Manual Técnico

Software completo de Farmacia Bethesda

Tabla de contenido

Introducción....4
Informaciòn destacada....4
Objetivos....4
Requerimientos....5
Instalacion y configuracion....6
Módulos del Software....7
Definición de Módulos....7
Clases del Código:....9
Diagrama de Clases:....9
Diagramas de Actividades....10
Diagrama de Inicio de Sesión....10
Diagrama de Módulo Selección....11
Diagrama de Módulo de Ventas....12
Diagrama de Módulo de Inventarios....13
Diagrama de Módulo de Usuarios....14
Diagrama de Módulo de Distribuidoras y Marcas....15
Diagramas de Casos de Uso....16
· Diagrama de Casos de Uso: Inicio de Sesión....16
· Diagrama de Casos de Uso: Selección....17
· Diagrama de Casos de Uso: Ventas....18
· Diagrama de Casos de Uso: Inventarios....19
· Diagrama de Casos de Uso: Administración de Usuarios....20
· Diagrama de Casos de Uso: Administración de Distribuidoras y Marcas....21
Solución de los Módulos....22
1. Módulo de Inicio de Sesión....22
2. Módulos de CRUD (Inventarios, Usuarios, Marcas y Distribuidoras)....25
3. Encriptación y comprobación de contraseñas en el módulo “Usuarios”....29
4. Validación de correos electrónicos “Usuarios”....29
5. Inserción de imágenes en los módulos CRUD....29
6. Módulo de Ventas....30
7. Módulo de Vista de Ventas....33
Estructura raíz programa....36
Directorio del programa....36

Estructura raíz Pagina 38
Directorio de la Página 38
Funcionamiento de la página web 39

Introducción

Este documento tiene la finalidad de explicar el funcionamiento, el desarrollo y el uso para el usuario de la aplicación. En esta guía se dará a conocer los aspectos técnicos del funcionamiento de la aplicación desde un punto de vista externo para familiarizar al personal con la instalación, solución de problemas y configuración del programa.

Informaciòn destacada

“Farmacia Bethesda” es una empresa dedicada a la venta de medicamentos de primera necesidad con un enfoque principal en la salud de la mujer ya que además de contar con medicina para todo público desde niños hasta ancianos destinadas a tratar diversos tipos de enfermedades también cuentan con medicinas enfocadas completamente en la mujer y en sus distintas necesidades biológicas.

Objetivos

Instruir al usuario en el uso correcto de la aplicación de “Farmacias Bethesda” garantizando que el ingreso de los datos sea el correcto y el óptimo para el funcionamiento correcto del programa también garantizar que el uso de las tablas y la organización del inventario sean correctos para evitar problemas con la optimización del programa.

Requerimientos

El sistema puede ser instalado en cualquier sistema operativo que cumpla con los siguientes requerimientos:

Administrador de servidores “XAMPP” o “WAMPP”
Servidor Web Apache 2.0 o Superior
Servidor SQL
PHP 7.0 o Superior
PHPMYADMIN
Navegador Web

Instalación y configuración

Si bien existen diversos métodos de instalación para el programa con base a visual studio esta es la forma más completa y más segura y completa para el usuario y el desarrollador.

Esta es la opción más recomendada ya que es la que permite una actualización mas rápida y exacta además de un funcionamiento más ágil del programa evitándose así tiempos de carga muy prolongados mejorando la experiencia de uso para el usuario.

El método a continuación está enfocado a Sistemas Windows en el cual se tiene XAMPP, Visual code y el visual studio instalado.

Para realizar los siguientes pasos es necesario que previamente haya importado la base de datos "registrofarmacia" en el programa "XAMPP" de su computadora.

- **Programa**

1. Descomprima la carpeta .zip "Farmacia Bethesda versión final" .
2. Ingrese a la carpeta "FarmaciaBethesda" .
3. Inicie los servidores "Apache" y "MySql" desde el panel de control del programa XAMPP.
4. Ejecute el programa "Farmacia Bethesda" en su acceso directo

- **Página**

1. En la carpeta ya descomprimida "Farmacia Bethesda versión final" ingrese a la carpeta "página"
2. Inicie los servidores "Apache" y "MySql" desde el panel de control del programa XAMPP.
3. Diríjase a su navegador y busque "localhost/dashboard/"
4. Reemplace esta última por "localhost/php_bethesda/index.php"

Módulos del Software

El programa orientado a la administración de ventas e inventarios de la Farmacia Bethesda, consiste en diferentes módulos de código. Estos a su vez contienen clases destinadas a ejecutar funciones por separado, permitiendo así un funcionamiento correcto del programa conectado a la base de datos creada desde el administrador en php de MySQL.

Los módulos del programa son los siguientes, separados por su área de enfoque:

1. Módulos de administración de base de datos (Inventarios de medicamentos y tabla de empleados del negocio)
    - Módulo de Inventarios
    - Módulo de Distribuidoras
    - Módulo de Marcas
    - Módulo de Administración de Usuarios
2. Módulos de Ventas
    - Módulo de Realización de Ventas
    - Módulo de Lectura, Facturación y Eliminación de Ventas
3. Módulos de Funcionamiento y Seguridad
    - Módulo de Menú de Selección
    - Módulo de Inicio de Sesión

Definición de Módulos

Cada módulo tiene un propósito específico dentro del software. A continuación se definirá cada uno de ellos.

1. **Módulos de administración de base de datos (Inventarios de medicamentos y tabla de empleados del negocio)**

- **Administración de usuarios (Administradores)**

En este Módulo se tendrá acceso a un tratamiento CRUD donde se podrá administrar el registro de empleados de la empresa con toda su información como la contraseña del usuario, su rango (administrador o vendedor), el nombre de la persona a la que pertenece, su nombre de usuario, dirección de correo electrónico, identificación única, número de teléfono, puesto, correo electrónico y una imagen del usuario. En este módulo también se podrán eliminar usuarios, actualizar su información y crear nuevos usuarios de diferentes rangos.

- **Inventario (Administradores)**

En este módulo, el usuario (administrador) tendrá acceso a un tratamiento CRUD del inventario de medicamentos donde se podrá tener un control de la existencia de los medicamentos, sus características, el tipo de síntomas que trata, el número de serie, la marca y se podrán organizar por orden alfabético además de que se podrán buscar por nombre. También se podrá agregar, modificar y eliminar medicamentos según las necesidades del usuario.

- **Administración de distribuidoras (Administradores)**

En este módulo, el usuario (administrador) tendrá acceso a tratamientos CRUD tanto de las distribuidoras de productos de medicina, como de las marcas originales de los mismos. En el tratamiento CRUD concerniente a las distribuidoras, se incluirán campos de entrada de datos como el nombre de la distribuidora, el departamento del establecimiento, la dirección del establecimiento y su número de teléfono. En este módulo se podrán buscar los datos de distribuidoras, insertar nuevos datos y actualizar o borrar datos ya existentes.

- **Administración de marcas (Administradores)**

En el tratamiento CRUD concerniente a las marcas se incluirán campos como el nombre de la marca, el país de origen y el logotipo. En este módulo se podrán buscar los datos de marcas, insertar nuevos datos y actualizar o borrar datos ya existentes.

2. **Módulos de Ventas**

- **Ventas (vendedores o administradores)**

En este Módulo se accederá a un formulario con los datos del pedido que el cliente le pedirá al empleado. El empleado podrá consultar por medicamentos y sus existencias además de que cada vez que se agregue cierto producto a la venta, su precio se sumará al monto del pedido y cuando el pedido sea finalizado se descontará la cantidad de productos vendidos al inventario en el número de existencias. En el formulario de este módulo se pedirán datos del cliente (nombre, NIT, número telefónico), cantidad e identificación de medicamentos a vender y se auto completará el campo correspondiente al usuario encargado de la venta. Al final se mostrará una cuenta con el precio final con el IVA incluido.

- **Vista de Ventas (vendedores o administradores)**

En este módulo se mostrará la tabla “ventas” de la base de datos, permitiendo al usuario (administrador o vendedor), buscar ventas según serie o nombre del cliente, así como también mostrar la factura de una venta seleccionada, o eliminarla.

3. **Módulos de Funcionamiento y Seguridad**

- **Login**

En este módulo se le pedirá el usuario y contraseña al usuario para así poder acceder al siguiente módulo de selección con el botón “Login”; con el botón “Logout” el usuario podrá salir del programa. En caso de que el usuario o contraseña no coinciden con lo registrado en la base de datos, el programa no concederá acceso a la persona.

- **Menú de selección**

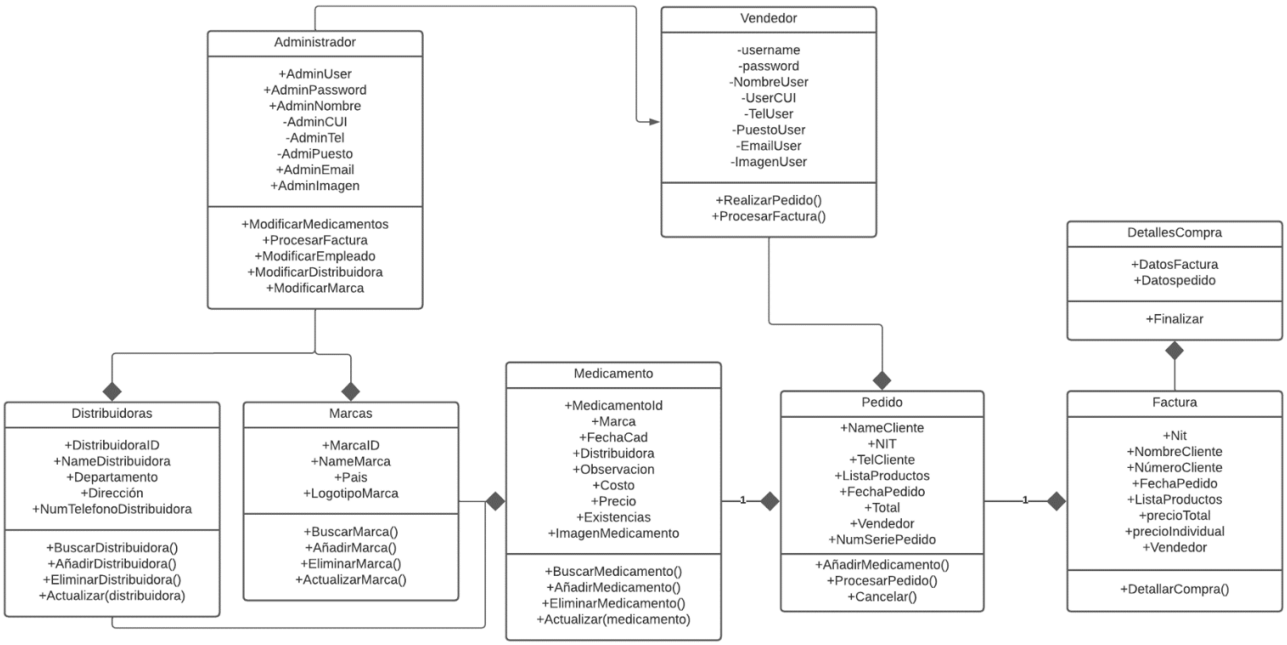
En este módulo, el usuario de rango administrador tendrá la opción de usar los botones de “Volver”, “Salir”, “Administrar Inventario”, “Administrar Usuarios”, “Ventas” y “Distribuidoras y Marcas” donde al seleccionar uno de ellos, tendrá acceso a un módulo dependiendo del botón que haya sido seleccionado; en el caso de el botón “volver” se regresa

al módulo anterior de inicio de sesión y en el caso del botón “salir” el programa se cerrará. Los usuarios de rango vendedor tendrán únicamente la opción de seleccionar el botón de “Ventas”, para ingresar a dicho módulo.

Clases del Código:

Dentro de los módulos anteriores, se encuentran las clases que hacen referencia a los objetos de la programación. Los cuales, estan descritos en el siguiente diagrama de clases:

Diagrama de Clases:



Diagramas de Actividades

Respectivamente, los diagramas de actividades describen el funcionamiento del software dependiendo de las decisiones del usuario al ejecutar instancias de código. En este caso, se describirá con un diagrama de actividades a cada módulo descrito anteriormente.

