

-nameNotFound- Team

“JavaChess”

DOCUMENTO DE REQUERIMIENTOS

Autor(es)

Cáceres, Juan Manuel
Hidalgo, Fabian Nicolás
Mamani, Franco Ivan
Mendez, Jorge Nicolas

Versión de documento

1.0.0

Tabla de contenidos

1	<i>HISTORIAL DE REVISIONES</i>
2	<i>PÁGINA DE APROBACIONES</i>
3	<i>INTRODUCCIÓN</i>
3.1	<i>Propósito</i>
3.2	<i>Enfoque</i>
3.3	<i>Referencias / Acrónimos / Glosario</i>
3.4	<i>Visión General</i>
4	<i>DESCRIPCIÓN GENERAL</i>
4.1	<i>Funciones de sistema</i>
4.2	<i>Características para el Usuario</i>
5	<i>RESTRICCIONES GENERALES</i>
5.1	<i>Limitaciones de Software</i>
5.2	<i>Limitaciones de Hardware</i>
6	<i>SUPUESTOS Y DEPENDENCIAS</i>
7	<i>MASTER LIST DE REQUERIMIENTOS</i>
7.1	<i>Requerimientos Propios del Juego Ajedrez</i>
7.2	<i>Requerimientos para Funcionalidades Adicionales</i>
7.3	<i>Requerimientos para la UI</i>
7.3.1	<i>Requerimientos para la UI de Consola</i>
7.3.2	<i>Requerimientos para la UI de Ventanas</i>
8	<i>DETALLES DE LOS REQUERIMIENTOS</i>
8.1	<i>Requerimientos Propios del Juego Ajedrez</i>
	<i>REQ 1: Cantidad y atributos inamovible de las piezas</i>
	<i>REQ 2: Jugadas especiales posibles predefinidas</i>
	<i>REQ 3: Cambio de turno</i>
	<i>REQ 4: Movimientos solo dentro del turno correspondiente</i>
	<i>REQ 5: Movimientos solo de piezas propias</i>
	<i>REQ 6: Límites del tablero de juego</i>
	<i>REQ 7: Campo rival definido</i>
	<i>REQ 8: Eliminación de una pieza propia o rival</i>
	<i>REQ 9: Posición de Jaque</i>
	<i>REQ 10: Posición de Jaque Mate</i>
8.2	<i>Requerimientos para Funcionalidades Adicionales</i>
	<i>REQ 1: Control de contadores de tiempo</i>
	<i>REQ 2: Control de tiempo para modo de juego competitivo</i>

REQ 3: Respuesta de la AI
REQ 4: Dificultades de la AI
REQ 5: Control de sincronización en el juego 1vs1
REQ 6: Funcionamiento del chat online 1vs1
REQ 7: Funcionamiento del chat online 1vsCOM
REQ 8: Recopilación de información de la partida

8.3 *Requerimientos para la UI*

8.3.1 *Requerimientos para la UI de Consola*

REQ 1: Visualización de las piezas y el tablero
REQ 2: Entrada de datos por consola
REQ 3: Simpleza de la UI

8.3.2 *Requerimientos para la UI de Ventanas*

REQ 1: Ventana para el arranque del sistema
REQ 2: Ventana de partida
REQ 3: Ventana de Scoreboard o estadísticas
REQ 4: Ventana de configuración

9 *REQUERIMIENTOS DE INTERFAZ EXTERNA*

10 *RESTRICCIONES DE DISEÑO GENERAL*

1 Historial de revisiones

Versión	Fecha	Observación	Autor(es)
1.0.0	26/04/2022	Versión base del documento	Cáceres, Juan Manuel Hidalgo, Fabian Nicolás Mamani, Franco Ivan Mendez Jorge Nicolas
1.0.1	18/06/2022	Correcciones en base a Feedback	Cáceres, Juan Manuel Hidalgo, Fabian Nicolás
1.0.2	19/06/2022	Arquitectura Preliminar	Cáceres, Juan Manuel Hidalgo, Fabian Nicolás

2 Página de Aprobaciones

A continuación se encuentra la información de las personas cuya aprobación es necesaria para realizar cambios significativos al siguiente plan.

* Aprobación sólo será necesaria en caso de cambios mayores, no será necesaria para cambios menores.

* Aprobación sólo será necesaria en caso de que el cambio sea realizado por personas que no son Change Managers.

Configuration Manager	Fecha	Firma
Hidalgo, Fabian Nicolás	26/04/2022	

Configuration Manager BackUp	Fecha	Firma
Cáceres, Juan Manuel	26/04/2022	

Build Manager	Fecha	Firma
Méndez, Jorge Nicolás	26/04/2022	

Release Manager - Issue Coordinator	Fecha	Firma
Mamaní, Franco Iván	26/04/2022	

3 Introducción

Esta sección es introductoria al documento de especificación de los requerimientos para el proyecto “JavaChess”, donde se provee el propósito, objetivos y foco del sistema a tener en cuenta a lo largo de todo el proceso de documentación, desarrollo y presentación del proyecto. Cualquier definición o referencia se encuentra listada en la visión general del siguiente documento de especificación de requerimientos.

3.1 Propósito

Este documento de especificación de requerimientos describe las funciones y requisitos especificados para el desarrollo del proyecto “JavaChess”, siendo éste un juego de ajedrez para plataforma de computadora desarrollado en el lenguaje de programación JAVA. El propósito del documento de especificación de requerimientos es el de poder llevar un control sobre cada uno de los puntos a desarrollar en el sistema, definiendo la funcionalidad y pudiendo entender como debe de desarrollarse, sabiendo con anterioridad el resultado esperado.

3.2 Enfoque

El enfoque que se toma en el desarrollo del sistema “JavaChess” es puramente práctico basándose en la teoría, donde buscamos aplicar las herramientas y metodologías ágiles para el desarrollo de software y poner a prueba el sistema de una forma controlada. La idea es la de llevar un método de integración continua al proyecto y poder cumplir con los requerimientos que hemos establecido durante el desarrollo del sistema.

El objetivo de este trabajo es la creación de un programa de computadora en donde se pueda implementar el juego de ajedrez con sus funcionalidades básicas, en una primera entrega, e ir agregando funcionalidades a medida que se especifiquen los requerimientos a desarrollar, como puede ser un modo contra la computadora, un modo contra otros usuarios, la integración de un chat en vivo, contadores de tiempo por turno y de partida, scoreboards, menús, sección de ajustes de la aplicación, etc.

3.3 Referencias / Acrónimos / Glosario

Acrónimo	Descripción
SCM	Software Configuration Management - Administración de las Configuraciones de Software
CI	Continuous Integration - Integración Continua
CD	Continuous Deployment - Implementación Continua
ID	Identification Code - Código de Identificación

AI	Artificial Intelligence - Inteligencia Artificial
UI	User Interface - Interfaz de Usuario
1vsCOM	Jugador vs Computadora
1vs1	Jugador vs Jugador
SRS	Software Requirements Specifications - Especificación de Requerimientos de Software

Tabla 1 - Acrónimos/Glosario

3.4 Visión General

En el desarrollo de este documento de requerimientos de software veremos un enfoque de alto nivel en lo que respecta a la descripción de las funcionalidades del sistema, pudiendo explicar en palabras su funcionamiento esperado y su funcionamiento no deseado, para así tener un detalle y análisis rápido. De esta manera, cualquier individuo que desee interiorizarse en los detalles del sistema, puede entender los requerimientos solicitados y cómo se reflejan en el producto final de cada release o fase de desarrollo. Los requerimientos se ordenarán en forma de lista, con un código único de identificación (ID) y luego los detalles del mismo que lo caracterizan y describen el objetivo del mismo.

