

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



**Proyecto.**

**Equipo desarrollador.**

<b>APELLIDOS, NOMBRES</b>	<b>CARNET</b>
Martínez Valladares, Ana Cecilia	MV01135209
Monge Cerón, Liliana De Los Ángeles	MC01136242
Trejo Franco, Douglas Rafael.	TF01136646
Orellana Quijada, Víctor Manuel.	OQ01136509

**CÁTEDRA:** Base de datos II.

**HORARIO:** 7:00am a 8:40am.

**FACILITADOR:** Irwin Guardado.

**SAN SALVADOR, 30 DE Noviembre DE 2021**

## Contenido

INTRODUCCIÓN.....	3
ANÁLISIS DEL PROBLEMA.....	4
OBJETIVOS.....	5
ANÁLISIS DE BENEFICIOS DEL PROYECTO.....	6
JUSTIFICACIÓN.....	7
BENEFICIOS DEL SISTEMA.....	8
Análisis.....	11
Diseño.....	12
Desarrollo.....	16
Documentación.....	21
CONCLUSIONES.....	22
Recomendaciones.....	23
Bibliografía.....	24
Anexos.....	25
Introducción.....	26
<b>Objetivos.....</b>	<b>27</b>
<b>Manual de usuario.....</b>	<b>28</b>
Manual del programador.....	33

# INTRODUCCIÓN.

Una de las principales situaciones por las que pasan a diario las empresas u organizaciones, es que el mundo se va globalizando, buscan ser más competitivas entre sí, es por ello que sufren de pérdidas ya sea en productos o capital, y por tal razón algunas empresas han optado en cerrar, para poder lograr un mejor rendimiento es necesario implementar herramientas informáticas para la gestión administrativa.

En la actualidad los sistemas de control de ventas van evolucionando a medida que la tecnología va tomando la vanguardia de casi todo, estos sistemas se realizan por medio de bases de datos digitales, dicho proyecto estará siendo desarrollado en una base de datos NoSQL, donde esta se adapta a aplicaciones modernas como lo es el sistema de venta el cual ayuda a recopilar las informaciones de los clientes y registrar las entradas y salidas de los productos. Mediante su desarrollo se busca facilitar el trabajo del control de ventas con la ayuda de las herramientas necesarias.

# ANÁLISIS DEL PROBLEMA

**TEMA:** Sistema de Ventas de Ropa

## **PROBLEMÁTICA A RESOLVER.**

El proyecto que está siendo presentado del sistema de ventas de ropa, desarrollado en la colonia Escalón, se ha observado la problemática que no cuentan con un sistema de ventas actualizado para poder llevar el control de las salidas y entradas de los productos o información. Lo tradicional está quedando atrás hoy en día la mayoría de personas se están actualizando con las tecnologías porque estas van innovándose día a día, para facilitar el trabajo a las personas.

Estos negocios prefieren realizar la gestión de sus recursos, los inventarios y presupuestos de manera manual y sin una organización adecuada para su realización, lo cual requiere mucho tiempo invertido para registrar las transacciones, en el cual la empresa sufre pérdida en su rendimiento y por lo tanto genera un mayor costo, además que los datos no son exactos y verídicos para la generación de los reportes que permitan la toma de mejores decisiones estratégicas que beneficien el crecimiento de la empresa Rascacielos.

El sistema de ventas será llevado de forma virtual para brindar un mejor beneficio, el sistema tendrá un mejor control de los productos que se irán adquiriendo para ofrecer a las personas, así dando una mejor efectividad al inventario y que esto pueda generar un mejor ingreso en las ventas del sistema. Las empresas y negocios están optando por la implementación de bases de datos para la organización y gestión de sus sistemas de ventas. Sin embargo, la mayoría de negocios pequeños o en crecimiento no cuentan con estos beneficios para su organización y gestión.

Además del retraso en la atención a los clientes por la falta de un sistema automático de control de ventas y registró también la poca capacidad de los métodos actuales que no benefician para la optimización del tiempo de operación en el proceso de ventas.

# OBJETIVOS

## GENERAL:

- Desarrollar un sistema de ventas para la empresa Rascacielos, para optimizar una mejor viabilidad en dicho sistema, ya que tendrá un mejor manejo del control de la empresa y facilitar así los resultados y efectividad para el beneficio de dicha empresa.

## ESPECÍFICOS:

- Definir los requisitos esenciales para llevar a cabo el funcionamiento del sistema de ventas que se desarrollará en las distintas etapas.
- Demostrar el desarrollo del sistema que tendrá la capacidad de realizar registros para mejorar el rendimiento de la empresa Rascacielos.
- Analizar las tareas en cuanto a las ventas para que estas sean más fáciles de realizar, además de que se pueden obtener reportes y todos los datos e información importante se encuentran almacenados dentro del sistema y se pueden consultar en cualquier momento que sea necesario.

# ANÁLISIS DE BENEFICIOS DEL PROYECTO.

El desarrollo del sistema de ventas, beneficiará a los empleados y clientes de la empresa Rascacielos ya que tendrán la facilidad de tener un mejor manejo a la hora de realizar procesos, que puedan hacer la diferencia entre la demás empresa y tener la capacidad de registrar las transacciones en el menor tiempo posible.

## LISTA DE LOS BENEFICIOS QUE TENDRÁ EL SISTEMA DE VENTAS.

- **Mejor manejo de datos:** Esto permitirá tener una habilidad a la hora de ingresar datos y mostrarlos.
- **Optimizar el control de inventario:** Un punto de venta permite ver en tiempo real la cantidad de productos disponibles para procesar algún requerimiento por parte del usuario.
- **Reducción de tiempo:** Al tener la capacidad de registrar las transacciones en el menor tiempo posible, se puede mejorar al máximo la experiencia de compra.
- **Generación de reporte:** Efectuará una mejor calidad que permitirá tomar mejores decisiones estratégicas que beneficien el crecimiento de la empresa.
- **Reducción el tiempo al cliente:** El sistema de ventas, podrá optimizar el tiempo de operación en el proceso de ventas al cliente ya que los estarán organizadas por categorías.
- **Acceso múltiple:** Todos los usuarios podrán tener acceso simultáneo al sistema de ventas, ya que no tendrá un límite de ingresos.

# JUSTIFICACIÓN.

El proyecto se desarrollará en base a los conocimientos adquiridos durante lo aprendido en clases, el sistema de venta hoy en día es muy importante dentro de una gran o pequeña empresa. Dicho sistema ayudará a recopilar la información sobre los productos, categorías y proveedores.

Dicho sistema, las funciones que ofrecerá el sistema es la actualización del inventario en tiempo real, conforme se ingresan los productos. Esto permitirá optimizar el control de inventarios e identificar los productos que son obsoletos.

## **PROBLEMA A RESOLVER.**

Hoy en día se debe considerar la importancia de un sistema de venta dentro y fuera de una empresa o tienda, ya que dicho sistema ayuda a resolver muchas problemáticas que se presentan día con día. El sistema estará diseñado para brindar facilidad a la empresa para guardar la información de los clientes con mayor confianza.

También el sistema podrá resolver la problemática sobre el manejo de inventario de toda la información, que se lleva dentro de la empresa o tienda, dicho sistema llevará de manera ordenada toda la información general de cada producto, proveedor y usuarios existentes dentro de la empresa. Ayudará a optimizar el tiempo de búsqueda para algún producto en específico.

# BENEFICIOS DEL SISTEMA.

El sistema tendrá los siguientes beneficios.

- El manejo sobre toda la información en general dentro de la empresa.
- Optimizar el tiempo de búsqueda de algún producto en específico.
- Ingresar productos de manera rápida.
- El administrador tendrá acceso a todas las funciones del sistema.
- Se podrán agregar nuevos usuarios.
- Control de inventario.

## ASPECTOS POR LOS CUALES ES NECESARIO DESARROLLARLO.

El sistema de ventas, se desarrollará con el fin de optimizar el tiempo en la empresa Rascacielos donde se implementará en el cual beneficiará a los empleados y clientes a la hora del proceso de ventas. Dicho sistema tendrá diferentes áreas de desarrollo donde el administrador podrá ingresar, visualizar, actualizar y eliminar datos en general.

El sistema tendrá un área específica de inventario donde se almacenarán los datos ingresados por el administrador como productos y categorías, a la hora de buscar un dato en específico no será muy complicado ya que el sistema contendrá una función la cual ayudará a buscar y obtener un resultado rápidamente.

También, al momento de generar un nuevo usuario dentro del sistema este se almacenará en una base de datos, seguidamente el sistema tendrá implementada una función la cual encripta la contraseña del usuario con el fin de generar confianza y seguridad. Con el fin de resguardar bien la informa<sup>1</sup>de los usuari

---

1

## Metodología y ciclo de vida utilizados:

### 1. Requerimientos funcionales

#### 1. El usuario podrá ingresar datos a los diferentes campos .

- 1.1 El sistema tendrá que mostrar la colección de productos.
- 1.2 El sistema tendrá que mostrar la colección de proveedores.
- 1.3 El sistema tendrá que mostrar la colección del cliente.
- 1.4 El sistema tendrá que mostrar la colección de categorías.
- 1.5 El sistema tendrá que mostrar la colección de ventas.

#### 2. El usuario podrá ingresar datos a la colección de productos.

- 2.1 El sistema tendrá que procesar los datos.
- 2.2 El sistema mostrará los datos almacenados.
- 2.3 El sistema tendrá la funcionalidad de actualizar y eliminar productos existentes.

#### 3. El usuario podrá actualizar los registros almacenados.

- 3.1 El sistema tendrá que mostrar los datos actuales.
- 3.2 El sistema tendrá que ocupar las funciones **updateOne** y **set** para poder actualizar los datos.
- 3.3 El sistema tendrá que validar la información actualizada por el usuario.

