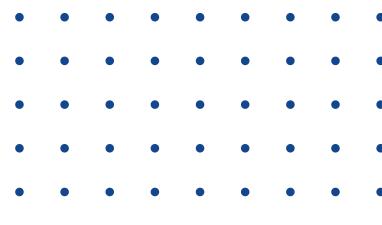
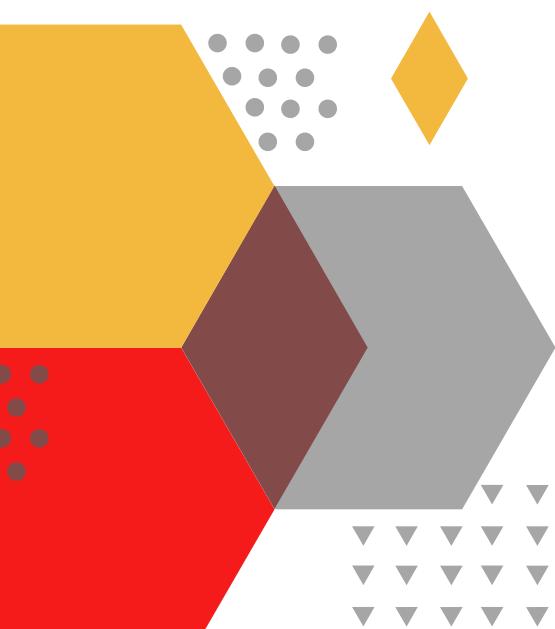


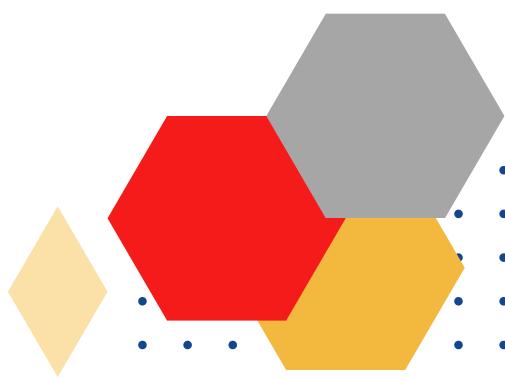
EASYDINING

OFRECIENDO UNA VISIÓN INTEGRAL
MANUAL TÉCNICO



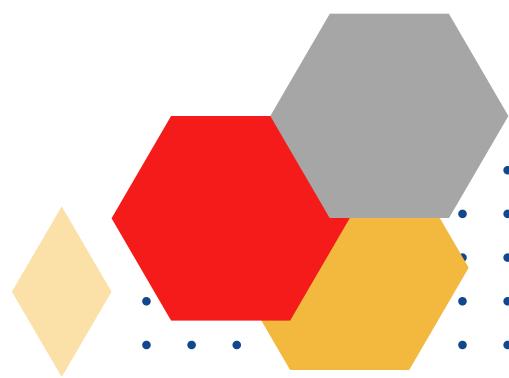
Introducción.

En la actualidad, la industria de la restauración enfrenta múltiples desafíos, entre los que destacan la gestión eficiente de reservas y la optimización de la experiencia del cliente. Estos desafíos han cobrado mayor relevancia en un entorno cada vez más digitalizado, donde los usuarios esperan soluciones tecnológicas que les faciliten el acceso a servicios cotidianos, como la reserva de mesas en restaurantes. A pesar de la creciente demanda de herramientas digitales, muchos restaurantes aún no cuentan con un sistema de reservas efectivo que se alinee con las expectativas de sus clientes.



