Buenas tardes, soy Gemma Selles y hoy os voy a presentar mi app para chatear: chApp

Para usar esta aplicacion necesitaremos 2 telefonos mobiles Android.

Esta aplicacion sigue la filosofia cliente/servidor implementada mediante Sockets tradicionales.

En resumidas cuentas, el funcionamiento es el siquiente:

- El **servidor** crea un *ServerSocket* en el puerto que hemos seleccionado (6000)
- *ServerSoket* se queda esperando a que el cliente se conecte.
- El **cliente** crea un *Socket* e intenta conectarse a la IP del servidor y al puerto que esta escuchando.
- Ambos se quedan **escuchando** si el otro les envia un mensaje.

¿Como he implementado la aplicacion? Os lo explico a traves de las clases que he creado.

Empezamos por la ***MainActivity*** donde el telefono elige si quiere ser el **servidor** o el **cliente**.

Hago uso de ***Intents*** para llevar al usuario a la siguiente actividad segun su eleccion.

Tiene este sencillo aspecto.

La funcion mas importantde del servidor es **initServidor** donde se inicia la conexion y espera a que el cliente acepte. Una vez conseguido se queda escuchando si recibe algun mensaje o si el usuario quiere enviar uno.

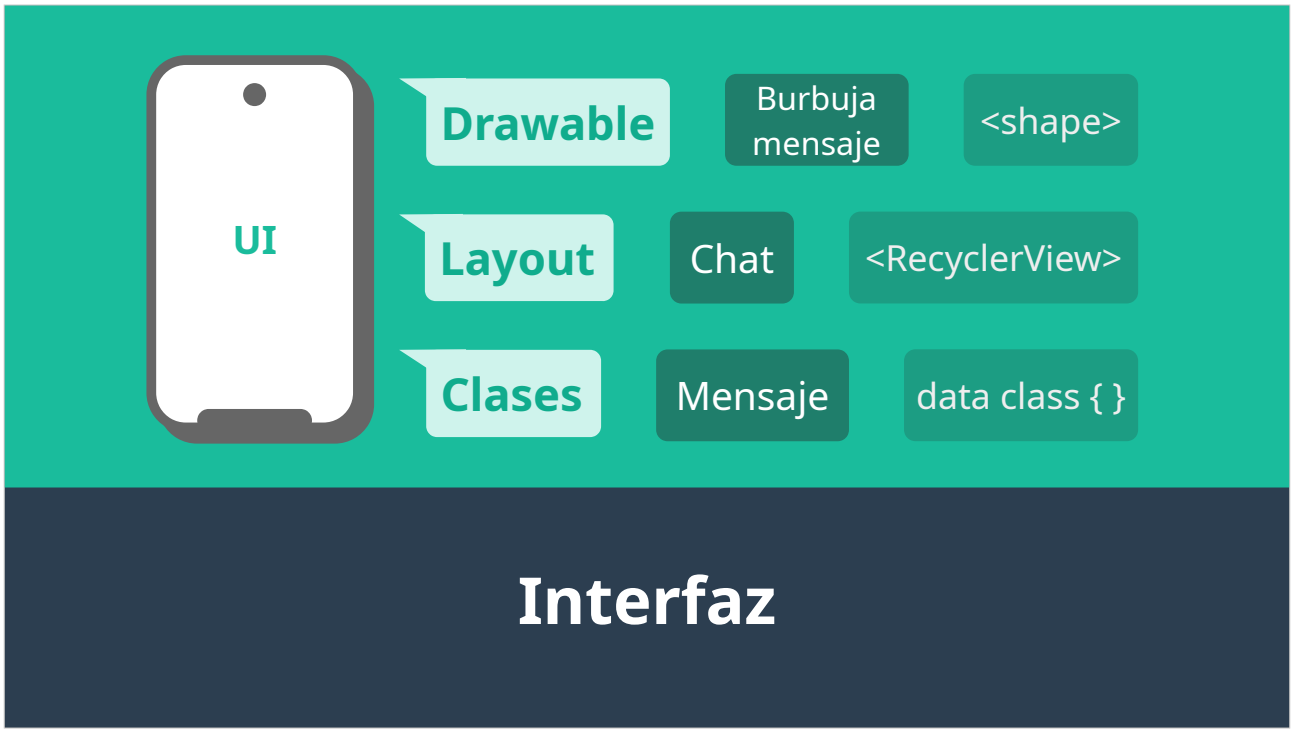
Para que funcione he tenido que usar **corrutinas**, para no cargarme el hilo principal y que la aplicacion lance errores (como veremos mas adelante), **streams** para recibir y enviar mensajes y **manejar errores** con try/catch.

