

MANUAL **TÉCNICO**

NADRO

México, Ciudad de México a 04 de junio del 2018

Primera Revisión

Este documento fue elaborado por la unidad operativa de sistemas, todos los nombres que se mencionan a continuación quedan reservados únicamente bajo el Copyright de NADRO.

Clave de la forma: NFS001

Getzemany Lesly Aviña León

Mario Axel Gutiérrez Reyes

Alan Jiménez Fernández Pidal

Xilonen Guadalupe Sánchez Xavier

Abigail Valaguez González

ÍNDICE

* Prólogo o Introducción
* Objetivo del manual
* Procedimientos

PRÓLOGO

La Farmacéutica NADRO, es una organización la cual se encarga de la venta de productos de higiene personal/belleza y medicamentos principalmente, por lo cual es de suma importancia contar con un sistema con el cuál se puedan llevar a cabos las actividades necesarias para el correcto funcionamiento de cada una de las sucursales con las que cuentas esta cadena de farmacias.

Es por eso, que el sistema que se desea implementar tiene mucha importancia, ya que como sabemos, el buen funcionamiento del mismo, define la eficiencia de la realización de las tareas que cada integrante de la organización tiene establecidas. Este sistema está hecho con ese fin, el mejorar la productividad y el rendimiento de la sucursal, es fundamental evaluar las técnicas actuales y la tecnología disponible para poder implementar este sistema, por lo cual estamos seguros que la implementación del sistema será un éxito.

Es importante también, saber cómo es que nuestro sistema trabaja y el contenido que este tiene, desde un punto técnico y administrativo, ya que, al ser implementado, se debe saber su funcionamiento desde la parte técnica.

Nuestro sistema proyecta funciones de compra-venta, inventario, búsqueda de productos, entre otras, las cuáles, claramente son fundamentales y propias de una farmacéutica, dentro de este se cubren necesidades que actualmente no han sido cubiertas y son necesarias para un mejor desarrollo.

Contamos con las funciones básicas que debe cubrir la relación sistema-base de datos y se tomó como principio de diseño una interfaz gráfica sencilla de manejar por el administrador, y un diseño de código que se guía de las buenas prácticas de programación.

Al usar este sistema, se puede tener total confianza y seguridad de que el software desarrollado es sólido, confiable, accesible y lo más importante, fácil de entender.

El alcance del sistema es ambicioso y estamos seguros que habrá una enorme mejora en la experiencia tanto de compra-venta, como en la de control de información, movimientos y personal dentro de la plataforma.

OBJETIVO DEL MANUAL

Al tratarse de un manual técnico, absolutamente todos los esfuerzos y procedimientos que se describen aquí son de carácter profesional y sistemático, ya que se describen cuidadosamente las entrañas del proyecto y las condiciones bajo las cuáles se opera.

Aquí se describen las acciones que el personal encargado de esta área técnica deberá de seguir para la modificación, implementación o adaptación de ciertos caracteres dentro del sistema.

La implementación de sistema conlleva los siguientes objetivos:

* Reducir el tiempo de compra/venta en cajas.
* Obtener un control efectivo y directo de todas las actividades, procesos e información que realiza la organización desde su interior (por sucursal).
* Integración de nuevas tecnologías y herramientas para el desarrollo de vanguardia.
* Alcanzar la máxima eficiencia posible en las actividades y procesos que se realizan dentro de la organización.
* Disminución de errores en las bases de datos que maneja la organización (productos, ventas, proveedores, inventarios y sucursales).

Algunos objetivos específicos del manual técnico son:

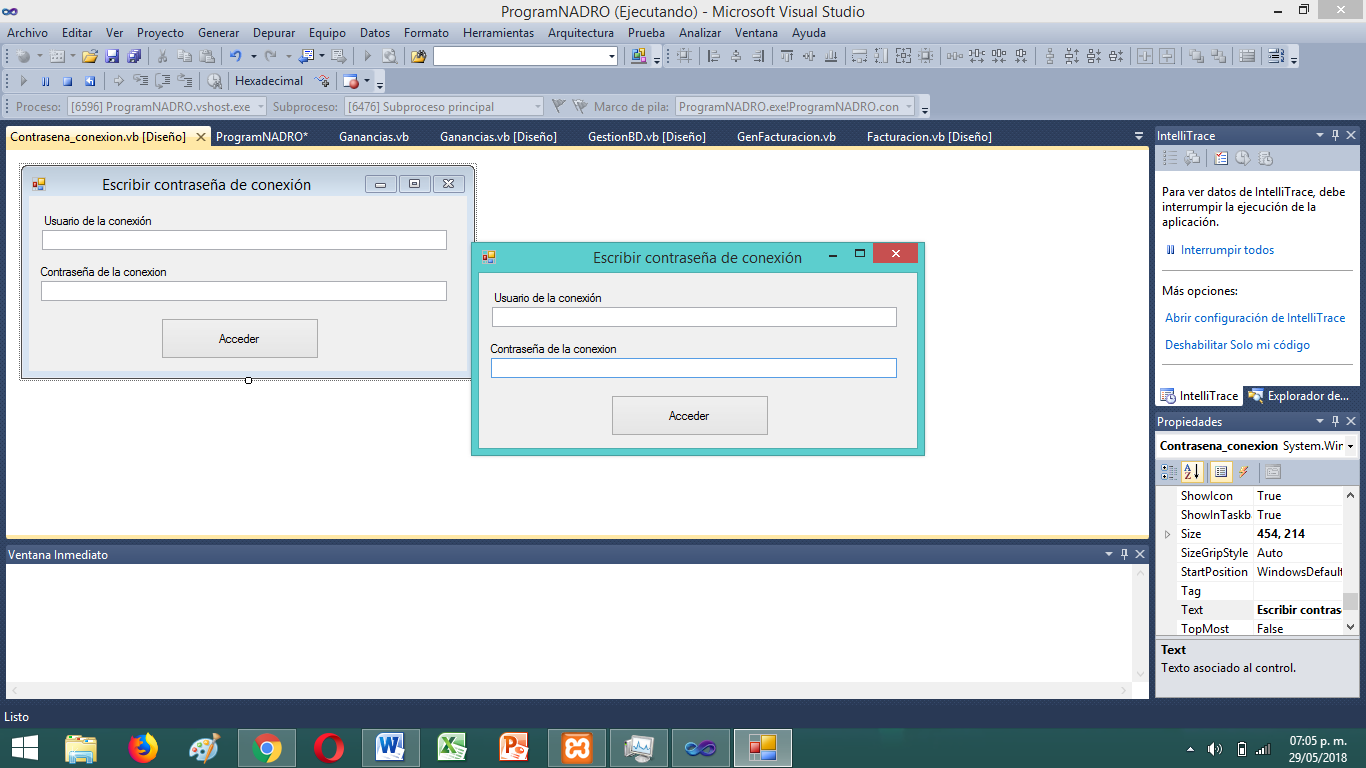
* Explicar lo más claramente posible el correcto funcionamiento del sistema, esto para poder utilizarlo correctamente y de la mejor manera posible.
* Dar mantenimiento efectivo al sistema con base en las buenas prácticas de programación.
* Descripción de los componentes generales y específicos que el sistema contiene.
* Explicar las bases fundamentales bajo las cuales se desarrolló este sistema.

