



# AHORCADO

**MANUAL Y JUEGO  
ELABORADO POR:  
EVELYN SANTIAGO  
DISEÑO Y  
MAQUETACIÓN:  
DEREK REVILLA**

# MANUAL DETALLADO DEL JUEGO DEL AHORCADO

Antes de que se utilicen los diferentes documentos que componen el juego, es importante comprender el propósito del proyecto y los lenguajes involucrados en su desarrollo.

El Juego del Ahorcado es una aplicación interactiva diseñada para ejecutarse directamente en el navegador web. Su objetivo principal es permitir que el usuario adivine una palabra secreta seleccionando letras del alfabeto. Por cada error cometido, se irá completando la figura de un ahorcado, hasta que el jugador logre descubrir la palabra o pierda el juego.

Para la creación de este proyecto se emplearon tres tecnologías fundamentales:

- **HTML (HyperText Markup Language):** se utiliza para estructurar el contenido de la página, es decir, crear los elementos visibles como el título, el lienzo de dibujo (canvas) y el botón de reinicio del juego.
- **CSS (Cascading Style Sheets):** se encarga de la parte visual, proporcionando colores, tamaños, posiciones y efectos a los elementos de la página para hacerla más atractiva y funcional.
- **JavaScript:** es el lenguaje encargado de la lógica y el comportamiento del juego. Permite generar el teclado, detectar las letras seleccionadas, manejar los aciertos y errores, y controlar las imágenes del ahorcado.

Estos tres lenguajes se combinan para dar forma a una experiencia interactiva completamente funcional.

Primero, el documento HTML establece la estructura del sitio; luego el CSS define su apariencia; y finalmente, el JavaScript dota al proyecto de movimiento, interactividad y lógica de juego.

Cada parte cumple una función esencial que, en conjunto, hace posible que el juego del ahorcado funcione correctamente en cualquier navegador moderno.

## 1. ESTRUCTURA HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Juego del ahorcado</title>
```

- **<!doctype html>** indica que el documento usa HTML5, la versión más actual del lenguaje.
- **<html>** abre el documento HTML.
- **<head>** contiene la configuración del sitio (no visible directamente).
- **<meta charset="utf-8">** define el tipo de codificación de caracteres, "UTF-8", para soportar acentos y ñ.
- **<title>** define el nombre que aparece en la pestaña del navegador: "Juego del ahorcado".

## 2. ESTILOS CSS (DISEÑO VISUAL)

```
body {  
  width: 960px;  
  margin: 0 auto;  
}
```

DEFINE QUE TODO EL CONTENIDO TENGA UN ANCHO DE 960 PÍXELES Y ESTÉ CENTRADO EN LA PANTALLA (MARGIN: 0 AUTO).

```
.imagen-derecha {  
  position: absolute;  
  top: 40px;  
  left: 60px;  
  width: 90px;  
}
```

DEFINE LA POSICIÓN Y TAMAÑO DE UNA IMAGEN QUE SE COLOCARÁ EN UNA POSICIÓN FIJA.  
POSITION: ABSOLUTE; PERMITE COLOCARLA EN COORDENADAS EXACTAS (40PX DESDE ARRIBA, 60PX DESDE LA IZQUIERDA).

```
h1 {  
  text-align: center;  
}
```

CENTRA EL TÍTULO PRINCIPAL DEL JUEGO ("EL JUEGO DEL AHORCADO").

```
#pantalla {  
  border: groove 8px gold;  
  background: lightgreen;  
}
```

#PANTALLA ES EL CANVAS, EL "LIENZO" DONDE SE DIBUJA EL JUEGO.  
BORDER: GROOVE 8PX GOLD; CREA UN BORDE DORADO EN RELIEVE.  
BACKGROUND: LIGHTGREEN; DA UN COLOR DE FONDO VERDE CLARO.  
DE NEGRO.

```
#boton {  
  background-color: red;  
  color: white;  
  font-size: 20px;  
  text-align: center;  
  font-weight: bolder;  
  padding: 3px;  
  border: solid 2px black;  
}
```

- DEFINE EL ESTILO VISUAL DEL BOTÓN "VOLVER A JUGAR".
- USA ROJO CON LETRAS BLANCAS, TEXTO CENTRADO Y GROSOR DE BORDE NEGRO.

```
#boton:hover {  
  background-color: lightcoral;  
  font-size: 22px;  
  border: groove 4px red;  
}
```

**HOVER SE ACTIVA CUANDO EL CURSOR PASA SOBRE EL BOTÓN. CAMBIA EL COLOR Y EL TAMAÑO DE LETRA, DANDO SENSACIÓN DE INTERACCIÓN (EFECTO VISUAL DINÁMICO).**

### 3. CONTENIDO VISIBLE EN HTML

