

Videojuego de ajedrez con modo de juego contra la máquina

Anexo II: Análisis y diseño del sistema

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Junio de 2025

Autor

Óscar Sánchez Rubio

Tutores

Luis Augusto Silva Zendron

Gabriel Villarrubia González

Índice

Contenidos

1.	Introducción	1
2.	Arquitectura inicial	2
3.	Modelo de análisis	3
3.1.	Modelo de dominio	3
3.2.	Realización de casos de uso a nivel de análisis	4
3.3.	Paquetes de análisis.....	15
3.4.	Propuesta de arquitectura	17
4.	Modelo de diseño	19
4.1.	Patrón arquitectónico	19
4.2.	Arquitectura de subsistemas	20
4.3.	Clases de diseño	21
4.4.	Realización de casos de uso a nivel de diseño	23
5.	Modelo de despliegue	31
6.	Diseño de la interfaz	32
6.1.	Mapa del sitio.....	32
6.2.	Wireframes.....	33
6.2.1.	Wireframe del menú principal	33
6.2.2.	Wireframe de configuraciones previas.....	34
6.2.3.	Wireframe de partida.....	35
6.2.4.	Wireframe de promoción	35
6.2.5.	Wireframe de final de partida	36
7.	Bibliografía	37

Ilustraciones

Ilustración 1: Diagrama de paquetes de servicio.....	2
Ilustración 2: Diagrama de clases	3
Ilustración 3: Diagrama de secuencia CU-01 Seleccionar pieza	5
Ilustración 4: Diagrama de secuencia CU-02 Mostrar movimientos legales para una pieza	5
Ilustración 5: Diagrama de secuencia CU-03 Deseleccionar pieza	6
Ilustración 6: Diagrama de secuencia CU-04 Mover pieza	6
Ilustración 7: Diagrama de secuencia CU-05 Detectar jaque	7
Ilustración 8: Diagrama de secuencia CU-06 Detectar jaque mate.....	7
Ilustración 9: Diagrama de secuencia CU-07 Detectar tablas.....	8
Ilustración 10: Diagrama de secuencia CU-08 Finalizar partida	8
Ilustración 11: Diagrama de secuencia CU-09 Cambiar nombre jugador	9

Ilustración 12: Diagrama de secuencia CU-10 Cambiar color de las piezas de los jugadores	9
Ilustración 13: Diagrama de secuencia CU-11 Elegir modo de juego contra otro jugador	10
Ilustración 14: Diagrama de secuencia CU-12 Elegir modo de juego contra la máquina	10
Ilustración 15: Diagrama de secuencia CU-13 Empezar partida	11
Ilustración 16: Diagrama de secuencia CU-14 Buscar movimientos para la IA.....	11
Ilustración 17: Diagrama de secuencia CU-15 Evaluar movimientos	12
Ilustración 18: Diagrama de secuencia CU-16 Decrementar tiempo restante a los jugadores	13
Ilustración 19: Diagrama de secuencia CU-17 Elegir dificultad de la IA.....	13
Ilustración 20: Diagrama de secuencia CU-18 Cargar distribución del tablero en formato FEN	14
Ilustración 21: Diagrama de secuencia CU-19 Calcular tiempo de búsqueda.....	14
Ilustración 22: Paquete de análisis Gestión de configuraciones previas.....	15
Ilustración 23: Paquete de análisis Gestión de partidas	16
Ilustración 24: Paquete de análisis Gestión de inteligencia artificial.....	17
Ilustración 25: Propuesta de arquitectura	18
Ilustración 26: Diagrama de la arquitectura basada en capas	20
Ilustración 27: Vista arquitectónica	21
Ilustración 28: Subsistema de diseño Capa de presentación.....	22
Ilustración 29: Subsistema de diseño Control.....	22
Ilustración 30: Subsistema de diseño Lógica	23
Ilustración 31: Subsistema de diseño Lógica	23
Ilustración 32: Diagrama de secuencia de diseño CU-01 Seleccionar pieza	24
Ilustración 33: Diagrama de secuencia de diseño CU-02 Mostrar movimientos legales para una pieza	24
Ilustración 34: Diagrama de secuencia de diseño CU-03 Deseleccionar pieza	25
Ilustración 35: Diagrama de secuencia de diseño CU-04 Mover pieza	25
Ilustración 36: Diagrama de secuencia de diseño CU-05 Detectar jaque	25
Ilustración 37: Diagrama de secuencia de diseño CU-06 Detectar jaque mate	26
Ilustración 38: Diagrama de secuencia de diseño CU-07 Detectar tablas	26
Ilustración 39: Diagrama de secuencia de diseño CU-08 Finalizar partida	26
Ilustración 40: Diagrama de secuencia de diseño CU-09 Cambiar nombre jugador ...	27
Ilustración 41: Diagrama de secuencia de diseño CU-10 Cambiar color de las piezas de los jugadores	27
Ilustración 42: Diagrama de secuencia de diseño CU-11 Elegir modo de juego contra otro jugador	27
Ilustración 43: Diagrama de secuencia de diseño CU-12 Elegir modo de juego contra la máquina.....	27
Ilustración 44: Diagrama de secuencia de diseño CU-13 Empezar partida	28
Ilustración 45: Diagrama de secuencia de diseño CU-14 Buscar movimientos para la IA	28
Ilustración 46: Diagrama de secuencia de diseño CU-15 Evaluar movimientos	29
Ilustración 47: Diagrama de secuencia de diseño CU-16 Decrementar tiempo restante a los jugadores	29
Ilustración 48: Diagrama de secuencia de diseño CU-17 Elegir dificultad de la IA ...	30
Ilustración 49: Diagrama de secuencia de diseño CU-18 Cargar distribución del tablero en formato FEN.....	30

Ilustración 50: Diagrama de secuencia de diseño CU-19 Calcular tiempo de búsqueda	30
Ilustración 51: Diagrama de despliegue	31
Ilustración 52: Mapa del sitio	32
Ilustración 53: Wireframe del menú principal	33
Ilustración 54: Wireframe de configuraciones previas	34
Ilustración 55: Wireframe de partida	35
Ilustración 56: Wireframe de promoción	35
Ilustración 57: Wireframe de final de partida	36

1. Introducción

En este anexo se recoge el análisis y diseño del sistema desarrollado, aplicando los principios de la ingeniería del software y utilizando el lenguaje de modelado unificado (UML) como herramienta principal de representación. El apartado de análisis describe cómo se han interpretado y estructurado los requisitos obtenidos durante la fase de especificación, mediante la identificación de las clases, sus relaciones y sus responsabilidades. Por su parte, el diseño detalla las decisiones arquitectónicas adoptadas, así como la organización de los componentes y su interacción para dar respuesta a los objetivos funcionales y no funcionales del proyecto.

2. Arquitectura inicial

La arquitectura inicial del sistema que se desarrollará se estructurará en torno a la funcionalidad, y estará compuesta por paquetes funcionales o de servicio. Estos paquetes agrupan las diferentes funciones del sistema y se corresponden con los paquetes establecidos en el modelo de casos de uso definido durante la etapa de requisitos. El diagrama de paquetes de servicio es el siguiente:

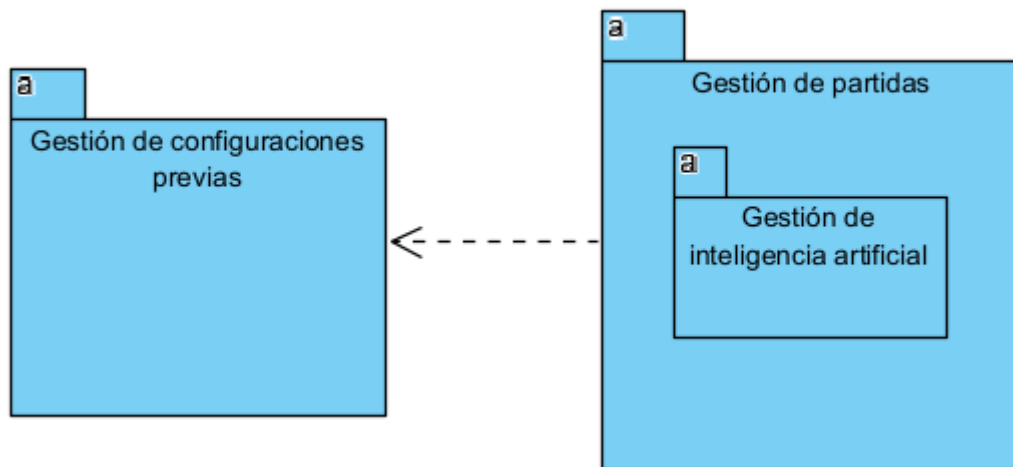


Ilustración 1: Diagrama de paquetes de servicio

Podemos observar que la jerarquización de los paquetes es idéntica a la del anexo anterior, topándonos con dos paquetes principales, como son: **Gestión de configuraciones previas** y **Gestión de partida**, además de un paquete de **Gestión de inteligencia artificial** dentro del paquete de partida. La agrupación que puede ser considerada como el paquete principal es la Gestión de partida, la cual depende del paquete de Gestión de configuraciones previas debido a que este determina el comportamiento de algunas funcionalidades propias de la partida.

3. Modelo de análisis

El modelo del análisis tiene como objetivo refinar y estructurar los requisitos funcionales del sistema mediante una representación más detallada y orientada a objetos. A partir del modelo de dominio y del estudio de los casos de uso, este modelo permite identificar las clases de análisis, sus responsabilidades, relaciones y comportamientos, sirviendo como puente entre la especificación de requisitos y el diseño técnico del sistema. En este apartado se describen los elementos clave del modelo del análisis, como son:

- **Modelo de dominio**
- **Casos de uso a nivel de análisis**
- **Paquetes de análisis**
- **Propuesta de arquitectura**

3.1. Modelo de dominio

El modelo de dominio constituye una representación conceptual del sistema, donde se identifican y describen las principales entidades, sus atributos y las relaciones existentes entre ellas. Este modelo sirve como base para comprender el funcionamiento del sistema desde una perspectiva orientada al problema, y facilita la transición hacia las fases de diseño e implementación. En este apartado se presentan los elementos clave del dominio de la aplicación desarrollada, reflejando de forma estructurada los objetos, conceptos y asociaciones que intervienen en el contexto del proyecto. El modelo de dominio se ha elaborado a partir de los requisitos obtenidos anteriormente, con el propósito de garantizar una correcta alineación entre los objetivos del sistema y la solución propuesta.

Aunque esta representación tiene un nivel de detalle más elevado (menor nivel de abstracción), no hay que olvidar que los atributos que se especificarán serán a alto nivel y no se especificarán operaciones.

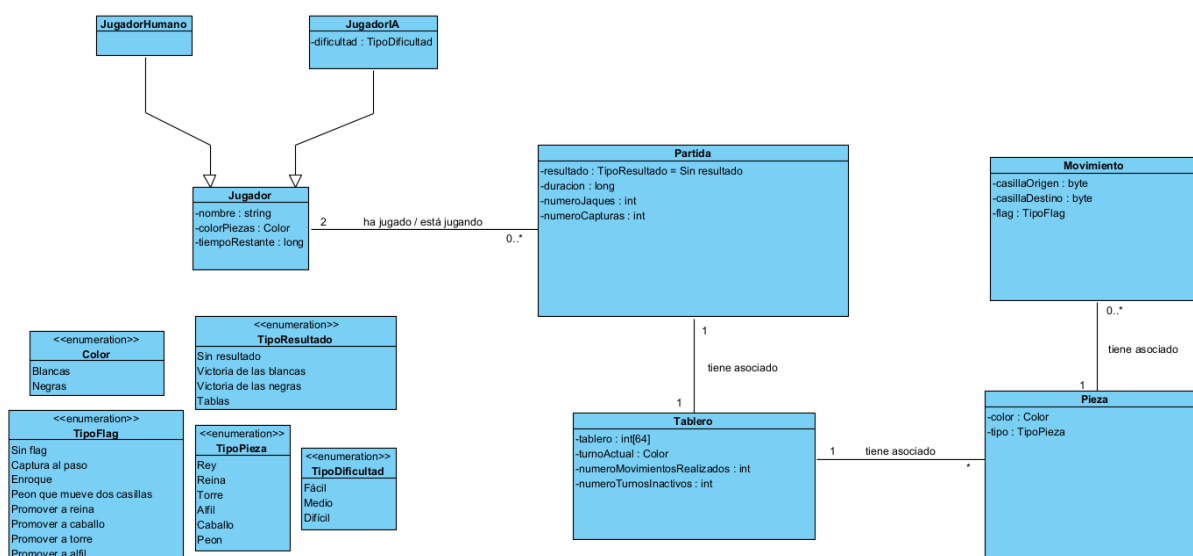


Ilustración 2: Diagrama de clases

Como puntos a destacar del modelo de dominio, se puede mencionar la definición de una clase principal de jugador, de la que heredan **JugadorHumano**, que aún no tiene ningún atributo exclusivo, y **JugadorIA**, con un atributo para especificar la dificultad de la inteligencia artificial, como se venía definiendo en los objetivos del sistema. Todas las clases, salvo JugadorHumano, han sido establecidas, partiendo de los requisitos de información que se habían identificado.

También se puede observar cómo se han definido tipos de atributos específicos (enumeraciones), los cuales se utilizan cuando el atributo sólo puede tomar un conjunto finito y bien definido de valores. Un ejemplo de esto es el mencionado en el párrafo anterior **TipoDificultad**, que puede tomar tres valores: fácil, medio, difícil.

Por último, y en cuanto a la relaciones entre las clases, se puede que cada usuario tiene asociado un jugador en el momento de entrar en partida. Y cada partida, a su vez, está siendo jugada siempre por dos jugadores.

3.2.Realización de casos de uso a nivel de análisis

Dentro del modelo de análisis se incluye la realización de los casos de uso en esta fase. Esta realización consiste en describir cómo se llevan a cabo los casos de uso definidos en la disciplina de requisitos, en función de las clases de análisis que se han identificado previamente. El objetivo es detallar con mayor precisión la secuencia de pasos necesaria para implementar cada funcionalidad del sistema, poniendo el foco en el flujo de información que se establece entre las distintas clases de análisis.

Para elaborar estas realizaciones, se emplea el **Diagrama de Secuencia**, una herramienta de modelado proporcionada por UML. Este tipo de diagrama permite representar el intercambio de mensajes entre los diferentes objetos (clases) implicados en la implementación de cada funcionalidad, definiendo así el comportamiento dinámico del sistema.

En los diagramas de secuencia a nivel de análisis se indican las operaciones o intercambios de mensajes de forma orientativa (García). En algunos casos, se pueden incluir atributos asociados a estas operaciones, aunque no de forma exhaustiva, ya que el objetivo principal en esta etapa es describir el intercambio de información. Será en la fase de diseño donde se especificarán con mayor nivel de detalle tanto las operaciones como los atributos asociados a cada una de ellas.

En el caso de uso **CU-01 Seleccionar pieza**, el actor Usuario registrado solicita, a través de la interfaz de tablero, seleccionar una pieza. El control de la partida se encargará de acceder a la clase entidad de la pieza, e informar si la pieza fue seleccionada, o si hubo un error.

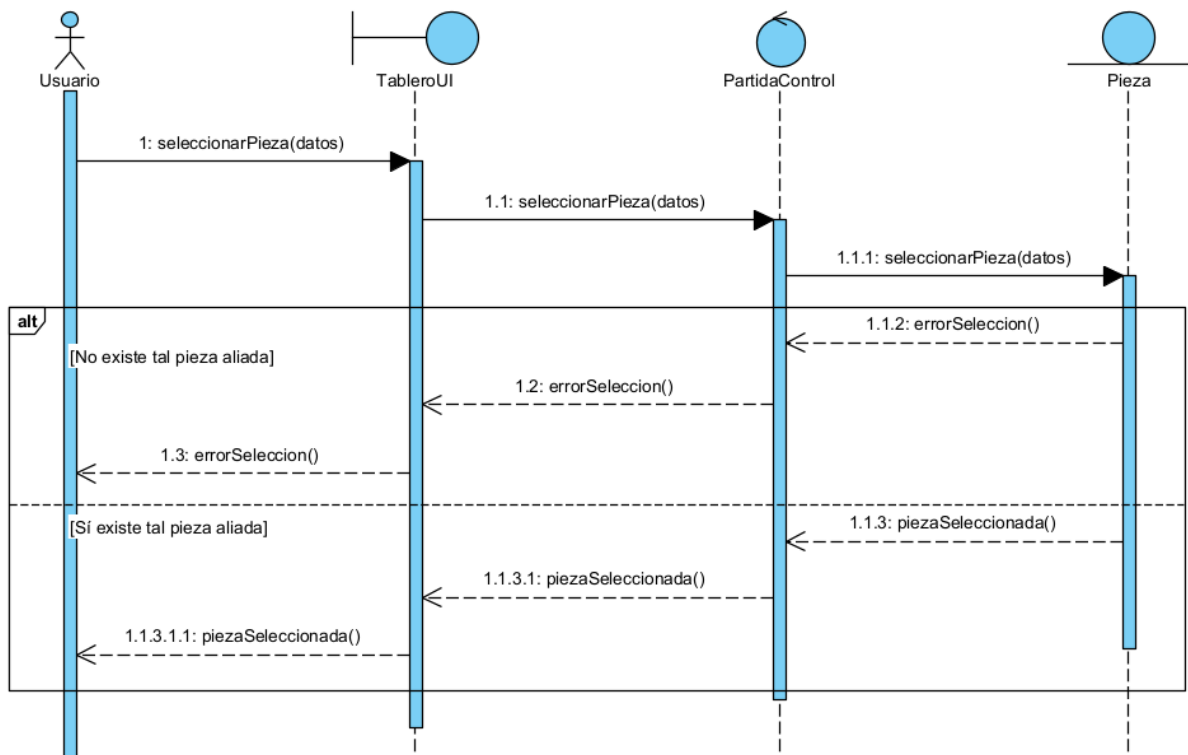


Ilustración 3: Diagrama de secuencia CU-01 Seleccionar pieza

En el caso de uso **CU-02 Mostrar movimientos legales para una pieza**, el actor Sistema solicita, a través de la interfaz de tablero, mostrar los movimientos legales para una pieza. El control de la partida se encargará de acceder a la clase entidad del tablero, e informar al actor Usuario no registrado de los movimientos legales que hay para esa pieza.

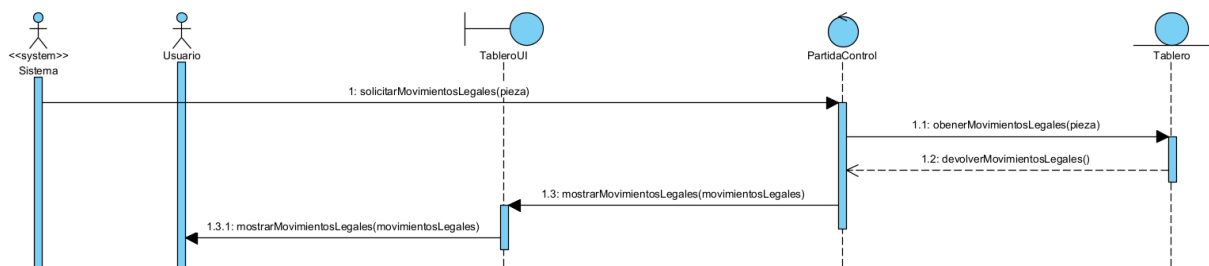


Ilustración 4: Diagrama de secuencia CU-02 Mostrar movimientos legales para una pieza

En el caso de uso **CU-03 Deseleccionar pieza**, el actor Usuario registrado solicita, a través de la interfaz de tablero, deseleccionar una pieza. El control de la partida se encargará de acceder a la clase entidad de la pieza, e informar que la pieza fue deseleccionada.

