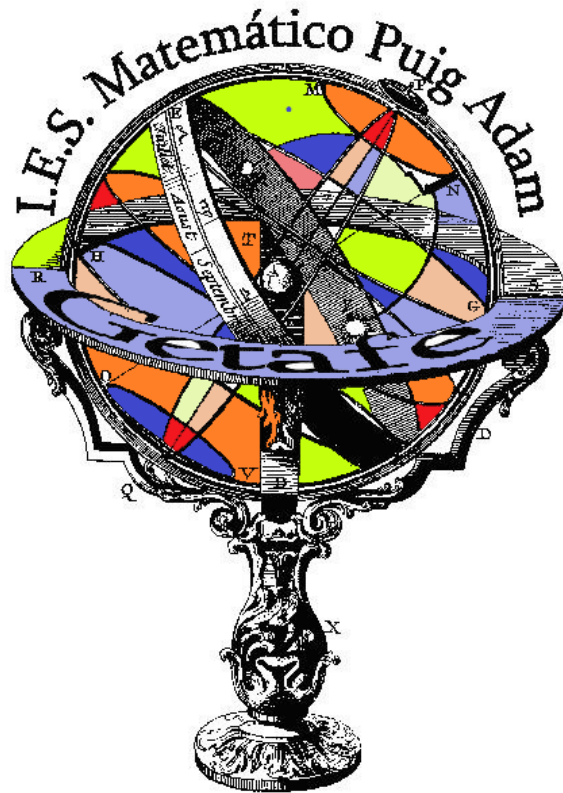


PROYECTO FINAL DE CICLO.



TIENDA DE ARTÍCULOS DE ALTA COSTURA EN LÍNEA

NELUMBO NUCIFERA

Gabriel Orellana Vásquez

2022-2023

INDICE.

1. Justificación del proyecto y su viabilidad.....	2
2. Objetivos.....	4
3. Planificación.....	5
4. Desarrollo.....	6
4.1. Diagrama entidad-relación y consultas.....	6
4.2. Estructura de archivos.....	8
4.3. Utilización de la función hash_hmac.....	13
4.4. Implementación del servicio de PayPal.....	14
5. Guía de uso.....	17
5.1. Menú.....	17
5.2. Sección de contacto y footer.....	18
5.3. Página de inicio (index.php).....	19
5.4. Registro (Registro.php) e inicio de sesión (inicio_sesion.php).....	21
5.5. Catálogo (catalogo.php).....	22
5.6. Página de producto (Pagina-producto.php).....	25
5.7. Checkout (checkout.php).....	27
5.8. Pasarela de pago(pago.php).....	27
5.9. Historial de compras (historial_compras.php) y detalles de compra (detalles_compra.php)...	29
6. Conclusiones.....	30
7. Bibliografía y webgrafía.....	31
8. Anexos.....	33
8.1. Detalles sobre Hash_hmac.....	33
8.3. Gráficos de la evolución de ventas derivadas de ecommerce.....	34

1. Justificación del proyecto y su viabilidad.

Crear una página web para realizar compras en línea es una opción muy viable y popular en la actualidad, es mas, es casi indispensable y más en el sector textil debido a la continua creación de negocios recomendados y gran competencia existente. El comercio electrónico ha experimentado un crecimiento significativo en la última década y se ha convertido en una forma conveniente y accesible de hacer compras (más información disponible en los anexos).

Antes de crear la página web de compras en línea, se deben considerar los siguientes aspectos clave:

- **Mercado objetivo:** Se debe realizar un análisis del mercado al que se desea dirigir y evaluar si existe suficiente demanda para los productos o servicios que se planea ofrecer en la tienda en línea. Es recomendable investigar la competencia existente y buscar formas de diferenciarse de ellos.
- **Infraestructura tecnológica:** Es necesario asegurarse de contar con los recursos y la infraestructura tecnológica necesarios para implementar y mantener una plataforma de comercio electrónico. Esto incluye tener un hosting web confiable, garantizar la seguridad en línea, ofrecer opciones de pago seguras y contar con una interfaz de usuario intuitiva.
- **Seguridad y protección de datos:** La seguridad de los datos de los clientes es un aspecto fundamental en una tienda en línea. Por lo tanto, es necesario implementar medidas de seguridad adecuadas, como el cifrado SSL, para proteger la información personal y financiera de los clientes.
- **Promoción y marketing:** Una vez que la página web esté en funcionamiento, será necesario promocionarla y atraer tráfico. Se deben considerar estrategias de marketing digital, como el SEO, la publicidad en línea y el uso de las redes sociales, para generar conciencia y atraer a potenciales clientes.
- **Atención al cliente:** Es importante establecer un sistema sólido de atención al cliente que permita responder rápidamente a las consultas y problemas de los clientes. Se deben proporcionar opciones de contacto, soporte técnico y tener una política de devolución clara.

En resumen, la viabilidad de crear una página web para compras en línea depende de factores como el mercado objetivo, la infraestructura tecnológica disponible y la capacidad para promocionar y brindar un buen servicio al cliente.

Sin embargo, tras decidir querer crear una página web teniendo en cuenta todos los puntos anteriores aplicables en la gran mayoría de mercados, ¿qué ofrece al mundo de los productos de alta costura? ¿Cómo afecta esta decisión en esta área? ¿Por qué es interesante desarrollar una vía en línea para su promoción?

Las tiendas online han tenido un impacto significativo en la industria de la moda de alta costura. Aunque la moda de alta costura se ha caracterizado tradicionalmente por su exclusividad y su enfoque en la atención personalizada, las tiendas en línea han logrado introducir cambios importantes en este sector. Aquí hay algunas formas en las que las tiendas online han influido en la moda de alta costura:

Acceso global: Las tiendas online han eliminado las barreras geográficas y han permitido que las marcas de alta costura lleguen a una audiencia global. Antes, los clientes tenían que viajar a las capitales de la moda, como París o Milán, para acceder a las colecciones de alta costura.

Ahora, pueden comprar estas prendas desde cualquier parte del mundo, lo que amplía el alcance y la visibilidad de las marcas.

Experiencia de compra conveniente: Las tiendas online ofrecen a los consumidores una experiencia de compra conveniente y sin problemas. Los clientes pueden explorar y comprar productos de alta costura desde la comodidad de sus hogares, sin tener que visitar las tiendas físicas. Esto ahorra tiempo y esfuerzo, lo que ha atraído a una nueva base de clientes que aprecian la comodidad de comprar en línea.

Mayor disponibilidad y variedad: Las tiendas online han ampliado la disponibilidad de productos de alta costura. Antes, solo unos pocos privilegiados tenían acceso a estas prendas exclusivas. Ahora, las tiendas en línea ofrecen una gama más amplia de productos de alta costura, lo que permite a un público más amplio explorar y adquirir estas prendas. Esto ha aumentado la demanda y la visibilidad de la moda de alta costura.

Personalización y adaptación: Las tiendas online también han permitido a las marcas de alta costura ofrecer una mayor personalización y adaptación a las preferencias individuales de los clientes. A través de herramientas de diseño en línea, los clientes pueden personalizar prendas de alta costura según sus gustos y necesidades específicos. Esto agrega un elemento de exclusividad y satisfacción personalizada a la experiencia de compra.

Visibilidad y marketing: Las tiendas online han facilitado la promoción y el marketing de las marcas de alta costura. A través de las redes sociales, los blogs de moda y otras plataformas en línea, las marcas pueden mostrar sus creaciones y conectarse directamente con los consumidores. Esto ha aumentado la visibilidad y la conciencia de la moda de alta costura, generando un mayor interés y demanda.

Si bien las tiendas en línea han influido positivamente en la moda de alta costura, es importante destacar que esta industria sigue valorando la artesanía, la atención personalizada y la experiencia única que ofrecen las tiendas físicas. Las tiendas en línea han complementado y enriquecido la industria, pero no la han reemplazado por completo.

El proyecto presentado está orientado a ser un medio para facilitar la venta online de productos originales diseñados y confeccionados por un grupo reducido de diseñadores y patronistas, aumentando el prestigio y facilitando el visionado de las colecciones gracias a una interfaz limpia, cómoda y bella, estando acorde a una temática identificativa de la marca. Se dotará de información de contacto, así como de un sistema de seguridad para evitar el tratamiento indebido de datos.

La marca sobre la que se basará el diseño del producto final recibe el nombre de Nelumbo Nucifera, marca de vestuario sin género y con disponibilidad en tallas variadas y colores que añadirán un punto de personalización y exclusividad. Los productos que ofrece la empresa se centran en conceptos de high couture siendo, por ello, de tono atrevido y extravagante pero manteniendo una imagen cuidada y apolínea que se refleja en el diseño final de la página web. El diseño es vital para atraer a los compradores.

2. Objetivos.

- Objetivos generales.

- Lograr aumentar el prestigio de la marca representada gracias a una presentación estéticamente bella del catálogo perteneciente a la parte interesada en contratar nuestro servicio como desarrolladores de aplicaciones web.
- Incrementar el nivel de ventas gracias a la extensión de mercado que permite la acción de posicionar la marca en el medio online, así como, utilizar algunas tácticas de marketing en el diseño de la web para resultar aún más apetecible para el usuario medio (en las cuales se incluye la utilización de tonos claros para evitar sensaciones de incomodidad manteniendo al usuario el mayor tiempo posible en la página web, una interfaz clara y de uso placentero debido al manejo cómodo, etc).
- Representar la marca de forma fiel adecuando la parte visual de la web a nuestro cliente logrando distinción e identidad.

- Objetivos específicos.

- Controlar los productos que se mostrarán en el catálogo gracias al registro “activo” de la BBDD que permitirá discernir un producto disponible (1) de un producto no disponible (0). Además, mostrar el catálogo empleando el filtro disponible en el menú para solo visualizar la categoría que se desee, así como, mostrar los datos concretos de cada producto (talla y color).
- Mantener un sistema de seguridad por medio de la utilización de tokens (hash_hmac) evitando intervenciones externas al cambiar de página una vez se envían datos (página de producto, página de pago, checkout). El enlace visible contendrá información oculta, la cual se utiliza para el acceso a datos concretos como puede observarse en el caso de la página de producto.
- Mantener la BBDD protegida. Los accesos a BBDD se encuentran gestionados por medio de PDO manteniendo la utilización de consultas preparadas para *evitar* inyecciones.
- Permitir el registro de usuarios, el inicio de sesión, cierre de sesión y, de la misma manera, compras asociadas al usuario que inicie sesión.
- Mostrar datos sobre las compras asociadas a cliente realizadas anteriormente.
- Permitir la compra por medio de PayPal y tarjeta de crédito. Una vez se realice la gestión se guardarán los datos correspondientes a la compra en la BBDD.

- Objetivos de buenas prácticas de programación.

Utilizar buenas prácticas de programación ofrece una serie de beneficios y objetivos clave que ayudan a mejorar la calidad del código, facilitar el mantenimiento y promover la colaboración entre desarrolladores. Aquí hay algunos objetivos importantes de utilizar buenas prácticas de programación:

1. **Legibilidad del código:** Las buenas prácticas de programación promueven el uso de convenciones y estándares que hacen que el código sea fácil de leer y entender. Esto es importante para facilitar la colaboración entre desarrolladores y permitir que otros puedan comprender y mantener el código en el futuro.
2. **Mantenibilidad:** El código bien estructurado y organizado es más fácil de mantener a largo plazo. Las buenas prácticas, como la modularidad y la cohesión, ayudan a reducir la complejidad y permiten realizar cambios o correcciones de manera más eficiente, minimizando el riesgo de introducir nuevos errores.
3. **Reutilización de código:** Al seguir buenas prácticas, se tiende a escribir código más modular y cohesivo, lo que facilita la reutilización de componentes en diferentes partes del proyecto o incluso en proyectos futuros. Esto ahorra tiempo y esfuerzo al no tener que volver a escribir funcionalidades similares desde cero.
4. **Eficiencia y rendimiento:** Al aplicar buenas prácticas de programación, se pueden evitar ineficiencias y cuellos de botella en el código. Esto puede implicar utilizar algoritmos más eficientes, optimizar el uso de recursos o evitar operaciones innecesarias, lo que conduce a un mejor rendimiento del programa.
5. **Calidad del software:** Las buenas prácticas de programación fomentan la adopción de estándares de calidad y la implementación de pruebas unitarias y de integración. Esto ayuda a identificar y corregir errores de manera temprana, mejorar la robustez del software y aumentar la confiabilidad del mismo.
6. **Facilitar la colaboración:** Al seguir buenas prácticas, se establece una base común de reglas y convenciones para el equipo de desarrollo. Esto facilita la colaboración, la revisión de código y la integración de cambios realizados por diferentes personas, ya que todos trabajan bajo una estructura coherente y comprensible.
7. **Escalabilidad:** Las buenas prácticas de programación permiten que el código sea más escalable, es decir, que pueda crecer y adaptarse a medida que los requisitos del proyecto evolucionan. La modularidad, la abstracción y el diseño adecuado del código ayudan a facilitar la incorporación de nuevas funcionalidades o cambios sin afectar negativamente al sistema existente.

