

**IC4700**

*Lenguajes de Programación*

**IV Proyecto**

*(Parchís Online)*

**Integrantes:**

Mynell Jemuel Myers Hall

Andrew Denilson López Herrera

Ingeniería en Computación

Prof. Allan Rodríguez Dávila

Semestre 1, 2024

16/06/2024

# Manual de usuario

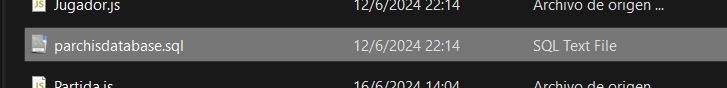
Este proyecto fue construido en Windows usando las herramientas MySQL, Node.js y React.js, por lo que el incumplimiento de este entorno o la falta de las herramientas puede ocasionar fallas o errores en el programa.

## Compilación

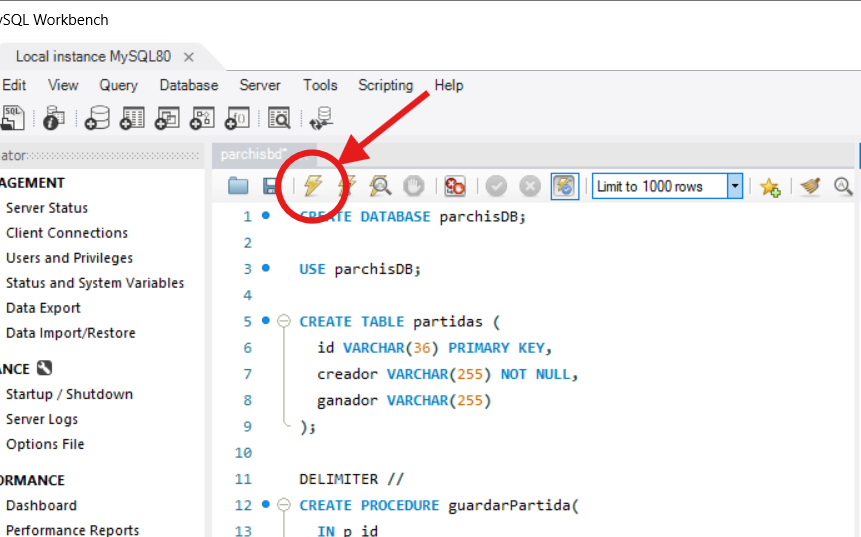
Ante todo, descargaremos el proyecto desde el repositorio de GitHub (Puede encontrar el link del repositorio en la sección de final).

Después debemos asegurarnos que los puertos 3000 y 4000 estén libres y públicos para acceso. La forma de asegurarse de eso lo dejamos a su elección.

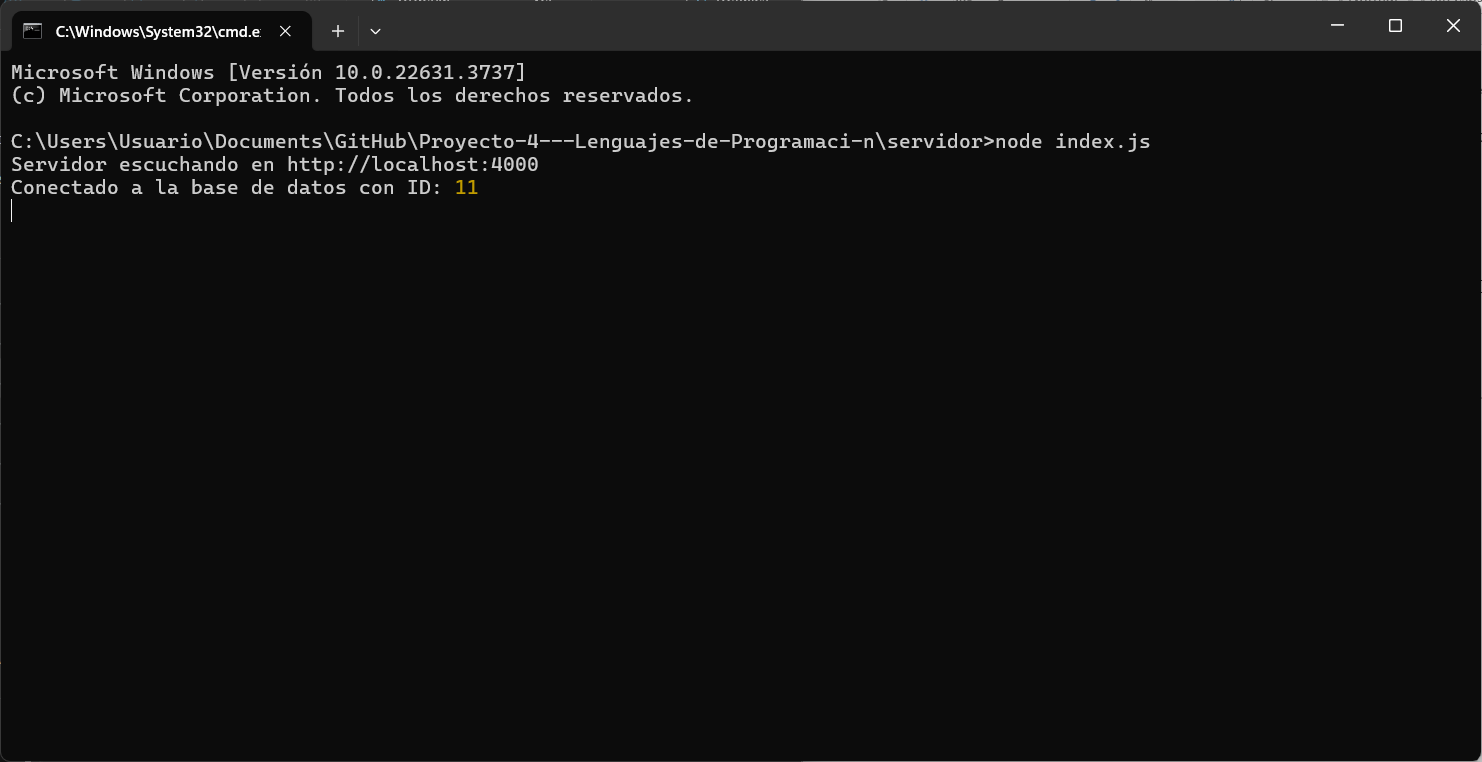
Y como ultimo paso inicial ejecutaremos el Query de la base de datos que encontraremos en la siguiente ruta dentro del proyecto “../programa/servidor/logicadejuego/parchisdatabase.sql”.



