

Instituto tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Campus tecnológico Central Cartago

Programación Orientada A Objetos

Grupo 02

Profesor: William Mata Rodríguez

Proyecto #2

Futoshiki

Elaborado por:

Kendall Rodríguez Camacho c. 2022049438

Jocsan Adriel Pérez Coto c. 2022437948

Fecha de entrega: Domingo 27 de noviembre del 2022

II Semestre 2022

Contenido

Enunciado	3
Temas investigados	6
Abrir archivo PDF	6
Archivos XML	6
El patrón de arquitectura Modelo/Vista/Controlador	6
Conclusiones	8
Lista de revisión y análisis de resultados	9
Diagrama de caso de uso	10



Enunciado

En este juego hay que llenar con dígitos las casillas de una cuadrícula de tal forma que cada dígito no se repita ni en la fila ni en la columna a la que pertenece y que los dígitos cumplan con restricciones de desigualdad: mayor que ($>$) o menor que ($<$). Las restricciones de desigualdad pueden estar a nivel de filas (entre dos casillas horizontales) o a nivel de columnas (entre dos casillas verticales). La cuadrícula debe tener la misma cantidad de filas y de columnas: 5x5.

Opciones del programa:

- A) Jugar: Esta opción permite jugar el Futoshiki con un tamaño de cuadrícula 5x5. Cuando se da esta opción se muestra una ventana considerando la configuración (nivel del juego, uso del reloj, posición del panel de dígitos). El programa tiene una serie de partidas que previamente han sido registradas y de ahí selecciona aleatoriamente una partida según el nivel de dificultad configurado.

Botones para presionar:

-  Iniciar Juego: Cuando el jugador pica este botón se inicia el juego. Una vez que se inicia el juego, el jugador selecciona un dígito picándolo en el panel de dígitos (círculos a la derecha de la partida) y luego pica en la casilla de la cuadrícula en donde quiere ponerlo. Cuando pica el dígito se marca su círculo con un color verde como se muestra en el ejemplo. El dígito permanece marcado, disponible para ponerlo en otra casilla. Si quiere seleccionar otro dígito lo pica y el dígito anterior se desmarca. Cuando el jugador pone un dígito se deben hacer las validaciones para que la jugada cumpla con las reglas del juego, de lo contrario se le envía algún mensaje. El juego termina cuando el jugador llena todas las casillas de la cuadrícula de forma correcta.
-  Borrar Jugada: Elimina la última jugada dejando la casilla vacía.

- ✚ Rehacer Jugada: Este botón se puede usar inmediatamente después de haber seleccionado el botón de “BORRAR JUGADA”. Rehace o reconstruye la última jugada que se borró.
- ✚ Terminar Juego: Cuando el jugador selecciona esta opción se le pregunta ¿ESTÁ SEGURO DE TERMINAR EL JUEGO (SI o NO)? Si responde SI termina de inmediato el juego y se vuelve a mostrar otro juego como si estuviera entrando a la opción de Jugar.
- ✚ Borrar Juego: Cuando el jugador selecciona esta opción se le pregunta ¿ESTÁ SEGURO DE BORRAR EL JUEGO (SI o NO)? Si responde SI vuelve a la opción de Jugar usando la misma partida pero eliminando todas las jugadas que hizo. Si responde NO sigue jugando con el mismo juego.
- ✚ Top 10: Despliega una sola ventana con los registros de los mejores 10 primeros jugadores por cada nivel: aquellos que hicieron menos tiempo para completar el juego.
- ✚ Guardar Juego: Guarda en el archivo “futoshiki2022juegoactual.dat” todo el estado del juego actual: configuración, cuadrícula, nombre del jugador, etc. El objetivo es que el jugador pueda en cualquier momento guardar el juego y posteriormente continuarlo en el punto donde hizo el guardado del juego. Este archivo solo va a contener una partida. En caso de que haya una partida en el archivo, se borra y se guarda la del momento.
- ✚ Cargar Juego: Trae del archivo “futoshiki2022juegoactual.dat” el juego que fue guardado y lo pone en la ventana como el juego actual. El juego continúa cuando el jugador usa el botón de INICIAR JUEGO.
- ✚ Posibles Jugadas: Se da la oportunidad al jugador de ver las posibles jugadas que posee un cuadro de la cuadrícula

