

# *Gestión Restaurante*

PL

**Proyecto Integrado 2º DAW**

**Tutor: José Piñero Berbel**

**Tutora de FCT y PI: Alicia Benítez Yáñez**

**Alumno: Pablo López Lozano**

## Resumen de los contenidos y alcances del proyecto

### Español:

Mi proyecto tiene como objetivo desarrollar una aplicación web para la gestión de un restaurante. La aplicación se enfocará en el control de mesas y comandas, permitiendo crear, gestionar y finalizar pedidos. Así para tener un mejor control del restaurante

### Inglés:

My project aims to develop a web application for the management of a restaurant. The application will focus on controlling tables and orders, allowing you to create, manage and finalize orders. So to have better control of the restaurant

## Índice

<b>Resumen de los contenidos y alcances del proyecto</b>	<b>2</b>
<b>Justificación</b>	<b>3</b>
a. Características generales	3
b. Restricciones generales, si las hay	3
c. Aspectos a cubrir, junto con los que no se van a tratar	3
d. Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría.	3
<b>Justificación de la tecnología empleada</b>	<b>3</b>
<b>Requerimientos hardware y software</b>	<b>4</b>
<b>Análisis y diseño</b>	<b>4</b>
Diagrama de casos de uso	5
Diagrama de clases	6
Árbol General de la Base de Datos	7
Descripción de los Datos de los Documentos	9
<b>Implementación</b>	<b>11</b>
<b>Evaluación y prueba</b>	<b>11</b>
<b>Manual de estilos</b>	<b>11</b>
a. Sketches	11
b. Criterios de accesibilidad	17
c. Criterios de usabilidad	17
d. Tipografía	17
f. Dispositivos/vistas para las que se ha diseñado el proyecto .	17
<b>Software utilizado</b>	<b>18</b>
Desarrollo:	18
Diseño	18
Funcionalidades	18
Back-end	18
<b>Mejoras posibles y aportaciones</b>	<b>18</b>

## Justificación

### *a. Características generales*

Cubrir una necesidad de los restaurantes  
Ganar eficiencia y eficacia para el negocio  
Seguridad de los datos

### *b. Restricciones generales, si las hay*

Todo el backend de la aplicación web está en los servicios de firebase (actualmente en una versión gratuita ya que el volumen de peticiones no sobrepasa el límite).  
La web está alojada en Firebase Hosting por lo que su disponibilidad depende de Firebase.  
Escalable fácilmente con los planes que ofrece Firebase

### *c. Aspectos a cubrir, junto con los que no se van a tratar*

Gestión del restaurante  
Organización de las mesas (Abrir, cerrar mesas)  
Envío de comandas  
Gestión de las comandas  
Registro y logueo de usuarios con verificación por correo electrónico y recuperación de contraseña.  
Administración de usuarios, mesas y carta

No se va a tratar:  
PWA

### *d. Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría.*

Las soluciones que ya existen, son caras y pueden llegar a ser difícil de implementar

## Justificación de la tecnología empleada

1. **React:** Framework de JavaScript facilidad de uso, componentes reutilizables, rendimiento, comunidad amplia y curva de aprendizaje leve.
2. **Progressive Web App:** Instalación sencilla, SEO.
3. **Firebase Store:** Base datos escalable fácilmente y obtención de datos en tiempo real para guardar los archivos (imágenes en mi caso).
4. **Firebase Database:** Base de datos NoSQL flexible para datos no estructurados, escalable fácilmente y obtención de datos en tiempo real para la gestión de comandas, mesas etc....

5. **Firestore Authentication:** Autenticación de usuarios segura y confiable.
6. **Firestore Hosting:** Alojamiento web escalable, seguro y gratis hasta un gran volumen de peticiones.
7. **Código QR:** Facilidad de acceso a la carta del restaurante.

## Requerimientos hardware y software

Servidor web con suficiente capacidad de procesamiento y memoria para alojar la aplicación. En mi caso usaré Firebase como servidor web, hasta que se no cubra un volumen límite de peticiones el servidor de la web será gratis una vez se cubra ese volumen, se puede escalar fácilmente con las tarifas de Firebase.

Dispositivos móviles o tablets para que los camareros puedan acceder a la aplicación y gestionar las mesas y comandas.

Pequeña impresora para tickets para que se puedan imprimir las cuentas generadas gracias a react-pdf

Sistema operativo compatible con el servidor web y los dispositivos móviles.  
Navegador web actualizado.