En resumen, el uso de buenas prácticas de programación busca mejorar la legibilidad, la mantenibilidad, la reutilización, la eficiencia, la calidad, la colaboración y la escalabilidad del código, lo que conduce a un desarrollo de software más robusto y exitoso.

3. Planificación.

La planificación llevada a cabo para completar varios puntos del desarrollo se ha cumplido dentro de los objetivos básicos para llevar a cabo una compra asociada a un producto y un usuario con registro en base de datos. En cuanto a los detalles tales como darse de baja y/o la implementación de tallas y colores junto a stock que surgieron más tarde no fueron totalmente llevados al producto final.

- Presentación de parte del proyecto: 26 de Abril.
Se presentará parte de la maquetación y diseño inicial perteneciente a la parte de registro, página de producto y galería o catálogo, incluyendo un navegador y un footer con la parte de contacto que se utilizarán en el resto de la web. Primará la utilización de html, css, js, bootstrap principalmente.
- Presentación de parte del proyecto: 10 de Mayo.
Se presentará la página de inicio junto a la página de catálogo o galería. Se ajustará el diseño para que resulte más cohesivo. Además el navegador comienza a ser funcional junto al catálogo por defecto (todos los artículos disponibles). Incorporación de php a todo el desarrollo.
- Presentación de parte del proyecto: 24 de Mayo.
Se presentó la BBDD finalizada junto a las partes restantes relacionadas con la compra online y su relación con la BBDD (páginas de checkout, historial de compras y detalle de compra) integrando el funcionamiento correcto de las variables de sesión empleadas a lo largo del desarrollo. Utilización de MariaDB para la base de datos.

4. Desarrollo.

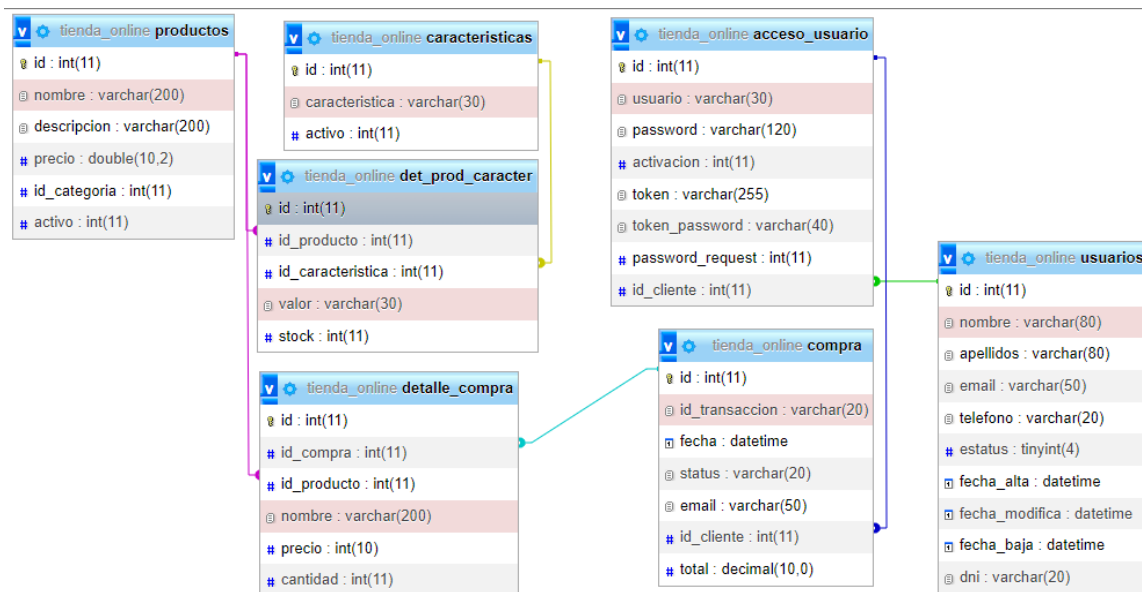
4.1. Diagrama entidad-relación y consultas.

La base de datos inicial fue más sencilla que la que se mostrará a continuación, prescindiendo inicialmente de características y la tabla que actúa de unión entre ella y la tabla de productos. La decisión de incluir características se tomó en referencia a poder dar mayor personalización a los productos que se pidiesen en línea junto a un stock controlado asociado a una talla y a un color.

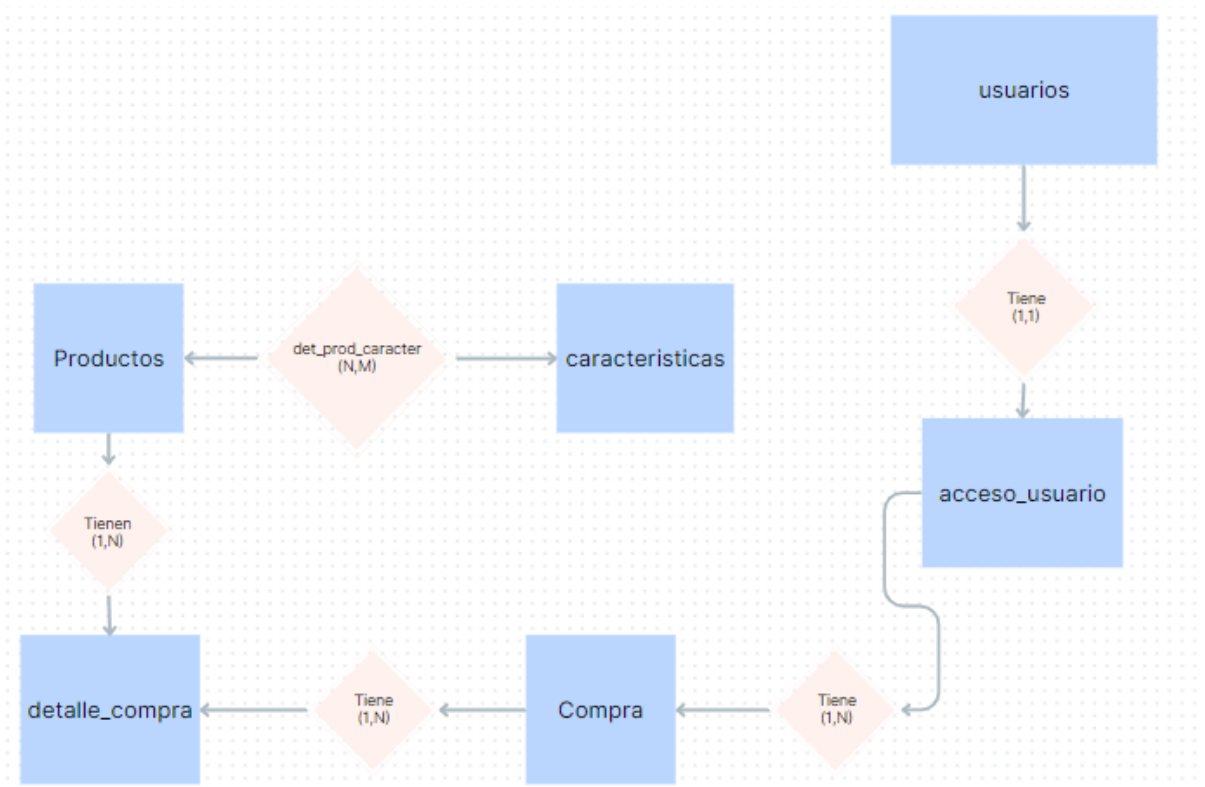
Por condiciones de tiempo y organización no pudo adecuarse la tabla al resto del desarrollo pero eso no impide que la funcionalidad sea correcta y que puedan venderse todo tipo de productos siempre y cuando la asociación de tallas o colores no esté implícita. Sin embargo, se mantuvo la existencia de ambas tablas para poder mostrar el despliegue de datos asociados en las opciones del producto una vez se accediese a cada uno de los artículos de la tienda.

La estructura que da lugar a la base de datos del producto final está constituida por “productos”, tabla relacionada a su vez, con “características” por medio de la tabla “det_prod_caracter” formando una relación N:M, “compra” que se relaciona con “detalle_compra” formando una relación 1:N, donde esta última se relaciona con “productos” por medio de una relación 1:N. Por último, “usuarios” se relaciona con “acceso_usuario” (1:1) y, esta última tabla, conecta con compra gracias a una relación 1:N.

En la siguiente imagen se muestran los campos que contienen las diferentes tablas que constituyen la base de datos.



Para mayor facilidad a la hora de visualizar las relaciones anteriormente descritas se muestra la siguiente imagen donde se puede ver de forma inmediata la manera en la que se relacionan las entidades de la BBDD.



Por último, las consultas realizadas siguen la estructura PDO junto a bindParam para aumentar la seguridad en el código. Aquí se presenta un ejemplo en el cual se están obteniendo datos de un producto gracias a su id. Normalmente se recurre al id por lo que cuando se utiliza bindParam se emplea PARAM_INT pero, en caso de que se tratase de un dato de tipo string, lo utilizado de forma

correcta es PARAM_STR (en alguna ocasión, como por ejemplo, con el nombre de usuario, se puede ver en código).

```
//Tras validar el token correctamente.
$sql = $con->prepare("select count(id) FROM productos where id=? and activo= 1");
$sql->bindParam(1, $id, PDO::PARAM_INT);
$sql->execute();

if ($sql->fetchColumn() > 0) {
    $sql = $con->prepare("select nombre, descripcion, precio, id_categoria FROM productos where"
        . " id=? and activo= 1 limit 1");
    $sql->bindParam(1, $id, PDO::PARAM_INT);
    $sql->execute();
    $row = $sql->fetch(PDO::FETCH_ASSOC);
    //Se recogen los valores tras realizar la consulta.
    $nombre = $row["nombre"];
    $descripcion = $row["descripcion"];
    $precio = $row["precio"];
    $id_categoria = $row["id_categoria"];
```

4.2. Estructura de archivos.

La estructura de la que se compone la página web toma la siguiente estructura de archivos (los directorios se mostrará subrayados):

Proyecto_final (directorio principal):

- Editorial: contiene las fotos utilizadas en la galería de la página de inicio (index.php).
- clases: directorio que contiene tres archivos.
 - actualizar_carrito.php: contiene datos que intercambia con checkout.php mediante AJAX. De esta forma se mantiene actualizado el contenido del carrito.
 - carrito.php: se encarga de hacer el recuento de productos que se acumulan en el carrito de compras, la información recogida se envía por medio de Json.
 - informacion_capturada: su funcionamiento consiste en, una vez aprobada la compra por PayPal, obtener los datos del array multidimensional que obtenemos derivados de PayPal.
- css: directorio que contiene archivos referentes a la hoja de estilos y el resto del funcionamiento aplicado sobre la interfaz.
 - all.min.css
 - mdb.min.css
 - styles.css: es el archivo principal en cuanto al diseño, aquí se recogen la mayoría de estilos aplicados al html.
- img: el directorio que contiene la mayoría de imágenes que serán utilizadas a lo largo del desarrollo. Uno de los archivos más importantes es el propio logotipo.
- assets: contiene las imágenes empleadas como emoticonos (teléfono, avatar, ubicación, correo electrónico).

