**I.E.S LAS SALINAS**



**PROYECTO FINAL FIN DE GRADO**

**CURSO 22-23**

|  |
| --- |
| **Mazo Comida** |

CICLO FORMATIVO GRADO SUPERIOR

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

|  |
| --- |
| **Autor: Daniel Eloy Rodríguez Pozuelo** |

**Índice:**

1. **Abstract**
2. **Justificación**
3. **Introducción**
4. **Objetivos**
5. **Desarrollo**
6. **Conclusiones**
7. **Bibliografía**

**1: Abstract**

La aplicación Mazo Comida contendrá una base de datos en firebase en la que el administrador (ya sea el encargado o el jefe del establecimiento) tendrá acceso y control sobre las comidas que se servirán en ese día (podrá cambiarlo cuando desee) de tal manera, que puede añadir comida, editar una ya puesta, y borrarla.

Luego está el cliente, que tiene acceso a solamente ver el menú del día (puesto por el Administrador).   
El cliente tiene dos funciones más:

1. Reservar (eligiendo un día y hora dentro del horario del bufé) poniendo su nombre y cuántas personas irán (como máximo podrán asistir 5 personas).

Eso sí, hay espacios para reservas como máximo. Si la base de datos de las reservas está llena, saldrá un mensaje de error.

1. Cancelar su reserva llamando al número de teléfono del bufé. El que esté a cargo de llevar la base de datos borrará la reserva.

The Mazo Comida application will contain a Firebase database in which the administrator (either the manager or the establishment's owner) will have access and control over the meals that will be served on that day (they can change it whenever they want). They can add food, edit an existing one and delete it.

On the other hand, there is the customer, who only Will be able to view the daily menu (set by the Administrator).

The customers also have two additional functions:

1. They can make a reservation by selecting a day and time within the buffet's operating hours. They need to provide their name and the number of people attending, with a maximum limit of 5 persons. However, there is a maximum capacity for reservations. If the reservation database is full, an error message will be displayed.
2. They can cancel their reservation by calling the buffet's phone number. The person in charge of managing the database will delete their reservation from it.

Please note that this translation is a direct representation of the provided text. If there are any specific technical terms or phrases related to the application that need to be accurately translated, please let me know.

**2: Justificación**

Una de las razones del porqué elegí hacer esta aplicación fue que adoro los bufés. No sería la primera vez que miro en la página web de “Muerde la Pasta”, por ejemplo, y veo que sirven el menú completo, solo para llegar y comprobar que sirven la mitad o que directamente hay cosas distintas a lo que vi en la web. Mi aplicación propone una solución más directa para evitar eso, y es que el “Administrador” que esté en el bufé se encargue de editar y actualizar en tiempo real lo que están sirviendo ese día. De esta forma, cuando un cliente entre en la aplicación y mire el menú, podrá decidir sin problema si quiere reservar, ya que estará seguro de que el menú que figure es el que se va a servir en realidad.

Otro aspecto que he tenido en cuenta ha sido el poner el formato de las reservas; de esta forma evitas hacer el viaje para encontrarte con el bufé lleno Esto es especialmente útil para bufés pequeños.

Por último, he procurado diseñar una aplicación sencilla. Dejando de lado el tiempo de entrega, que no es poco, a veces una aplicación bien optimizada y simple puede ser más satisfactoria, intuitiva y fácil de usar para el usuario que una app llena de detalles y funciones que pueden llegar a ser abrumadoras para los usuarios. En estos casos, “menos es más”.

**3: Introducción**

Las **características** principales de la aplicación son:

* Simple e intuitiva para el usuario
* Fácil acceso a las funciones
* Útil y aplicable para casos reales
* Manejo y efectividad en tiempo real

Además, me planteé si mi aplicación del bufé debería tener formato de reservas o no. Tras investigar, finalmente me decidí por las reservas, ya que, aunque el hecho de hacer una reserva puede resultar más tedioso, tiene muchas más ventajas de las que identifiqué a primera vista. Después de ver en internet los comentarios de los clientes y los trabajadores de los locales con ese formato, y pensando en el contexto de mi aplicación, hice una lista de las **ventajas principales** que aporta:

1. **Actualización en tiempo real del menú:** Al mostrar el menú del día actualizado en tiempo real, los clientes pueden ver qué platos están disponibles en ese momento. Esto les permite tomar decisiones informadas sobre qué alimentos desean probar y les da la seguridad de que los platos que eligen están disponibles.
2. **Experiencia gastronómica exclusiva:** Al establecer un límite de reservas cuando el bufé está lleno, se puede controlar la cantidad de personas en el establecimiento y brindar una experiencia gastronómica más exclusiva. Esto puede ayudar a mantener un ambiente más relajado y garantizar que los clientes puedan disfrutar de su comida sin aglomeraciones.
3. **Evita desperdicio de alimentos:** Al limitar las reservas y ajustar la cantidad de comida preparada según la demanda, es más probable que se reduzca el desperdicio de alimentos. Al ofrecer solo el menú del día actualizado, se puede optimizar la cantidad de ingredientes utilizados y minimizar las sobras innecesarias.
4. **Gestión eficiente de recursos:** Con el sistema de reservas y límite de capacidad, el bufé puede gestionar eficientemente sus recursos, como el personal y los ingredientes. Esto ayuda a evitar la sobrecarga de trabajo del personal y a garantizar que haya suficiente comida disponible para satisfacer la demanda de los clientes.
5. **Experiencia personalizada:** Al utilizar una aplicación en Android Studio para la reserva, se puede recopilar información sobre las preferencias de los clientes, como restricciones dietéticas o alergias. Esto permite adaptar los platos según las necesidades individuales de los comensales.

**4: Objetivos**

Los objetivos que quiero alcanzar con mi aplicación “Mazo Comida” son:

* Implementar un sistema de reservas en línea que permita a los clientes acceder al menú del día actual, realizar reservas y cancelarlas según su disponibilidad y preferencia.
* Desarrollar una aplicación móvil en Android Studio con dos modos: el modo cliente, que les permite ver el menú del día, realizar reservas y cancelarlas; y el modo administrador, que requiere una contraseña para acceder y permite editar en tiempo real el menú del bufé, y subirlo a una base de datos en la nube.
* Garantizar una experiencia de reserva fluida y conveniente para los clientes, brindándoles la capacidad de seleccionar la fecha, la hora y la cantidad de personas que van a venir, y recibir confirmación o no, dependiendo de si pudieron hacer las reserva.
* Permitir al administrador del bufé acceder de manera segura (con contraseña) al modo administrador de la aplicación para realizar cambios en tiempo real en el menú del bufé, como agregar, modificar o eliminar platos, según la disponibilidad de ingredientes y las preferencias del bufé.

**5: Desarrollo**

A continuación, mostraré de forma técnica el código y los métodos que he empleado a lo largo del desarrollo de mi aplicación, con capturas explicadas.

**5.1: Desarrollo (Diseño)**

Captura de pantalla de computadora

Descripción generada automáticamenteSi bien es cierto que esta suele ser la última parte en este tipo de proyectos, yo decidí dejar hecho el diseño primero y así tener una mejor idea de cómo iban a ser las cosas.

Aquí diseñé e implementé una pantalla de carga que tiene el degradado de los colores del logo del bufé “MazoComida”, que diseñé yo mismo en Photoshop:

Luego me puse a trabajar en el diseño del “modo cliente” y “modo administrador” con sus respectivos botones.

Captura de pantalla de computadora

Descripción generada automáticamente

Aquí, en el modo cliente, apliqué también otro fondo degradado. Los botones son de color rojo con diferentes tonalidades para cada botón por variedad. Para el RecycleView puse el espacio de la imagen en la parte izquierda, el ID de la comida en pequeño arriba, ya que es un detalle poco importante (menos para el cliente).