4 Descripción General

Esta sección del documento de requerimientos de software proveerá una descripción de alto nivel en 2 ámbitos principales del sistema.

4.1 Funciones de Sistema

El juego deberá ser capaz de poder desarrollarse en un ámbito donde se encuentren todas las piezas necesarias para cada color, sin posibilidad de ser repetidas, o habiendo existencia de faltantes, desarrollándose el juego sobre un tablero de 8x8. Además deberá contar con la posible realización de los movimientos específicos de cada pieza y las jugadas especiales como ser, el “Enroque de Torre y Rey” o la “Coronación de Peón”. También deberá poder ejecutarse en consola como una primera interfaz gráfica, facilitando el desarrollo en las primeras instancias del proyecto, para luego poder realizar una UI más amigable con el usuario. También deberá poder jugar contra la máquina. Estas funcionalidades básicas del sistema serán el punto de partida inicial del proyecto.

Una vez que se tenga una versión base del juego se buscarán implementaciones que añadan mejor experiencia, pero siempre basándose en el juego como foco principal. Estas implementaciones serán, por ejemplo, interfaz de usuario en ventanas, menús, distintos modos de juego o selección de estrategias para el juego contra la computadora, contadores de tiempo por turnos, un chat para juego online, etc.

4.1 Características para el Usuario

Como características para el usuario le daremos un enfoque más destinado a la parte gráfica del proyecto, por lo que estas características se verán más reflejadas en los requerimientos destinados a la segunda parte de la realización del juego, ya que como primera medida buscaremos poder tener un juego muy limitado en cuanto a opciones de juego, con solamente funciones básicas relacionadas a un juego de ajedrez y siendo éste mostrado por consola, con funcionalidades muy reducidas.

Una vez que se empiece con el desarrollo de los requerimientos posteriores a la primera release base del juego, se empezará con el “embellecimiento” del mismo, buscando una implementación cómoda en los menús y pudiendo tener cada apartado organizado y de rápido acceso para el jugador. También queremos guardar estadísticas de las partidas para el desarrollo de un “Scoreboard” pudiendo ver por ejemplo cuántas partidas se realizaron, cantidad y posición de los movimientos que se realizaron en total, duración de la partida en tiempo real, etc, para luego con esos datos además de mostrarlos en una tabla, poder representarlos en una unidad de puntos de juego a modo de puntaje de un determinado jugador. También en el modo de un jugador, como un requerimiento bueno a considerar, es el de que un jugador al cerrar una partida, pueda retomarla en otro momento en donde quedó el último movimiento antes de salir de la aplicación.

5 Restricciones Generales

5.1 Limitaciones de Software

El sistema deberá contar con Windows 7 o superior, versión de JAVA 8 en adelante, CMD de Windows para las versiones iniciales con interfaz gráfica por consola y posteriormente Microsoft.NET Framework para las versiones con interfaz gráfica en ventanas.

5.2 Limitaciones de Hardware

El hardware necesario para la ejecución es una computadora de escritorio con procesador de 64 bits, entrada de teclado para poder interactuar al juego por consola y luego un ratón opcional para la interfaz gráfica de próximas releases.

‘Supuestos y Dependencias

El programa asume que se poseen las entradas mínimas de teclado y monitor necesarias para interaccionar con el sistema y que se puede ejecutar sin inconvenientes a lo que software y hardware se refiere a nivel de procesamiento de la computadora donde se ejecuta la aplicación de escritorio.

Se da por hecho que el usuario no tendrá inconvenientes con el uso básico de un sistema informático como ser una computadora de uso personal, tanto para el “setup” del programa, como puede ser la ejecución del ejecutable “.exe” como también luego la interacción dentro del mismo y el “exit” al escritorio al finalizar la sesión de juego.

Otro punto que se da por sentado es que no se contempla todo uso erróneo e inesperado que el usuario haga del sistema, o sea, que el usuario plantee situaciones que no se hayan analizado y especificado en los requerimientos del diseño del software, dando al equipo de desarrollo la posibilidad de encontrar posibles defectos inesperados que no fueron planteados en un principio.

7 Master List de Requerimientos

Esta sección del documento contiene una lista de todos los requerimientos que se plantean en el desarrollo del sistema, donde en cada uno de éstos se detalla la ID del requerimiento a ser testeado y una descripción breve de lo que se espera en cada punto. En la siguiente sección se ve en detalle cada uno de los requerimientos.

7.1 Requerimientos Propios del Juego Ajedrez

REQP_1: Cantidad y atributos inamovible de las piezas: Las piezas de la partida no deben ser modificables en cantidad ni en movimientos, cada pieza debe ser única junto con sus atributos, o sea que no pueden ni instanciarse o destruirse, o variar su rol, siempre que no sea contemplado por las reglas de juego.

REQP_2: Movimientos permitidos de piezas: Limitar los movimientos de las piezas a aquellos regidos por las reglas de ajedrez, de manera que solo puedan desplazarse a posiciones dentro de este límite.

REQP_3: Cambio de turno: Luego de que un jugador realice un movimiento y éste haya sido válido, su turno debe finalizar e iniciar el turno de su oponente.

REQP_4: Movimientos solo dentro del turno correspondiente: Si un jugador no se encuentra en su turno correspondiente, le debe ser imposible poder efectuar movimientos de sus piezas.

REQP_5: Movimientos solo de piezas propias: Cada jugador solamente podrá tener acción sobre las piezas que haya elegido al arranque de la partida.

REQP_6: Límites del tablero de juego: Los límites de juego del tablero deben estar definidos, para así evitar posibles salidas del campo de juego hacia casillas inexistentes.

REQP_7: Campo rival definido: Se debe definir a la primera fila del tablero de cada jugador como la zona en la que, al llegar un Peón contrincante, éste pueda coronarse como una pieza cualquiera, a excepción del Rey.

REQP_8: Eliminación de una pieza propia o rival: Ya sea por un movimiento propio o del contrincante, el sistema se debe asegurar que al capturar una pieza ésta desaparezca del tablero y no sea elegible en turnos posteriores.

REQP_9: Posición de Jaque: Cuando un jugador realice un movimiento que ponga en jaque (amenaza directa) al rey del rival, el oponente deberá realizar obligatoriamente un movimiento para liberar a su rey del jaque.

REQP_10: Posición de Jaque Mate: Si el rey de un jugador se coloca en posición de jaque y no existe algún movimiento que el jugador pueda hacer para sacarlo de allí, entonces el rey está en jaque mate, el juego termina y ese jugador pierde.