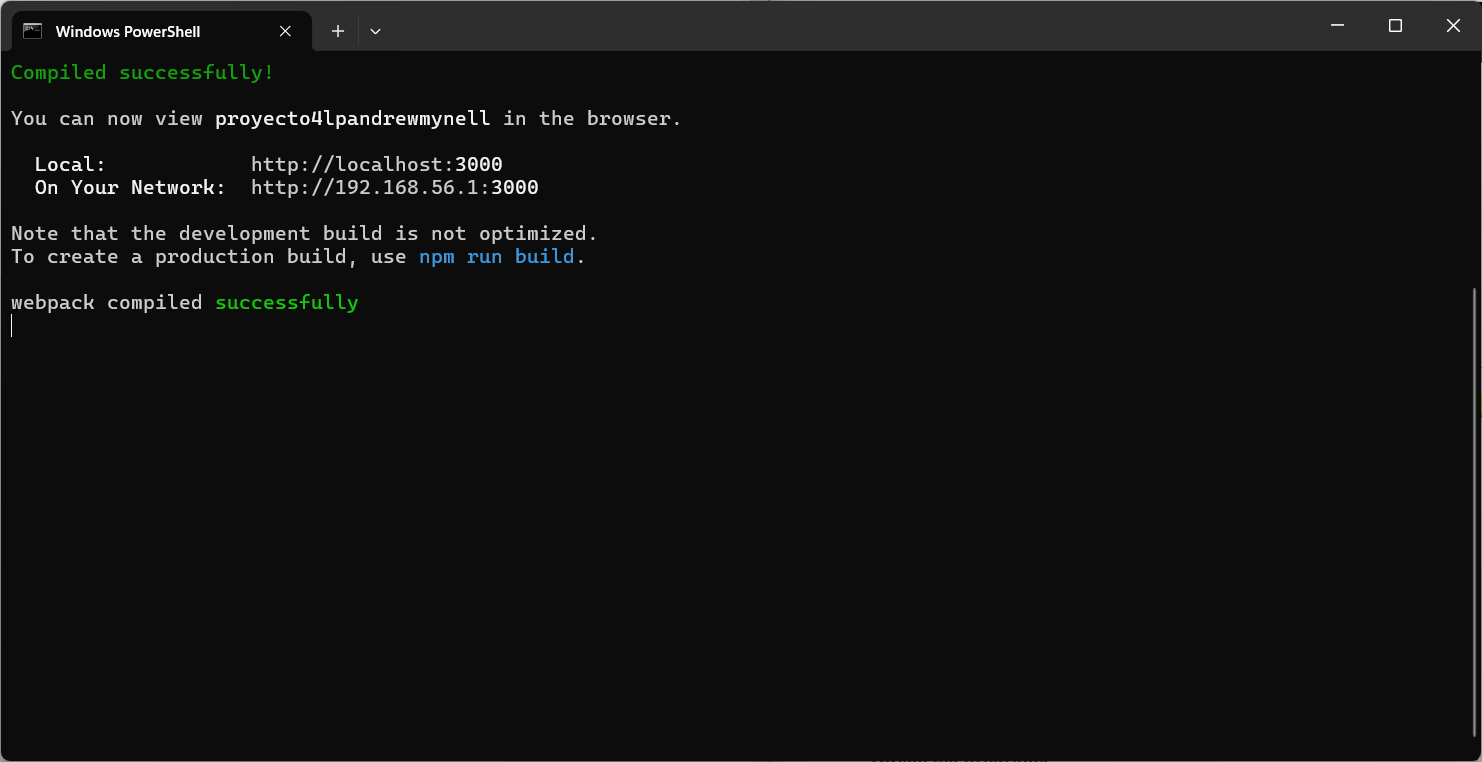
La ejecución deberá ser en MySQL en tu instancia local de la plataforma:



Ahora con los puertos libres, la base de datos y descargado el proyecto nos posicionaremos en la carpeta programa y abriremos la carpeta servidor, luego abriremos el CMD desde esa ruta e ingresaremos el comando “node index.js” (recordar NO cerrar el CMD, ya que de este depende el servidor prendido):

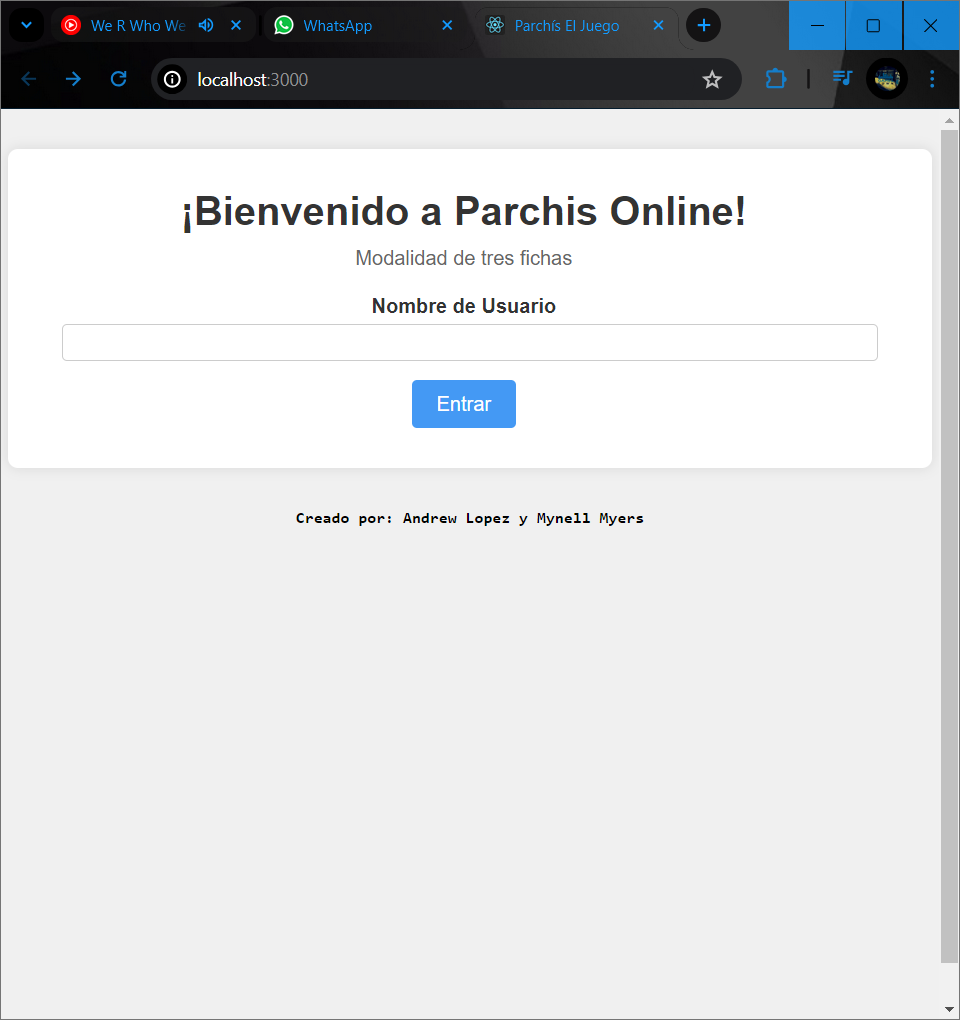


Con el servidor listo seguiremos con la interfaz yendo a la carpeta cliente de la misma carpeta de programa, abriendo otro CMD y ejecutando el comando “npm start” (recordar NO cerrar el CMD, ya que de este depende la interfaz):



