17:00
2	Planteamiento de la idea	3 days	19/09/18 8:00	21/09/18 17:00
3	<input type="checkbox"/> Contexto y justificación	2 days	21/09/18 8:00	24/09/18 17:00
4	<input type="checkbox"/> Definir los objetivos del trabajo	1 day	25/09/18 8:00	25/09/18 17:00
5	<input type="checkbox"/> Enfoque y método seguido	1 day	26/09/18 8:00	26/09/18 17:00
6	<input type="checkbox"/> Planificación del trabajo	3 days	27/09/18 8:00	1/10/18 17:00
7	<input type="checkbox"/> Sumario de productos obtenidos	3 days	3/10/18 8:00	5/10/18 17:00
8	<input type="checkbox"/> Descripción de otros capítulos	3 days	8/10/18 8:00	10/10/18 17:00
9	<input type="checkbox"/> PEC 2 - Diseño	15 days	11/10/18 8:00	31/10/18 17:00
10	<input type="checkbox"/> Usuarios y contexto de uso	3 days	11/10/18 8:00	15/10/18 17:00
11	<input type="checkbox"/> Diseño conceptual	5 days	15/10/18 8:00	19/10/18 17:00
12	<input type="checkbox"/> Prototipado	5 days	22/10/18 8:00	26/10/18 17:00
13	<input type="checkbox"/> Evaluación	3 days	29/10/18 8:00	31/10/18 17:00
14	<input type="checkbox"/> PEC 3 - Implementación	29 days?	1/11/18 9:00	12/12/18 9:00
15	<input type="checkbox"/> Montar arquitectura para desarrollo	1 day	1/11/18 9:00	2/11/18 9:00
16	<input type="checkbox"/> Desarrollo	20 days	5/11/18 8:00	30/11/18 17:00
17	<input type="checkbox"/> Revisión y testeo	3 days	5/12/18 8:00	7/12/18 17:00
18	<input type="checkbox"/> Documentación de la PEC	3 days?	3/12/18 8:00	5/12/18 17:00
19	<input type="checkbox"/> Creación de paquete de la solución	2 days	10/12/18 9:00	12/12/18 9:00
20	<input type="checkbox"/> Entrega Final	14 days	13/12/18 9:00	2/01/19 9:00
21	<input type="checkbox"/> Corrección de errores en la APP	11 days	17/12/18 9:00	1/01/19 9:00
22	<input type="checkbox"/> Finalización de la memoria	12 days	13/12/18 9:00	31/12/18 9:00
23	<input type="checkbox"/> Preparar presentación	2 days	31/12/18 9:00	2/01/19 9:00

Figura 1: Planificación en fechas

El trabajo planificado es de unas 300 horas para todo el proyecto dedicando 20 horas por semana. En caso de que en cada hito se requiera más tiempo de lo

previsto, hay un plan de contingencia en invertir horas no previstas. Para cada periodo hay un margen del 10% de horas de margen por hito.

El mayor riesgo de retraso de tareas es en la fase de implementación, dónde se va a desarrollar la aplicación. En caso de que haya imprevistos en esta fase, se hará un sobreesfuerzo de horas alargando las jornadas y dedicando más tiempo en días festivos si fuera necesario.

Aquí tenemos el diagrama de Gantt:

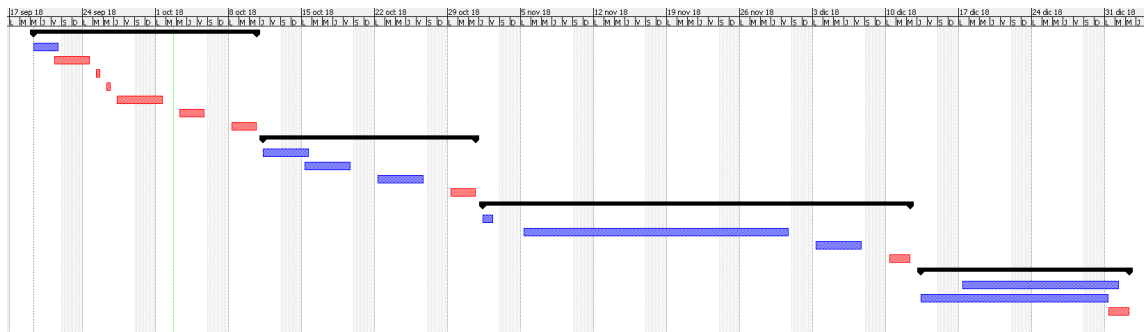


Figura 2: Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

Al final del trabajo se obtendrá una aplicación funcional con las características descritas anteriormente. Además, siendo multiplataforma se podrá acceder desde cualquier dispositivo móvil.

Habrán varios manuales dependiendo del rol que se tenga (si es cliente o hostelero) para ayudar a iniciarse en la aplicación.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos, se entrará en detalle la parte de diseño de la aplicación, la arquitectura, el modelo de negocio y para acabar se explicará toda la fase de desarrollo.

En la parte del diseño de la aplicación, se conocerá los requisitos y necesidades de los usuarios para así focalizar el trabajo en esas funcionalidades más necesarias. A partir de eso se preparará un diseño conceptual para definir los escenarios. Para finalizar con la parte de diseño se elaborará un prototipo de la aplicación y así poder definir los posibles casos de uso de manera adecuada.

El modelo de negocio es una de las partes importantes a la hora de iniciar un proyecto de este calibre. Es algo que se debe detallar para ver la viabilidad de la idea.

Por último, habrá todo un capítulo para la parte más técnica del trabajo. Se explicará la arquitectura escogida y se detallará todo el proceso de desarrollo e implementación.

2. Análisis: Usuarios y contexto de uso

Selfo, es una plataforma que se centra en un ámbito muy en concreto. Gracias a eso se puede definir muy claramente a los actores que tienen un impacto directo con la plataforma.

Una de las grandes ventajas que conlleva el proceso de definir las características de los usuarios, sus necesidades y objetivos es mi gran conocimiento del negocio ya que mi ámbito familiar siempre se ha dedicado a ello.

Una parte importante es definir claramente las necesidades del usuario tanto al usuario final que utilizará la aplicación como comensal como al dueño del establecimiento que se quiera adherir a la plataforma para mejorar el volumen de su negocio o mejorar los servicios que se le ofrecen.

2.1 Metodología en los análisis de usuario

Para detectar la aceptación y averiguar el segmento de personas que utilizarían este servicio, se ha creado un pequeño formulario o encuesta que he realizado a varios clientes de un restaurante familiar.

Preguntas:

¿Sueles usar apps a la hora de decidir el sitio dónde ir a comer?

¿Utilizarías un dispositivo móvil para tomar nota?

¿Qué crees que aportaría un servicio así?

¿Pagarías la cuenta con tu dispositivo?

Los resultados fueron más que satisfactorios y se puede segmentar a los posibles usuarios por edades:

Menos de 50 años: tienen más asimilado el uso de las nuevas tecnologías en su día a día. De 8 entrevistados, 7 ya utilizan diversas aplicaciones relacionadas con el mundo de la hostelería.

Todos ellos, han contestado que un dispositivo móvil personal para tomar nota es algo que podrían utilizar y que les parecería una innovación para restaurantes que no son de comida rápida. Sin embargo, a dos encuestados han comentado que a veces la interacción con el camarero es más eficiente para preguntar y obtener recomendaciones. 5 de 8 usuarios no tendrían problemas con pagar la cuenta desde la aplicación ya que están acostumbrados a las compras online. El resto prefiere pagar al a vieja usanza.

Mayores de 50 años: no tienen tan asimilado las nuevas tecnologías y son más tradicionales. Los dos únicos encuestados no utilizan el dispositivo móvil para decidir dónde ir a comer. Prefieren un servicio con camarero físico, aunque creen que la idea podría mejorar el servicio para otras personas.

Otro aspecto muy importante es identificar las necesidades del dueño del restaurante y el valor que le puede aportar para adherirse a la plataforma como restaurante colaborador.

Para recoger sus necesidades a partir de la idea el proyecto se ha utilizado un método de entrevista a dos de los pocos dueños que han accedido a entrevistarse.

Las preguntas que se han formulado han sido:

¿Está tu restaurante adherido a alguna plataforma o app?

¿Qué te aparece la idea de camarero virtual auto gestionado?

¿Estarías dispuesto a utilizar Selfo?

¿Qué crees que aportaría este servicio?

En ambos casos, sus establecimientos no utilizan ninguna plataforma para promocionarse, pero también lo han valorado a corto-medio plazo. Pero por el tema económico no se han decidido aún.

La idea Selfo, de tomar la nota a través del propio dispositivo del cliente, creen que es una buena idea que puede ayudar a liberar la carga de trabajo de los camareros y evitar posibles errores humanos con lo que conlleva. Sin embargo, no creen que al implementar esta plataforma se tenga que prescindir de trabajadores para ello ya que al final se requiere un trato humano.

Entendiendo que es un negocio, lo podrían utilizar si económicamente es rentable y les aporta un crecimiento significativo del negocio.

2.2 Ficha de los perfiles seleccionados

Para entender mejor las necesidades de los usuarios y del contexto de uso, vamos a seleccionar y crear una pequeña ficha de cada uno de los dos perfiles identificados en el apartado anterior.

Usuario 1

Nombre: Ángel

Edad: 32

Estudios: Licenciado

Profesión: Ingeniero

Descripción de la persona: Ángel está casado y tiene un hijo de 1 año. Vive en Esplugues de Llobregat, Barcelona. Trabaja como ingeniero en una consultora tecnológica. En casa están acostumbrado a las últimas tecnologías y disponen de 2 ordenadores portátiles, 1 ordenador de sobremesa, 2 dispositivos tableta y un teléfono Smartphone por persona adulta.

