

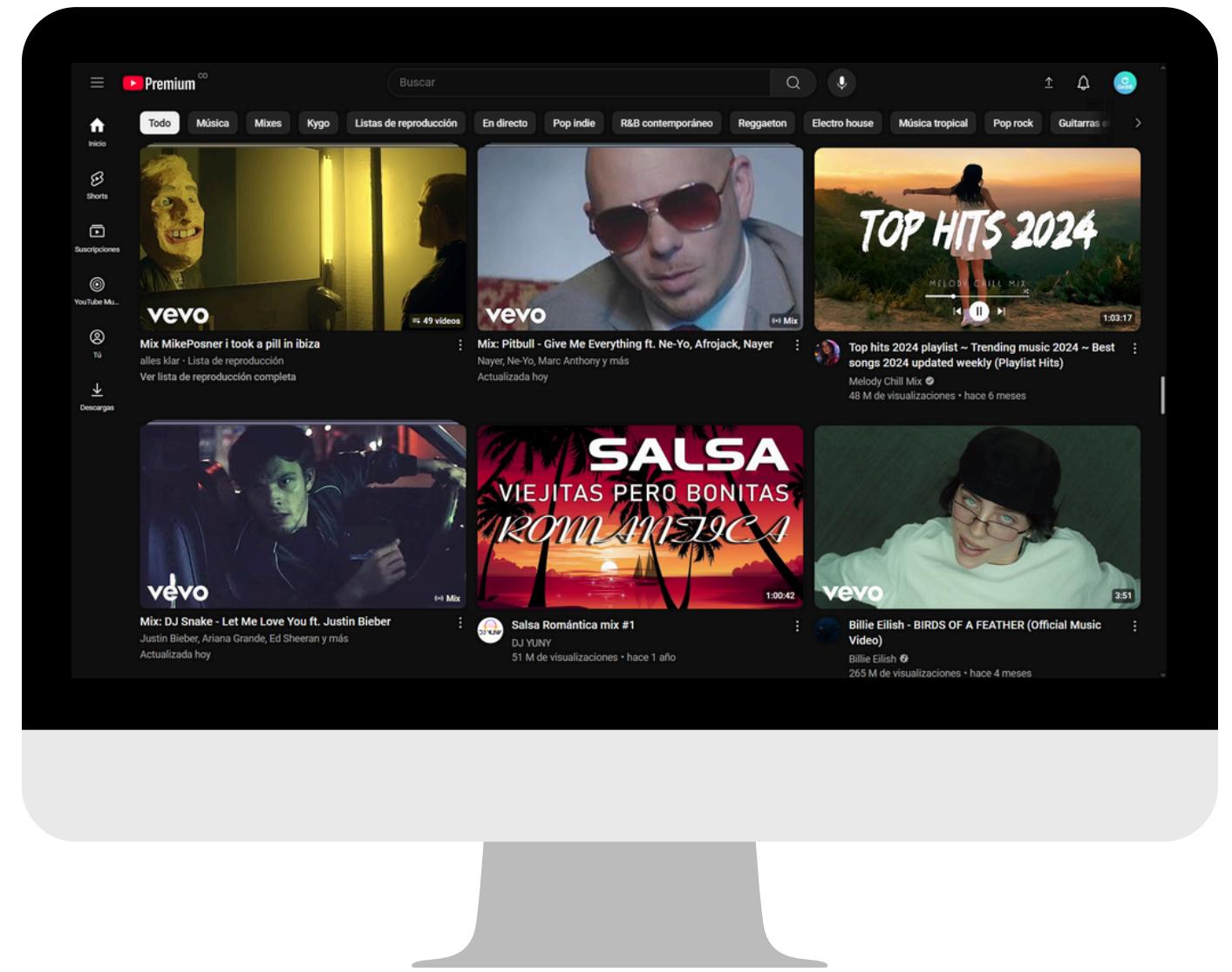
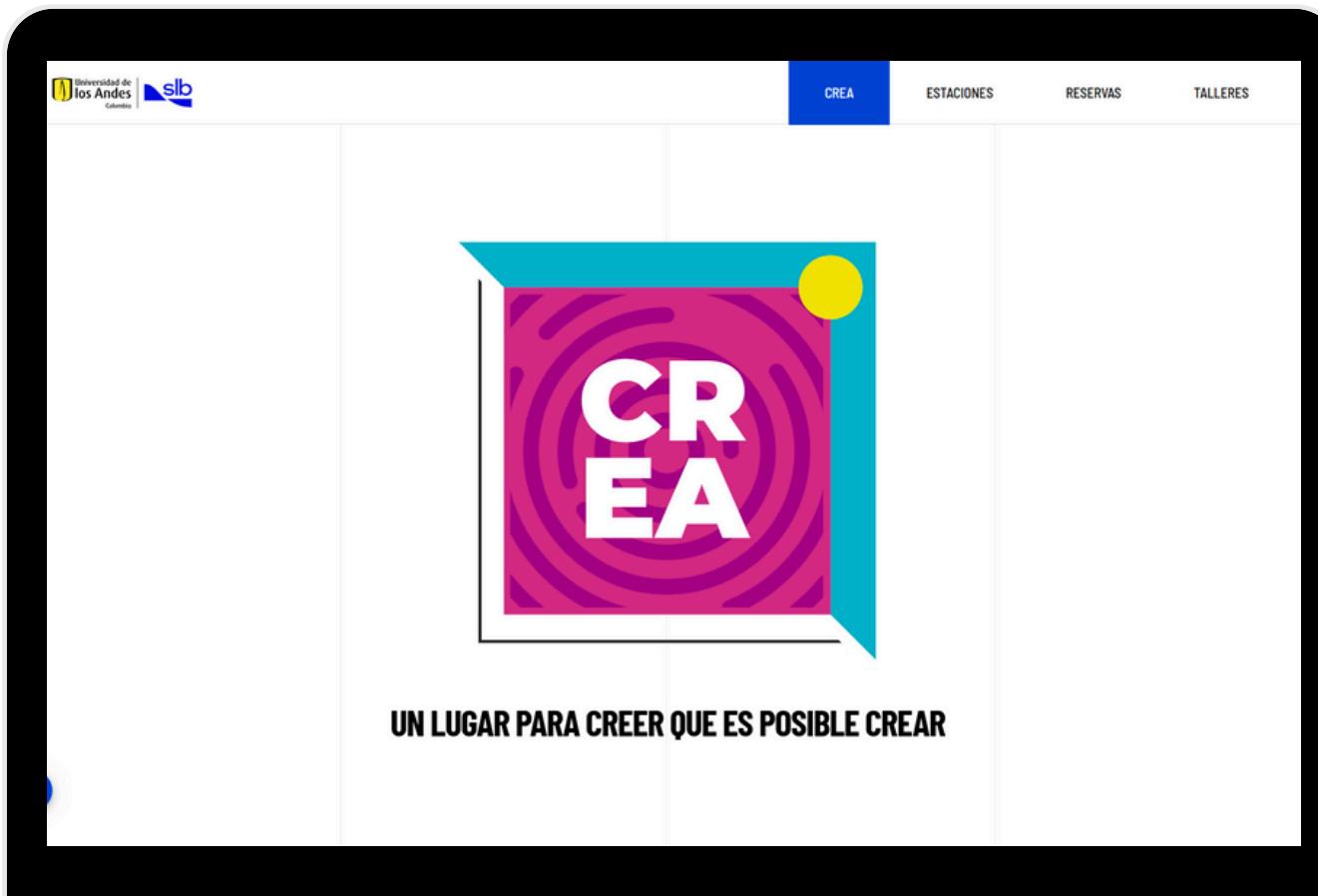


¡CREA tu Propio Videojuego!

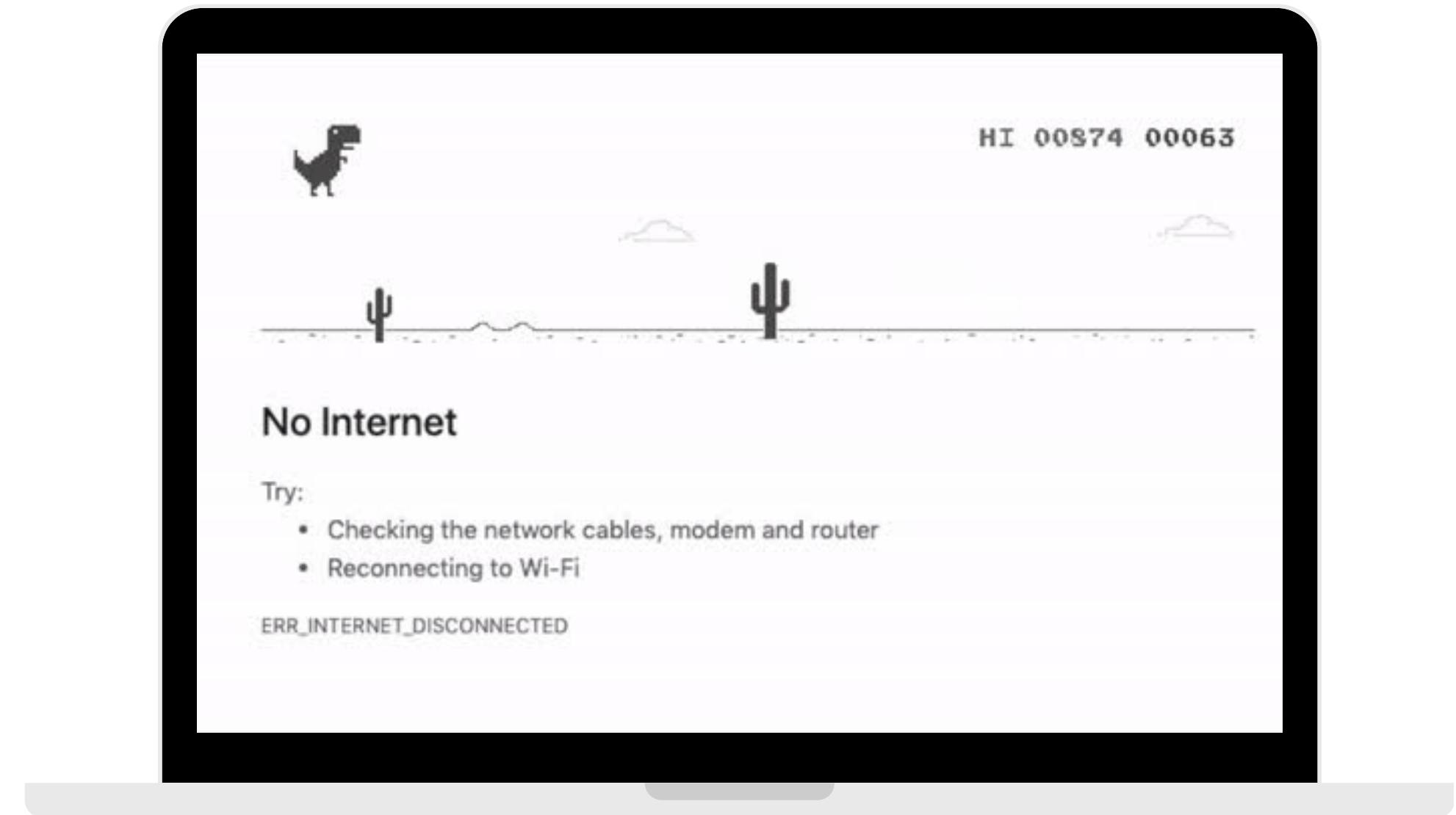
Explorando la programación web



Programación web



Programación web - Juegos



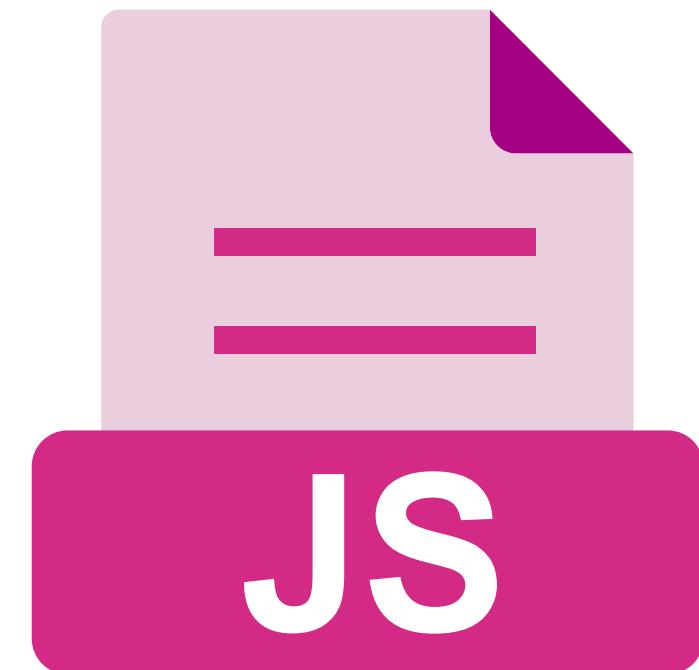
¿Cómo funciona una página web?



Estructura

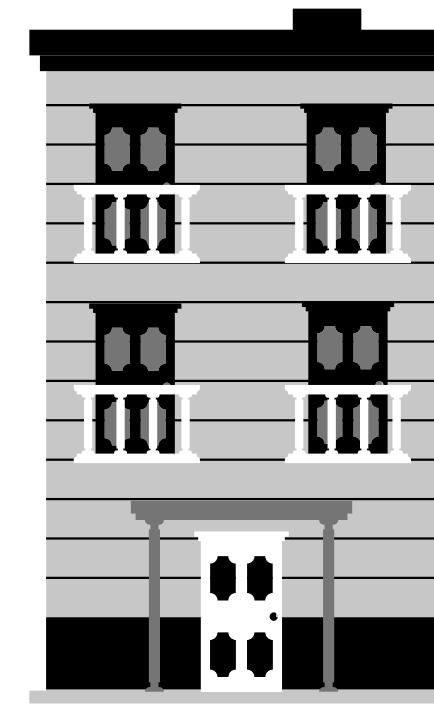


Estilo y Diseño

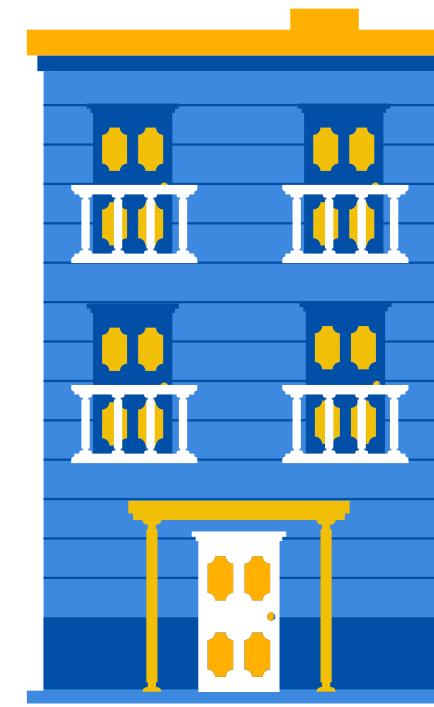


Interacción

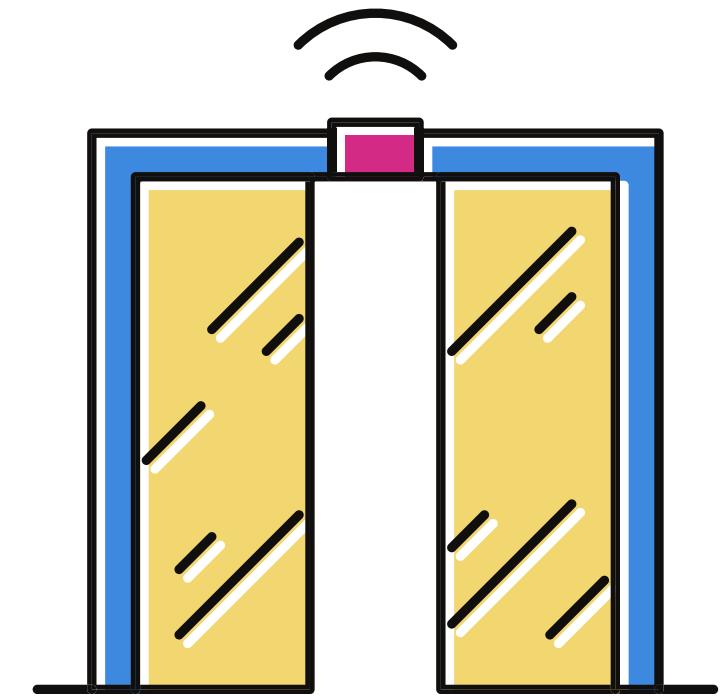
¿Cómo funciona una página web?



