

Projecte Final de Cicle

# MesaYa



<b>Alumne:</b>	Jose Adrián Aglio Gascón
<b>DNI:</b>	23855131Z
<b>Tutor Individual:</b>	Jordi Cidoncha Navarrete
<b>Tutor del grup:</b>	Jose Alfredo Murcia Andres



**IES Jaume II El Just**  
Tavernes de la Valldigna

# Dades del Projecte

## Dades de l'alumne

Nom i cognoms	Jose Adrián Aglio Gascón
NIF/NIE	23855131Z
Curs i CF	2n DAM

## Dades del projecte

Títol del projecte	MesaYa
Nom del tutor individual	Jordi Cidoncha Navarrete
Nom del tutor del grup	Jose Alfredo Murcia Andres
Resum	<i>Una aplicació en la que pugues reservar taules en restaurants o altre establiment, y també tindre una llista de locals filtrats per categoria i resenyes positives, per a que pugues triar el millor local per a tu..</i>
Abstract	<i>An application where you can book a table in restaurants or another establishment, and also have a list of premises filtered by category and by positive reviews, so you can choose the best site for you.</i>
Mòduls implicats	<ul style="list-style-type: none"><li>● PMDM</li><li>● PSP</li><li>● AD</li></ul>
Data de presentació	3 de juny de 2025

# Índex

Dades del Projecte.....	1
Dades de l'alumne.....	1
Dades del projecte.....	1
Índex.....	2
1. Introducció/Marc del projecte.....	3
1.1. Descripció del projecte.....	3
1.2. Objectius.....	3
1.3. Tipus de projecte.....	3
1.4. Orientacions per al desenvolupament i recursos.....	3
2. Anàlisi de Requeriments.....	8
2.1. Estudi de l'estat de l'art/Estudi de mercat.....	8
2.2. Definició de requeriments.....	8
3. Metodologia de Desenvolupament.....	10
3.1. Descripció de la metodologia àgil.....	10
3.2. Backlog del producte.....	11
3.3. Planificació temporal.....	11
4. Desenvolupament iteratiu per sprints.....	11
4.1. Sprint 1: Crear Backend.....	11
4.2. Sprint 2: Autenticació i Configuració MySQL.....	13
4.3. Sprint 3: Frontend Bàsic i API Local.....	16
4.4. Sprint 4: Implementar funcions.....	18
4.5. Sprint 5: Ajustos finals.....	19
5. Conclusions i Treballs Futurs.....	21
5.1. Resum dels resultats.....	21
5.2. Possibles millores.....	22

# 1. Introducció/Marc del projecte

## 1.1. Descripció del projecte

El meu projecte és una aplicació amb la funció de poder reservar taula non sols en restaurants, si no en més establiments com bars, hamburgueseries, pizzeries, etc. L'aplicació compta amb un inici de sessió, on les funcions estan limitades si l'usuari no ha iniciat sessió. Des de la pantalla principal, on podràs veure els locals disponibles, ordenats per categoria, valoració o preu, podràs accedir a qualsevol restaurant i reservar una taula. També inclou un apartat de cercador amb filtres per a facilitar l'elecció del millor lloc segons els gustos de l'usuari, además de tindre una pantalla per a vore les teues reserves, com per a vore els teus restaurants favorits. Aixina podràs accedir als restaurants d'una manera rapida i sencilla.

## 1.2. Objectius

- Permetre a l'usuari iniciar sessió i registrar-se.
- Crear una API amb la informació dels establiments.
- Poder consultar els establiments.
- Reservar taula en els establiments disponibles.
- Poder cancel·lar reserves.
- Guardar establiments com a favorits.
- Filtrar establiments per categoria, valoració y preu.

## 1.3. Tipus de projecte

- **Extern:** Desenvolupament per a tercers, porque l'aplicació esta pensat per a persones independents.

## 1.4. Orientacions per al desenvolupament i recursos

El desenvolupament ha sigut iteratiu, o s'ha utilitzat la metedología Scrum.

