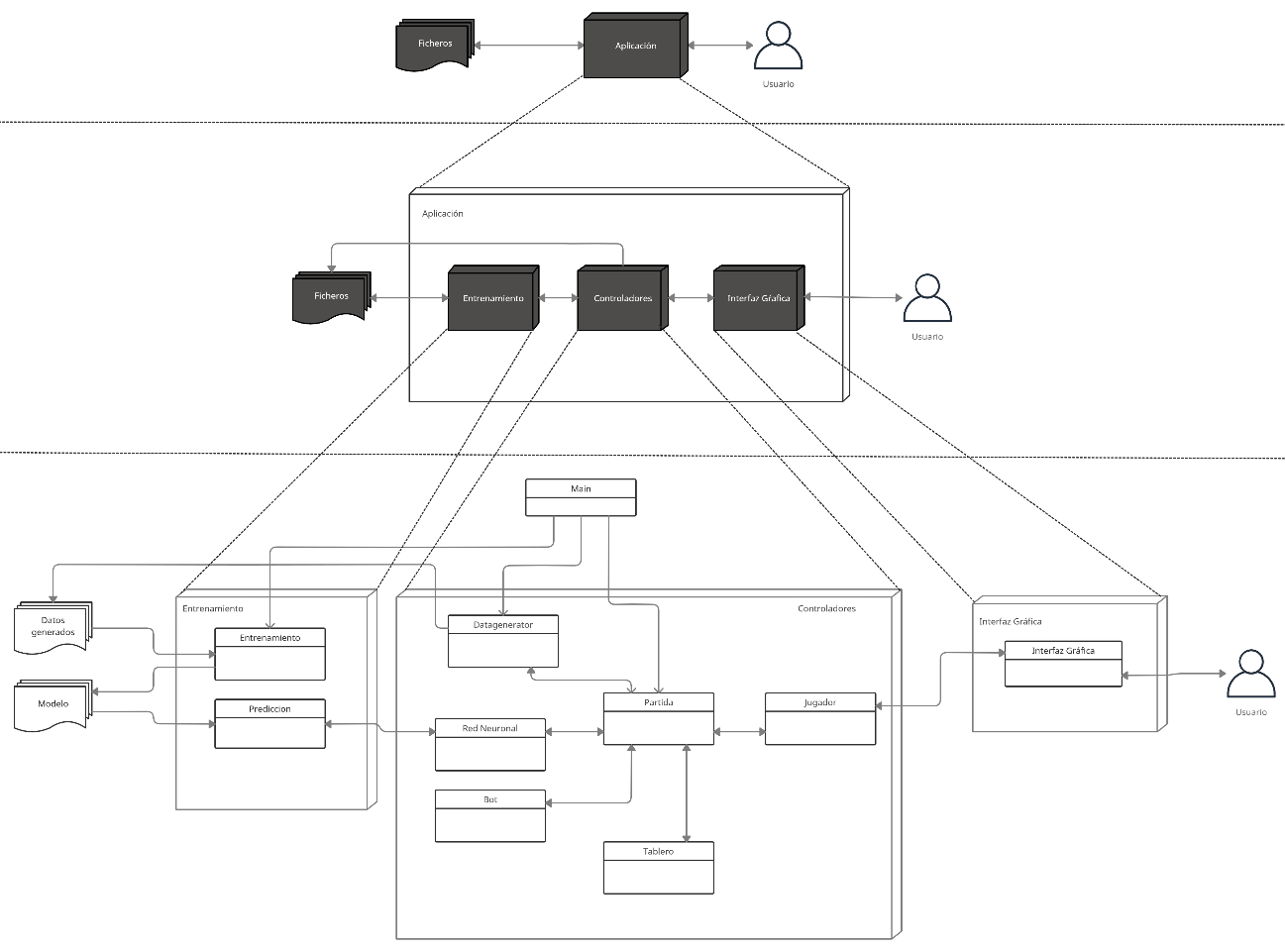
# Diseño del Sistema

## Vista de bloques de construcción

En la siguiente imagen (Imagen 5.2.a) podemos ver una vista general de los bloques de construcción que se describirán en los siguientes puntos. La leyenda de la imagen se comparte con la de todos los puntos de esta sección de la documentación.