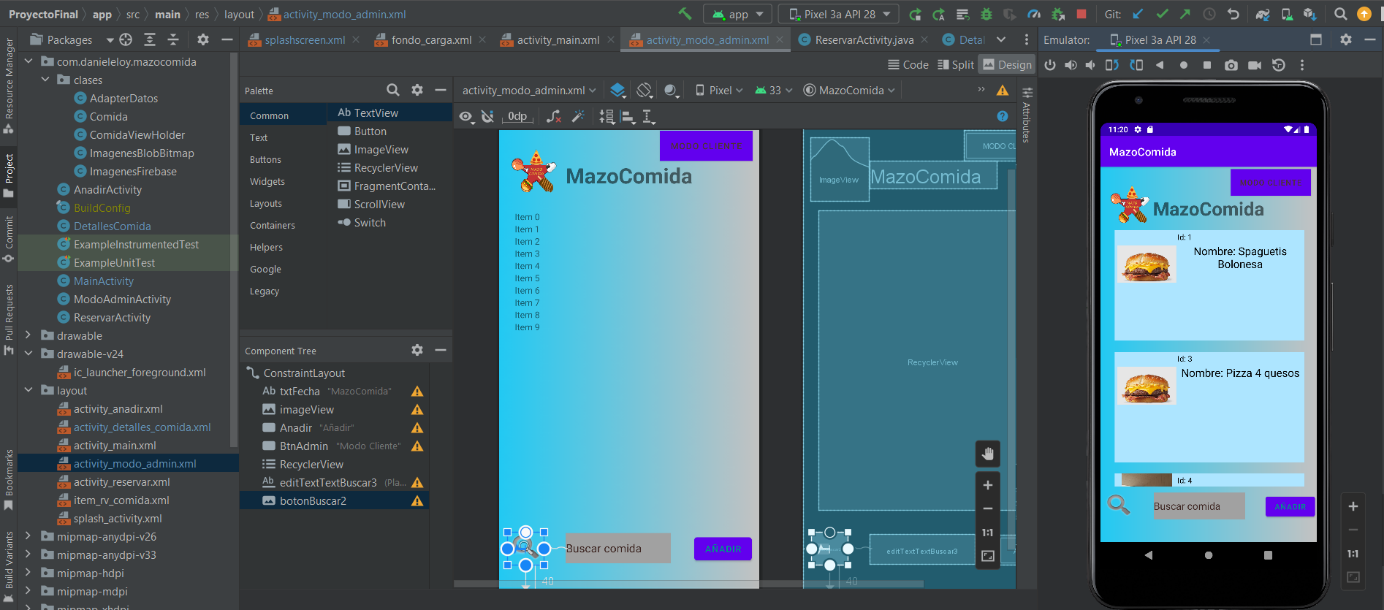
En la parte de abajo se encuentra una barra de búsqueda en gris, con el fin de conseguir un mayor contraste con el color de fondo y resultar fácil de localizar por el usuario. Con este fin agregué también un mensaje guía que dice “Buscar comida” junto a un botón con una lupa, indicando que es el que hay que pulsar para que empiece a funcionar y busque la comida deseada.

Interfaz de usuario gráfica

Descripción generada automáticamente Si el cliente clica sobre el botón “Reservar”, le llevará a esta ventana, en la que tendrá que rellenar los datos y volver a clicar sobre “Reservar” para mandar la reserva a la base de datos.

Captura de pantalla de un celular

Descripción generada automáticamentePor otro lado, si el cliente quiere cancelar la reserva, tendrá que pulsar sobre el botón “Cancelar reserva”. Le saldrá entonces un mensaje para que llame a un número de teléfono (el que figura ahora mismo es el mío personal) y así cancelar la reserva:

Para el modo admin es casi lo mismo, solo que el fondo es diferente, con degradado azul y botones también en azul, más oscuro para destacar:

Si el administrador pulsa sobre cualquier comida que esté en el RecyclerView, le saltará esta pantalla que le da los detalles. Podrá entonces actualizar, cambiar y/o borrar los datos de ese plato en concreto (lo mismo para la imagen):

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego, si pulsa el botón “Añadir”, le saldrá esta pantalla, que sirve exclusivamente para añadir un nuevo plato a la base de datos. De momento, y hasta elegir una imagen, se verá la el logo por defecto en la aplicación:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**5.2: Desarrollo (Código y funciones)**

Con el diseño listo, era el momento de ponerlo en marcha. Para ello, en primer lugar había que conectar la aplicación con la base de datos Firebase.