Les tecnologies utilitzades son:

- **Frontend**

Per al desenvolupament del frontend s'ha utilitzat Flutter, un framework creat per Google que permet crear aplicacions mòbils amb una sola base de codi. Utilitza el llenguatge Dart i destaca pel seu sistema de widgets i la funcionalitat hot reload, que facilita el desenvolupament ràpid i visual de la interfície d'usuari.



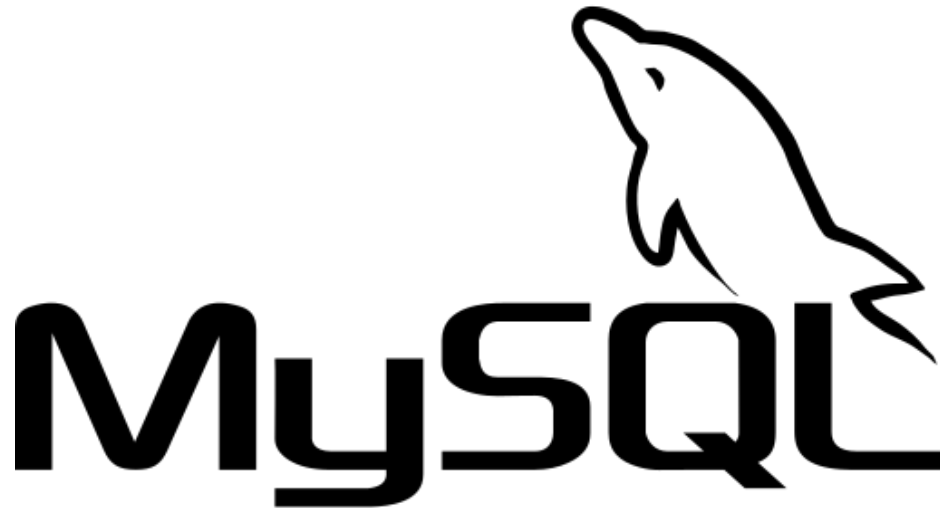
- **Backend**

S'ha utilitzat Node.js juntament amb Express, un entorn d'execució i un framework minimalista que permeten crear un servidor web lleuger per gestionar les rutes, l'autenticació i la connexió amb la base de dades.



- **Base de dades**

Per a emmagatzemar la informació, s'ha utilitzat MySQL, un sistema de gestió de bases de dades relacional, molt utilitzat per la seua robustesa i facilitat d'ús.



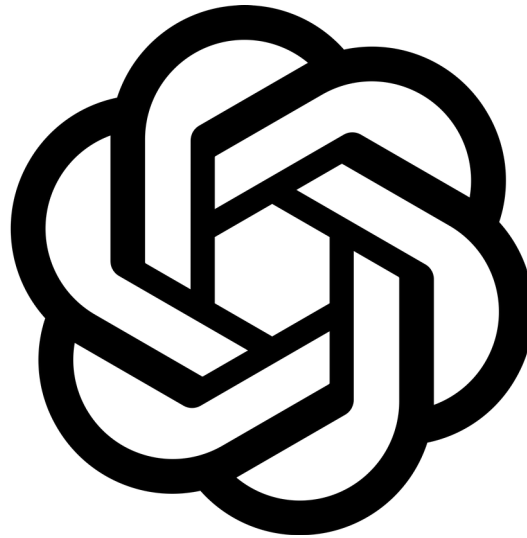
- **Control de versions**

S'ha utilitzat GitHub per a portar un control de versions del projecte, gestionar el codi i treballar de forma segura i ordenada.



- **Fonts d'ajuda**

- **ChatGPT:** ha sigut utilitzat per a resoldre dubtes puntuals durant el desenvolupament, redactar parts del projecte i generar el logo de l'aplicació.