HTML

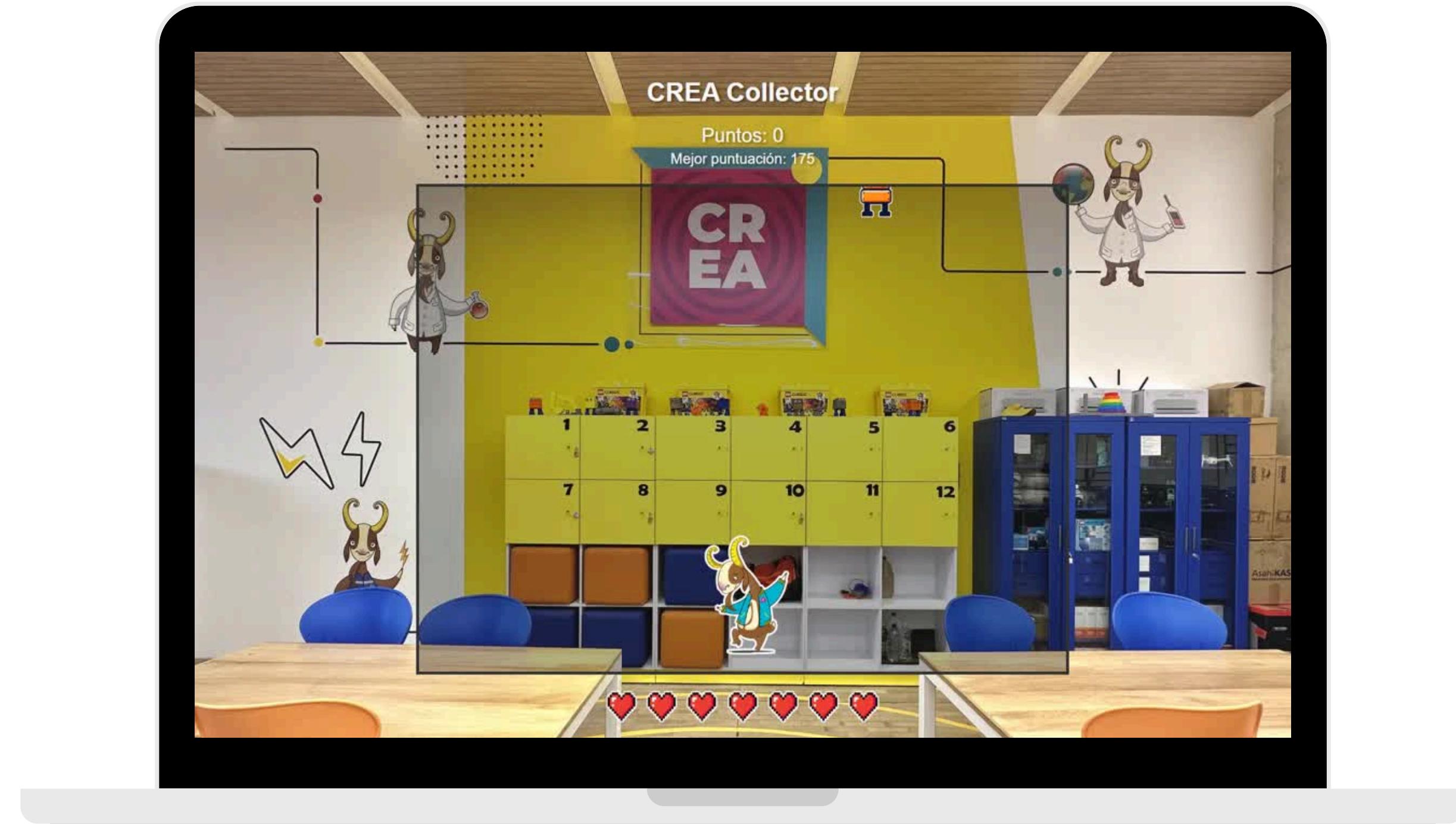


CSS

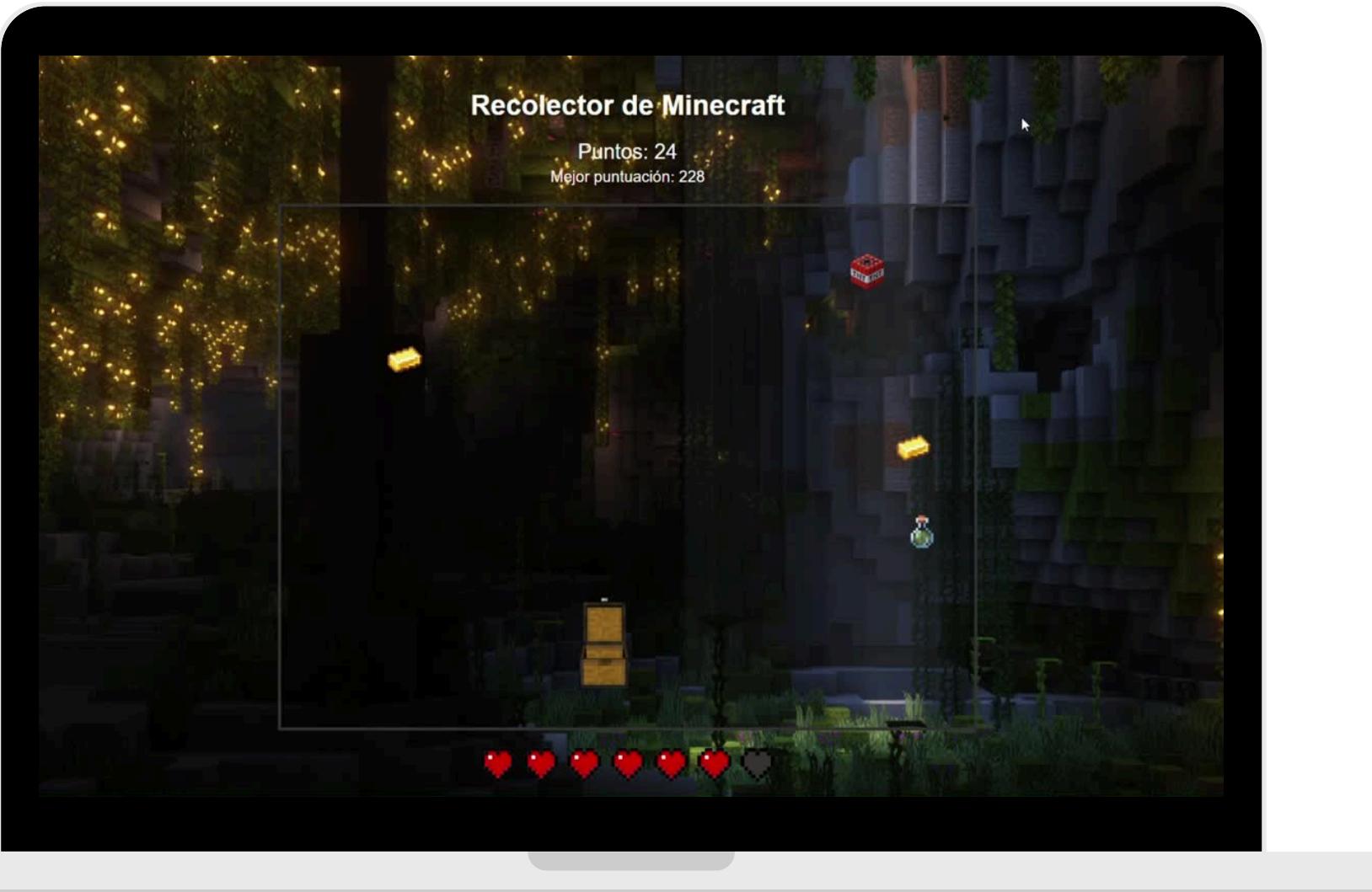


JavaScript

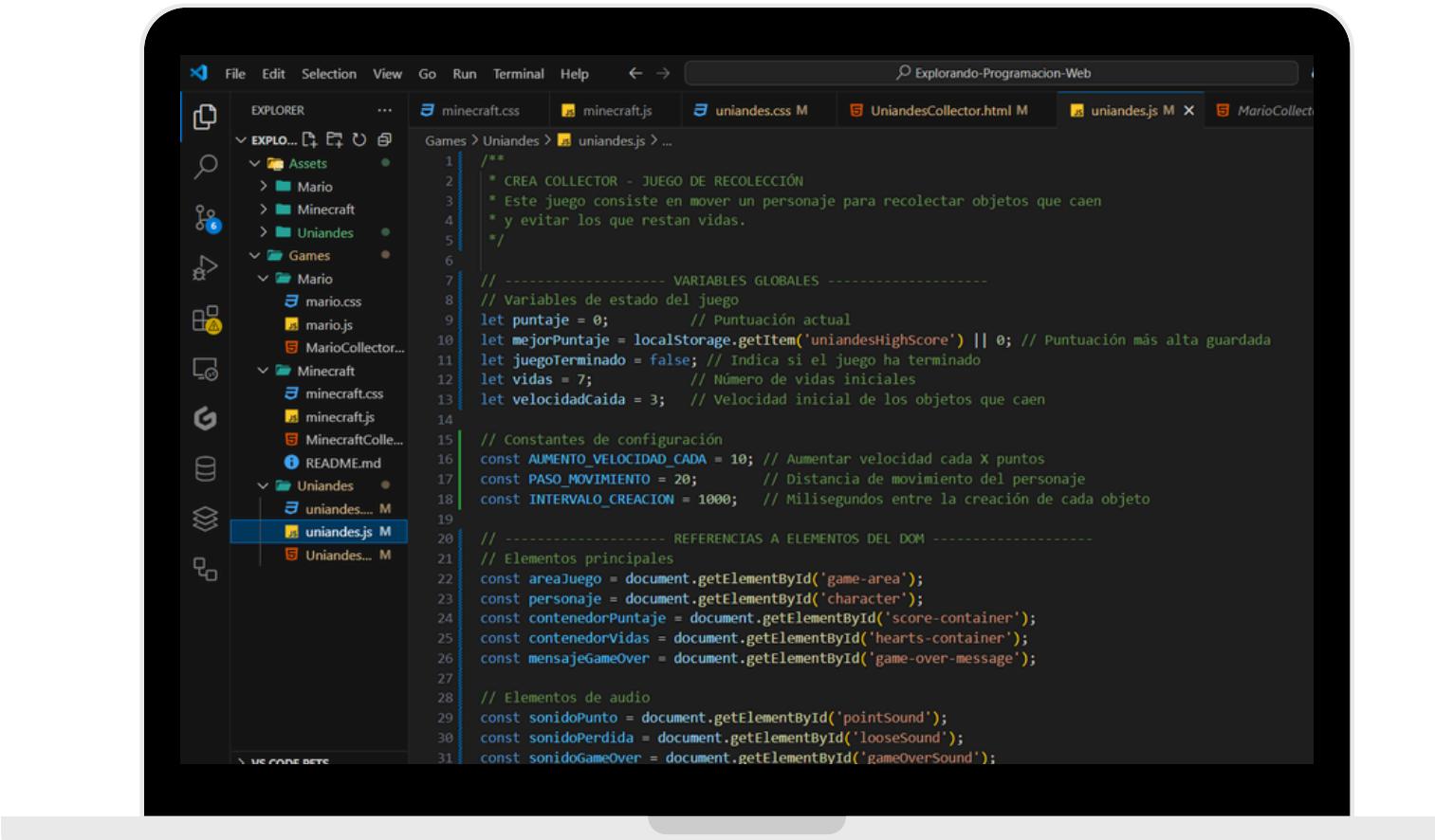
Ejemplos de lo que vamos a crear



Ejemplos de lo que vamos a crear



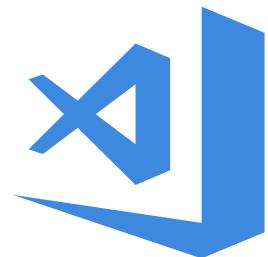
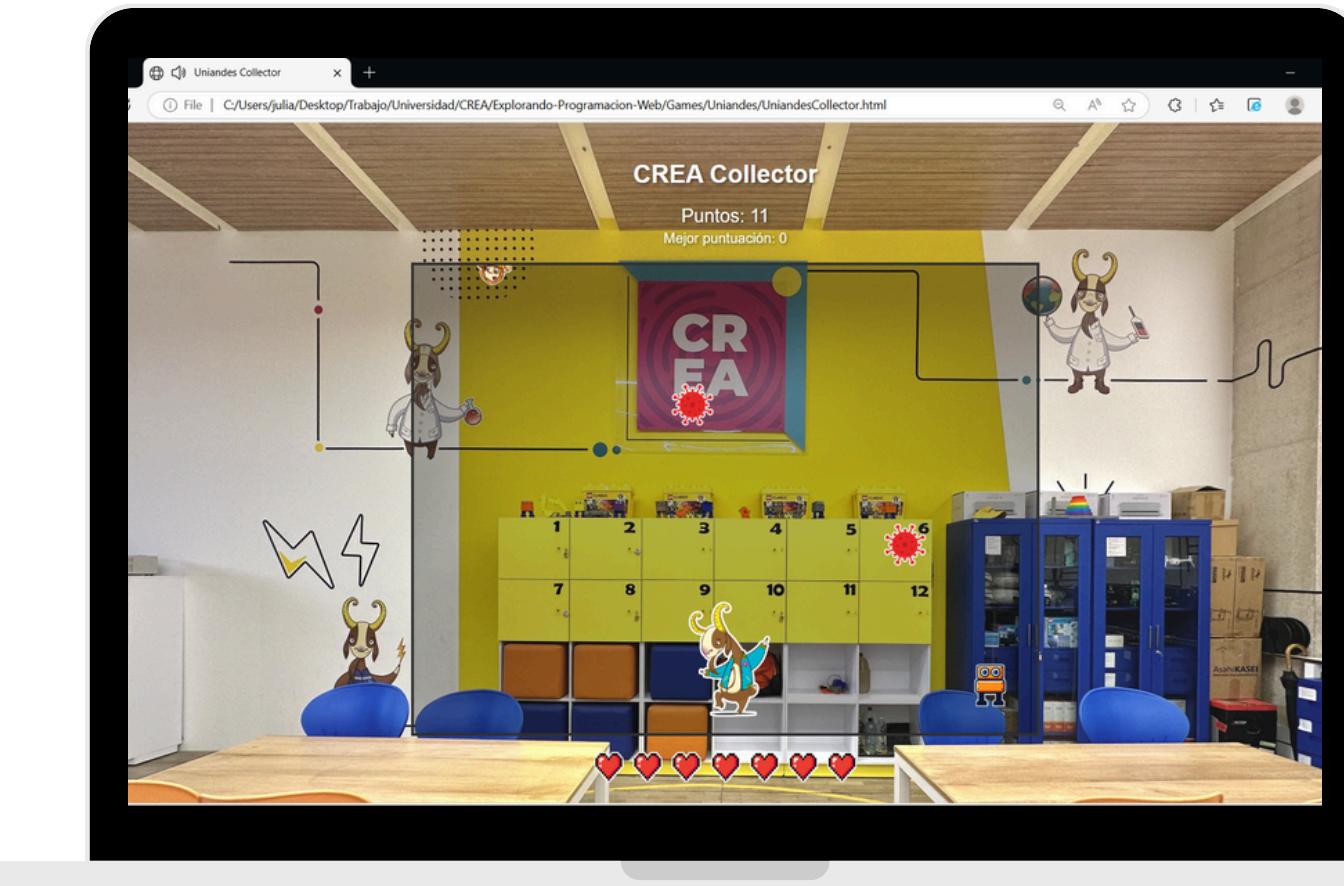
¿Qué vamos a utilizar?



```

1 /**
 * CREA COLLECTOR - JUEGO DE RECOLECCIÓN
 * Este juego consiste en mover un personaje para recolectar objetos que caen
 * y evitar los que restan vidas.
 */
2
3 // ----- VARIABLES GLOBALES -----
4
5 let puntaje = 0; // Puntuación actual
6 let mejorPuntaje = localStorage.getItem('uniandesHighScore') || 0; // Puntuación más alta guardada
7 let juegoTerminado = false; // Indica si el juego ha terminado
8 let vidas = 7; // Número de vidas iniciales
9 let velocidadCaida = 3; // Velocidad inicial de los objetos que caen
10
11 // Constantes de configuración
12 const AUMENTO_VELOCIDAD_CADA = 10; // Aumentar velocidad cada X puntos
13 const PASO_MOVIMIENTO = 20; // Distancia de movimiento del personaje
14 const INTERVALO_CREACION = 1000; // Milisegundos entre la creación de cada objeto
15
16 // ----- REFERENCIAS A ELEMENTOS DEL DOM -----
17
18 // Elementos principales
19 const areaJuego = document.getElementById('game-area');
20 const personaje = document.getElementById('character');
21 const contenedorPuntaje = document.getElementById('score-container');
22 const contenedorVidas = document.getElementById('hearts-container');
23 const mensajeGameover = document.getElementById('game-over-message');
24
25 // Elementos de audio
26 const sonidoPunto = document.getElementById('pointsSound');
27 const sonidoPerdida = document.getElementById('looseSound');
28 const sonidoGameOver = document.getElementById('gameOverSound');
29
30
31

```



**Entorno de Desarrollo Integrado
IDE**



Explorador



Para iniciar

A screenshot of the Google Drive web interface. The left sidebar shows navigation options like 'Nuevo', 'Página principal', 'Mi unidad', 'Computadoras', etc. The main area shows a folder named 'Compartidos conmigo' containing two items: 'Entregas' and 'CREA-TU-PROPIO-VIDEOJUEGO.zip'. The file 'CREA-TU-PROPIO-VIDEOJUEGO.zip' is highlighted with a blue selection bar.

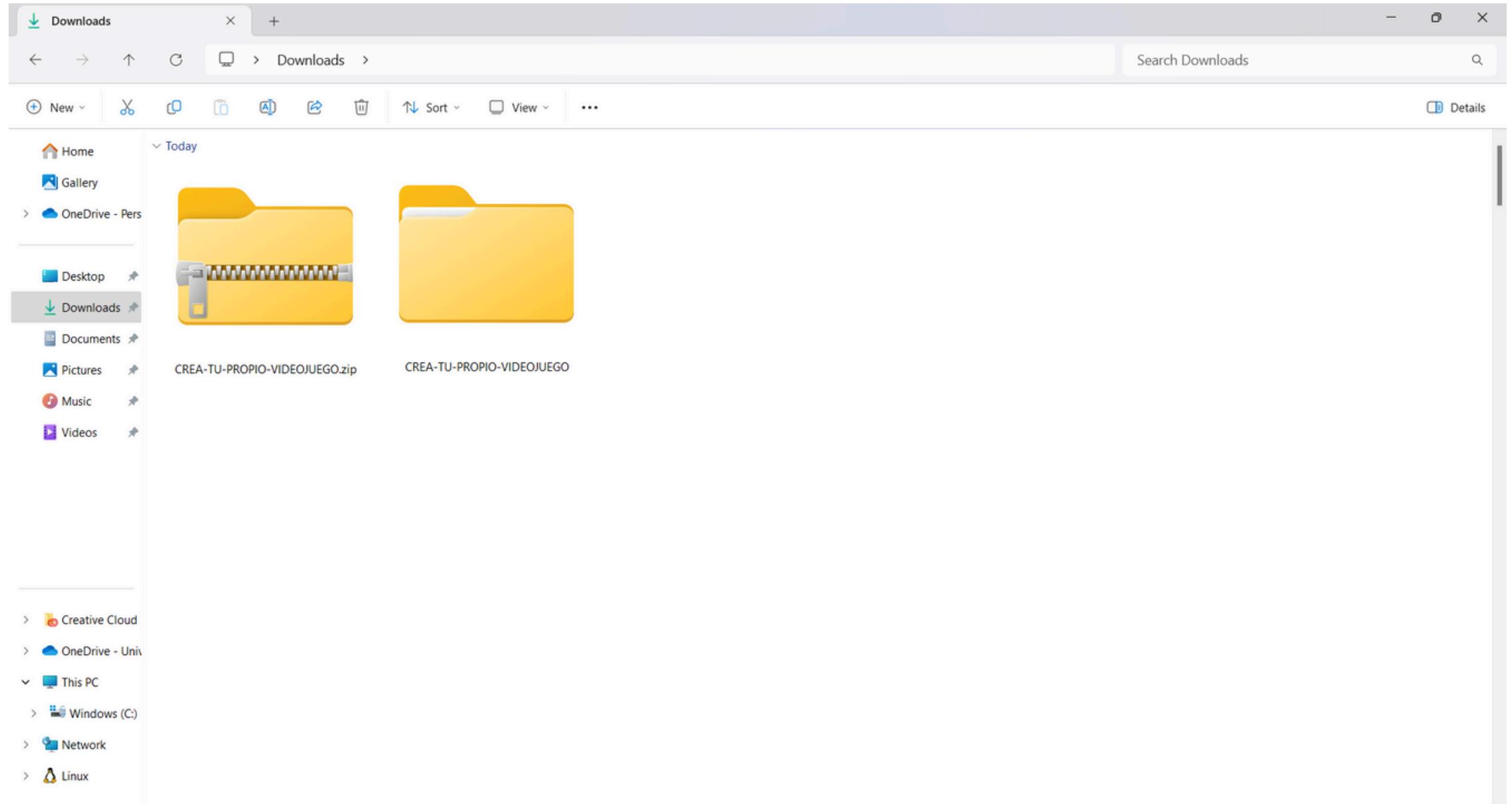
Nombre	Propietario	Última ...	Tamaño del:
Entregas	creauniandes	7:02 p.m.	—
CREA-TU-PROPIO-VIDEOJUEGO.zip	creauniandes	7:03 p.m.	43 MB

**En tu computador
podrás ver la
siguiente carpeta.**

**Descarga el archivo
**CREA-TU-PROPIO-
VIDEOJUEGO.zip****



Para iniciar

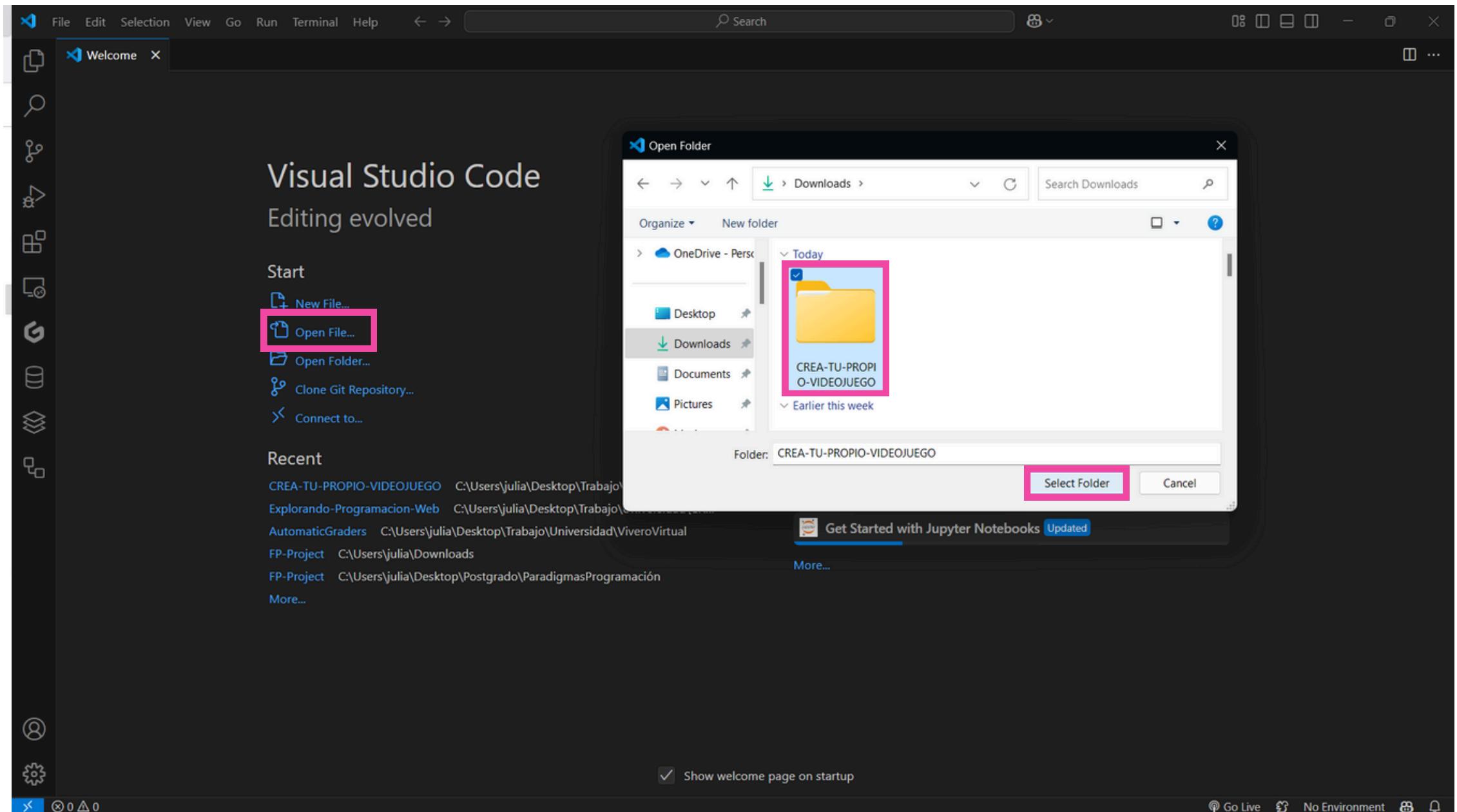


**Descomprime el
archivo que
descargaste en el
paso anterior.**

Tendrás la carpeta
**CREA-TU-PROPIO-
VIDEOJUEGO**



Para iniciar



Dentro de **Visual Studio Code**, abre la carpeta **CREA-TU-PROPIO-VIDEOJUEGO**. La misma del paso anterior.

