


29 DE NOVIEMBRE DE 2022

## PROYECTO PROGRAMADO 4

LUIKI-KART

JEAN HUNT

LENGUAJES DE PROGRAMACIÓN  
Ingeniería en computación



## Tabla de contenido

Manual de usuario .....	1
Compilación y ejecución .....	2
Tutorial .....	3
1. Inicio.....	3
2. Menú principal .....	3
3. Crear partida .....	4
4. Unirse a partida .....	5
5. Lobby.....	5
6. Tablero de juego .....	6
7. Partida finalizada.....	7
8. Ranking .....	7
Documentación general.....	9
1. Descripción del problema.....	10
2. Diseño del programa: .....	10
3. Librerías utilizadas .....	11
4. Análisis de resultados .....	12

Ilustración 1:Home .....	3
Ilustración 2: Menú principal.....	4
Ilustración 3: Crear partida .....	4
Ilustración 4: Partidas disponibles .....	5
Ilustración 5: Lobby .....	6
Ilustración 6:Tablero .....	6
Ilustración 7: Partida finalizada .....	7
Ilustración 8:Ranking.....	8
Ilustración 9:Diagrama de paquetes.....	10
Ilustración 10:Diagrama de despliegue.....	11

## *Manual de usuario*

## Compilación y ejecución

Este programa fue desarrollado haciendo uso del framework nodejs + reactjs+socket.io por lo que deberá contar nodejs en su equipo para poder **hostear** este programa, en caso de que solo quisiera conectarse basta con tener acceso a un navegador web.

Para instalar las herramientas anteriormente mencionadas deberá seguir los siguientes links o ingresar los siguientes comandos en su consola de preferencia:

- <https://nodejs.org/es/download/>

Además, durante el desarrollo del proyecto se hizo uso de la herramienta nodemon la cual ayuda a refrescar el servidor cada vez que se realiza un cambio. Para instalar esta herramienta deberá ingresar el siguiente comando:

- Npm install -D nodemon

Una vez completados los pasos anteriores el programa está listo para ser **hosteado**, para ello:

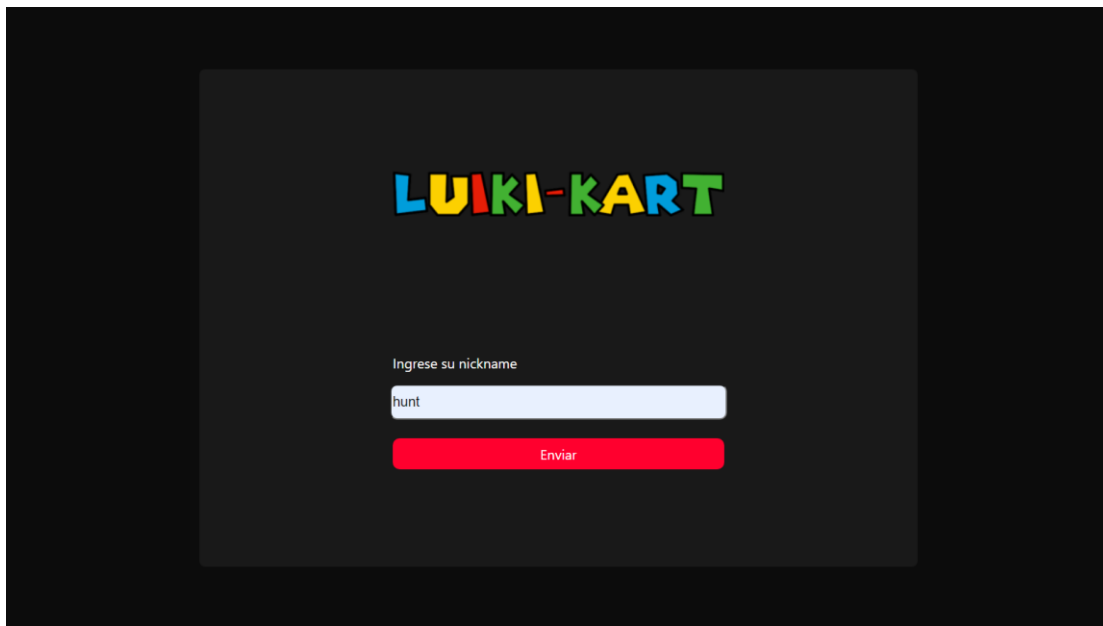
1. Abrir una terminal de comandos, ya sea integrada o de terceros.
2. Dirigirse a la ruta en donde se encuentra el archivo proyecto.
3. Dirigirse a la carpeta servidor
4. Ingresar el comando “npm start”, esto iniciará el servidor back end
5. En otra consola dirigirse a la carpeta client
6. Ingresar el comando “npm start”, esto iniciará el servidor front end