B) Configurar: Esta opción es para indicar las condiciones con que se va a jugar. Contiene los siguientes datos:

- ✓ Nivel
- ✓ Reloj

- ✓ Posición del panel de dígitos
 - ✓ Tamaño de la cuadrícula
- C) Ayuda: Esta opción despliega el manual de usuario
- D) Acerca de: Esta opción despliega la información “Acerca del programa” donde pondremos al menos los datos del nombre del programa, la versión, la fecha de creación y el autor intelectual del producto.
- E) Salir: Esta opción se usa para salirse del programa.

Temas investigados

Abrir archivo PDF

Para la elaboración de este proyecto se tuvo que realizar una investigación para averiguar como se podía abrir o desplegar un archivo PDF en Java. Luego de realizar dicha investigación se obtuvo la respuesta a dicho problema. El algoritmo utilizado fue crear un objeto de la clase FILE y llamar al método constructor con la ruta y nombre del archivo PDF. Luego se utilizó el método `getDesktop()` para abrir el archivo.

Archivos XML

XML o Lenguaje de Marcado Extensible es un formato de texto que se utiliza para almacenar e intercambiar datos estructurados, bien sea que se trate de documentos, configuraciones, transacciones o simplemente datos. Es un lenguaje de marcado que define la estructura y el significado de los datos.

Los archivos XML se componen de etiquetas que aportan datos e información que se desea procesar. Estas etiquetas pueden estar de forma individual o anidadas.

Habitualmente un fichero XML incluye mucha información y debe de ser procesada correctamente por el usuario, en este caso el desarrollador. Cuanto más grande sea un fichero XML estará indicando que más información trae.

Este tipo de archivos siempre contendrá una única etiqueta dando la introducción a dicho fichero, y a partir de ella se podrá crear cualquier tipo de archivo XML. Dentro de estos archivos XML habrá una o varias etiquetas, que a su vez tendrán otras **etiquetas**, que podrán estar o no anidadas.

El patrón de arquitectura Modelo/Vista/Controlador

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

- ✚ El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- ✚ La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.
- ✚ El Controlador es el objeto que proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Conclusiones