- **DesarrolloWeb.com:** ha sigut una font de consulta útil per trobar mòduls i exemples relacionats amb Node.js i Express.



desarrolloweb.com

- **Eines de disseny i diagrama**

- **Mermaid:** Per a representar visualment el funcionament de l'aplicació (esquema casos de ús i diagrama de Gantt), s'ha utilitzat Mermaid, una eina que permet generar diagrames de manera senzilla mitjançant codi.



- **Draw.io:** S'ha fet ús de draw.io per dissenyar els esquemes visuals i pantalles de l'aplicació abans d'implementar-les amb Flutter. Aquesta eina ha facilitat la planificació de la interfície.



draw.io



## 2. Anàlisi de Requeriments

### 2.1. Estudi de l'estat de l'art/Estudi de mercat

Existixen aplicacions com TheFork, que s'utilitza per a cercar i reservar en restaurants, però està limitat a restaurants formals y en zones específiques. Lo que vullc buscar amb MesaYa es ser més inclusiu amb xicotets locals i oferir una experiència personalitzada amb els filtres i també donar la opció de que la aplicació trie per tu amb recomanacions aleatòries.

### 2.2. Definició de requeriments

- Requeriments funcionals i no funcionals.

#### **Requeriments funcionals:**

- Registre i inici de sessió.
- Visualització de locals amb detalls.
- Filtres per categoria, valoració y preu.
- Sistema de favorits.
- Gestió de reserves.

#### **Requeriments no funcionals:**

- Interfície accessible.
- Alta disponibilitat i resposta ràpida.
- Seguretat de dades d'usuari.

- Casos d'ús i històries d'usuari.

#### **1. CU-1: Registrar-se a l'aplicació**

- Actor: Usuari nou.
- Descripció: L'usuari accedeix a l'aplicació i es registra introduint les dades necessàries.

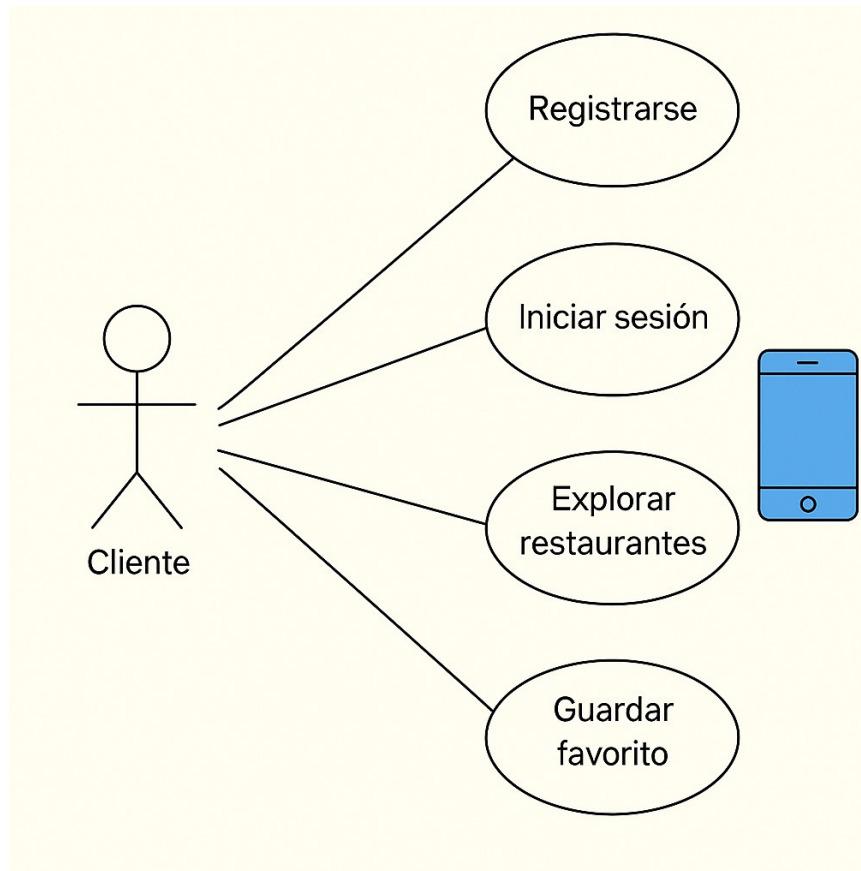
## 2. CU-2: Cercar i filtrar locals

- Actor: Usuari registrat.
- Descripció: L'usuari pot cercar locals (restaurants, bars, pizzeries, etc.) mitjançant un buscador i aplicar filtres per categoria, valoració o preu.

