



UCAM
UNIVERSIDAD
CATÓLICA DE MURCIA

PRÁCTICA FINAL

Programación Web

Oscar David Romero Humber - B05553852

odromerohumber@alu.ucam.edu

Ingeniería Informática

Departamento de Ciencias Politécnicas

UCAM

Introducción	4
Vídeo de funcionalidad	4
Front End	5
Navbar	5
Buscador	6
Footer	6
Contenido de páginas	7
Index	7
Nosotros	8
Preguntas frecuentes	8
Login	9
Back End	9
Productos	10
Producto	11
Carrito	12
MiCarrito	12

Introducción

La realización de este proyecto ha sido con las tecnologías básicas de la pila de la programación web, HTML para la estructura de las páginas, CSS para su presentación y estilos, Javascript para las funcionalidades del lado de cliente, XAMPP para lanzar un servidor web Apache y una base de datos MySQL, y PHP para recuperar los datos de la BBDD.

Estas tecnologías también han sido extendidas por diferentes frameworks, Bootstrap para CSS y JQuery para Javascript, facilitando así su uso al momento de desarrollar la aplicación web. Además de esto se han utilizado diferentes APIs para acceder a cierta información, específicamente Google Maps para utilizar la geolocalización de HTML 5, Google Fonts para utilizar una fuente diferente a las por defecto y Randomuser.me para generar imágenes genéricas de personas.

Combinando todo esto más unas librerías de JS y CSS, Fullcalendar.io, Fontawesome.com y CSS Neumophic_buttons, se ha realizado la aplicación web CSSkate.com.

Vídeo de funcionalidad

https://youtu.be/xzT4rQRFo_c

Front End

Todas las diferentes pantallas se presentan divididas en 4 secciones:

- Navbar
- Buscador + botón de login
- Contenido principal
- Footer

Principalmente se ha utilizado Bootstrap mezclado con modificaciones de CSS propias, principalmente para colores, tamaño y posición, pero la mayoría del CSS está basado en las clases de BS por su facilidad de implementación para márgenes y padding.

Navbar

```
<!-- Navbar -->
<nav class="container navbar navbar-expand-lg navbar-dark ">
  <a id="a-logo" class="navbar-brand" href="index.php">
    
  </a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
    aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <section class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav ms-auto">
      <li class="nav-item">
        <a class="nav-link" href="productos.php#skateboards">
          <h5>Skateboards</h5> </span>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="productos.php#longboards">
          <h5>Longboards</h5>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="faq.php">
          <h5>Preguntas frecuentes</h5>
        </a>
      </li>
    </ul>
  </section>
</nav>
```

Basada en los ejemplos de Bootstrap simplemente presenta el logo de la página principal con las diferentes pestañas a la derecha, para resoluciones más angostas se contrae a un menú colapsable.

Buscador

```
<!-- Buscador -->
<section class="container p-0 pt-2 pb-2">
  <nav id="buscador" class="navbar navbar-light bg-light justify-content-between">
    <form class="form-inline " method="POST" action="productos.php">
      <input id="buscador-text" name="buscador-text" class="form-control my-auto" type="search" placeholder="M
        aria-label="Search">
      <button id="buscador-boton" class="btn btn-outline-dark my-auto" type="submit">Buscar</button>
    </form>
  </nav>
  <?php
  if(!isset($_SESSION['cliente'])){
    ?>
    <a style="text-align: left; display: inline;" href="login.php"><img class="img-fluid " alt="Login" id="pe
  <?php }?>
  </nav>
</section>
```

Buscador basado en un tipo de navbar de bootstrap compuesto por un formulario que envía a la página de productos.php y una imagen que lleva a la página de login.php en caso de no haber iniciado sesión.

Footer

```
<!-- Footer-->
<footer class="container bg-dark">
  <section class="row">
    <section class="col-2">
      <canvas class="img-fluid" id="canvasarea" width="150" height="150"></canvas>
    </section>
    <!--  -->
    <address class="col-6">
      <strong>CSS, Inc.</strong><br>
      Calle Francisco de Asis, 1<br>
      Murcia, ES 30009<br>
      <abbr title="Telefono">T:</abbr> (+34) 658-645-546
    </address>
  </section>
  <section class="row">
    <span class="col-11"></span>
    <address class="col-11">
      <strong>Oscar Romero</strong><br>
      <a href="mailto:#">odromerohumber@alu.ucam.edu</a>
    </address>
  </section>
</footer>
<!-- Bootstrap JS Bundle-->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
  crossorigin="anonymous"></script>
```

El footer contiene el logo realizado con Canvas, una dirección y un método de contacto, además de el script de bootstrap.