Para ingresar al sitio web deberá solicitar el enlace al host del huego ya que este varía según la maquina en donde se ejecute el programa.

## Tutorial

### 1. Inicio

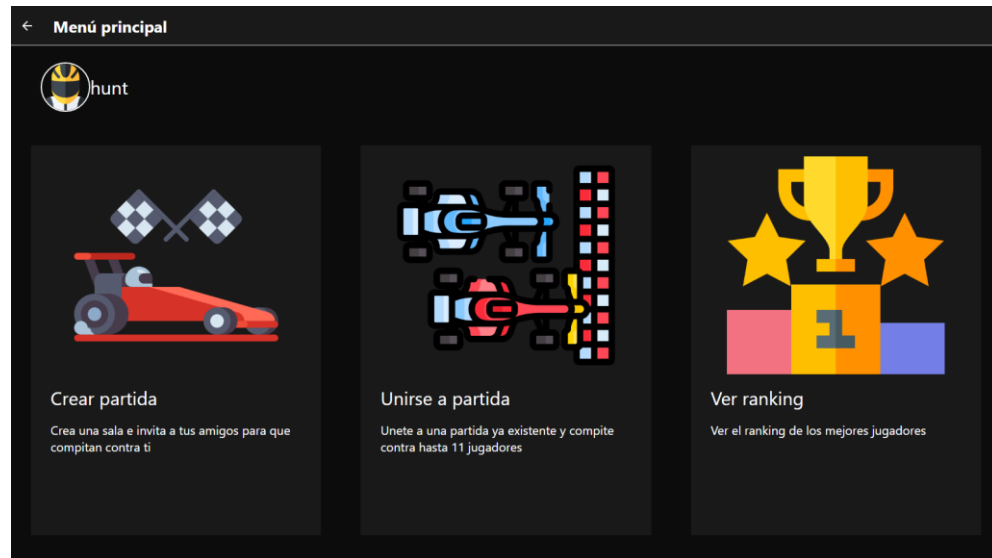
La primera pantalla que se muestra al ejecutar el programa es la pantalla de inicio o “home”, en esta pantalla deberá ingresar su nombre de usuario o nickname para poder acceder al resto de las funcionalidades.



*Ilustración 1: Home*

### 2. Menú principal

Una vez en el menú principal el usuario tendrá 3 opciones, “crear partida”, “unirse a partida” y “ranking”.