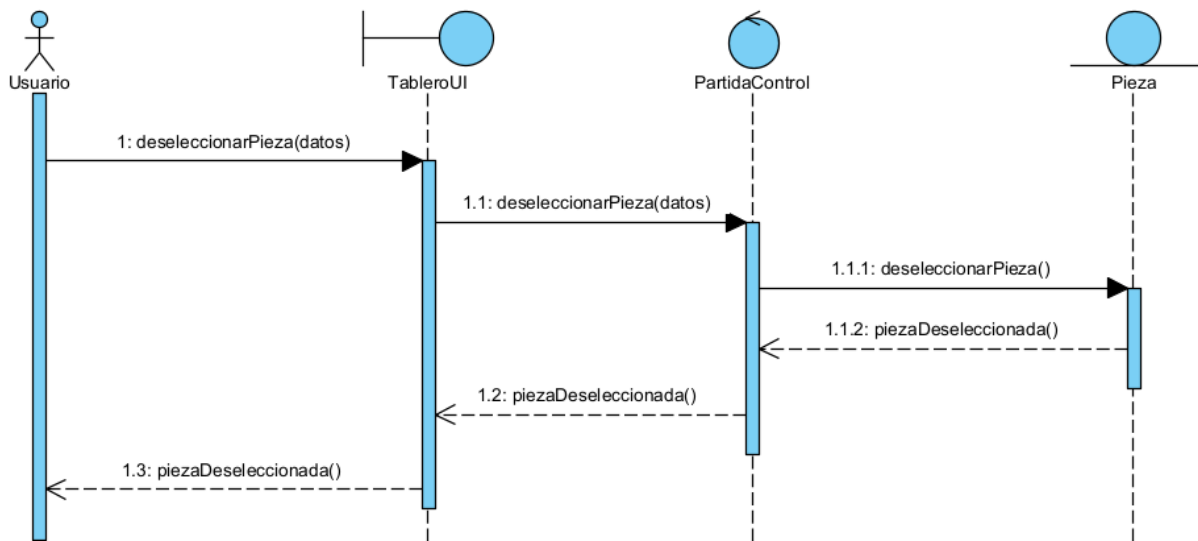


Ilustración 5: Diagrama de secuencia CU-03 Deseleccionar pieza

En el caso de uso **CU-04 Mover pieza**, el actor Usuario registrado solicita, a través de la interfaz de tablero, mover una pieza. El control de la partida se encargará de acceder a la clase entidad del tablero, e informar si el movimiento se pudo realizar, o si es un movimiento ilegal.

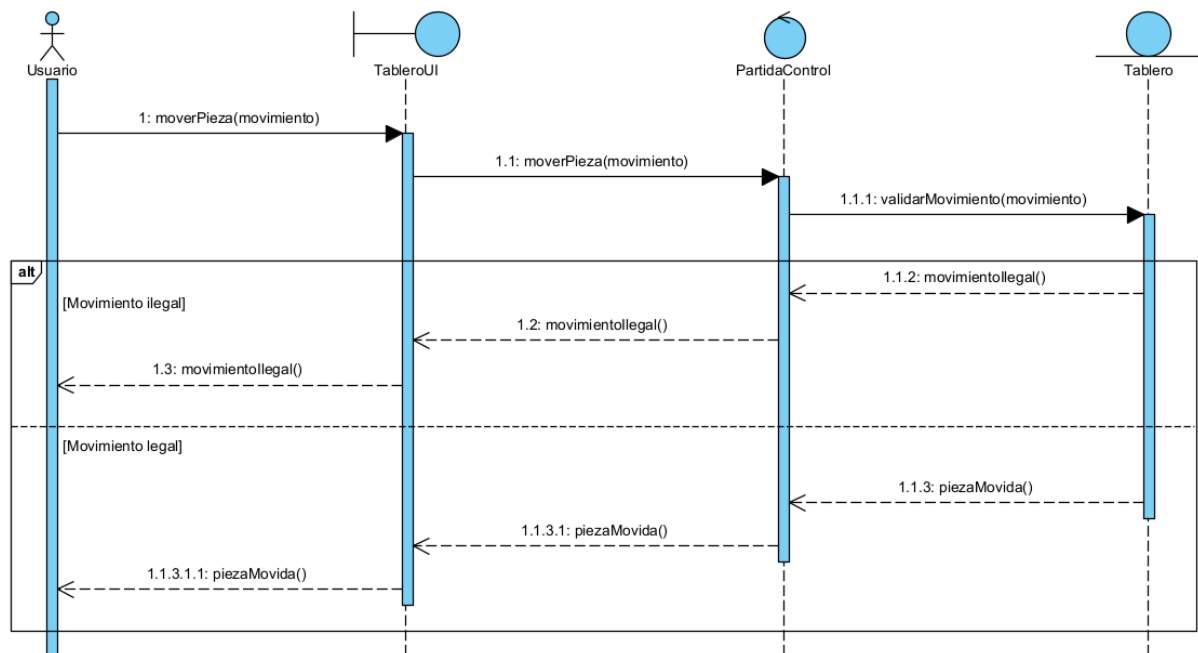


Ilustración 6: Diagrama de secuencia CU-04 Mover pieza

En el caso de uso **CU-05 Detectar jaque**, el actor Sistema solicita detectar el jaque. El control de la partida se encargará de acceder a la clase entidad del tablero, e informar al actor Usuario registrado, a través de la interfaz de partida, si no hay jaque, o restringir los movimientos pseudolegales si sí lo hubo.

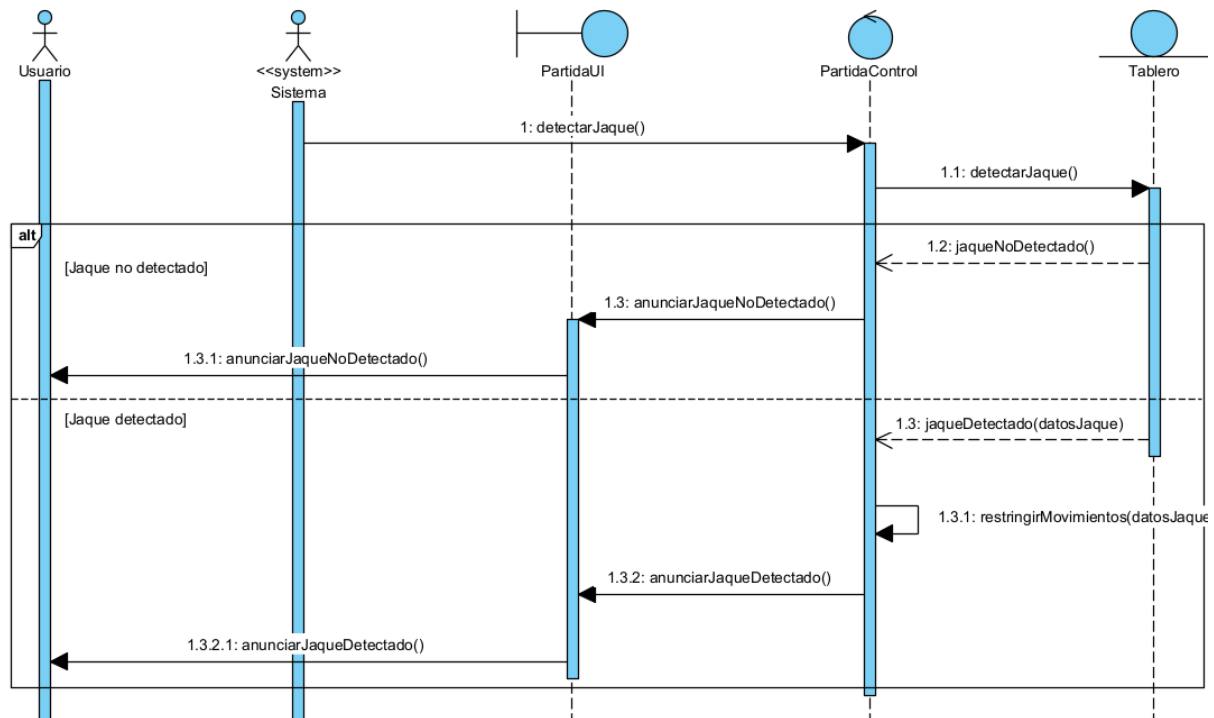


Ilustración 7: Diagrama de secuencia CU-05 Detectar jaque

En el caso de uso **CU-06 Detectar jaque mate**, el actor Sistema solicita detectar el jaque mate. El control de la partida se encargará de acceder a la clase entidad del tablero, e informar al actor Usuario registrado si no hay mate jaque, a través de la interfaz de partida, si sí lo hubo.

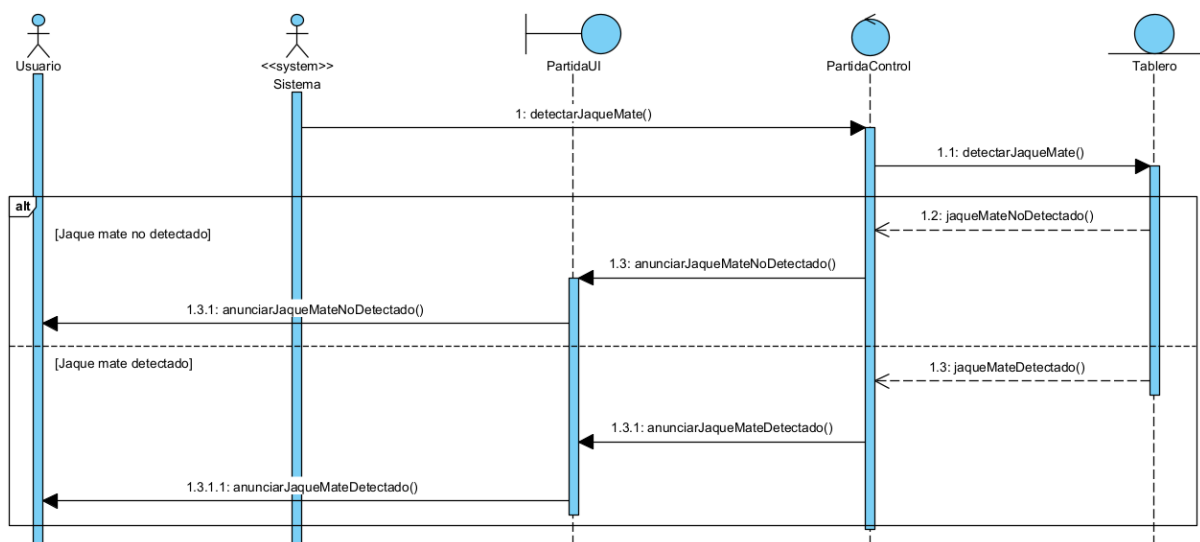


Ilustración 8: Diagrama de secuencia CU-06 Detectar jaque mate

En el caso de uso **CU-07 Detectar tablas**, el actor Sistema solicita detectar las tablas. El control de la partida se encargará de acceder a la clase entidad del tablero, e informar al actor Usuario registrado si no hay tablas, a través de la interfaz de partida, si sí las hubo.

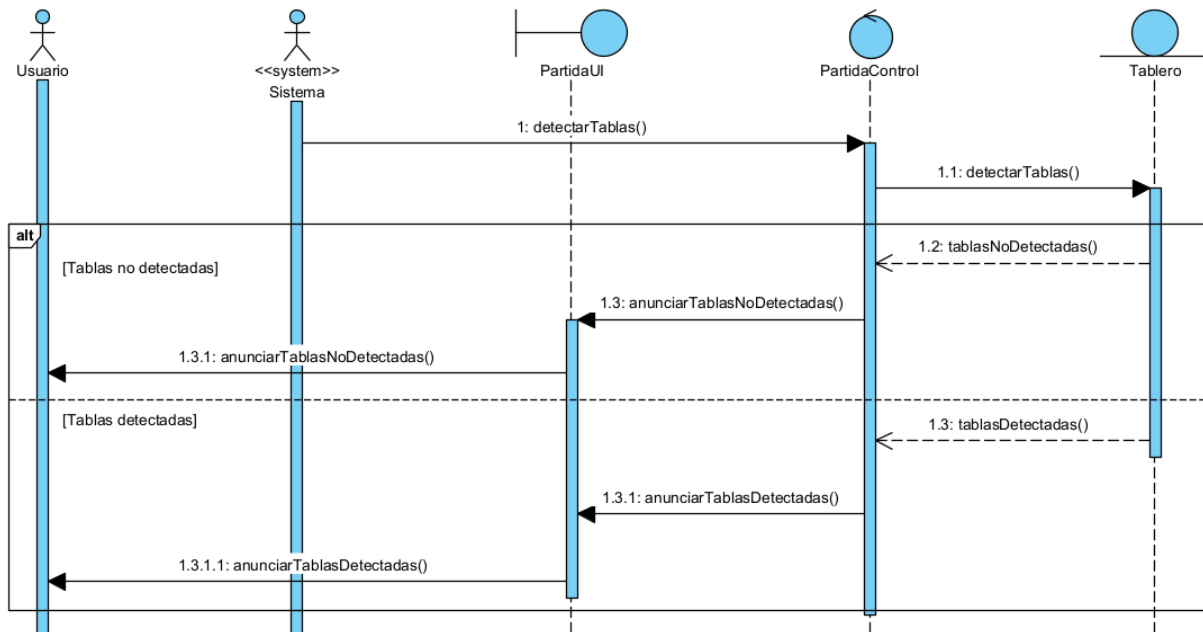


Ilustración 9: Diagrama de secuencia CU-07 Detectar tablas

En el caso de uso **CU-08 Finalizar partida**, el actor Sistema solicita finalizar la partida. El control de la partida se encargará de acceder a la clase entidad de la partida, e informar al actor Usuario registrado del resultado de la partida, a través de la interfaz de partida.

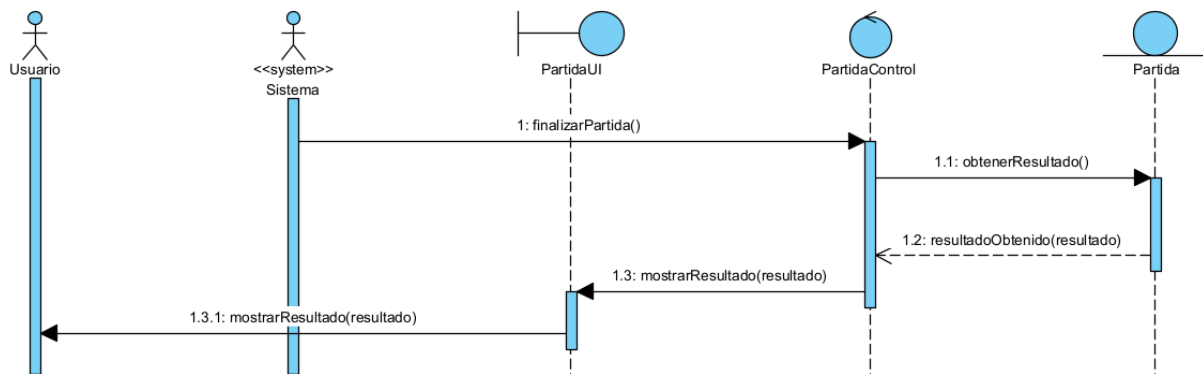


Ilustración 10: Diagrama de secuencia CU-08 Finalizar partida

En el caso de uso **CU-09 Cambiar nombre jugador**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, cambiar el nombre de un jugador, para después introducir el nuevo nombre. El control de las configuraciones previas se encargará de acceder a la clase entidad del jugador, e informar si el nombre fue cambiado exitosamente, o si no es un nombre válido.

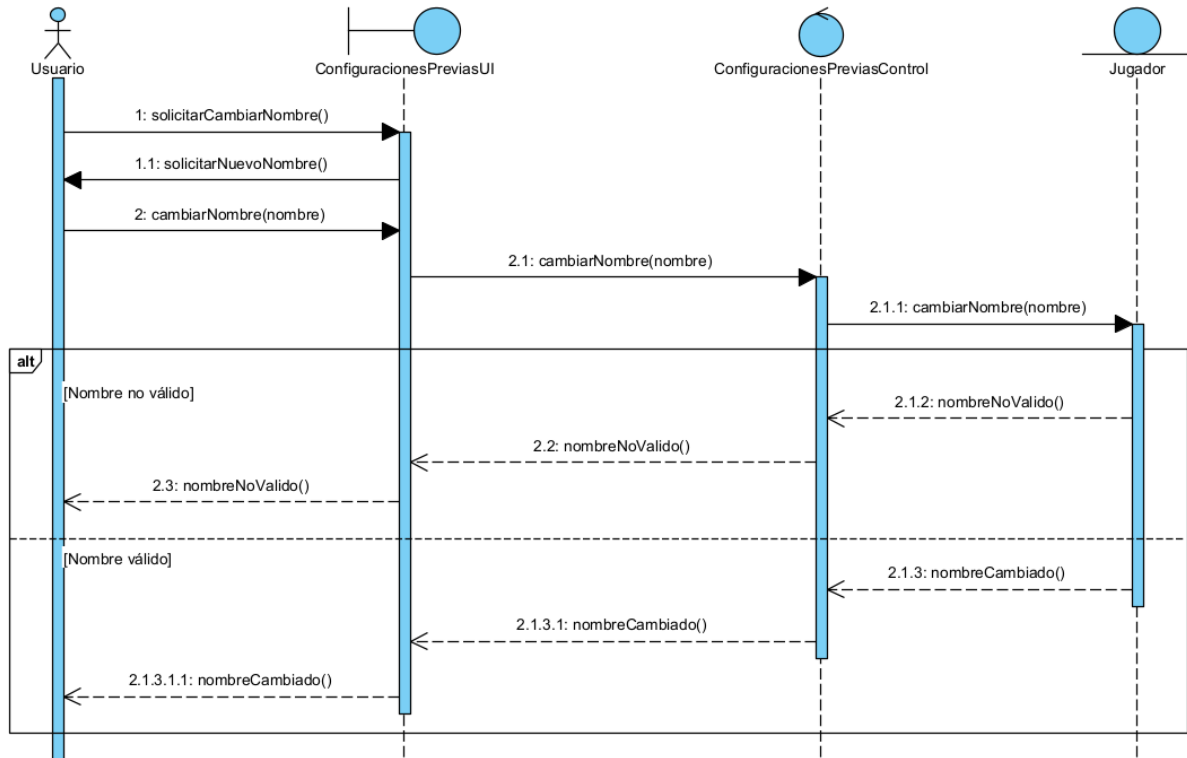


Ilustración 11: Diagrama de secuencia CU-09 Cambiar nombre jugador

En el caso de uso **CU-10 Cambiar color de las piezas de los jugadores**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, cambiar el color de las piezas de los jugadores. El control de las configuraciones previas se encargará de acceder a la clase entidad del jugador, e informar cuando el cambio se haya realizado.

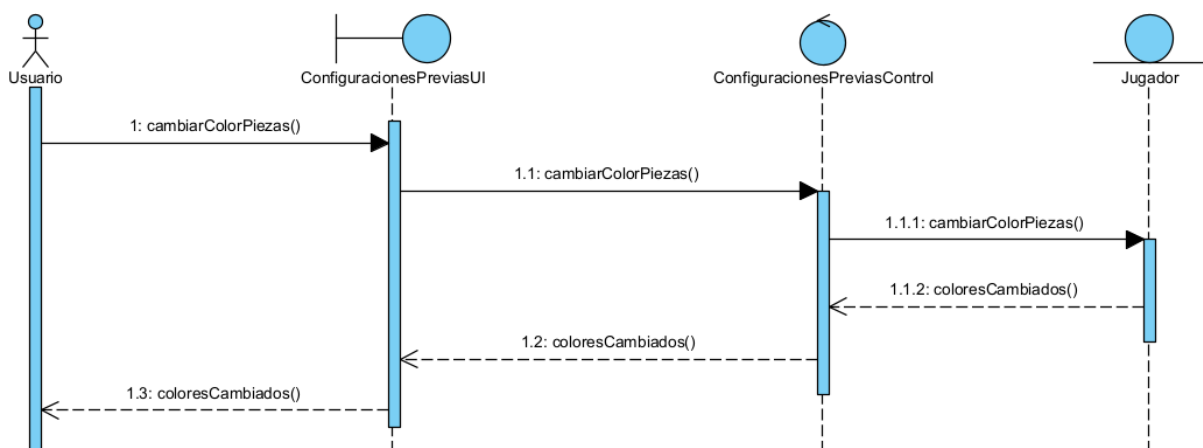


Ilustración 12: Diagrama de secuencia CU-10 Cambiar color de las piezas de los jugadores

En el caso de uso **CU-11 Elegir modo de juego contra otro jugador**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, cambiar al modo de juego contra otro jugador. El control de las configuraciones previas se encargará de acceder a la clase entidad de la partida, e informar cuando el cambio se haya realizado.