PROCEDIMIENTOS (GENERACIÓN DEL SISTEMA)

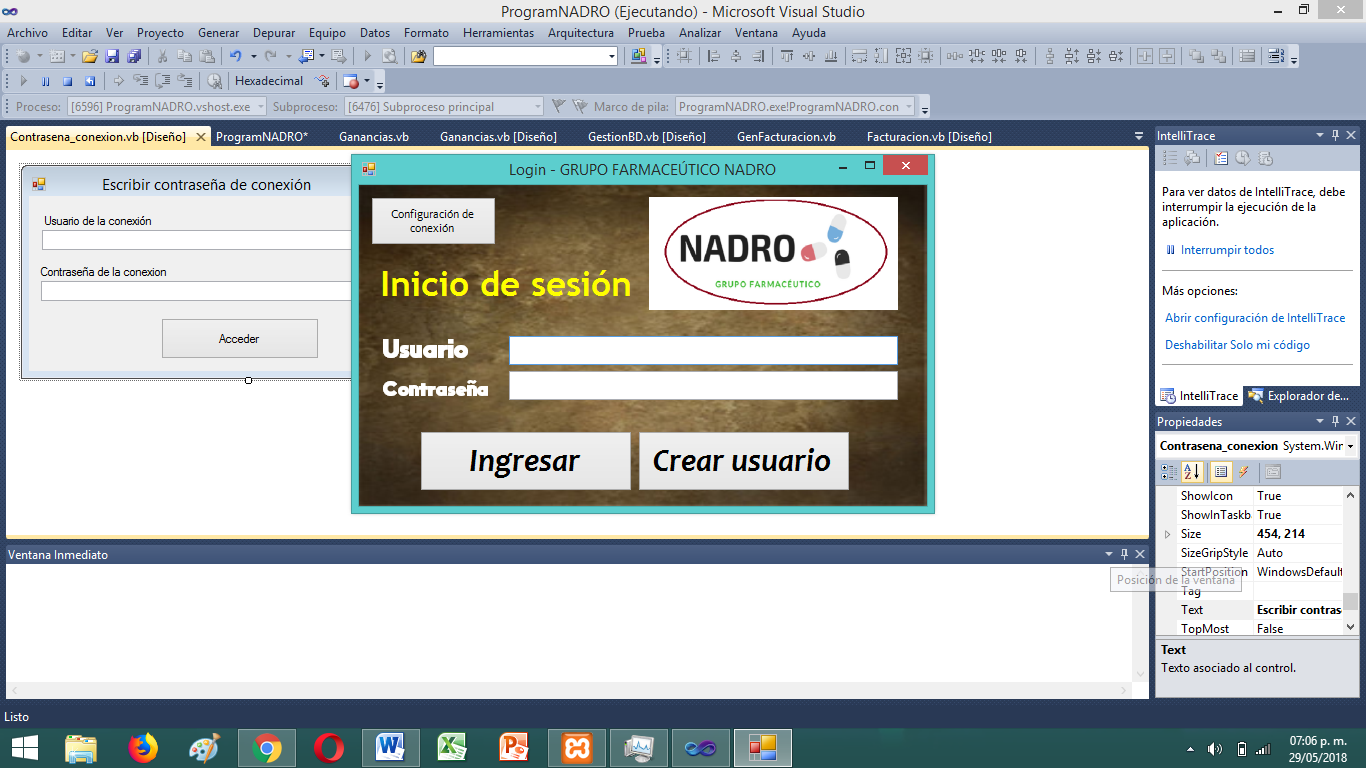
Todo el sistema fue realizado con Visual Studio 2010, las bases de datos en Php my admin y se cargaron al servidor local como prueba en su realización sin embargo se pueden montar en cualquier servidor.

Por ello se instalaron los programas de Visual Studio 2010 y Xampp.

En cada formulario se usaron botones, colores para darle diseño, cajas de texto, labels, imágenes, radio buttons, listas desplegables y las secciones, los controladores que dan paso a las bases de datos y en la ventana principal se usa una plantilla de menú.

**CONECTAR A LAS BASES DE DATOS**

Es indispensable conectar a las bases de datos ya que si no, no tendríamos acceso a los registros o simplemente el sistema no funcionaría correctamente, en el código se usaron las sentencias de control if para poder coincidir cuando se pone el usuario y la contraseña y si no mandar mensajes de alerta o que se guardó exitosamente

**VENTANA FORM\_CONFIG\_CONEXION**

Public Class Form\_Config\_Conexion

   Public direccion As String

   Dim conexion As conexion = New conexion()

   Private Sub btn\_aplicar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_aplicar.Click

       Dim condicion As Boolean = True

       Try

           Dim txtoct1 As Integer = Int(txt\_oct1.Text)

           Dim txtoct2 As Integer = Int(txt\_oct2.Text)

           Dim txtoct3 As Integer = Int(txt\_oct3.Text)

           Dim txtoct4 As Integer = Int(txt\_oct4.Text)

           If txtoct1 > 255 Or txtoct1 < 0 Then

               condicion = False

           End If

           If txtoct2 > 255 Or txtoct2 < 0 Then

               condicion = False

           End If

           If txtoct3 > 255 Or txtoct3 < 0 Then

               condicion = False

           End If

           If txtoct4 > 255 Or txtoct4 < 0 Then

               condicion = False

           End If

           If condicion = False Then

               MessageBox.Show("Los valores deben situarse entre 0 y 255")

           Else

               direccion = txt\_oct1.Text + "." + txt\_oct2.Text + "." + txt\_oct3.Text + "." + txt\_oct4.Text

               conexion.NuevaIP(direccion)

               MessageBox.Show("CONFIGURACIÓN GUARDADA EXITOSAMENTE")

           End If

       Catch ex As Exception

           MessageBox.Show(ex.ToString)

       End Try

   End Sub

   Private Sub Form\_Config\_Conexion\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

   End Sub

End Class

Al abrir el programa es así como luce la ventana de acceso a la cuenta de usuario de cada empleado. En el apartado se usan las cajas de texto y botones que es donde se configura los códigos que están debajo donde importamos la base de datos para ver la coincidencia de los mismos, se declaran las variables y se verifica la conexión con la base de datos.

**VENTANA LOGIN**

Imports MySql.Data

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class Form\_Login

   Dim conexion As conexion = New conexion()

   Dim acceso As Boolean

   Public tipouser As String

   Private Sub Form\_Login\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       conexion.ObtenerIP()

       txtb\_contrasena.PasswordChar = "•"

   End Sub

   Public Sub doLogin()

       Dim usuario As String = txtb\_usuario.Text

       Dim contrasena As String = txtb\_contrasena.Text

       Dim verifcontra As String

       conexion.Consulta("select Contrasena, Tipo from login where Usuario= '" + usuario + "'", "login")

       dgVLogin.DataSource = conexion.ds.Tables("login")

       verifcontra = dgVLogin.Rows(0).Cells(0).Value.ToString

       tipouser = dgVLogin.Rows(0).Cells(1).Value.ToString

       If verifcontra = contrasena Then

           acceso = True

       Else

           acceso = False

           MessageBox.Show("Cuenta de usuario o contraseña incorrectas")

       End If

   End Sub

   Private Sub DataGridView1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)

   End Sub

   Private Sub btn\_ingresar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_ingresar.Click

       If txtb\_usuario.Text = "" Or txtb\_contrasena.Text = "" Then

           MessageBox.Show("¿No te has olvidado de llenar algún campo?")

