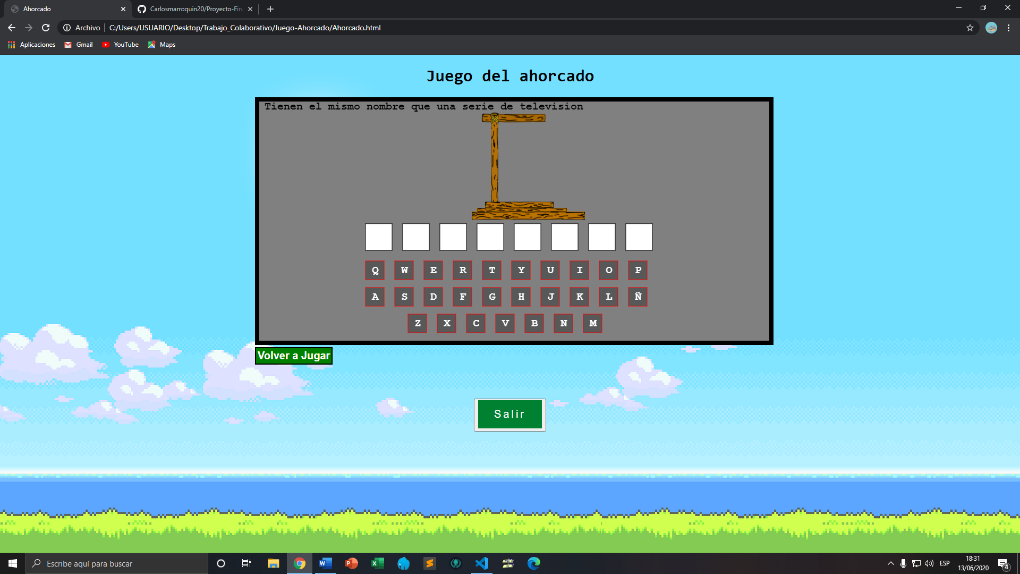
**Documentación Externa**

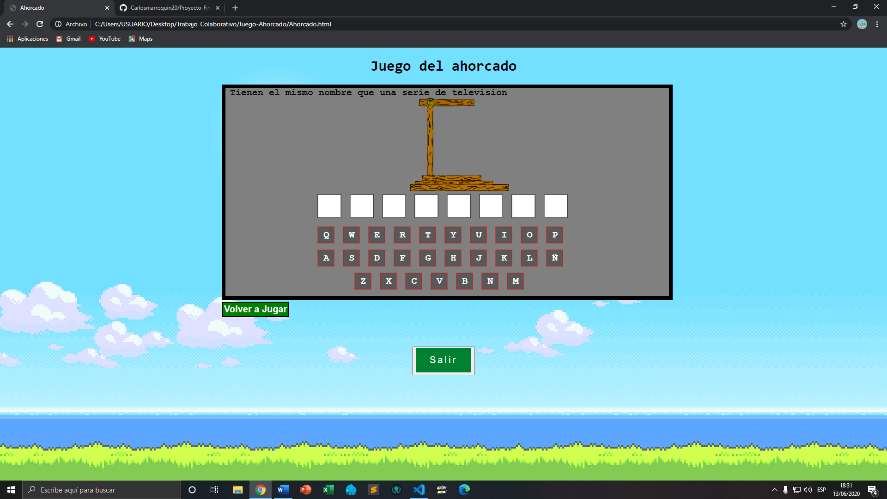
**El Ahorcado**

* **Descripción paso a paso.**

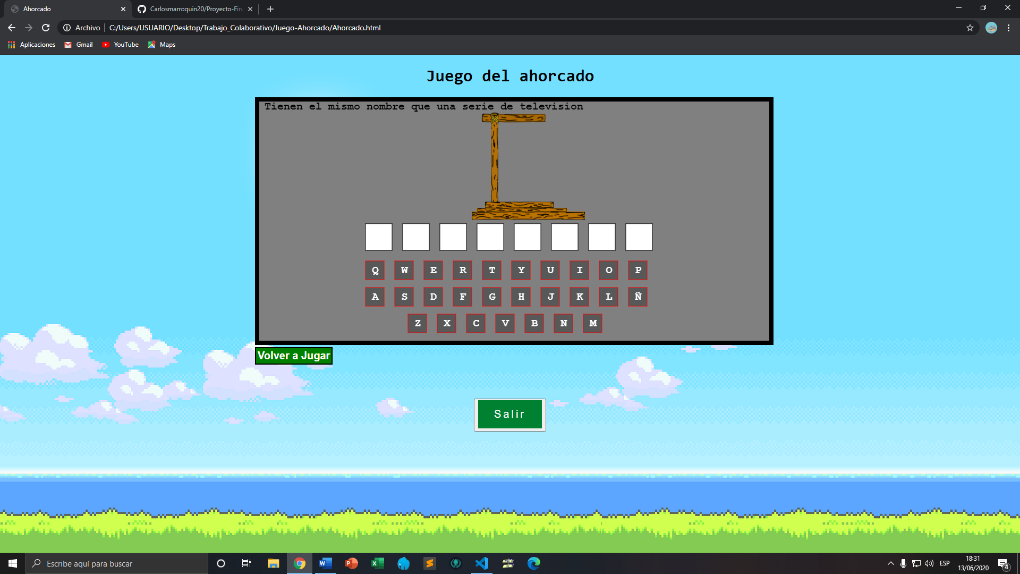
Se requiere del cargado de un canva para el dibujado de las animaciones que va a tener el programa, se requieren de ocho cuadro de texto en los cuales se muestran las letras que servirán de pistas para formar la palabra, así mismo, se requiere de todas las letras del abecedario en botones individuales para poder formas las palabras, también se necesita la función del dibujado del cambas para que al momento de fallar una letra se dibuje otra imagen en el lienzo, también se requiere de un botón para reiniciar y una función que este ligada a este botón que permita generar otras pistas, por último se necesita un botón para poder Salir a la pagina principal.

* **Programa.**
* **Manual de Usuario.**

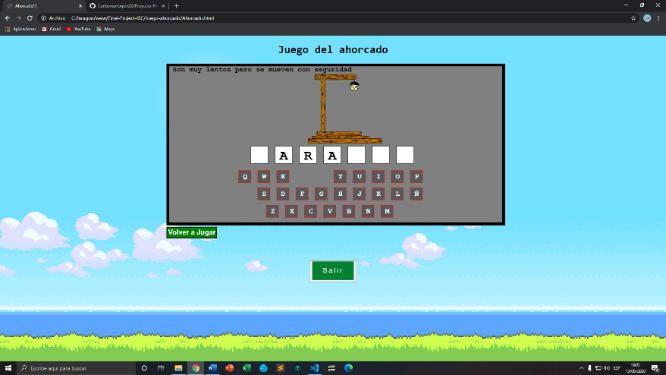
1. Al momento de ingresar a la página se le mostrara lo siguiente:

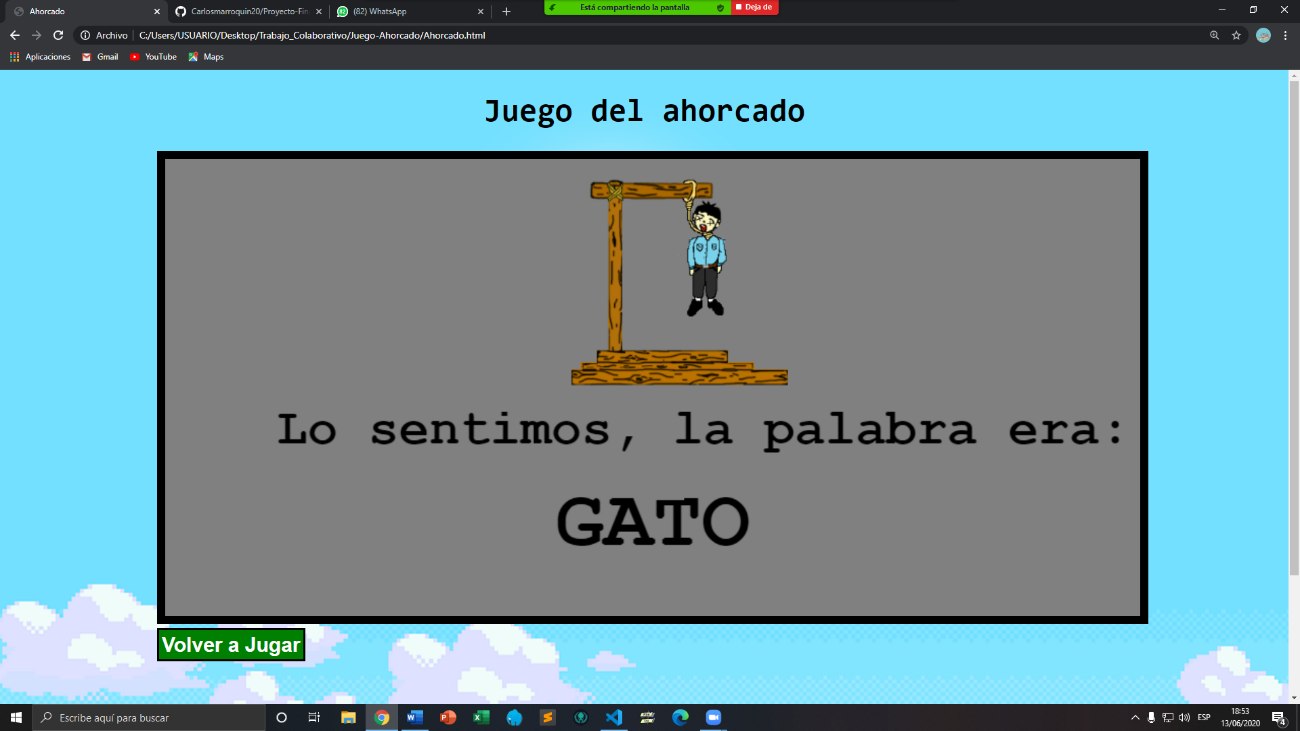


1. Se le mostrara ciertas pistas.

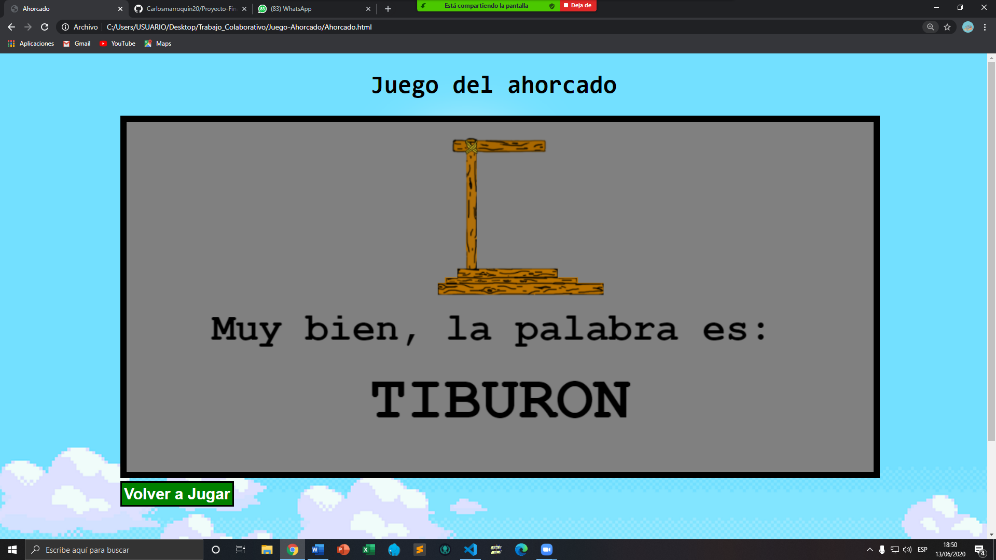


1. Seleccionar las letras con las que crea que se formara la palabra



1. Si el usuario gana el juego se le mostrara lo siguiente:
2. Si se coloca una letra que no es la correcta se dibujara lo siguiente en el canvas:

1. Si pierde el juego se le mostrara lo siguiente:



* **Descripción del problema.**

Se requieren de diferentes funciones que estén ligadas y que estas al momento de ingresar la respuesta incorrecta: Se dibuje cierta parte de la imagen y si perdemos que nos muestre el mensaje de que es incorrecto, y si es correcta que las letras que presionamos se muestren en el cuadro de texto y que nos diga que la respuesta es correcta. También se necesita de un botón que reinicie el juego y un botón que permita salir.

* **Nombre del autor.**

Carlos Emanuel Marroquin Ortega (Carlosmarroquin20)

* **Algoritmo.**

1. Abrir el programa
2. Ver si el canvas cargo correctamente
3. Generacion de la pista
4. Verificar si la letra es correcta
5. Mensaje que nos mostrara si ganamos o perdemos

* **Diagrama de Flujo.**

ABRIR EL PROGRAMA

NO

Ver si el canvas cargo correctamente correctamente

Generación de la pista

Verificación de la respuesta

Verificación si ganamos o perdemos

si

no

Mensaje que ganamos

Mensaje si perdemos presionar volver a jugar

* **Diccionario de Datos**

|  |  |  |
| --- | --- | --- |
| Termino | Tipo | Descripción |
| **HTML** | | |
| head | Metadatos | Provee información general del documento. |
| body | Contenedor | Contiene todos los aspectos de un documento. |

**CSS**

|  |  |  |
| --- | --- | --- |
| background-image | Propiedad | Establece una o más imágenes de fondo para un elemento. |
| background-size | Propiedad | Especifica el tamaño de las imágenes de fondo. |
| text-align | Propiedad | Especifica la alineación horizontal del texto en un elemento. |
| font-family | Propiedad | Especifica la fuente para un elemento. |
| text-decoration | Propiedad | Especifica la decoración agregada al texto |
| font-size | Propiedad | Especifica la dimensión de la letra. |
| background | Propiedad | Especifica el color de fondo de un elemento. |

**Js**

|  |  |  |
| --- | --- | --- |
| if | Condición | Ejecuta una sentencia si una condición especificada es evaluada como verdadera en JS. |
| onload | Evento | Un evento tipo handler para el evento load event del window. |
| function | Función | La función se ejecuta cuando "algo" lo invoca o llama. |
| window | Objeto | Esta sección proporciona una pequeña referencia a todos los métodos. |
| setTimeout | Método | El método setTimeout () llama a una función o evalúa una expresión después de un número específico de milisegundos. |
| array | Objeto | Son objetos tipo-lista, cuyo prototipo tiene métodos para efectuar operaciones de recorrido y de mutación. |
| document | Objeto | Se controla la página web y todos los elementos que contiene. |
| addEventListener | Cadena tipo Evento | Registra un evento a un objeto en específico. |
| var | Sentencia | Declara una variable, opcionalmente inicializándola con un valor. |
| this | Sentencia | Atajo que refiere al objeto desde donde se invocó la función. |
| else | Condición | Si la condición es evaluada como falsa, otra sentencia puede ser ejecutada. |

* **Codigo fuente**

**Html**

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" a href= "AHORCADOs.css"></lin>

    <title>Ahorcado!!!</title>

    <style>

        body{

            width: 960px;

            margin: 0 auto;

        }

        h1{

            text-align: center;