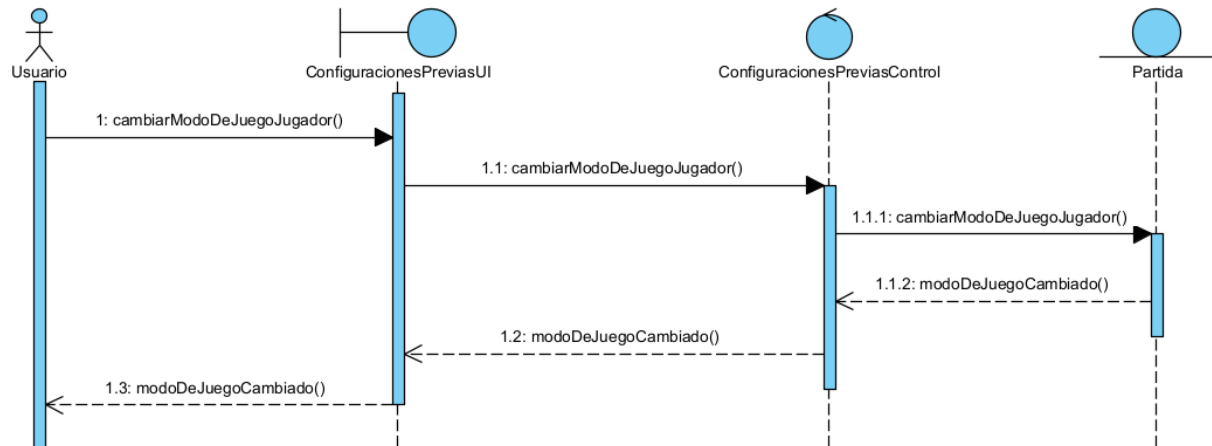


Ilustración 13: Diagrama de secuencia CU-11 Elegir modo de juego contra otro jugador

En el caso de uso **CU-12 Elegir modo de juego contra la máquina**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, cambiar al modo de juego contra la máquina. El control de las configuraciones previas se encargará de acceder a la clase entidad de la partida, e informar cuando el cambio se haya realizado.

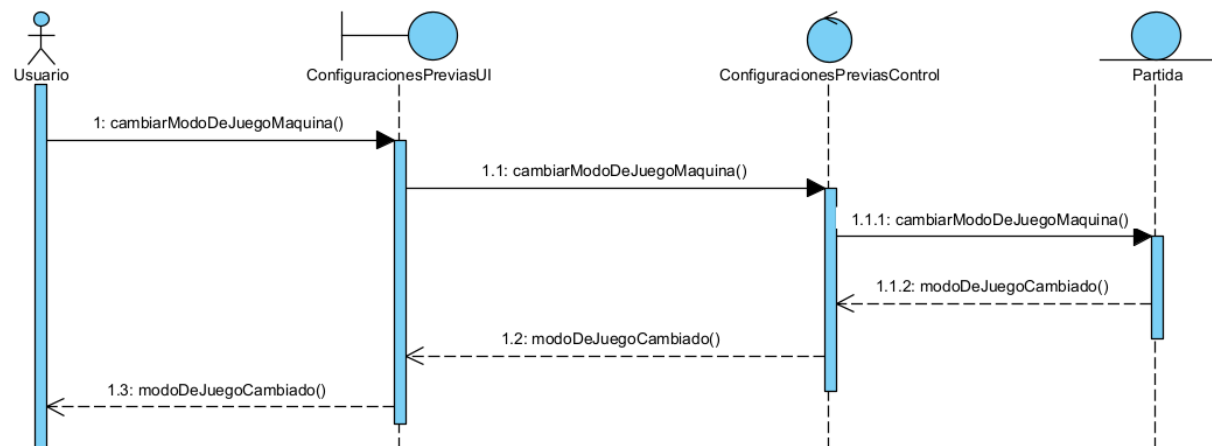


Ilustración 14: Diagrama de secuencia CU-12 Elegir modo de juego contra la máquina

En el caso de uso **CU-13 Empezar partida**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, empezar la partida. El control de las configuraciones previas se encargará de acceder a la clase entidad de la partida, e informar si la partida ha empezado, o si las configuraciones previas no son válidas.

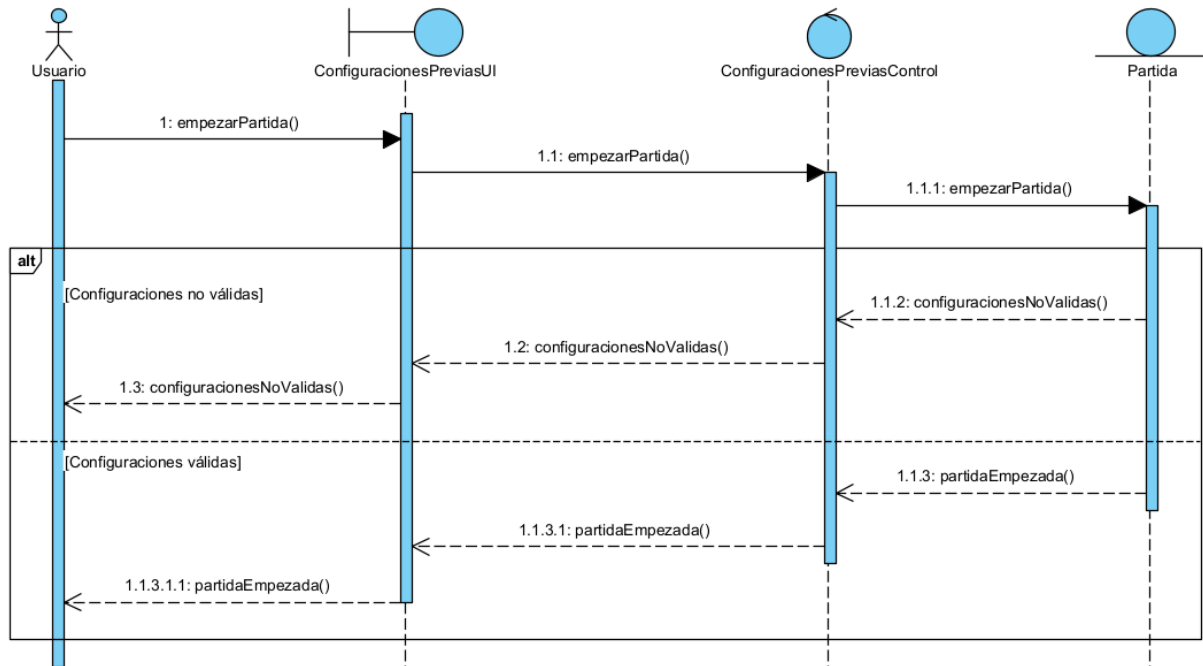


Ilustración 15: Diagrama de secuencia CU-13 Empezar partida

En el caso de uso **CU-14 Buscar movimientos para la IA**, el actor Sistema solicita buscar movimientos para la IA. El control de búsqueda se encargará de acceder a la clase entidad del tablero y devolver los movimientos legales para la IA.

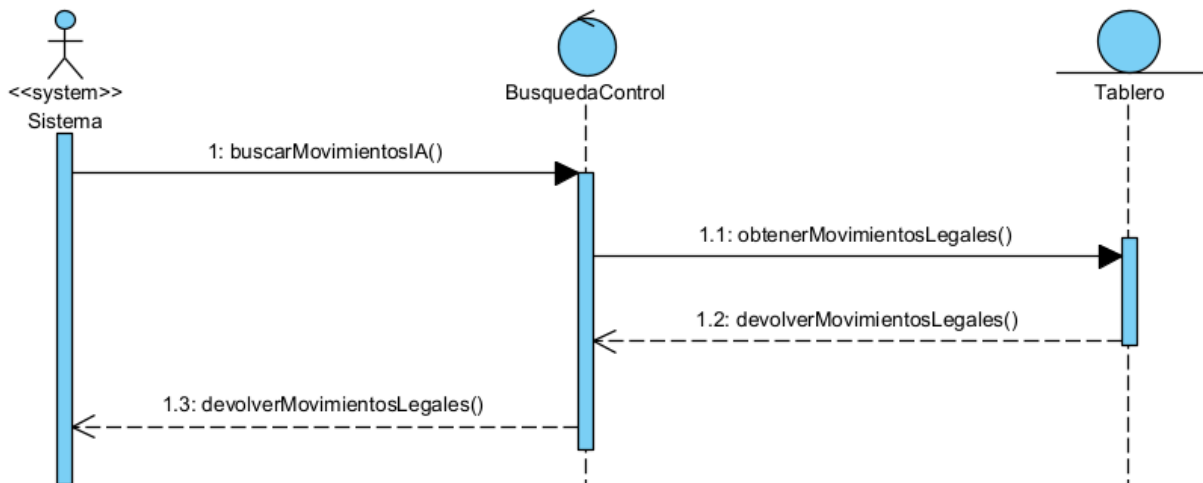


Ilustración 16: Diagrama de secuencia CU-14 Buscar movimientos para la IA

En el caso de uso **CU-15 Evaluar movimientos**, el actor Sistema solicita evaluar los movimientos de la IA. El control de evaluación se encargará de acceder a la clase entidad del tablero, aplicar cada movimiento en el tablero, evaluar la posición y deshacerlo, para finalmente devolver el movimiento con mejor evaluación.

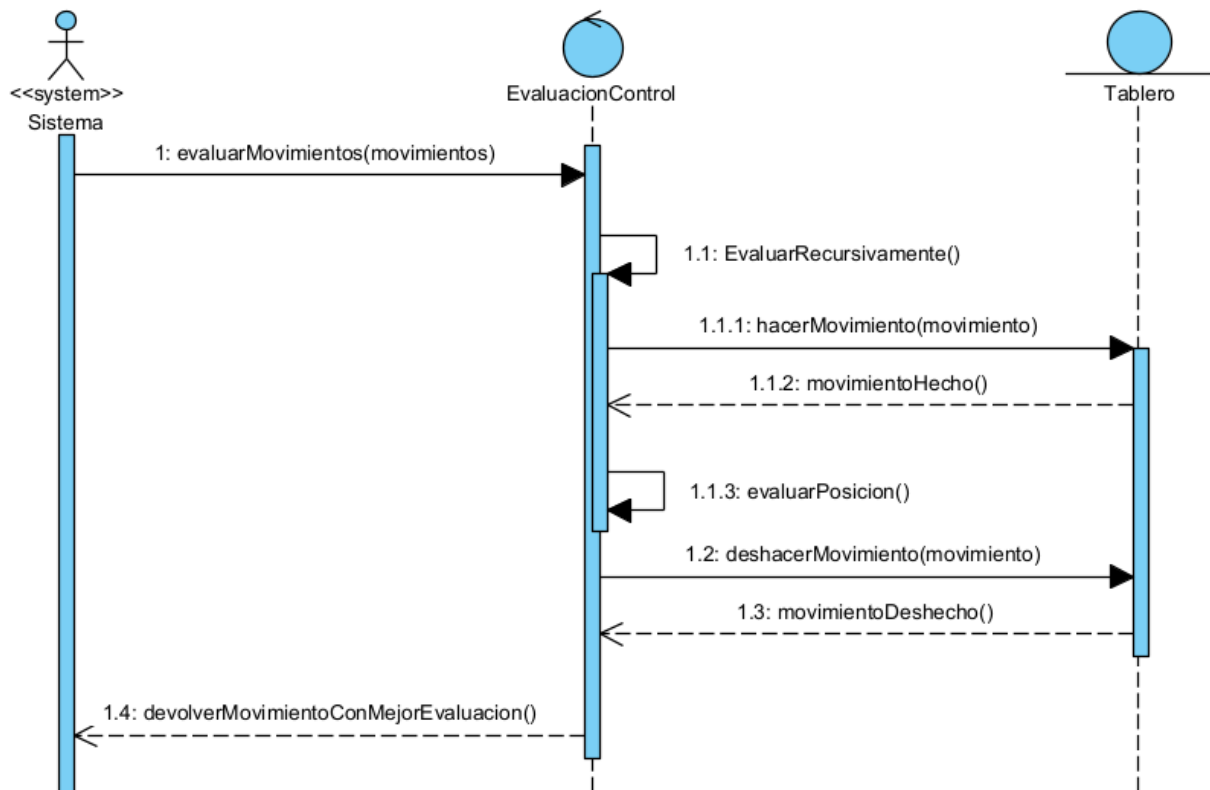


Ilustración 17: Diagrama de secuencia CU-15 Evaluar movimientos

En el caso de uso **CU-16 Decrementar tiempo restante a los jugadores**, el actor Sistema solicita decrementar el tiempo restante al jugador correspondiente. El control de la partida se encargará de hallar el jugador actual y decrementar su tiempo, si este se ha agotado, comunicará el fin de la partida al usuario mediante la interfaz de partida, de lo contrario, mostrará el nuevo tiempo restante al jugador.

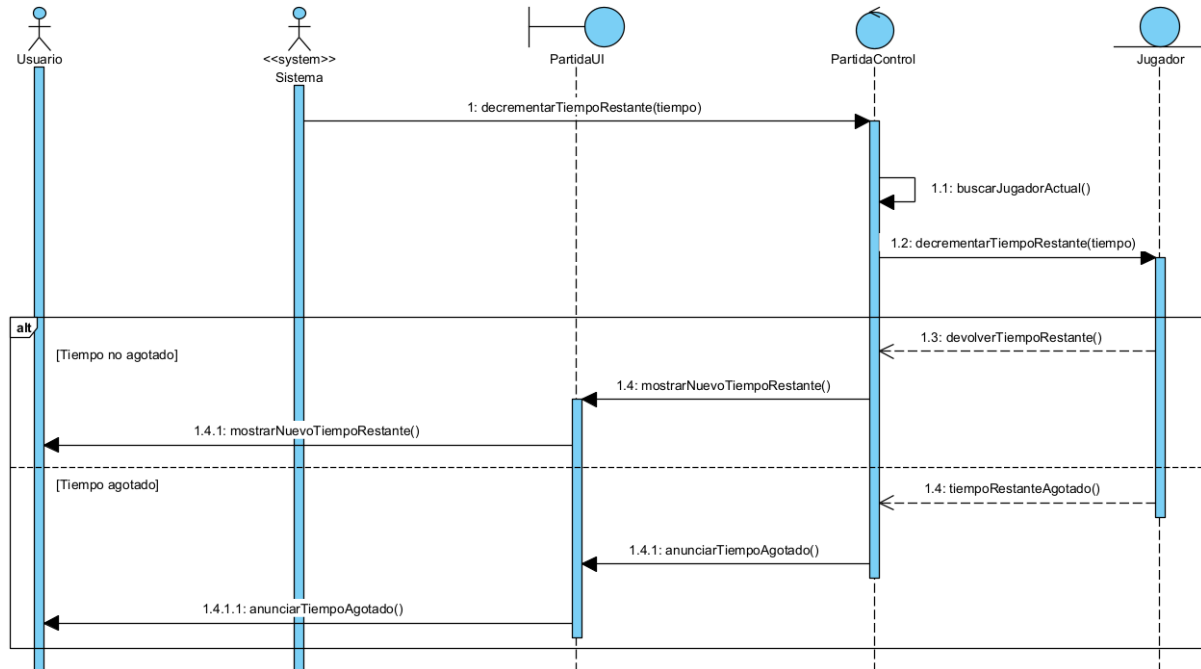


Ilustración 18: Diagrama de secuencia CU-16 Decrementar tiempo restante a los jugadores

En el caso de uso **CU-17 Elegir dificultad de la IA**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, elegir la dificultad de la IA, para después seleccionar la dificultad de la IA. El control de las configuraciones previas se encargará de acceder a la clase entidad del jugadorIA, e informar cuando la dificultad se haya cambiado.

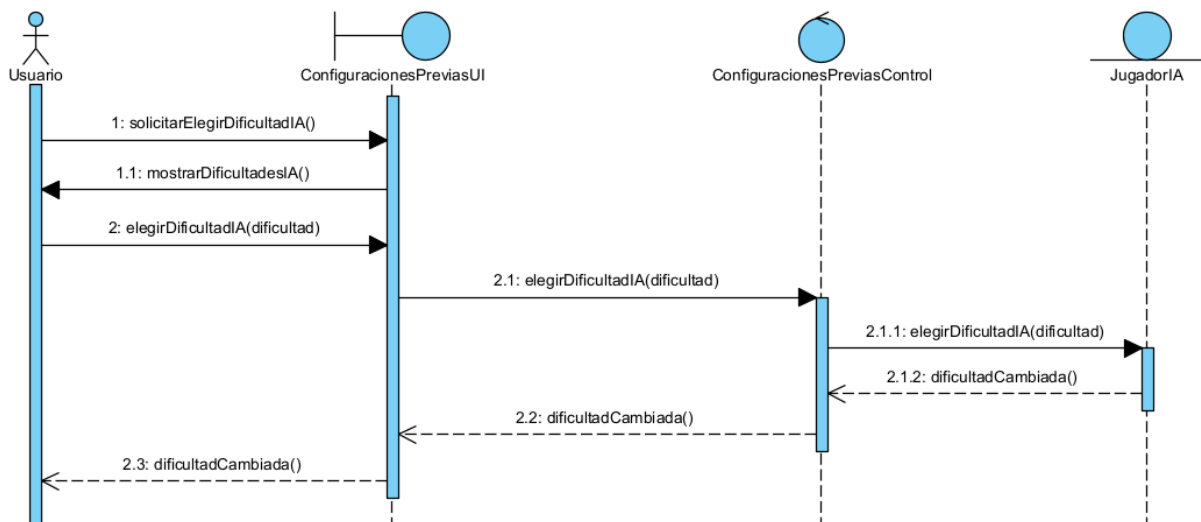


Ilustración 19: Diagrama de secuencia CU-17 Elegir dificultad de la IA

En el caso de uso **CU-18 Cargar distribución del tablero en formato FEN**, el actor Usuario registrado solicita, a través de la interfaz de configuraciones previas, cargar la distribución del tablero en formato FEN. El control de las configuraciones previas se encargará de acceder a la clase entidad del tablero, e informar si la distribución fue cargada, o si no es una distribución válida.