           Exit Sub

       End If

       doLogin()

       If acceso = True Then

           MessageBox.Show("INICIO DE SESIÓN")

           txtb\_usuario.Text = ""

           txtb\_contrasena.Text = ""

           MenuPrincipal.Show()

           Me.Hide()

       End If

   End Sub

   Private Sub btn\_crear\_user\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_crear\_user.Click

       Form\_Nuevo\_Usuario.Show()

   End Sub

   Private Sub btn\_ConexionIP\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_ConexionIP.Click

       Form\_Config\_Conexion.Show()

   End Sub

   Private Sub txtb\_contrasena\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtb\_contrasena.TextChanged

   End Sub

End Class

Cuando se verifican también con una sentencia de control IF y se arrojan los mensajes.

**VENTANA NUEVO USUARIO**

Public Class Form\_Nuevo\_Usuario

   Dim conexion As conexion = New conexion

   Dim categoria As String

   Private Sub dgVNuevo\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgVNuevo.CellContentClick

   End Sub

   Private Sub btn\_Nuevo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Nuevo.Click

       Dim verificador As String = ConfirmaContra(txta\_contrasena.Text, txta\_confirm.Text)

       Dim verifrepetido As Boolean = True

       Dim verifespacios As Boolean = True

       Dim revision As String

       Dim esp As String

       Dim lar As String

       Dim u As Integer

       Dim o As Integer

       For o = 1 To txta\_usuario.Text.Length

           esp = Mid(txta\_usuario.Text, o, 1)

           If esp = " " Then

               verifespacios = False

           End If

       Next

       For o = 1 To txta\_contrasena.Text.Length

           esp = Mid(txta\_contrasena.Text, o, 1)

           If esp = " " Then

               verifespacios = False

           End If

       Next

           If verifespacios Then

           If (radbtn\_Gerencial.Checked = True Or radbtn\_Otros.Checked = True) And txta\_usuario.Text.Length <> 0 And txta\_contrasena.Text.Length <> 0 Then

               If txta\_contrasena.Text.Length >= 8 Then

                   If (verificador) Then

                       conexion.Consulta("select Usuario from login", "login")

                       dgVNuevo.DataSource = conexion.ds.Tables("login")

                       lar = dgVNuevo.RowCount - 2

                       Dim reg\_usuario As String = txta\_usuario.Text

                       For u = 0 To lar

                           revision = dgVNuevo.Rows(u).Cells(0).Value.ToString

                           verifrepetido = confirmrepeat(reg\_usuario, revision)

                           If verifrepetido = False Then

                               Exit For

                           End If

                       Next

                       If verifrepetido = True Then

                           conexion.Consulta("select Usuario from req\_nuevos\_usuarios", "req\_nuevos\_usuarios")

                           dgVNuevo.DataSource = conexion.ds.Tables("req\_nuevos\_usuarios")

                           lar = dgVNuevo.RowCount - 2

                           MsgBox(lar)

                           For u = 0 To lar

                               revision = dgVNuevo.Rows(u).Cells(0).Value.ToString

                               verifrepetido = confirmrepeat(reg\_usuario, revision)

                               If verifrepetido = False Then

                                   Exit For

                               End If

                           Next

                       End If

                       If verifrepetido = True Then

                           Dim adduser As String

                           adduser = "INSERT INTO req\_nuevos\_usuarios VALUES(NULL, '" + txta\_usuario.Text + "', '" + txta\_contrasena.Text + "', '" + categoria + "')"

                           MessageBox.Show("USUARIO AGREGADO, Ahora sólo espera la verificación de tu cuenta")

                           conexion.Insertar(adduser)

                       Else

                           MsgBox("Este nombre de usuario YA EXISTE. Prueba otro nombre")

                       End If

                   Else

                       MsgBox("CONTRASEÑA Y CONFIRMACIÓN NO COINCIDEN")

                   End If

               Else

                   MessageBox.Show("La contraseña debe contener al menos 8 caracteres")

               End If

           Else

               MessageBox.Show("Has olvidado llenar algún rubro")

           End If

       Else

           MessageBox.Show("No se permite el uso de espacios ni en el usuario ni en la contraseña")

       End If

   End Sub

   Function ConfirmaContra(ByVal contrasena, ByVal confirm)

       If contrasena = confirm Then

           Return True

       Else

           Return False

       End If

   End Function

   Function confirmrepeat(ByVal usuario As String, ByVal registro As String)

       If usuario = registro Then

           Return False

       Else

           Return True

       End If

   End Function

   Private Sub radbtn\_Gerencial\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radbtn\_Gerencial.CheckedChanged

       categoria = "A"

   End Sub

   Private Sub radbtn\_Otros\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radbtn\_Otros.CheckedChanged

       categoria = "B"

   End Sub

   Private Sub Form\_Nuevo\_Usuario\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       dgVNuevo.Visible = False

   End Sub

End Class

Como parte del sistema aparece la opción que es para poder agregar un nuevo usuario en esta parte como muestra el código se declaran bastantes variables, otra sentencia de control en este caso FOR para un contador de los caracteres introducidos y también un IF para corroborar los nuevos usuarios y la contraseña de activación en el sistema, obviamente con los ELSE mostrando los mensajes correspondientes y la conexión al nuevo formulario si todo es correcto con el SHOW. **FACTURACIÓN**

Public Class Facturacion

   Dim conexion As conexion = New conexion()

   Dim m As Integer = 0

   Dim idnuevo As String = ""

   Dim celda As Integer

   Private Sub Facturacion\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       lbl\_fact\_realizada.Visible = False

       btn\_Cancelar.Visible = False

       btn\_Elim.Visible = False

       conexion.Conectar("127.0.0.1")

       btn\_NuevoItem.Visible = False

   End Sub   Private Sub btn\_ObtenerInfo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_ObtenerInfo.Click

       conexion.Consulta("select \* from medicamentos where IDMed = '" + txt\_id\_medic.Text + "'", "medicamentos")

       dgVMedicamentos.DataSource = conexion.ds.Tables("medicamentos")

       btn\_NuevoItem.Visible = True

   End Sub

   Private Sub btn\_NuevoItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_NuevoItem.Click

       If m = 12 Then

           MessageBox.Show("Sólo se pueden añadir 12 artículos diferentes por factura")

           Exit Sub

       End If

       Try

           Dim subtotal As Decimal

           If txt\_piezas.Text = "" Or txt\_id\_medic.Text = "" Or dgVMedicamentos.Rows(0).Cells(0).Value = "" Then

               MessageBox.Show("Asegúrate de haber llenado los campos correctamente y haber obtenido la información del medicamento")

               Exit Sub

           End If

           Try

               Convert.ToDecimal(txt\_piezas.Text)

           Catch ex As Exception

               MessageBox.Show("La cantidad de piezas debe ser un NÚMERO")

               Exit Sub

           End Try

           dgVFactura.Rows.Add()

           dgVFactura.Rows(m).Cells(1).Value = dgVMedicamentos.Rows(0).Cells(1).Value

           subtotal = dgVMedicamentos.Rows(0).Cells(3).Value

           dgVFactura.Rows(m).Cells(2).Value = subtotal

           dgVFactura.Rows(m).Cells(3).Value = subtotal \* Convert.ToDecimal(txt\_piezas.Text)

           dgVFactura.Rows(m).Cells(0).Value = txt\_piezas.Text

           m += 1

       Catch ex As Exception

           MessageBox.Show("Asegúrate de haber llenado los campos correctamente y haber obtenido la información del medicamento")

