

Proyecto 1: Laberinto

Estudiantes

Geovanni González Aguilar

Daryll Martínez Rodríguez

Escuela de Ingeniería en Computación

Curso

IC3002 - Análisis de Algoritmos

Profesor

Joss Pecou Johnson

Grupo

60

II Semestre 2024

Introducción

El software desarrollado, consiste en un laberinto con un enfoque en backtracking, en el cual se creará un mapa de un tamaño de 10x10 hasta 50x50 (elegido por el usuario), en el cual se podrá iniciar desde cualquier parte del mapa y este se podrá solucionar de 2 maneras distintas, cada una con un enfoque personalizado al tipo de algoritmo que se usa, uno para buscar el camino más optimo de todos el cual tiene como nombre “Optimización”, el otro se encargará de buscar todos los caminos posibles sin lógicas que lo ayuden a descartar posibles rutas, el nombre de este enfoque es “Fuerza Bruta”.

Manual De Usuario

Crear un laberinto:

Laberinto

Generar Laberinto

Tamaño del laberinto: 10

Fuerza bruta

Resolver Laberinto

Cargar Laberinto

Guardar Laberinto

En este apartado se creará el mapa, en el cual se escogen las dimensiones desplazando el punto azul sobre su eje, estas irán desde 10x10 siendo el menor hasta 50x50 siendo el de mayor tamaño (los niveles aumentan de 10 en 10).

Laberinto creado:

Laberinto

Generar Laberinto

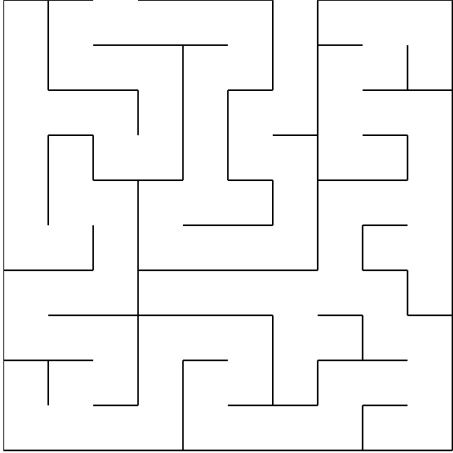
Tamaño del laberinto: 20

Fuerza bruta

Resolver Laberinto

Cargar Laberinto

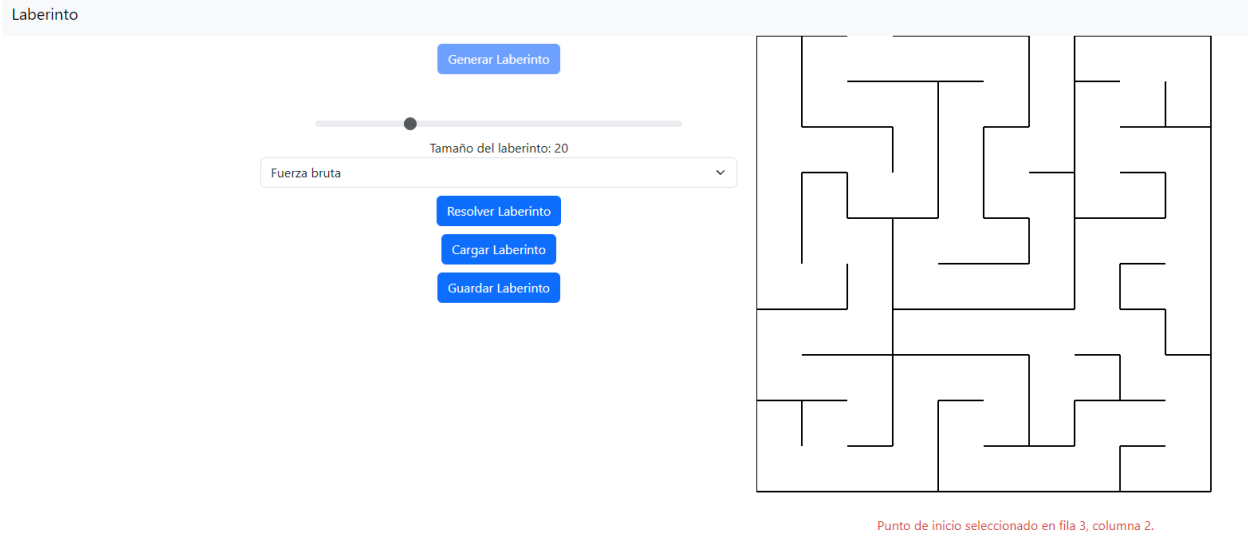
Guardar Laberinto



Haga clic en el laberinto para seleccionar el punto de inicio.

En esta parte, con el laberinto ya creado se puede probar las distintas jugabilidades, se tiene que seleccionar un punto en la figura del mapa, el cual será el punto de inicio para el juego, desde ahí se iniciará la ruta hasta la salida.

Enfoque Fuerza Bruta:



Luego de iniciar el enfoque de fuerza bruta se ejecutará el algoritmo DFS o Búsqueda en profundidad, el cual buscará todas las rutas posibles desde ese punto y cuando encuentre la que redirecciona a la salida esta se pintará en pantalla para que se pueda visualizar, cabe recalcar que este algoritmo no busca la eficiencia, sino más bien busca encontrar el camino correcto, aunque se usen muchos recursos computacionales.

