

**Hay que tener en cuenta que este no es el estilo final del documento. En la entrega final, cuando estén todos los apartados completos haré el documento con el estilo requerido.**

### **3. Arquitectura.**

#### **3.1. Topología de Servidores y Apps:**

##### **Diagrama de Arquitectura:**

La aplicación móvil (desarrollada en Android Studio) estará conectada a Firebase para la autenticación de usuarios, la base de datos en tiempo real y la gestión de datos. La comunicación con Firebase se realizará mediante la SDK de Firebase, que ofrece soporte para operaciones CRUD sobre la base de datos y autenticación.

- **Aplicación Móvil Android con Firebase Authentication** para autenticación de usuarios.
- **Aplicación Móvil Android con Firebase Firestore** para almacenar y recuperar datos de las builds y el estado de los planetas.

##### **Descripción de la Topología:**

El flujo de datos será el siguiente:

##### **1. Consulta de los planetas:**

- La aplicación móvil realiza una solicitud a la API de Helldivers para obtener los estados de los planetas. La respuesta es procesada y presentada al usuario.

##### **2. Subida de Builds:**

- Cuando un usuario crea o sube una build, la aplicación realiza una operación en Firebase Firestore/Realtime Database para guardar los datos de la build (como las armas, armaduras, estrategias, etc.).

### 3. Sistema de Likes/Dislikes:

- Los usuarios pueden interactuar con las builds de otros usuarios, registrando un "like" o "dislike". Esto se realiza mediante la modificación de un campo en el documento correspondiente a la build en Firestore/Realtime Database.

### Especificación de Tecnologías:

- **Aplicación Móvil:**
  - **Android Studio 4.2, Kotlin 1.5, Firebase SDK 28.0.**
- **Base de Datos:**
  - **Firebase Firestore** (para almacenar las builds, datos de usuarios, etc.).
- **Autenticación:**
  - **Firebase Authentication** para la autenticación de usuarios.

### Justificación de las Tecnologías:

- **Firebase:** Elegido por su escalabilidad, facilidad de integración con Android y capacidad de manejo de datos en tiempo real.
- **Android Studio:** La opción natural para desarrollar aplicaciones Android, ya que es la plataforma oficial para crear apps nativas.
- **Firebase Authentication:** Proporciona un sistema robusto y fácil de integrar para gestionar la autenticación de usuarios de manera segura.

### Consideraciones de Seguridad:

- **Firebase** asegura la transmisión de datos mediante HTTPS.
- **Reglas de seguridad de Firebase** se configurarán para garantizar que los usuarios solo puedan acceder a los datos relevantes para ellos.

- **Validación en el lado del cliente y servidor** para evitar la manipulación de datos.

### **Consideraciones de Escalabilidad:**

- **Firebase** se escala automáticamente según el tráfico, lo que garantiza un rendimiento óptimo incluso con un aumento en el número de usuarios.
- La base de datos está diseñada para soportar consultas eficientes a gran escala, aprovechando las capacidades de Firestore/Realtime Database.

## **3.2. Descripción de las Apps.**

### **Descripción Detallada de la Aplicación Móvil (Android):**

#### **Funcionalidades Principales:**

- **Pantalla Principal:**  
La pantalla principal mostrará el estado de los planetas, con un **botón central** que servirá para consultar los estados de los planetas a través de la API de Helldivers 2.
- **Botón Izquierdo - Builds de Jugadores:**  
Este botón llevará a la vista que muestra las **builds** subidas por los jugadores. Los usuarios podrán ver, dar "like" o "dislike" a las builds de otros usuarios. Además, en la esquina inferior de esta pantalla, habrá un **botón flotante (+)** que permitirá a los usuarios subir sus propias builds.
- **Botón Derecho - Major Orders:**  
Este botón permitirá ver las **Major Orders**, proporcionando una lista o vista de las órdenes principales que están disponibles.

### **Interfaz de Usuario (UI):**

#### **Barra de Navegación:**

La barra de navegación consta de **cuatro botones con íconos**, distribuidos de forma horizontal en la parte inferior de la pantalla:

- **Primer botón (izquierda):** Estado de los planetas.

- **Segundo botón:** Builds de jugadores.
- **Tercer botón:** Crear nueva build.
- **Cuarto botón (derecha):** Major Orders.