¿Qué hay dentro de nuestro proyecto?

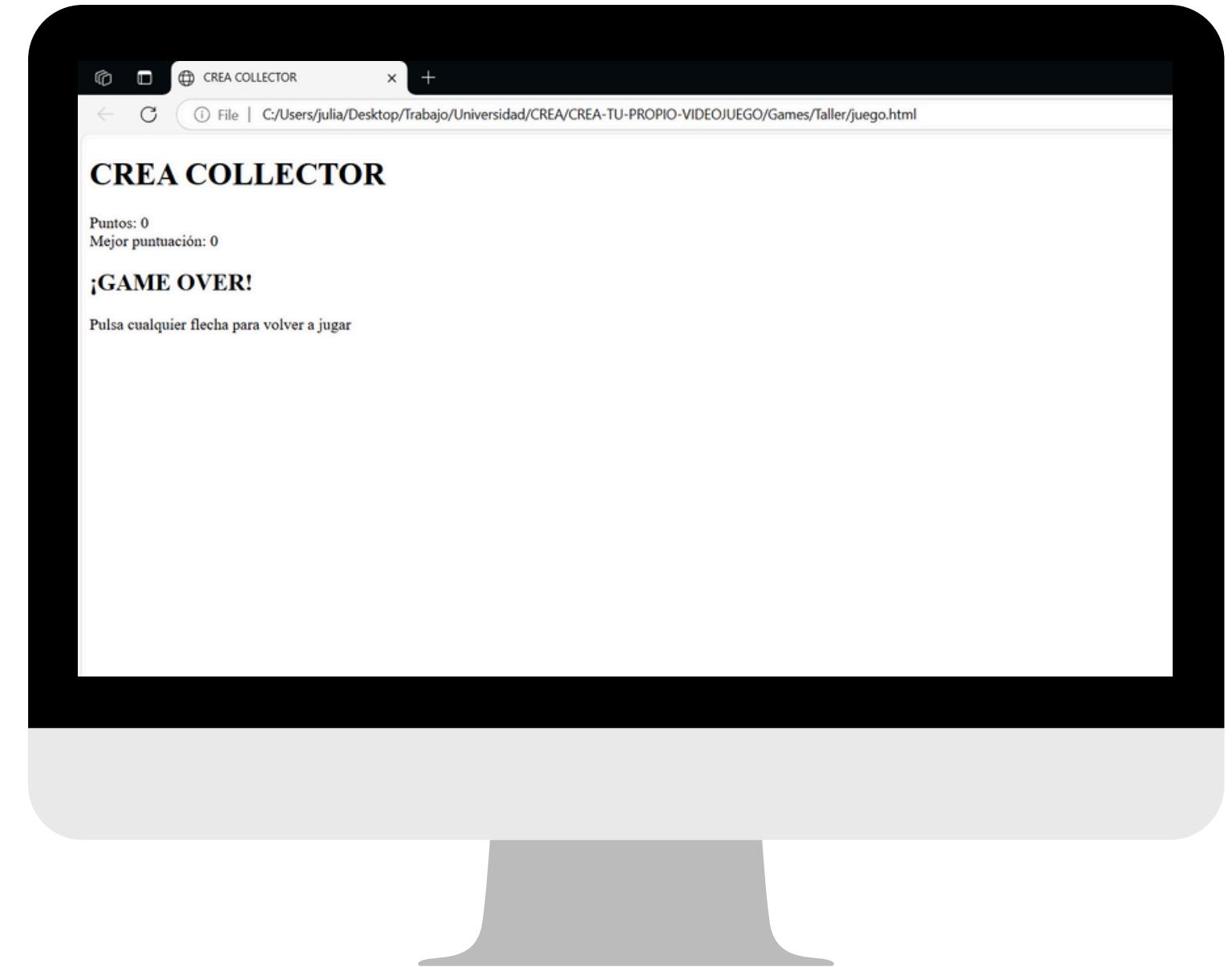
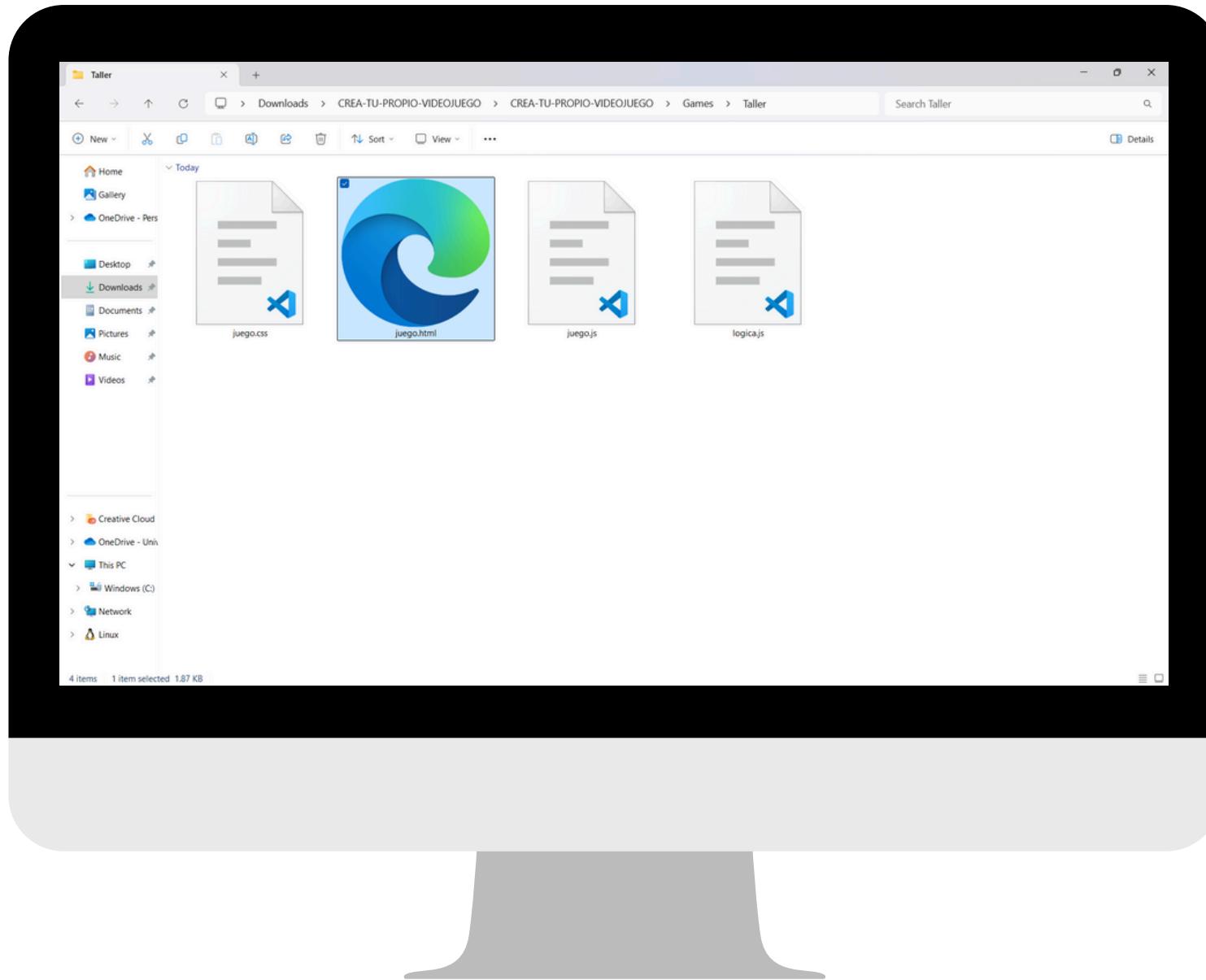
A screenshot of the Visual Studio Code interface. The title bar says "CREA-TU-PROPIO-VIDEOJUEGO". The left sidebar shows a file tree with a folder "CREA-TU-PROPIO-VIDEOJUEGO" containing "Assets", "Games", "Otros Juegos", and "Taller". Inside "Taller", there are files "juego.css", "juego.html" (which is selected), "juego.js", and "logica.js". A "README.md" file is also present. The main editor area displays the content of "juego.html", which includes a header with game details, a meta section, a title, and a body containing a game container with a score section and a game area.

El archivo **logica.js** contiene las funciones más complejas de nuestro juego. Por lo tanto, no vamos a modificarlo durante el taller.

Vamos a desarrollar todas nuestras actividades dentro de la carpeta **Taller**.

Dentro de esta, veremos 4 archivos **juego.css**, **juego.html**, **juego.js** y **logica.js**

¿Cómo probar nuestro juego?



Abre el archivo **juego.html** en el explorador,
haciendo doble click.



Conceptos básicos

¡**iCREA** tu Propio Videojuego!



¿Qué es HTML?

- Define la estructura de una página web.
- Funciona con contenedores.
- **<head>**: Provee información general, como el título.
- **<body>**: Representa el contenido de la página web.
- **<div>**: Contenedor genérico. Se usa para agrupar contenido.
- **id**: Identifica elementos únicos.

HyperText Markup Language

```

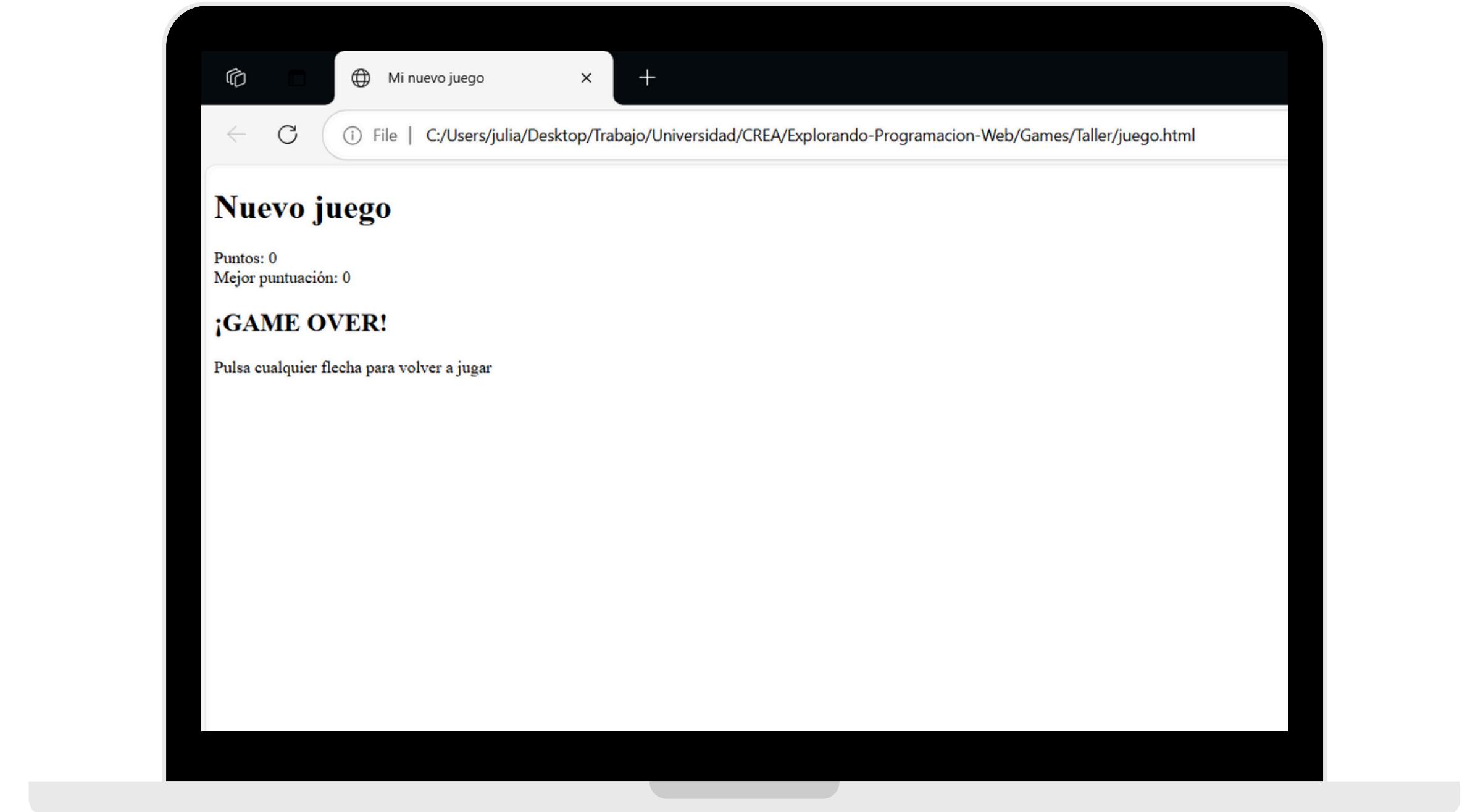
Games > Taller > juego.html > html > body > div#game-container
1  <!--
2   Archivo HTML con el esqueleto del taller ¡CREA tu propio videojuego!
3   Autor: CREA Uniandes (crea@uniandes.edu.co)
4   Fecha de creación: 21/01/2025
5   Última modificación: 07/03/2025
6  -->
7  <!DOCTYPE html>
8  <html lang="es">
9  <head>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <!-- Encabezado del juego -->
13 <!-- Tarea 1: Cambia el encabezado de tu página -->
14 <title>CREA COLLECTOR</title>
15 <!-- Conexión a juego css -->
16
17 </head>
18 <body>
19 <div id="game-container">
20 <!-- Título del juego -->
21 <!-- Tarea 1: Cambia el título de tu página -->
22 <h1>CREA COLLECTOR</h1>
23
24 <!-- Contenedor de puntaje -->
25 <div id="score-container">
26   Puntos: <span id="score">0</span>
27   <div id="high-score">Mejor puntuación: 0</div>
28 </div>
29
30 <!-- Área de juego -->
31 <div id="game-area">
32   <!-- Personaje jugable -->
33   <div id="character"></div>
34

```

¿Qué es HTML?

**Hagamos algo
sencillo...**

**En el archivo
juego.html, cambia
el nombre de la
página web y el título
de tu juego**



**Recuerda guardar siempre que realices un cambio para que tu
proyecto se actualice (ctrl+s)**



¿Qué es HTML?

Solución

Hagamos algo sencillo...

En el archivo **juego.html**, cambia el nombre de la página web y el título de tu juego

```
4  Fecha de creación: 21/01/2025
5  Última modificación: 07/03/2025
6  -->
7  <!DOCTYPE html>
8  <html lang="es">
9  <head>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <!-- Encabezado del juego -->
13 <!-- Tarea 1: Cambia el encabezado de tu página -->
14 <title>Nuevo Juego</title>
15 <!-- Conexión a juego css -->
16 </head>
17 <body>
18 <div id="game-container">
19 <!-- Título del juego -->
20 <!-- Tarea 1: Cambia el título de tu página -->
21 <h1>Mi nuevo Juego</h1>
22
23 <!-- Contenedor de puntaje -->
24 <div id="score-container">
25   Puntos: <span id="score">0</span>
26   <div id="high-score">Mejor puntuación: 0</div>
27 </div>
28
29 <!-- Área de juego -->
30 <div id="game-area">
```

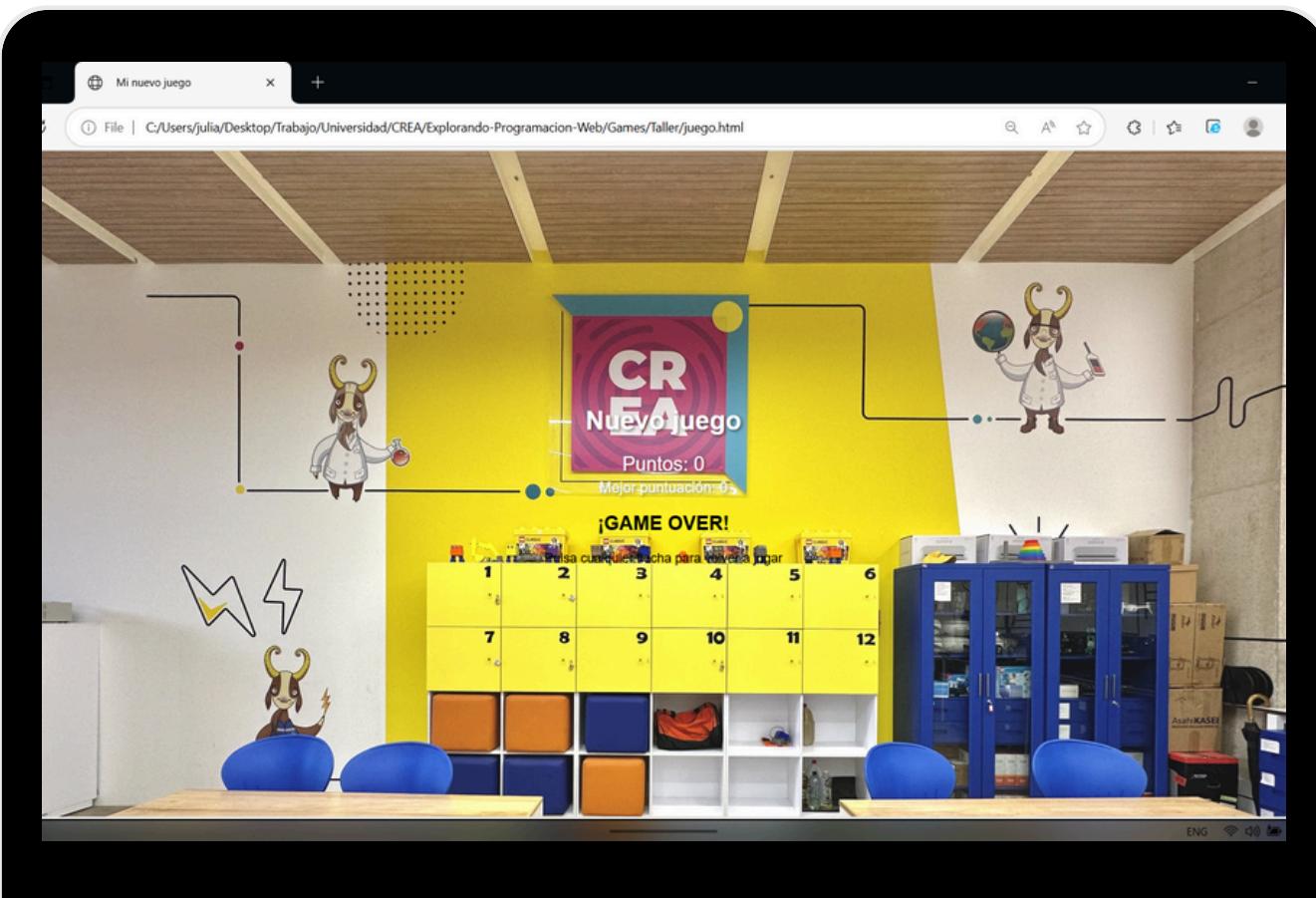
¿Qué es CSS?

```

Games > Taller > juego.css > ...
4  * Fecha de creación: 21/01/2025
5  * Última modificación: 07/03/2025
6  */
7
8  /* Estilos generales para la página */
9  body {
10    margin: 0;
11    padding: 0;
12    height: 100vh;
13    display: flex;
14    justify-content: center;
15    align-items: center;
16    font-family: Arial, sans-serif;
17    background-size: cover;
18    background-image: url(' ../../Assets/Wallpaper_CREA.jpg'); /* Tarea 2: Cambia la imagen de fondo */
19  }
20
21  /* Contenedor principal del juego */
22  #game-container {
23    text-align: center;
24  }
25
26  /* Estilo para el título del juego */
27  h1 {
28    color: white;
29    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
30  }
31
32  /* Estilo para el contador de puntos */
33  #score-container {
34    font-size: 24px;
35    color: white;
36    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
37    margin: 20px 0;
38  }
39
40  /* Estilo para el contador de mejor puntaje */
41  #high-score {
42    font-size: 18px;
43    margin-top: 5px;
44  }
45
46  /* Área de juego donde ocurre la acción */
47  #game-area {
48    width: 800px;
49    height: 600px;
50    position: relative;