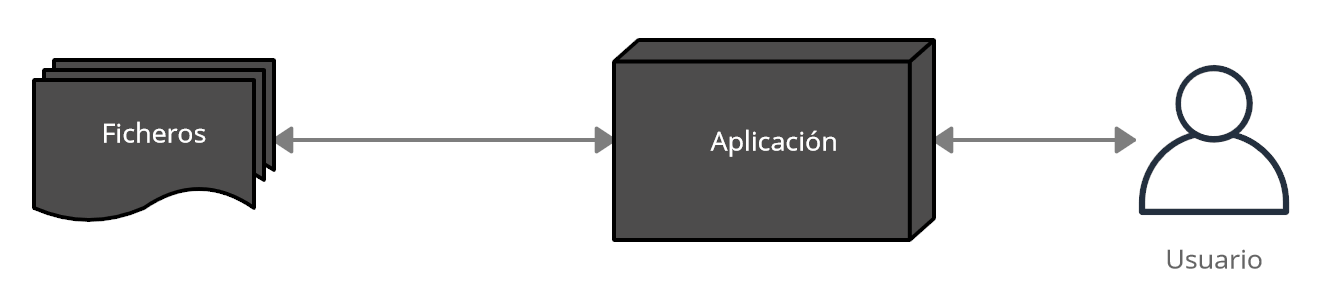


*Imagen 5.2.a*

Leyenda:

|  |  |
| --- | --- |
|  | Persona que interactua con la aplicación |
|  | Bloque de caja negra |
|  | Bloque de caja blanca |
|  | Ficheros de caja negra |
|  | Ficheros de caja blanca |
|  | Clase o archivo de funciones |
|  | Llamada a función sin paso de datos |
|  | Llamada a función con paso de datos |

### Nivel 1

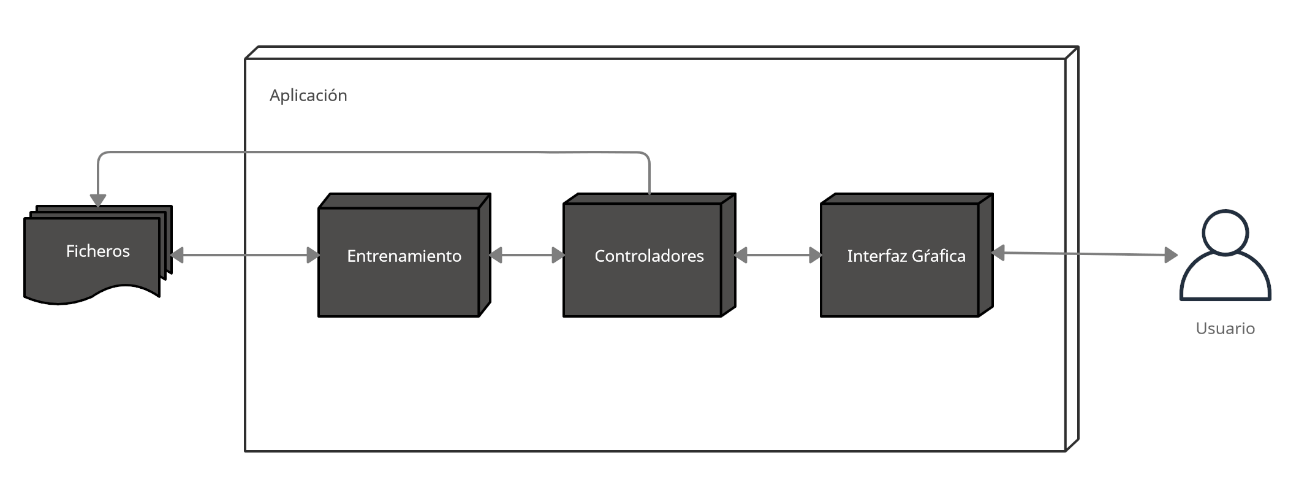


*Imagen 5.2.1.a*

En el nivel 1 podemos ver como la aplicación puede comunicar y recibir información del usuario que la esté ejecutando. Cabe destacar que en caso de que se ejecutando en modo generación de datos o entrenamiento de la red neuronal el usuario no interactúa con la aplicación, esta comunicación se produce únicamente cuando se ejecuta en modo jugar.

También podemos ver como la aplicación se comunica con ficheros externos. Esta comunicación si se realiza para los 3 modos de ejecución como veremos más adelante

