

GameHubTM 2025

Manual técnico

Ian Franco Iwanczuk

Introducción

GameHub es una idea que comienza con la finalidad de dar entretenimiento al usuario con juegos simples y familiares, que la mayoría de personas conocen de su infancia. Es una aplicación que estando conectada a internet, es una herramienta de ocio, al simple alcance de la mano de cualquiera de nuestros clientes.



Al ser una aplicación de juegos sencillos, nuestro público objetivo serán personas mayormente jóvenes, que estando con el teléfono quieran entretenerse unos breves minutos mientras están a la espera de alguna otra cosa. Por eso nuestro foco es, realizar juegos sencillos donde el usuario pueda ganar o perder en poco tiempo, así el mismo se puede entretener la mayor cantidad posible, en el menor tiempo posible.

Análisis del problema

La problemática vendría a la hora de la monetización, ya que una aplicación sencilla como la nuestra, es muy difícil ponerla de pago y que usuarios se interesen en ella. Ya hay muchas aplicaciones de juegos offline y con mucha más variación. Pero a partir de esto, hay ciertas cosas en las que nuestra aplicación destaca:

Debilidades

- Nuestra debilidad principal es la sencillez de la aplicación a los ojos del usuario, por más soluciones complejas que encontremos, el usuario simplemente ve nuestra interfaz y nada más. Dentro de un mercado lleno de aplicaciones como la nuestra, es complicado resaltar.

Amenazas

- La mayor amenaza son las demás aplicaciones que buscan hacer lo mismo, entretenimiento fácil y rápido para el usuario. Es una amenaza ya que hay miles de aplicaciones así, la nuestra en principio no resaltaría.

Fortalezas

- Para combatir las amenazas, nuestra aplicación lo que tiene es que, a diferencia de otras aplicaciones que tienen gran cantidad de juegos y hechos con poco esfuerzo, la nuestra tiene menos juegos pero cada uno de ellos tiene mucho más trabajo y dedicación.

Oportunidades

- Al estar lanzando una primera versión, esto no se queda quieto. Tenemos la oportunidad de seguir desarrollando juegos para nuestra aplicación, nuevos juegos más completos y más pulidos que los anteriores.

Monetización y beneficios

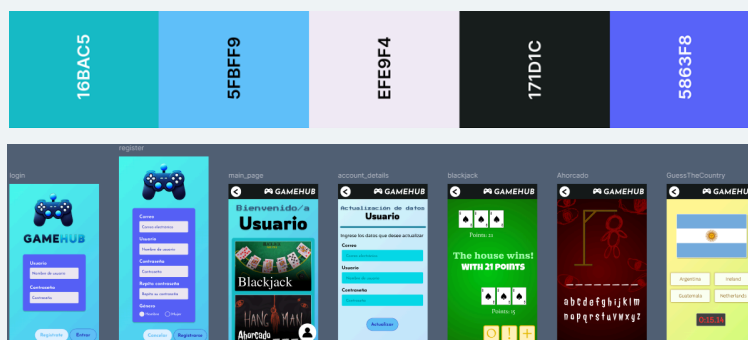
Nuestra aplicación al ser gratis, obtendrá la mayor cantidad ingresos a base de publicidad. Pero algo que haremos a diferencia de otras aplicaciones, es no atacar al usuario con ellas. Siempre habrá un banner en la parte inferior de la aplicación de publicidad, cada 5 derrotas o 5 victorias, antes de reiniciar la partida, pondremos una publicidad más larga, pero nos vamos a asegurar de no saturar al usuario.

- También, vamos a tener una suscripción mensual y permanente para remover completamente todo el contenido publicitario.

Diseño de la solución

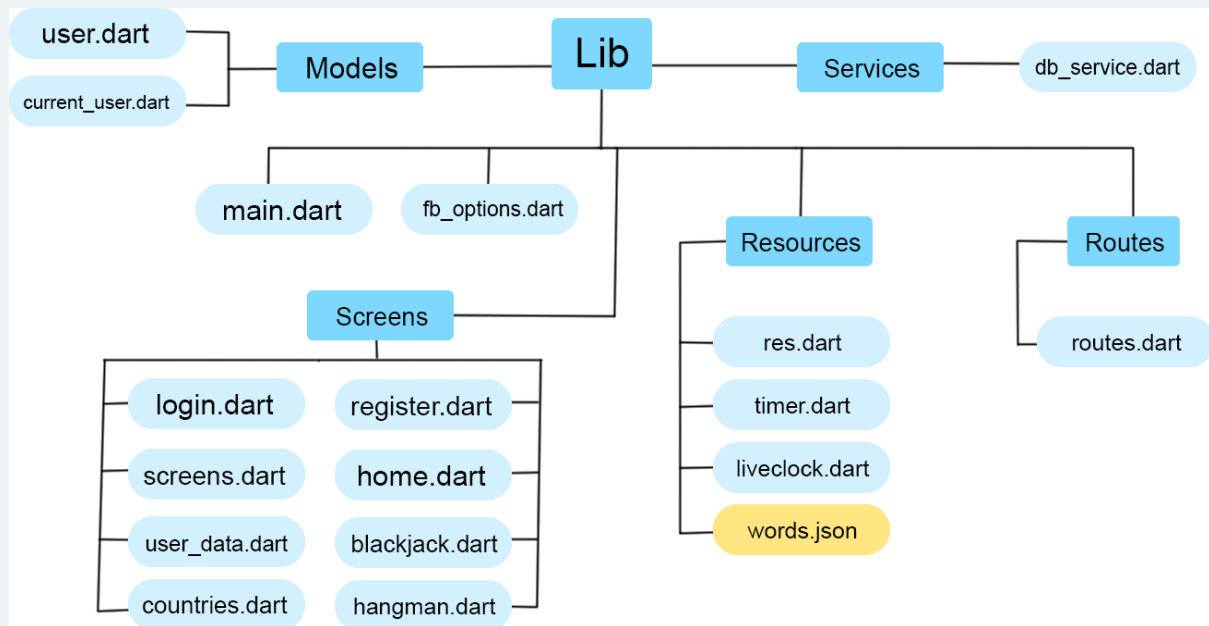
En cuanto al lenguaje de programación, hemos utilizado Dart junto con el framework Flutter para la programación de la aplicación. Estamos usando una conexión a una base de datos en Firebase donde vamos a guardar todos los datos del usuario.

