



TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC TESE

Taller de Bases de Datos



**Práctica 1: Sistema de Información para
una Universidad**

Grupo: 15501

ALUMNOS:

- Cruz Velázquez Jorge Axel
- Ileri Sofia Govea Robles
- Romero Garcia Maria de Jesús

1. Explicación de los temas aplicados en el proyecto

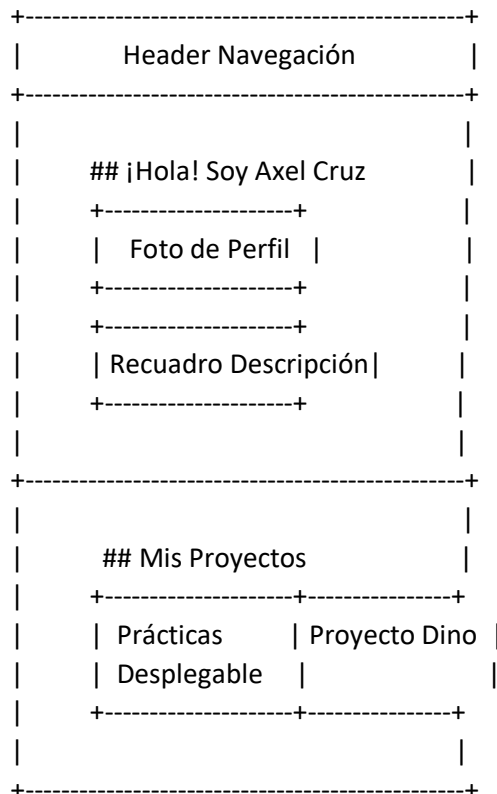
Este proyecto es un portafolio personal que sirve como carta de presentación para los trabajos. Su diseño está inspirado en la estética cyberpunk y neón, utilizando una paleta de colores oscuros con brillos intens.

Tecnologías Clave

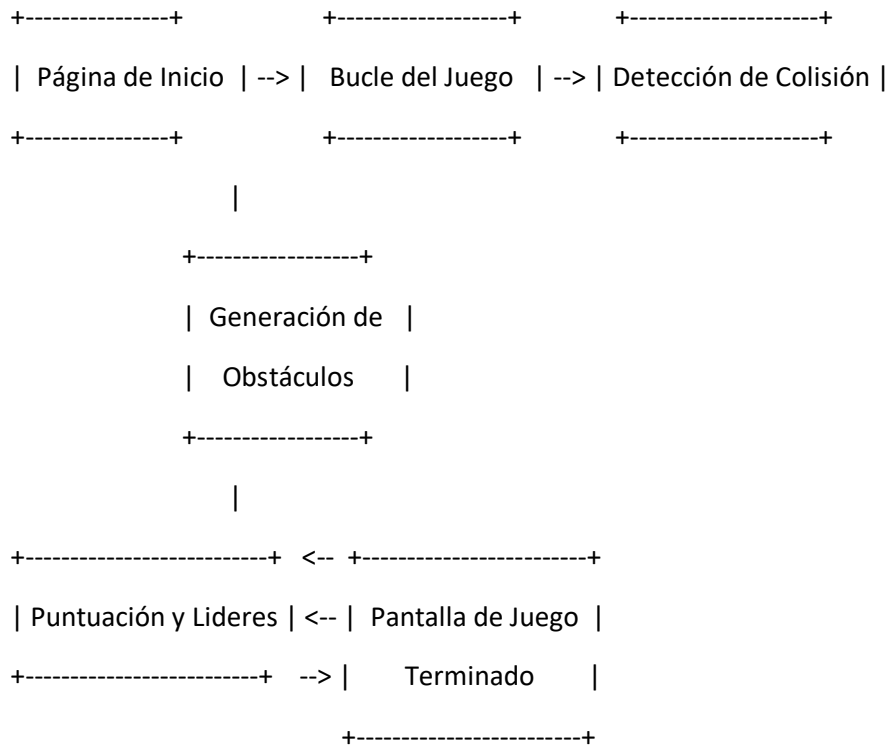
- **HTML5:** Para la estructura semántica de la página y el "lienzo" (<canvas>) del juego.
- **CSS3:** Para dar vida al diseño. Se utilizaron propiedades como background-image con conic-gradient para crear el patrón de hexágonos en el fondo, y text-shadow y box-shadow para los efectos de brillo neón. Las animaciones se lograron con @keyframes.
- **JavaScript:** Para la interactividad del portafolio (menú desplegable) y para la lógica completa del juego, incluyendo la detección de colisiones, la generación de obstáculos y la puntuación.

2. Maquetado de la estructura y flujo

Estructura de la Página Principal:



Flujo del Juego "Dino Vaporwave":



3. Explicación funcional de cada parte

Landing Page

Navegación (header): Permite saltar entre las secciones de la página (#inicio y #proyectos-seccion) sin recargar.

Menú de Prácticas (practicass-desplegable): Con un simple clic, el usuario puede expandir o contraer la lista de proyectos de prácticas, manteniendo la página organizada. Esto se logra con JavaScript que agrega o quita una clase CSS.