## 3. CU-3: Fer una reserva

- Actor: Usuari registrat.
- Descripció: L'usuari selecciona un local, tria data, hora i nombre de persones, i confirma la reserva.

Funcionalitat	Història	Prioritat
Visualitzar detalls del establiment	Com a client m'agradaria poder vore els detalls del establiment	Alta
Reservar taula	Com a client m'agradaria poder reservar una taula en qualsevol establiment	Alta
Marcar un establiment com a favorit	Com a client m'agradaria poder guardar un establiment en una llista de favorits	Mitja



## 3. Metodologia de Desenvolupament

### 3.1. Descripció de la metodologia àgil

He utilitzat la metodologia Scrum amb sprints semanals, per així organitzar el treball d'una manera àgil i eficient:

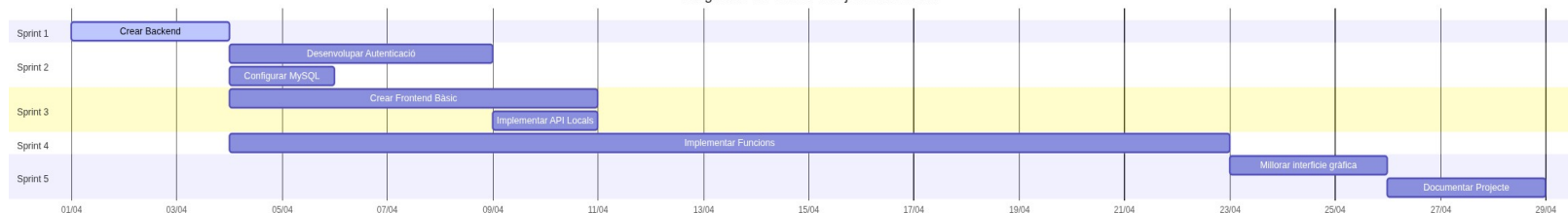
- Sprints: 2 setmanes.
- Eines: GitHub Projects per a gèstio de tasques.
- Reunions: Diàries, revisant que tinc que fer y problemes a resoldre i a millorar.

### 3.2. Backlog del producte

Prioritat	Funcionalitat	Temps estimat
Alta	Registre i login	2 dies
Alta	Llista de locals	Mitja setmana
Mitjana	Filtres per categoria, valoració i preu	2 dies
Mitjana	Favorits	Mitja setmana
Alta	Reserves	1 setmana
Alta	Backend	1 setmana

### 3.3. Planificació temporal

Diagrama de Gantt - Projecte MesaYa



## 4. Desenvolupament iteratiu per sprints

El projecte s'ha organitzat en 5 sprints seguint el diagrama de Gantt proporcionat, amb una durada total de 34 dies de desenvolupament actiu. A continuació es detalla cada fase:

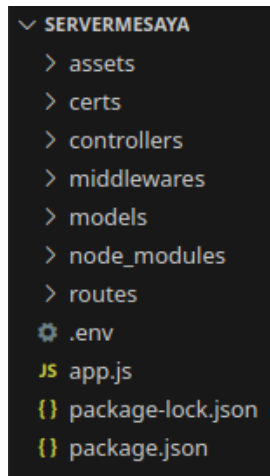
### 4.1. Sprint 1: Crear Backend

- Disseny de la solució

- Creació del servidor amb Node.js y Express.
- Definició d'endpoints.
- Desenvolupament dels fitxers necessaris per al servidor.

