

# ADONVINA EL NUMERO

creado por

CASTILLO ESPIONOSA

MAURICIO EMMANUEL

DISEÑO Y MAQUETACION

CASTILLO ESPIONOSA

MAURICIO EMMANUEL

# Manual de uso del juego "Adivina el Número"

## Introducción

El juego "Adivina el Número" es una aplicación interactiva desarrollada con HTML, CSS y JavaScript. Su objetivo es que el jugador adivine un número aleatorio entre 1 y 100 que el sistema genera automáticamente.

Cada intento recibe una respuesta que guía al jugador: si el número es demasiado alto, bajo o correcto.

Este manual explica detalladamente qué hace cada parte del código, para qué sirve y cómo se conectan los archivos que hacen funcionar el juego.

## Documentos utilizados

En este proyecto se emplean dos archivos principales:

- `index.html`: contiene la estructura del juego (lo que el usuario ve y con lo que interactúa).
- `styles.css`: define el aspecto visual del juego (colores, fuentes, alineación y botones).

## Archivo 1: index.html

Este archivo contiene el esqueleto del juego y el código JavaScript que le da funcionalidad.

A continuación se muestra su contenido y la explicación de cada parte:

### . Estructura general HTML

```
<!DOCTYPE html>
<html lang="es">
<head> ... </head>
<body> ... </body>
</html>
```

Define que el documento es una página HTML. Dentro de `<head>` se agregan configuraciones (como el título y el enlace a la hoja de estilos) y dentro de `<body>` se coloca el contenido visible del juego.

### 2. Encabezado del juego

```
<h1>Adivina el Número</h1>
<p>Estoy pensando en un número entre 1 y 100.</p>
```

Muestra el nombre del juego y las instrucciones para el jugador.

El texto explica brevemente la meta: adivinar un número del 1 al 100.

## Entrada y botón

```
<input type="number" id="guess" placeholder="Introduce tu intento">  
<button onclick="checkGuess()">Adivinar</button>
```

- El `<input>` permite que el jugador escriba su número.
- El botón ejecuta la función `checkGuess()` cuando se hace clic.
- Esa función está en el código JavaScript y contiene la lógica del juego.

### 4. Mensaje de resultado

```
<p id="message"></p>
```

Aquí aparecerán los mensajes del juego:

- Si el número es demasiado alto o bajo.
- Si el jugador ha adivinado correctamente.

## ⚙️ 5. Código JavaScript

```
<script>
let randomNumber = Math.floor(Math.random() * 100) + 1;
let attempts = 0;

function checkGuess() {
  let guess = document.getElementById("guess").value;
  attempts++;

  if (guess < randomNumber) {
    document.getElementById("message").textContent = "Demasiado bajo."
  } else if (guess > randomNumber) {
    document.getElementById("message").textContent = "Demasiado alto."
  } else {
    document.getElementById("message").textContent = `¡Felicidades! Adivinaste el número secreto en ${attempts} intentos.`
  }
}
</script>
```



- `let randomNumber = Math.floor(Math.random() * 100) + 1;`
  - Genera un número aleatorio entre 1 y 100.
  - Este será el número secreto que el jugador debe adivinar.
- `let attempts = 0;`
  - Inicia un contador que aumenta cada vez que el jugador intenta adivinar.
- `function checkGuess() { ... }`
  - Define la función que se ejecuta al hacer clic en el botón "Adivinar".
  - Dentro de la función:
    - `let guess = document.getElementById("guess").value;`
      - Obtiene el número que el usuario escribió.
    - `attempts++;`
      - Suma un intento cada vez que el jugador adivina.
- Las condiciones `if`, `else if`, `else` comparan el número del jugador con el número secreto:
  - Si es menor, muestra "Demasiado bajo".
  - Si es mayor, muestra "Demasiado alto".
  - Si es igual, muestra un mensaje de victoria con el número de intentos realizados.

## Explicación detallada del archivo styles.css

El archivo styles.css define la apariencia visual del juego.

Mientras que el archivo HTML organiza el contenido, el CSS le da color, forma, tamaño y estilo.

Gracias a este archivo, el juego se ve atractivo, centrado y ordenado, en lugar de ser solo texto plano.

  1. body { ... }

```
body {  
  background-color: #e0f7fa;  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
  margin: 0;  
  color: #333;  
}
```

Define el fondo, el tipo de letra y la alineación general del juego.

- `background-color: #e0f7fa;`
- Pone un color de fondo azul claro. Da una sensación fresca y agradable al jugar.
- `font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;`
- Cambia el tipo de letra de toda la página a una fuente moderna y fácil de leer.
- `display: flex;`
- Activa el modo flexbox, que permite centrar los elementos fácilmente tanto vertical como horizontalmente.
- `flex-direction: column;`
- Hace que los elementos se organicen uno debajo del otro (como en una columna).
- `align-items: center;`
- Centra los elementos horizontalmente.
- `justify-content: center;`
- Centra los elementos verticalmente dentro de la pantalla.
- `height: 100vh;`
- Hace que el cuerpo ocupe toda la altura visible del navegador.
- `margin: 0;`
- Elimina los márgenes por defecto del navegador.
- `color: #333;`
- Define el color del texto en gris oscuro (más suave que el negro).

