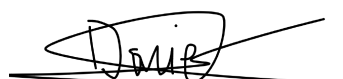


PORTADA

Nombre Alumno / DNI	Daniel Rodríguez García
Título del Programa	BD Computer Science & Artificial Intelligence and Software Engineering
Nº Unidad y Título	UNIT 1-PROGRAMMING & CODING
Año académico	2023/2024
Profesor de la unidad	Gabriela García
Título del Assignment	AB final
Día de emisión	20/01/2024
Día de entrega	23/01/2024
Nombre IV y fecha	
Declaración del estudiante	<p>Certifico que la presentación del assignment es completamente mi propio trabajo y entiendo completamente las consecuencias del plagio. Entiendo que hacer una declaración falsa es una forma de mala práctica.</p> <p>Fecha: 23/01/2024</p> <p>Firma del alumno: </p>

Plagio

El plagio es una forma particular de hacer trampa. El plagio debe evitarse a toda costa y los alumnos que infrinjan las reglas, aunque sea inocentemente, pueden ser sancionados. Es su responsabilidad asegurarse de comprender las prácticas de referencia correctas. Como alumno de nivel universitario, se espera que utilice las referencias adecuadas en todo momento y mantenga notas cuidadosamente detalladas de todas sus fuentes de materiales para el material que ha utilizado en su trabajo, incluido cualquier material descargado de Internet. Consulte al profesor de la unidad correspondiente o al tutor del curso si necesita más consejos.

Documentación AB final programación y bases de datos

La página web es una tienda online de videojuegos que también tiene dos videojuegos incluidos que son el Snake y el Simón dice. También tiene un foro para que nuestros clientes puedan charlar.

Index.php

Header

```
<header>
<nav class="navbar navbar-expand-lg navbar-light bg-body-tertiary">
  <div style="background-color: #FF7000;" class="container-fluid">
    <a class="navbar-brand" href="index.php">
      
    </a>
    <button data-mdb-collapse-init class="navbar-toggler" type="button" data-mdb-target="#navbarText" aria-controls="navbarText" aria-expanded="false" aria-label="Toggle navigation">
      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarText">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="pages/foro.php">Foro</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="Snake/index.html">Snake</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="SimonDice/index.html">Simon dice</a>
        </li>
      </ul>
      <a class="navbar-brand" href="pages/iniciosesion.php">
        
      </a>
    </div>
  </div>
</nav>
</header>
```

El header está creado a través del framework Bootstrap y el resultado final sería este.



Tienda

La tienda está creada llamando a una función de PHP en el index. La función a la que llama sería esta:

```

function mostrarTienda(){
    $servername = "localhost";
    $username = "root";
    $password = "9455";
    $dbname = "abfinal";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT nombre,imagen,precio,id FROM productos";
    $result = $conn->query($sql);

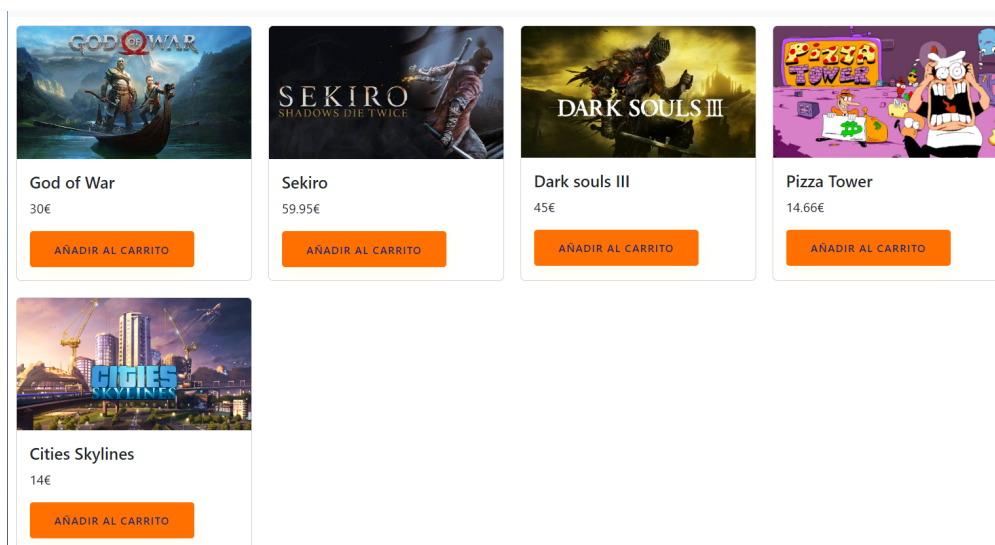
    if ($result->num_rows > 0) {
        echo "<section style='display: flex; flex-wrap: wrap;'>";
        while ($row = $result->fetch_assoc()) {
            echo '
            <div class="card" style="width: 18rem; margin: 10px;">
                
                <div class="card-body">
                    <h5 class="card-title">' . $row["nombre"] . '</h5>
                    <p class="card-text">' . $row["precio"] . '</p>
                    <button style="background-color:#FF7000" class="btn btn-primary" onclick="agregarAlCarrito(\'' . $row["nombre"] . '\', \'' . $row["precio"] . '\')">Añadir al carrito</button>
                </div>
            </div>';
        }
        echo "</section>";
    } else {
        echo "No hay usuarios registrados en este momento";
    }
    $conn->close();
}

