



# Genshin Impact Team Building



**Genshin Impact Teambuilder**

**Proyecto Final de Grado**

**Autor: Rubén Grau Padilla**

**Grado Superior:** Desarrollo de Aplicaciones Multiplataforma

Curso académico 2020-2021

# **ÍNDICE**

# **1. INTRODUCCIÓN**

## **1.1 Resumen y justificación del proyecto**

Hace 8 meses apareció un juego llamado Genshin Impact que provocó bastante revuelo en lo que se llama, “la comunidad gaming”, y fue tan grande que después de tan solo un mes desde su salida, entró en los “gaming awards”, que son premios otorgados a los juegos mejor valorados por la comunidad, y fue candidato a ser el juego del año.

Dicho esto y centrándonos en el juego en si, este es un juego que se encuentra en la categoría de los “gacha”, esto quiere decir que básicamente son lo contrario a los juegos clásicos en los que adquirías personajes avanzando en la historia.

En este tipo de juegos los personajes se adquieren usando una moneda del juego que consigues por eventos, recompensas o con dinero real y, además, aparte de su historia que avanza conforme avanza el tiempo ya que meten misiones de historia en cada parche, tenemos una zona que requiere de tener diferentes personajes bien equipados y de tener conocimientos del juego para superarla llamada, el abismo.

Como bien he mencionado al principio, han pasado casi 9 meses desde que este juego apareció, por lo que por lo general todo el mundo puede vencer el contenido actual del juego, pero, hay gente que ya sea por ver poco contenido del juego o simplemente no fijarse, no es capaz de vencer ese contenido.

Mi aplicación está hecha para ese tipo de gente que solo quiere jugar sin tener que calentarse la cabeza, ya que con esta tendrán indicaciones detalladas de como mejorar su cuenta en pocos minutos usandola.

## Abstract

Eight months ago, a game appeared from nowhere, making a lot of noise in the gaming community, believe it or not but it was so big that, having only a month of live, it got into “the gaming awards”, these awards are the ones awarded by the community and genshin got into the best game of the year category.

That said and getting a bit deeper on the game itself, it falls in the category of “gacha” games, these are the ones that, unlike the classical games in which you get new characters as you progress into the history, you get characters using a special coin that you get in the game.

These coins can be obtained in events, rewards the the company gives for playing the game or using real money to buy them. Also, the way the history works is different too, because in this kind of games you get history missions as the game gets updates and it has endgame content called the abyss, this game mode requires you to have well built characters and some knowledge of the game.

As i said at the beginning, it’s been almost nine months since the game came out, so in general, almost all the players are able to complete the abyss, but there’s people that are not able to beat it mainly because they dont really look to progress as they should.

My application is made for that kind of person that doesn’t want to loose a ton of time looking how to progress int the game and just want to chill out while playing, and thats because they’ll only need a few minutes to know how to get better character builds and teams.

## **2.ANÁLISIS**

En la época que salió el juego, que aún estábamos con confinamiento, este tipo de juegos en los que puedes tirar horas y horas sin aburrirte han florecido bastante, y por eso quiero ayudar a que la gente que se pueda sentir frustrada con el juego por no ser capaces de avanzar y que así puedan disfrutar de este.

### **2.1 Objetivos específicos**

En función del tiempo disponible, procedo a dividir los objetivos en dos categorías:

#### **Realizados:**

- Recomendaciones específicas por personaje.
- Dos zonas diferentes, una para el equipo y otra en la que se encuentren todos los personajes.
- Recomendación de equipos.

#### **Deseables(futuras mejoras):**

- Recomendación más específica sobre el equipo, viendo este personaje por personaje y dando los consejos adecuados en cada situación.
- Pestaña de equipo funcional, ya que por ahora no se pueden añadir ni quitar personajes del equipo.
- Recomendaciones de personajes más específicas dependiendo del equipo en el que se encuentren.
- Un apartado en el que poder poner los artefactos de tus personajes y así poder saber si necesitas trabajar más en ellos o no.
- Un login para poder guardar y actualizar los datos de tus personajes y así tener un progreso más eficaz en el juego.

## 2.2 Fases de realización

En este apartado se describen las fases seguidas a lo largo de la realización del proyecto:

- Estudio: Investigación sobre las posibles tecnologías a utilizar. Indagación sobre aplicaciones parecidas a la que busco realizar.
- Análisis: Definición de requisitos con el objetivo de definir exactamente qué es lo que se desea construir.
- Diseño: Tras finalizar las fases de estudio y análisis, y con el listado de requisitos, se procede a diseñar la arquitectura de la aplicación. Esto incluye diversos diagramas que definirán el diseño de pantallas, interfaz y la base de datos.
- Implementación: Programación de la aplicación móvil en Java.
- Pruebas: Una vez finalizadas todas las fases anteriores, se ha realizado una serie de pruebas generales.
- Documentación: La documentación es algo indispensable para el correcto mantenimiento de cualquier producto software. Tanto a nivel de código como de documento.

### 2.2.1 Planificación

La planificación para la realización del proyecto se ha realizado de la siguiente forma:

	Estudio	Análisis	Diseño	Implementación	Pruebas	Documentación
Abril	X					
Mayo		X	X			
Junio				X	X	X

## 2.3 Tecnologías y herramientas utilizadas

- Android SDK y Android Studio

El Android Studio junto con Android SDK contiene las herramientas necesarias para el desarrollo de aplicaciones Android. Contiene emuladores y un depurador bastante potente.

- Base de datos con Room

Room proporciona una capa de abstracción sobre SQLite que permite acceder a la base de datos sin problemas y, al mismo tiempo, aprovechar toda la potencia de SQLite.

- Parcelables

Los objetos Parcelable y Bundle están diseñados para usarse entre límites de procesos como con transacciones de IPC y Binder, entre actividades con intents y para almacenar el estado transitorio en los cambios de configuración.

- Fragments

Un fragment representa un comportamiento o una parte de la interfaz de usuario en una fragmentActivity.

- Glide

Glide es una popular librería Android de código abierto para cargar imágenes, videos y GIFs animados. Con Glide puedes cargar y mostrar medios de muchas fuentes diferentes, tales como servidores remotos o el sistema local de archivos.