- js: directorio que contiene todos los archivos referidos a hojas de script que se utilizarán en el resto del desarrollo principalmente para manejar la parte estética de la página web.
 - all.min.js
 - bootstrap.min.js
 - jquery.min.js
 - mdb.min.js
 - nbproject: directorio que contiene varios archivos xml. Su contenido, así como su propia existencia, se debe a los archivos que vienen por defecto con el resto de prestaciones de netbeans.
 - productos: directorio que contiene una segunda tanda de directorios con un número (1, 2, 3, 4, 5, etc). Cada subdirectorio contiene una imagen diferente referenciando a un producto diferente que recibe el nombre de principal.jpg. La idea inicial es poder utilizar varias imágenes que recibirán el mismo nombre en cada directorio para poder llamar a su ruta de manera eficiente y mostrar diferentes ángulos del producto a ofrecer en caso de guardar más imágenes que principal.jpg.
 - ***Pagina_producto.php*** : Muestra la información perteneciente al producto seleccionado en el catálogo, así como algunos datos de otros productos de categorías diferentes (si hay un pantalón como producto principal debajo se mostrará una lista de 4 items que serán de cualquier categoría menos la del pantalón).
Al mostrar los datos se conecta tanto con productos para mostrar precio, nombre y descripción como talla y color que pertenecen a la tabla de características que incluirán ambos detalles mencionados.
- Al recibir el id de producto (catálogo.php) que se utilizará para acceder a los datos se emplearán tokens para proteger la información enviada, tras ello se comparará el token creado anteriormente con el token creado en esta parte validando que ambos pertenecen al mismo id que se envió.
- ***Registro.php***: Este archivo se conforma de un formulario y las validaciones pertenecientes a todos los campos que encontramos en el mismo.
Se utilizan funciones que encontraremos en config.php para validar datos como el correo electrónico o ,incluso, comprobar si el nombre de usuario ya existe o, el mismo correo, ya está registrado en la bbdd.
Además, una vez el sistema haya comprobado error a error, en caso de darse, se incluirán en un array que se manejará finalmente en una función para mostrar dichas incongruencias en un alert a juego con la estética de la página que podrá eliminarse gracias a un evento de js.
 - ***catalogo.php***: Este archivo muestra una galería de productos con datos generales sobre el producto junto a un botón que nos redirige a la *Pagina_producto.php* correspondiente a cada producto mediante el uso de su id.
Se utiliza hash_mac para cifrar información que se enviará al siguiente destino (en este caso se cifra el id). Una vez se reciba la información del id cifrado (token) en el archivo de producto se comprobará si es correcto.

El funcionamiento de las categorías consiste en filtrar la selección de artículos según requiera el cliente. El cifrado y la comprobación del token debe realizarse incluso aquí mismo ya que se cifra la categoría en la parte de `menu.php`, siendo enviada de la misma manera que se envía desde `catalogo.php` a `Pagina-producto.php`.

- **`cerrar_sesion.php`**: Archivo encargado de cerrar la sesión de usuario, tras ello se redirige a la página de inicio `index.php`.
- **`checkout.php`**: muestra la lista de productos que se comprarán, dichos artículos están conectados por su id.
Se ofrece la posibilidad de modificar la cantidad de los productos lo que sucede en tiempo real gracias a `actualizar_carrito.php` que contiene una función específica para enviar datos por Json y actualizar subtotales y totales o incluso eliminar el producto seleccionado con el emoticono con apariencia de papelera (se le atribuye esta posibilidad gracias a otra función incluida también en dicho archivo).

Se puede acceder a la ventana de `pago.php` mediante el botón realizar pago pero, solo en caso de haber iniciado sesión (`$_SESSION['user_id']`), de lo contrario se redirige a la página de inicio de sesión (`inicio_sesion.php`).

- **`config.php`**: Es uno de los archivos que se utilizarán globalmente junto a `database.php`.
En este caso se define la moneda del euro para utilizar en el resto del código y los tokens para cifrar la información que recorrerá el catálogo y la página de producto.
Además, se definen también datos correspondientes a la conexión con PayPal, utilizando también el euro como moneda principal.

Por último, además del inicio de las variables de sesión, encontramos diferentes funciones pertenecientes a las validaciones del registro de usuarios y acceso del usuario ya registrado. Se comprueban tanto si existen ciertos datos como el formato o si se deja algún campo vacío que deba rellenarse mediante el uso de funciones.

```
//Funciones para el registro de usuarios.

function nulo(array $datos){
    foreach ($datos as $dato){
        if(strlen(trim($dato)) < 1){
            return true;
        }
    }
    return false;
}

function email($email){
    if(filter_var($email, FILTER_VALIDATE_EMAIL)){
        return true;
    }
    return false;
}

function validarPassword($password, $repassword){
    if(strcmp($password, $repassword) === 0){
        return true;
    }
    return false;
}
```

- **database.php**: conecta con la BBDD de forma segura, en caso de error de conexión no realizará ninguna operación evitando comprometer la integridad de la BBDD. En la siguiente imagen se muestra la manera en la que se ha codificado utilizando PDO junto a try y catch reduciendo así la posibilidad de obtener fallos. Como se mencionó en el anterior punto (config.php), este archivo será llamado en el resto de archivos.

```
class Database {
    private $hostname = "localhost:3307";
    private $database = "tienda_online";
    private $username = "root";
    private $password = "";
    private $charset = "utf-8";

    function conectar() {

        try{
            $conexion = "mysql:host=". $this->hostname . ";dbname=" . $this->database;

            /*Para lograr mayor seguridad*/

            $options = [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_EMULATE_PREPARES => false
            ];

            $pdo = new PDO($conexion, $this->username, $this->password, $options);

            return $pdo;
        } catch (PDOException $e){
            echo 'Error de conexión'. $e->getMessage();
            exit; //Si se genera un error ya no realizará ninguna operación extra.
        }
    }
}
```

- **detalles_compra.php**: Como ya sucedía anteriormente, gracias a la utilización de tokens para enviar información de un lado a otro, se validará que todo sea correcto, junto a ello, si no se encontrase nada que mostrar o sucediese un error, se mostrará de nuevo el *historial_compras.php*. Lo común es que si se encuentra una tarjeta con datos de compra en el historial de compras se pueda acceder a esta página con más detalles sobre los productos que conformaron dicha gestión. Se mostrarán los datos de la compra general así como los datos de los productos incluidos en la compra correspondiente.
- **footer.php**: Parte de contacto y footer que se colocará, gracias a include, en todas las páginas mostradas a lo largo de la navegación.
- **historial_compras.php**: El historial de compras mostrará las compras realizadas por el usuario registrado que haya iniciado sesión y haya realizado compras por medio de PayPal (en nuestro caso). Cada compra está asociada a un id de cliente por lo que a la hora de mostrar la lista de gestiones de compra la consulta utilizará la variable de sesión del id del usuario que se haya conectado. De no haberse registrado y conectado posteriormente esta opción no aparecerá. Cada tarjeta contendrá un botón que nos redirigirá a *detalles_compra.php* para mostrar detalles sobre los productos incluidos en la gestión.

- **index.php**: La página de inicio mostrará una pequeña presentación de la marca sobre la cual se ha centrado el diseño junto a una galería de imágenes de los diferentes diseños que incluye la colección más actual mostrando en modelos como funcionan los artículos de moda que se presentan en catálogo.
La galería de fotos incluye modales para cada una de las fotos lo que ayuda a ver en detalle la prenda junto al modelo.
- **inicio_sesion.php**: El archivo de inicio de sesión contiene un pequeño formulario que recoge los datos de un usuario que en principio ya debe estar registrado en nuestra BBDD. Se comprueba siempre que esté todo relleno antes de enviar los datos para comprobar con la función login.
Una vez los datos son procesados se comprobarán las coincidencias y en caso de que no se encuentre al usuario en los registros se procederá a enviar el mensaje de error, de lo contrario se enviará al usuario a la página de inicio con la sesión ya inicializada.
La sesión inicializada se puede comprobar cuando clicamos sobre el emoticono de usuario del menú (derecha).
Siempre dispondremos de la opción de dirigirnos a la pantalla de registro.
- **menu.php**: El menú mostrará todas las opciones de acceso a las diferentes partes (inicio de sesión, registro, inicio, catálogo en todas sus formas). Aunque es importante mencionar que varias de las opciones dependerán de si el usuario ha iniciado sesión o no pues en caso de hacerlo las opciones de usuario serán un saludo y cerrar sesión así como la opción de acceder a pago.php una vez se quiera relizar el pago en checkout.php. En el caso contrario, al no iniciar sesión se mostrarán las posibilidades de iniciar, registrar y el acceso a pago.php se verá imposibilitado. La parte visual junto a la funcionalidad será tratada más adelante en la guía de usuario pero ¿cómo funciona por dentro? para responder a esta pregunta se mostrará parte del código junto a pequeñas aclaraciones.

Aquí encontramos la estructura en código que sigue el menú donde se puede detectar la existencia de varias clases que forman parte del funcionamiento de bootstrap (el diseño final que toma a ojos del cliente) pero también se encuentra código embebido de php que, en esta parte concreta, impedirá que “Tus pedidos” se muestre en caso de que el cliente no haya iniciado sesión con una cuenta.

```
<div class="col-md-4 d-flex justify-content-center justify-content-md-end align-items-center">
  <div class="d-flex">
    <div class="dropdown">
      <a class="text-reset me-3 dropdown-toggle hidden-arrow" href="#"
        id="navbarDropdownMenuLink" role="button" data-mdb-toggle="dropdown"
        aria-expanded="false">
        <i class="fas fa-shopping-cart"></i>
      </a>
      <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdownMenuLink">
        <?php if (isset($_SESSION['user_id'])) {
          ?>
          <li><a class="dropdown-item" href="historial_compras.php">Tus Pedidos</a></li>
          <?php } ?>
          <li>
            <a class="dropdown-item" href="checkout.php">Comprar</a>
          </li>
        </ul>
      </div>
    </div>
  </div>
```

Esto ocurre a lo largo de todo el menú a la hora de desplegar o no opciones. Las opciones mostradas según la situación se tratarán en la guía de usuario.

Otro detalle interesante del menú es como diferencia las categorías a la hora de mostrar un catálogo específico y no siempre los mismos productos.

```
.i class="sidenav-item">
  <?php $categoria = 1; ?>
  <a class="sidenav-link ripple-surface" data-mdb-toggle="collapse" href="#sidenav-collapse-1-0-1" role
    <i class="fas fa-layer-group pe-3"></i><span>Pantalones</span><i
      class="fas fa-angle-down rotate-icon" style="transform: rotate(180deg);"></i></a>
  <ul class="sidenav-collapse show collapse" id="sidenav-collapse-1-0-0">
    <li class="sidenav-item">
      <a class="sidenav-link ripple-surface" href = "catalogo.php?categoria=<?php echo $categoria;
        <?php echo hash_hmac('sha1', $categoria, KEY_TOKEN); ?>">Largos</a>
    </li>
    <li class="sidenav-item">
      <?php $categoria = 3; ?>
      <a class="sidenav-link ripple-surface" href = "catalogo.php?categoria=<?php echo $categoria;
        <?php echo hash_hmac('sha1', $categoria, KEY_TOKEN); ?>">Leggins</a>
    </li>
  </ul>
..
```

Para cada opción se cambiará, mediante php, el número de la categoría de forma que el token cifrado que se envía a catálogo junto al mismo id de la categoría que servirá para validar en este último archivo determinarán que productos se enseñarán en el catálogo como ya se explicó en el apartado de *catalogo.php*. Para comprender el funcionamiento de la función `hash_hmac()` y su forma de cifrar información se podrá leer más en el siguiente punto (4.3) y, en caso de querer ahondar algo más, se podrá ver información extra en el anexo dedicado a ello.