```

La función se conecta a MySQL a través de msqli y realiza una consulta select para sacar la información de los productos.

Una vez recoge los datos los muestra dentro de una card a creada a través de bootstrap.

El resultado final en el index seria tal que así:



Carrito

```

<section id="carrito" class="text-center mx-auto my-5">
    <h2>Carrito</h2>
    <div id="itemsCarrito"></div>
    <div id="total"></div>
    <button class="btn btn-danger mt-3" onclick="vaciarCarrito()" style="background-color:#FF7000">Vaciar Carrito</button>
</section>

```

Para utilizar el carrito pulsamos los botones de añadir al carrito que tiene cada producto que llamara a la función agregarAlCarrito()

```
function agregarAlCarrito(producto, precio) {  
  if (carrito[producto]) {  
    carrito[producto].cantidad++;  
  } else {  
    carrito[producto] = { precio, cantidad: 1 };  
  }  
  actualizarCarrito();  
}
```

La cual añadirá un producto al array carrito.

La función actualizarCarrito() mostrara la información que hay dentro del array con su nombre, precio, cantidad y total en euros.

```
function actualizarCarrito() {  
  const itemsCarrito = document.getElementById('itemsCarrito');  
  itemsCarrito.innerHTML = '';  
  
  let total = 0;  
  
  Object.keys(carrito).forEach(producto => {  
    const div = document.createElement('div');  
    div.innerHTML = `${producto} - Cantidad: ${carrito[producto].cantidad} <button onclick="quitarDelCarrito('${producto}')">Quitar uno</button>`;  
    itemsCarrito.appendChild(div);  
    total += carrito[producto].precio*carrito[producto].cantidad;  
  });  
  
  document.getElementById("total").innerHTML = `Total: ${total}€`;  
}
```

El carrito también tiene un botón que vacía el carrito llamando a la función vaciarCarrito() y un botón que quita elementos del carrito llamando a la función quitarDelCarrito()

```
function vaciarCarrito() {
  carrito = {};
  actualizarCarrito();
}

function quitarDelCarrito(producto) {
  if (carrito[producto] && carrito[producto].cantidad > 0) {
    carrito[producto].cantidad--;
    if (carrito[producto].cantidad === 0) {
      delete carrito[producto];
    }
    actualizarCarrito();
  }
}
```

El resultado final seria:

Carrito

Sekiro - Cantidad: 1

Dark souls III - Cantidad: 1

Pizza Tower - Cantidad: 1

Total: 119.61€

VACIAR CARRITO

Footer

El footer esta creado a traves del framework de bootstrap.

```

<footer class="text-center text-lg-start text-white" style="background-color: #FF7000">
  <section class="d-flex justify-content-between p-4" style="background-color: #ff9900">
    <div class="me-5">
      <span>Redes Sociales:</span>
    </div>
    <div>
      <a href="" class="text-white me-4">
        <i class="fab fa-facebook-f"></i>
      </a>
      <a href="" class="text-white me-4">
        <i class="fab fa-twitter"></i>
      </a>
      <a href="" class="text-white me-4">
        <i class="fab fa-google"></i>
      </a>
      <a href="" class="text-white me-4">
        <i class="fab fa-instagram"></i>
      </a>
      <a href="" class="text-white me-4">
        <i class="fab fa-linkedin"></i>
      </a>
      <a href="" class="text-white me-4">
        <i class="fab fa-github"></i>
      </a>
    </div>
  </section>
  <section>
    <div class="container text-center text-md-start mt-5">
      <div class="row mt-3">
        <div class="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">
          <h6 class="text-uppercase fw-bold"><a style="color: white;" href="">Quienes somos</a></h6>
        </div>
        <div class="col-md-2 col-lg-2 col-xl-2 mx-auto mb-4">
          <h6 class="text-uppercase fw-bold"><a style="color: white;" href="">Contacto</a></h6>
        </div>
        <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">
          <h6 class="text-uppercase fw-bold"><a style="color: white;" href="">Política privacidad</a></h6>
        </div>
        <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">
          <h6 class="text-uppercase fw-bold"><a style="color: white;" href="">Política cookies</a></h6>
        </div>
      </div>
    </div>
  </section>
  <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2)">
    © 2023 Copyright:
    <a class="text-white" href="index.php">JuegaConGalleta</a>
  </div>
</footer>

