

**UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN**



ESTUDIANTES:

Moisés Mozo García

Samuel Ernesto Beltrán Castillo

Diana Carolina Alvarado Mejía

Gerardo Alfonso Rivas Barrios

ASIGNATURA: Base de Datos 1

DOCENTE: Pedro Trejo Noble.

CICLO/AÑO: 1/2023

Introducción

En el presente proyecto pretendemos realizar un sistema de compra y venta de carne abordando como problemática principal el desarrollo de un sistema de compra y venta para este negocio.

Un sistema de compra y venta consiste en un conjunto de normas y seguimientos que utilizan las empresas para conocer la cantidad exacta productos que se disponen. Para crear dicho sistema usaremos el modelo vista controlador con el apoyo de Visual Studio Code para la estructuración del programa.

En esta etapa del proyecto avanzaremos con el modelado y la creación de la base de datos que todas las compras se registren sin pérdidas de información.

ÍNDICE

DESARROLLO	4
OBJETIVO GENERAL	4
OBJETIVOS ESPECIFICO	4
JUSTIFICACIÓN	4
PROBLEMÁTICA A RESOLVER	5
TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO DEL PROYECTO	6
DIAGRAMAS.....	12
CONCLUSIONES	14
ANEXOS.....	15
MANUALES.....	15

DESARROLLO

SISTEMA DE COMPRA Y VENTA DE CARNE (CARNICERÍA)

OBJETIVO GENERAL

- Implementar un sistema de compra y venta eficiente capaz de controlar y gestionar de manera óptima el flujo de inventario en la empresa

OBJETIVOS ESPECIFICO

- Mejorar la gestión de los recursos internos.
- Reducir los costos asociados a la falta o exceso de inventario.

Justificación

Un sistema de compra y venta agiliza los trámites dentro de una organización al mantener de forma ordenada la cantidad exacta de los productos de los cuales se dispone en tiempo real, flexibilizando la operación, mantener un stock es de suma importancia ya que de ello depende mantener la venta de un producto o cubrir la demanda que los clientes requieren, además de esto ayudan a evitar el exceso de compras disminuyendo pérdidas de productos que no son muy demandados y que estos se queden estancados.

Es necesario llevar este control virtual de los recursos físicos que se manejan dentro del negocio o quiosco para evitar gastos innecesarios en tiempo, recurso humano y financiero, cómo equipo de trabajo notamos la escasez de sistemas empleados objetivamente en carnicerías que como negocios rentables lo necesitan ya que sus productos tienen muchas clasificaciones cómo carnes y despojos frescos (refrigerados o congelados), con o sin hueso, en sus diferentes modalidades (fileteado, troceado, picado, mechado y otras análogas, según se trate) etc.

Dada la necesidad de las tiendas de llevar este registro interno de su inventario para que el control sea más estructurado y organizado ayudando tanto a sus encargados como empleados a agilizar este proceso, nos vemos en el deber de crear un sistema de inventario que puede ser usado para cualquier tipo de negocio, pero en este caso especificado para una carnicería

PROBLEMÁTICA A RESOLVER

Tema: Sistema de compra y venta de carnicería

Problemática a resolver

El sistema de compra y venta busca facilitar la organización de las actividades y tareas de los trabajadores de la carnicería además de ordenar y jerarquizar sus productos para que de esta forma facilite la contabilidad de los productos, con este sistema de compra y venta también se benefician los trabajadores tanto administrativos como los carniceros, porque les da una idea más clara de sus productos y cantidades más exactas de manera directa y organizada que facilita el trabajo dentro de la tienda. Como también es importante mantener actualizados los precios de los productos, ya que estos pueden tener un cambio en el costo de la materia prima o en la competencia. Con nuestro sistema de carnicería los precios se actualizarán de manera eficiente y precisa. La carnicería debe garantizar que los productos que vende sean de alta calidad y estén en buen estado, por lo tanto, es esencial contar con un sistema de control de calidad adecuado, que permita a los empleados y dueño verificar la calidad de los productos entrantes y salientes, monitorear las fechas de caducidad y llevar un registro de los productos devueltos por el cliente. Finalmente, la seguridad es una preocupación importante en cualquier sistema de inventario, ya que los datos del inventario y los registros de ventas pueden ser sensibles y confidenciales. Por lo tanto, es fundamental que el sistema tenga medidas de seguridad adecuadas para proteger la información del cliente y la empresa.