        }

        #pantalla{

            border: groove 8px black;

            background: gray;

        }

        #boton{

            background-color: green;

            color: white;

            font-size: 20px;

            text-align: center;

            font-weight: bolder;

            padding: 3px;

            border: solid 2px black;

        }

        /\*Estilo que se activa al pasar el puntero sobre el boton\*/

        #boton:hover{

            background-color: rgb(2, 146, 33);

            font-size: 22px;

            border: groove 4px gray;

        }

    </style>

</head>

<body>

    <h1>Juego del ahorcado</h1>

<!--Etiqueta del canvas con sus medidas en la pantalla-->

<canvas id="pantalla" width="960px" height="450px">

    Tu navegador no soporta Canvas.

    </canvas>

<!--El boton que nos sirve para recargar la pagina y asi generar una nueva palabra y volver a jugar-->

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

<script>

  /\*Variables\*/

  var ctx;

        var canvas;

        var palabra;

        var letras = "QWERTYUIOPASDFGHJKLÑZXCVBNM";

        var colorTecla = "#585858";

        var colorMargen = "red";

        var inicioX = 200;

        var inicioY = 300;

        var lon = 35;

        var margen = 20;

        var pistaText = "";

      /\*Arreglos\*/

      var teclas\_array = new Array();

        var letras\_array = new Array();

        var palabras\_array = new Array();

       /\*Variables de control\*/

       var aciertos = 0;

        var errores = 0;

        /\*Palabras\*/

        palabras\_array.push("LEON");

        palabras\_array.push("CABALLO");

        palabras\_array.push("PERRO");

        palabras\_array.push("GATO");

        palabras\_array.push("LAGARTIJA");

        palabras\_array.push("RINOCERONTE");

        palabras\_array.push("TIBURON");

        palabras\_array.push("CARACOL");

        palabras\_array.push("ALACRAN");

        palabras\_array.push("ARAÑA");

        palabras\_array.push("CHAPULIN");

        palabras\_array.push("AVESTRUZ");

        palabras\_array.push("OCELOTE");

        palabras\_array.push("MUSARAÑA");

        palabras\_array.push("AGUILA");

      /\*Objetos\*/

      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;

        }

        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;

        }

       /\*Funciones\*/

        /\*Dibujar Teclas\*/

        function dibujaTecla(){

            ctx.fillStyle = colorTecla;

            ctx.strokeStyle = colorMargen;

            ctx.fillRect(this.x, this.y, this.ancho, this.alto);

            ctx.strokeRect(this.x, this.y, this.ancho, this.alto);

            ctx.fillStyle = "white";

            ctx.font = "bold 20px courier";

            ctx.fillText(this.letra, this.x+this.ancho/2-5, this.y+this.alto/2+5);

        }

       /\*Dibua la letra y su box\*/

       function dibujaLetraLetra(){

            var w = this.ancho;

            var h = this.alto;

            ctx.fillStyle = "black";

            ctx.font = "bold 40px Courier";

            ctx.fillText(this.letra, this.x+w/2-12, this.y+h/2+14);

        }

        function dibujaCajaLetra(){

            ctx.fillStyle = "white";

            ctx.strokeStyle = "black";

            ctx.fillRect(this.x, this.y, this.ancho, this.alto);

            ctx.strokeRect(this.x, this.y, this.ancho, this.alto);

        }

       ///Pista para el usuario (Funcion)///

       function pistaFunction(palabra){

            let pista = ""; //Variable para dar pistas

            switch(palabra){  //Switch para la ordenacion de las pistas segun la pregunta

                case 'LEON':   //Case individual para cada palabra

                    pista = "Ruge y es fuerte";

                    break;     //Break para cada funsion

                case 'CABALLO':

                    pista = "Hay de tierra y hay de mar";

                    break;

                case 'PERRO':

                    pista = "El mejor amigo del hombre";

                    break;

                case "GATO":

                    pista = "Son tiernos pero arañan";

                    break;