- Implementació

- Estructura inicial del projecte.



- Configuració del servidor HTTPS.

```
// Definim app com a una aplicació Express
const app = express();

// Creem el servidor HTTPS amb les credencials i l'aplicació Express
const httpsServer = https.createServer(credentials, app);

// Escoltar al port 3000
httpsServer.listen(3000, () => {
  console.log('Servidor HTTPS en execució a https://localhost:3000/');
});
```

- Configuració de Middleware per a JSON.

```
// Incorporem el middleware per gestionar sol·licituds JSON
app.use(express.json());

app.use('/auth', authRouter);

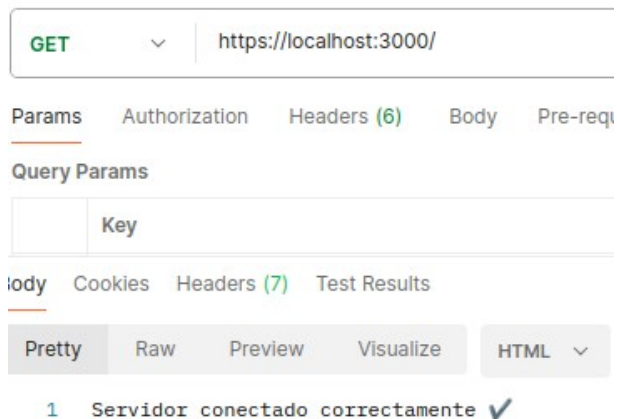
app.use('/reservas', reservas);

app.use('/favoritos', favoritos);

app.use('/restaurantes', restaurantes);
```

## ● Proves

- Verificació de resposta dels endpoints amb Postman.



## 4.2. Sprint 2: Autenticació i Configuració MySQL

- Disseny de la solució

- Implementar seguretat JWT.
- Crear script de MySQL.
- Configuració del servidor per implementar MySQL.

- Implementació

- Codi implementant JWT.

```
import jwt from 'jsonwebtoken';
import { registerUser, authenticateUser } from '../models/users.js';

// Llibreria per a les variables d'entorn
import dotenv from 'dotenv';

// Aquesta és la forma correcta d'accedir a les claus secretes
dotenv.config();
const SECRET_KEY = process.env.SECRET_KEY;
```