Tecnologías a utilizar para el desarrollo del proyecto

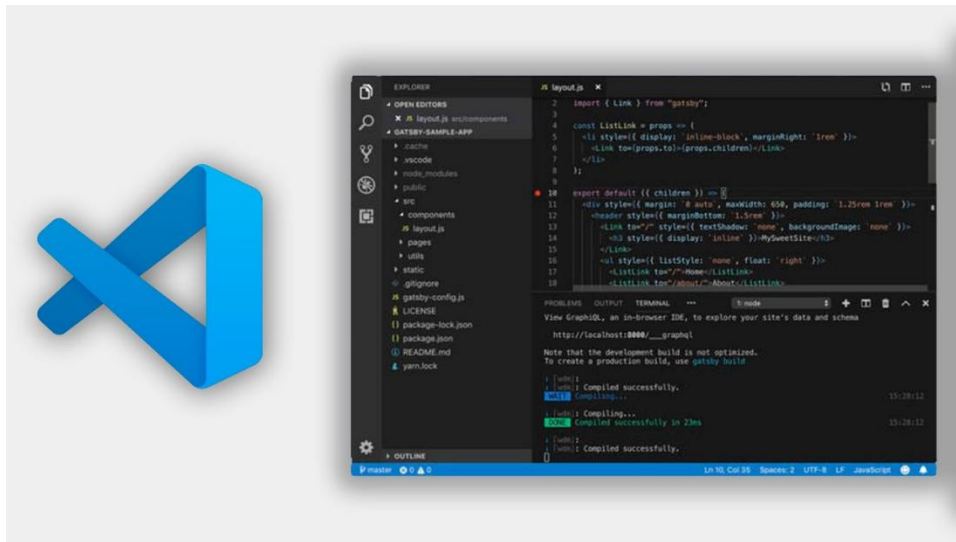
App Serv

App Serv es una plataforma de alojamiento web y aplicación web gratuita desarrollada por Microsoft. Con App Serv, se puede alojar y administrar aplicaciones web y móviles en la nube de Microsoft Azure. Ofrece un conjunto de características y herramientas que permiten crear y alojar aplicaciones web y móviles escalables y de alta disponibilidad. App Serv admite una variedad de tecnologías de desarrollo web, como ASP.NET, PHP, Node.js, Python y Ruby, lo que permite crear aplicaciones utilizando estas herramientas de desarrollo tan conocidas. Además, App Serv ofrece una amplia variedad de herramientas de desarrollo integradas, como Visual Studio, que permiten crear aplicaciones de manera rápida y eficiente. App Serv también admite la integración con bases de datos y sistemas de almacenamiento en la nube.



Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft que está disponible de forma gratuita y se puede utilizar en múltiples sistemas operativos, como Windows, macOS y Linux. Es una herramienta muy popular entre los desarrolladores debido a su interfaz intuitiva, sus características personalizables y la amplia gama de extensiones disponibles. Es un editor de código fuente altamente personalizable y fácil de usar que ofrece una amplia variedad de características y herramientas que facilitan el desarrollo de software.



MySQL-SQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto. Fue desarrollado por la compañía sueca MySQL AB y ahora es propiedad de Oracle Corporation. MySQL es uno de los sistemas de gestión de bases de datos más populares en el mundo y es ampliamente utilizado por desarrolladores web y empresas de todos los tamaños. MySQL utiliza el lenguaje SQL (Structured Query Language) para administrar y manipular los datos almacenados en una base de datos. Ofrece una amplia variedad de características y herramientas para la gestión de bases de datos, como la creación de tablas, la definición de relaciones entre tablas, la consulta y actualización de datos, y la administración de usuarios y permisos de acceso. MySQL es un sistema de gestión de bases de datos relacional de código abierto muy popular que ofrece una amplia variedad de características y herramientas para la gestión de datos empresariales.

SQL (Structured Query Language) es un lenguaje de programación utilizado para administrar y manipular bases de datos relacionales. SQL es ampliamente utilizado por desarrolladores web y empresas de todo el mundo para interactuar con bases de datos y realizar tareas como la consulta, actualización y eliminación de datos. SQL se utiliza para comunicarse con sistemas de gestión de bases de datos relacionales, como MySQL, PostgreSQL, Microsoft SQL Server, Oracle, entre otros. SQL permite a los usuarios interactuar con los datos almacenados en la base de datos y realizar tareas complejas como la unión de datos de múltiples tablas, la agregación de datos y la creación de procedimientos almacenados.



PHP

PHP es un lenguaje de programación de código abierto utilizado principalmente para la creación de sitios web y aplicaciones web dinámicas. Fue desarrollado originalmente por Rasmus Lerdorf en 1995 y ha evolucionado a lo largo de los años con la contribución de una gran comunidad de desarrolladores.

PHP es un lenguaje de programación del lado del servidor, lo que significa que se ejecuta en el servidor web y genera código HTML que se envía al navegador del usuario. PHP se puede utilizar para crear aplicaciones web dinámicas, como foros, carritos de compra, blogs, redes sociales y muchas otras aplicaciones. PHP es un lenguaje de programación de código abierto utilizado principalmente para la creación de sitios web y aplicaciones web dinámicas. PHP es fácil de usar, flexible, se integra bien con bases de datos, es compatible con múltiples plataformas y tiene una gran comunidad de desarrolladores.



HTML y CSS

HTML (HyperText Markup Language) se utiliza para crear la estructura de una página web, mientras que CSS (Cascading Style Sheets) se utiliza para definir el aspecto visual de la página.

HTML es un lenguaje de marcado que se utiliza para definir la estructura de una página web. Con HTML, se pueden crear encabezados, párrafos, listas, tablas, formularios y otros elementos que componen una página web. HTML utiliza etiquetas para definir estos elementos y atributos para especificar sus propiedades.

CSS es un lenguaje de estilo que se utiliza para definir la apariencia visual de una página web. Con CSS, se pueden definir el color, la fuente, el tamaño y la posición de los elementos de una página web. CSS utiliza selectores para identificar los elementos a los que se aplicarán los estilos y propiedades para definir cómo se verán esos elementos.



Bootstrap

Bootstrap es un marco de diseño front-end de código abierto creado por Twitter para ayudar a los desarrolladores web a crear sitios web y aplicaciones móviles responsivas y modernas. Bootstrap proporciona un conjunto de herramientas y estilos CSS predefinidos que se pueden utilizar para crear rápidamente páginas web y aplicaciones móviles con un aspecto y sensación consistentes.

Bootstrap incluye una amplia variedad de componentes, como botones, formularios, tablas, navegación, carruseles, alertas y mucho más. Además, Bootstrap también incluye una cuadrícula de diseño sensible que se puede utilizar para crear diseños de página personalizados que se adaptan automáticamente a diferentes tamaños de pantalla.



