

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS

PROGRAMACIÓN AVANZADA

PROF. OSWALDO ALBERTO ROMERO VILLALOBOS

INFORME PROYECTO FINAL

JORGE ANDRÉS BOHÓRQUEZ CASTELLANOS - 20181020016

JOHAN AGUIRRE DÍAZ - 20181020072

CHRISTIAN GALINDO -20181020111

Objetivo :

Como proyecto final se realizó una pagina web de ventas con el propósito de ver cómo se implementa el HTML, el CSS y Funciones integradas al php; además de cómo se vinculan una base de datos en este entorno.

Índice	pág
I. Html _____	3
II. Php _____	4
III. Css _____	5
IV. Funciones Php _____	9
V. Base de Datos _____	10

I. HTML

index.php

En esta hoja de código se encuentra todo el código HTML y en sí, toda la configuración que se usará de manera global

```
<?php
include "configs/config.php";
include "configs/abrirBD.php";
include "configs/funciones.php";

if(!isset($p)){
    $p="inicio";
}else{
    $p = $p;
}
?>

<!DOCTYPE html>
<html lang="es">
```

En este primer trozo de código incluimos la configuración que se usará de manera global en el programa

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/fontello.css">
    <link rel="stylesheet" href="css/estilos.css">
    <!--<link rel="stylesheet" href="fontawesome/css/all.css">-->
    <script type="text/javascript" src="bootstrap/js/bootstrap.js"></script>
    <script type="text/javascript" src="fontawesome/js/all.js"></script>
    <script type="text/javascript" src="js/jquery.js"></script>
    <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
    <title>Games</title>
</head>
```

Luego creamos la cabeza del HTML donde se llaman los estilos que se van a usar y otros elementos del bootstrap

```
<body>
    <header>
        <div class="contenedor">
            <h1 class="icon-gamepad">Games</h1>
            <input type="checkbox" id="menu-bar">
            <label class="icon-menu" for="menu-bar"></label>
            <nav class="menu">
                <a href="?p=inicio">Inicio</a>
                <a href="?p=juegos">Juegos</a>
                <a href="?p=ofertas">Ofertas</a>
                <a href="?p=carrito">Carrito</a>
                <?php
                if(!isset($_SESSION['k_id'])){
                >
                    <a href="?p=ingresar">Ingresar</a>
                <?php
                }
                <?php
                if(isset($_SESSION['k_id'])){
                >
                    <a href="?p=compras">Usuario: <?=nombre_cliente($_SESSION['k_id'])?></a>
                    <a href="?p=salir" id="ocultar">Salir</a>
                <?php
                }
                >
            </nav>
        </div>
    </header>
```

```

<div class="cuerpo">
  <main>
    <?php
      if(file_exists("modulos/".$p.".php")){
        include "modulos/".$p.".php";
      }else{
        echo "<i>No se ha encontrado el modulo <b>".$p."<b> <a href='.'/'>Regresar</a></i>";
      }
    <?>
  </main>
</div>

<footer>
  <div class="contenedor">
    <p class="copy">Games &copy; 2020</p>
    <div class="sociales">
      <a class="icon-facebook-official" href="#"></a>
      <a class="icon-twitter" href="#"></a>
      <a class="icon-instagram" href="#"></a>
      <a class="icon-gplus-squared" href="#"></a>
    </div>
  </div>
</footer>
</body>
</html>

```

Luego de eso creamos el body, donde diseñamos y generamos la estructura de la página con ayuda del CSS especificado para el banner y todas las imágenes usadas en la página principal.

II. PHP

carrito.php:

En esta hoja de código está toda la parte lógica detrás de la realización de una compra. Gracias a esta se dan las opciones de completar la compra, modificar la cantidad de productos que deseamos o, incluso, eliminar un producto del carrito. Toda esta información también se pasa a la base de datos por medio de la creación de la tabla y sentencias SQL que pasan la información a la base de datos.

compras.php:

Esta hoja de código nos permite ver en la página el registro de compras, es decir, que hemos comprado, y por supuesto, cuánto hemos gastado en cada compra.

ingresar.php:

Esta hoja de código nos permite hacer el inicio de sesión en la página web, por medio tanto de un usuario como de un contraseña, las cuales (sí se hizo correctamente el registro) se encuentran guardadas en la base de datos.

inicio.php:

Esta hoja de código lo que hace es aportar la parte lógica a la página principal, aquí llamamos los banners y también están los encabezados de texto,

juegos.php:

Esta hoja de código es la encargada de crear nuestra barra para filtrar por categoría y los botones para agregar los juegos al carrito en la página de juegos. En esta es donde se encuentra todo nuestro catálogo de juegos disponibles.

ofertas.php:

Esta hoja de código es la encargada de crear nuestra barra para filtrar por categoría y los botones para agregar los juegos al carrito en la página de ofertas. En esta es donde se encuentra todo nuestro catálogo de juegos disponibles en oferta.

registro.php:

Esta hoja de código lo que hace es crear los campos de texto para que el usuario pueda hacer su registro, pidiendo un nombre de usuario, una contraseña y la reconfirmación de la contraseña. Toda esta información se pasa como un nuevo registro a la base de datos, también se verifica si el usuario ya existe, en tal caso la página sacará un aviso advirtiéndole que el registro no es posible debido a esto.

config.php:

Estas líneas de código tienen como función controlar las sesiones que se hagan en la página.