       End Try

   End Sub

   Private Sub txt\_id\_medic\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_id\_medic.TextChanged

   End Sub

   Private Sub txt\_piezas\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_piezas.TextChanged

   End Sub

   Private Sub btn\_GenerarFactura\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_GenerarFactura.Click

       If txt\_nombreCliente.Text = "" Or txt\_Direccion.Text = "" Or txt\_Ciudad.Text = "" Or txt\_CodigoP.Text = "" Or txt\_Telefono.Text = "" Or txt\_Telefono.Text = "" Or txt\_RFC.Text = "" Or m = 0 Then

           MessageBox.Show("Asegúrate de haber llenado todos los datos del cliente y haber ingresado medicamentos")

           Exit Sub

       End If

       If MsgBox("Los datos de la factura serán ingresados con los siguientes datos del cliente: " +

                       vbCrLf + "Nombre: " + txt\_nombreCliente.Text + vbCrLf + vbCrLf +

                       "Direccion: " + txt\_Direccion.Text + vbCrLf +

                       "Ciudad: " + txt\_Ciudad.Text + vbCrLf +

                       "Código Postal: " + txt\_CodigoP.Text + vbCrLf +

                       "Teléfono: " + txt\_Telefono.Text + vbCrLf +

                       "RFC: " + txt\_RFC.Text + vbCrLf + vbCrLf + "¿Deseas proceder con la operación?", vbYesNo) = vbYes Then

           Try

               GenerarIDFactura()

               If idnuevo = "ZZZZZZZZZZ" Then

                   MessageBox.Show("IDs de Facturación agotadas")

                   Exit Sub

               End If

               If conexion.Insertar("INSERT INTO idfacturas VALUES (NULL, '" + idnuevo + "')") Then

               End If

               If conexion.Insertar("INSERT INTO facturaciones VALUES ('" + idnuevo + "', '" + txt\_nombreCliente.Text + "', '" + txt\_Direccion.Text + "', '" + txt\_Ciudad.Text + "', '" + txt\_CodigoP.Text + "', '" + txt\_Telefono.Text + "', '" + txt\_RFC.Text + "')") Then

               End If

               m = 0

               Dim s As Integer

               Dim piezas As String

               Dim desc As String

               Dim precio As String

               Dim subtotal As String

               s = dgVFactura.RowCount - 2

               conexion.Consulta("SELECT IDFactura FROM idfacturas where ID = (SELECT MAX(ID) FROM idfacturas)", "idfacturas")

               dgVUltimoregistro.DataSource = conexion.ds.Tables("idfacturas")

               For o = 0 To s

                   piezas = dgVFactura.Rows(o).Cells(0).Value.ToString

                   desc = dgVFactura.Rows(o).Cells(1).Value.ToString

                   precio = dgVFactura.Rows(o).Cells(2).Value.ToString

                   subtotal = dgVFactura.Rows(o).Cells(3).Value.ToString

                   conexion.Insertar("INSERT INTO reg\_facturas VALUES (NULL, '" + idnuevo + "', '" + piezas + "', '" + desc + "', '" + precio + "', '" + subtotal + "')")

               Next

               btn\_NuevoItem.Visible = False

               lbl\_fact\_realizada.Visible = True

           Catch ex As Exception

               MessageBox.Show("Error en el proceso")

           End Try

       Else

           Exit Sub

       End If

   End Sub

   Public Sub GenerarIDFactura()

       Dim ultimoid As String

       Dim esnum As Boolean

       Dim arregloid As Array

       Dim nuevoid() As String = {"0", "0", "0", "0", "0", "0", "0", "0", "0", "0"}

       Dim verif As Integer

       idnuevo = ""

       conexion.Consulta("SELECT IDFactura FROM idfacturas where ID = (SELECT MAX(ID) FROM idfacturas)", "idfacturas")

       dgVUltimoregistro.DataSource = conexion.ds.Tables("idfacturas")

       ultimoid = dgVUltimoregistro.Rows(0).Cells(0).Value.ToString

       arregloid = ultimoid.ToCharArray

       For o = 0 To 9

           Try

               verif = Int(arregloid(9 - o))

               esnum = True

           Catch ex As Exception

               esnum = False

           End Try

           If esnum = True Then

               If arregloid(9 - o) = "9" Then

                   nuevoid(9 - o) = "A"

                   Exit For

               Else

                   verif = verif + 1

                   nuevoid(9 - o) = verif.ToString

                   Exit For

               End If

           Else

               If arregloid(9 - o) = "Z" Then

                   nuevoid(9 - o) = "0"

                   Continue For

               Else

                   Dim caracter As String

                   Dim valor As Integer

                   valor = Asc(arregloid(9 - o))

                   valor = valor + 1

                   caracter = Chr(valor)

                   nuevoid(9 - o) = caracter

                   Exit For

               End If

           End If

       Next

       For u = 0 To 9

           idnuevo = idnuevo + nuevoid(u)

       Next

   End Sub

   Private Sub btn\_Selecc\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Selecc.Click

       Try

           celda = dgVFactura.CurrentRow.Index

           btn\_Elim.Visible = True

       Catch ex As Exception

           MessageBox.Show("No ha seleccionado ninguna celda")

       End Try

   End Sub

   Private Sub btn\_Elim\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Elim.Click

       Dim tabceldas, ind As Integer

       tabceldas = dgVFactura.RowCount - 2

       If tabceldas = -1 Then

           Exit Sub

       End If

       For o = celda To tabceldas

           dgVFactura.Rows(o).Cells(0).Value = ""

           dgVFactura.Rows(o).Cells(1).Value = ""

           dgVFactura.Rows(o).Cells(2).Value = ""

           dgVFactura.Rows(o).Cells(3).Value = ""

           dgVFactura.Rows(o).Cells(0).Value = dgVFactura.Rows(o + 1).Cells(0).Value

           dgVFactura.Rows(o).Cells(1).Value = dgVFactura.Rows(o + 1).Cells(1).Value

           dgVFactura.Rows(o).Cells(2).Value = dgVFactura.Rows(o + 1).Cells(2).Value

           dgVFactura.Rows(o).Cells(3).Value = dgVFactura.Rows(o + 1).Cells(3).Value

       Next

       ind = dgVFactura.Rows(tabceldas).Index

       dgVFactura.Rows.Remove(dgVFactura.Rows(ind))

       m -= 1

   End Sub

   Private Sub btn\_Cancelar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Cancelar.Click

       btn\_Elim.Visible = False

       btn\_Cancelar.Visible = False

       btn\_Selecc.Visible = True

   End Sub

   Private Sub btn\_Archivo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Archivo.Click

   End Sub

End Class

**VENTANA FACTURACIÓN**

      Private Sub txt\_piezas\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_piezas.TextChanged

   End Sub

   Private Sub btn\_GenerarFactura\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_GenerarFactura.Click

       If txt\_nombreCliente.Text = "" Or txt\_Direccion.Text = "" Or txt\_Ciudad.Text = "" Or txt\_CodigoP.Text = "" Or txt\_Telefono.Text = "" Or txt\_Telefono.Text = "" Or txt\_RFC.Text = "" Or m = 0 Then

           MessageBox.Show("Asegúrate de haber llenado todos los datos del cliente y haber ingresado medicamentos")

