

===== HTML =====

- - - - - D : \ Documents \ 3 . C E S D E - C a s o c a m b i o
de programa \ Tecnica - Proyecto \ Semestre 2 \
Frontend I \ Cesde_Frontend - I \ Juego memoria
V1 \ index . html - - - - -

< ! D O C T Y P E h t m l >

< h t m l l a n g = " e s " >

< h e a d >

< m e t a c h a r s e t = " U T F - 8 " >

< m e t a n a m e = " v i e w p o r t " c o n t e n t = " w i d t h =
d e v i c e - w i d t h , i n i t i a l - s c a l e = 1 . 0 " >

< t i t l e > J u e g o M e m o r i a < / t i t l e >

< ! - - B o o t s t r a p - - >

< l i n k h r e f = " h t t p s : / / c d n . j s d e l i v r . n e t /
n p m / b o o t s t r a p @ 5 . 3 . 7 / d i s t / c s s / b o o t s t r a p . m i n
. c s s " r e l = " s t y l e s h e e t "

integrity = " s h a 3 8 4 - L N + 7 f d V z j 6 u 5 2 u 3
0 K p 6 M / t r l i B M C M K T y K 8 3 3 z p b D + p X d C L u T u s P j 6 9 7 F H
4 R / 5 m c r " c r o s s o r i g i n = " a n o n y m o u s " >

< l i n k r e l = " s t y l e s h e e t " h r e f = " e s t i l o s .
c s s " >

< / h e a d >

< b o d y >

< d i v c l a s s = " c o n t a i n e r " >

< d i v c l a s s = " r o w " >

< d i v c l a s s = " c o l - 1 2 " >

< h 1 c l a s s = " m t - 3 m b - 2 t e x t
- c e n t e r " > J u e g o d e M e m o r i a < / h 1 >

< / d i v >

```

        <!-- Barra de informaci3n -->
        <div class="col-12">
            <ul class="nav justify-content-center">
                <li class="nav-item">
                    <a class="nav-link fw-bold text-dark" href="#">Jugador: <span id="jugador">Player 1</span></a>
                </li>
                <li class="nav-item">
                    <a class="nav-link fw-bold text-dark" href="#">Intentos: <span id="intentos">0</span></a>
                </li>
                <li class="nav-item">
                    <a class="nav-link fw-bold text-dark" href="#">Aciertos: <span id="aciertos">0</span></a>
                </li>
                <li class="nav-item">
                    <a class="nav-link fw-bold text-dark" href="#">Tiempo: <span id="tiempo">0</span>s</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link fw-bold text-dark" href="#"><span id="em
oji">=</span></a>
                </li>
            </ul>
        </div>
    </div>

```

```

        <li class="nav-item">
            <a class="nav-link fw-bold text-dark" href="#">Nivel: <span
id="nivel">1</span></a>
        </li>
        <li class="nav-item">
            <span id="btnIniciar" class="btn btn-success">Iniciar</span>
        </li>
    </ul>
</div>

<div class="row">
    <!-- TABLERO -->
    <div class="col-7">
        <h2>Tablero de juego</h2>
        <hr>
        <div class="row g-3 mb-5
tablero">
            <!-- Las cartas se ag
regan dinamicamente -->
        </div>
    </div>

    <!-- TABLA DE RECORDS -->
    <div class="col-5">

```

```

        <table class="table my-3
records table-striped">

        <thead class="table-d
anger">

            <tr>

                <th>#</th>

                <th>Jugador</
th>

                <th>Tiempo to
tal</th>

                <th>Intentos
totales</th>

            </tr>

        </thead>

        <tbody></tbody>

    </table>

    </div>

</div>

</div>

<!-- Bootstrap -->

<script src="https://cdn.jsdelivr.net
/npm/bootstrap@5.3.7/dist/js/bootstrap.bun
dle.min.js"

    integrity="sha384-ndDqU0GzaU9qJ1l
fW4pNL1hNTkCfHzAVBREH9diLvGRem5+R9g2FzA8ZG
N95405Q"

    crossorigin="anonymous"></script>