#### 4. El usuario podrá eliminar los registros almacenados.

- 4.1 El sistema tendrá que mostrar los datos actuales.
- 4.2 El sistema tendrá que validar que el código de registro sea igual al ID para poder eliminar los datos.
- 4.3 El sistema tendrá que ocupar la función **deleteOne** para poder eliminar los datos solicitados

## **Requerimientos no funcionales.**

1. El sistema tendrá que procesar los datos del cliente.
2. El sistema tendrá una interfaz sencilla.
3. El sistema tendrá que procesar las ventas realizadas.
4. El sistema deberá indicar al usuario que debe rellenar todos los campos obligatoriamente.
5. El sistema tendrá que permitir la actualización de datos existentes.
6. El sistema deberá procesar todas las consultas que se soliciten dentro de ella.

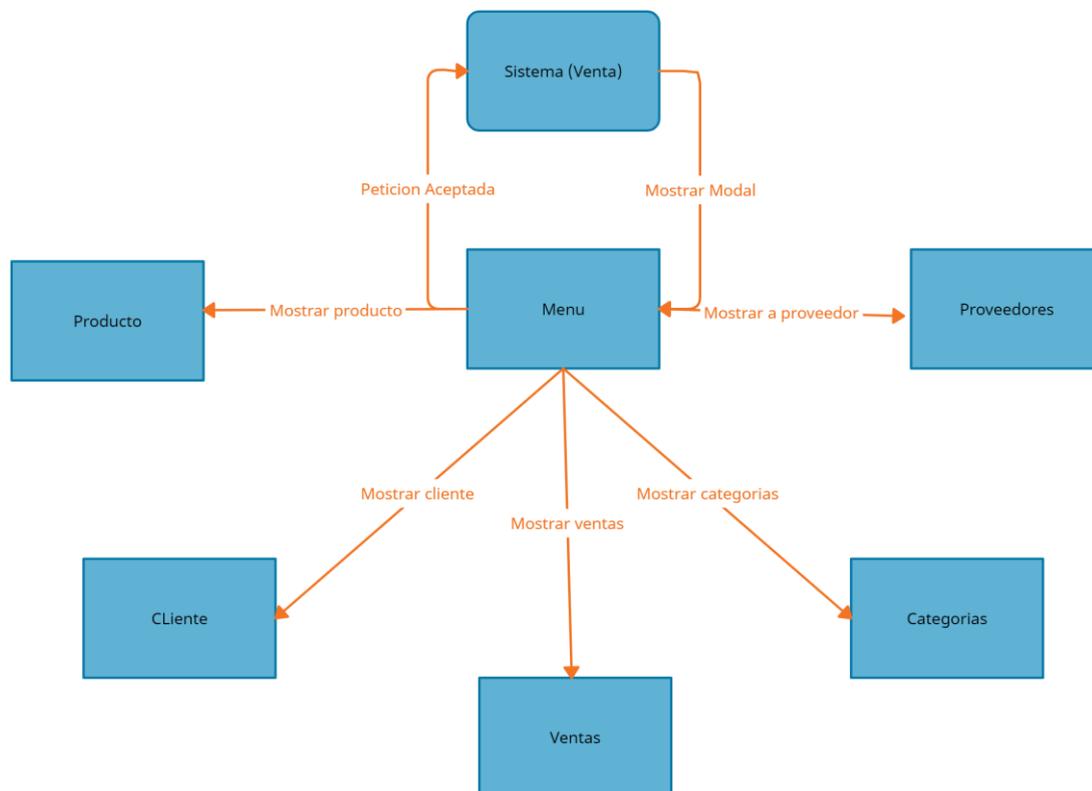
## **Modelo ciclo de vida.**

Para el desarrollo de nuestro proyecto usamos el modelo V ya que, el cual nos permite ir actualizando nuestro sistema e ir haciendo pruebas para facilitar la localización de posibles fallos dentro del sistema. El objetivo de ocupar el modelo V fue comprobar que cada módulo del sistema funcione como se espera, de forma independiente, y facilitar así las pruebas de integración.

- Optimización del sistema a la hora de ingresar datos.
- Mejor manejo
- Colaboración del cliente sobre como requiere el sistema del que creamos.
- Mejorar la satisfacción del cliente.
- Los objetivos del modelo ágil incluyen:
- Creación eficiente de software de trabajo.
- Satisface su propósito.
- Es inteligible.
- Es suficientemente preciso.
- Es suficientemente consistente.
- Es suficientemente detallado.
- Aporta valor positivo.
- Es lo más simple posible

# Análisis

## Análisis Estructurado.



Se ha trabajado la programación estructurada para la realización del sistema de proyecto, se ha creado un CRUD sencillo que permite insertar, mostrar, actualizar y eliminar registros existentes.

El diseño del sistema se realizó en una división de acuerdo a módulos, donde cada archivo estará enlazado a través de direcciones y métodos POST. Ya que se utiliza un diseño estructural, se utilizó el lenguaje de programación php para poder realizar las conexiones necesarias con la base de datos y sus respectivas colecciones.

El sistema tendrá un ciclo y funcionamiento donde se mostrará primeramente una pantalla de inicio y su respectivo menú de acceso con los módulos ya designados, donde los módulos tendrán que mostrar los datos existentes, seguidamente actualizar y eliminar los datos.

```

collection.json
1 //Tabla Categoria
2 {
3   "_id": "ObjectId",
4   "codcategoria": "String",
5   "categoria": "String"
6 }
7
8 //Tabla Cliente
9 {
10  "_id": "ObjectId",
11  "cod_cliente": "String",
12  "nombres": "String",
13  "apellidos": "String",
14  "direccion": "String",
15  "telefono": "Number",
16  "correo": "String"
17 }
18
19 //Tabla producto
20 {
21  "_id": "ObjectId",
22  "codproducto": "String",
23  "nombre": "String",
24  "descripcion": "String",
25  "precio": "Number",
26  "idproveedor": "Number"
27 }
28
29 //Tabla Proveedor
30 {
31  "_id": "ObjectId",
32  "codprov": "String",
33  "nombre": "String"
34 }
35
36 //Tabla Venta
37 {
38  "_id": "ObjectId",
39  "cod_venta": "001",
40  "producto": "String",
41  "cantidad": "String",
42  "descripcion": "String",
43  "precio_venta": "Number",
44  "precio_compra": "Number",
45  "fecha_venta": "String"
46 }
47
48 }
49

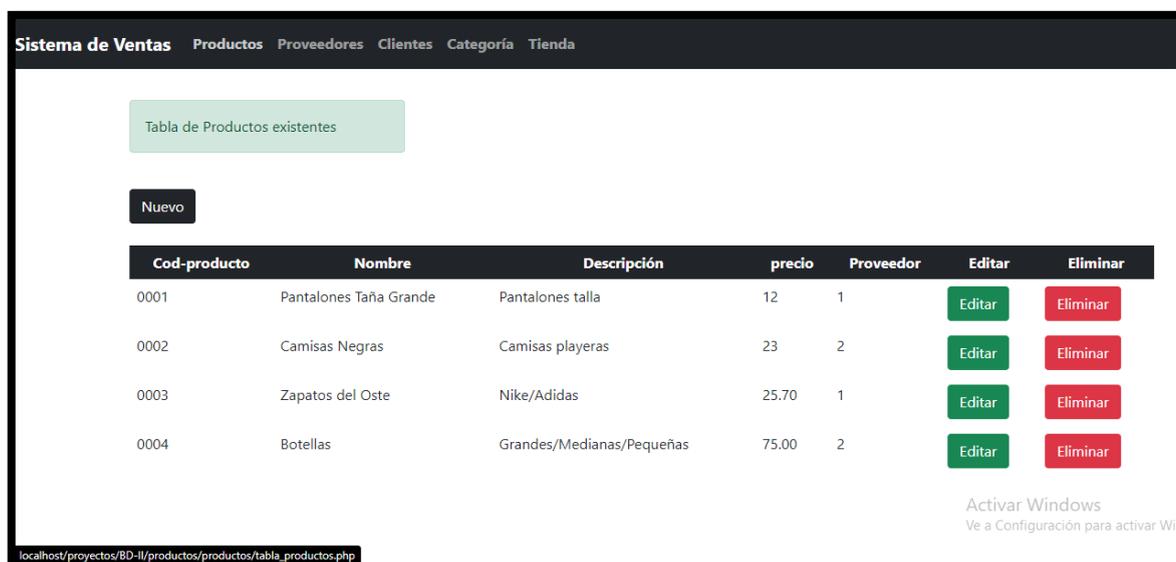
```

## Diseño

El diseño del sistema consta de las siguientes acciones:

### “Tabla Productos”

1. Si el usuario escoge un módulo a través del menú, podrá ingresar a una pantalla donde se muestran los productos.



1.1 Una vez mostrada la tabla de productos se podrá tener la elección de editar o eliminar cada uno de los registros existentes.

The screenshot shows a web application interface for updating a product. At the top, there is a navigation menu with 'Sistema de Ventas' and sub-items: 'Productos', 'Proveedores', 'Clientes', 'Categoría', and 'Tienda'. Below the menu, a light green box displays 'Código del producto a actualizar: 0001'. The main form contains several input fields: 'Nombre del producto:' with the value 'Pantalones Taña Grande', 'Descripción del producto:' with 'Pantalones talla', 'Precio del producto:' with '12', and 'ID Proveedor:' with '1'. At the bottom left, there are two buttons: a green 'Actualizar' button and a red 'Cancelar' button. At the bottom right, there is a watermark that says 'Activar Windows' and 'Ve a Configuración para activar Windows'.

### “Tabla Proveedores”

2. Si el usuario escoge un módulo a través del menú, podrá ingresar a una pantalla donde se muestran los proveedores ya existentes.

