



PROFESIONALES EN SISTEMAS

Equipo de trabajo:

Jim Kimpatrick Ascencio Ramírez
Nehemías Adonay Morales Reyes
Bryan Alexander Mata Cáceres
Azucena Eleticia Merlos Sandoval

Sistema de compra y ventas

Manual de Programador

San Miguel, El Salvador Junio 2021

Índice

Introducción	3
Información destacada	3
Objetivos	3
Requerimientos.....	4
Instalación y Configuración	5
Diagrama de Caso de Uso	7
Diagrama Entidad Relación	8
Configuración del Sistema	9
- Estructura Raíz	9
- Directorio Ajax.....	10
- Directorio Modelos	11
- Directorio Vistas.....	12
- Directorio Reportes	13
- Directorio Config.....	13
- Directorio Public	14
- Directorio Fpdf.....	15
Diccionario de datos y descripción de tablas.....	16
- Tabla articulo.....	16
- Tabla categoría	17
- Tabla detalle ingreso.....	17
- Tabla de detalle de venta	18
- Tabla ingreso	19
- Tabla permiso	20
- Usuario	21
- Tabla de venta.....	22
Definición y descripción de variables usadas en el sistema	23
Estructura del Sistema	26

Introducción

El siguiente documento tiene como objetivo, describir los aspectos técnicos e informáticos del sistema de compra y venta. Este permitirá al personal técnico y especializado, encargado de las actividades de mantenimiento, revisión y solución de problemas a futuro, instalación, modificación y configuración del sistema de manera eficaz.

Información destacada

El manual técnico proporciona la información necesaria con el fin de orientar al personal en la concepción, planteamiento análisis programación e instalación del sistema. Es notorio que el manual está dirigido a personal especializado con previos conocimientos en sistemas y tecnologías informáticas, conocimientos previos sobre programación avanzada y entornos web, administración de base de datos, responsables el mantenimiento del sistema.

Objetivos

Instruir al uso adecuado el Sistema Informático, para el acceso oportuno y adecuado en la instalación del mismo, mostrando los pasos a seguir, durante el proceso de instalación, así como la descripción de los archivos relevantes el sistema los cuales orienten en la configuración y el soporte del mismo.

Requerimientos

El sistema puede ser instalado y utilizado en cualquier sistema operativo que cumpla con los siguientes requerimientos.

- PHP 7.0 (O superior)
- MySql 5.0 (O superior)
- Navegador Web
- Servidor Web Apache 2.0 o superior

Instalación y Configuración

Para realizar la instalación del programa utilizaremos una ruta específica.

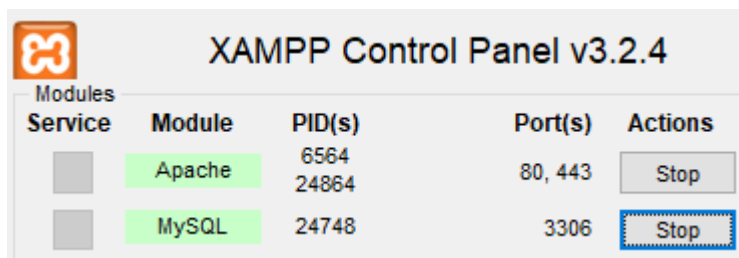
No es la opción más fácil, pero si la más segura y recomendada.

El método está enfocado a Sistemas Windows que poseen XAMPP (Distribución de Apache completamente gratuita).

1. Descargamos el archivo .Zip del programa que instalaremos.
2. Como segundo paso, entraremos a nuestro explorador de archivos, posteriormente en la ubicación donde se encuentra instalado XAMPP (usualmente en C:\xampp) y luego dentro de la carpeta htdocs (C:\xampp\htdocs).
3. Como tercer paso, crearemos una carpeta dentro de la ruta de htdocs
4. Como último paso para la instalación copiaremos el archivo .Zip el programa y lo llevaremos a la carpeta que hemos creado en htdocs posteriormente procederemos a descomprimir el archivo dentro de la carpeta.

A continuación, procederemos a instalar la base de datos del sistema:

1. Como primer paso, iremos a nuestro administrador de bases de datos, por defecto puede ser usado <<localhost/phpmyadmin>> para esto debemos tener activos el servidor de Apache y MySQL en el panel de control de XAMPP.