```

Cascading Style Sheets

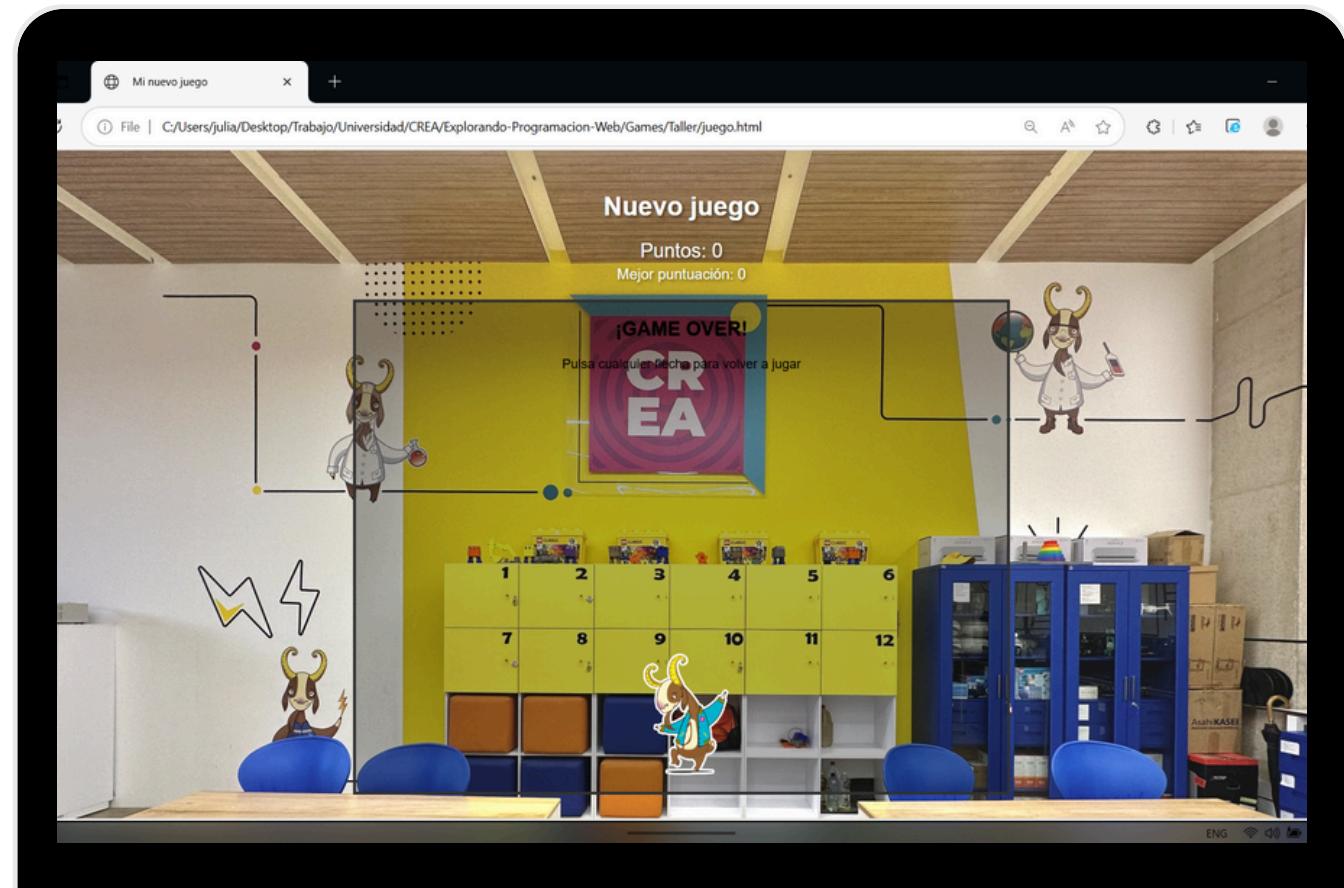


En este archivo se definen los estilos del título, el fondo y los demás elementos visuales que contenga la página web

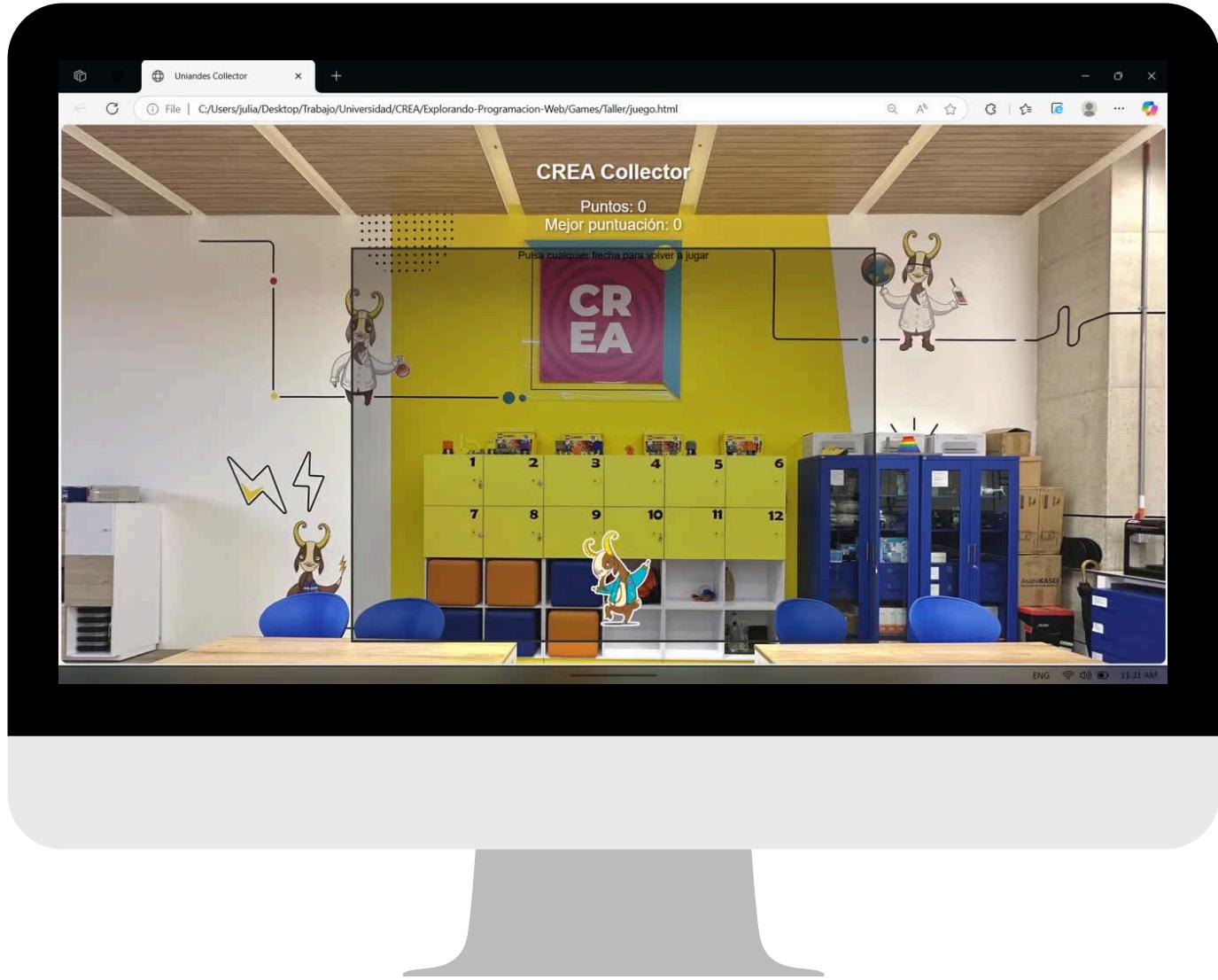
¿Qué es CSS?

```
43 /* Área de juego donde ocurre la acción */
44 #game-area {
45     width: 800px;
46     height: 600px;
47     position: relative;
48     border: 4px solid #333;
49     overflow: hidden;
50     background-color: #rgba(0, 0, 0, 0.3);
51 }
52
53 /* Estilo para el personaje controlado por el jugador */
54 #character {
55     width: 150px;
56     height: 150px;
57     position: absolute;
58     bottom: 20px;
59     left: 350px;
60     background-image: url('../Assets/Taller/Uniandes_Seneca.png'); /* Tarea 2: Cambia la imagen de fondo */
61     background-size: contain;
62     background-repeat: no-repeat;
63 }
64 }
```

Área de juego y Personaje



¿Qué es CSS?



¿Cómo conectar mi CSS?

```
1  <!--
2   Archivo HTML con el esqueleto del taller ¡CREA tu propio videojuego!
3   Autor: CREA Uniandes (crea@uniandes.edu.co)
4   Fecha de creación: 21/01/2025
5   Última modificación: 07/03/2025
6  -->
7  <!DOCTYPE html>
8  <html lang="es">
9  <head>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <!-- Encabezado del juego -->
13 <!-- Tarea 1: Cambia el encabezado de tu página -->
14 <title>Nuevo Juego</title>
15 <!-- Conexión a juego css -->
16 <link rel="stylesheet" href="juego.css">
17 </head>
18 <body>
19 <div id="game-container">
20 <!-- Título del juego -->
21 <!-- Tarea 1: Cambia el título de tu página -->
22 <h1>Mi nuevo Juego</h1>
23
24 <!-- Contenedor de puntaje -->
25 <div id="score-container">
26   Puntos: <span id="score">0</span>
27   <div id="high-score">Mejor puntuación: 0</div>
28 </div>
```



Para poder ver los estilos definidos en el archivo CSS es necesario que agreguemos la siguiente conexión en el encabezado.

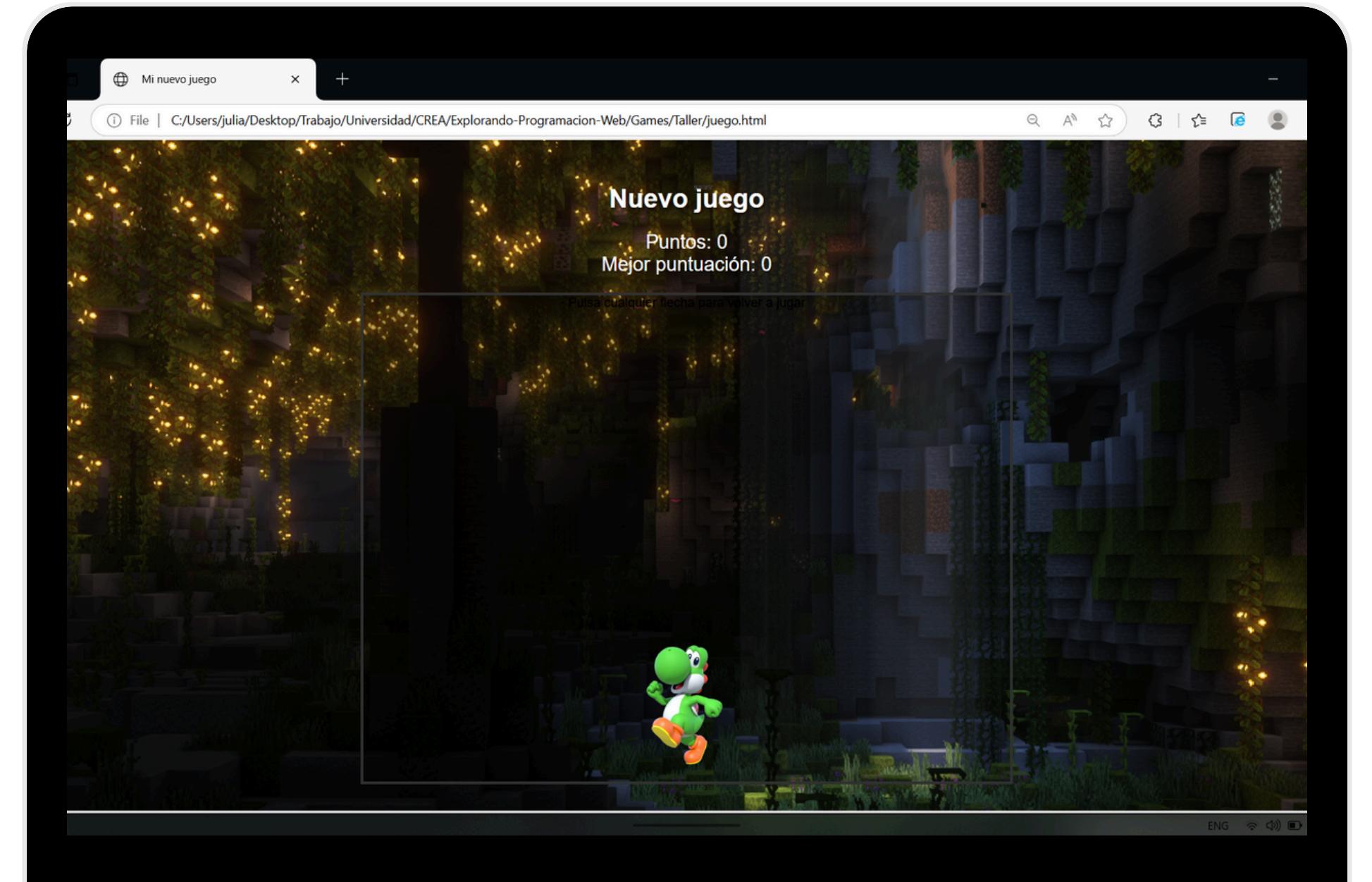


¿Qué es CSS?



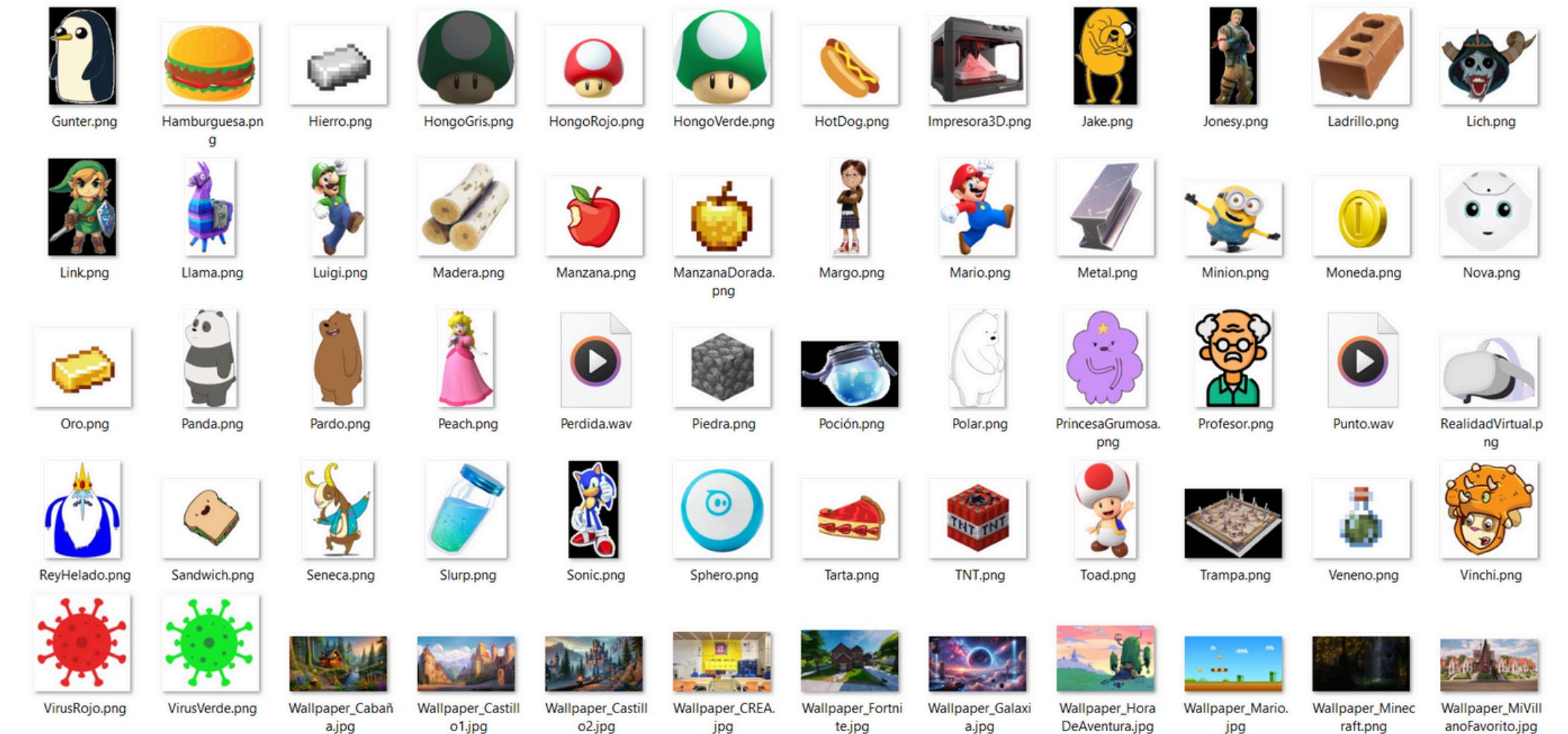
Hagamos algo sencillo...

Cambia la imagen de fondo y el personaje principal



¿Qué es CSS?

Para ello pueden utilizar cualquier imagen dentro de la carpeta **Assets**.



Cuando quieras usar una de las imágenes debes llamarla de la siguiente forma:

'**.../Assets/Gunter.png**'



¿Qué es CSS?

Solución

Hagamos algo sencillo...

Cambia la imagen de fondo y el personaje principal

```
8  /* Estilos generales para la página */
9  body {
10    margin: 0;
11    padding: 0;
12    height: 100vh;
13    display: flex;
14    justify-content: center;
15    align-items: center;
16    font-family: Arial, sans-serif;
17    background-size: cover;
18    background-image: url('../Assets/Wallpaper_Minecraft.png'); /* Tarea 2: Cambia la imagen de fondo */
19  }
20
21  /* Contenedor principal del juego */
22 > #game-container {
23
24 }
25
26  /* Estilo para el título del juego */
27 > h1 { ...
28 }
29
30  /* Estilo para el contador de puntos */
31 > #score-container { ...
32 }
33
34  /* Estilo para el contador de mejor puntaje */
35 > #high-score { ...
36 }
37
38  /* Área de juego donde ocurre la acción */
39 > #game-area { ...
40 }
41
42  /* Estilo para el personaje controlado por el jugador */
43 #character {
44   width: 150px;
45   height: 150px;
46   position: absolute;
47   bottom: 20px;
48   left: 350px;
49   background-image: url('../Assets/Yoshi.png'); /* Tarea 2: Cambia la imagen de fondo */
50   background-size: contain;
51   background-repeat: no-repeat;
52 }
```

¿Qué es JavaScript? ¡Démosle Vida a la Web!



- **JavaScript permite la interactividad.**
- **Controla el comportamiento de la página web.**
- **Ejecuta acciones en el navegador del usuario.**



¿Qué es JavaScript?

Conceptos

- **VARIABLES:** Almacenan datos que pueden cambiar a lo largo del programa.
- **CONSTANTES:** Almacenan datos que permanecen fijos durante toda la ejecución del programa.

```
9 // Variables globales del juego
10 let objetosEnJuego = []; // Lista de objetos activos en el juego
11 let gameArea = null;
12 let personaje = null;
13 let scoreDisplay = null;
14 let highScoreDisplay = null;
15 let heartsContainer = null;
16 let gameOverMessage = null;
17
18 // Variables de estado del juego
19 let score = 0; // Puntuación actual
20 let highScore = 0; // Mejor puntuación
21 let gameOver = false; // Estado de fin de juego
22 let vidas = 3; // Número de vidas
23 let intervaloCaida; // Intervalo para mover objetos
24 let intervaloCreacion; // Intervalo para crear objetos
25
26 // Sonidos
27 let pointSound;
28 let looseSound;
29 let gameOverSound;
30
31 // Constantes de juego
32 const ANCHO_JUEGO = 800;
33 const ALTO_JUEGO = 600;
34 const ANCHO_PERSONAJE = 110;
35 const ALTO_PERSONAJE = 150;
36
```

Vamos a analizar el archivo logica.js. Sin embargo este no lo vamos a modificar porque contiene funciones avanzadas para la interacción del juego.

¿Qué es JavaScript?

Conceptos

- **Funciones:** Las funciones agrupan código reutilizable para ejecutar acciones específicas.

¿Qué hace esta función?

```
481 function reiniciarJuego() {  
482     console.log("Reiniciando juego");  
483  
484     // Limpiar objetos existentes  
485     while (objetosEnJuego.length > 0) {  
486         if (objetosEnJuego[0].parentNode) {  
487             gameArea.removeChild(objetosEnJuego[0]);  
488         }  
489         objetosEnJuego.shift();  
490     }  
491  
492     // Reiniciar variables  
493     score = 0;  
494     gameOver = false;  
495     if (scoreDisplay) {  
496         scoreDisplay.textContent = score;  
497     }  
498     if (gameOverMessage) {  
499         gameOverMessage.classList.add('hidden');  
500     }  
501  
502     // Reiniciar vidas  
503     vidas = VIDAS_INICIALES;  
504  
505     // Actualizar los corazones  
506     actualizarVidasVisualmente();  
507  
508     // Reiniciar posición del personaje  
509     if (personaje) {  
510         personaje.style.left = "350px";  
511     }  
512  
513     // Iniciar el juego de nuevo  
514     iniciarIntervalos();  
515 }
```



¿Qué es JavaScript?

Conceptos



- **Eventos:** Los eventos permiten que una página web reaccione a acciones del usuario, como presionar teclas, hacer clic o mover el mouse.

```
144 function configurarControles(velocidad) {
145   document.addEventListener('keydown', function(event) { <<<
146     // Verificar que el personaje existe
147     if (!personaje) {
148       console.error("Error: Personaje no encontrado");
149       return;
150     }
151
152     // Si el juego terminó, reiniciar con cualquier flecha
153     if (gameOver && (event.key === 'ArrowLeft' || event.key === 'ArrowRight')) {
154       reiniciarJuego(); <<<
155       return;
156     }
157
158     // Obtener posición actual del personaje de manera más robusta
159     let posX;
160     // Si el personaje ya tiene estilo left definido, usarlo
161     if (personaje.style.left) {
162       posX = parseInt(personaje.style.left);
163     }
164     // Si no, obtener el valor computado
165     else {
166       const computedStyle = window.getComputedStyle(personaje);
167       posX = parseInt(computedStyle.left);
168     }
169
170     // Si no se pudo obtener un valor válido, usar el valor predeterminado
171     if (isNaN(posX)) {
172       posX = 350;
173     }
174
175     // Ancho del área de juego y del personaje
176     const anchoJuego = gameArea.offsetWidth;
177     const anchoPersonaje = personaje.offsetWidth;
178
179     // Mover a la izquierda o derecha según la tecla presionada
180     if (event.key === 'ArrowLeft') { <<<
181       posX = Math.max(0, posX - velocidad);
182     } else if (event.key === 'ArrowRight') {
183       posX = Math.min(anchoJuego - anchoPersonaje, posX + velocidad);
184     }
185
186     // Establecer la nueva posición
187     personaje.style.left = posX + 'px';
188   });
189 }
```

¿Qué es JavaScript?