```

```
<!-- Tu script -->
<script src="guardarDatos.js"></script>
t>
<script src="funciones.js"></script>
</body>
</html>
```

```
==== CSS ====
- - - - - D:\Documents\3. CESDE - Caso cambio
de programa\Tecnica - Proyecto\Semestre 2\
Frontend I\Cesde_Frontend-I\Juego memoria
V1\estilos.css - - - - -
```

```
.altura-img{
    height: 150px;
    width: 150px;
}
```

```
/* =====
FONDO GENERAL DE LA PAGINA
===== */
```

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    color: #222;
    position: relative;
    min-height: 100vh;
```

```
/* Imagen de fondo general (marca de ag
```

```

ua de pagina) */

background: url('imagenes/fondo2.jpg')
center/cover no-repeat fixed;

}

/* Efecto de marca de agua suave sobre to
da la pagina */

body::before {
    content: " ";
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(255, 255, 255, 0.68);
/* suaviza la imagen */

/* backdrop-filter: blur(2px); /* efecto
de difuminado */

    z-index: -1;
}

/* =====
TITULOS Y ELEMENTOS PRINCIPALES
===== */

h1 {
    text-align: center;
    color: #470202;

```

```
font-weight: bold;
}
```

```
h2, h4 {
  color: #3d0101;
}
```

```
/* =====
PANEL SUPERIOR DE INFORMACION
===== */
```

```
.info {
  display: flex;
  justify-content: center;
  gap: 15px;
  margin: 10px 0 20px;
  color: #01152c;
  font-weight: 500;
}
```

```
#btnIniciar {
  background-color: #2e7d32;
  color: white;
  border: none;
  padding: 5px 10px;
  border-radius: 4px;
```

```
    cursor: pointer;
    transition: background-color 0.3s ease;
}
```

```
#btnIniciar:hover {
    background-color: #1b5e20;
}
```

```
/* =====
    TABLERO DE JUEGO
    ===== */
```

```
.tablero {
    position: relative;
    width: 100%;
    min-height: 300px;
    border: 2px dashed #ccc;
    border-radius: 10px;
    padding: 10px;
    margin: 10px auto 30px;
```

```
/* Imagen de fondo del tablero (marca d
e agua interna) */
```

```
background:
    rgba(255, 255, 255, 0.85)
    url('imagenes/marca-agua.png')
    center / 50% no-repeat;
```



```

        background-blend-mode: lighten;
    }

    /* Cada carta o imagen dentro del tablero
    */

    .tablero img {
        width: 90px;
        height: 90px;
        margin: 5px;
        object-fit: cover;
        border-radius: 8px;
        border: 1px solid #ccc;

        /* Marca de agua sobre cada carta */

        background: rgba(255, 255, 255, 0.7) url('
imagenes/marca-agua.png') center/80% no-re
peat;

        background-blend-mode: lighten;

        transition: transform 0.3s ease, opacit
y 0.3s ease;
    }

    .tablero img:hover {
        transform: scale(1.05);
        opacity: 0.9;
    }

```

```

/* =====
   TABLA DE PUNTAJES
   ===== */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
    background-color: rgba(255, 255, 255, 0.9);
;
    border-radius: 6px;
    overflow: hidden;
}

th, td {
    border: 1px solid #ccc;
    text-align: center;
    padding: 8px;
}

th {
    background-color: #f2f2f2;
}

/* =====

```

AJUSTES RESPONSIVOS

===== */

```
@media (max-width: 768px) {
```

```
  .info {
```

```
    flex-direction: column;
```

```
    align-items: center;
```

```
  }
```

```
  .tablero img {
```

```
    width: 70px;
```

```
    height: 70px;
```

```
  }
```

```
}
```

===== JS =====

```
- - - - - D:\Documents\3. CESDE - Caso cambio  
de programa\Tecnica - Proyecto\Semestre 2\  
Frontend I\Cesde_Frontend-I\Juego memoria  
V1\funciones.js - - - - -
```