Contenido de páginas

Index

```
<main class="container p-0">
  <!-- Contenido del inicio -->
  <section id="contenido-inicio">
    <section class="row">
      <section class="col-md-12"><a href="productos.php"><
      <section class="col-md-12 p-0 jumbotron text-center bg-light bg-opacity-75">

        <a id="enlace-jumbo" href="productos.php">
          <h1 class="display-3">Ofertas Semana Santa</h1>
        </a>
        <p class="lead">Dile hola a las vacaciones con un nuevo Skate.</p>

      </section>
      <section class="col-md-12 col-lg-7"><a href="productos.php">
      <section class="col-md-6 col-lg-5"><a href="productos.php">
      <section class="col-md-6 col-lg-5"><a href="productos.php">
      <section class="col-md-12 col-lg-7"><a href="productos.php">
      <section class="col-md-12"><a href="productos.php">

    </section>
  </section>
  <!-- Newsletter -->
  <section id="newsletter" class=" text-light p-5">
    <section class="container">
      <section class="d-md-flex justify-content-between align-items-center">
        <h3 class="mb-3 mb-md-0 p-1">¡Suscríbete a nuestra lista para recibir promociones!</h3>

        <section class="input-group news-input">
          <input type="text" class="form-control" placeholder="Correo electronico" aria-label="Correo del recep">
          <button class="btn btn-dark btn-lg" type="button">Enviar</button>
        </section>
      </section>
    </section>
  </section>
</main>
```

Para el index se ha realizado un juego con imágenes y otros elementos de BS, ubicando cada imagen a la izquierda o derecha para que en navegadores más estrechos se reubiquen y se muestren mejor, así notándose la capacidad responsive de la página. También incluye una sección de newsletter para que un usuario se pueda suscribir a nuestra lista de promociones.

Nosotros

```
<main class="container p-0">
  <section id="equipo" class="p-5 bg-gradient">
    <section class="container">
      <h2 class="text-center text-white">¡Conoce a nuestro equipo!</h2>
      <p class="lead text-center text-white mb-5">
        Todo nuestro equipo tiene 5+ años de experiencia sobre la tabla y detrás del mostrador
      </p>
      <section class="row g-4">
        <section class="col-md-6 col-lg-3">
          <section class="card bg-light">
            <section class="card-body text-center">
              
              <h3 class="card-title mb-3">Jose Ramiro</h3>
              <p class="card-text">
                Skate or die! Skater desde los 8 años, de Murcia centro y me encanta ayudar a los c
              </p>
            </section>
          </section>
        </section>
      </section>
    </section>
  </main>
```

Para la página de nosotros mostramos 4 tarjetas de BS diferentes para 4 empleados ficticios con su respectiva fotografía, nombre y descripción. También se incluye un formulario básico para dudas o comentarios de los clientes.

Preguntas frecuentes

```
<!-- Question Accordion -->
<section id="questions" class="p-5">
  <section class="container">
    <h2 class="text-center mb-4">Preguntas frecuentes</h2>
    <section class="accordion accordion-flush" id="questions">
      <!-- Item 1 -->
      <section class="accordion-item rounded-3">...
      </section>
      <!-- Item 2 -->
      <section class="accordion-item rounded-3">...
      </section>
      <!-- Item 3 -->
      <section class="accordion-item rounded-3">...
      </section>
      <!-- Item 4 -->
      <section class="accordion-item rounded-3">...
      </section>
      <!-- Item 5 -->
      <section class="accordion-item rounded-3">...
      </section>
      <!-- Item 6 -->
      <section class="accordion-item rounded-3">
        <h2 class="accordion-header">
          <button class="accordion-button collapsed mx-0 rounded-3" type="button" data-bs-toggle="collapse" data-bs-target="#question-six">
            ¿Dónde está vuestra tienda?
          </button>
        </h2>
        <section id="question-six" class="accordion-collapse collapse" data-bs-parent="#questions">
          <section class="accordion-body">
            <label for="geoloc" class="form-label">No tenemos tienda física, solo online, así que es donde tú te encuentres (; </label>
            <section id="dvMap" style="width: 500px; height: 500px">
            </section>
          </section>
        </section>
      </section>
    </section>
    <!-- Item 7 -->
    <section class="accordion-item rounded-3">
      <h2 class="accordion-header">
        <button class="accordion-button collapsed mx-0 rounded-3" type="button" data-bs-toggle="collapse" data-bs-target="#question-seven">
          Enlaces de interes
        </button>
      </h2>
    </section>
  </section>
</section>
```