Hagamos algo sencillo para que nuestro juego se mueva...

```
Games > Taller > juego.js > ...
1  /**
2   * Archivo JavaScript personalizable del taller ¡CREA tu propio videojuego!
3   * Autor: CREA Uniandes (crea@uniandes.edu.co)
4   * Fecha de creación: 21/01/2025
5   * Última modificación: 07/03/2025
6   */
7
8  VELOCIDAD_PERSONAJE = 0; // Tarea 3: Cambiar la velocidad del personaje (ejemplo 20)
9
10 VELOCIDAD_CAIDA = 0; // Tarea 3: Cambiar la velocidad de caída de los objetos (ejemplo 3)
11
12 INTERVALO_CREACION = 0; // Tarea 3: Cambiar la frecuencia de aparición de objetos (ejemplo 1000)
13
```

Cambia las variables de velocidad del personaje, caída de objetos y creación de objetos

¿Qué es JavaScript?

Hagamos algo sencillo para que nuestro juego se mueva...

```
Games > Solución > juego.js > ...
1  /**
2   * Archivo JavaScript personalizable del taller ¡CREA tu propio videojuego!
3   * Autor: CREA Uniandes (crea@uniandes.edu.co)
4   * Fecha de creación: 21/01/2025
5   * Última modificación: 07/03/2025
6   */
7
8  VELOCIDAD_PERSONAJE = 20; // Tarea 3: Cambiar la velocidad del personaje (ejemplo 20)
9
10 VELOCIDAD_CAIADA = 3; // Tarea 3: Cambiar la velocidad de caída de los objetos (ejemplo 3)
11
12 INTERVALO_CREACION = 1000; // Tarea 3: Cambiar la frecuencia de aparición de objetos (ejemplo 1000)
13
```

Cambia las variables de velocidad del personaje, caída de objetos y creación de objetos

¿Qué es JavaScript?

Recuerda la conexión

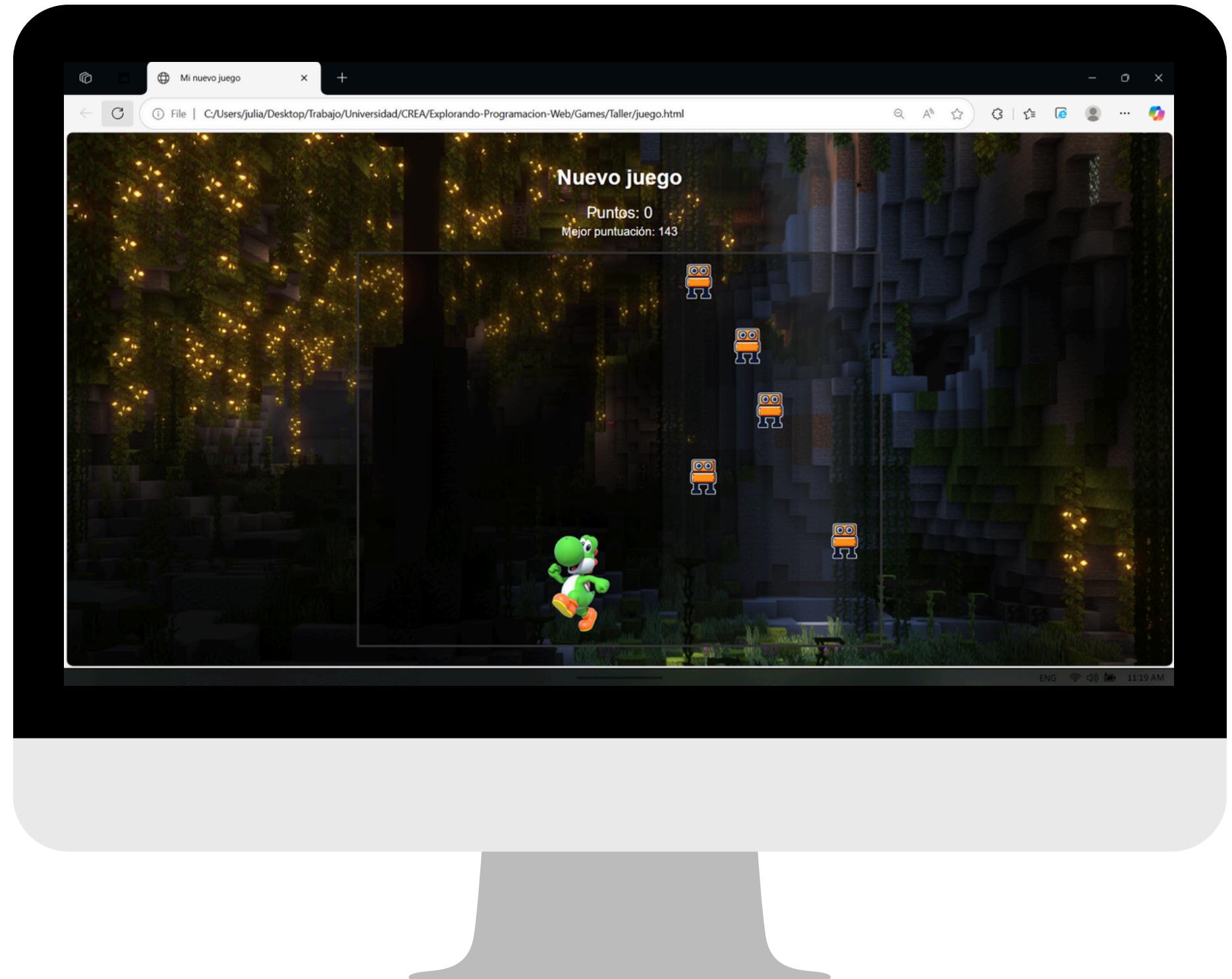
```
42      <!-- Contenedor de vidas -->
43      <!-- Tarea 7: Completa con la cantidad de vidas totales -->
44      <div id="hearts-container">
45          </div>
46      </div>
47
48      <!-- Sonidos del juego -->
49      <audio id="sonidoPunto" src="../../Assets/Punto.wav" preload="auto"></audio>
50      <audio id="sonidoPerdida" src="../../Assets/Perdida.wav" preload="auto"></audio>
51      <audio id="sonidoGameOver" src="../../Assets/GameOver.wav" preload="auto"></audio>
52
53      <!-- Conexión al archivo de lógica, luego el de juego -->
54      <!-- Tarea 3: Conecta tus archivos JavaScript -->
55      <script src="logica.js"></script> <<<
56      <script src="juego.js"></script>
57  </body>
58  </html>
59
```

<script src="logica.js"></script>
<script src="juego.js"></script>

Se conectan en ese orden porque **juego.js** depende de **logica.js**

¿Qué es JavaScript?

Resultado



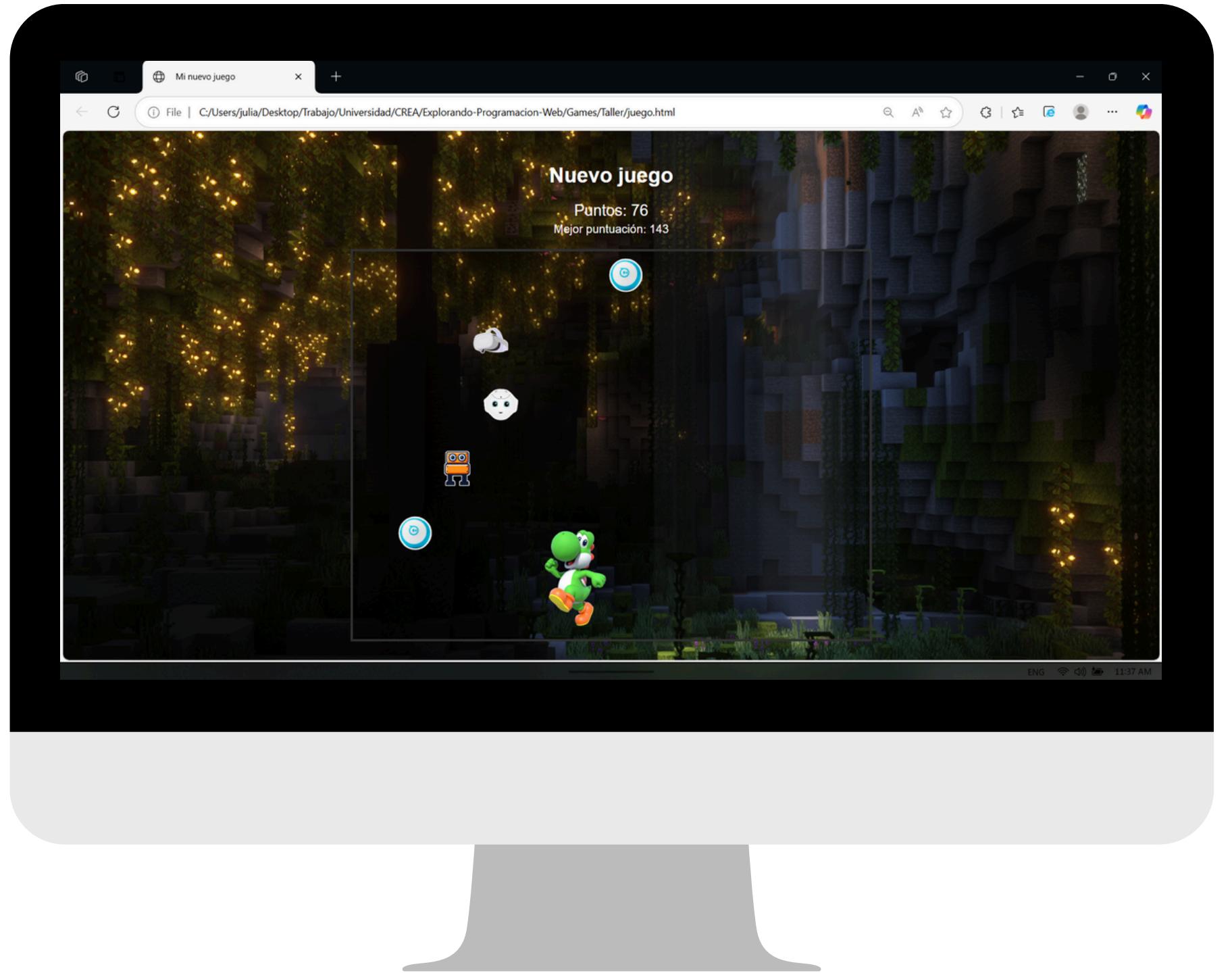


¡A programar!

¡**iCREA** tu Propio Videojuego!

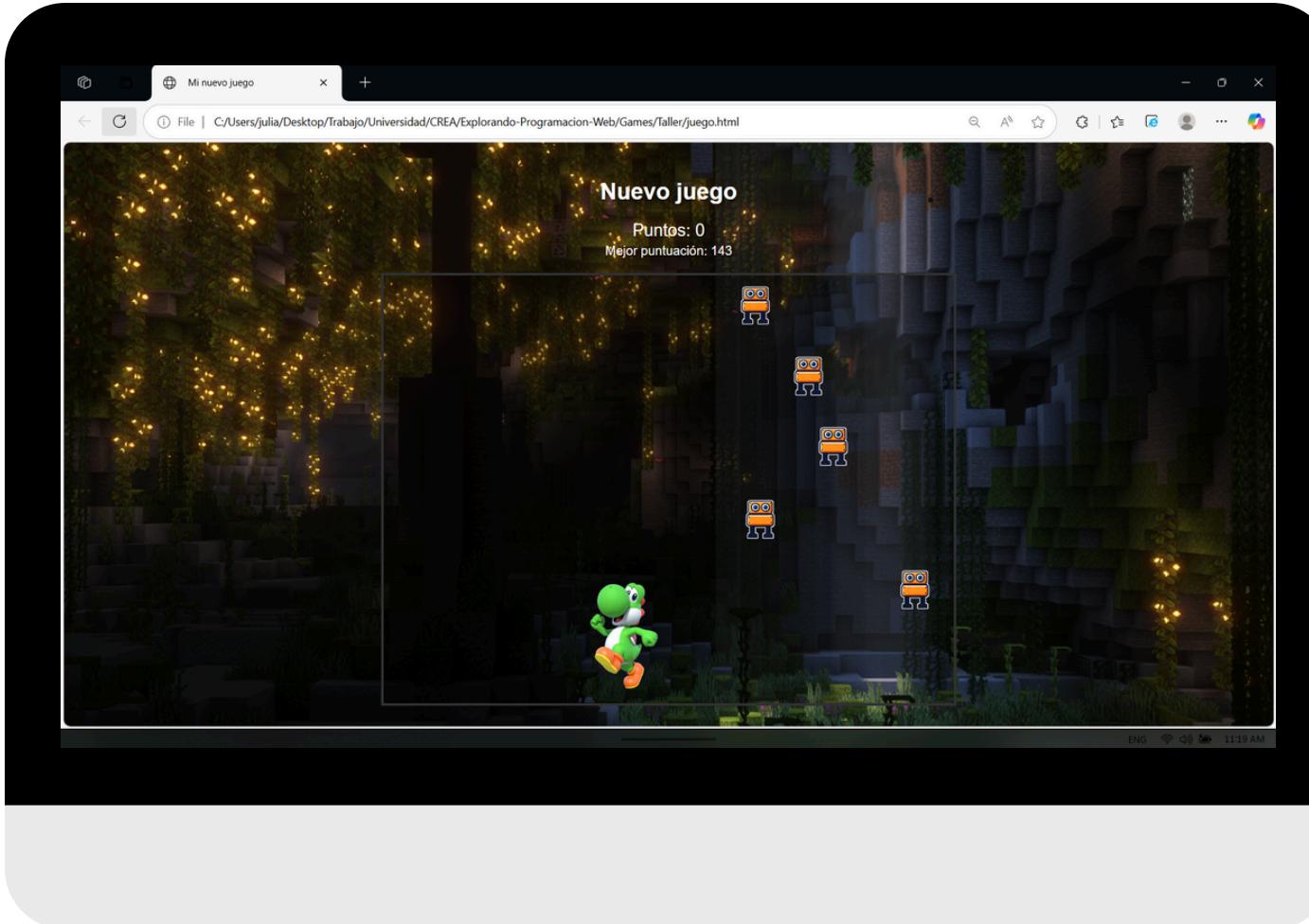


Objetos en caída



Objetos en caída

juego.js

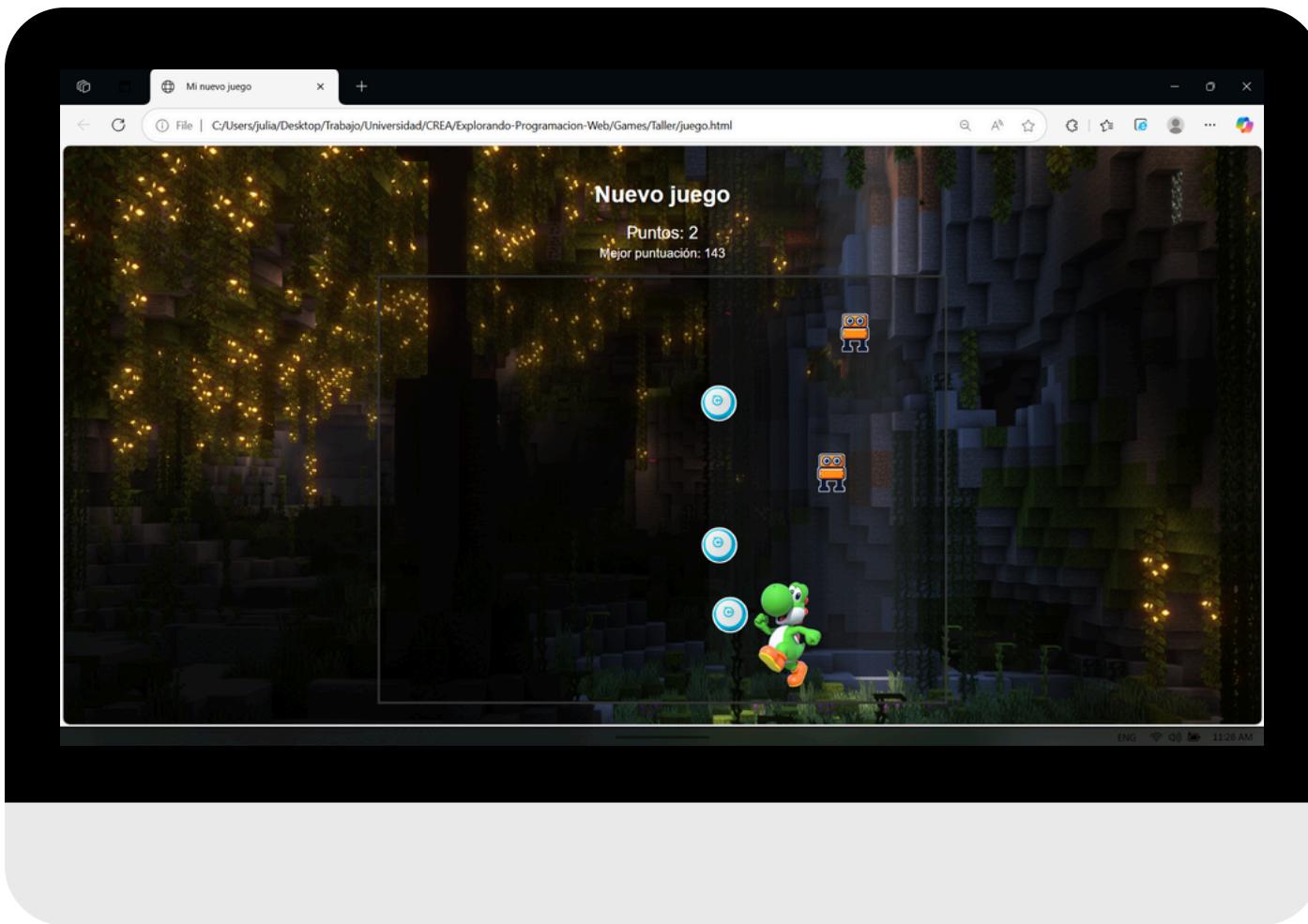


```
17 // Cada objeto debe tener: tipo, clase, puntos, y probabilidad
18 // La suma de todas las probabilidades debe ser menor o igual a 1
19 OBJETOS = [
20   {
21     tipo: 'bot',          // Nombre del objeto
22     clase: 'objeto-bot', // Clase CSS para dar estilo
23     puntos: 1,           // Puntos que da al recogerlo
24     probabilidad: 0.25   // Probabilidad de que aparezca (entre 0 y 1)
25   }
26   // Tarea 4: Agrega 3 objetos que den puntos positivos
27 ]
```

juego.css

```
78 /* Estilos para los objetos que caen */
79 /* Tarea 4: Agrega 3 objetos que den puntos positivos */
80 .objeto-bot {
81   background-image: url('../Assets/CREABOT.png');
82 }
```

Objetos en caída



Ayuda

juego.js

```
// Cada objeto debe tener: tipo, clase, puntos, y probabilidad
// La suma de todas las probabilidades debe ser menor o igual a 1
OBJETOS = [
  {
    tipo: 'bot',          // Nombre del objeto
    clase: 'objeto-bot', // Clase CSS para dar estilo
    puntos: 1,            // Puntos que da al recogerlo
    probabilidad: 0.25   // Probabilidad de que aparezca (entre 0 y 1)
  },
  // Tarea 4: Agrega 3 objetos que den puntos positivos
  {
    tipo: 'sphero',        // Nombre del objeto
    clase: 'objeto-sphero', // Clase CSS para dar estilo
    puntos: 2,              // Puntos que da al recogerlo
    probabilidad: 0.20     // Probabilidad de que aparezca (entre 0 y 1)
  }
]
```

juego.css

```
/* Estilos para los objetos que caen */
/* Tarea 4: Agrega 3 objetos que den puntos positivos */
.objeto-bot {
  background-image: url('../Assets/CREABOT.png');
}
.objeto-sphero {
  background-image: url('../Assets/Sphero.png');
}
```

Objetos en caída

juego.css

```

78  /* Estilos para los objetos que caen */
79  /* Tarea 4: Agrega 3 objetos que den puntos positivos */
80  .objeto-bot {
81    background-image: url('../Assets/CREABOT.png');
82  }
83  .objeto-sphero {
84    background-image: url('../Assets/Sphero.png');
85  }
86  .objeto-vr {
87    background-image: url('../Assets/RealidadVirtual.png');
88  }
89  .objeto-nova {
90    background-image: url('../Assets/Nova.png');
91  }