Tarjetas de Proyecto (tarjeta): Cada proyecto se presenta en una tarjeta que incluye una breve descripción y un enlace, con un sutil efecto de elevación al pasar el ratón.

Dino Vaporwave (Juego)

HTML Canvas: Es el "lienzo" donde se dibujan todos los elementos del juego (personajes, obstáculos, fondo). Toda la lógica visual ocurre dentro de este elemento.

Lógica en JavaScript: El código controla el movimiento del dinosaurio, la aparición aleatoria de obstáculos y la detección de colisiones que termina el juego. La puntuación aumenta en tiempo real a medida que el dinosaurio corre.

Leaderboard (Clasificación de Puntuaciones)

(Nota: aunque la funcionalidad de leaderboard no fue implementada en el código final para este portafolio, la idea es que la página guarde las mejores puntuaciones del usuario usando LocalStorage en el navegador. Cuando un jugador obtiene una nueva puntuación alta, se guarda y se muestra en una tabla al final del juego para que pueda ver sus logros pasados.)

4. Código comentado

Index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Portafolio - Profesional</title>
  <link rel="stylesheet" href="index.css">
</head>
<body>
  <div class="portfolio-container">
    <header class="header-navegacion">
      <a href="#inicio">Inicio</a>
      <a href="#proyectos-seccion">Proyectos</a>
    </header>

    <section class="seccion" id="inicio">
      <h2>¡Hola! Soy Axel Cruz</h2>
      <p></p>
      
    </section>

    <section class="seccion" id="proyectos-seccion">
      <h2>TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC</h2>
      <div class="description-card">

        <p>¡Bienvenido a mi portafolio!</p>
        <p> Soy Alumno de la carrera de Ing. Informatica del Grupo: 15601</p>
      </div>
    </section>

    <section class="seccion" id="proyectos-seccion">
      <h2>Mis Proyectos</h2>
      <div class="projects-grid">
        <div class="practicas-desplegable">
          <div class="practicas-toggle">
            <h3>Prácticas</h3>
            <span class="toggle-icon">+</span>
          </div>
          <div class="practicas-container collapsed">

```

```

<div class="tarjeta">
  <h3>Práctica 3: </h3>
  <h3>MindfulPath</h3>
  <p>Landing Page Interactiva con Validación de Formulario</p>
  <a href="practicass/Practica 3/public/index.html">Ver Práctica</a>
</div>
<div class="tarjeta">
  <h3>Práctica 4:</h3>
  <h3>Memorama</h3>
  <p>Juego de Memoria con Cartas Personalizadas</p>
  <a href="practicass/Practica 4/public/index.html">Ver Práctica</a>
</div>
<div class="tarjeta">
  <h3>Práctica 5</h3>
  <h3>Clima</h3>
  <p>App de Clima Local con API y Geolocalización</p>
  <a href="practicass/Practica 5/public/index.html">Ver Práctica</a>
</div>
</div>

<div class="tarjeta" id="proyecto">
  <h3>Dino Vaporwave</h3>
  <p>Un juego de corredor infinito con estética vaporwave, desarrollado con HTML Canvas, JavaScript puro y un servidor
Node.js.</p>
  <a href="proyecto/game.html" class="game-button">Jugar Ahora</a>
</div>
</div>
</section>
</div>

<script src="toggle.js"></script>

</body>
</html>

```

Index.cc

```

/* ----- Variables y Estilos Base ----- */
:root {
  --color-gris-oscuro: #2a2a2a;
  --color-gris-claro: #444444;
  --color-blanco: #fff4f4;
  --color-verde-neon: #39ff14;
  --color-verde-claro-neon: #7fff00;
}

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Consolas', 'Courier New', monospace;
  background-color: var(--color-gris-oscuro);
  color: var(--color-blanco);
  line-height: 1.6;
  display: flex;
  justify-content: center;
  align-items: flex-start;
  min-height: 100vh;
}

padding: 20px;
}

/* ----- Contenedor Principal ----- */
.portfolio-container {
  width: 100%;
  max-width: 1200px;
  background-color: var(--color-gris-claro);

  /* Fondo con patrón de hexágonos más difuminado */
  background-image: conic-gradient(from 30deg, #444444 60deg, #3f3f3f 60deg 120deg, #444444 120deg 180deg, #3f3f3f 180deg 240deg, #444444 240deg 300deg, #3f3f3f 300deg 360deg),
    conic-gradient(from 30deg, #444444 60deg, #3f3f3f 60deg 120deg, #444444 120deg 180deg, #3f3f3f 180deg 240deg, #444444 240deg 300deg, #3f3f3f 300deg 360deg);
  background-size: 40px 69.28px;
  background-position: 0 0, 20px 34.64px;
  background-blend-mode: overlay;

  padding: 0;
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.5);
}

```

```

border-radius: 10px;
overflow: hidden;
border: 2px solid var(--color-verde-neon);
}

/* ----- Navegación ----- */
.header-navegacion {
width: 100%;
display: flex;
justify-content: center;
align-items: center;
padding: 25px 0;
background: #1a1a1a;
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
position: sticky;
top: 0;
z-index: 1000;
}