- ***pago.php***: El acceso a esta página se ve posibilitado solo a usuarios registrados con sesión iniciada.

Tras aprobar la compra rellenando los datos disponibles se realiza la inserción de datos del carrito de compra en la BBDD y se registra la compra (esto se realizará en *informacion_capturada.php*).

El fin de esta página es mostrar la información de los artículos a comprar, los precios finales y los botones de pago junto a los cuestionarios correspondientes al sistema de PayPal.

Para comprender totalmente como se añade esta funcionalidad de pago se podrá ver en detalle como se implementó en el punto 4.4.

4.3. Utilización de la función `hash_hmac`.

La función `hash_hmac` se utiliza para generar un código hash HMAC (Keyed-Hash Message Authentication Code). Un HMAC es un algoritmo de verificación de integridad y autenticación que utiliza una clave secreta para procesar los datos y producir un resumen (hash) único.

Los parámetros de la función son los siguientes:

- \$algorithm: Especifica el algoritmo hash a utilizar. Puede ser un valor como "md5", "sha256", "sha512", etc. PHP admite una variedad de algoritmos hash. Es importante elegir un algoritmo adecuado según los requisitos de seguridad y compatibilidad.
- \$data: Los datos que se utilizarán para calcular el HMAC. Puede ser una cadena de texto o un flujo de bytes.
- \$key: La clave secreta utilizada para calcular el HMAC. También puede ser una cadena de texto o un flujo de bytes.
- \$raw_output (opcional): Un parámetro booleano opcional que indica si se debe devolver el resultado como una cadena de caracteres hexadecimal (false, valor predeterminado) o como una cadena de bytes crudos (true).

Un ejemplo dentro del código que se puede encontrar es el siguiente (Pagina-producto.php):

```
$id = isset($_GET['id']) ? $_GET['id'] : ''; //En caso de que exista tomará el valor anterior y sino ''.
$token = isset($_GET['token']) ? $_GET['token'] : '';

if ($id == '' || $token == '') {
    echo 'Error al procesar la petición.'; //De no tener el token disponible.
    exit;
} else {
    //En caso de que todo se procese de forma correcta.
    $token_tmp = hash_hmac('sha1', $id, KEY_TOKEN);

    if (trim($token) == trim($token_tmp)) {
        //Tras validar el token correctamente.
        $sql = $con->prepare("select count(id) FROM productos where id=? and activo= 1");
        $sql->bindParam(1, $id, PDO::PARAM_INT);
        $sql->execute();
    }
}
```

Se recibe el token (variable \$token) que originalmente se originó de la variable \$id y, tras ello, se crea otro token con los mismos parámetros (\$token_tmp). Ambos tokens serán comparados y en caso de resultar ser iguales se procederá a realizar las consultas pertinentes.

¿De dónde se obtiene el primer token? Se envía tras seleccionar un producto asociado a su id (se ha obtenido de una consulta sql) en la parte de catálogo.

```
<a href="Pagina-producto.php?id=<?php echo $row['id']; ?>&token=
<?php echo hash_hmac('sha1', $row['id'], KEY_TOKEN); ?>">
    <button>Más detalles</button></a>
</div> <!-- hash permite tomar datos, añadir una contraseña y cifrarla.-->
```

Para más información sobre el funcionamiento de la función puedes consultar el anexo que se encuentra al final del documento.

4.4. Implementación del servicio de PayPal.

Para lograr mostrar botones que generen la opción de pago de los productos una vez se decida realizar el pago se necesita una cuenta de PayPal (de esta forma también se podrán realizar pruebas).

En resumen se procede a crear una cuenta de PayPal con los datos de empresa y, a continuación, tras haber añadido a nuestro código los scripts y el resto de codificación necesaria, accedemos a sandbox

para la realización de una simulación de pago mediante cuentas de prueba de la que nos dota la página.

Los primeros pasos dentro de la codificación son los siguientes:

- Una vez se ha creado la cuenta y se ha iniciado sesión deberemos recoger el código que se nos indique en el apartado de developer. Esta información se ha repartido de la siguiente manera:

Se definen dos constantes en *config.php* con la información que la aplicación nos ha dado y la moneda en la que se realizarán los pagos.

```
//Datos correspondientes a PayPal
define("CLIENT_ID", "AbENXdZfJM_uaXbgApbkbZEtotnDWze3ml0pZ5LbuMWxR4x1TlW4SCmpbe15xdSFlp7JCeIy1liUD0tcY");
define("CURRENCY", "EUR");
```

A continuación se añade un pequeño script al archivo *pago.php*.

```
<script src="https://www.paypal.com/sdk/js?client-id=?php echo CLIENT_ID; ?>
&currency=?php echo CURRENCY; ?>"></script>
```

Este script conectará nuestra cuenta de negocios con las transacciones que sucedan en la página web que se ha desarrollado.

Finalmente se añade el código perteneciente a los botones.

```
<!-- Inicio de la zona de botones para realizar el pago. -->

<!-- Set up a container element for the button -->
<div id="paypal-button-container"></div>
</section>
```

- Para que funcione correctamente y ocurra el intercambio de información se utiliza JavaScript como se ha ido utilizando anteriormente solo que con algunas opciones más específicas. Además también incluye algunas opciones de diseño del botón (contenido en style donde color dotará al botón de un color negro y shape decidirá la forma, en este caso pill redondeará los bordes).

```
// Render the PayPal button into #paypal-button-container
paypal.Buttons({
  style: {
    color: 'black',
    shape: 'pill',
    label: 'pay'
  },
  createOrder: function (data, actions) {
    return actions.order.create({
      purchase_units: [{
        amount: {
          value: <?php echo $total; ?>
        }
      }]
    });
  },
});
```

El código no finaliza en esta parte pues ahora se describe lo que ocurre en caso de que se acepte el pago o, de lo contrario, cómo solventará la aplicación un pago cancelado. Para la primera opción se utiliza “onApprove” y para la segunda “OnCancel” y para onApprove se

emplea un archivo para capturar la información derivada del array multidimensional que crea PayPal.

```
//Si el usuario aprueba el pago se pasarán los datos a informacion_capturada.
onApprove: function (data, actions) {
    let URL = "clases/informacion_capturada.php";
    actions.order.capture().then(function (detalles) {
        console.log(detalles);
        let url = "clases/informacion_capturada.php";

        return fetch(url, {
            method: 'post',
            headers: {
                'content-type': 'application/json'
            },
            body: JSON.stringify({
                detalles: detalles
            })
        });
    });
},
//Si el usuario cancela el pago.
onCancel: function (data) {
    alert('Pago cancelado');
    console.log(data);
}
}).render('#paypal-button-container');
```

La información capturada se puede ver en el archivo de *informacion_capturada.php* que contendrá lo siguiente:

```
$json = file_get_contents('php://input'); //Toma la información desde la función de AJAX.
$datos = json_decode($json, true);

if (is_array($datos)) {

    $id_usuario = $_SESSION['user_id'];

    //Se toman como referencia los índices del array resultante de $datos tras realizar la compra.
    $id_transaccion = $datos['detalles']['id'];
    $total = $datos['detalles']['purchase_units'][0]['amount']['value'];
    $status = $datos['detalles']['status'];
    $fecha = $datos['detalles']['update_time'];
    $fecha_nueva = date('Y-m-d H:i:s', strtotime($fecha)); //Conversión.
    $email = $datos['detalles']['payer']['email_address'];
    $id_cliente = $datos['detalles']['payer']['payer_id'];
```

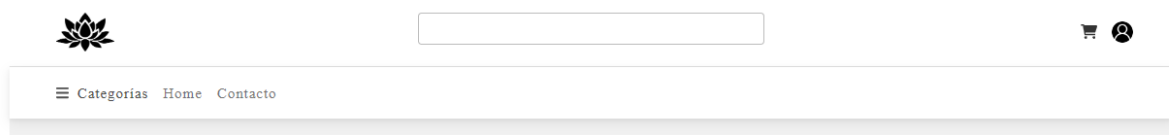
Como se explica en los comentarios se han recogido los valores del array mediante el AJAX del archivo *pago.php* y se han utilizado los índices del array (inspeccionando) para recogerlos.

5. Guía de uso.

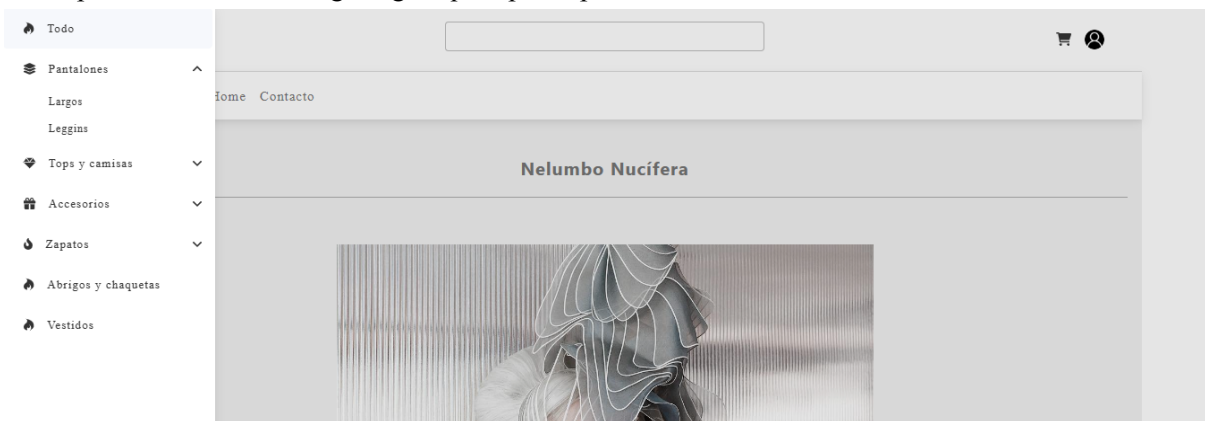
5.1. Menú.

La utilización de la página web es sencilla. Se explicará su funcionamiento a la vez que se describe una justificación de su diseño a nivel usuario en las siguientes líneas.

En la parte superior se dispone de un menú que, según la zona sobre la que se pulse redirigirá a una zona u otra.



Pulsando sobre las líneas horizontales a la izquierda de Categorías podremos encontrar los diferentes filtros para acceder al catálogo según que tipo de prenda nos interese.



La zona de Home no es mas que una forma de dirigirnos a la página de inicio en la que actualmente estamos realizando la explicación.

En cuanto a “Contacto”, justo a la derecha de “Home”, en caso de pulsar sobre él, nos dirigirá hacia la sección de contacto o footer que se encuentra al terminar la galería. En este caso, sea donde sea que nos encontremos esta sección siempre estará disponible para su acceso en la misma página ya sea inicio, catálogo, u otras.

Tras finalizar con la revisión de la zona izquierda del menú podemos avanzar hacia la derecha donde se encuentran dos emoticonos con diferentes accesos.

El carrito de compra contendrá una o varias secciones.

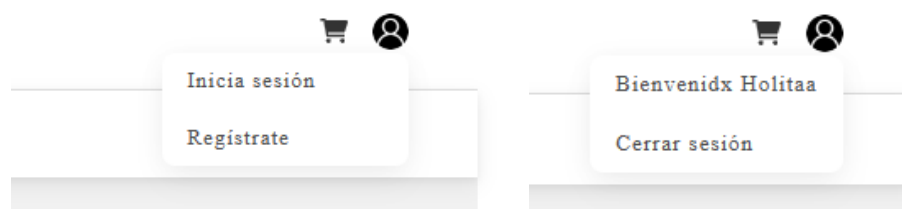


Por un lado, en caso de no haber iniciado sesión nos mostrará la posibilidad de solamente “Comprar”, enseñándonos los productos que hayamos sumado a nuestro carrito (para realizar esta operación se explicará en la parte de Página de producto) y si decidimos pagar se nos llevará a inicio de sesión evitando comprar sin registro de usuario.

Por otro lado, si el usuario está registrado y ha iniciado sesión se mostrará una opción más llamada “Tus pedidos” tratándose de un historial de compras registradas y, además, se permitirá el pago al acceder a la parte de “Comprar”.