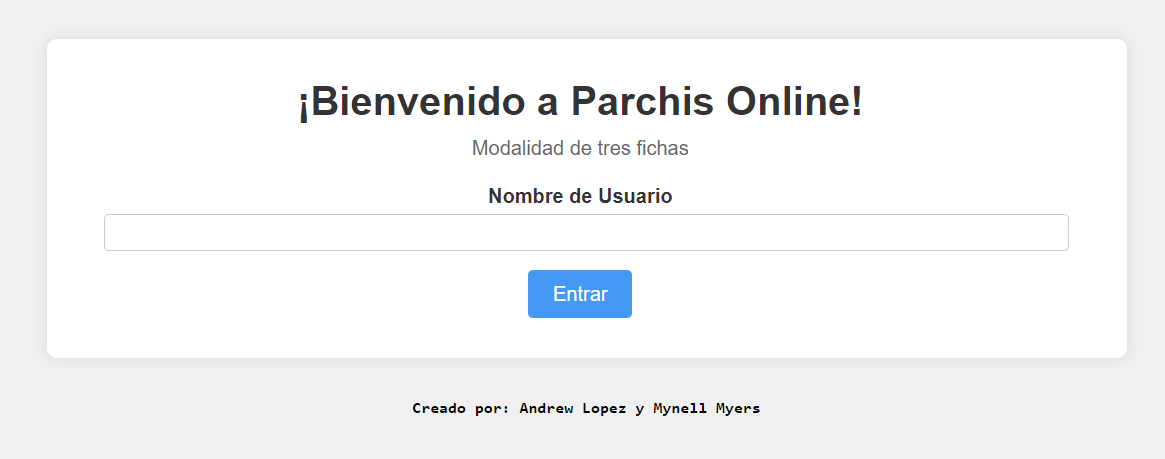
## Ejecución

Para comenzar a usar el programa, solo deberá de ir a <http://localhost:3000/> sí accedes de forma local, gracias a que el programa se ejecuta en tu computador, o con <https://parchistecgame.loca.lt> si estas accediendo de forma remota. Estas rutas a cualquier navegador de tu preferencia, pero preferiblemente utilizar Chrome o Microsoft Edge.



## Uso del programa

Al comenzar el programa se te dará la bienvenida y deberas de poner un nombre con el cual se te podrá identificar en el programa, el nombre no podrá ser vacío.



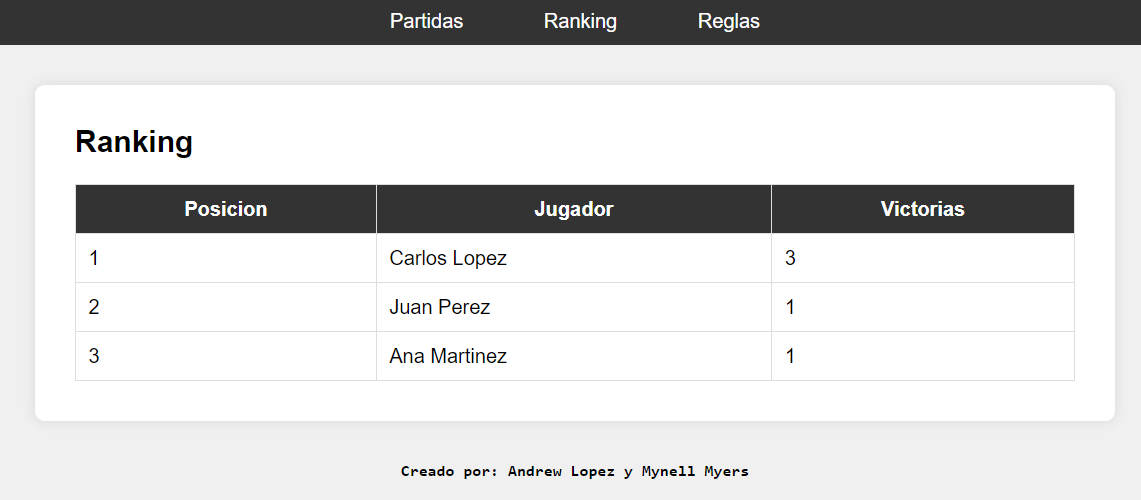
Ahora explicaremos brevemente las 3 áreas que consisten la interfaz, los cuales son; Partidas (Donde crear sala para jugar junto a tus amigos), Ranking (Donde veras esos campeones que han logrado la mayor cantidad de victorias entre los demás) y Reglas (Donde podrás ver en que consiste el juego, si es que se te ha olvidado alguna regla o eres nuevo en el juego).

En partidas estará la lista de partidas, podrás ver las partidas vigentes, cuantos participantes poseen y buscar alguna en particular según el id de la partida o el nombre del usuario que creo la partida. Al tocar una de las partidas se extenderá el área de información de la partida, donde podrás tocar el botón de ingresar donde, ingresaras a la partida y esperaras a más participantes si es necesario.



También estará el pequeño espacio donde podrás generar una partida que este a tu nombre, entrando inmediatamente en ella y solo existiendo esa partida si tu estas dentro de ella, eligiendo antes la cantidad de jugadores en el cuadro de selección al lado del botón de crear partida.