The screenshot shows the 'Sistema de Ventas' interface for managing providers. The navigation menu is the same as in the previous screenshot. Below the menu, a light green box says 'Tabla de Proveedores existentes'. There is a dark grey button labeled 'Nuevo'. Below this is a table with the following data:

Código Proveedor	Proveedor	Teléfono	Dirección	Correo	Editar	Eliminar
0001	Administrador77	76324151	San Salvador	admin@gmail.com	Editar	Eliminar
0002	Orellana-08	22229999	Puerto de La Libertad	orellana08@gmail.com	Editar	Eliminar
0003	Juana Perez	76324151	Mexicanos	juanperez@gmail.com	Editar	Eliminar

At the bottom right, there is a watermark that says 'Activar Windows' and 'Ve a Configuración para activar Windows'.

2.1 Una vez mostrada la tabla de proveedores se podrá tener la elección de editar o eliminar cada uno de los registros existentes.

Sistema de Ventas | Productos | Proveedores | Clientes | Categoría | Tienda

Código del proveedor a actualizar: 0001

Proveedor: Administrador77

Teléfono: 76324151

Dirección: San Salvador

Correo: admin@gmail.com

Actualizar Cancelar

Activar Windows  
Ve a Configuración para activar

### “Tabla Clientes”

3. Si el usuario escoge un módulo a través del menú, podrá ingresar a una pantalla donde se muestran los clientes ya existentes.

Sistema de Ventas | Productos | Proveedores | Clientes | Categoría | Tienda

Tabla de Clientes existentes

Nuevo

Cod-producto	Nombres	Apellidos	Dirección	Teléfono	Correo Electrónico	Editar	Eliminar
0001	Victor Manuel	Orellana Quijada	Zacatecoluca, La Paz	00002222	orellana606@gmail.com	Editar	Eliminar

3.1 Una vez mostrada la tabla de clientes se podrá tener la elección de editar o eliminar cada uno de los registros existentes.

**Sistema de Ventas** Productos Proveedores Clientes Categoría Tienda

Código del producto a actualizar: 0001

Nombre: Victor Manuel

Apellidos: Orellana Quijada

Dirección: Zacatecoluca, La Paz

Teléfono: 00002222

Correo Electrónico: orellana606@gmail.com

Guardar Cancelar

Activar Windows  
Ve a Configuración para activar

### “Tabla Categorías”

4. Si el usuario escoge un módulo a través del menú, podrá ingresar a una pantalla donde se muestran las categorías ya existentes.

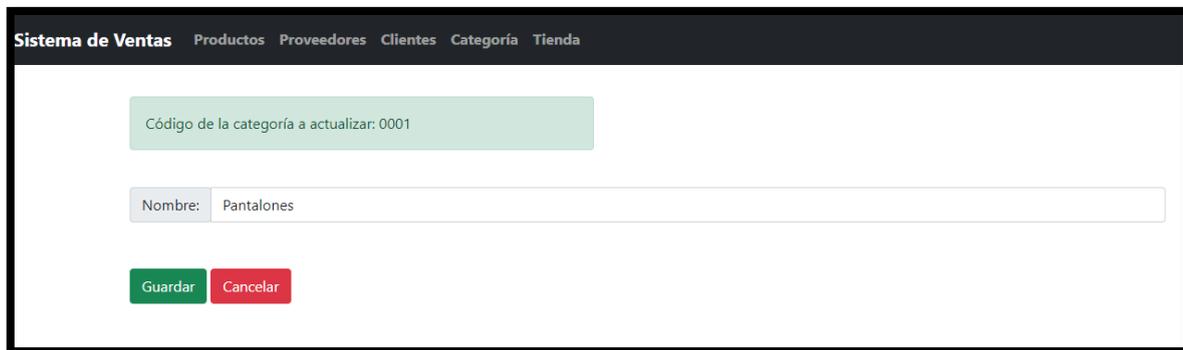
**Sistema de Ventas** Productos Proveedores Clientes Categoría Tienda

Tabla de Categorías existentes

Nuevo

Cod-categoría	Categoría	Editar	Eliminar
0001	Pantalones	Editar	Eliminar
0002	Camisas	Editar	Eliminar

4.1 Una vez mostrada la tabla de clientes se podrá tener la elección de editar o eliminar cada uno de los registros existentes.



The screenshot shows a web application interface with a dark header containing the text "Sistema de Ventas" and navigation links: "Productos", "Proveedores", "Clientes", "Categoría", and "Tienda". Below the header, there is a light green box containing the text "Código de la categoría a actualizar: 0001". Underneath this is a text input field with the label "Nombre:" and the value "Pantalones". At the bottom of the form, there are two buttons: a green "Guardar" button and a red "Cancelar" button.

## Desarrollo

El sistema se ha desarrollado con el fin de brindar un mejor entorno dentro de la empresa, ahorrando tiempo a la hora de registrar nuevos productos, categorías, clientes de igual manera, se podrán visualizar los registros existentes dentro del sistema.

### a) Técnicas de programación aplicadas.

La técnica que implementamos en nuestro sistema fue la programación estructurada, ha sido elegida porque no es complicada de entender, hace más fácil la estructura y verificación de programas se adapta perfectamente al diseño. De igual manera se hizo uso del lenguaje de programación PHP ya que es de código abierto.

#### Conceptos para la programación del sistema.

- **Secuencia:** La secuencia se utilizó para las instrucciones del programa en cual se ejecuta una después de otra, ambas con una sola entrada y una única salida.
- **Selección (bifurcación):** Equivale a la instrucción IF de todos los lenguajes de programación, también conocida como la estructura SI-CIERTO-FALSO, plantea la selección entre dos alternativas con base en el resultado de la evaluación de una condición.

- **Iteración o Repetitivas (bucle).** Repetición de una operación mientras se cumple una condición. Corresponde a la ejecución repetida de una instrucción o un conjunto de instrucciones mientras se cumpla una determinada condición.
- **Variables.** Las variables pueden considerarse las técnicas más fundamentales de programación. Está formada por un espacio en el sistema de almacenaje y un nombre simbólico que está asociado a dicho espacio.
- **Array .** El array es un tipo de datos que asocia valores con claves, se optimiza para varios usos diferentes.

## b) Herramientas de desarrollo utilizadas.

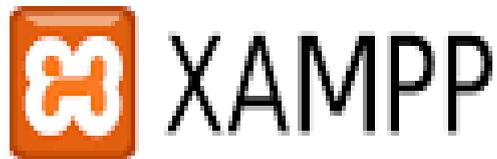
Para el desarrollo del sistema se utilizaron diferentes herramientas las cuales son las siguientes.

1. **Computadora:** Es una máquina electrónica digital programable que ejecuta una serie de comandos para procesar los datos de entrada, obteniendo

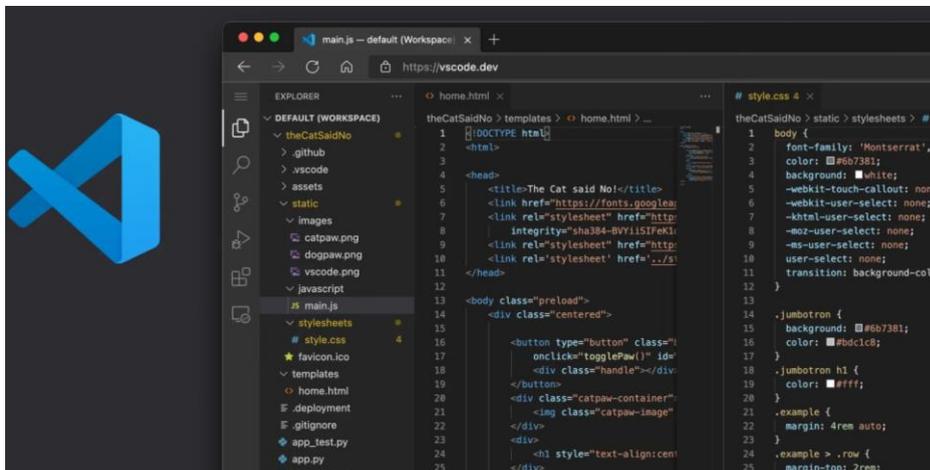


### 2. Acceso a internet.

3. **XAMPP:** XAMPP paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. En este caso solo ocuparemos Apache para poder cargar el sistema en el navegador y cómo BD ocuparemos mongodb no MySQL



4. **Visual Studio Code.** Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.



5. **Composer.** Es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP. Fue desarrollado por Nils Adermann y Jordi Boggiano quienes continúan dirigiendo el proyecto.



6. **Compass.** Es una herramienta interactiva para consultar, optimizar y analizar sus datos de MongoDB. Obtenga información clave, arrastre y suelte para crear canalizaciones y más.



## Lenguaje de programación y gestores de bases de datos u otras herramientas utilizadas en el desarrollo del proyecto.

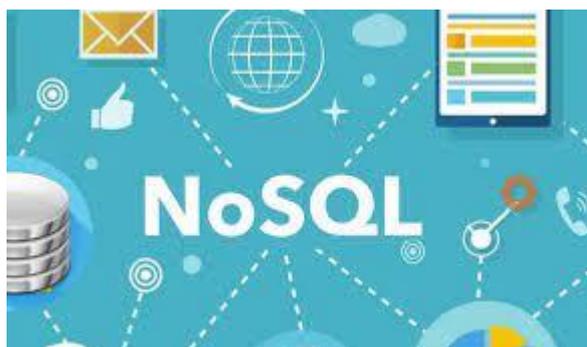
1. **PHP es un lenguaje de programación.** Uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group.



2. **Gestor de Base de Datos MongoDB.** Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda.



3. **NoSQL.** Es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales) en aspectos importantes, siendo el más destacado que no usan SQL como **lenguaje** principal de consultas.