```
<?php
    @session_start();
    @extract($_REQUEST);
?>
```

salir.php:

Estas líneas de código lo que hacen es destruir la sesión cuando el usuario decide cerrar sesión.

```
<?php
    @session_destroy();
    redir("?p=inicio");
?>
```

ver_compra.php:

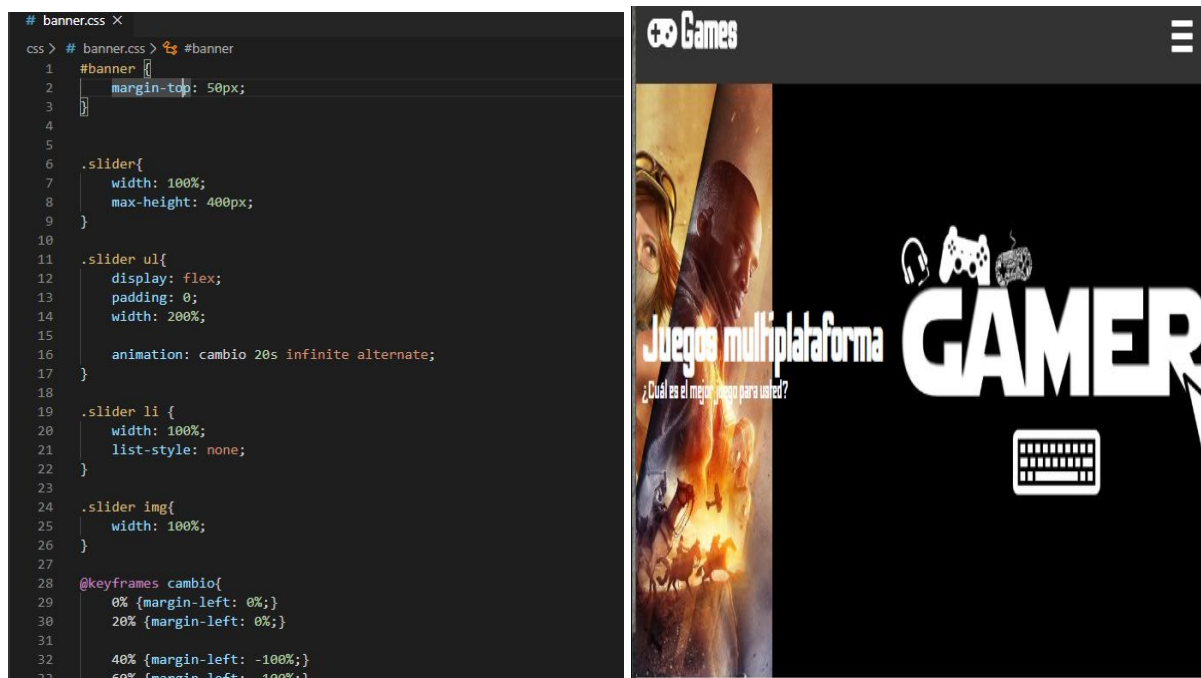
Esta hoja de código es la que nos permite los detalles de una compra realizada, es decir, qué productos compramos, la cantidad y el monto final de esa compra.

III. CSS

Dentro de las diferentes hojas de CSS podemos encontrar los siguientes códigos y su funcionamiento que cumplen con el criterio responsive.

Banner.css

Esta hoja de CSS es la encargada de las Imágenes Principales



Como podemos ver en el código, cada clase que se encuentra ahí tiene como objetivo el permitir que la imagen se pueda redimensionar. Tanto en largo como ancho esto con el objetivo de que la imagen quede en proporción con el tamaño de la página. Con el uso de los keyframes podemos realizar una animación con las dos imágenes que podemos ver; esto se logra trasladando la margen de la imagen de tal forma que se convierta una transición alternada entre las dos.

Catalogo.css

Esta hoja se encarga del catálogo de su posición y estilo específicamente de las imágenes de nuestro catálogo.

