

# Rescate en el Laberinto

## Índice

Descripción del Juego.....	1
Instrucciones del Juego.....	1
Paso a Paso para Implementar el Juego.....	1
Imágenes del juego.....	3

## Descripción del Juego

El juego "Rescate en el Laberinto" consiste en guiar a un personaje a través de un laberinto, recolectando objetos especiales y evitando trampas.

El objetivo es alcanzar la salida antes de quedarse sin energía o de que se acabe el tiempo. Cada paso que toma el jugador consume energía, y al caer en una trampa, la energía se reduce más. Los objetos especiales, sin embargo, permiten recuperar energía.

"Rescate en el Laberinto" se implementará usandi HTML, CSS, jQuery y TypeScript.

## Instrucciones del Juego

1. Controla al jugador con las teclas de dirección (arriba, abajo, izquierda, derecha).
2. Cada paso consume energía. Si la energía llega a cero, el juego termina.
3. Las trampas restan 20 puntos de energía si caes en una.
4. Los objetos especiales suman 10 puntos de energía al jugador.
5. El objetivo es alcanzar la salida antes de que se acabe el tiempo o te quedes sin energía.

## Paso a Paso para Implementar el Juego

Creación de la lógica el juego con jQuery y TypeScript.

### 1. Crear el archivo TypeScript (script.ts)

Usa un `namespace` llamado `LaberintoGame` para organizar el código del juego.

- Define una clase `Celda` como base para todos los elementos del laberinto.
- Crea una clase `Jugador` que herede de `Celda`. Incluye un atributo `energia` que disminuye con cada movimiento y un método `mover()`.
- Añade clases para `Trampa`, `ObjetoEspecial`, y `Salida` que heredan de `Celda` y que representan los elementos del juego.

### 2. Crear el sistema de laberinto

- ◆ Define una clase `Matriz<T>` con uso de Generics que represente la cuadrícula del laberinto y permita almacenar los elementos.
- ◆ Implementa métodos `set()` y `get()` para manipular la cuadrícula.

### 3. Lógica de Juego

Crea la clase `Juego` dentro del `namespace LaberintoGame`. Esta clase debe contener:

- ◆ Un objeto `laberinto` del tipo `Matriz<Celda>`.
- ◆ Un `jugador` del tipo `Jugador`.
- ◆ Métodos para inicializar el laberinto con elementos (trampas, objetos especiales y la salida).
- ◆ Un método `moverJugador(dx, dy)` para mover al jugador según las teclas de dirección.
- ◆ Métodos `actualizarEnergia()` y `actualizarTiempo()` que actualicen la energía y el tiempo en la interfaz.

### 4. Renderizar el laberinto con jQuery

- ◆ Dentro de la clase `Juego`, implementa el método `renderizarLaberinto()`. Este método debe actualizar la cuadrícula en el contenedor `#laberinto` en el HTML.
- ◆ Usa jQuery para crear y añadir los elementos HTML correspondientes a cada celda según su tipo (`trampa`, `objeto`, `salida`, o `jugador`).

### 5. Control de eventos y movimiento del jugador

- ◆ Usa jQuery para capturar eventos de teclado que muevan al jugador en la cuadrícula.
- ◆ Maneja el evento `keydown` para las teclas de dirección y llama a `moverJugador()` para actualizar la posición del jugador.

### 6. Temporizador del juego

- ◆ Implementa un temporizador en el método `iniciarTemporizador()`. Disminuye el tiempo cada segundo y termina el juego cuando llegue a 0.

### 7. Estilo del juego

- ◆ Usa el archivo `styles.css` para definir los estilos de la cuadrícula y de cada elemento en el laberinto (jugador, trampas, objetos y salida).

# Imágenes del juego

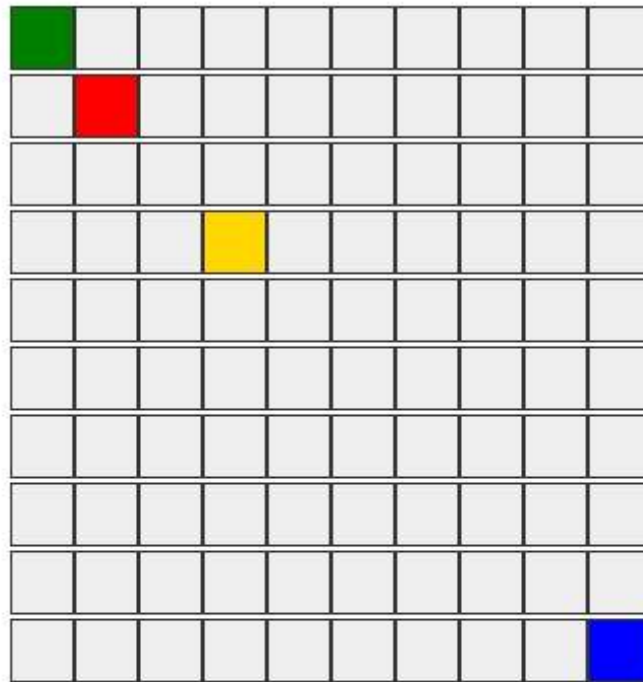
El inicio del juego es el siguiente.

Leyenda:

- cuadrado **verde**: jugador
- cuadrado **rojo**: obstáculo (resta energía)
- cuadrado **amarillo**: zona de cura (aumenta energía)
- cuadrado **azul**: salida

Objetivo:

Alcanzar la salida con la menor pérdida de energía en el menor tiempo posible.



Energía: 100

Tiempo: 120s

Iniciar Juego