### Nivel 2



*Imagen 5.2.2.a*

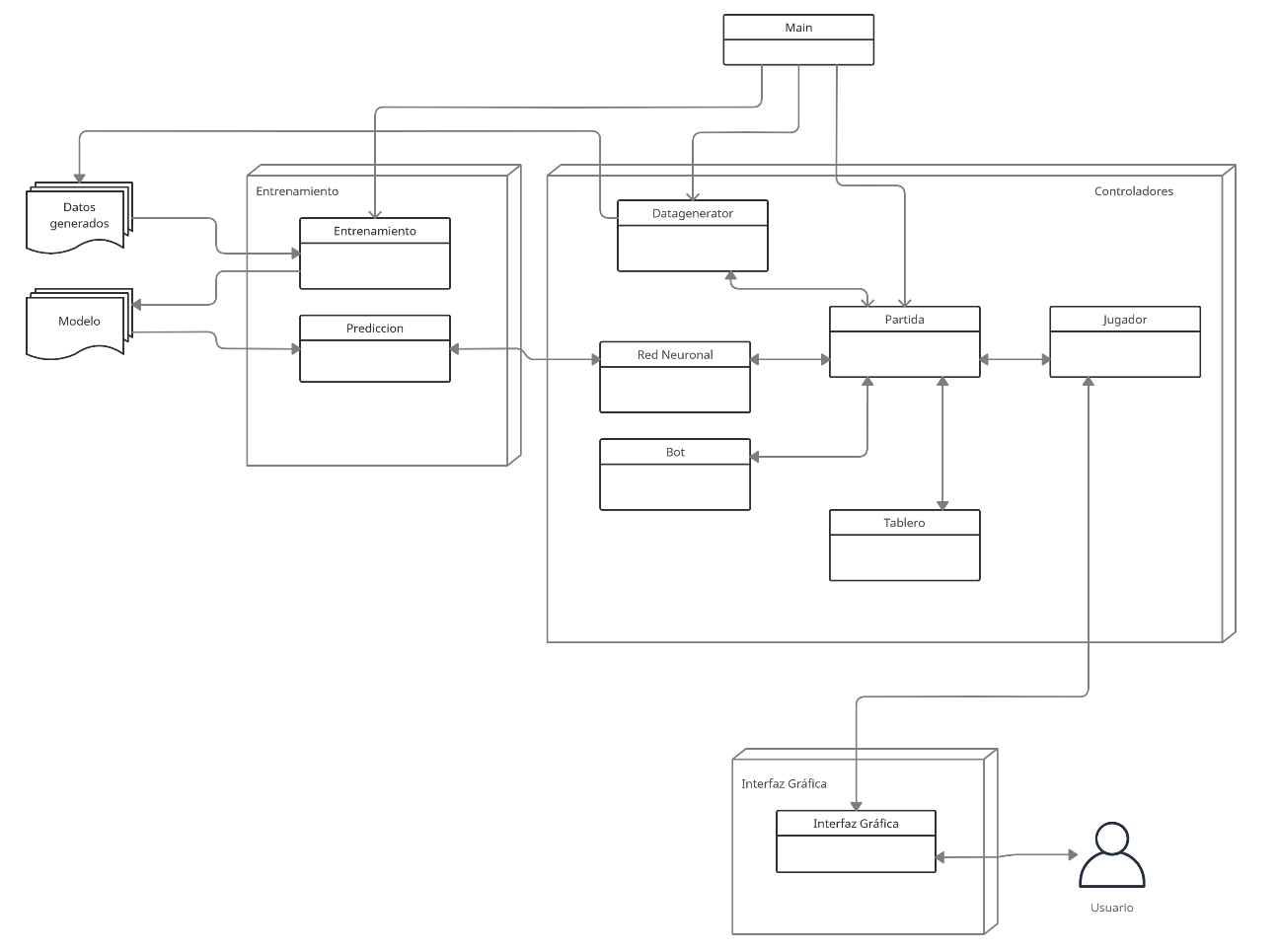
En el nivel 2 podemos ver como la aplicación tiene 3 paquetes principales.

El paquete de controladores está en el centro de ellos porque es el encargado de gestionar el resto, este se comunica con el paquete de entrenamiento de la red neuronal tanto para pedir datos como para recibirlos, así como con el de la interfaz gráfica. También puede generar información que se guarda en ficheros.

El paquete de entrenamiento es el encargado de gestionar la red neuronal. Hace uso y guarda archivos y se comunica con el paquete de controladores.

El paquete de interfaz gráfica es el encargado de comunicar el paquete de controladores con el del usuario.