El cuerpo queda centrado completamente, con un fondo suave y un diseño limpio que prepara el espacio para el contenedor del juego.

## 2. .container { ... }

```
.container {  
  background-color: white;  
  padding: 30px;  
  border-radius: 10px;  
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
  text-align: center;  
  width: 80%;  
  max-width: 500px;  
}
```

Define el cuadro blanco donde se muestra el juego (texto, campo y botón).

Explicación:

- `background-color: white;`
- Pone el fondo del cuadro en blanco, para que contraste con el fondo azul.
- `padding: 30px;`
- Agrega espacio interno dentro del cuadro, evitando que el texto toque los bordes.
- `border-radius: 10px;`
- Redondea las esquinas del contenedor, dándole un estilo moderno.
- `box-shadow: 0 4px 8px rgba(0,0,0,0.1);`
- Agrega una sombra suave debajo del cuadro, creando un efecto de "flotante".
- `text-align: center;`
- Centra todo el texto dentro del contenedor.
- `width: 80%;` y `max-width: 500px;`
- Hace que el cuadro sea responsivo: ocupa el 80% de la pantalla, pero nunca más de 500 píxeles de ancho.
- Así se adapta a pantallas grandes y pequeñas.

El juego aparece dentro de un recuadro blanco, centrado y con un diseño limpio y profesional.

### 3. h1 { ... }

```
h1 {  
    color: #009688;  
    margin-bottom: 20px;  
}
```

Estiliza el título principal del juego.

- `color: #009688;` → le da un tono verde azulado (color característico del tema).
- `margin-bottom: 20px;` → agrega espacio debajo del título para separarlo del texto.

El título “Adivina el Número” destaca en la parte superior con un color vibrante y espacio adecuado.

#### 4. p { ... }

```
p {  
    font-size: 16px;  
    margin-bottom: 20px;  
}
```

Controla el tamaño y espaciado de los párrafos.

- font-size: 16px; → hace el texto cómodo de leer.
- margin-bottom: 20px; → deja espacio entre los párrafos para que no se amontonen.

Los textos de instrucciones y mensajes se ven claros y bien espaciados.

## 5. input[type="number"] { ... }

```
input[type="number"] {  
  padding: 12px;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
  margin-bottom: 20px;  
  width: 70%;  
  font-size: 16px;  
}
```

Diseña el campo donde el jugador escribe su número.

Explicación:

- padding: 12px; → agrega espacio dentro del cuadro de texto.
- border: 1px solid #ccc; → le da un borde gris claro, limpio y discreto.
- border-radius: 4px; → redondea los bordes ligeramente.
- margin-bottom: 20px; → deja espacio entre el campo y el botón.
- width: 70%; → hace que el campo tenga un tamaño adecuado en relación al contenedor.
- font-size: 16px; → agranda el texto dentro del input para hacerlo más visible.

👉 Resultado:

El campo de número se ve elegante, fácil de usar y alineado con el diseño general.

## 6. button { ... } y button:hover { ... }

```
button {
  padding: 14px 28px;
  background-color: #009688;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #00796b;
}
```

Define el aspecto del botón y su efecto al pasar el cursor.

Explicación:

- padding: 14px 28px; → hace el botón más grande y fácil de presionar.
- background-color: #009688; → color verde azulado que combina con el título.
- color: white; → texto blanco para buen contraste.
- border: none; → elimina los bordes por defecto.
- border-radius: 4px; → redondea las esquinas.
- cursor: pointer; → cambia el cursor al pasar encima, indicando que es clicable.
- transition: background-color 0.3s ease; → hace que el color cambie suavemente cuando se pasa el cursor.

Y el bloque:

```
button:hover {  
    background-color: #00796b;  
}
```

Oscurece el botón cuando el jugador pasa el ratón encima, dando interactividad visual.

Un botón moderno con efecto dinámico y profesional.

7. #message { ... }

```
#message {  
    font-size: 18px;  
    font-weight: bold;  
    margin-top: 20px;  
}
```

Define cómo se verá el texto del resultado del juego (mensajes de “Demasiado bajo”, “Demasiado alto” o “¡Felicidades!”).

Explicación:

- font-size: 18px; → agranda el texto para que destaque.
- font-weight: bold; → lo pone en negritas.
- margin-top: 20px; → deja espacio arriba para separarlo del botón.

## Conclusión

El juego “Adivina el Número” es un excelente ejemplo de cómo se pueden integrar los tres lenguajes fundamentales del desarrollo web: HTML, CSS y JavaScript, para crear una aplicación interactiva, funcional y visualmente atractiva.

Por medio del archivo HTML, se construye la estructura del juego y los elementos que el usuario ve e interactúa, como el título, las instrucciones, el campo de entrada y el botón.

El archivo CSS aporta el diseño visual, logrando una interfaz limpia, centrada y con colores agradables que mejoran la experiencia del jugador.

Finalmente, el JavaScript brinda la lógica del juego, permitiendo que el sistema genere un número aleatorio, reciba los intentos del usuario y muestre mensajes dinámicos según el resultado.

En conjunto, estos tres componentes permiten que el juego funcione de manera ordenada, fluida y comprensible.

Además, este proyecto refuerza los conocimientos sobre la manipulación del DOM, la comparación de valores, la interacción con eventos y la aplicación de estilos modernos.