Diagramas

Diagrama de clases

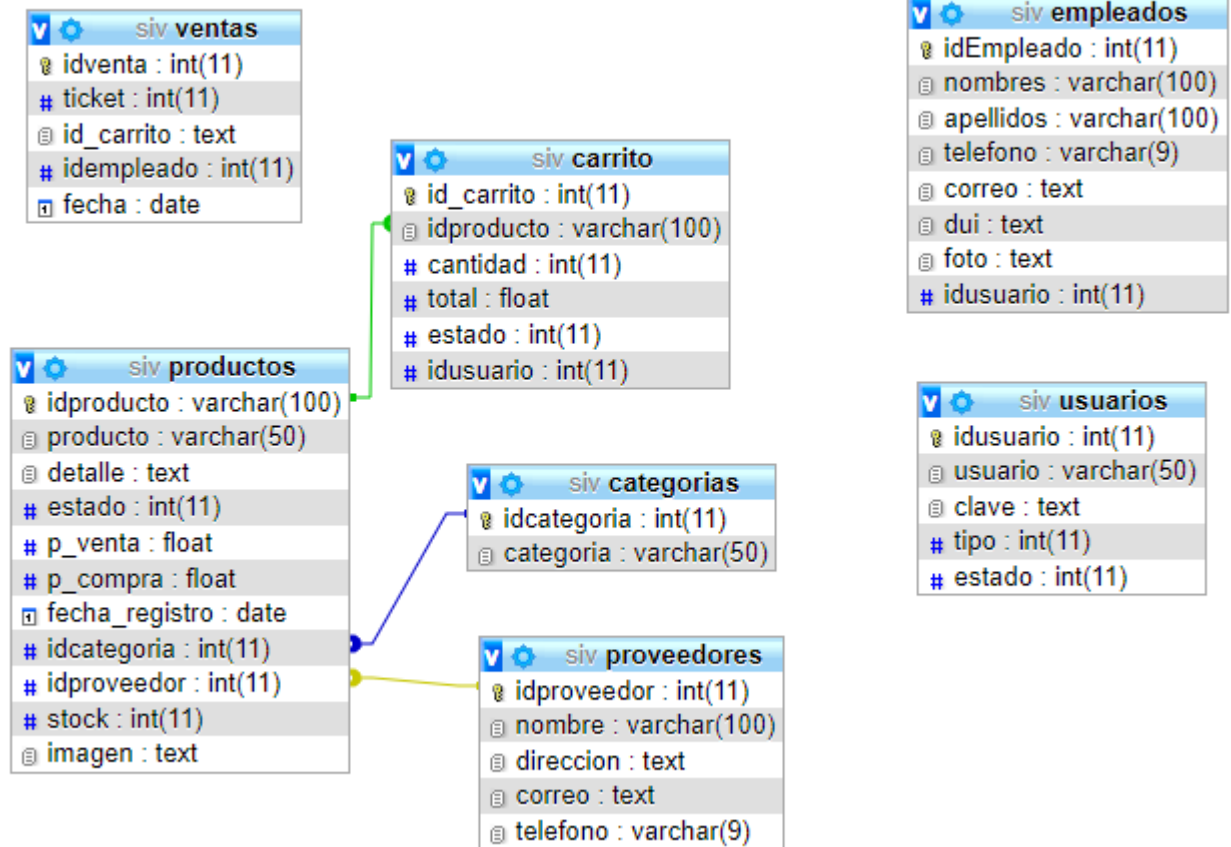
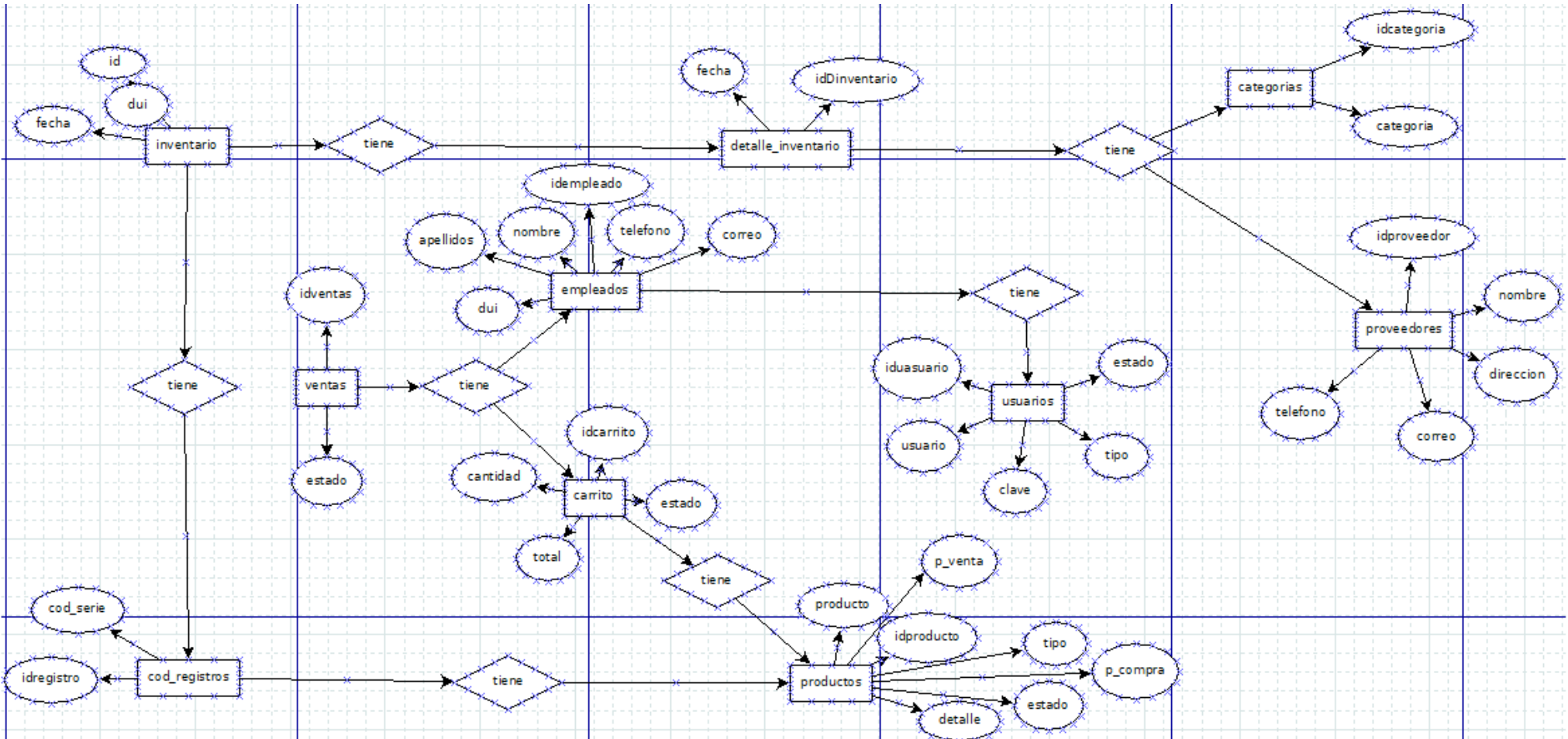