```

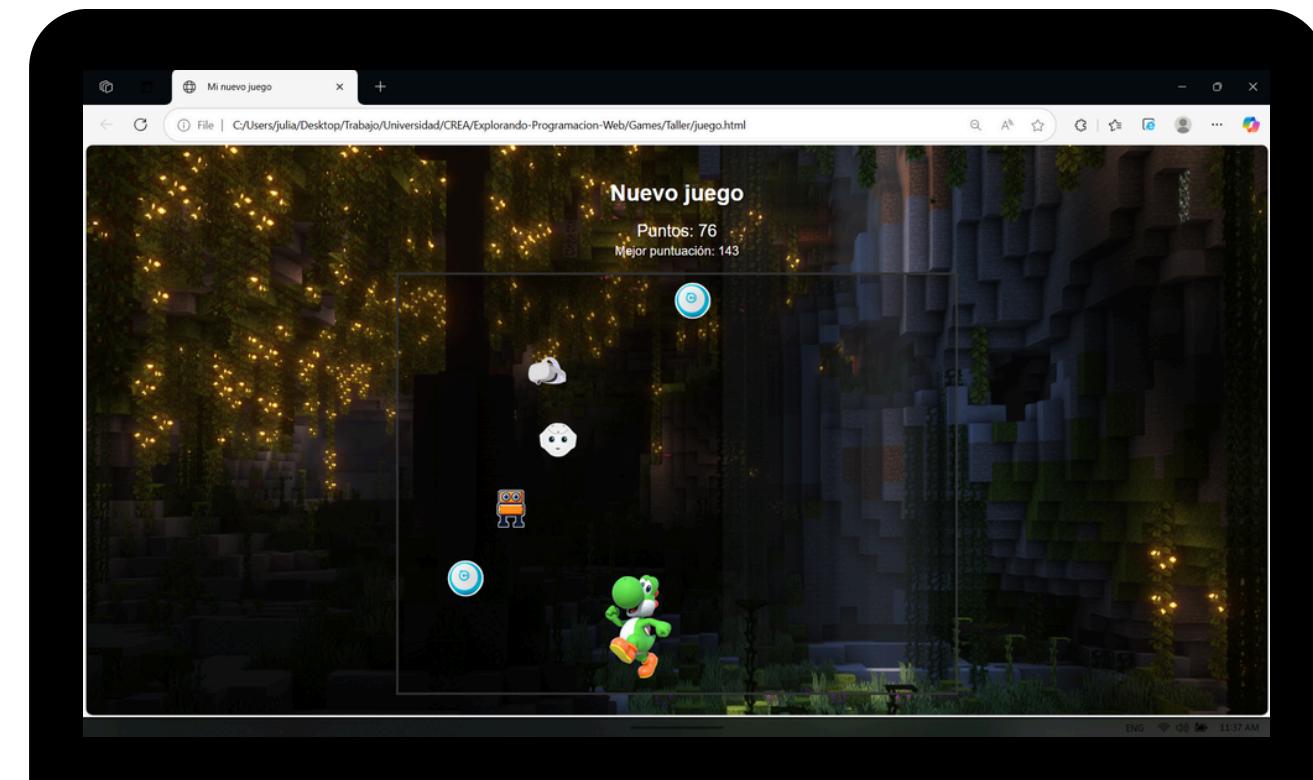
juego.js

```

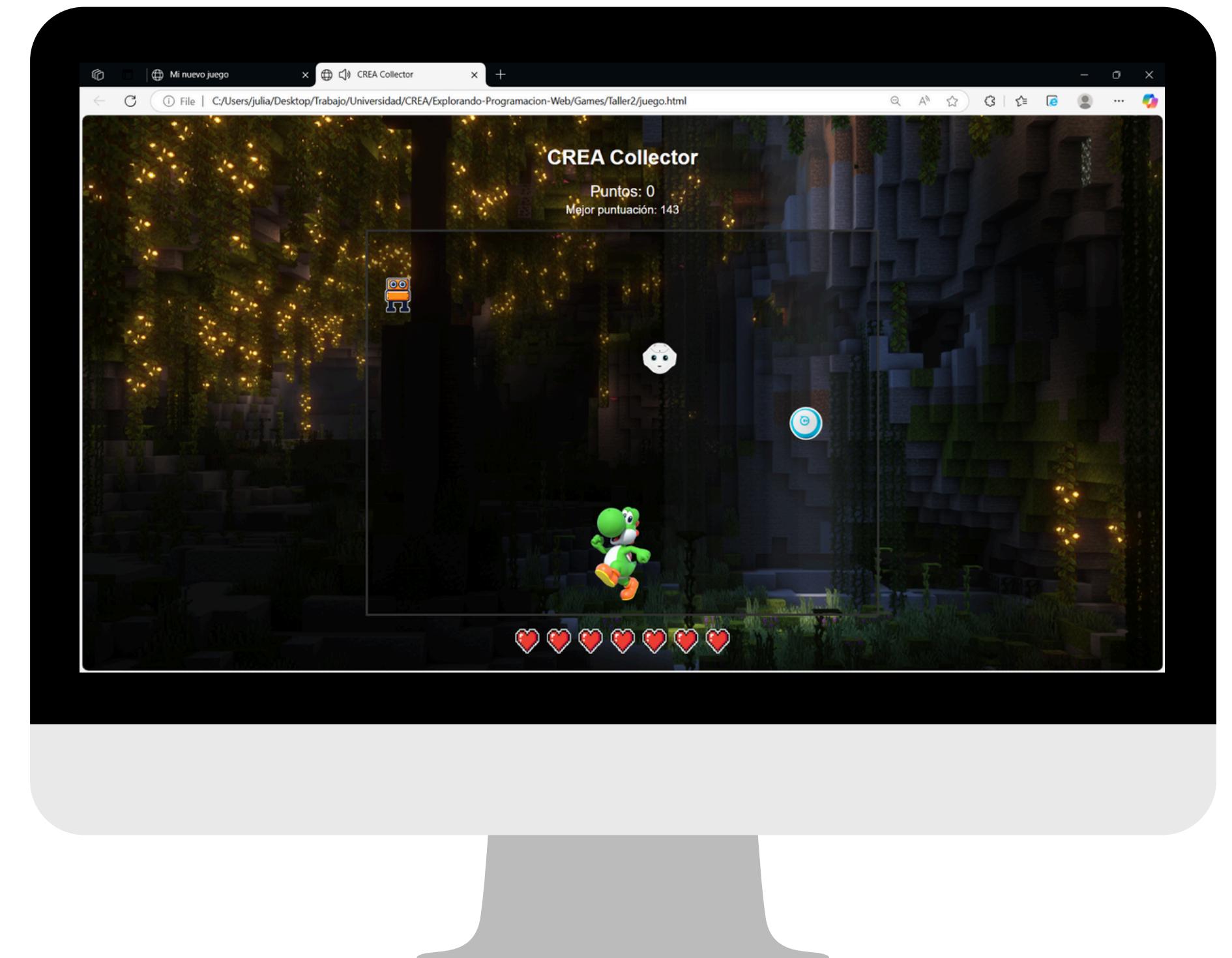
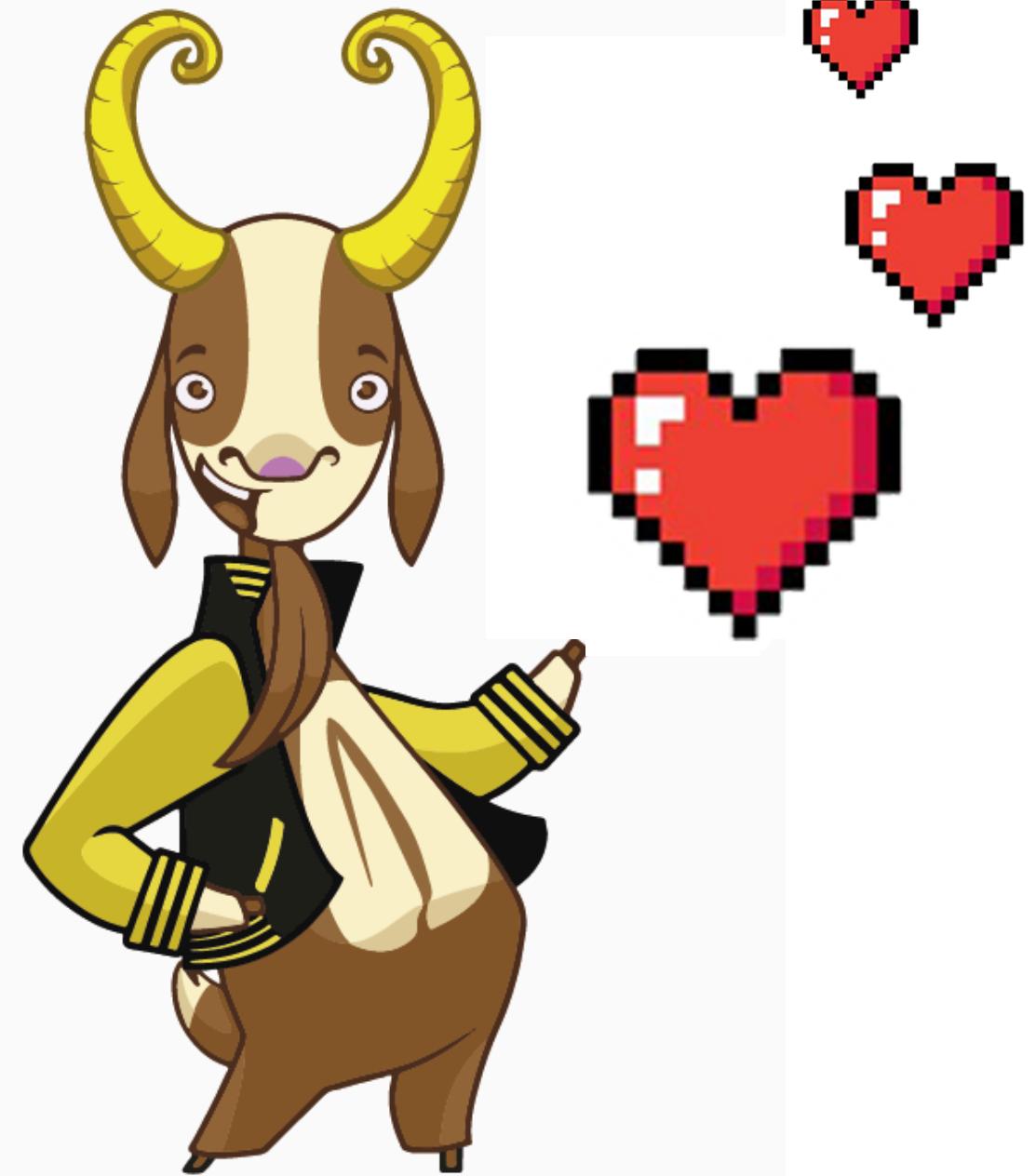
// Cada objeto debe tener: tipo, clase, puntos, y probabilidad
// La suma de todas las probabilidades debe ser menor o igual a 1
OBJETOS = [
{
  tipo: 'bot',          // Nombre del objeto
  clase: 'objeto-bot', // Clase CSS para dar estilo
  puntos: 1,            // Puntos que da al recogerlo
  probabilidad: 0.25   // Probabilidad de que aparezca (entre 0 y 1)
},
// Tarea 4: Agrega 3 objetos que den puntos positivos
{
  tipo: 'sphero',        // Nombre del objeto
  clase: 'objeto-sphero', // Clase CSS para dar estilo
  puntos: 2,              // Puntos que da al recogerlo
  probabilidad: 0.20    // Probabilidad de que aparezca (entre 0 y 1)
},
{
  tipo: 'realidad virtual',      // Nombre del objeto
  clase: 'objeto-vr', // Clase CSS para dar estilo
  puntos: 5,            // Puntos que da al recogerlo
  probabilidad: 0.15   // Probabilidad de que aparezca (entre 0 y 1)
},
{
  tipo: 'nova',          // Nombre del objeto
  clase: 'objeto-nova', // Clase CSS para dar estilo
  puntos: 10,             // Puntos que da al recogerlo
  probabilidad: 0.10   // Probabilidad de que aparezca (entre 0 y 1)
}
];

```

Possible solution



Vidas



Vidas

Definamos la cantidad de **vidas** iniciales y **vidas en total** que va a tener nuestro personaje

juego.js

```
8  // VELOCIDAD_PERSONAJE = 20; // Tarea 3: Cambiar la velocidad del personaje (ejemplo 20)
9
10 // VELOCIDAD_CAIDA = 3; // Tarea 3: Cambiar la velocidad de caída de los objetos (ejemplo 3)
11
12 // INTERVALO_CREACION = 1000; // Tarea 3: Cambiar la frecuencia de aparición de objetos (ejemplo 1000)
13
14 // Tarea 5: Crea el número de vidas iniciales y máximas (VIDAS_INICIALES, VIDAS_TOTAL)
15
16
```

Vidas

Se crean las variables **VIDAS_INICIALES** y **VIDAS_TOTAL** para guardar el número de vidas.

Solución

juego.js

```
VELOCIDAD_PERSONAJE = 20; // Tarea 3: Cambiar la velocidad del personaje (ejemplo 20)

VELOCIDAD_CAIADA = 3; // Tarea 3: Cambiar la velocidad de caída de los objetos (ejemplo 3)

INTERVALO_CREACION = 1000; // Tarea 3: Cambiar la frecuencia de aparición de objetos (ejemplo 1000)

// Tarea 5: Crea el número de vidas iniciales y máximas (VIDAS_INICIALES, VIDAS_TOTAL)
VIDAS_INICIALES = 7;

VIDAS_TOTAL = 7;
```



Vidas

**Hay que completar
la función que crea
los corazones
iniciales.**

juego.js

```
39 // Tarea 6: Completa la función para crear los corazones
40 function crearCorazones() {
41     /*
42     const contenedor = document.getElementById('hearts-container');
43
44     contenedor.innerHTML = '';
45
46     // Aquí crearemos tantos corazones como VIDAS INICIALES hayamos definido
47     for (let i = 0; i < VIDAS; i++) { // Tarea 6: Reemplaza VIDAS por la variable creada
48         const corazon = document.createElement('img');
49         corazon.src = ''; // Tarea 6: Agrega la imagen del corazón rojo
50         corazon.classList.add('heart');
51         contenedor.appendChild(corazon);
52     }
53     */
54 }
```

La función está comentada. Para “activarla” deberás borrar las líneas 41 y 53.



Vidas

Se reemplaza VIDAS
por
VIDAS_INICIALES y
se agrega la imagen
de un **corazón rojo** a
las vidas.

Solución

juego.js

```
// Tarea 6: Completa la función para crear los corazones
function crearCorazones() {
    const contenedor = document.getElementById('hearts-container');

    contenedor.innerHTML = '';

    // Aquí crearemos tantos corazones como VIDAS_INICIALES hayamos definido
    for (let i = 0; i < VIDAS_INICIALES; i++) { // Tarea 6: Reemplaza VIDAS por la variable creada
        const corazon = document.createElement('img');
        corazon.src = '../../../../../Assets/Taller/Uniandes_Heart.png'; // Tarea 6: Agrega la imagen del corazón rojo
        corazon.classList.add('heart');
        contenedor.appendChild(corazon);
    }
}
```

La imagen del corazón también se puede cambiar si se desea. Pero esto tiene una implicación adicional que explicaremos más adelante.

Vidas

juego.html

Completa el **contenedor** de las vidas en el **esqueleto** de la página web.

Debes agregar el mismo número de líneas como vidas iniciales hayas creado.

¿Qué ponemos dentro del **src** en una imagen?

```
35      <!-- Mensaje de Game Over -->
36      <div id="game-over-message" class="hidden">
37          <h2>¡GAME OVER!</h2>
38          <p>Pulsa cualquier flecha para volver a jugar</p>
39      </div>
40  </div>
41
42  <!-- Contenedor de vidas -->
43  <!-- Tarea 7: Completa con la cantidad de vidas totales -->
44  <div id="hearts-container">
45  </div>
46 </div>
47
```



```
<div id="hearts-container">
    <img src="" class="heart">
    ...
</div>
```



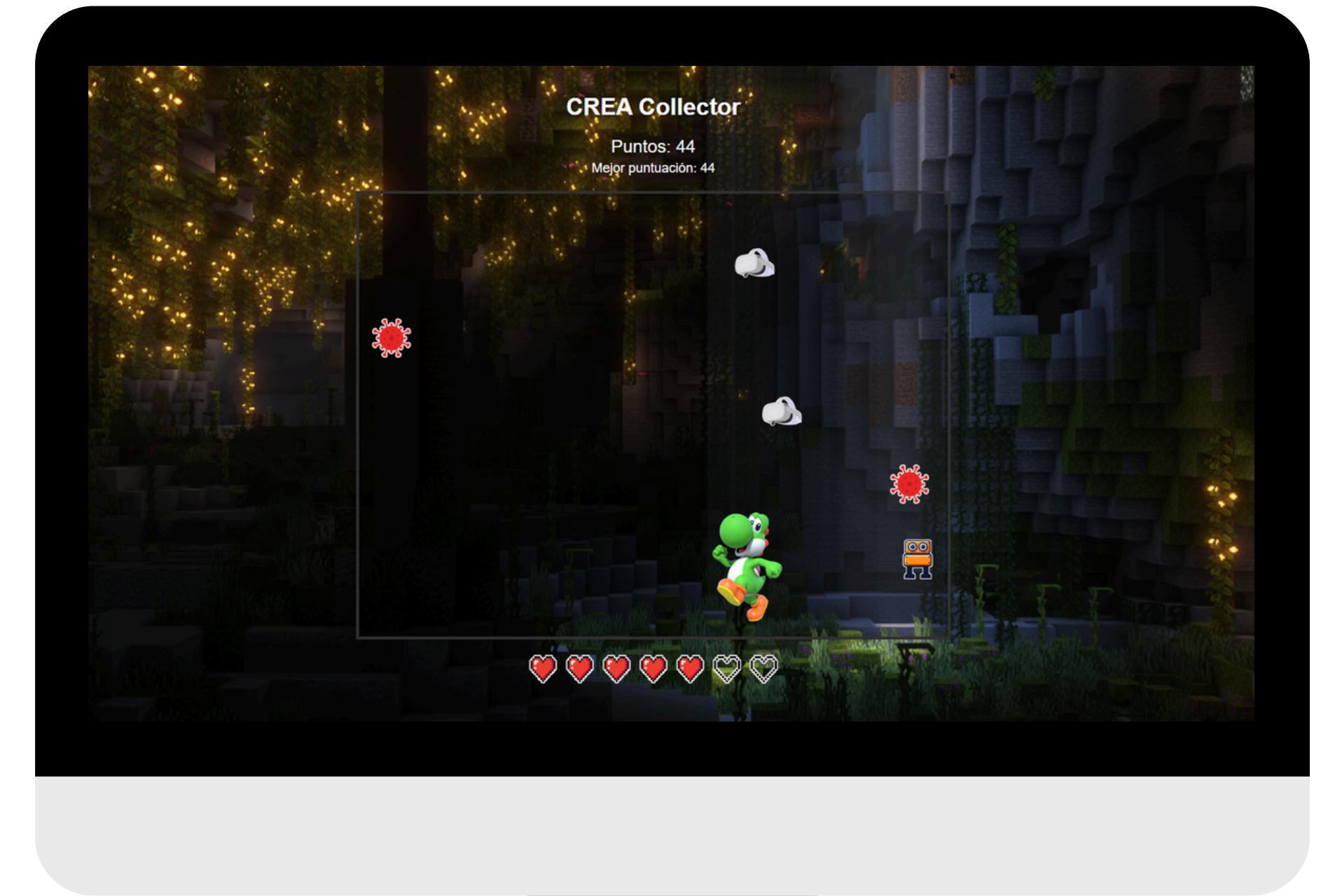
Vidas

Solución

Se completa el contenedor con las imágenes del corazón que definimos para los corazones en la tarea anterior.

juego.html

Perder Vidas



Perder Vidas

juego.js

Creemos un objeto que le quite vidas a nuestro personaje.

Ayuda: Los puntos negativos quitan vidas

```
20 OBJETOS = [
21 {
22   tipo: 'bot',           // Nombre del objeto
23   clase: 'objeto-bot',  // Clase CSS para dar estilo
24   puntos: 1,             // Puntos que da al recogerlo
25   probabilidad: 0.25    // Probabilidad de que aparezca (entre 0 y 1)
26 },
27 // Tarea 4: Agrega 3 objetos que den puntos positivos
28 {
29   tipo: 'sphero',        // Nombre del objeto
30   clase: 'objeto-sphero', // Clase CSS para dar estilo
31   puntos: 2,             // Puntos que da al recogerlo
32   probabilidad: 0.20    // Probabilidad de que aparezca (entre 0 y 1)
33 },
34 {
35   tipo: 'realidad virtual', // Nombre del objeto
36   clase: 'objeto-vr',     // Clase CSS para dar estilo
37   puntos: 5,              // Puntos que da al recogerlo
38   probabilidad: 0.15    // Probabilidad de que aparezca (entre 0 y 1)
39 },
40 {
41   tipo: 'nova',          // Nombre del objeto
42   clase: 'objeto-nova',  // Clase CSS para dar estilo
43   puntos: 10,             // Puntos que da al recogerlo
44   probabilidad: 0.10    // Probabilidad de que aparezca (entre 0 y 1)
45 }
46 // Tarea 8: Agrega 1 objeto que quite vidas
47 // Ayuda: Los puntos negativos quitan vidas
48 ];
49
50 ];
```

Perder Vidas

En este ejemplo creamos un objeto que **quita 1 vida**. Es de **tipo virus** y su **clase** es '**objeto-virus**'.