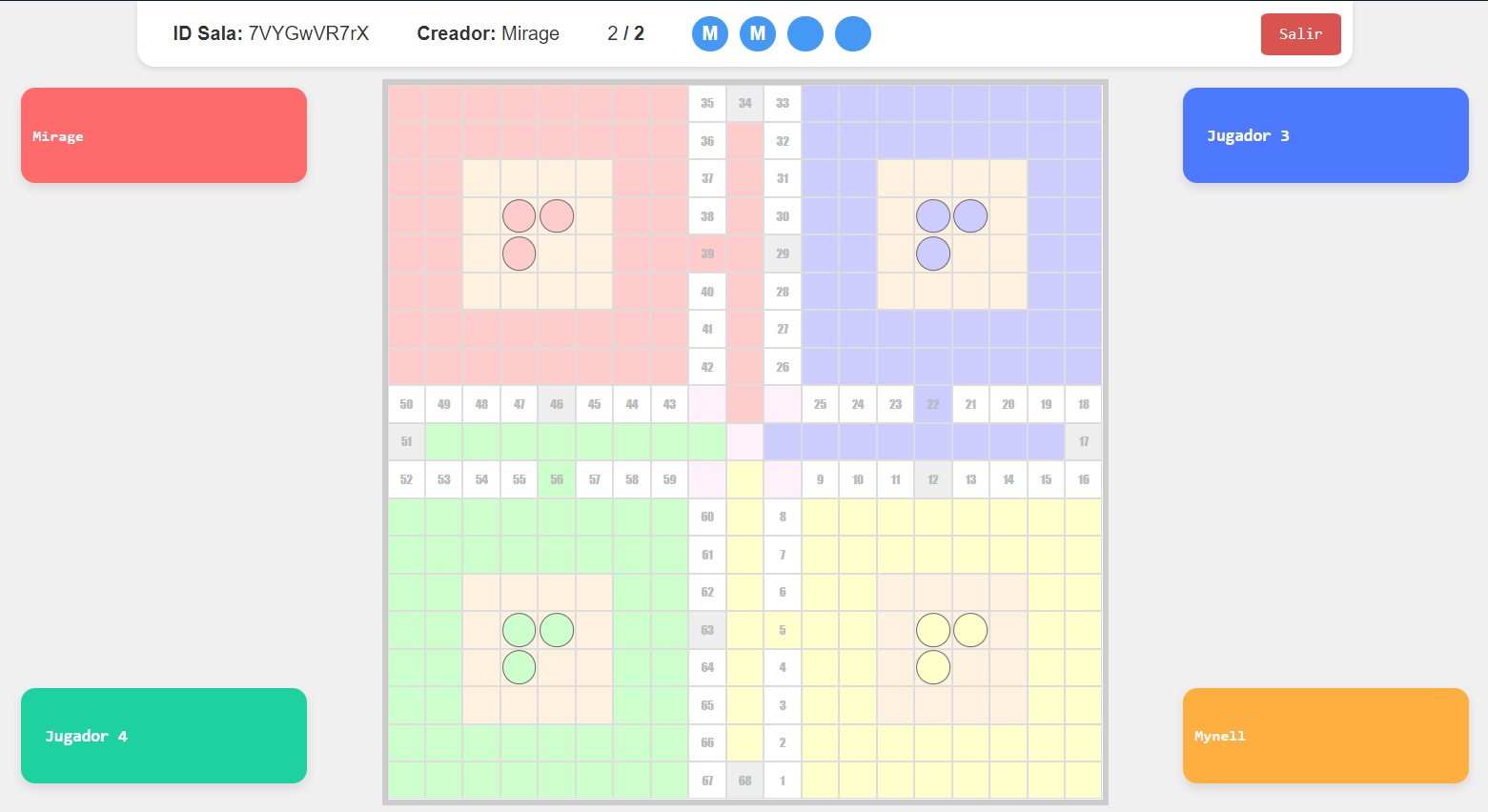
Pasando con el área de ranking, se mostrará una tabla con jugadores registrados en la base de datos, con su posición en el ranking, su nombre y la cantidad de victorias que posee.



Y por último se podrá ver las reglas, con cada uno de sus detalles:



Con respecto al juego, sigue la lógica y funcionamiento descritos en el área de reglas dentro de la interfaz, pero de igual forma la interfaz contara con la información de la partida y el botón para girar el dado.



# Pruebas de funcionalidad

# Descripción del problema

Se requiere una investigación acerca de las funcionalidades detalladas del juego Parchís. Antes de iniciar el trabajo debe analizar por completo el funcionamiento del juego, ya que lo que se desea es reproducir el comportamiento del mismo. El juego debe disponer de las siguientes funcionalidades:

## Cliente

Se debe desarrollar la funcionalidad en la aplicación del cliente en la cual el usuario pueda elegir entre tres opciones:

a) Crear partida

b) Unirse a Juego

c) Ver ranking

## Área de juego predefinida

El sistema debe diseñar un área de juego, con la temática clásica del juego (colores por casas) y con la dimensión y estructura estándar, con las áreas definidas, identificando las casillas, seguros, meta y pasillos.

Esta área de juego se deberá prever para un máximo de cuatro jugadores, cada uno identificado por un color con las estructuras asociadas según el tablero estándar del juego.

## Autenticación

Para ingresar a las funcionalidades del juego, el usuario deberá autenticarse escribiendo un nickname (no hay gestión de usuarios).

## Crear partida

Debe desarrollar la funcionalidad para generar un nuevo juego, donde el usuario podrá personalizar el juego por medio de las siguientes opciones:

a) Nikname del creador:

i. Se debe indicar el nombre de jugador que crea la partida

b) Seleccionar el tipo de juego: i. Vs: los usuarios juegan durante el tiempo que deseen hasta que haya un ganador.

c) Seleccionar cantidad de jugadores:

i. El usuario selecciona la cantidad de jugadores de la partida, dos o cuatro.

d) Una vez creado se generará un código identificador de partida. La partida estará creada por 3 minutos (esto deberá ser un parámetro de configuración que puede ser modificable), si en ese tiempo no es iniciada (debe “arrancar” el juego), la partida se cierra.