Diagrama entidad relación



Conclusiones

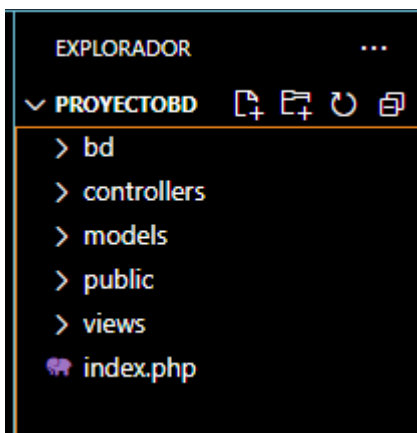
La implementación de una base de datos para un proyecto de compra y venta de carne en una página web permite la gestión eficiente y automatizada del inventario, lo que conlleva a una mejora en la gestión del negocio. Los datos almacenados en la base de datos pueden ser utilizados para analizar la demanda y las preferencias de los clientes, así como también para monitorear el stock de los productos y generar reportes financieros. Además, la utilización de una base de datos facilita la integración con otros sistemas, lo que puede ampliar las posibilidades de venta y mejorar la experiencia de compra del cliente.

Una página web con una base de datos bien estructurada y mantenida, puede generar un mayor nivel de confianza y satisfacción en los clientes. La disponibilidad de información clara y actualizada sobre los productos, su origen y su calidad, así como también el uso de herramientas de pago seguras, son elementos clave para fomentar la fidelización de los clientes y el crecimiento del negocio. Además, el uso de la tecnología para la venta de productos de carne en línea puede contribuir a la reducción de los costos operativos y mejorar la eficiencia en la gestión del negocio.

Anexos

Manuales

Modelo-vista-controlador



Carpeta “bd”: En esta carpeta se encuentra la base de datos del proyecto.

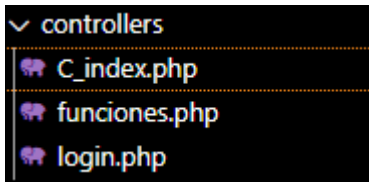
```
ss.sql x
bd > ss.sql
22 -- Base de datos: `ss`
23 --
24
25 -----
26
27 --
28 -- Estructura de tabla para la tabla `carrito`
29 --
30
31 CREATE TABLE `carrito` (
32   `idcarrito` int(11) NOT NULL,
33   `idproducto` int(11) DEFAULT NULL,
34   `cantidad` int(11) DEFAULT NULL,
35   `total` int(11) DEFAULT NULL,
36   `estado` int(11) DEFAULT NULL
37 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
38
39 -----
40
41 --
42 -- Estructura de tabla para la tabla `categorias`
43 --
44
45 CREATE TABLE `categorias` (
46   `idcategoria` int(11) NOT NULL,
47   `categoria` text
48 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
49
50 -----
51
52 --
53 -- Estructura de tabla para la tabla `cod_registros`
54 --
55
56 CREATE TABLE `cod_registros` (
57   `idregistro` int(11) NOT NULL,
58   `cod_serie` varchar(25) DEFAULT NULL
59 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
60
```

Carpeta “controllers”: En esta carpeta contiene tres archivos php

C_index.php: Este archivo contiene la clase llamada “index”, con una función publica para poder llamar a la base de datos siempre que nosotros queramos.

Funciones. php: Este archivo contiene las funciones que utilizaremos para el proyecto, algunas de estas son la funciones “Crud” y “MaxID”. Ambas contienen consultas que ayudan a facilitar el proyecto al momento de programar.

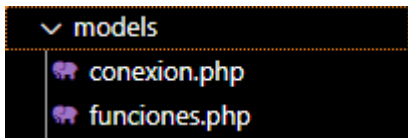
Login.php: Este archivo contiene el inicio de sesión y la validación para registrar un usuario.



Carpeta “models”: Esta carpeta contiene 3 archivos php

Conexión.php: En este archivo se encuentra la conexión a la base de datos alojada en phpmyadmin.