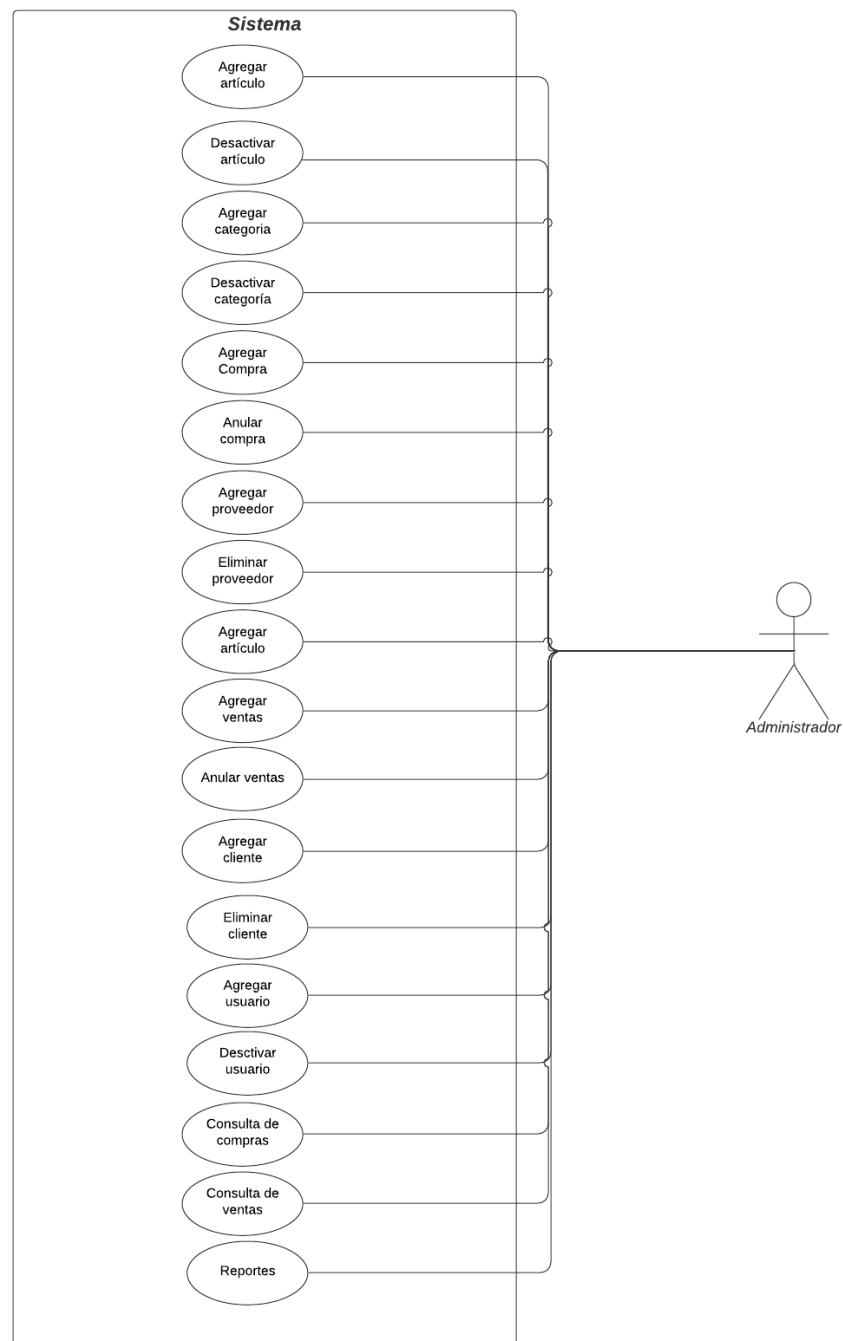
2. En nuestro navegador, en la pestaña <<localhost/phpmyadmin>> crearemos una nueva base de datos con el nombre

“bdcomprayventa” que nos servirá para importar nuestra información de tablas.

3. Como tercer paso, entraremos en la base de datos que creamos y seleccionaremos la opción “Importar”
4. En botón “seleccionar archivo” nos dirigimos a la carpeta de htdocs donde instalamos nuestro programa (C:\xampp\htdocs) y seleccionaremos el archivo “bdcomprayventa.sql” el cual nos importará la información necesaria para nuestra base de datos.

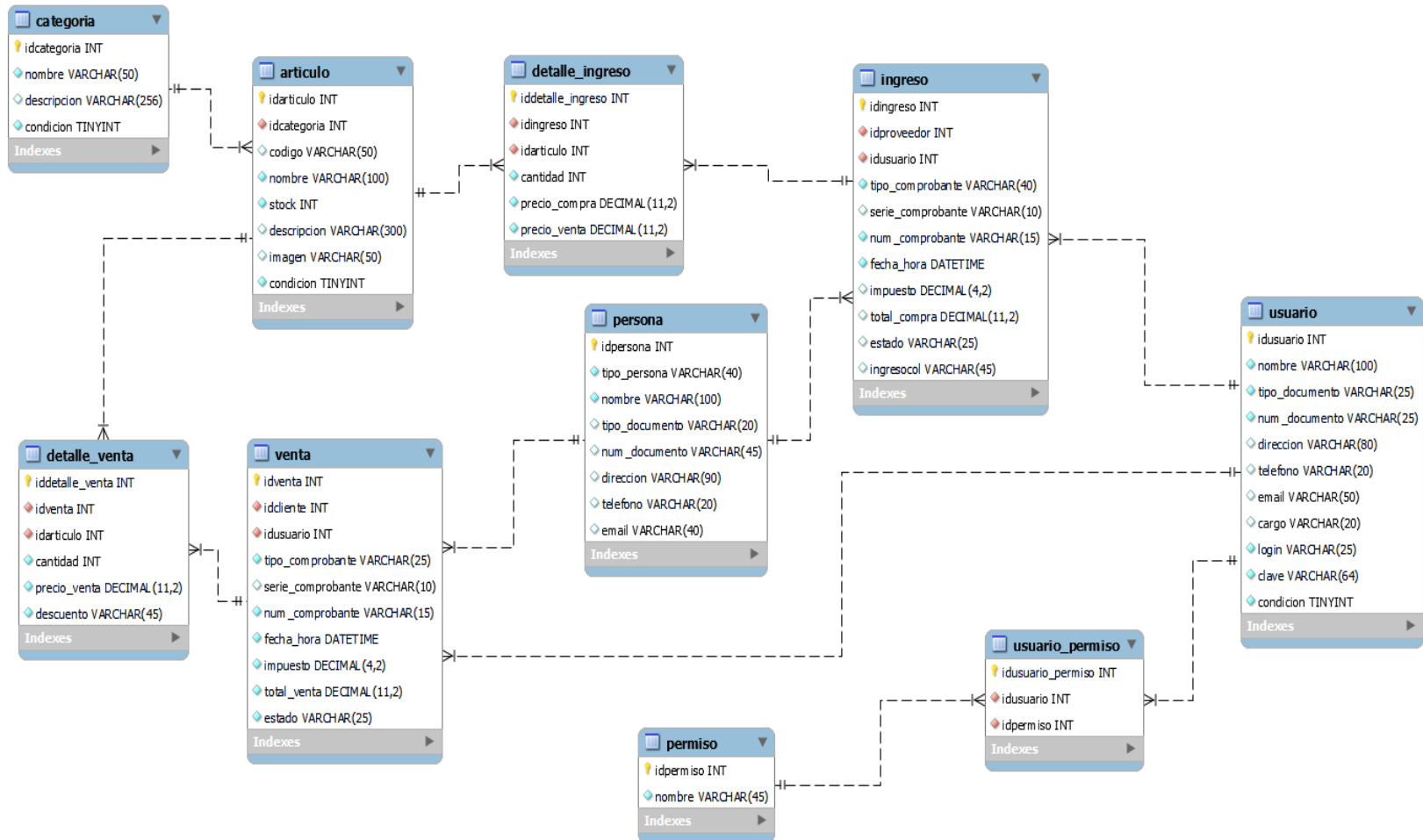
Y ya tendríamos las bases principales para poder ocupar nuestro programa.