Diagrama de Inicio de Sesión

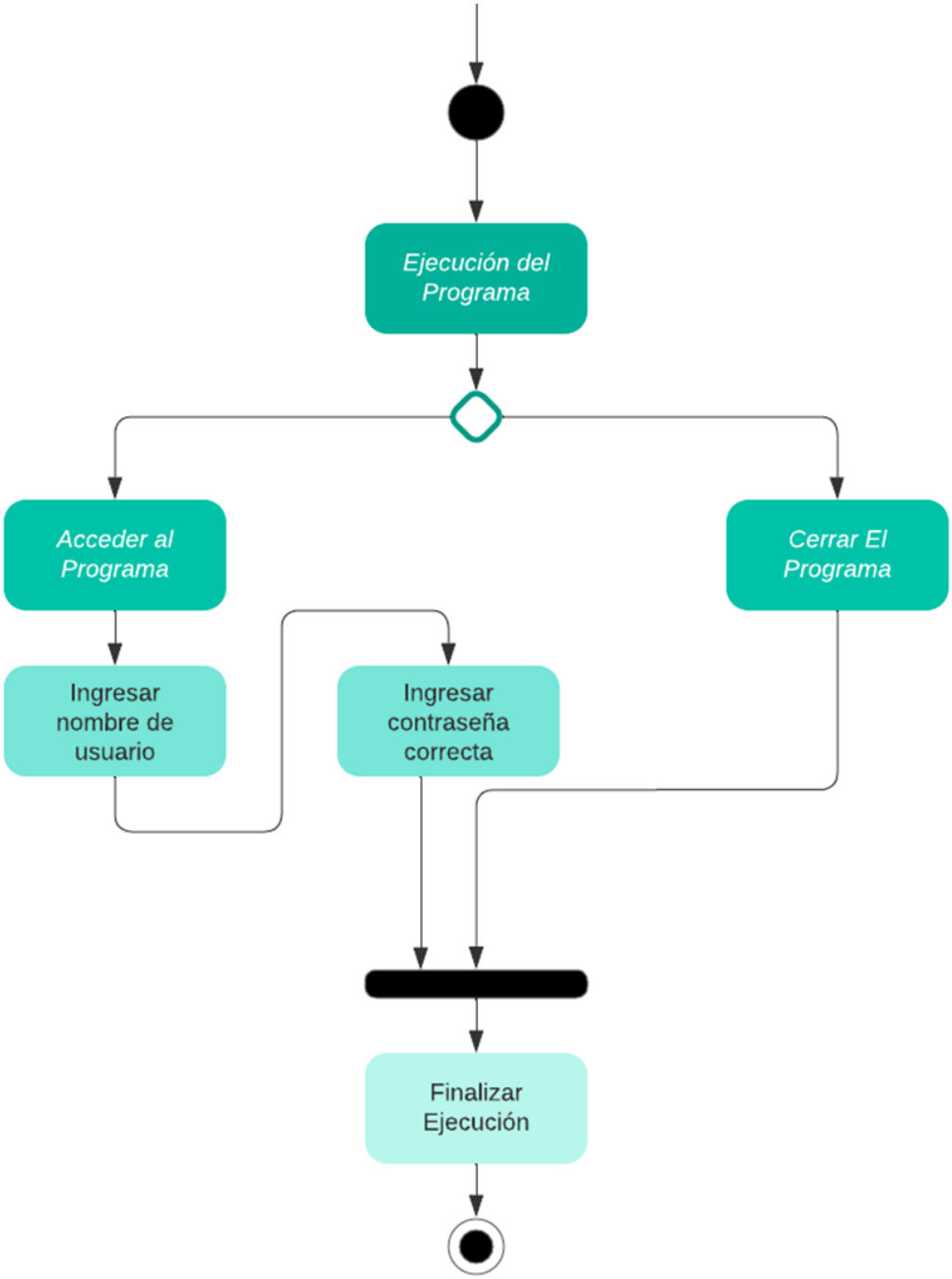


Diagrama de Módulo Selección

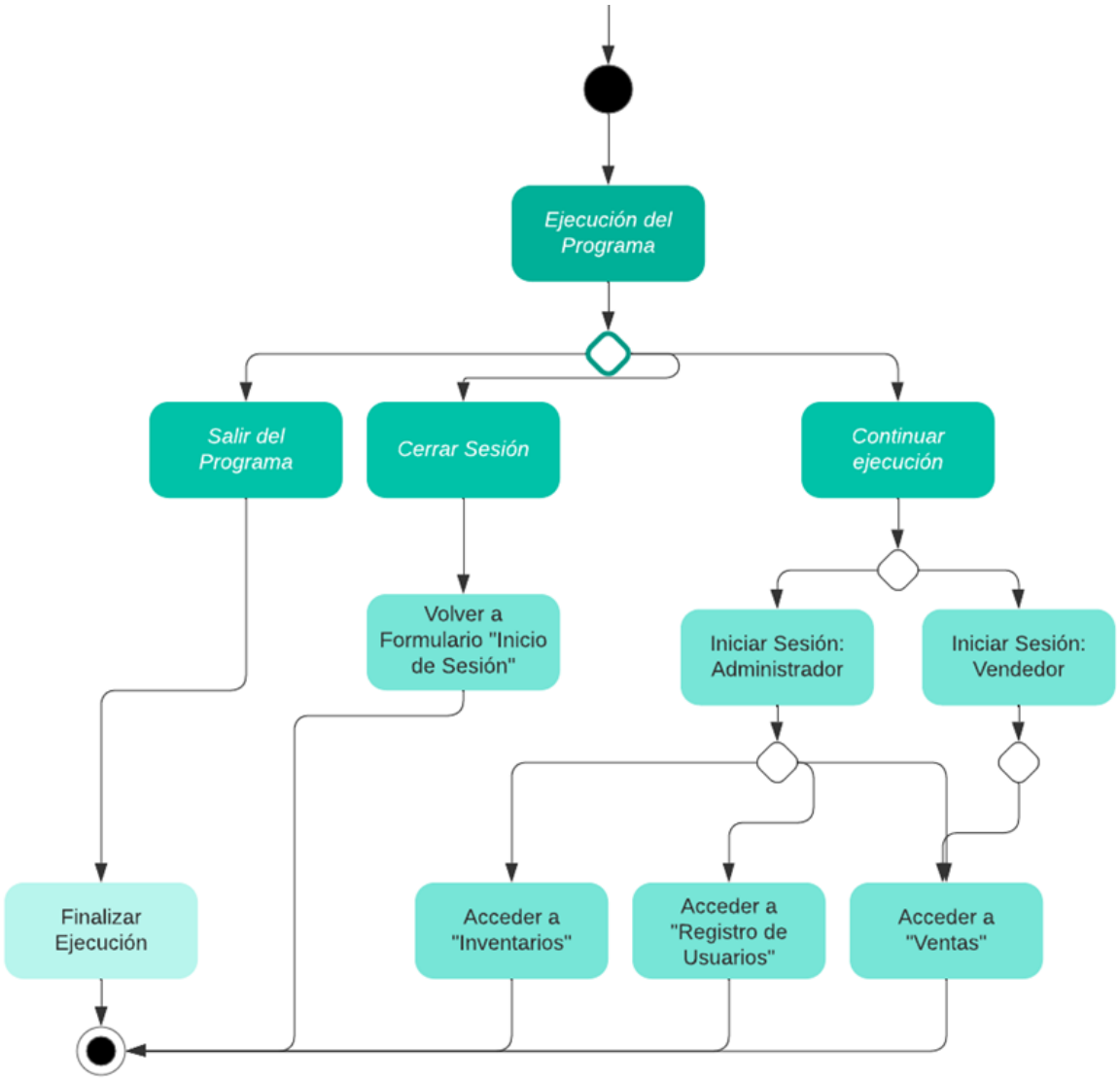


Diagrama de Módulo de Ventas

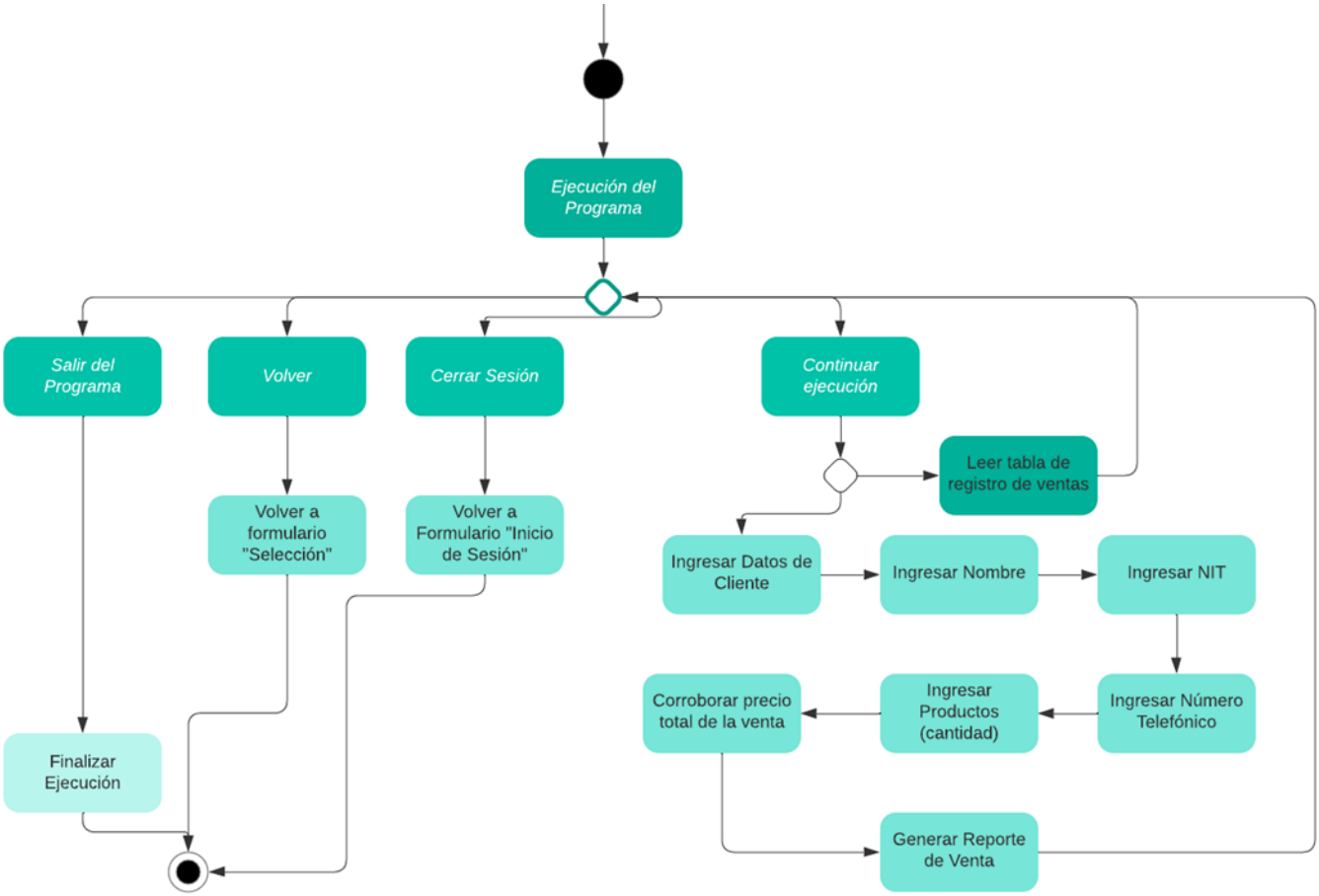


Diagrama de Módulo de Inventarios