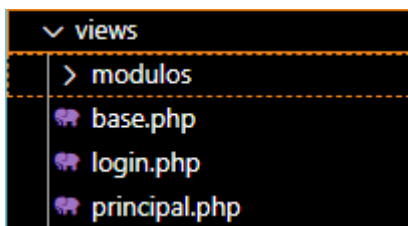
Funciones.php: Este archivo contiene otras funciones que van de la mano con el archivo “funciones”, ubicado en “controllers”.



Carpeta “public”: En ella hay carpetas con “frameworks”, carpeta “CSS” donde esta nuestra hoja de estilos para el diseño de nuestro proyecto, una carpeta llamada “JS” que contiene funciones con JavaScript, para el desarrollo de nuestro proyecto y la carpeta “IMG” donde están alojadas las imágenes a utilizar.

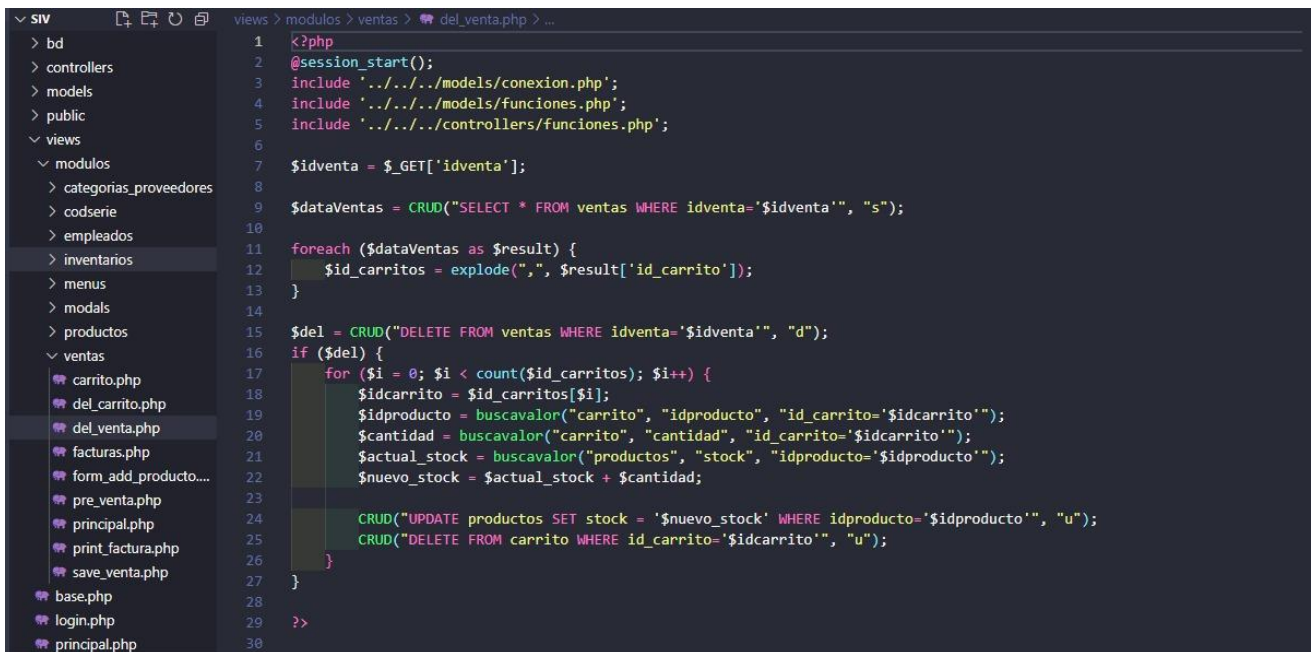
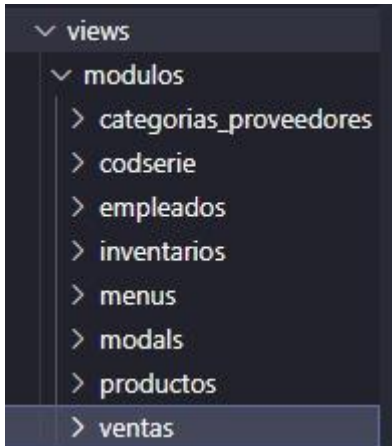


Carpeta “views”: Esta carpeta contiene las vistas de nuestra página web.



Carpeta “modulos” que se encuentra dentro de la carpeta “views”

Esta carpeta contiene el resto de vistas de nuestro proyecto, algunas de ellas son el “navbar” que es nuestra barra para navegar en la página web. También se encuentran los archivos para “crear”, “eliminar” y “actualizar”, esto funciona dentro de los formularios que contiene nuestra pagina web para agregar, eliminar y editar productos e inventarios.



```
views > modulos > ventas > del_venta.php > ...
1 <?php
2 @session_start();
3 include '../../../models/conexion.php';
4 include '../../../models/funciones.php';
5 include '../../../controllers/funciones.php';
6
7 $idventa = $_GET['idventa'];
8
9 $dataVentas = CRUD("SELECT * FROM ventas WHERE idventa='$idventa'", "s");
10
11 foreach ($dataVentas as $result) {
12     $id_carritos = explode(",", $result['id_carrito']);
13 }
14
15 $del = CRUD("DELETE FROM ventas WHERE idventa='$idventa'", "d");
16 if ($del) {
17     for ($i = 0; $i < count($id_carritos); $i++) {
18         $idcarrito = $id_carritos[$i];
19         $idproducto = buscavalor("carrito", "idproducto", "id_carrito='$idcarrito'");
20         $cantidad = buscavalor("carrito", "cantidad", "id_carrito='$idcarrito'");
21         $actual_stock = buscavalor("productos", "stock", "idproducto='$idproducto'");
22         $nuevo_stock = $actual_stock + $cantidad;
23
24         CRUD("UPDATE productos SET stock = '$nuevo_stock' WHERE idproducto='$idproducto'", "u");
25         CRUD("DELETE FROM carrito WHERE id_carrito='$idcarrito'", "u");
26     }
27 }
28
29 >>
30
```