## Análisis y diseño

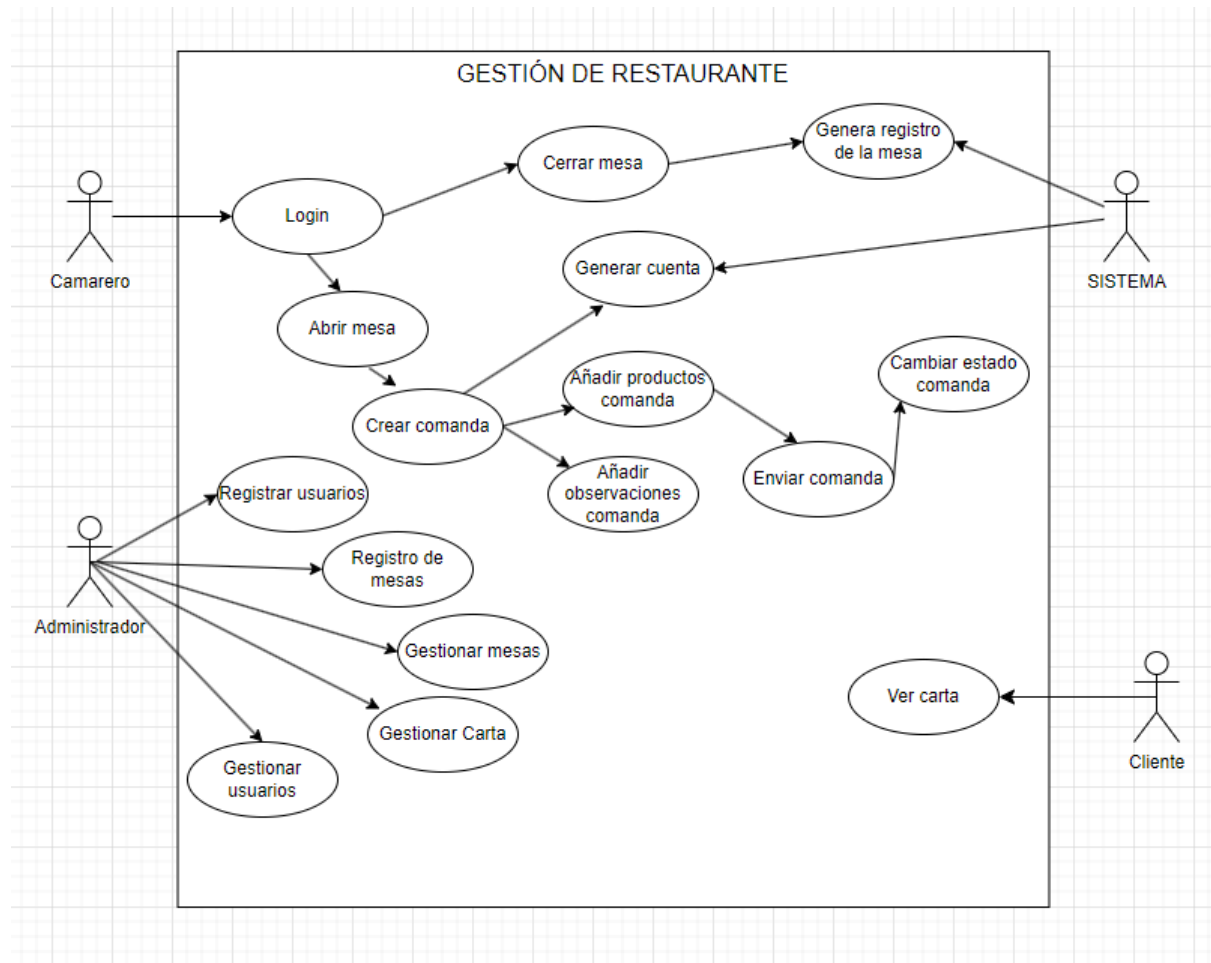
Mi aplicación está diseñada para solventar el problema de los restaurantes a la hora de gestionar el sistema de las comandas, con lo cual la gran parte de la aplicación es la parte privada en la que hay que estar registrado para poder acceder. Para registrar un nuevo empleado, los únicos que pueden hacerlo son los administradores.

La pequeña parte pública que tiene mi aplicación es la carta del restaurante y la landing page de la web la cual la puede visitar todo el mundo sin necesidad de tener una sesión iniciada.