- Recycler view

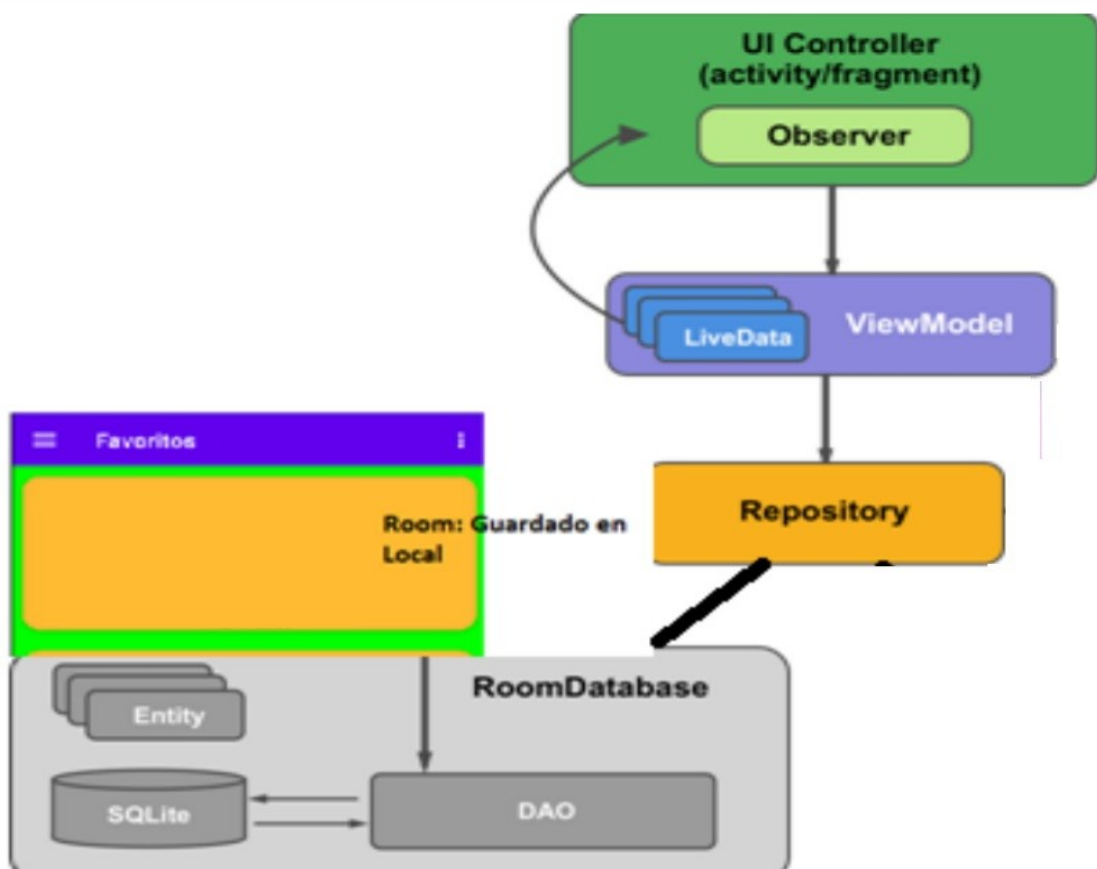
RecyclerView facilita que se muestren de manera eficiente grandes conjuntos de datos. Tú proporcionas los datos y defines el aspecto de cada elemento, y la biblioteca RecyclerView creará los elementos de forma dinámica cuando se los necesite.

- Git

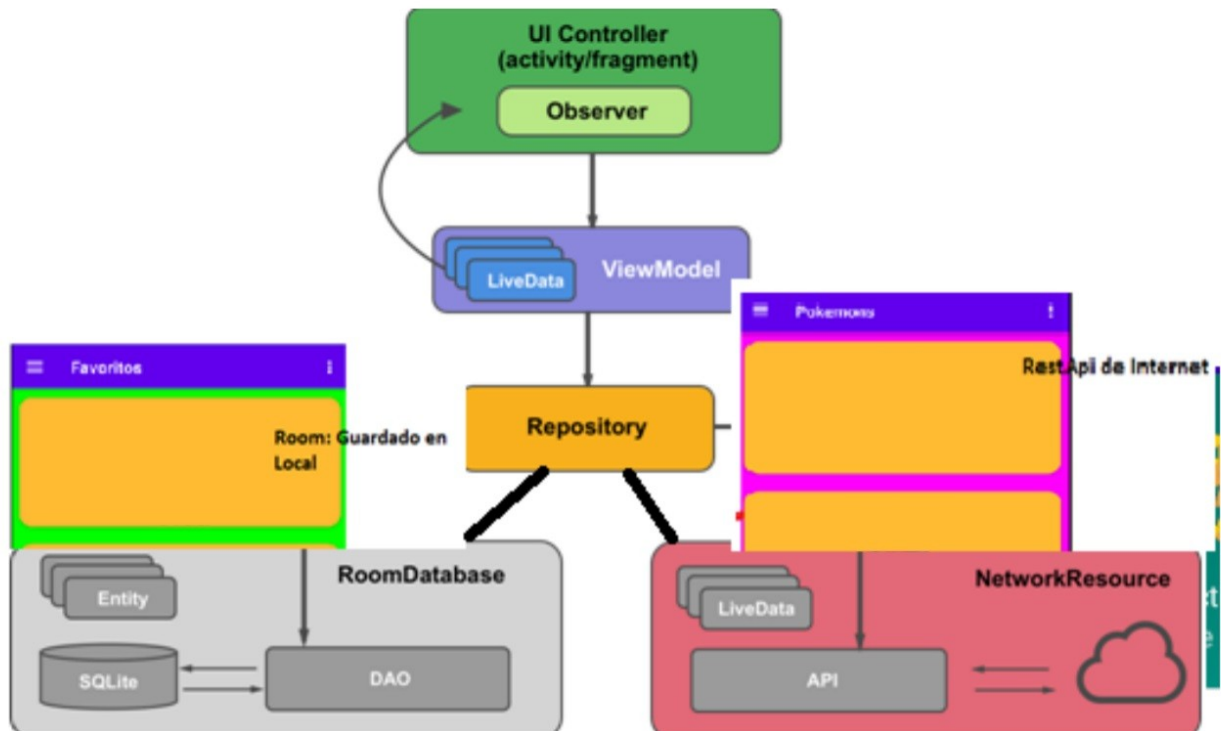
Control de versiones compatible con Android Studio.