4. **BSON** es un formato de intercambio de datos usado principalmente para su almacenamiento y transferencia en la base de datos MongoDB. Es una representación binaria de estructuras de datos y mapas. El nombre **BSON** está basado en el término JSON y significa Binary JSON (JSON Binario).

```
Python 3.6.8 (default, Jan 14 2019, 11 :02 :34)
[GCC 8.0.1 20180414 (experimental) [trunk revision 259383]] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import bson
>>> import pymongo
>>> from bson import BSON
>>> bson_example = BSON.encode({"Object" : "Rocket"})
>>> bson_example
b'\x18\x00\x00\x00\x02Object\x00\x07\x00\x00\x00Rocket\x00\x00'
>>> type(bson_example)
<class 'bson.BSON'>
>>>
```

5. **JSON.** Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

```
chrome://extensions/manifest.json
// 20150625171154
// https://api.github.com/repos/tulloz/json-viewer

{
  "id": "f263585",
  "name": "json-viewer",
  "full_name": "tullos/json-viewer",
  "version": "0",
  "update_url": "https://clients.google.com/cr/x/updates",
  "background": {
    "scripts": [
      "background.js"
    ]
  },
  "content_scripts": [
    {
      "matches": ["*://*/*"],
      "js": [
        "content.js"
      ]
    }
  ],
  "devtools_page": "devtools.html",
  "description": "It is a Chrome extension for printing JSON and JSONP.",
  "homepage_url": "https://github.com/tulloz/json-viewer",
  "icons": {
    "16": "icons/icon16.png",
    "32": "icons/icon32.png",
    "48": "icons/icon48.png",
    "96": "icons/icon96.png",
    "128": "icons/icon128.png",
    "256": "icons/icon256.png",
    "512": "icons/icon512.png"
  },
  "permissions": [
    "https://api.github.com/*",
    "https://raw.githubusercontent.com/*",
    "https://gist.github.com/*",
    "https://www.googleapis.com/*",
    "https://www.googleapis.com/auth/userinfo.email",
    "https://www.googleapis.com/auth/userinfo.profile",
    "https://www.googleapis.com/auth/userinfo.name",
    "https://www.googleapis.com/auth/userinfo.email",
    "https://www.googleapis.com/auth/userinfo.profile",
    "https://www.googleapis.com/auth/userinfo.name",
    "https://www.googleapis.com/auth/userinfo.email",
    "https://www.googleapis.com/auth/userinfo.profile",
    "https://www.googleapis.com/auth/userinfo.name"
  ],
  "optional_permissions": [
    "https://www.googleapis.com/auth/userinfo.email",
    "https://www.googleapis.com/auth/userinfo.profile",
    "https://www.googleapis.com/auth/userinfo.name"
  ],
  "type": "browser_action",
  "site_access": false,
  "private": false,
  "web_accessible_resources": [
    "icons/*",
    "content.js"
  ],
  "description": "It is a Chrome extension for printing JSON and JSONP.",
  "homepage_url": "https://github.com/tulloz/json-viewer",
  "icons": {
    "16": "icons/icon16.png",
    "32": "icons/icon32.png",
    "48": "icons/icon48.png",
    "96": "icons/icon96.png",
    "128": "icons/icon128.png",
    "256": "icons/icon256.png",
    "512": "icons/icon512.png"
  }
}
```

6. **HTML** (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Lagunaje) es el componente más básico de la Web. Define el significado y la estructura del contenido web. ... "Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web

# HTML



## Documentación

### Técnicas de prueba del sistema utilizadas o propuestas para la puesta en marcha de la aplicación

Dicho sistema que se ha presentado anteriormente, se sometió a diferentes pruebas durante el desarrollo, para el mejor funcionamiento óptico y el uso del sistema dentro de la empresa pueda ser más eficiente al momento de registrar y visualizar los datos.

# CONCLUSIONES

Con dicha actividad que ha sido desarrollada se busca que las personas interesadas en el proyecto de sistema de venta puedan interactuar de una forma más sencilla, los sistemas de ventas día a día se van innovando y las tiendas, empresas, etc., van optando por obtener estos sistemas , este sistema ha sido desarrollado en NoSQL las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y cuentan con esquemas flexibles, el actual sistema es un sistema estable con la finalidad de llevar un completo control de las ventas, no obstante el objetivo que se persigue es que los procesos se puedan realizar de forma segura y rápida.

# Recomendaciones

De acuerdo al contenido presentado anteriormente se listan algunas recomendaciones para las personas beneficiadas con el sistema de ventas:

- La principal ventaja del proyecto es brindar una mejor atención a nuestros usuarios haciendo así el trabajo para ellos más factible.
- El acceso al sistema es muy sencillo para el usuario, ya que no requiere un mayor esfuerzo de acceso

# Bibliografía

- Blog mantenimiento de una pc ( 12 de noviembre del 2011), que es xampp, obtenido de: <https://mantenimientosdeunapc.blogspot.com/2011/11/que-es-xampp-y-para-que-sirve.html>
- Wikipedia (05 de octubre de 2020), editor de texto Visual Studio Code, obtenido de: [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code)
- Styde (23 de diciembre), que es composer y cómo usarlo, obtenido de: <https://styde.net/que-es-composer-y-como-usarlo/>
- Europa-Kompass,(21 de diciembre del 2020), que es compas, obtenido de: <https://www.europa-kompass.de/es/que-es-compass>
- Blog Agencia la Nave (10 de octubre de 2013), que es php, obtenido de: <https://www.google.com/search?q=que+es+php&oq=que+es+php&aqs=chrome..69i57j0i512l9.2948i0j7&sourceid=chrome&ie=UTF-8>
- Revista digital ( 16 de abril del 2019), gestores de base más utilizados en la actualidad, obtenido de: <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>
- Wikipedia (24 de noviembre del 2021), NoSQL, obtenido de: <https://es.wikipedia.org/wiki/NoSQL>
- Wikipedia (23 de noviembre del 2021), bson, obtenido de: [https://es.wikipedia.org/wiki/BSON#:~:text=BSON%20es%20un%20formato%20de,Binary%20JSON%20\(JSON%20Binario\).](https://es.wikipedia.org/wiki/BSON#:~:text=BSON%20es%20un%20formato%20de,Binary%20JSON%20(JSON%20Binario).)
- Wikipedia (23 de noviembre del 2021), json, obtenido de: <https://es.wikipedia.org/wiki/JSON>
- Developer ( 11 de marzo del 2020), html, obtenido de: <https://developer.mozilla.org/es/docs/Web/HTML>

# Anexos

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



**TITULO:**

**Manual de usuario Sistema de ventas**

Nombre	Carnet
Liliana de Los Ángeles Monge Cerón	Mc01136242
Victor Manuel Orellana Quijada	OQ01136509
Douglas Rafael Trejo Franco	TF01136646
Ana Cecilia Martínez Valladares	MV01135209

**ASIGNATURA:**

**Bases de datos II**

**DOCENTE:**

**Lic. Irwin Guardado**

**CICLO/AÑO:**

**2-2021**

**FECHA**

**30/11/2021**

# Introducción

Hoy en día las empresas tienen una mayor participación en el mercado, los sistemas de ventas se han convertido en un elemento crítico e importante en el crecimiento de las ventas; el sistema de venta ha sido diseñado con el propósito de llevar el control interno de los movimientos u operaciones. Elegir un software y hardware apropiado para una empresa puede ofrecer muchos beneficios tales como, llevar un control y encontrar rentabilidad; por lo que es importante no esperar para explorar las opciones valiosas. A continuación, se presenta los pasos y lineamientos a seguir para la utilización y navegación dentro del sistema de ventas. Dado los objetivos propuestos para el funcionamiento del sistema, este se convierte en una herramienta de fácil acceso para sus usuarios.

## Objetivos

### Objetivo general:

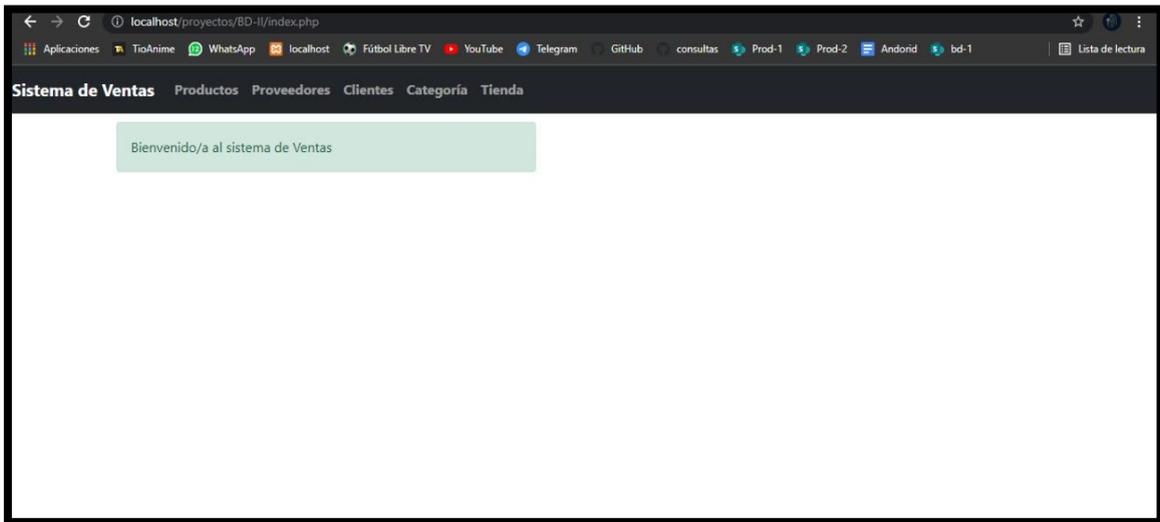
Mostrar y explicar a los usuarios el funcionamiento del sistema de ventas, incluyendo las distintas áreas de acceso y visibilidad para los mismos.

### Objetivos específicos:

- ✓ Mostrar las diferentes pantallas gráficas que muestran el proceso a realizar para cada acción al que el usuario tendrá acceso.
  
- ✓ Listar los pasos y requisitos para acceder al sistema.
  
- ✓ Fácil acceso al sistema de venta.

## Manual de usuario

Al iniciar sesión, se muestra la pantalla con todas las opciones disponibles: Productos, Proveedores, Clientes, Categorías y tienda. Cada una de estas opciones, cuenta con un objetivo principal, lo cual permite facilitar el trabajo del usuario.