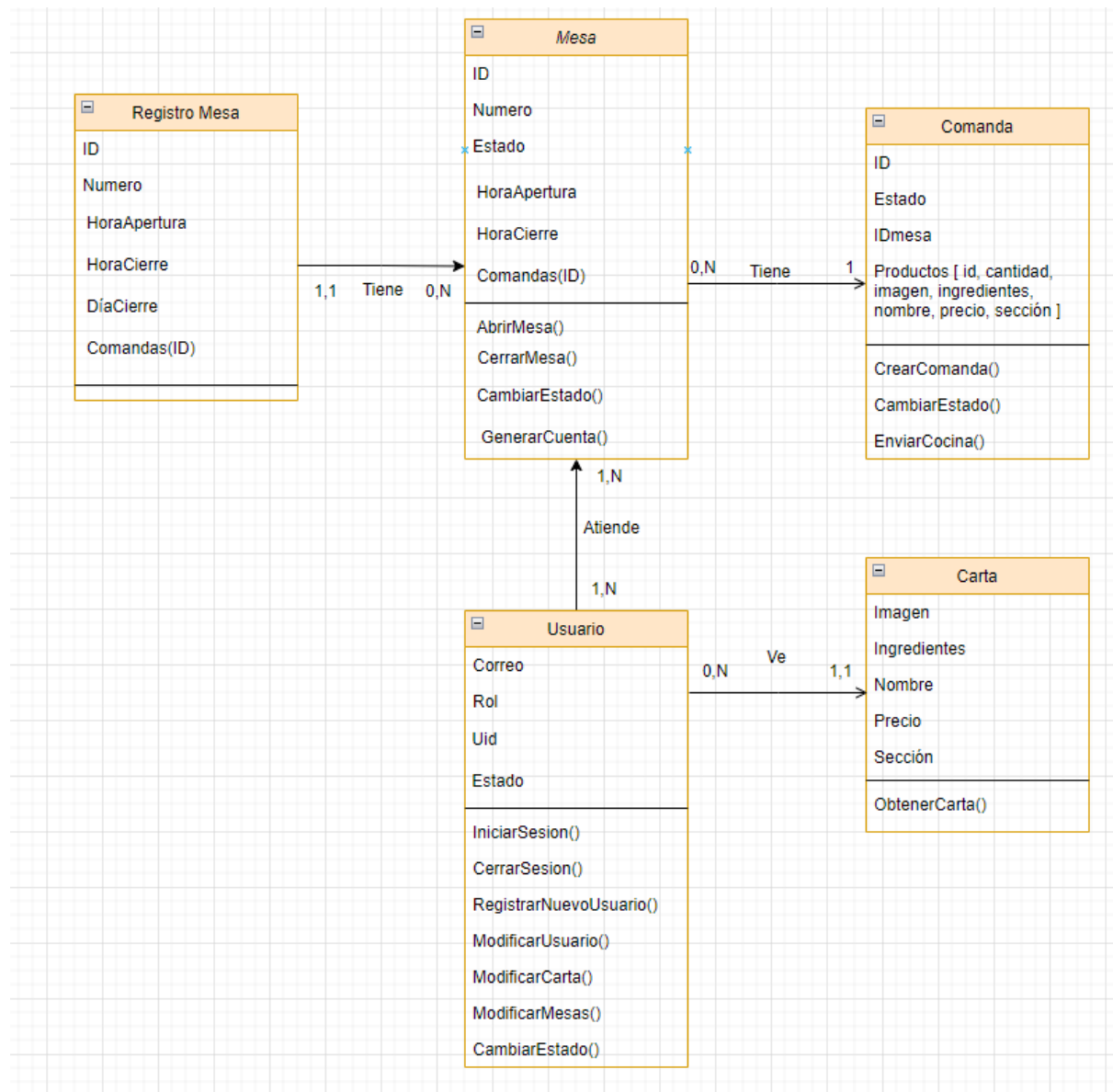
Volviendo a la parte privada, cualquier usuario con sesión iniciada puede abrir y cerrar mesas, crear comandas, calcular cuentas, procesar las comandas (cambiar el estado de las comandas).

Los administradores, accederán a toda la parte de la gestión. Podrán gestionar los empleados, crear, borrar y editar mesas, y actualizar la carta como se requiera.

## Diagrama de casos de uso



## Diagrama de clases



## *Árbol General de la Base de Datos*

Las colecciones principales de la base de datos serán:

Base de datos no relacional:

---

Leyenda:

- COLECCIÓN
  - CAMPO
  - ELEMENTOS ARRAY
- 

- Carta

La colección de **Carta** contendrá documentos que representan los diferentes platos y bebidas ofrecidos por el restaurante. Se compondrá de categoría y productos el cual es una nueva colección dentro del documento.

Cada documento tiene la siguiente estructura:

- carta
  - categoría
  - productos
    - categoria\_id
    - id
    - imagen
    - ingredientes
    - nombre
    - precio

- Comandas

La colección de **Comandas** contendrá documentos que representan los pedidos realizados por los clientes.

Cada documento tiene la siguiente estructura:

- comandas
  - horacreacion
  - idMesa
  - numeroMesa
  - productos[]
    - cantidad
    - categoría
    - id
    - imagen
    - nombre
    - precio
    - ingredientes

- Mesas

La colección de **Mesas** contendrá documentos que representan las mesas del restaurante.

Cada documento tiene la siguiente estructura:

- Mesas
  - id
  - número
  - estado
  - capacidad
  - horaapertura
  - horacierre
  - dia

- Registro Mesas

La colección de **Registro Mesas** contendrá documentos que representan el historial de ocupación de las mesas.

Cada documento tiene la siguiente estructura:

- id
- número
- dia
- horaapertura
- horacierre
- comandas[]
  - horacreacion
  - idMesa
  - numeroMesa
  - productos[]
    - cantidad
    - categoría
    - id
    - imagen
    - nombre
    - precio
    - ingredientes



- Users

La colección de **Users** contendrá documentos que representan a los usuarios del sistema, que pueden ser clientes o personal del restaurante.

Cada documento tiene la siguiente estructura:

- Users
  - correo
  - rol
  - uid
  - estado

### *Descripción de los Datos de los Documentos*

- Carta

**categoría:** Categoría a la que pertenece el producto (e.g., entrante, carne, pescado, postre, bebida).

**productos:**

**categoría\_id:** Identificador único de la categoría.

**id:** Identificador único del producto.

**imagen:** URL de la imagen del producto.

**ingredientes:** Lista de ingredientes del producto.

**nombre:** Nombre del plato o bebida.

**precio:** Precio del plato o bebida.

- Comandas

**comandas:**

- **horacreacion:** Hora de creación de la comanda.
- **idMesa:** Identificador de la mesa.
- **numeroMesa:** Número de la mesa.
- **productos:**
  - **cantidad:** Cantidad del producto.
  - **categoría:** Categoría del producto.
  - **id:** Identificador del producto.
  - **imagen:** URL de la imagen del producto.
  - **nombre:** Nombre del producto.
  - **precio:** Precio del producto.
  - **ingredientes:** Lista de ingredientes del producto.