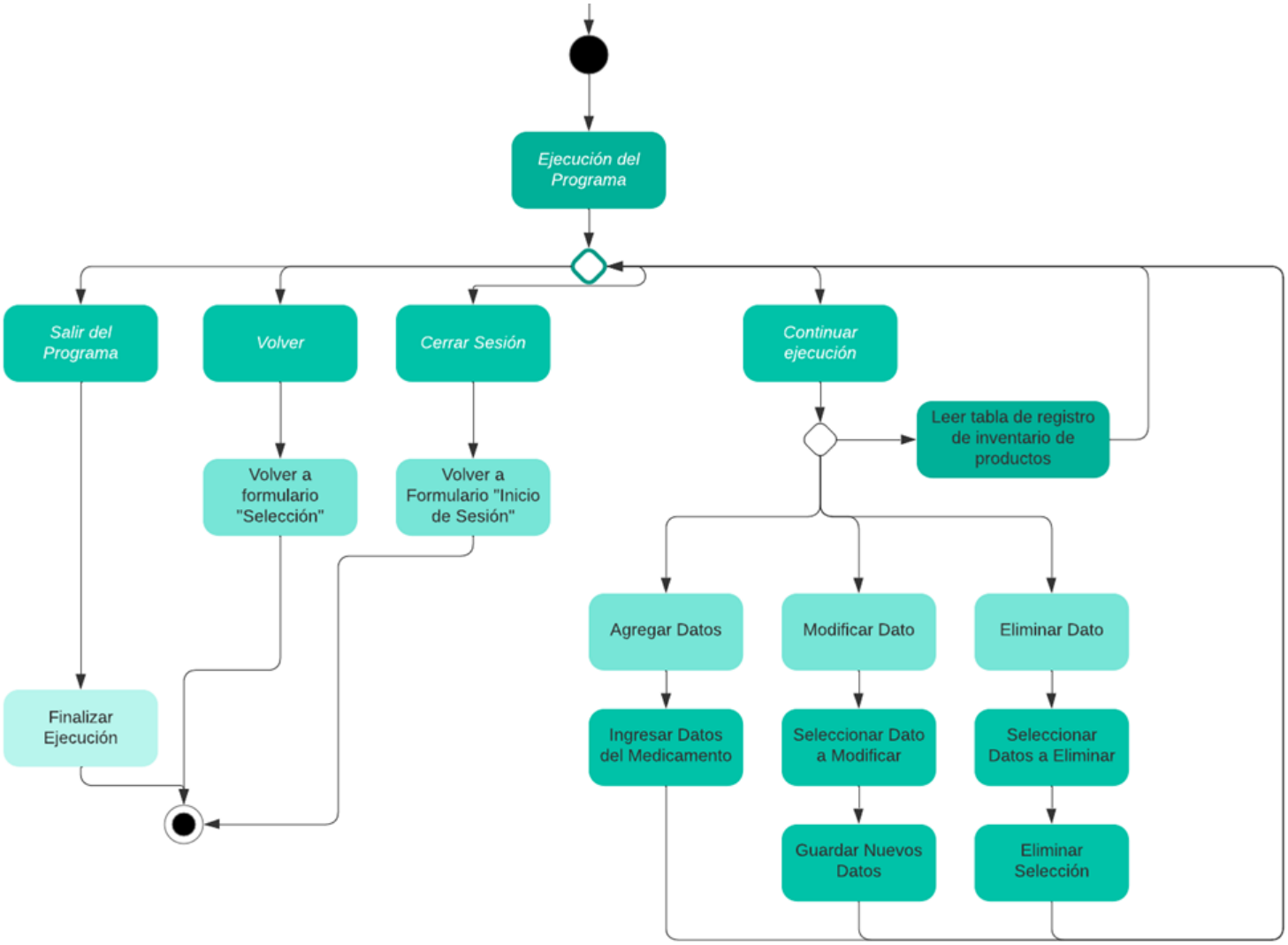


Diagrama de Módulo de Usuarios

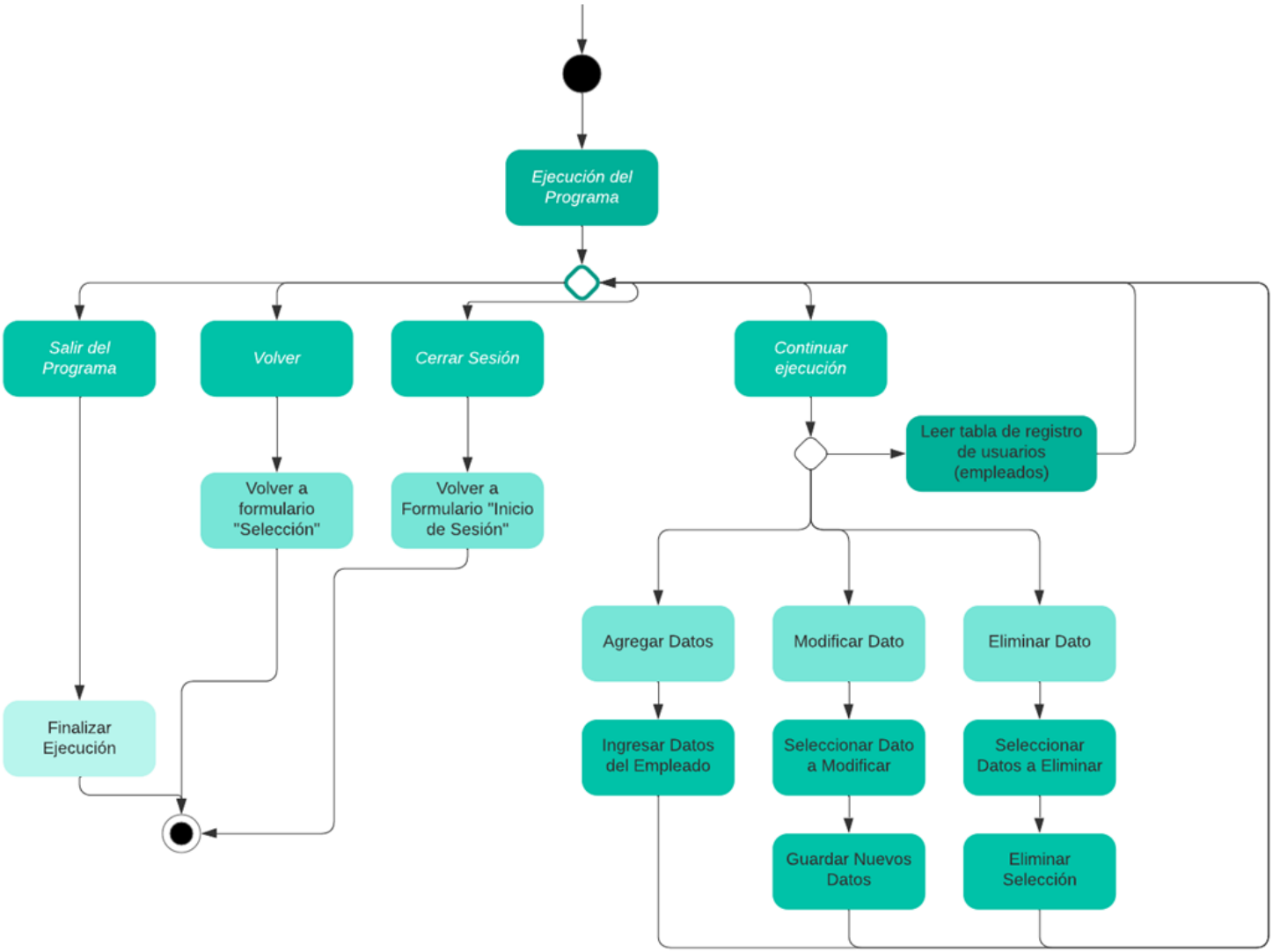
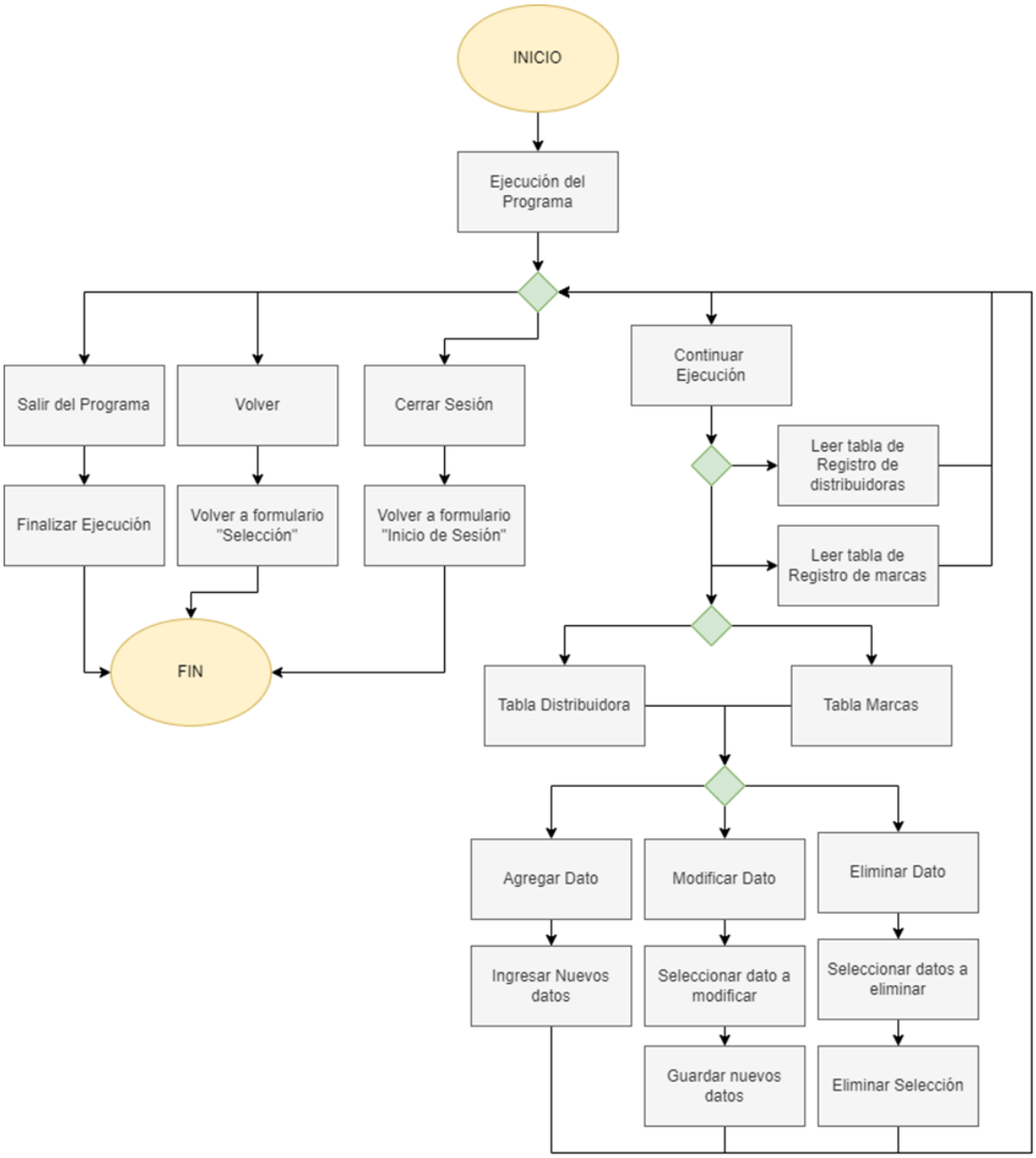


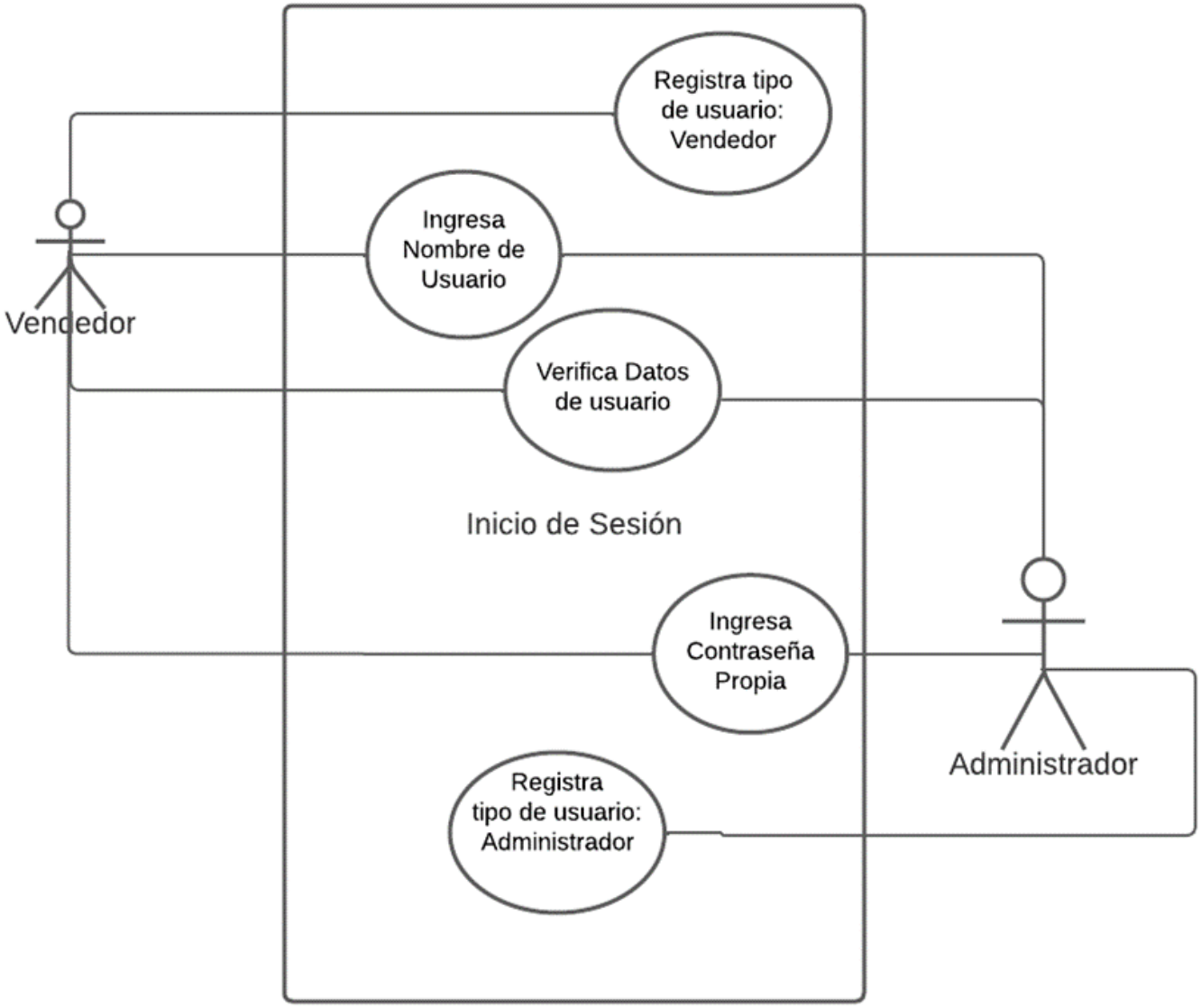
Diagrama de Módulo de Distribuidoras y Marcas