## 2.4 Casos de uso

En el estado actual de la aplicación solo hay un caso de uso real:



Y en el futuro se buscaría algo así:

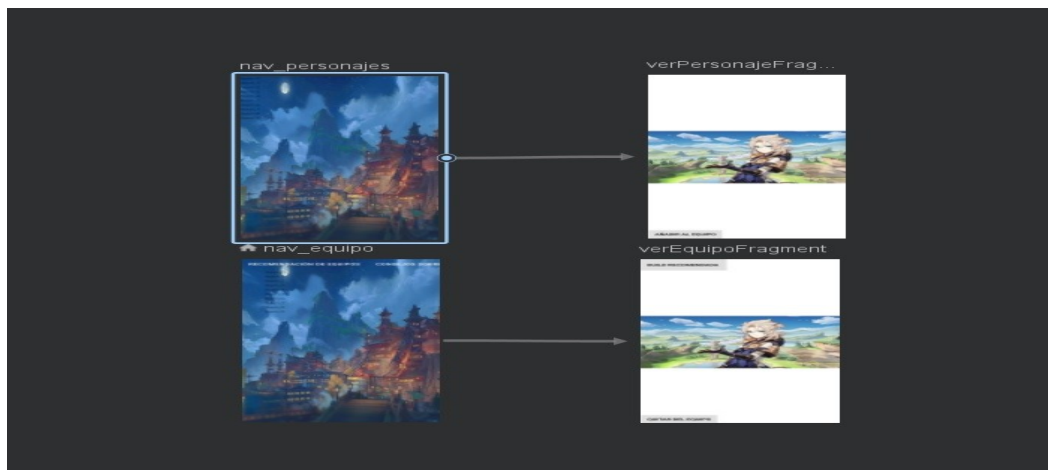


## 3.DISEÑO

### 3.1 Prototipo

El prototipo está realizado en base al `mobile_navigation`. En este se muestran en vertical, pero dentro de la aplicación se muestra todo en horizontal:

En el diseño final, todo se muestra en



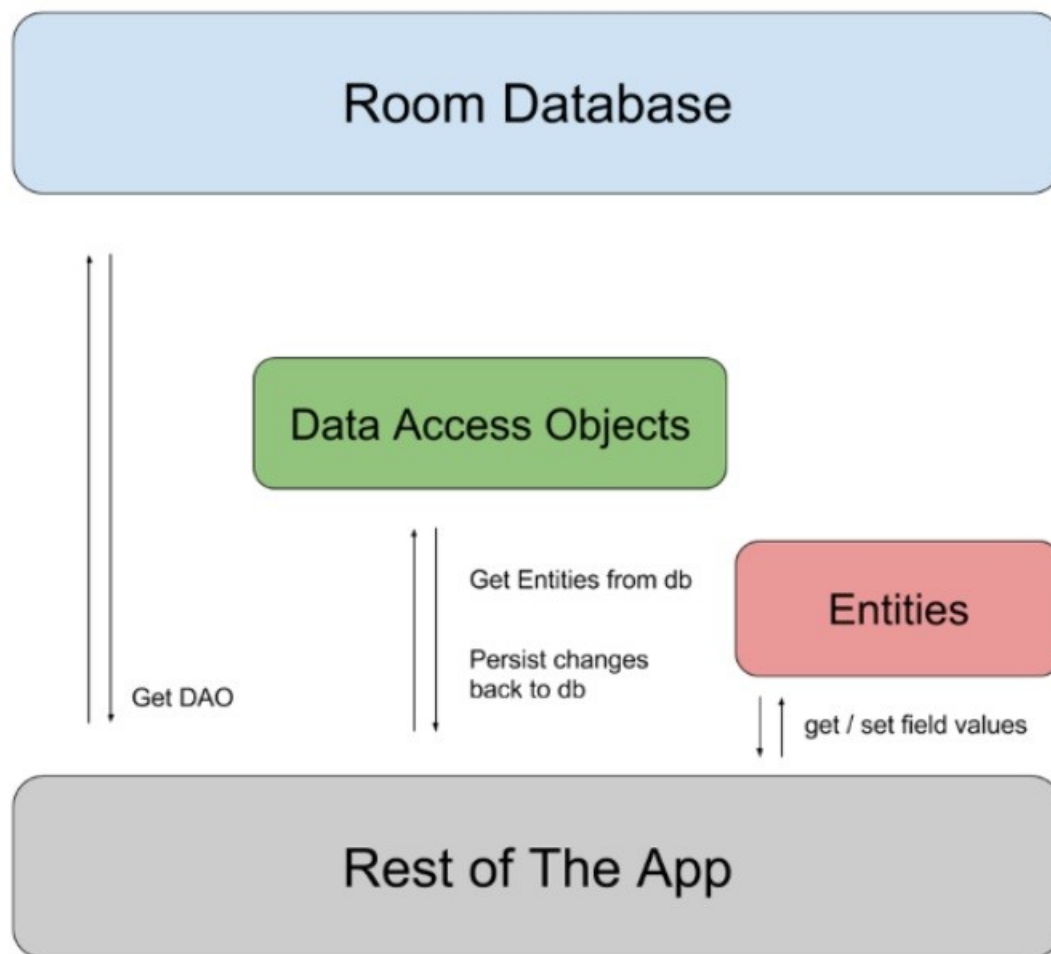


horizontal y se utilizan iconos relacionados con el juego para que los jugadores se acostumbren a ellos.

Al ser una aplicación pensada para ser utilizada durante poco tiempo, se ha buscado un diseño sencillo y práctico de ventanas.

## 3.2 Arquitectura

Se ha utilizado la arquitectura de la base de datos de room:



Como la app se ejecuta en un solo proceso, se sigue el patrón de diseño singleton cuando se crea una instancia de un appDataBase. Cada instancia de RoomDataBase es bastante costosa y rara vez necesitas acceder a varias instancias en un mismo proceso.

### 3.3 Modelado de datos

RoomDataBase es una base de datos basada en Sqlite, la cual se define como HSQLDB (Hyperthreaded Structured Query Language Database), basado en HypersonicSQL, es un gestor de bases de datos relacional y se escribe completamente en Java.

Todo sale de una sola base de datos, que es la de personajes, y de la cual salen dos funciones diferentes:

- Equipo:

En esta se busca tener 4 personajes, de los cuales poder ver como se construye cada uno, aunque en la versión actual se encuentran todos los personajes por falta de tiempo para implementar el añadir y quitar personajes del equipo.