- Sistema de registre i login.

```
// Registre d'usuari
export const register = async (req, res) => {
  const { username, password } = req.body;
  if (!username || !password) {
    return res.status(400).send({ error: 'Nom d\'usuari i contrasenya requereix' });
  }
  const newUser = await registerUser(username, password);
  res.status(201).send(newUser);
};

// Login d'usuari
export const login = async (req, res) => {
  console.log("Recibido login para:", req.body);
  const { username, password } = req.body;
  if (!username || !password) {
    console.log("Error: faltan credenciales");
    return res.status(400).send({ error: 'Nom d\'usuari i contrasenya requereix' });
  }

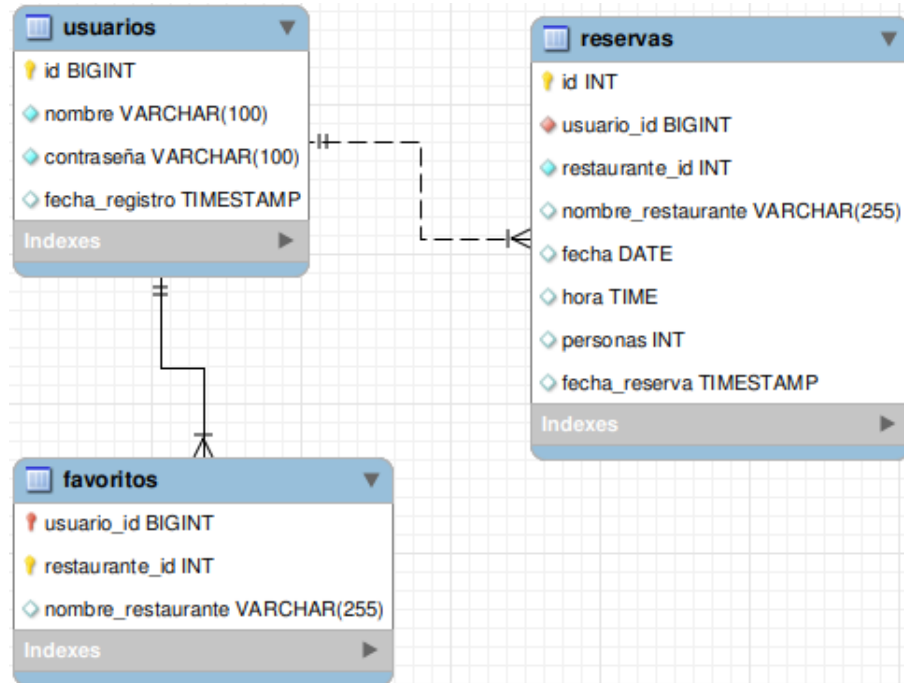
  try {
    console.log("Intentando autenticar:", username);
    const user = await authenticateUser(username, password);
    console.log("Resultado autenticación:", user ? "Exitoso" : "Fallido");

    if (!user) {
      return res.status(401).send({ error: 'Credenciales incorrectes' });
    }

    const token = jwt.sign({
      username: user.username,
      id: user.id
    }, SECRET_KEY, { expiresIn: '1h' });

    res.send({ token, id: user.id });
  } catch (err) {
    console.error("Error en login:", err);
    res.status(500).send({ error: 'Error de servidor' });
  }
};
```

- Taules de la base de dades.



- Conexió servidor-base de dades.

```
import mysql from 'mysql2/promise';

// Configuración de la conexión a la base de datos
const pool = mysql.createPool({
  user: 'root',
  host: 'localhost',
  database: 'MesaYa',
  password: 'root',
  port: 3307,
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0
});

export { pool };
```

- **Proves**

- Proves d'integració frontend-backend.

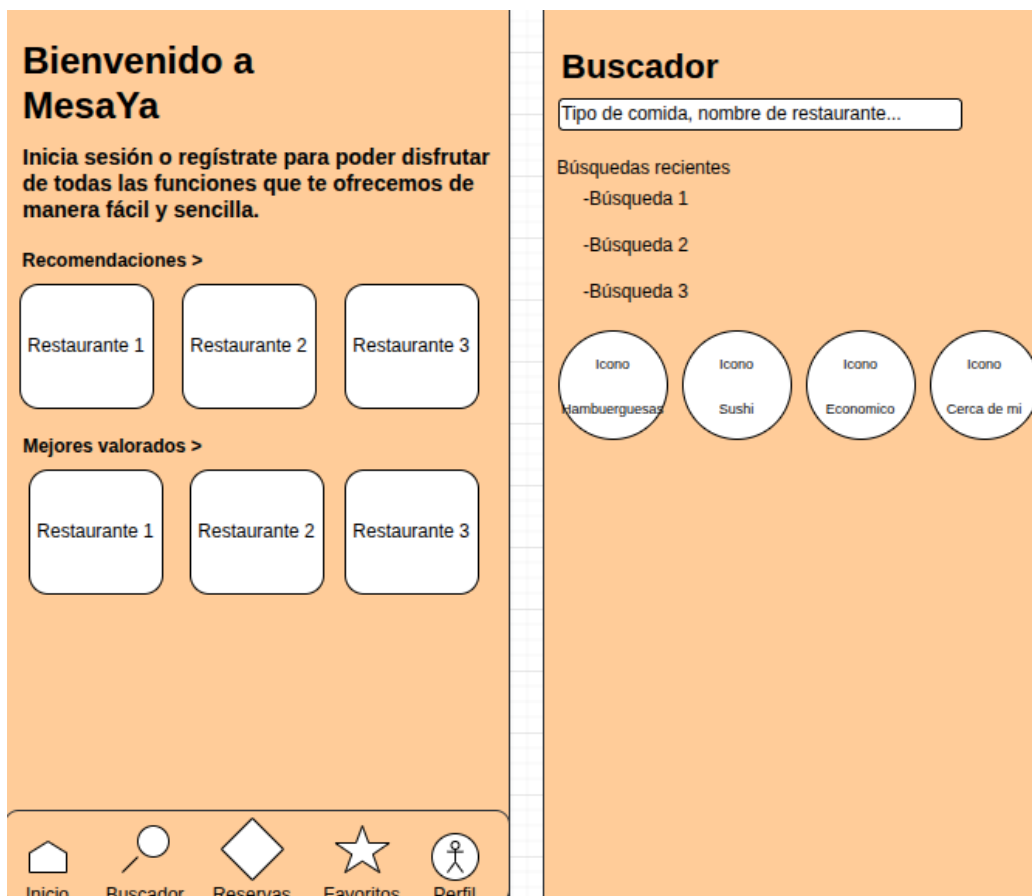
## 4.3. Sprint 3: Frontend Bàsic i API Local