                case "LAGARTIJA":

                    pista = "Son mas pequeñas que sus parientes los lagartos";

                    break;

                case "RINOCERONTE":

                    pista = "Tienen un gran cuerno encima de su ocico";

                break;

                case "TIBURON":

                    pista = "Son los reyes del mar";

                    break;

                case "CARACOL":

                    pista = "Son muy lentos pero se mueven con seguridad";

                break;

                case "ALACRAN":

                    pista = "Tienen un agijon poderoso";

                break;

                case "ARAÑA":

                    pista = "Tejen telarañas";

                break;

                case "CHAPULIN":

                    pista = "Tienen el mismo nombre que una serie de television";

                break;

                case "AVESTRUZ":

                    pista = "Son las aves mas grandes del mundo";

                break;

                case "OCELOTE":

                    pista = "Los gatos salvajes";

                break;

                case "MUSARAÑA":

                    pista = "Animal que convina a un mouse y a una araña";

                break;

                case "AGUILA":

                    pista = "Son las aves que mejor vista tienen";

                break;

                default:  //Creacion del default//

                    pista="No hay pista aun xP";

            }

           //Dibujado en el canvas//

           ctx.fillStyle = "black";  //Color de la letra

            ctx.font = "bold 20px Courier";  //Tipo y tamaño de letra

            ctx.fillText(pista, 10, 15);  //Variable pista

        }

       /\*Distribucion del teclado con sus letras respectivas al acomodo de nuestro array\*/

       function teclado(){

            var ren = 0;

            var col = 0;

            var letra = "";

            var miLetra;

            var x = inicioX;

            var y = inicioY;

            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);

                x += lon + margen;

                col++;

                if(col==10){

                    col = 0;

                    ren++;

                    if(ren==2){

                        x = 280;

                    } else {

                        x = inicioX;

                    }

                }

                y = inicioY + ren \* 50;

            }

        }

        /\*Obtencion de palabra aleatoria y division en letras\*/

        function pintaPalabra(){

            var p = Math.floor(Math.random()\*palabras\_array.length);

            palabra = palabras\_array[p];

            pistaFunction(palabra);

            var w = canvas.width;

            var len = palabra.length;

            var ren = 0;

            var col = 0;

            var y = 230;

            var lon = 50;

            var x = (w - (lon+margen) \*len)/2;

            for(var i=0; i<palabra.length; i++){

                letra = palabra.substr(i,1);

                miLetra = new Letra(x, y, lon, lon, letra);

                miLetra.dibuja();

                letras\_array.push(miLetra);

                x += lon + margen;

            }

        }

       /\* dibujado del pj\*/

       function horca(errores){

            var imagen = new Image();

            imagen.src = "imagenes/ahorcado"+errores+".png";

            imagen.onload = function(){

                ctx.drawImage(imagen, 390, 0, 230, 230);

            }

        }

        /\*Ajuste de coordenadas\*/

        function ajusta(xx, yy){

            var posCanvas = canvas.getBoundingClientRect();

            var x = xx-posCanvas.left;

            var y = yy-posCanvas.top;

            return{x:x, y:y}

        }

        /\*Deteccion de tecla clickeada y comparacion con las de la palabra ya elegida al azar\*/

        function selecciona(e){

            var pos = ajusta(e.clientX, e.clientY);

            var x = pos.x;

            var y = pos.y;

            var tecla;

            var bandera = false;

            for (var i = 0; i < teclas\_array.length; i++){

                tecla = teclas\_array[i];

                if (tecla.x > 0){

                    if ((x > tecla.x) && (x < tecla.x + tecla.ancho) && (y > tecla.y) && (y < tecla.y + tecla.alto)){

                        break;

                    }

                }

            }

            if (i < teclas\_array.length){

                for (var i = 0 ; i < palabra.length ; i++){

                    letra = palabra.substr(i, 1);

                    if (letra == tecla.letra){ /\*Comparado de resultado con la letra correcta\*/

                        caja = letras\_array[i];