Las preguntas frecuentes están compuestas por un acordeón, dentro de estas está el uso de la geolocalización y los enlaces de interés. Al final de la página también se encuentra un calendario de eventos.

Login

```
<!-- Login -->

<section id="login" class="row">
  <form class="form-signin p-5 col-6 mx-auto py-4 bg-light bg-opacity-50 my-5 rounded-3" method="POST" action="login_accion.php">
    <h2 class="form-signin-heading">Introduce tus credenciales</h2>
    <br>
    <label for="correo" class="sr-only">Correo Electrónico</label>
    <input type="text" id="correo" name="inputUsuario" class="form-control" placeholder="Introduce tu correo" required autofocus>
    <br>
    <label for="pass" class="sr-only">Contraseña</label>
    <input type="password" id="pass" name="inputPassword" class="form-control" placeholder="Contraseña" required>
    <br>
    <button class="custom-btn btn-4 m-0" type="submit">Iniciar sesión</button>
  </form>
</section>
```

Se utiliza un formulario que recibe correo electrónico y contraseña y es autenticado en el login_accion.php. En este formulario se puede apreciar mi mayor manera de programar CSS a través de BS con las clases de los elementos HTML, modificando así los colores, formas y posiciones.

Las páginas de Producto, Productos, Carrito y MiCuenta son explicadas en el siguiente apartado porque la mayoría del HTML es generado por PHP.

Back End

Aprovechando la modularidad de PHP todas las páginas solamente incluyen el head, navbar, buscador y footer, así centrándose solamente en el contenido propio de cada una.

```
<head>
  <?php include "head.php"; ?>
</head>

<body onload="setcanvas()">

  <?php include "navbar.php" ?>
  <?php include "buscador.php" ?>

  <main class="container p-0">...
</main>
  <?php include "footer.php" ?>

</body>

</html>
```

Productos

```
<!-- Contenido de los productos -->

<section id="instructors" class="p-5 ">
  <section class="container">
    <h2 id="skateboards" class="text-center text-white">¡Nuestros skateboards!</h2>
    <p class="lead text-center text-white mb-5">
      Todos nuestros skateboards son hechos con 100% madera de roble
    </p>
    <section class="row g-4">

      <?php
      if (isset($_POST["buscador-text"])) {
        buscarProductos($_POST["buscador-text"]);
      } else {
        getAllSkates();
      }
      ?>

      <h2 id="longboards" class="text-center text-white mt-5">¡Nuestros longboards!</h2>
      <p class="lead text-center text-white mb-5">
        Todos nuestros longboards son hechos con 100% madera de arce
      </p>
      <section class="row g-4">
        <?php
        getAllLongs();
        ?>
      </section>
    </section>
  </section>
  <?php
  ?>
</pre>
```

La página de productos sirve como página de búsqueda también, llamando al método php respectivo en caso de recibir una petición post con 'buscador-text' o no, de cualquier manera mostrando tarjetas de los distintos productos.

El método de buscarProductos realiza una consulta SQL donde compara el nombre y la descripción del producto con el texto para generar tarjetas de productos, igual que los métodos getAllSkates y getAllLongs para el caso de no haber realizado una búsqueda pero comparando el atributo longboard de la tabla.

```
function getAllLongs()
{
  include 'conexion.php';
  $sentencia = "SELECT * FROM producto WHERE longboard=1";
  $resultado = $conexion->query($sentencia) or die("Error al comprobar producto: " . mysqli_error($conexion));

  $count = mysqli_num_rows($resultado);

  if ($count > 0)
  {
    while ($prod = mysqli_fetch_object($resultado)) {
      echo '
      <section class="col-md-6 col-lg-3">
      <section class="card bg-light">
        <section class="card-body text-center">
          nombre . '"
          />
        </section>
      </section>
      </section>
      ';
```

