

# Informe Mini Mario Bros

## Mini Mario Bros

**Autores:** Marcelo Carabajal, Gisela Grisel Lanzillotta, Gastón Carlos Carabajal

**Fecha:** 19/10/2025

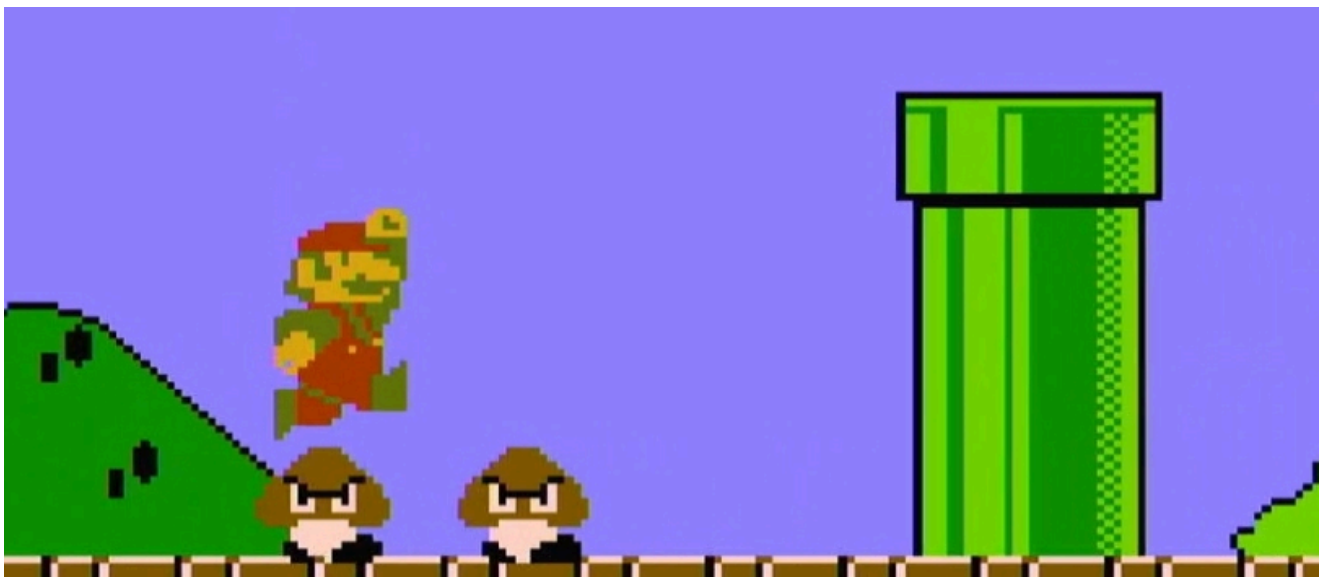
---

## Historia

El Reino Pixelado vive en paz desde hace muchos años, hasta que un grupo de criaturas llamadas *Goombas* invade el territorio.

Tu misión es recorrer el nivel, saltar por plataformas y esquivar enemigos para llegar a la bandera final y restaurar la paz.

El héroe de esta historia, **Mini Mario**, no posee armas, pero tiene su salto, su astucia y su habilidad para esquivar los peligros del mundo.



## Personajes y Entidades

### Jugador

El protagonista controlado por el usuario. Puede correr y saltar, posee varias vidas y acumula puntos al eliminar enemigos o al llegar al final del nivel.

- Organización: Básica
- Tecnología: Ninguna

- Armamento: Salto

## Enemigos

Criaturas que patrullan el nivel moviéndose de un lado a otro. Si el jugador las pisa, mueren; si lo tocan de costado, el jugador pierde una vida.

