

Instituto Tecnológico de Costa Rica

Proyecto 2

Curso: Introducción a la programación

Profesor: Diego Mora

Estudiantes: Oved Zamora Alvarado, Edier Isaac Barrantes Azofeifa

Proyecto 2

Fecha: 28/11/2025

Introducción

El presente documento corresponde a la documentación del Proyecto 2, desarrollado en el Instituto Tecnológico de Costa Rica. Este proyecto consiste en un juego programado en Python utilizando la biblioteca Pygame. El software implementa dos modos de juego: *Escapa* y *Cazador*, donde el jugador debe cumplir objetivos distintos según el modo seleccionado.

En este documento se presenta la descripción del problema, la solución diseñada, el diagrama de clases del programa, un manual básico de usuario y la conclusión general del proyecto.

Descripción del Problema

El problema consiste en diseñar un juego tipo laberinto donde intervienen un jugador, enemigos y distintos tipos de casillas generadas de forma procedural.

El juego debe permitir dos modos:

1. **Escapa:**

El jugador inicia en la esquina superior izquierda y debe llegar a la salida ubicada en la esquina inferior derecha, evitando a los *cazadores*. El jugador puede colocar trampas limitadas con tiempo de recarga para detener temporalmente a los enemigos.

2. **Cazador:**

El jugador debe atrapar a los *corredores* antes de que estos logren escapar por cualquiera de las salidas designadas. Cada corredor capturado otorga puntos.

El sistema debe:

- Generar mapas.
- Administrar trampas, energía del jugador, movimiento de NPCs, spawn y respawn.
- Mostrar interfaz gráfica completa.
- Guardar puntajes en archivo usando formato JSON.
- Detectar victoria o derrota según las reglas del modo.

Descripción de la Solución

La solución está implementada completamente en Python usando la biblioteca Pygame. El programa está organizado alrededor de una clase principal llamada **Juego**, la cual controla:

- El estado actual del juego (registro, menú, juego, fin de partida).
- Los mapas generados.
- La lógica del jugador y enemigos.
- El sistema de trampas y energía.
- El tiempo de juego, número de capturas y puntaje.
- El dibujo de la interfaz gráfica, entidades, HUD y menús.
- La carga y guardado de puntajes.

Generación del mapa

Se utilizan funciones auxiliares para construir el mapa:

- `crear_mapa()`
- `agregar_camino_principal()`
- `agregar_caminos_secundarios()`
- `agregar_red_tuneles()`
- `crear_tunel()`
- `agregar_lianas()`

El mapa se compone de casillas de distintos tipos (camino, túnel, liana, pared), generados aleatoriamente para crear variación en cada partida.

Lógica de movimiento

- El jugador se mueve con W/A/S/D o flechas.
- Puede hacer *sprint* si tiene energía suficiente.
- Los NPCs se mueven con un comportamiento simple:

- Cazadores persiguen al jugador.
- Corredores huyen del jugador.

Trampas

En el modo “Escapa”, el jugador puede colocar trampas con clic derecho sobre casillas de camino.

Cada trampa:

- Tiene un límite máximo.
- Tiene tiempo de recarga.
- Captura enemigos temporalmente y los respawnea luego de unos segundos.