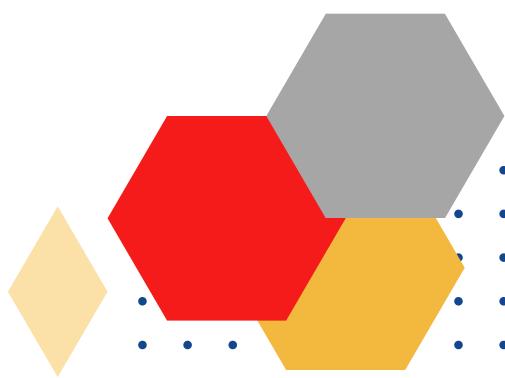
El proyecto "EasyDining" surge como una respuesta a esta necesidad, proponiendo el desarrollo de una aplicación móvil que permita a los usuarios reservar mesas en una variedad de restaurantes, seleccionando tanto la fecha como la hora deseada, y ofreciendo una visión integral de las opciones disponibles en función de la ubicación y las calificaciones proporcionadas por otros usuarios. Esta solución no solo mejorará la experiencia del cliente, sino que también optimizará la gestión operativa de los restaurantes, permitiendo una mejor planificación y aprovechamiento de los recursos.

La metodología de desarrollo de "EasyDining" se basará en enfoques ágiles, utilizando Scrum para gestionar las tareas de manera eficiente y colaborativa. La aplicación será desarrollada en Kotlin, un lenguaje moderno y robusto para aplicaciones Android, asegurando una experiencia de usuario fluida y receptiva



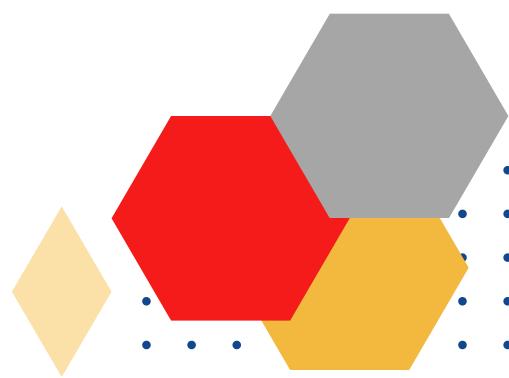
Índice.

- Objetivo
- Inicio
- Registrar
- Menu
- Reservaciones
- Buscador

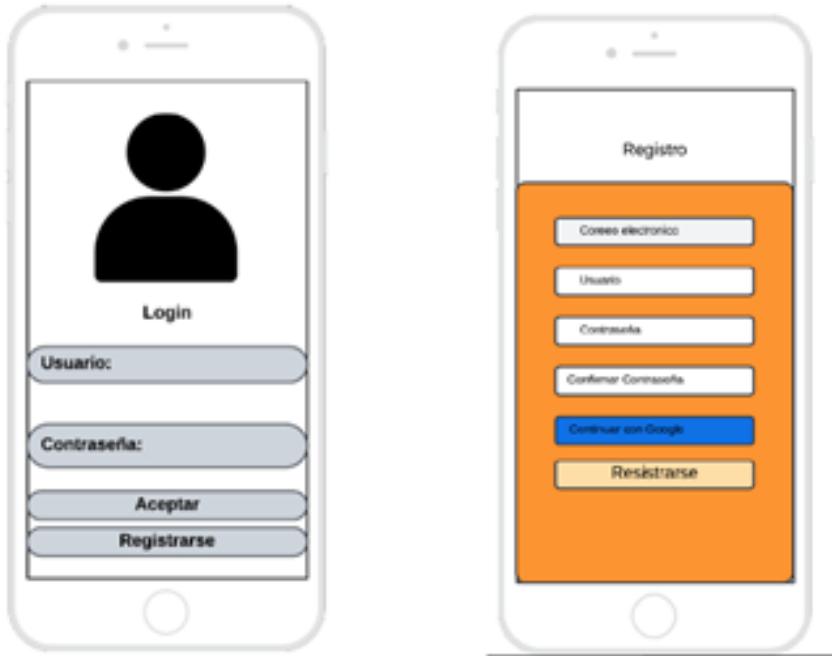


OBJETIVO:

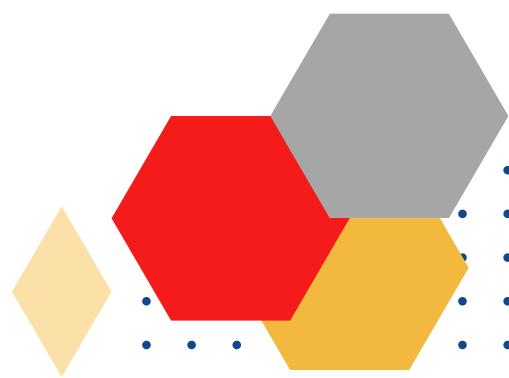
Desarrollar una aplicación móvil, “EasyDining”, que permita a los usuarios realizar reservas de mesas en restaurantes de manera intuitiva y eficiente, con funcionalidades que incluyen la selección de fechas y horarios, visualización de la ubicación de los restaurantes en un mapa interactivo, y acceso a calificaciones y opiniones de otros usuarios.



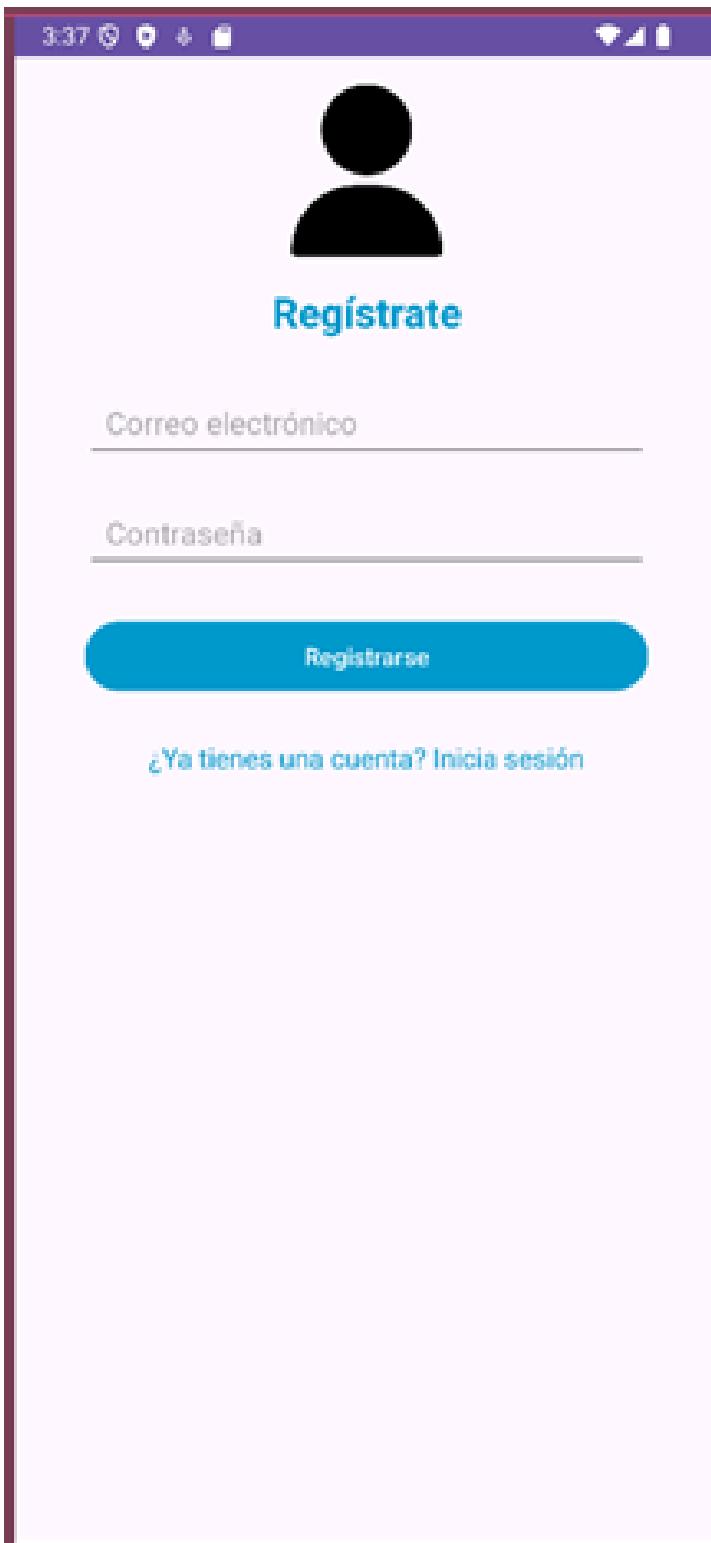
Inicio.