7.2 Requerimientos para Funcionalidades Adicionales

REQFA_1: Control de contadores de tiempo: Se debe garantizar que tanto el contador de duración de partida, el de turno actual, y el tiempo total disponible para cada jugador, inicien y detengan en sus debidos instantes de tiempo correspondientes. Así como existe un contador de tiempo ascendente global de partida, habrá otro contador ascendente que cada jugador dispondrá para guardar su propio conteo de tiempo por turno, y por último un contador que existirá en modos de juego competitivos, donde serán contadores descendentes en el cual el jugador deberá realizar su movimiento con velocidad antes de que finalice el conteo.

REQFA_2: Control de tiempo para modo de juego competitivo: En modo competitivo, según el nivel de dificultad, el tiempo disponible restante para cada jugador se debe ir reduciendo y será una variable propia de cada participante, donde al cambiar de turno en un muy corto periodo de tiempo, éste se podrá aumentar en una pequeña proporción a modo de beneficio al jugador.

REQFA_3: Respuesta de la AI: En un modo de juego contra la computadora, se debe garantizar mínimamente que al encontrarse en el turno de la computadora, ésta efectúe un movimiento válido obligatoriamente, donde luego la coherencia del movimiento responderá al modo elegido en la partida.

REQFA_4: Modos de la AI: Cada modo de dificultad deberá ser seleccionable en tiempo real, haciendo que la computadora cambie la lógica detrás de sus movimientos..

REQFA_5: Control de sincronización en el juego 1vs1: Se debe poder sincronizar los movimientos de cada jugador, para que puedan verse reflejado en el tablero del adversario, luego de haberse realizado el movimiento deseado.

REQFA_6: Funcionamiento del chat online 1vs1: Se debe poder enviar y recibir mensajes de manera instantánea mientras se desarrolla una partida en el modo 1vs1, mostrando los mensajes en orden cronológico y siendo éstos inalterables una vez enviados.

REQFA_7: Funcionamiento del chat online 1vsCOM: En partidas contra la computadora, el chat debe estar inhabilitado completamente.

REQFA_8: Recopilación de información de la partida: Se debe poder recopilar información sobre todos los movimientos que se realizan durante la partida. Esta información contiene qué jugador realizó el movimiento, de que casilla inicial a cuál casilla final se desplazó, que pieza se movió, qué tiempo se tardó en el turno, que pieza se capturó en el movimiento, que Peón se coronó y que pieza se le asignó, si hubo Jaque al finalizar el movimiento, etc.

7.3 Requerimientos para la UI

En esta sección se dispondrán los requerimientos solicitados para ambas UI gráficas, tanto para la consola en sus versiones más base, como la UI en ventana para las versiones más avanzadas.

7.3.1 Requerimientos para la UI de Consola

REQUIC_1: Visualización de las piezas y el tablero: Se deberá poder visualizar correctamente las piezas y el tablero, así como las columnas y filas deben estar unívocamente definidas.

REQUIC_2: Entrada de datos por consola: Se deberá poder recibir datos por consola y analizarlos para poder así manejar tanto el menú principal y las configuraciones o diferentes pestañas, como en partida poder ingresar la posición de la pieza a mover, y su casilla final.

7.3.2 Requerimientos para la UI de Ventanas

REQUIV_1: Ventana para el arranque del sistema: Al iniciar la aplicación se deberá presentar un menú principal con las opciones principales, como partida, scoreboard o estadísticas, configuraciones y salir.

REQUIV_2: Ventana de partida: Al entrar en el apartado de partida, luego de haber configurado a gusto las opciones como el modo de juego, la dificultad, etc, se debe presentar una pantalla en la cual se visualiza el tablero inicializado con sus respectivas fichas y los contadores en sus valores iniciales, según sea cada caso.

REQUIV_3: Ventana de Scoreboard o estadísticas: En esta sección se debe visualizar una lista de últimas partidas jugadas, con opción de ver todos sus datos como cantidad de movimientos, duración en tiempo real, etc, también se debe visualizar los mejores jugadores con sus puntajes correspondientes y debe existir la opción de ordenar los resultados mostrados.

REQUIV_4: Ventana de configuración: En esta sección se deberá visualizar una lista de opciones configurables (AÚN NO DEFINIDAS)

8 Detalles de los Requerimientos

Aquí en esta sección se detallarán los requerimientos listados en el apartado de Master List de Requerimientos, brindando una descripción, que entrada de datos debe darse, como se procesan los datos, que se tiene que presentar en pantalla, y que restricciones posee el requerimiento.

8.1 Requerimientos Propios del Juego Ajedrez

REQP_1: Cantidad y atributos inamovible de las piezas

DESCRIPCIÓN:

Las piezas de la partida no deben ser modificables en cantidad ni en movimientos, cada pieza debe ser única junto con sus atributos, o sea que no pueden ni instanciarse o destruirse, o variar su rol, siempre que no sea contemplado por las reglas de juego.

ENTRADA DE DATOS:

Se tiene que poder iniciar una partida nueva y las fichas se deben inicializar de acuerdo a como es una partida de ajedrez normal, luego se tiene que seleccionar una pieza y elegir la casilla a la que mover, y repetir este proceso varias veces..

PROCESAMIENTO DE DATOS:

Al inicializar el tablero se deben instanciar las piezas por defecto y solo se podrán instanciar otras en determinados casos a lo largo del desarrollo de la partida y si es que se cumplen las condiciones de juego necesarias. También cuando se elige una pieza en una casilla y luego se elige la casilla de destino, esta deberá ser validada por el sistema si es que la pieza elegida tiene válido ese movimiento deseado. Estos atributos de cada pieza no deben cambiar de un momento a otro y deben ser siempre los mismos, o sea que no debe cambiar ni las opciones de movimiento propias de cada pieza ni el tipo de pieza

SALIDA DE DATOS:

A la salida de los datos al iniciar una partida se espera que se encuentren todas las piezas por defecto necesarias instanciadas para ambos jugadores, también, al mover una pieza a una posición válida, y repetir este proceso varias veces, se debe poder seguir efectuando el movimiento.

RESTRICCIONES:

Para la prueba de verificación de los movimientos de cada pieza se realiza la prueba pieza por pieza en un tablero vacío y sin variar el turno del jugador para agilizar el procedimiento, donde una vez probada una pieza se inicia la prueba nuevamente con otra pieza sola en tablero.

REQP_2: Movimientos permitidos de piezas

DESCRIPCIÓN:

Limitar los movimientos de las piezas a aquellos regidos por las reglas de ajedrez, de manera que solo puedan desplazarse a posiciones dentro de este límite.

ENTRADA DE DATOS:

Se debe elegir una pieza a mover y seleccionar una casilla válida según el movimiento teórico de esa pieza, luego se debe repetir el proceso pero esta vez a una casilla en la cual no se cumpla su posibilidad de movimiento.

PROCESAMIENTO DE DATOS:

El sistema debe poder testear que los movimientos de la pieza con sus atributos designados son los correctos, y solo debe dejar desplazar una pieza si la entrada del usuario es factible según lo dicten las reglas del juego.

SALIDA DE DATOS:

En el primer movimiento, el sistema debe poder ser capaz de efectuar el movimiento solicitado, ya que es válido, y en el movimiento siguiente, deberá informar al usuario que el movimiento no es válido y que se debe ingresar nuevamente una posición en el tablero válida.