El poco tiempo libre que encuentra, lo intenta disfrutar con su familia y salir a pasar una buena velada conociendo nuevos restaurantes de la ciudad. También suele repetir en sus restaurantes preferidos. Utiliza mucho las aplicaciones para encontrar nuevos sitios y se fija bastante de los posibles servicios que ofrecen en casa establecimiento.

Descripción del escenario: llega el fin de semana y Ángel quiere pasar una buena mañana con su familia y salen a comer fuera. A parte de la buena

comida le interesa que haya un servicio eficiente para no perder mucho el tiempo.

Conocen un restaurante que está adherida a Selfo y deciden probarlo. Ángel se descarga la aplicación para su dispositivo móvil y se registra como cliente. Cuando llegan al restaurante, el camarero los sienta en una mesa y les pregunta si desean la carta o si conocen la aplicación Selfo. Ángel decide que esta vez usará Selfo, el servicio de camarero electrónico ya que ve que el restaurante está muy lleno.

Con su propio dispositivo, Ángel puede ver la carta del menú y van eligiendo los platos que quieren pedir. Finaliza la comanda sabiendo en todo momento el precio final del pedido. De manera automática, el restaurante sabe el pedido de Ángel y empiezan a servirle. Al acabar, Ángel puede elegir dos métodos de pago, directamente a través de la app o que un camarero le traiga la cuenta. En este caso, Ángel paga a través de la app y se marcha satisfecho del servicio.

Usuario 2

Nombre: Miguel

Edad: 49

Estudios: Secundaria

Profesión: Autónomo

Descripción de la persona: Miguel es un padre de familia que tiene dos hijas de 20 y 14 años. Desde hace más de 15 años regenta un restaurante de comida mediterránea en Cornellà de Llobregat, Barcelona con capacidad de 50 comensales y tiene contratado a 6 empleados que se reparten en varios turnos. No está acostumbrado a las últimas tecnologías, pero sabe de su existencia y que esto puede mejorar el servicio de su negocio e incluso darse a conocer en la red. Dispone de teléfono Smartphone, pero no utiliza muchas apps, solo para comunicarse y leer noticias.

Descripción del escenario: En algunos momentos su restaurante se llena de golpe sobretodo en la hora de la comida. Los camareros no pueden ofrecer la misma calidad de servicio en ese pico de trabajo. Selfo, puede ayudar a minimizar un poco la carga de trabajo ya que el tiempo en el que le dedica un camarero a tomar nota puede estar sirviendo otras mesas.

Miguel, puede ver qué pedidos hay activos que se han hecho por la app y ayuda a coordinar los camareros con la cocina.

2.3 Necesidades de los usuarios

Después de hablar, realizar encuestas y entrevistas con ambas partes de los posibles usuarios tenemos que diferenciar las necesidades de ambas partes.

Para los usuarios que son cliente de un restaurante, podemos identificar sus necesidades por:

- Autonomía a la hora de realizar el pedido. El usuario se siente sin presión a la hora realizar el pedido.
- Evitar esperas en realizar el pedido o a la hora de pagar.
- Minimizar la posibilidad de cometer un error a la hora de hacer el pedido.

En cambio, para el dueño que quiera ofrecer el servicio a sus clientes tiene otras necesidades.

- Mejorar el servicio a sus clientes.
- Disminuir la carga de trabajo a sus empleados.
- Aumentar le volumen de negocio.

2.4 Benchmarking

Una vez que se ha identificado las necesidades, se ha hecho una búsqueda de aplicaciones o plataformas que cubren dichas necesidades que al fin y al cabo será la competencia.

Actualmente en el mercado español no existe ninguna aplicación o plataforma que haga lo mismo. Muchos establecimientos tienen su propia solución de camarero electrónico, pero lo controlan los propios camareros del establecimiento. Sin embargo, hay muchos sitios de comida rápida que ya ofrecen un servicio con quioscos electrónicos para tomar pedidos. Estos sitios cuando hay mucha afluencia de gente, se siguen formado colas y al final es lo que se quiere evitar, ya que muchos clientes sienten la presión de hacer el pedido lo más rápido posible.

En cambio, en el panorama europeo sí que existen aplicaciones parecidas. Las dos más importantes son:

1. Menu.app. Empresa suiza que ofrece soluciones tecnológicas para restaurantes y hoteles. Se centra en el mercado suizo y alemán.
 - Fortalezas: Empresa bastante asentada, muchos restaurantes utilizan su plataforma.
 - Debilidades: Solo se encuentra en Suiza y Alemania.
 - Oportunidades: Puede extenderse al resto de países europeos.
 - Amenazas: Nuevos competidores ya que aún existe mucho margen de negocio que mejorar.

2. MyMenu.info. Empresa inglesa que ofrece soluciones tecnológicas para restaurantes solamente. Actualmente, se cree que ya no están operativos ya que su página web hace más de 1 año que no se actualiza. Pasa lo mismo con la app en las plataformas de iOS y Android no han actualizado desde hace tiempo.

La conclusión que se puede sacar de este benchmarking, es que es un servicio que actualmente solo está implementado en los establecimientos de comida rápida. Dado la poca información que se ha encontrado sobre aplicaciones parecidas, es un buen indicador que hay un nicho de mercado que está sin explotar y que está esperando a la aplicación o plataforma correcta para penetrar en el mercado hostelero.

3. Diseño conceptual

Para realizar el diseño conceptual, se deben de partir de escenarios de uso detallado para luego poder definir el flujo que se requiere de la aplicación. Hay que dividir los escenarios en dos partes, como cliente o comensal y como dueño del establecimiento.

3.1 Escenarios de uso

Primer escenario

Perfil de usuario: Usuario 1 (Ángel).

Objetivos: Pasar una buena velada para comer con su familia.

El usuario se encuentra con su familia en una mañana de sábado y deciden pasar una buena mañana comiendo fuera de casa sin estrés y sin tener que esperar largas colas. Se ha enterado de un nuevo servicio que ofrece uno de sus restaurantes preferidos de la zona de donde viven, Selfo. Sabiendo que, gracias a esta app, mejora el servicio que ofrece el restaurante, decide bajársela y registrarse gratuitamente.

Las largas esperas para que le atendieran, era uno de los factores por que casi nunca iba a ese restaurante en fin de semana, ya que siempre había mucha gente. Deciden darle una oportunidad al restaurante que tanto le gusta sabiendo que los fines de semanas siempre hay mucho volumen de trabajo.

Al llegar al restaurante, ven que efectivamente hay mucha clientela. Sin embargo, un camarero sienta a Ángel y a su familia en una mesa. Ángel comunica al camarero que tiene la app de Selfo y no requiere la carta del menú. Entonces, Ángel saca su dispositivo Smartphone y a través de la APP de Selfo, escanea un código QR pegado a la mesa para iniciar el proceso de hacer el pedido. Automáticamente se le carga el menú del restaurante y Ángel y su familia, van escogiendo lo que quieren comer y beber y sus respectivas cantidades. Cuando ya tienen todo elegido, confirman el pedido ya pueden esperar a que les traigan el pedido.

Mientras disfrutan de su comida, Ángel se ha quedado sin agua por lo que quiere añadir otra a su pedido. Coge el teléfono y en la app de Selfo añade un agua a su pedido y automáticamente un camarero le servirá el agua.

Al finalizar toda la comida, paga la cuenta a través de la app y recogen sus cosas y se marchan. La impresión de Ángel fue muy positiva, no ha tenido que esperar para que le tomen la comanda, ni para pedir un agua a mitad de la comida ni tampoco al pedir la cuenta y pagar.

Segundo escenario

Perfil de usuario: Usuario 2 (Miguel)

Objetivos: Ofrecer un mejor servicio a sus clientes.

Miguel se encuentra en su restaurante y sabe que como cada sábado por la mañana la gente acude a su establecimiento para pasar una buena velada disfrutando de la comida. Para Miguel, es muy importante que los clientes queden satisfechos con el servicio que ofrece y que sobretodo se vayan contentos y vuelvan en otra ocasión.

Gracias a que se ha adherido a la plataforma Selfo, muchos de sus clientes tienen total autonomía en realizar los pedidos una vez sentados en la mesa. Entonces sus camareros quedan más liberados en realizar otras tareas necesarias del servicio. Miguel sabe que sobre todo en los picos de clientela que sus clientes utilicen Selfo es una gran ventaja tanto para él como para el cliente. Además, Miguel dispone de una tableta para ver todos los pedidos que hay activos en su restaurante que se han hecho a través de la plataforma.

Cuando una mesa que ha utilizado el servicio de la plataforma decide marcharse, Miguel sabe en todo momento si ha pagado o no. Entonces se puede despedir a los clientes sabiendo que no ha habido ningún problema durante el servicio que ha ofrecido.

3.2 Flujos de interacción

Gracias a los escenarios ficticios definidos anteriormente, se puede definir mejor el flujo de interacción de la app. Hay que tener en cuenta que la aplicación va a ser diferente dependiendo del tipo de perfil, si es comensal o administrador de un establecimiento.

Para un cliente de un restaurante tendrá más posibilidades a la hora de interactuar dentro de la app, en cambio para el perfil de administrador del restaurante estará más limitado.