           Exit Sub

       End If

       If MsgBox("Los datos de la factura serán ingresados con los siguientes datos del cliente: " +

                       vbCrLf + "Nombre: " + txt\_nombreCliente.Text + vbCrLf + vbCrLf +

                       "Direccion: " + txt\_Direccion.Text + vbCrLf +

                       "Ciudad: " + txt\_Ciudad.Text + vbCrLf +

                       "Código Postal: " + txt\_CodigoP.Text + vbCrLf +

                       "Teléfono: " + txt\_Telefono.Text + vbCrLf +

                       "RFC: " + txt\_RFC.Text + vbCrLf + vbCrLf + "¿Deseas proceder con la operación?", vbYesNo) = vbYes Then

           Try

               GenerarIDFactura()

               If idnuevo = "ZZZZZZZZZZ" Then

                   MessageBox.Show("IDs de Facturación agotadas")

                   Exit Sub

               End If

               If conexion.Insertar("INSERT INTO idfacturas VALUES (NULL, '" + idnuevo + "')") Then

               End If

               If conexion.Insertar("INSERT INTO facturaciones VALUES ('" + idnuevo + "', '" + txt\_nombreCliente.Text + "', '" + txt\_Direccion.Text + "', '" + txt\_Ciudad.Text + "', '" + txt\_CodigoP.Text + "', '" + txt\_Telefono.Text + "', '" + txt\_RFC.Text + "')") Then

               End If

               m = 0

               Dim s As Integer

               Dim piezas As String

               Dim desc As String

               Dim precio As String

               Dim subtotal As String

               s = dgVFactura.RowCount - 2

               conexion.Consulta("SELECT IDFactura FROM idfacturas where ID = (SELECT MAX(ID) FROM idfacturas)", "idfacturas")

               dgVUltimoregistro.DataSource = conexion.ds.Tables("idfacturas")

               For o = 0 To s

                   piezas = dgVFactura.Rows(o).Cells(0).Value.ToString

                   desc = dgVFactura.Rows(o).Cells(1).Value.ToString

                   precio = dgVFactura.Rows(o).Cells(2).Value.ToString

                   subtotal = dgVFactura.Rows(o).Cells(3).Value.ToString

                   conexion.Insertar("INSERT INTO reg\_facturas VALUES (NULL, '" + idnuevo + "', '" + piezas + "', '" + desc + "', '" + precio + "', '" + subtotal + "')")

               Next

               btn\_NuevoItem.Visible = False

               lbl\_fact\_realizada.Visible = True

           Catch ex As Exception

               MessageBox.Show("Error en el proceso")

           End Try

       Else

           Exit Sub

       End If

   End Sub

   Public Sub GenerarIDFactura()

       Dim ultimoid As String

       Dim esnum As Boolean

       Dim arregloid As Array

       Dim nuevoid() As String = {"0", "0", "0", "0", "0", "0", "0", "0", "0", "0"}

       Dim verif As Integer

       idnuevo = ""

       conexion.Consulta("SELECT IDFactura FROM idfacturas where ID = (SELECT MAX(ID) FROM idfacturas)", "idfacturas")

       dgVUltimoregistro.DataSource = conexion.ds.Tables("idfacturas")

       ultimoid = dgVUltimoregistro.Rows(0).Cells(0).Value.ToString

       arregloid = ultimoid.ToCharArray

       For o = 0 To 9

           Try

               verif = Int(arregloid(9 - o))

               esnum = True

           Catch ex As Exception

               esnum = False

           End Try

           If esnum = True Then

               If arregloid(9 - o) = "9" Then

                   nuevoid(9 - o) = "A"

                   Exit For

               Else

                   verif = verif + 1

                   nuevoid(9 - o) = verif.ToString

                   Exit For

               End If

           Else

               If arregloid(9 - o) = "Z" Then

                   nuevoid(9 - o) = "0"

                   Continue For

               Else

                   Dim caracter As String

                   Dim valor As Integer

                   valor = Asc(arregloid(9 - o))

                   valor = valor + 1

                   caracter = Chr(valor)

                   nuevoid(9 - o) = caracter

                   Exit For

               End If

           End If

       Next

       For u = 0 To 9

           idnuevo = idnuevo + nuevoid(u)

       Next

   End Sub

   Private Sub btn\_Selecc\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Selecc.Click

       Try

           celda = dgVFactura.CurrentRow.Index

           btn\_Elim.Visible = True

       Catch ex As Exception

           MessageBox.Show("No ha seleccionado ninguna celda")

       End Try

   End Sub

   Private Sub btn\_Elim\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Elim.Click

       Dim tabceldas, ind As Integer

       tabceldas = dgVFactura.RowCount - 2

       If tabceldas = -1 Then

           Exit Sub

       End If

       For o = celda To tabceldas

           dgVFactura.Rows(o).Cells(0).Value = ""

           dgVFactura.Rows(o).Cells(1).Value = ""

           dgVFactura.Rows(o).Cells(2).Value = ""

           dgVFactura.Rows(o).Cells(3).Value = ""

           dgVFactura.Rows(o).Cells(0).Value = dgVFactura.Rows(o + 1).Cells(0).Value

           dgVFactura.Rows(o).Cells(1).Value = dgVFactura.Rows(o + 1).Cells(1).Value

           dgVFactura.Rows(o).Cells(2).Value = dgVFactura.Rows(o + 1).Cells(2).Value

           dgVFactura.Rows(o).Cells(3).Value = dgVFactura.Rows(o + 1).Cells(3).Value

       Next

       ind = dgVFactura.Rows(tabceldas).Index

       dgVFactura.Rows.Remove(dgVFactura.Rows(ind))

       m -= 1

   End Sub

   Private Sub btn\_Cancelar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Cancelar.Click

       btn\_Elim.Visible = False

       btn\_Cancelar.Visible = False

       btn\_Selecc.Visible = True

   End Sub

   Private Sub btn\_Archivo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Archivo.Click

   End Sub

End Class

En el código volvemos a ratificar la conexión con las bases de datos que vendría siendo lo mismo con casi todos los procesos que tiene el sistema con Dim conexión

Después con clases y funciones asignamos las diferentes tareas y apoyándose de las sentencias de control, como para sólo añadir 12 artículos, asegurarte de haber llenado todos los campos y entre otras tareas necesarias para que el proceso se pueda entender. Se indica en las cajas donde se depositará la información y se usaron arreglos para ordenar la factura.