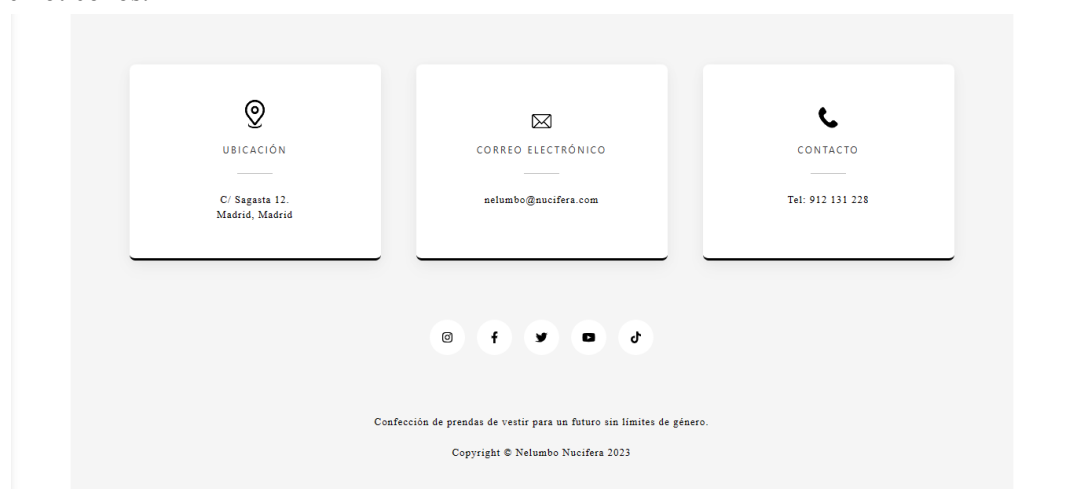
El emoticono de color negro de una persona que se aprecia justo a la derecha del carrito contendrá también diferentes opciones en caso de haber iniciado sesión o no.

Al no realizar el inicio de sesión las opciones serán: “Inicio de sesión” y “Regístrate” como se indica en la primera imagen de debajo, de lo contrario, se indicará un pequeño mensaje “Bienvenidx Holitaa [Nombre de usuario]” y debajo “Cerrar sesión”. La parte del mensaje de bienvenida solamente es visual, si pulsamos sobre ella no ocurrirá nada, sin embargo, si pulsamos sobre “Cerrar sesión” finalizaremos la sesión y volveremos a la apariencia anterior, es decir a las primeras imágenes pertenecientes a no haber iniciado nuestra sesión.



5.2. Sección de contacto y footer.

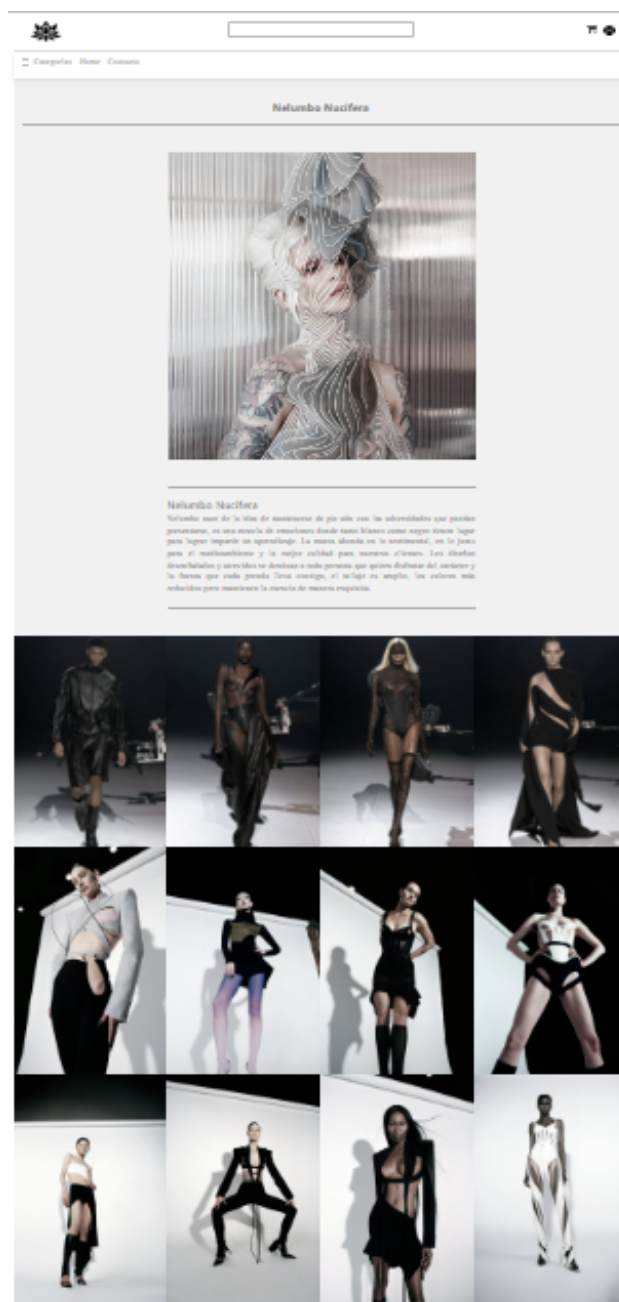
En la parte inferior se dispone de una pequeña sección y un footer de contacto con emoticonos de redes sociales pertenecientes a la marca, así como la ubicación de la tienda física. Su funcionamiento es sencillo y para acceder a alguna red social concreta bastará con pulsar sobre cada uno de los emoticonos.



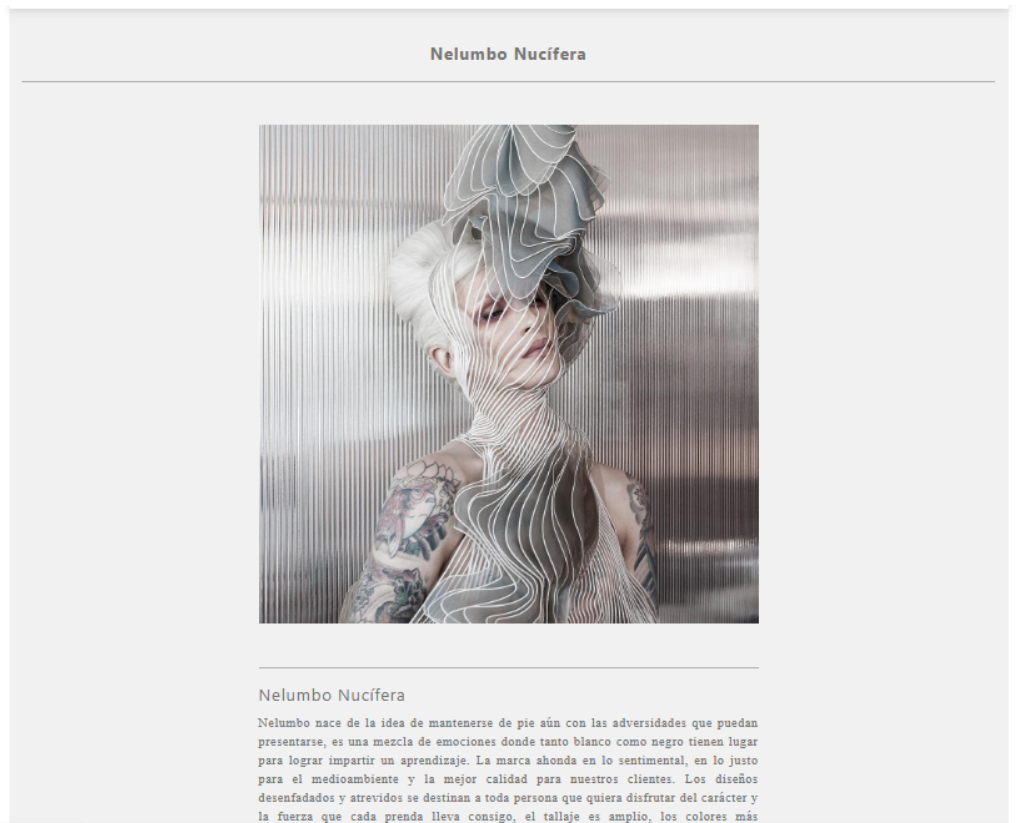
5.3. Página de inicio (index.php).

La página de inicio, en el resto de su cuerpo, mostrará una pequeña presentación junto a una imagen de estilo editorial que sumará atractivo a la imagen global. Además de la descripción se puede acceder a la parte de abajo donde encontramos una galería de fotografías que, tras pulsarse ocupan toda la pantalla enseñando en detalle el diseño de los productos puestos en un modelo, para salir de la imagen completa basta con hacer click en cualquier parte de la pantalla o pulsar sobre la x que se encuentra a la derecha y parte superior.

Imagen global sin footer incluido.



Descripción junto a una imagen tipo editorial.



Galería de fotografías. Al pasar el ratón se mostrará un efecto leve como ocurre en la primera imagen.



Tras pulsar la primera imagen se podrá ver en detalle la misma junto a un fondo negro para focalizar la atención en el modelo y el diseño. Para salir de la fotografía en tamaño completo basta con pulsar sobre cualquier zona o la x que encontramos en la esquina derecha.



5.4. Registro (Registro.php) e inicio de sesión (inicio_sesion.php).

La página de registro cuenta con un formulario que requiere de todos los datos para enviar la información pertinente a la base de datos. De la misma manera, el formulario de inicio de sesión cuenta con dos campos que también deben ser completados.

The registration form is titled "Regístrate" with a user icon. It contains the following fields: "Nombre*" with a person icon, "Apellidos*" with a pencil icon, "Usuario*" with a person icon, "Email*" with an envelope icon, "Contraseña*" with a lock icon, "Reintroduzca su contraseña*" with a lock icon, "N.I.F./C.I.F." with a document icon, and "Teléfono" with a phone icon. At the bottom, there is a "Registrar" button and a link: "¿Ya estás dado de alta? Inicia sesión".The login form is titled "Iniciar sesión" with a user icon. It contains the following fields: "Usuario" with an envelope icon and "Contraseña" with a lock icon. Below these fields is a large "Entrar" button. At the bottom, there is a link: "¿No estás dado de alta? Regístrate".

En caso de que las validaciones determinen que los datos no son correctos se avisará con una ventana tipo alert que informará de cuales son los fallos encontrados tras revisar la información obtenida.