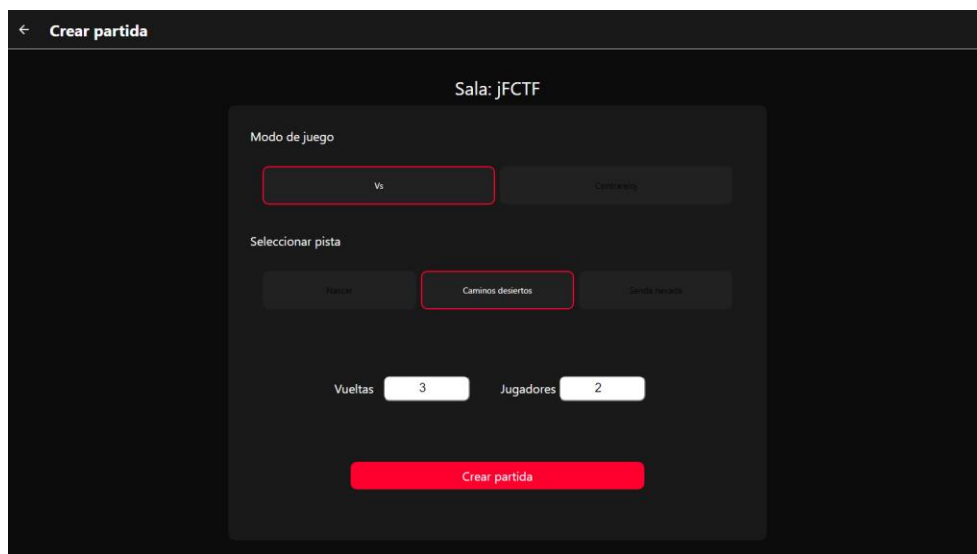


*Ilustración 2: Menú principal*

### 3. Crear partida

Para crear una partida deberá seleccionar la configuración que desee para esta ajustando parámetros como:

- Modo de juego
- Pista
- Cantidad de vueltas
- Cantidad máxima de jugadores (mínimo 2)



*Ilustración 3: Crear partida*

Una vez configurada la partida deberá hacer clic en “crear partida” para ser redirigido al lobby y esperar a los otros jugadores. La partida estará a la espera de sus jugadores durante 3 minutos, si pasado este tiempo no se ha arrancado la partida se cerrará la sala.

#### 4. Unirse a partida

Al seleccionar esta opción se le mostrarán en pantalla una lista de todas las partidas disponibles en este momento, para unirse a una de ellas bastará con hacer click.