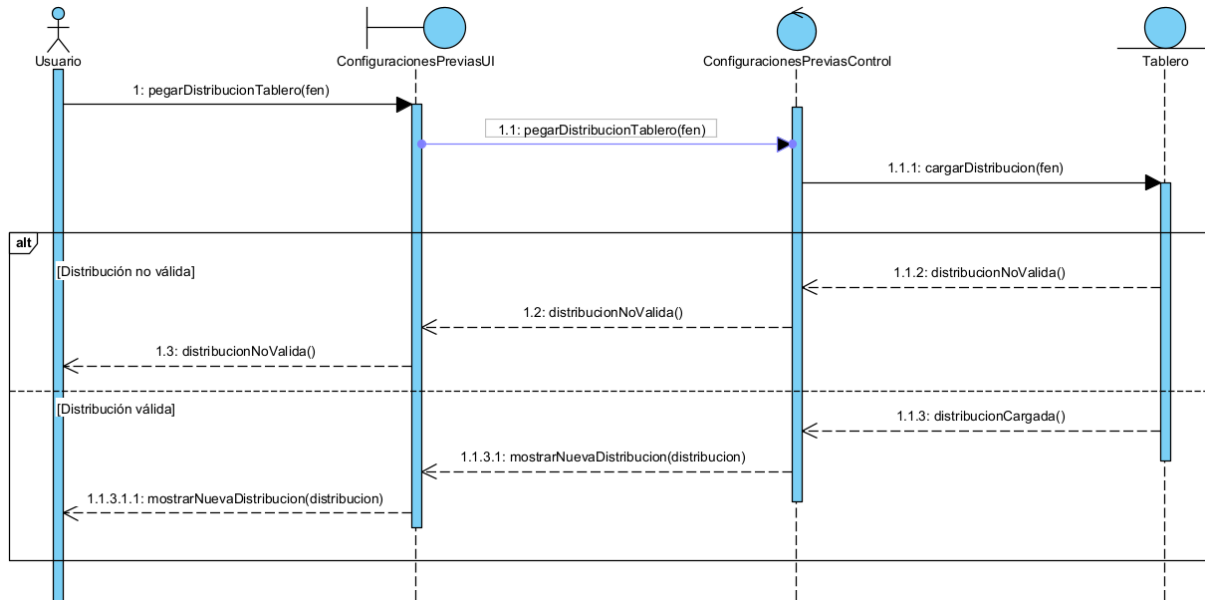


Ilustración 20: Diagrama de secuencia CU-18 Cargar distribución del tablero en formato FEN

En el caso de uso **CU-19 Calcular tiempo de búsqueda**, el actor Sistema solicita calcular el tiempo de búsqueda que empleará la IA. El control del tiempo de búsqueda se encargará de acceder a la clase entidad del jugadorIA y devolver su tiempo restante, en base al cual, calculará dicho tiempo de búsqueda.

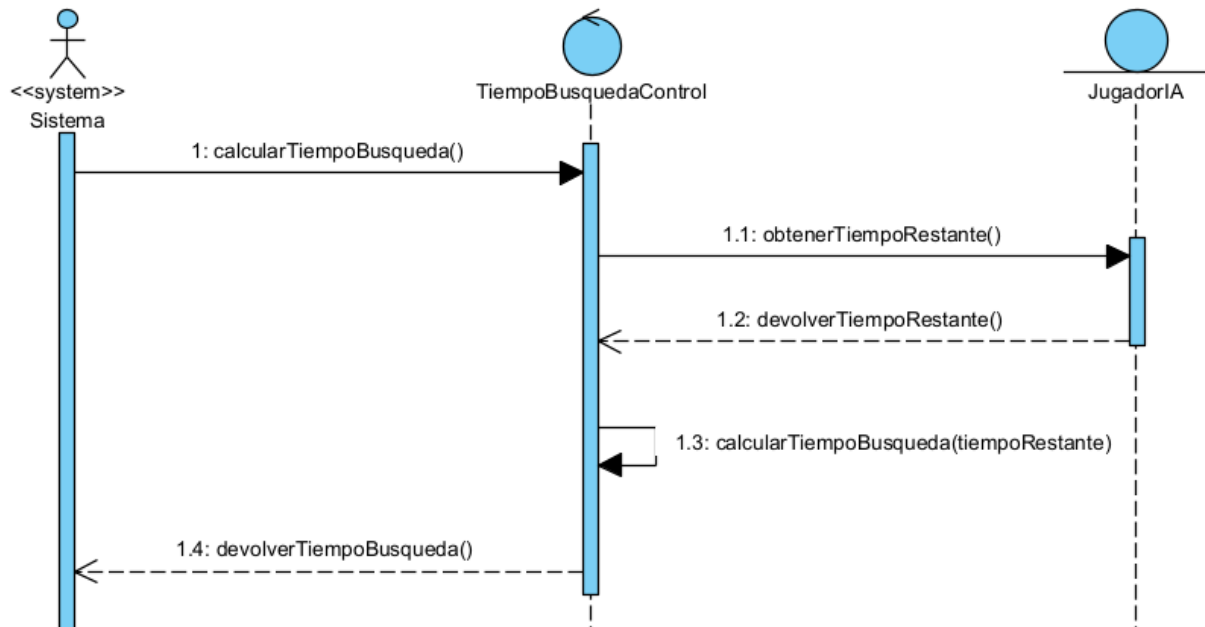


Ilustración 21: Diagrama de secuencia CU-19 Calcular tiempo de búsqueda

3.3.Paquetes de análisis

Al comienzo de este anexo se definió una arquitectura inicial basada en la funcionalidad, organizada en paquetes funcionales o de servicio. A partir de esta estructura inicial, el modelo ha ido evolucionando, por un lado, mediante la identificación de las clases entidad en el modelo de dominio y, por otro, a través del desarrollo de la realización de los casos de uso a nivel de análisis, lo que ha permitido refinar la funcionalidad del sistema.

Una vez completadas estas actividades, el siguiente paso en el análisis consiste en identificar los paquetes de análisis. Estos paquetes se derivan de los paquetes funcionales definidos previamente, pero no estarán vacíos, sino que contendrán las clases de análisis identificadas hasta el momento, así como las relaciones existentes entre ellas organizadas dentro de los paquetes correspondientes.

Al igual que ocurre con otros tipos de paquetes, los paquetes de análisis pueden incluir subpaquetes y establecer relaciones con otros paquetes de análisis dentro de la estructura general del sistema.

Empezando por el paquete de **Gestión de configuraciones previas**, nos encontramos con varias clases entidad, que se acceden desde el control de las configuraciones:

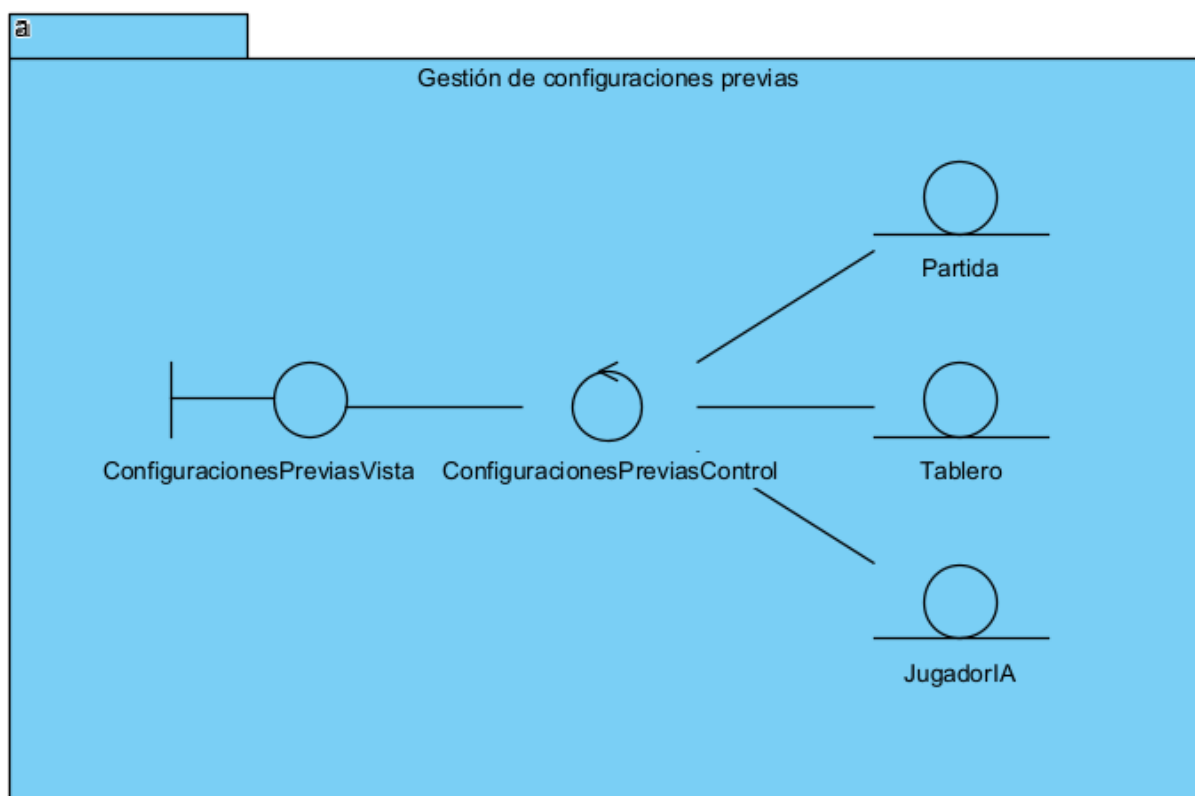


Ilustración 22: Paquete de análisis Gestión de configuraciones previas

El paquete **Gestión de partida** contiene la interfaz del tablero y la de la partida, las cuales tienen propósitos distintos. Además, un único control coordina el acceso a las clases entidad:

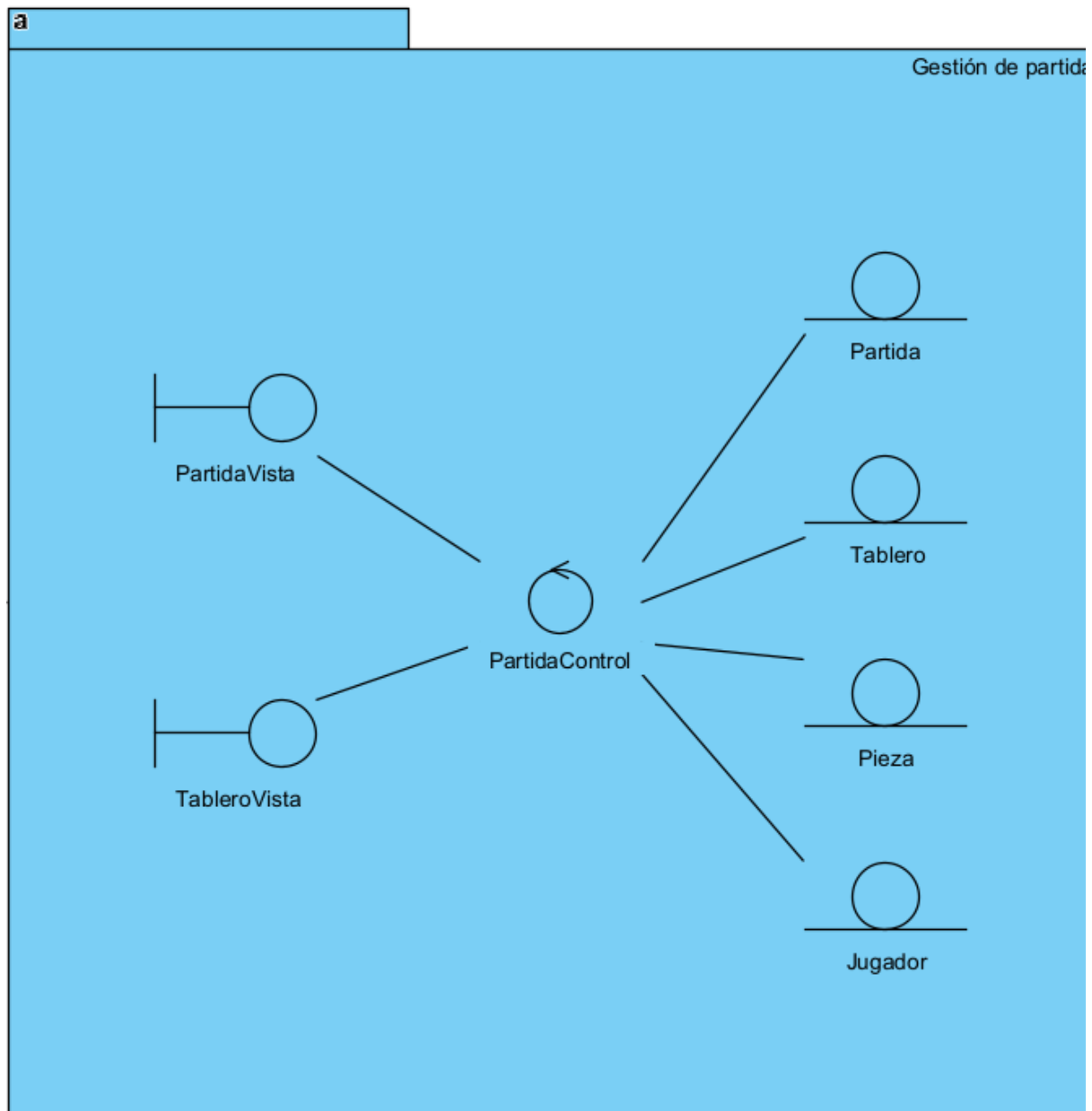


Ilustración 23: Paquete de análisis Gestión de partidas

En el paquete **Gestión de inteligencia artificial**, tenemos dos clases control que acceden a la clase entidad del tablero:

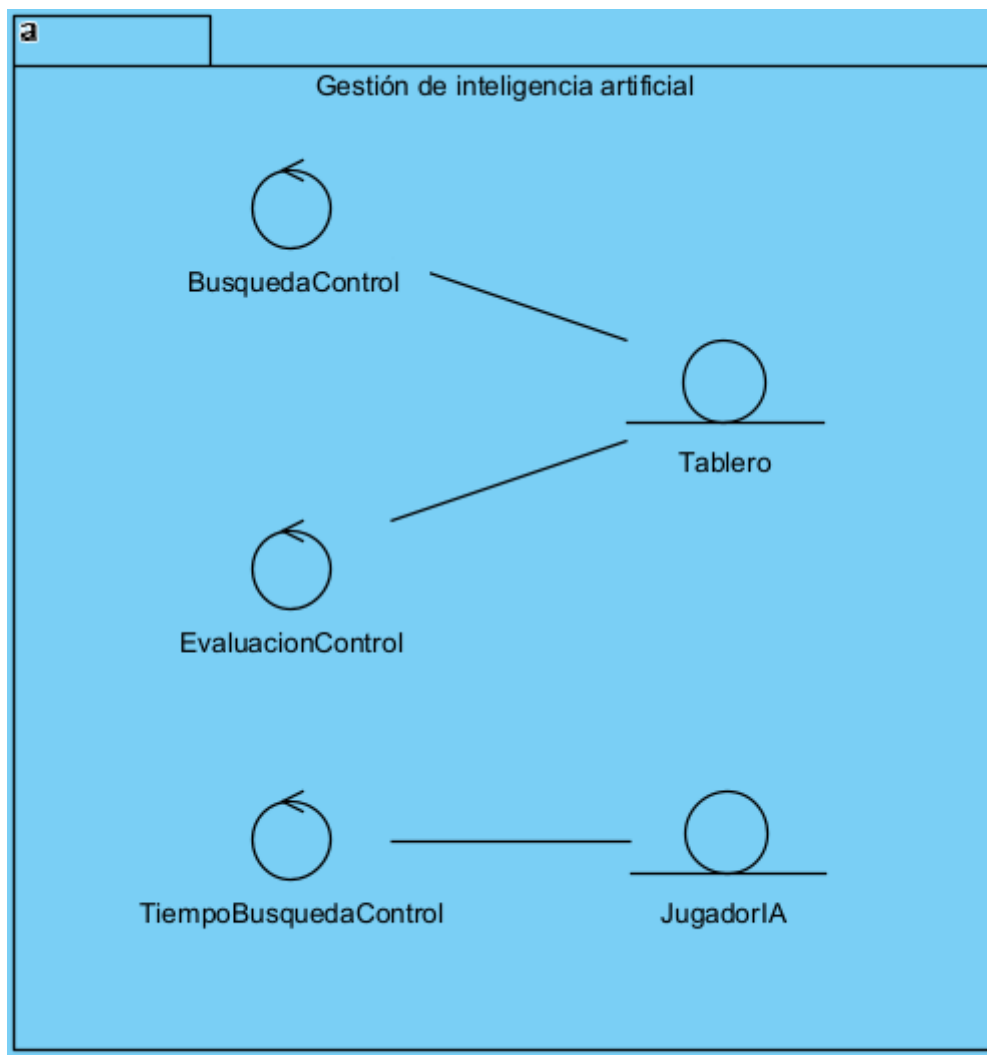


Ilustración 24: Paquete de análisis Gestión de inteligencia artificial

3.4.Propuesta de arquitectura

Para finalizar la disciplina del análisis, es necesario plantear una propuesta de arquitectura que tome como base la arquitectura inicial definida al comienzo de esta fase. Dicha propuesta debe incluir los paquetes funcionales o de servicio establecidos en esa arquitectura preliminar, y dentro de estos paquetes se integrarán tanto las clases entidad identificadas en el modelo de dominio como las clases de interfaz y de control obtenidas durante la realización de los casos de uso en el análisis.

De este modo, los paquetes funcionales de la propuesta de arquitectura no estarán vacíos, sino que contendrán las clases identificadas a lo largo de la disciplina del análisis. Asimismo, será preciso definir las relaciones entre estas clases, de acuerdo con las asociaciones establecidas en la realización de los casos de uso a nivel de análisis. Esto es lo que hemos denominado paquetes de análisis, y lo que se ha descrito en apartados anteriores.

La propuesta de arquitectura del sistema a construir incluye todas las clases identificadas en el análisis y sus relaciones entre los distintos paquetes y subpaquetes del análisis. El resultado sería el siguiente:

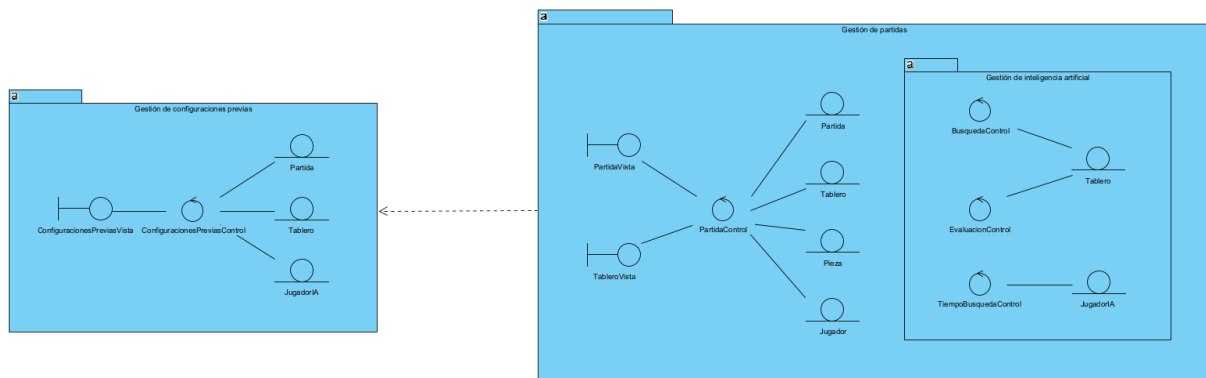


Ilustración 25: Propuesta de arquitectura

4. Modelo de diseño

El modelo de diseño proporciona una visión detallada de cómo se estructurará el sistema y cómo interactuarán sus componentes, garantizando que se cumplan los objetivos funcionales y no funcionales del proyecto, como la mantenibilidad, escalabilidad y eficiencia.

A lo largo de este capítulo se presentan los distintos aspectos que componen el modelo de diseño. En primer lugar, se detalla el **patrón arquitectónico** seleccionado, justificando su elección en función de las necesidades y características del sistema. A continuación, se expone la **arquitectura de sistemas**, donde se definen los distintos módulos y su interacción. Posteriormente, se describen las **clases de diseño**, que especifican los elementos principales a nivel de código y su relación entre sí. Finalmente, se realiza la **realización de casos de uso a nivel de diseño**, mostrando cómo se implementan los principales casos de uso mediante los elementos definidos en el modelo de diseño.