### Nivel 3

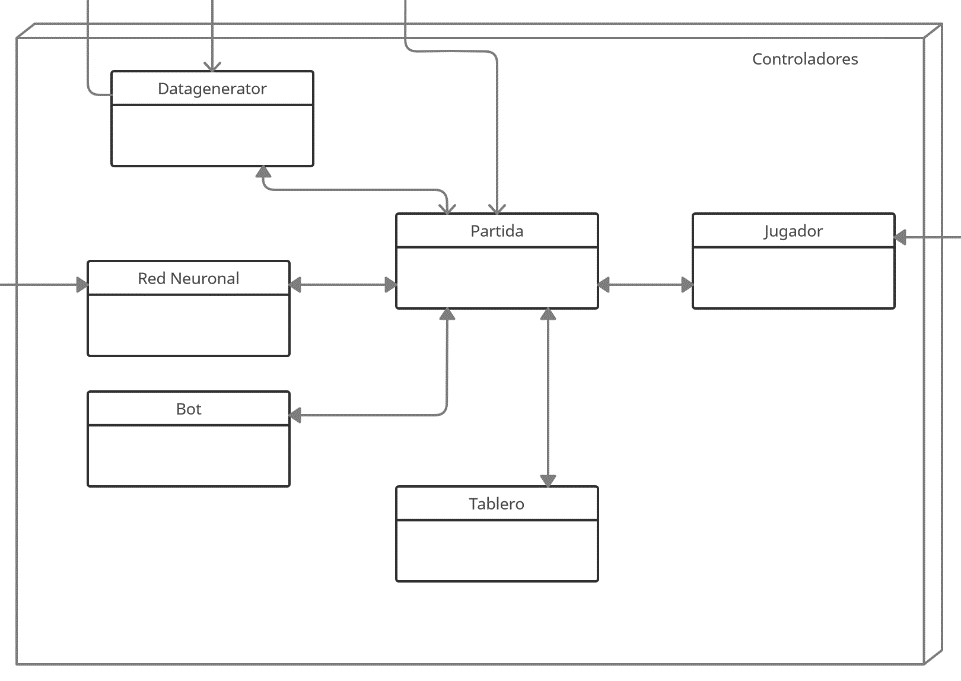


*Imagen 5.2.3.a*

En este nivel podemos ver todas las clases principales de la aplicación y la relación entre ellas.

Desde el main se puede llamar a 3 clases dependiendo del modo de ejecución en el que estemos. Para el modo de generación de datos se llama la clase DataGeneratorController, para el modo de entrenamiento de la red neuronal se llama a la clase Entrenamiento y para el modo jugar se llama a la clase PartidaController.

#### Controladores



*Imagen 5.2.3.1.a*

En el paquete de controladores tenemos 2 clases que se llaman desde el main.

La clase DataGeneratorController sirve para llamar a la clase PartidaController el número de veces que se establezcan en la parametrización, de esta clase recibe los datos de los tableros y acciones que se han realizado en cada partida y se guardan en los ficheros de datos generados.

La clase PartidaController es el centro de la aplicación, ya que es la que se encarga de gestionar y comunicar entre sí gran parte del resto de clases. En el caso de que la aplicación se ejecute en modo de generación de datos, la PartidaController será llamada un numero parametrizado de veces, pero en todas ellas hará lo mismo que si fuera una ejecución en modo jugar, como podemos ver en el apartado 5.3.2 de la documentación. Recibe el tablero de la clase TableroController y se la envía a la clase Red Neuronal (NeuralNetworkController), JugadorController o BotTonto dependiendo de que jugadores estén jugando esa partida. De esas clases recibirá la acción realizada y se la pasará a clase TableroController.

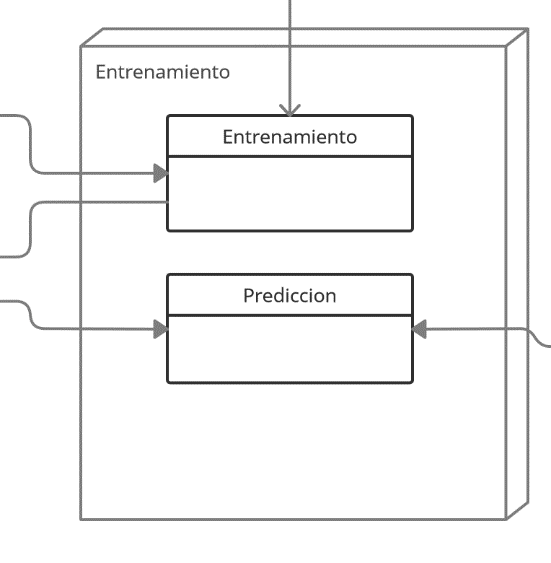
La clase JugadorController sirve de interfaz intermediaria entre la clase PartidaController y la interfaz de usuario.

La clase Red Neuronal (NeuralNetworkController) sirve de intermediaria entre la clase PartidaController y la clase de Predicción que forma parte del paquete de entrenamiento de la red neuronal.

La clase TableroController es la que se encarga de toda la gestión del tablero, lo inicializa al inicio de cada turno para enviárselo a la clase PartidaController, luego recibe las acciones que realicen los diferentes jugadores desde la clase PartidaController y guarda los cambios que realizan esas acciones en el tablero, devolviéndoselo de nuevo para la siguiente acción.

La clase del Bot sirve como jugador aleatorio. Esta clase recibe un tablero y realiza una acción completamente aleatoria dentro de las acciones posibles y con unas cartas aleatorias suficientes para realizar esa acción que estén disponibles en la mano de este jugador. Sirve como jugador inicial para el entrenamiento y como jugador de dificultad baja para el modo jugar si así se define en el archivo de parametrización.