Diagrama de Caso de Uso



Nota: El DCU se deja de esta manera debido a que el administrador tiene acceso a todos los procesos, pero él tiene la decisión cuando está agregando un usuario de ponerle los permisos que él desee, por lo tanto, es el único actor dentro del diagrama que tiene fijo sus accesos.

Diagrama Entidad Relación

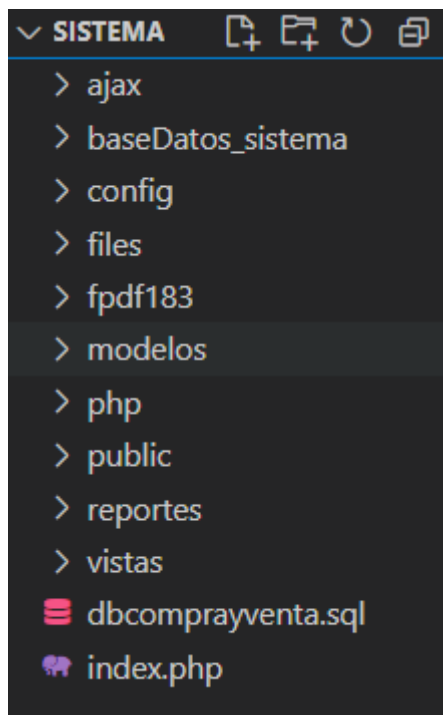


Configuración del Sistema

El sistema de Compra y Venta está desarrollado principalmente en el lenguaje PHP en su versión 7.0 utilizando otras herramientas para la visualización del programa como Bootstrap en su versión 3.0 para el entorno gráfico y Fpdf en su versión 1.83 para la creación de reportes que se requieren dentro del sistema.

- Estructura Raíz

El sistema de Compra y Venta posee la siguiente estructura de directorios

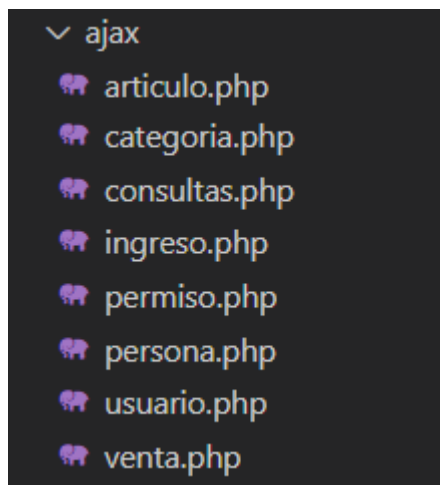


A continuación, se describen los directorios y archivos principales del sistema:

- **Directorio Ajax**

Ajax es utilizado para reducir demoras en la navegación web evitando recargar la página cada vez que se hace una petición al servidor. Estos nos permiten mostrar los datos de nuestra base al sistema de forma dinámica sin necesidad de cambiar o recargar la página.

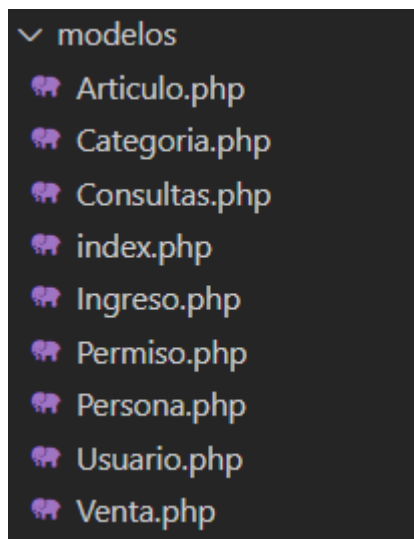
Este directorio posee los archivos **.php** los cuales nos permiten extraer y enviar datos a nuestra base en cada una de las rutas, de forma dinámica.