```
<h1>-El juego del ahorcado-</h1>
<canvas id="pantalla" width="960px" height="450px">
  Tu navegador no soporta Canvas.
</canvas>
```

**<H1>: TÍTULO PRINCIPAL DEL JUEGO.**

**<CANVAS>: ES EL "LIENZO" DONDE SE DIBUJARÁN LETRAS, TECLAS E IMÁGENES (AHORCADO).**

**EL TEXTO DENTRO ("TU NAVEGADOR NO SOPORTA CANVAS.") APARECE SOLO SI EL NAVEGADOR ES ANTIGUO Y NO SOPORTA LA ETIQUETA <CANVAS>.**

```
<button id="boton" type="reset" onclick="javascript:window.location.reload();">Volver a
```

**CREA UN BOTÓN QUE RECARGA TODA LA PÁGINA CUANDO SE HACE CLIC, REINICIANDO EL JUEGO.**

**WINDOW.LOCATION.RELOAD() VUELVE A CARGAR EL DOCUMENTO HTML DESDE CERO.**

### 4. SCRIPT JAVASCRIPT (LÓGICA DEL JUEGO)

**EL CORAZÓN DEL PROGRAMA. AQUÍ SE CREAN LAS FUNCIONES QUE PERMITEN DIBUJAR, DETECTAR CLICS, MOSTRAR LETRAS, ETC.**

#### VARIABLES PRINCIPALES

```
var ctx;
var canvas;
var palabra;
var letras = "QWERTYUIOPASDFGHJKLÑZXCVBNM";
```

**CANVAS Y CTX: SE USAN PARA DIBUJAR EN EL <CANVAS>.**

**CTX (CONTEXTO) ES LO QUE REALMENTE PERMITE USAR HERRAMIENTAS DE DIBUJO (TEXTO, RECTÁNGULOS, IMÁGENES).**

**PALABRA: ALMACENARÁ LA PALABRA SECRETA DEL JUEGO.**

**LETRAS: CONTIENE TODAS LAS LETRAS DISPONIBLES DEL TECLADO (EN MAYÚSCULAS).**

```
var ctx;
var canvas;
var palabra;
var letras = "QWERTYUIOPASDFGHJKLÑZXCVBNM";
```

- **DEFINE COLORES, POSICIONES Y DIMENSIONES PARA EL TECLADO QUE APARECE EN PANTALLA.**
- **INICIOX Y INICIOY: COORDENADAS DONDE EMPEZARÁN A DIBUJARSE LAS**

LON: TAMAÑO DE CADA TECLA (35PX).

MARGEN: ESPACIO ENTRE TECLAS.

## ARREGLOS PRINCIPALES

```
var teclas_array = new Array();  
var letras_array = new Array();  
var palabras_array = new Array();
```

TECLAS\_ARRAY: GUARDA TODOS LOS OBJETOS DE TIPO "TECLA".

LETRAS\_ARRAY: GUARDA LOS CUADROS DONDE SE MOSTRARÁN LAS LETRAS ADIVINADAS.

PALABRAS\_ARRAY: CONTIENE LA LISTA DE PALABRAS POSIBLES PARA JUGAR.

## PALABRAS DISPONIBLES

```
palabras_array.push("LEON");  
palabras_array.push("CABALLO");  
...  
palabras_array.push("AGUILA");
```

SE AGREGAN PALABRAS AL ARREGLO PALABRAS\_ARRAY.

.PUSH() METE UN ELEMENTO NUEVO AL FINAL DEL ARREGLO.

ESTAS SERÁN LAS POSIBLES PALABRAS QUE SALDRÁN ALEATORIAMENTE.

## 5. CONSTRUCTORES DE OBJETOS

ESTOS CREAN "PLANTILLAS" PARA LAS TECLAS Y LAS LETRAS.

### TECLAS

```
function Tecla(x, y, ancho, alto, letra){  
  this.x = x;  
  this.y = y;  
  this.ancho = ancho;  
  this.alto = alto;  
  this.letra = letra;  
  this.dibuja = dibujaTecla;  
}
```

DEFINE UNA TECLA CON SU POSICIÓN, TAMAÑO Y LETRA QUE REPRESENTA. GUARDA TAMBIÉN QUÉ FUNCIÓN USARÁ PARA DIBUJARSE (DIBUJATECLA).

### LETRAS

```
function Letra(x, y, ancho, alto, letra){  
  this.x = x;  
  this.y = y;  
  this.ancho = ancho;  
  this.alto = alto;  
  this.letra = letra;  
  this.dibuja = dibujaCajaLetra;  
  this.dibujaLetra = dibujaLetraLetra;  
}
```

Representa cada espacio en blanco de la palabra.  
Puede dibujar su "caja vacía" y también mostrar la letra correcta cuando se acierta.

## 6. FUNCIONES DE DIBUJO

### Dibujar teclas

```
function dibujaTecla(){
  ctx.fillStyle = colorTecla;
  ctx.strokeStyle = colorMargen;
  ctx.fillRect(this.x, this.y, this.anch, this.alto);
  ctx.strokeRect(this.x, this.y, this.anch, this.alto);

  ctx.fillStyle = "white";
  ctx.font = "bold 20px courier";
  ctx.fillText(this.letra, this.x+this.anch/2-5, this.y+this.alto/2+5);
}
```

Dibuja una tecla con un rectángulo gris oscuro y borde rojo.  
Escribe la letra blanca encima, centrada.

Dibujar letras de la palabra  
Se ejecuta cuando el jugador acierta una letra.  
Muestra la letra en negro grande en el cuadro correspondiente.

### Dibujar caja de letra vacía

