

ENTREGA EVALUACIÓN 1

Documentacion de errores

Álvaro Del Valle Fernández

November 17, 2025

1 Introducción

En este documento hablaré sobre los errores y problemas que me encontré al realizar esta práctica.

2 Android Studio

Android Studio fue mi fuente principal de problemas, tardando demasiado en completar la interfaz gráfica tal y como la había realizado en figma, lo mas complicado fue añadir sombras junto a bordes curvados y con diferentes colores. Tras terminar esa parte la complicación fue gestionar correctamente la conexión entre el servidor y android, tardando numerosas horas en arreglar esos problemas.

2.1 XML

El xml me resulto algo complicado en comparacion con el FXML, con falta de muchos elementos sencillos del CSS como el borderradius, mi principal problema fue crear los elementos curvos como son los bordes de los botones y los rectángulos, teniendo que crear una clase en drawable como por ejemplo radius.xml. La complicacion no fue el radius si no la combinacion de este con la sombra, Teniendo que crear otro elemento xml para este y situarlo en una capa inferior. Otro problema fue situar elementos de forma libre, con numeros problemas en los que el visor mostraba un resultado y la app lanzada otro. El arreglo fue usar en los androidx.constraintlayout.widget.ConstraintLayout. añadir:

```
app:layout_constraintStart_toStartOf="parent"
```

2.2 .java

Mi principal problema con los .java fue enviar datos entre vistas de forma correcta, hasta que encuentre un metodo de guardarlos internamente de forma temporal.

```
sharedPreferences = getSharedPreferences("RestaurantePrefs", MODE_PRIVATE);
```

Necesite usar hilos debido a multiples problemas de crasheos, usando el metodo:

```
runOnUiThread(() -> {})
```

Gran parte de mi codigo son elementos de gestion de errores, centrados en obtener informacion detallada de que punto esta fallando.

```
java.io.InputStream errorStream = connection.getErrorStream();
```

Websocket me costo multiples horas en hacer que funcionase bien, viendo numerosas guias.

Cree un elemento llamado overlay que se superpone a la vista principal, diseñarlo visualmente fue sencillo pero que funcionase correctamente al superponerse fue muy complicado, primero el problema fue que no mostraba la ventana debajo y luego que podia salir de ese menu haciendo click.

En PayActivity fue donde me aparecieron mas problemas, teniendo que añadir funciones enteras destinadas a que me indique que falló exactamente o si funcionó. Crear el boton de eliminar dentro de cada elemento fue bastante complicado y necesité cambiar numerosos puntos de la version original. No mosntandolo con cantidad si no repitiendo el producto en el array y sin poner el boton de sumar pedido.

3 Server

Mi principal problema fue que en ciertos puntos puse localhost, en otros 0.0.0.0 y en otros 10.0.2.2, siendo http://10.0.2.2:5000/clientes, la correcta. Esto me dio numerosos problemas y creo que algunos elementos no fucionan correctamente y los tengo duplicados debido a eso.

Muchos son una mezcla entre el emtodo que estaba intentando antes de usar websocket, por lo que esos elementos no se utilizan, los manejos de errores tambien ocupan espacio por lo que mas de la mitad del codigo no se usa en el resultado final.

El problema de las ip es el motivo principal, con algunos elementos del websocket en el puerto 8080.

4 MongoDB Atlas

Mi problema principal con mongoDB fue con la ip, al crear una lista blanca y mover el proyecto en git a otros ordenadores me encuentre con multiples problemas, al no indicarme con los logs de errores si se estaba conectando correctamente.

La ip ahora esta abierta por lo que no deberia de volver a dar problemas al mover el proyecto de ordenador a ordenador.

Otro de los problemas fue que añadi el campo de "estado:" en la mesa, junto con

el "estado:" del cliente, al no darle un nombre claro me encontré con muchos errores al intentar cambiar el estado de la mesa y de los clientes.

```
0
_id "6913f603bd20090f1876023c"
numero 1
clientes
0
_id "691a8f18bf50cdce2cb7a210"
numeroCliente 1
pedidos
0
plato "6913f5e9bd20090f1876022c"
cantidad 2
precio 12
nombre "Spaghetti Carbonara"
_id "691a8f1bbf50cdce2cb7a21a"
totalPedidos 24
estado "pidiendo"
mesaId "6913f603bd20090f1876023c"
__v 1
estado "esperando"
__v 93
```

5 Administracion restaurante

5.1 FXML

Dentro de las complicaciones el XML me resulta lo mas sencillo al parecerse a CSS, el unico problema fue que las columnas con elementos las coloqué de forma desordenada por lo que necesite cambiar las ids de sitio para que funcionase correctamente. Fue mas el problema de mapear todos los circulos.

5.2 .java

Uno de los problemas principales fue encontrar una forma de actualizar el contenido de la vista sin consumir demasiados recursos, use un timeline que se efecuta de forma indefinida y actualiza los datos cada dos segundos. Aun con dos segundos creo que da problemas de rendimiento. Me encuentre con los problemas de mi mongodb con los estados, teniendo problemas cuando o no mostraba nada pero si que se actualizaba en el servidor, o los colores se cambiaban de forma erronea. Mi mayor problema y que tarde varias horas en solucionar fue que cuando actualizaba los datos al servir, estos cambiaban al color correcto pero volvian tras el timer de 2 segundos al color anterior. Esto me llevo horas y necesite de volver a una version anterior del trabajo para poder arreglarlo.

Otro de los problemas fue al mapear incorrectamente los círculos que representan las mesas, dando errores o mostrando los colores en posiciones erróneas.