4.1. Patrón arquitectónico

El patrón arquitectónico empleado para este sistema será la **arquitectura basada en capas** (Cherif, 2024), que se trata un modelo de diseño en el que el sistema se organiza en un conjunto de capas o niveles, donde cada capa tiene una responsabilidad específica y se comunica únicamente con las capas adyacentes (normalmente la capa inmediatamente superior o inferior). Cada capa tiene un rol claramente definido, oculta su implementación interna a otras capas y se encarga de interactuar solo con la capa que está justo debajo o encima. Un sistema con arquitectura en capas suele tener:

- **Capa de presentación (o interfaz de usuario):** Se encarga de la interacción con el usuario. Muestra datos y recoge entradas.
- **Capa de lógica de negocio (o dominio):** Contiene las reglas del negocio o la lógica central de la aplicación. Procesa datos y coordina el flujo entre presentación y datos.
- **Capa de acceso a datos (o persistencia):** Se encarga de comunicarse con la base de datos u otros sistemas de almacenamiento. Traduce las operaciones de la lógica de negocio a consultas sobre los datos.
- **Capa de datos (o base de datos):** Almacena la información en un formato persistente.

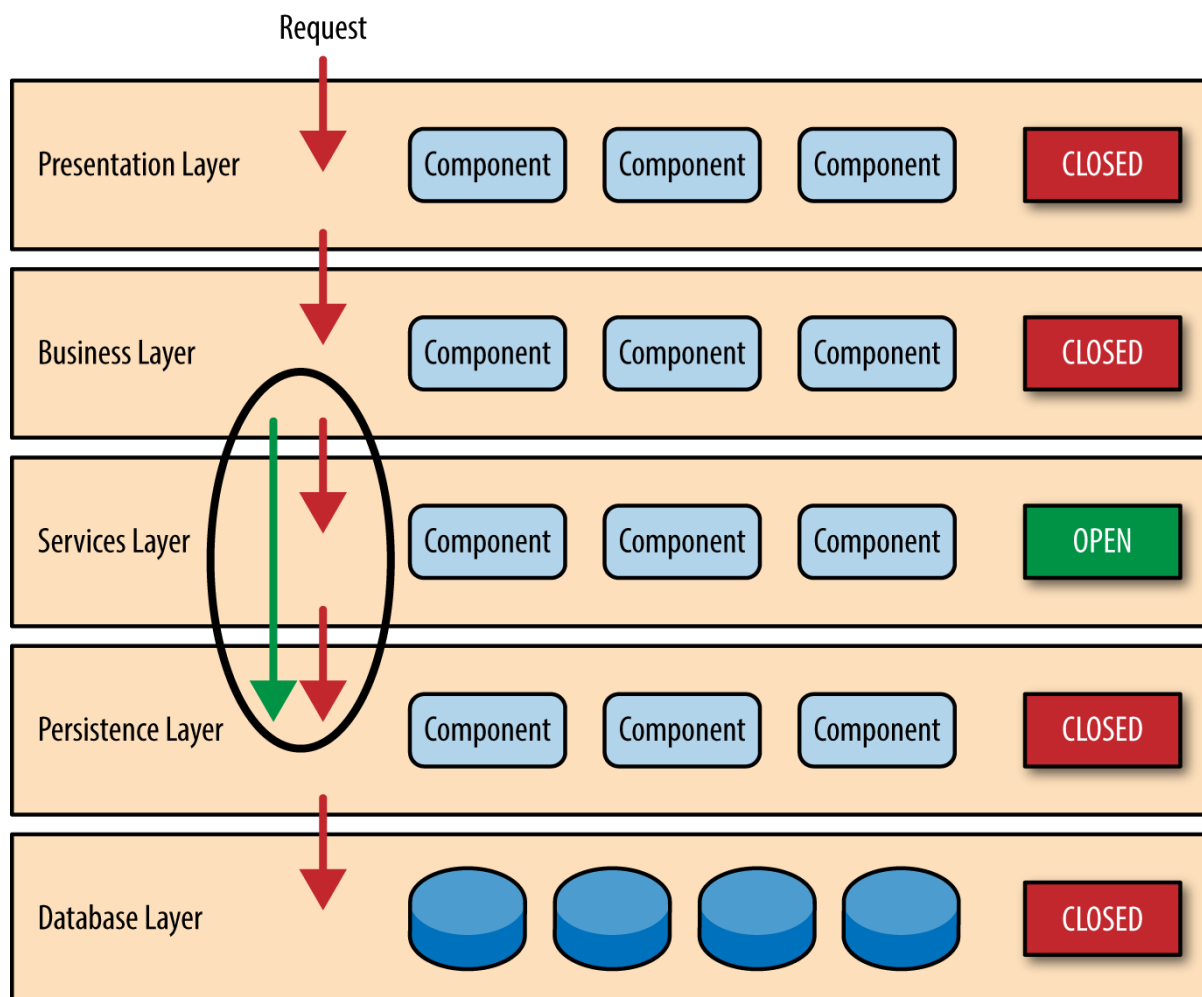


Ilustración 26: Diagrama de la arquitectura basada en capas

En algunos casos se pueden añadir más capas o agrupar algunas, y estas pueden ser abiertas o cerradas, en función de si se pueden sobre pasar o no. La elección de este patrón arquitectónico para este sistema, frente a otros más comunes como el modelo-vista-controlador o el modelo-vista-modelo de vista, se debe al hecho de abstraer el conjunto de código destinado a la inteligencia artificial, y agruparlo en su propia capa.

4.2.Arquitectura de subsistemas

Las capas que se han identificado para este proyecto han sido cuatro, y aunque podrían haberse separado algunas capas en otras más pequeñas, estas han sido las que finalmente se han establecido:

- **Capa de presentación:** Es la entrada y salida de todas las interacciones con el usuario, desde pulsaciones de teclas y movimiento del ratón, hasta actualización de la interfaz.
- **Capa de control:** Coordina y orquesta el funcionamiento general de las funcionalidades de la aplicación.
- **Capa de inteligencia artificial:** Esta capa alberga todo lo relativo al funcionamiento de la inteligencia artificial, aunque podría haberse dividido en una capa de búsqueda y

otra de evaluación. Se trata de una capa abierta, y los componentes de la capa de control no tienen por qué acceder siempre a esta capa, antes de llegar a la capa siguiente.

- **Capa de lógica:** En esta capa se agrupa todas las clases cuyo propósito sea la representación interna de la aplicación.

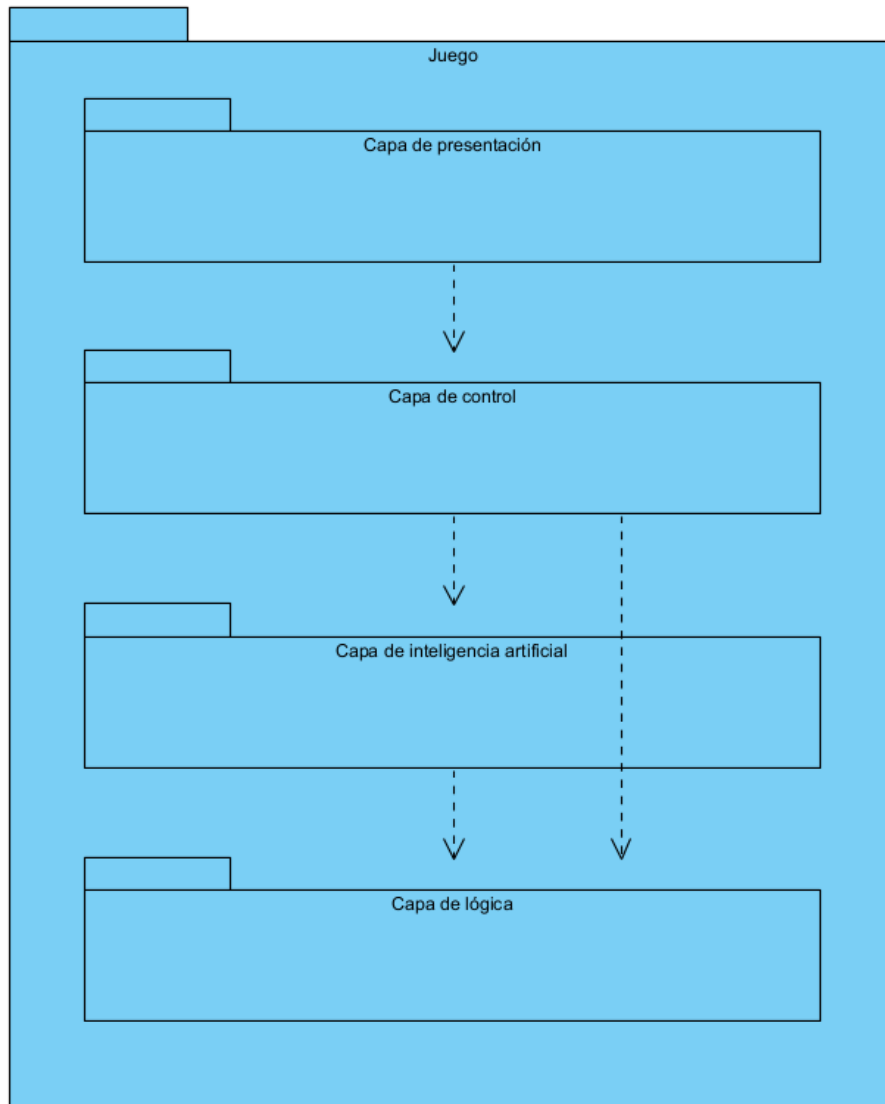


Ilustración 27: Vista arquitectónica

4.3. Clases de diseño

Las clases del diseño pertenecen al ámbito de la solución y están pensadas para la implementación en código. En estas clases se definen de forma completa todos los atributos y métodos, a diferencia de las clases del análisis donde solo se indicaban algunos de ellos. Además, se detalla la visibilidad de dichos atributos y métodos (por ejemplo, si son públicos o privados). También se establecen las interfaces y los nombres de los métodos, de acuerdo con el lenguaje de programación que se usará para construir el sistema.

A continuación, se mostrarán todas las clases que se utilizarán en la implementación final del sistema, y no solo las clases entidad como ocurría en el modelo de clases del análisis, además de indicar la capa a la que pertenece cada clase:

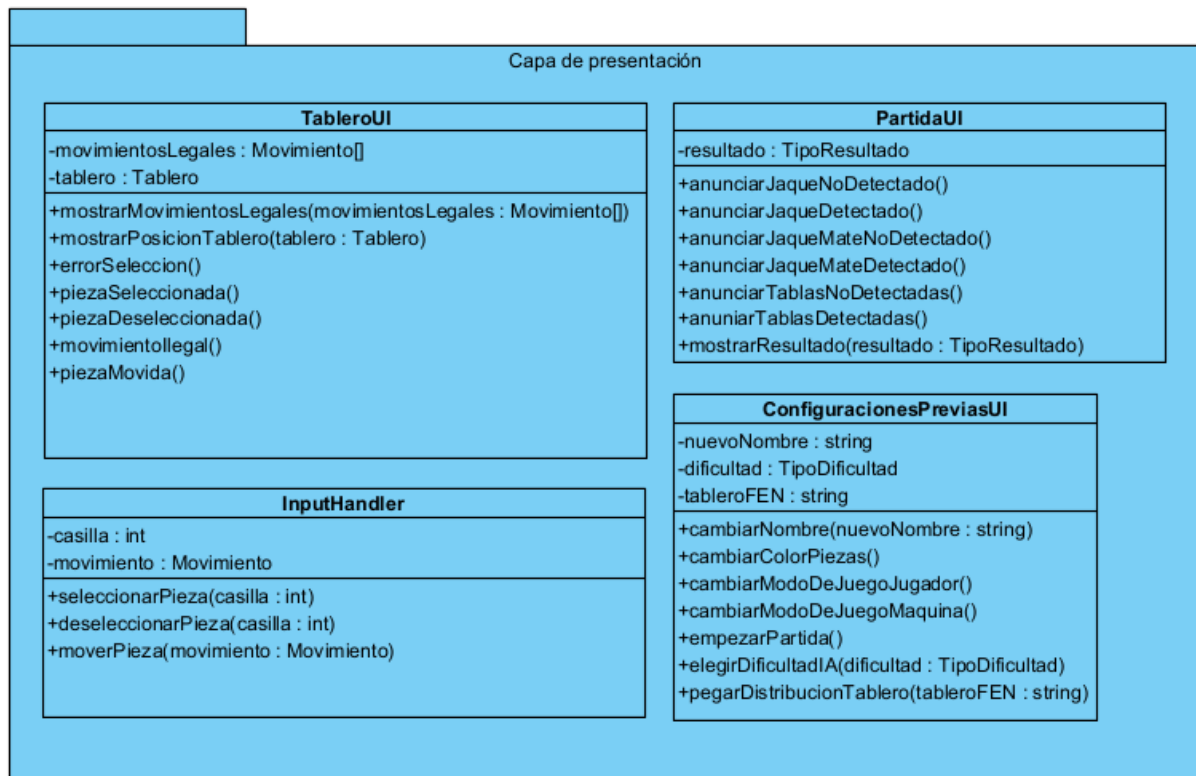


Ilustración 28: Subsistema de diseño Capa de presentación

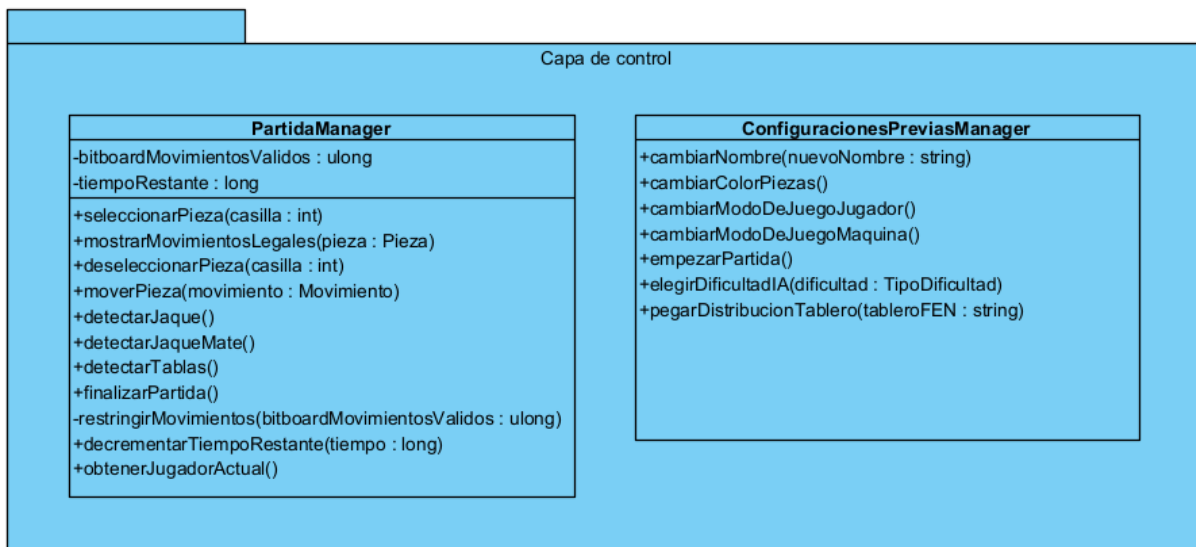


Ilustración 29: Subsistema de diseño Control

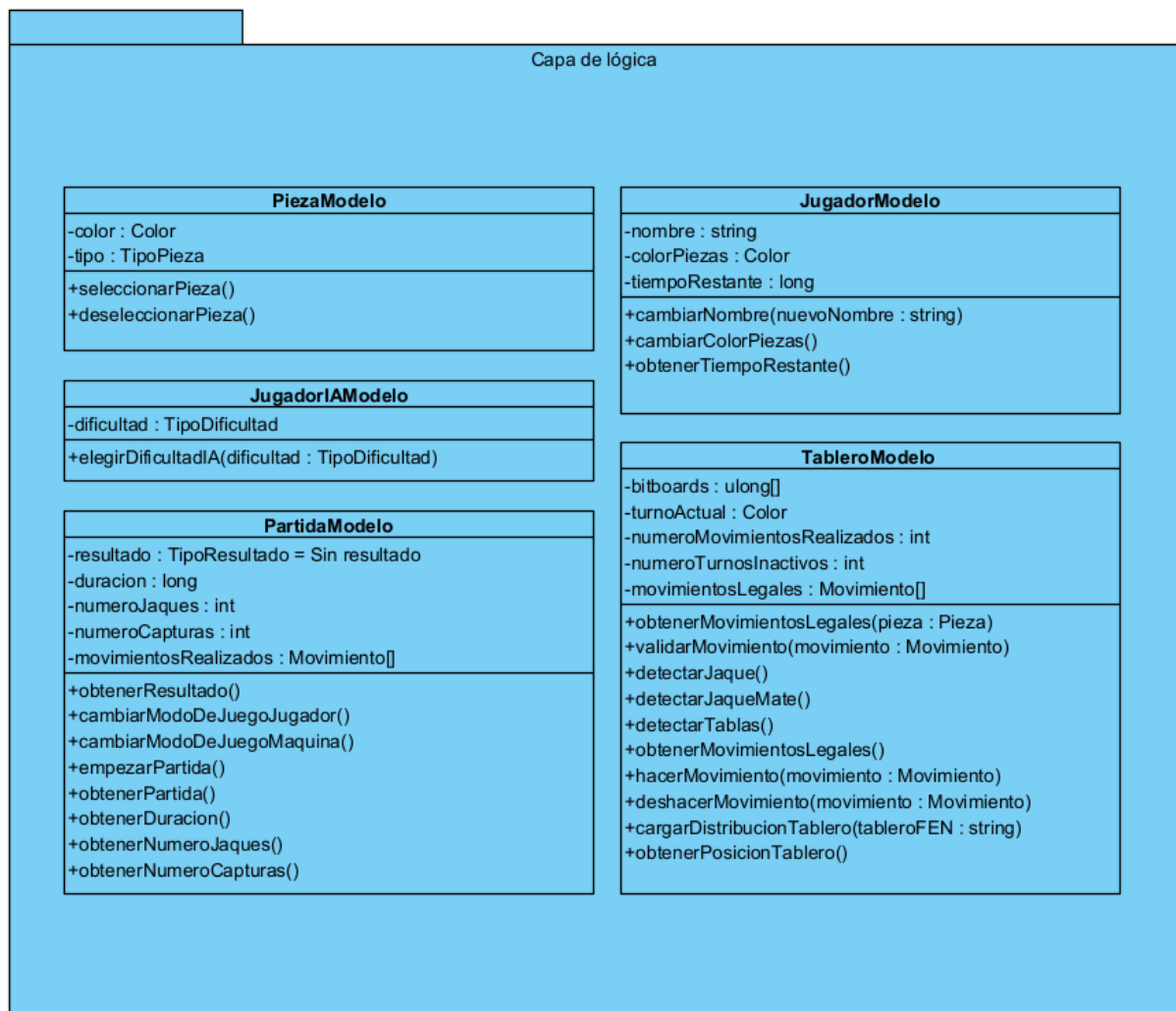


Ilustración 30: Subsistema de diseño Lógica

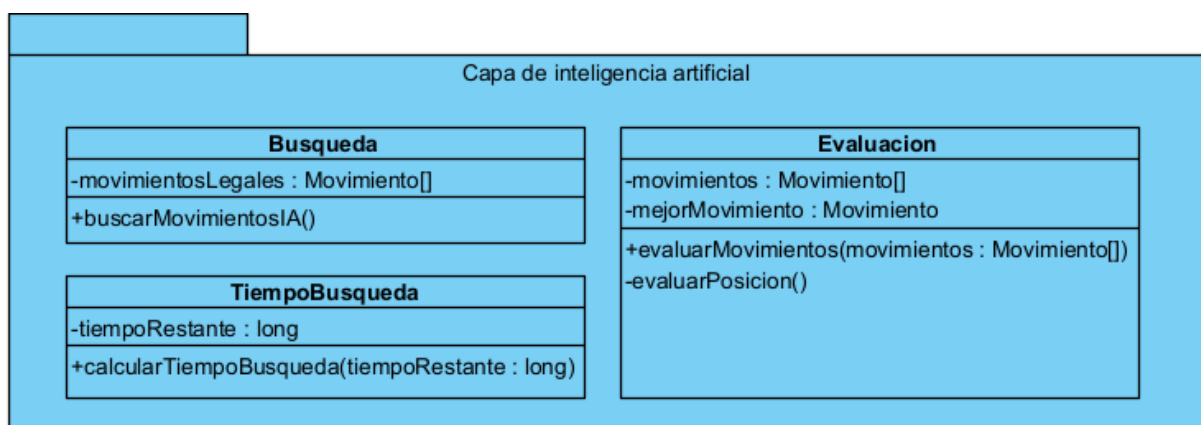


Ilustración 31: Subsistema de diseño Lógica

4.4.Realización de casos de uso a nivel de diseño

De forma similar al modelo de análisis, en el de diseño se encuentra la realización de casos de uso a nivel de diseño, que muestran con mayor detalle la secuencia de pasos necesarios para implementar una funcionalidad del sistema. En concreto, la realización de casos de uso a nivel de diseño se centra en el flujo de información que se produce entre las distintas clases del diseño.