#### Entrenamiento



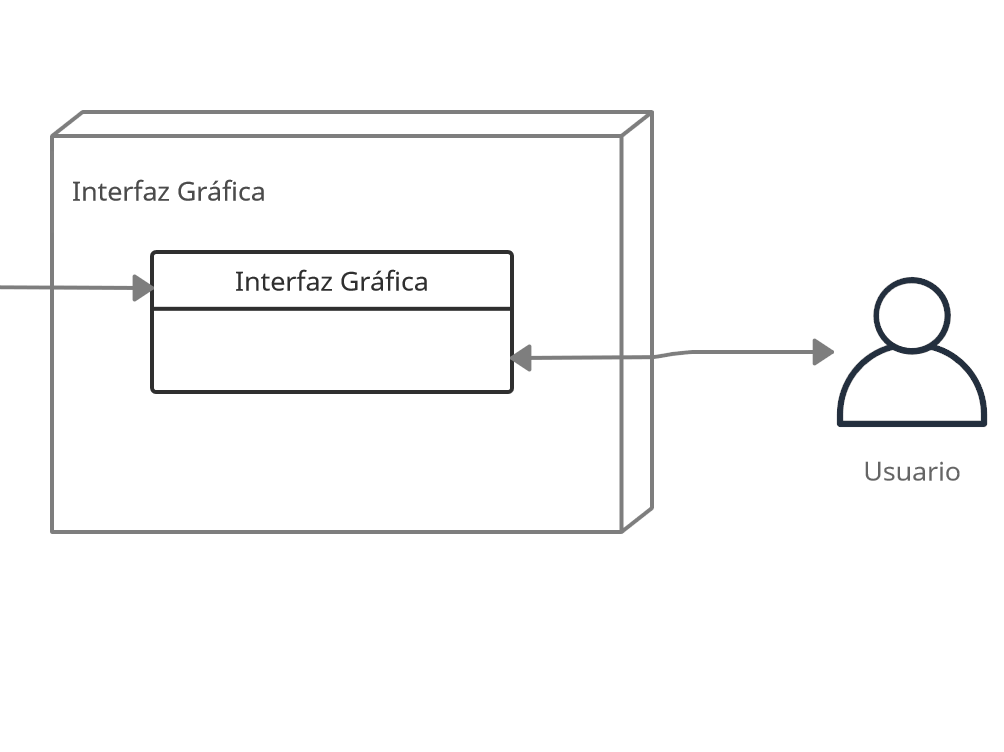
*Imagen 5.2.3.2.a*

En el paquete de entrenamiento hay únicamente 2 clases, la clase de entrenamiento y la de predicción.

La clase de entrenamiento se ejecuta directamente desde el main, lee los archivos de datos y genera los archivos del modelo.

La clase de predicción carga los archivos del modelo, recibe el tablero de la clase de red neuronal y le devuelve una predicción de una acción que será la que realice como jugador.

#### Interfaz grafica



*Imagen 5.2.3.2.a*

En la interfaz gráfica tenemos una única clase que utiliza Tkinter para hacer la ventana gráfica donde el usuario interactuará como jugador con la aplicación, de manera que esta clase pinta el tablero y ofrece al usuario las posibles acciones que puede realizar. Una vez seleccionada la acción esta vuelve a los controladores para seguir la partida.

En este paquete también existe una clase auxiliar que no aparece representada en el modelo ya que solo sirve para informar al usuario de la finalización del programa.

## Vista de tiempo de ejecución

### Opciones

Como hemos visto anteriormente existen 3 opciones parametrizables a la hora de ejecutar la aplicación. Estas opciones se arrancan desde el mismo main y llaman a su clase inicial correspondiente.

* Generar datos
* Entrenar a la red neuronal
* Jugar

La vista de tiempo de ejecución de entrenar a la red neuronal es la más simple de las 3, ya que únicamente se establece el modelo de la red neuronal y se arranca, no hay más que el uso de una sola clase.

Las vistas de generación de datos y jugar es prácticamente la misma, únicamente se diferencian en quienes son los jugadores de cada partida y el número de partidas que se juegan.

Para la opción de generación de datos el main llaman a la clase DataGeneratorController, que se encarga de hacer las llamadas a la clase PartidaController y guardar los datos en un archivo.

Para la opción de Jugar se llama desde el main directamente a la clase PartidaController que es la encargada de gestionar el ciclo completo de ejecución de una partida.

### Ciclo de una partida

Como acabamos de ver la clase PartidaController se encarga de gestionar la partida con un bucle que realiza una ronda completa hasta que uno de los 2 jugadores gana.

Cada ronda se compone de a su vez de un bucle de 4 turnos tal y como se ve representado en la imagen 5.3.2.a.

Al inicio de cada ronda la clase PartidaController pide a la clase TableroController

En cada ciclo del turno cada jugador realiza una acción, siempre en el mismo orden de jugadores para cada turno y intercambiando el orden para cada ronda. El jugador inicial recibe el estado actual del tablero y decide que acción realizar. Si la acción es la acción de “secreto” o la acción de “renuncia” (es decir, es una acción simple) esta acción llega a la clase PartidaController que se encarga de enviarla a la clase TableroController para que haga los cambios correspondientes en el tablero y devolvérselo a la clase PartidaController. Si la acción es de “regalo” o “competición” (es decir, una acción compleja) después de que el jugador realice la acción y la clase TableroController devuelva el resultado en el tablero igual que en una acción simple, la clase PartidaController envía la acción de decisión en el tablero al jugador 2 para que elija las cartas, este una vez ha elegido las cartas le devuelve esa acción a la clase PartidaController y esta vuelve a enviársela a la clase TableroController para que guarde dicha decisión en el tablero.