Producto

Para generar la página de cada producto se recibe el id del producto por una petición get con idProd para realizar una consulta a la base de datos y rellenar los campos

```
function getProducto($idProd)
{
    include 'conexion.php';
    $sentencia = "SELECT * FROM producto where id_producto=" . $idProd;
    $resultado = $conexion->query($sentencia) or die("Error al comprobar producto: " . mysqli_error($conexion));

    $count = mysqli_num_rows($resultado);

    if ($count > 0)
    {
        $prod = mysqli_fetch_object($resultado);

        echo '
            nombre . '" class="col-lg-5 img-thumbnail">
            <section id="info-producto" class="col-lg-7">
                <h1 id="nombre" class="mt-5 ms-3 text-black">' . $prod->nombre . '</h3>
                <p id="precio" class="ms-3"><b>' . $prod->precio . '€</b></p>
                <h3 class="ms-3">Detalles del producto</h3>
                <p id="detalles" class="mx-3 text-black opacity-75">
                    ' . $prod->detalles . '
                </p>
            </section>
        '
```

respectivamente.

También se utiliza otro método php para recibir productos relacionados al buscado, esto realmente solo realiza una consulta sql aleatoria para generar tres tarjetas de productos y mostrarlas al lado del producto principal.

```
function getRelacionados()
{
    include 'conexion.php';

    $sentencia = "SELECT * FROM producto ORDER BY RAND() LIMIT 3";
    $resultado = $conexion->query($sentencia) or die("Error al comprobar producto: " . mysqli_error($conexion));

    $count = mysqli_num_rows($resultado);

    if ($count > 0)
    {
        while ($prod = mysqli_fetch_object($resultado)) {
            echo '
                <section class="col-lg-4">
                    <section class="card bg-light">
                        <section class="card-body text-center">
                            nombre . '">
                            <a href="producto.php?id=' . $prod->id_producto . '">
                                <h3 class="card-title mb-3">' . $prod->nombre . '</h3>
                            </a>
                        </section>
                    </section>
                </section>
            '
```

Carrito

Al añadir un producto al carrito se crea o recupera el carrito de la sesión, este es un map que utiliza el id de producto como clave y la cantidad de productos comprados el valor, llevando luego a la página de miCarrito.php.

```
session_start();
$idProd = $_GET['id'];
echo '<script>';

if (isset($_SESSION['carrito'])) {

    $carrito = $_SESSION['carrito'];
    if(array_key_exists($idProd, $carrito)){
        $cant = $carrito[$idProd];
        $carrito[$idProd] = $cant + 1;

        echo 'alert("Producto ya en el carrito");';

    }else{
        $carrito[$idProd] = 1;
        echo 'alert("Producto añadido al carrito!");';
    }
    $_SESSION['carrito'] = $carrito;
}else{
    $carrito[$idProd] = 1;
    $_SESSION['carrito'] = $carrito;
    echo 'alert("Producto añadido al carrito!");';
}
```

MiCarrito

Se muestra una tabla con los productos dentro del carrito, con un botón para comprarlo y otro para borrarlo.

```
<main class="container p-0">
<section id="miCarrito" class="p-5 bg-gradient">
  <h2 class="text-center text-white">El carrito de <?php if(isset($_SESSION['cliente'])) { echo $_SESSION['cliente']->nombre;}else{ echo 'visi
  <p class="lead text-center text-white mb-5">
    Este es el detalle de tu carrito:
  </p>

  <?php if(isset($_SESSION['carrito']) && !isset($_GET['borrar']))){
    getCarrito();
  } else{
    borrarCarrito();
  }
  ?>

  <a id="comprarCarrito" class="custom-btn text-center btn-14 w-auto" style="display: block;" href="comprarCarrito.php">Comprar carrito</a>
  <br>
  <a id="borrarCarrito" class="custom-btn text-center btn-15 w-auto" style="display: block;" href="miCarrito.php?borrar=1">Borrar carrito</a>
```