Para representar la realización de los casos de uso en el nivel de diseño, UML ofrece distintas opciones:

- **Diagramas de secuencia a nivel de diseño:** Estos diagramas son similares a los usados en la etapa de análisis, pero en este caso se emplean las clases de diseño que se han definido en el modelo de diseño, así como los métodos y operaciones propios de esas clases. Esto permite describir el flujo de información entre las clases reales con mayor precisión y un nivel de detalle más bajo que el de los diagramas de análisis.
- **Diagramas de actividad:** Muestran el flujo de información según las actividades y acciones realizadas. Al igual que los diagramas de secuencia, sirven para reflejar el comportamiento dinámico del sistema.
- **Máquinas de estados:** Modelan el comportamiento dinámico mediante estados y transiciones.

Cualquiera de estas tres alternativas que ofrece UML es válida para documentar la realización de los casos de uso en el diseño. En este caso concreto, se optará por los **diagramas de secuencia a nivel de diseño**, ya que permiten mostrar de forma clara y detallada el flujo de información y las tareas necesarias para llevar a cabo una funcionalidad, con un nivel de especificación cercano al de la implementación.

A continuación, se mostrarán los diagramas de secuencia de los 19 casos de uso identificados en la disciplina de requisitos:

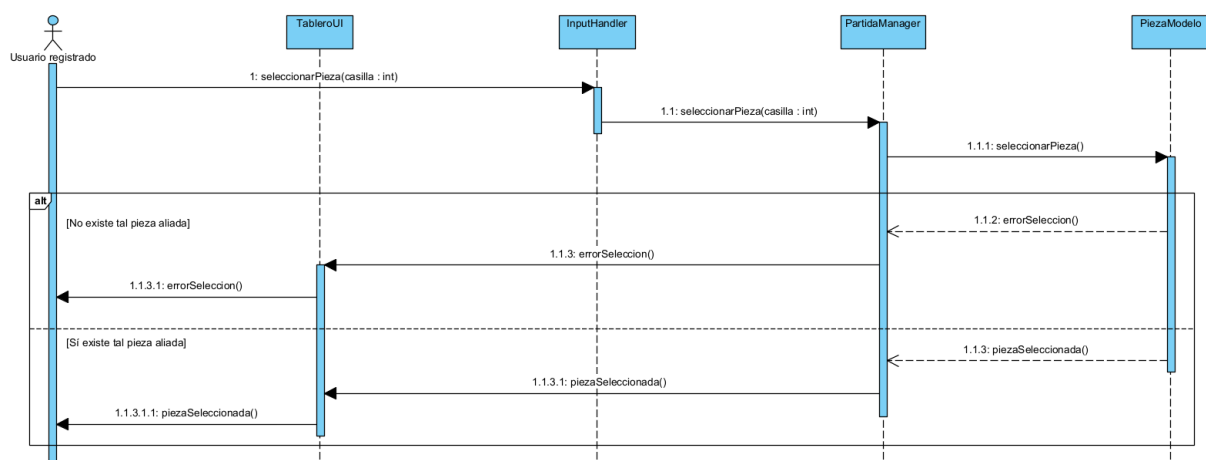


Ilustración 32: Diagrama de secuencia de diseño CU-01 Seleccionar pieza



Ilustración 33: Diagrama de secuencia de diseño CU-02 Mostrar movimientos legales para una pieza

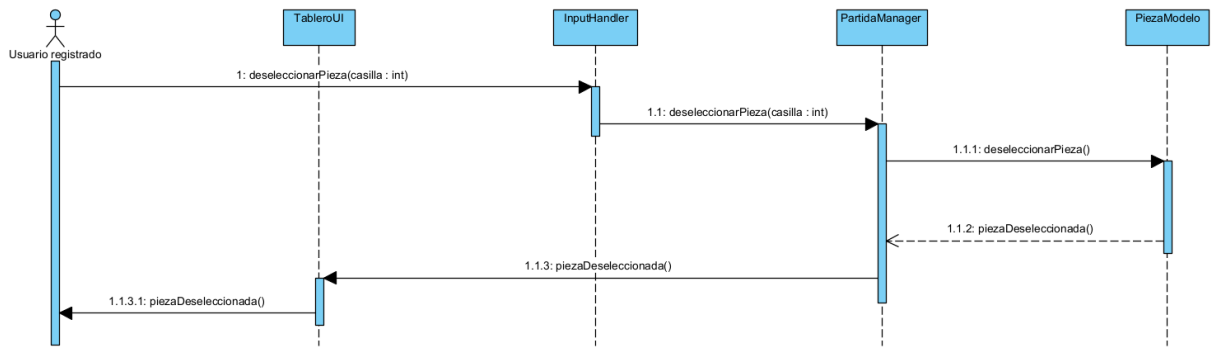


Ilustración 34: Diagrama de secuencia de diseño CU-03 Deseleccionar pieza

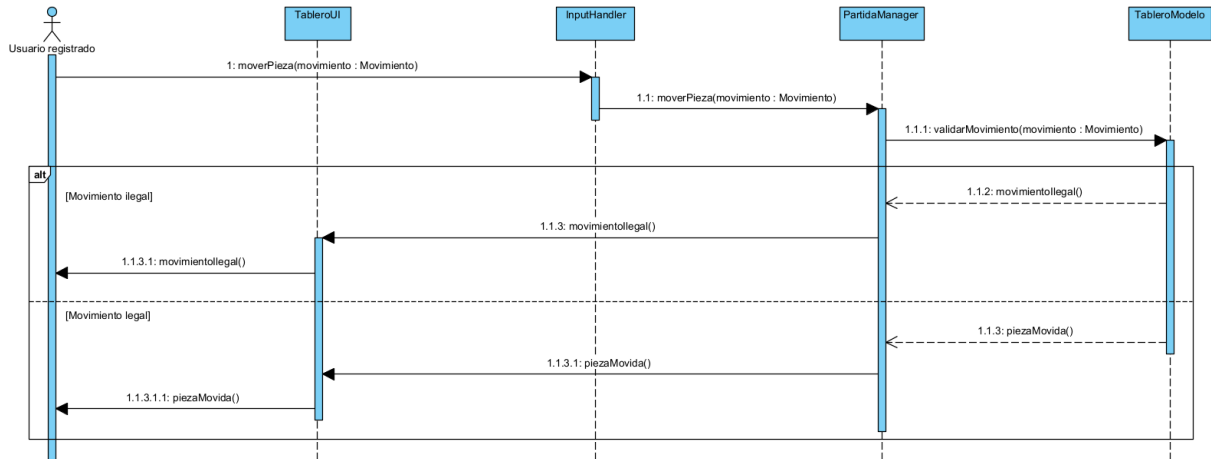


Ilustración 35: Diagrama de secuencia de diseño CU-04 Mover pieza

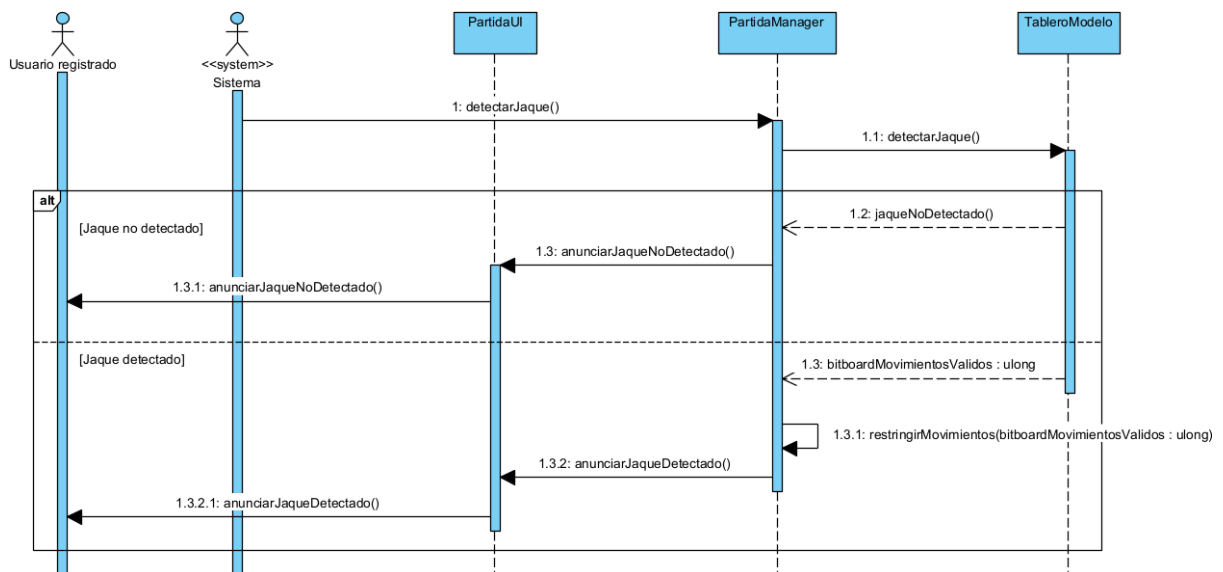


Ilustración 36: Diagrama de secuencia de diseño CU-05 Detectar jaque

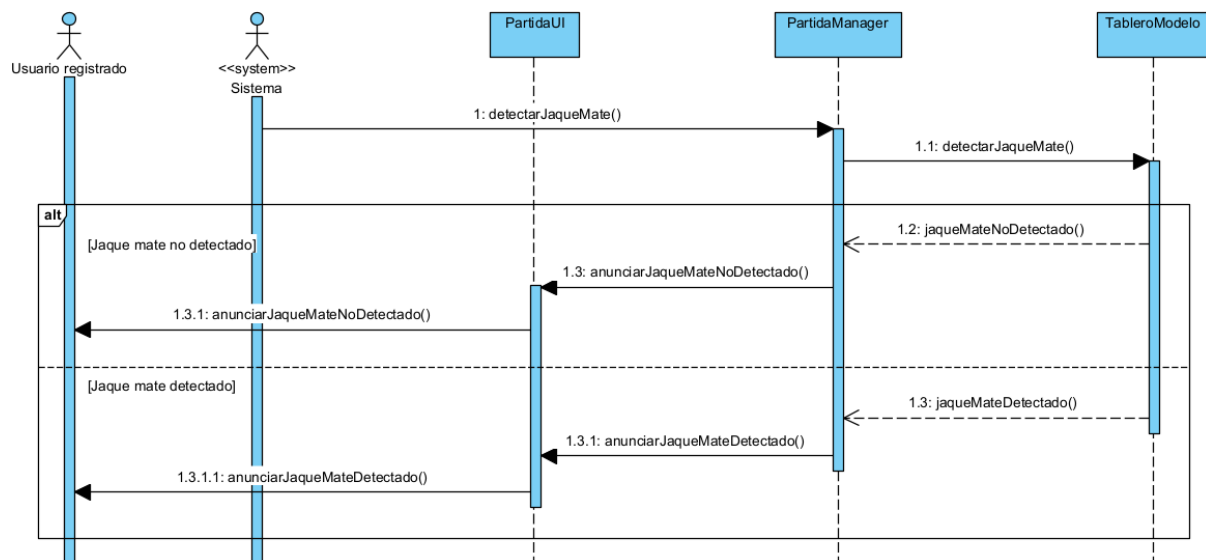


Ilustración 37: Diagrama de secuencia de diseño CU-06 Detectar jaque mate

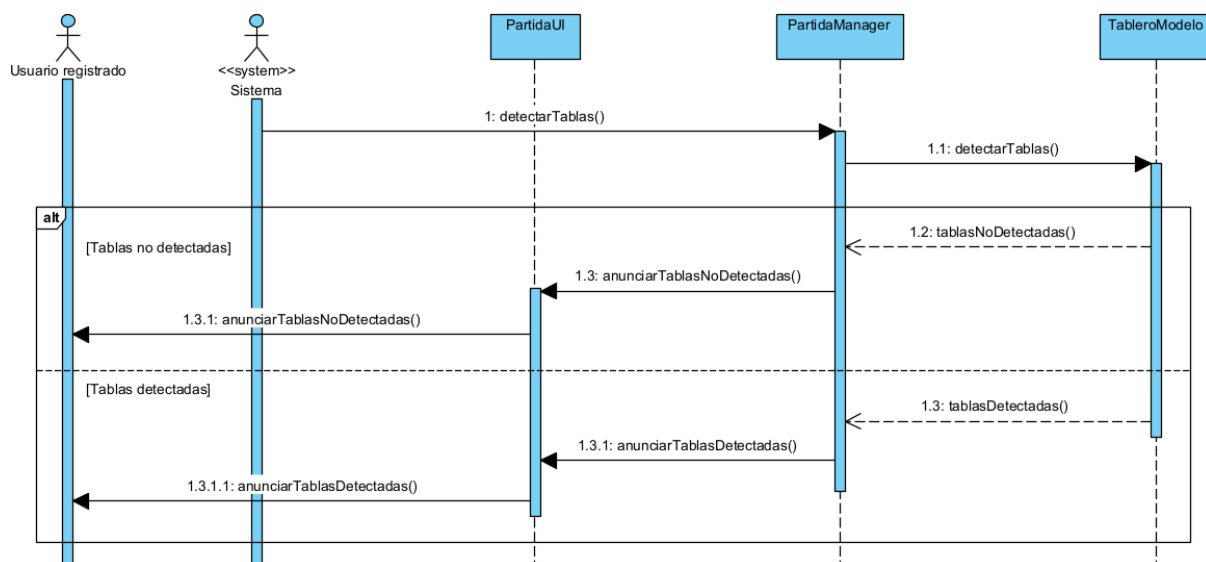


Ilustración 38: Diagrama de secuencia de diseño CU-07 Detectar tablas

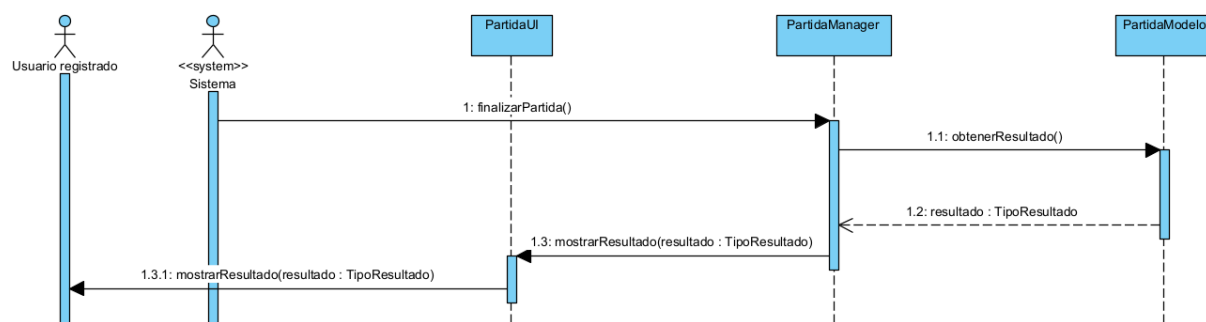


Ilustración 39: Diagrama de secuencia de diseño CU-08 Finalizar partida

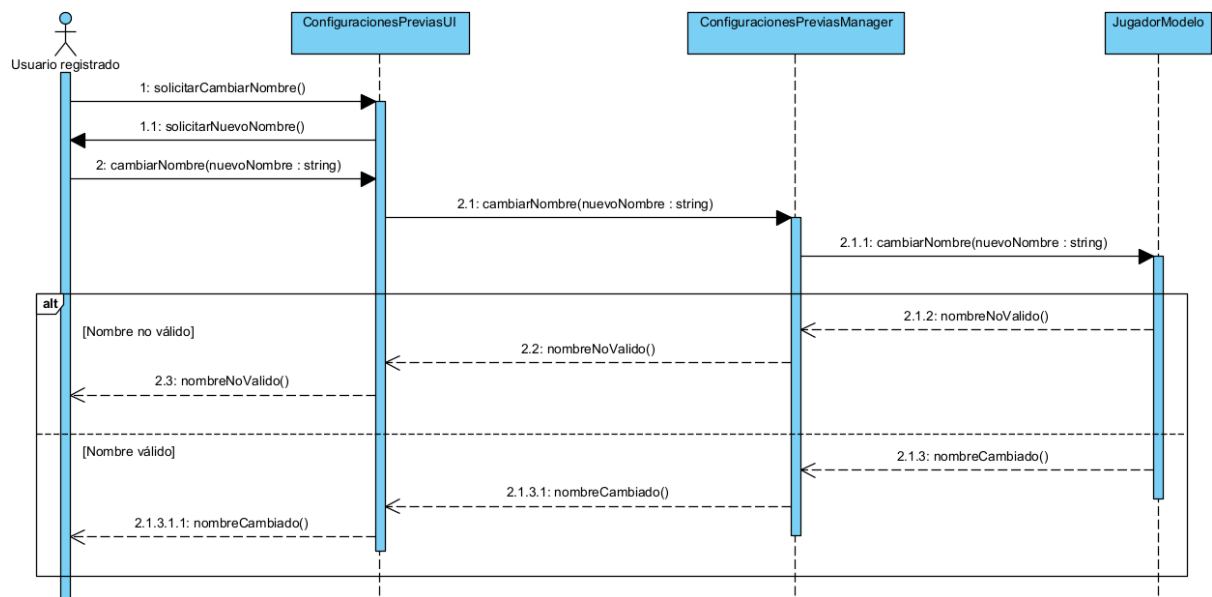


Ilustración 40: Diagrama de secuencia de diseño CU-09 Cambiar nombre jugador

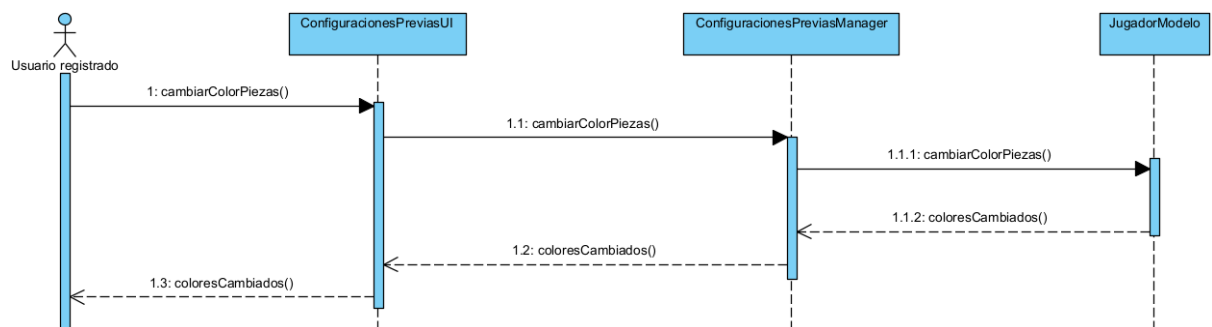


Ilustración 41: Diagrama de secuencia de diseño CU-10 Cambiar color de las piezas de los jugadores