Query para eliminar datos/productos

```
del.php x
views > modulos > codserie > del.php
1 <?php
2 @session_start();
3 include '../models/conexion.php';
4 include '../models/funciones.php';
5 include '../controllers/funciones.php';
6
7 $codserie = $_GET['codserie'];
8 $del = CRUD("DELETE FROM cod_registros WHERE idregistro='$codserie'", "d");
9 ?>
10 <?php if($del):?>
11     <script>
12         $(document).ready(function(){
13             alertify.success("Producto eliminado");
14             $("#Data-Panel-1").load("./views/modulos/codserie/principal.php");
15             return false;
16         });
17     </script>
18 <?php else:?>
19     <script>
20         alertify.error("Error al eliminar producto");
21         $("#Data-Panel-1").load("./views/modulos/codserie/principal.php");
22         return false;
23     </script>
24
25 <?php endif:?>
```

Query para actualizar/modificar datos/productos

```
update.php X
views > modulos > codserie > update.php
1 <?php
2 @session_start();
3 include '../../models/conexion.php';
4 include '../../models/funciones.php';
5 include '../../controllers/funciones.php';
6
7 $idregistro = $_POST['idregistro'];
8 $cod_serie = $_POST['cod_serie'];
9
10 $tabla = "cod_registros";
11 $campos = "cod_serie= '$cod_serie'";
12 $condicion = "idregistro= '$idregistro'";
13
14 $update = CRUD("UPDATE $tabla SET $campos WHERE $condicion","u");
15 ?>
16 <?php if($update):?>
17     <script>
18         $(document).ready(function(){
19             alertify.success("Producto Actualizado");
20             $("#Data-Panel-1").load("../views/modulos/codserie/principal.php");
21             return false;
22         });
23     </script>
24 <?php else:?>
25     <script>
26         alertify.error("Error al actualizar producto");
27         $("#Data-Panel-1").load("../views/modulos/codserie/principal.php");
28         return false;
29     </script>
30
31 <?php endif;?>
```