Os muestro el resultado:

El **Cliente** hace practicamente lo mismo que el servidor a diferencia que en lugar de iniciar la comunicacion, crea un ***socket*** con los datos para establecerla y aceptarla.

El resto de funciones y elementos que he usado son iguales.

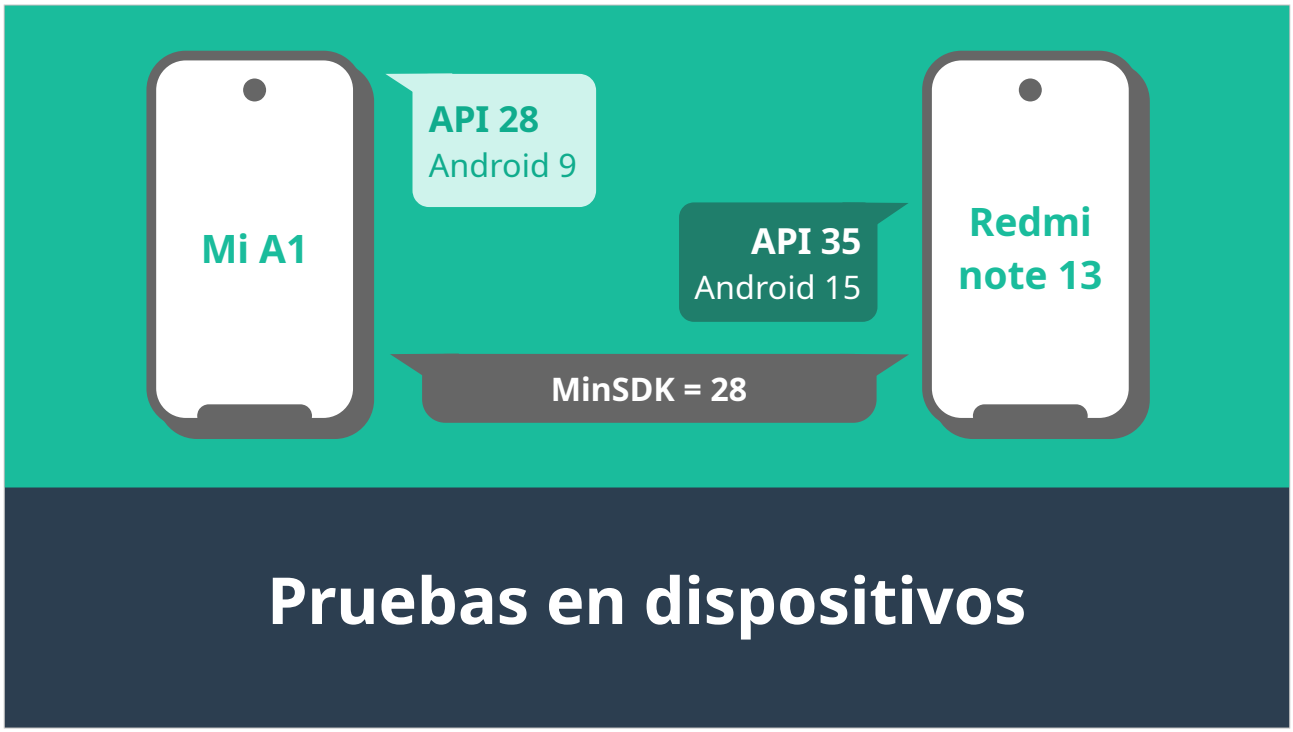
Ha quedado asi:



Otros aspectos a destacar de la aplicacion es que para que tenga un aspecto de aplicacion de chat tradicional he tenido que customizar algunos elementos.

He creado en la carpeta de «**res**» drawables y layouts personalizados y tambien clases que los manegen.