RESTRICCIONES:

Se realizará la prueba pieza por pieza en un tablero vacío y sin variar el turno del jugador, donde una vez probada una pieza se inicia la prueba nuevamente con otra pieza sola en tablero.

REQP_3: Cambio de turno

DESCRIPCIÓN:

Luego de que un jugador realice un movimiento y éste haya sido válido, su turno debe finalizar e iniciar el turno de su oponente.

ENTRADA DE DATOS:

Se debe seleccionar una pieza del tablero, y realizar un movimiento a una casilla válida.

PROCESAMIENTO DE DATOS:

El sistema debe entender que el movimiento es válido, y una vez ejecutado tiene que ser capaz de cambiar el turno al contrincante, realizando todas las acciones que eso implica, como por ejemplo pausar los contadores de tiempo de un jugador y reanudando el del adversario.

SALIDA DE DATOS:

Se debe poder observar como el control pasa a ser del contrincante y los contadores e indicadores se ponen a la espera de la jugada del oponente para efectuar luego de una jugada válida un nuevo cambio de turno.

RESTRICCIONES:

No se encuentran restricciones.

REQP_4: Movimientos solo dentro del turno correspondiente**DESCRIPCIÓN:**

Si un jugador no se encuentra en su turno correspondiente, le debe ser imposible poder efectuar movimientos de sus piezas.

ENTRADA DE DATOS:

Le debe ser imposible ingresar datos de posición al jugador que se encuentra en espera.

PROCESAMIENTO DE DATOS:

El juego debe anular las opciones de entrada de datos del jugador que se encuentra esperando su turno correspondiente, impidiendo que ingrese órdenes a la partida que causarían conflictos.

SALIDA DE DATOS:

No hay ninguna salida de datos.

RESTRICCIONES:

No se encuentran restricciones.

REQP_5: Movimientos solo de piezas propias**DESCRIPCIÓN:**

Cada jugador solamente podrá tener acción sobre las piezas que haya elegido al arranque de la partida.

ENTRADA DE DATOS:

Se deberá seleccionar una pieza del adversario.

PROCESAMIENTO DE DATOS:

El sistema deberá entender que el dato ingresado de casilla pertenece a una pieza que no es de su propiedad y por lo tanto deberá denegar la solicitud, pidiendo al usuario que ingrese una casilla con una pieza propia a mover.

SALIDA DE DATOS:

Deberá indicar en la pantalla al jugador que la pieza que fue seleccionada es inválida y que debe seleccionar nuevamente, habilitando la posibilidad de elegir nuevamente.

RESTRICCIONES:

No se encuentran restricciones.

REQP_6: Límites del tablero de juego**DESCRIPCIÓN:**

Los límites de juego del tablero deben estar definidos, para así evitar posibles salidas del campo de juego hacia casillas inexistentes.

ENTRADA DE DATOS:

Se debe elegir una pieza que se encuentre en un borde y con un movimiento “válido” hacia una posición fuera del tablero.

PROCESAMIENTO DE DATOS:

El sistema no tiene que ser capaz de habilitar ese movimiento y deberá avisar al usuario que ese movimiento es inexistente y necesita que se ingrese una nueva posición válida.

SALIDA DE DATOS:

El juego deberá avisar que la casilla a la que se desea mover es inexistente, y que ingrese nuevamente una casilla válida.

RESTRICCIONES:

Se deberá realizar la prueba con una sola pieza en el tablero que pueda moverse en cualquier dirección, colocándola en cada borde del tablero e intentando moverla hacia afuera del mismo.

REQP_7: Campo rival definido**DESCRIPCIÓN:**

Se debe definir a la primera fila del tablero de cada jugador como la zona en la que, al llegar un Peón contrincante, éste pueda coronarse como una pieza cualquiera, a excepción del Rey.

ENTRADA DE DATOS:

Se debe seleccionar un Peón que esté a un movimiento de coronarse, y efectuar el movimiento hacia la última fila del tablero.

PROCESAMIENTO DE DATOS:

El sistema tiene que interpretar correctamente que esa zona es la zona rival y si se ejecuta la jugada de “Coronación de Peón” correctamente se puede afirmar que esa zona se valida como el punto de partida del contrincante.

SALIDA DE DATOS:

Se debe poder elegir en qué pieza queremos que se convierta nuestro Peón.

RESTRICCIONES:

Se realiza la prueba 2 veces intercambiando de jugador, con un solo peón del equipo contrario, en la posición de la anteúltima fila, para ejecutar la jugada una sola vez y analizar el resultado.

REQP_8: Eliminación de una pieza propia o rival**DESCRIPCIÓN:**

Ya sea por un movimiento propio o del contrincante, el sistema se debe asegurar que al capturar una pieza ésta desaparezca del tablero y no sea elegible en turnos posteriores.

ENTRADA DE DATOS:

Se debe seleccionar una pieza propia y mediante un movimiento válido se tiene que seleccionar una casilla donde podamos tomar una pieza enemiga.

PROCESAMIENTO DE DATOS:

El sistema debe ser capaz de interpretar la jugada de forma correcta y realizar la toma de la pieza adversaria que fue capturada, quitando del tablero y realizando el movimiento de la pieza que se desea mover.

SALIDA DE DATOS:

Se debe informar que el movimiento fue válido y que se capturó una pieza enemiga, informando cual es, y quitando del tablero.

RESTRICCIONES:

Se realiza la prueba con únicamente 2 piezas en el tablero, ubicadas estratégicamente para que se pueda capturar de un solo movimiento.

REQP_9: Posición de Jaque

DESCRIPCIÓN:

Cuando un jugador realice un movimiento que ponga en jaque (amenaza directa) al rey del rival, el oponente deberá realizar obligatoriamente un movimiento para liberar a su rey del jaque.

ENTRADA DE DATOS:

Se debe mover una pieza que comprometa al Rey enemigo, y luego el oponente deberá primero intentar mover una pieza distinta a su Rey sin éxito, para luego seleccionar y poder mover únicamente su Rey a una posición válida.

PROCESAMIENTO DE DATOS:

El sistema debe interpretar que el movimiento del jugador pone en jaque al Rey del oponente y en el turno siguiente, debe anular todos los movimientos que no sean sobre el Rey.

SALIDA DE DATOS:

Se le debe indicar al jugador que su movimiento en un primer intento es invalido, y luego se debe poder realizar el movimiento solicitado al Rey.

RESTRICCIONES:

Se prueba en una jugada con 3 piezas sobre el tablero, donde una es del jugador que hace Jaque y que debe estar a un movimiento de hacerlo, y las otras 2 piezas son el Rey y otra pieza al azar.

REQP_10: Posición de Jaque Mate**DESCRIPCIÓN:**

Si el rey de un jugador se coloca en posición de jaque y no existe algún movimiento que el jugador pueda hacer para sacarlo de allí, entonces el rey está en jaque mate, el juego termina y ese jugador pierde.

ENTRADA DE DATOS:

Se debe seleccionar una pieza y realizar un movimiento estratégico válido a una casilla que logre dejar en Jaque Mate al oponente.