- **Directorio Modelos**

Los modelos son los archivos que representan la información lo cual opera el sistema, por lo tanto, son los archivos encargados de gestionar tantas consultas como actualizaciones dentro de la base de datos, implementando también privilegios de acceso que se hayan descrito, y todas las inserciones permitidas dentro de la base.

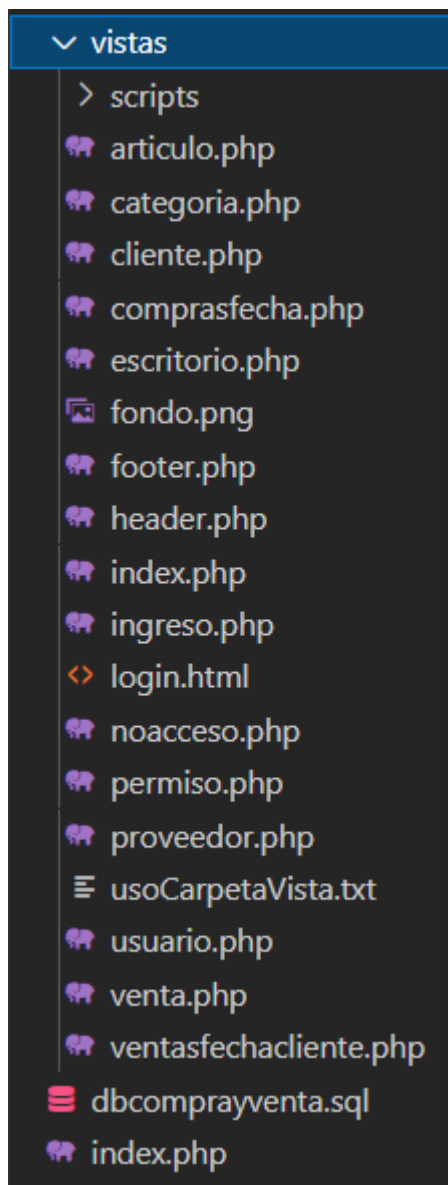
Permiten realizar todas las consultas requeridas, desde inserción y modificación, eliminaciones y extracción de datos requeridos que se encuentren dentro de la base.



- Directorio Vistas

Este directorio está compuesto principalmente por archivos **.PHP** los cuales poseen el entorno gráfico mostrado al usuario, como el formulario de inicio de sesión, las diferentes secciones en las que se divide el programa, la tabulación de los datos en tablas que ayudan a dar un mejor aspecto al programa.

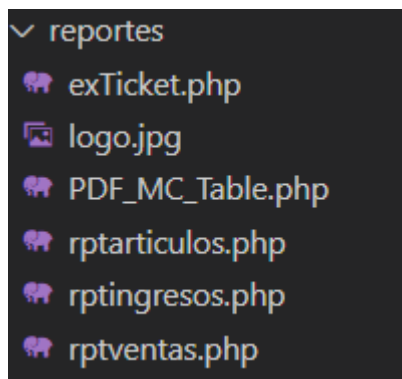
Dentro de este también se encuentra un subdirectorio con el nombre de "scripts" su función principal es enlazar el entorno gráfico con las consultas requeridas en cada parte del programa.



- Directorio Reportes

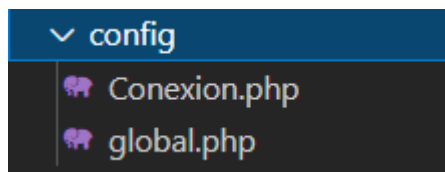
Dentro del directorio Reportes, encontramos archivos PHP que poseen la información para generar reportes de dentro del sistema, y la generación de tickets de compras.

Este directorio es dependiente de la biblioteca FPDF la cual permite generar informes en formato .pdf los cuales permiten generar reportes sobre cada una de las tablas que se encuentran dentro del sistema.



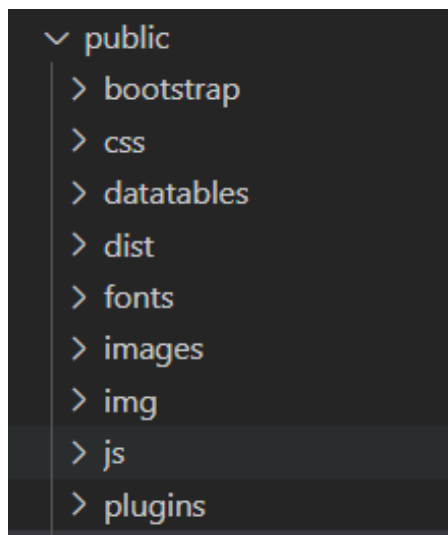
- Directorio Config

Dos archivos **.php** que poseen la configuración que permite la conexión entre la base de datos y el sistema, permitiendo realizar las consultas desde la base de datos específica.