Cada botón estará acompañado de un ícono representativo e intuitivo, y la interfaz se desarrollará siguiendo las guías de **Material Design** para asegurar una experiencia de usuario coherente, moderna y accesible.

#### **Pantalla de Builds:**

La pantalla de builds mostrará las builds subidas por los jugadores en un diseño de lista. Cada ítem mostrará información básica de la build (nombre, autor, puntuación, etc.). La opción para **crear una nueva build** estará accesible desde el **tercer botón de la barra de navegación**, lo que ofrecerá un flujo de navegación más claro y estructurado que el clásico botón flotante "+".

#### **Experiencia de Usuario (UX):**

- La aplicación será intuitiva, con botones de navegación accesibles y fáciles de entender.
- Las animaciones de transición serán suaves para mejorar la experiencia.
- Al subir o ver una build, se podrá ver un listado con la opción de dar "like" o "dislike" a otras builds, haciendo que la interacción sea dinámica.

#### **Interacción con la API:**

- La aplicación se conectará a la **API de Helldivers 2** para consultar los estados de los planetas. Además, se comunicará con Firebase para gestionar la autenticación de usuarios y para interactuar con las builds subidas por los usuarios (subir nuevas builds, dar like/dislike).
- Retrofit se utilizará para hacer las solicitudes HTTP a la API de Helldivers y Firebase para el manejo de la base de datos y autenticación.

#### **Gestión de Sesiones:**

- Los usuarios podrán iniciar sesión mediante **Firestore Authentication**. Además, se implementará la opción de cerrar sesión y de recordar al usuario.

### 3.3. Diagrama de despliegue.

En este proyecto, el sistema está compuesto por una aplicación móvil Android que interactúa con varios servicios en la nube, incluyendo Firebase y una API externa (Helldivers).

La **aplicación móvil Android** se despliega en los dispositivos móviles de los usuarios en formato APK o AAB. Esta aplicación utiliza la SDK de Firebase para interactuar con los servicios en la nube, como **Firestore** para almacenar y recuperar datos de los usuarios (por ejemplo, las builds subidas por los jugadores) y **Firebase Authentication** para gestionar la autenticación de los usuarios. La comunicación con **Firestore** es en tiempo real, lo que garantiza que las actualizaciones, como la subida de nuevas builds, se sincronicen de inmediato entre los dispositivos. La autenticación también se realiza de forma segura a través de Firebase, permitiendo que los usuarios se registren, inicien sesión y mantengan su sesión activa de manera eficiente.

Adicionalmente, la **aplicación móvil** realiza peticiones HTTP a la **API de Helldivers** para obtener información actualizada sobre los planetas y sus características. Esta API RESTful devuelve respuestas en formato JSON, que luego son procesadas y utilizadas dentro de la aplicación móvil. La API de Helldivers proporciona datos dinámicos como el estado de los planetas, que pueden variar con el tiempo, lo que justifica el uso de esta fuente externa de datos.

La **API de Helldivers** se encuentra en un servidor remoto y está diseñada para responder a las solicitudes HTTP con información en formato JSON. La aplicación móvil realiza solicitudes GET para obtener los datos de los planetas, que se actualizan regularmente según la información disponible en la API externa. Esto permite que los usuarios siempre tengan acceso a datos precisos y actualizados sobre los planetas y sus estados.

El **servicio de Firebase** maneja la infraestructura de la base de datos en tiempo real (Firestore) y la autenticación de los usuarios (Firebase Authentication). Al elegir Firebase, se aprovecha su escalabilidad y sus funcionalidades en tiempo real para asegurar que los datos de los usuarios y las actualizaciones de las builds se sincronicen automáticamente entre todos los dispositivos. Además, Firebase es una

solución de fácil implementación que permite evitar la necesidad de gestionar servidores adicionales, lo que reduce la complejidad del sistema.

En cuanto a la **seguridad**, Firebase proporciona reglas de seguridad para Firestore, lo que asegura que los datos de los usuarios estén protegidos y que cada usuario solo tenga acceso a su propia información, como sus builds. Además, **Firestore Authentication** garantiza que solo los usuarios autenticados puedan interactuar con los servicios de la aplicación, protegiendo así la privacidad y la integridad de los datos.

En resumen, la arquitectura de este sistema se basa en una aplicación móvil que se conecta tanto a Firebase para la gestión de usuarios y almacenamiento en tiempo real como a la API de Helldivers para obtener información externa sobre los planetas. Esta integración permite una experiencia de usuario fluida y dinámica, con datos actualizados y sincronizados automáticamente.