PROCESAMIENTO DE DATOS:

El sistema debe ser capaz de entender la jugada de Jaque Mate al rival y terminar la partida, recolectando todos los datos y dando por victorioso al jugador correspondiente.

SALIDA DE DATOS:

Se debe declarar la partida como terminada y se debe informar del jugador ganador de la sesión de juego.

RESTRICCIONES:

Se prueba con varias fichas del jugador que ataca, y en el equipo contrario solo se debe tener al Rey, donde mediante un solo movimiento estratégico válido se debe dejar en Jaque Mate al Rey, sin opción de movimiento en el turno siguiente.

8.2 Requerimientos para Funcionalidades Adicionales**REQFA_1: Control de contadores de tiempo****DESCRIPCIÓN:**

Se debe garantizar que tanto el contador de duración de partida, el de turno actual, y el tiempo total disponible para cada jugador, inicien y detengan en sus debidos instantes de tiempo correspondientes. Así como existe un contador de tiempo ascendente global de partida, habrá otro contador ascendente que cada jugador dispondrá para guardar su propio conteo de tiempo por turno, y por último un contador que existirá en modos de juego competitivos, donde serán contadores descendentes en el cual el jugador deberá realizar su movimiento con velocidad antes de que finalice el conteo.

ENTRADA DE DATOS:

Se debe iniciar una partida nueva, se debe dejar correr el turno unos segundos y al cabo de un tiempo se debe cambiar de turno, también dejándolo correr un tiempo y finalmente terminando la prueba.

PROCESAMIENTO DE DATOS:

El sistema en primera instancia debe setear los contadores en sus valores correctos por defecto, y luego de dejarlos correr un tiempo se debe comprobar que se ha

transcurrido un tiempo correcto, y luego del cambio de turno se debe comprobar que se cumple nuevamente con los tiempos pero con el rival.

SALIDA DE DATOS:

Se debe esperar que los tiempos de los contadores sean correctos en ambos casos.

RESTRICCIONES:

Se hace cambio de turno sin necesidad de realizar movimientos, para llevar un control únicamente del tiempo transcurrido.

REQFA_2: Control de tiempo para modo de juego competitivo

DESCRIPCIÓN:

En modo competitivo, según el nivel de dificultad, el tiempo disponible restante para cada jugador se debe ir reduciendo y será una variable propia de cada participante, donde al cambiar de turno en un muy corto periodo de tiempo, éste se podrá aumentar en una pequeña proporción a modo de beneficio al jugador.

ENTRADA DE DATOS:

Se debe empezar un turno y ejecutar un movimiento válido, en un tiempo lo suficientemente lento, y luego realizar la misma prueba pero en un tiempo lo suficientemente rápido.

PROCESAMIENTO DE DATOS:

El sistema debe de poder aplicar los beneficios de tiempo en los casos en los que se cumple la condición, que es cuando se efectúa un movimiento lo suficientemente rápido en el turno.

SALIDA DE DATOS:

En la primera ejecución se debe ver que el tiempo total se redujo y en la segunda ejecución el tiempo total se debe haber incrementado un determinado valor.

RESTRICCIONES:

Se realiza la prueba con solo una pieza en el tablero para evitar tiempos de carga innecesarios.

REQFA_3: Respuesta de la AI

DESCRIPCIÓN:

En un modo de juego contra la computadora, se debe garantizar mínimamente que al encontrarse en el turno de la computadora, ésta efectúe un movimiento válido obligatoriamente, donde luego la coherencia del movimiento responderá a la dificultad elegida en el momento previo al inicio de la partida.

ENTRADA DE DATOS:

Se debe elegir un modo de juego, iniciar una partida con movimientos previamente definidos, y luego repetir la prueba con un modo de dificultad superior, e iterar este procedimiento hasta barrer todos los niveles de dificultad.

PROCESAMIENTO DE DATOS:

Se espera que el sistema sea capaz de variar y seleccionar el modo de dificultad, y que a la hora de que la computadora deba realizar un movimiento, estos sean más coherentes a medida que el nivel de dificultad aumenta.

SALIDA DE DATOS:

Luego de un movimiento del jugador, previamente definido el movimiento siguiente de la máquina deberá ser cada vez más efectivo, intentando acercarse a piezas rivales para capturarlas o efectuando jugadas guardadas que sean defensivas.

RESTRICCIONES:

No se encuentran restricciones.

REQFA_4: Dificultades de la AI**DESCRIPCIÓN:**

Cada nivel de dificultad deberá ser incremental en dificultad, haciendo que la computadora ejecute movimientos más “pensados” y no tan aleatorios.

ENTRADA DE DATOS:

Se elegirá el modo de dificultad de la AI en el menú de configuraciones del juego.

PROCESAMIENTO DE DATOS:

Según la dificultad seleccionada. Se seleccionarán los grupos de métodos necesarios para el movimiento de las piezas, dependiendo de la ubicación de las fichas del jugador. Además, se realizará un ajuste de tiempo dependiendo la escala de dificultad (a más dificultad, menos tiempo).

SALIDA DE DATOS:

Configurar la partida con el modo de dificultad seleccionado, antes del inicio de la misma.

RESTRICCIONES:

No se encuentran restricciones.

REQFA_5: Control de sincronización en el juego 1vs1**DESCRIPCIÓN:**

Se debe poder sincronizar los movimientos de cada jugador, para que puedan verse reflejados en el tablero del adversario, luego de haberse realizado el movimiento deseado.

ENTRADA DE DATOS:

Un jugador deberá seleccionar una casilla y realizar un movimiento válido.

PROCESAMIENTO DE DATOS:

El sistema deberá saber a qué jugador otorgarle el poder de ingresar datos y cual deberá quedarse inactivo esperando su turno, además de que una vez efectuado el movimiento, ambos jugadores deberán ver la misma actualización en sus pantallas

SALIDA DE DATOS:

Se deberá corroborar que cada jugador se encuentre visualizando el mismo tablero con las piezas en las mismas posiciones ambos, y que los cambios se actualicen por igual.

RESTRICCIONES:

No se encuentran restricciones.

REQFA_6: Funcionamiento del chat online 1vs1

DESCRIPCIÓN:

Se debe poder enviar y recibir mensajes de manera instantánea mientras se desarrolla una partida en el modo 1vs1, mostrando los mensajes en orden cronológico y siendo éstos inalterables una vez enviados.

ENTRADA DE DATOS:

El mensaje se deberá ingresar en un text box en el menú del chat de la partida multijugador.

PROCESAMIENTO DE DATOS:

El sistema debe procesar la entrada, guardarla en una lista en orden cronológico, y representarlos tanto en la ventana del jugador 1 como en la del jugador 2.

SALIDA DE DATOS:

Se deberían representar los mensajes enviados y recibidos en una ventana aparte a la del tablero con las piezas de la partida en curso.

RESTRICCIONES:

No se encuentran restricciones.

REQFA_7: Funcionamiento del chat online 1vsCOM

DESCRIPCIÓN:

En partidas contra la computadora, el chat debe estar inhabilitado completamente.

ENTRADA DE DATOS:

No debe ser posible ningún tipo de entrada.