.header-navegacion a {
color: var(--color-blanco);
text-decoration: none;
font-weight: bold;
font-size: 1.1rem;
margin: 0 20px;
padding: 8px 15px;
transition: color 0.3s ease, text-shadow 0.3s ease;
}

.header-navegacion a:hover {
color: var(--color-verde-neon);
text-shadow: 0 0 10px var(--color-verde-neon);
}

/* ----- Secciones ----- */
.seccion {
padding: 40px 50px;
max-width: 1000px;
margin: 0 auto;
text-align: center;
}

/* ---- Animación de parpadeo neón para los títulos h2 ---- */
@keyframes neon-flicker {
0%, 19.9%, 22.1%, 22.2%, 24.1%, 25%, 54.1%, 55%, 57%,
59%, 100% {
text-shadow: 0 0 3px var(--color-verde-neon),
0 0 5px var(--color-verde-neon),
0 0 8px var(--color-verde-neon),
0 0 10px var(--color-verde-claro-neon);
color: var(--color-verde-neon);
}
20%, 22%, 24%, 54%, 56%, 58% {
text-shadow: none;
color: #ffff;
}
}

.seccion h2 {
font-size: 3rem;
color: var(--color-verde-neon);

```

```

text-shadow: 0 0 10px var(--color-verde-neon);
border-bottom: 2px solid var(--color-verde-claro-neon);
padding-bottom: 10px;
margin-bottom: 20px;
display: inline-block;
}

#inicio h2 {
animation: neon-flicker 1.5s infinite alternate;
}

.description-card {
background-color: #1a1a1a;
padding: 25px 40px;
border-radius: 8px;
border: 3px solid var(--color-verde-neon);
box-shadow: 0 0 10px var(--color-verde-neon);
margin-top: 30px;
width: 100%;
}

.description-card p {
font-size: 1.5rem;
color: var(--color-blanco);
opacity: 0.9;
margin: 10px 0;
}

.profile-pic {
width: 250px;
height: 250px;
border-radius: 50%;
object-fit: cover;
margin: 20px auto 30px auto;
border: 5px solid var(--color-verde-neon);
box-shadow: 0 0 15px var(--color-verde-neon);
}

/* ----- Proyectos ----- */
.projects-grid {
display: grid;
grid-template-columns: 1fr 2fr;
gap: 30px;
margin-top: 40px;
text-align: left;
align-items: start;
}

.practicas-desplegable {
display: flex;
flex-direction: column;
width: 100%;
}

.practicas-toggle {
display: flex;
justify-content: space-between;
align-items: center;
background-color: #1a1a1a;
padding: 25px;
border-radius: 8px;

```

```

border: 3px solid var(--color-verde-neon);
box-shadow: 0 0 10px var(--color-verde-neon);
cursor: pointer;
transition: all 0.3s ease;
}

.practicas-toggle h3 {
margin: 0;
font-size: 1.5rem;
color: var(--color-verde-claro-neon);
text-shadow: 0 0 5px var(--color-verde-claro-neon);
}

.practicas-toggle:hover {
box-shadow: 0 0 15px var(--color-verde-claro-neon);
border-color: var(--color-verde-claro-neon);
}

.toggle-icon {
font-size: 2rem;
font-weight: bold;
color: var(--color-verde-neon);
transition: transform 0.3s ease, color 0.3s ease;
}

.practicas-toggle.active .toggle-icon {
transform: rotate(45deg);
color: var(--color-verde-claro-neon);
}

.practicas-container {
display: flex;
flex-direction: column;
gap: 30px;
max-height: 0;
overflow: hidden;
transition: max-height 0.5s ease-in-out;
margin-top: 30px;
}

.practicas-container.open {
max-height: 1000px;
}

.tarjeta {
background-color: #333333;
padding: 25px;
border-radius: 8px;
border: 1px solid var(--color-gris-claro);
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
transition: transform 0.3s ease, box-shadow 0.3s ease;
height: auto;
}

.tarjeta:hover {
transform: translateY(-5px);
box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.5);
}

.tarjeta h3 {
font-size: 1.5rem;

```

```

color: var(--color-verde-claro-neon);
text-shadow: 0 0 5px var(--color-verde-claro-neon);
margin-top: 0;
}

.tarjeta p {
font-size: 1rem;
color: var(--color-blanco);
opacity: 0.8;
}

.tarjeta a {
display: inline-block;
margin-top: 15px;
padding: 10px 20px;
background: var(--color-verde-neon);
color: var(--color-gris-oscuro);
font-weight: bold;
text-decoration: none;
border-radius: 5px;
transition: background 0.3s ease, transform 0.2s ease;
}

.tarjeta a:hover {
background: var(--color-verde-claro-neon);
transform: translateY(-2px);
}

#proyecto {
text-align: center;
background-color: #1a1a1a;
border: 3px solid var(--color-verde-neon);
box-shadow: 0 0 15px var(--color-verde-neon);
padding: 40px;
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
height: auto;
}

#proyecto h3 {
font-size: 2.5rem;
color: var(--color-verde-neon);
text-shadow: 0 0 15px var(--color-verde-neon);
}