## 4. Diseño.

### 4.1. Modelo de datos. Firebase Store.

Se han definido tres colecciones principales para organizar los datos de forma flexible y escalable:

#### Colección: **usuarios**

Contiene un documento por cada usuario registrado.

```
{
  "userID": "usuario123",
  "email": "usuario123@gmail.com",
  "profilePic": "https://.../profile.jpg",
  "signUpDate": "2025-04-27T14:20:00Z"
}
```

#### Colección: **builds**

Contiene builds personalizadas subidas por los usuarios, utilizando datos que pueden consultarse de la API de Helldivers.

```
{
  "userID": "usuario123",
```

```
"armor": "SC-34 Infiltrator",
"booster": "Hellpod Space Optimization",
"enemyType": "Terminids",
"primaryWeapon": "Railgun",
"secondaryWeapon": "Ultimatum",
"stratagems": [
  "M-105 Stalwart",
  "Orbital Airburst Strike",
  "Eagle Strafing Run",
  "LIFT-850 Jump Pack"
],
"like": 0,
"dislike": 0,
"time": "2025-04-30T15:55:05Z"
}
```

## Colección: **votos**

Almacena los votos individuales de cada usuario para evitar votos múltiples por build.

```
{
  "userID": "",
  "buildID": "build789xyz",
  "type": "like"
}
```

## Relaciones y Referencias

- ❖ Cada documento en **builds** guarda el ID del usuario que la creó (**userID**), referenciando a un documento en **usuarios**.
- ❖ Cada documento en **votos** guarda:
  - **userID**: el usuario que ha votado.
  - **buildID**: la build que ha sido votada.
  - **type**: tipo de voto (**like** o **dislike**).

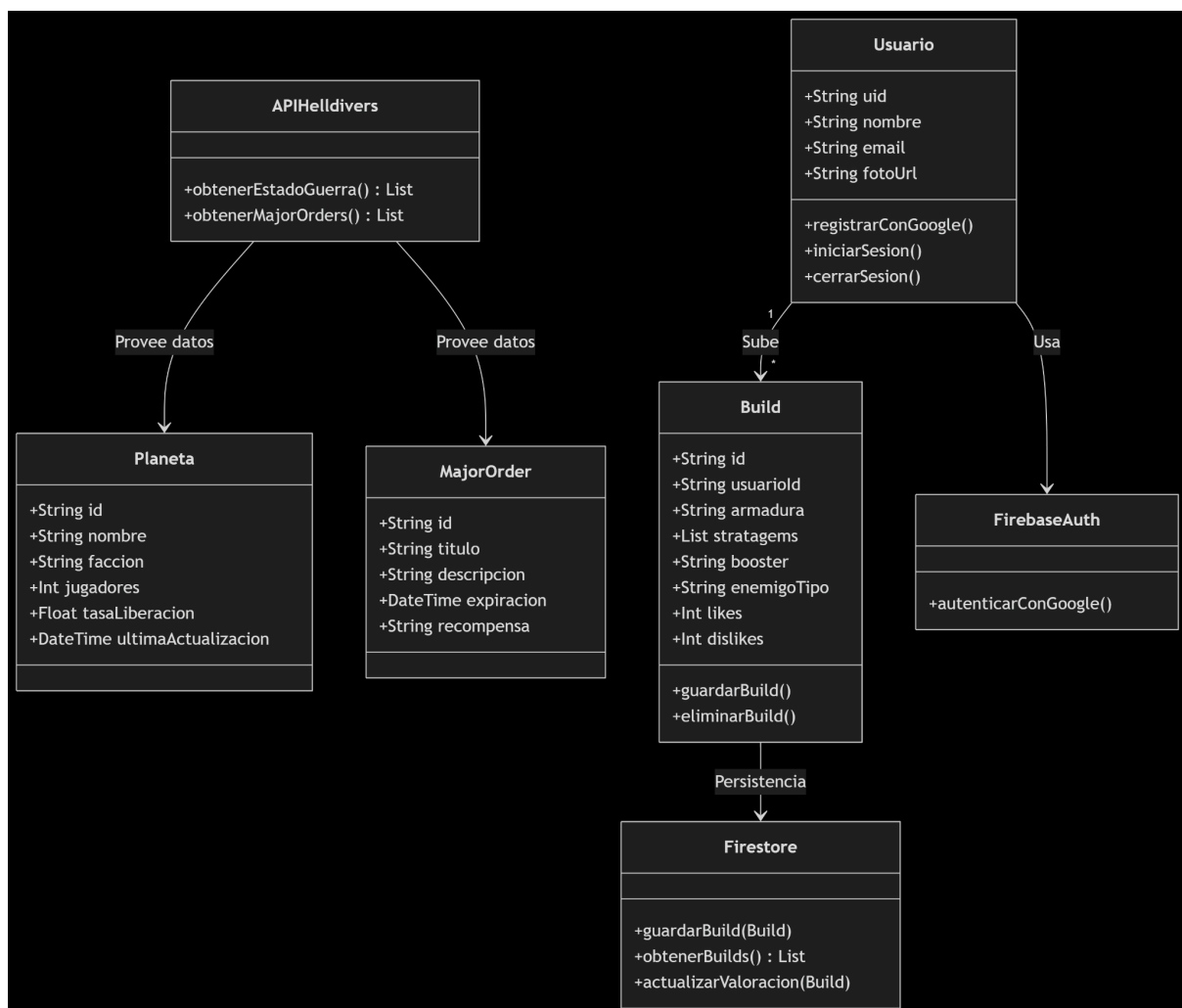
## Justificación del uso de Firebase Firestore:

Se ha elegido Firebase Firestore como base de datos para esta aplicación por su estructura flexible, ideal para trabajar con colecciones y documentos como *builds* o usuarios, sin necesidad de esquemas rígidos como en MySQL.

Firestore permite trabajar en tiempo real, mostrando cambios como votos o nuevas builds al instante en la app.

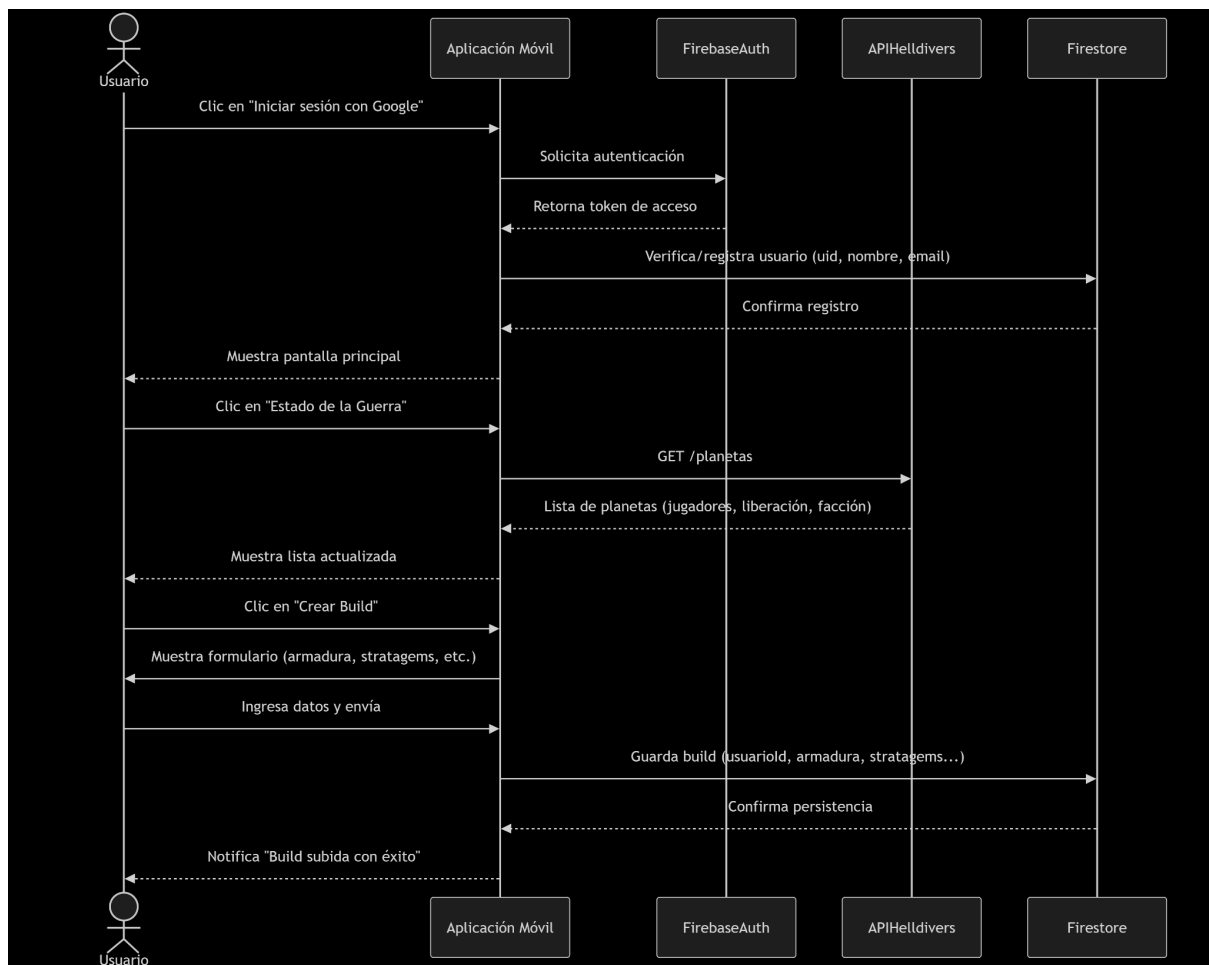
Además, se integra fácilmente con Firebase Authentication, lo que simplifica el manejo de usuarios y sus acciones dentro del sistema (por ejemplo, quién sube o vota una build). También es escalable y gestionado en la nube, lo que evita preocuparse por mantenimiento o servidores. Por estas razones, es una solución adecuada para una app social y colaborativa como esta.