Texto

Descripción generada automáticamente

Primero había que descargarse el archivo “google-services.json” e incluirlo en el proyecto. Ese código contiene códigos e ids que conectan el proyecto con la nube Firebase.

Texto

Descripción generada automáticamente

Lo siguiente fue implementar las dependencias de Firebase al apartado build.gradle(app). Esto permite a mi programa usar las librerías y funciones de Firebase sin problemas.

Texto

Descripción generada automáticamente

El último paso fue implementar los servicios de Google a mi build.gradle a nivel de proyecto.

Una vez completados estos pasos, ya podía empezar a programar.

Texto

Descripción generada automáticamenteLo primero, y muy importante, era importar las librerías de Firebase en todas las Activities de mi aplicación para poder usar sus funciones. Con el gradle ya sincronizado, no surgieron errores.

**5.2: Desarrollo (Código del modo cliente)**

Texto

Descripción generada automáticamenteLuego tenía que crear las variables de Firebase (amén del resto, para sincronizarlas con los elementos gráficos XML del Activity en cuestión)

Texto

Descripción generada automáticamenteEn esta parte se indica al programa que, antes de empezar, se tiene que mostrar el “SplashTheme”, la pantalla de carga (degradado con los colores del logo que mencioné anteriormente), que se mostrará durante 2 segundos para después pasar a la pantalla principal que es el modo cliente.

Texto

Descripción generada automáticamenteDespués, el código pone en el TextView directamente el día actual y el mes actual, para que el cliente sepa que, en efecto, el menú que está viendo, es el menú de HOY y no de MAÑANA, o PASADO.

Ahora pasamos a la parte de adaptar el RecyclerView y a desactivar (en la parte señalada de rojo) que el cliente pueda hacer la función de darle click al RecyclerView y con eso activar la función que se definió en la clase “AdapterDatos” junto con “ComidaViewHolder” de ir al ActivityDetalles que hace la función de borrar y editar/actualizar el plato.

Texto

Descripción generada automáticamente

Aquí la clase de Comida que tiene la Id con la que se verá en la base de datos y así no se reescribirá si una comida tiene un nombre similar.

Texto

Descripción generada automáticamente

Esta es la función que se desactivó en la parte antes mencionada, en la que si haces click en cualquiera de los elementos, te llevará al “DetallesComida”.

Texto

Descripción generada automáticamente

Ya en esta parte, se hace referencia a que lo que esté en Firebase, se coja y se muestre en el RecyClerView, y a su vez llenar un Array de la clase “Comida”, ya que tiene variables con el mismo nombre; de lo contrario, no se podría llenar.

Texto

Descripción generada automáticamente

Con este método, cuando se pulse sobre el botón, saldrá un mensaje que preguntará por la contraseña (“ModoAdmin123”). Si se rellena bien y coincide, la aplicación pasará al Modo Administrador; de lo contrario, saldrá un error indicando que la contraseña no es correcta y dará la opción de “Cancelar”, por si se ha pulsado el botón por error.

Texto

Descripción generada automáticamente

Este es el método que se encarga de que lo que sea que esté en la barra de búsqueda, pase por la lista que hay en el RecyclerView, y si se encuentra, se muestre en pantalla.

Texto

Descripción generada automáticamente

**Estos son los métodos de reservar y cancelar reserva:**

El reservar te lleva directamente a la ventana de reserva que mostraré más adelante-El de cancelar reserva hace que salte un mensaje en el que te informa que, si deseas cancelar, llames al número que te aparece en pantalla para que la persona encargada cancele tu reserva de la base de datos.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Una vez dentro de la ventana de reserva, tenemos el método Reservar, en el que primero se comprueba que los datos de los clientes están llenos para proseguir.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteLuego hay un botón que te hará elegir fecha en un calendario (gracias a los métodos “calendar”) y, si todo es correcto, se pondrán los datos de la reserva (nombreCliente, numeroClientes, y fechaReservada) en un textView para subirlos. Además, servirá también como identificador de la reserva, para evitar que se sobrescriban clientes que tengan el mismo nombre en Firebase, tal que así:

Se puede comprobar que se guarda el nombre junto con la fecha, hora y minuto en la que se hizo esa reserva.