```
// ===== VARIABLES GLO  
BALES =====
```

```
const d = document;
```

```
// - - - al inicio, donde defines variables  
globales - - -
```

```
let startTime = null; // marca de tiempo  
cuando inicia cada juego
```

```
let usuario = "";
```

```

let nivel = 1;
let tiempo = 60;
let timer = null;
let juegoIniciado = false;

let imagenNombre = [];
let imagenID = [];
let cartasEmparejadas = new Set();
let intentos = 0;
let aciertos = 0;

// ===== ELEMENTOS DEL DOM =====

const tablero = d.querySelector(".tablero");

const spanIntentos = d.getElementById("intentos");

const spanAciertos = d.getElementById("aciertos");

const spanTiempo = d.getElementById("tiempo");

const spanNivel = d.getElementById("nivel");

const emoji = d.getElementById("emoji");

const btnIniciar = d.getElementById("btnIniciar");

const tablaRecords = document.querySelector(".records tbody");

```

```
// ===== MODAL DE USUARIO =====
```

```
function pedirNombre() {  
    usuario = prompt("Escribe tu nombre de usuario:");  
    if (!usuario) usuario = "Jugador";  
    localStorage.setItem("usuario", usuario);  
    console.log(`Bienvenido, ${usuario}`);  
};  
  
pedirNombre();
```

```
// ===== IMAGENES POR NIVEL =====
```

```
const niveles = {  
    1: [  
        { nombre: "carta1", url: "imagenes/carta1.jpg" },  
        { nombre: "carta2", url: "imagenes/carta2.jpg" },  
        { nombre: "carta3", url: "imagenes/carta3.jpg" },  
        { nombre: "carta4", url: "imagenes/carta4.jpg" },  
        { nombre: "carta5", url: "imagenes/carta5.jpg" },  
        { nombre: "carta6", url: "imagenes/carta6.jpg" }  
    ],  
};
```

```
2: [
    { nombre: "carta7", url: "imagenes / carta7.jpg" },
    { nombre: "carta8", url: "imagenes / carta8.jpg" },
    { nombre: "carta9", url: "imagenes / carta9.jpg" },
    { nombre: "carta10", url: "imagenes / carta10.jpg" },
    { nombre: "carta11", url: "imagenes / carta11.jpg" },
    { nombre: "carta12", url: "imagenes / carta12.jpg" },
    { nombre: "carta13", url: "imagenes / carta13.jpg" },
    { nombre: "carta14", url: "imagenes / carta14.jpg" }
],
```

```
3: [
    { nombre: "carta15", url: "imagenes / carta15.jpg" },
    { nombre: "carta16", url: "imagenes / carta16.jpg" },
    { nombre: "carta17", url: "imagenes / carta17.jpg" },
    { nombre: "carta18", url: "imagenes / carta18.jpg" },
    { nombre: "carta19", url: "imagenes / carta19.jpg" },
    { nombre: "carta20", url: "imagenes / carta20.jpg" },
    { nombre: "carta21", url: "imagenes / carta21.jpg" },
]
```