## 4.2. Diagrama de clases.

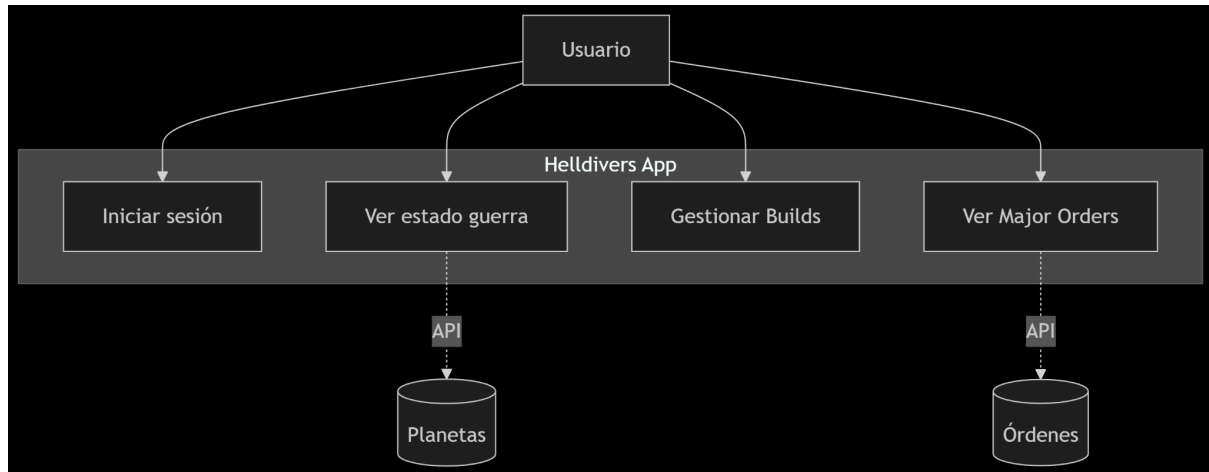




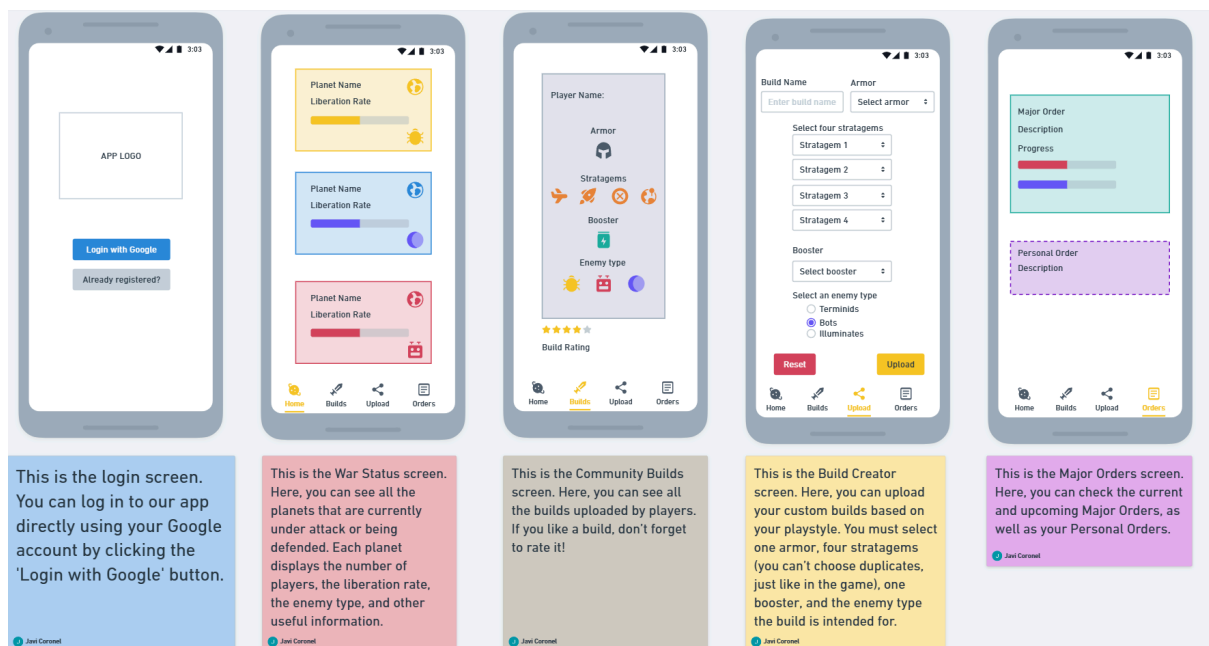
### 4.3. Diagrama de secuencia.



## 4.4. Diagrama de casos de uso.



## 4.5. Mockup.



Para más claridad visitar el enlace:

<https://whimsical.com/helldivers-buildhub-RYNohu5EvfwqDBurAHWgeB>

## 4.6. Guía de estilos.

### Guía de Estilos

La guía de estilos define los elementos visuales clave de la aplicación, garantizando una experiencia coherente y reconocible para el usuario, en sintonía con la estética del videojuego *Helldivers*.

### Paleta de Colores

Se han seleccionado colores que evocan el estilo visual de *Helldivers* y que, a la vez, garantizan buena legibilidad en dispositivos móviles.

- **Color primario:** Amarillo (#FFD700) — utilizado en botones y navegación, inspirado en el color corporativo de Helldivers.
- **Color secundario:** Negro (#000000) — usado para textos principales y fondo en elementos contrastados.
- **Colores según tipo de enemigo:**
  - Bichos (Terminids): Amarillo (#FFD700)
  - Iluminados: Azul (#3498DB)
  - Bots: Rojo (#E74C3C)
- **Fondo general:** Color neutro (ej. #F5F5F5 o #1E1E1E), adaptado a la visibilidad y estética general.