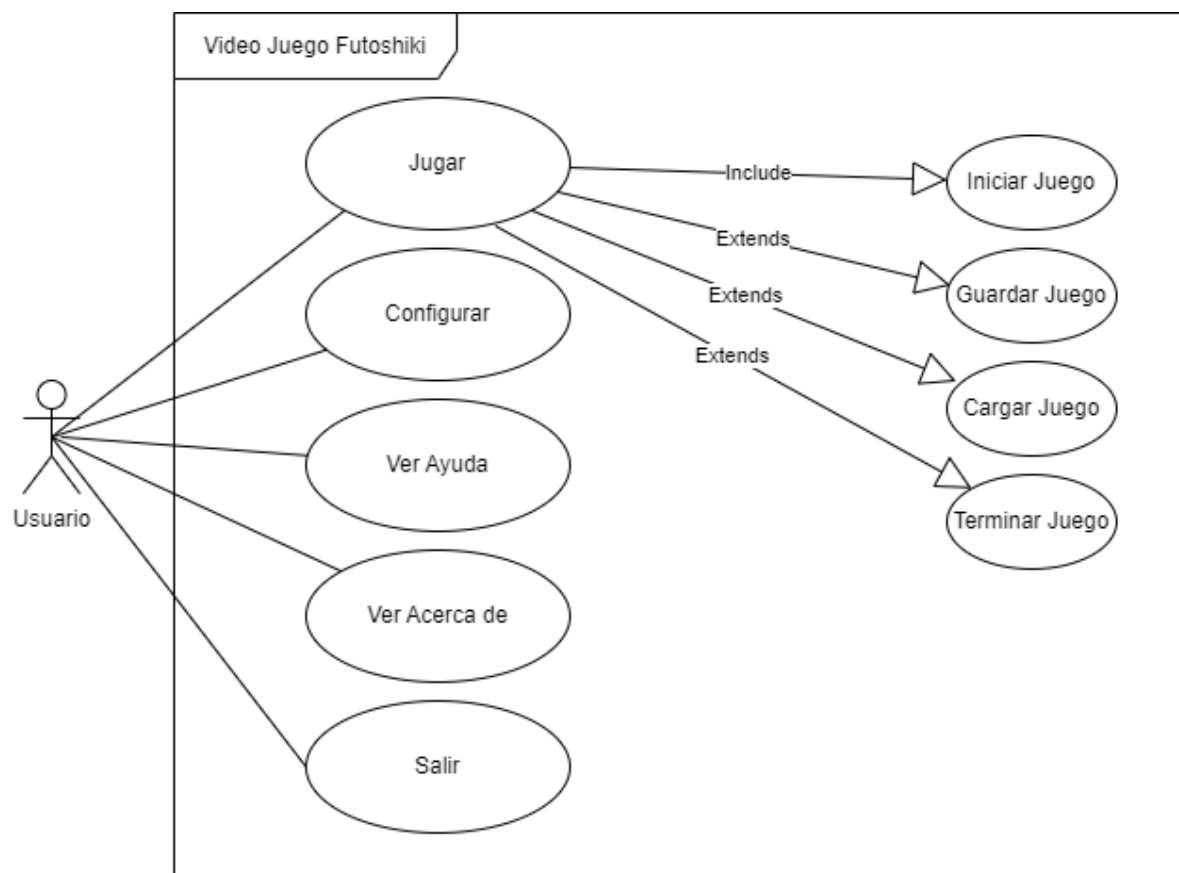
El trabajo realizado fue un éxito, se pudieron resolver la mayoría de las indicaciones que el proyecto indicaba. Algunos inconvenientes que se encontraron en la elaboración del programa, pero se logró resolverlos investigando sobre el tema.

La elaboración de este trabajo fue de gran importancia para apañar y reforzar nuestros conocimientos sobre las interfaces gráficas en el IDE de NetBeans, así como el uso de clases en java, Además, gracias a este programa mejoramos considerablemente nuestros pensamientos lógicos al realizar algunos algoritmos que en un pasado no lográbamos resolver y que ahora los ejecutamos con gran facilidad.

Lista de revisión y análisis de resultados

Concepto	Puntos Originales	Avance 100%/0	Puntos Obtenidos	Análisis de resultados
Juegos multinivel	18	100%	18	100: Totalmente desarrollado
Posibles jugadas para una casilla	12	100%	12	100: Totalmente desarrollado
Cambios: Opción Configurar Ventana del juego Grabar juego Cargar juego Archivo Partidas del juego	10 10 5 5 5	100%	10 10 5 5 5	100: Totalmente desarrollado
Aplicación correcta del patrón de arquitectura MVC	25	100%	25	100: Totalmente desarrollado
Ayuda	10	100%	10	100: Totalmente desarrollado
TOTAL	100	100%	100	
Partes desarrolladas adicionalmente				

Diagrama de caso de uso



Nombre del caso de uso	<i>Jugar</i>
Actor participante	Llamado por el <i>Usuario</i> , se comunica con <i>JUEGO</i> , <i>Cuadro</i> , <i>CargarPartida</i> , <i>GuardarPartida</i> , <i>Partida</i> , y <i>FuncionesListaEnlazada</i> .
Condición inicial	1. El botón <i>Jugar</i> activa la interfaz donde se ve el juego y esta va a responder según las indicaciones que se envíen en los botones asignados.
Flujo de eventos	<p>2. botón <i>Iniciar Juego</i>, se activa cuando detecta que se ingresó un nombre en la casilla que almacena el nombre del jugador, debajo de <i>Nombre del Jugador</i>, luego no permite cambiarse el nombre y empieza el cronometro (este cambia si se configura anteriormente), y activa los botones de la interfaz para poder usarlos.</p> <p>3. botón <i>Borrar Jugada</i>, borra las jugadas realizadas dentro del cuadro por el <i>Usuario</i></p> <p>4. botón <i>Rehacer Jugada</i>, rehace las jugadas que fueron borradas, solamente tiene un uso luego de haber borrado la jugada.</p> <p>5. botón <i>Top 10</i>, muestra el top 10 de los mejores tiempos, en cada dificultad, de cada usuario que terminó el juego.</p> <p>6. botón <i>Salir</i>, vuelve al menú principal.</p> <p>7. botón <i>Cargar Juego</i>, se carga el juego que fue guardado con anterioridad, se cargan todas las características del juego guardado</p>

