

Especificación de

Requisitos de Software

Proyecto Taller

2013

Reversi Game

Integrantes:

- Karim Abuzaid, DNI: 35.472.429
- Pablo Marconi, DNI: 33.359.204
- Dario Astorga, DNI: 35.134.346

Índice

Registro de Cambios

1. Introducción

- 1.1. Propósito
- 1.2. Alcance
- 1.3. Definición, acrónimos y abreviaciones
- 1.4. Glosario
- 1.5. Referencias
- 1.6. Personal involucrado
- 1.7. Visión global

2. Descripción global

- 2.1. Perspectiva del producto
- 2.2. Funciones del producto
- 2.3. Características del usuario
 - 2.3.1. Perfil de usuario
 - 2.3.2. Jerarquía de usuario
- 2.4. Restricciones generales
 - 2.4.1. Políticas reguladoras
 - 2.4.2. Limitaciones de hardware
 - 2.4.3. Interfaces con otras aplicaciones
 - 2.4.4. Funcionamiento paralelo
 - 2.4.5. Funcionamiento de auditoría
 - 2.4.6. Funciones de control
 - 2.4.7. Requisitos del lenguaje
 - 2.4.8. Protocolos señalados
 - 2.4.9. Credibilidad de la aplicación
 - 2.4.10. Consideraciones de seguridad
- 2.5. Suposiciones y dependencias
- 2.6. Prorratear los requisitos

3. Requerimientos específicos

- 3.1. Requerimientos de interfaces externas
- 3.2. Requerimientos funcionales
 - 3.2.1. Inicio un juego nuevo.
 - 3.2.2. Cambio de vista.
 - 3.2.3. Guardar partida.
 - 3.2.4. Abrir partida.
 - 3.2.5. Realizar jugada.
 - 3.2.6. Guardar puntaje

Registro de cambios:

Fecha	Autor	Versión	Comentario	# Horas
13/09/2020	Grupo	v 0.1	-	2
16/09/2013	Grupo	v 0.2	-	2
24/09/2013	Grupo	V0.3	-	3
03/11/2013	Grupo	V0.4	-	4

1. Introducción

1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación JAVA que permitirá a un usuario jugar al Reversi.

1.2. Alcance

Diseño, desarrollo e Implementación del juego Reversi.

Para desarrollar este juego se utilizará el lenguaje JAVA, utilizando el patrón arquitectural MVC (modelo-vista-controlador), y el patron Observer, implementando las funcionalidades requeridas para la funcionalidad del juego, además permitirá al usuario alternar el juego entre dos vistas.

1.3. Definiciones, acrónimos y abreviaciones

MVC: modelo vista controlador

1.4. Glosario

Usuario: Persona.

Java: Es un lenguaje de programación orientado a objetos. Que permite el desarrollo de una aplicación para algún uso específico.

MVC: Modelo vista controlador es un patrón de arquitectura de software que separa los datos y la lógica de negocios de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

1.5. Referencias

- IEEE 830 esp - http://www.ctr.unican.es/asignaturas/is1/IEEE830_esp.pdf
- SRS para el desarrollo del SIS-WEB de la Universidad del Valle - http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:pis:ejemplo_de_especificacion_de_requerimientos_-_para_sesion_9.pdf

1.6. Personal involucrado

Nombre	Pablo Marconi
Rol	Diseño y desarrollo
Responsabilidades	

Nombre	Karim Abuzaid
Rol	Diseño y desarrollo
Responsabilidades	

Nombre	Dario Astorga
Rol	Diseño y desarrollo
Responsabilidades	

1.7. Visión Global

En el presente documento se encontrará la información acerca de las características del producto de software, interacción del usuario, vistas del sistema, comportamiento del juego, descripción de los requerimientos funcionales, no funcionales y del sistema, los cuales se representarán mediante el siguiente formato:

Caso de uso: Nombre del caso de uso Precondición: - Poscondición: - Descripción: Breve descripción del caso de uso	
USUARIO	SISTEMA
Acciones que realiza el usuario	Respuesta a las acciones del usuario
FLUJO DE EVENTOS ALTERNATIVO	

2 Descripción global

2.1 Perspectiva del producto

El juego a desarrollar será una aplicación de escritorio.

2.2 Funciones del producto

La aplicación permitirá realizar las siguientes funciones:

- Ofrecer al usuario la funcionalidad del juego Reversi.

2.3. Características del usuario

La aplicación será ejecutada por un usuario sin necesidad de registro. Solamente se le consultará al usuario su nombre de usuario en caso de querer almacenar su puntaje al haber finalizado la partida.

2.3.1. Perfil del usuario

La aplicación será usada por un usuario de escritorio sin necesidad de registro.

2.3.2. Jerarquía de usuario

No es condición del Proyecto

2.4. Restricciones generales

No es condición del Proyecto

2.4.1 Políticas Reguladoras

La aplicación se desarrollará mediante software de licencia abierta, con el lenguaje de programación JAVA, JUnit para la automatización del testing y por lo tanto la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

2.4.2 Limitaciones de Hardware

Para esta aplicación será necesario un computador con una máquina virtual java.

2.4.3 Interfaces con otras aplicaciones

Debido a que la aplicación no interactúa con otros sistemas y es autónomo no se desarrollan interfaces con otras aplicaciones o sistemas.

2.4.4 Funcionamiento Paralelo

No es condición del Proyecto

2.4.5 Funciones de Auditoría

No es condición del Proyecto

2.4.6 Funciones de Control

El usuario no tendrá reestricciones con respecto a la jugabilidad siempre que se respeten las reglas del juego.