Tendremos un apartado donde el usuario podrá registrarse si es primera vez, pero si ya se registró y lo hizo con el correo el iniciara sesión y entrara a la aplicación



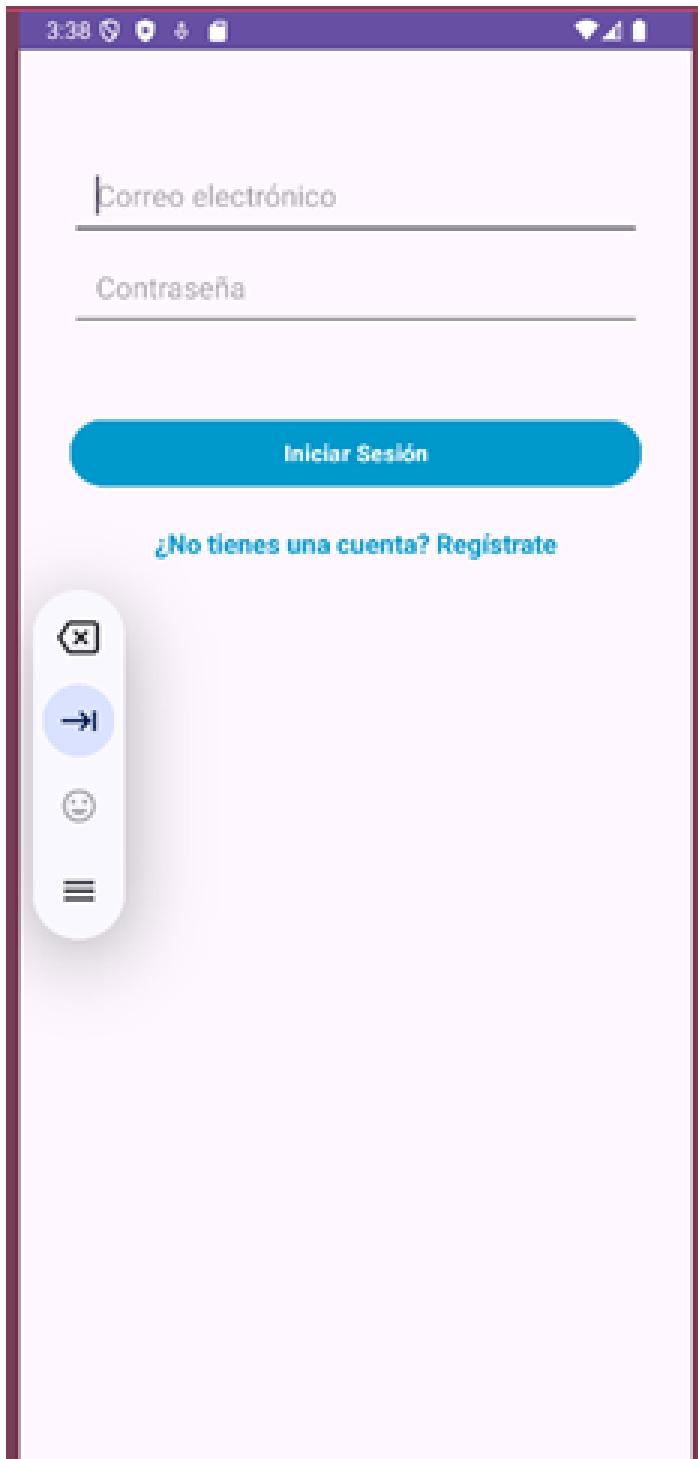
Registrar.