```
#catalogo h3,h4{
  text-align: center;
}

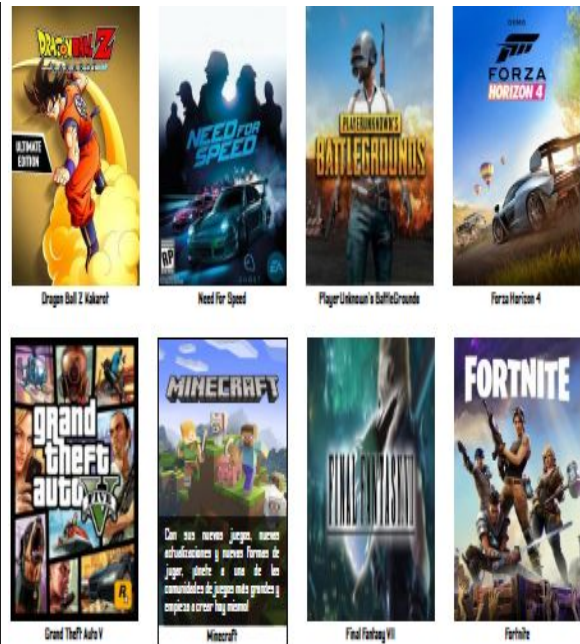
#catalogo .contenedor{
  display: flex;
  justify-content: center;
  flex-wrap: wrap;
}

#catalogo article{
  margin: 15px;
}

#categoria{
  margin: auto;
  height: 20px;
  width: 98%;
}

#titulo-cat{
  margin: 10px auto;
  text-align: center;
  color: rgba(219, 106, 72, 0.966);
}

.item2{
  position: relative;
  overflow: hidden;
}
```



Como podemos ver en el código de la imagen cada clase aparte de encargarse de darle tamaño y estilo se encarga de la animación cuando emerge el texto como se ve en la imagen de la derecha. Esto se logra con la etiqueta overflow con el valor de hidden la cual permite ocultar el texto sobrante y luego con una transición se desplaza para poder ver todo el texto como se ve en la otra imagen.

Carrito.css

Esta hoja es la encargada del estilo del carrito de los productos

```
# carrito.css > #titulo-tabla
#titulo-tabla{
  width: 98%;
  margin: 10px auto;
  text-align: left;
}

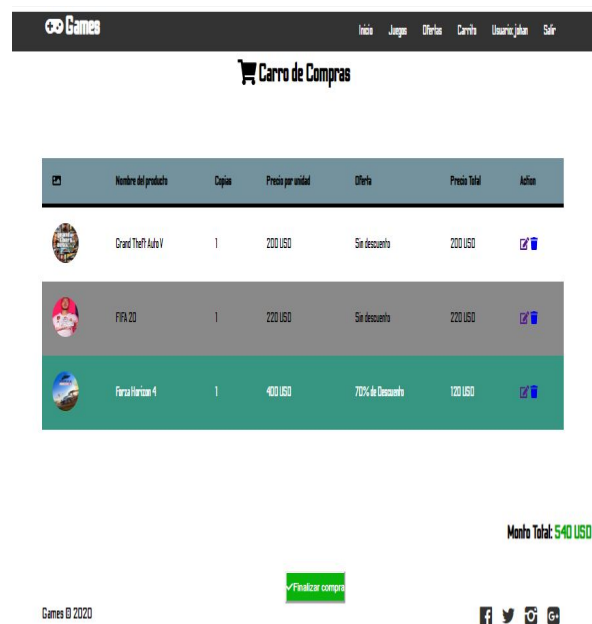
.monto{
  margin-right: 300px;
  text-align: left;
}