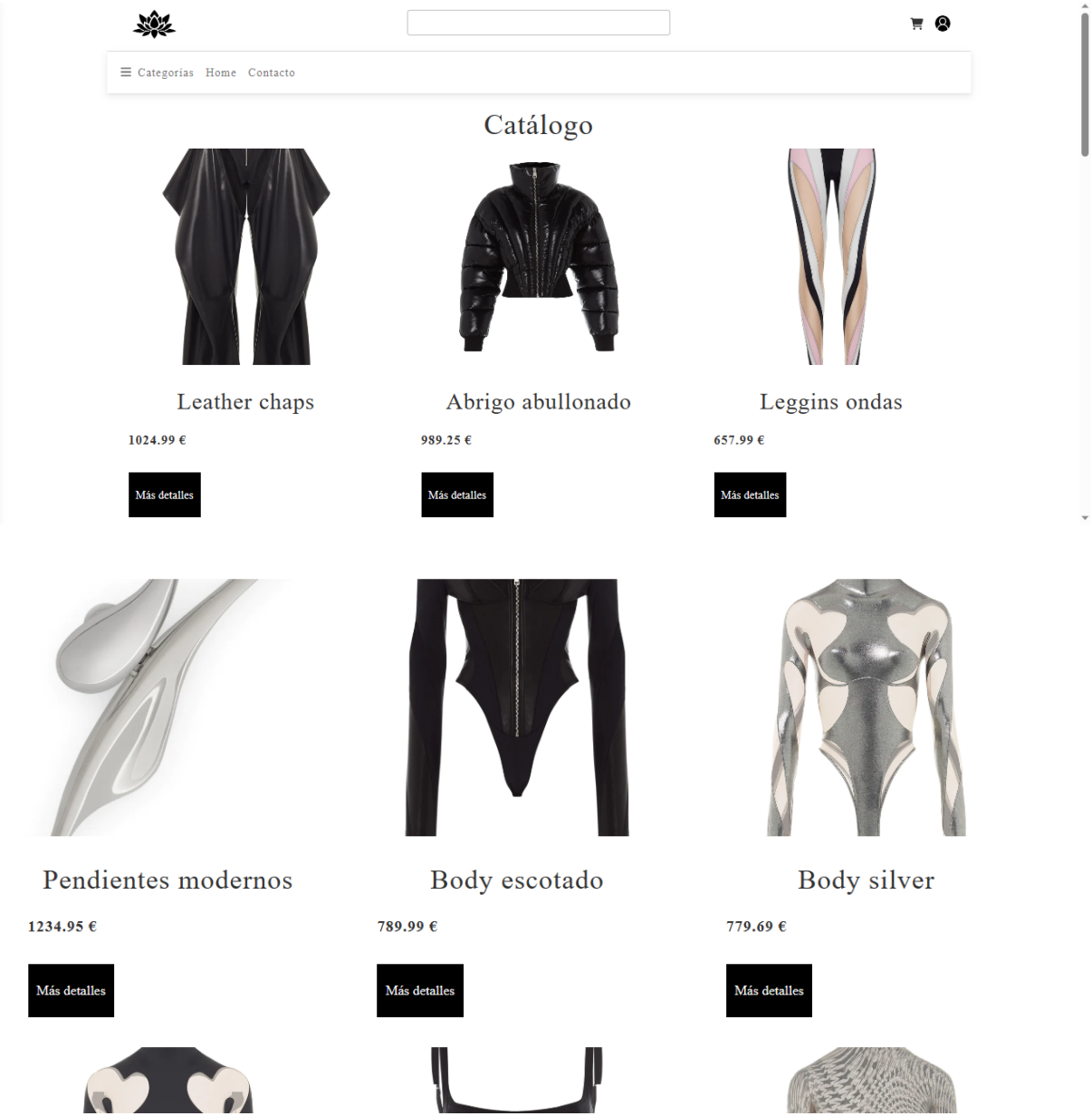
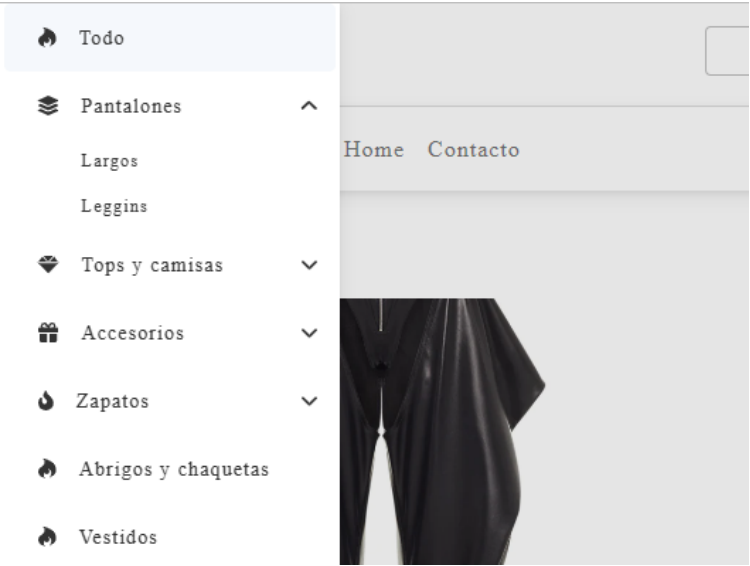
The image displays three screenshots related to a registration form. The top screenshot shows a single alert message: "El usuario y/o contraseña no son correctos." with a close button (X). The middle screenshot shows the registration form itself, which includes input fields for "Email*", "Contraseña*", "Reintroduzca su contraseña*", "N.I.F./C.I.F.", and "Teléfono", each with a corresponding icon. Below the fields is a "Registrar" button and a link that says "¿Ya estás dado de alta? Inicia sesión". The bottom screenshot shows a multi-point alert message: "La dirección de correo no es válida.", "Las contraseñas no coinciden.", and "El nombre de Holitaa ya existe.", with a close button (X).

Además, cada ventana de formulario cuenta con un texto al final, debajo del botón, que lleva de inicio de sesión a registro y viceversa facilitando al usuario la operación de registro en caso de que no disponga de cuenta o el mero cambio de una zona a otra con mayor facilidad.


5.5. Catálogo (catalogo.php).

Esta parte de la web está totalmente conectada con el menú. Dependiendo de que categoría se escoja se obtendrá un catálogo que muestre todos los elementos o solo los pertenecientes a categorías concretas.

Al pulsar sobre “Todo” obtendremos todos los productos disponibles en la BBDD. Cada producto contiene una imagen inicial junto al nombre y precio correspondiente. Si se pasa el ratón sobre cada tarjeta de producto se verá una pequeña interacción agrandando la imagen y delimitando los bordes de su tarjeta añadiendo con ello unas sombras para destacar más los bordes.



En caso de pulsar sobre el resto de opciones se filtrará por la categoría de producto. Por ejemplo: si se pulsa sobre “Tops” solo aparecerán los productos que sean de dicho tipo.


 Tops y camisas ^

Tops

Camisas

Home



Contacto




≡ Categorías

Home

Contacto




Catálogo



Body escotado

789.99 €


Más detalles



Body silver

779.69 €


Más detalles



Body ondas negro

755.99 €


Más detalles



Corset transparencias

569.99 €

Más detalles



Top estrellado

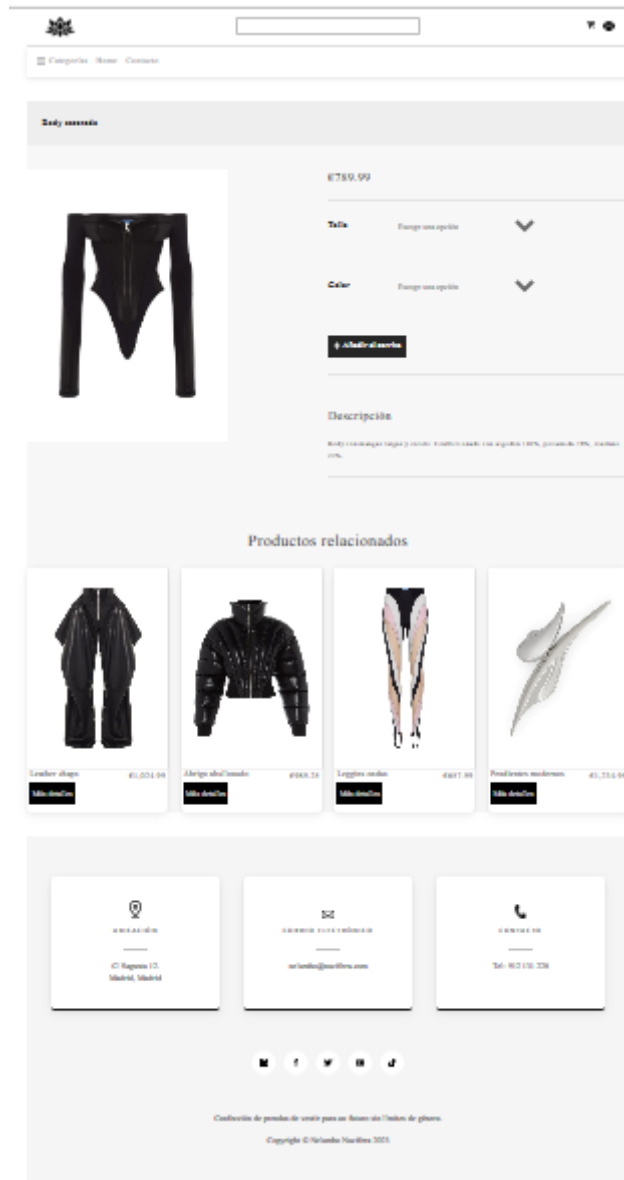
388.95 €

Más detalles

5.6. Página de producto (Pagina-producto.php).

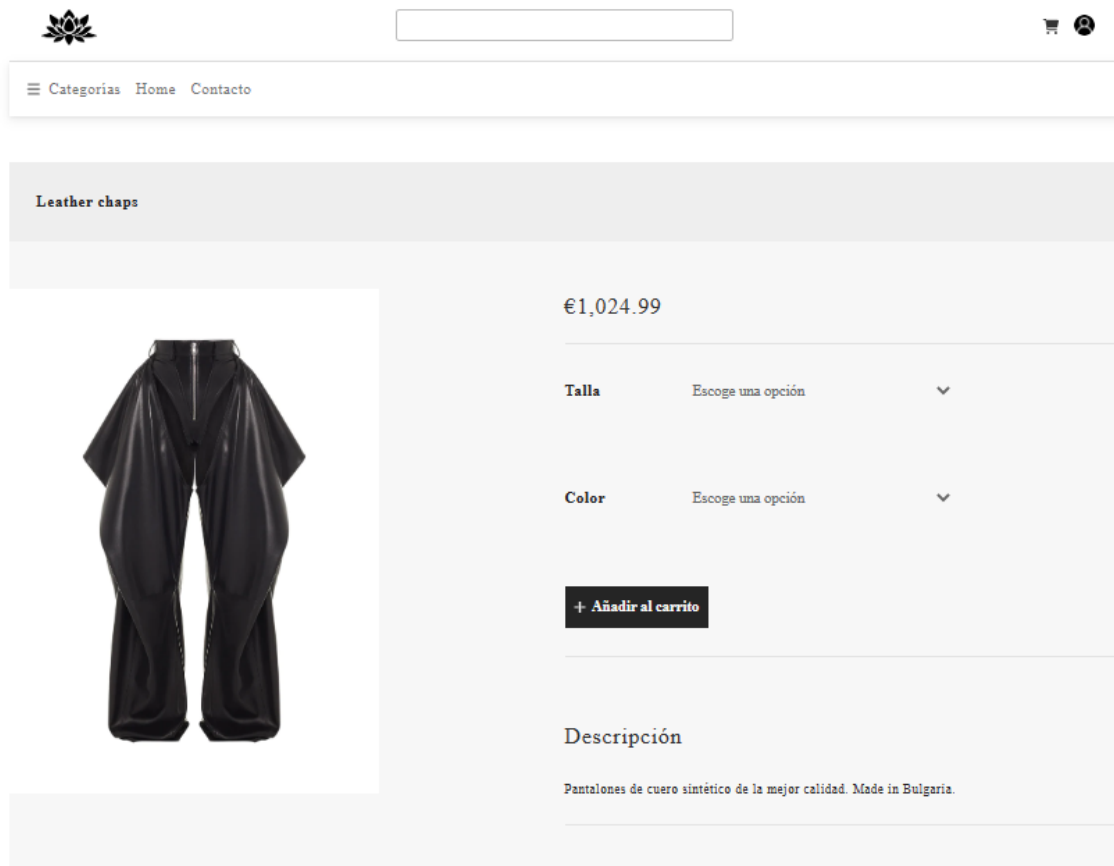
Tras pulsar sobre el botón “Más detalle” encontrado dentro de cada tarjeta de producto pasaremos a la página de producto específica del producto sobre el cual pulsamos el botón asociado.

Aparecerá la imagen principal junto a los datos del producto seleccionado permitiendo escoger la opción de los dos desplegables que encontramos a la derecha (talla y color). Cada producto tiene sus tallas y colores específicos mostrándose en este apartado. Una vez seleccionado o no (valores por defecto) se podrá pulsar en “+añadir al carrito” que sumará una unidad de dicho producto al carrito.