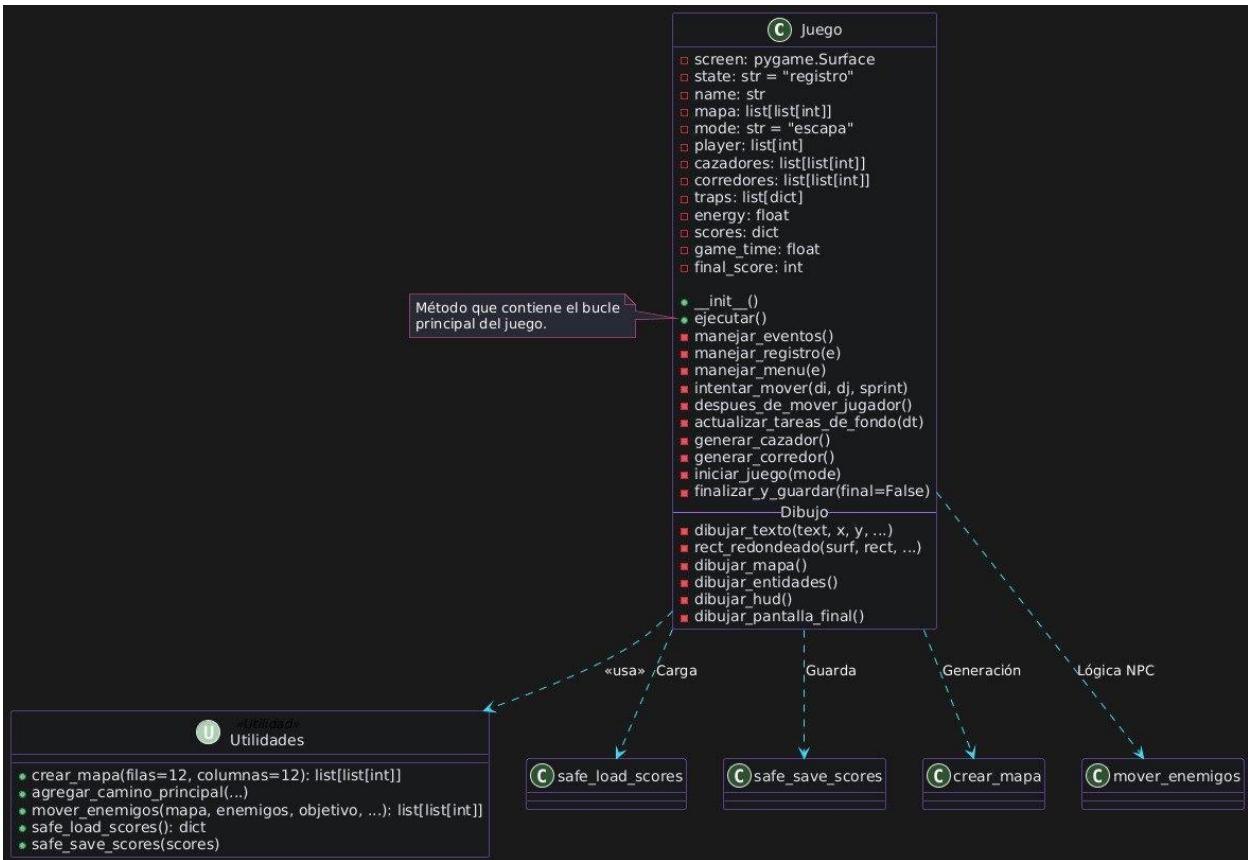
Puntajes

- El sistema guarda los puntajes más altos en “scores.json”.
- Usa funciones seguras:
 - `safe_load_scores()`
 - `safe_save_scores()`

Pantallas

- Registro (ingreso del nombre)
- Menú principal
- Puntajes
- Juego
- Pantalla de fin de partida (victoria/derrota)

Diagrama de clases



Manual de Usuario Requisitos

- Python 3 instalado
- Pygame instalado

Cómo ejecutar

1. Colocar los archivos del proyecto en una carpeta.
2. Abrir terminal dentro de esa carpeta.
3. Ejecutar:
`python nombre_del_archivo.py`
4. En la pantalla de registro, escribir el nombre del jugador y presionar ENTER.

Controles del juego

- **Movimiento:** W/A/S/D o flechas
- **Sprint:** Mantener SHIFT
- **Colocar trampa (modo Escapa):** clic derecho
- **Salir al menú:** ESC

Objetivos

- **Modo Escapa:** Llegar a la salida sin ser atrapado.
- **Modo Cazador:** Atrapar corredores antes de que escapen.

Notas

- La energía se regenera.
- Las trampas tienen recarga.
- Los puntajes se guardan automáticamente.

Conclusión

El Proyecto 2 permitió aplicar conceptos de programación en Python, generación procedural de mapas, manejo de entidades y creación de interfaces gráficas con Pygame. La estructura modular del programa facilita mantenerlo y expandirlo. El sistema cumple con los requisitos del enunciado y ofrece una experiencia interactiva completa con almacenamiento de puntajes, modos de juego diferenciados y elementos de estrategia como trampas y energía.