Query para insertar/agregar datos/productos

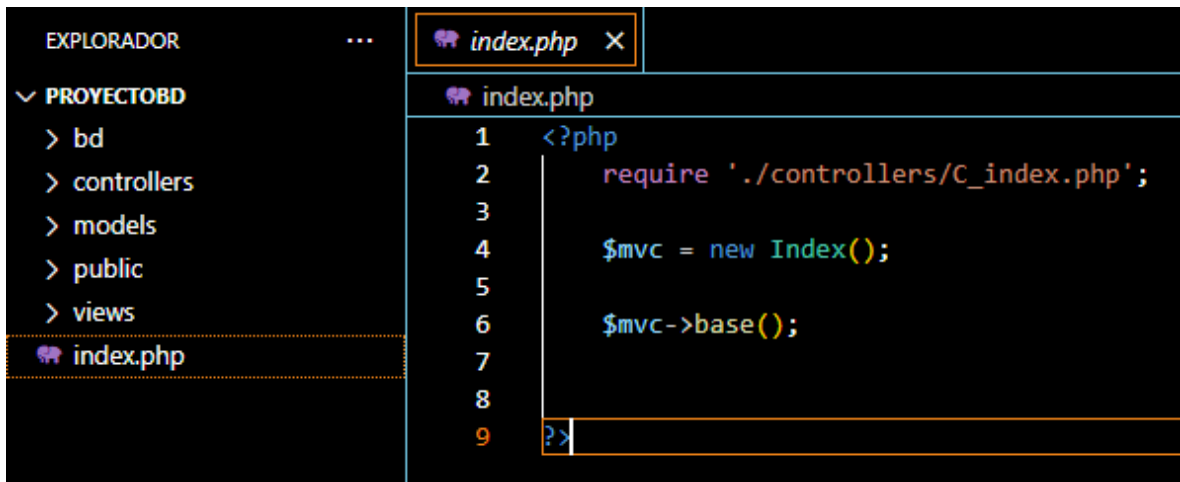
```
insert.php ×
views > modulos > codserie > insert.php
1  <?php
2  @session_start();
3  include '../models/conexion.php';
4  include '../models/funciones.php';
5  include '../controllers/funciones.php';
6
7  //$obj = new ConexionBD();
8  //$conexion = $obj->get_conexion();
9
10 echo $maxid = MaxID("SELECT MAX(idRegistro) FROM cod_registros")+1;
11
12 $cod_serie = $_POST['codserie'];
13
14 $tabla = "cod_registros";
15 $campos = "idRegistro, cod_serie";
16 $valores = "$maxid, '$cod_serie'";
17
18 $insert = CRUD("INSERT INTO $tabla ($campos) VALUES ($valores)","i");
19 ?>
20 <?php if($insert):?>
21     <script>
22         $(document).ready(function(){
23             alertify.success("Datos Registrados");
24             $("#Data-Panel-1").load("../views/modulos/codserie/principal.php");
25             return false;
26         });
27     </script>
28 <?php else:?>
29     <script>
30         alertify.error("Error al registrar datos");
31         $("#Data-Panel-1").load("../views/modulos/codserie/principal.php");
32         return false;
33     </script>
34
35 <?php endif;?>
```

```

principal.php x
views > modulos > codserie > principal.php
22 } else {
23 | $inicio = ($pagina - 1) * $registros;
24 | }
25
26 $query = "SELECT * FROM cod_registros";
27 $dataCodSerie = CRUD("SELECT * FROM cod_registros LIMIT $inicio,$registros", "s");
28 $num_registro = CountReg($query);
29 $paginas = ceil($num_registro / $registros);
30 $cont = 0;
31 ?>
32 <script src="./public/js/funciones-codSerie.js"></script>
33 <div class="row">
34 | <div class="col-md-3">
35 | | <a href="" id="BtnNewCod" class="btn btn-primary">Nuevo Producto</a>
36 | </div>
37 | <div class="col-md-9" id="Panel-Cod-Serie">
38 | | <table class="table table-striped table-bordered table-borderless" style="margin:0 auto;width:50%; margin-bottom:50px;">
39 | | | <thead class="centrado">
40 | | | | <tr>
41 | | | | | <th>Nº</th>
42 | | | | | <th>Producto</th>
43 | | | | | <th colspan="2">Acciones</th>
44 | | | | </tr>
45 | | | </thead>
46 | | | <tbody>
47 | | | | <?php foreach ($dataCodSerie as $result) : ?>
48 | | | | | <tr>
49 | | | | | | <td><?php echo $cont += 1; ?></td>
50 | | | | | | <td><?php echo $result['cod_serie']; ?></td>
51 | | | | | | <td>
52 | | | | | | | <a href="" class="btn btn-success EditCS" cod-serie="<?php echo $result['idregistro']; ?>"><i class="fa-sharp fa-solid fa-pen
53 | | | | | | </td>
54 | | | | | | <td>
55 | | | | | | | <a href="" class="btn btn-danger DelCS" cod-serie="<?php echo $result['idregistro']; ?>"><i class="fa-solid fa-trash"></i></a

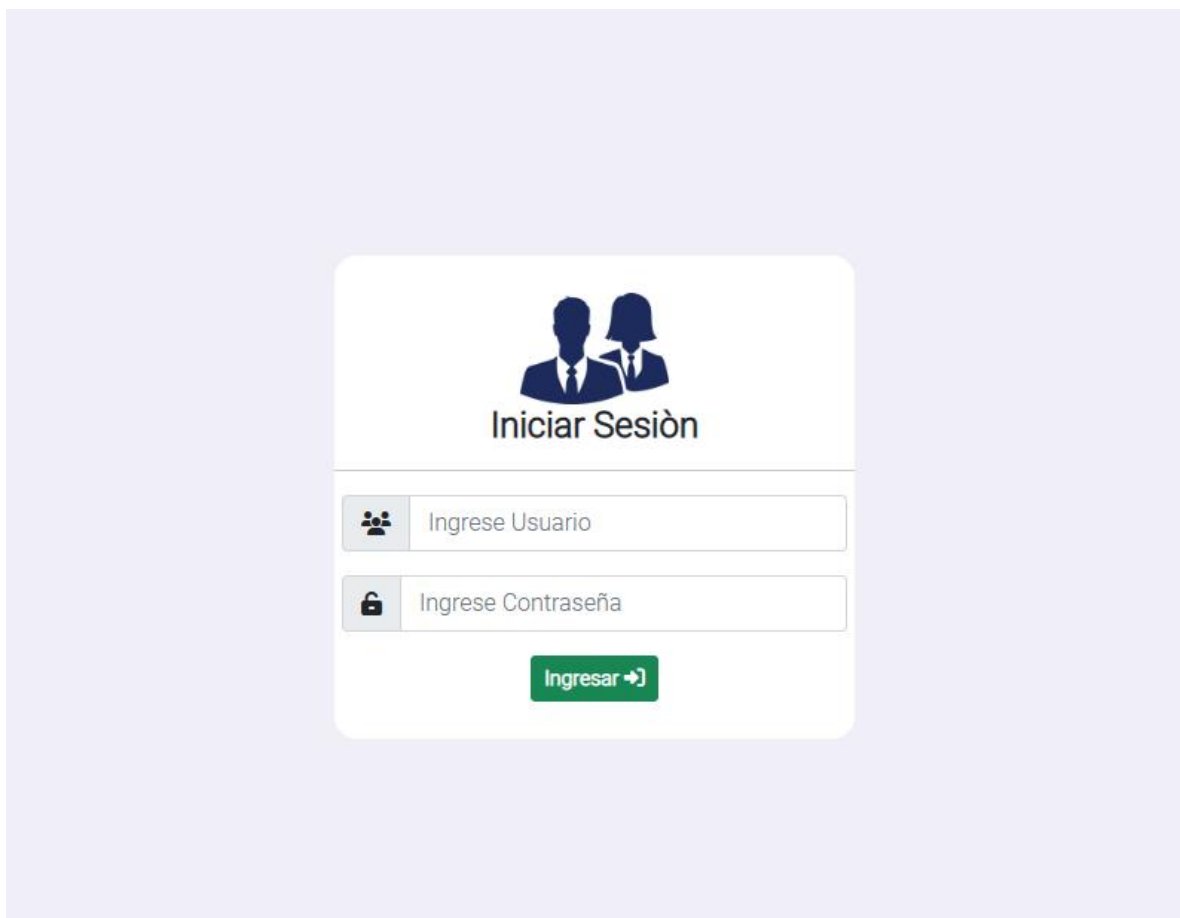
```

Archivo index.php: Este archivo es el índice de nuestra página web, este archivo está conectado con el archivo “index” ubicado en la carpeta controllers anteriormente mencionada.