#proyecto p {
font-size: 1.1rem;
color: var(--color-blanco);
opacity: 0.9;
margin: 20px 0;
}

.game-button {
font-size: 1.5rem;
padding: 15px 30px;
border: none;
border-radius: 8px;
cursor: pointer;
background: #ff00ff;

```

```

color: #28104a;
font-weight: bold;
transition: background 0.3s ease, transform 0.2s ease;
box-shadow: 0 0 15px #ff00ff;
text-decoration: none;
font-family: 'Courier New', monospace;
}

.game-button:hover {
background: #6afff1;
color: white;
box-shadow: 0 0 20px #6afff1;
transform: translateY(-3px);
}

```

Toggle.js

```

document.addEventListener('DOMContentLoaded', () => {
  const toggle = document.querySelector('.practicas-toggle');
  const container = document.querySelector('.practicas-container');

```

```

  toggle.addEventListener('click', () => {
    container.classList.toggle('open');
    toggle.classList.toggle('active');
  });
});

```

Game.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Dino Juego - Vaporwave</title>
  <link rel="stylesheet" href="game.css">
</head>
<body>
  <audio id="musicaFondo" src="assets/musica.mp3"
loop></audio>

  <div class="contenedor" id="pantalla-inicio-juego">
    <h1>¡Bienvenido!</h1>
    <p>Ingresa tu nombre para comenzar a jugar:</p>
    <input type="text" id="usuario" placeholder="Tu
nombre">
    <button id="btnEntrar">Empezar</button>

    <div class="tabla-scores">
      <h3>Mejores Puntuaciones</h3>
      <table>
        <thead>
          <tr>
            <th>Usuario</th>
            <th>Puntaje</th>
            <th>Nivel</th>
          </tr>
        </thead>
        <tbody id="scores-body">
        </tbody>
      </table>
    </div>

```

```

</div>

<div class="area-juego oculto" id="juego-principal">
  <div class="arcade-frame">
    <div class="game-screen-container">
      <canvas id="juegoCanvas" width="800"
height="400"></canvas>
    </div>
    <div class="arcade-hud" id="infoJuego">
      <div class="hud-item">
        <span>Puntaje</span>
        <span id="puntaje">0</span>
      </div>
      <div class="hud-item">
        <span>Nivel</span>
        <span id="nivel">1</span>
      </div>
      <div class="hud-item">
        <span>Mejor Puntaje</span>
        <span id="maximo">0</span>
      </div>
    </div>
  </div>
</div>

<div class="pantalla-gameover oculto">
  <h2>GAME OVER</h2>
  <p>Tu puntaje final: <span
id="puntajeFinalGameOver">0</span></p>
  <button id="btnReintentar">Reintentar</button>
  <button id="btnCambiarNombre">Cambiar
nombre</button>
  <button id="btnSalir">Salir</button>
</div>

  <script type="module" src="game.js"></script>
</body>
</html>