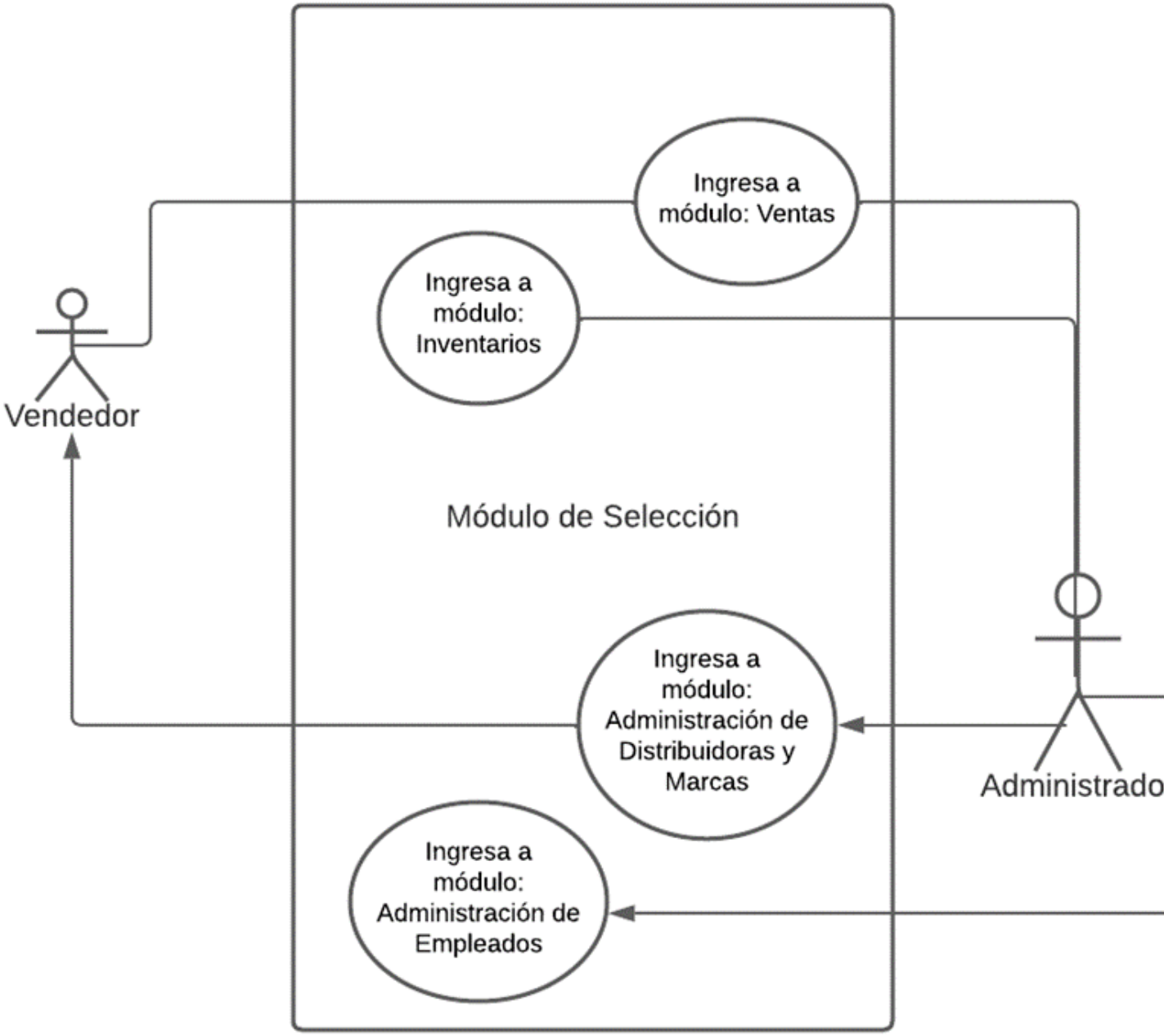
Diagramas de Casos de Uso

Los diagramas de uso, se utilizan para concretar quiénes y en qué momento de necesidad, van a hacer uso de las módulos del programa y de sus clases, para generar mas datos, modificarlos o eliminarlos. Se mostraran los diagramas de casos de uso a continuación:

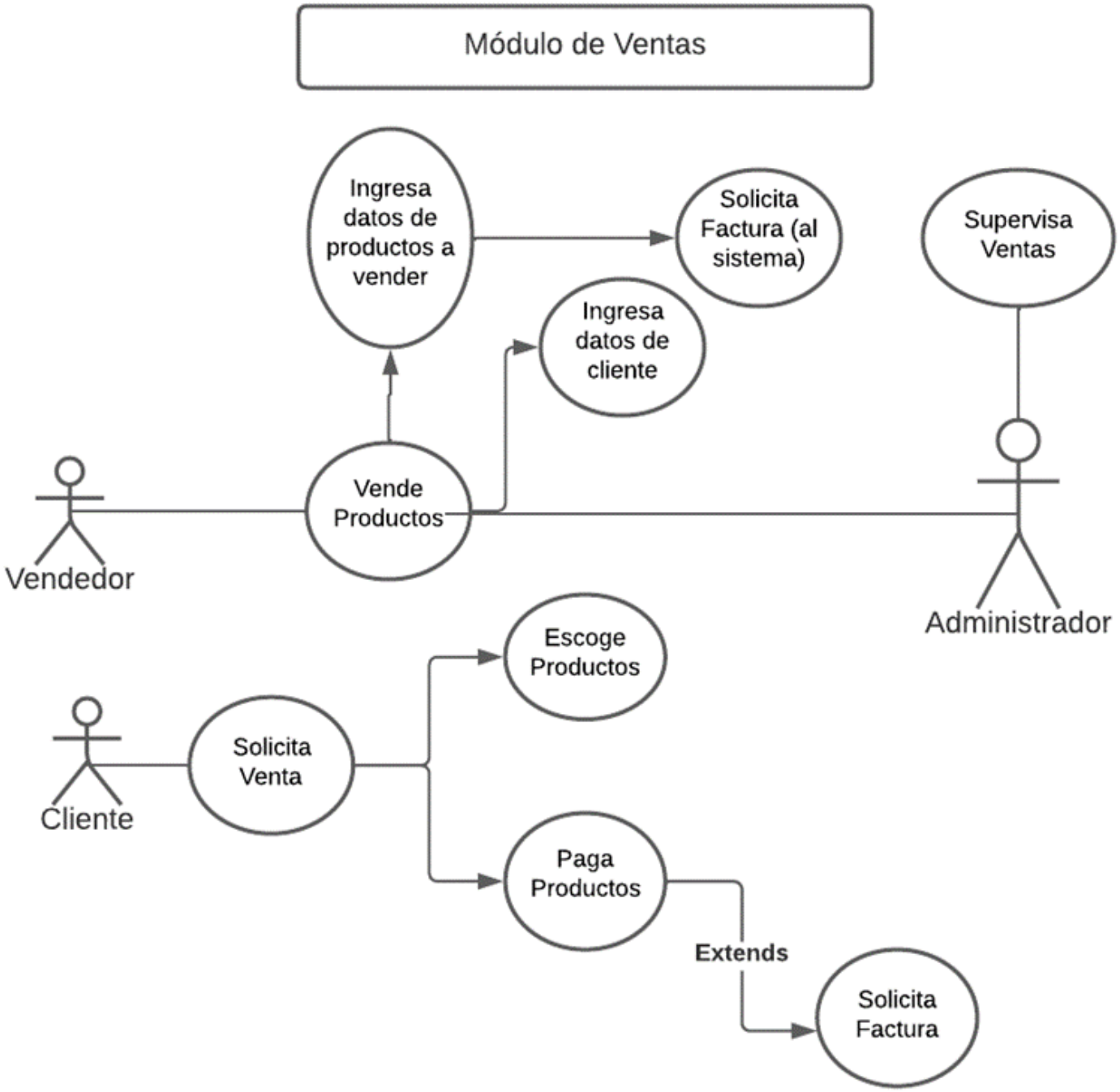
- **Diagrama de Casos de Uso: Inicio de Sesión**



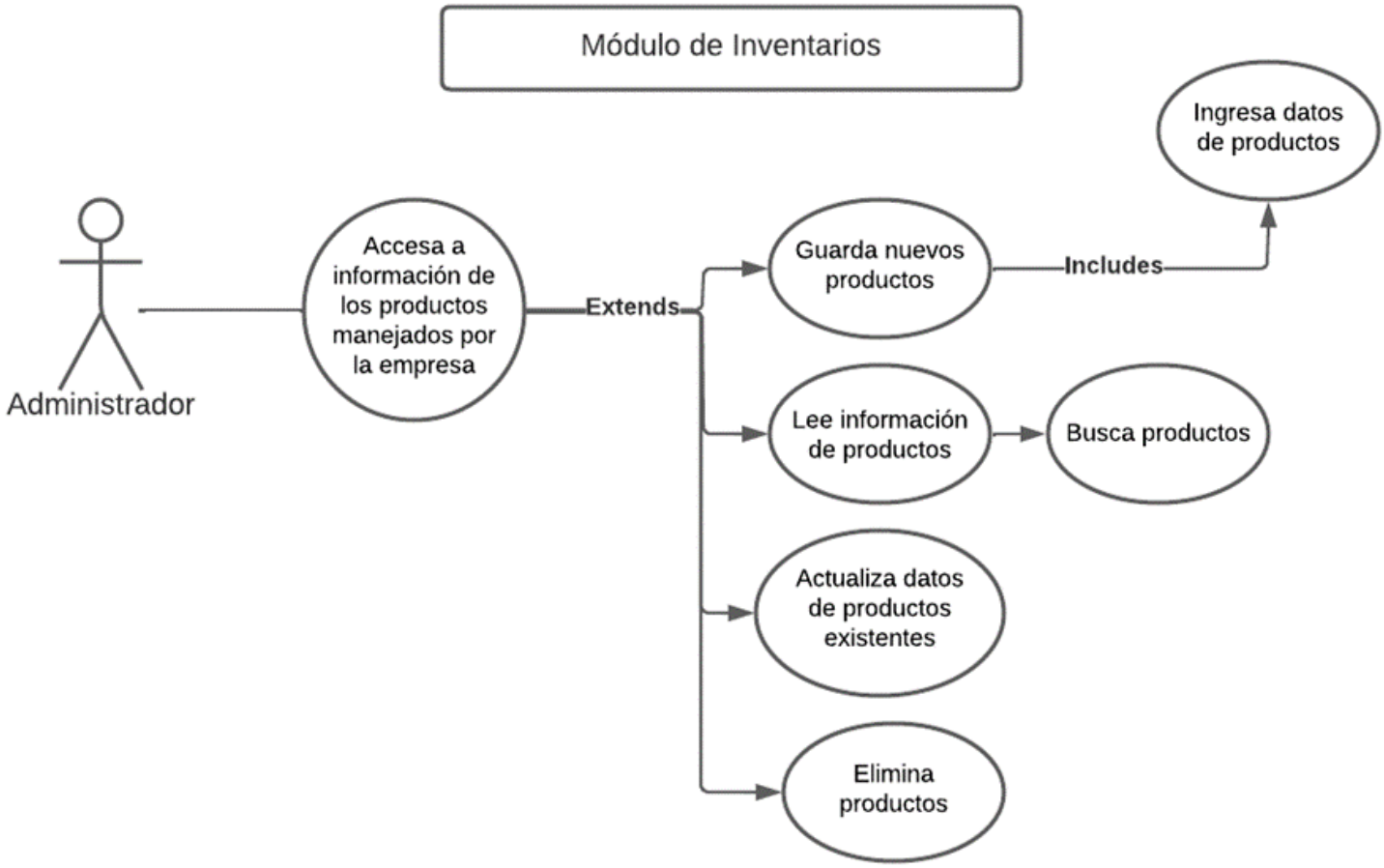
- **Diagrama de Casos de Uso: Selección**