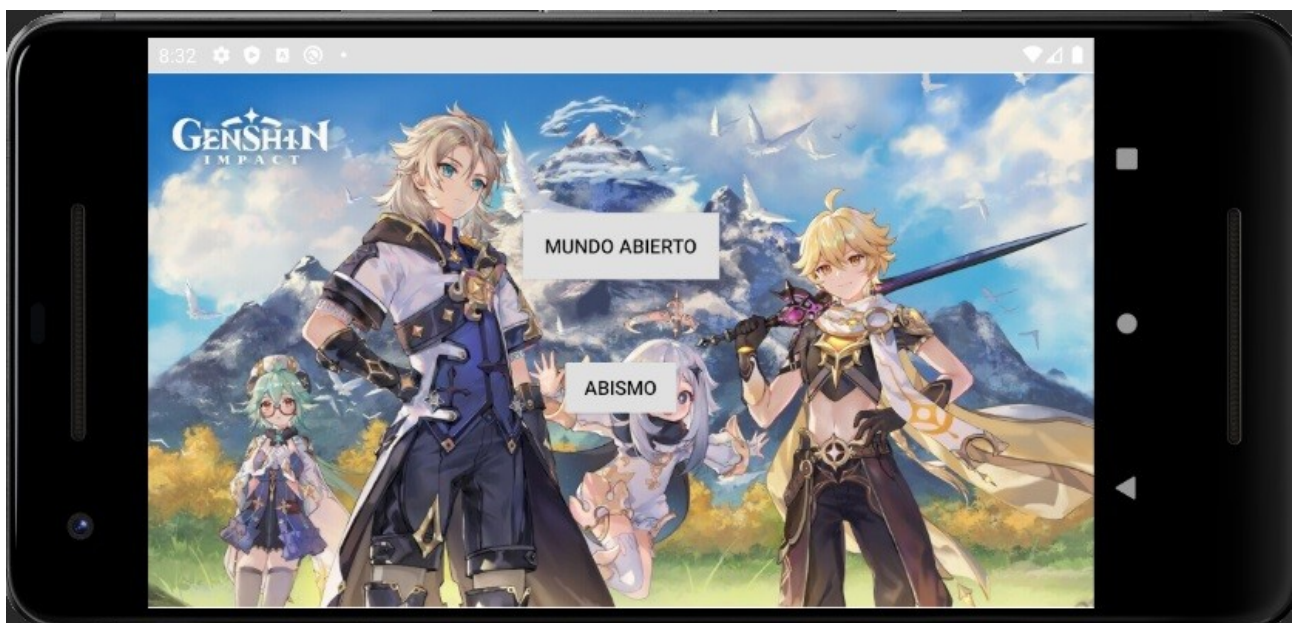
- Personajes:

En esta simplemente se busca poder ver todos los personajes disponibles y poder añadirlos al equipo.

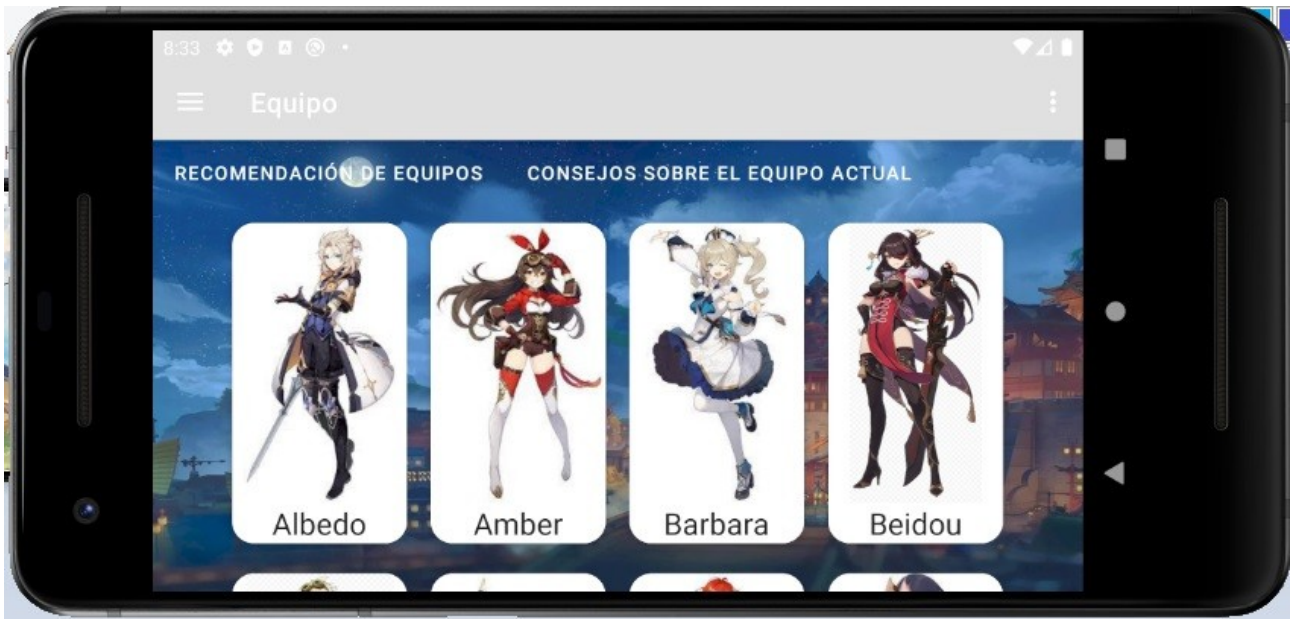
### 3.4 Vistas de la aplicación

A continuación voy a mostrar algunas vistas de la aplicación:

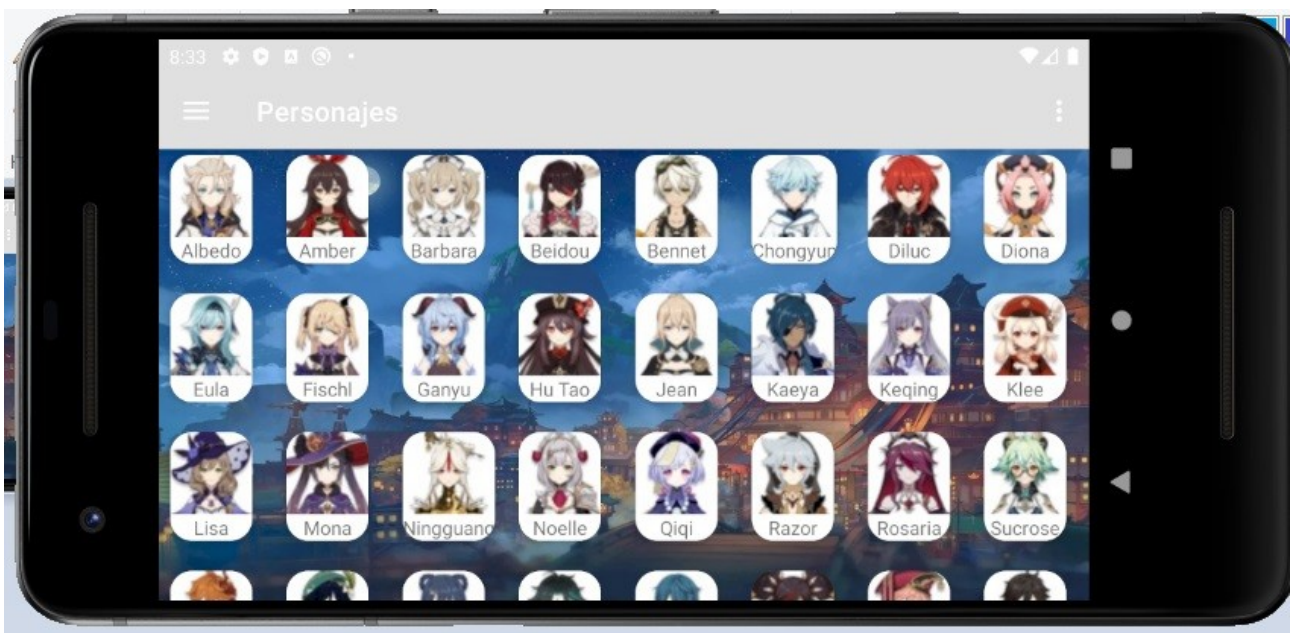
**Pantalla de inicio:**



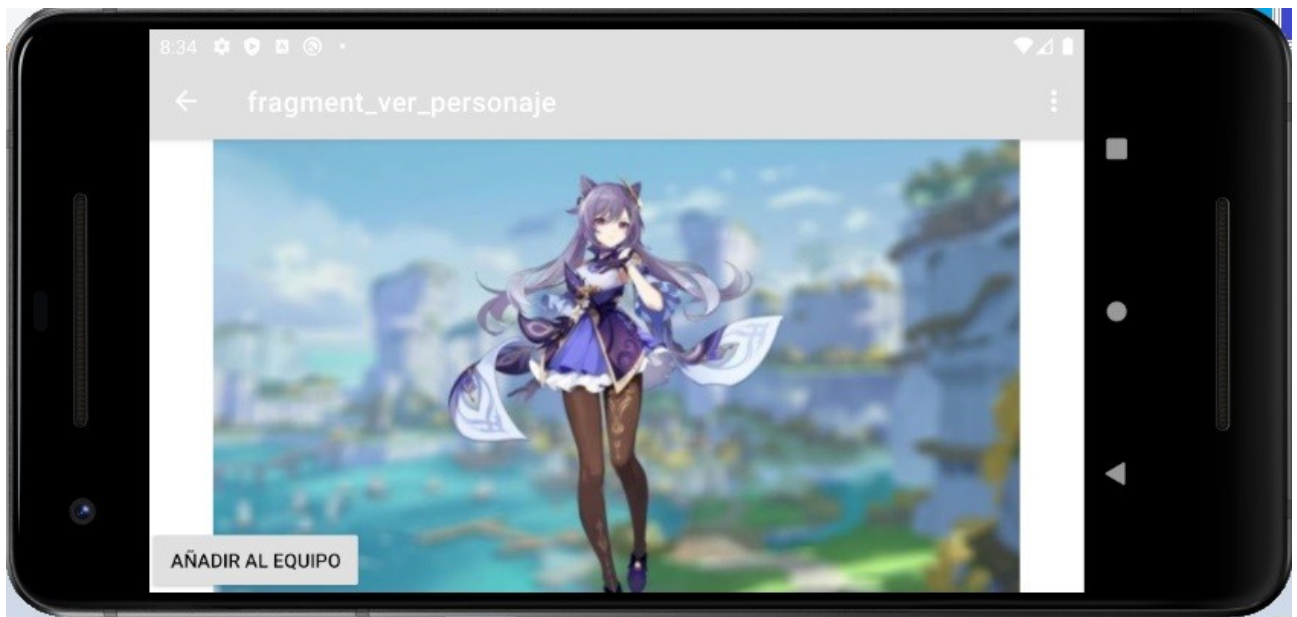
**Pestaña de equipo(no se deberían ver mas de 4 pero por funcionalidad por el momento están todos los personajes):**



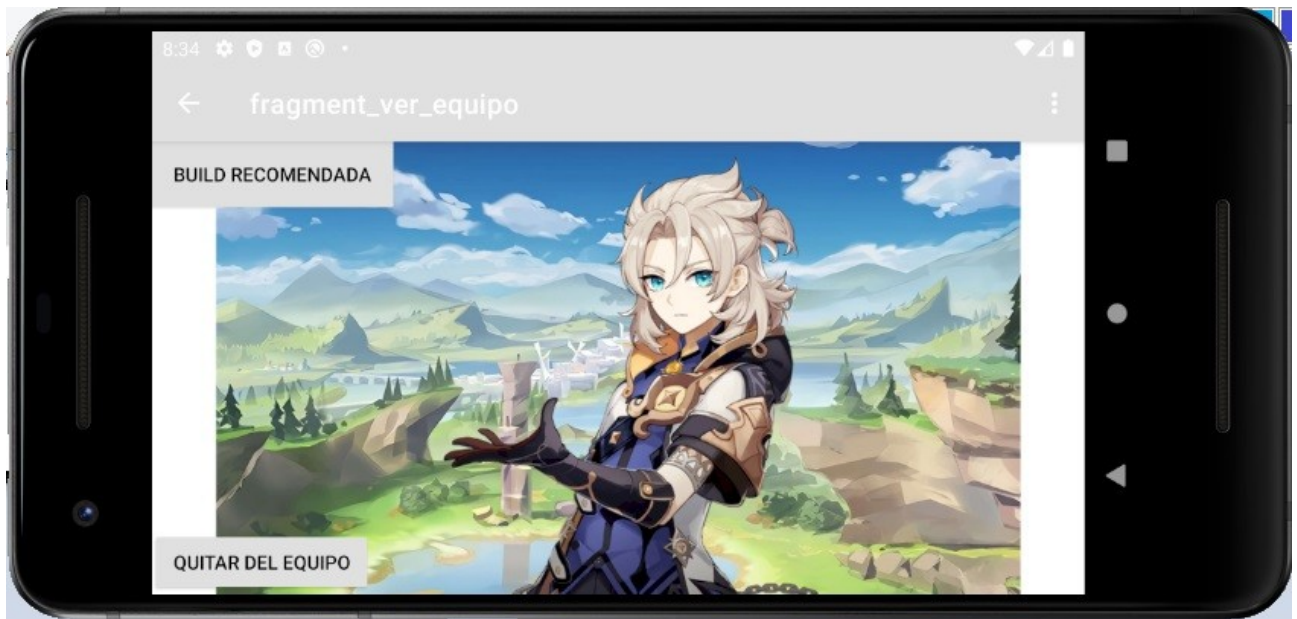
**Pestaña de personajes:**



**Al pulsar un personaje(flata conseguir que android estudio le haga resize a la imagen pero no me funciona):**



**Al pulsar un personaje del equipo(lo mismo con la imagen que en el anterior):**





## **4. IMPLEMENTACIÓN**

### **4.1 Configuración del entorno de desarrollo**

Para la implementación de la aplicación se ha usado:

- Portatil con Intel i7 y 8gb de RAM.