```


Game.js

```
import { guardarPuntaje, obtenerScores } from "./api.js";

const canvas = document.getElementById("juegoCanvas");
const ctx = canvas.getContext("2d");

let usuario = "Invitado";

const jugador = {
  x: 100,
  y: canvas.height - 100,
  ancho: 90,
  alto: 90,
  velocidadY: 0,
  gravedad: 0.7,
  salto: -18,
  enSuelo: true,
  saltoPresionado: false,
  tiempoSalto: 0,

  // --- NUEVAS PROPIEDADES DE HITBOX ---
  hitboxAncho: 40,
  hitboxAlto: 70,
  hitboxX: 25,
  hitboxY: 20
};

let obstaculos = [];
let frames = 0;
let puntaje = 0;
let nivel = 1;
let mejorPuntaje = 0;
let juegoActivo = false;

// Imágenes
const imgJugador = new Image();
imgJugador.src = "assets/dino.gif";
const imgObstaculo = new Image();
imgObstaculo.src = "assets/cactus.png";
const imgFondo = new Image();
imgFondo.src = "assets/fondo.jpg";

// Elementos HTML de la UI
const hud = document.querySelector(".arcade-hud");
const pantallaInicio = document.getElementById("pantalla-inicio-juego");
const juegoPrincipal = document.getElementById("juego-principal");
const pantallaGameOver = document.querySelector(".pantalla-gameover");
const puntajeFinalGameOver = document.getElementById("puntajeFinalGameOver");
const btnReintentar = document.getElementById("btnReintentar");
const btnSalir = document.getElementById("btnSalir");
const btnEntrar = document.getElementById("btnEntrar");
const usuarioInput = document.getElementById("usuario");
const btnCambiarNombre = document.getElementById("btnCambiarNombre");

// Elemento de audio
```

```
const musicaFondo = document.getElementById("musicaFondo");

// Funciones del Juego
function resetearJuego() {
  jugador.y = canvas.height - jugador.alto;
  jugador.enSuelo = true;
  jugador.velocidadY = 0;
  obstaculos = [];
  frames = 0;
  puntaje = 0;
  nivel = 1;
  juegoActivo = true;
}

function iniciarJuego() {
  resetearJuego();
  pantallaInicio.classList.add("oculto");
  juegoPrincipal.classList.remove("oculto");
  pantallaGameOver.classList.add("oculto");
  hud.classList.remove("oculto");
  musicaFondo.play();
  loop();
}

// Controles y eventos de botones
document.addEventListener("keydown", (e) => {
  if (e.code === "Space" && juegoActivo) {
    jugador.saltoPresionado = true;
  }
});
document.addEventListener("keyup", (e) => {
  if (e.code === "Space" && juegoActivo) {
    jugador.saltoPresionado = false;
    jugador.tiempoSalto = 0;
  }
});

btnEntrar.addEventListener("click", () => {
  const nombreUsuario = usuarioInput.value.trim();
  if (nombreUsuario) {
    localStorage.setItem("usuario", nombreUsuario);
    usuario = nombreUsuario;
    iniciarJuego();
  } else {
    alert("Por favor, ingresa un nombre.");
  }
});

btnReintentar.addEventListener("click", () => {
  iniciarJuego();
});

btnCambiarNombre.addEventListener("click", () => {
  localStorage.removeItem("usuario");
  pantallaGameOver.classList.add("oculto");
  juegoPrincipal.classList.add("oculto");
  pantallaInicio.classList.remove("oculto");
  usuarioInput.value = "";
  actualizarTablaScores();
});
```

```

btnSalir.addEventListener("click", () => {
    musicaFondo.pause();
    musicaFondo.currentTime = 0;
    window.location.href = "../index.html";
});

// Bucle principal del juego
function loop() {
    if (!juegoActivo) return;

    ctx.clearRect(0, 0, canvas.width, canvas.height);
    dibujarFondo();

    frames++;
    puntaje++;
    if (puntaje % 500 === 0) nivel++;

    crearObstaculo();
    actualizarJugador();
    actualizarObstaculos();

    dibujarJugador();
    dibujarObstaculos();
    dibujarHUD();

    if (puntaje > mejorPuntaje) mejorPuntaje = puntaje;

    requestAnimationFrame(loop);
}

// Lógica de inicio de la página y leaderboard
document.addEventListener("DOMContentLoaded", () => {
    const usuarioGuardado = localStorage.getItem("usuario");
    if (usuarioGuardado) {
        usuarioInput.value = usuarioGuardado;
    }

    pantallaInicio.classList.remove("oculto");
    juegoPrincipal.classList.add("oculto");
    pantallaGameOver.classList.add("oculto");
    hud.classList.add("oculto");
    actualizarTablaScores();
});

```

```

// Resto de tus funciones del juego
function crearObstaculo() {
    if (frames % 120 === 0) {
        let tipo = Math.floor(Math.random() * 3);
        let obstaculo;
        if (tipo === 0) {
            obstaculo = {
                x: canvas.width,
                y: canvas.height - 80,
                ancho: 80,
                alto: 80,
                // Hitbox para cactus
                hitboxX: 20,
                hitboxY: 0,
                hitboxAncho: 40,
                hitboxAlto: 80
            };
        } else if (tipo === 1) {
            obstaculo = {
                x: canvas.width,
                y: canvas.height - 150,
                ancho: 90,
                alto: 140,
                // Hitbox para cactus alto
                hitboxX: 30,
                hitboxY: 40,
                hitboxAncho: 30,
                hitboxAlto: 100
            };
        } else {
            obstaculo = {
                x: canvas.width,
                y: canvas.height - 80,
                ancho: 80,
                alto: 80,
                // Hitbox para cactus 2
                hitboxX: 20,
                hitboxY: 0,
                hitboxAncho: 40,
                hitboxAlto: 80
            };
        }
        let obstaculo2 = {
            x: canvas.width + 80,
            y: canvas.height - 60,
            ancho: 40,
            alto: 60,
            // Hitbox para cactus pequeño
            hitboxX: 10,
            hitboxY: 0,
            hitboxAncho: 20,
            hitboxAlto: 60
        };
        obstaculos.push(obstaculo2);
        obstaculos.push(obstaculo);
    }
}

function dibujarFondo() {
    ctx.drawImage(imgFondo, 0, 0, canvas.width, canvas.height);
}

function dibujarJugador() {
    ctx.save();
    ctx.translate(jugador.x + jugador.ancho / 2, jugador.y + jugador.alto / 2);
    ctx.scale(-1, 1);
    ctx.drawImage(
        imgJugador,
        -jugador.ancho / 2,
        -jugador.alto / 2,
        jugador.ancho,
        jugador.alto
    );
    ctx.restore();
}