¡Pueden hacer que cualquier objeto quite vidas!

Possible solución juego.js

```
39  },
40  {
41    tipo: 'nova',           // Nombre del objeto
42    clase: 'objeto-nova',  // Clase CSS para dar estilo
43    puntos: 10,            // Puntos que da al recogerlo
44    probabilidad: 0.10    // Probabilidad de que aparezca (entre 0 y 1)
45  },
46  // Tarea 8: Agrega 1 objeto que quite vidas
47  // Ayuda: Los puntos negativos quitan vidas
48  {
49    tipo: 'virus',          // Nombre del objeto
50    clase: 'objeto-virus', // Clase CSS para dar estilo
51    puntos: -1,            // Puntos que da al recogerlo
52    probabilidad: 0.15    // Probabilidad de que aparezca (entre 0 y 1)
53  }
54 ];
```



Perder Vidas

juego.css

Usando la clase que acabas de crear, agrega la imagen del objeto que va a quitar las vidas.

```
78  /* Estilos para los objetos que caen */
79  /* Tarea 4: Agrega 3 objetos que den puntos positivos */
80  .objeto-bot {
81      background-image: url('../Assets/CREABOT.png');
82  }
83  .objeto-sphero {
84      background-image: url('../Assets/Sphero.png');
85  }
86  .objeto-vr {
87      background-image: url('../Assets/RealidadVirtual.png');
88  }
89  .objeto-nova {
90      background-image: url('../Assets/Nova.png');
91  }
92
93  /* Tarea 9: Agrega el objeto que quita las vidas */
94  /* Puede ser cualquier objeto que deseas */
95  /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */
96  |
```

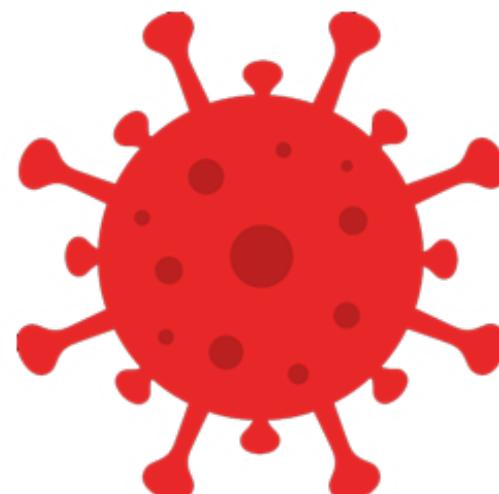
Perder Vidas

En esta solución el objeto que quita vidas es un **virus**.

Sin embargo, ustedes pueden asignar la imagen que quieran.

Possible solución juego.css

```
89 .objeto-nova {  
90 |   background-image: url('../Assets/Nova.png');  
91 | }  
92 /* Tarea 9: Agrega el objeto que quita las vidas */  
93 /* Puede ser cualquier objeto que deseas */  
94 /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */  
95 .objeto-virus {  
96 |   background-image: url('../Assets/VirusRojo.png');  
97 | }  
98 |  
99 |
```



Perder Vidas

juego.js

Hay que completar la función que actualiza las vidas usando nuestras imágenes.

Esta función se encarga de cambiar la imagen de los corazones cuando el personaje pierde una vida.

Por esta razón se recomienda usar corazones rojos y vacíos. Pero pueden usar otras imágenes.



```

71 // Función para actualizar vidas visualmente
72 function actualizarVidasVisualmente() {
73 /*
74  * console.log("Actualizando vidas visualmente:", vidas);
75
76 // Obtener el contenedor de corazones
77 const heartsContainer = document.getElementById('hearts-container');
78 if (!heartsContainer) {
79   console.error("Error: Contenedor de corazones no encontrado");
80   return;
81 }
82
83 // Obtener todas las imágenes de corazones
84 const corazones = heartsContainer.querySelectorAll('.heart');
85
86 console.log("Corazones encontrados:", corazones.length);
87
88 // Si no hay corazones, crearlos de nuevo
89 if (corazones.length === 0) {
90   console.log("No hay corazones, creándolos de nuevo");
91   crearCorazones();
92   return;
93 }
94
95 // Actualizar cada corazón
96 // Tarea 10: Completa la función
97 for (let i = 0; i < corazones.length; i++) {
98   if (i < vidas) {
99     corazones[i].src = ''; // Tarea 10: Imagen para vida llena
100   } else {
101     corazones[i].src = ''; // Tarea 10: Imagen para vida vacía
102   }
103
104 // Asegurarse de que el corazón sea visible
105 corazones[i].style.display = 'inline-block';
106 }
107 */
108 }
```

La función está comentada. Para “activarla” deberás borrar las líneas 73 y 107.

Perder Vidas

Solución

juego.js

```
// Actualizar cada corazón
// Tarea 10: Completa la función
for (let i = 0; i < corazones.length; i++) {
  if (i < vidas) {
    corazones[i].src = '../../../../../Assets/CorazónRojo.png'; // Tarea 10: Imagen para vida llena
  } else {
    corazones[i].src = '../../../../../Assets/CorazónVacio.png'; // Tarea 10: Imagen para vida vacía
  }

  // Asegurarse de que el corazón sea visible
  corazones[i].style.display = 'inline-block';
}
```

La imagen de la línea 134 debe ser la misma que utilizaste en la tarea 6.

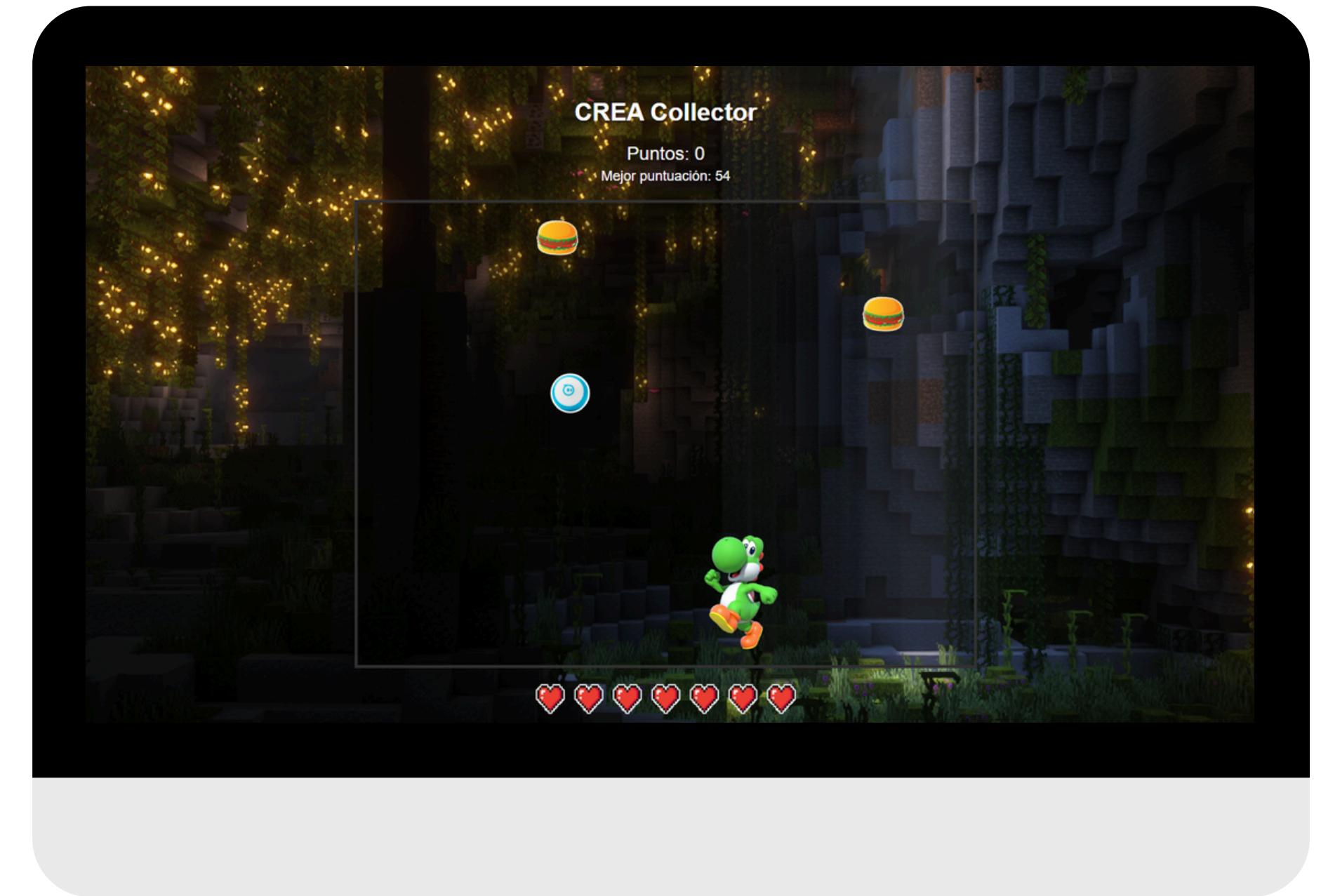
La primera línea representa el corazón completo.



La segunda línea representa el corazón después de perder una vida.



Ganar Vidas





Ganar Vidas

juego.js

Creemos un objeto que le devuelva vidas a nuestro personaje.

```
48      {
49        tipo: 'virus',           // Nombre del objeto
50        clase: 'objeto-virus', // Clase CSS para dar estilo
51        puntos: -1,            // Puntos que da al recogerlo
52        probabilidad: 0.15    // Probabilidad de que aparezca (entre 0 y 1)
53      }
54      // Tarea 11: Agrega 1 objeto que devuelva vidas
55      // Ayuda: ¿Debería dar puntos? ¿Podemos agregar otro atributo?
56    ];
57
```

Ayuda: ¿Debería dar puntos?

¿Podemos agregar algo extra?



Ganar Vidas

En este ejemplo creamos un objeto que devuelve vidas.

Es de tipo hamburguesa y su clase es ‘objeto-hamburguesa’.

Usamos un nuevo atributo llamado vidaExtra el cual es un booleano.

Possible solución juego.js

```
54 // Tarea 11: Agrega 1 objeto que devuelva vidas
55 // Ayuda: ¿Debería dar puntos? ¿Podemos agregar otro atributo?
56 {
57     tipo: 'hamburguesa',           // Nombre del objeto
58     clase: 'objeto-hamburguesa', // Clase CSS para dar estilo
59     puntos: 0,                  // Puntos que da al recogerlo
60     vidaExtra: true,            // Indica que da vida extra
61     probabilidad: 0.10         // Probabilidad de que aparezca (entre 0 y 1)
62 }
63 ];
```



Ganar Vidas

juego.css

```
93  /* Tarea 9: Agrega el objeto que quita las vidas */
94  /* Puede ser cualquier objeto que deseas */
95  /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */
96  .objeto-virus {
97      background-image: url('../Assets/VirusRojo.png');
98  }
99  /* Tarea 12: Agrega el objeto que devuelve las vidas */
100 /* Puede ser cualquier objeto que deseas */
101 /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */
```

Usando la **clase que acabas de crear, agrega la **Imagen** del objeto que va a dar las vidas.**

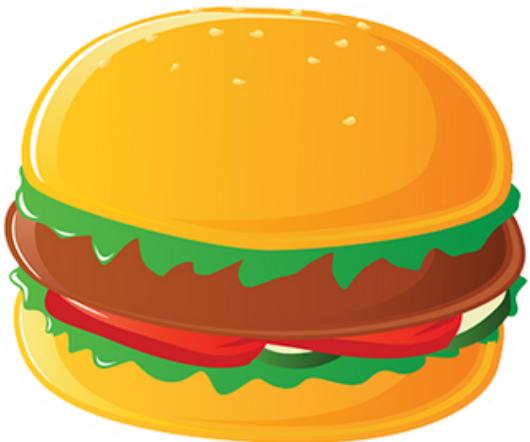
Ganar Vidas

En esta solución el objeto que devuelve vidas es una hamburguesa.

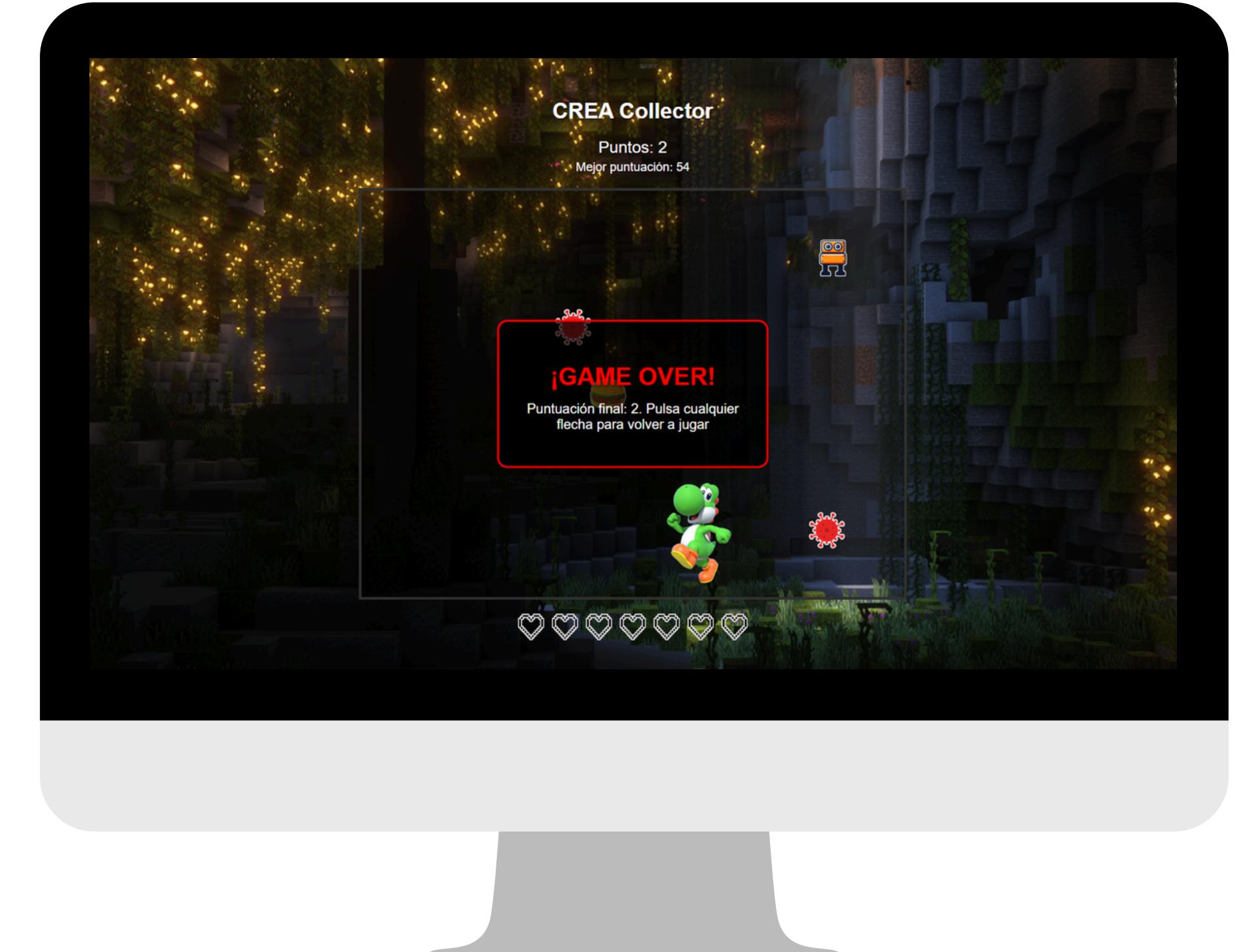
Sin embargo, ustedes pueden asignar la imagen que prefieran.

Possible solución juego.css

```
99  /* Tarea 12: Agrega el objeto que devuelve las vidas */  
100 /* Puede ser cualquier objeto que deseas */  
101 /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */  
102 .objeto-hamburguesa {  
103     background-image: url('../Assets/Hamburguesa.png');  
104 }
```



Game Over





Game Over

juego.js

**Creemos un objeto que le
ocasione un Game Over.**

```
54 // Tarea 11: Agrega 1 objeto que devuelva vidas
55 // Ayuda: ¿Debería dar puntos? ¿Podemos agregar otro atributo?
56 {
57     tipo: 'hamburguesa',           // Nombre del objeto
58     clase: 'objeto-hamburguesa',   // Clase CSS para dar estilo
59     puntos: 0,                   // Puntos que da al recogerlo
60     vidaExtra: true,            // Indica que da vida extra
61     probabilidad: 0.10          // Probabilidad de que aparezca (entre 0 y 1)
62 }
63 // Tarea 13: Agrega 1 objeto que termine el juego
64 // Ayuda: Recuerda lo que hicimos anteriormente (gameOver)
65 ];
```

Ayuda: Recuerda lo hecho en
el paso anterior (gameOver).

Game Over

En este ejemplo creamos un objeto que ocasiona un Game Over.

Es de **vinchi** y su clase es '**objeto-vinchi**'.

Usamos un nuevo atributo llamado **gameOver** el cual es un booleano.

Possible solución juego.js

```
62 },
63 // Tarea 13: Agrega 1 objeto que termine el juego
64 // Ayuda: Recuerda lo que hicimos anteriormente (gameOver)
65 {
66     tipo: 'vinchi',           // Nombre del objeto
67     clase: 'objeto-vinchi',  // Clase CSS para dar estilo
68     puntos: 0,               // Puntos que da al recogerlo
69     gameOver: true,         // Indica que termina el juego
70     probabilidad: 0.15     // Probabilidad de que aparezca (entre 0 y 1)
71 }
72 ];
```



Game Over

Usando la clase que acabas de crear, agrega la imagen del objeto que va causar un Game Over.

juego.css

```
102 .objeto-hamburguesa {  
103     background-image: url('../Assets/Hamburguesa.png');  
104 }  
105 /* Tarea 14: Agrega el objeto que acaba el juego */  
106 /* Puede ser cualquier objeto que deseas */  
107 /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */  
108  
109
```

Game Over

En esta solución el objeto que causa el Game Over es **Vinchi** (el mejor amigo de Seneca).

Sin embargo, ustedes pueden asignar la imagen que quieran.

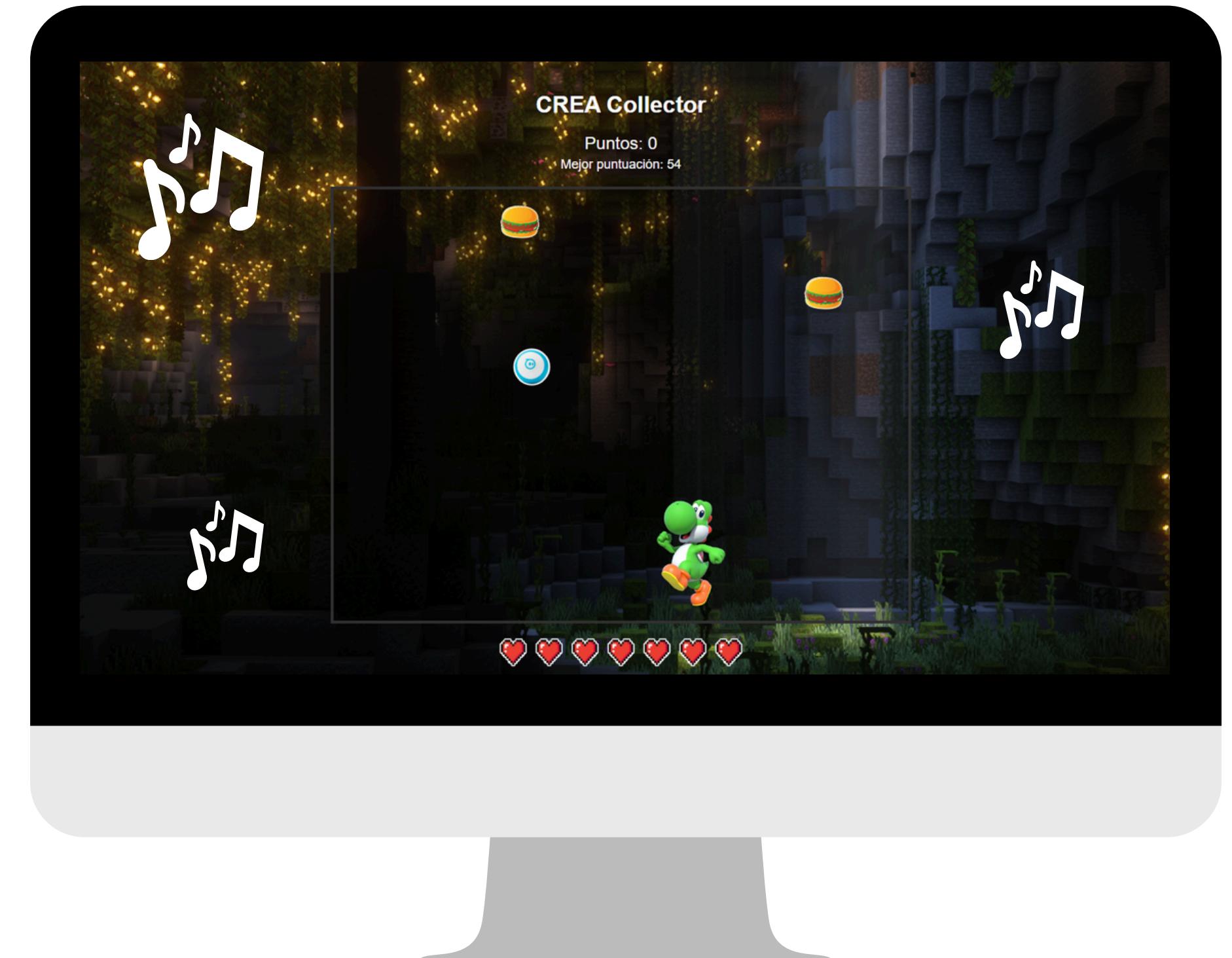
Possible solución

juego.css

```
105 /* Tarea 14: Agrega el objeto que acaba el juego */  
106 /* Puede ser cualquier objeto que deseas */  
107 /* Lo importante es que su nombre coincida con la clase que creaste en el paso anterior */  
108 .objeto-vinchi {  
109   background-image: url('../Assets/Vinchi.png');  
110 }  
111
```



Sonido



Sonido

juego.html

Revisemos los audios incluidos en el esqueleto de la página.