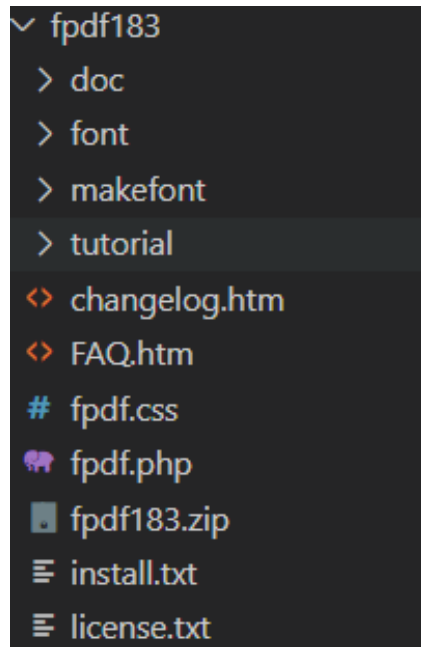
- **Directorio Public**

Dentro de este directorio tenemos todos los recursos estáticos de nuestra aplicación, es decir, archivos css, js, imágenes y fuentes, son librerías estáticas que permiten un mejor entorno gráfico de nuestro programa, entre ellos hojas de estilo css, archivos javaScript, imagenes, fuentes, librerias js, plugins, además nos permiten generar códigos de barra.



- Directorio Fpdf

Es una librería utilizada para la generación de reportes dentro de las tablas principales del sistema, creación de tickets de compra y su visualización a la hora de imprimir cada informe.



Diccionario de datos y descripción de tablas

- Tabla artículo

Aquí se guardan los artículos que hay en existencia para la venta

Columna	Tipo	Nulo	Predefinido	Enlaces a	Descripción	Media (MIME) type
idarticulo (Primaria)	int(11)	No			Id del artículo que se agrega, este se va autoincrementando	
idcategoria	int(11)	No		categoria -> idcategoria	El id que se trae de la tabla categoría	
codigo	varchar(50)	Sí	NULL		El código del producto que se va ingresar	
nombre	varchar(100)	No			El nombre del artículo ingresado	
stock	int(11)	No			La cantidad de producto que hay del artículo	
descripcion	varchar(256)	Sí	NULL		Descripción del artículo	
imagen	varchar(50)	Sí	NULL		Imagen ilustrativa del producto	
condicion	tinyint(1)	No	1		Aquí se sabe si el producto esta habilitado o no	

- Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idarticulo	3	A	No	La llave primaria es el id del artículo
nombre_UNIQUE	BTREE	Sí	No	nombre	3	A	No	El nombre del artículo tiene que ser único
fk_articulo_categoria_idx	BTREE	No	No	idcategoria	3	A	No	La llave foránea es el id de la categoría de los artículos

- Tabla categoría

Aquí se guardan las categorías donde pertenecen los artículos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
idcategoria (Primaria)	int(11)	No			Id de la categoría y es auto incrementable	
nombre	varchar(50)	No			Nombre de la categoría a la cual pertenecen los productos, como quesos, panes, etc.	
descripcion	varchar(256)	Sí	NULL		Descripción en lo que consiste la categoría	
condicion	tinyint(1)	No	1		Aquí se sabe si la categoría está habilitado o no	

- Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idcategoria	2	A	No	La llave primaria es el id de la categoría
nombre_UNIQUE	BTREE	Sí	No	nombre	2	A	No	El nombre debe ser único

- Tabla detalle ingreso

Aquí se guardan los detalles de los artículos que se compraron para la venta.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
iddetalle_ingreso (Primaria)	int(11)	No			Id del detalle de ingreso y es auto incrementable	
idingreso	int(11)	No		ingreso -> idingreso	El id que se trae de la tabla ingreso	
idarticulo	int(11)	No		articulo -> idarticulo	El id que se trae de la tabla articulo	
cantidad	int(11)	No			Cantidad de ingreso del artículo que se compro	
precio_compra	decimal(11,2)	No			Es el costo de la compra	
precio_venta	decimal(11,2)	No			Precio de venta que se le dará al articulo	

- Índice

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	iddetalle_ingreso	3	A	No	Primary key
fk_detalle_ingreso_ingreso_idx	BTREE	No	No	idingreso	3	A	No	La llave foránea traído de la tabla de ingresos
fk_detalle_ingreso_articulo_idx	BTREE	No	No	idarticulo	3	A	No	La llave foránea es el id de la tabla de ingreso de artículos

- Tabla de detalle de venta

En esta tabla se lleva el control del detalle de los artículos con sus especificaciones en las ventas

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
iddetalle_venta (Primaria)	int(11)	No			Id del detalle de venta y es auto incrementable	
idventa	int(11)	No		venta -> idventa	El id que se trae de la tabla venta	
idarticulo	int(11)	No		articulo -> idarticulo	El id que se trae de la tabla articulo	
cantidad	int(11)	No			Cantidad solicitada	
precio_venta	decimal(11,2)	No			Precio de venta por articulo	
descuento	decimal(11,2)	No			Descuento de venta	

- Índice

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	iddetalle_venta	10	A	No	Primary key
fk_detalle_venta_venta_idx	BTREE	No	No	idventa	10	A	No	La llave foránea traído de la tabla de ventas
fk_detalle_venta_articulo_idx	BTREE	No	No	idarticulo	10	A	No	La llave foránea traído de la tabla de articulo

