



Instituto Tecnológico de Costa Rica

Introducción a la programación

Escape del Laberinto

Proyecto II

Claudia Olivas Morera

Estudiante de Ingeniería en Computación
Semestre II, 2025

San Carlos, Costa Rica

1. Atributo de Análisis de Problema

El proyecto **Escape del Laberinto** modela un problema complejo de ingeniería: la navegación en un laberinto dinámico con recursos limitados. Se combinan:

- Matemáticas: uso de matrices, conteo de movimientos y distancias simples.
- Ciencias naturales: agentes que reaccionan a estímulos (perseguir o huir).
- Ingeniería: diseño modular del sistema y simulación discreta por turnos.

Las variables relevantes son la estructura del laberinto, el estado del jugador y de los enemigos, la dificultad (cantidad/velocidad de enemigos y energía disponible) y las restricciones de movimiento.

El plan de solución se basa en:

- Generar un laberinto aleatorio siempre conectable entre entrada y salida.
- Modelar jugador, enemigos y bombas como objetos con atributos y métodos claros.
- Definir reglas de movimiento y consumo/recuperación de energía para cada modo.
- Integrar una interfaz gráfica que visualiza el estado del sistema y permite la interacción en tiempo real.

Evaluación breve de la solución:

- *Pros*: diseño modular y reutilizable, lógica común para ambos modos, control explícito de recursos y parámetros de dificultad ajustables.
- *Contras*: la aleatoriedad del laberinto puede generar escenarios difíciles de balancear; se requiere amplia prueba para garantizar jugabilidad justa.

2. Atributo de Herramientas de Ingeniería

Las principales técnicas y herramientas utilizadas son:

- **Python** como lenguaje principal y **Pygame** para la interfaz gráfica, entrada por teclado y manejo del bucle del juego.
- **Programación orientada a objetos** para modelar casillas, jugador, enemigos, bombas, dificultades y modos de juego.
- Archivos **JSON** para el almacenamiento simple y eficiente de puntajes.

En el proyecto, las técnicas se aplican así:

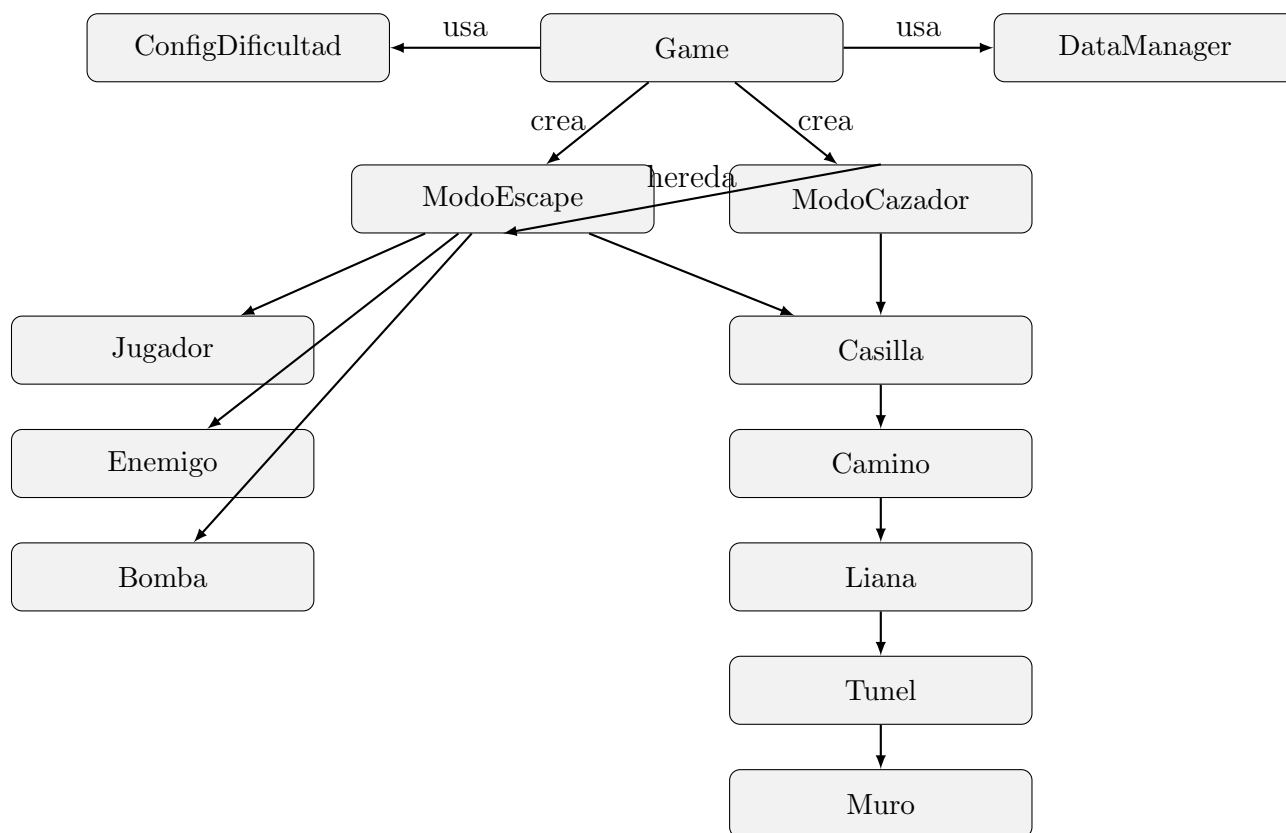
- `main.py` concentra la lógica de laberinto, tipos de casilla, jugador, enemigos, bombas y parámetros de dificultad.