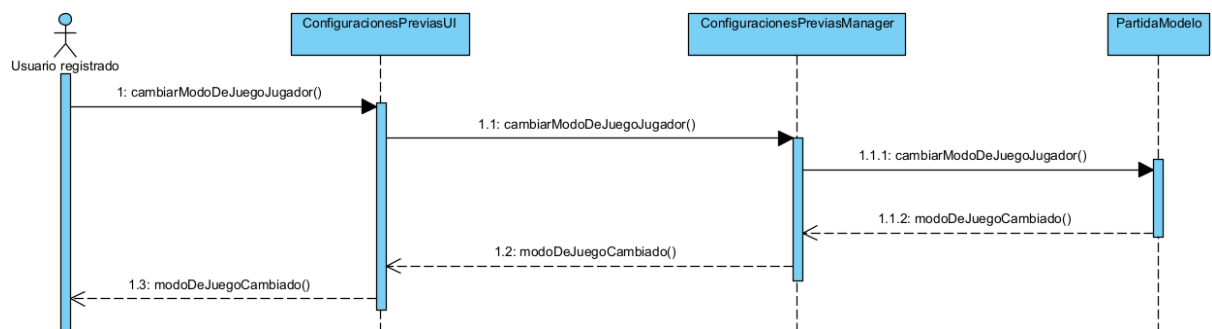


Ilustración 42: Diagrama de secuencia de diseño CU-11 Elegir modo de juego contra otro jugador

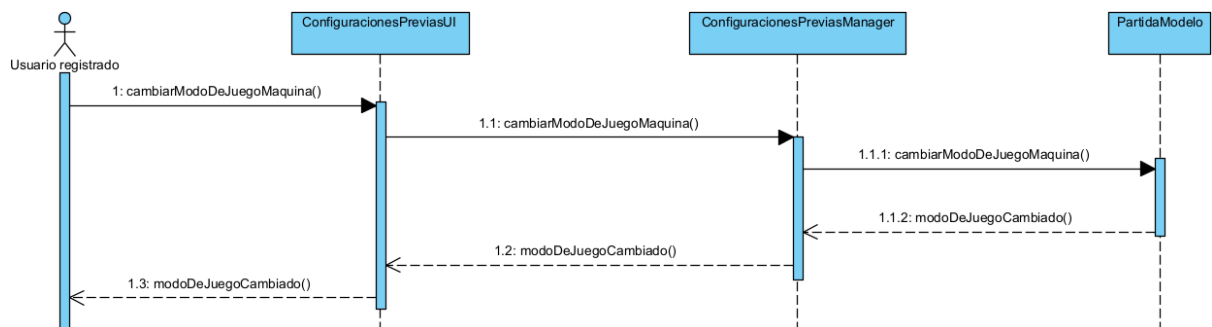


Ilustración 43: Diagrama de secuencia de diseño CU-12 Elegir modo de juego contra la máquina

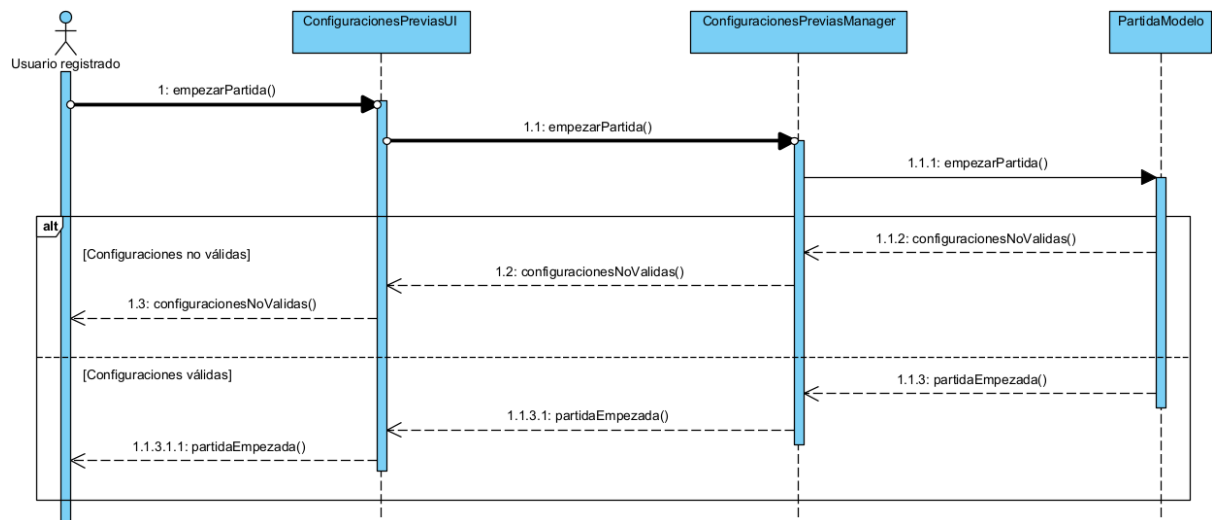


Ilustración 44: Diagrama de secuencia de diseño CU-13 Empezar partida

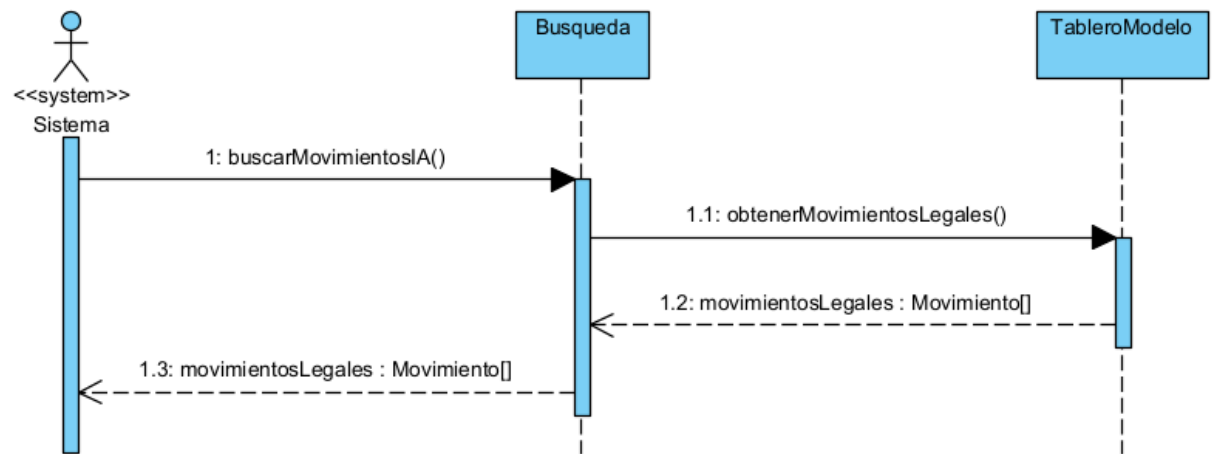


Ilustración 45: Diagrama de secuencia de diseño CU-14 Buscar movimientos para la IA

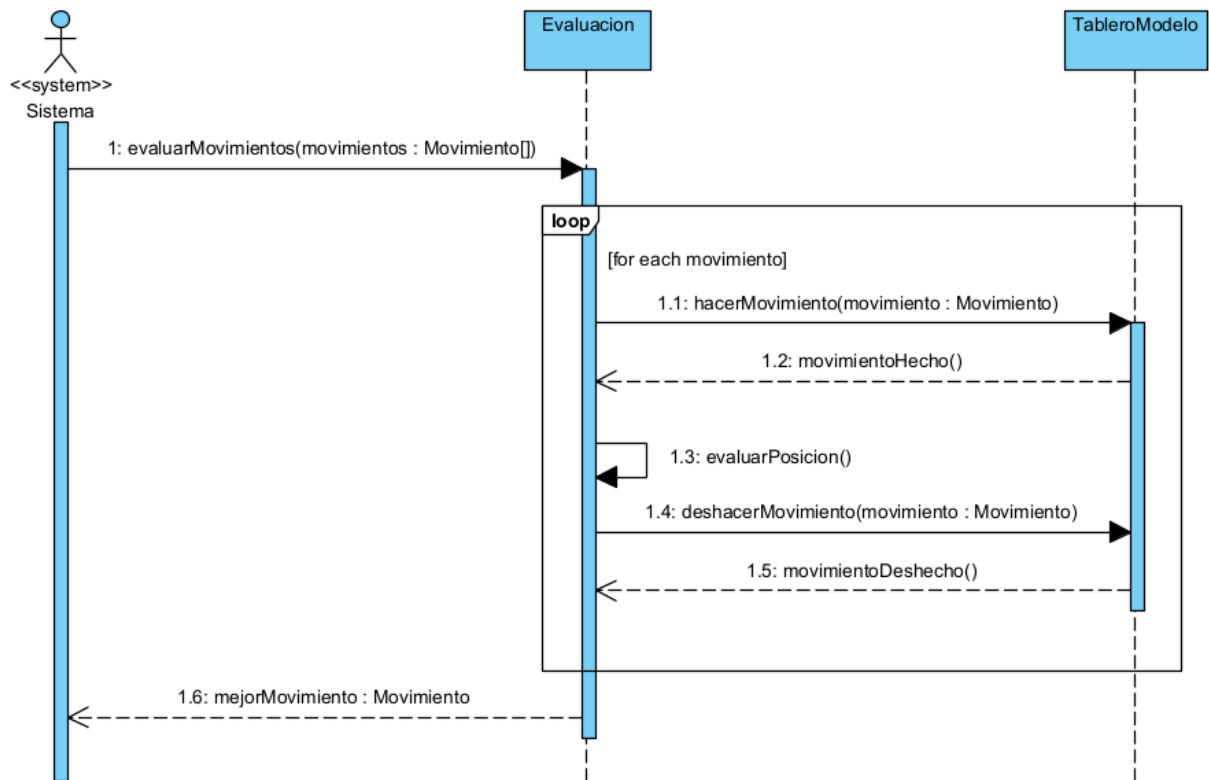


Ilustración 46: Diagrama de secuencia de diseño CU-15 Evaluar movimientos

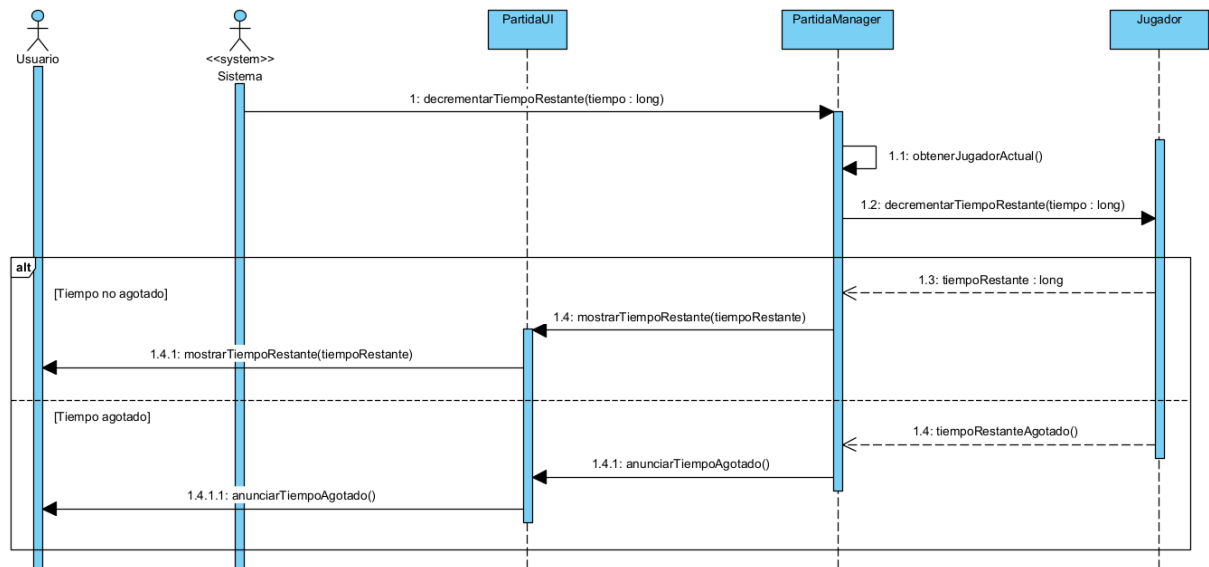


Ilustración 47: Diagrama de secuencia de diseño CU-16 Decrementar tiempo restante a los jugadores

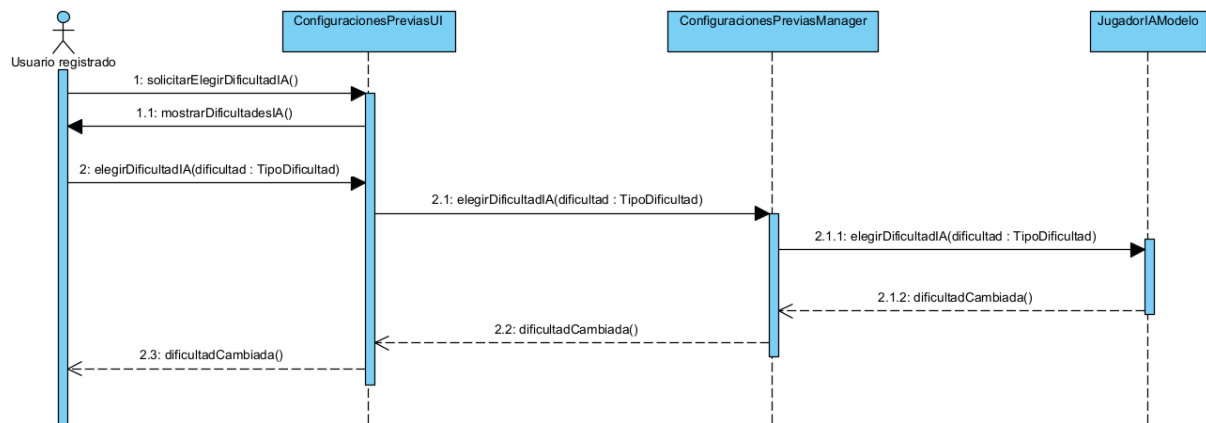


Ilustración 48: Diagrama de secuencia de diseño CU-17 Elegir dificultad de la IA

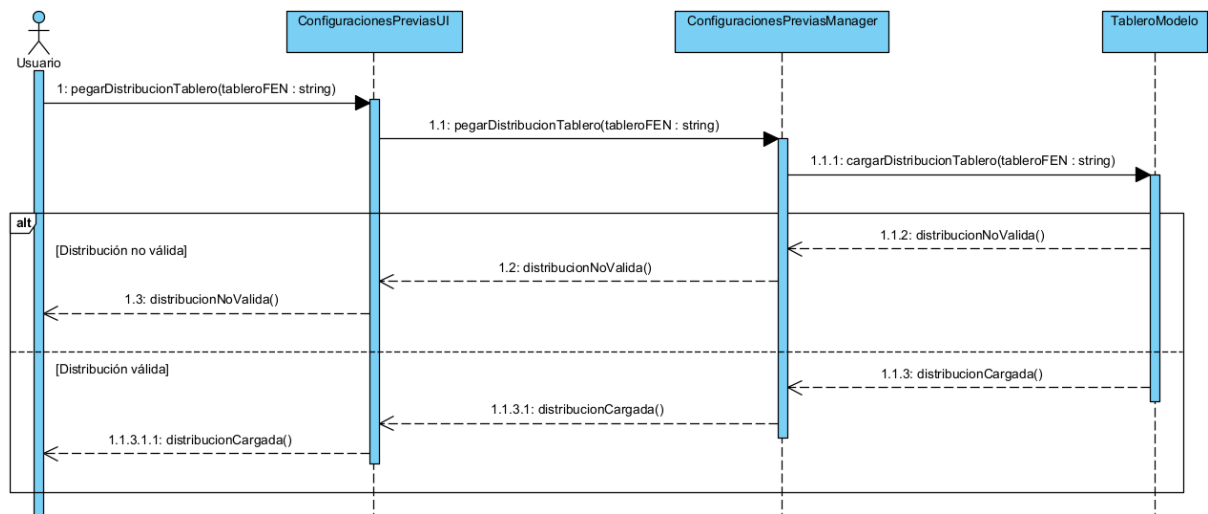


Ilustración 49: Diagrama de secuencia de diseño CU-18 Cargar distribución del tablero en formato FEN

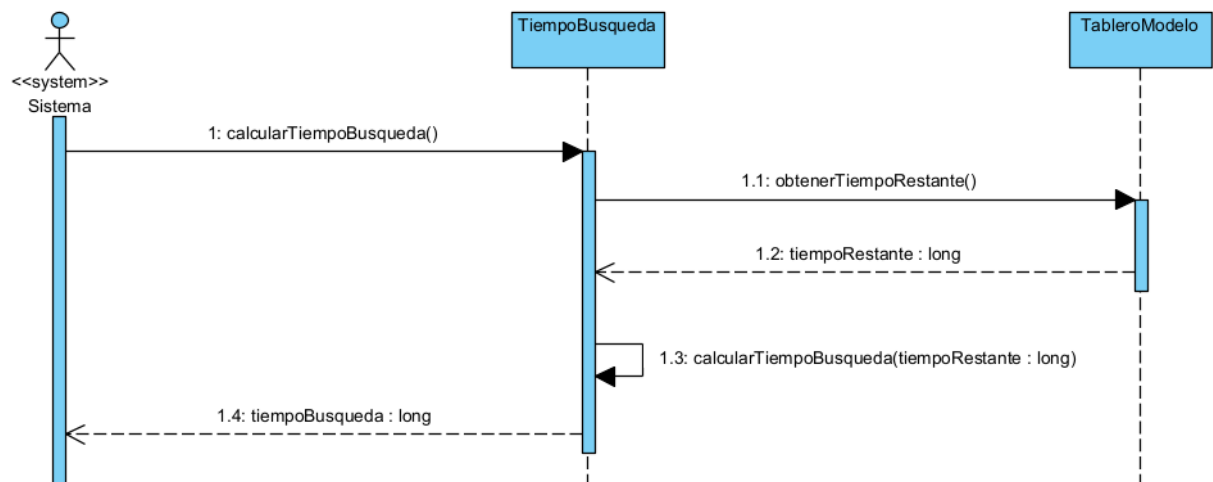


Ilustración 50: Diagrama de secuencia de diseño CU-19 Calcular tiempo de búsqueda

5. Modelo de despliegue

El **modelo de despliegue** describe cómo se distribuyen los componentes software del sistema sobre la infraestructura *hardware* en la que se ejecutan. Este modelo representa la disposición física de los nodos (servidores, dispositivos, estaciones de trabajo, etc.) y la forma en que estos se comunican entre sí mediante redes u otros medios.

En el contexto del diseño del sistema, el modelo de despliegue es fundamental porque permite visualizar la arquitectura física y comprender cómo los distintos elementos interactúan en el entorno real de ejecución. Además, facilita la planificación de aspectos como la instalación, la configuración, la seguridad, la escalabilidad y el rendimiento del sistema.

En este apartado se presenta el modelo de despliegue correspondiente al sistema desarrollado, detallando los nodos principales, los componentes software asignados a cada uno de ellos y las relaciones de comunicación entre los diferentes elementos de la infraestructura. Puesto que en el proyecto no existe ningún tipo de servicio en línea, el diagrama de despliegue sólo cuenta con un nodo físico que representa donde se ejecuta la aplicación del motor de ajedrez desarrollada en Unity.