- Tabla ingreso

Brinda el detalle completo de la compra, a quien se compra, cuando, el comprobante de compra y le agrega los impuestos.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
idingreso (<i>Primaria</i>)	int(11)	No			Id del ingreso, ósea de la compra y es auto incrementable	
idproveedor	int(11)	No		persona -> idpersona	El id que se trae el usuario del proveedor de la tabla persona	
idusuario	int(11)	No		usuario -> idusuario	El id que se trae de la tabla usuario para ver la persona que registro la compra	
tipo_comprobante	varchar(20)	No			El ticket de la compra	
serie_comprobante	varchar(7)	Sí	NULL		El número de serie	
num_comprobante	varchar(10)	No			Número del comprobante de compra	
fecha_hora	datetime	No			Fecha y hora de la compra	
impuesto	decimal(4, 2)	No			Impuesto de la compra	
total_compra	decimal(11, 2)	No			Total monetario de la compra	
estado	varchar(20)	No			Si esta anulado o aceptada la compra	

- Índice

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idingreso	3	A	No	Primary key
fk_ingreso_persona_idx	BTREE	No	No	idproveedor	3	A	No	La llave foránea traído de la tabla de persona
fk_ingreso_usuario_idx	BTREE	No	No	idusuario	3	A	No	La llave foránea traído de la tabla de usuario

- Tabla permiso

Aquí se guardan los permisos de los usuarios, las secciones donde puede acceder de acuerdo a los permisos que le dé el admin.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
idpermiso (<i>Primaria</i>)	int(11)	No			Id de permiso de usuario y es auto incrementable	
nombre	varchar(30)	No			Nombre del permiso	

- Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idpermiso	7	A	No	Primary key

- Tabla persona

Aquí se guardan los clientes y los proveedores

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media (MIME) type
idpersona (<i>Primaria</i>)	int(11)	No			Id de persona y es auto incrementable	
tipo_persona	varchar(20)	No			Tipo de persona, si es cliente o si es proveedor	
nombre	varchar(100)	No			Nombre de la persona	
tipo_documento	varchar(20)	Sí	NULL		Tipo de documento personal para registrarse	
num_documento	varchar(20)	Sí	NULL		Numero de documento	
direccion	varchar(70)	Sí	NULL		Dirección de la sucursal del proveedor, o de la residencia del cliente	
telefono	varchar(20)	Sí	NULL		Número telefónico	
email	varchar(50)	Sí	NULL		Email de la persona o empresa	

- Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idpersona	4	A	No	Primary key

- Usuario

Aquí se guardan los datos de los usuarios de los empleados de la empresa y el tipo de cargo que tiene

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media (MIME) type
idusuario (Primaria)	int(11)	No			Id de usuario y es auto incrementable	
nombre	varchar(100)	No			Nombre del usuario	
tipo_documento	varchar(20)	No			Tipo de documento personal para registrarse	
num_documento	varchar(20)	No			Numero de documento	
direccion	varchar(70)	Sí	NULL		Dirección del usuario	
telefono	varchar(20)	Sí	NULL		Teléfono del empleado	
email	varchar(50)	Sí	NULL		Email del empleado	
cargo	varchar(20)	Sí	NULL		Cargo del empleado	
login	varchar(20)	No			Usuario con el que hará inicio de sesión	
clave	varchar(64)	No			Contraseña	
imagen	varchar(50)	No			Imagen del empleado	
condicion	tinyint(1)	No	1		Si está activo o si está inactivo el usuario	

- Índice

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idusuario	2	A	No	Primary key
login_UNIQUE	BTREE	Sí	No	login	2	A	No	El nombre de usuario debe ser único

- Tabla de usuario permiso

Tabla donde se guardan los permisos a los que tiene acceso cada usuario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
idusuario_permiso (Primaria)	int(11)	No			Id de permiso usuario y es auto incrementable	
idusuario	int(11)	No		usuario -> idusuario	El número de id del usuario	
idpermiso	int(11)	No		permiso -> idpermiso	Registro del permiso	

- Índice

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idusuario_permiso	16	A	No	Primary key
fk_usuario_permiso_permiso_idx	BTREE	No	No	idpermiso	16	A	No	La llave foránea traído de la tabla de persona
fk_usuario_permiso_usuario_idx	BTREE	No	No	idusuario	8	A	No	La llave foránea traído de la tabla de usuario

- Tabla de venta

Aquí se registran los datos completos de las ventas, cuando, a quien, con comprobante y los impuestos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Descripción	Media (MIME) type
idventa (Primaria)	int(11)	No			Id de ventas y es auto incrementable	
idcliente	int(11)	No		persona -> idpersona	Id del cliente al que se le cargara la venta	
idusuario	int(11)	No		usuario -> idusuario	Id del usuario quien hizo la venta	
tipo_comprobante	varchar(20)	No			El tipo de comprobante de venta	
serie_comprobante	varchar(7)	Sí	NULL		El número de serie	
num_comprobante	varchar(10)	No			El número del comprobante	
fecha_hora	datetime	No			La fecha y la hora en la que se realizó la venta	
impuesto	decimal(4,2)	No			El impuesto de venta	
total_venta	decimal(11,2)	No			El total monetario	
estado	varchar(20)	No			Si esta anulada la venta o si fue aceptada	