```
        { nombre: "carta22", url: "imagenes / carta22.jpg" }
```

```
    ]
```

```
};
```

```
// ===== BOT N INICIAR =====
```

```
btnIniciar.addEventListener("click", () => {
```

```
    if (!juegoIniciado) {
```

```
        iniciarJuego();
```

```
        btnIniciar.textContent = "Reiniciar";
```

```
        btnIniciar.style.backgroundColor = "#dc3545";
```

```
    } else {
```

```
        reiniciarJuego();
```

```
    }
```

```
});
```

```
// ===== INICIAR JUEGO =====
```

```
function iniciarJuego() {
```

```
    console.clear();
```

```
    console.log(`<Nivel ${nivel} iniciado por ${usuario}`);
```

```
    intentos = 0;
```

```
    a c i e r t o s   =   0 ;
    c a r t a s E m p a r e j a d a s . c l e a r ( ) ;
    i m a g e n N o m b r e   =   [ ] ;
    i m a g e n I D   =   [ ] ;
    t i e m p o   =   6 0 ;
    s p a n T i e m p o . s t y l e . c o l o r   =   " b l a c k " ;
    i f   ( e m o j i )   e m o j i . t e x t C o n t e n t   =   " =
";

    j u e g o I n i c i a d o   =   t r u e ;

    s p a n N i v e l . t e x t C o n t e n t   =   n i v e l ;
    s p a n I n t e n t o s . t e x t C o n t e n t   =   i n t e n t o s ;
    s p a n A c i e r t o s . t e x t C o n t e n t   =   a c i e r t o s ;
    s p a n T i e m p o . t e x t C o n t e n t   =   t i e m p o ;

    // Guardamos el momento de inicio par
a   c a l c u l a r   t i e m p o   t o t a l   d e s p u e s

    s t a r t T i m e   =   D a t e . n o w ( ) ;

    a g r e g a r I m a g e n e s ( ) ;

    c l e a r I n t e r v a l ( t i m e r ) ;

    t i m e r   =   s e t I n t e r v a l ( a c t u a l i z a r T i e m p o ,
1 0 0 0 ) ;

    // Guardar datos iniciales en localSt
o r a g e   s i   q u i e r e s
```



```
o ( ) ; guardarDatosJuego && guardarDatosJueg  
}
```

```
// ===== AGREGAR IMAGENES =====
```

```
function agregarImagenes ( ) {  
    tablero.innerHTML = " ";  
    let imgs = [ ...niveles[nivel], ...niveles[nivel] ];  
    imgs.sort ( ( ) => Math.random ( ) - 0.5 );  
  
    imgs.forEach ( ( imagen , i ) => {  
        let div = d.createElement ( "div" );  
        div.className = "col-3";  
  
        let img = d.createElement ( "img" );  
        img.className = "img-fluid altura  
- img ";  
        img.id = i;  
        img.src = "imagenes/ocultar.jpg";  
        img.dataset.nombre = imagen.nombre;  
        img.dataset.url = imagen.url;  
        img.addEventListener ( "click" , mostrarImg );  
    } );  
}
```

```

        div.appendChild(img);
        tablero.appendChild(div);
    });
}

// ===== ACTUALIZAR TIEMPO =====
function actualizarTiempo() {
    tiempo--;
    spanTiempo.textContent = tiempo;

    if (tiempo <= 10 && tiempo > 0) {
        spanTiempo.style.color = "red";
        if (emoji) emoji.textContent = "=
, ";
    }

    // Guardar estado si tienes esa funci
n
    guardarDatosJuego && guardarDatosJueg
o();

    if (tiempo <= 0) {
        clearInterval(timer);

        const tiempoTotal = Math.round((D
ate.now() - (startTime || Date.now())) / 1
000);

        console.log("# Tiempo agotado. Gu
ardando record parcial:", { usuario, nivel

```

```
, tiempoTotal, intentos }));
```

```
        // Guardamos registro del intento  
        aunque haya perdido (nivel alcanzado)
```

```
        guardarRecord(usuario, nivel, tiempoTotal, intentos);
```

```
        // Reiniciamos el juego y forzamos  
        volver a nivel 1
```

```
        nivel = 1;
```

```
        localStorage.setItem("nivel", nivel);
```

```
        reiniciarJuego(true);
```

```
        alert("# Tiempo agotado! Has perdido.  
        Volviendo al nivel 1.");
```