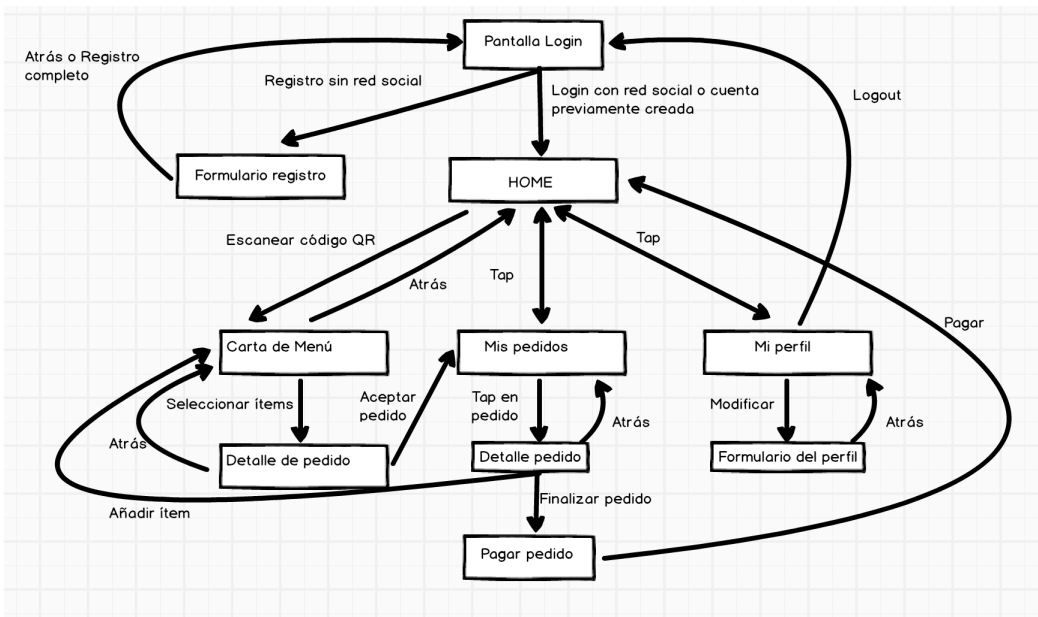


Figura 3: Árbol de navegación comensal

Para poder hacer pedidos el usuario se tendrá que registrar previamente, la razón es para tener identificado siempre quién hace el pedido cuando se utiliza el servicio. A partir de allí, podrá ir a cualquiera de los establecimientos adheridos a la plataforma para disfrutar de las ventajas de Selfo.

Para iniciar el proceso de realizar el pedido, el usuario solo tendrá que escanear el código QR que habrá en la mesa y le aparecerá el menú del restaurante. Entonces, podrá ir eligiendo las cantidades de cada ítem de la carta del menú y al finalizar podrá ver un resumen de todo lo que ha seleccionado antes de confirmar el pedido. Automáticamente, el pedido pasa a estar activo dentro del restaurante y el usuario podrá encontrarlo dentro de la sección de “Mis pedidos”. En cualquier momento del servicio, podrá añadir nuevos ítems al pedido. Para finalizar el pedido y proceder a pagar solo se tendrá que confirmar la finalización del pedido y seguir el proceso de pago correspondiente.

El usuario en cualquier momento podrá modificar su perfil sin necesidad de tener ningún pedido activo.

Para el perfil de administrador el árbol de navegación es mucho más sencillo, a parte de poder modificar el perfil, podrá ver todos los pedidos activos que hay en ese momento en el restaurante y que se han generado a través de la app.

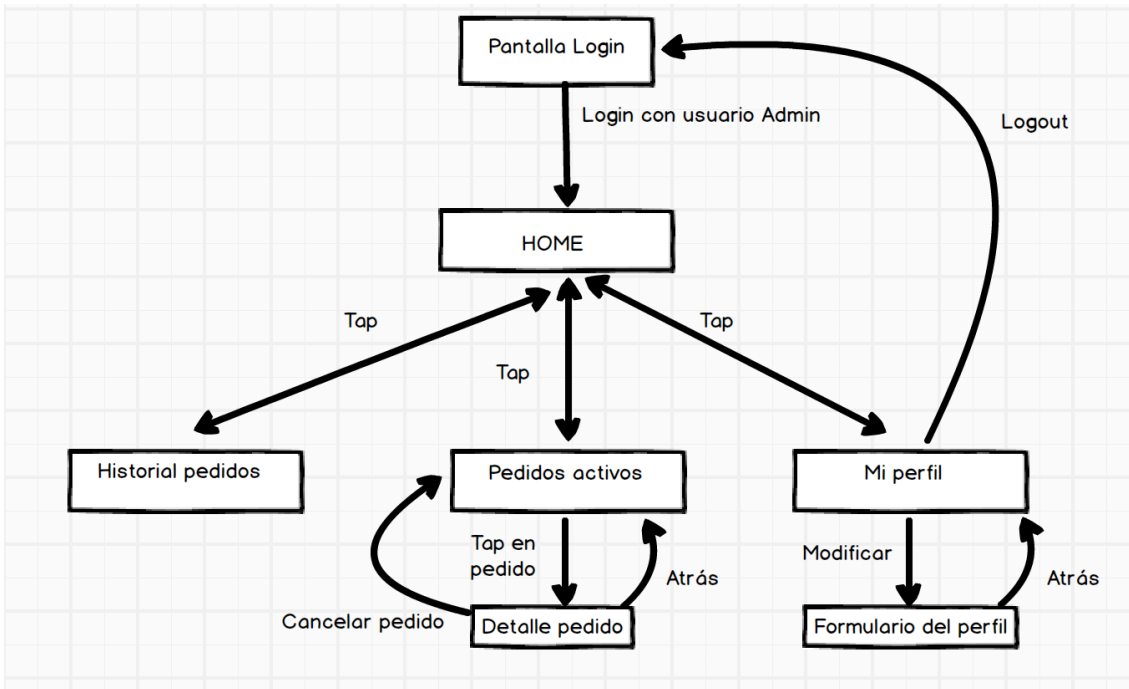


Figura 4: Árbol de navegación dueño establecimiento

El administrador del restaurante también podrá consultar un historial de pedidos.

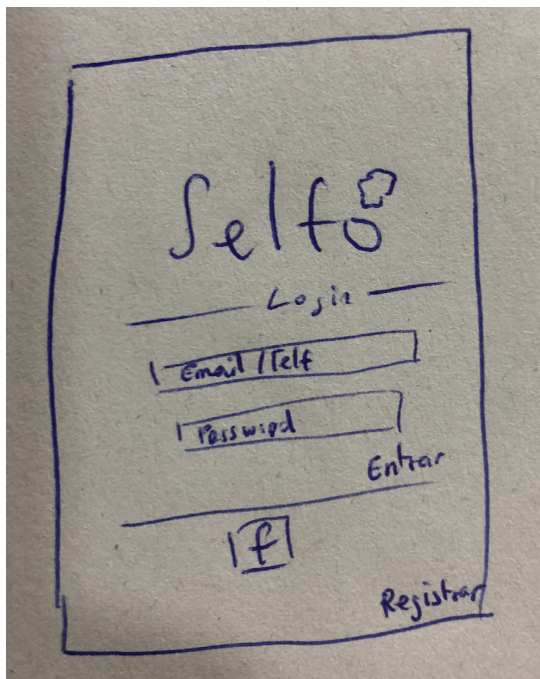
El administrador podrá ver qué pedidos están en Borrador, activos o finalizados. Es uno de los puntos importantes para el negocio, saber la carga de trabajo actual y la que le puede venir.

3.3 Prototipo de baja fidelidad

Tras la definición de los flujos de interacción, toca plasmar las ideas junto con los requisitos en unos bocetos detallados de cómo serán la aplicación y en cómo interactuarán entre ellas.

Se han hecho bocetos de todas las pantallas más importantes de la aplicación. Como anteriormente se ha descrito que el flujo es diferente dependiendo del tipo de usuario que lo utiliza, con lo que conlleva que algunas pantallas solo estarán para un perfil en concreto y otras se compartirán dependiendo del perfil.

Pantalla Login

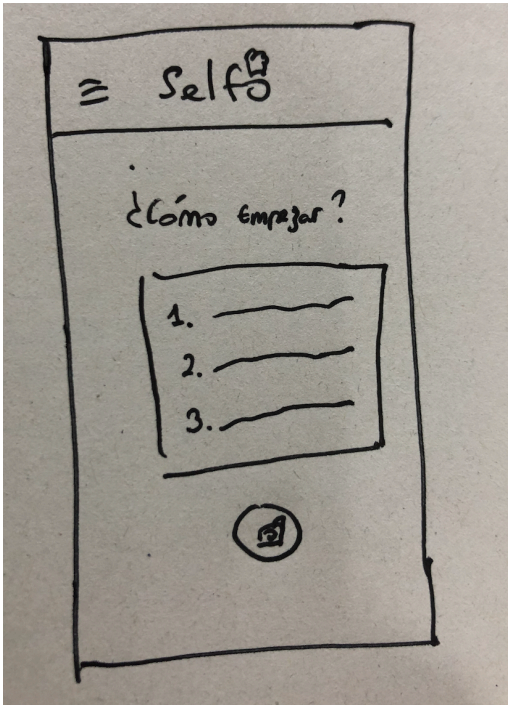


La pantalla de inicio de sesión o “login” es la primera pantalla que aparecerá cuando cualquier usuario entre en la aplicación sin importar el tiempo de perfil, comensal o administrador del restaurante. Una vez que el usuario haya introducido sus credenciales, se conocerá el tipo de perfil.