¿Qué nombres (*id*) tienen?

```
42      <!-- Contenedor de vidas -->
43      <!-- Tarea 7: Completa con la cantidad de vidas totales -->
44      <div id="hearts-container">
45          
46          
47          
48          
49          
50          
51          
52      </div>
53  </div>
54
55  <!-- Sonidos del juego -->
56  <audio id="sonidoPunto" src="../../Assets/Punto.wav" preload="auto"></audio>
57  <audio id="sonidoPerdida" src="../../Assets/Perdida.wav" preload="auto"></audio>
58  <audio id="sonidoGameOver" src="../../Assets/GameOver.wav" preload="auto"></audio>
59
60  <!-- Conexión al archivo de lógica, luego el de juego -->
61  <!-- Tarea 3: Conecta tus archivos JavaScript -->
62  <script src="logica.js"></script>
63  <script src="juego.js"></script>
64  </body>
65  </html>
```



Sonido

juego.js

Teniendo en cuenta los nombres anteriores, completa la función para reproducir un sonido.

¿Qué significan los **case**?

```
126 // Tarea 15: Completa la función para reproducir sonidos
127 function reproducirSonido(tipo) {
128     /*
129     switch(tipo) {
130         case 'punto':
131             document.getElementById('').currentTime = 0; // Tarea 14: Introducir el sonido apropiado
132             document.getElementById('').play(); // Tarea 14:
133             break;
134         case 'perdida':
135             document.getElementById('').currentTime = 0; // Tarea 14:
136             document.getElementById('').play(); // Tarea 14:
137             break;
138         case 'gameover':
139             document.getElementById('').currentTime = 0; // Tarea 14:
140             document.getElementById('').play(); // Tarea 14:
141             break;
142     }
143 */
144 }
```

La función está comentada. Para “activarla” deberás borrar las líneas 128 y 143.



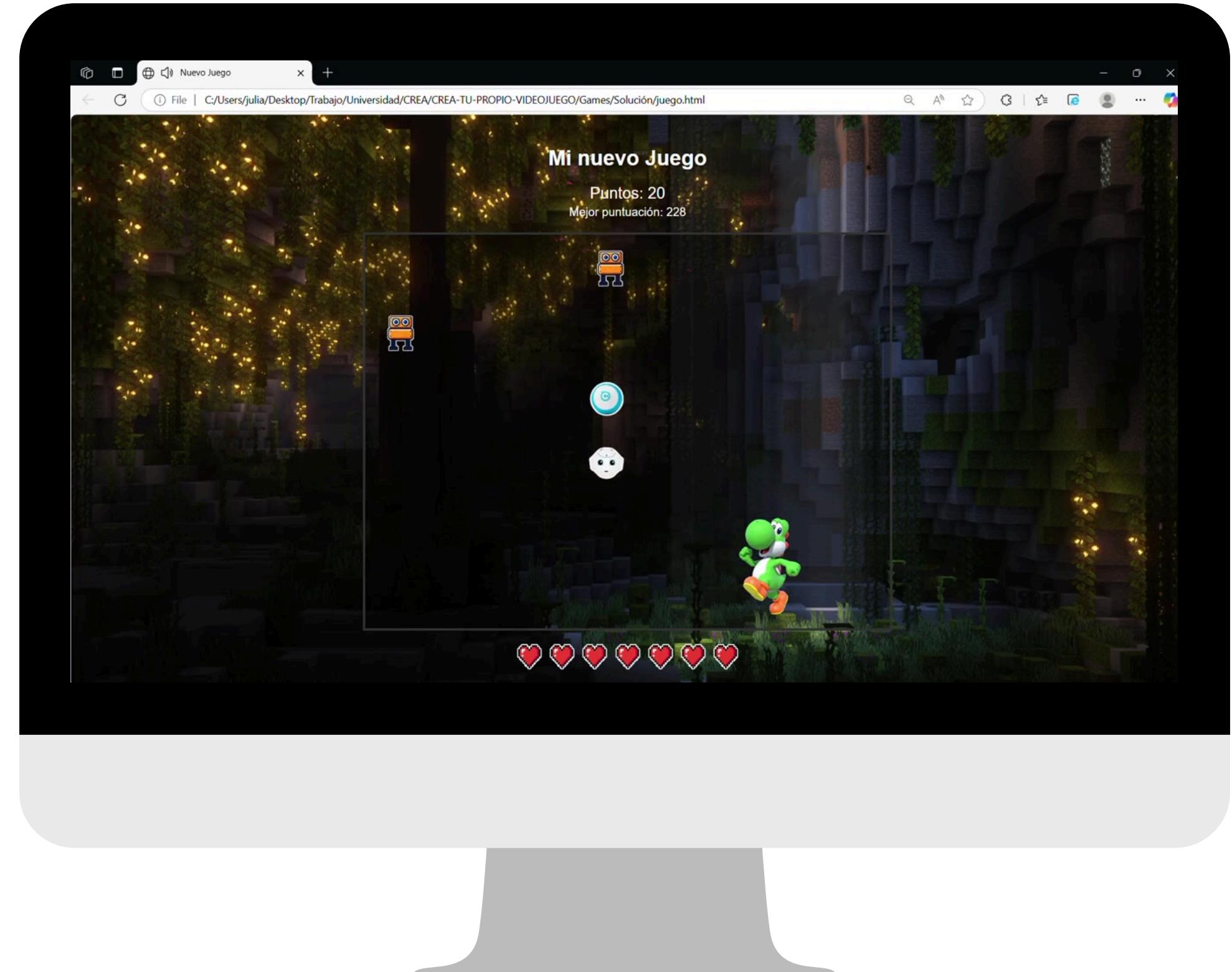
Sonido

Para cada uno de los casos, activamos un sonido diferente

Possible solución juego.js

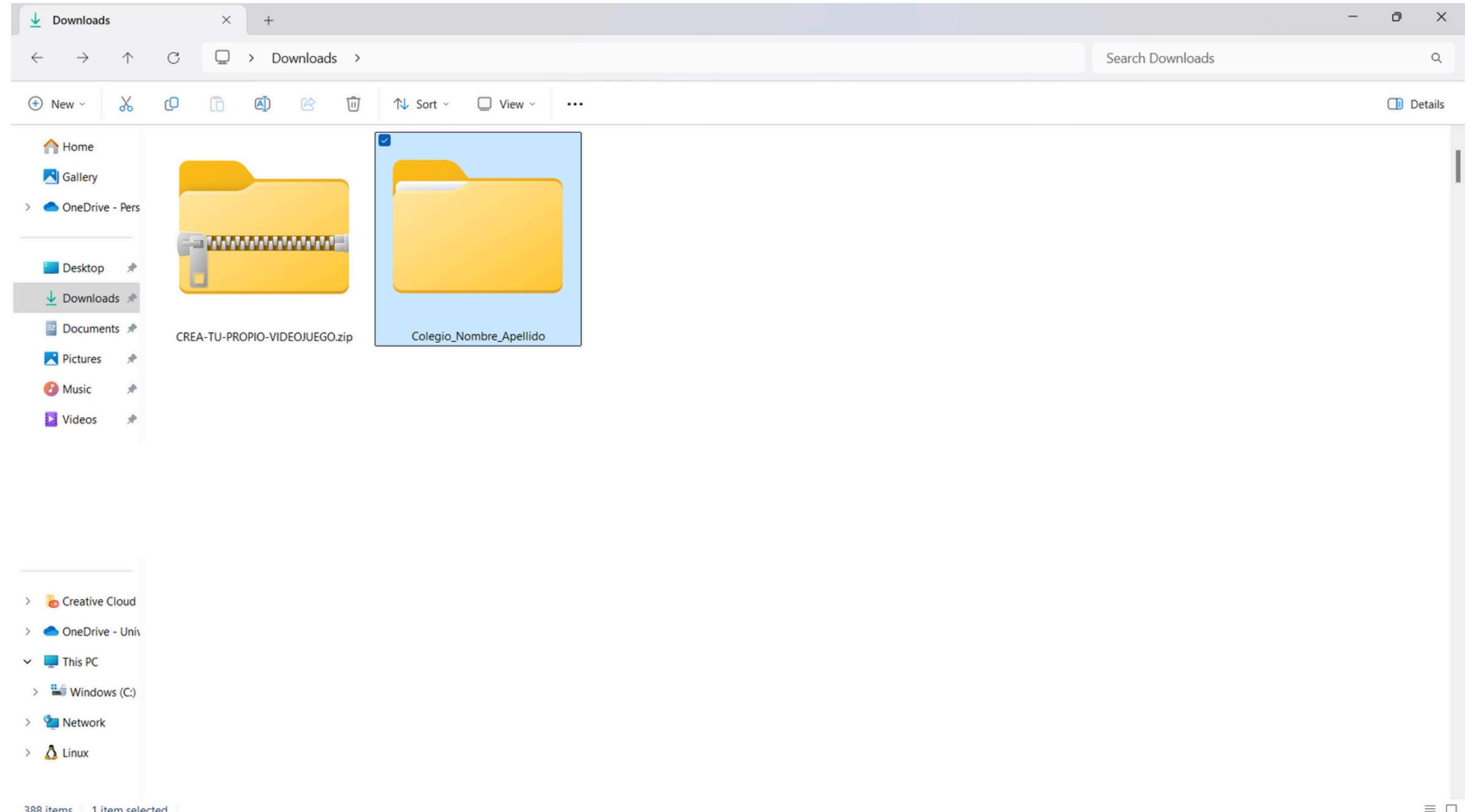
```
126 // Tarea 15: Completa la función para reproducir sonidos
127 function reproducirSonido(tipo) {
128     switch(tipo) {
129         case 'punto':
130             document.getElementById('sonidoPunto').currentTime = 0; // Tarea 14: Introducir el sonido apropiado
131             document.getElementById('sonidoPunto').play(); // Tarea 14:
132             break;
133         case 'perdida':
134             document.getElementById('sonidoPerdida').currentTime = 0; // Tarea 14:
135             document.getElementById('sonidoPerdida').play(); // Tarea 14:
136             break;
137         case 'gameover':
138             document.getElementById('sonidoGameOver').currentTime = 0; // Tarea 14:
139             document.getElementById('sonidoGameOver').play(); // Tarea 14:
140             break;
141     }
142 }
143 }
```

Resultado final





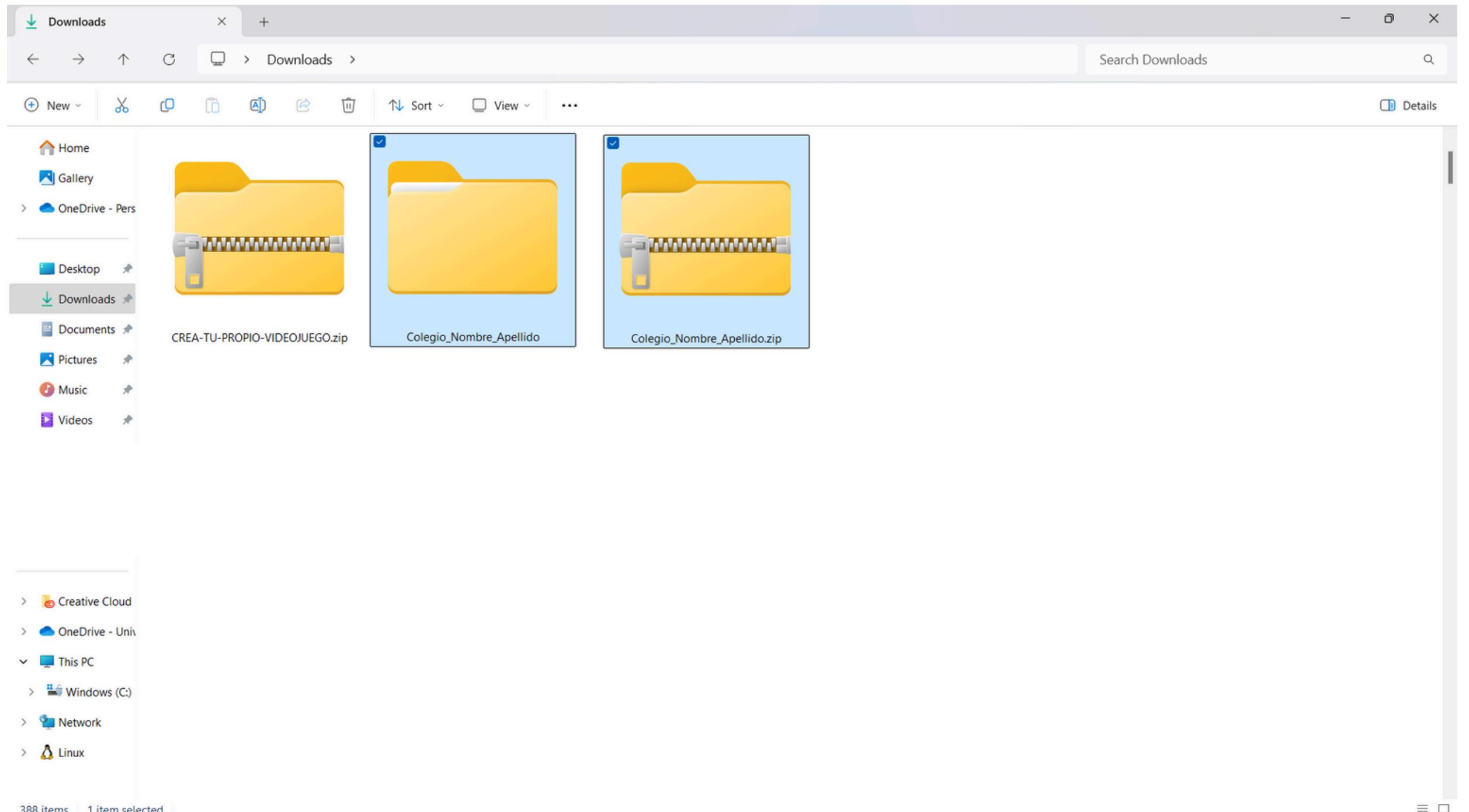
Para finalizar



**Cámbiale el
nombre a la
carpeta de tu
proyecto.**

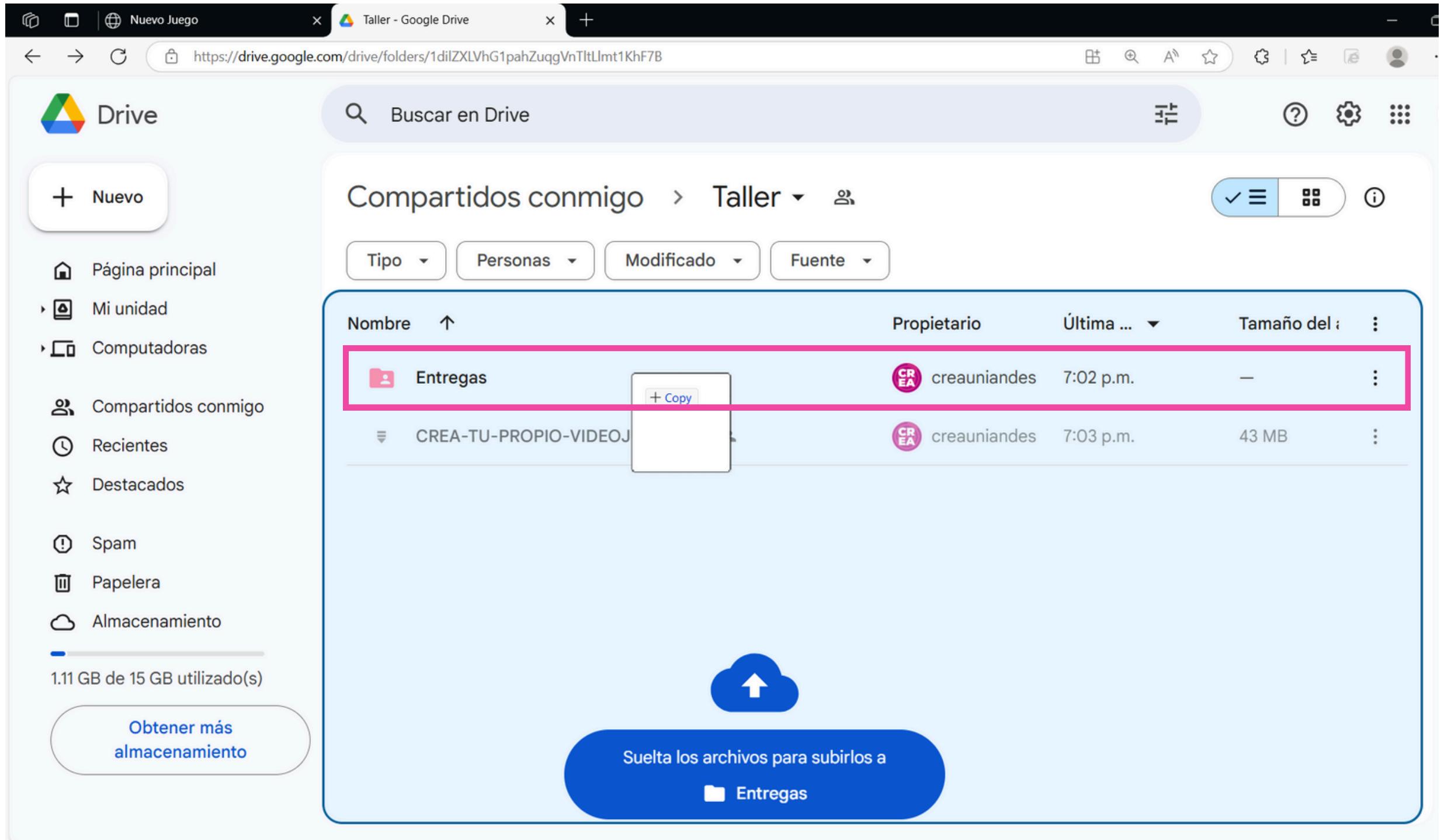
**Colegio_Nombre_
Apellido**

Para finalizar



**Comprime la
carpeta en un
archivo .zip.**

Para finalizar



The screenshot shows a Google Drive interface. The address bar indicates the URL is <https://drive.google.com/drive/folders/1dilZXLVhG1pahZuqqVnTltLlmt1KhF7B>. The left sidebar includes links like 'Nuevo', 'Página principal', 'Mi unidad', 'Computadoras', 'Compartidos conmigo', 'Recientes', 'Destacados', 'Spam', 'Papelera', and 'Almacenamiento'. The main area shows a list of files and folders under the 'Compartidos conmigo > Taller' folder. A folder named 'Entregas' is highlighted with a pink border. Below the list is a large blue button with a cloud icon and the text 'Suelta los archivos para subirlos a Entregas'.

Nombre	Propietario	Última ...	Tamaño del
Entregas	creauniandes	7:02 p.m.	—
CREA-TU-PROPIO-VIDEOJ	creauniandes	7:03 p.m.	43 MB

**Sube tu archivo
.zip a la carpeta
Entregas.**



Para finalizar

A screenshot of a web browser displaying the Google Drive interface. The address bar shows the URL https://drive.google.com/drive/folders/1WPv6uQCyB3zqfTd4hxJKmjjG8lav8wBq. The page title is "Entregas - Google Drive". The left sidebar shows navigation links like "Nuevo Juego", "Drive", "Nuevo", "Página principal", "Mi unidad", "Computadoras", "Compartidos conmigo", "Recientes", "Destacados", "Spam", "Papelera", and "Almacenamiento". The main content area shows a file named "Colegio_Nombre_Apellido.zip" in the "Entregas" folder. The file is owned by "yo" and was modified at "11:38 p.m.". The file size is "43 MB".

Nombre	Propietario	Última modificación	Tamaño del archivo
Colegio_Nombre_Apellido.zip	yo	11:38 p.m.	43 MB

Tu archivo deberá verse dentro de la carpeta.

De esta manera, podremos ver juntos todas sus increíbles creaciones.

¡Felicidades!

Hoy no solo jugaron un videojuego, ¡lo **crearon!** Imaginen todo lo que pueden lograr si siguen **aprendiendo**.



Califica tu experiencia