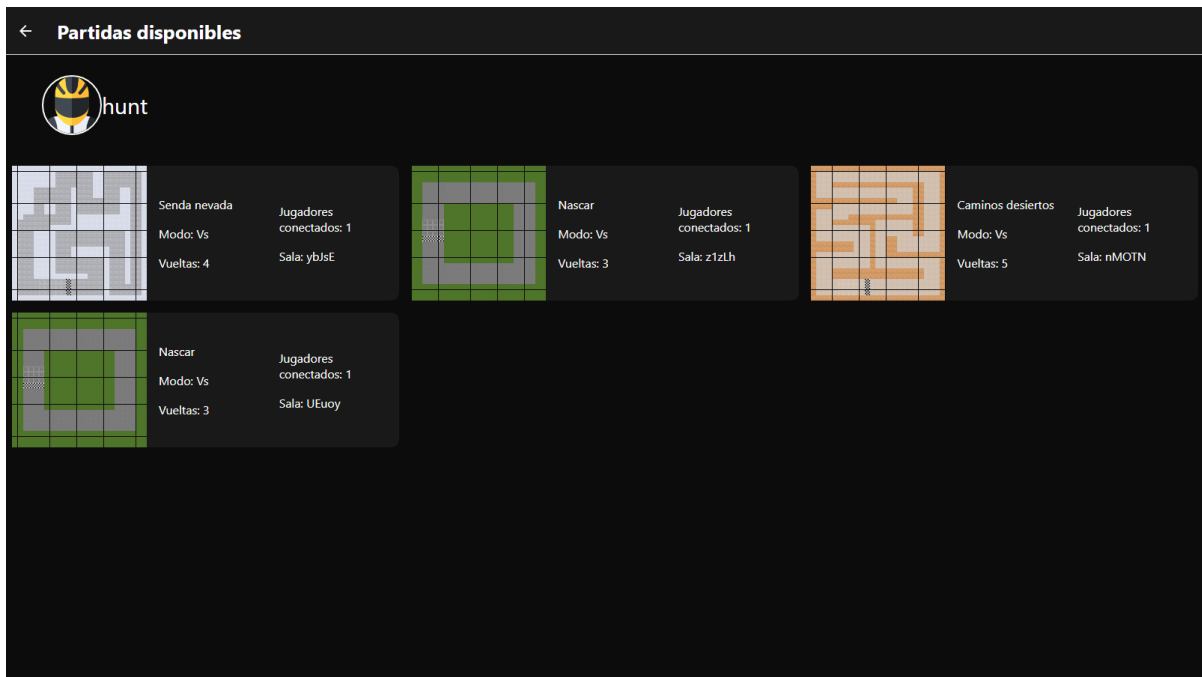


Ilustración 4: Partidas disponibles

#### 5. Lobby

Al unirse o crear una partida se mostrará la lista de usuarios que se han unido a la sala correspondiente, en esta ventana podrán seleccionar un color para su vehículo. Este lobby se cerrará si el creador no inicia la partida en 3 minutos. Únicamente se necesitan 2 jugadores para iniciar la partida.



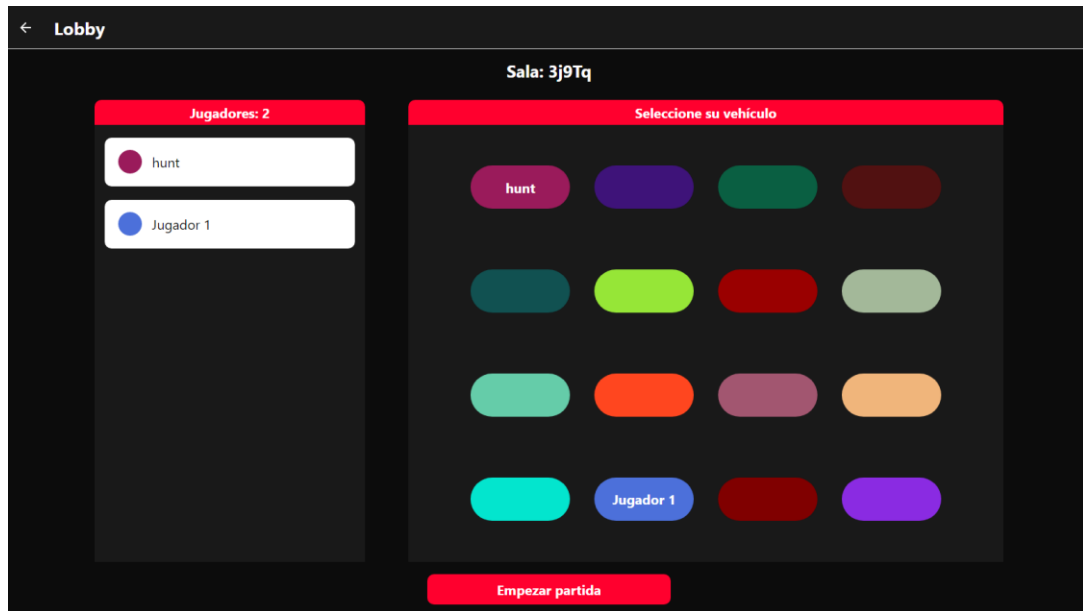


Ilustración 5: Lobby

## 6. Tablero de juego

Una vez el creador de la partida haya iniciado la carrera (presionando la tecla “u”) se desplegará la pista seleccionada y se ubicarán a los jugadores en sus posiciones de salida