- Mesas

**id:** Identificador único de la mesa.

**número:** Número identificativo de la mesa.

**estado:** Estado de la mesa (e.g., ocupada, libre).

**capacidad:** Número de asientos disponibles en la mesa.

**horaapertura:** Hora de apertura de la mesa.

**horacierre:** Hora de cierre de la mesa.

**dia:** Día de uso de la mesa

- Registro Mesas

**id:** Identificador único del registro.

**número:** Número identificativo de la mesa.

**dia:** Día de operación de la mesa.

**horaapertura:** Fecha y hora de apertura de la mesa.

**horacierre:** Fecha y hora de cierre de la mesa.

**comandas:**

- **horacreacion:** Hora de creación de la comanda.
- **idMesa:** Identificador de la mesa.
- **numeroMesa:** Número de la mesa.
- **productos:**
  - **cantidad:** Cantidad del producto.
  - **categoría:** Categoría del producto.
  - **id:** Identificador del producto.
  - **imagen:** URL de la imagen del producto.
  - **nombre:** Nombre del producto.
  - **precio:** Precio del producto.
  - **ingredientes:** Lista de ingredientes del producto.

- Users

**correo:** Correo electrónico del usuario.

**rol:** Rol del usuario en el sistema (e.g., cliente, camarero, gerente).

**uid:** Identificador único del usuario.

**estado:** Indica si el usuario puede trabajar o no (activo inactivo)

Si está inactivo, no podrá iniciar sesión.

## Implementación

La implementación de la aplicación web se ha realizado utilizando React, desarrollando componentes para que esté todo bien organizado y sea más fácil de mantener y ampliar la aplicación en un futuro.

El diseño de la web está formado con Bootstrap, componentes de Bootstrap y CSS3. En archivos separados para mejorar la organización del código, además cargando solamente un archivo css donde se encuentran los imports de los demás.

Las funciones con acceso a la base de datos, gran parte de estas, están enfocadas a la gestión de las comandas lo cual conlleva que haya usado el método `onSnapshot()` para obtener actualizaciones en tiempo real del documento y no tener que recargar la página. También hay acceso a documentos normalmente sin el método `onSnapshot()` para estar escuchando el documento, usado para la carga de la carta por ejemplo.

Se han usado métodos de escritura en la base de datos también necesarios para la gestión de las comandas, edición de la carta y la gestión de usuarios y mesas.

La conexión con la base de datos es simple, se ha creado un archivo de configuración `firebase.js` donde se encuentran las credenciales de acceso a la base de datos del proyecto y simplemente se inicializa la app de firebase creando así la conexión.

## Evaluación y prueba

El testeo de la aplicación se ha realizado en gran parte por mi, en busca de cualquier error posible.

Pruebas unitarias:

- Carga de renderizado de componentes, respuesta a eventos etc..

Pruebas de funcionamiento de la aplicación:

- Verificación de entradas del usuario con la app, validaciones.

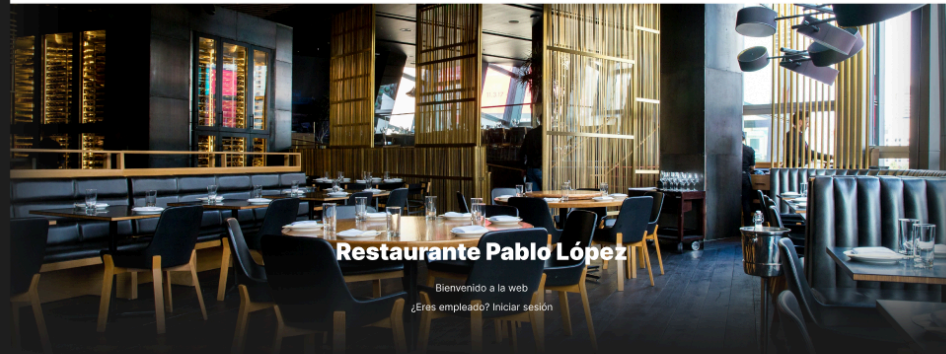
Pruebas para el rendimiento de la web con LightHouse

A parte de mis testeos de la aplicación web, dí a probar la web a familiares y amigos.