Es una pantalla simple y con buena usabilidad ya que no está llena de elementos, la idea es no estresar al usuario en su primera pantalla. Contiene dos campos rellenables para usuario y contraseña y un botón para entrar una vez introducidas las credenciales. Más abajo habrá la opción de utilizar la red social Facebook para entrar y como última opción un botón para registrarse en

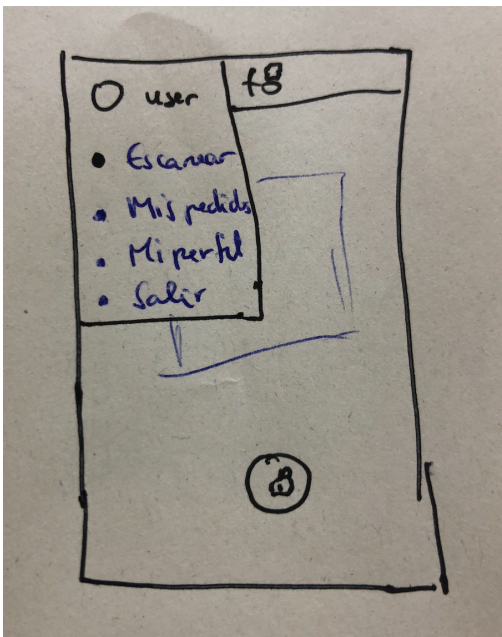
el caso de que no lo esté ya.

Home



La pantalla home, es la pantalla principal de la aplicación y esta pantalla tendrá diferente contenido dependiendo del perfil. En este caso se muestra el perfil de comensal, ya que le aparece un pequeño resumen de qué pasos hay que seguir para utilizar todo el servicio. Además está mapeado directamente el botón para abrir la cámara y empezar a escanear. En la parte superior, se sitúa el botón para abrir el menú.

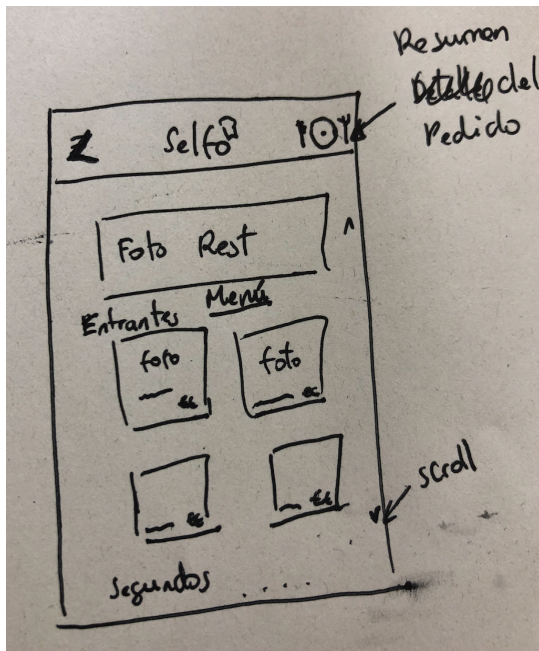
Menú



El menú contiene todas las acciones que puede hacer el usuario en ese momento. El usuario podrá:

- Escanear código QR.
- Mis pedidos. Podrá ver todos los pedidos del usuario, no solo el activo también los que ya ha finalizado.
- Mi perfil. Información del perfil del usuario.
- Salir. Cerrar sesión.

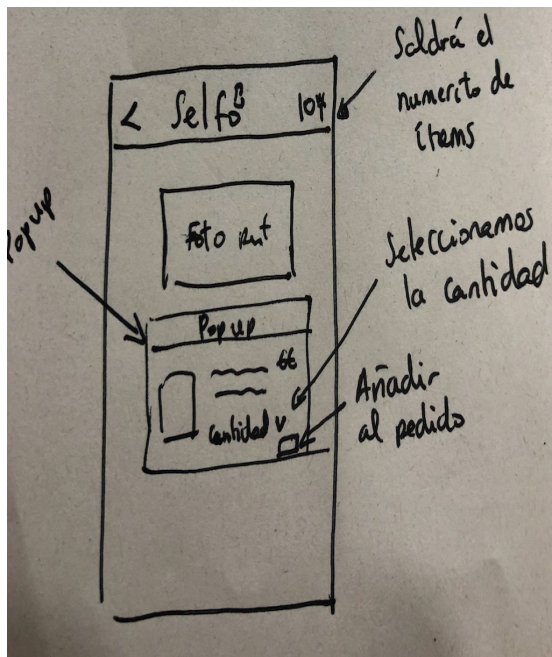
Carta de menú



Esta pantalla también es muy importante, ya que una vez escaneado el código QR, automáticamente se cargará la carta del menú del restaurante. En esta pantalla se verá una foto del restaurante para indicar en qué restaurante pertenece el menú y a continuación la carta con foto, título y precio de cada ítem de la carta. Esta pantalla tendrá scroll.

En la parte superior izquierda, hay un botón para volver atrás que será la pantalla Home. En la parte superior derecha, podrá ir al detalle del pedido actual.

Añadir ítem



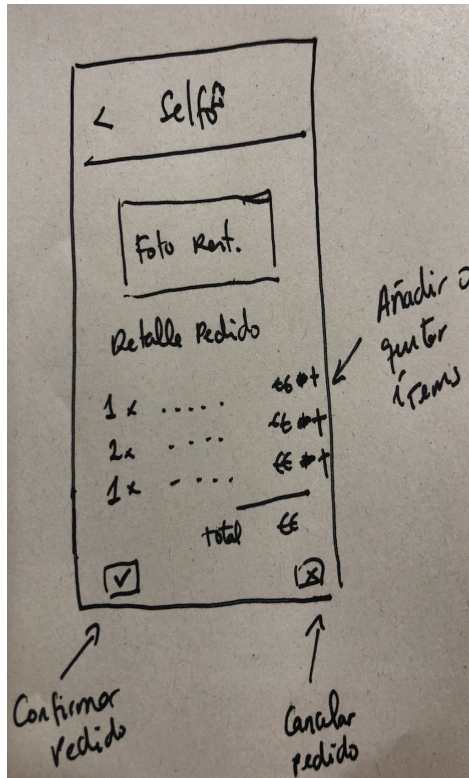
Este esbozo, es una extensión de la pantalla anterior de la carta de menú. Simplemente es una pequeña ventana emergente que aparecerá cuando se seleccione un ítem en la carta del menú.

Esta ventana contendrá toda la información detallada sobre el ítem seleccionado. A parte, contendrá un selector de cantidad para indicar las unidades que se quieran pedir del producto.

En la parte inferior de la ventana hay un botón para añadir el ítem y las cantidades al pedido actual, y también otro para cerrar la ventana y volver a la carta del menú.

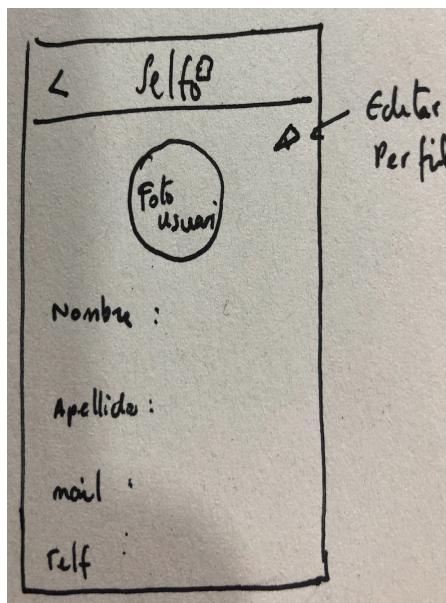
Detalle de pedido

Esta pantalla es el resumen del pedido. Muestra todos los ítems y las cantidades escogidos con su precio correspondiente y el total del pedido. Dependiendo del estado del pedido se cargará diferentes botones con diferentes acciones.



- Borrador. El pedido no se ha confirmado, el usuario solo ha seleccionado los ítems y las cantidades. En la parte inferior le aparecerán dos botones con las opciones de confirmar el pedido o cancelar el pedido. El botón superior de volver sirve para volver a la carta del menú,
- Activo. El pedido ha sido confirmado por el usuario y está en servicio. Entonces, aparece las opciones de añadir ítem al pedido o finalizar el pedido, que llevará al usuario al proceso de pago.
- Finalizado. El usuario podrá consultar pedido finalizados anteriormente. No contiene ninguna acción.

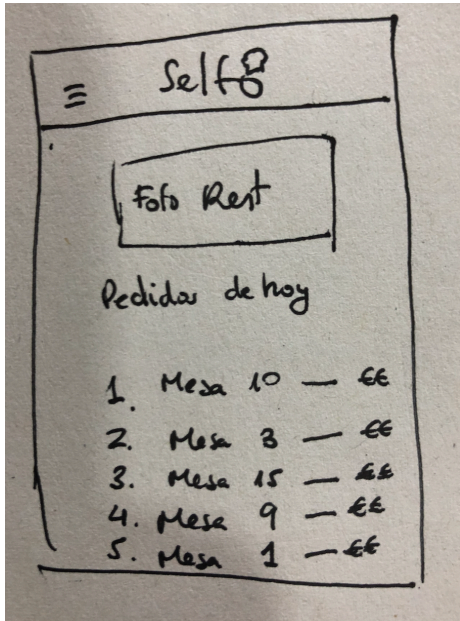
Mi perfil



Pantalla de información del perfil de usuario. Muestra toda la información del usuario y una foto, tiene una opción de poder modificar el perfil.