Public Class Ganancias

   Dim fecha As String

   Dim conexion As conexion = New conexion()

   Dim id As String

   Dim fechaventa As String

   Dim ingreso As Decimal

   Dim suma As Decimal = 0

   Private Sub btn\_Ingresos\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Ingresos.Click

       fecha = DateString

       fecha = Now.ToString("yyyy/MM/dd")

       id = 0

       If radbtn\_Mes.Checked = True Then

           Do

               Try

                   conexion.Consulta("select Monto, Fecha from ventas where ID" = id, "ventas")

                   dgVGenVentas.DataSource = conexion.ds.Tables("ventas")

                   fechaventa = dgVGenVentas.Rows(0).Cells(1).Value.ToString

                   If fechaventa(5) = fecha(5) And fechaventa(6) = fecha(6) Then

                       suma += Convert.ToDecimal(dgVGenVentas.Rows(0).Cells(0).Value)

                   End If

                   id += 1

               Catch ex As Exception

                   Exit Do

               End Try

           Loop

           ingreso = suma

           txt\_Ingreso.Text = ingreso

       End If

       If radbtn\_Anual.Checked = True Then

           Do

               Try

                   conexion.Consulta("select Monto, Fecha from ventas where ID" = id, "ventas")

                   dgVGenVentas.DataSource = conexion.ds.Tables("ventas")

                   fechaventa = dgVGenVentas.Rows(0).Cells(1).Value.ToString

                   If fechaventa(0) = fecha(0) And fechaventa(1) = fecha(1) And fechaventa(2) = fecha(2) And fechaventa(3) = fecha(3) Then

                       suma += Convert.ToDecimal(dgVGenVentas.Rows(0).Cells(0).Value)

                   End If

                   id += 1

               Catch ex As Exception

                   Exit Do

               End Try

           Loop

           ingreso = suma

           txt\_Ingreso.Text = ingreso

       End If

   End Sub

   Private Sub Ganancias\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       conexion.Conectar("localhost")

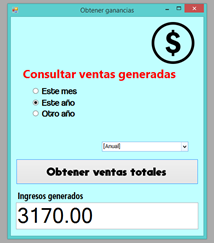
   End Sub

End Class

**VENTANA GANANCIAS**

Obtener ganancias de un periodo

En este apartado dentro de ventas se pueden consultar todo el dinero o ingresos de la farmacia en tres niveles, mes, años y cualquier otro año que no sea el actual.



**VENTANA GENERAR FACTURACIÓN**

Public Class GenFacturacion

   Dim conexion As conexion = New conexion()

   Private Sub btn\_Busqueda\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Busqueda.Click

       conexion.Consulta("select \* from facturaciones where IDFacturacion = '" + txt\_IDFactura.Text + "'", "facturaciones")

       dgVFacturacion.DataSource = conexion.ds.Tables("facturaciones")

       Try

           txt\_nombre.Text = dgVFacturacion.Rows(0).Cells(1).Value.ToString

           txt\_direccion.Text = dgVFacturacion.Rows(0).Cells(2).Value.ToString

           txt\_ciudad.Text = dgVFacturacion.Rows(0).Cells(3).Value.ToString

           txt\_codigop.Text = dgVFacturacion.Rows(0).Cells(4).Value.ToString

           txt\_telefono.Text = dgVFacturacion.Rows(0).Cells(5).Value.ToString

           txt\_RFC.Text = dgVFacturacion.Rows(0).Cells(6).Value.ToString

       Catch ex As Exception

           MessageBox.Show("No hay facturas asociadas al ID ingresada (" + txt\_IDFactura.Text + ")")

       End Try

       conexion.Consulta("select Piezas, Descripcion, Precio, Subtotal from reg\_facturas where IDFactura = '" + txt\_IDFactura.Text + "'", "reg\_facturas")

       dgVArticulos.DataSource = conexion.ds.Tables("reg\_facturas")

   End Sub

   Private Sub GenFacturacion\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       dgVFacturacion.Visible = False

       txt\_nombre.ReadOnly = True

       txt\_direccion.ReadOnly = True

       txt\_ciudad.ReadOnly = True

       txt\_telefono.ReadOnly = True

       txt\_codigop.ReadOnly = True

       txt\_RFC.ReadOnly = True

   End Sub

End Class

Digamos que esta parte es de las más difíciles por el manejo correcto de la base de datos, los datos acomodados correctamente a través de los arreglos, con el único resultado de tener una factura con un llenado automático y bueno con los valores y datos correctos, las estructuras antes mencionadas también aplican para este apartado.

**VENTANA GESTIÓN DE LA BASE DE DATOS**

Public Class GestionBD

   Dim fecha, periodo As String

   Dim ventas As Boolean = False

   Dim prods As Boolean = False

   Dim facturaciones As Boolean = False

   Dim regfacturas As Boolean = False

   Dim conexion As conexion = New conexion()

   Dim validar As String

   Private Sub GestionBD\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       conexion.Conectar("localhost")

   End Sub

   Private Sub dgVRenovacion\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgVRenovacion.CellContentClick

   End Sub

   Private Sub btn\_Renovar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Renovar.Click

       Dim ext As Integer

       periodo = ""

       fecha = DateString

       fecha = Now.ToString("yyyy/MM/dd")

       conexion.Consulta("SELECT \* FROM permitrenovacion", "permitrenovacion")

       dgVRenovacion.DataSource = conexion.ds.Tables("permitrenovacion")

       Try

           validar = dgVRenovacion.Rows(0).Cells(0).Value.ToString

           ext = dgVRenovacion.RowCount

           If dgVRenovacion.Rows(0).Cells(1).Value = "TRUE" Then

               ventas = True

           Else

               ventas = False

           End If

           If dgVRenovacion.Rows(0).Cells(2).Value = "TRUE" Then

               prods = True

           Else

               prods = False

           End If

           If dgVRenovacion.Rows(0).Cells(3).Value = "TRUE" Then

               facturaciones = True

           Else

               facturaciones = False

           End If

           If dgVRenovacion.Rows(0).Cells(4).Value = "TRUE" Then

               regfacturas = True

           Else

               regfacturas = False

           End If

       Catch ex As Exception

           validar = ""

           ext = 0

       End Try

       MessageBox.Show(ext)

       MessageBox.Show(validar)

       If validar = "TRUE" Then

       Else

           If fecha(5) = "1" And fecha(6) = "2" And fecha(8) = "3" And fecha(9) = "1" Then

               If ext = 0 Then

                   MessageBox.Show("Soltó excepción")

                   conexion.Insertar("INSERT INTO permitrenovacion VALUES('TRUE', 'FALSE', 'FALSE', 'FALSE', 'FALSE')")

               Else

                   conexion.Insertar("UPDATE permitrenovacion SET Validar='TRUE' WHERE Validar = 'FALSE'")

               End If

           Else

               If ext = 0 Then

                   MessageBox.Show("Entró a la condición")

                   conexion.Insertar("INSERT INTO permitrenovacion VALUES('FALSE', 'FALSE', 'FALSE', 'FALSE', 'FALSE')")

               Else

                   conexion.Insertar("UPDATE permitrenovacion SET Validar='FALSE' WHERE Validar = 'FALSE'")

               End If

               Exit Sub

           End If

       End If

       If radbtn\_Ventas.Checked = True Then

           If ventas = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE ventas\_" + periodo + " like ventas")

               conexion.Insertar("INSERT INTO ventas\_" + periodo + " SELECT \* FROM ventas")

               conexion.Insertar("delete from ventas")

               conexion.Insertar("alter table ventas AUTO\_INCREMENT = 1")

               ventas = True

               conexion.Insertar("UPDATE permitrenovacion SET Ventas='TRUE' WHERE Ventas = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de la tabla Ventas")

           End If

       End If

       If radbtn\_Productos\_vendidos.Checked = True Then

           If prods = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE prod\_vendidos\_" + periodo + " like prod\_vendidos")

               conexion.Insertar("INSERT INTO prod\_vendidos\_" + periodo + " SELECT \* FROM prod\_vendidos")

               conexion.Insertar("delete from prod\_vendidos")

               conexion.Insertar("alter table prod\_vendidos AUTO\_INCREMENT = 1")

               prods = True

               conexion.Insertar("UPDATE permitrenovacion SET Prods='TRUE' WHERE Prods = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Productos vendidos")

