
CARTA DE PRESENTACIÓN

Laberinto ERA

UNIVERSIDAD CENTRAL DEL ECUADOR



FACULTAD DE INGENIERÍA CIENCIAS FÍSICAS Y MATEMÁTICA

CARRERA DE SISTEMAS DE LA INFORMACIÓN

GRUPO DE DESARROLLO DE SOFTWARE

ALGORITMOS

**ERICK ENRIQUE DÍAZ PASTAZ
ROYER FABIAN MORENO MIRANDA
ANDY JOEL LLUMIQUINGA VASCO**

2020

RESUMEN EJECUTIVO.

Para poder resolver y poder crear este aplicativo, se necesitó de la ayuda de material extra, tanto videos como blogs informativos, que nos dio suficiente información para la creación de este juego. Al ser un programa realizado completamente en Java, no se necesitó de demasiado tiempo para utilizar el código. Pero se necesitó de tiempo para poder encontrar librerías que fueran de utilidad, para simular la creación y generación del laberinto para simular una interfaz más amigable que reaccione a las acciones del usuario.

Hemos llegado a la conclusión de que tanto la planificación, realización y uso de este juego, es muy práctica y conveniente, ya que se puede utilizar la teoría de la materia estudiada no solo para resolver problemas de eficiencia de algoritmos, sino que también se puede utilizar para generar un divertido juego que sea del agrado del usuario.

Recomendamos realizar este juego con ya conocimientos sobre algoritmos y técnicas de programación avanzadas, ya que, para la elaboración de este juego del Laberinto, se requiere que el aplicativo sea lo más eficaz y rápido posible al momento de mostrar las rutas posibles para resolver el laberinto, ya que con métodos de programación normales se va a llegar a una solución de una manera no tan eficaz o rápida.

ÍNDICE.

| | |
|--|---|
| RESUMEN EJECUTIVO..... | 2 |
| ÍNDICE. | 3 |
| 1. INTRODUCCIÓN..... | 4 |
| 1.1 Objetivo General. | 4 |
| 1.2 Objetivos específicos. | 4 |
| 2. METODOLOGÍA..... | 4 |
| 3. DESCRIPCIÓN DEL PROBLEMA..... | 5 |
| 3.1 Antecedentes | 5 |
| 3.2 Detalles..... | 5 |
| 3.3 Justificación | 5 |
| 3.4 Beneficios del Sistema | 5 |
| 4. FACTIBILIDAD. | 6 |
| 4.1 Técnica. | 6 |
| 4.2 Operacional. | 6 |
| 5. ANÁLISIS. | 6 |
| 5.1 Requerimientos para el sistema | 6 |
| 6. RECOMENDACIONES. | 6 |
| 7. CONCLUSIONES. | 7 |

1. INTRODUCCIÓN.

1.1 Objetivo General.

El objetivo principal de este proyecto es crear una solución para el problema de resolver un laberinto mediante los algoritmos estudiados en el presente semestre. Entender el funcionamiento de cada uno de estos algoritmos para encontrar una solución óptima y aplicarlos al programa que se desarrollará mediante el lenguaje de programación java.

1.2 Objetivos específicos.

- Desarrollar un aplicativo que simule un juego de un laberinto a partir de una matriz de $n \times n$ (2-200) con distintas dificultades dónde un jugador deba llegar desde un punto de salida hasta un punto de llegada.
- Demostrar cómo se aplica los algoritmos vistos en el curso para poder crear un juego dónde se puede entretener a un usuario, además de mostrar la interacción que se da entre este y la máquina.
- Analizar los algoritmos vistos en clases para así poder determinar cuál algoritmo es el indicado para poder aplicarlo a la resolución del problema.

2. METODOLOGÍA.

La metodología que se usó dentro del lenguaje Java fue Programación Orientada a Objetos, debido a que según nuestro criterio era la óptima que podíamos implementar para facilitar la creación de código, por la facilidad que se nos da para la reutilización, ahorrando tiempo y líneas de código.

Y la investigación aplicada fue utilizada, tanto para practicar los conocimientos que íbamos adquiriendo durante el transcurso de creación del aplicativo.