- Índice

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Descripción
PRIMARY	BTREE	Sí	No	idventa	8	A	No	Primary key
fk_venta_persona_idx	BTREE	No	No	idcliente	8	A	No	El número de id del cliente
fk_venta_usuario_idx	BTREE	No	No	idusuario	4	A	No	El número del id del usuario que hizo la venta

Definición y descripción de variables usadas en el sistema

Script	Función	Ubicación	Líneas
Index.php	Página que redirección al login.	Carpeta Raiz	4
header.php	Genera la parte del encabezado de las páginas del sitio con una plantilla de administración.	vistas	221
footer.php	Genera el pie de la página donde están los enlaces de scripts.	Carpeta rais/vistas	27
login.html	Página con un formulario para ingresar sus datos para ingresar al sistema, la cual se cargará siempre que se quiera acceder a cualquier parte del sitio sin iniciar sesión.	Carpeta rais/vistas	68
articulo.php	Página que muestra una lista de artículos del sistema y agregar artículos e imprimir un informe en diferentes formatos.	Carpeta rais/vistas	118
categoría.php	Página que muestra una lista de las categorías del sistema y la opción de agregar categorías nuevas, deshabilitar categorías o editar categorías.	Carpeta rais/vistas	89
escritorio.php	Página que muestra una gráfica con las compras en los últimos días y las ventas en los últimos 12 meses	Carpeta rais/vistas	241

ingreso.php	Página que muestra una tabla con los ingresos al almacén del sistema (solo se pueden ingresar artículos ya registrados en el almacén) con su respectivo proveedor.	Carpeta rais/vistas	185
proveedor.php	Página que muestra una tabla con los proveedores registrados en el sistema con las opciones de agregar nuevos proveedores, editar proveedor y eliminar proveedor.	Carpeta rais/vistas	113
venta.php	Página que muestra una tabla con las ventas registradas en el sistema con las opciones de generar reportes en diferentes formatos, ver venta, agregar venta, generar ticket y anular venta.	Carpeta rais/vistas	190
cliente.php	Página que muestra los clientes registrados en el sistema en el cual puede agregar nuevos clientes, editar clientes y eliminar clientes.	Carpeta rais/vistas	113
usuario.php	Página que muestra una tabla con los usuarios del sistema con las opciones de agregar nuevos usuarios con sus respectivos permisos,	Carpeta rais/vistas	143

	editar usuarios y desactivar usuarios.		
permiso.php	Página que muestra una lista con los permisos del sistema.	Carpeta rais/vistas	63
comprasfecha.php	Página que muestra una tabla en la cual se visualizarán las compras en un intervalo de tiempo ingresado en fecha inicio y fecha fin.	Carpeta rais/vistas	88
ventasfechacliente.php	Página que muestra las ventas realizadas según el cliente seleccionado en un intervalo de tiempo que ingresa el usuario.	Carpeta rais/vistas	94
noacceso.php	Página que se muestra cuando no tienes permiso de ver el escritorio.	Carpeta rais/vistas	25

Estructura del Sistema

El sistema está hecho de forma semimodular, cada página que se carga dentro del sistema se compone de scripts.

Estos se encuentran dentro del directorio denominado *vistas* este incluye un módulo llamado, *header.php*, es el módulo principal que se carga en la parte central de cada página del sistema y el módulo denominado *footer.php*, se muestra en la parte inferior de cada página del directorio *vistas*. Estos módulos se integran mediante un solo archivo *index.php*

El script que se carga dentro del archivo *index.php* es invocado agregando una variable identificada como *header*. Este archivo funciona como una cadena para redireccionar la página que se desea cargar si no existe una sesión activa.

```
<? php
    //redireccionar a la vista de login
    header ('Location: vistas/login.html');
?>
```

Para el inicio de sesión es necesario crear una conexión con la base de datos del sistema que nos permita extraer datos de las tablas de las diferentes entidades, para eso utilizamos los archivos que se muestran en nuestro directorio *config*, dentro del cual se encuentran los dos archivos *global.php* el cual utiliza el host local de la pc servidor para establecer una conexión del sistema con la base de datos definida, con un usuario administrador principal que gestionara los demás usuarios del sistema

```
<?php
//ip de la pc servidor de la base de datos
define("DB_HOST", "localhost");

//nombre de la base de datos
define("DB_NAME", "dbcomprayventa");

//usuario de la base de datos
define("DB_USERNAME", "root");

//contraseña del usuario
define("DB_PASSWORD", "");

//definimos la codificacion de los caracteres
```

```

define("DB_ENCODE", "utf8");

//guarda el nombre del proyecto
define("DB_NOMBRE", "La vaca Lola");

?>

```

Y un archivo con el nombre de Conexión.php que hace referencia al archivo global.php en el cual se valida la conexión con la base.

```

if(mysqli_connect_errno()){

    printf("fallo en la conexion a la base de datos: %s \n",mysqli_connect_error());
    exit();
}
if(!function_exists('ejecutarConsulta')){

    function ejecutarConsulta($sql){
        global $conexion;
        $query = $conexion->query($sql);
        return $query;
    }

    function ejecutarConsultaSimpleFila($sql){
        global $conexion;
        $query = $conexion->query($sql);
        $row = $query->fetch_assoc();
        return $row;
    }

    function ejecutarConsulta_retornarID($sql){
        global $conexion;
        $query = $conexion->query($sql);
        return $conexion->insert_id;
    }

    function limpiarCadena($str){
        global $conexion;
        $str = mysqli_real_escape_string($conexion,trim($str));
        return htmlspecialchars($str);
    }

}
?>

```

Dentro del código podemos observar tres funciones que permiten ejecutar consultas primarias a la base de datos.