Enfoque Optimización:

Laberinto

Generar Laberinto

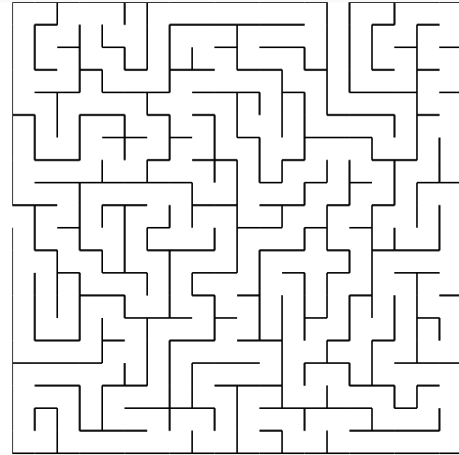
Tamaño del laberinto: 40

Optimización

Resolver Laberinto

Cargar Laberinto

Guardar Laberinto



Punto de inicio seleccionado en fila 9, columna 9.

Se selecciona el punto de inicio

Laberinto

Generar Laberinto

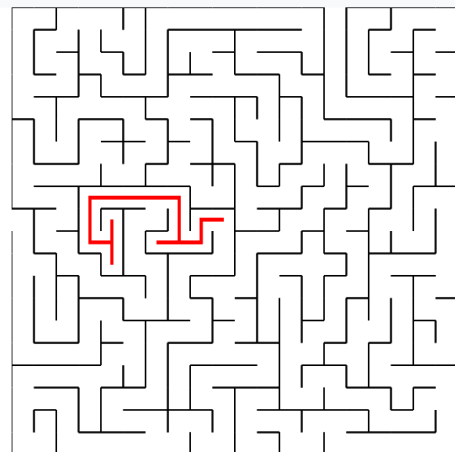
Tamaño del laberinto: 40

Optimización

Resolver Laberinto

Cargar Laberinto

Guardar Laberinto



Punto de inicio seleccionado en fila 9, columna 9.

El camino se empieza a marcar en el mapa

Laberinto

Generar Laberinto

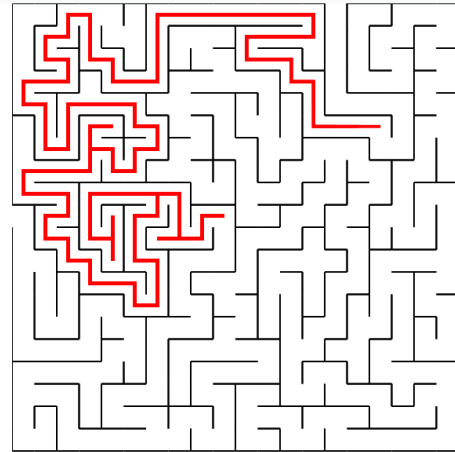
Tamaño del laberinto: 40

Optimización

Resolver Laberinto

Cargar Laberinto

Guardar Laberinto



Punto de inicio seleccionado en fila 9, columna 9.

Busca la ruta más optima tomando en cuenta los lugares ya visitados

Laberinto

Generar Laberinto

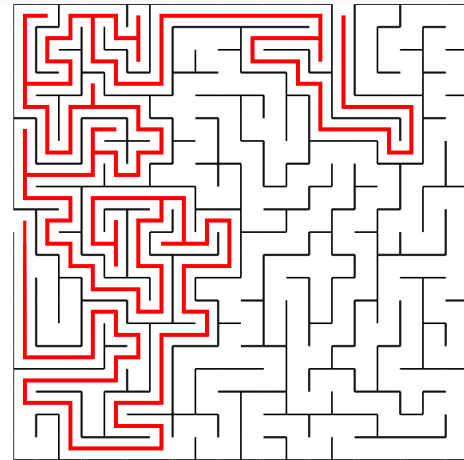
Tamaño del laberinto: 40

Optimización

Resolver Laberinto

Cargar Laberinto

Guardar Laberinto



Punto de inicio seleccionado en fila 9, columna 9.

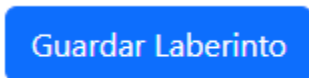
Se detiene cuando encuentra el punto de salida.

Cargar

Cargar Laberinto

Con este botón se puede cargar un laberinto previamente guardado.

Guardar



Con este botón se puede guardar la información de un laberinto que se haya creado en el sistema.

Herramientas para el desarrollo del software

En este proyecto para la funcionalidad del backend se utilizó Python como lenguaje principal con un énfasis en programación orientada a objetos, con clases como: Laberinto, Celda, Solucionador, Administrador y Algoritmos. Para el frontend se utilizó una herramienta grafica llamada Flask, la cual es compatible con Python y trabaja de la mano con HTML, en el frontend se encuentran archivos importantes para el proyecto los cuales serían: app.py (Controlador principal), #main.css (para la apariencia web) y index.html (para el desarrollo de la página web y sus componentes).

Aparte de Flask se utilizaron librerías externas como math, time y random.

Conclusión

En este proyecto se puso a prueba los conocimientos sobre los tipos de algoritmos y sus usos, así también como sus enfoques, se usaron estructuras de datos y distintos tipos de programación por lo cual creemos que fue de mucho provecho y con un trabajo continuo y ordenado se logró la mayor parte del objetivo.