

JUEGO INTERACTIVO DE MEMORIA SOBRE 10 ANIMALES

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<style>
  /* Estilo General del Juego de Animales */
  .animal-game-container {
    font-family: 'Verdana', sans-serif;
    max-width: 600px;
    margin: 20px auto;
    padding: 25px;
    background-color: #fff9db;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.15);
    text-align: center;
    border: 3px solid #f59e0b;
  }

  .game-title { color: #b45309; margin: 0 0 10px 0; font-size: 1.8em; }
  .game-info { color: #d97706; margin-bottom: 20px; font-weight: bold; }

  /* Grilla del Tablero */
  .animal-grid {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 12px;
    perspective: 1000px;
  }

  /* Diseño de las Tarjetas */
  .card-animal {
    height: 100px;
```

```
position: relative;
transform-style: preserve-3d;
transition: transform 0.6s;
cursor: pointer;
}

.card-animal.flipped { transform: rotateY(180deg); }
```

```
.face-front, .face-back {
width: 100%;
height: 100%;
position: absolute;
border-radius: 12px;
backface-visibility: hidden;
display: flex;
align-items: center;
justify-content: center;
font-size: 2.8em;
box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}
```

```
.face-front {
background-color: #ffffff;
transform: rotateY(180deg);
border: 2px solid #fbbf24;
}
```

```
.face-back {
background-color: #f59e0b;
color: white;
border: 2px solid #d97706;
}
```

```
.face-back::after {
```

```
content: "🐾";
font-size: 0.6em;
}

/* Botón de Reinicio */
.btn-reset {
  margin-top: 30px;
  padding: 12px 35px;
  background-color: #d97706;
  color: white;
  border: none;
  border-radius: 50px;
  font-weight: bold;
  cursor: pointer;
  font-size: 1em;
  transition: background 0.3s;
}
.btn-reset:hover { background-color: #b45309; }

/* Estilo del mensaje de Victoria */
.victory-msg {
  display: none;
  background-color: #dcfce7;
  color: #166534;
  padding: 15px;
  border-radius: 10px;
  margin-top: 20px;
  border: 2px solid #22c55e;
  font-weight: bold;
  animation: bounce 0.5s infinite alternate;
}

@keyframes bounce {
  from { transform: translateY(0); }
```

```

    to { transform: translateY(-5px); }
  }

  @media (max-width: 480px) {
    .card-animal { height: 80px; }
    .face-front { font-size: 2.2em; }
  }
</style>
</head>
<body>

<div class="animal-game-container">
  <h2 class="game-title">🦁 Memoria Animal</h2>
  <div class="game-info">Movimientos: <span id="count">0</span></div>

  <div class="animal-grid" id="board"></div>

  <div id="victory" class="victory-msg">
    ¡Increíble! 🎉 Has completado el safari con éxito. 🏆
  </div>

  <button class="btn-reset" onclick="initGame()">Reiniciar Safari</button>
</div>

<script>
const animals = ['🦁', '🐘', '🦒', '🦓', '🐵', '🐯', '🐼', '🐻', '🦊', '🐸'];
let deck = [...animals, ...animals];
let card1, card2;
let busy = false;
let moveCount = 0;
let matchesFound = 0;

function initGame() {
  const board = document.getElementById('board');

```

```

const victoryMsg = document.getElementById('victory');

board.innerHTML = "";

victoryMsg.style.display = 'none';

moveCount = 0;

matchesFound = 0;

document.getElementById('count').innerText = moveCount;

deck.sort(() => Math.random() - 0.5);

deck.forEach(emoji => {
  const cardElement = document.createElement('div');
  cardElement.classList.add('card-animal');
  cardElement.dataset.icon = emoji;
  cardElement.innerHTML = `
    <div class="face-front">${emoji}</div>
    <div class="face-back"></div>
  `;
  cardElement.addEventListener('click', flip);
  board.appendChild(cardElement);
});
}

function flip() {
  if (busy) return;
  if (this === card1) return;

  this.classList.add('flipped');

  if (!card1) {
    card1 = this;
    return;
  }

  card2 = this;

```

```
    moveCount++;
    document.getElementById('count').innerText = moveCount;
    checkMatch();
}
```

```
function checkMatch() {
    const isMatch = card1.dataset.icon === card2.dataset.icon;
    if (isMatch) {
        matchesFound++;
        lockCards();
        if (matchesFound === animals.length) {
            setTimeout(() => {
                document.getElementById('victory').style.display = 'block';
            }, 500);
        }
    } else {
        unflipCards();
    }
}
```

```
function lockCards() {
    card1.removeEventListener('click', flip);
    card2.removeEventListener('click', flip);
    resetState();
}
```

```
function unflipCards() {
    busy = true;
    setTimeout(() => {
        card1.classList.remove('flipped');
        card2.classList.remove('flipped');
        resetState();
    }, 1000);
}
```

```
function resetState() {  
  [card1, card2] = [null, null];  
  busy = false;  
}
```

```
  initGame();
```

```
</script>
```

```
</body>
```

```
</html>
```