MyBinder

Alejandro Marín Bermúdez

Contenido

[Resumen 3](#_Toc134550966)

[1. INTRODUCCIÓN 5](#_Toc134550967)

[2. Objetivos y características 7](#_Toc134550968)

[3. Diseño final de la aplicación 19](#_Toc134550969)

[4. Matemáticas en el proyecto. 27](#_Toc134550970)

[5. Estructura del proyecto 29](#_Toc134550971)

[6. Finalidad 31](#_Toc134550972)

[7. Fase de pruebas 33](#_Toc134550973)

[8. Conclusiones 35](#_Toc134550974)

[9. Problemas en el desarrollo. 37](#_Toc134550975)

[10. Bibliografía 39](#_Toc134550976)

# 

# Resumen

Los que juegan juegos de cartas coleccionables o simplemente las coleccionan, tienen un problema en común, el manejar la colección y la lista de cartas que ellos poseen.

Para ello se ha creado este programa llamado MyBinder donde se introduce los datos de la carta más una foto y podrás llevar siempre en tu dispositivo móvil tu colección de cartas y poder gestionarla de manera rápida y eficaz donde te apetezca, aparte de esto también se le da una solución a varios problemas más, se crea una función para crear mazos y comprobar probabilidades de las manos que elijas y por ultimo relacionado con la funcionalidad principal del programa, hay una lista para meter las cartas que quizás ya no te interese tener, y quieras soltarla, además de poder ver la lista en la aplicación se podrá crear un archivo PDF para poder enseñar la pull de “Cambio” que tienes a posibles interesados.

En este Documento se verá también de manera más extensa: los objetivos, los medios utilizados y la planificación que se ha realizado para llevar a cabo el proyecto, también las posibles mejoras que tendrá en el futuro y los fallos que tiene.

# INTRODUCCIÓN

**1.1 ¿Qué es MyBinder?**

Es una herramienta que permitirá a los jugadores del juego de cartas “Yu-Gi-Oh!” a poder gestionar sus colecciones y poder crear listas con cartas que ya no quieran y por ultimo crear mazos con las cartas de su colección y poder ver la probabilidad que tiene de robar la mano que ellos quieran.

**1.2 ¿Por qué nace este proyecto?**

Para mí, la comunidad de este juego de cartas es importante, está lleno de amigos y de gente a la que admiro, por lo que quiero hacer una herramienta que ayude tanto a los más expertos como a la gente que este inicializando en este juego.

Hay aspectos en este juego que no son sencillos como la creación de mazo, que por razones de “consistencia” es mejor llevar más o menos copias iguales de una carta para que no te salgan “malas manos”.

También hay gente con colecciones extensas que suelen perder u olvidar que tenían alguna carta y gracias a esta aplicación pueden llevar un registro de las cartas que tienen.

Este tipo de aplicación existe pero son de pagos y no son muy accesibles al público, para ello, por eso quiero compartir esta aplicación con todos a los que le interese este hobby.

# Objetivos y características

El objetivo de este proyecto como anteriormente se ha comentado es realizar una aplicación con unas especificaciones concretas, siendo en literalmente una típica aplicación del tipo CRUD y para ellos se van a realizar diversos preparativos para llevar un correcto desarrollo de esta.

* 1. **Historia de usuario**
* Como usuario, me gustaría poder insertar cartas, para poder ver mi colección de manera rápida.
* Como usuario, me gustaría poder editar cartas insertadas, para poder corregir los datos que me he equivocad al insertar la.
* Como usuario, me gustaría poder eliminar unas cartas insertadas, para poder quitar una carta que ya no tengo.
* Como usuario, me gustaría poder filtrar en mi colección, para poder encontrar más rápidamente la carta de mi colección.
* Como usuario, me gustaría poder crear mis mazos, para poder tener mi colección aun más controlada.
* Como usuario, me gustaría poder editar mis mazos, para poder actualizarlo a la versión más actual.
* Como usuario, me gustaría eliminar mis mazos, porque ya no me interesan tenerlos o los he deshecho.
* Como usuario, me gustaría tener una lista de cambio, para poder tener un control de las cartas que quiero y las que no.
* Como usuario, me gustaría editar mi lista de cambio, para poder añadir o eliminar cartas que ya no quiera cambiar o ya lo he hecho.
* Como usuario, me gustaría poder sacar un pdf de mi cambio, para poder enseñarlo más fácilmente a las personas.
* Como usuario, me gustaría tener una opción para ver las probabilidades que tengo de robar cada carta en mi mazo, porque quiero comprobar que tal “consistente” es.
  1. **Fases del desarrollo**

**Fase 1:** Funciones Básicas (Crear, editar, Eliminar y Mostrar Cartas de la base de datos)

Duración: 1.5 semanas

**Fase 2:** Filtrado

Duración: 0.5 semanas

**Fase 3:** Lista de cambio

Duración: 1 semana

**Fase 4:** Opción de Mis mazos

Duración: 2 semanas

**Fase 6:** Opción de probabilidades

Duración 1 semana

**Fase 7:** Corrección de estilo

Duración: x semanas

**2.3 Tecnologías usadas.**

Las tecnologías usadas en este proyecto son las mismas que las que hemos aprendido en clase este curso: kotlin y mysqlLite.

Kotlin:



Kotlin es un lenguaje de programación multiplataforma, estático y fuertemente tipado, desarrollado por jetBrains en 2011. Kotlin se basa en la JVM (Java Virtual Machine) y es compatible con Java, lo que permite a los desarrolladores de Java adoptar Kotlin fácilmente. Además, Kotlin también puede ser compilado a Java Script para su uso en aplicaciones web.