Una vez terminada la acción del jugador 1 se realiza la acción del jugador 2.

A modo de resumen, cada partida consta de al menos 1 ronda. Cada ronda consta de 4 turnos en los que se realizan 2 acciones, una por cada jugador.

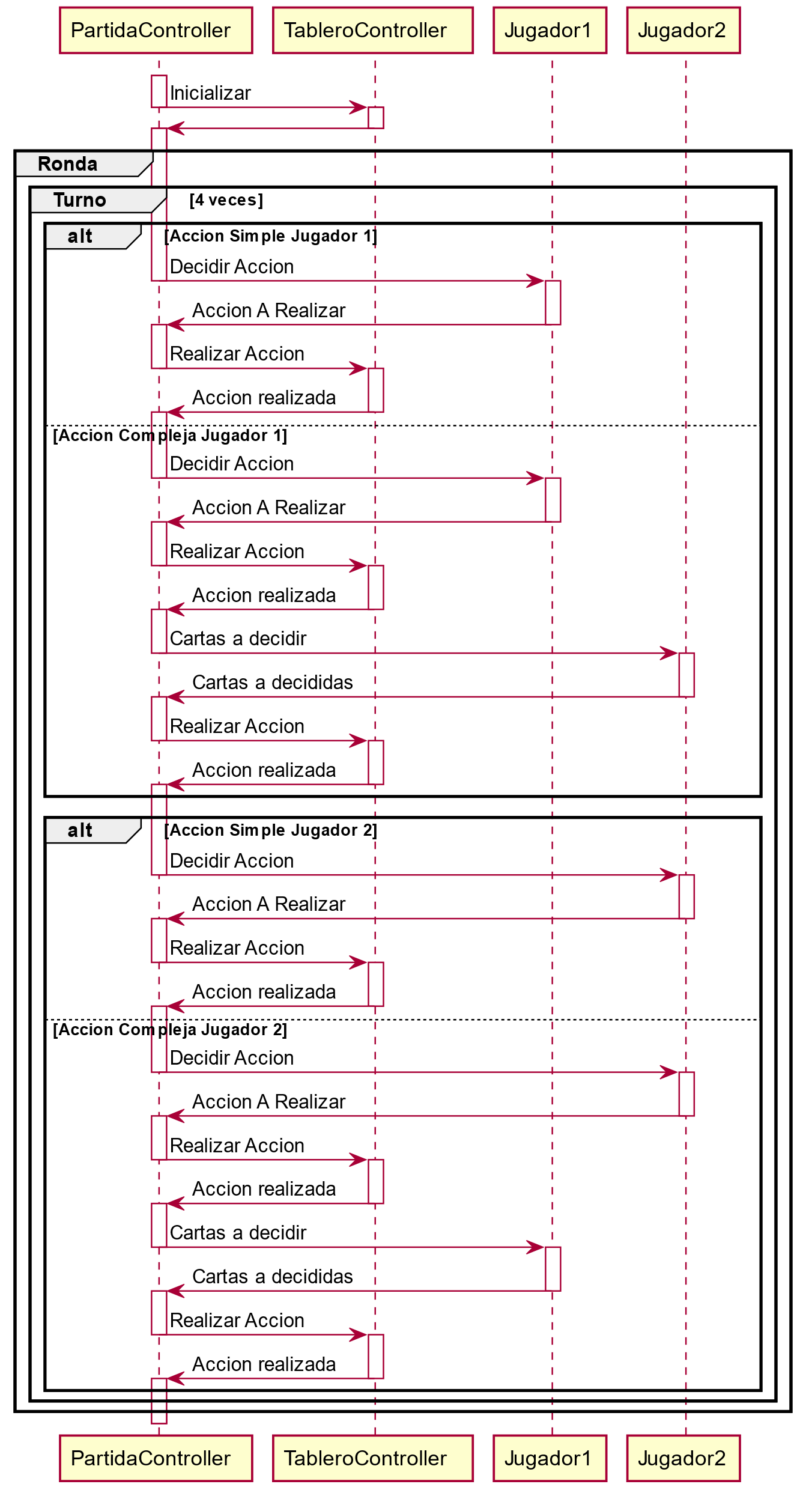
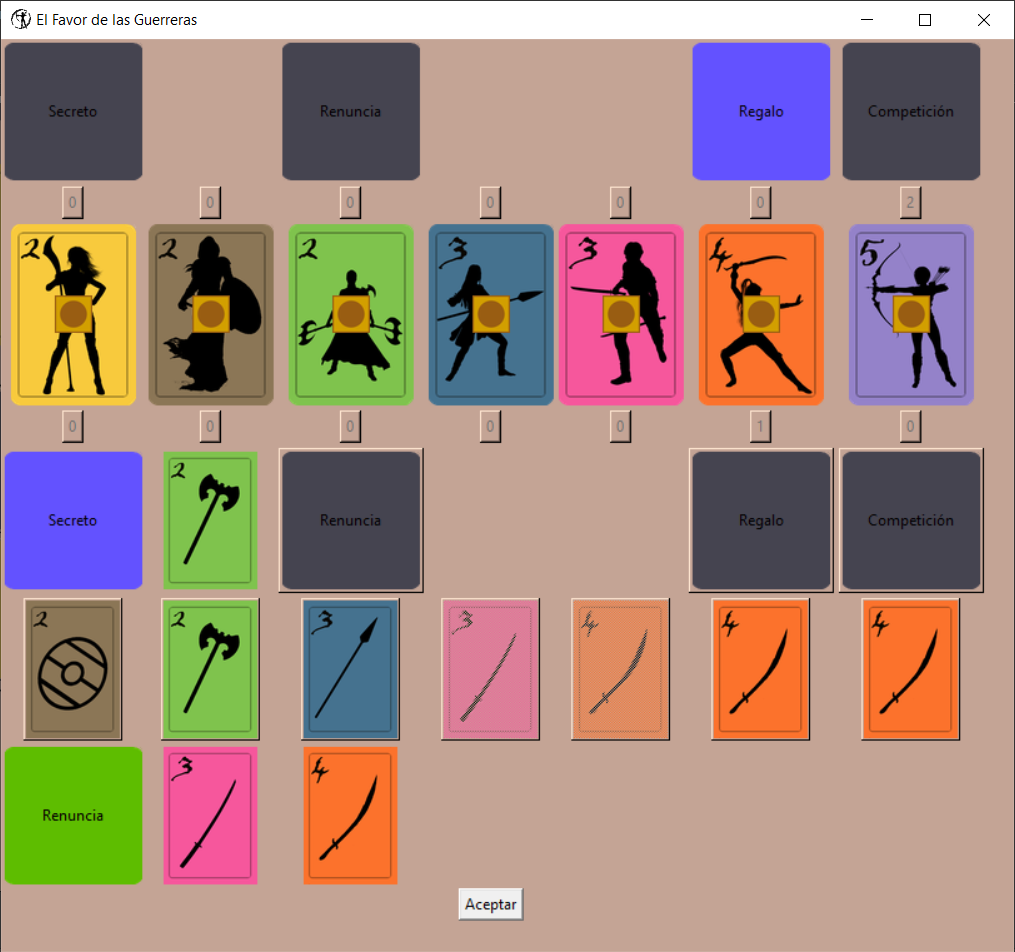