Lo primero fué la instalación de Android Studio y las herramientas del SDK, junto a Git para el control de versiones.

- Android Studio 4.1
- Git 2.24

### **4.2 Desarrollo**

Después de configurar el entorno de desarrollo se creó la aplicación en Android Studio desde una plantilla vacía que te llevaba a una *Navigation Drawer Activity*.

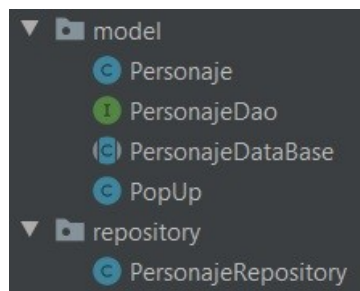
#### **4.2.1 RoomDataBase**

Lo primero fué configurar RoomDataBase en el proyecto, añadiendo al gradle:

```
implementation 'androidx.room:room-runtime:2.2.6'
```

```
annotationProcessor 'androidx.room:room-compiler:2.2.6'
```

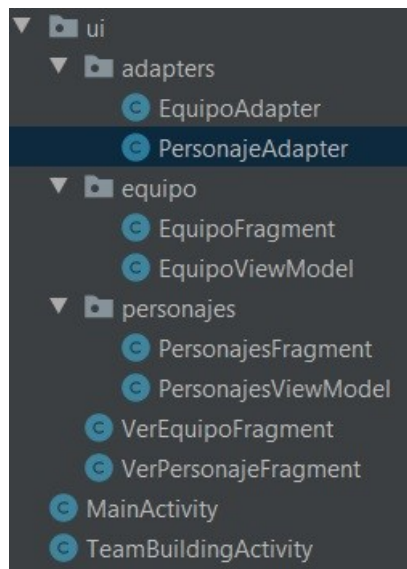
Para acceder a Room se utiliza una clase DAO que se crea en base a la clase principal y se controlan las acciones desde el repository.



#### **4.2.2 Fragmentos**

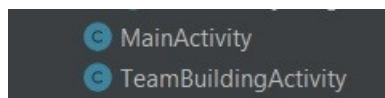
Hay dos clases encargadas de mostrar los fragment de equipo y de personajes respectivamente, ya que las vistas son diferentes.

Se utilizan los métodos `onCreateViewHolder()`, `onBindViewHolder()` y `getItemCount()` que se heredan de `RecyclerView.Adapter<>` que a su vez está heredando de `RecyclerView.ViewHolder`.



### 4.2.3 Actividades

Estas se limitan a esperar información y realizar sus respectivas tareas.



### 4.2.4 Utilidades















Adicionalmente, hay un paquete de utilidades que basicamente se encarga de cargar la imagen que hay que cargar en cada actividad que la requiera.

## 5. PRUEBAS

### 5.1 Pruebas en dispositivos

Se han realizado multitud de pruebas en diferentes dispositivos Android con versiones:

- Android 9.0 (API 28)
- Android 10.0 (API 29)
- Android 11.0 (API 30)

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Galaxy Nexus API 30		720 x 1280: xhdpi	30	Android 11.0 (Google APIs)	x86	513 MB	  
	Phone		1080 x 1920: 420dpi	29	Android 10.0 (Google Play)	x86	9.9 GB	  
	Pixel 4 API 28		1080 x 2280: 440dpi	28	Android 9.0 (Google Play)	x86	513 MB	  

Prueba	Resultado
Inicia la aplicación	OK
Salta el aviso del boton abismo en proceso	OK
Entra correctamente a la pestaña de equipos	OK
Entra correctamente a la pestaña de personajes	OK
Entra en el fragment del personaje desde equipo	OK
Entra en el fragment del personaje desde personajes	OK
Salta el aviso al usar el boton de consejo equipo actual	OK
Se puede hacer zoom en la imagen de equipos recomendados	OK
Los popup aparecen correctamente	OK

### 5.2 Evaluación

En la fase de diseño se tuvieron en cuenta las diversas opiniones de amigos tras realizar el prototipo. Las mismas personas han realizado distintas acciones sin ayuda alguna una vez acabado:

- Iniciar la app
- Intentar acceder al botón Abismo sin conseguirlo
- Entrar correctamente a las pestañas de equipo y personajes
- Intentar pedir consejo sobre el equipo actual y que les salte el aviso de que está en desarrollo

- Pueden hacer zoom en la imagen que aparece cuando pulsas el botón de recomendación de equipos

Una vez hecho esto, han respondido varias preguntas y se se llegado a la misma conclusión: el funcionamiento es sencillo y directo. A todos ellos les ha parecido útil ya que son del tipo de persona para los que está hecha esta aplicación.

## **6 CONCLUSIÓN**

El resultado de realizar este proyecto ha sido satisfactorio a nivel personal pues he adquirido y asentado una gran cantidad de conocimientos nuevos.

Cosas como las popUpWindows, que era un concepto nuevo y el Navigation Drawer, con el cual no estaba muy comodo ya que se trabaja en una actividad que era bastante guiada. Trabajar con ellos conlleva utilizar fragmentos, los cuales son mas complicados que utilizar exclusivamente actividades.

La planificación temporal ha sido bastante mayor de las 40h del proyecto. Me he dado cuenta que realizar una aplicación en Android e intentar seguir unas líneas de calidad requiere mucho tiempo y esfuerzo(más del que he podido realizar ya que hay cosas que me gustaría haber hecho que no he sido capaz).

Como se ha comentado en el punto 2.1 Objetivos específicos, los objetivos deseables son mejoras que tenia pensadas para la aplicación pero que por tiempo no es posible su implementación. He intentado pues, que sea una aplicación sencilla y útil.

Dar las gracias a José Antonio Lopez de Merlo por responder a las dudas que he tenido durante el desarrollo y una mención especial a StackOverflow, ya que tienen respuestas a la mayoría de problemas que he tenido.