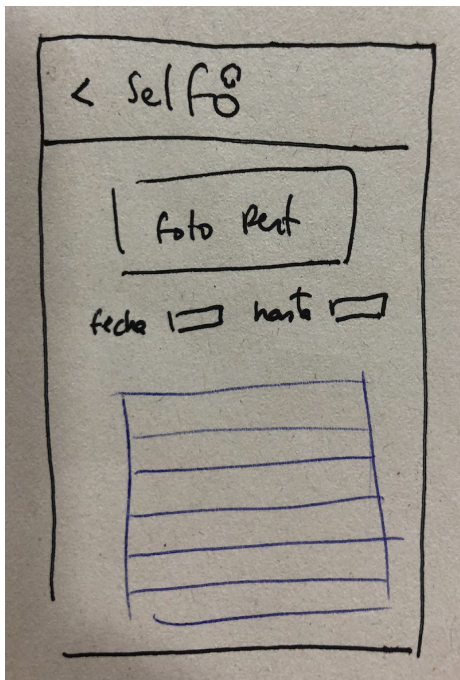
Hay menor cantidad de pantallas para un administrador de un establecimiento, ya que el flujo de interacción es menor. Se ha esbozado de manera detallada las pantallas más relevantes que sólo se mostrará cuando el usuario que acceda a la aplicación sea de tipo administrador.

Home – Usuario administrador



Las necesidades de un usuario administrador son diferentes a la de un cliente. Por esa razón la pantalla inicial muestra contenido relevante para el administrador. En este caso todos los pedidos actuales del día actual. A parte de una foto del restaurante. En todos los pedidos se podrá ver el detalle si se selecciona.

Historial de pedidos



Esta pantalla es para mostrar el historial de pedidos finalizados del restaurante. El usuario administrador podrá introducir un rango de fechas y entonces se mostrará en forma de lista. Se podrá seleccionar cada elemento de la lista y ver más detallado el pedido.

3.4 Prototipo alta fidelidad

Una vez terminado el proceso de esbozo de las pantallas, se intenta plasmar las ideas a un prototipo más formal. Para esta parte, el diseño se ha realizado en con el programa Balsamiq.

Pantalla login



La pantalla login será bastante simple y con la información necesaria. Contiene el logo, y dos campos para que el usuario pueda hacer login.

El usuario podrá utilizar una red social para registrarse o entrar. También puede registrarse si es la primera vez que entra y no quiere hacerlo a través de una red social.

Home para perfil usuario



En la pantalla Home del perfil usuario como cliente o comensal, se mostrará información de como funciona el proceso de todo el servicio. También habrá un acceso directo a la cámara para que el usuario pueda a empezar el proceso de escaneo del código QR.

Home perfil usuario con menú



El icono del menú está en la parte superior izquierda. Muestra todas las opciones que puede hacer dependiendo del tipo de perfil que está logueado. Las opciones son:

- Escanear código QR. Inicia la cámara.
- Mis pedidos. Historial de pedidos.
- Mi perfil. Perfil del usuario.
- Salir. Cerrar sesión actual.

Cartá menú

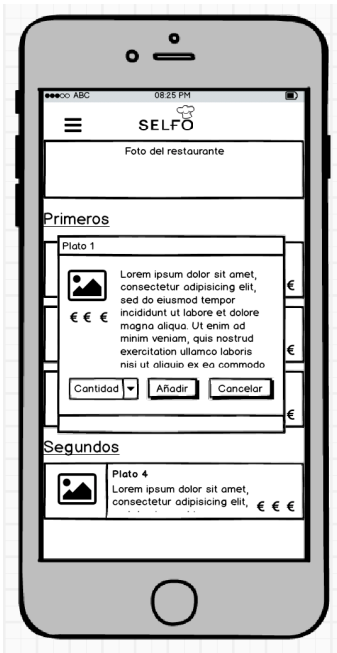


En la parte superior aparecerá una foto del restaurante para identificarlo. Entonces, justo a continuación se mostrará la carta con todos sus ítems. Estará separado por tipo de ítem y a parte de mostrar una foto y el precio habrá una pequeña descripción.

En el esbozo inicial, se había diseñado en forma de cajas, sin embargo a la hora de diseñarlo con más detalle, se mostraba mejor la información en forma de lista.

Además, al estar toda la carta integrada en una misma pantalla, el usuario podrá añadir ítems a su antojo.

Añadir ítem



Cuando se selecciona un ítem de la carta, se abre una ventana auxiliar para mostrar mejor toda la información y descripción del ítem. El usuario podrá seleccionar la cantidades si lo quiere añadir al pedido o cancelar si no lo quiere añadir al pedido.

Detalle de pedido



En esta pantalla, el usuario podrá entrar ver el datalle del pedido. Hay 3 estados del pedido:

1. Borrador. El usuario está eligiendo platos. Una vez finalizado tendrá que aceptar el pedido para que se haga efectivo.
2. En proceso. Se está sirviendo el pedido, el usuario puede añadir ítems al pedido.
3. Finalizado. El usuario ha finalizado el pedido y se dispone a pagar.

Mi perfil



Pantalla para ver información del perfil. Cambia la información dependiendo del tipo de usuario que está logueado. En este caso es la pantalla de un usuario comensal. Puede modificar la información básica y la foto de perfil.

Home Admin



Esta es la pantalla home de un usuario con perfil administrador. La información relevante son la lista de pedidos actuales de hoy. Si selecciona un ítem de la lista de pedidos podrá ver el detalle de ese pedido.

Historial de pedidos



Otras de las pantallas importantes para el perfil administrador es poder consultar el historial de pedidos. Seleccionando un rango de fechas se mostrará todos los pedidos que entren en el rango de fechas. Al seleccionar un objeto de la lista podrá ir al detalle del pedido.

4. Evaluación

La parte para probar el prototipo y poder definir los posibles casos de uso de la mejor manera posible es dejando al usuario final opinar y simular como si fuera real. Es un proceso muy importante ya que la aceptación de los usuarios condicionará muchas decisiones.

Para obtener la retroalimentación de los usuarios se ha creado un pequeño diseño más interactivo en invisio.

https://invis.io/RSOUZ8LZFNM#/328576612_Pantalla_Login

Se ha dejado probar el diseño a varios usuarios potenciales realizando las siguientes tareas:

1. Hacer login.
2. Simular todo el proceso de realizar el pedido.

Los tres usuarios han sido curiosos y han intentado ir a las diferentes secciones que estaban habilitadas como el perfil o el historial de pedidos. Después de dejarles un tiempo se ha recogido las opiniones en forma de las siguientes preguntas:

- ¿Te ha parecido fácil de utilizar?
- ¿Crees que hace el cometido del servicio?
- ¿Qué echas de menos?
- ¿Qué cambiarías?

Se ha probado el diseño en tres usuarios diferentes. A los tres usuarios les ha parecido bastante sencilla la utilización de la aplicación. Echan de menos el colorido, pero se ha comentado que solo era el diseño de un prototipo.

Creen que cumplen perfectamente para el cometido de tomar un pedido, y echan de menos el poder ver vídeos o filtros por tipos de comida. Son posibles mejoras que se pueden incorporar en futuras versiones.

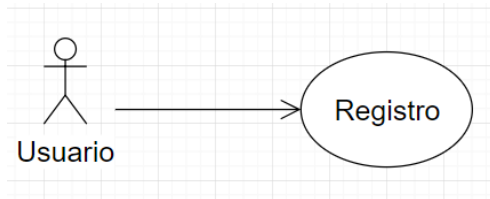
Comentan que el logo es original igual que el nombre y la idea de escanear un código QR es divertido. Sin embargo, han sugerido la posibilidad de qué pasaría si no reconoce el código QR, debería existir otro método para poder utilizar el servicio.

Una vez recogido las percepciones y encontrado los posibles puntos de mejora y evaluado el diseño, se tiene que definir los casos de uso y la arquitectura de todo el sistema.

4.1 Casos de uso

La definición de los casos de uso es un proceso que es muy importante estudiarla antes de la fase de desarrollo e implementación. Hay que separar los casos de uso de los tipos de perfiles que hay, comensal y administrador de restaurante.

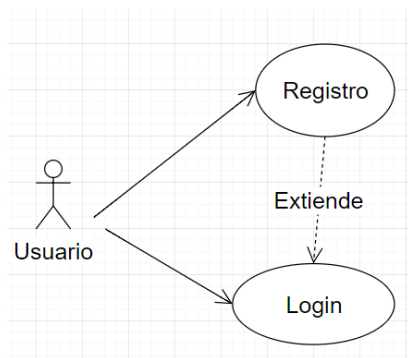
Caso de uso registro



Identificador	CU-001
Nombre	Registro
Prioridad	Alta
Descripción	El usuario se registra en la plataforma
Actores	Usuario (Comensal)
Precondiciones	Tener la aplicación bajada
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none">1. Abrir la aplicación.2. Hacer clic en "Registrarse".3. Rellenar el formulario con los datos necesarios.4. Enviar formulario.
Post Condiciones	Se da de alta al usuario y le garantiza acceso a la plataforma.
Notas	Ninguna

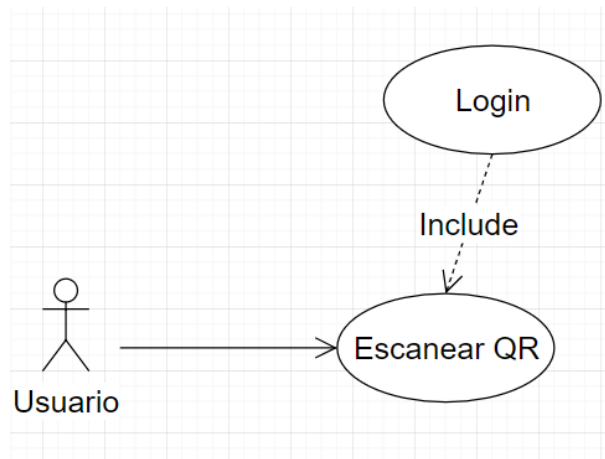
Este caso de uso sirve para ambos perfiles de usuario.