```

El resultado final seria:



InicioSesion.php

Esta página esta dividida en dos, la parte para iniciar sesión y registrarse y por otra parte la sección de mi cuenta que se muestra solo si has iniciado sesión en la pagina web.

Lo primero que hace la pagina es revisar si tienes la cookie id para verificar si has iniciado sesión o no.

```
<script>
if (!isset($_COOKIE['id'])) {
```

Inicio sesión/Registro

El formulario de inicio/registro esta creado a través de bootstrap y es un formulario dinámico que cambia entre inicio de sesión y registro.

```
<div class="section">
  <div class="container">
    <div class="row full-height justify-content-center">
      <div class="col-12 text-center align-self-center py-5">
        <div class="section pb-5 pt-5 pt-sm-2 text-center">
          <h6 class="mb-0 pb-3">
            <span>Inicio Sesión </span><span>Registro</span>
          </h6>
          <input class="checkbox" type="checkbox" id="reg-log" name="reg-log" /> <label for="reg-log"></label>
          <div class="card-3d-wrap mx-auto">
            <div class="card-3d-wrapper">
              <div class="card-front">
                <div class="center-wrap">
                  <div class="section text-center">
                    <h4 class="mb-4 pb-3">Inicio Sesión</h4>
                    <div class="form-group">
                      <input type="text" name="correo" class="form-style" placeholder="Tu correo electronico" id="correo" autocomplete="off">
                    </div>
                    <div class="form-group mt-2">
                      <input type="password" name="contrasenaInicio" class="form-style" placeholder="Tu contraseña" id="contrasenaInicio" autocomplete="off">
                    </div>
                    <button id="botoninicio" class="btn mt-4" style='background-color: #333; color: white;'>Iniciar sesión</button>
                    <p id="respuestaInicio"></p>
                  </div>
                </div>
              </div>
              <div class="card-back">
                <div class="center-wrap">
                  <div class="section text-center">
                    <h4 class="mb-4 pb-3">Registro</h4>
                    <div class="form-group mt-2">
                      <input type="email" name="correoRegistro" class="form-style" placeholder="Tu correo electronico" id="correoRegistro" autocomplete="off">
                    </div>
                    <div class="form-group mt-2">
                      <input type="password" name="contrasenaRegistro" class="form-style" placeholder="Tu contraseña" id="contrasenaRegistro" autocomplete="off">
                    </div>
                    <button id="botonregistro" class="btn mt-4" style='background-color: #333; color: white;'>Registrarse</button>
                    <p id="respuestaRegistro"></p>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

El resultado de este código sería este:

INICIO SESIÓN

REGISTRO



Inicio Sesión

Tu correo electronico

Tu contraseña

INICIAR SESIÓN

INICIO SESIÓN

REGISTRO



Registro

Tu correo electronico

Tu contraseña

REGISTRARSE

```
$(document).ready(function() {  
    $("#botonregistro").click(verificarRegistro);  
    $("#botoninicio").click(verificarInicio);  
    $("#borrarcookie").click(borrarcookie);  
})
```

La función de jQuery espera a que se pulse el botón de iniciar sesión para llamar a la función de JavaScript verificarInicio()