La opción de Productos, muestra la lista de los mismos ingresados al sistema, cada uno con su respectiva identificación (refiriéndose a cod-producto, nombre, descripción, etc.). Con los datos ya ingresados en la tabla, el sistema cuenta con dos acciones muy importantes permite editar y eliminar si el usuario así lo desea.

The screenshot shows the 'Sistema de Ventas' dashboard with the 'Productos' menu item selected. A green box displays the message 'Tabla de Productos existentes'. Below the message is a 'Nuevo' button and a table of products.

Cod-producto	Nombre	Descripción	precio	Proveedor	Editar	Eliminar
0001	Pantalones XG	Pantalones talla	12	1	Editar	Eliminar
0002	Camisas Negras	Camisas playeras	23	2	Editar	Eliminar
0003	Zapatos del Oste	Nike/Adidas	25.70	1	Editar	Eliminar
0004	Botellas	Grandes/Medias/Pequeñas	75.00	2	Editar	Eliminar

La tabla producto cuenta con la opción de agregar y/o registrar un nuevo producto, muestra un formulario el cual me permite ingresar la información del producto deseado. En el mismo, se encuentra el botón Guardar Producto el cual permite almacenar la información en dicha tabla. De igual forma, el botón Cancelar, si el usuario ya no desea guardar el producto.

The screenshot shows the 'Nuevo Producto' form in the 'Sistema de Ventas' application. The form includes the following fields and buttons:

- Código Producto:** N°. Producto
- Nombres del producto:** Nombre del producto
- Descripción del producto:** (Text area)
- Precio del producto:** \$00.00
- ID Proveedor:** ID. Prov
- Buttons:** Guardar producto (green), Cancelar (red)

The application header shows 'Sistema de Ventas' and navigation links for 'Productos', 'Proveedores', 'Clientes', 'Categoría', and 'Tienda'. A watermark 'Activar Windows' is visible in the bottom right corner.

El botón Editar muestra el formulario con la información que previamente ha sido ingresada ,donde le permitirá al usuario cambiar uno o más datos del producto, y si el usuario ya no desea actualizarlo puede cancelar dicha acción.

The screenshot shows the 'Editar Producto' form in the 'Sistema de Ventas' application. The form includes the following fields and buttons:

- Código del producto a actualizar:** 0001
- Nombre del producto:** Pantalones XG
- Descripción del producto:** Pantalones talla
- Precio del producto:** 12
- ID Proveedor:** 1
- Buttons:** Actualizar (green), Cancelar (red)

The application header shows 'Sistema de Ventas' and navigation links for 'Productos', 'Proveedores', 'Clientes', 'Categoría', and 'Tienda'. A watermark 'Activar Windows' is visible in the bottom right corner.

El botón Eliminar, permite quitar un producto de la tabla si el usuario lo considera necesario. Es necesario recalcar que en todas las tablas del sistema de venta estas acciones están disponibles en todas las tablas. En cada tabla, dichas acciones de editar y eliminar tienen diferentes funciones, de acuerdo a lo que el sistema solicite.

## PROVEEDORES

Esta opción muestra la lista de los proveedores encargados de surtir al sistema de ventas. De igual forma, se muestra el botón Agregar, el cual despliega un formulario para guardar la información de dicho proveedor.

Sistema de Ventas [Productos](#) [Proveedores](#) [Clientes](#) [Categoría](#) [Tienda](#)

Tabla de Proveedores existentes

Nuevo

Código Proveedor	Proveedor	Teléfono	Dirección	Correo	Editar	Eliminar
0001	Administrador77	76324151	San Salvador	admin@gmail.com	Editar	Eliminar
0002	Orellana-08	22229999	Puerto de La Libertad	orellana08@gmail.com	Editar	Eliminar
0003	Juana Perez	76324151	Mexicanos	juanperez@gmail.com	Editar	Eliminar

Sistema de Ventas [Productos](#) [Proveedores](#) [Clientes](#) [Categoría](#) [Tienda](#)

Código del proveedor a actualizar: 0001

Proveedor:

Teléfono:

Dirección:

Correo:

Activar Windows

## CLIENTES

Muestra la lista de los clientes que han realizado una compra. Aunque el sistema está diseñado para facilitar el trabajo interno, la tabla cliente, pretende guardar la información de los clientes que necesiten cambio de un producto, encargo o devolución. También se muestra el botón Nuevo, el cual facilita al usuario guardar la información de un nuevo cliente, y además cuenta con la opción de editar o eliminar un cliente.

Sistema de Ventas [Productos](#) [Proveedores](#) [Clientes](#) [Categoría](#) [Tienda](#)

Tabla de Clientes existentes

Nuevo

Cod-producto	Nombres	Apellidos	Dirección	Teléfono	Correo Electrónico	Editar	Eliminar
0001	Victor Manuel	Orellana Quijada	Zacatecoluca, La Paz	00002222	orellana606@gmail.com	Editar	Eliminar
00026	VictorManuel	Orellana	asdasdasd	1111-1111	cookie011@gmail.com	Editar	Eliminar

Sistema de Ventas [Productos](#) [Proveedores](#) [Clientes](#) [Categoría](#) [Tienda](#)

**Nuevo Cliente**

Código del Cliente: N°. Cliente

Nombres: Nombre del cliente

Apellidos: Apellido del cliente

Dirección:

Teléfono: N° teléfono

Correo Electrónico: correo@gmail.com

Guardar Cliente Cancelar

Activar Windows  
Ve a Configuración para activar Windows.

Sistema de Ventas [Productos](#) [Proveedores](#) [Clientes](#) [Categoría](#) [Tienda](#)

Código del producto a actualizar: 0001

Nombre: Victor Manuel

Apellidos: Orellana Quijada

Dirección: Zacatecoluca, La Paz

Teléfono: 00002222

Correo Electrónico: orellana606@gmail.com

Guardar Cancelar

Activar Windows  
Ve a Configuración para activar Windows.

## CATEGORIA:

La tabla categoría contiene sus propios atributos como son pantalones, camisas jeans. También se muestra el botón Nuevo, el cual facilita al usuario guardar la información de una nueva categoría, y además cuenta con la opción de editar o eliminar un cliente.

Sistema de Ventas Productos Proveedores Clientes Categoría Tienda

Tabla de Categorías existentes

Nuevo

Cod-categoría	Categoría	Editar	Eliminar
0001	Pantalones	Editar	Eliminar
0002	Camisas	Editar	Eliminar
0003	Jeans	Editar	Eliminar

Sistema de Ventas Productos Proveedores Clientes Categoría Tienda

### Nueva Categoría

Código categoría:

Categoría:

Sistema de Ventas Productos Proveedores Clientes Categoría Tienda

Código de la categoría a actualizar: 0001

Nombre:

# Manual del programador

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



**TITULO:**

**Manual de usuario Sistema de ventas**

Nombre	Carnet
Liliana de Los Ángeles Monge Cerón	Mc01136242
Victor Manuel Orellana Quijada	OQ01136509
Douglas Rafael Trejo Franco	TF01136646
Ana Cecilia Martínez Valladares	MV01135209

**ASIGNATURA:**

**Bases de datos II**

**DOCENTE:**

**Lic. Irwin Guardado**

**CICLO/AÑO:**

**2-2021**

**FECHA**

**30/11/2021**

## Introducción.

El propósito de este manual del programador, es dar a conocer al lector los códigos fuentes del programa realizado. Para ello tratamos de la forma más concisa de explicar cada uno de los códigos, junto con la programación utilizada en el desarrollo del sistema, esto con el fin de que el usuario pueda modificar a gusto alguno de los valores y parámetros de las funciones que se encuentran expuestas en la programación de sistema inventario de venta.

## Objetivos

### Objetivos General.

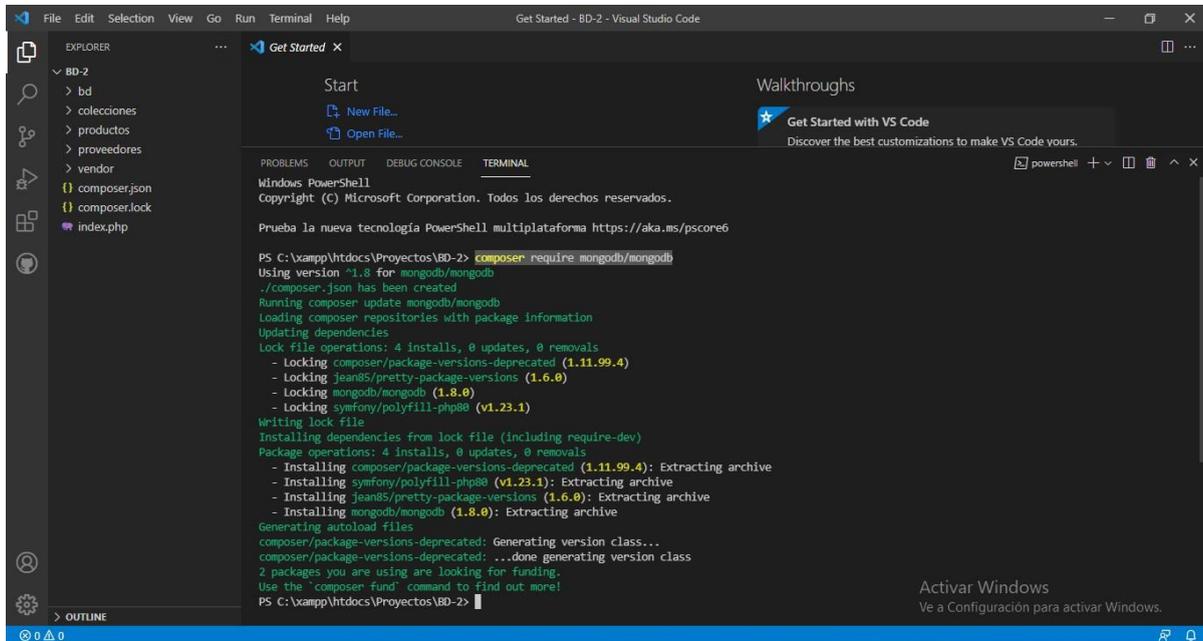
- El desarrollo del manual del programador es dar a conocer a los usuarios los códigos fuentes del control y venta del sistema y mostrar los pasos a seguir para la configuración del sistema.