- **Disseny de la solució**

- Dissenyar la part gràfica de l'aplicació.
- Programació de les pantalles principals.
- Crear API.

- **Implementació**

- Disseny gràfic.





### Reservas

#### Proximas Reservas

Restaurante 1  
Fecha/Hora  
Num. Personas

#### Historial Reservas

Restaurante 1  
Fecha/Hora  
Num. Personas  
Comentario/Valoración

### Favoritos

Restaurante 1  
Dirección  
Precio  
Estimado

### Mi Perfil

Nombre

Editar Perfil

Cerrar sesión

- **Proves**

- Proves unitàries dels components de Flutter.
- Validació dels endpoints amb diferents paràmetres.

## 4.4. Sprint 4: Implementar funcions

- **Disseny de la solució**

- Ampliació i creació de fitxers amb les funcions (reservar, cercar, marcar com favorit, etc).
- Modificació i millora del backend.

- **Implementació**

- Integració amb Google Maps.

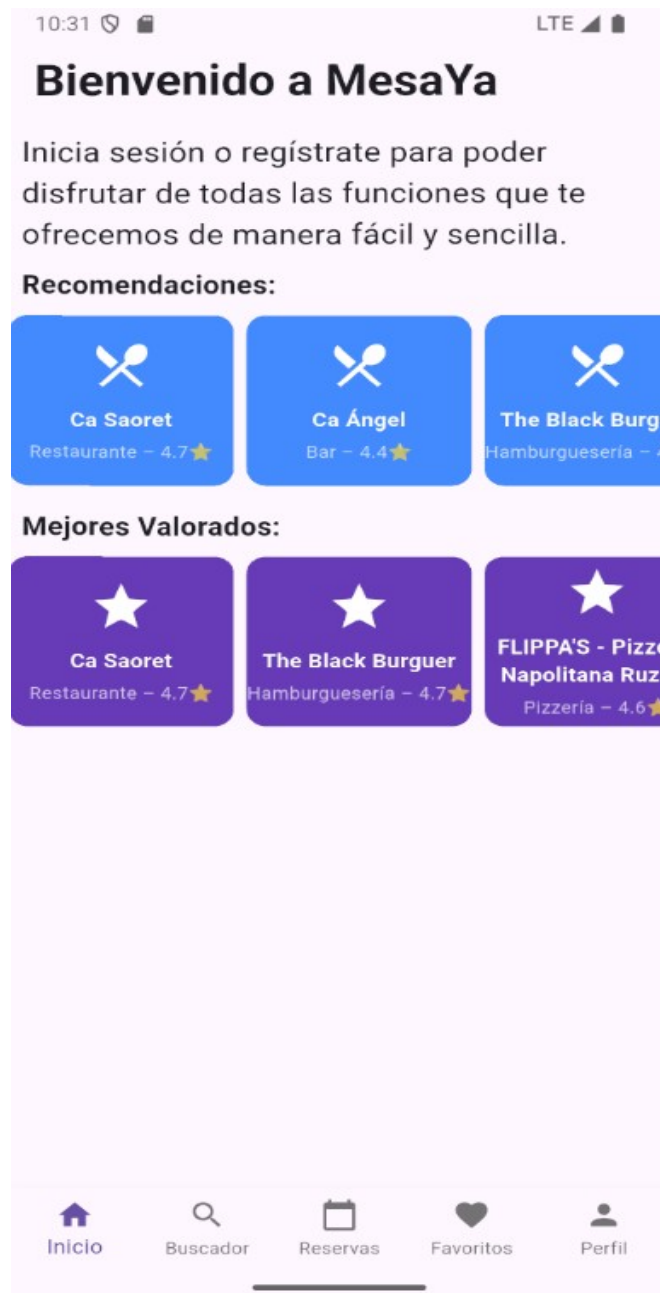
```
void _abrirEnMapa(double latitud, double longitud) async {  
  final url =  
    'https://www.google.com/maps/search/?api=1&query=$latitud,$longitud';  
  if (await canLaunchUrl(Uri.parse(url))) {  
    await launchUrl(Uri.parse(url));  
  } else {  
    throw 'No se pudo abrir el mapa';  
  }  
}
```