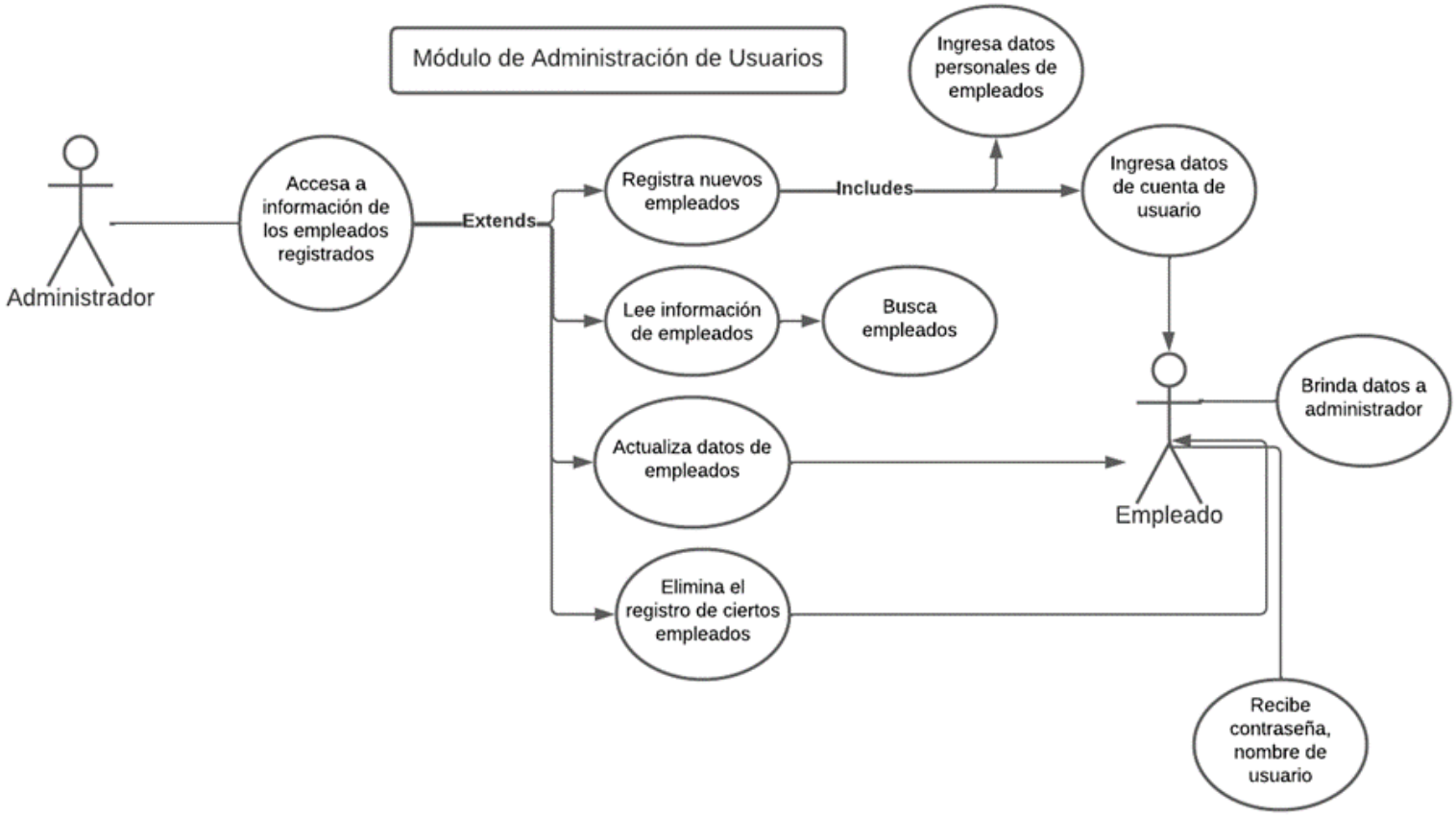
- **Diagrama de Casos de Uso: Ventas**



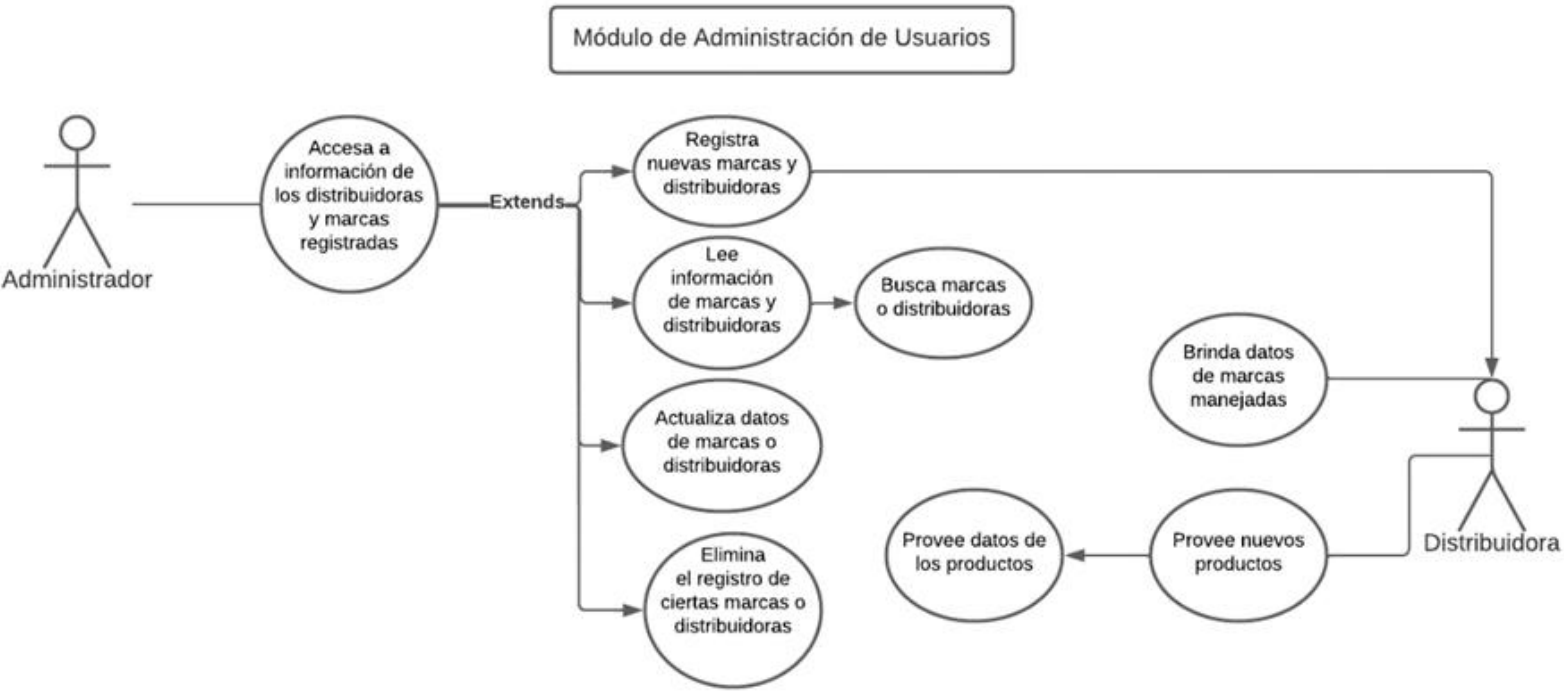
- **Diagrama de Casos de Uso: Inventarios**



- **Diagrama de Casos de Uso: Administración de Usuarios**



- **Diagrama de Casos de Uso: Administración de Distribuidoras y Marcas**



Solución de los Módulos

El software de “Farmacia Bethesda”, funciona con una referencia de conexión integrada “MySql.Data”, el cual permite la conexión de las funciones de los módulos con la base de datos en PhpMyAdmin. Aunado a esto, al conectarse por medio de XAMPP, al servicio de bases de datos, los módulos podrán funcionar sin ningún problema.

Los módulos de administración de datos CRUD (inventarios, marcas, distribuidoras y usuarios), están conformados por un sistema de envío de consultas a MySQL, destinadas a crear registros, leer registros, actualizar registros y eliminar registros concretamente.

A continuación se explicará el funcionamiento de los sistemas de consultas por cada módulo:

1. Módulo de Inicio de Sesión

El módulo de inicio de sesión se conforma en diseño, de dos cajas de texto haciendo referencia al nombre y la contraseña del usuario que desea ingresar, junto con un botón para comprobar los datos y otro para salir del programa. en este módulo se utilizara un código para comprobar si los datos corresponden con los establecido en la base de datos además también se usará un código para que el programa coincidan los usuarios y contraseñas de manera acertada.

El inicio de sesión incluye una herramienta (MD5), para encriptar la contraseña ingresada por el usuario, y comparar la encriptación, con la disponible en la base de datos. Además, se utilizan dos funciones de código por separado, para transmitir a los demás formularios el índice del usuario que ingresa y el tipo, para condicionar la accesibilidad a los módulos dependiendo del tipo de usuario.

Función para comprobar el usuario y la contraseña (con MD5)

```
Private Sub BtLogin_Click(sender As Object, e As EventArgs) Handles BtLogin.Click
    Try
        Dim codigo As New UnicodeEncoding()
        Dim md5 As New MD5CryptoServiceProvider()
        Dim clave As String
        clave = TbContraseña.Text
        Dim Hash() As Byte = md5.ComputeHash(codigo.GetBytes(clave))
        Dim nuevaclave As String
        nuevaclave = Convert.ToBase64String(Hash)
        Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password="
        Dim conn As New MySqlConnection(cadenaConexion)
        conn.Open()
        Dim leer As MySqlDataReader
        Dim codigo2 As MySqlCommand
        codigo2 = New MySqlCommand()
        codigo2.Connection = conn
        codigo2.CommandText = "SELECT * FROM usuarios WHERE username
            = '" & TbUsuario.Text & "' and userpass = '" & nuevaclave & "'"
        leer = codigo2.ExecuteReader()
        If leer.Read() Then
            getusertype()
            getuserid()
            Me.Hide()
            MenuM.Show()
        Else
            MessageBox.Show("Nombre de usuario o contraseña incorrectos",
                            "Acceso Denegado", MessageBoxButtons.OK, MessageBoxIcon.Stop)
        End If
        conn.Close()
    Catch ex As Exception
        MessageBox.Show("Hubo un error de conexión con la base de datos",
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
    TbUsuario.Clear()
    TbContraseña.Clear()
End Sub
```

Función para obtener el tipo de usuario

```
Sub getusertype()
    Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
    Dim conn As New MySqlConnection(cadenaConexion)
    Dim da As MySqlDataAdapter
    Dim dt As DataTable
    conn.Open()
    Dim sQuery = "SELECT usertype FROM usuarios WHERE username = '" & TbUsuario.Text & "';"
    da = New MySqlDataAdapter(sQuery, conn)
    dt = New DataTable
    da.Fill(dt)
    tokenusertype = dt.Rows(0)("usertype").ToString
    conn.Close()
End Sub
```

Función para obtener en índice del usuario

```
Sub getuserid()
    Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
    Dim conn As New MySqlConnection(cadenaConexion)
    Dim da As MySqlDataAdapter
    Dim dt As DataTable
    conn.Open()
    Dim sQuery = "SELECT id_user FROM usuarios WHERE username = '" & TbUsuario.Text & "';"
    da = New MySqlDataAdapter(sQuery, conn)
    dt = New DataTable
    da.Fill(dt)
    iduser = dt.Rows(0)("id_user").ToString
    conn.Close()
End Sub
```

El módulo de selección consiste en los botones de enlace a cada uno de los módulos internos del programa, así como un botón para cerrar sesión y otro para cerrar completamente el programa. Cuando el módulo de selección cargue, se ejecutará un código con el fín de comprobar el tipo de usuario obtenido en el inicio de sesión y en base al resultado, habilitar o deshabilitar botones de acceso a ciertos módulos disponibles condicionalmente para vendedores o administradores.

Funciones para mostrar otros formularios

```
Private Sub BtSalir_Click(sender As Object, e As EventArgs) Handles BtSalir.Click
    End
End Sub

Private Sub BtUsuarios_Click(sender As Object, e As EventArgs) Handles BtUsuarios.Click
    Me.Hide()
    Empleados.Show()
End Sub

Private Sub BtInventario_Click(sender As Object, e As EventArgs) Handles BtInventario.Click
    Me.Hide()
    Medicamentos.Show()
End Sub

Private Sub TbMarcas_Click(sender As Object, e As EventArgs) Handles BtMarcas.Click
    Me.Hide()
    Marcas.Show()
End Sub

Private Sub TbDistrib_Click(sender As Object, e As EventArgs) Handles BtDistrib.Click
    Me.Hide()
    Distribuidoras.Show()
End Sub

Private Sub BtVender_Click(sender As Object, e As EventArgs) Handles BtVender.Click
    Me.Hide()
    Ventas.Show()
End Sub
```

Función para comprobar el tipo de usuario

```
Private Sub MenuM_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    If usertoken = "Vendedor" Then
        BtInventario.Enabled = False
        BtUsuarios.Enabled = False
        BtDistrib.Enabled = False
        BtMarcas.Enabled = False
    End If
End Sub
```

2. Módulos de CRUD (Inventarios, Usuarios, Marcas y Distribuidoras)

Estos módulos consisten en que el usuario pueda administrar y leer los registros de diferentes tablas de la base de datos. Cada uno de estos formularios cuenta con botones y cajas de texto para buscar medicamentos y mostrarlos en un DataGridView donde se muestra cada fila de la tabla. Tambien cuenta con cajas de texto y de selección de datos, donde el usuario podrá ingresar los datos de un nuevo medicamento, editar uno ya existente, o eliminarlos con el botón de "Eliminar".

Los formularios CRUD, cuentan con varias sentencias de conexión con la base de datos y una sentencia para seleccionar datos desde el DataGridView presente.

Sentencia para lectura de datos (SELECT)

```
Sub ReadQuery()
    Try
        Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
        Dim conn As New MySqlConnection(cadenaConexion)
        Dim da As MySqlDataAdapter
        Dim dt As DataTable
        conn.Open()
        Dim sQuery = "SELECT m.`id_medicamento` as 'ID', m.`name_medicamento` as 'Nombre',
             ma.`name_marca` as 'Marca', di.`name_distribuidora` as 'Distribuidora' ,
             m.`fechacad_medicamento` as 'Fecha de Caducidad', m.`obs_medicamento` as 'Observación',
             m.`precio_medicamento` as 'Precio',m.`costo_medicamento` as 'Costo',
             m.`medica_existencias` as 'Existencias' , m.`img_medicamento` as 'Imagen' FROM
             medicamento m INNER JOIN marca ma ON m.id_marca = ma.id_marca INNER JOIN distribuidoras di ON
             m.id_distribuidora = di.id_distribuidora;"
        da = New MySqlDataAdapter(sQuery, conn)
        dt = New DataTable
        da.Fill(dt)
        DTGmedicamentos.DataSource = dt
        conn.Close()
    Catch ex As Exception
        MessageBox.Show("Hubo un error de conexión con la base de datos", "Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

Función para buscar datos desde TextBox

```
Sub designbuscar()

    If cbbuscar.Text = "Nombre" And tbbuscar.Text <> "" Then
        columnabuscar = " WHERE m.name_medicamento LIKE "
        datobuscar = "'" + tbbuscar.Text + "%" + "'"
    ElseIf cbbuscar.Text = "Marca" And tbbuscar.Text <> "" Then
        columnabuscar = " WHERE ma.name_marca LIKE "
        datobuscar = "'" + tbbuscar.Text + "%" + "'"
    ElseIf cbbuscar.Text = "Distribuidora" And tbbuscar.Text <> "" Then
        columnabuscar = " WHERE di.name_distribuidora LIKE "
        datobuscar = "'" + tbbuscar.Text + "%" + "'"
    ElseIf cbbuscar.Text = "" And tbbuscar.Text = "" Then
        columnabuscar = ""
        datobuscar = ""
    ElseIf cbbuscar.Text <> "" And tbbuscar.Text = "" Then
        MessageBox.Show("Debe ingresar un dato a buscar", "Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
    ElseIf cbbuscar.Text = "" And tbbuscar.Text <> "" Then
        MessageBox.Show("Debe seleccionar la categoría del dato a buscar.",
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
    End If
End Sub
```

Sentencia para inserción de datos (INSERT)