### Objetivos Específicos.

- El desarrollo del manual del programador instruirá en el uso adecuado del sistema de ventas, para el acceso oportuno y adecuado del inventario mismo.
- El desarrollo del manual del programador mostrará los pasos a seguir en el proceso de configuración del sistema, así como la descripción de los archivos relevantes del sistema.
- El desarrollo del manual del programador se orientará en la configuración y soporte del control y venta del mismo.

- El desarrollo del sistema de venta está dividido en tres partes importantes para el funcionamiento del sistema y funcionan de la siguiente manera.
  - **Visual Studio Code.** Visual será nuestro gestor de código quien nos ayudará a generar el código fuente del sistema.
  - **Xampp.** Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache quien nos ayudará con el sistema.
  - **MongoDB.** Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON.
  - **Composer.** Es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP.
  - **Compass.** Es una herramienta interactiva para consultar, optimizar y analizar sus datos de MongoDB. Obtenga información clave, arrastre y suelte para crear canalizaciones y más.

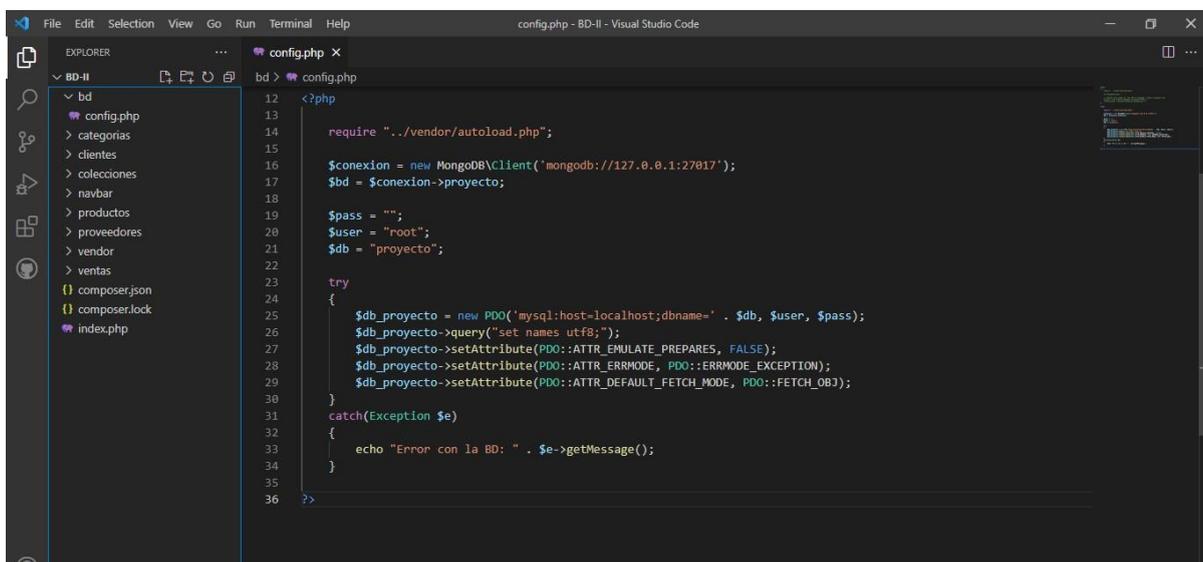
1. Antes de comenzar a desarrollar un sistema ocupa como gestor de BD mongodb, debemos ocupar el siguiente comando **composer require mongodb/mongodb** en cual nos instalará los archivos que nos harán posible comenzar a desarrollar nuestro sistema. El cual debemos irnos al apartado de terminal/new terminal y colocaremos el comando anteriormente mencionado,



```
PS C:\xampp\htdocs\Proyectos\BD-2> composer require mongodb/mongodb
Using version ^1.8 for mongodb/mongodb
./composer.json has been created
Running composer update mongodb/mongodb
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
  - Locking composer/package-versions-deprecated (1.11.99.4)
  - Locking jean85/pretty-package-versions (1.6.0)
  - Locking mongodb/mongodb (1.8.0)
  - Locking symfony/polyfill-php80 (v1.23.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
  - Installing composer/package-versions-deprecated (1.11.99.4): Extracting archive
  - Installing symfony/polyfill-php80 (v1.23.1): Extracting archive
  - Installing jean85/pretty-package-versions (1.6.0): Extracting archive
  - Installing mongodb/mongodb (1.8.0): Extracting archive
Generating autoload files
composer/package-versions-deprecated: Generating version class...
composer/package-versions-deprecated: ...done generating version class
2 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
PS C:\xampp\htdocs\Proyectos\BD-2>
```

y nos quedaría de la siguiente manera.

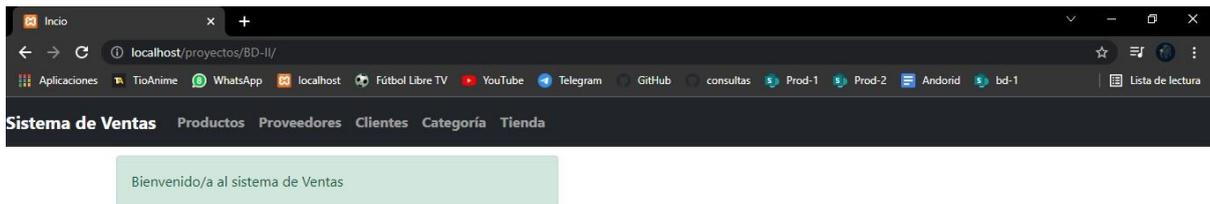
2. Una vez instalados los archivos comenzamos a hacer la conexión de la DB de mongodb. Si tenemos instalado el programa **Studio 3T** directamente podemos sacar la conexión de mongodb con php. nuestra BD se llama proyecto en la cual, se guardarán los registros que ingresemos por medio del sistema.



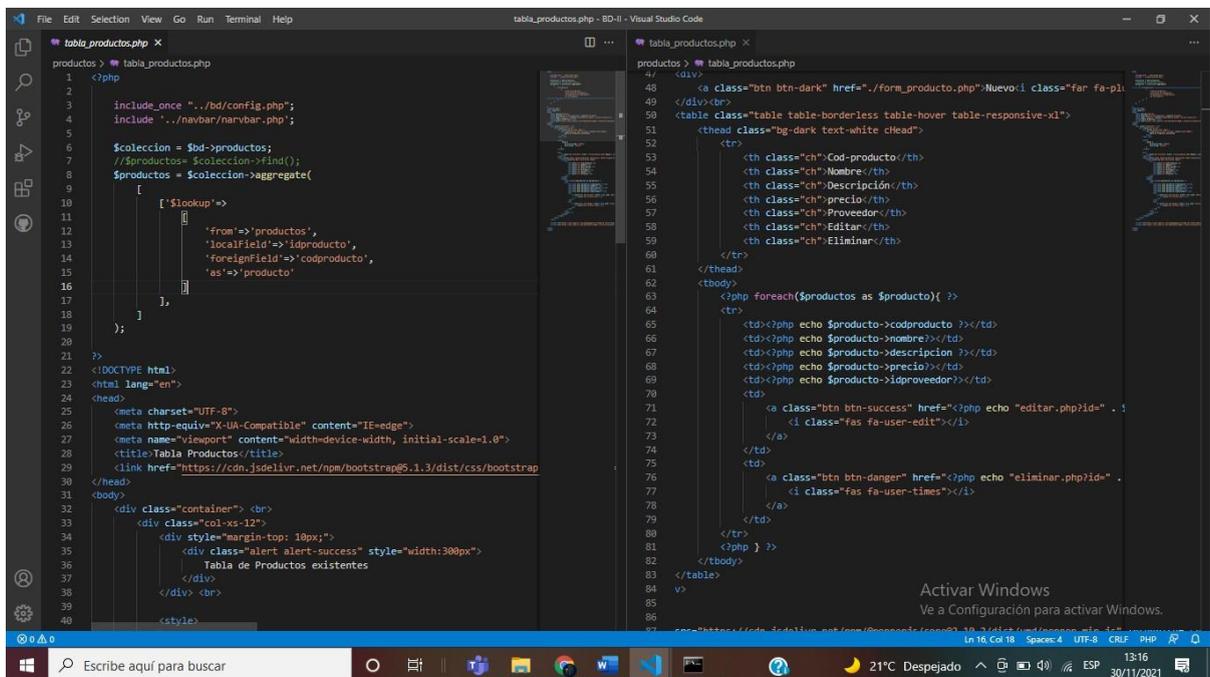
```
12 <?php
13
14 require "../vendor/autoload.php";
15
16 $conexion = new MongoClient("mongodb://127.0.0.1:27017");
17 $bd = $conexion->proyecto;
18
19 $pass = "";
20 $user = "root";
21 $db = "proyecto";
22
23 try
24 {
25     $db_proyecto = new PDO("mysql:host=localhost;dbname=". $db, $user, $pass);
26     $db_proyecto->query("set names utf8;");
27     $db_proyecto->setAttribute(PDO::ATTR_EMULATE_PREPARES, FALSE);
28     $db_proyecto->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
29     $db_proyecto->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
30 }
31 catch(Exception $e)
32 {
33     echo "Error con la BD: " . $e->getMessage();
34 }
35
36 >>
```