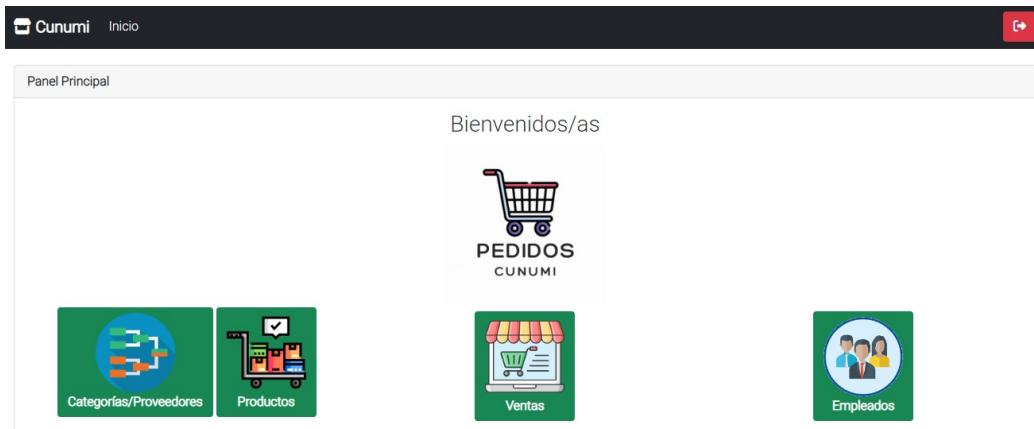


```
1 <?php
2     require './controllers/C_index.php';
3
4     $mvc = new Index();
5
6     $mvc->base();
7
8
9 ?>
```

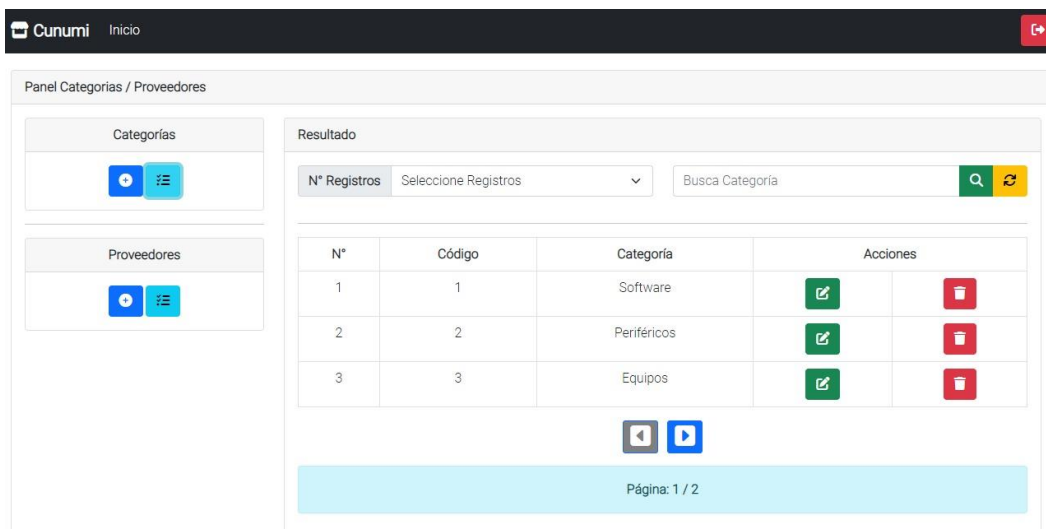
El usuario deberá iniciar sesión en el sistema para poder acceder a los paneles principales.



Luego de haber iniciado sesión el usuario será capaz de visualizar la ventana de inicio o panel principal donde encontrara una serie de paneles como el panel categorías/proveedores, panel productos, panel ventas y el panel empleados.



Dentro del panel *Categorías/Proveedores* encontrara una vista con diferentes acciones a realizar como lo es el ingreso de nuevas categorías/proveedores, siendo capaz listarlos y limitar el numero de registros a mostrar así como realizar su búsqueda y cambiar de página.







En el panel correspondiente a productos encontrar una vista con botones donde podrá ingresar nuevos productos dentro de un formulario, editarlos y eliminarlos de ser necesario.

Panel Productos

Panel Productos

N° Registros Seleccione Registros



Busca Categoría


N°	Código	Producto	Stock	Detalle	Acciones
1	195133118740	CARNE ACER ASPIRE 5 A515-54-55HZ	27	PROC. INTEL CORE I5 10210U RAM 8GB ALMACENAMIENTO SSD NVMe 256GB PANTALLA FHD IPS 15.6P W10HSL WIFI AX+BT5 BLACK	 
2	195133118741	CARNE ACER ASPIRE 5 A515-54-55HZ	15	PROC. INTEL CORE I5 10210U RAM 8GB ALMACENAMIENTO SSD NVMe 256GB PANTALLA FHD IPS 15.6P W10HSL WIFI AX+BT5 BLACK	 

Página: 1 / 1

Vista Ventas. Dentro de este panel el usuario podrá agregar productos al carrito para realizar la compra, procesarla y al completar esta acción podrá reclamar su comprobante de veta.


Panel Ventas



LAPTOP ACER ASPIRE 5 A515-54-55HZ

Precio: \$699.00





Stock: 28

En el panel empleados será posible agregar información de los empleados, editarla, bloquear/habilitar un usuario o eliminarlo.

Cunumi Inicio [+]

Panel Empleados



N°	Foto	Código	Empleado	DUI	Teléfono	Correo	Usuario	Acciones
1		1	Bartholomew J. Simpson	11111111-1	1111-1111	bart@gmail.com	gerardo	