Imagen 5.3.2.a

## Diseño de la Interfaz

La interfaz gráfica se compone de una única pantalla en la que el usuario tiene una visión parcial de la situación actual del tablero.

Se compone de una matriz de 7x7 casillas donde cada fila corresponde a un conjunto diferente de datos que explicaremos a continuación.



*Imagen 4.1.a*

### Acciones del adversario



*Imagen 4.1.1.a*

Al inicio de la partida el adversario no habrá utilizado ninguna de las acciones, por lo que se mostraran todas en un color oscuro indicando esto tal y como se muestra en la Imagen 4.1.1.a.



*Imagen 4.1.1.b*

Cuando el adversario utilice alguna acción esta pasará a estar en un color azul más claro.

Existe un espacio entre la acción de secreto y la acción de renuncia que representa la carta que el adversario haya usado para la acción de secreto. Si no se ha utilizado el hueco estará vacío, mientras que si se ha utilizado aparecerá una carta gris, ya que es una carta oculta y el jugador no puede saber que carta es.

También existen dos espacios entre la acción de renuncia y la de regalo que representan las dos cartas de renuncia del adversario en caso de haber usado ya esta acción al igual que con la carta de secreto.

### Guerreras, armas usadas y favor

**

*Imagen 4.1.2.a*

Estas 3 filas están orientadas, es decir, la parte superior corresponde al adversario y la parte inferior corresponde al jugador.