8. botón *Guardar Juego*, guarda todas las características de la partida, nombre, tiempo y números escritos, así como las desigualdades y la dificultad.

9. botón *Terminar Juego*, termina el juego actual y busca otro con las mismas características, ya sea dificultad y cronometro, de esta manera se puede iniciar otra partida nuevamente.

10. botón *Borrar Juego*, borra todas las jugadas que se realizaron en el juego actual (el cronometro y el nombre no se borran), de esta manera podemos iniciar de nuevo, pero en la misma partida.

11. botón *Posibles Jugadas*, luego de darle click a este botón se debe de darle click al botón donde se quiera mostrar las jugadas posibles y de esta manera se muestra una pequeña ventana con las jugadas.

12. botones *1,2,3,4,5*, se utilizan para ponerlos dentro de las casillas, se presionan primero y luego se ingresa en la casilla deseada.

Condición de salida

13. El botón salir recibe la acción de salir de la interfaz y vuelve al menú principal. También el juego termina cuando se completa el nivel o los niveles y así mismo si se termina el tiempo.

Requerimientos especiales

En este caso, cada partida necesitaría configuraciones necesarias, por lo que, si al darle al botón de *Jugar* no hay configuraciones, se inicia con características predeterminadas

Nombre del caso de uso	<i>Configuración</i>
Actor participante	Llamado por el <i>Usuario</i> , se comunica con <i>JUEGO</i>
Condición inicial	1. El botón <i>Configuración</i> activa la interfaz donde se ve las maneras de configurar nuestra siguiente partida.
Flujo de eventos	2. <i>Marcar botón Nivel</i> , en este se debe de marcar la dificultad en la que se desea tener el juego. 3. <i>Marcar botón Reloj</i> , en este se debe de marcar el tipo de cronometro que se desea tener en el juego. 4. <i>Marcar botón Posición en la pantalla del panel de dígitos</i> , con este cambiamos la posición donde se van a ubicar los botones del juego. 5. botones <i>Horas, minutos, segundo</i> , este se utiliza si la opción de timer fue seleccionada, se debe de ingresar la cantidad de horas, minutos y segundos que se desea en el juego.
Condición de salida	6. botón <i>Atrás</i> , vuelve al menú principal con las configuraciones que se seleccionaron.
Requerimientos especiales	En este caso, si las configuraciones no se ingresan nada, se van a quedar las configuraciones predeterminadas por los creadores.

Nombre del caso de uso	<i>Ayuda</i>
Actor participante	Llamado por el <i>Usuario</i> , se comunica con <i>La ayuda del juego</i>
Condición inicial	1. El botón <i>Ayuda</i> activa la interfaz donde se ve el manual de usuario ya creado
Flujo de eventos	La interfaz donde se muestra el manual de usuario.
Condición de salida	2. botón <i>Atrás</i> , vuelve al menú principal
Requerimientos especiales	
Nombre del caso de uso	<i>Acerca de</i>
Actor participante	Llamado por el <i>Usuario</i> , se comunica con <i>La interfaz AcercaDe.</i>
Condición inicial	1. El botón <i>Acercade</i> activa la interfaz donde se la información acerca del programa creado
Flujo de eventos	La interfaz donde se muestra la información.
Condición de salida	2. botón <i>Atrás</i> , vuelve al menú principal
Requerimientos especiales	
Nombre del caso de uso	<i>Salir</i>
Actor participante	Llamado por el <i>Usuario</i>
Condición inicial	1. El botón <i>Salir</i> termina el programa.
Flujo de eventos	
Condición de salida	
Requerimientos especiales	