Debajo de la sección del producto principal tenemos más opciones a las que accederemos a través de su botón asociado, se trata de los productos asociados. Estos productos pertenecen a otras categorías y podrán servir para complementar el estilismo. Si se pulsa sobre sus botones accederemos a la misma página solo que cambiando el producto principal, siendo éste el seleccionado en última instancia en

vez del anterior (Body escotado). Por ejemplo, pulsando sobre el botón más detalles de “Leather chaps”, el primer producto de productos relacionados, se obtendrá la siguiente pantalla:



Junto a sus productos relacionados donde se puede apreciar que las categorías de producto mostrada siguen no coincidiendo con el producto principal.



5.7. Checkout (checkout.php).

Tras haber realizado la selección de uno o más artículos (botón de añadir al carrito) y se pulse sobre “Comprar” en el menú (opción desplegada tras clicar en el emoticono del carrito de la parte derecha) aparecerá una página con la lista de elementos contenidos en el carrito.

The screenshot shows the checkout page layout. At the top, there is a navigation bar with a logo, a search input field, and a shopping cart icon. Below the navigation bar is a menu with links: "Categorías", "Home", and "Contacto". The main content area displays a table of items in the cart:

Producto	Color	Talla	Precio	Cantidad	Subtotal	Eliminar
Leather chaps	Color	Talla	1,024.99€	<input type="text" value="2"/>	2,049.98€	

Below the cart table is a summary section titled "Total":

Total	
Gastos de envío	0 €
Total	€2,049.98

At the bottom of the summary section is a button labeled "Realizar pago".

En caso de que no se haya añadido nada al carrito se mostrará la lista vacía.

The screenshot shows the checkout page layout for an empty cart. The main content area displays a table with the following structure:

Producto	Color	Talla	Precio	Cantidad	Subtotal	Eliminar
Lista vacía						

Below this table is a summary section titled "Total":

Total	
Gastos de envío	0 €
Total	Lista vacía

Una de las ventajas de esta ventana es que, si queremos modificar los productos de nuestro carrito antes de realizar el pago se tiene dicha opción. Gracias a las flechas o campo editable que se encuentra debajo de “Cantidad” se podrá cambiar el número de unidades de producto que se quiera adquirir, siempre con un límite mínimo de 1 pero si se quisiese eliminar directamente un producto se puede realizar la operación pulsando sobre la papelera que se encuentra debajo de “Eliminar”. Todos los cambios en los subtotales y total se modificarán en tiempo real para mayor comodidad del comprador.

*Actualmente encontramos la posibilidad de escoger un producto gracias a su identificador pero se espera añadir una mejora futura para añadir talla y color (datos recogidos en la página de producto).

5.8. Pasarela de pago(pago.php).

Una vez se pulse el botón “Realizar pago” del checkout se mostrará una nueva pantalla con los productos seleccionados y botones correspondientes a las diferentes formas disponibles de pago. Después de elegir una de las tres opciones se podrá realizar la gestión introduciendo los datos que se requieran.

Finalmente al haber realizado el pago exitosamente volveremos a la misma pantalla de la pasarela de pago. Para comprobar que la compra se ha realizado se puede consultar en “Tus pedidos”. Si aparece se habrá realizado y sino no, pues toda compra queda registrada en la BBDD y se muestra posteriormente en esta parte.

≡ Categorías Home Contacto


Detalles de pago

Producto	Color	Talla	Subtotal
Leather chaps	Color	Talla	2,049.98€

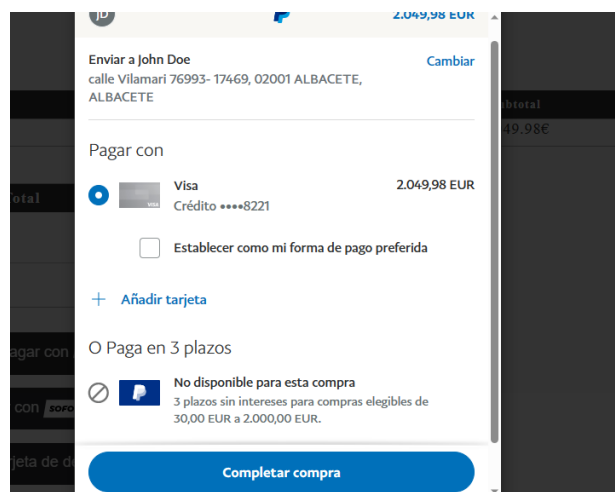
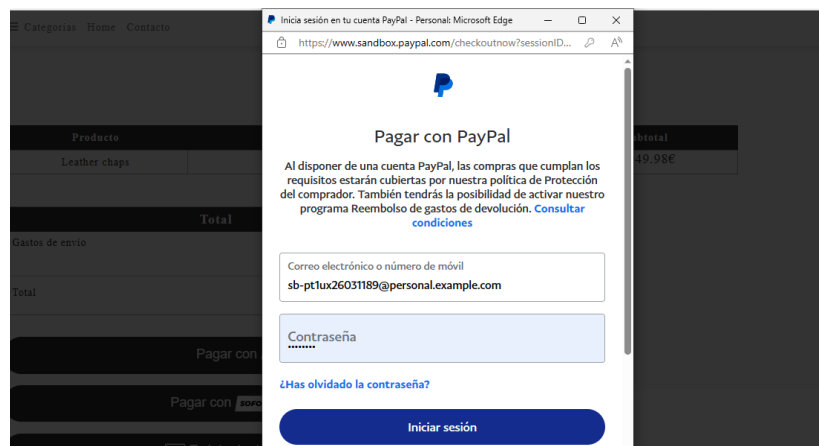
Total	
Gastos de envío	0 €
Total	€2,049.98

Pagar con **PayPal**

Pagar con **SOFORT** SOFORT

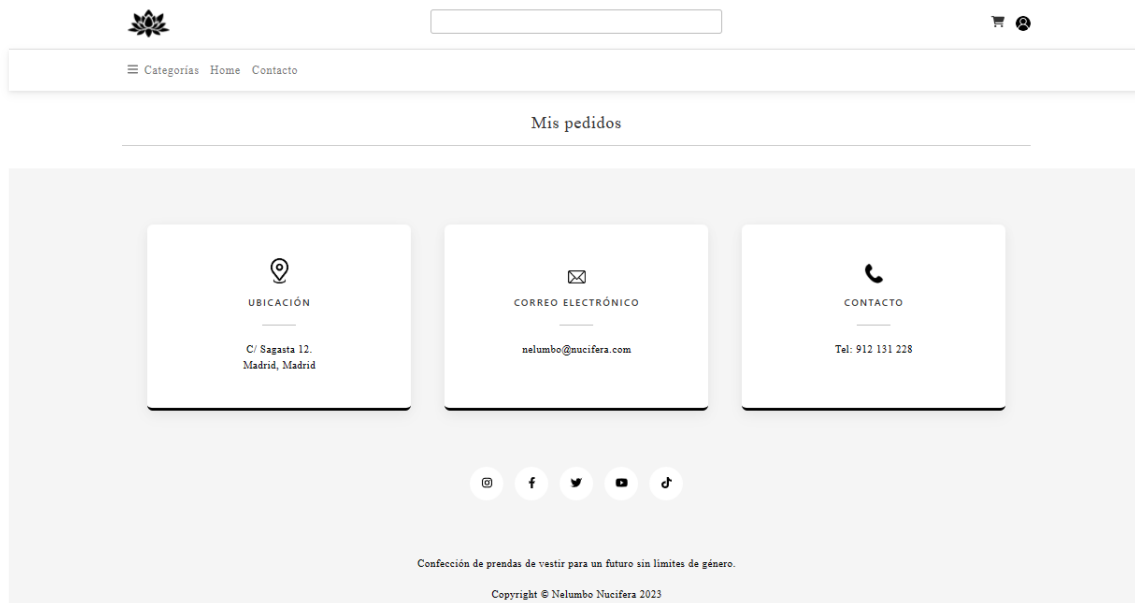
 Tarjeta de débito o crédito

Desarrollado por **PayPal**

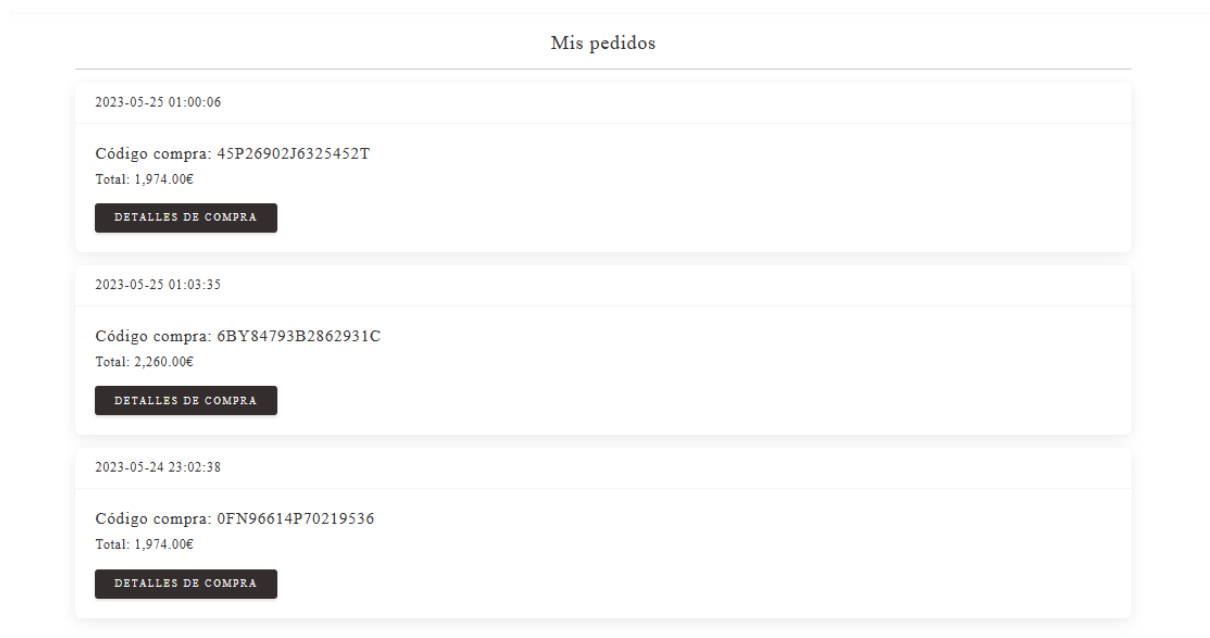


5.9. Historial de compras (historial_compras.php) y detalles de compra (detalles_compra.php).

Otra de las opciones posibles al pulsar sobre el carrito del menú (esquina derecha) es “Tus pedidos” que no es más que un historial de las compras realizadas asociadas al cliente. De darse la situación de no haber realizado ninguna compra aún aparecerá la siguiente pantalla:



En cambio, tras haber realizado alguna compra la pantalla mostrará las operaciones realizadas con su fecha, código de compra y total correspondientes.



De querer verse en detalle el contenido de la compra se podrá acceder a éste simplemente pulsando sobre “detalles de compra”, se podrá acceder entonces a detalles_compra.php, archivo que mostrará los detalles anteriores junto a los detalles de los productos adquiridos.

Detalles de la compra

Fecha: 2023-05-25 01:03:35

Código de compra: 6BY84793B2862931C

Total: 2,260.00€

Producto	Talla	Color	Precio	Cantidad	Subtotal
Leather chaps	Talla	Color	1,025.00€	2	2,050.00€
Pendientes modernos	Talla	Color	1,235.00€	1	1,235.00€

6. Conclusiones

La realización de una página web para compra de productos online puede resultar dificultosa si se añaden varias prestaciones. Posiblemente lograr adquirir una imagen acorde a la marca puede resultar menos complicado en cuanto a codificar gracias al contenido de los diferentes productos que incluye la marca sobre la cual se basa la página que nos permite tener una guía en lo referente a la interfaz pero dar por finalizado dicho diseño y asegurarse de que la página web sea segura utilizando los elementos adecuados (keys, bind_param para evitar inyecciones, trim(), etc) puede resultar menos alentador pues, por un lado, se pierde tiempo en caso de indecisión y, por otro, se pueden encontrar errores derivados de la validación de tokens que retrasen la entrega de la tarea en fecha, un ejemplo de ello fue una de las primeras validaciones que se realizó que resultó ser solucionable con la función trim() pues, por alguna razón, sin ésta, comprobar la veracidad de la igualdad de dos códigos resultaba imposible.

