

# Ángel Barceló – Documentación Proyecto KillerGame

## 4 - DESCRIPCIÓN DETALLADA DEL SUBPROYECTO INDIVIDUAL

### 4.2 - ROL ASIGNADO

Desarrollar la parte de la aplicación móvil correspondiente al control del juego. Añadiendo las funciones necesarias para poder mover el cursor para configurar y jugar la partida.

#### 4.2.2 – RESPONSABILIDADES ADICIONALES

Llevar un seguimiento del proyecto y elaborar la documentación del mismo.

## 5 - PLANIFICACIÓN DE LAS TAREAS DEL SUBPROYECTO INDIVIDUAL

Tarea	Duración	Fecha Inicio	Fecha Fin
Repartición de tareas del equipo y preparación del GitHub	13 Días	03/03/2023	15/03/2023
Diseño general de la aplicación	20 Días	16/03/2023	03/04/2023
Diseño de cada actividad de la aplicación	20 Días	04/04/2023	24/04/2023
Implementación del Código	37 Días	25/04/2023	31/05/2023
Pruebas y correcciones	7 Días	01/06/2023	07/06/2023
Entrega Final		12/06/2023	

## 6 - DETALLE DE LAS DEDICACIONES Y TAREAS REALIZADAS

### 6.1 – PREPARACIÓN DEL REPOSITORIO GITHUB

La parte de la aplicación de móvil es desarrollada en Android Studio, y subida a un repositorio de GitHub diferente a la parte de PC del proyecto, ya que la estructura de los proyectos es diferente, debido al entorno de desarrollo.

### 6.2 – ASIGNACIÓN DE GRUPOS E INTEGRANTES

El equipo de la aplicación de móvil se compone de tres integrantes:

Lisbeth - Coordinación y diseño de pantallas de conexión
Angel - Diseño de pantallas de control del juego
Karina - Implementación de la lógica de comunicaciones.

### 6.3 – DEFINICIÓN DE REQUISITOS FUNCIONALES

El mando debe tener cuatro botones, dos de dirección y dos de acción.

Dichos botones deben enviar paquetes de datos al cluster, para que éste pueda tomar acciones en respuesta al input del usuario.