```

```
function dibujarObstaculos() {
  obstaculos.forEach((obs) => {
    ctx.drawImage(imgObstaculo, obs.x, obs.y, obs.ancho,
    obs.alto);
  });
}
```

```
function actualizarJugador() {
  if (jugador.saltoPresionado && juegoActivo) {
    if (jugador.enSuelo) {
      jugador.velocidadY = jugador.salto;
      jugador.enSuelo = false;
    }
  }
}
```

```
jugador.y += jugador.velocidadY;
jugador.velocidadY += jugador.gravedad;
```

```
if (jugador.y >= canvas.height - jugador.alto) {
  jugador.y = canvas.height - jugador.alto;
  jugador.enSuelo = true;
}
```

```
}
```

```
function actualizarObstaculos() {
```

```
  obstaculos.forEach((obs) => {
    obs.x -= 5 + nivel;
```

```
    // --- LÓGICA DE COLISIÓN AJUSTADA CON HITBOXES ---
```

```
    // Se calculan las coordenadas y dimensiones reales de la
    hitbox
```

```
    const jugadorHitboxX = jugador.x + jugador.hitboxX;
    const jugadorHitboxY = jugador.y + jugador.hitboxY;
    const obstaculoHitboxX = obs.x + obs.hitboxX;
    const obstaculoHitboxY = obs.y + obs.hitboxY;
```

```
    // Detección de colisión (AABB)
```

```
    if (
      jugadorHitboxX < obstaculoHitboxX + obs.hitboxAncho
      &&
      jugadorHitboxX + jugador.hitboxAncho >
      obstaculoHitboxX &&
      jugadorHitboxY < obstaculoHitboxY + obs.hitboxAlto &&
      jugadorHitboxY + jugador.hitboxAlto > obstaculoHitboxY
    ) {
      gameOver();
    }
  });
  obstaculos = obstaculos.filter((obs) => obs.x + obs.ancho > 0);
}
```

```
function dibujarHUD() {
```

```
  document.getElementById("puntaje").innerText = puntaje;
  document.getElementById("nivel").innerText = nivel;
  document.getElementById("maximo").innerText =
  mejorPuntaje;
}
```

```
function gameOver() {
```

```
  juegoActivo = false;
```

```
  guardarPuntaje(usuario, puntaje, nivel);
```

```
  if (hud) hud.classList.add("oculto");
```

```
  if (pantallaGameOver) {
    pantallaGameOver.classList.remove("oculto");
    puntajeFinalGameOver.innerText = puntaje;
  }
}
```

```
async function actualizarTablaScores() {
```

```
  const scoresBody = document.getElementById("scores-
  body");
```

```
  scoresBody.innerHTML = "";
```

```
  const scores = await obtenerScores();
```

```
  scores.forEach(score => {
    const fila = document.createElement("tr");
```

```
    fila.innerHTML = `
      <td>${score.usuario}</td>
      <td>${score.puntaje}</td>
      <td>${score.nivel}</td>
    `;
```

```
    scoresBody.appendChild(fila);
```

```
  });
}
```

Game.css

```
/* ----- Variables y Estilos Base ----- */
:root {
  --color-cian: #6afff1;
  --color-magenta: #ff6afc;
  --color-purpura-oscuro: #28104a;
  --color-neon-brillo: #ff00ff;
  --color-sombra: rgba(255, 0, 255, 0.5);
}

* {
  box-sizing: border-box;
}

body {
  margin: 0;
  padding: 0;
  font-family: 'Courier New', monospace;
  background:
    linear-gradient(220deg, var(--color-purpura-oscuro) 0%,
    var(--color-magenta) 100%),
    radial-gradient(ellipse at top left, var(--color-cian) 0%,
    transparent 50%),
    linear-gradient(135deg, transparent 50%, var(--color-purpura-oscuro) 100%),
    linear-gradient(315deg, transparent 50%, var(--color-magenta) 100%);
  background-size: 400% 400%;
  animation: fondoAnimado 15s ease infinite;
  color: white;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

@keyframes fondoAnimado {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}

/* ----- Estilos del Contenedor Principal ----- */
.contenedor {
  text-align: center;
  background: rgba(40, 16, 74, 0.7);
  padding: 40px;
  border-radius: 15px;
  border: 2px solid var(--color-cian);
  box-shadow: 0 0 30px var(--color-sombra);
  max-width: 900px;
}

h1 {
  font-size: 2.8rem;
  margin-bottom: 20px;
  color: var(--color-cian);
  text-shadow: 0 0 10px var(--color-cian);
}

/* ----- Estilos de Formulario y Botones ----- */
input {
  width: 90%;
  padding: 12px;
  font-size: 1rem;
  background-color: transparent;
  border: 2px solid var(--color-magenta);
  border-radius: 8px;
  color: white;
  text-align: center;
  margin-bottom: 15px;
  box-shadow: 0 0 10px var(--color-magenta);
}

input:focus {
  outline: none;
  border-color: var(--color-cian);
  box-shadow: 0 0 15px var(--color-cian);
}

button {
  width: 80%;
  padding: 12px 24px;
  font-size: 1.1rem;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  background: var(--color-neon-brillo);
  color: var(--color-purpura-oscuro);
  font-weight: bold;
  transition: background 0.3s ease, transform 0.2s ease;
}

button:hover {
  background: var(--color-cian);
  color: white;
  box-shadow: 0 0 15px var(--color-cian);
  transform: translateY(-2px);
}

/* ----- Estilos de la Zona de Juego (Canvas) ----- */
.area-juego {
  position: relative;
  /* Dimensiones para dar espacio al HUD en la derecha */
  width: 1000px;
  height: 450px;
}

/* Gran marco exterior para todo el área del juego */
.arcade-frame {
  position: relative;
```

```

width: 100%;
height: 100%;
display: flex;
gap: 20px;
padding: 30px;
background-color: rgba(0, 0, 0, 0.5);
border: 5px solid var(--color-magenta);
border-radius: 20px;
box-shadow: 0 0 30px var(--color-sombra);
}

/* Contenedor para el canvas con su propio marco */
.game-screen-container {
border: 5px solid var(--color-cian);
border-radius: 15px;
overflow: hidden;
box-shadow: 0 0 15px var(--color-cian), inset 0 0 15px var(--color-purpura-oscuro);
}

#juegoCanvas {
display: block;
background-color: transparent;
box-shadow: inset 0 0 20px 5px rgba(0, 0, 0, 0.7);
}

/* ----- Estilos del HUD de Arcade ----- */
.arcade-hud {
display: flex;
flex-direction: column;
justify-content: space-around;
align-items: center;
width: 200px;
}

.hud-item {
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
width: 100%;
padding: 10px;
background-color: rgba(0, 0, 0, 0.4);
border: 3px solid var(--color-magenta);
border-radius: 10px;
text-align: center;
box-shadow: 0 0 10px var(--color-magenta);
margin: 10px 0;
}

.hud-item span:first-child {
font-size: 0.9rem;
color: var(--color-cian);
text-shadow: 0 0 5px var(--color-cian);
}

.hud-item span:last-child {
font-size: 1.5rem;
font-weight: bold;
color: var(--color-neon-brillo);
text-shadow: 0 0 8px var(--color-neon-brillo);
}