Para poder acceder a la vista principal del sistema es necesario un arreglo que permita verificar una sesión activa por parte del usuario, para esto es necesario un arreglo como se muestra en el siguiente fragmento de código.

```
<?php
ob_start();
if (strlen(session_id()) < 1){
    session_start();//Validamos si existe o no la sesión
}
if (!isset($_SESSION["nombre"]))
{
    header("Location: ../vistas/login.html");//Validamos el acceso solo a los usuarios
    logueados al sistema.
}
```

Observamos que adicionalmente las páginas están protegidas por sesiones. Lo que significa que los usuarios no pueden acceder a ninguna página a menos que exista una sesión activa, esto también se verifica dentro del archivo usuario.php del directorio Ajax.

```
$logina=$_POST['logina'];
$clavea=$_POST['clavea'];

$clavehash=hash("SHA256",$clavea);

$rspta=$usuario->verificar($logina, $clavehash);

$fetch=$rspta->fetch_object();

if (isset($fetch))
{
    //Declaramos las variables de sesión
    $_SESSION['idusuario']=$fetch->idusuario;
    $_SESSION['nombre']=$fetch->nombre;
    $_SESSION['imagen']=$fetch->imagen;
    $_SESSION['login']=$fetch->login;

    //Obtenemos los permisos del usuario
    $marcados = $usuario->listarmarcados($fetch->idusuario);

    //Declaramos el array para almacenar todos los permisos marcados
    $valores=array();

    //Almacenamos los permisos marcados en el array
    while ($sper = $marcados->fetch_object())
    {
        array_push($valores, $sper->idpermiso);
    }
}
```

```

    }

    //Determinamos los accesos del usuario
    in_array(1,$valores)?$_SESSION['escritorio']=1:$_SESSION['escritorio']=0;
    in_array(2,$valores)?$_SESSION['almacen']=1:$_SESSION['almacen']=0;
    in_array(3,$valores)?$_SESSION['compras']=1:$_SESSION['compras']=0;
    in_array(4,$valores)?$_SESSION['ventas']=1:$_SESSION['ventas']=0;
    in_array(5,$valores)?$_SESSION['acceso']=1:$_SESSION['acceso']=0;
    in_array(6,$valores)?$_SESSION['consultac']=1:$_SESSION['consultac']=0;
    in_array(7,$valores)?$_SESSION['consultav']=1:$_SESSION['consultav']=0;

}
echo json_encode($fetch);

```

En el segmento de código anterior se observa la información que obtenemos para validar la sesión del usuario, verificando su nombre e inicio, su contraseña, y los permisos de acceso que se toman dentro el mismo archivo en el caso de permisos

```

case 'verificar':

    require_once "../modelos/Permiso.php";
    $permiso = new Permiso();
    $rspta = $permiso->listar();

    //Obtener los permisos asignados al usuario
    $id=$_GET['id'];
    $marcados = $usuario->listarmarcados($id);
    //Declaramos el array para almacenar todos los permisos marcados
    $valores=array();

    //Almacenar los permisos asignados al usuario en el array
    while ($per = $marcados->fetch_object())
    {
        array_push($valores, $per->idpermiso);
    }

    //Mostramos la lista de permisos en la vista y si están o no marcados
    while ($reg = $rspta->fetch_object())
    {
        $sw=in_array($reg->idpermiso,$valores)?'checked':'';
        echo '<li> <input type="checkbox" '.$sw.' name="permiso[]" value="'.$reg->idpermiso.'">'.$reg->nombre.'</li>';
    }
    break;

```

Una vez autenticado el usuario y sus credenciales con permisos, se carga la página que verá el usuario autenticado. Esto se encuentra dentro del

archivo header.php del directorio vistas y a continuación una serie de condiciones "if" validan si existen sesiones activas por cada ventana del sistema, como pudimos observar en el código anterior, si el usuario no posee permiso para una ventana según la obtención de permisos de la parte del código anterior, el usuario no podrá visualizar esa ventana, un ejemplo de ello se muestra a continuación con el escritorio principal del sistema que se encuentra dentro el header.php

```
if ($_SESSION['escritorio']==1)
{
    echo '<li id="mEscritorio">
    <a href="escritorio.php">
    <i class="fa fa-tasks"></i> <span>Escritorio</span>
    </a>
    </li>';
}
?>
```

Podemos observar dentro de la condición, que el escritorio solo se mostrará si la sesión del usuario posee permisos para el escritorio

Finalmente, para que el usuario pueda cerrar su sesión utilizamos el caso de nombre salir, que se encuentra dentro del directorio Ajax en el archivo usuario.php el cual se encarga de limpiar las variables de la sesión y destruir la sesión activa, para posteriormente redireccionar al login.

```
case 'salir':

    session_unset();

    session_destroy();

    header("Location: ../index.php");

    break;
```

Cabe resaltar que no se puede acceder a otra página de la cual no se tienen permisos utilizando el url de una sesión activa.