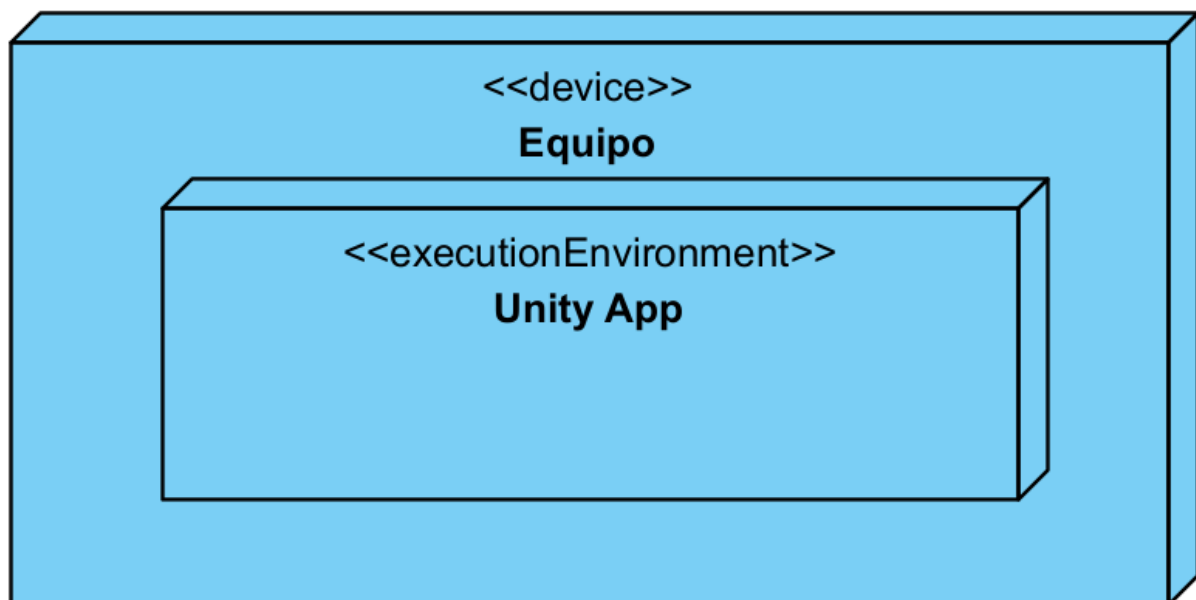


Ilustración 51: Diagrama de despliegue

6. Diseño de la interfaz

6.1. Mapa del sitio

El mapa del sitio tiene como objetivo diferenciar las pantallas con las que va a interactuar el usuario, dentro de la aplicación. Esta diferenciación ayuda que enormemente a la hora de diseñar las interfaces individualmente.

La representación de este mapa del sitio se va a hacer a partir de cuadrados, los cuáles van a indicar las pantallas de las que se compone, además de líneas que conectan dichos cuadrados, para establecer la conexión de las distintas pantallas y saber el flujo que puede tener el usuario a través de todas ellas.

El proyecto va a ser dividido en cinco pantallas:

- **Menú principal:** Será la primera pantalla con la que interactuará el usuario, nada más entrar en la aplicación, y servirá como presentación de la misma.
- **Configuraciones previas:** Es el preámbulo a la pantalla de partida, y agrupa todas las funcionalidades que se describieron anteriormente en el paquete de Gestión de configuraciones previas.
- **Partida:** Es la interfaz con la que interactuará el jugador mientras juegue.
- **Promoción:** En el momento en el que un peón promocióne o corone, esta pantalla aparecerá para permitir al usuario seleccionar el tipo de pieza al que promocionar.
- **Final de partida:** Comunica al usuario que la partida ha concluido, y muestra el resultado de la misma. También debe permitir regresar al usuario a la pantalla del menú principal.

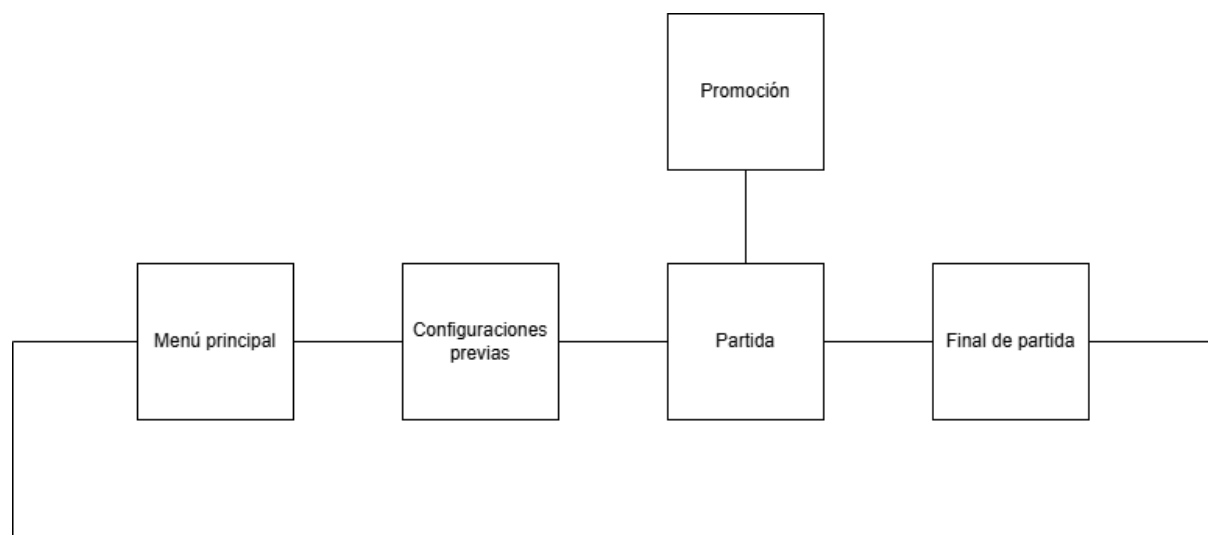


Ilustración 52: Mapa del sitio

6.2. Wireframes

Una vez definido el esquema de navegación de la aplicación, es momento de elaborar los esquemas visuales de cada pantalla de forma individual. Este esquema tiene como finalidad representar de forma gráfica el diseño de la aplicación, mostrando su estructura visual y facilitando la organización su contenido, es decir, se trata de elaborar el “*esqueleto*” de la aplicación.

Los *wireframes* cumplen de un conjunto de características comunes como es el hecho de que se centran en la funcionalidad y en la distribución del contenido, y no ahondan en el funcionamiento, las tipografías o los colores. Tampoco incluyen imágenes, sino que se representan como bloques de contenido, los cuáles también pueden representar texto que todavía no esté elaborado, o botones.

6.2.1. Wireframe del menú principal

En primer lugar, se va a realizar el *wireframe* del menú principal. Esta pantalla, al servir de presentación para el usuario, requerirá de un título, y de un conjunto de opciones para viajar a las pantallas secundarias. Aunque no se ha definido exactamente el nombre que tendrá cada botón, sus funcionalidades serán parecidas a: “*jugar*”, “*salir*”, “*opciones*”, etc.

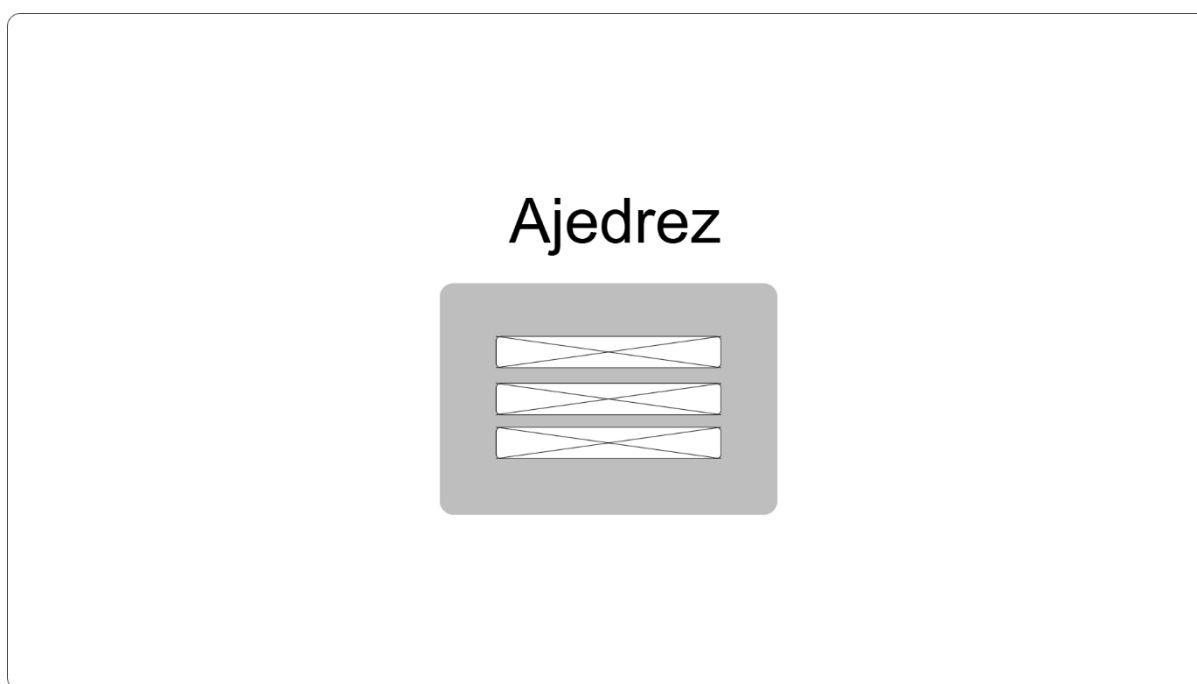


Ilustración 53: Wireframe del menú principal

6.2.2. Wireframe de configuraciones previas

Para la pantalla de configuraciones previas, se han representado en primer lugar un botón de “*volver*” y otro de “*empezar*”, en la parte superior de la pantalla. En el centro, se hallará un tablero en el que se podrá visualizar la posición de partida que se va a jugar, la cual se determinará por la notación FEN que se introduzca en el formulario que aparece justo debajo de tablero. A la derecha de este formulario, se encuentra un botón para intercambiar los colores de las piezas de los jugadores, los cuales se indican en el margen izquierdo y derecho de la pantalla, respectivamente.

Debajo de donde se indicar el color de cada jugador, habrá en primer lugar, un selector del tipo de jugador; humano o inteligencia artificial, el cuál determinará el contenido del bloque adyacente. Si se elige jugador humano, aparecerá un formulario para introducir el nombre del mismo, y si se elige inteligencia artificial, aparecerá un selector para indicar la dificultad de la misma (fácil, media o difícil).

Por último, encima del tablero, se encuentran dos selectores para establecer la duración y el incremento de partida, representado de la forma estándar que se hace en ajedrez: 1+0, 3+5, 15+10, etc.

El wireframe muestra una interfaz de configuración de partida. En la parte superior, hay dos botones: "Volver" a la izquierda y "Empezar" a la derecha. Debajo de ellos, en el centro, hay un selector para "Duración" (con un menú desplegable) seguido de "min +", luego un selector para "Incremento" (con un menú desplegable) seguido de "seg". A la izquierda del tablero, hay un selector para "Color del jugador 1" (con un menú desplegable) y un formulario para el nombre. A la derecha del tablero, hay un selector para "Color del jugador 2" (con un menú desplegable) y un formulario para el nombre. En el centro, hay un tablero de ajedrez de 8x8 casillas. Debajo del tablero, hay un selector para "Tipo de jugador" (con un menú desplegable) y un formulario para el nombre.

Ilustración 54: Wireframe de configuraciones previas

6.2.3. Wireframe de partida

En la pantalla de partida, a parte del representar el tablero en el centro, nos encontramos con el tiempo restante de cada jugador en el margen derecho, siendo el reloj superior el del jugador 2, y el inferior el del jugador 1. Además, en la esquina inferior derecho y en la superior izquierda, se visualizarán los nombres de cada jugador, que se habrán establecido en la pantalla anterior.

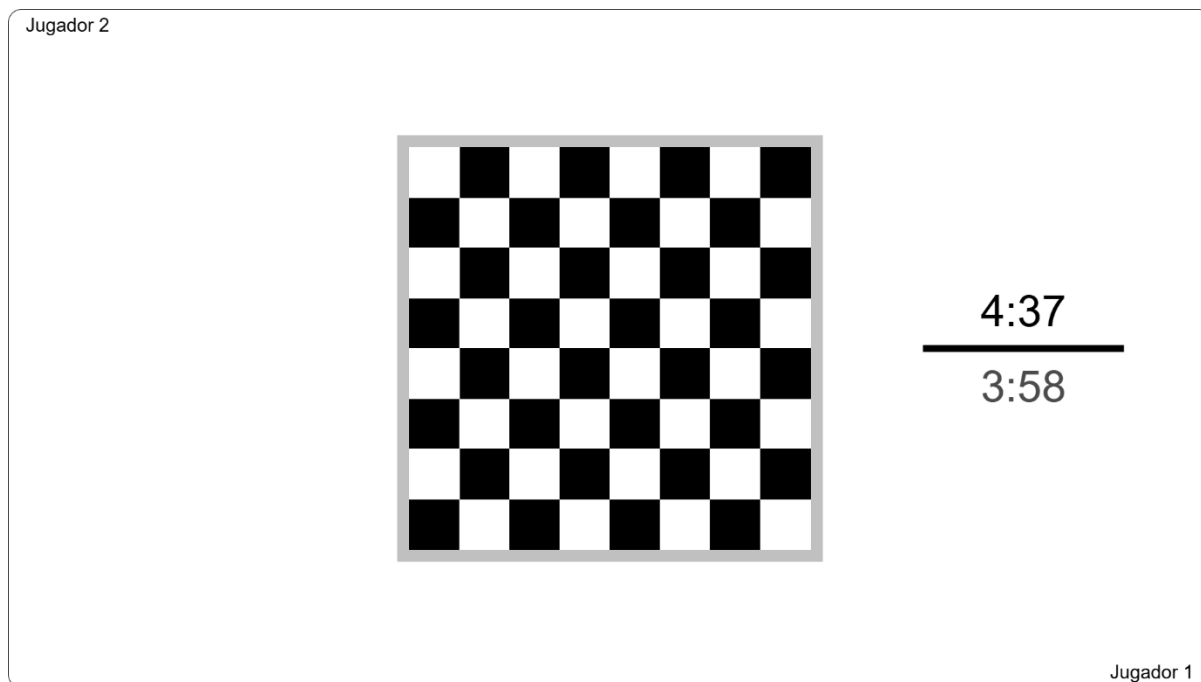


Ilustración 55: Wireframe de partida

6.2.4. Wireframe de promoción

Como se ha mencionado antes, los jugadores deben poder seleccionar el tipo de promoción que desean, siendo las posibilidades: reina, torre, alfil o caballo. Cada una de estas opciones se representará con un botón en un pequeño menú que aparecerá sobre la pantalla de partida.

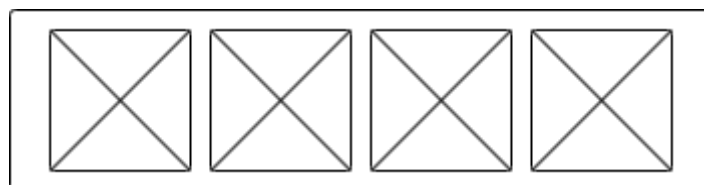


Ilustración 56: Wireframe de promoción

6.2.5. Wireframe de final de partida

Por último, queda definir cómo será la pantalla de final de partida, la cual también aparecerá encima de la de partida, e informará del resultado de la misma. En la parte inferior de esta ventana, habrá un botón que permitirá volver a la pantalla del menú principal.

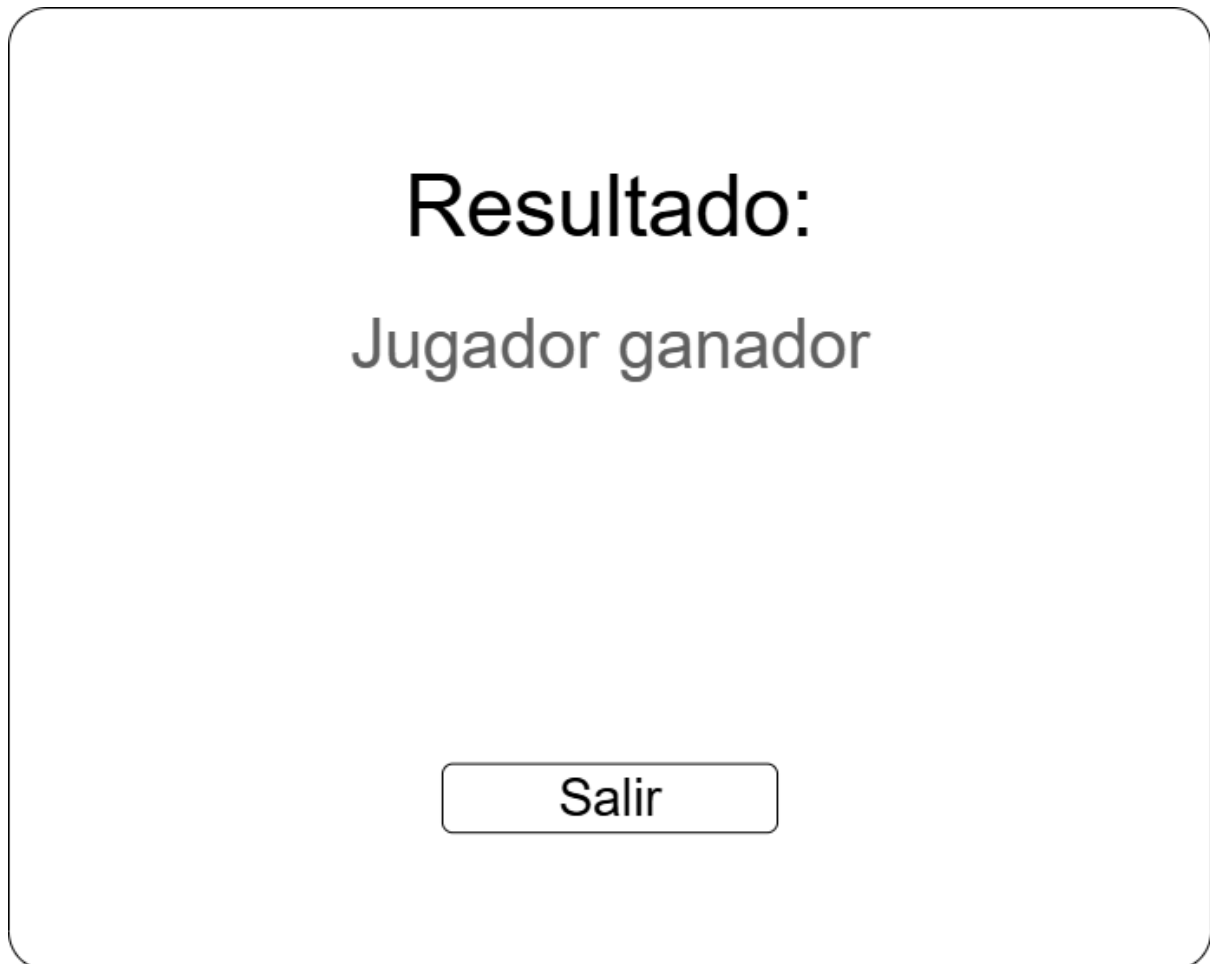


Ilustración 57: Wireframe de final de partida

7. Bibliografía

Cherif, Y. (2024). Understanding the Layered Architecture Pattern: A Comprehensive Guide.

García, M. N. (n.d.). Transparencias de Ingeniería del Software II, Tema 2 - UML. Unified Modeling Language. Salamanca.