## 7. MANUAL DE USUARIO

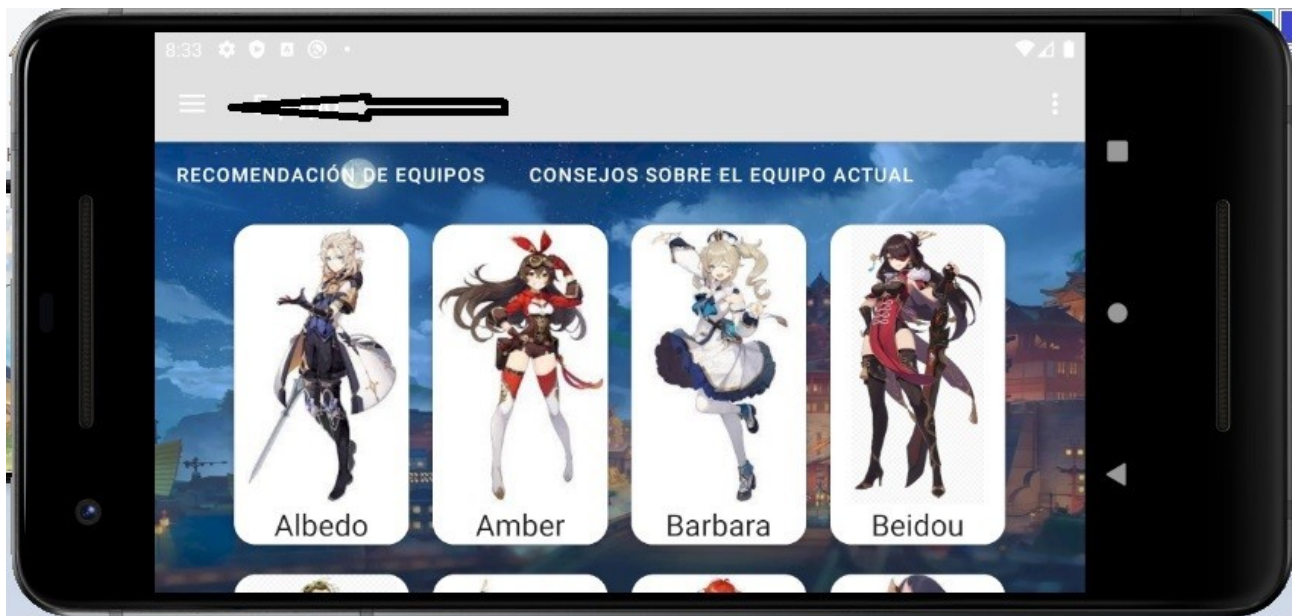
### 7.1 Acceso al teambuilder

Simplemente hay que pulsar el botón “Mundo Abierto”.



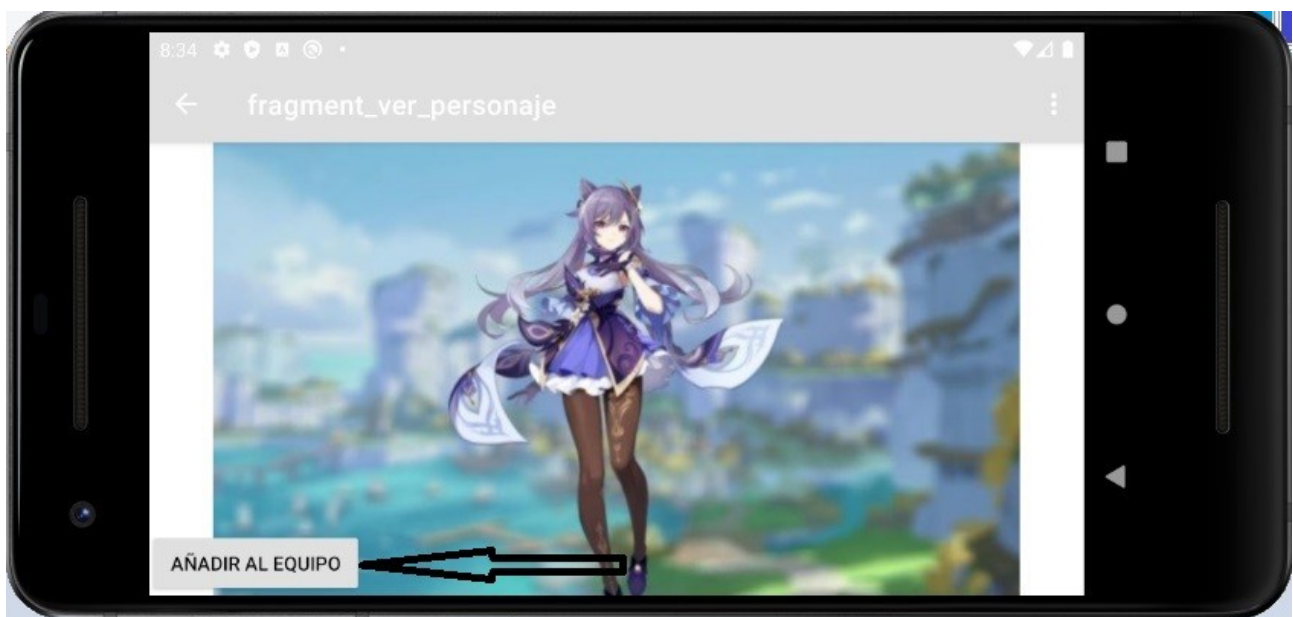
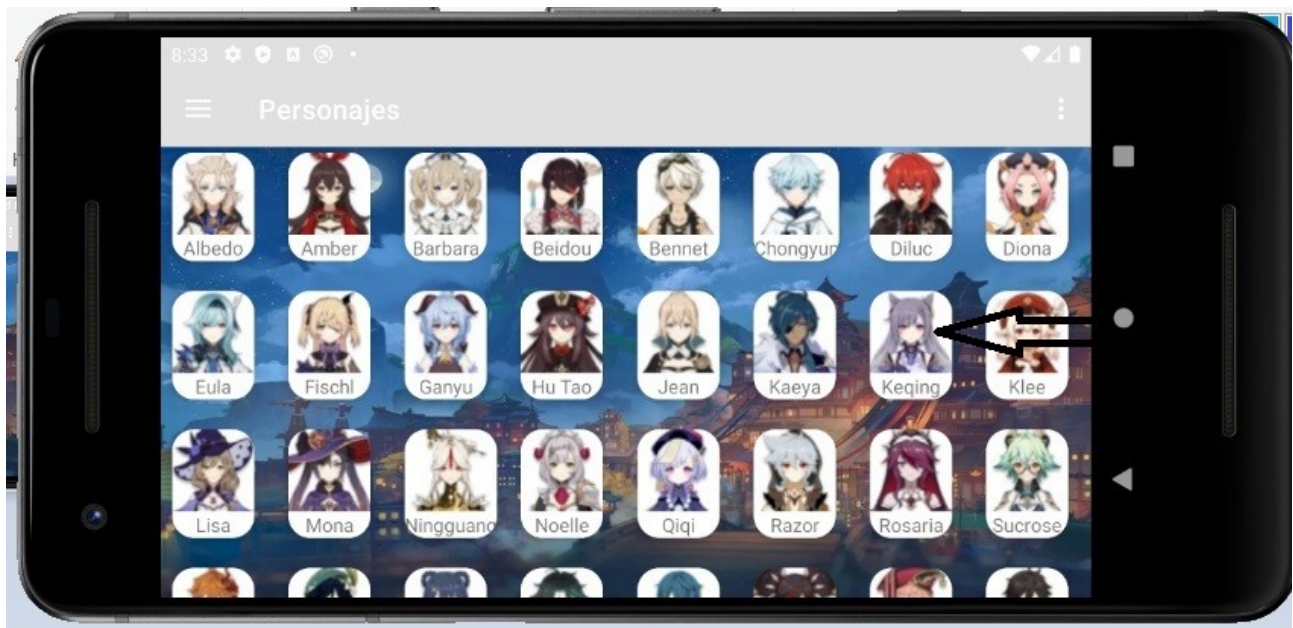
### 7.2 Moverse entre las pestañas

Pulsar el botón señalado y en este pulsar la pestaña a la que queráis acceder.