```
    }
```

```
}
```

```
// ===== MOSTRAR IMAGEN =====
```

```
function mostrarImg() {
```

```
    if (!juegoIniciado) {
```

```
        alert("⚠ Presiona 'Iniciar' para  
        comenzar.");
```

```
        return;
```

```
    }
```

```
    let id = this.id;
```

```

        // Carta ya emparejada
        if (cartasEmparejadas.has(id)) {
            alert(" &[] Esa carta ya fue empare
jada, escoge otra.");
            return;
        }

        // Mismo clic
        if (imagenID.length === 1 && imagenID
[0] === id) {
            alert(" &[] Debes seleccionar otra
imagen.");
            return;
        }

        this.src = this.dataset.url;
        imagenNombre.push(this.dataset.nombre
);
        imagenID.push(id);

        if (imagenNombre.length === 2) {
            setTimeout(compararImg, 700);
        }
    }

    // ===== COMPARAR =====
    =====

```

```

function compararImg() {
    let cartas = d.querySelectorAll( ".tablero img" );

    intentos++;

    spanIntentos.textContent = intentos;

    if ( imagenNombre[0] === imagenNombre[1] ) {
        cartas[imagenID[0]].src = "imagenes/acierto-png-5.png";
        cartas[imagenID[1]].src = "imagenes/acierto-png-5.png";

        cartasEmparejadas.add( imagenID[0] );
        cartasEmparejadas.add( imagenID[1] );

        aciertos++;
        spanAciertos.textContent = aciertos;

        guardarDatosJuego();

        if ( aciertos === niveles[nivel].length ) {
            subirNivel();
        }
    } else {
        cartas[imagenID[0]].src = "imagenes/ocultar.jpg";
        cartas[imagenID[1]].src = "imagenes/ocultar.jpg";
    }
}

```

```
es/ocultar.jpg";
```

```
}
```

```
imagenNombre = [];
```

```
imagenID = [];
```

```
}
```

```
// ===== SUBIR DE NIVEL =====  
=====
```

```
function subirNivel() {
```

```
    clearInterval(timer);
```

```
    // Guardar el progreso del jugador en  
    este nivel (antes de incrementar)
```

```
    const tiempoTotal = Math.round((Date.  
now() - (startTime || Date.now())) / 1000)  
;
```

```
    console.log("🎉 Subiste de nivel. Guar  
dando record provisional:", { usuario, niv  
el, tiempoTotal, intentos });
```

```
    guardarRecord(usuario, nivel, tiempoT  
otal, intentos);
```

```
    alert(`<% Felicidades ${usuario}! Comp  
letaste el nivel ${nivel}`);
```

```
    nivel++;
```

```
    localStorage.setItem("nivel", nivel);
```

```

// Si ya pas el 3, finalizamos el juego
(y ya registramos el nivel 3 arriba)

    if (nivel > 3) {

        alert(" < Has completado todos los niveles!");

        // opcional: borrar datos guardados de juego

        borrarDatosJuego && borrarDatosJuego();

        reiniciarJuego(true);

        return;

    }

// iniciar el siguiente nivel

iniciarJuego();

}

```

```

// ===== REINICIAR =====
=====

```

```

function reiniciarJuego(final = false) {

    clearInterval(timer);

    juegoIniciado = false;

    btnIniciar.textContent = "Iniciar";

    btnIniciar.style.backgroundColor = "#198754";
}

```

```

    if (final) nivel = 1; // Reinicio total si el juego se completó o se perdió

```

```
// Actualizar en interfaz
spanNivel.textContent = nivel;
spanIntentos.textContent = 0;
spanAciertos.textContent = 0;
spanTiempo.textContent = 0;

// Mostrar mensaje emoji
if (emoji) emoji.textContent = " = 4 ";

// Reiniciar tablero
agregarImagenes();
```

```
}
```

```
// ===== LOCAL STORAGE (guardarDatos.js integrado) =====
```

```
function guardarDatosJuego() {
    const datos = {
        usuario,
        nivel,
        tiempo,
        intentos,
        aciertos,
        fecha: new Date().toLocaleString(
    )
}
```