```

function verificarInicio() {
    let correo1 = document.getElementById("correo").value;
    let contra1 = document.getElementById("contrasenaInicio").value;
    let correoIni = correo1.trim();
    let contraIni = contra1.trim();
    let errores = [];
    if (validarcorreo(correoIni) == false) {
        errores.push("El correo no esta en un formato correcto \n");
    }
    if (errores.length == 0) {
        let peticion = $.ajax({
            url: "../funciones/funciones.php",
            type: "POST",
            data: {
                correoIni: correoIni,
                contraIni: contraIni
            },
            async: true,
            success: function(data) {
                if(data=="false"){
                    $("#respuestainicio").html("El correo o la contraseña son incorrectos");
                }else{
                    setCookie("id",data,1);
                    location.href = '../index.php';
                }
            }, error: function(data) {
                alert("Error en la trasmision");
            }
        });
    } else {
        alert(errores);
    }
}

```

La función de verificar el inicio recoge los datos del formulario, y llama a la función validarcorreo() para validar que cumple los requisitos de un correo electrónico

```

function validarcorreo(dato) {
    let verdadero = true;
    let ex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
    if (!ex.test(dato)) {
        verdadero = false;
    }
    return verdadero;
}

```

Cuando la respuesta es true manda a través de Ajax la información de correo y contraseña al php para realizar la consulta en la base de datos.

```

if (isset($_POST['correoIni']) && isset($_POST['contraIni'])) {
    $correo = $_POST['correoIni'];
    $contra = $_POST['contraIni'];
    $servername = "localhost";
    $username = "root";
    $dbname = "abfinal";
    $password = "9455";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT id FROM usuarios WHERE correo = '$correo' AND contrasenia = '$contra'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // output data of each row
        while ($row = $result->fetch_assoc()) {
            echo $row['id'];
        }
    } else {
        echo "false";
    }
}

```

El php recoge los datos enviados y realiza la consulta a la base de datos, si el resultado de la consulta es mayor que 0 por lo tanto existe un usuario con esa contraseña y ese correo recoge el id y lo envía de nuevo a la respuesta de Ajax.

Si la respuesta es false escribe en el formulario que no hay ningún usuario que coincida con esa contraseña.

Si la respuesta es diferente de false, en este caso el id, le crea una cookie con nombre id y el valor de su id para verificar que el usuario a iniciado sesión, y lo manda al index.php.

Para el formulario de registro funciona bastante parecido, el jQuery llama a la función verificarRegistro()

```

function verificarRegistro() {
    let correo1 = document.getElementById("correoRegistro").value;
    let contra1 = document.getElementById("contrasenaRegistro").value;
    let correo = correo1.trim();
    let contra = contra1.trim();

    let errores = [];
    if (validarcorreo(correo) == false) {
        errores.push("El correo no esta en un formato correcto \n");
    }
    if (validarcontra(contra) == false) {
        errores.push("La contraseña tiene que tener 8 caracteres, una mayuscula, una minuscula, un numero y un simbolo(!@#$$%^&*) \n");
    }
    if (errores.length == 0) {
        let peticion = $.ajax({
            url: "../funciones/funciones.php",
            type: "POST",
            data: {
                correo: correo,
                contra: contra
            },
            async: true,
            success: function(data) {
                if(data=="false"){
                    $("#respuestaregistro").html("Ya existe una cuenta con ese correo electronico");
                }else{
                    $("#respuestaregistro").html("La cuenta se ha creado correctamente");
                }
            }, error: function(data) {
                alert("Error en la trasmision");
            }
        });
    } else {
        alert(errores);
    }
}

```

Recoge los datos del formulario, llama a la función validarcorreo() que he mostrado antes y a la función validarcontra() para validar la contraseña

```

function validarcontra(dato) {
    let verdadero = true;
    let ex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$$%^&*])[A-Za-z\d!@#$$%^&*]{8,}$/;
    if (!ex.test(dato)) {
        verdadero = false;
    }
    return verdadero;
}

```

Si las validaciones son correctas manda a través de Ajax la información al php, si no es correcta muestra por pantalla que la contraseña o el correo no son correctos.

```

if (isset($_POST['correo']) && isset($_POST['contra'])) {
    $correo = $_POST['correo'];
    $contra = $_POST['contra'];
    $servername = "localhost";
    $username = "root";
    $dbname = "abfinal";
    $password = "9455";