Al entrar a la app podremos ver este apartado en donde nos pedirá un correo electronico y nos pedirá crear una contraseña.

Al terminar podremos continuar a dar click en “Reistrate”.

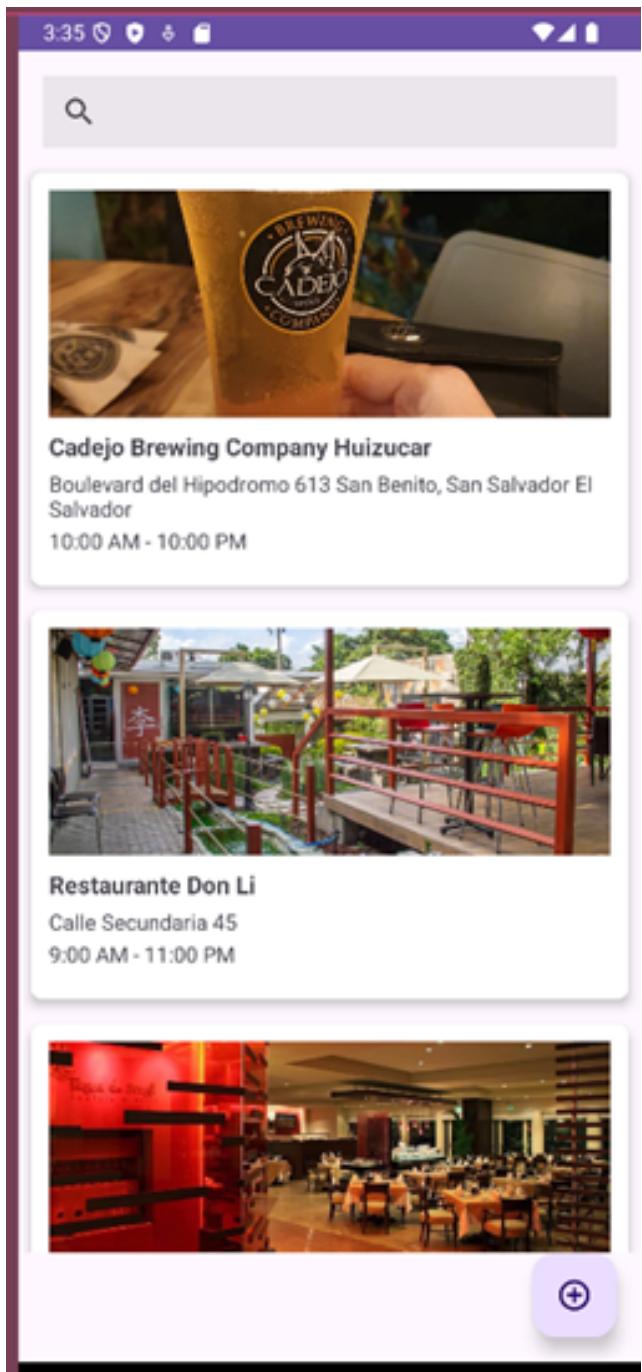
Inicio.



Aqui podemos encontrar la pantalla en la cual podremos iniciar sesión si ya se tiene una cuenta creada.

Como podremos observar en la parte inferior existe el apartado de registrarse si aún no se tiene una cuenta previa.

Menú.



Al iniciar sesión o registrarse, encontramos el menu en donde podemos ver los restaurantes disponibles y el horario en que estos trabajan.

se encuentra un botón por la parte inferior, este es en el que podemos realizar una reservación.