Caso de uso login



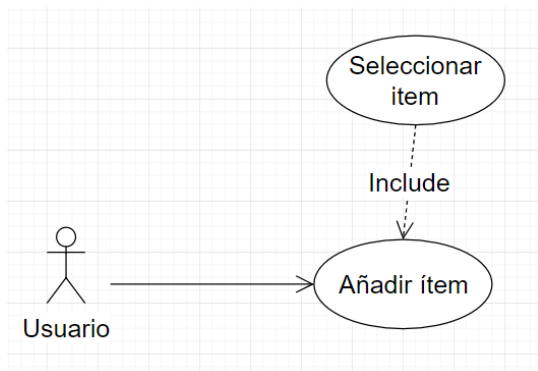
Identificador		CU-002
Nombre	Login	
Prioridad	Alta	
Descripción	El usuario se logea en la aplicación	
Actores	Usuario (Comensal)	
Precondiciones	Estar registrado en la plataforma	
Iniciado por	Usuario	
Flujo	<ol style="list-style-type: none">1. Abrir la aplicación.2. Rellenar credenciales.3. Clic en el botón de Entrar.	
Post Condiciones	Usuario accede a la aplicación con su usuario	
Notas	Ninguna	

Caso de uso escanear código QR



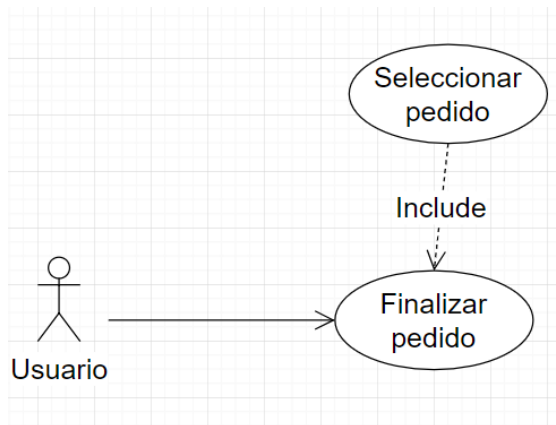
Identificador		CU-003
Nombre	Escanear código QR	
Prioridad	Alta	
Descripción	El usuario se escanea el código QR de la mesa	
Actores	Usuario (Comensal)	
Precondiciones	Estar logueado en la plataforma, garantizar permisos a la aplicación para acceder a la cámara.	
Iniciado por	Usuario	
Flujo	<ol style="list-style-type: none">1. Clic en el botón de la cámara.2. Escanear el código QR.	
Post Condiciones	Aparece el menú del restaurante al dispositivo del usuario.	
Notas	Ninguna	

Caso de uso añadir ítem al pedido



Identificador	CU-004
Nombre	Añadir ítem al pedido
Prioridad	Baja
Descripción	El usuario añade un ítem al pedido
Actores	Usuario (Comensal)
Precondiciones	Tener seleccionado un ítem
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none">1. Seleccionar la cantidad.2. Clic en añadir al pedido.
Post Condiciones	Se añade un ítem al pedido con la cantidad seleccionada.
Notas	Ninguna

Caso de uso finalizar pedido

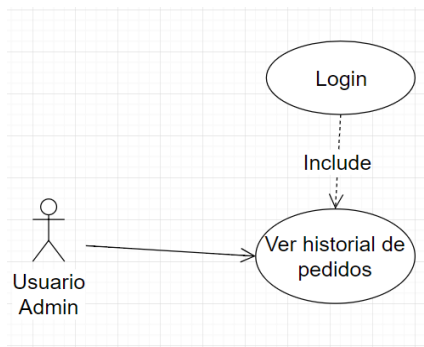


Identificador	CU-005
Nombre	Finalizar pedido
Prioridad	Alta
Descripción	El usuario añade un ítem al pedido
Actores	Usuario (Comensal)
Precondiciones	Tener un pedido activo
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none">1. Entrar al detalle del pedido.2. Clic en finalizar pedido.
Post Condiciones	Se añade un ítem al pedido con la cantidad seleccionada.
Notas	Ninguna

Este caso de uso también sirve para el usuario administrador, cuando selecciona un pedido activo puede tener la posibilidad de cancelarlo en cualquier momento por las razones que sean.

Hay algunos casos de uso que solo el usuario administrador podrá tener.

Caso de uso historial de pedidos



Identificador	CU-005
Nombre	Historial de pedidos
Prioridad	Alta
Descripción	El usuario puede ver los pedidos que se han realizado en un periodo en concreto.
Actores	Usuario (Administrador)
Precondiciones	Estar logueado como usuario administrador.
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none">1. Entrar al detalle del pedido.2. Clic en finalizar pedido.
Post Condiciones	Se muestra la lista de pedidos seleccionados.
Notas	Ninguna

4.2 Diseño de la arquitectura

Una vez realizado el prototipo de la aplicación, se va a entrar un poco en detalle sobre la arquitectura de ésta planteando un diseño de las clases que se van a utilizar y el diseño de la base de datos donde se guardará toda la información.

Diagrama de clases

La aplicación tendrá cinco clases:

- Usuario. Se guarda toda la información del usuario que está dentro de la sesión. Puede guardar la información de los pedidos que tiene si en algún momento lo requiere.
- Pedido. Clase que representa un pedido contiene todos ítems del pedido.
- Restaurante. Alberga toda la información del restaurante en el que el usuario interactúa en ese momento.
- Menú. Clase que representa la carta del menú con todos los ítems del restaurante.
- Ítem. Clase para representar un objeto del menú.

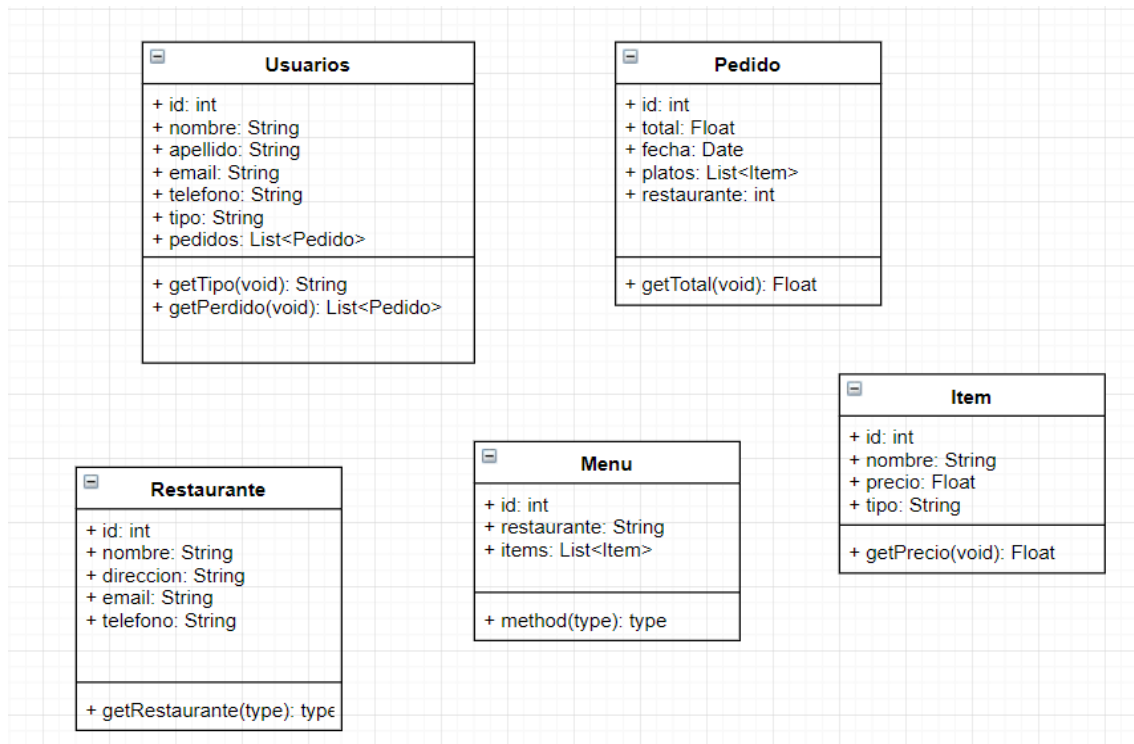


Fig. 3 Diagrama de clases

Diagrama de base de datos

En el diagrama de base de datos se representa las relaciones de las tablas donde se va a aguaradar la información de toda la plataforma. Se ha diseñado con el modelo Entidad Relación.