           End If

       End If

       If radbtn\_Facturas.Checked = True Then

           If facturaciones = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE facturaciones\_" + periodo + " like facturaciones")

               conexion.Insertar("INSERT INTO facturaciones\_" + periodo + " SELECT \* FROM facturaciones")

               conexion.Insertar("delete from facturaciones")

               conexion.Insertar("delete from idfacturas")

               conexion.Insertar("alter table idfacturas AUTO\_INCREMENT = 1")

               conexion.Insertar("INSERT INTO idfacturas VALUES(NULL, '0000000000')")

               facturaciones = True

               conexion.Insertar("UPDATE permitrenovacion SET Facturaciones='TRUE' WHERE Facturaciones = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Facturas")

           End If

       End If

       If radbtn\_regfacturas.Checked = True Then

           If regfacturas = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE reg\_facturas\_" + periodo + " like reg\_facturas")

               conexion.Insertar("INSERT INTO reg\_facturas\_" + periodo + " SELECT \* FROM reg\_facturas")

               conexion.Insertar("delete from reg\_facturas")

               conexion.Insertar("alter table reg\_facturas AUTO\_INCREMENT = 1")

               regfacturas = True

               conexion.Insertar("UPDATE permitrenovacion SET RegFacturas='TRUE' WHERE RegFacturas = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Registros de facturas")

           End If

       End If

       If ventas = True And prods = True And facturaciones = True And regfacturas = True Then

           conexion.Insertar("UPDATE permitrenovacion SET Validar='FALSE', Prods='FALSE', Facturaciones ='FALSE', Ventas='FALSE', RegFacturas='FALSE' WHERE Validar = 'TRUE'")

           prods = False

           facturaciones = False

           ventas = False

           regfacturas = False

       End If

   End Sub

   Private Sub radbtn\_regfacturas\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radbtn\_regfacturas.CheckedChanged

   End Sub

End Class

                   conexion.Insertar("UPDATE permitrenovacion SET Validar='TRUE' WHERE Validar = 'FALSE'")

               End If

           Else

               If ext = 0 Then

                   MessageBox.Show("Entró a la condición")

                   conexion.Insertar("INSERT INTO permitrenovacion VALUES('FALSE', 'FALSE', 'FALSE', 'FALSE', 'FALSE')")

               Else

                   conexion.Insertar("UPDATE permitrenovacion SET Validar='FALSE' WHERE Validar = 'FALSE'")

               End If

               Exit Sub

           End If

       End If

       If radbtn\_Ventas.Checked = True Then

           If ventas = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE ventas\_" + periodo + " like ventas")

               conexion.Insertar("INSERT INTO ventas\_" + periodo + " SELECT \* FROM ventas")

               conexion.Insertar("delete from ventas")

               conexion.Insertar("alter table ventas AUTO\_INCREMENT = 1")

               ventas = True

               conexion.Insertar("UPDATE permitrenovacion SET Ventas='TRUE' WHERE Ventas = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de la tabla Ventas")

           End If

       End If

       If radbtn\_Productos\_vendidos.Checked = True Then

           If prods = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE prod\_vendidos\_" + periodo + " like prod\_vendidos")

               conexion.Insertar("INSERT INTO prod\_vendidos\_" + periodo + " SELECT \* FROM prod\_vendidos")

               conexion.Insertar("delete from prod\_vendidos")

               conexion.Insertar("alter table prod\_vendidos AUTO\_INCREMENT = 1")

               prods = True

               conexion.Insertar("UPDATE permitrenovacion SET Prods='TRUE' WHERE Prods = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Productos vendidos")

           End If

       End If

       If radbtn\_Facturas.Checked = True Then

           If facturaciones = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE facturaciones\_" + periodo + " like facturaciones")

               conexion.Insertar("INSERT INTO facturaciones\_" + periodo + " SELECT \* FROM facturaciones")

               conexion.Insertar("delete from facturaciones")

               conexion.Insertar("delete from idfacturas")

               conexion.Insertar("alter table idfacturas AUTO\_INCREMENT = 1")

               conexion.Insertar("INSERT INTO idfacturas VALUES(NULL, '0000000000')")

               facturaciones = True

               conexion.Insertar("UPDATE permitrenovacion SET Facturaciones='TRUE' WHERE Facturaciones = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Facturas")

           End If

       End If

       If radbtn\_regfacturas.Checked = True Then

           If regfacturas = False Then

               For o = 0 To 3

                   periodo = periodo + fecha(o)

               Next

               conexion.Insertar("CREATE TABLE reg\_facturas\_" + periodo + " like reg\_facturas")

               conexion.Insertar("INSERT INTO reg\_facturas\_" + periodo + " SELECT \* FROM reg\_facturas")

               conexion.Insertar("delete from reg\_facturas")

               conexion.Insertar("alter table reg\_facturas AUTO\_INCREMENT = 1")

               regfacturas = True

               conexion.Insertar("UPDATE permitrenovacion SET RegFacturas='TRUE' WHERE RegFacturas = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Registros de facturas")

           End If

       End If

       If ventas = True And prods = True And facturaciones = True And regfacturas = True Then

           conexion.Insertar("UPDATE permitrenovacion SET Validar='FALSE', Prods='FALSE', Facturaciones ='FALSE', Ventas='FALSE', RegFacturas='FALSE' WHERE Validar = 'TRUE'")

           prods = False

           facturaciones = False

           ventas = False

           regfacturas = False

       End If

   End Sub

   Private Sub radbtn\_regfacturas\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radbtn\_regfacturas.CheckedChanged

   End Sub

End Class

               conexion.Insertar("delete from idfacturas")

 conexion.Insertar("alter table reg\_facturas AUTO\_INCREMENT = 1")

               regfacturas = True

               conexion.Insertar("UPDATE permitrenovacion SET RegFacturas='TRUE' WHERE RegFacturas = 'FALSE'")

           Else

               MessageBox.Show("Ya fue RENOVADO el registro de Registros de facturas")

           End If

       End If

       If ventas = True And prods = True And facturaciones = True And regfacturas = True Then

           conexion.Insertar("UPDATE permitrenovacion SET Validar='FALSE', Prods='FALSE', Facturaciones ='FALSE', Ventas='FALSE', RegFacturas='FALSE' WHERE Validar = 'TRUE'")

           prods = False

           facturaciones = False

           ventas = False

           regfacturas = False

       End If

   End Sub

   Private Sub radbtn\_regfacturas\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radbtn\_regfacturas.CheckedChanged

   End Sub

End Class

**VENTANA MEDICAMENTO**

Public Class Medicamentos

   Dim conexion As conexion = New conexion()

   Private Sub Label5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label5.Click

   End Sub

   Private Sub Medicamentos\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       mostrarMedicamentos()

   End Sub

   Public Sub mostrarMedicamentos()

       conexion.Consulta("select \* from medicamentos", "medicamentos")

       dgVMedicamentos.DataSource = conexion.ds.Tables("medicamentos")

   End Sub

   Public Sub mostrarConsultaMedic(ByVal atributo, ByVal valor)

       conexion.Consulta("select \* from medicamentos where " + atributo + "= '" + valor + "'", "medicamentos")

       dgVMedicamentos.DataSource = conexion.ds.Tables("medicamentos")