PROCESAMIENTO DE DATOS:

Las opciones relacionadas al chat deben ser satisfactoriamente anuladas por el sistema al detectar que se encuentra jugando contra la COM.

SALIDA DE DATOS:

No debe haber ningún tipo de interacción con el usuario.

RESTRICCIONES:

No se encuentran restricciones.

REQFA_8: Recopilación de información de la partida**DESCRIPCIÓN:**

Se debe poder recopilar información sobre todos los movimientos que se realizan durante la partida. Esta información contiene qué jugador realizó el movimiento, de qué casilla inicial a cuál casilla final se desplazó, que pieza se movió, qué tiempo se tardó en el turno, que pieza se capturó en el movimiento, que Peón se coronó y que pieza se le asignó, si hubo Jaque al finalizar el movimiento, etc.

ENTRADA DE DATOS:

Se deberá poder captar y almacenar datos de las partidas, como los tiempos de cada turno, como el tiempo total de juego, así como la cantidad de movimientos o piezas que fueron tomadas por el oponente, etc.

PROCESAMIENTO DE DATOS:

Todas las entradas de datos deben ser almacenadas en listas de objetos para luego ser escritas sobre un documento de texto que pueda ser modificado para almacenar los datos a lo largo del tiempo y luego deberá poder ser leído para recuperar las estadísticas.

SALIDA DE DATOS:

No se espera una salida de datos.

RESTRICCIONES:

No se encuentran restricciones.

8.3 Requerimientos para la UI

En esta sección se dispondrán los requerimientos solicitados para ambas UI gráficas, tanto para la consola en sus versiones más base, como la UI en ventana para las versiones más avanzadas.

8.3.1 Requerimientos para la UI de Consola**REQUIC_1: Visualización de las piezas y el tablero****DESCRIPCIÓN:**

Se deberá poder visualizar correctamente las piezas y el tablero, así como las columnas y filas deben estar unívocamente definidas.

ENTRADA DE DATOS:

No se esperan entradas de datos.

PROCESAMIENTO DE DATOS:

Se desea que el usuario sea capaz de poder reconocer sus piezas propias y las del rival, así como su tipo. Además, en lo que respecta al tablero el usuario deberá asimilar las filas y columnas. Todo esto considerando de que no demore un tiempo mayor al de 5 segundos en interpretar todo y pueda realizar una acción de acuerdo a su estrategia de juego para seleccionar una casilla.

SALIDA DE DATOS:

No se esperan salidas de datos.

RESTRICCIONES:

No se encuentran restricciones.

REQUIC_2: Entrada de datos por consola

DESCRIPCIÓN:

Se deberá poder recibir datos por consola y analizarlos para poder así manejar tanto el menú principal y las configuraciones o diferentes pestañas, como en partida poder ingresar la posición de la pieza a mover, y su casilla final.

ENTRADA DE DATOS:

Se deberá poder ingresar por teclado un comando indicado por pantalla tanto para seleccionar una posición del tablero, como en el menú algún tipo de interacción para ser capaz de realizar una selección.

PROCESAMIENTO DE DATOS:

Una vez que se reciba una entrada, se debería analizar y se espera que el usuario tarde en interpretar y realizar una entrada en menos de 5 segundos y en menos de 2 intentos. Esta entrada deberá ser capaz de ser tomada por el sistema e interpretarla, tanto para moverse por el menú, como para seleccionar posiciones en el tablero.

SALIDA DE DATOS:

Como salida de datos, depende del caso que se esté ejecutando en el momento, pudiendo ser ésta la selección válida de una casilla del tablero o una opción de un menú en particular.

RESTRICCIONES:

No se detectan restricciones.

REQUIC_3: Simpleza de la UI

DESCRIPCIÓN:

La presentación de los datos como el tablero, piezas, instrucciones de juego, o los menús, deberán poder ser interpretados por el usuario en un tiempo menor a 5 segundos (tiempo máximo contemplado para leer y esperar una respuesta) y de manera inequívoca. También, a su vez, se espera que el usuario no realice más de 2 intentos fallidos en lo relacionado a las posibles interacciones con la UI..

ENTRADA DE DATOS:

Cualquier interacción que requiera ingresar datos por teclado.

PROCESAMIENTO DE DATOS:

Se espera un tiempo de respuesta del usuario relativamente corto, de no más de 5 segundos y en caso de ingresar datos erróneos, que no lleve más de 2 intentos fallidos.

SALIDA DE DATOS:

No se esperan salidas de datos.

RESTRICCIONES:

No se detectan restricciones.

8.3.2 Requerimientos para la UI de Ventanas

REQUIV_1: Ventana para el arranque del sistema

DESCRIPCIÓN:

Al iniciar la aplicación se deberá presentar un menú de inicio con las opciones principales, como partida, scoreboard o estadísticas, configuraciones y salir.

ENTRADA DE DATOS:

Inicialización del programa desde su ejecutable.

PROCESAMIENTO DE DATOS:

El programa deberá entregar un menú de opciones para poder seleccionar su previsualización como también poder configurar las estrategias vs COM.

SALIDA DE DATOS:

Se debe mostrar un menú visual.

RESTRICCIONES:

No se detectan restricciones.

REQUIV_2: Ventana de partida

DESCRIPCIÓN:

Al entrar en el apartado de partida, luego de haber configurado a gusto las opciones como el modo de juego, la dificultad, etc, se debe presentar una pantalla en la cual se visualiza el tablero inicializado con sus respectivas fichas y los contadores en sus valores iniciales, según sea cada caso.

ENTRADA DE DATOS:

Selección desde el menú para iniciar la partida.

PROCESAMIENTO DE DATOS:

El programa deberá ser capaz de interpretar la selección de iniciar una partida, además de tomar la configuración determinada previamente para seguidamente instanciar una ventana adicional en la cual se visualizará el tablero con las piezas.

SALIDA DE DATOS:

Se deberá mostrar una ventana aparte en la cual se desplegarán todos los datos que respondan al tablero y las fichas.

RESTRICCIONES:

No se detectan restricciones.

REQUIV_3: Ventana de Scoreboard o estadísticas

DESCRIPCIÓN:

En esta sección se debe visualizar una lista de las últimas partidas jugadas, con opción de ver todos sus datos como cantidad de movimientos, duración en tiempo real, etc, también se debe visualizar los mejores jugadores con sus puntajes correspondientes y debe existir la opción de ordenar los resultados mostrados.

ENTRADA DE DATOS:

El usuario debe ser capaz de realizar una selección sobre el menú en el apartado de estadísticas.

PROCESAMIENTO DE DATOS:

Una vez que el sistema detecte la selección del usuario, deberá ser posible que el programa instancie una ventana aparte en la cual se muestren datos acerca de las puntuaciones previas como también estadísticas de tiempos.

SALIDA DE DATOS:

Debería ser posible ver una ventana aparte a la del menú en la cual se seleccionó la opción de estadísticas, donde se visualicen los datos almacenados.

RESTRICCIONES:

No se detectan restricciones.