## Manual de estilos

*a. Sketches*

Página inicio



## Restaurante Pablo López

Bienvenido a la web  
¿Eres empleado? Iniciar sesión



### Nuestros mejores platos

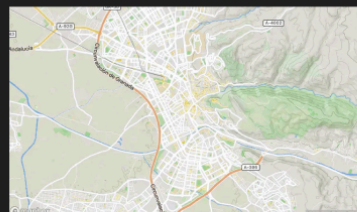
Descubrir

Descubrir

Descubrir


### Contacta con nosotros

Dirección: C/ Recogidas, Granada, España  
Teléfono: +34 900 00 00 00  
Email: [piopezlozano12@gmail.com](mailto:piopezlozano12@gmail.com)



## Inicio Sesión

Desktop - 2

CartaInicia sesión

### Inicio de Sesión


¿Eres un empleado y aún no tienes tu cuenta? [Contacta con tu administrador](#)

[¿Has olvidado tu contraseña? Recuperar Aquí](#)

Iniciar sesión

## Carta

Desktop - 3

CartaInicia sesión

### Carta

Entrantes

^

Pizzas

^

Pescados

^

Carnes

^

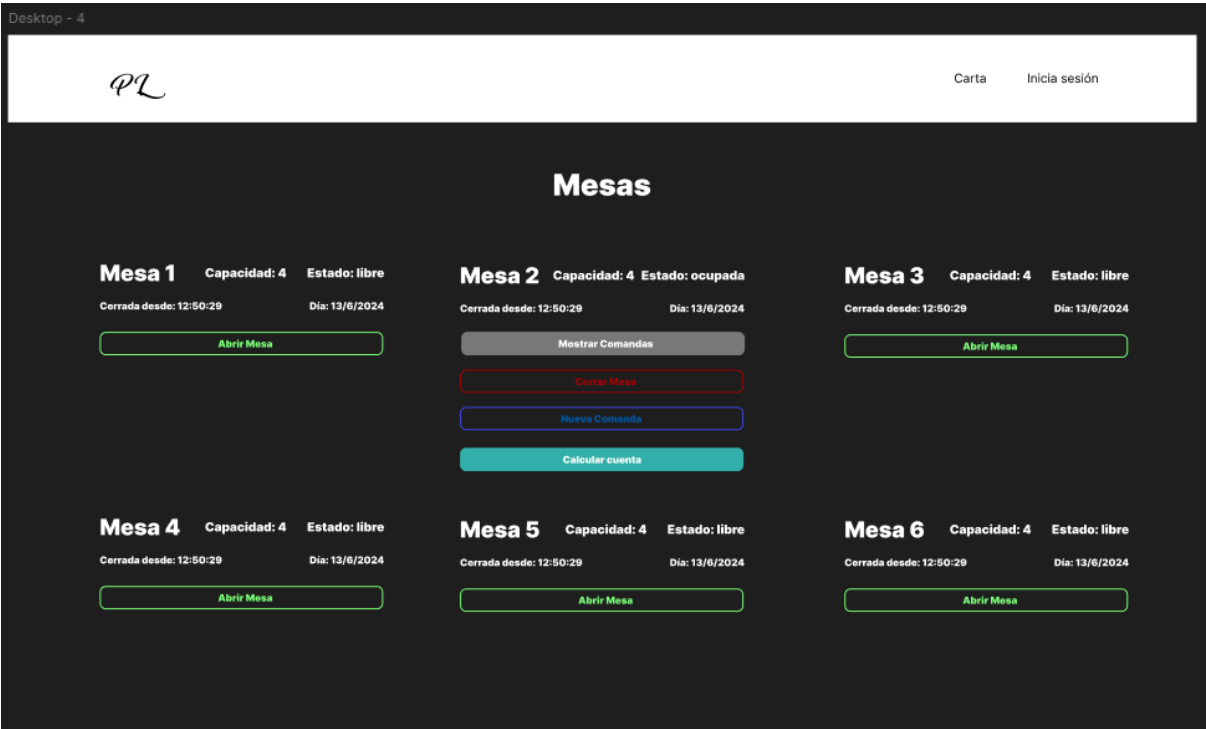
Bebidas

^

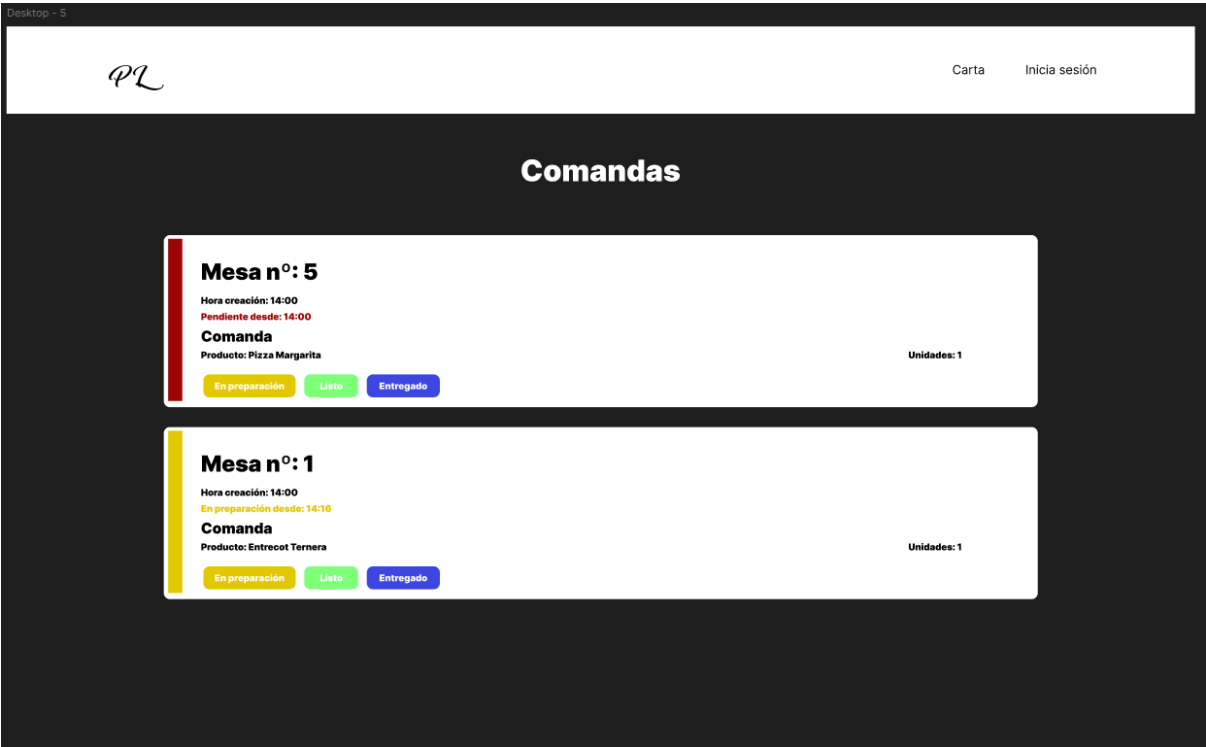