5. Dicho sistema se podrá ver en el navegador de la siguiente manera.

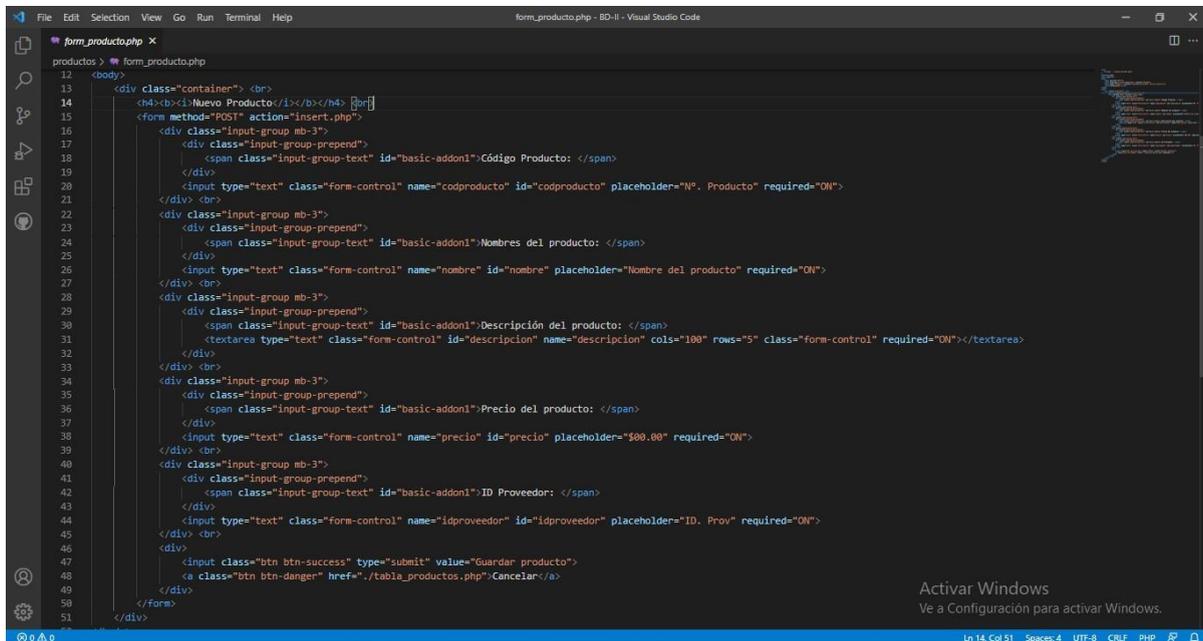


6. En el caso de este sistema no ocuparemos el gestor de BD MySQL. Si no que MongoDB/NoSQL dicho esto, para traer algún registro no ocuparemos las sentencias sql, si no que las de NoSQL ... Como podemos observar tenemos abiertas dos pestañas lo hago de la siguiente manera por que el código es un poco largo. En la primera pestaña incluimos la conexión y el navbar. Después creamos una variable que contendrá la BD seguidamente la colección. Nuevamente creamos otra variable que contendrá la variable anterior seguida del operador aggregate quien procesa los registros de datos y devuelven resultados calculados. En este caso nos traerá los registros que ingresamos en la colección productos. En la segunda parte hacemos un foreach (array) en la cual mandamos a llamar la variable que contiene todo el proceso y le damos



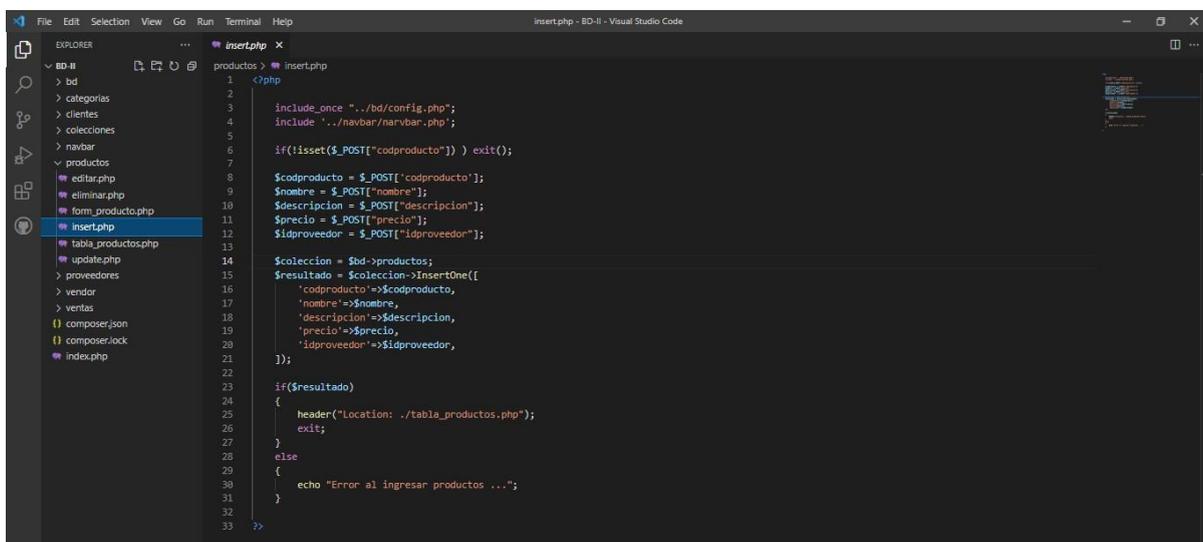
un alias (AS) y con ese alias definido mandamos a traer los demás datos.

7. En el archivo form\_producto.php creamos el formulario donde el usuario o administrador deberá llenar los campos que se le soliciten. Este una vez llenado nos direccionará al archivo insert.php donde está la lógica que nos permitirá insertar los datos en la BD y mostrarlos.



```
12 <body>
13 <div class="container"> <br>
14 <h3><b><i>Nuevo Producto</i></b></h3></div>
15 <form method="POST" action="insert.php">
16 <div class="input-group mb-3">
17 <div class="input-group-prepend">
18 <span class="input-group-text" id="basic-addon1">Código Producto: </span>
19 </div>
20 <input type="text" class="form-control" name="codproducto" id="codproducto" placeholder="Nº. Producto" required="ON">
21 </div> <br>
22 <div class="input-group mb-3">
23 <div class="input-group-prepend">
24 <span class="input-group-text" id="basic-addon1">Nombres del producto: </span>
25 </div>
26 <input type="text" class="form-control" name="nombre" id="nombre" placeholder="Nombre del producto" required="ON">
27 </div> <br>
28 <div class="input-group mb-3">
29 <div class="input-group-prepend">
30 <span class="input-group-text" id="basic-addon1">Descripción del producto: </span>
31 <div class="input-group-append">
32 <div class="input-group-text" id="basic-addon1">
33 </div>
34 </div> <br>
35 <div class="input-group mb-3">
36 <div class="input-group-prepend">
37 <span class="input-group-text" id="basic-addon1">Precio del producto: </span>
38 </div>
39 <input type="text" class="form-control" name="precio" id="precio" placeholder="$00.00" required="ON">
40 </div> <br>
41 <div class="input-group mb-3">
42 <div class="input-group-prepend">
43 <span class="input-group-text" id="basic-addon1">ID Proveedor: </span>
44 </div>
45 <input type="text" class="form-control" name="idproveedor" id="idproveedor" placeholder="ID. Prov" required="ON">
46 </div> <br>
47 <input class="btn btn-success" type="submit" value="Guardar producto">
48 <a class="btn btn-danger" href="./tabla_productos.php">Cancelar</a>
49 </div>
50 </form>
51 </div>
```

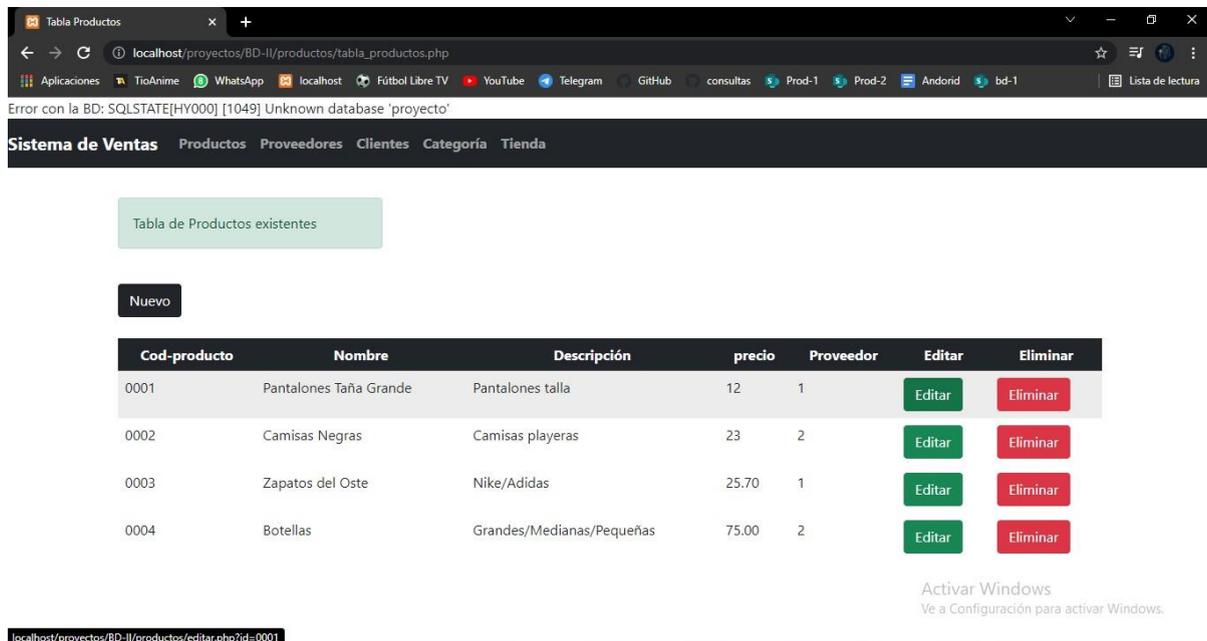
8. Una vez estamos en el archivo insert.php incluimos la conexión de BD y el navbar primero ocupamos el **isset** en cual nos determina si una variable está definida y no es null. Seguidamente mandamos a llamar el nombre de cada campo del formulario por el método \$\_POST, después creamos una variable que contenga la BD y la colección, seguidamente creamos una variable nueva que contendrá la variable anterior ocupando el operador insertOne el cual nos