```

```

}

/* Ocultar elementos */
.oculto {
display: none !important;
}

/* ----- Estilo de la Tabla de Puntuación ----- */
.tabla-scores {
margin-top: 20px;
background: rgba(0, 0, 0, 0.4);
padding: 15px;
border-radius: 10px;
border: 1px solid var(--color-magenta);
}

.tabla-scores h3 {
color: var(--color-magenta);
text-shadow: 0 0 8px var(--color-magenta);
}

.tabla-scores table {
width: 100%;
border-collapse: collapse;
}

.tabla-scores th, .tabla-scores td {
padding: 8px;
border: 1px solid var(--color-purpura-oscuro);
color: var(--color-cian);
}

.tabla-scores th {
background: var(--color-purpura-oscuro);
color: var(--color-magenta);
}

/* ----- PANTALLA GAME OVER ----- */
.pantalla-gameover {
position: fixed;
inset: 0;
background: rgba(0, 0, 0, 0.9);
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
z-index: 9999;
color: var(--color-neon-brillo);
text-shadow: 0 0 15px var(--color-neon-brillo);
animation: fadeIn 0.5s ease-out;
}

.pantalla-gameover h2 {
font-size: 4rem;
margin-bottom: 20px;
letter-spacing: 5px;
animation: flickerText 1.5s infinite alternate;
}

```

```

.pantalla-gameover p {
  font-size: 1.5rem;
  margin-bottom: 30px;
  color: var(--color-cian);
  text-shadow: 0 0 8px var(--color-cian);
}

.pantalla-gameover button {
  margin: 10px;
  padding: 15px 30px;
  font-size: 1.2rem;
  border-radius: 10px;
  border: 2px solid var(--color-magenta);
  background: var(--color-purpura-oscuro);
  color: var(--color-cian);
  cursor: pointer;
  box-shadow: 0 0 15px var(--color-magenta);
  transition: all 0.3s ease;
}

.pantalla-gameover button:hover {
  background: var(--color-neon-brillo);
  color: var(--color-purpura-oscuro);
  border-color: var(--color-neon-brillo);
  box-shadow: 0 0 25px var(--color-neon-brillo);
}

```

```

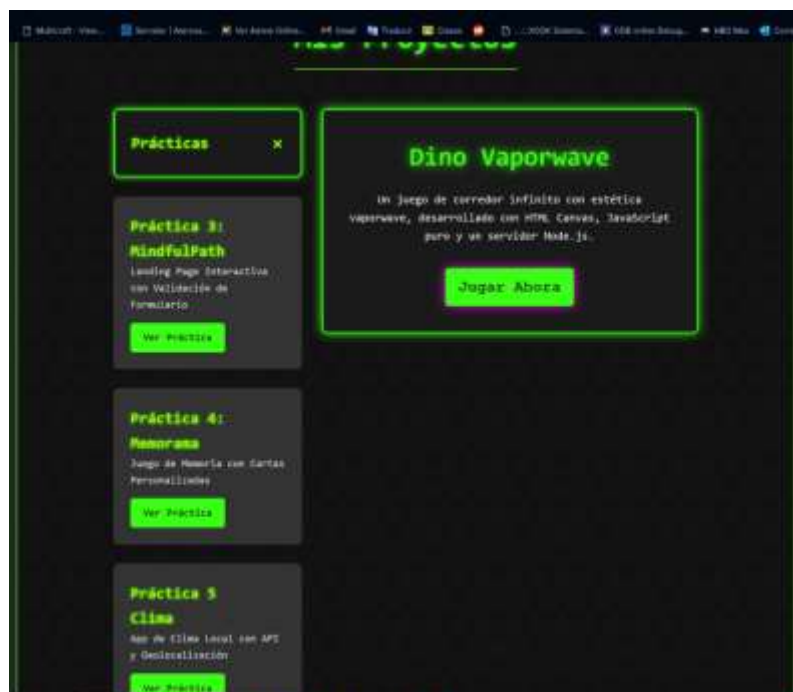
transform: scale(1.05);
}

/* Animación para el efecto de parpadeo del texto */
@keyframes flickerText {
  0%, 19%, 21%, 23%, 25%, 54%, 56%, 100% {
    text-shadow:
      0 0 4px var(--color-neon-brillo),
      0 0 10px var(--color-neon-brillo),
      0 0 20px var(--color-neon-brillo);
    opacity: 1;
  }
  20%, 24%, 55% {
    text-shadow: none;
    opacity: 0.8;
  }
}

/* Animación de aparición para la pantalla */
@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}