Postres

^

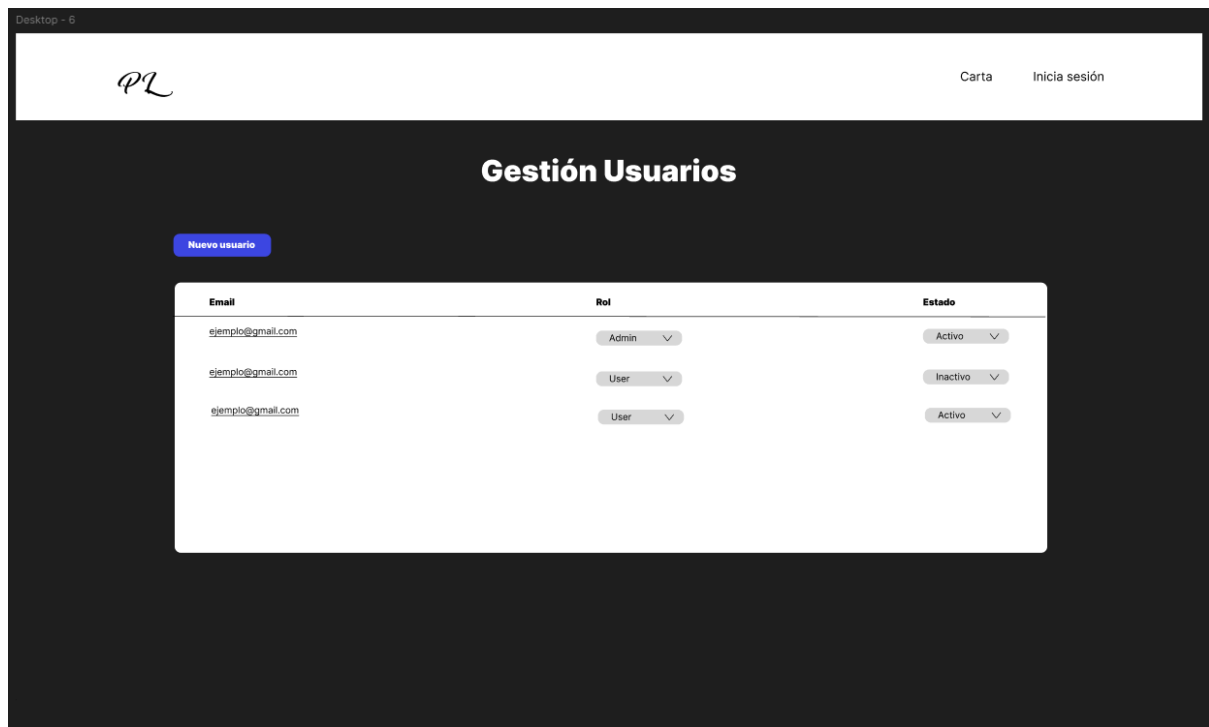
Mesas



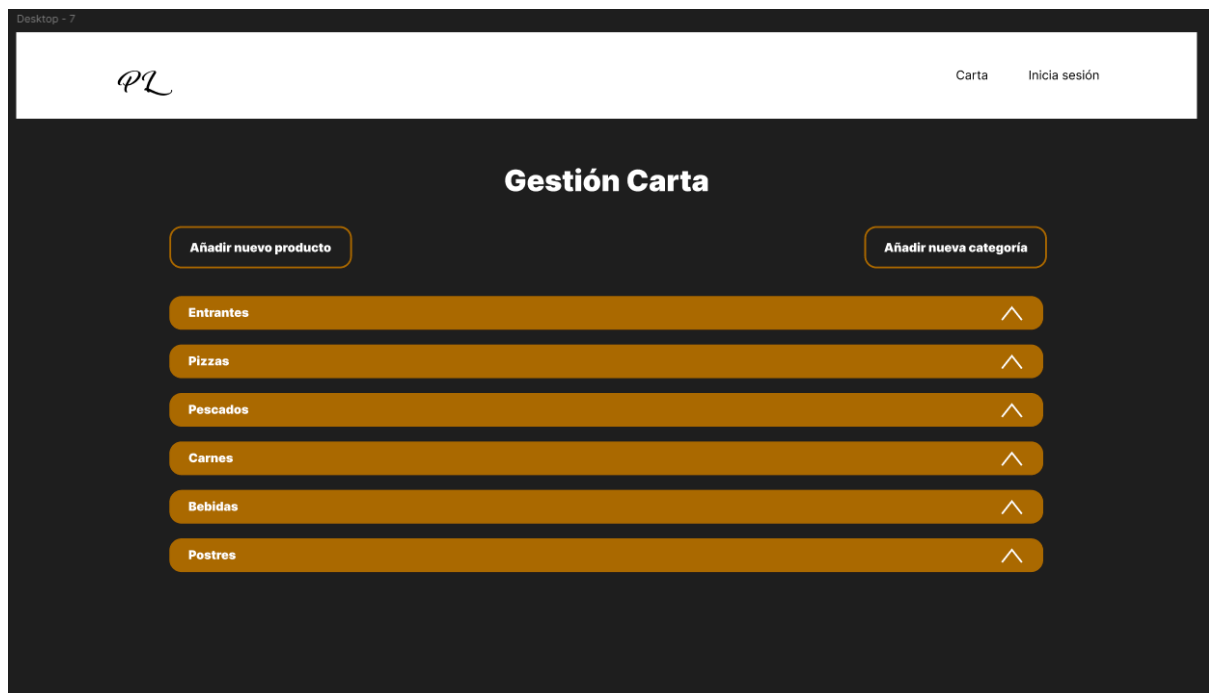
Comandas



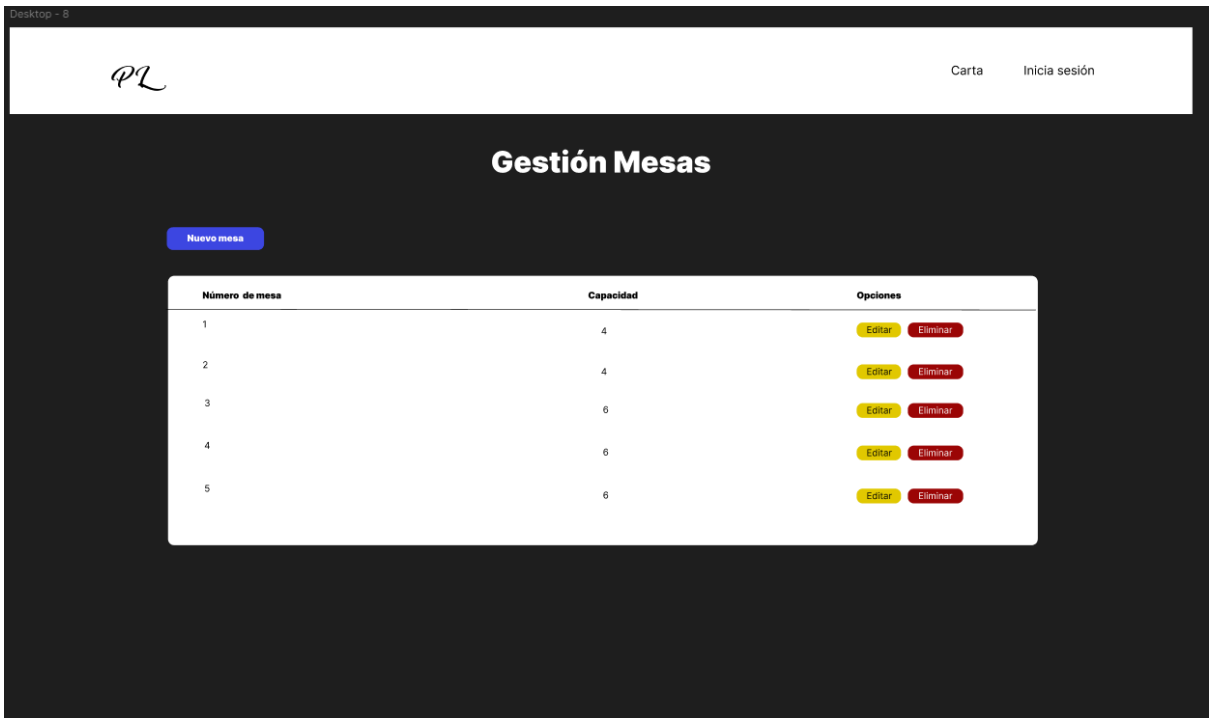
## Gestión Usuarios



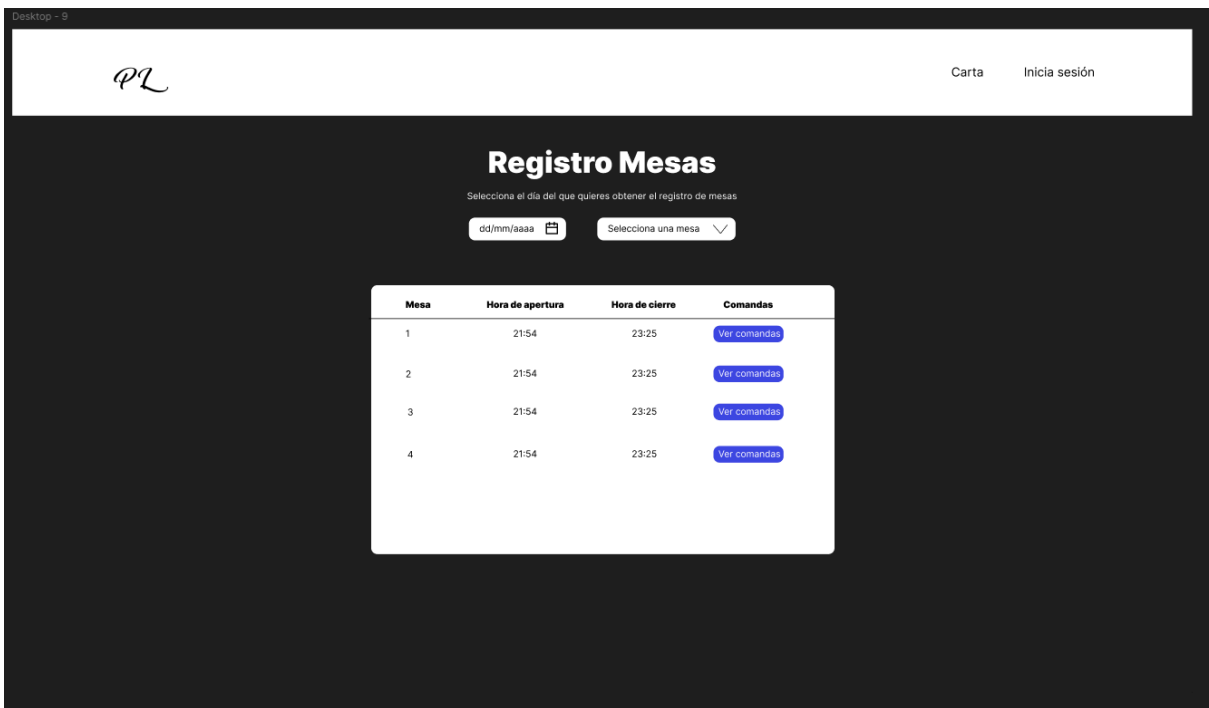
## Gestión Carta



## Gestión Mesas



## Registro Mesas





*b. Criterios de accesibilidad*

- Uso de colores de contraste con el fondo
- Navegación a través del teclado
- Texto alternativo en las imágenes

*c. Criterios de usabilidad*

- Responsive
- Diseño sencillo y claro

*d. Tipografía*

Teachers (<https://fonts.google.com/specimen/Teachers>)

Diseño simple, bonito y legible para toda la aplicación

Playwrite IS (<https://fonts.google.com/specimen/Playwrite+IS>)

Diseño más estilizado, bonito para título

e. Mapa de colores de proyecto en tres formatos (RGB, Hexadecimal, nombre junto con el color).