REQUIV_4: Ventana de configuración

DESCRIPCIÓN:

En esta sección del menú se deberá visualizar una lista de opciones potencialmente configurables de la aplicación, principalmente para la selección de la estrategia de juego contra la COM.

ENTRADA DE DATOS:

Se debería ser realizable una selección desde el menú, sobre la opción de configuración.

PROCESAMIENTO DE DATOS:

El sistema, deberá captar la selección del usuario, y sobre la misma ventana previa, instanciar las opciones referidas a las configuraciones posibles.

SALIDA DE DATOS:

El menú previo deberá ser capaz de cambiar su instancia para visualizar las opciones de configuración.

RESTRICCIONES:

No se detectan restricciones.

» Requerimientos No Funcionales

REQNF_1: Simpleza de la UI: Deberá ser una interfaz gráfica básica y simple, que muestre los datos puntuales que son necesarios sobre todo para agilizar el desarrollo backend del software de la aplicación y las pruebas del mismo, y no debe enfocarse en la belleza de la presentación del juego. La misma deberá mostrar el tablero y las piezas presentes de acuerdo al formato estándar de vista de las mismas del mundo del ajedrez. Deberá proveer al usuario información con respecto al turno e informar el resultado de sus selecciones. No presenta efectos visuales o auditivos, solo presenta el estado actual del tablero y las piezas. Toda acción debe ser realizable dentro de 2 clicks o 2 inputs por teclado.

REQNF_2: Tiempo para el arranque del sistema: Al iniciar la aplicación la misma deberá ser capaz de inicializarse dentro de un margen de 10 segundos máximo.

REQNF_3: Facilidad de agregar nuevas funcionalidades: El agregado de nuevas funcionalidades a la aplicación no debería requerir de una reestructuración o reescritura del código previamente establecido. Se busca una gran modularización del programa.

REQNF_4: Facilidad de mantenimiento y reparación: La identificación de la fuente de un problema deberá ser rápida y directa, el tipo de problema deberá darnos una clara de idea de que parte del código fuente lo produce según su naturaleza.

REQNF_5: Rapidez de actualización de pantallas: La vista de todos los usuarios deberá actualizarse dentro del lapso de 1 segundo luego de producirse un movimiento.

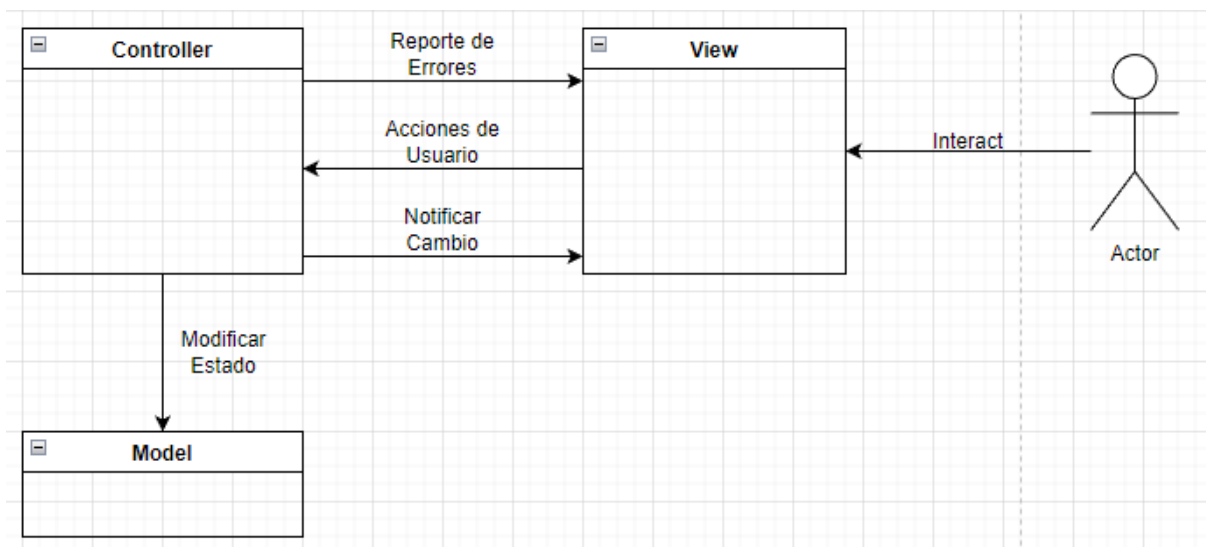
10 Arquitectura Preliminar

En base a los requerimientos definidos hasta el momento hemos decidido adoptar una arquitectura MVC, donde el modelo será el tablero, contaremos con un controlador para realizar la lógica de negocio, y podremos utilizar una vista inicial por consola y en un futuro release desarrollar una vista por ventanas sin necesidad de alterar la lógica del juego.

Esta arquitectura nos permitirá separar la vista del juego de forma que será fácilmente agregar nuevas funcionalidades, y nos permite no atarnos a una sola manera de visualizar el juego.

Además esta arquitectura nos permitirá fácilmente identificar los bugs que se encuentren durante el desarrollo al poder identificar en qué parte de la arquitectura se producen en base a su naturaleza.

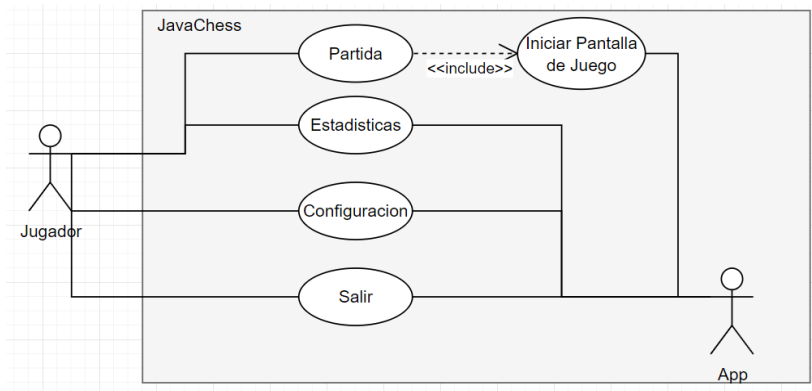
Por último esta arquitectura permite la implementación relativamente sencilla de patrones de diseño, lo que permitirá una rápida actualización de la pantalla en base a la implementación del patrón observer.