Kotlin se caracteriza por ser un lenguaje moderno, conciso y seguro. Ofrece una sintaxis más simple y legible que Java, lo que lo hace más fácil de aprender y usar. Además, Kotlin cuenta con características avanzadas como la inferencia de tipos, funciones de orden superior, lambdas, null safety y data clases, que facilitan la escritura de código seguro y eficiente.

Kotlin es utilizado en una ampliada variedad de proyectos, desde aplicaciones móviles Android hasta aplicaciones de escritorio, servidores y aplicaciones web. Además, Kotlin es respaldado por una gran comunidad de desarrolladores y empresas, lo que garantiza un soporte continuo y un crecimiento a futuro

MysqlLite:



SQLite es un sistema de gestión de bases de datos relacionales ligero, rápido y autónomo. A diferencia de otros sistemas de gestión de base de datos, SQLite no funciona como un proceso separado y no requiere de un servidor de bases de datos para su funcionamiento. En su lugar, SQLite se ejecuta como una biblioteca dentro de la aplicación, lo que lo hace muy adecuado para aplicaciones que requieren de una base de datos local.

SQLite es ampliamente utilizado en aplicaciones móviles y de escritorio, asi como en sistemas integrados y dispositivos de loT, debido a su portabilidad y facilidad de integración. Además, SQLite soporta el lenguaje SQL estándar, lo que lo hace compatible con la mayoría de herramientas y frameworks que utilizan SQL.

Una de las principales ventajas de SQLite es su rendimiento, ya que es capaz de manejar grandes volúmenes de datos con una alta eficacia. Además, SQLite también cuenta con una amplia variedad de características como transacciones ACID, triggers, vistas, índices y subconsultas, lo que lo hace muy versátil para una amplia variedad de aplicaciones y casos de uso.

**2.4 Relación de entidades.**

En la base de datos tendríamos 4 entidades (monstruo, MagicaTrampa, Mazo y Carta).

* La entidad monstruo está compuesta de los siguientes campos:

Id: Int, categoría: String, categoria2: String, nombre: String, atributo: String, nivel: Int, tipo: String, ataque: Int, defensa: Int, código: String, escala: Int, cantidad: Int, imagen: String, cambio: Boolean.

* La entidad MagicaTrampa es muy similar a la de monstruo, la diferencia es que no tiene algunos campos:

Id: Int, nombre: String, categoría: String, tipo: String, código: String, cantidad: Int, imagen: String, cambio: Boolean.

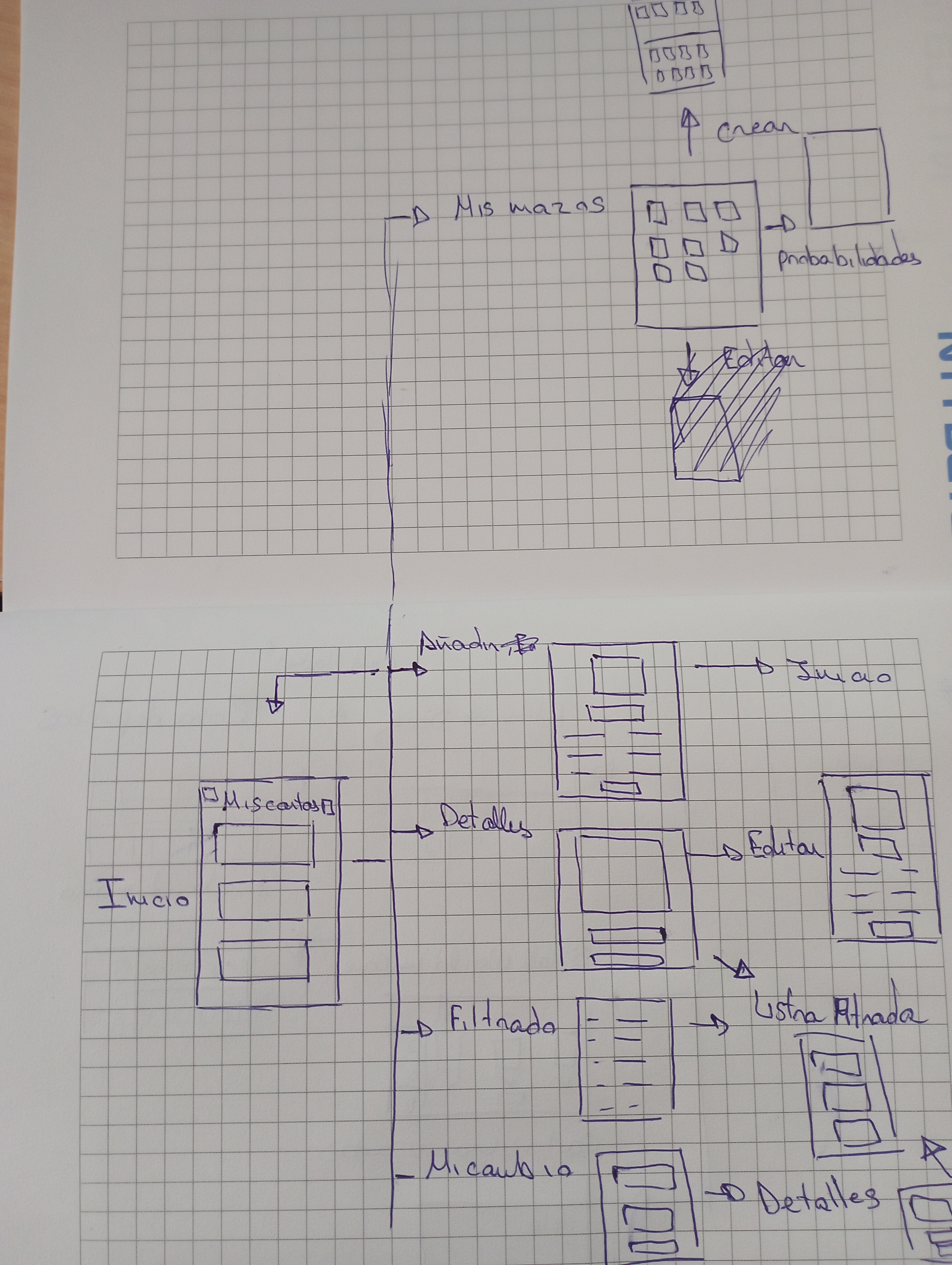
* La entidad Carta está compuesta por lo siguiente:

Id: Int, Foto: String, id \_Mazo: Int.

* La entidad Mazo está compuesta por la entidad Carta

carta :Carta.

**2.5 Flujo de pantallas.**



El flujo de la aplicación es simple.

**Pantalla principal**: La pantalla principal es la lista de todas las cartas que tienes en tu colección, son mostradas atreves de un Recycler View con CardsView en su interior para poder mostrar de manera más uniforme tu lista de cartas en la base de datos, desde aquí puedes ir a varios sitios como el menú, el filtrado o a detalles de cartas.

**Detalles de cartas**: Se accede desde cualquier lista que muestre cartas pulsando en uno de los card View que se muestran en ellas, aquí dentro podremos ver todas las características de las cartas, también hay varios botones que son editar, borrar y mandar al cambio, si pulsamos a cualquiera de estos botones o a una de las características de la carta nos mandara a otra página.

**Menú:** Este menú es accesible desde cualquier actividad relacionada con las listas, desde este menú podremos acceder a varias páginas como: Mis cartas, Añadir Cartas, Mi cambio, Mis mazos.

**Añadir carta:** En esta actividad aparecerá un formulario donde podremos crear nuestras cartas para meterla en la base der datos, desde aquí podremos acceder a la cámara de nuestro teléfono donde podremos hacer una foto a la carta que queramos insertar, una vez insertada nos mandara de nuevo a la pantalla principal ( Mis Cartas).

**Mi cambio:** En esta pantalla mostraremos todas las cartas que han sido mandadas desde la actividad de detalles, apareceremos aquí cada vez que se mande una o si entramos desde la pestaña del menú, aquí nos encontraremos un botón donde podremos descargar un pdf de la lista que se nos muestra por pantalla.

**Filtrador:** Se accede desde el botón con forma de lupa que hay en algunas listas como la de la pantalla principal, aquí se nos mostrara un formulario donde podremos rellenarlo o dejar campos vacios y pasaremos a la actividad ‘filtrado’.

**Filtrado:** Se puede acceder desde la pantalla filtrador o después de pulsar cualquier característica en “detalles”, aquí nos mostrara una lista de cartas que cumplan con los requisitos que tengamos puesto.

**Mis mazos:** Esta pantalla se accede desde el menú, aquí nos mostrara la lista de todos los mazos que tendremos creados y un botón para poder crearlos.

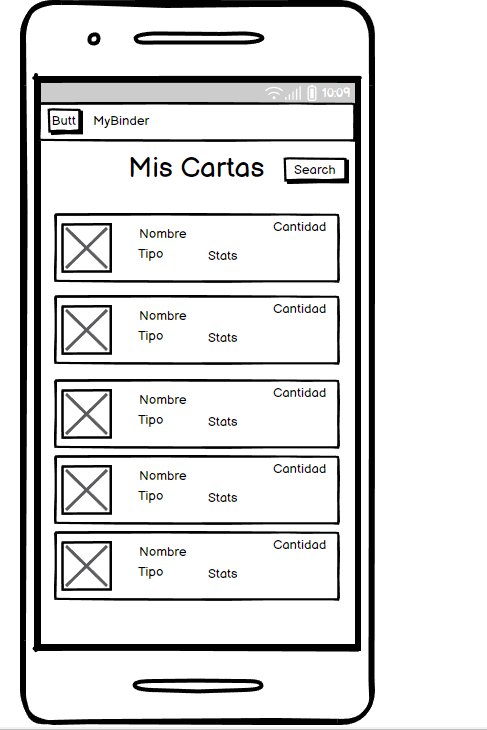
**Nuevo mazo:** Aquí tendremos una lista de todas las cartas que tengamos a nuestra disposición, al hacer click sobre una de estas aparecerá en una lista que tenemos en la pantalla, aquí tenemos un botón que es guardar donde nos guardara el mazo y nos enviara de nuevo a mis mazos.

**Editar Mazos:** Se accede presionado sobre uno de los elementos de la lista de mis mazos, aquí podremos editar los mazos en una pantalla muy similar a nuevo mazo, una vez guardado regresaremos a la pantalla de mis mazos.