```
function dibujaCajaLetra(){
  ctx.fillStyle = "white";
  ctx.strokeStyle = "black";
  ctx.fillRect(this.x, this.y, this.anch, this.alto);
  ctx.strokeRect(this.x, this.y, this.anch, this.alto);
}
```

Dibuja los cuadros blancos (vacíos) donde van las letras de la palabra secreta.

## 7. FUNCIÓN DE PISTAS

```
function pistaFunction(palabra){
  let pista = "";
  switch(palabra){
    case 'LEON':
      pista = "Ruge y es fuerte";
      break;
    ...
    default:
      pista="No hay pista aun xP";
  }
  ctx.fillStyle = "black";
  ctx.font = "bold 20px Courier";
  ctx.fillText(pista, 10, 15);
}
```

- Muestra una pista en la parte superior del lienzo según la palabra seleccionada.
- Usa switch para elegir la frase correcta.

## ◆ 8. DIBUJAR TECLADO

```
function teclado(){
  var ren = 0;
  var col = 0;
  ...
  for(var i = 0; i < letras.length; i++){
    letra = letras.substr(i,1);
    mIletra = new Tecla(x, y, lon, lon, letra);
    mIletra.dibuja();
    teclas_array.push(mIletra);
    ...
  }
}
```

Dibuja todas las teclas (de la A a la Z y la Ñ).  
Las coloca en tres filas, con coordenadas calculadas dinámicamente.  
Cada tecla se guarda en el arreglo `teclas_array`.

## ◆ 9. ELEGIR Y DIBUJAR PALABRA

```
function pintaPalabra(){
  var p = Math.floor(Math.random()*palabras_array.length);
  palabra = palabras_array[p];
  pistaFunction(palabra);
  ...
}
```

Escoge una palabra al azar (`Math.random()`).  
Llama a `pistaFunction()` para mostrar la pista correspondiente.  
Dibuja los cuadros vacíos de cada letra en el centro del canvas.

## ◆ 10. DIBUJAR IMAGEN DEL AHORCADO

```
function horcaerrores{
  var imagen = new Image();
  imagen.src = "imagenes/ahorcado"+errores+".png";
  imagen.onload = function(){
    ctx.drawImage(imagen, 390, 0, 230, 230);
  }
}
```

Muestra la imagen del ahorcado según el número de errores.  
Cada imagen (`ahorcado0.png`, `ahorcado1.png`...) representa una etapa del dibujo.

## ◆ 11. DETECTAR CLICS

```
function selecciona(e){
  var pos = ajusta(e.clientX, e.clientY);
  ...
}
```

Se activa cuando el jugador da clic sobre una tecla dibujada.  
Detecta en qué coordenadas del canvas ocurrió el clic.  
Revisa si esa posición coincide con alguna tecla.

Si la letra coincide:

```
if (letra == tecla.letra){
  caja = letras_array[i];
  caja.dibujaLetra();
  aciertos++;
}
```

Dibuja la letra dentro del cuadro correcto y suma un acierto.

Si no coincide:

```
errores++;  
horca(errores);  
if (errores == 5) gameOver(errores);
```

Aumenta el contador de errores y actualiza la imagen del ahorcado.  
Si llega a 5 errores, el juego termina.

## 12. FINAL DEL JUEGO

```
function gameOver(errores){  
  ctx.clearRect(0, 0, canvas.width, canvas.height);  
  ctx.fillStyle = "black";  
  ctx.font = "bold 50px Courier";  
  ...  
}
```

Limpia toda la pantalla.

Muestra el mensaje final:

Si se adivina la palabra, dice "Muy bien".

Si no, "Lo sentimos".

Vuelve a dibujar la palabra completa y la horca final.

## 13. INICIO AUTOMÁTICO

```
window.onload = function(){  
  canvas = document.getElementById("pantalla");  
  if (canvas && canvas.getContext){  
    ctx = canvas.getContext("2d");  
    if(ctx){  
      teclado();  
      pintaPalabra();  
      horca(errores);  
      canvas.addEventListener("click", selecciona, false);  
    }  
  }  
}
```

Se ejecuta automáticamente cuando la página termina de cargar.

Inicializa el canvas y su contexto (ctx).

Dibuja el teclado, la palabra y la primera imagen.

Activa el evento de clics para jugar.

## CONCLUSIÓN GENERAL

Este código combina HTML (estructura), CSS (diseño visual) y JavaScript (lógica del juego).

El uso del canvas permite representar gráficamente el teclado, la palabra y las imágenes del ahorcado sin usar elementos HTML adicionales.

Cada función tiene una responsabilidad clara:

dibujaTecla / dibujaCajaLetra: control visual.

selecciona: detección de clics.

horca: retroalimentación visual.

gameOver: cierre de partida.

pintaPalabra: gestión de palabra y posición.