- **Proves**

- Proves d'integració de tots els mòduls.
- Proves amb múltiples usuaris.
- Refactorització final del codi.

## 4.5. Sprint 5: Ajustos finals

- **Optimització i validació final**
  - Millora del rendiment del backend.
  - Optimització de consultes SQL.
  - Millora del funcionament del frontend.
- **Proves finals**
  - Prova final de l'aplicació



## 5. Conclusions i Treballs Futurs

### 5.1. Resum dels resultats

El resultat del projecte MesaYa ha superat amb èxit els objectius inicials, convertint-se en una aplicació funcional per a la reserva d'establiments gastronòmics. S'ha conseguit que l'aplicació siga més inclusiva que altres aplicacions amb xicotets locals i restaurants, i que siga facil d'utilitzar per a l'usuari, amb una interfície simple i sencilla.

#### ● Implementació tècnica

- **Backend robust:**
  - Arquitectura modular amb Node-js i Express.
  - Sistema d'autenticació segur utilitzant JWT i bcrypt.
  - Connexió eficient amb base de dades MySQL.
  - API REST ben documentada amb 12 endpoints funcionals.
- **Frontend intuïtiu:**
  - Interfície desenvolupada amb Flutter.
  - Cerca avançada amb múltiples filtres (categoria, valoració, preu).
  - Gestió completa de reserves i favorits.

#### ● Aprenentatges clau

- Domini de Node.js i Express.
- Integració efectiva entre Flutter i APIs REST.
- Configuració correcta per a connexions entre frontend i backend.
- Optimització de consultes SQL.
- Gestió de projecte amb metodologies àgils.

- **Dificultats superades**

- Sincronització de l'estat entre frontend i backend.
- Configuració dels fitxers d'autenticació.
- Intercambi d'informació entre frontend, backend i base de dades.

## 5.2. Possibles millores

El projecte té un gran potencial d'expansió amb les següents millores:

- **Sistema de pagaments**

Seria una funció més enfocada al pagament de la reserva en grup, donar opció a que cada client pose la seua part fins arribar al preu del conter del restaurant, i el pagament vaja directament al restaurant.

- **Aplicació d'escriptori per els establiments**

Una aplicació amb funció de administrador per a que els establiments puguem gestionar les seues reserves i pagaments.

- **Notificacions**

Que l'aplicació enviara recordatoris de les reserves, o un mensaje si la reserva a sigut cancelada per el restaurant, per exemple.