

# Proyecto:

## Objetivo

Programar un resolvidor de laberintos en un entorno individual.

## Instrucciones generales

1. Cada estudiante deberá decidir cómo desarrollar su resolvidor. Las funcionalidades esperadas de su programa se especifican en la sección **Especificación** más adelante.
2. La entrega del código deberá ir acompañada de un informe técnico, cuya especificación se incluye más adelante.
3. Cada estudiante tendrá una entrevista de 10 minutos con el profesor donde se discuta su proyecto. El día de entrega se programarán las entrevistas de revisión.

## Fechas de entrega

Programa: 15 de julio de 2022 a las 23:59.

Entrevistas: a convenir con el profesor.

## Especificación

1. Al correrse, su programa deberá abrir un archivo, **laberinto.txt**, donde encontrará la definición de un laberinto. Como resultado, su programa debe imprimir las coordenadas **(x, y)** donde se encuentra la solución, si puede encontrarla; si no puede encontrarla debe imprimir el texto: **No hay solucion.**
2. El laberinto se encontrará definido como una matriz de números enteros, separados por un espacio, de tamaño **m \* n**, en donde:
  - a. 0 indica una pared.
  - b. 1 indica un camino.

c. 2 indica la solución.

d. Un laberinto de ejemplo podría ser el siguiente:

```
0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 2 0
0 0 0 2 0 0 0 0 0
1 1 0 0 0 0 0 0 0
```

3. Su programa debe comenzar a recorrer el laberinto desde uno de los bordes, y solamente puede llegar a una solución si sigue caminos válidos. No puede solamente encontrar dónde hay una solución en el archivo, tiene que seguir un camino válido.
4. Como parte de la evaluación, se tomarán en cuenta:
  - a. Organización de su programa en diferentes archivos.
  - b. Utilización adecuada de estructuras y memoria dinámica.
  - c. Claridad del código.
  - d. Documentación interna.
  - e. Correctitud del programa.
5. Como entregable, deben crear un repositorio en GitHub y darle acceso al profesor (usuario `@jccotoucr`) y su trabajo debe estar en la rama `master` para la fecha de entrega. Solamente se tomará en cuenta el trabajo realizado hasta la fecha límite.
6. El repositorio debe tener un archivo README en su nivel raíz, donde incluya el informe técnico.

## Especificación: Informe técnico



- Deberá tener una extensión mínima de 5 páginas y máxima de 10 páginas, sin contar portada e índice.
- Deberá utilizar fuente Arial, a espacio y medio, con tamaño de 12 puntos para el texto de párrafos, 14 en negrita para subtítulos y 16 en negrita para títulos. Pueden utilizar imágenes para ilustrar conceptos importantes de su trabajo, pero cada

imagen debe tener un propósito claro, debe ser utilizada en el texto y debe tener una explicación adecuada.

- Deberá contener las siguientes secciones:
  - **Portada:** al menos nombre del estudiante.
  - **Índice**
  - **Introducción:** identificar claramente el propósito del documento.
  - **Diseño general:** las partes del programa y cómo decidió programarlas.
  - **Principales retos:** mencionar y elaborar sobre al menos 5.
  - **Conclusiones:** resumir la experiencia e identificar lecciones aprendidas y preguntas sin responder.

## Evaluación

### Desglose

 Name	 Porcentaje
<u>Organización</u>	10%
<u>Informe</u>	20%
<u>Lectura de archivo de laberinto</u>	15%
<u>Resolución del laberinto</u>	45%
<u>Entrevista</u>	10%
<u>Total</u>	<b>100%</b>

Juan Carlos Coto Ulate. Todos los derechos reservados.