## Unirse a juego

Los usuarios tendrán la funcionalidad unirse a una partida: se le muestra los usuarios la lista de partidas por iniciar con su información: creador, identificador, cupo y los usuarios unidos. El usuario le dará unirse y estará a la espera de completar los usuarios de la partida.

Una vez que se completen los usuarios en una partida se habilitará un botón para “entrar” al juego. El sistema cargará la partida mostrando el área de juego con todos sus componentes según la cantidad de usuarios definidos para la partida.

## Juego:

a) El área para poder jugar deberá mostrar las casas, casillas, casillas de salida, seguros, meta, pasillos de meta, y fichas para cada jugador.

La dirección de juego será en sentido contrario a las manecillas del reloj.

La gestión de juego será por turnos según las reglas del juego, para elegir quién inicia, cada jugador “tira un dado” y el orden será de mayor a menor número obtenido, en caso de empate, se le elige por orden alfabético del nickname. Según el orden obtenido, así se asignan los colores, iniciando con el amarillo, en sentido contrario a las manecillas del reloj (considerando si son 2 o 4).

b) Se debe ver el nickname de cada jugador con color que le corresponde.

c) Para el desarrollo del juego se deben aplicar todas las reglas estándar del juego.

d) Se utilizará un dado “cargado”, es decir que los números 4, 5 y 6 tendrán el doble de probabilidades de salir (1,2,3,4,4,5,5,6,6)

e) Se utilizarán 3 fichas en lugar de 4 por cada jugador.

f) Indicar gane: el sistema debe validar el momento en que algún jugador gane.

g) El equipo decidirá la estrategia para simular el juego (controles o acciones).

h) Se deberán visualizar los efectos en el tablero, a los jugadores, una vez que se complete cada turno.

## Estadísticas por juego:

Por cada juego se debe indicar lo siguiente (al finalizar):

a) Nickname de jugadores, indicar el ganador

b) Creador

c) Identificador de partida

## Ranking:

Se debe mostrar información de los ganadores de cada partida. Por cada partida se debe mostrar la siguiente información:

a) Nombre del ganador

b) Creador

c) Identificador de partida

# Diseño del programa

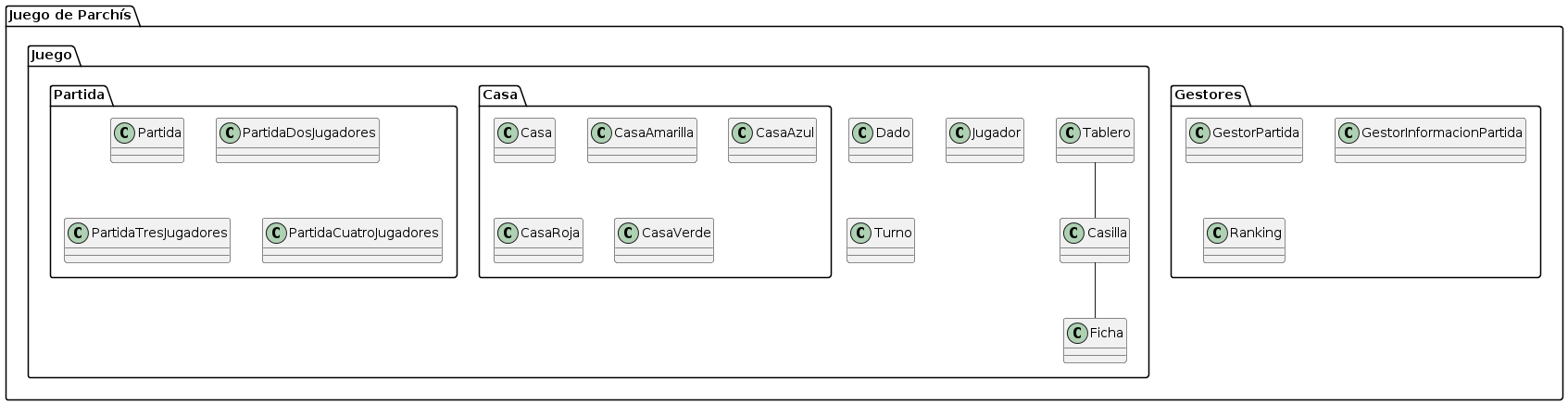
Diagrama de paquetes de planeamiento del proyecto: 

Diagrama de paquetes final:

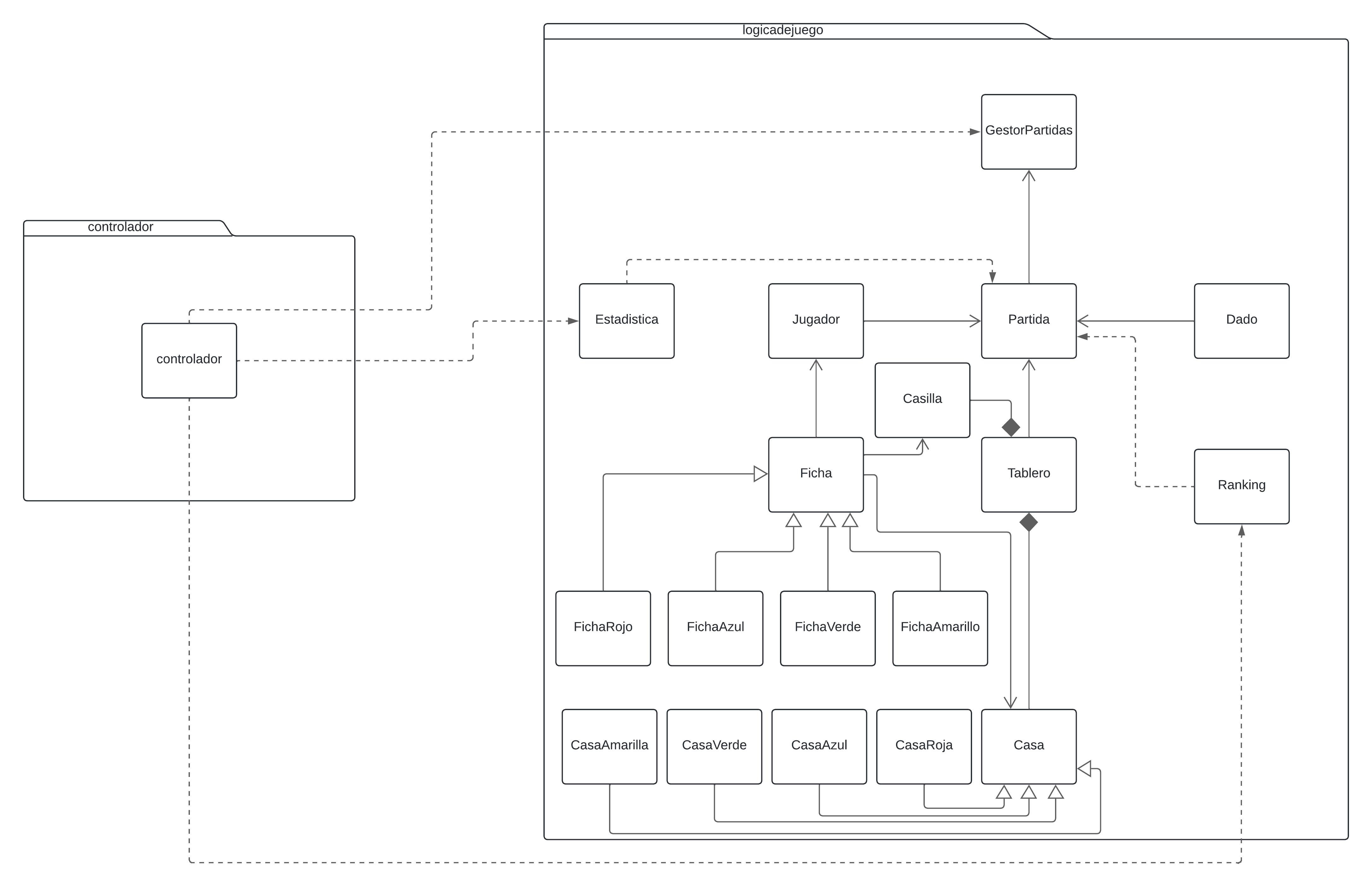


Diagrama de implementación:

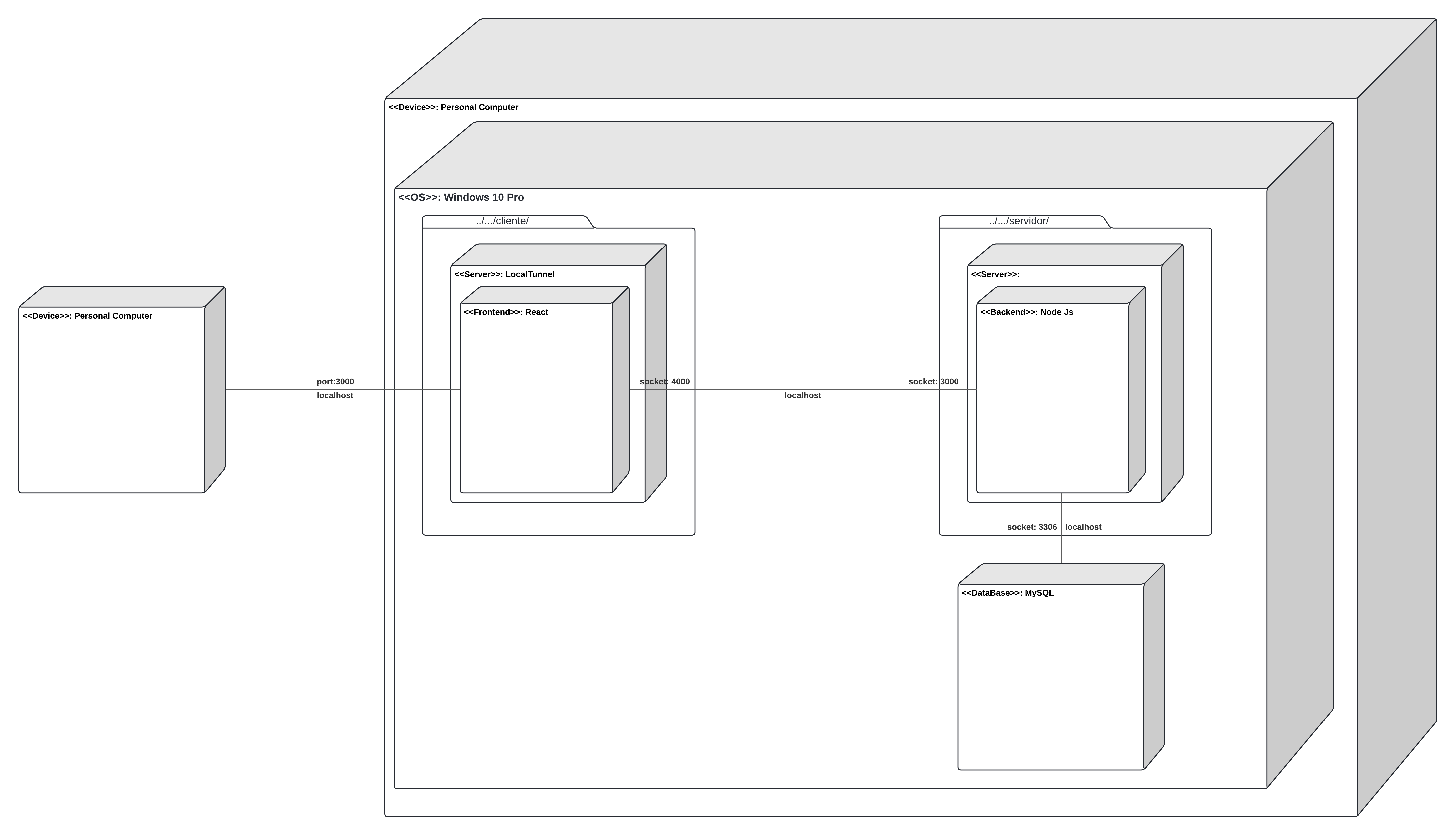
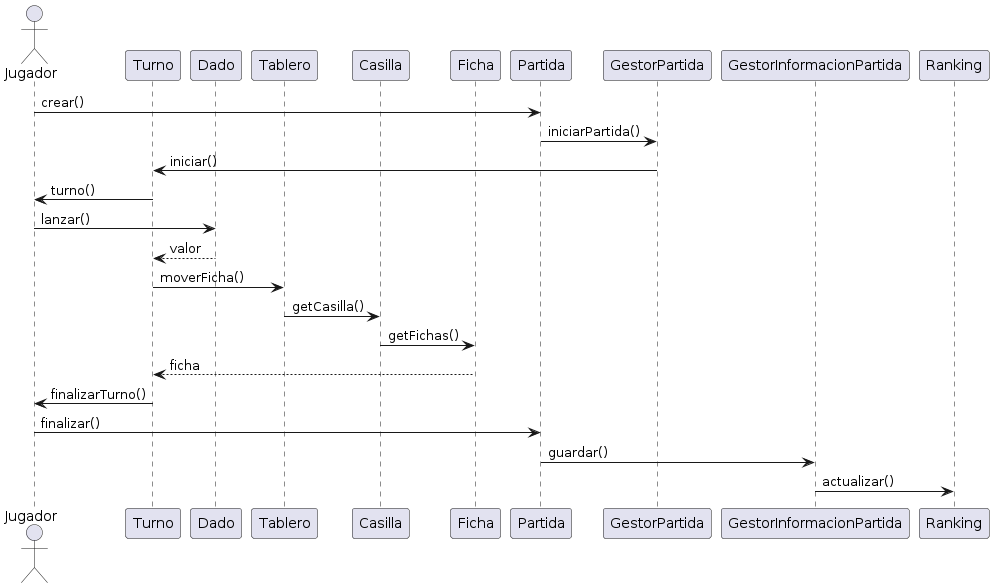


Diagrama de comunicación:



# Librerías utilizadas

React: Se utilizo para emplear algunos módulos propios de react y manejar un correcto funcionamiento del sistema.

Socket.io: Se utilizo para la comunicación a tiempo real entre el servidor y los clientes.

Axios: Principal herramienta utilizada para realizar peticiones por medio del HTTP al servidor.

MySQL2: Importación para crear un cliente de MySQL en Node.js para emplear la base de datos para los registros de partidas y el ranking.

# Análisis de resultados

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivos | Sí | No | Observaciones |
| Documentación interna | X |  |  |
| Documentación externa | X |  |  |
| Crear Partida | X |  |  |
| Unirse a Partida | X |  |  |
| Ver Ranking | X |  |  |
| Autenticación | X |  |  |
| Juego | - | - | El juego inicia, se es capaz de mover las fichas, tirar el dado, pero no de completar un juego ni funcionar correctamente los turnos. |
| Estadísticas | - | - | En el caso de estadísticas, no se pueden mostrar ya que un juego no se puede completar, pero existe un espacio para mostrar las estadísticas si es que se hubiera completado el juego. |
| Extra: Archivo de Requerimiento | X |  |  |

# Repositorio

<https://github.com/AndrewLopezHerrera/Proyecto-4---Lenguajes-de-Programaci-n.git>