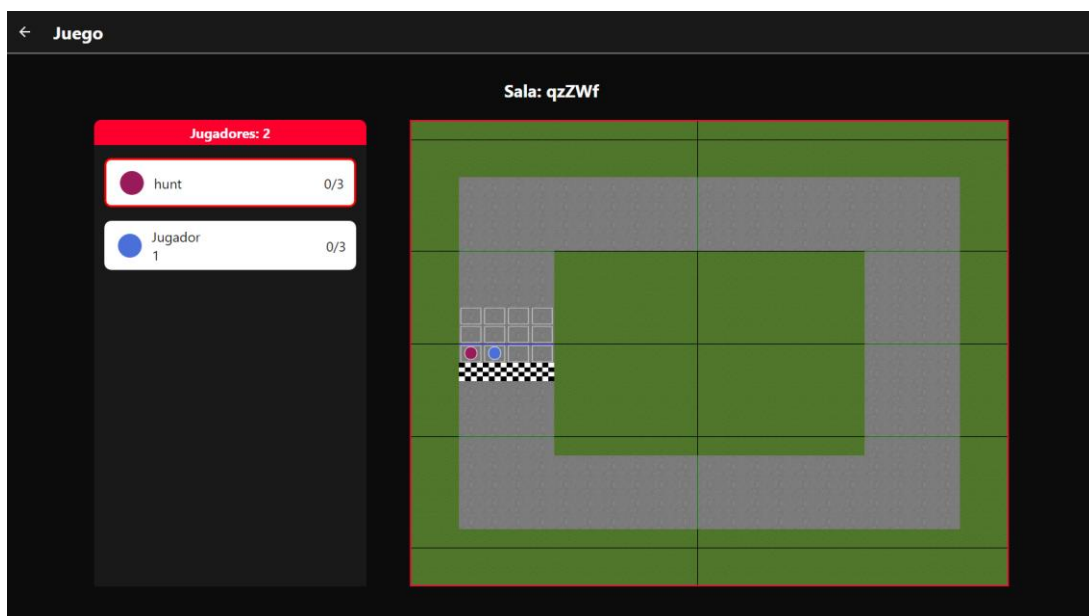


Ilustración 6: Tablero

Para desplazarse sobre la pista deberá utilizar las flechas direccionales, el vehículo se desplazará una celda por cada vez que presione la tecla. La partida finalizará una vez que todos los jugadores hayan completado el circuito.

### 7. Partida finalizada

Al finalizar la partida se desplegará la lista de posiciones en orden ascendente además de la información de la carrera.

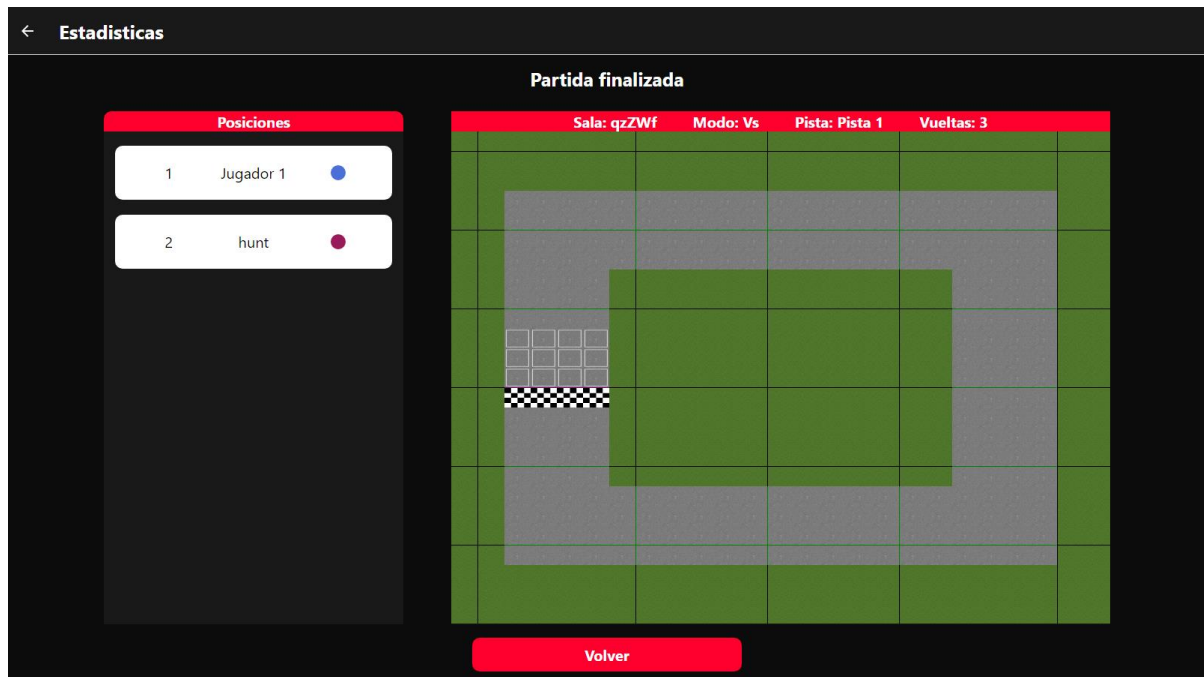


Ilustración 7: Partida finalizada

### 8. Ranking

Al ingresar a esta ventana se desplegará una lista de partidas finalizadas con la información de dicha carrera y su respectivo ganador.