### Tipografía

Se utiliza una fuente legible, moderna y adecuada para pantallas móviles.

- **Fuente principal:** *Titillium Web*, una fuente sans-serif de estilo militar, fácil de leer y en línea con el diseño del juego.
- **Tamaños típicos:**
  - Títulos: 24sp (negrita)
  - Texto principal: 16sp (regular)
  - Texto secundario: 14sp (ligero)

## Iconografía

- Se emplean iconos de estilo Material Design para las funciones generales.
- Los iconos específicos del universo *Helldivers* (estratagemas, enemigos, armaduras) se importarán como SVG/PNG personalizados.
- Tamaño estándar: 24x24 dp
- Color:
  - Activo: Amarillo (#FFD700)
  - Inactivo: Gris oscuro (#666666) o transparente.

## Elementos de Interfaz

- **Barra de navegación inferior:** Cuatro botones con iconos representativos sin fondo. Al estar seleccionados, se rellenan de amarillo.
- **Botones:** Silueta o fondo amarillo con esquinas redondeadas y sombra suave.
- **RecyclerView (planetas y builds):** Estilo tipo tarjeta, con bordes definidos, colores de tipo de enemigo y diseño responsive.
- **Campos de texto:** Bordes finos, fondo claro, espaciado interno de 8dp.
- **Botón flotante (+):** Circular, amarillo, ubicado en la esquina inferior derecha para añadir builds.