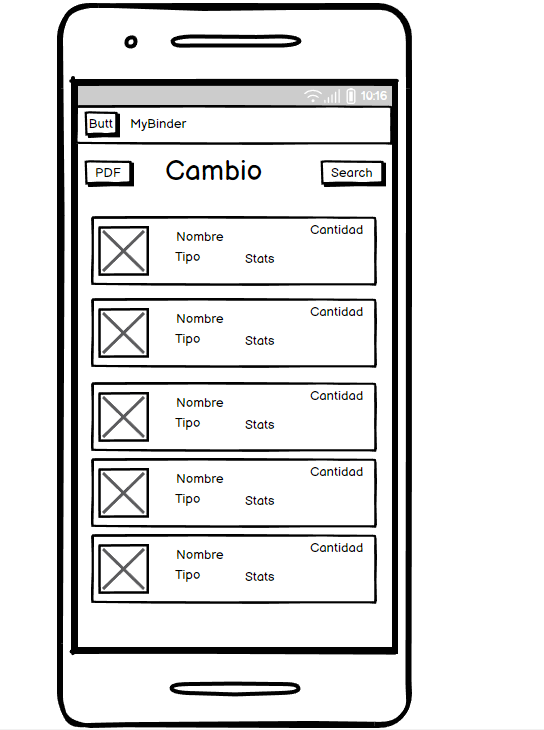
**Probabilidades:** Se accede presionando también sobre uno de los elementos de la lista de mis mazos aquí aparecerá una lista con todas las cartas que tenemos en el mazo y al pulsar sobre ellas aparecerán en otra lista que será la “mano” aquí nos calculara en tanto porciento cual sería la probabilidad de robar cada carta de nuestra mano, desde aquí saldaremos de nuevo a mis mazos.