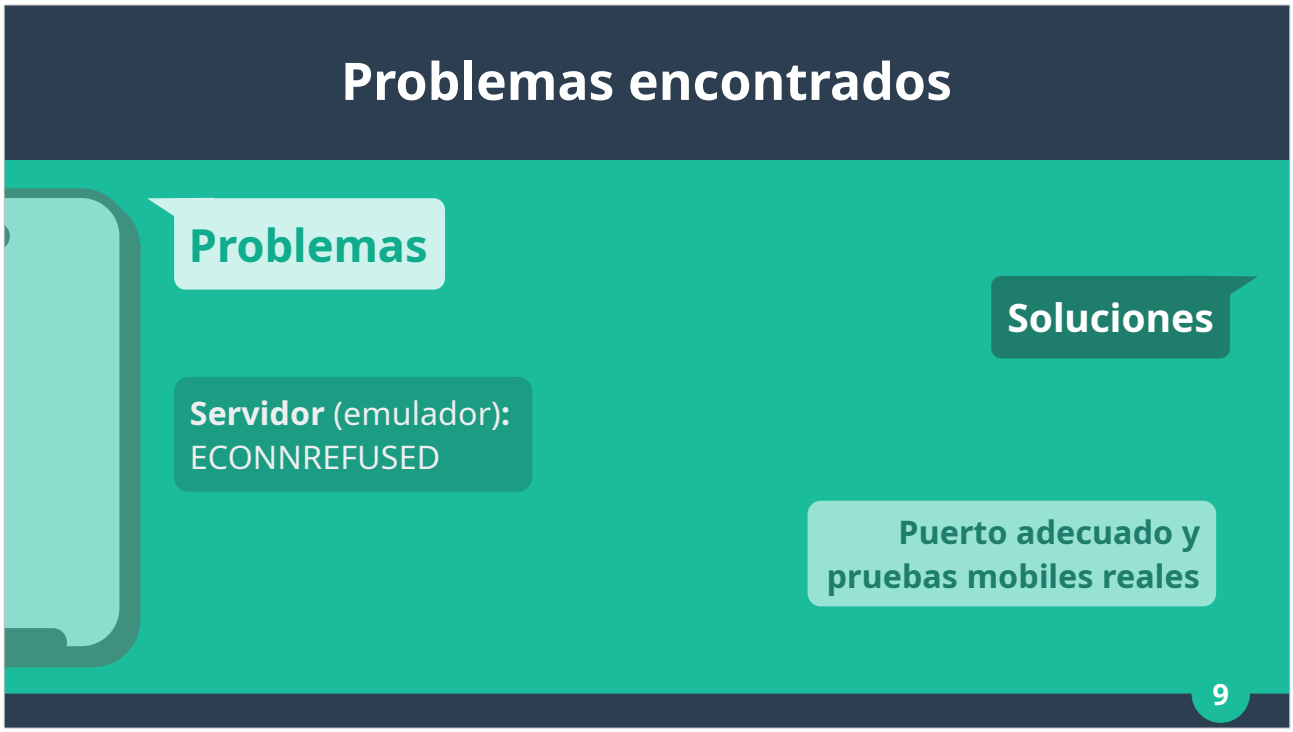
- Tenemos las burbujas de mensaje, que se muestran en el chat segun si se han recibido o enviado y se muestran una detras de la otra con los datos correspondientes.
- Las burbujas son shapes, el chat funciona con RecyclerView que tiene a su vez un adaptador que lo hace funcionar de forma personalizada y cada mensahe se ha hecho cob la data clas mensaje.



La aplicacion se ha probado y funciona en dispositivos mobiles reales.