## Aplicación de la Guía en Pantallas

- **Pantalla de planetas:** Uso del RecyclerView con tarjetas diferenciadas por color según enemigo, con texto legible y disposición clara.

- **Pantalla de builds comunitarias:** Lista con estilo uniforme, tipografía clara, iconografía personalizada y botones visibles para interactuar.

### **Justificación del Diseño**

- El color amarillo se asocia a la identidad visual de *Helldivers*, y se emplea para reforzar la ambientación.
- Se ha optado por una fuente moderna y de fácil lectura que recuerde a interfaces militares.
- La iconografía personalizada mejora la inmersión y claridad visual.
- El diseño general busca un equilibrio entre estética temática y funcionalidad, cuidando la accesibilidad visual con contrastes adecuados.

## **4.7. Justificación del diseño.**

Las decisiones de diseño tomadas en el desarrollo de esta aplicación móvil se han basado en principios de usabilidad, adaptabilidad y en las necesidades específicas del público objetivo: jugadores de *Helldivers*. Se ha procurado garantizar una experiencia de usuario accesible, coherente y cómoda en distintos dispositivos e idiomas.

### **Adaptabilidad a distintos dispositivos**

El diseño de la interfaz se ha concebido para que se adapte correctamente tanto a smartphones como a tablets. Se han empleado componentes de diseño responsivo, como [ConstraintLayout](#) y [RecyclerView](#), que permiten distribuir los elementos de forma flexible, manteniendo su proporción y legibilidad en pantallas de diferentes tamaños y resoluciones. Esto asegura que tanto las listas de builds como la visualización de los planetas se presenten de forma clara y ordenada, independientemente del dispositivo.

### **Soporte multilingüe**

Para facilitar el acceso a jugadores de distintas regiones, se ha implementado soporte multilingüe. Toda la interfaz ha sido internacionalizada utilizando archivos [strings.xml](#) separados para español e inglés, lo que permite que la aplicación se adapte automáticamente al idioma configurado en el dispositivo del usuario. Además, se han evitado textos incrustados directamente en los layouts, siguiendo buenas prácticas de desarrollo accesible y multilingüe.

## **Disposición de elementos y navegación**

La ubicación de los elementos de la interfaz se ha definido priorizando la comodidad y la facilidad de uso. La barra de navegación inferior agrupa los accesos más frecuentes (planetas, builds, crear build y major orders) en una posición fácilmente accesible con el pulgar, especialmente útil en móviles de pantalla grande. Esta estructura de navegación ayuda al usuario a comprender rápidamente la organización de la aplicación y acceder de forma ágil a las funciones más utilizadas.

## **Retroalimentación visual al usuario**

La aplicación proporciona retroalimentación inmediata en las situaciones necesarias. Por ejemplo, si un usuario intenta subir una build con datos incorrectos, los campos erróneos se resaltan en rojo. Esto permite identificar fácilmente el problema y corregirlo sin ambigüedad. También se utilizan mensajes de confirmación y validación contextual para guiar al usuario en acciones críticas.

## **Principios de diseño aplicados**

Las decisiones de diseño se han inspirado en los principios de diseño visual y usabilidad aprendidos durante el curso, así como en la experiencia personal analizando otras aplicaciones Android. Se ha aplicado el principio de **consistencia**, utilizando los mismos estilos visuales (botones, tipografías, colores) en todas las pantallas. También se ha seguido un diseño visual basado en Material Design para garantizar una experiencia coherente, moderna y accesible.

## **Prevención de errores**

Se han tomado medidas para minimizar errores del usuario. Por ejemplo, en el proceso de creación de una build, las opciones como estrategias o tipos de enemigo se seleccionan mediante [Spinner](#), evitando la introducción libre

de texto. De esta forma se garantiza que los datos introducidos siempre sean válidos y que el usuario no cometa errores al escribir.

### **Atención al usuario objetivo**

La aplicación está diseñada específicamente para la comunidad de jugadores de *Helldivers*, un público que valora la claridad, rapidez y personalización. Por ello, se ha implementado una interfaz que permite subir builds personalizadas, explorar builds de otros jugadores, y consultar el estado de los planetas y las *Major Orders* con facilidad. Asimismo, el soporte multilingüe (inglés y español) asegura que la herramienta pueda ser usada por una mayor parte de la comunidad internacional.