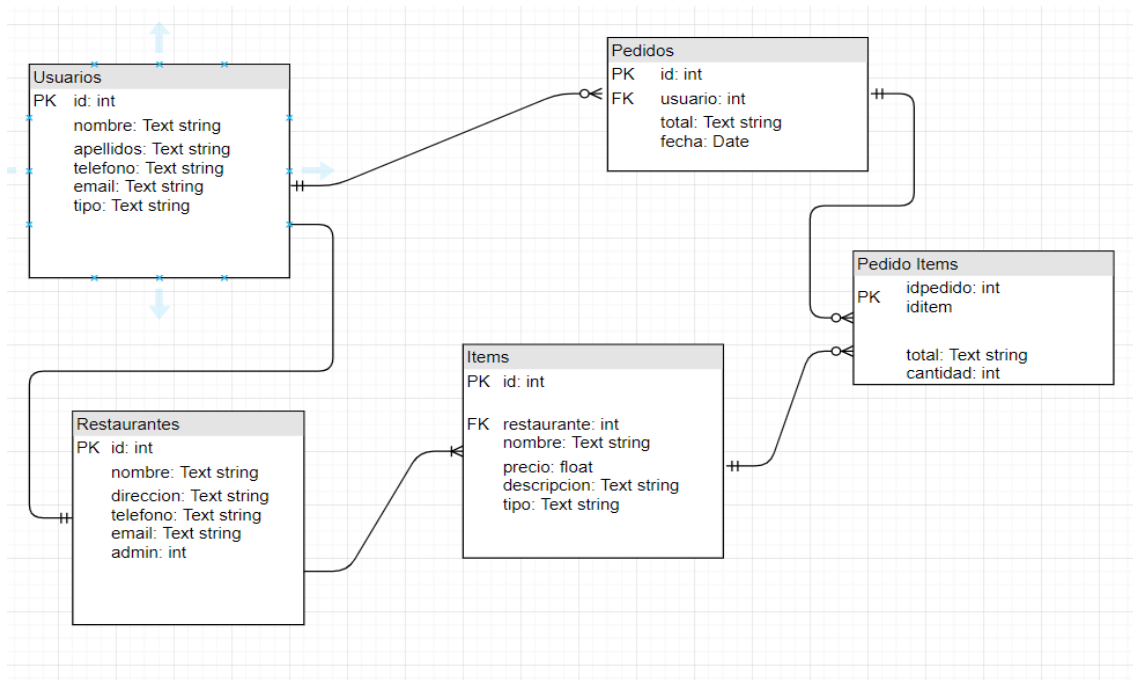


Fig 4. Diagrama de base de datos

Cuando un usuario se registra, automáticamente se crea un registro en la tabla de usuarios. Este usuario puede tener ningún pedido si es nuevo o tener varios pedidos si ya ha utilizado el servicio. Dependiendo del perfil, si el usuario es del tipo administrador de un restaurante, se le asignará un restaurante en la tabla de restaurantes.

Un restaurante tendrá un administrador y también tendrá ítems que serán todos los platos y productos que tendrá ese restaurante. Todos los pedidos tienen que tener al menos un producto, y al ser una relación que muchos productos pueden estar en diferentes pedidos, se genera una tabla auxiliar Pedido-Ítems donde se guardará toda la información de cada pedido con sus ítems añadidos.

Diagrama de arquitectura

Otro aspecto muy importante del diseño de la plataforma es la arquitectura que se va a utilizar. Se ha definido previamente la tecnología que se va a utilizar, que en este caso es ionic. La tecnología de ionic puede desarrollarse en varias arquitecturas y para este proyecto se ha escogido Modelo Vista Controlador (MVC). Las razones han sido:

- Modelo muy asentado en el mundo del desarrollo.
- Facilidad de uso y reusabilidad de vistas.
- Capa de datos separada mostrando siempre la información actualizada en todo momento.
- Más experiencia con este patrón de diseño.

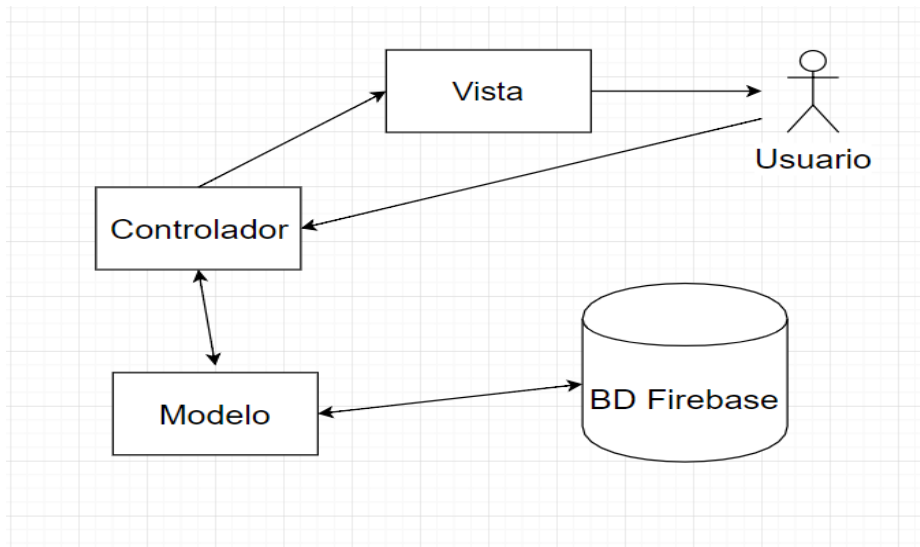


Fig. 5 Diagrama de arquitectura

El usuario verá toda la información a través de la vista, si realiza una acción el controlador se encarga de recogerla y procesarla. Si requiere datos o actualización de la base de datos el modelo recoge los cambios y los aplica en la base de datos o recoge la información para pasarsela al controlador por si tiene que tratarla y al final se muestra al asuario a través de la vista.

5. Desarrollo

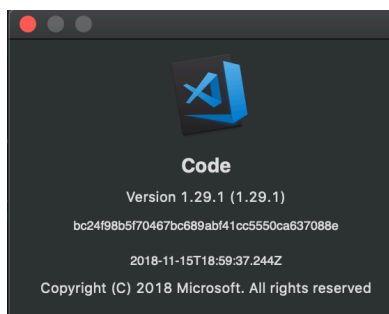
Una vez explicado la parte de diseño del proyecto, en este apartado se va a centrar únicamente la parte de desarrollo, detallando toda la información y explicando cada una de las decisiones tomadas.

El sistema utilizado para el desarrollo ha sido MacOS Mojave versión 10.14, algunas pruebas y testeos también se han ejecutado en entorno Windows 10.

5.1 Herramientas para el desarrollo.

Al iniciar el desarrollo, elegir las mejores herramientas para el tipo de proyecto que se va a desarrollar afecta directamente a la calidad y tiempo de finalización del proyecto.

Como herramienta principal de desarrollo para este proyecto se ha utilizado “Visual Studio Code”, que es un editor de texto creado por Microsoft. La versión utilizada ha sido 1.29.1.



La razón de utilizar este editor de texto ha sido que se integra muy bien con los proyectos web y multiplataforma y ofreciendo grandes ventajas a la personalización con pluguins de terceros. Gracias a dicho pluguins se puede optimizar el tiempo en realizar muchas tareas. Otra característica importante de este editor es su gran “intellisense”, es la capacidad de autocompletar el código que estás escribiendo.

Como se ha mencionado anteriormente, el framework utilizado para este proyecto ha sido Ionic en su versión 3. Gracias a su integración con capacidades nativas a los diferentes recursos del dispositivo móvil y su capacidad de multiplataforma.



Esta versión de Ionic, utiliza Angular 5 que es un framework de código abierto basado en JavaScript para crear aplicaciones web.

Como servicio de base de datos también se ha mencionado a Firebase. Utilizando su servicio de bases de datos en tiempo real. Es una de las características principales de Firebase, cuando se actualiza un dato en Firebase automáticamente se refleja en la app.



Estas son las herramientas principales del desarrollo de la aplicación. Sin embargo, también hay que hacer especial mención a plugins y librerías de terceros que se han utilizado para llevar a cabo el proyecto.

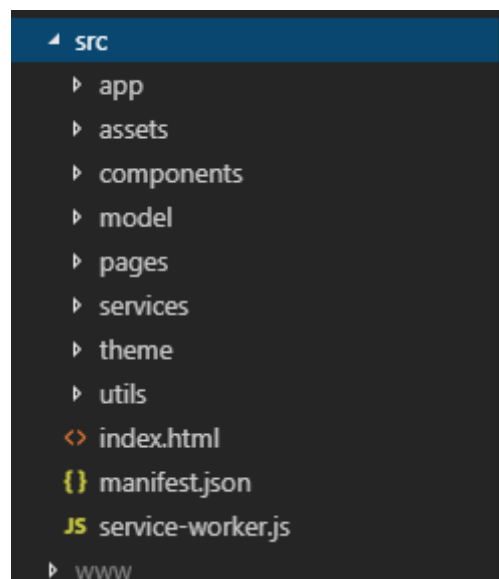
- AngularFire 2. <https://github.com/angular/angularfire2>
Librería oficial de Angular para trabajar con Firebase. Muy útil para crear, escribir y leer en la base de datos hospedada en Firebase.
- Camera. <https://ionicframework.com/docs/native/camera/>
Plugin para acceder a las características de la cámara del dispositivo donde se está ejecutando la aplicación. Se requiere para acceder a la cámara.
- Camera preview. <https://ionicframework.com/docs/native/camera-preview/>
Otro plugin nativo de Ionic que sirve para mostrar la previsualización de la cámara.
- QR Scanner. <https://ionicframework.com/docs/native/qr-scanner>
Como su nombre indica es un plugin para leer códigos QR. Funcionalidad implementada en la aplicación.
- Reactive Extensions (RxJS). <https://rxjs-dev.firebaseio.com/>
Es una librería que contiene modelo de objetos de Firebase, haciendo más sencillo la sincronización de la app con la base de datos.

5.2 Estructura del proyecto

Se definió que el patrón de arquitectura sería Modelo Vista Controlador, sin embargo, el framework de Ionic 3 con Angular 5 se integra de manera estándar al Modelo Vista Vista Modelo (MVVM). Es la manera más eficiente de trabajar con el framework elegido, por esa razón se ha decidido este patrón.

Realmente no cambia mucho, ya que MVVM es una evolución del MVC. Cada vista (pantalla) interactúa con el modelo y contiene su propio controlador para hacerlo.

La estructura final del proyecto es la siguiente:



- App: residen todos los archivos principales para la inicialización de la aplicación junto con la configuración principal.
- Assets: albergan las imágenes necesarias para el proyecto.
- Components: son pequeñas funcionalidades que se pueden aprovechar en varios sitios y así poder tener una extracción del código.
- Model: están todas las clases utilizadas en el proyecto.
- Pages: todas las pantallas (vistas) que hay dentro de la aplicación.
- Services: servicios utilizados para gestionar la base de datos en Firebase.
- Utils: conjunto de funcionalidades que son utilizadas por diferentes módulos.