El diseño artístico de la aplicación, se ha creado a partir de la siguiente paleta de colores y de las siguientes pantallas:



- Imágenes del diseño inicial de la aplicación, y la paleta de colores

Y el diseño de la arquitectura está basado en el siguiente esquema:



Login y registro

Para las pantallas de login y registro, tenemos siempre dos botones al lado inferior derecho, para registrarse/iniciar sesión o volver hacia atrás. Tenemos primero, el logo de la aplicación, y luego tenemos un contenedor azul que encasilla todos los campos del formulario.

- En la parte del **login** le pedimos al usuario su nombre de usuario y contraseña
- En la parte del **registro**, le pedimos nuevamente al usuario que introduzca su nombre de usuario, correo electrónico, su contraseña dos veces para asegurarnos, y que seleccione su sexo correspondiente.

Pantalla principal de juegos

En este apartado, daremos un mensaje de bienvenida al usuario según su sexo y mostraremos su nombre. Luego, tendremos un contenedor deslizable con todos los juegos de la aplicación. Debajo del todo tendremos un reloj con el tiempo del sistema, para que el usuario pueda revisar la hora si así lo necesitase.

También tenemos un botón fijo en la parte inferior derecha, que nos llevará a la pantalla de actualización de datos del usuario.

Actualización de datos del usuario

En esta pantalla, nuevamente le recordamos al usuario qué cuenta es la que está gestionando, y debajo le damos campos para ingresar datos nuevos. Le aclaramos, que solo se actualizarán los datos cuyo valor se diferencie de los datos originales.

Juego 1: Blackjack

Utilizamos una interfaz en donde mostramos los dos mazos de cartas, la casa por arriba y el jugador por debajo, y en el medio mostramos un texto indicando el resultado de la partida. El jugador tiene tres botones como opciones, pedir una carta, plantarse, o reiniciar.

Las reglas son las siguientes:

- Tiene prioridad la casa, si ambos llegan a 21, gana la casa.
- Si la casa se pasa de 21, gana el jugador. Si el jugador se pasa, gana la casa.
- Se gana por menor diferencia a 21. Si la casa tiene 17 y el jugador 18, gana el jugador.
- El jugador no puede reiniciar la partida hasta que termine la ronda.

Juego 2: Ahorcado

Para este juego, tenemos la imagen del ahorcado en medio de la pantalla, y debajo tenemos unos guiones, que representan la palabra a adivinar. Justo debajo de ella, tenemos los botones para cada una de las letras, para que el usuario que quiera adivinar una letra, simplemente pulse en ella y el código actualice el ahorcado, o la palabra secreta si es que la letra está dentro de la palabra.

Juego 3: Adivina la bandera

El nombre de este juego lo dice todo, utilizamos una interfaz simple en donde colocamos una bandera en el centro de la pantalla, y le damos 4 opciones al jugador para que escoja la bandera correcta. El jugador también está cronometrado por 6 segundos, que si se acaban, pierde la ronda.

- Este juego utiliza una API que nos trae banderas (*Enlaces de interés*)

Diagrama de clases

En nuestro código, únicamente utilizamos una clase Usuario, que simplemente guarda el correo, nombre de usuario, contraseña y sexo en Firestore Database:

```
email: "usuario@gmail.com"
pwd: "usuario"
sex: "Hombre"
user: "usuario"
```

No disponemos de diagrama E/R ya que estamos utilizando una base de datos documental. Podemos también tomar en cuenta, que al ser documental, podemos añadirle cualquier tipo de variable a una colección, podríamos por ejemplo añadir un récord histórico de puntos en el juego de las banderas, o del ahorcado.

Documentación de la solución

Nuestro proyecto es Open Source, aquí un enlace directo al repositorio de la aplicación:

- <https://github.com/IanIwanczuk/ProyectoFinalFlutter>

Enlaces de interés

Distintos enlaces a tutoriales que pueden ayudar a comprender y estudiar la estructura del proyecto:

- https://www.youtube.com/watch?v=VCWRxy0G4_Y&ab_channel=HussainMustafa
- https://www.youtube.com/watch?v=G0rsszX4E9Q&t=1289s&ab_channel=HussainMustafa
- https://www.youtube.com/watch?v=k7TVYn5jwQk&ab_channel=HussainMustafa

API utilizada para el juego de adivinar la bandera:

- <https://flagpedia.net/download/api>

API utilizada para las imágenes de las cartas de poker:

- <https://deckofcardsapi.com/>