La fila del medio representa a las guerreras, y el marcador central representan su favor actual. Al inicio de la partida este favor estará siempre en el centro representando que son neutrales. Sin embargo, tras la primera ronda, si ningún jugador ha ganado el favor cambiará dependiendo de las acciones de la ronda anterior de manera que estará en la parte superior de la carta de la guerrera en caso de que esa guerrera este a favor del adversario (imagen 4.1.2.b) o en la parte inferior en caso de este a favor del jugador (imagen 4.1.2.c).

|  |  |
| --- | --- |
| *Imagen 4.1.2.b* | *Imagen 4.1.2.c* |
|  |  |

Los marcadores numéricos que aparecen encima y debajo de las cartas de las guerreras representan el número de cartas que cada jugador ha jugador en favor de esa guerrera durante esta ronda. Los marcadores superiores corresponden al adversario y los inferiores al jugador. Hay que tener en cuenta, las acciones de secreto y renuncia guardan las cartas en la parte de acciones usadas y no se tienen en cuenta hasta el final de la ronda.

### Acciones del jugador



*Imagen 4.1.3.a*

Esta fila es muy similar a la de acciones del adversario al inicio, pero tiene dos diferencias fundamentales. Para empezar las acciones son interactivas, por lo que el usuario puede pulsar encima de ellas para seleccionarla. Cuando se selecciones una acción aparecerá marcada más abajo. Si se desea se puede cambiar de acción seleccionando una diferente, además seleccionar de nuevo una acción reinicia las cartas que hayan sido seleccionadas para esta acción.



*Imagen 4.1.3.b*

Al igual que con las acciones del adversario cuando ya han sido utilizadas en esta ronda se marcan en azul mas claro, pero a diferencia del adversario en esta ocasión si se podrán ver las cartas seleccionadas para las acciones de secreto y renuncia. Además, las acciones utilizadas dejan de ser interactiva, ya que el usuario no puede volver a utilizarlas.



*Imagen 4.1.3.b*

Cuando el jugador se encuentre con una acción de decisión en la que tenga que elegir cartas del adversario, las acciones del jugador quedarán bloqueadas, tornándose en un tono mas grisáceo, pero mostrando diferencia entre las acciones usadas y no usadas, así como las cartas usadas para las acciones de secreto y renuncia para poder tomar una decisión con toda la información del tablero.

### Mano del jugador



*Imagen 4.1.4.a*

En esta fila encontramos las cartas que tiene en la mano el jugador. El color y el símbolo representan a que guerrera corresponden. Estas cartas son interactivas, para poder seleccionarlas hay que haber elegido antes una acción, y dependiendo de la acción elegida se podrán seleccionar un numero distinto de cartas.



*Imagen 4.1.4.b*

Cuando se seleccione alguna carta esta quedará bloqueada, de manera que no se podrá volver a seleccionar. Si se desea cambiar de cartas se puede volver a seleccionar la acción borrando así las cartas seleccionadas.



*Imagen 4.1.4.c*

Cuando el jugador se encuentre con una acción de decisión en la que tenga que elegir cartas del adversario, las cartas del jugador quedarán bloqueadas al igual que las acciones.

### Acción elegida



*Imagen 4.1.5.a*

Esta fila aparecerá en un principio vacía, pero irá cambiando, dependiendo de la acción y las cartas que seleccione el jugador. No es interactiva, ya que el jugador no puede pulsar en ninguna casilla. En caso de que el jugador tenga que realizar una acción de decisión esta aparecerá en esta fila.

#### Acción por realizar

En la primera casilla se muestra la acción que ha seleccionado el jugador (ver sección 4.1.3). A continuación, se muestran las cartas que ha seleccionado el jugador. Es importante destacar que en para la acción de competición el orden en el que se seleccionen las cartas tiene relevancia, ya que las dos primeras formarán un grupo y las dos segundas el otro grupo. Para el resto de las acciones el orden es indiferente.

#### Acción de decisión



*Imagen 4.1.5.2.a*

Cuando el jugador tenga que realizar una acción de decisión en el lugar de la acción aparecerá una breve descripción de lo que debe hacer. En caso de ser una acción de regalo tendrá que seleccionar una carta de las 3, la cual aparecerá bloqueada después de ser seleccionada, pudiendo seleccionar cualquiera de las otras dos si el jugador cambia de opinión.

En caso de ser una acción de competición el jugador podrá elegir entre el grupo de las dos primeras o el grupo de las dos últimas. Al pulsar sobre una carta se quedan marcadas las dos cartas de ese grupo, pudiendo cambiar de grupo en caso de cambio de opinión.

### Botón de aceptar

En último lugar tenemos el botón de aceptar, este botón aparece únicamente cuando el jugador ha seleccionado una acción y el correspondiente grupo de cargas o ha seleccionado las cartas que desea en la acción de decisión. En caso de que el jugador cambie de acción y las cartas seleccionadas se reinicien el botón desaparecerá hasta que el jugador vuelva a seleccionar el numero correcto de cartas.

Al pulsar este botón la acción queda registrada y se envía a la aplicación, para que el adversario realice su acción y le vuelva a tocar el turno al jugador.

### Pantalla de información final

Cuando se acaba la partida se cierra la pantalla principal y aparece una pequeña ventana anunciando el resultado de la partida.