                        caja.dibujaLetra();

                        aciertos++;

                        bandera = true;

                    }

                }

                if (bandera == false){ /\*Si falla aumenta los errores y checa si perdio para mandar a la funcion gameover\*/

                    errores++;

                    horca(errores);

                    if (errores == 5) gameOver(errores);

                }

                /\* Borra la tecla que se a presionado \*/

                ctx.clearRect(tecla.x - 1, tecla.y - 1, tecla.ancho + 2, tecla.alto + 2);

                tecla.x - 1;

                /\* checa si se gano y manda a la funcion gameover \*/

                if (aciertos == palabra.length) gameOver(errores);

            }

        }

        /\*Borramos las teclas y la palabra con sus cajas y mandamos msj segun el caso si se gano o se perdio\*/

        function gameOver(errores){

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

            ctx.fillStyle = "black";

            ctx.font = "bold 50px Courier";

            if (errores < 5){

                ctx.fillText("Muy bien, la palabra es: ", 110, 280);

            } else {

                ctx.fillText("Lo sentimos, la palabra era: ", 110, 280);

            }

            ctx.font = "bold 80px Courier";

            lon = (canvas.width - (palabra.length\*48))/2;

            ctx.fillText(palabra, lon, 380);

            horca(errores);

        }

        /\*Detectar si se a cargado nuestro contexco en el canvas\*/

        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);

                } else {

                    alert ("Error al cargar el contexto!");

                }

            }

        }

</script>

<p id= "p1">

    <button><a href="C:\laragon\www\Final-Project-ISC\Pagina Principal\Indexp.html" class="btn-neon">

        <span id="span1"></span>

        <span id="span2"></span>

        <span id="span3"></span>

        <span id="span4"></span>

    Salir

    </a></button>

</p>

</body>

</html>

**CSS**

body{

    background-image: url("C:/laragon/www/Final-Project-ISC/Juego-ahorcado/imagenes/ahorcado.jpg") ;

}

h1{

    font-family: consolas;

}

p{

    display: flex;

    justify-content: center;

    align-items: center;

    height: 150px;

    width: center;

    background-size: auto;

    font-family: consolas;

    margin: 20px;

    }

    .btn-neon{

    position: relative;

    display: inline-block;

    padding: 15px 30px;

    color: white;

    letter-spacing: 4px;

    text-decoration: none;

    font-size: 20px;

    overflow: hidden;

    transition: 0.5s;

    background-color: rgb(0, 128, 49);

    }

    .btn-neon:hover{

     background: gray;

     box-shadow: 0 0 10px gray, 0 0 40px gray, 0 0 80px gray;

     transition-delay: 1s;

     }

    .btn-neon span{

     position: absolute;

     display: block;

            }

     #span1{

     top: 0;

     left: -100%;

     width: 100%;

     height: 2px;

     background: linear-gradient(90deg, transparent,#a945c7);

     }

     .btn-neon:hover #span1{

     left: 100%;

     transition: 0.5s;

     }

     #span3{

     bottom: 0;

     right: -100%;

     width: 100%;

     height: 2px;

     background: linear-gradient(270deg, transparent,#a945c7);

     }

     .btn-neon:hover #span3{

      right: 100%;

      transition: 1s;

      transition-delay: 0.5s;

      }

      #span2{

      top: -100%;

      right: 0;

      width: 2px;

      height: 100%;

      background: linear-gradient(180deg, transparent,#a945c7);

      }

      .btn-neon:hover #span2{

      top: 100%;

      transition: 1s;

      transition-delay: 0.25s;

      }

      #span4{

      bottom: -100%;

      left: 0;

      width: 2px;

      height: 100%;

      background: linear-gradient(360deg, transparent,#a945c7);

      }

      .btn-neon:hover #span4{

      bottom: 100%;

      transition: 1s;

      transition-delay: 0.75s;

      }