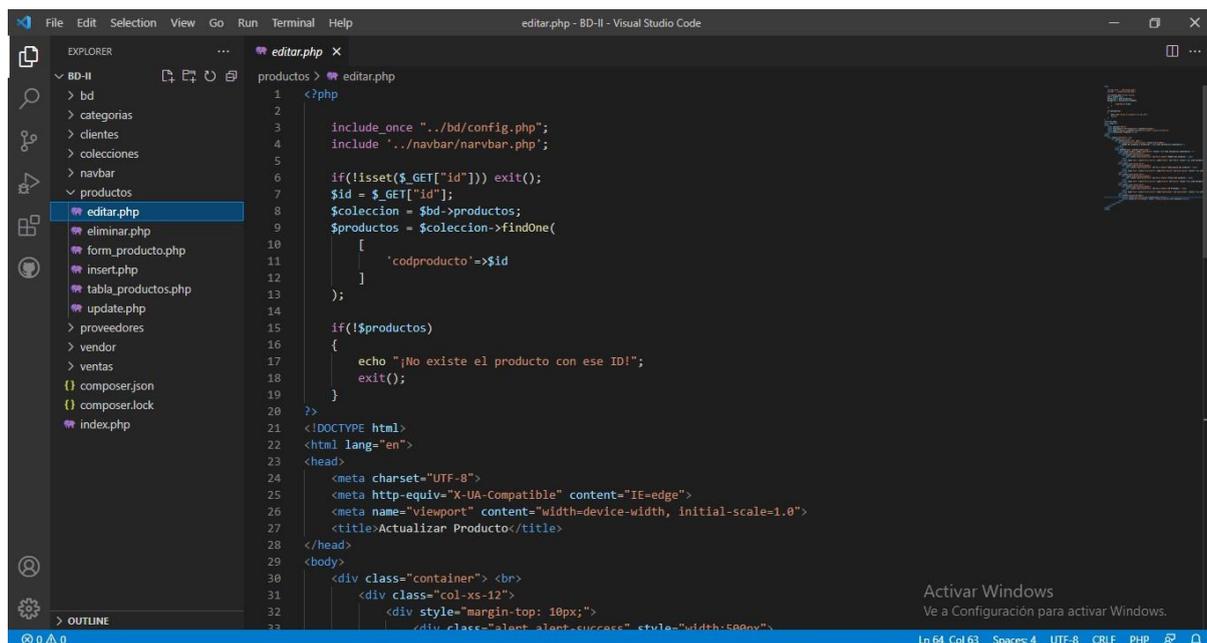
```
1 <?php
2
3 include_once "../bd/config.php";
4 include "../navbar/navbar.php";
5
6 if(!isset($_POST["codproducto"])) exit();
7
8 $codproducto = $_POST["codproducto"];
9 $nombre = $_POST["nombre"];
10 $descripcion = $_POST["descripcion"];
11 $precio = $_POST["precio"];
12 $idproveedor = $_POST["idproveedor"];
13
14 $coleccion = $bd->productos;
15 $resultado = $coleccion->insertOne([
16     'codproducto'=>$codproducto,
17     'nombre'=>$nombre,
18     'descripcion'=>$descripcion,
19     'precio'=>$precio,
20     'idproveedor'=>$idproveedor,
21 ]);
22
23 if($resultado)
24 {
25     header("Location: ../tabla_productos.php");
26     exit;
27 }
28 else
29 {
30     echo "Error al ingresar productos ...";
31 }
32
33 ?>
```

ayudará a registrar los datos en la BD.

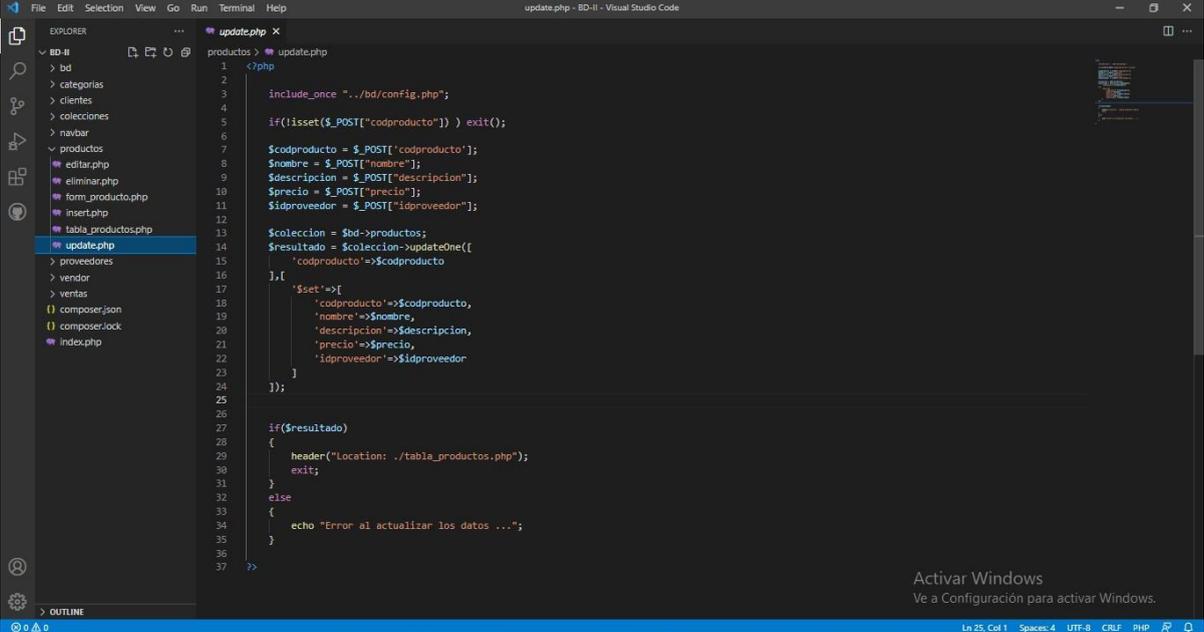
10. Si la condición se cumple está nos direccionará al tabla\_productos.php donde nos mostrará los datos de la siguiente manera. El cual contiene dos botones para editar como eliminar.



11. Una vez presionado el botón de actualizar, pero visto por medio de código, incluimos los archivos necesarios, hacemos uso del isset nuevamente, y traemos el id por el método \$\_GET, creamos otra variable que contendrá la variable anterior donde ocupa el operador findOne el cual le decimos que nos traiga un solo registro, siempre y cuando el codproducto sea igual al id. y En el formulario, mandamos a traer los datos para poderlos actualizar



12. Una vez actualizados los datos presionamos el botón actualizar quien nos dirige al archivo update.php, quien incluye los archivos de la conexión ocupando nuevamente el isset traemos los archivos actualizados por el método \$\_POST. creamos una variable que contenga la BD y la colección, seguidamente creamos una variable nueva que contendrá la variable anterior ocupando el operador updateOne donde le decimos que me actualice un registro siempre y cuando el codproducto sea igual codproducto también ocupamos el operador \$set si esto se cumple los datos se



```
1 <?php
2
3 include_once "../bd/config.php";
4
5 if(!isset($_POST["codproducto"]) ) exit();
6
7 $codproducto = $_POST["codproducto"];
8 $nombre = $_POST["nombre"];
9 $descripcion = $_POST["descripcion"];
10 $precio = $_POST["precio"];
11 $idproveedor = $_POST["idproveedor"];
12
13
14 $coleccion = $bd->productos;
15 $resultado = $coleccion->updateOne([
16     'codproducto'=>$codproducto
17 ],[
18     '$set'=>[
19         'codproducto'=>$codproducto,
20         'nombre'=>$nombre,
21         'descripcion'=>$descripcion,
22         'precio'=>$precio,
23         'idproveedor'=>$idproveedor
24     ]
25 ]);
26
27 if($resultado)
28 {
29     header("Location: ../tabla_productos.php");
30     exit;
31 }
32 else
33 {
34     echo "Error al actualizar los datos ...";
35 }
36
37 ?>
```

actualizarán sin ningún problema.

13. Por último para terminar el CRUD del modal de producto, tener el archivo eliminar.php quien incluye los archivos de la conexión ocupando nuevamente el isset traemos el \_id por el método \$\_GET. creamos una variable que contenga la BD y la colección, seguidamente creamos una variable nueva que contendrá la variable anterior ocupando el operador deleteOne quien nos elimina un registro cuando el codproducto sea igual al id

```
1 <?php
2
3 include_once "../bd/config.php";
4
5 if(!isset($_GET["id"])) exit();
6
7 $id = $_GET["id"];
8 $coleccion = $bd->productos;
9 $resultado = $coleccion->deleteOne(
10     [
11         'codproducto'=>$id
12     ]
13 );
14
15 if($resultado)
16 {
17     header("Location: ../tabla_productos.php");
18     exit;
19 }
20 else
21 {
22     echo "Error al eliminar productos existente ...";
23 }
24
25 >>
```

BD proyecto con las colecciones existentes dentro de ella.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
categorias	2	69.5 B	139.0 B	1	36.9 KB	
clientes	1	196.0 B	196.0 B	1	36.9 KB	
productos	4	143.5 B	574.0 B	1	36.9 KB	
proveedores	3	149.7 B	449.0 B	1	36.9 KB	
ventas	2	246.0 B	492.0 B	1	36.9 KB	

## Conclusión.

Como se observó en todo el manual el modal de producto cumple con las funciones de un CRUD, cada modal como el de proveedor, productos, cliente, categorías y ventas están separadas por carpetas ... Y dichas carpetas, contiene los archivos que hacen posible que dicho CRUD se cumpla en cada modal, lo único que cambiaría en cada modal es el nombre de la colección y los datos que se han creado en ella.