    // Crear la conexión
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Verificar la conexión
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT id FROM usuarios WHERE correo = '$correo'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        echo "false";
    } else {
        $sql = "INSERT INTO usuarios (correo, contrasenia) VALUES ('$correo', '$contra')";
        if ($conn->query($sql) === TRUE) {
            echo "true";
        } else {
            echo "false";
        }
    }
}
}

```

El php recoge los datos enviado por Ajax y hace una consulta para ver si ya existe un usuario con ese correo electrónico, si la respuesta es mayor que cero envía false, si no, inserta los datos a la base de datos y envía de respuesta true.

Si la respuesta es false se muestra por pantalla que ya existe un usuario con ese usuario y si es correcta muestra por pantalla que se ha creado la cuenta correctamente

Si el usuario tiene sesión iniciada en lugar de mostrarle los formularios le muestra un panel de mi cuenta. En el caso que la id de usuario sea 1 mostrara el panel de administrador, que llama a la función mostrarUsu()

```

if ($_COOKIE['id'] == 1) {
    include("../funciones/funciones.php");
    mostrarUsu();
}

```

```

function mostrarUsu()
{
    $servername = "localhost";
    $username = "root";
    $password = "9455";
    $dbname = "abfinal";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT id,correo FROM usuarios";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // output data of each row
        echo "<table style='margin-top:3%;>";
        while ($row = $result->fetch_assoc()) {
            $id = $row['id'];
            echo "<tr style='border:solid 2px orange;>";
            echo "<td style='border:solid 2px orange;text-align:center;' . $row['correo'] . "</td><td style='border:solid 2px orange;text-align:center;'>";
            <button onclick='borrar($id)'>Borrar</button></td><td style='border:solid 2px orange;text-align:center;'><button onclick='modificar($id)'>Modificar</button></td>";
            echo "</tr>";
        }
        echo "</table>";
    } else {
        echo "No hay usuarios registrados en este momento";
    }
    $conn->close();
}

```

La función mostrar usuarios realiza una consulta a la base de dato en la que muestra en una tabla los usuarios existentes en la base de datos.

Mostrara también varios botones, el botón de borrar y el botón de modificar, que cuando son pulsado llaman a la función modificar() o borrar()

La función borrar() envía una petición de Ajax al php con los datos del id.

```

function borrar(id) {
    let peticion = $.ajax({
        url: "../funciones/funciones.php",
        type: "POST",
        data: {
            id: id
        },
        async: true,
        success: function(data) {
            if (data == "false") {
                alert("No se ha podido eliminar");
            } else {
                alert("Eliminado");
            }
        },
        error: function(data) {
            alert("Error en la trasmision");
        }
    });
}

```

```

if (isset($_POST['id'])) {
    $id = $_POST['id'];
    $servername = "localhost";
    $username = "root";
    $dbname = "abfinal";
    $password = "9455";

    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // sql to delete a record
    $sql = "DELETE FROM usuarios WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "True";
    } else {
        echo "False";
    }

    $conn->close();
}

```

Y el php borra el usuario que coincide con id enviado.

La función modificar()

```

function modificar(idmod) {
    let actualizado = prompt("Escribe el nuevo correo electronico");
    let peticion = $.ajax({
        url: "../funciones/funciones.php",
        type: "POST",
        data: {
            idmod: idmod,
            actualizado,
            actualizado
        },
        async: true,
        success: function(data) {
            if (data == "false") {
                alert("No se ha podido modificar");
            } else {
                alert("Modificado");
                location.reload();
            }
        },
        error: function(data) {
            alert("Error en la trasmision");
        }
    });
}

```

Pregunta el nuevo correo al que quiere cambiar, y luego manda a través de una petición de Ajax al php el id y el nuevo correo.

```
if (isset($_POST['actualizado'])) {
    $nuevo = $_POST['actualizado'];
    $id = $_POST['idmod'];
    $servername = "localhost";
    $username = "root";
    $dbname = "abfinal";
    $password = "9455";

    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "UPDATE usuarios SET correo='$nuevo' WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "true";
    } else {
        echo "false";
    }

    $conn->close();
}
```

El php realiza la consulta de actualización a la base de datos y actualiza el usuario.
Por ultimo el botón para cerrar sesión

```
<button id="borrarcookie" style="margin-top:3%">Cerrar sesion</button>
```

El botón llama a la función borrarcookie()

```
function borrarcookie() {
    document.cookie = "id=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";
    window.location.href = "iniciosesion.php";
}
```

La cual borra la cookie y refresca la pagina la usuario.

El resultado final seria:

unan1515@gmail.com	<input type="button" value="Borrar"/>	<input type="button" value="Modificar"/>
daniel@gmail.com	<input type="button" value="Borrar"/>	<input type="button" value="Modificar"/>

Cerrar sesion

Foro.php

Por último esta la página de foro, que es un chat entre los usuarios registrados en la pagina.

Lo primero que hacemos en la pagina es revisar que el usuario este con la sesión iniciada.