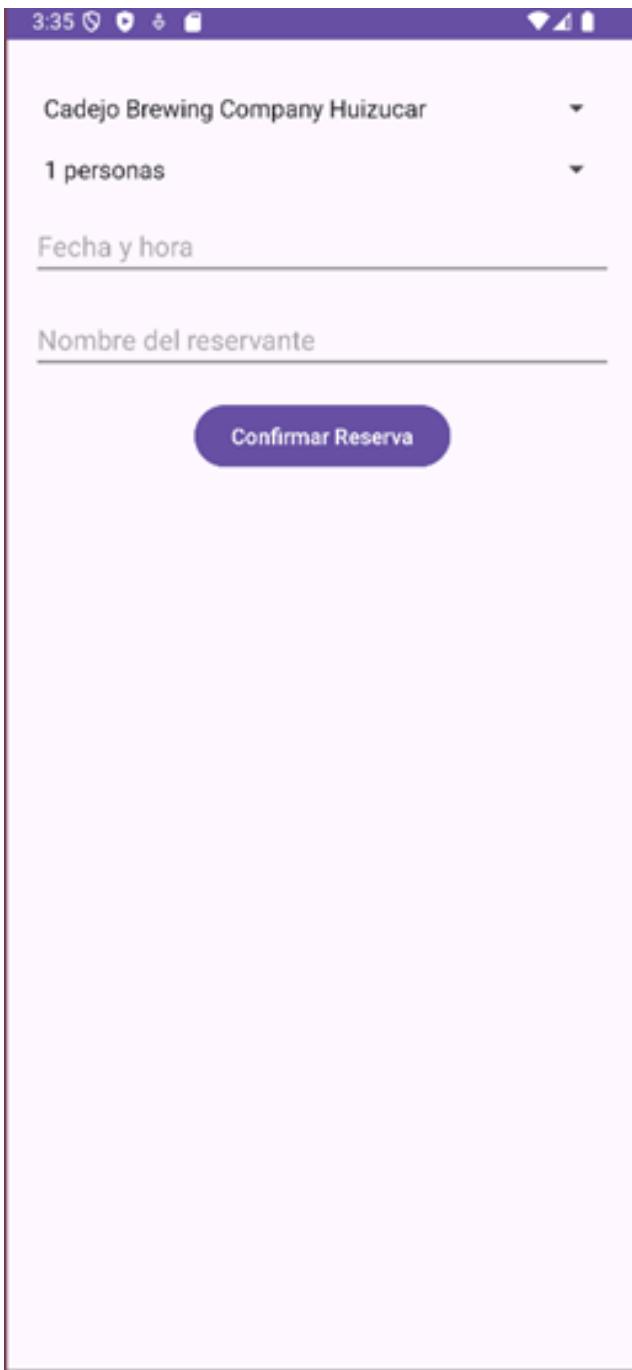
Menú

Los restaurantes se encuentran almacenados en un API, esta se puede observar que se encuentra en el apartado de RetrofitClient.kt

```
object RetrofitClient {  
    private const val BASE_URL =  
        "https://6701b49fb52042b542d86557.mockapi.io/"  
  
    val retrofit: Retrofit = Retrofit.Builder()  
        .baseUrl(BASE_URL)  
  
        .addConverterFactory(GsonConverterFactory.create())  
        .build()  
  
    val apiService: ApiService =  
        retrofit.create(ApiService::class.java)  
}
```