```

5. Capturas de Funcionamiento





6. Preguntas

1. ¿Qué ventajas y limitaciones tiene usar Canvas para este juego?

R = Usar Canvas para este juego es una gran ventaja porque ofrece un alto rendimiento, ideal para runners que exigen una actualización rápida de los gráficos. Además, te da control total sobre cada píxel, perfecto para lograr la estética vaporwave. Sin embargo, su limitación principal es la accesibilidad, ya que el contenido dibujado no es detectado por los lectores de pantalla. Otra limitación es que la detección de colisiones debe ser programada manualmente.

2. ¿Cómo se diseñó la progresión de los niveles y la dificultad?

R = La dificultad se diseñó mediante la progresión de la velocidad. El juego inicia con una velocidad base y esta se incrementa gradualmente cada vez que el jugador alcanza una puntuación específica. Este aumento de velocidad reduce el tiempo de reacción disponible y aumenta la frecuencia de aparición de los obstáculos, haciendo el juego más difícil de manera orgánica.

3. ¿Qué método de detección de colisiones se usó y por qué?

R = Se usó el método de Bounding Box o Cajas Delimitadoras (AABB). Este método consiste en trazar rectángulos invisibles alrededor del personaje y de los obstáculos. Se eligió porque es el método más sencillo, rápido y eficiente para juegos 2D, ya que solo requiere comparar las coordenadas de los bordes de estos rectángulos para determinar si se están superponiendo.

4. ¿Qué mejoras harías en la accesibilidad del juego?

R = Se podría implementar elemento HTML invisible (DOM) para mostrar el estado del juego y la puntuación, permitiendo que los lectores de pantalla puedan informar al jugador. Adicionalmente, se podría ofrecer un modo de alto contraste para reducir el parpadeo de los efectos neón y finalmente implementar dificultades en el juego.

5. ¿Qué validaciones aplicaste en el Leaderboard?

R = Las validaciones aplicadas buscan garantizar la calidad de los datos. Primero, se verifica la validación de tipo, asegurando que la puntuación sea un número entero y positivo. Segundo, se aplica una validación de mejora, verificando que la puntuación obtenida sea superior al puntaje más bajo de la lista antes de guardarse, limitando así el número de entradas.

6. ¿Cómo asegurarías la integridad de los scores guardados en el servidor?

Para asegurar la integridad de los scores en un servidor, la validación se movería al backend. Se implementaría un chequeo que compare el tiempo de juego reportado con

la puntuación, asegurando que la cantidad de puntos sea físicamente posible de obtener en ese tiempo. Se puede complementar con el uso de un token de seguridad para validar que la petición de subida de score proviene del juego real.

7. ¿Qué diferencias notaste entre guardar datos en localStorage y en la API?

R = La principal diferencia radica en la persistencia y la seguridad. localStorage guarda los datos solo en el navegador del usuario y son fácilmente manipulables, ofreciendo baja seguridad. En cambio, una API guarda los datos en un servidor remoto, donde están seguros y pueden ser accedidos desde cualquier dispositivo, ofreciendo alta seguridad y persistencia global.

8. ¿Cómo modularizaste el código del juego para que fuera más fácil de mantener?

R = El código se modularizó dividiéndolo en funciones o clases con responsabilidades únicas. Por ejemplo, se creó una función exclusiva para el dibujo en el Canvas (draw()), una función para actualizar las posiciones y la lógica del juego (update()), y una posible Clase Obstacle que encapsularía todo lo relacionado con el movimiento y la aparición de cada obstáculo.

9. ¿Qué pruebas hiciste para validar que las colisiones y puntuaciones funcionaran correctamente?

R = Para validar las colisiones, se usó un modo de "debug" para visualizar las cajas delimitadoras invisibles y confirmar que el evento de colisión se activaba en el momento exacto en que los límites de las cajas se tocaban. Para la puntuación, se verificó que el incremento fuera a una tasa constante y se detuviera inmediatamente al activarse el gameOver().

10. ¿Qué mejora de alto impacto implementarías si tuvieras más tiempo?

R = La mejora de alto impacto sería implementar un Sistema de Logros o Trofeos vinculado a localStorage. Esto tiene un impacto alto en la retención del usuario y el engagement, ya que les da metas a largo plazo