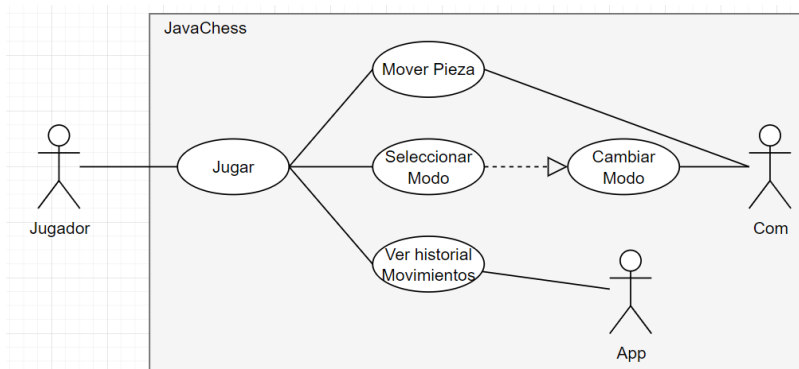
11 Diagramas

DIAGRAMAS DE CASOS DE USO

Iniciar Partida



Partida Local vs Com



Partida Multijugador

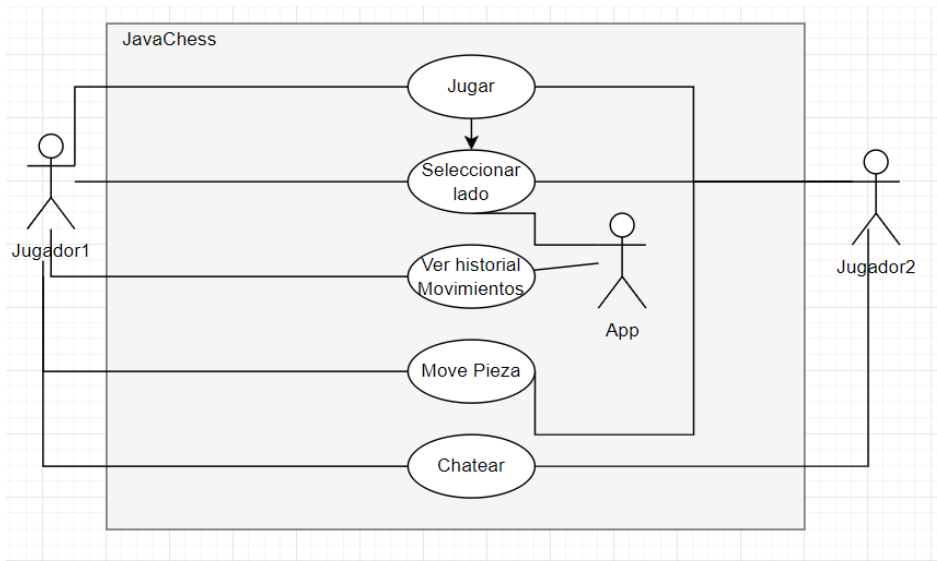


DIAGRAMA DE ACTIVIDADES

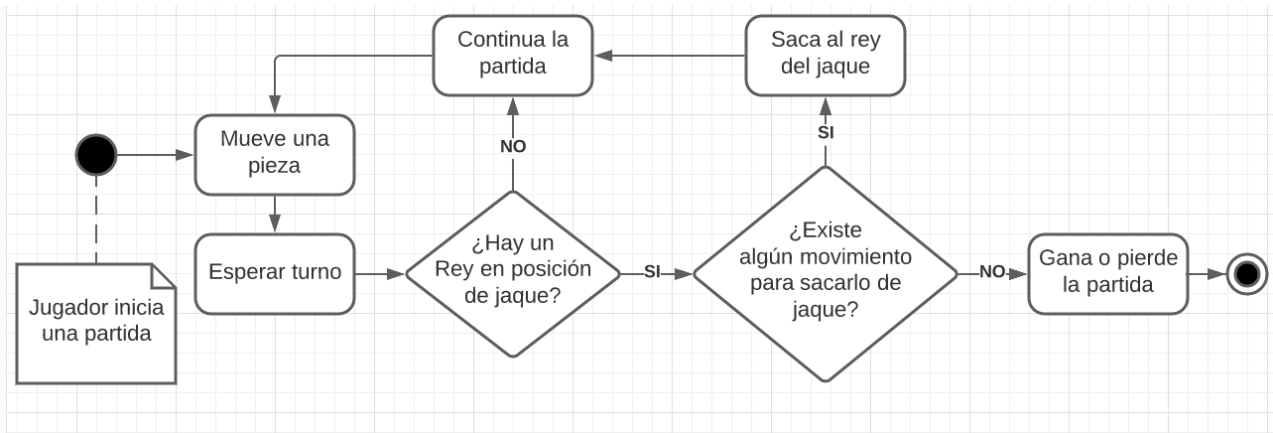


DIAGRAMA DE PAQUETES

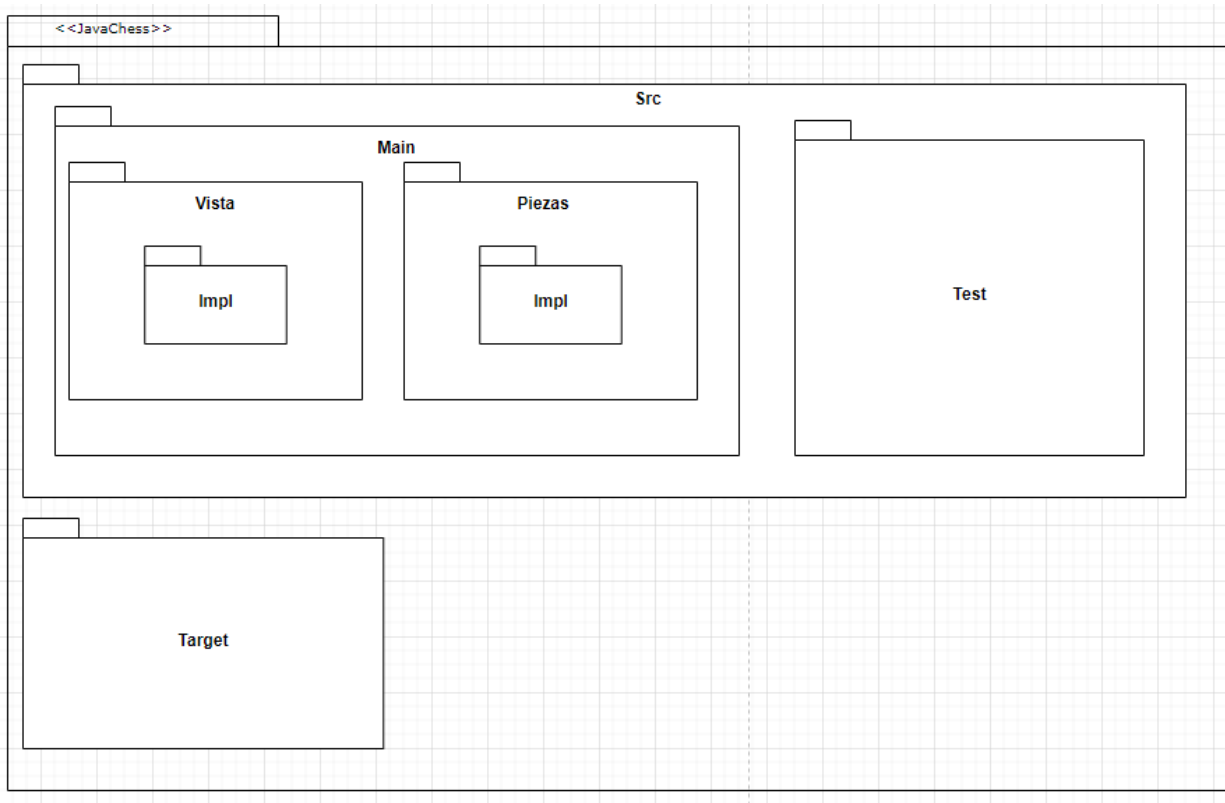


DIAGRAMA DE CLASES

