

Universidad San Carlos de Guatemala
Facultad de ingeniería.
Ingeniería en ciencias y sistemas



Título del Proyecto:

MazeBot

Nombre del Catedrático	Sección
M.Sc. Luis Fernando Espino Barrios	A

Tabla de contenido

1. Resumen Ejecutivo	3
2. Competencia que desarrollaremos.....	3
3. Objetivos del Aprendizaje	3
3.1 Objetivo General.....	3
3.2 Objetivos Específicos	3
4. Enunciado del Proyecto	4
4.1 Descripción del problema a resolver.....	4
4.2 Alcance del proyecto	4
4.4 Entregables	6
5. Metodología.....	6
6. Desarrollo de Habilidades Blandas.....	7
6.1 Proyectos en Grupo.....	7
7. Cronograma.....	7
8. Valores.....	8

1. Resumen Ejecutivo

MazeBot es un simulador tridimensional de resolución de laberintos desarrollado por los estudiantes del curso de Inteligencia Artificial 1 de la Universidad de San Carlos de Guatemala. Su propósito es introducir a los estudiantes en la simulación de entornos inteligentes, combinando elementos de lógica, visualización tridimensional y algoritmos de búsqueda de caminos.

MazeBot permite diseñar laberintos de manera dinámica mediante archivos JSON configurables, e implementa un robot virtual que analiza su entorno y determina la mejor ruta desde un punto de inicio hasta una meta. El proyecto aplica conocimientos adquiridos en programación, estructuras de datos, simulación y uso de bibliotecas como p5.js y Three.js, consolidando la formación técnica de los estudiantes y su capacidad para construir soluciones inteligentes visualmente atractivas.

2. Competencia que desarrollaremos

Simulación de comportamiento autónomo en entornos virtuales, diseño 3D interactivo, implementación de algoritmos de búsqueda y aplicación de conceptos de inteligencia artificial en simuladores.

3. Objetivos del Aprendizaje

3.1 Objetivo General

El estudiante será capaz de desarrollar un simulador tridimensional de un robot que resuelva de forma automática laberintos generados dinámicamente mediante configuraciones en archivos JSON, aplicando principios de inteligencia artificial, programación y simulación de comportamientos inteligentes.

3.2 Objetivos Específicos

Al finalizar el proyecto, los estudiantes deberán ser capaces de:

1. Diseñar e implementar un entorno web 3D interactivo que represente la estructura del laberinto.
2. Desarrollar un sistema de lectura e interpretación de laberintos definidos por archivos JSON.

3. Programar un robot virtual capaz de recorrer el laberinto desde un punto inicial a uno final utilizando algoritmos de búsqueda.

4. Enunciado del Proyecto

Este proyecto tiene como objetivo el desarrollo de MazeBot, una aplicación web tridimensional que simula un entorno de laberinto y un robot autónomo capaz de resolverlo utilizando algoritmos de búsqueda. El sistema permitirá la carga dinámica de laberintos mediante archivos JSON, facilitando la experimentación con diferentes configuraciones.

MazeBot está diseñado para fomentar la comprensión de conceptos de inteligencia artificial aplicados a entornos virtuales, integrando visualización 3D, simulación de comportamientos inteligentes y estrategias de resolución de problemas. La aplicación se desarrollará utilizando bibliotecas como p5.js y Three.js, y será accesible desde navegadores web modernos.

4.1 Descripción del problema a resolver

En el aprendizaje de la inteligencia artificial, es común que los conceptos teóricos como la búsqueda de caminos, percepción del entorno y toma de decisiones resulten abstractos para los estudiantes. La falta de herramientas visuales interactivas limita la comprensión de estos procesos.

El objetivo del proyecto es construir un simulador en el que los estudiantes puedan observar el comportamiento de un agente inteligente dentro de un laberinto generado dinámicamente. A través de este entorno, los usuarios podrán visualizar cómo el robot analiza su entorno, aplica un algoritmo de búsqueda y encuentra la ruta más corta hacia un objetivo.

El sistema también permitirá cargar distintos laberintos definidos en archivos JSON, lo cual fomentará la experimentación, el diseño de escenarios personalizados y la comprensión del impacto que tienen distintas estructuras en el rendimiento de los algoritmos.

4.2 Alcance del proyecto

- **Alcance obligatorio:**

Se debe desarrollar una aplicación web utilizando JavaScript, CSS, HTML y Bibliotecas de diseño 2D y 3D (p5.js, Three.js).

Se debe desarrollar o buscar un modelo de un robot, el cual será el personaje principal de la aplicación y se encargará de simular al agente inteligente que resolverá los laberintos. Este modelo debe contar con animaciones para movimiento.

La aplicación debe representar un entorno tridimensional del laberinto utilizando Three.js, incluyendo muros, caminos, punto de inicio y punto de meta.

La estructura del laberinto debe construirse de forma dinámica a través de la carga e interpretación de archivos JSON que contengan la configuración respectiva del laberinto, incluyendo dimensiones, posición de muros, inicio y fin. La estructura del JSON será la siguiente:

```
{
  "ancho": 5,
  "alto": 5,
  "inicio": [0, 0],
  "fin": [4, 4],
  "paredes": [
    [0, 1],
    [1, 1],
    [2, 1],
    [3, 1],
    [1, 3],
    [2, 3],
    [3, 3]
  ]
}
```

Se deben escoger e implementar **tres algoritmos de búsqueda** (por ejemplo, BFS, DFS, A*, Dijkstra, etc.). Debe incluirse en la documentación la descripción de cada uno de los algoritmos y la justificación del por qué se eligió.