**2.6 Mockups.**

**- Feed**

****

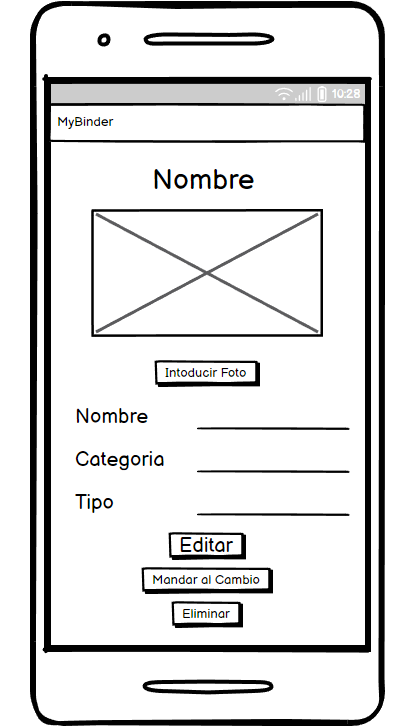
* **Filtrado**



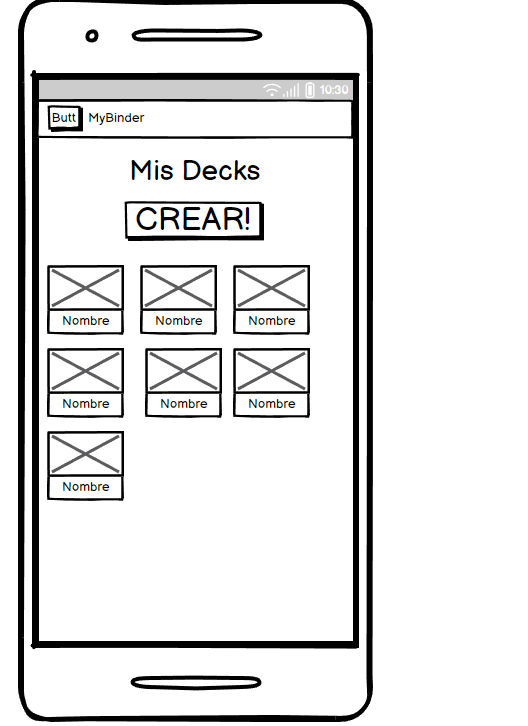
* **Añadir Carta/Editar**

****

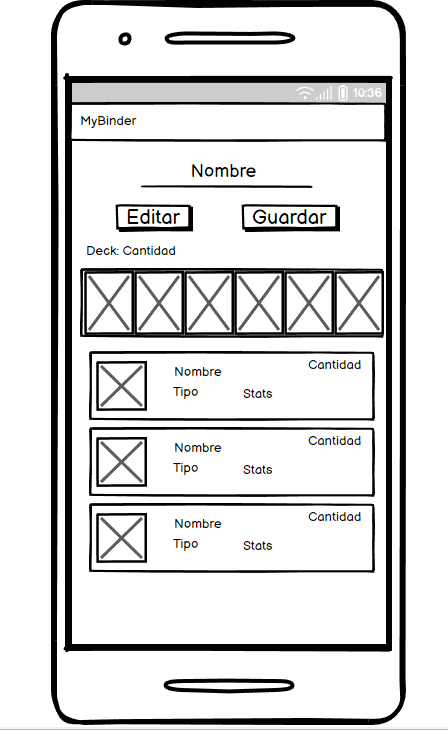
* **Detalles**

****

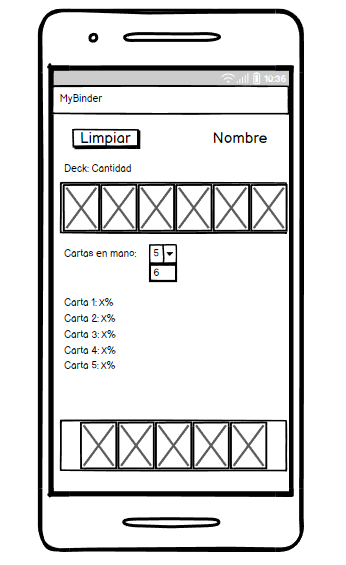
* **Mis Mazos**

****

* **Nuevo/editar Mazo**

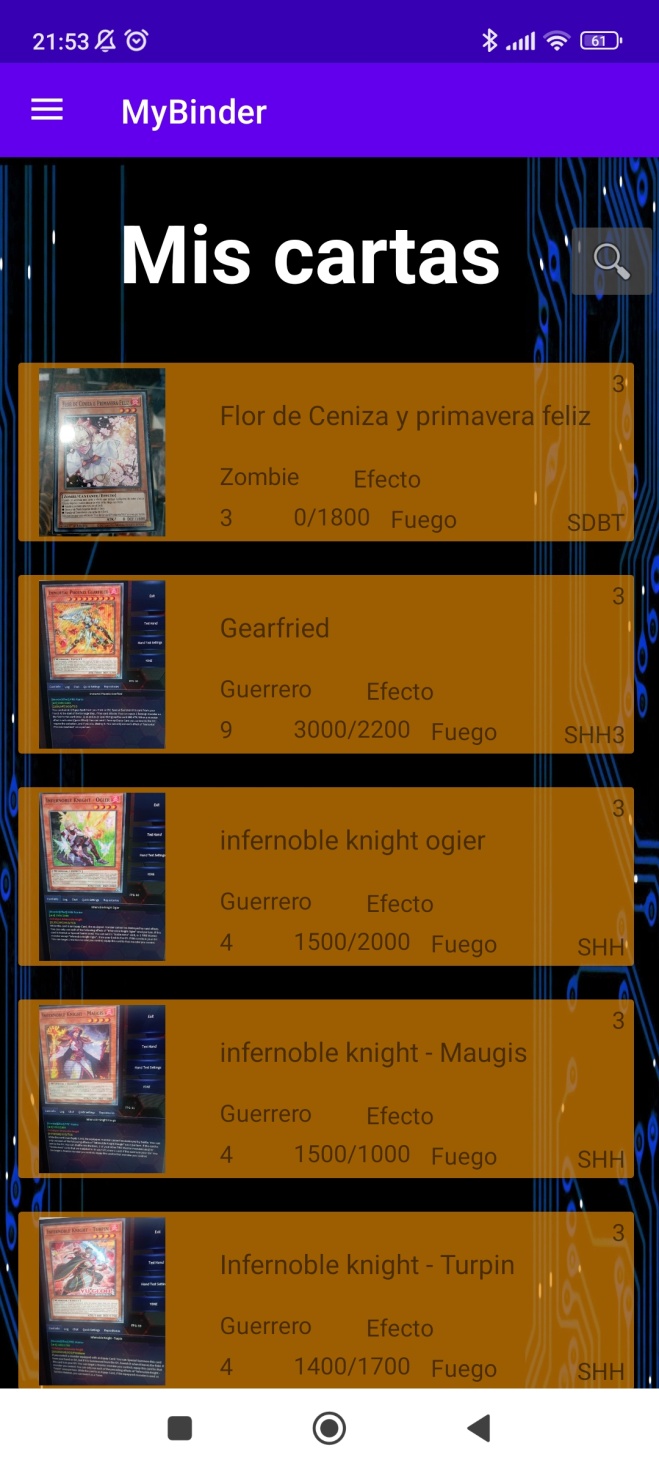
****

* **Probabilidades**

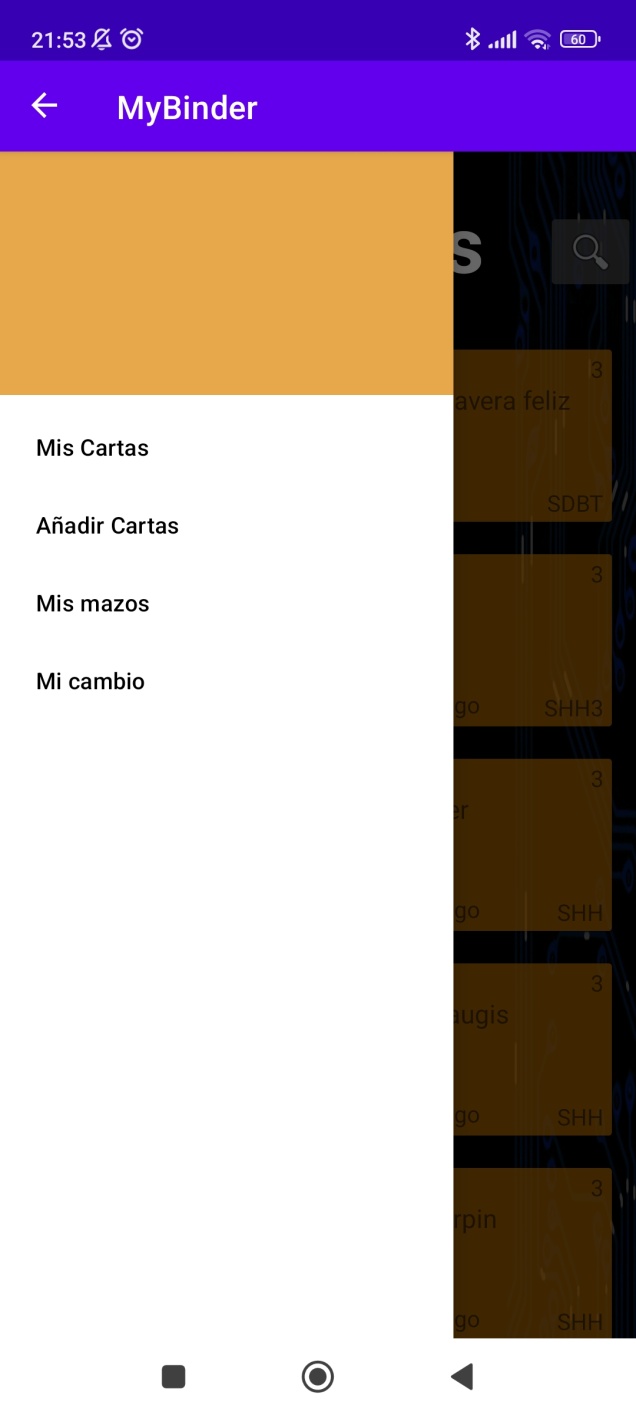
****

# 3. Diseño final de la aplicación

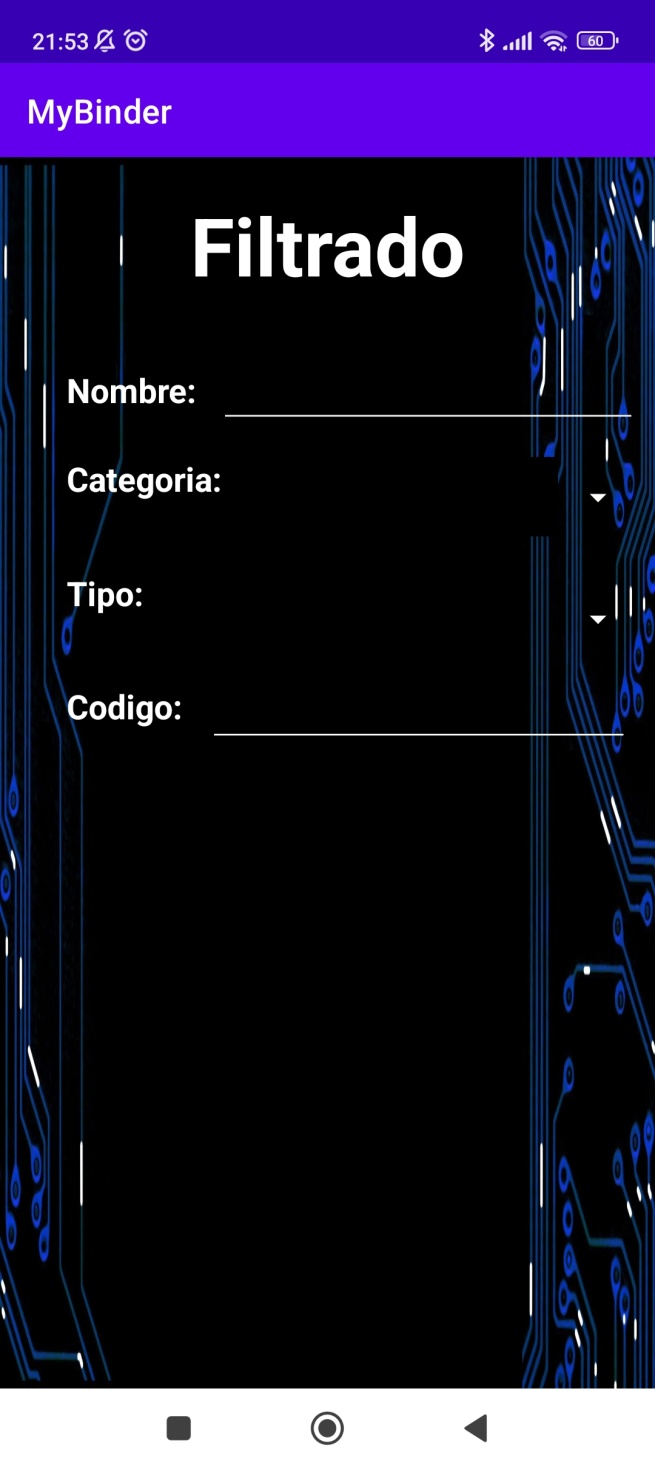
* Diseño de mis cartas



* Diseño del menú



* Diseño de Filtrado



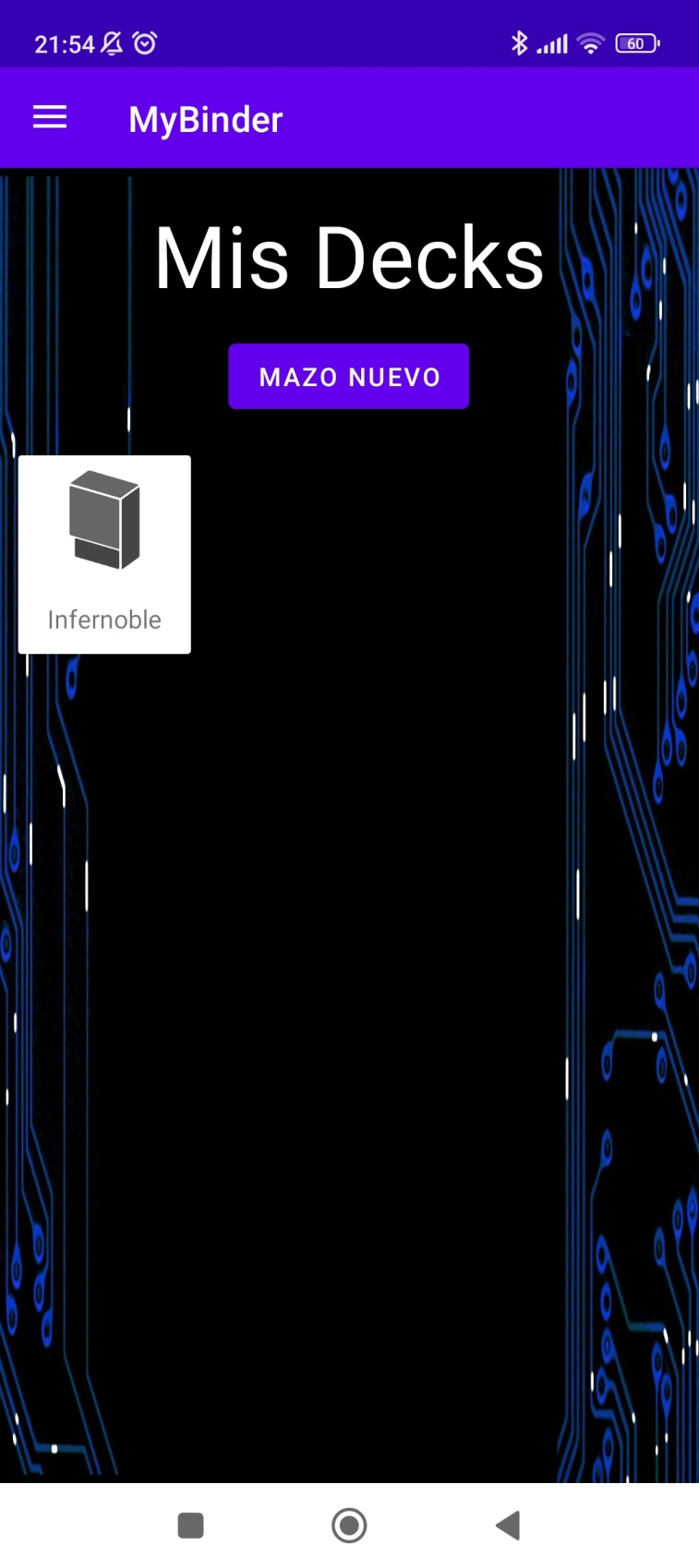
* Diseño Añadir Cartas/Editar



* Diseño de Detalles



* Diseño Mis Decks



* Diseño Crear/Editar Deck



* Diseño Probabilidades



# Matemáticas en el proyecto.

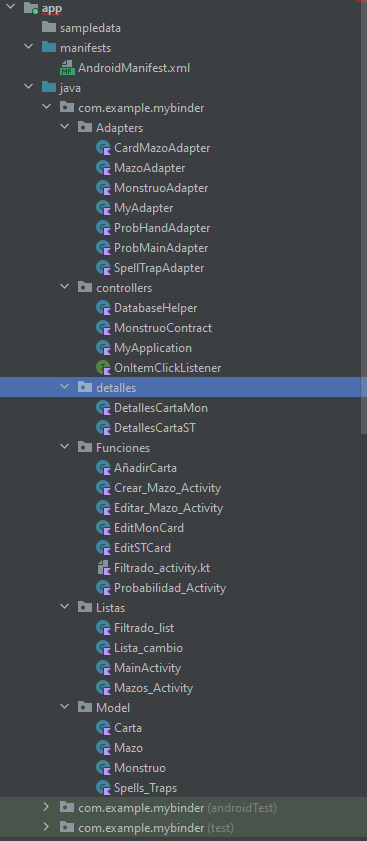
En el proyecto, hay una funcionalidad en la que se basa en calcular las probabilidades para ello se ha usado una fórmula que es la siguiente:



Esta parte de código se usa para calcular las probabilidades de robar la carta que tú elijas del mazo dependiendo de cuantas copias queden en el mazo, cuantas cartas haya en el mazo restante y cuantas cartas tengas en la mano.

Aparte de esta parte del código, ya no hay mas partes con matemáticas complejas.

# 5. Estructura del proyecto

****

Como se puede ver en la imagen se ha ordenado las activities de la siguiente manera:

Adapters: En la carpeta adapter hemos metido todos las adapters que rellenan las listas que tenemos en nuestro proyecto.

Controllers: En esta carpeta hemos metido todo lo relacionado con la base de datos y algunas interfaces.

Detalles: En esta carpeta solo metemos los dos activity de detalles que tenemos.

Funciones: En esta carpeta hemos metido todas las activity que son funcionales y acciones que el usuario puede realizar dentro de la a aplicación.

Listas: En esta carpeta están todas las listas que compone el programa.

Model: Por ultimo la carpeta model se usa para guardar todos los modelos de los objetos que tenemos y usaremos en el programa.

# 6. Finalidad

La finalidad de este proyecto es realizar una aplicación que gestione las cartas que estén en la posesión del usuario de manera rápida y sencilla para poder ayudar a la gente nueva o ya mas iniciada en uno de mis “hobbies” favoritos como es el jugar a juego de cartas coleccionables (TCG para abreviar). La manera de ayudar con esta aplicación es sencilla, el usuario podrá añadir toda su colección de cartas dentro de la aplicación y poder hacer sus mazos y a la hora de querer construir un mazo tiene la herramienta probabilidades, donde podrá ver que chances hay de robar la carta que desea en manos iniciales de tanto turno 1 como turno 2, ya que dependiendo del turno en el que empieces a jugar tendrás mas o menos cartas en mano.