```

    };

    localStorage.setItem("datosJuego", JSON.stringify(datos));
}

function cargarDatosJuego() {
    const datos = localStorage.getItem("datosJuego");
    return datos ? JSON.parse(datos) : null;
}

function borrarDatosJuego() {
    localStorage.removeItem("datosJuego");
}

// Cargar records previos al iniciar
window.addEventListener("DOMContentLoaded", mostrarRecords);

function guardarRecord(jugador, nivelRecord, tiempoRecord, intentosRecord) {
    if (!jugador) jugador = "Jugador";

    const records = JSON.parse(localStorage.getItem("records")) || [];

```

```

    const nuevo = {
        jugador,
        nivel: Number(nivelRecord) || 0,
        tiempo: Number(tiempoRecord) || 0
    },
    intentos: Number(intentosRecord)
    || 0,
    fecha: new Date().toLocaleString()
    };

    records.push(nuevo);

    // Ordenar: nivel desc, tiempo asc, i
    ntentos asc
    records.sort((a, b) => {
        if (b.nivel !== a.nivel) return b
        .nivel - a.nivel;

        if (a.tiempo !== b.tiempo) return
        a.tiempo - b.tiempo;

        return a.intentos - b.intentos;
    });

    const top10 = records.slice(0, 10);

    localStorage.setItem("records", JSON.
    stringify(top10));

    mostrarRecords();
}

```

```

function mostrarRecords() {
    const records = JSON.parse(localStorage.getItem("records")) || [];
    if (!tablaRecords) {
        console.warn("No se encontró tablaRecords (.records tbody)");
        return;
    }
    tablaRecords.innerHTML = "";

    records.forEach((r, i) => {
        const tr = document.createElement("tr");
        tr.innerHTML = `
            <td>${i + 1}</td>
            <td>${r.jugador}</td>
            <td>Nivel ${r.nivel}</td>
            <td>${r.tiempo}s</td>
            <td>${r.intentos}</td>
        `;
        tablaRecords.appendChild(tr);
    });

    console.log("< Tabla de records actualizada:", records);
}

```

```

// Mostrar records en carga de pagina
window.addEventListener("DOMContentLoaded", () => {

    try { mostrarRecords(); } catch(e) { console.error(e); }

});

- - - - - D:\Documents\3. CESDE - Caso cambio
de programa\Tecnica - Proyecto\Semestre 2
\Frontend I\Cesde_Frontend-I\Juego memoria
V1\guardarDatos.js - - - - -

// ===== GUARDAR Y CAR
GAR DATOS =====

// Guarda el estado actual del juego en l
ocalStorage

function guardarDatosJuego(usuario, nivel
, tiempo, intentos, aciertos) {

    const datos = {

        jugador,

        nivel,

        tiempo,

        intentos,

        aciertos,

        fecha: new Date().toLocaleString(
)

    };

// Guardar los datos en localStorage

```

```
        localStorage.setItem( "datosJuego" , JSON.stringify( datos ) );

        console.log( " = Datos guardados: " , datos );

    }
}
```

```
// Carga los datos almacenados del juego anterior
```

```
function cargarDatosJuego( ) {

    const datosGuardados = localStorage.getItem( "datosJuego" );

    if ( datosGuardados ) {

        const datos = JSON.parse( datosGuardados );

        console.log( " = Datos cargados desde localStorage: " , datos );

        return datos;

    } else {

        console.log( " & No hay datos guardados en localStorage. " );

        return null;

    }

}
```

```
// Borra los datos guardados (por ejemplo, al reiniciar completamente)
```

```
function borrarDatosJuego( ) {

    localStorage.removeItem( "datosJuego" );

};
```

```
        console.log( " = [] Datos del juego eliminados del localStorage. " );  
    }
```