Al momento de tener cargado y renderizado el laberinto, el robot debe situarse al inicio del laberinto, el usuario debe poder seleccionar el algoritmo que desea utilizar y posteriormente debe correr la animación. El robot deberá buscar la ruta de solución del laberinto ejecutando los movimientos paso a paso y ejecutando sus correspondientes animaciones de movimiento.

Al finalizar la simulación, el usuario deberá contar con las opciones de reiniciar la animación, utilizar otro algoritmo o cargar un nuevo laberinto.

- **Alcance opcional:**

Desarrollar su propio modelo de personaje incluyendo sus animaciones.

4.3 Requerimientos técnicos

- La aplicación debe ser desarrollada utilizando JavaScript y las bibliotecas de p5.js y Three.js para la implementación de diseño y modelos 2D y 3D.
- El sistema debe ser una aplicación web, permitiendo su uso en diversos dispositivos sin necesidad de instalación.
- Los laberintos deben ser cargados de forma dinámica desde un archivo JSON.
- Se deben utilizar GitHub y GitHub Pages para el control de versiones y la entrega final del proyecto.

4.4 Entregables

Tipo	Descripción
Aplicación Web	Enlace de la aplicación Web publicada en GithubPages.
Manual Técnico	<p>Incluye guías de instalación, diagramas y descripciones de la arquitectura o flujo del sistema desarrollado y herramientas utilizadas.</p> <p>También debe incluir la descripción y justificación para la elección de cada uno de los algoritmos de búsqueda utilizados.</p>
Manual de Usuario	Documento que explica cómo usar el sistema desarrollado, incluyendo capturas de pantalla, pasos detallados y resolución de problemas comunes.
Código Fuente	El código completo del proyecto, con una estructura clara y comentada, que refleja las buenas prácticas de programación y uso de control de versiones.

- **Fecha límite:** Domingo, 04 de mayo del 2025 a las 23:59.
- Se debe agregar al auxiliar como colaborador en el repositorio (robinbuezo11).
- El nombre del repositorio debe ser IA1_1S2025_P3_G#.
- Se hará entrega del enlace al repositorio y a la página web en la plataforma UEDI.

5. Metodología

1. **Investigación preliminar:** Los estudiantes deben realizar una investigación sobre los diferentes algoritmos de búsqueda y elegir tres de estos para su implementación.
2. **Diseño del sistema:** Crear diagramas de flujo o de arquitectura que representen cómo debería funcionar y estar estructurada la aplicación.

- 3. **Desarrollo:** Los estudiantes deberán implementar su solución, asegurándose de cumplir con el alcance y los requerimientos técnicos descritos.
- 4. **Pruebas y ajustes:** Se realizarán pruebas para verificar que la aplicación funciona correctamente en diferentes condiciones.

6. Desarrollo de Habilidades Blandas

6.1 Proyectos en Grupo

La colaboración efectiva es esencial para el desarrollo exitoso de MazeBot. Los estudiantes trabajarán en equipos, aplicando estrategias de organización, comunicación y toma de decisiones para cumplir con los objetivos del proyecto. A lo largo del proceso, se fomentará el pensamiento crítico, la adaptabilidad y el liderazgo técnico, claves en el desarrollo de soluciones inteligentes.

6.1.1 Trabajo en Equipo

Cada grupo debe conformarse de forma colaborativa, asignando roles como líder técnico, programador de lógica, desarrollador de entorno 3D y responsable de documentación. El equipo deberá coordinarse usando herramientas que considere necesarias y GitHub para control de versiones y colaboración en el código fuente.

6.1.2 Comunicación Efectiva

La comunicación del equipo debe ser continua y organizada, con reuniones periódicas y una colaboración activa en el repositorio. Todos los integrantes deben aportar valor mediante commits progresivos. Se espera que cada miembro contribuya al desarrollo técnico y la documentación del proyecto, asegurando una integración colaborativa y un aprendizaje compartido.

6.1.3 Resolución de Conflictos

Durante el proyecto, es posible que surjan desacuerdos sobre enfoques de diseño, distribución de tareas o implementación. Los equipos deberán enfrentar estos retos con apertura al diálogo, priorizando soluciones colectivas y el cumplimiento de metas comunes, fortaleciendo así su capacidad de negociación y empatía profesional.

7. Cronograma

Tipo	Fecha Inicio	Fecha Fin
Asignación del Proyecto	09 de abril del 2025.	
Desarrollo del Proyecto	09 de abril del 2025.	04 de mayo del 2025.

Calificación del Proyecto	05 de mayo del 2025.	07 de mayo del 2025.
---------------------------	----------------------	----------------------

8. Valores

- Copias parciales o totales serán sancionadas con una nota de cero y los responsables serán reportados a la Escuela de Ciencias y Sistemas.
- El proyecto debe ser OpenSource con licencia MIT.
- El lenguaje a utilizar será JavaScript, no está permitido el uso de frameworks ni runtimes.
- El código, manual de usuario, manual técnico y todo lo relacionado con el proyecto debe alojarse en un repositorio de GitHub, este debe ser privado y luego de la calificación se debe dejar público.
- No se permiten ediciones o modificaciones después de la fecha de entrega.