2.4.7 Requisitos del Lenguaje

Todo el material que se realice para el usuario se encuentra en Español. La aplicación propiamente dicha en Ingles.

2.4.8 Protocolos Señalados

No es condición del proyecto.

2.4.9 Credibilidad de la aplicación

Para garantizar una buena credibilidad el sistema deberá ser sometido a una serie de pruebas para establecer que se encuentra acorde a los requerimientos que se plasman en el documento en tanto a la consistencia de datos como al rendimiento de la aplicación, tales como tiempos de respuesta.

2.4.10 Consideraciones de seguridad

No es condición del proyecto.

2.5 Suposiciones y dependencias

No es condición del proyecto.

2.6 Prorratear los Requisitos

No es condición del proyecto.

3.1 Requerimientos de interfaces externas

La aplicación no tendrá interconexión con otros sistemas de información, por lo tanto no es necesaria la utilización de interfaz externa alguna.

3.2 Requerimientos funcionales

3.2.1 Inicio un juego nuevo.

Caso de uso: Inicia de juego nuevo. Pre condición: True Pos condición: Se inicio un juego nuevo Descripción: Inicia un juego nuevo.	
Usuario	Sistema
1. Desea crear un juego nuevo. 2. Presiona sobre el boton NewGame. 4. Selecciona pieza con la que desea jugar y luego presiona Start.!	3. Muestra por pantalla ventana emergente para la selección de una pieza. 5. Realiza los cambios necesarios para la creacion de un juego nuevo.
FLUJOS DE EVENTOS ALTERNATIVOS	
Nota: en cualquier momento el usuario puede cancelar la ejecución del caso de uso.	

3.2.2 Cambiar vista.

Caso de uso: Cambiar vista Pre condición: Existe una vista a la cual cambiar. Pos condición: Cambia la vista del juego en ejecución. Descripción: Se cambia la vista de la aplicacion por otra.	
Usuario	Sistema
1. El usuario se dirige a la barra de menu Game, luego luego presiona sobre la opcion Skins.	2. Verifica que el cambio pedido no es la vista en ejecucion. 3. Realiza los cambios en la vista.
FLUJOS DE EVENTOS ALTERNATIVOS	

2.1 Si se confirma la vista que ya esta en ejecucion, no se realizan cambios. Fin de caso de uso.
Nota: en cualquier momento el usuario puede cancelar la ejecución del caso de uso.

3.2.3 Guardar partida

Caso de uso: Guardar partida

Precondición: Existe una partida ejecutada que no ha sido finalizada.

Poscondición: Se guardó la partida actual existente.

Descripción: Se guarda la partida en ejecucion no finalizada en un archivo.

Usuario	Sistema
1. Se dirige a la barra de menu Game y presiona sobre la opcion SaveGame.	3. Guarda la partida actual en un archivo.
FLUJOS DE EVENTOS ALTERNATIVOS	
Nota: en cualquier momento el usuario puede cancelar la ejecución del caso de uso.	

3.2.4 Abrir partida

Caso de uso: Abrir partida

Precondición: Existe una partida guardada a ser abierta.

Poscondición: Partida cargada

Descripción: Se abre una partida guardada en un archivo.

Usuario	Sistema
1. Se dirige a la barra de menu Game y presiona sobre la opcion LoadGame.	2. Carga la partida. 3. Visualizacion de la partida cargada.
FLUJOS DE EVENTOS ALTERNATIVOS	
3.1 E el caso de que no halla ninguna partida guarda anteriormente por el usuario, carga la partida por defecto que el sistema tiene guardada. Nota: en cualquier momento el usuario puede cancelar la ejecución del caso de uso.	

3.2.5 Realizar jugada

Caso de uso: Realizar jugada Pre condición: Existe una jugada a ser realizada por el usuario y es su turno de juego. Pos condición: Se realizó la jugada en la partida exitosamente. Descripción: Realizar jugada	
Usuario	Sistema
2. Usuario elige que jugada realizar. 4. Realiza la jugada.	1. Verifica si el jugador tiene jugadas posibles. 3. Verifica si la jugada es posible. 5. Verifica si el jugador tiene fichas. 6. Realiza los cambios de la jugada en el juego, y visualiza por pantalla tablero con jugada realizada correctamente.
FLUJOS DE EVENTOS ALTERNATIVOS	
3.1 Si el jugador no tiene jugadas posibles, termina el juego ganando el otro jugador. Fin de caso de uso. 3.1 Si la jugada no es posible, no se efectua ningun cambio en el tablero. Fin de caso de uso. 5.1 Si el jugador no tiene mas fichas pasa su turno, termina el juego ganando el otro jugador. Fin de caso de uso. Nota: en cualquier momento el usuario puede cancelar la ejecución del caso de uso.	

3.2.6 Guardar puntaje

Caso de uso: Guarda el puntaje Precondición: Existe una partida finalizada. Poscondición: Puntaje nuevo guardado. Descripción: Se guarda el puntaje con un nombre ingresado por el usuario.	
Usuario	Sistema
1. Finalizo la partida. 3. Ingresa el nombre con el que desea almacenar el puntaje. 4. Presiona en boton Submit que posee la	2. Muestra ventana emergente llamada New Winner con el puntaje obtenido y un espacio para ingresar un nombre.

ventanilla.	5. Guarda el puntaje con determinado nombre ingresado por el usuario.
FLUJOS DE EVENTOS ALTERNATIVOS	
Nota: en cualquier momento el usuario puede cancelar la ejecución del caso de uso.	