Texto

Descripción generada automáticamenteAsí es el método que saca el calendario para que elijas el día y el mes.

Texto

Descripción generada automáticamente

Además, esta es la parte de seleccionar la hora junto con una condición que indica que solo puedes seleccionar entre las 01:00PM y las 22:00 PM, que es el horario del restaurante en cuestión; sino, saltará un mensaje de error.

**5.2: Desarrollo (Código del modo administrador)**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Para el modo administrador, al cual se accede pulsando el botón con el mismo nombre e introduciendo la contraseña, el primer paso es casi el mismo que en el modo cliente, solo que esta vez no se desactiva el “onClick” de la clase Adapter, ya que el administrador tiene como función principal eso mismo: añadir los platos del menú del día, editarlos y/o borrarlos.

Texto

Descripción generada automáticamente

El método buscar es el mismo que el modo cliente.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Este método es el que te lleva directamente a la pantalla de añadir comida, que contiene el método de añadir en la base de datos

Texto

Descripción generada automáticamente

Se pone una condición para que haya que poner algo en los espacios que lo piden (el nombre y el id de la comida, ya que la imagen tiene una por defecto).

Texto

Descripción generada automáticamente

Ya en esta parte, se sube al Firebase los datos puestos aquí, que actualizará automáticamente el RecyclerView.

Texto

Descripción generada automáticamente

Luego está este método que selecciona la imagen que tengas en la galería de tu teléfono (tendría que ser la imagen de la comida en cuestión). Si no se sube nada, se pondrá la imagen por defecto hasta que se la actualice.

Texto

Descripción generada automáticamenteAl clicar cualquier elemento del RecyclerView, se llega a la “DetallesComida” Activity. Aquí lo primero es poner los nuevos Ids y Nombre de comida, además de actualizar la imagen.

Texto

Descripción generada automáticamenteEl método de borrar es sencillo. Tras obtener la instancia y la referencia de la base de datos, simplemente, cogiendo como referencia el Id de la comida que quieres borrar, pones el comando señalado, y se borrará el plato.

Texto

Descripción generada automáticamente

Con este método actualizas los datos de la comida deseados. Puedes cambiar el Id o dejarlo como estaba, a tu elección. Y una vez puesto el id como referencia, sobrescribes los datos

Texto

Descripción generada automáticamentePara cambiar la imagen es lo mismo que con el método añadir, hay que elegir una imagen de la galería y más tarde se subirá al apartado del Storage de Firebase.

Captura de pantalla de computadora

Descripción generada automáticamenteP

or ejemplo lo seleccionado en rojo fue una imagen que se añadió junto con la comida.

**6. Conclusiones**

Mientras estaba avanzando en el proyecto, originalmente tenía planeado que el cliente tuviese un “carrito” en el que iba seleccionando los platos que quería, y ya alguien se los iría llevando, pero al ver que era demasiado complejo, además de llegar a la conclusión de que meter un método de reservas sería más eficaz, decidí abandonar la idea del “carrito” e incluir las reservas.

Por otro lado, me fui dando cuenta de lo mucho que se podría mejorar el funcionamiento de muchos negocios similares con una gestión así; es decir, que al hacer este proyecto me he dado cuenta de que, a veces, por muy simples que puedan parecer algunas aplicaciones y la ignorancia de lo que importa la base de datos, estos últimos elementos hacen muy eficaz y satisfactoria la experiencia del cliente/usuario, que es lo que al fin y al cabo se busca en todo negocio.

**7. Bibliografía**

A lo largo del desarrollo del proyecto, he usado los siguientes programas:

* Microsoft Word para redactar la memoria del proyecto.
* Adobe Photoshop para realizar el logo y otros elementos gráficos.
* Android Studio para la codificación y diseño base de la aplicación
* GitHub para la subida de copias de seguridad y actualizaciones de la app
* ChatGPT que se usó para la solución de errores que retrasaban el avance del proyecto.