Como se menciono anteriormente la finalidad de este proyecto es ayudar a la comunidad de jugadores en la que estoy, pero no me gustaría limitarme solo a ella también me gustaría poder expandir la aplicación para ayudar a otras comunidades de otros juegos de cartas de una manera que se comentara en otro punto más adelante.

# 7. Fase de pruebas

Las pruebas se han ido realizando conforme los avances se iban consiguiendo, permitiendo así conseguir localizar los fallos de manera más rápida y precisas.

Además se ha dado la aplicación a algunas personas dentro del público objetivo para que probase la aplicación y diese sus opiniones sobre esta. Dentro de este público ha habido perfiles distintos, más experimentados y menos experimentados, tanto en el ámbito de la funcionalidad de la aplicación como de cómo funciona el código (personas que ha estudiado o trabaja como programador).

Los resultados obtenidos son los siguientes:

Sujeto de pruebas 1: “La aplicación está bien, y es algo que yo usaría, es cómoda de usar y tiene potencial para que la pueda usar.”

Sujeto de prueba 2: “La aplicación está bien, pero hay que pulir cosas como los formularios, hay demasiados huecos vacios que afea la aplicación, aparte de eso tienes que tener cuidado con los campos que se dejen vacios, usa validaciones para ello.”

Sujeto de pruebas 3: “La aplicación es fea, y no me parece intuitiva, me cuesta saber que es que”

Sujeto de pruebas 4: “Me encanta que puedas hacer fotos a las cartas y que puedas filtrar por la expansión donde salió la carta, así puedo ver que mas salió en ese mismo sobre”

Sujeto de pruebas 5: “Es interesante para controlar todo lo que tengo ya que a veces se me olvida si tengo algo que los demás buscan y a si puedo buscar de manera más sencilla las cosas que tengo pero creo que es bastante simple y habría que añadir más cosas para que fuera mejor para usarlo”

Sujeto 6: “No tengo muchas cartas pero aun así me parece bastante chula la aplicación, podrías intentar añadir más cosa como una opción para poder probar los mazos que haga”

# 8. Conclusiones

Gracias a la realización de este proyecto he aprendido mas del lenguaje kotlin y de las cosas que se pueden hacer (por ejemplo llamar a la aplicación de fotografías o de crear pdfs dentro de la memoria del teléfono).

Aun así siento que la aplicación se puede mejorar y en el futuro me gustaría poder ampliar la aplicación con las siguientes especificaciones:

* Poder soportar varios juegos de cartas simultáneos (por ejemplo los más famosos que son “Magic” y “PokemonTCG”) para poder llevar un registro de otros juegos de cartas que se jueguen.
* Poder importar y exportar archivos de los mazos y de las cartas para poder insertar más fácilmente las cartas que compres o vendas a otros usuarios.
* Tener la base de datos en la nube , aun que no ocupe mucha memoria en el teléfono igual es una ventaja mayor poder tenerlo en la nube y además esto podría ayudar a hacer conexiones con otros usuarios dentro de la aplicación para poder ver incluso de manera más sencilla el cambio que estos tengan.
* Añadir una opción de prueba de mazos, donde podrás aprender a usar estos y practicar los combos que tenga.
* Poder usar un api para poder insertar más fácilmente las cartas de la colección y ahorrarnos los formularios de crear y editar cartas.
* Mejorar Visualmente la aplicación.

# 9. Problemas en el desarrollo.

Durante el transcurso de este proyecto ha habido varios problemas y obstáculos que se han tenido que cruzar, uno de ellos el primero y el más difícil fue, buscar una api con la base de datos más completa posible, la manera de superar este obstáculo fue literalmente reenfocando la aplicación desde cero, al principio la aplicación iba a contar con una base de datos ya hecha y tendría menos funcionalidades pero las que tendría serian algo más extensas y complicadas.

Para llegar a esta conclusión gaste una semana y media del tiempo asignado para buscar una api, y otra media experimentando con archivos Json por una solución que se había pensado. Ya después de haber cambiado tanto de plataforma como planteamiento del proyecto aun se planteaban nuevos problemas, como por ejemplo cuando se pulsaban algunos elementos no se actualizaban las listas que se mostraban, para solucionar esto sencillo, solo había que poner DataSetChange después de cada modificación de los datos de estas listas, este error fue bastante recurrente durante todo el desarrollo del programa. Aparte de esto hubo problemas con la base de datos, un ejemplo de esto el filtrado no funciona del todo bien, hay un bug el cual no muestra todos los elementos que se filtran a no ser que se hayan realizado un “update” de los datos de ese objeto, por mucho que investigue y debuge no puedo encontrar el fallo que está provocando todo este bug.

# 10. Bibliografía

Guillermo Hernández Manzano. (2022, 5 octubre). *Uso de la Cámara en KOTLIN* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=USSkwTdR_no>

*HIPERGEOMETRICA*. (s. f.). https://www.uv.es/ceaces/base/modelos%20de%20probabilidad/hipergeometrica.htm

eddy. (2022, 5 febrero). *COMO HACER UN RECYCLERVIEW SENCILLO CON KOTLIN* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=D93sD08Hly4>

Viera Class. (2022, 20 julio). *Base de datos Sqlite Dispositivos Móviles en Kotlin* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=23IFKNIZm4c>

dev.xcheko51x. (2022, 22 abril). *Crear PDF usando PDFDocument Kotlin Android Studio* [Vídeo]. YouTube. https://www.youtube.com/watch?v=bAz3GelYVjQ