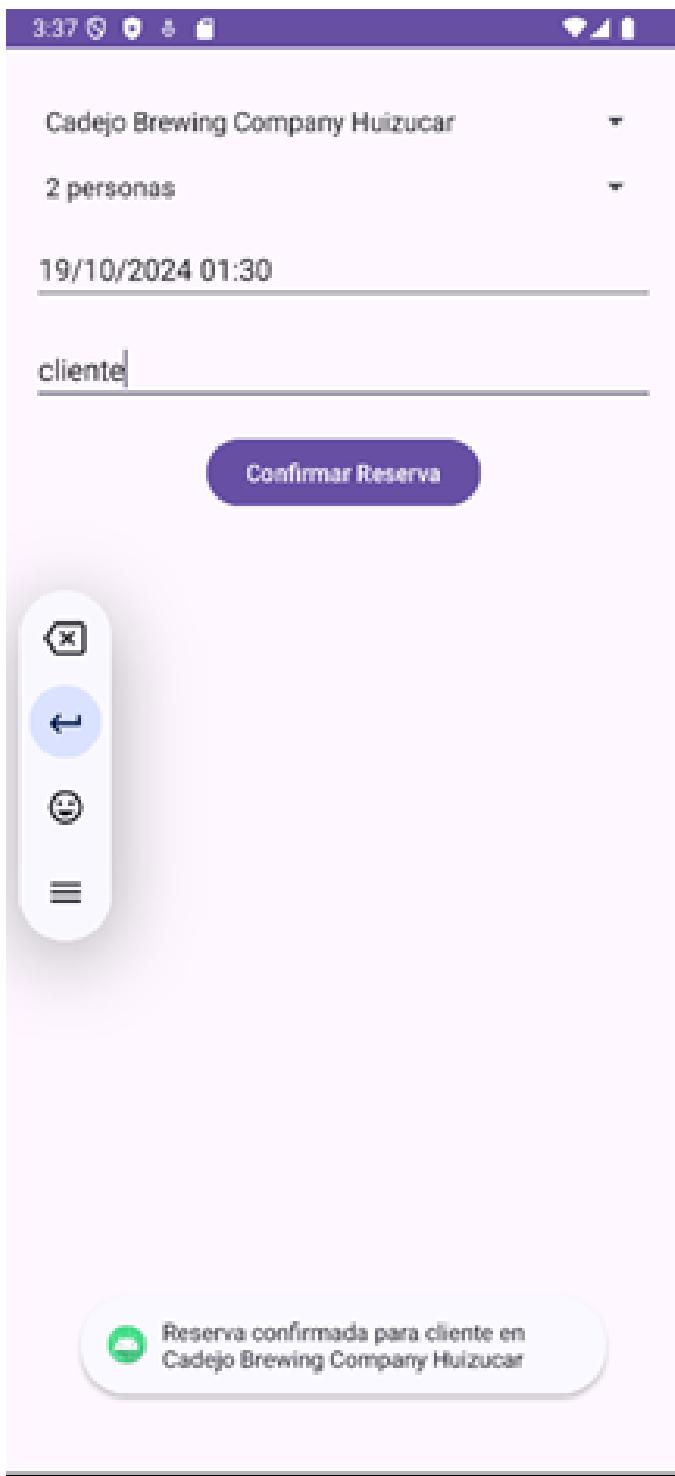
Reserva



Cuando hacemos click en el boton, nos lleva a este apartado en el cual se haran las reservas en su restaurante favorito.

Podemos encontrar el codigo en funcion en el apartado de ReservaActivity.kt

Confirmación de reserva

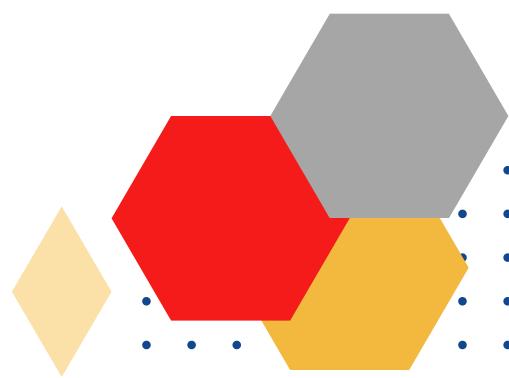


En la parte inferior de nuestro móvil podemos observar la notificación de que la reserva se ha hecho exitosamente.