Ranking				
Ganador	Tiempo	Pista	Vueltas	Sala
hunt	----	Nascar	3	LyQe2
jean	----	Nascar	3	fQoAX
hunt	----	Camino desiertos	5	bMqK9
Jugador 1	----	Nascar	3	qzZWf

*Ilustración 8: Ranking*

## *Documentación general*

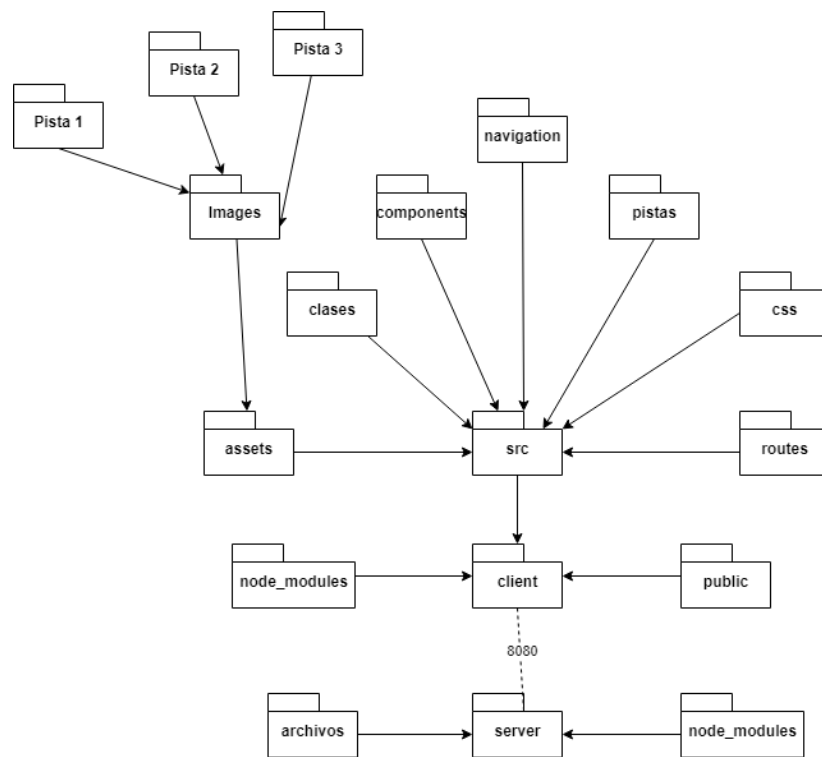
## 1. Descripción del problema

El objetivo general de este proyecto es desarrollar una mejor herramienta Web para el desarrollo de un juego en línea desde la perspectiva de la programación orientada a objetos, la investigación de estrategias de manejo de concurrencia y multiusuarios; de manera que las partes de la lógica y datos (Backend) que lo enmarcan sean diseccionadas minuciosamente para desarrollar una solución web (Frontend). El Backend deberá desarrollarse con .Net Core (C#) o Node js (Express) aplicando orientación a objetos y el Frontend deberá desarrollarse con React, Vue.js, Angular o Net –Asp.Net– (manteniendo la separación de proyectos).