#main-container{
  margin: 100px auto;
  width: 1000px;
}

table{
  background: white;
  text-align: left;
  width: 100%;
  border-collapse: collapse;
}

th,td{
  padding: 20px;
}

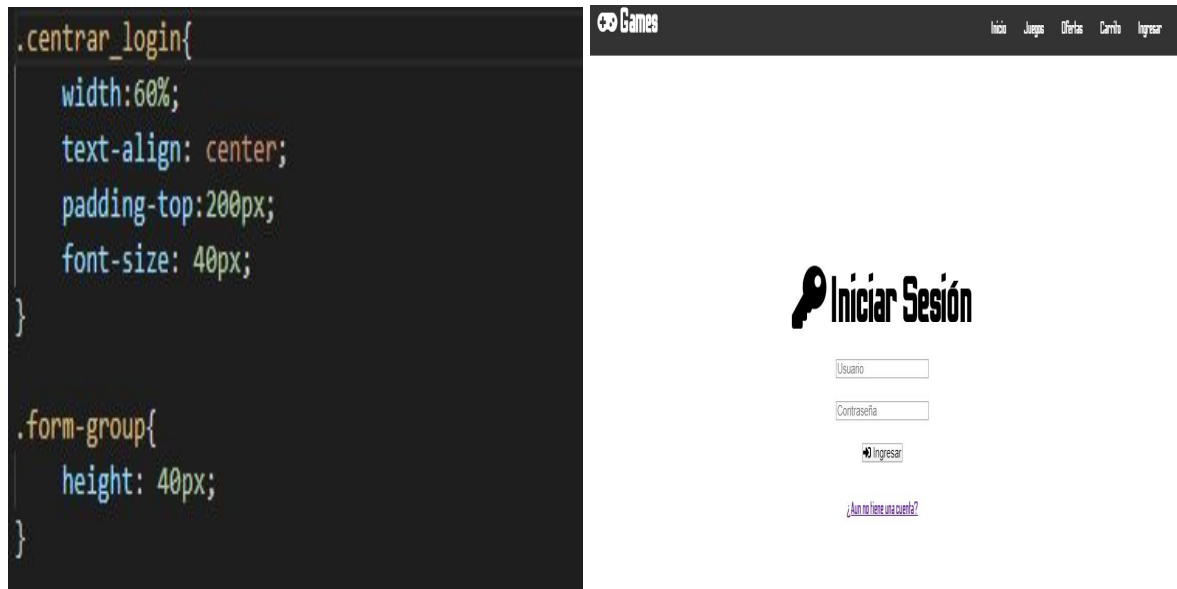
thead{
  background-color: #245365a2;
  border-bottom: solid 5px;
}
```



Dentro del código podemos ver cómo se construye el diseño de la tabla donde se van agregando los productos que se compran, mostrando las opciones para eliminar y agregar la

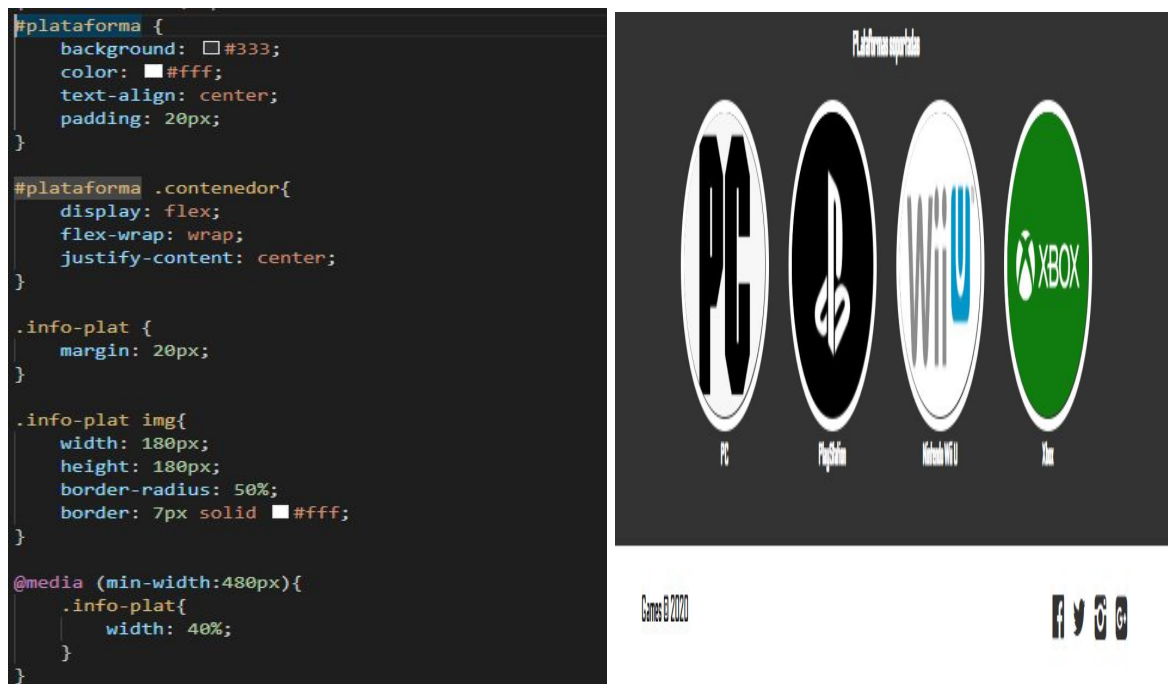
cantidad deseada del producto y también una pequeña animación al pasar el mouse sobre la tabla. Además de agregar un estilo al balance de la compra que se va realizar y del botón para realizar la compra.

Login.css



Siendo la parte más básica de la página frente a estilo , el código que vemos solo se encarga de la disposición del texto y tamaño.

Plataforma.css



Al igual que la anterior hoja, esta se encarga de ubicar y redimensionar las imágenes para que se vean de manera circular y el fondo de la sección donde está ubicadas sea de color negro agregando además un marco a la imagen. El @media es el encargado del

redimensionamiento del contenido respecto al tamaño de la página como se puede ver en esta y las anteriores imágenes.

IV. Funciones Php

Para realizar estas funciones nos apoyamos en los lenguajes de programación y php como se encuentran las siguientes imágenes con su respectivo funcionamiento.