5.3 Pruebas

Muchas de las pruebas que se han realizado han sido manualmente a través de la app. Sin embargo, ciertas pruebas sí que se han podido automatizar.

- Creación de usuario.

```
myCredentials = {
  email: 'luishong.wu@gmail.com',
  password: [REDACTED]
};

testCreationOfUsers(public afAuth: AngularFireAuth) {

  this.signInWithEmail(this.myCredentials);
  afAuth.authState.subscribe(user => {
    this.user = user;
  });
}
```

- Creación de pedido.

```
testCrearPedido(){
  var myTestPedido = {
    R0000P0001: 2,
    R0000P0002: 1,
    R0000P0003: 3
  }
  this.CrearPedido(myTestPedido);
}
```

Para crear un pedido se requiere el identificador del ítem pedido y la cantidad.

Otras pruebas se han realizado en un emulador a medida que se iban implementando.

- Inicio de sesión. Se ha probado que un usuario ha podido iniciar sesión dentro de la aplicación. Avisando al usuario si ha fallado.
- Recuperar contraseña. Se ha probado que el usuario recibe un mail para poder reiniciar la contraseña.
- Lector de código QR. Se ha probado que el dispositivo inicia la cámara previo aceptación de permisos y lee correctamente los códigos QR.
- Carga de la carta de menú. Se ha probado que carga la carta del menú de manera correcta una vez escaneado el código QR.

- Creación de pedidos. Poder crear el pedido con los datos correctos del restaurante y la fecha.
- Añadir ítems al pedido. Pruebas de añadir y borrar ítems al pedido y actualizando el precio del pedido.
- Aceptar y finalizar pedido. Cambiar status del pedido cuando el usuario quiere aceptar o finalizar el pedido.

5.4 Estado actual del desarrollo.

El estado actual del desarrollo está al 75% a falta de la parte de administrador y corregir algunos detalles de interfaz y control de errores.

Existe una pequeña batería de datos en Firebase para probar la aplicación en modo usuario.

Hay una desviación de un 20% respecto a la planificación inicial. La razón es desconocimiento de la plataforma de desarrollo y falta de experiencia en crear aplicaciones de este calibre. Es un riesgo que existía y se ha intentado minimizar al máximo posible.

La valoración de ese 20% faltante, es que la parte de administrador se puede reaprovechar el 80% de lo que hay creado para cumplir la planificación inicial.

Problemas actuales en la aplicación

Hay varios problemas que se encuentran actualmente en la aplicación pendientes de resolver. Son pequeñas correcciones que no afectan al funcionamiento actual de la aplicación.

- Al volver a la pantalla Home después de utilizar la cámara, el fondo se vuelve transparente (solo pasa en dispositivo físico, no en el emulador).
- Se puede borrar ítems de un pedido habiendo aceptado o finalizado el pedido.
- Al mostrar el teclado para introducir los datos, se mueve la vista.
- No se ha creado la posibilidad de modificación de perfil.

6. Conclusiones

La realización de este trabajo ha supuesto un gran reto personal. Se ha podido adentrar en diferentes aspectos sobre un proyecto para el desarrollo de aplicaciones móviles. Estoy satisfecho por el aprendizaje que he obtenido a lo largo del máster y que muchas de las cosas aprendidas se han podido aplicar para este proyecto. También, al adentrarme en tecnologías desconocidas he podido ver de primera mano la manera en que se desarrollan las aplicaciones y sinceramente creo que es una experiencia valiosa para futuros proyectos.

Sin embargo, no se ha podido lograr todos los objetivos que se había planteado, ya que el proyecto era muy ambicioso y el tiempo muy ajustado. Pero destacar que lo más importante se ha implementado con éxito. En parte gracias a una buena planificación, realizada desde el principio. Había un riesgo en la fase de implementación, dado que se iba a desarrollar con una tecnología en la cual no tenía experiencia.

Con la metodología para la realización del proyecto, contemplaba cambios y han ido muy bien a la hora de adaptarse a las necesidades y los requisitos. Sobre todo en el aspecto del desarrollo y diseño, no ha sido exactamente lo que se había planteado en un inicio, ha variado un poco respecto a la valoración inicial.

Como trabajo futuro es muy importante el estudio y elaboración de un modelo de negocio, ya que ello depende el éxito de este proyecto. Así como ofrecer más valor a los dueños para que accedan a adherirse a la plataforma. Sobre las funcionalidades de la aplicación, un trabajo pendiente es el diseño y la usabilidad, uno de los aspectos que no se ha podido tocar en este proyecto pero que creo que es muy importante hoy en día.

7. Glosario

- App – Aplicación
- Comensal – Cliente de un restaurante.
- iOS – Sistema operativo para dispositivos móviles Apple.
- Android – Sistema operativo para dispositivos móviles desarrollado por Google de código abierto.
- Código QR – Es un módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional.
- Framework - Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- Ionic – Framework de desarrollo para dispositivos multiplataforma.
- HTML – Lenguaje de marcado para construir páginas web.
- CSS – Hojas de estilo en cascada.
- JavaScript – Lenguaje de programación que se ejecuta en el navegador.
- Retroalimentación – Opinión
- Plugin – Componente que añade una funcionalidad adicional o una nueva característica al programa.

8. Bibliografía

<https://www.projectlibre.com> – 24/09/2018

<http://mymenu.info/es/> - 22/09/2018

<https://menu.app/> - 22/09/2018

<https://sistemas.com/casos-de-uso.php> - 17/10/2018

<https://www.unirioja.es/cu/arjaime/Temas/03.UML.pdf> - 18/10/2018

<https://draw.io> - 23/10/2018

<https://firebase.google.com/docs/firestore/manage-data/data-types> - 25/10/2018

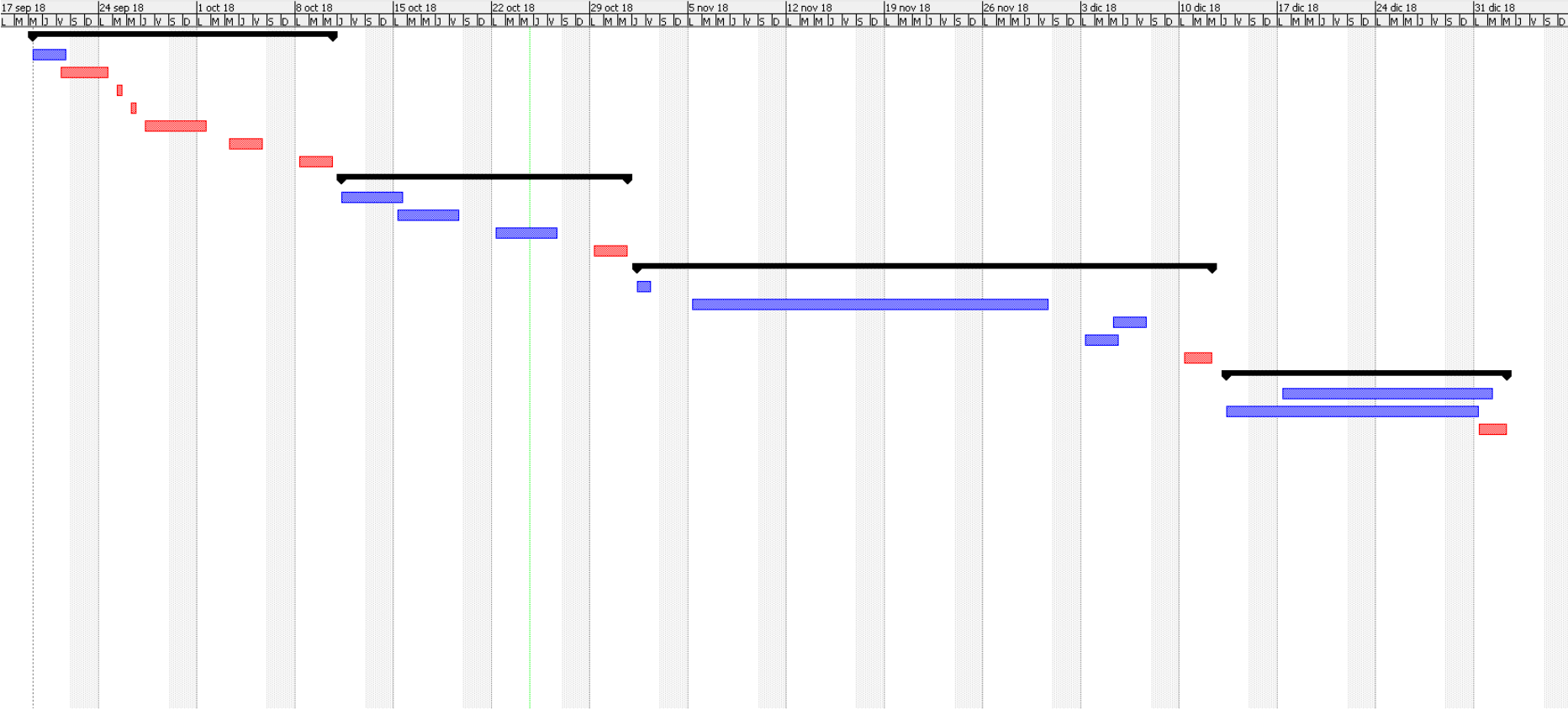
<https://www.invisionapp.com/> - 26/10/2018

<https://ionicframework.com/docs/> - 12/11/2018

<https://stackoverflow.com> – 12/11/2018

7. Anexos

Diagrama de Gantt



Capturas de la aplicación