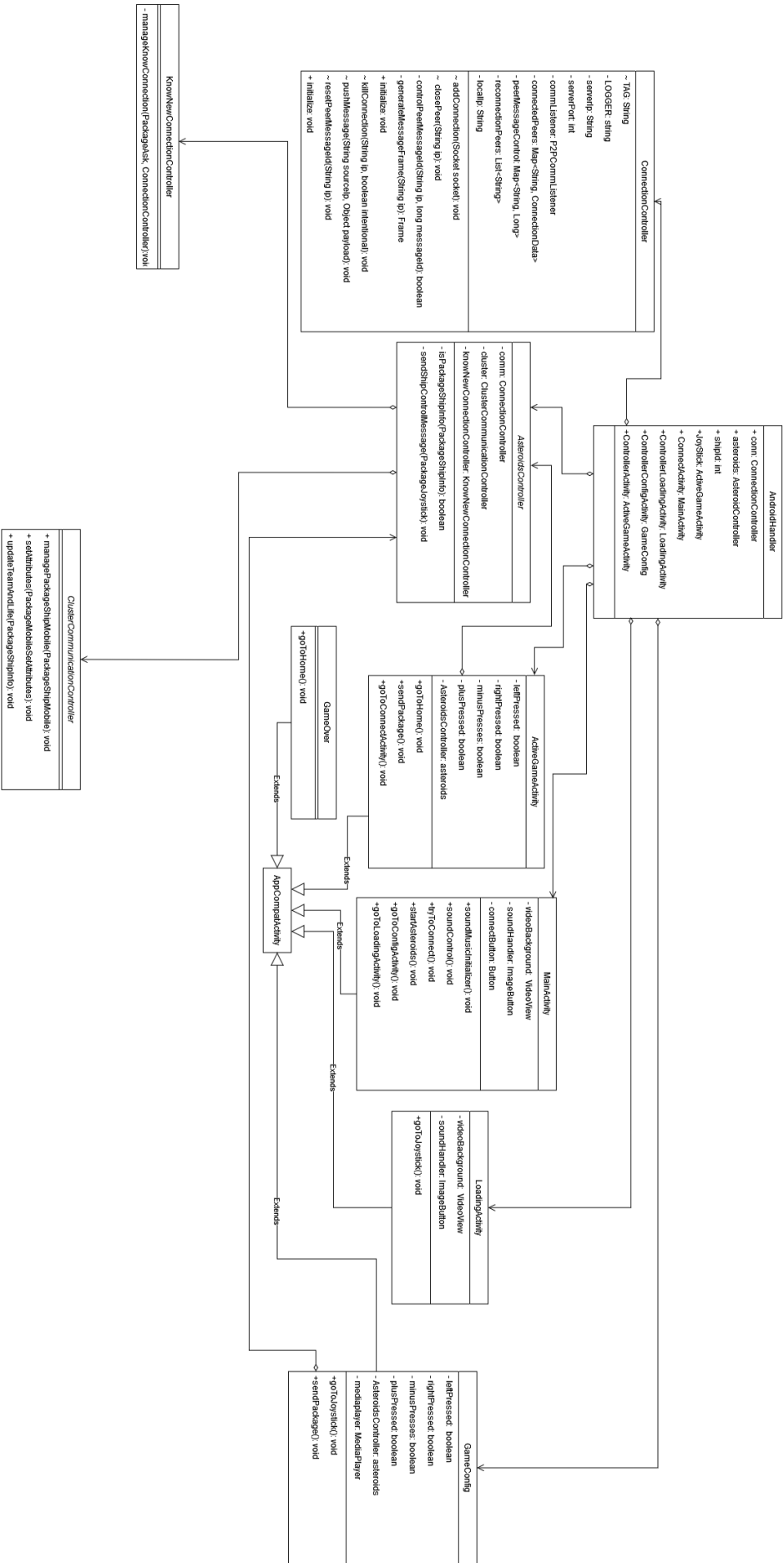
## 8 - DESCRIPCIÓN DE LA ARQUITECTURA DEL SUBPROYECTO

El Subproyecto del movil consiste de varias clases, una por cada pantalla, más otras clases necesarias para implementar las funcionalidades de conexión y comunicaciones.

Cada jugador usará la aplicación movil para conectarse al cluster de ordenadores, el primero en hacerlo es designado como máster, y se encargará de configurar las reglas de la partida, mientras los demás jugadores esperan a que termine la configuración. A continuación, todos los jugadores que estén conectados y listos pasarán a la pantalla de juego, y comienza la partida.

Cuando la nave de un jugador es destruida, éste queda a la espera, y cuando la partida termina, todos los jugadores vuelven al principio, donde se vuelve a designar un master para configurar la siguiente partida.

En la siguiente imagen se puede ver el UML de la aplicación:

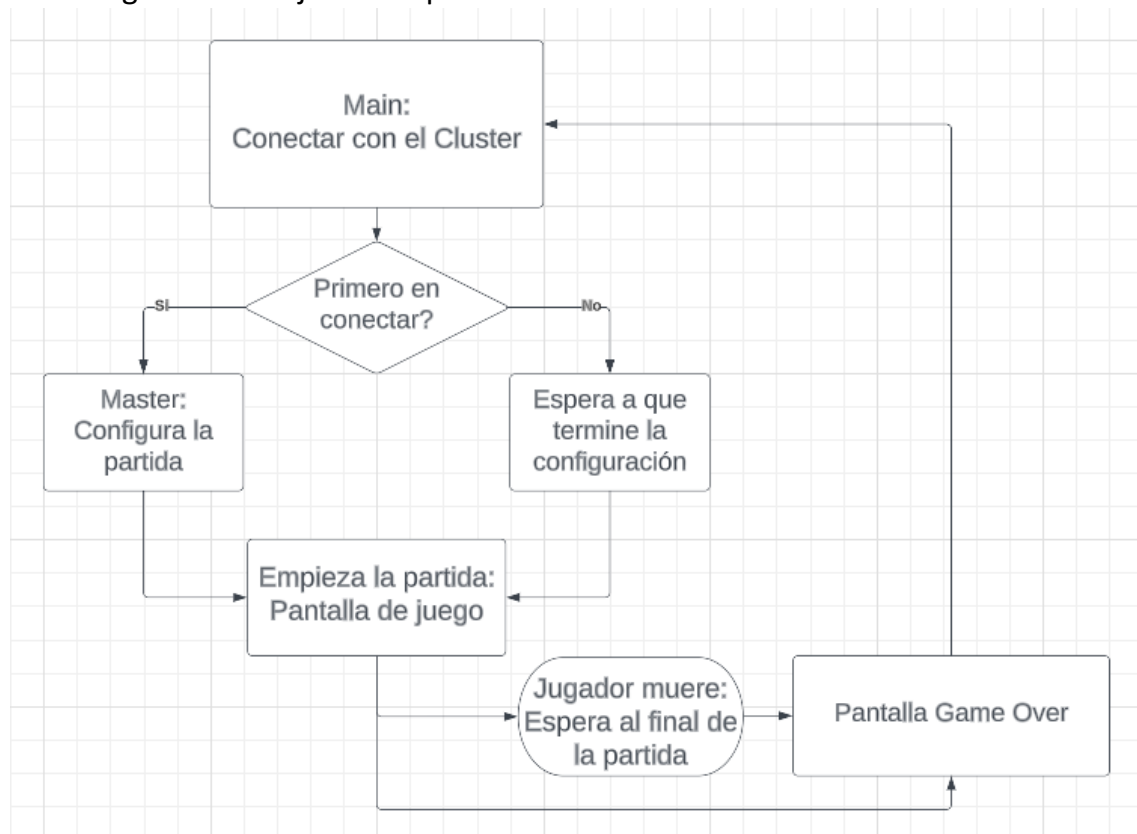


Para una mejor visualización del UML se puede abrir desde Google Drive, o directamente desde Drawio, a través de los siguientes enlaces:

<https://drive.google.com/file/d/1P2BnkUiUjbFCVaV8P4hIGKSg8E-PLQ8/view?usp=sharing>

<https://app.diagrams.net/#G1P2BnkUiUjbFCVaV8P4hIGKSg8E-PLQ8>

Y un diagrama de flujo de la aplicación:



## 9 - DESCRIPCIÓN DE LOS MÓDULOS

Para la parte de control del juego se han desarrollado 3 clases:

- GameConfigActivity:
  - Contiene los controles para mover un cursor que se muestra en el cluster, para seleccionar las opciones y configurar la partida.
  - Principalmente contiene cuatro botones, dos de dirección (Siguiete y Anterior) para navegar entre las opciones, otros dos, "más" o "aceptar" y "menos" o "cancelar".
  - Cada botón reproduce un sonido al ser pulsado, y envía un paquete al cluster indicando qué boton se ha pulsado, y luego el cluster reacciona en consecuencia.

- Cuando el jugador pulsa el botón “más” sobre la opción de terminar la configuración, la aplicación de todos los jugadores avanza a la siguiente pantalla.
- **ActiveGameActivity:**
  - Contiene una barra de vida que indica al jugador el estado de su nave, y cuatro botones, dos de dirección, para controlar la rotación de la nave, en sentido anti-horario y horario, en ese orden de izquierda a derecha, y otros dos botones, uno para disparar y otro para acelerar.
  - Los botones tienen el mismo aspecto que los de la pantalla de configuración, pero funcionan de forma diferente; Cada botón tiene un atributo asignado, que es un boolean, inicializado en False por defecto.
  - Cuando se *pulsa* un botón, su atributo cambia a True y se envía un paquete al cluster que contiene el estado actual de todos los atributos.
  - De forma similar, cuando se *deja de pulsar* un botón, su atributo vuelve a False, y también se envía un paquete al cluster con el estado actual de cada atributo.
  - De esta forma se consigue que una nave pueda acelerar o rotar durante una cantidad de tiempo indefinida, sin tener que enviar paquetes de datos continuamente al cluster para comprobar si un botón se está manteniendo pulsado o no.
- **GameOverActivity:**
  - Contiene una simple pantalla de Game Over, su función es principalmente hacer que los jugadores eliminados esperen al final de la partida, y luego redirigir a todos los jugadores de vuelta al inicio, donde pueden configurar y empezar una nueva partida.

# 11 – CONCLUSIÓN

A continuación se analizarán los objetivos logrados en el desarrollo de la aplicación, comparándolos con las expectativas iniciales y sugiriendo posibles mejoras.

## 11.1.1.2 – EXPECTATIVAS Y MEJORAS

En términos generales, el proyecto no ha cumplido con las expectativas, debido a problemas y desacuerdos en la definición inicial de los requisitos.

Aunque el proyecto es funcional en su estado actual, se encuentra mas bien en una fase de demostración y haría falta terminar de implementar los requisitos que faltan.

## 11.1.1.3 - LECCIONES APRENDIDAS

El proyecto ha sido una buena experiencia que nos ha enseñado las dificultades que puede tener un proyecto real, especialmente si es algo complejo y se deben repartir las tareas entre varios integrantes de un equipo, y la importancia de una buena planificación.