**Nota:** Esta descripción fue tomada de la especificación del proyecto.

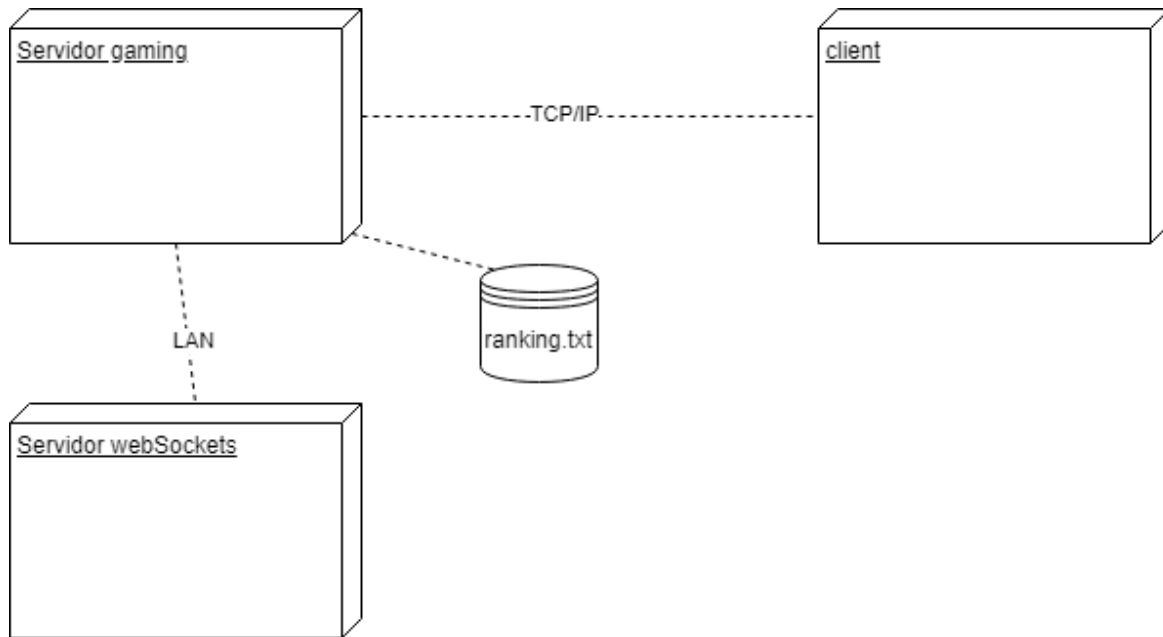
## 2. Diseño del programa:

En la siguiente imagen se muestra el diagrama de paquetes del programa:



*Ilustración 9: Diagrama de paquetes*

En la siguiente imagen se muestra el diagrama de despliegue del programa:



*Ilustración 10: Diagrama de despliegue*

### 3. Librerías utilizadas

Para el desarrollo del programa se utilizaron las siguientes librerías:

- **Nodemon:** Permite realizar cambios en tiempo real al servidor
- **Socket.io:** Permite la gestión de sockets en el servidor
- **Socket.io-client:** Permite conectarse a un servidor web socket desde el cliente
- **Cors:** Permite crear un servidor que atenderá múltiples clientes
- **React-navigation:** Facilita la navegación entre componentes react

#### 4. Análisis de resultados

En la siguiente tabla se muestra una lista de objetivos para los cuales se evaluará si fueron cumplidos o no, en caso de no cumplirse también se indicará la razón del porqué.

Objetivo	Alcanzado	Razón
Realizar una investigación sobre el funcionamiento de web sockets	Si	
Conseguir un diseño simple pero agradable de UI	Si	
Cumplir con las funcionalidades del programa solicitadas	Si	
Obtener una alta rigidez durante la ejecución	Si	
Publicar programa con ngrok	no	El programa se publicaba, pero los sockets no se creaban. En su lugar se publicó el programa usando portFowarding
Acceso al programa desde fuera del localhost	Si	
Documentación interna del programa	Si	
Documentación externa del programa	Si	

<b>Extra:</b> Creación de un modo contrarreloj	No	Falta de tiempo por cuestiones de salud
<b>Extra:</b> Habilitar diferentes temáticas de juego	Si	

*Ilustración 13: Objetivos alcanzados*

En el siguiente enlace se puede encontrar el repositorio que contiene el proyecto  
<https://github.com/JPHuntV/P4-Lenguajes-S2-2022.git>