```
<?php
if (isset($_COOKIE['id'])) {
```

Si el usuario esta con la sesión iniciada llama a la función mostrarMensajes()

```
<section class="d-flex flex-column align-items-center mt-4">
  <div>
    <?php
    include("../funciones/funciones.php");
    mostrarMensajes();
    ?>
  </div>
  <div class="mt-3">
    <form action="../funciones/funciones.php" method="post">
      Escribe aquí: <textarea rows="2" cols="33" name="mensaje"></textarea>
      <input type="submit" name="enviar" value="Enviar">
    </form>
  </div>
</section>
```

```

function mostrarMensajes()
{
    $servername = "localhost";
    $username = "root";
    $password = "9455";
    $dbname = "abfinal";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT m.id,u.correo,m.mensaje FROM mensajes m INNER JOIN usuarios u ON u.id = m.id_usuario";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // output data of each row
        while ($row = $result->fetch_assoc()) {
            echo $row['correo'] . ": " . $row['mensaje']."<br>";
        }
    } else {
    }
    $conn->close();
}

```

La cual hace una consulta a la base de datos que muestra en pantalla el correo de la persona que ha escrito el mensaje y el mensaje.

En el momento en el que un usuario envía un mensaje a través del formulario pasa los datos a funciones.php

```

}
if (isset($_POST['mensaje'])) {
    $mensaje = $_POST['mensaje'];
    $servername = "localhost";
    $username = "root";
    $password = "9455";
    $dbname = "abfinal";
    if($mensaje == "" || $mensaje == " "){
        header("location:../pages/foro.php");
    }else{
        $id = $_COOKIE['id'];
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }
        $sql = "INSERT INTO mensajes (id_usuario, mensaje) VALUES ($id, '$mensaje')";
        if ($conn->query($sql) === TRUE) {
            header("location:../pages/foro.php");
        } else {
            echo "Algo ha fallado";
        }
        $conn->close();
    }
}
}

```

Recoge los datos del formulario y hace una consulta de inserción en la que registra el mensaje con el id del usuario, si todo ha salido correctamente refresca la pagina, si no envía un mensaje de fallo.

El resultado final seria este:

```
unan1515@gmail.com: Hola soy el primer mensaje
unan1515@gmail.com: hola
daniel@gmail.com: hola persona del foro!
daniel@gmail.com: soy nuevo aqui
daniel@gmail.com: me gusta tu foro!
daniel@gmail.com: hola de nuevo administrador, me encanta los colores naranjas
daniel@gmail.com: hola
```

Escribe aquí:

Enviar

El usuario de administrador es Correo: unan1515@gmail.com y la contraseña: Callof13

AB intermedio 5

¿Qué es un framework?

Un framework es un conjunto de herramientas o estructuras que sirve para desarrollar software facilitando el mantenimiento de la aplicación a través de esquemas predefinidos.

Su principal ventaja es que están preparadas para ser mas productivas al estar creadas para acelerar el trabajo y la productividad.

Es un código más organizado y legible para el resto de desarrolladores.

Suelen ser muy escalables e integran los estándares de programación.

¿Qué es una librería?

Una librería es un conjunto de códigos reutilizables que se utilizan comúnmente para solucionar problemas. Están formadas por clases y funciones.

Una de sus principales ventajas es que puedes elegir que partes de la biblioteca te interesa utilizar y no tener que utilizar la librería entera.

La segunda es que no tienes que seguir ningún esquema predefinido.

Por último, son más ligeras.

Framework para tienda online

Por lo que he estado investigando diría que uno de los mejores frameworks para crear una tienda online es vue.js ya que es un framework ligero y una de las mejores características que tiene vue.js es la capacidad para crear interfaces dinámicas y una gran comunidad detrás de ella.

Por último, utiliza un enlace de base de datos bidireccional lo que significa que no tiene necesidad de modificar el DOM de la pagina web.

Justificación

La principal razón por la que no utilizo un framework como Bootstrap para la entrega intermedia ha sido que ya he utilizado Bootstrap para realizar el AB final.

También he de decir que me gusta más la utilización de media queries que de Bootstrap ya que me parece poco intuitivo y el lenguaje un poco feo.

Link al github:<https://github.com/Danielesko/AbfinalProgramacion.git>