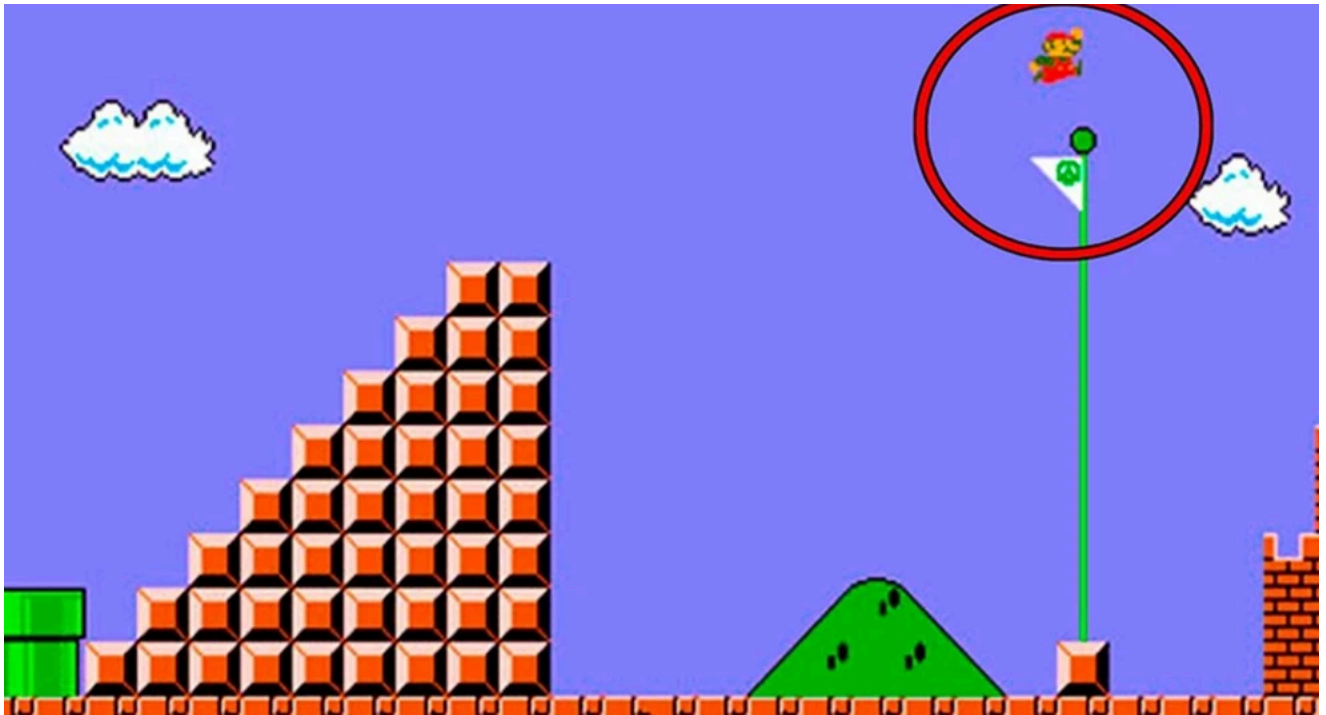
- Organización: Nula
- Tecnología: Nula
- Armamentos: Cuerpo físico

## Plataformas

Estructuras sólidas que permiten al jugador avanzar y alcanzar zonas elevadas.

## Meta (Bandera final)

Marca el fin del nivel. Al alcanzarla, el jugador gana y se guarda su puntaje en archivo.



## Mecánica de Juego

El jugador debe desplazarse lateralmente por un nivel con desplazamiento 2D. Puede saltar sobre plataformas, esquivar o eliminar enemigos y alcanzar la meta.

## Reglas básicas

- Movimiento lateral con A/D o flechas.
- Salto con la barra espaciadora.

- Gravedad constante que afecta al jugador.
- Colisiones:
  - Suelo/plataforma → detienen la caída.
  - Enemigo desde arriba → muere enemigo (+100 pts).
  - Enemigo lateral → jugador pierde una vida.
- Al llegar a la meta → fin del nivel, se calcula el puntaje y el tiempo.

(inicio → jugar → colisiones → ganar/perder).

---

## Controles

Acción	Tecla
Mover izquierda	A / ←
Mover derecha	D / →
Saltar	Barra espaciadora
Salir del juego	Esc

---

## Persistencia de Datos (POO + FILE)

La persistencia se maneja **exclusivamente mediante clases**, utilizando la API de archivos de C ( FILE\* ) para lectura y escritura binaria.

De este modo se cumple con la consigna de usar FILE , pero dentro de un enfoque completamente orientado a objetos.

## Ejemplo de clase de persistencia

```
class GestorArchivos {
private:
    const char* nombreArchivo = "Puntajes.dat";

public:
    void guardarPuntaje(const std::string& nombre, int puntaje, float
tiempo) {
        FILE* archivo = fopen(nombreArchivo, "ab");
        if (!archivo) return;

        // Escribir datos en orden binario
        fwrite(nombre.c_str(), sizeof(char), 20, archivo);
        fwrite(&puntaje, sizeof(int), 1, archivo);
    }
}
```

```

        fwrite(&tiempo, sizeof(float), 1, archivo);

        fclose(archivo);
    }

    void mostrarPuntajes() {
        FILE* archivo = fopen(nombreArchivo, "rb");
        if (!archivo) return;

        char nombre[20];
        int puntaje;
        float tiempo;

        while (fread(nombre, sizeof(char), 20, archivo)) {
            fread(&puntaje, sizeof(int), 1, archivo);
            fread(&tiempo, sizeof(float), 1, archivo);

            printf("Jugador: %s | Puntaje: %d | Tiempo: %.2f\n", nombre,
puntaje, tiempo);
        }

        fclose(archivo);
    }

    long obtenerTamaño() {
        FILE* archivo = fopen(nombreArchivo, "rb");
        if (!archivo) return 0;
        fseek(archivo, 0, SEEK_END);
        long bytes = ftell(archivo);
        fclose(archivo);
        return bytes;
    }
};

```