```
Sub guardardatos()
    Try
        indexmarca()
        indexdistrib()
        dateformato()
        If String.IsNullOrEmpty(TbNombre.Text) Or String.IsNullOrEmpty(TbObservacion.Text) Or
        String.IsNullOrEmpty(direccionimagen) Then
            MessageBox.Show("No ingresó los datos completos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        Else
            Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
            Dim conn As New MySqlConnection(cadenaConexion)
            Dim da As MySqlDataAdapter
            Dim dt As DataTable
            conn.Open()
            Dim sQuery = "INSERT INTO medicamento (name_medicamento, id_marca,
                fechacad_medicamento, obs_medicamento, id_distribuidora, precio_medicamento,
                costo_medicamento, medica_existencias, img_medicamento) VALUES ('" & TbNombre.Text & "',
                " & idmarcacombobox & ",'" & fechasql & "','" & TbObservacion.Text & "'," & iddistribcombobox & ",
                '" & Convert.ToString(NUDprecio.Value) & "','" & Convert.ToString(NUDcosto.Value) & "',
                '" & Convert.ToString(NUDexistencias.Value) & "','" & direccionfinal & "');"
            da = New MySqlDataAdapter(sQuery, conn)
            dt = New DataTable
            da.Fill(dt)
            conn.Close()
            ReadQuery()
            cleartxt()
        End If
    Catch ex As Exception
        MessageBox.Show("Hubo un error de conexión con la base de datos", "Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

Sentencia para actualización de datos (UPDATE)

```
Private Sub BtEditar_Click(sender As Object, e As EventArgs) Handles BtEditar.Click

    indexmarca()
    indexdistrib()
    dateformato()
    If String.IsNullOrEmpty(TbNombre.Text) Or String.IsNullOrEmpty(TbObservacion.Text) Or
        String.IsNullOrEmpty(direccionimagen) Then
        MessageBox.Show("No ingresó los datos completos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
        Dim conn As New MySqlConnection(cadenaConexion)
        Dim da As MySqlDataAdapter
        Dim dt As DataTable
        conn.Open()
        Dim sQuery = "UPDATE medicamento SET name_medicamento='" & TbNombre.Text & "',
                id_marca=" & idmarcacombobox & ",fechacad_medicamento='" & fechasql & "',
                obs_medicamento='" & TbObservacion.Text & "',id_distribuidora = " & iddistribcombobox & ",
                precio_medicamento='" & Convert.ToString(NUDprecio.Value) & "',
                costo_medicamento='" & Convert.ToString(NUDprecio.Value) & "',
                medica_existencias='" & Convert.ToString(NUDexistencias.Value) & "',
                img_medicamento='" & direccionfinal & "' WHERE id_medicamento=" & stringindex
        da = New MySqlDataAdapter(sQuery, conn)
        dt = New DataTable
        da.Fill(dt)
        conn.Close()
        ReadQuery()
        cleartxt()
    End If

End Sub
```

Sentencia para eliminación de datos (DELETE)

```
Private Sub BtEliminar_Click(sender As Object, e As EventArgs) Handles BtEliminar.Click
    Try
        If String.IsNullOrEmpty(stringindex) Then
            MessageBox.Show("Seleccione una fila de la tabla para eliminarla correctamente",
                            "Error de tabla", MessageBoxButtons.OK, MessageBoxIcon.Error)
        Else
            Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
            Dim conn As New MySqlConnection(cadenaConexion)
            Dim da As MySqlDataAdapter
            Dim dt As DataTable
            conn.Open()
            Dim sQuery = "DELETE FROM medicamento WHERE id_medicamento =" & stringindex
            da = New MySqlDataAdapter(sQuery, conn)
            dt = New DataTable
            da.Fill(dt)
            conn.Close()
            ReadQuery()
            cleartxt()
        End If
    Catch EX As Exception
        MessageBox.Show("Hubo un error de conexión con la base de datos",
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

Sentencia para selección de datos (Eventos de DataGridView "CellClick")

```
Private Sub DTGmedicamentos_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DTGmedicamentos.
    index = e.RowIndex
    Dim selectedrow As DataGridViewRow

    selectedrow = DTGmedicamentos.Rows(index)
    stringindex = selectedrow.Cells(0).Value.ToString
    TbNombre.Text = selectedrow.Cells(1).Value.ToString
    TbObservacion.Text = selectedrow.Cells(5).Value.ToString
    ComboBox1.SelectedItem = selectedrow.Cells(2).Value.ToString
    ComboBox2.SelectedItem = selectedrow.Cells(3).Value.ToString
    NUDprecio.Value = selectedrow.Cells(6).Value
    NUDcosto.Value = selectedrow.Cells(7).Value.ToString
    NUDexistencias.Value = selectedrow.Cells(8).Value.ToString
    direccionfinal = selectedrow.Cells(9).Value.ToString
    direccionimagen = direccionproduc & direccionfinal
    PbMedicamento.Image = Image.FromFile(direccionimagen, True)
    DateTimePicker1.Value = selectedrow.Cells(4).Value.ToString
End Sub
```

Además, con el fín de obtener el índice de una clave foránea seleccionada desde una caja de selección, se ejecuta una función por cada ComboBox de lectura, obteniendo una tabla de datos y convirtiendo la única celda en una cadena de texto:

```
Sub indexmarca()
    Try
        Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
        Dim conn As New MySqlConnection(cadenaConexion)
        Dim da As MySqlDataAdapter
        Dim dt As DataTable
        conn.Open()
        Dim sQuery = "SELECT id_marca FROM marca WHERE name_marca = '" & ComboBox1.Text & "'"
        da = New MySqlDataAdapter(sQuery, conn)
        dt = New DataTable
        da.Fill(dt)
        idmarcacombobox = dt.Rows(0)("id_marca").ToString
        conn.Close()
    Catch ex As Exception
        MessageBox.Show("Hubo un error de conexión con la base de datos",
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

3. Encriptación y comprobación de contraseñas en el módulo “Usuarios”

En el módulo “Usuarios” se encuentran dos cajas de texto destinadas a que el usuario escriba dos veces una nueva contraseña (para un usuario nuevo o a editar), con el fín de comprobar efectivamente que la contraseña se haya escrito de manera correcta. El programa escribe una condición en caso de que las contraseñas sean iguales o desiguales para brindar un resultado:

```
If TbPassword.Text <> TbVerify.Text Then
    MessageBox.Show("La contraseña que ingresó, no coincide con la verificación", "Verificación",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
Else
    encriptacion()
    Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
    Dim conn As New MySqlConnection(cadenaConexion)
    Dim da As MySqlDataAdapter
    Dim dt As DataTable
    conn.Open()

    Dim sQuery = "UPDATE usuarios SET username='" & TbUserName.Text & "',
            userpass='" & nuevaclave & "',usertype='" & tipouser & "',
            nombre_user='" & TbNombre.Text & "',CUI_usuario='" & TbCui.Text & "',
            num_usuario='" & TbNumero.Text & "',puesto_usuario='" & TbPuesto.Text & "',
            correo_usuario='" & TbEmail.Text & "',img_usuario='" & direcciontemporal & "'
            WHERE id_user=" & stringindex
    da = New MySqlDataAdapter(sQuery, conn)
    dt = New DataTable
    da.Fill(dt)
    conn.Close()
    ReadQuery()
    cleartxt()
End If
```

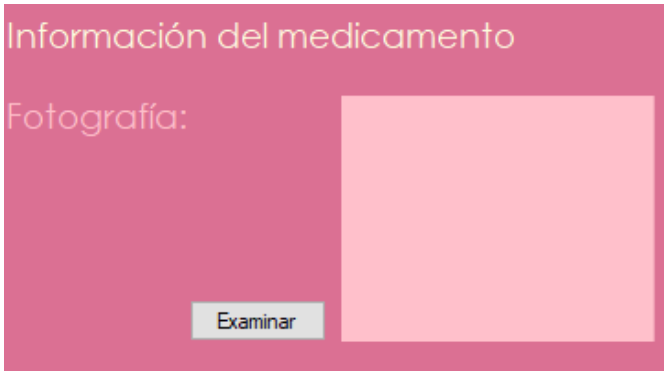
4. Validación de correos electrónicos “Usuarios”

En el módulo usuarios se encuentra una condición, donde se verifican los carácteres del email ingresado por el usuario, para comprobar si es un correo válido:

```
Dim regexPattern As String = "^[\w._%-]+@[\w.-]+\.[a-zA-Z]{2,4}$"
Dim emailRegEx As System.Text.RegularExpressions.Regex = New System.Text.RegularExpressions.Regex(regexPattern)
Dim result As Boolean = emailRegEx.IsMatch(TbEmail.Text)
If Not result Then
    MessageBox.Show("Ingrese un correo válido", "Verificación de Email", MessageBoxButtons.OK,
                    MessageBoxIcon.Exclamation)
End If
```

5. Inserción de imágenes en los módulos CRUD

En cada módulo CRUD, se encuentra un PictureBox y un botón de “Examinar”. Estos elementos, funcionan para permitirle al usuario ingresar una imagen para el campo de la tabla a modificar.



Código para ingresar una imagen y obtener una dirección de imagen válida:

```
Private Sub BtExaminar_Click(sender As Object, e As EventArgs) Handles BtExaminar.Click
    Try
        OpenFileDialog1.ShowDialog()
        direccionimagen = OpenFileDialog1.FileName

        direccionfinal = direccionimagen.Replace("C:\xampp\htdocs\php_bethesda\ImgFarmacia\Productos\", "")
        PbMedicamento.Image = Image.FromFile(direccionimagen, True)

    Catch ex As Exception
        MessageBox.Show("Hubo un error al incorporar la ruta o formato de la imagen al programa",
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

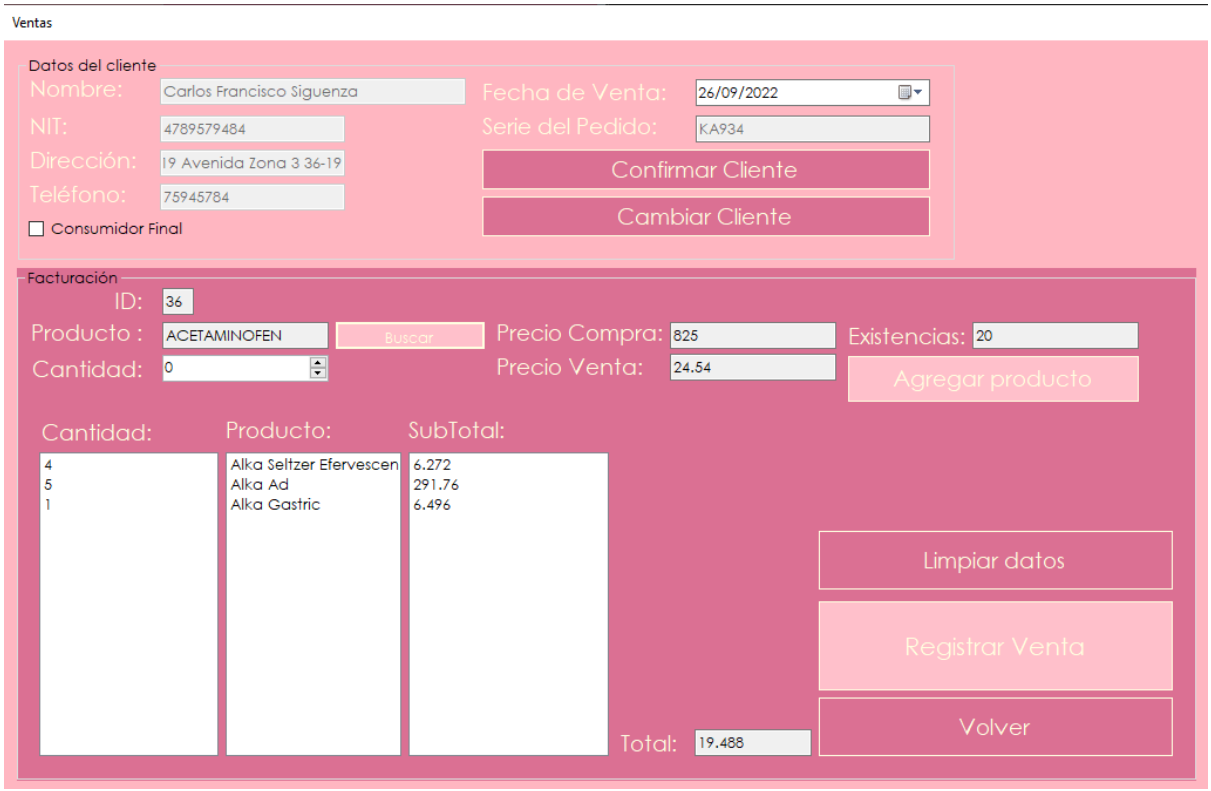
En caso de tener la necesidad de ingresar la dirección completa de la imagen a la base de datos, se debe escribir el siguiente código para confirmar las diagonales de la dirección:

```
Sub cadenanueva()
    Dim nuevacadena() As Char
    nuevacadena = direccionimagen.ToCharArray()
    For i = 0 To nuevacadena.Length - 1 Step 1
        If nuevacadena(i) = "\" Then
            direcciontemporal = direcciontemporal & "\\"
        Else
            direcciontemporal = direcciontemporal & nuevacadena(i)
        End If
    Next
End Sub
```

6. Módulo de Ventas

El módulo de ventas permite la entrada de datos del cliente, así como los productos seleccionados por el mismo para la compra. El módulo de ventas cuenta con una opción para omitir el uso de un NIT y datos de un cliente, habilitando una caja de selección haciendo referencia a un cliente anónimo (consumidor final). El mismo cuenta con una opción para elegir una fecha donde se haya realizado una venta, así como una opción para escribir una serie única identificando la venta.

En este apartado, la entrada para el NIT, y el Número de teléfono, poseen un límite de dígitos, así como poseer como única opción la inserción de números en los eventos KeyPress.

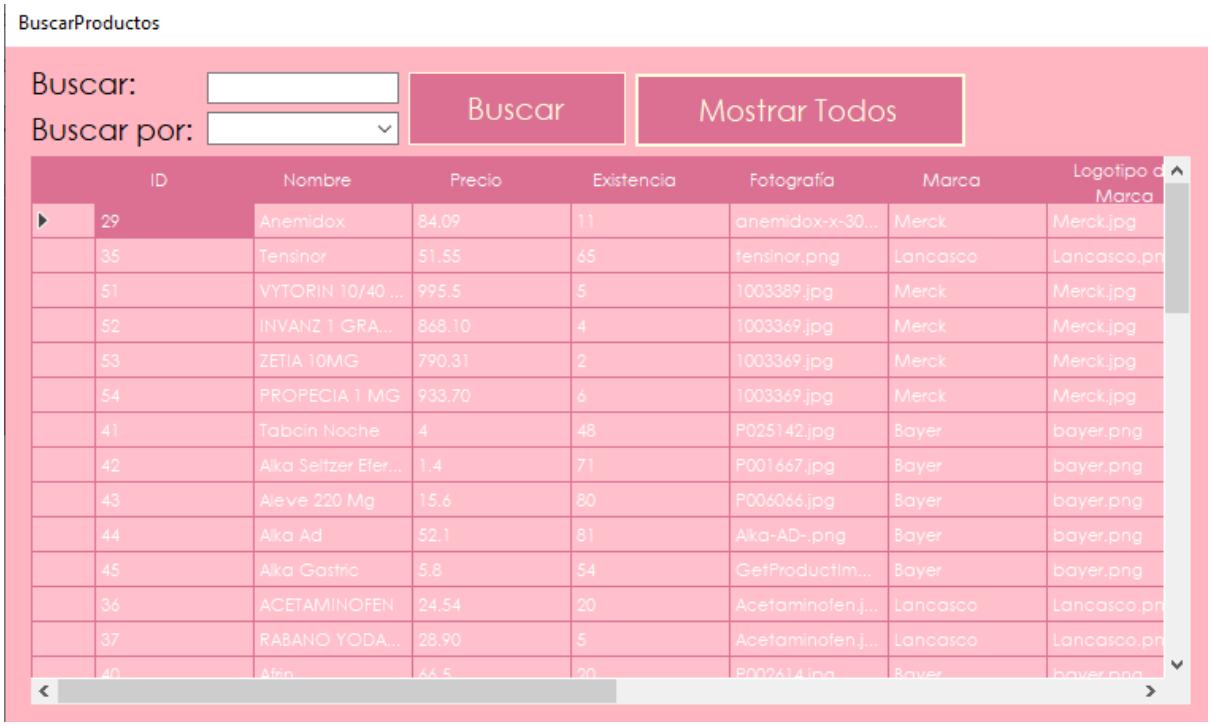


En el módulo de ventas, es posible buscar medicamentos para vender al cliente por medio de un formulario externo que escribe los datos de la fila seleccionada (medicamento), en variables distintas que se auto incluyen en las cajas de texto específicas para cada dato. Al tener los datos listos en las cajas de texto, se podrán insertar medicamentos en la lista y está recorrerá cada dato para obtener el total y un vector con el índice de cada producto seleccionado.

Función de Lectura de datos de Medicamento

```
Public Sub setarticulo(par0, par1, par2, par3, par4)
    tbid.Text = par0
    TbProd.Text = par1
    TbCompra.Text = par2
    TbVenta.Text = par3
    TbExistencias.Text = par4
End Sub
```

Formulario de Selección de Medicamentos



Código de selección de medicamentos

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
    Dim conn As New MySqlConnection(cadenaConexion)
    Dim da As MySqlDataAdapter
    Dim dt As DataTable
    conn.Open()
    Dim sQuery = "SELECT m.id_medicamento as 'ID', m.name_medicamento as 'Nombre',m.precio_medicamento as 'Precio',m.medica_existencias as 'Existencia',
    da = New MySqlDataAdapter(sQuery, conn)
    dt = New DataTable
    da.Fill(dt)
    DataGridView1.DataSource = dt
    conn.Close()
End Sub

Private Sub DataGridView1_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick
    Index = e.RowIndex
    Dim selectedrow As DataGridViewRow
    selectedrow = DataGridView1.Rows(Index)
    idmedicamento = selectedrow.Cells(0).Value.ToString
    varname = selectedrow.Cells(1).Value.ToString
    varcompra = selectedrow.Cells(12).Value.ToString
    varventa = selectedrow.Cells(2).Value.ToString
    varexistencias = selectedrow.Cells(3).Value.ToString
    Ventas.setarticulo(idmedicamento, varname, varcompra, varventa, varexistencias)
    Me.Hide()
End Sub
```

Al registrar la venta, el vector que posee los índices de cada producto en la lista, así como el vector que posee la cantidad de dichos productos, son recorridos por un ciclo FOR, incluyéndose en una consulta para ser insertados en la tabla “detalle_pedidos” junto con el índice de la venta del momento.

Además, al evaluar la lista de productos de una venta a registrar, el programa ejecutará una consulta UPDATE para la tabla de medicamentos, con el fin de restar las existencias de los productos registrados, con la cantidad de los vendidos.

Inserción de datos en “detalle_pedidos”

```
Sub insertardetalleventa()
    getidpedido()
    Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
    Dim conn As New MySqlConnection(cadenaConexion)
    Dim da As MySqlDataAdapter
    Dim dt As DataTable
    conn.Open()
    Dim sQuery As String = ""
    For i = 0 To Productos.Items.Count - 1 Step 1
        sQuery = sQuery + "INSERT INTO detalle_pedidos (id_pedido, id_medicamento,
        medicamento_cantidad) VALUES (" & idpedido & "," & vectorid.Items(i) & ",
        " & Convert.ToString(Cantidad.Items(i)) & ");" + vbCrLf +
        "UPDATE medicamento SET medica_existencias= " & restavector.Items(i) &
        " WHERE id_medicamento = " & vectorid.Items(i) & ";"
    Next
    da = New MySqlDataAdapter(sQuery, conn)
    dt = New DataTable
    da.Fill(dt)
    conn.Close()
End Sub
```

Inserción de datos en "pedidos"

```
Sub insertarventa()
    dateformato()
    Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
    Dim conn As New MySqlConnection(cadenaConexion)
    Dim da As MySqlDataAdapter
    Dim dt As DataTable
    conn.Open()
    Dim sQuery = "INSERT INTO pedidos (name_cliente, nit_cliente,
      num_cliente, dir_cliente, fecha_pedido, id_user, serie_pedido,
      total_pedido) VALUES ('" & namecliente & "','" & nitcliente & "',
      '" & telefonocliente & "','" & dircliente & "','" & fechasql & "',
      " & iduser & ",'" & seriepedido & "','" & TbTotal.Text & "');"
    da = New MySqlDataAdapter(sQuery, conn)
    dt = New DataTable
    da.Fill(dt)
    conn.Close()
End Sub
```

7. Módulo de Vista de Ventas

El módulo de vista de ventas, permitirá al usuario facturar, buscar por fecha o por serie, o eliminar ventas realizadas según se de la necesidad.

Formulario de Vista de Ventas

Para hacer una operación con una venta determinada, el usuario deberá seleccionar una de las filas de la tabla que se muestra. Automáticamente, un grupo de variables públicas, obtendrán los datos de cada celda de la fila seleccionada.
Se ejecutará una consulta DELETE, para eliminar una fila con el índice seleccionado en la base de datos y para facturar la venta seleccionada, un nuevo formulario de impresión, se abrirá con diferentes etiquetas de texto, que contendrán el valor de las variables previamente determinadas.

Botón de "Eliminar"

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Try
        If String.IsNullOrEmpty(numero) Then
            MessageBox.Show("Seleccione una fila de la tabla para eliminarla correctamente",
                            "Error de tabla", MessageBoxButtons.OK, MessageBoxIcon.Error)
        Else
            Dim cadenaConexion = "server=localhost;database=registrofarmacia;userid=root;password=;port=3306"
            Dim conn As New MySqlConnection(cadenaConexion)
            Dim da As MySqlDataAdapter
            Dim dt As DataTable
            conn.Open()
            Dim sQuery = "DELETE FROM pedidos WHERE id_pedido =" & numero & ";" &
                "DELETE FROM detalle_pedidos WHERE id_pedido =" & numero & ";"
            da = New MySqlDataAdapter(sQuery, conn)
            dt = New DataTable
            da.Fill(dt)
            conn.Close()
            ReadQuery()
        End If
    Catch EX As Exception
        MessageBox.Show("Hubo un error de conexión con la base de datos", "Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

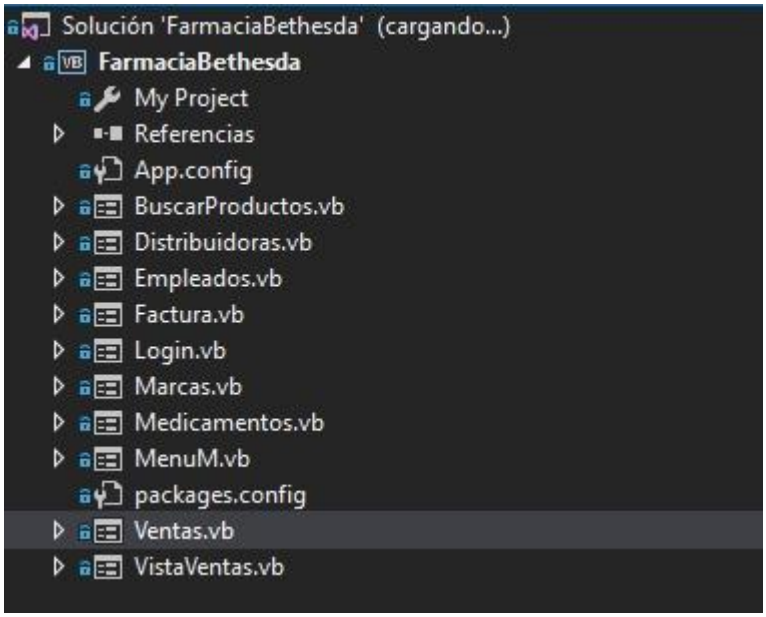
Botón de "Mostrar Factura"