```
function clear($var){
    htmlspecialchars($var);
    return $var;
}

function redir($var){
    <?>
    <script>
        window.location="<?=$var?>";
    </script>
    <?php
        die();
    >
}
```

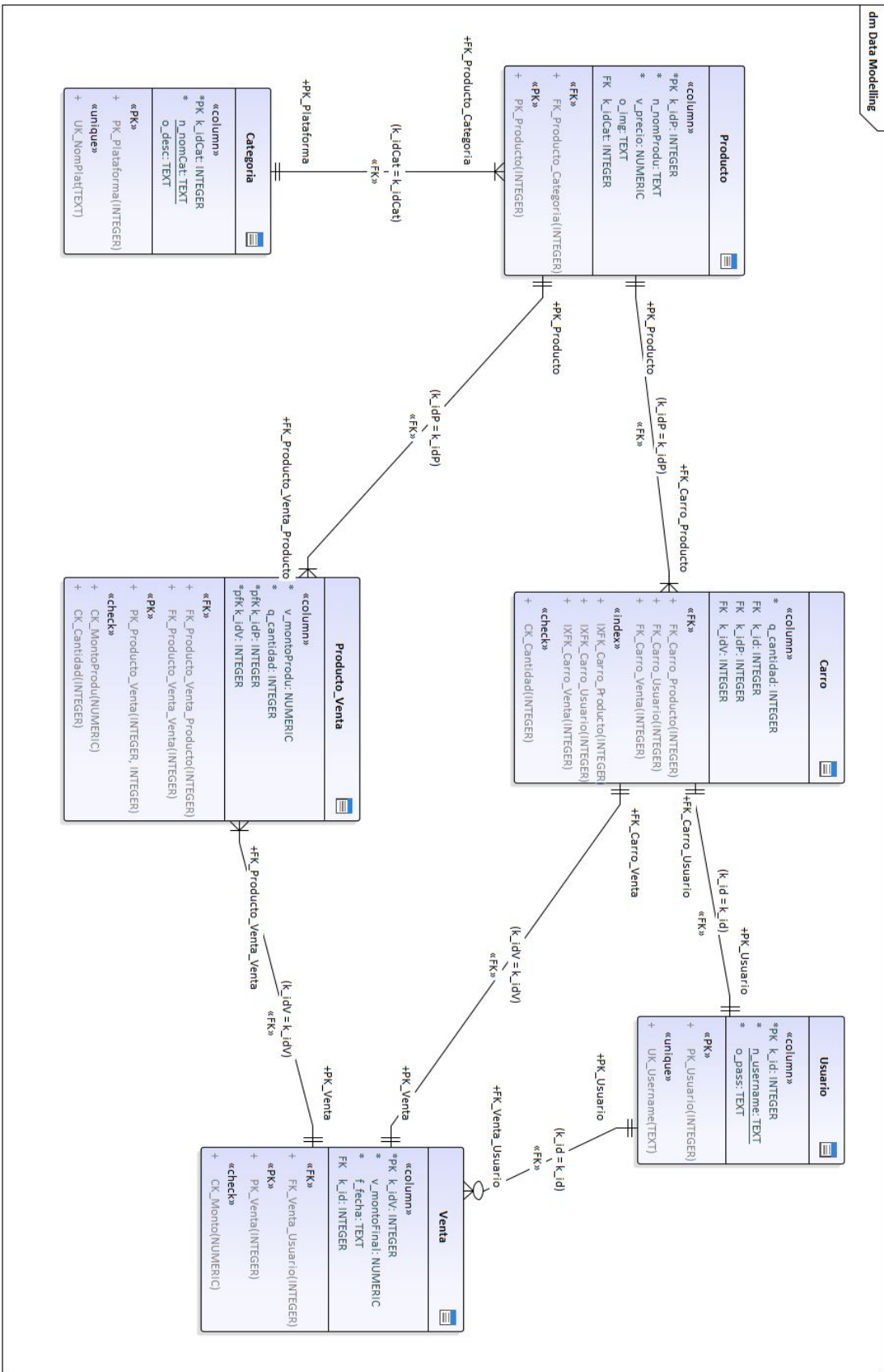
La primera función se encarga de cambiar los caracteres especiales para que el html sea capaz de interpretarlos.

La segunda redirecciona las páginas cuando se hace clic sobre algún apartado relacionado con el menú.