Además, tras haber realizado el proyecto puede entenderse que el tiempo y la organización correcta del orden de tareas por áreas específicas (estructura de base de datos, diseño y desarrollo) es esencial para cumplir con la totalidad de los objetivos fijados o al menos con la mayor parte de los mismos.

Debido a que no se tomaron las mejores decisiones de organización para la elaboración del proyecto no se pudieron incluir algunas partes como la talla y el color debido a que inicialmente no se había codificado incluyendo ambas características ni en la base de datos ni en el mismo código por lo que se ha decidido incluirlas de forma hipotética, mostrarlas incluidas en la base de datos y asociadas a algunos productos a la hora de mostrar detalles concretos de los mismos. Fuera de mostrarse en los desplegables no se incluyen en el almacenamiento del carrito pues la dificultad requería de algo más de tiempo de desarrollo para acabar adaptándolo adecuadamente sin que el resultado fuese inestable.

Aún con estas pequeñas dificultades el sistema de compra funciona perfectamente pudiendo asociarse a un producto con precio y cantidad realizando la transacción y registrando dicha operación en la tabla de BBDD correspondiente para posteriores consultas o el mismo procedimiento de adquirir datos por formulario para validar datos registrando usuarios y/o iniciando sesión de usuario así como un diseño de interfaz agradable a la vista y que no sufre deformación alguna en las diferentes pantallas manteniendo la funcionalidad completa en todo momento.

7. Bibliografía y webgrafía.

1. Manuales bindParam.

<https://manuales.guebs.com/php/pdostatement.bindparam.html>
<https://es.stackoverflow.com/questions/59971/php-cu%C3%A1l-es-la-diferencia-entre-bindparam-y-bindvalue-en-pdo>
https://www.youtube.com/watch?v=jpdhO_9B76Y
<https://www.youtube.com/watch?v=QQ3x9-pViQE>
<https://stackoverflow.com/questions/34869498/php-passing-array-to-pdo-bindparam>
<https://www.php.net/manual/es/pdostatement.bindparam.php>
<http://www.sellosmoreno.com/>
<https://learn.microsoft.com/es-es/sql/connect/php/pdostatement-bindparam?view=sql-server-ver16>
<https://www.php.net/manual/es/pdo.constants.php>

2. Bootstrap y diseño css.

<https://getbootstrap.com/>
<https://startbootstrap.com/themes>
<https://getbootstrap.com/docs/5.0/components/card/>
<https://getbootstrap.com/docs/4.0/components/alerts/>
<https://www.youtube.com/watch?v=S7UFRKljX0A>
<https://bbbootstrap.com/snippets/bootstrap-4-simple-catalog-template-59154867>
<https://www.bootdey.com/snippets/tagged/catalogs>
<https://mdbootstrap.com/docs/standard/extended/gallery/>
<https://developer.mozilla.org/es/docs/Web/CSS>
<https://www.youtube.com/watch?v=F-3cvd10zHE>
<https://developer.mozilla.org/es/docs/Web/JavaScript>
<https://es.stackoverflow.com/questions/274590/incrementar-valor-de-varios-inputs-con-botones-m%C3%A1s-y-menos-en-html-con-javascript>

3. Seguridad y hash_hmac.

<https://www.youtube.com/watch?v=pFYxSPvyUEA>
<https://es.wikipedia.org/wiki/HMAC>
<https://www.php.net/manual/es/function.hash-hmac.php>
<http://www.sellosmoreno.com/>

4. Errores y detalles sobre PHP, JS y SQL.

<https://desarrolloweb.com/articulos/318.php>
<https://stackoverflow.com/questions/22193390/sqlstate42s22-column-not-found-1054-unknown-column-array-in-field-list>
<https://stackoverflow.com/questions/10129120/sqlstate42s22-column-not-found-1054-unknown-column>
<https://developer.mozilla.org/es/docs/Web/JavaScript>
https://fmhelp.filemaker.com/help/18/fmp/es/index.html#page/FMP_Help%2Fone-to-many-relationships.html%23

5. Información sobre ámbitos ajenos a la programación. Moda y e-commerce.
<https://www.federicoasorey.es/2019/11/vender-por-internet-principales-obligaciones-legales/>
<https://www.mugler.es/es/perfume-y-fragancia>
<https://www.vogue.es/moda/modapedia/disenadores/iris-van-herpen/513>
<https://enriqueortegaburgos.com/>
<https://app.creately.com/d/yQjaIRhBeS3/edit/s/S7RajCgzkip>
<https://es.fashionnetwork.com/news/La-alta-costura-de-paris-un-negocio-vibrante-y-multimillonario-comenzara-el-lunes-su-proxima-temporada.1478398.html>
<https://www.elmundo.es/yodona/2014/07/08/53ba8538268e3e3d288b4583.html>
<https://enriqueortegaburgos.com/e-commerce-y-moda-i-cifras-y-tendencias-del-sector-en-espana-y-el-mundo/>
6. Funcionamiento e implementación de Paypal.
<https://www.baulphp.com/paypal-php-integracion-con-ejemplo-completo/>
<https://www.youtube.com/watch?v=EdGmUHW5Fdk>
<https://www.youtube.com/watch?v=nAz8xRQaPZQ>
<https://developer.paypal.com/home>
<https://developer.paypal.com/sdk/js/configuration/>
<https://developer.paypal.com/demo/checkout/#/pattern/server>

8. Anexos.

8.1. Detalles sobre Hash_hmac.

El cálculo del HMAC se realiza utilizando la función `hash_hmac` en PHP. Esta función acepta diferentes parámetros, como el algoritmo hash, los datos, la clave y un parámetro opcional para el formato de salida.

En primer lugar, es necesario seleccionar el algoritmo hash adecuado según los requisitos de seguridad y compatibilidad. PHP ofrece diversos algoritmos hash, como MD5, SHA-256, SHA-512, entre otros. Sin embargo, se recomienda optar por algoritmos más seguros, como SHA-256 o SHA-512.

Posteriormente, se deben tener los datos sobre los cuales se calculará el HMAC. Estos datos pueden ser una cadena de texto o un flujo de bytes. Es importante asegurarse de que los datos sean consistentes y estén en el formato correcto antes de realizar el cálculo del HMAC.

A continuación, se requiere una clave secreta que será utilizada por el algoritmo HMAC. La clave también puede ser una cadena de texto o un flujo de bytes. Es fundamental mantener esta clave de manera segura y evitar compartirla con personas no autorizadas.

Por último, se utiliza la función `hash_hmac` para calcular el HMAC. A continuación se muestra un ejemplo:

```
$data = "Hello, world!";  
$key = "secret_key";  
$algorithm = "sha256";  
  
$hash = hash_hmac($algorithm, $data, $key);  
  
echo $hash;
```

En el ejemplo proporcionado, se realiza el cálculo del HMAC utilizando el algoritmo SHA-256, los datos "Hello, world!" y la clave "secret_key". El resultado se guarda en la variable `$hash` y se muestra en la pantalla.

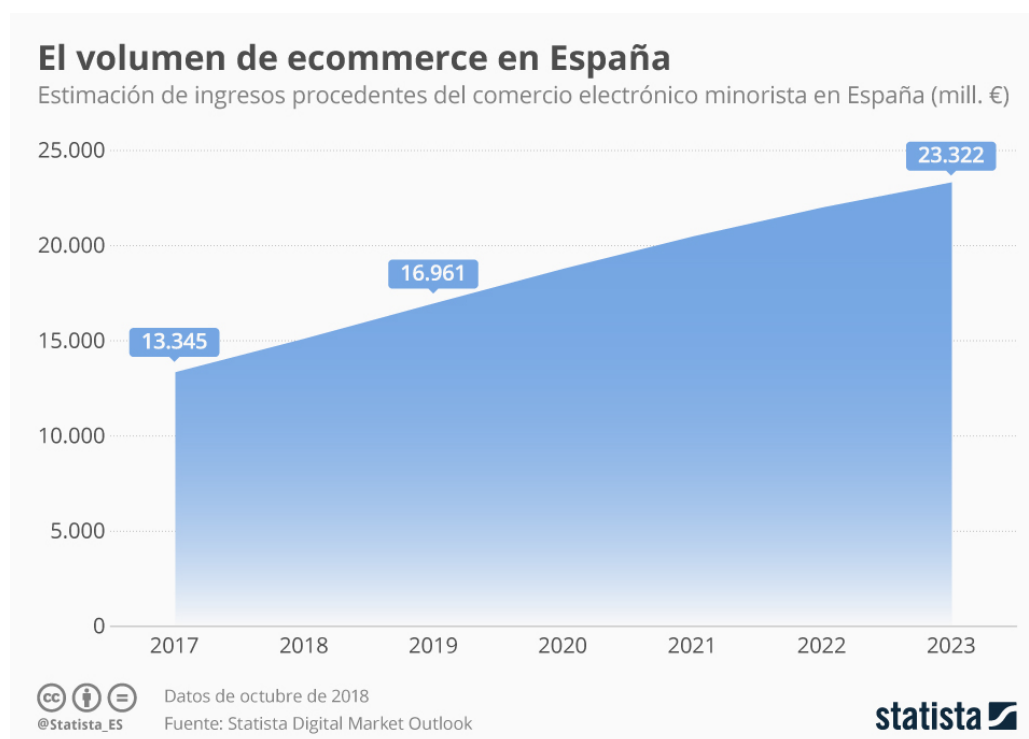
Es importante tener en cuenta que el HMAC genera un valor único y específico para los datos y la clave utilizados. Si se modifican los datos o la clave, el resultado del HMAC será completamente diferente. De esta manera, al verificar la integridad de los datos, se puede calcular nuevamente el HMAC y compararlo con el HMAC original para detectar modificaciones no autorizadas.

Es necesario tener en cuenta que el HMAC no proporciona confidencialidad, ya que cualquier persona puede calcularlo utilizando los mismos datos y la misma clave. Si se requiere confidencialidad, es necesario utilizar algoritmos de cifrado adicionales junto con HMAC.

En resumen, el HMAC en PHP es una herramienta útil para garantizar la integridad y autenticidad de los datos. Al combinar una función hash con una clave secreta, se genera un código de autenticación que permite verificar si los datos han sido modificados o falsificados y si provienen de una fuente confiable. Es importante elegir algoritmos hash seguros y mantener la clave secreta de manera segura.

8.3. Gráficos de la evolución de ventas derivadas de ecommerce.

De 2017 a 2023 se puede apreciar un incremento exponencial en los ingresos resultantes del medio en línea como se indica en este gráfico de *statista*.



En la siguiente tabla se refleja como en 2018 España entró en el top 10 de países con mayores ingresos gracias a la utilización del e-commerce. Tabla realizada por Enrique Ortega Burgos.

Posición	País	Total de ventas e-commerce (millones de \$)	Proporción del total de ventas de e-commerce en el PIB (%)	Ventas e-commerce B2B (millones de \$)	Proporción de e-commerce B2B en las ventas totales de e-commerce (%)	Ventas e-commerce B2C (millones de \$)
1	Estados Unidos	8.640.000	42	7.542.000	87	1.098.000
2	Japón	3.280.000	66	3.117.000	95	163.000
3	China	2.304.000	17	943.000	41	1.361.000
4	Corea del Sur	1.364.000	84	1.263.000	93	102.000
5	Reino Unido	918.000	32	652.000	71	266.000
6	Francia	807.000	29	687.000	85	121.000
7	Alemania	722.000	18	620.000	86	101.000
8	Italia	394.000	19	362.000	92	32.000
9	Australia	348.000	24	326.000	94	21.000
10	España	333.000	23	261.000	78	72.000
	TOTAL	19.110.000	35	15.773.000	83	3.337.000
	TOTAL MUNDIAL	25.648.000	30	21.258.000		4.390.000