HEXADECIMAL	RGB	HSL	Elementos donde se usa
AD6903	rgb(173,105,3)	hsl(36, 97%, 35%)	Botones, resaltes, efectos...
#1E1E1E	rgb(30,30,30)	hsl(0, 0%, 12%)	Color fondo
#FFFFFF	rgb(255, 255, 255)	hsl(0, 0%, 100%)	Nombre white, color letra y color fondo secundario, menú header
#0D6EFD	rgb(13,110,253)	hsl(216, 98%, 52%)	Botones secundarios
#F7EDCA	rgb(247,237,202)	hsl(47, 74%, 88%)	Color terciario, fondo de la carta

*f. Dispositivos/vistas para las que se ha diseñado el proyecto .*

- Ordenador
- Portátil
- Tablet
- Móvil

Se ha usado Bootstrap y CSS3 para un diseño responsive adaptado a varios tamaños.

## **Software utilizado**

### *Desarrollo:*

React: usado para el desarrollo de la aplicación

JavaScript: lenguaje de programación utilizado

### *Diseño*

Bootstrap: clases de bootstrap y componentes para un diseño atractivo y responsive

CSS3: Adaptaciones de bootstrap y creación de algunos diseños

### *Funcionalidades*

React-pdf: Librería usada para generar pdfs con react. En mi caso usada para generar las facturas de los clientes de cada mesa

### *Back-end*

Firebase Store: Base de datos donde se almacena archivos que utilizo en la aplicación a parte de las fotos de los platos de la carta

Firebase Database: Base de datos no relacional para guardar toda la información de la gestión del restaurante en tiempo real.

Firebase Auth: Autenticación de usuarios segura y fácil con verificación por correo y recuperación de contraseña

## **Mejoras posibles y aportaciones**

Mejoras en funcionalidad: sistema de reservas integrado en la aplicación para los clientes.

Organización de código

Organización de la información