3. DESCRIPCIÓN DEL PROBLEMA

3.1 Antecedentes

Se decide crear el aplicativo con la orientación hacia un juego, ya que para hacer más entretenido el tema que se va a tratar se ve conveniente que el usuario pueda encontrar un aplicativo divertido y que le llame la atención al usarlo y sea fácil de manejar. Ya que el tema es encontrar la solución a un laberinto, se ve muy provechoso crear el aplicativo orientado a que el usuario sea un jugador que trate de descifrar el camino y en el caso de que no pueda, se presentará la solución mediante las técnicas que se aplicarán para resolver el problema.

3.2 Detalles

Resolver un laberinto puede ser tan fácil, así como también tan difícil dependiendo de el grado de dificultad que tenga éste. Si tomamos en cuenta un laberinto representado por una matriz $n \times n$, mientras más grandes sean las dimensiones que tenga el laberinto más complicado será encontrar la solución ya que existirán más paredes y más caminos erróneos que el jugador podrá tomar.

3.3 Justificación

Se realizará el aplicativo para poder entender sobre los algoritmos estudiados en todo el semestre y a su vez poder aplicarlos a un programa que en nuestro caso es un juego.

3.4 Beneficios del Sistema

3.4.1 Beneficios Tangibles

Al realizarse el programa en lenguaje abierto, se asegura un mínimo coste en la realización del aplicativo, y para maquinas que realicen el proceso, se utilizará aquellas que sean lo suficientemente potentes para soportar las funciones implementadas.

3.4.2 Beneficios intangibles

Al ser desarrollado la aplicación en Java, nos aseguramos de que sea multiplataforma, es decir, que trabaja en cualquier sistema operativo disponible, por lo que reduciría el costo.

4. FACTIBILIDAD.

4.1 Técnica.

El aplicativo funciona como un juego en el que usuario ingresará por teclado una dimensión entre 2 y 200, la cual será la dimensión que tendrá el laberinto, mientras mayor sea el número más grande será el laberinto y por ende será más complicado de resolver. El jugador tendrá la opción de tratar de resolverlo o solicitar al juego la respuesta en caso de no encontrar el camino de solución.

4.2 Operacional.

El juego será capaz de ejecutarse en el sistema operativo Windows 10 y posteriores, siempre y cuando tengan los requisitos básicos los cuales son: 4 GB de memoria RAM y por lo menos 50 Mb de almacenamiento disponible.

5. ANÁLISIS.

5.1 Requerimientos para el sistema

El aplicativo va a poder crear y generar laberintos mediante una matriz de orden $n \times n$ la cual será la base para crear mediante cuadrados que serían las posiciones de la matriz. EL orden de la matriz será un valor ingresado por teclado. Además, el aplicativo será capaz de ser jugable, es decir, que el usuario debe encontrar un camino solución desde un punto de salida hasta un punto de llegada generado por el laberinto. Si el usuario es capaz de resolver el laberinto, se dará por ganador y en caso de no poder encontrar una solución, el juego deberá ser capaz de indicar cual sería el camino solución al usuario y declararlo perdedor del juego.

6. RECOMENDACIONES.

Se recomienda que el aplicativo sea llamativo para el usuario y fácil de manejar para evitar confusiones al momento de ejecutarlo, ya que varios usuarios suelen aburrirse o rechazar un aplicativo que sea complejo de manejar.

El juego debe ser capaz de tener un tiempo de respuesta rápido al momento de tener interacción con el usuario.

Generar el camino solución debe ser instantáneo, para lo cual se deberá elegir el correcto algoritmo que satisfaga este problema. Teniendo el correcto algoritmo, se garantiza que el tiempo de respuesta será el óptimo.

7. CONCLUSIONES.

Llegamos a la conclusión de que existen múltiples librerías que se pueden utilizar en Java para mejorar la experiencia entre la interacción Usuario-Máquina.

Concluimos que se puede mejorar la manera de reutilizar el código en Java que para futuros proyectos nos será de ayuda para evitar tantas líneas de código en el programa.

Se llegó a la conclusión de que un programa no simplemente se desarrolla, si no, debe pasar por múltiples procesos para poder ser validado y poder llegar a utilizarse de manera normal para el campo que ha sido diseñado.

Por último, se concluye que los algoritmos estudiados en el curso son de mucha utilidad para poder resolver problemas tan simples, así como también problemas muy complejos.