El método de `getCarrito` recupera los ids del carrito, los recorre para realizar una consulta SQL e inserta en la tabla una fila con los datos de cada uno, calculando un total al final de la tabla.

```
function getCarrito()
{
    include 'conexion.php';

    $carrito = $_SESSION['carrito'];
    $total = 0;

    echo '
    <br>
    <table class="w-100 table table-striped table-dark">
    <thead class="thead-dark">
    <tr>
    <th scope="col">Nombre</th>
    <th scope="col">Detalles</th>
    <th scope="col">Cantidad</th>
    <th scope="col">Precio</th>
    </tr>
    </thead>
    <tbody>';

    foreach ($carrito as $idprod => $cant) {
        $sentencia = "SELECT nombre, detalles, precio FROM producto WHERE id_producto= " . $idprod;
        $resultado = $conexion->query($sentencia) or die("Error al comprobar producto: " . mysqli_error($conexion));
        $prod = mysqli_fetch_object($resultado);
        echo '
        <tr>
        <td scope="row">' . $prod->nombre . '</td>
        <td>' . $prod->detalles . '</td>
    '
```

El método de comprar carrito verifica que exista un usuario y un carrito en sesión, recorre el carrito y va insertando en la tabla de compra el id de usuario, id de producto, cantidad y el id de compra, la fecha se inserta automáticamente por la base de datos. El método `comprarYa` realiza la misma acción solamente tomando el id del producto por `get` sin utilizar el carrito.

```
function comprarCarrito($carrito, $idUser)
{
    include 'conexion.php';

    $sentencia = "SELECT MAX(`id_compra`) as `max` FROM `compra`";
    $resultado = $conexion->query($sentencia) or die("Error al comprobar productos: " . mysqli_error($conexion));
    $max = mysqli_fetch_object($resultado)->max;

    foreach ($carrito as $idprod => $cant) {
        $sentencia = "INSERT INTO `compra`(`id_compra`, `id_cliente`, `id_producto`, `cantidad`) VALUES ('" . $max + 1 . "
        " . $idUser . ", " . $idprod . ", " . $cant . ")";

        if ($conexion->query($sentencia)) {
            $sentencia = "UPDATE `producto` SET `stock`=`stock` - " . $cant . " WHERE `id_producto` = " . $idprod;
            $conexion->query($sentencia);
        }
    }

    unset($_SESSION['carrito']);
}
```

MiCuenta

La página de miCuenta muestra una tabla con todas las compras realizadas por el usuario en sesión, el formato es igual al de la tabla de mi carrito pero en esta página se generan varias tablas para cada compra. Se realiza una consulta join de compra con producto para recuperar los datos del producto a imprimir en cada fila y de la fecha y cantidades de cada compra.

```
function getCompras($id_cliente)
{
    include 'conexion.php';
    $sentencia = "SELECT id_compra, nombre, detalles, precio, cantidad, fecha FROM `compra` INNER JOIN producto ON compra.id_producto = producto.id_producto WHERE compra.id_cliente = $id_cliente";
    $resultado = $conexion->query($sentencia) or die("Error al comprobar usuario: " . mysqli_error($conexion));

    $count = mysqli_num_rows($resultado); //Numero de filas del resultado de la consulta
    $prevId = -1;
    $total = 0;
    if ($count > 0) //si la variable count es mayor a 0
    {
        while ($compras = mysqli_fetch_object($resultado)) {
            if($prevId != -1 && $prevId != $compras->id_compra){
                echo '
                <tr class="table-info">
                <td>Total</td>
                <td></td>
                <td></td>
                <td>'.$total.'€</td>
                </tr>
                </tbody>
                </table>';
            }

            if ($prevId != $compras->id_compra) {
                $prevId = $compras->id_compra;
                $total = 0;
                echo '
                <br>
                <p class="lead text-white">
                | Fecha: '.$compras->fecha.'
                </p>
                <table class="w-100 table table-striped table-dark">
                <thead>
                <tr>
                <th>Nombre</th>
                <th>Detalles</th>
                <th>Cantidad</th>
                <th>Precio</th>
                </tr>
                </thead>
                <tbody>
                <tr>
                <td>'.$compras->nombre.'</td>
                <td>'.$compras->detalles.'</td>
                <td>'.$compras->cantidad.'</td>
                <td>'.$compras->precio.'</td>
                </tr>
                </tbody>
                </table>';
            }
        }
    }
}
```

Además de ver las compras se puede salir de sesión, limpiando el objeto cliente de la sesión y retornando al index.