- `map.py` implementa `ModoEscape` y `ModoCazador`, que reutilizan la lógica de `main.py` y añaden sprites, HUD y controles.
- `main_menu.py` gestiona el ciclo principal, los menús, el registro de jugadores y el acceso a `DataManager` para los puntajes.

Durante el desarrollo se fueron adaptando las herramientas:

- Primero se probó la lógica de movimientos en consola; luego se integró Pygame sin cambiar la estructura básica del modelo.
- Se refactorizaron funciones repetidas en módulos independientes para mejorar mantenimiento y claridad.
- Los parámetros de dificultad se ajustaron empíricamente para equilibrar consumo de energía, velocidad de enemigos y nivel de reto.

A continuación se presenta el diagrama de clases simplificado del sistema.



Esquema de puntaje, dificultad y condiciones de juego

El esquema de puntaje fue definido por el grupo y se describe a continuación para cada modo de juego, considerando la dificultad, las condiciones de victoria/derrota y el desempeño del jugador.

Modo 1: Escapa

En este modo el jugador debe llegar a la salida sin ser alcanzado por ningún cazador.

- **Dificultad:** se controla mediante el objeto `ConfigDificultad`, que define la cantidad de enemigos, su frecuencia de movimiento y la energía máxima del jugador. Dificultades más altas implican más cazadores y mayor presión sobre el jugador.
- **Condición de victoria:** el jugador gana si alcanza la casilla de salida con energía mayor que cero.
- **Condición de derrota:** el jugador pierde si un cazador entra en su misma casilla o si su energía llega a cero.
- **Puntaje:** el puntaje depende principalmente de la eficiencia del recorrido:
 - Se parte de un puntaje base que disminuye con el número de movimientos (a menor tiempo/movimientos, mayor puntaje).
 - Luego se aplica un multiplicador según la dificultad: niveles con más enemigos y mayor velocidad otorgan una recompensa mayor.

Este esquema refleja la idea de que escapar rápido en una dificultad alta es más valioso que hacerlo lentamente en una dificultad baja.

Modo 2: Cazador

En este modo el jugador es el cazador y los demás se convierten en objetivos que intentan huir hacia la salida.

- **Dificultad:** al igual que en el modo anterior, `ConfigDificultad` ajusta la cantidad de enemigos y su velocidad. A mayor dificultad, hay más enemigos moviéndose con mayor frecuencia.
- **Condición de victoria:** el jugador gana al alcanzar un número objetivo de capturas (`OBJETIVO_CAPTURAS`) antes de quedarse sin energía.
- **Condición de derrota:** la partida termina en derrota si la energía del jugador llega a cero.
- **Puntaje:** se acumula dinámicamente durante la partida:
 - Cada enemigo atrapado suma una cantidad fija de puntos (`PUNTOS_ATRAPAR_ENEMIGO`).
 - Cada enemigo que logra escapar por la salida resta puntos (`PENALIZACION_ENEMIGO_ESCAPA`), sin permitir que el puntaje baje de cero.

En este modo se premia la rapidez y precisión al interceptar enemigos, y se penaliza cada escape, reforzando el objetivo de controlar el laberinto de forma eficiente.