```
function alert($txt,$type,$url){
    if($type==0){
        $t = "error";
    }elseif($type==1){
        $t = "success";
    }elseif($type==2){
        $t = "info";
    }else{
        $t = "info";
    }
    echo '<script>swal({ title: "Alerta", text: "'. $txt. '", icon: "'. $t. '"});';
    echo '$(".swal-button").click(function(){ window.location="?p='.$url.'"; });';
    echo '</script>';
}
```

Esta función se encarga de mostrar un mensaje de “alerta” cuando se realiza una acción entre la página y el usuario como por ejemplo al iniciar sesión al realizar una compra etc.

V. Base de Datos



Interacciòn con la base de datos:

Para interactuar con la base de datos usamos SQLite3 y para gestionar la base de datos y crear las tablas con los respectivos atributos a guardar usamos DB Browser, el cual es compatible con SQLite.

Ejemplos de interacción con la base de datos:

1. Estas sentencias son las más importantes, debido a que con estas sentencias hacemos la conexión con la base de datos, así podremos trabajar sobre esta, guardar datos, modificar datos y eliminar datos.

```
k?php
    error_reporting(0);

    class MyDB extends SQLite3
    {
        function __construct()
        {
            $this->open('VentasDB.db');
        }
    }

    $db = new MyDB();
    $db->close();

?>
```

2. En este primer ejemplo, la base de datos interactúa de tal manera que recoge la información de un compra realizada.

```
("INSERT INTO Venta (k_id,f_fecha,v_montoFinal,i_estado) VALUES ('$id_cliente',date('now'),$monto,'0')");
("SELECT * FROM Venta WHERE k_id = '$id_cliente' ORDER BY k_idV DESC LIMIT 1");
```

3. En esta segunda sentencia SQL se modifica la cantidad de artículos que se encuentran en el carro de compras cuando se agrega o elimina un artículo.

```
$db->query("UPDATE Carro SET q_cantidad = '$modificar' WHERE k_idP = '$id'");
alert("Cantidad modificada",2,'carrito');
```

4. En la segunda sentencia SQL de esta imagen se accede a la información de un producto , la cual se pasará a la tabla venta:producto por medio de la primer sentencia SQL.

```
$res = $db->query("SELECT * FROM Producto_Venta WHERE k_idV = '". $r['k_idV']."'");
while($r2 = $res->fetchArray(SQLITE3_ASSOC)){
    $res2 = $db->query("SELECT * FROM Producto WHERE k_idP = '". $r2['k_idP']."'");
```

5. En esta imagen vemos dos sentencias SQL. En la primera, después de verificar la creación de un usuario y que todos los datos insertados en la página sean correctos,

estos datos se insertan en la tabla Usuario para así en un futuro verificar le inicia de sesión. En la segunda sentencia se verifica que el usuario haya sido creado

```
$db->query("INSERT INTO Usuario (n_username,o_pass) VALUES ('$username','$password')");  
  
$q2 = $db->query("SELECT * FROM Usuario WHERE n_username = '$username'");
```

NOTA: La entidad carro guarda de manera temporal los datos de la compra, luego, esta pasa los datos a la entidad producto_venta. En pocas palabras, la entidad carro funciona de manera temporal durante la ejecución de la venta o compra