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    If String.IsNullOrEmpty(numero) Then
        MessageBox.Show("Debe seleccionar una fila de la tabla para mostrar su factura",
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
    Else
        Factura.Show()
    End If
End Sub
```

Formulario de Impresión de Factura

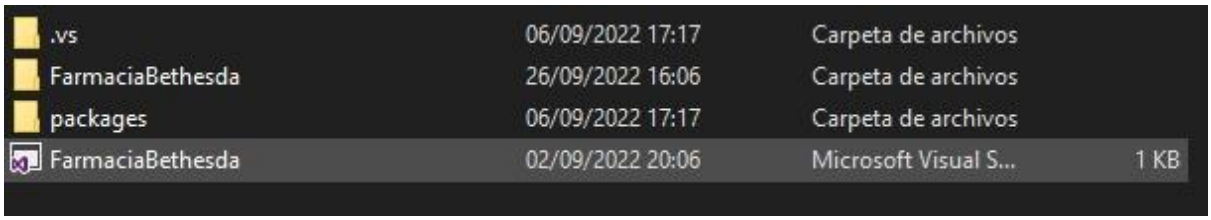
Estructura raíz programa

El proyecto en Visual Studio tiene la siguiente estructura de módulos:



Directorio del programa

En esta carpeta se encuentran todos los módulos del programa incluyendo el respectivo código que cada una requiere para funcionar además de los medios utilizados en el programa y también de el ejecutable necesario para abrir el programa en el visual studio y poder ser utilizado.



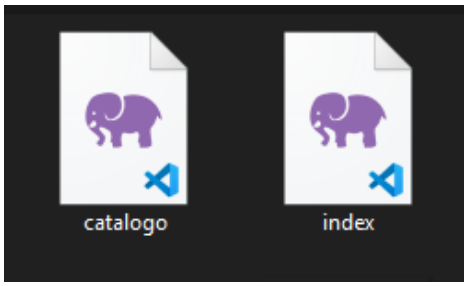
Dentro de la carpeta “FarmaciaBetehesda” se encuentran los archivos de código del programa

bin	06/09/2022 17:59	Carpeta de archivos	
My Project	06/09/2022 17:17	Carpeta de archivos	
obj	06/09/2022 17:17	Carpeta de archivos	
App	02/09/2022 20:06	XML Configuratio...	1 KB
BuscarCliente	06/09/2022 17:59	.NET Managed Re...	6 KB
BuscarProductos.Designer	23/09/2022 00:15	Archivo de origen ...	10 KB
BuscarProductos	23/09/2022 00:15	.NET Managed Re...	6 KB
BuscarProductos	23/09/2022 00:15	Archivo de origen ...	6 KB
Distribuidoras.Designer	09/09/2022 15:39	Archivo de origen ...	21 KB
Distribuidoras	09/09/2022 15:39	.NET Managed Re...	403 KB
Distribuidoras	09/09/2022 15:55	Archivo de origen ...	11 KB
Empleados.Designer	23/09/2022 19:51	Archivo de origen ...	30 KB
Empleados	23/09/2022 19:51	.NET Managed Re...	404 KB
Empleados	23/09/2022 19:51	Archivo de origen ...	14 KB
Factura.Designer	09/09/2022 13:21	Archivo de origen ...	22 KB
Factura	09/09/2022 13:21	.NET Managed Re...	302 KB
Factura	09/09/2022 13:21	Archivo de origen ...	2 KB
FarmaciaBethesda	09/09/2022 11:05	Visual Basic Projec...	9 KB
Form1.Designer	02/09/2022 20:06	Archivo de origen ...	2 KB
Form1	02/09/2022 20:06	.NET Managed Re...	6 KB
Form1	02/09/2022 20:06	Archivo de origen ...	1 KB
Login.Designer	23/09/2022 20:12	Archivo de origen ...	8 KB
Login	23/09/2022 20:12	.NET Managed Re...	698 KB
Login	23/09/2022 20:12	Archivo de origen ...	4 KB
Marcas.Designer	09/09/2022 15:01	Archivo de origen ...	23 KB
Marcas	09/09/2022 15:01	.NET Managed Re...	404 KB
Marcas	22/09/2022 16:53	Archivo de origen ...	10 KB
Medicamentos.Designer	09/09/2022 15:01	Archivo de origen ...	29 KB
Medicamentos	09/09/2022 15:01	.NET Managed Re...	404 KB
Medicamentos	23/09/2022 18:17	Archivo de origen ...	15 KB
MenuM.Designer	09/09/2022 16:05	Archivo de origen ...	11 KB
MenuM	09/09/2022 16:05	.NET Managed Re...	698 KB
MenuM	26/09/2022 15:52	Archivo de origen ...	2 KB
packages	02/09/2022 20:06	XML Configuratio...	1 KB
Ventas.Designer	26/09/2022 16:06	Archivo de origen ...	28 KB
Ventas	26/09/2022 16:06	.NET Managed Re...	403 KB
Ventas	26/09/2022 16:06	Archivo de origen ...	11 KB
VistaVentas.Designer	09/09/2022 16:05	Archivo de origen ...	13 KB
VistaVentas	09/09/2022 16:05	.NET Managed Re...	6 KB
VistaVentas	21/09/2022 15:23	Archivo de origen ...	8 KB

Estructura raíz Pagina

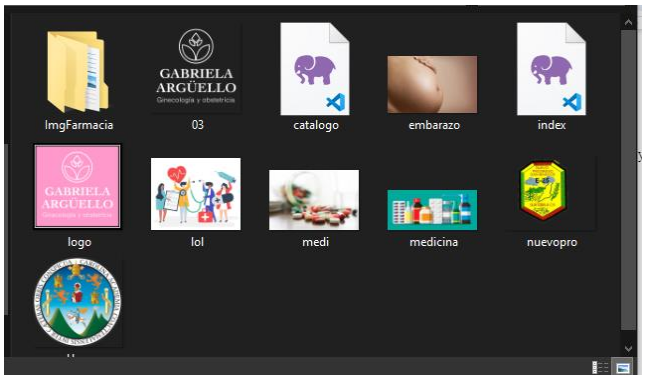
En el proyecto de visual code estas son los documentos de texto utilizados donde se encuentra el código de la página web y se tienen las conexiones con la base de datos.

- Index: Donde se encuentra la conexión principal de la página y el diseño de la primera parte de la página donde se muestra una mayor cantidad de texto, imágenes e información de la farmacia.
- Catálogo:En esta se encuentra el código utilizado para mostrar diversos elementos de la base de datos en la página web.

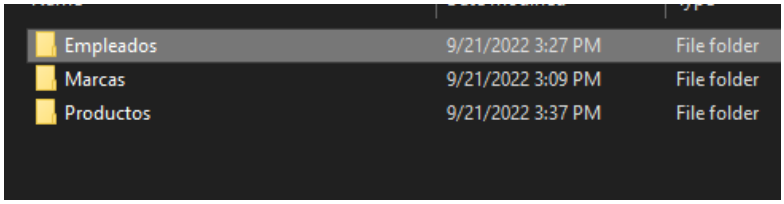


Directorio de la Página

En esta carpeta se encuentran almacenados los documentos de texto donde se encuentran los códigos de la página y las imágenes necesarias para la presentación de la página web incluyendo la carpeta “ImgFarmacia”.



Dentro de la carpeta “ImgFarmacia” se encuentran las imágenes de la base de datos que serán mostradas en la página web en este caso la imagen de los productos, distribuidoras y marcas. Es estrictamente necesario que las imágenes se encuentren dentro de esta carpeta en su respectiva categoría.



Empleados	9/21/2022 3:27 PM	File folder
Marcas	9/21/2022 3:09 PM	File folder
Productos	9/21/2022 3:37 PM	File folder

Funcionamiento de la página web

Para desarrollar la página web para Farmacias Bethesda fue necesario de 2 archivos de texto .php con funciones distintas cada uno pero simples en cuestión de programación, en estos 2 archivos se mezcló el uso del PHPMYADMIN, CSS y Bootstrap además del php y el html de base para que de esta manera la página funcionara correctamente además de mostrar un diseño minimalista pero visualmente atractivo.

Funcionamiento

```
F: > pablo > php_bethesda > catalogo.php
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Document</title>
8       <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-0evHe/X+R7YkIZD
9   <!-- JavaScript Bundle with Popper -->
10  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUs
11
12  </head>
13  <div class="container"  >¡Todo lo necesario para una buena salud!</div>
14  <body>
15  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
16  <a class="navbar-brand" href="index.php">
17      <img src="logo.png" width="30" height="30" class="d-inline-block align-top" alt="">
18      Farmacia Bethesda
19    </a>
20
21      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText" aria-controls="navbarText" aria-expanded="false
22          <span class="navbar-toggler-icon"></span>
23      </button>
24      <div class="collapse navbar-collapse" id="navbarText">
25          <ul class="navbar-nav mr-auto">
26          <li class="nav-item active">
27              <a class="nav-link" href="index.php">Inicio<span class="sr-only"></span></a>
28          </li>
29          <li class="nav-item">
30              <a class="nav-link" href="catalogo.php">Catálogo de Medicamentos</a>
31          </li>
32          </ul>
```

En la primera imagen se puede apreciar la primera parte del código donde se establecen cosas como el uso del bootstrap para la mejora inicial del programa, así designado títulos, subtítulos textos entre otros.

```
<font size="5">
<?php
$user = "root";
$pass = "";
$host = "localhost";
$coneccion = mysqli_connect($host, $user, $pass);
if(!$coneccion)
        {
            echo "No se ha podido conectar con el servidor" . mysql_error();
        }

$DB = "registrofarmacia";
$dbselect = mysqli_select_db($coneccion,$DB);
if (!$dbselect)
{
echo "No se ha podido encontrar la base de Datos";
}

$consulta = "SELECT m.id_medicamento as 'ID', m.name_medicamento as 'Nombre',m.precio_medicamento as 'Precio',m.medica_existencias a
$result = mysqli_query($coneccion,$consulta);

if(!$result)
{
    echo "No se ha podido realizar la consulta";
};
echo '<table class="table table-dark table-striped-columns">';
echo "<tr>";
echo '<th scope="col"><h1>ID</th></h1>';
echo '<th scope="col"><h1>Medicamento</th></h1>';
echo '<th scope="col"><h1>Precio</th></h1>';
echo '<th scope="col"><h1>Observación</th></h1>
```

en esta parte del código se puede ver cómo se realizan las conexiones entre el .php y el PHPMYADMIN con la base de datos utilizada para la farmacia, se hace el select y se le asignan códigos a los miembros de la base de datos ademas de tambien usar el codigo de error de conexión para comprobar si la pagina se conecto de forma correcta.

```
echo '<table class="table table-dark table-striped-columns">';
echo "<tr>";
echo '<th scope="col"><h1>ID</th></h1>';
echo '<th scope="col"><h1>Medicamento</th></h1>';
echo '<th scope="col"><h1>Precio</th></h1>';
echo '<th scope="col"><h1>Observación</th></h1>';
echo '<th scope="col"><h1>Existencia</th></h1>';
echo '<th scope="col"><h1>Fotografía</th></h1>';
echo '<th scope="col"><h1>Marca</th></h1>';
echo '<th scope="col"><h1>Distribuidora</th></h1>';
echo '<th scope="col"><h1>Costo</th></h1>';
echo "</tr>";
while ($colum = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td><h2>" . $colum['ID']. "</td></h2>";
    echo "<td><h2>" . $colum['Nombre'] . "</td></h2>";
    echo "<td><h2>" . $colum['Precio'] . "</td></h2>";
    echo "<td><h2>" . $colum['Observación'] . "</td></h2>";
    echo "<td><h2>" . $colum['Existencia'] . "</td></h2>";
    echo "<td><h2><img src= ImgFarmacia/Productos/" . $colum['Fotografía'] . " height=190></td></h2>";
    echo "<td><h2><img src= ImgFarmacia/Marcas/" . $colum['Logotipo de Marca'] . " height=190></td></h2>";
    echo "<td><h2>" . $colum['Distribuidora'] . "</td></h2>";
    echo "<td><h2>" . $colum['Costo'] . "</td></h2>";
    echo "</tr>";
}
echo "</table>"; ?>

<style>
html, body{
  margin: 0;
  padding: 0;
```

En esta parte del código se crea una tabla donde en texto se muestran los datos que corresponden con lo que queremos mostrar en la página, en la parte inferior se muestra el ingreso de los datos de la base de datos y se especifican en qué lugar van de esta manera se especifica en qué lugar se mostrará cada base de datos y en qué columna y fila dependiendo del lugar donde se especificó .

```
body{
    width: 100%;
    height: 100%;
    font-family: sans-serif;
    letter-spacing: 0.03em;
    line-height: 1.6;

}
.title{
    text-align: center;
    font-size: 40px;
    color: #6a6a6a;
    margin-top: 100px;
    font-family: 'Roboto', sans-serif;

}

.container{
    width: auto;
    max-width: auto;
    height: auto;
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    overflow: hidden;
}
.container .card{
    width: 275px;
    height: auto;
    border-radius: 8px;
    box-shadow: 0 2px 2px rgba(0, 0, 0, 0.2);
    overflow: hidden;
```

En esta parte se muestra una parte del código escrito en css este código se encuentra escrito entre <style></style> puesto a que de otra manera no se aplicaría en el código, este código fue necesario para lograr dale un estilo más fluido y minimalista a la página puesto a que el estilo base que brindaba la plantilla de bootstrap era básico y no permite animaciones fluidas.

Página web en funcionamiento:

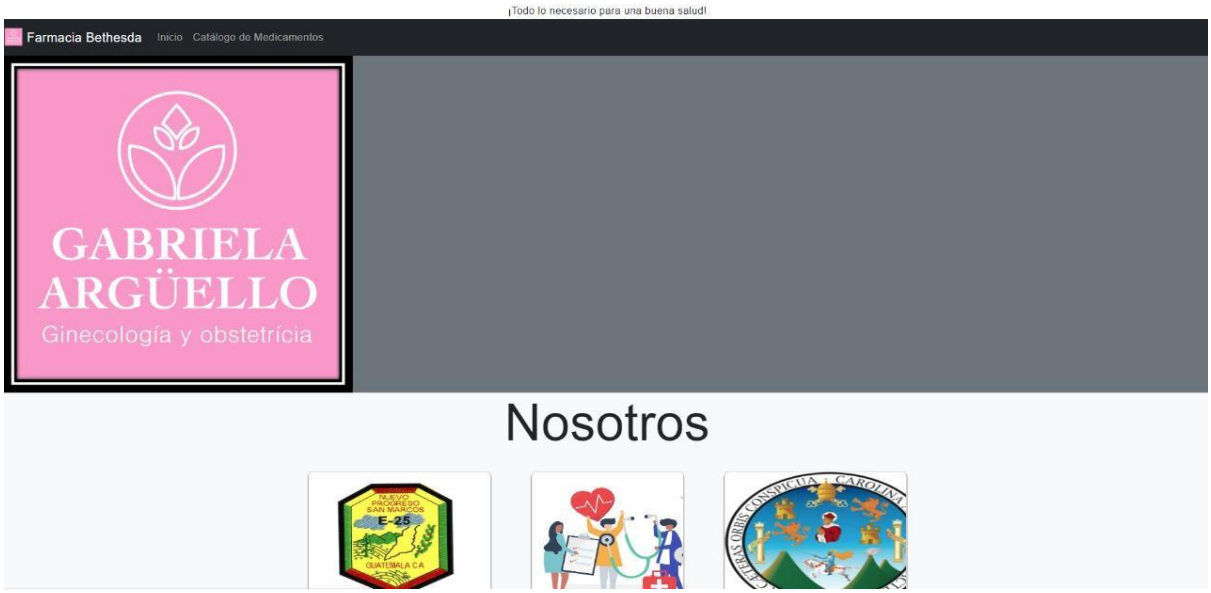


Tabla donde se muestran los elementos de la base de datos dentro de la página web:

¡Todo lo necesario para una buena salud!

Farmacia Bethesda Inicio Catálogo de Medicamentos

ID	Medicamento	Precio	Observación	Existencia	Fotografía	Marca	Distribuidora	Costo
29	Anemidox	Q84.09	Tratamiento preventivo de la anemia por déficit de Hierro, Ácido Fólico y Vitamina B12	11		MERCK	PharMar	Q115.2
35	Tensinor	Q51.55	Valsartan + Hidroclorotiazida	65		LANCASCO	PharMar	Q451
51	VYTORIN 10/40 MG	Q995.5	28 COMPRIMIDOS - CAJA	5		MERCK	PharMar	Q1231.5
52	INVANZ 1 GRAMO IV/IM	Q868.10	POLVO - UNIDAD Requiere receta médica	4		MERCK	PharMar	Q8256.2