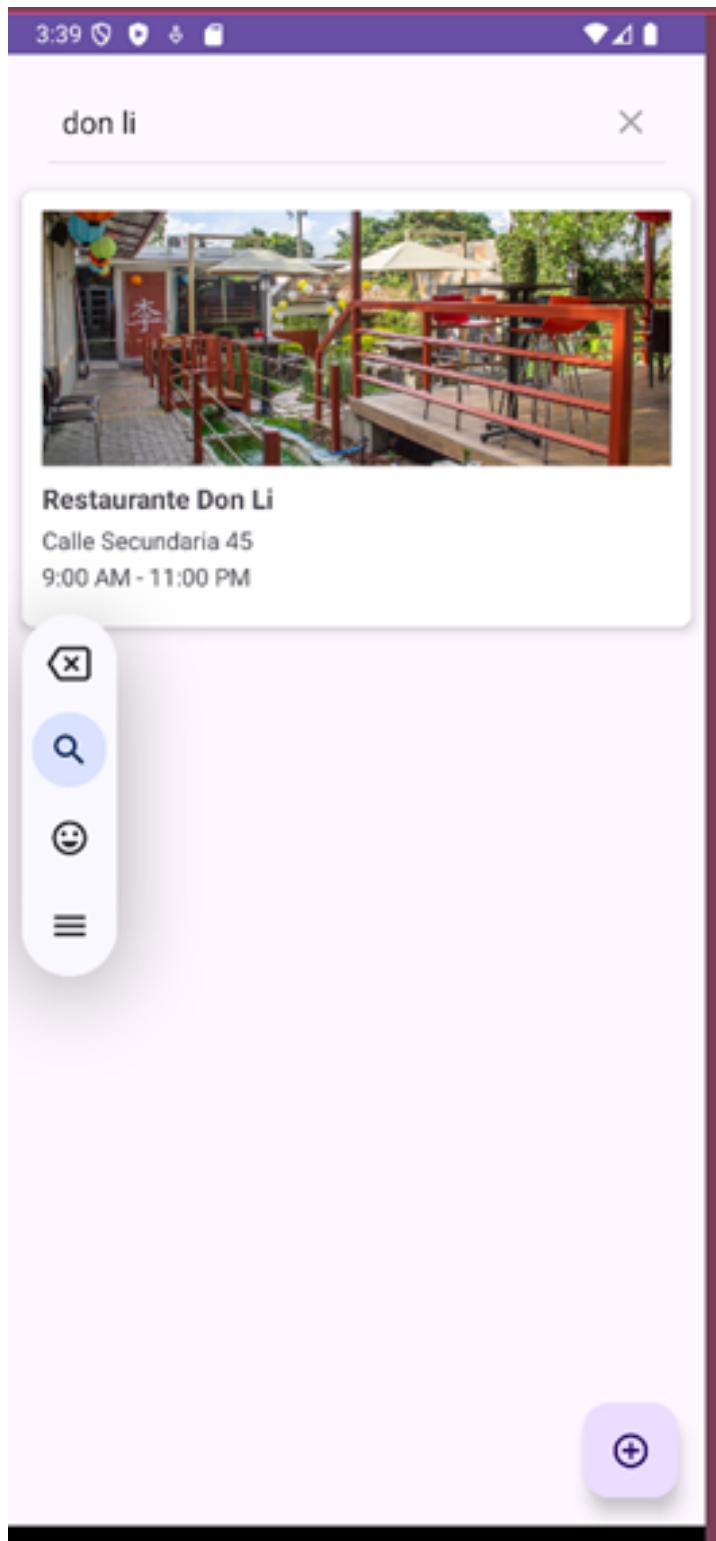
Confirmacion de reserva

```
btnConfirmarReserva.setOnClickListener {  
    val restauranteSeleccionado =  
spinnerRestaurante.selectedItem.toString()  
    val personasSeleccionadas = spinnerMesa.selectedItem.toString()  
    val fechaHora = editTextFechaHora.text.toString()  
    val nombreReservante = editTextNombre.text.toString()  
  
    if (restauranteSeleccionado.isNotEmpty() &&  
personasSeleccionadas.isNotEmpty() && fechaHora.isNotEmpty() &&  
nombreReservante.isNotEmpty()) {  
        Toast.makeText(this, "Reserva confirmada para $nombreReservante  
en $restauranteSeleccionado", Toast.LENGTH_LONG).show()  
    } else {  
        Toast.makeText(this, "Por favor, complete todos los campos",  
Toast.LENGTH_LONG).show()  
    }  
}
```

Aqui podemos ver en donde se está aplicando esta notificacion de confirmación



Buscador.



Se tiene un buscador el cual facilita dar con su restaurante de preferencia.

Muchas gracias.