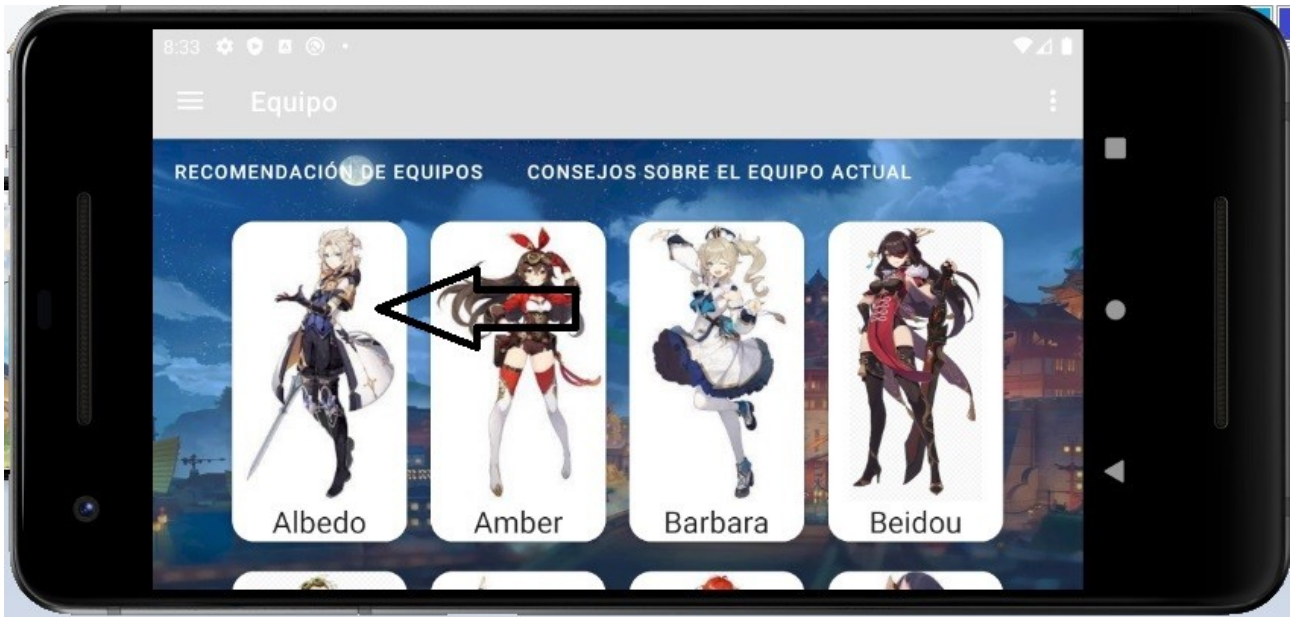
### 7.3 Añadir al equipo(en desarrollo actualmente)

En la pestaña personajes, pulsar sobre el personaje que quieres añadir y luego pulsar el botón añadir.



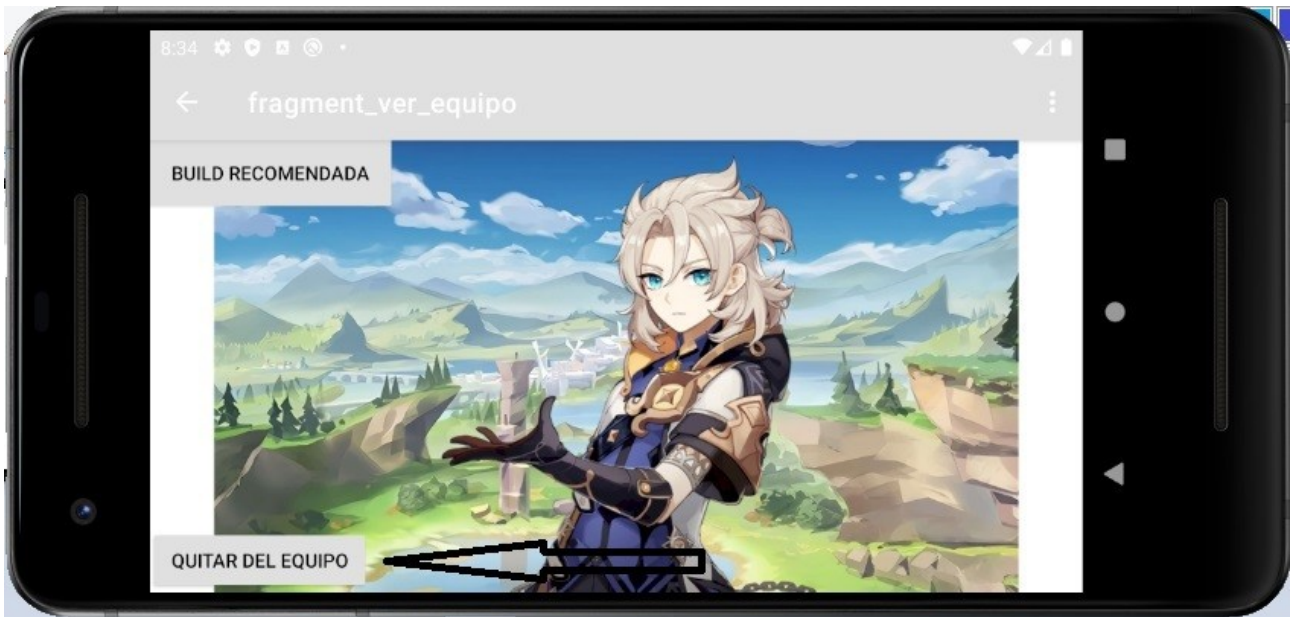
#### 7.4 Pestaña equipo acceder a personaje

Pulsar sobre un personaje en la pestaña de equipo.



##### 7.4.1 Quitar personaje del equipo(en desarrollo)

Una vez hecho el paso 7.4, pulsar el botón quitar del equipo.





### 7.4.2 Ver build recomendada

Una vez hecho el paso 7.4, pulsar el botón build recomendada.