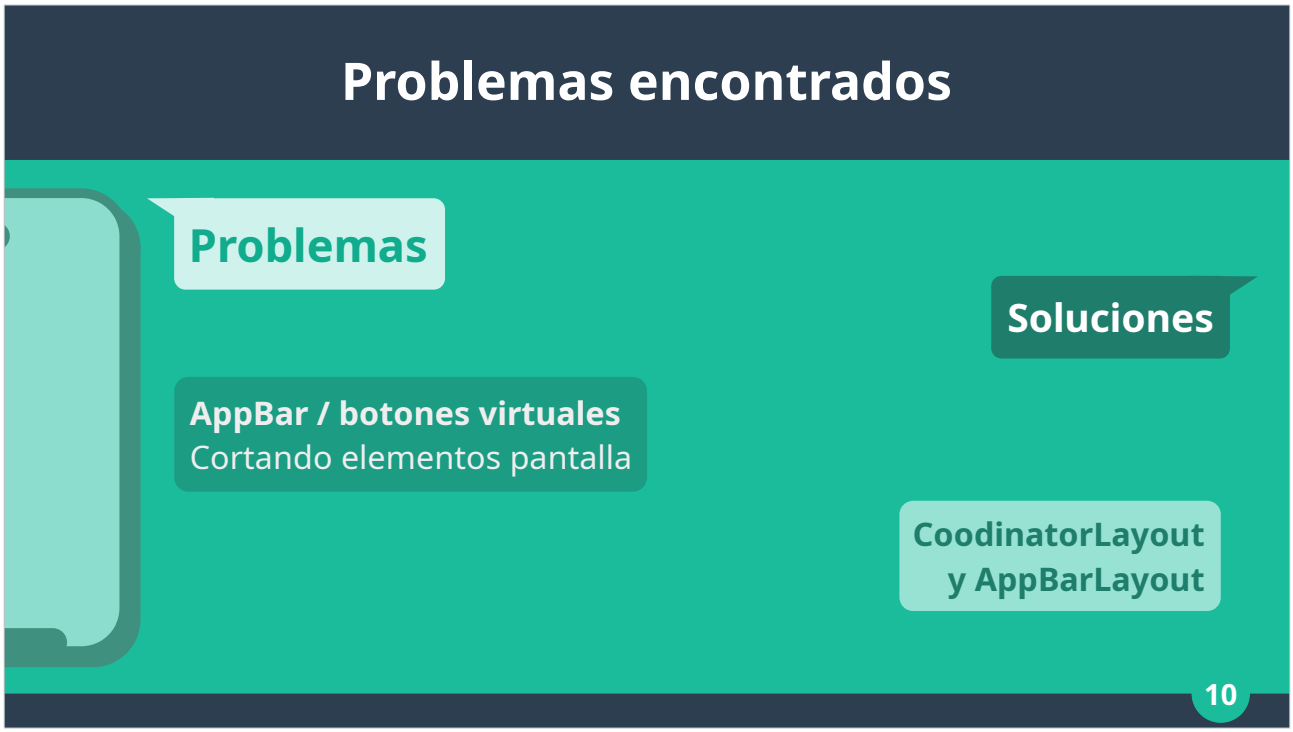
Las pruebas las he hecho con un Mi A1 y un Readme Note 13 que respectivamete tienen API 28, Android 9, y API 35, Android 15. Para adaptarnos al mobil mas limitado, he usado un SDK minimo de 28.

Otras diferencias entre dispositivos es que la pantalla del MiA1 era mas pequeña y tenia los botones fisicos a diferencia del Redmi. Estas diferencias seran destacables a continuacion.



Todo desarrollo que se precie tiene problemas que al principio parecen inexplicables y detienent tu progreso. Ese fue el caso de este primer problema:

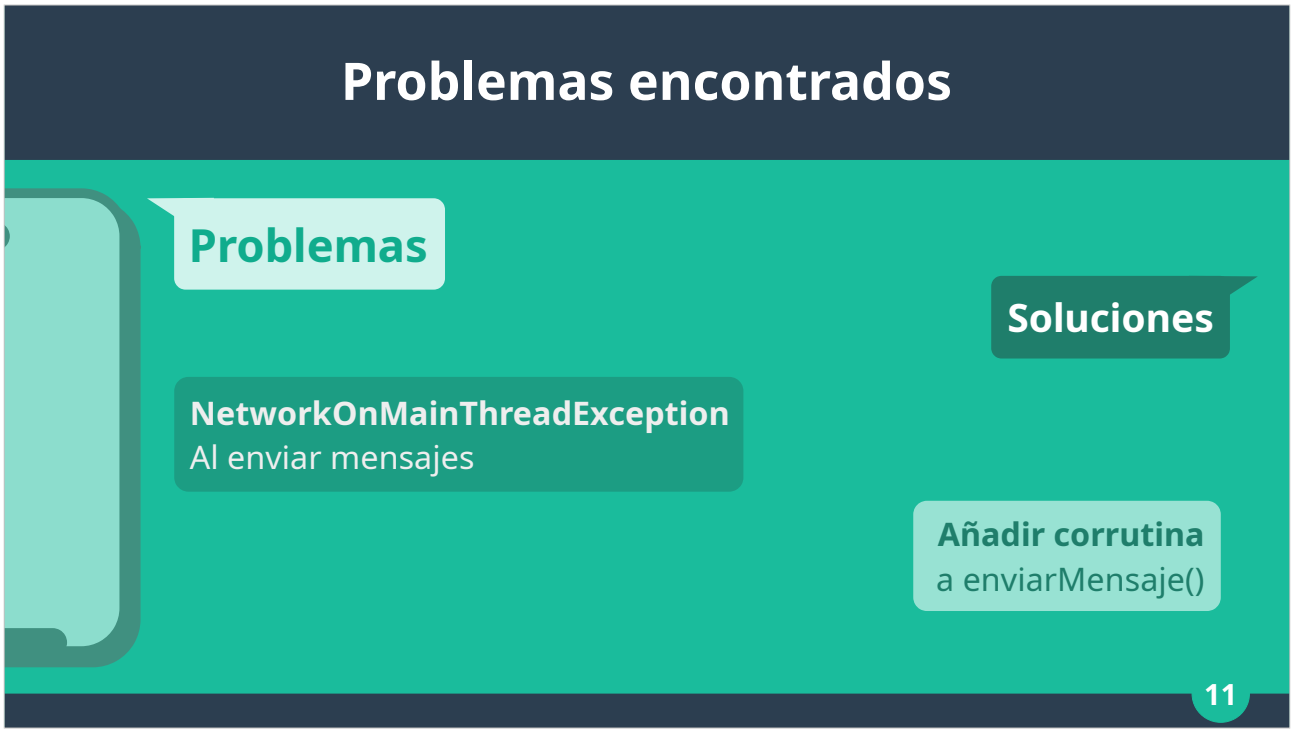
Al hacer pruebas, primero lo intente con dos emuladores dentrp de Android Studio y aunque ya tuviese programadas la parte de Cliente/Servidor para probar si se conectaban, daba el error de ECONNREFUSED. No entendia el motivo si la IP era la adecuada, però al probar en dispositivos fisicos, me di cuenta que el problema no era poner mal la IP (como pensaba al principio) si no por los emuladores.



Seguimos con los problemas visuales, en ambos mobiles se cortaba la aplicacion si el chat crecia mucho, primero lo solucione poniendo margenes manualmente para que se adaptara però no quedaba igual para las ds pantallas.

Luego tambien añadi una AppBar para distinguir que rol tenia el usuario y eso estropeo mas el aspecto, todo esto le sumamos que el redmi con los botones virtuales tambien tapaban parte de la aplicacion.

Se soluciono con CoordinatorLayout ya que cogia perfectamnte las dimensiones de la pantalla sin importar el tamaño y respentando los botones y el AppBar.



Y por ultimo però no menos importante, antesde ser capaz d encir un mensaje me topçe con este error, NetworkOnMainThreadException. Como el propionombre indica estaba estropeando la ejecucion del hilo principal cosa que hacia que se cerrara la aplicacion. Este hilo principal esta reservado para la interfaz, cosa que yo pensaba que respetaba con «Dispatch.Main» en mis corrutias, però se me paso con el metodo de escuchar mensajes y me lanzaba dicho error.

Con ponerle correctamente la corrutina a esa funcion no me volvio a salir.

Y hasta aquí Chapp. Aunque de funcionamiento es sencilla ha tenido mucho trabajo detras. Me ha gustado hacerla y me he qudado con ganas de haber tenido mas tiempo para implementarle mas cosas. Pero el tiempo es el que tenemos pra poder brindar la mejor solucion possible y esta ha sido la mia.

Muchas gracias por vuestra atencion y hasta a proxima.