   End Sub

   Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

       Dim addmedim As String = "insert into medicamentos values('" + txt\_id\_med.Text + "', '" + txt\_nombre.Text + "', " + txt\_id\_prov.Text + ", " + txt\_prec\_pub.Text + ", " + txt\_prec\_prov.Text + ", " + txt\_stock.Text + ", '" + txt\_sustancia.Text + "', '" + txt\_gramaje.Text + "', '" + txt\_presentacion.Text + "')"

       If (conexion.Insertar(addmedim)) Then

           mostrarMedicamentos()

           MessageBox.Show("DATOS AGREGADOS EXITOSAMENTE")

       Else

           MessageBox.Show("ERROR")

       End If

   End Sub

   Private Sub btn\_modif\_med\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_modif\_med.Click

       ModifMedic.Show()

   End Sub

   Private Sub btn\_id\_medim\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_id\_medim.Click

       Dim valor As String = txtc\_id\_medim.Text

       mostrarConsultaMedic("IDMed", valor)

   End Sub

   Private Sub btn\_nombre\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_nombre.Click

       Dim valor As String = txtc\_nombre.Text

       mostrarConsultaMedic("Nombre", valor)

   End Sub

   Private Sub btn\_id\_prov\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_id\_prov.Click

       Dim valor As String = txtc\_id\_prov.Text

       mostrarConsultaMedic("IDProv", valor)

   End Sub

   Private Sub btn\_prec\_pub\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_prec\_pub.Click

       Dim valor As String = txtc\_prec\_pub.Text

       mostrarConsultaMedic("Precio\_publico", valor)

   End Sub

   Private Sub btn\_prec\_prov\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_prec\_prov.Click

       Dim valor As String = txtc\_prec\_prov.Text

       mostrarConsultaMedic("Precio\_proveedor", valor)

   End Sub

   Private Sub btn\_stock\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_stock.Click

       Dim valor As String = txtc\_stock.Text

       mostrarConsultaMedic("Stock", valor)

   End Sub

   Private Sub btn\_sustancia\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sustancia.Click

       Dim valor As String = txtc\_sustancia.Text

       mostrarConsultaMedic("Sustancia", valor)

   End Sub

   Private Sub btn\_gramaje\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gramaje.Click

       Dim valor As String = txtc\_gramaje.Text

       mostrarConsultaMedic("Gramaje", valor)

   End Sub

   Private Sub btn\_presentacion\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_presentacion.Click

       Dim valor As String = txtc\_presentacion.Text

       mostrarConsultaMedic("Presentacion", valor)

   End Sub

   Private Sub btn\_mostrar\_todo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_mostrar\_todo.Click

       mostrarMedicamentos()

   End Sub

   Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

       If (conexion.Eliminar("medicamentos", "IDMed= " + txt\_clave\_med.Text)) Then

           MessageBox.Show("Datos eliminados exitosamente")

           mostrarMedicamentos()

           If (conexion.ResetID("proveedores")) Then

               MessageBox.Show("ID CAMBIADO EXITOSAMENTE")

           End If

       Else

           MessageBox.Show("ERROR")

       End If

   End Sub

   Private Sub txt\_clave\_med\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_clave\_med.TextChanged

   End Sub

End Class

**V**

   End Sub

   Private Sub btn\_prec\_prov\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_prec\_prov.Click

       Dim valor As String = txtc\_prec\_prov.Text

       mostrarConsultaMedic("Precio\_proveedor", valor)

   End Sub

   Private Sub btn\_stock\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_stock.Click

       Dim valor As String = txtc\_stock.Text

       mostrarConsultaMedic("Stock", valor)

   End Sub

   Private Sub btn\_sustancia\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_sustancia.Click

       Dim valor As String = txtc\_sustancia.Text

       mostrarConsultaMedic("Sustancia", valor)

   End Sub

   Private Sub btn\_gramaje\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_gramaje.Click

       Dim valor As String = txtc\_gramaje.Text

       mostrarConsultaMedic("Gramaje", valor)

   End Sub

   Private Sub btn\_presentacion\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_presentacion.Click

       Dim valor As String = txtc\_presentacion.Text

       mostrarConsultaMedic("Presentacion", valor)

   End Sub

   Private Sub btn\_mostrar\_todo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_mostrar\_todo.Click

       mostrarMedicamentos()

   End Sub

   Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

       If (conexion.Eliminar("medicamentos", "IDMed= " + txt\_clave\_med.Text)) Then

           MessageBox.Show("Datos eliminados exitosamente")

           mostrarMedicamentos()

           If (conexion.ResetID("proveedores")) Then

               MessageBox.Show("ID CAMBIADO EXITOSAMENTE")

           End If

       Else

           MessageBox.Show("ERROR")

       End If

   End Sub

   Private Sub txt\_clave\_med\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_clave\_med.TextChanged

   End Sub

End Class

**VENTANA MENÚ PRINCIPAL**

Imports System.Windows.Forms

Public Class MenuPrincipal

   Public dirIP As String

   Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs)

       ' Cree una nueva instancia del formulario secundario.

       Dim ChildForm As New System.Windows.Forms.Form

       ' Conviértalo en un elemento secundario de este formulario MDI antes de mostrarlo.

       ChildForm.MdiParent = Me

       m\_ChildFormNumber += 1

       ChildForm.Text = "Ventana " & m\_ChildFormNumber

       ChildForm.Show()

   End Sub

   Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)

       Dim OpenFileDialog As New OpenFileDialog

       OpenFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

       OpenFileDialog.Filter = "Archivos de texto (\*.txt)|\*.txt|Todos los archivos (\*.\*)|\*.\*"

       If (OpenFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

           Dim FileName As String = OpenFileDialog.FileName

           ' TODO: agregue código aquí para abrir el archivo.

       End If

   End Sub

   Private Sub SaveAsToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       Dim SaveFileDialog As New SaveFileDialog

       SaveFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

       SaveFileDialog.Filter = "Archivos de texto (\*.txt)|\*.txt|Todos los archivos (\*.\*)|\*.\*"

       If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

           Dim FileName As String = SaveFileDialog.FileName

           ' TODO: agregue código aquí para guardar el contenido actual del formulario en un archivo.

       End If

   End Sub

   Private Sub ExitToolsStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       Me.Close()

   End Sub

   Private Sub CutToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       ' Utilice My.Computer.Clipboard para insertar el texto o las imágenes seleccionadas en el Portapapeles

   End Sub

   Private Sub CopyToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       ' Utilice My.Computer.Clipboard para insertar el texto o las imágenes seleccionadas en el Portapapeles

   End Sub

   Private Sub PasteToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       'Utilice My.Computer.Clipboard.GetText() o My.Computer.Clipboard.GetData para recuperar la información del Portapapeles.

   End Sub

   Private Sub CascadeToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       Me.LayoutMdi(MdiLayout.Cascade)

   End Sub

   Private Sub TileVerticalToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       Me.LayoutMdi(MdiLayout.TileVertical)

   End Sub

   Private Sub TileHorizontalToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       Me.LayoutMdi(MdiLayout.TileHorizontal)

   End Sub

   Private Sub ArrangeIconsToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       Me.LayoutMdi(MdiLayout.ArrangeIcons)

   End Sub

   Private Sub CloseAllToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

       ' Cierre todos los formularios secundarios del principal.

       For Each ChildForm As Form In Me.MdiChildren

           ChildForm.Close()

       Next

   End Sub

   Private m\_ChildFormNumber As Integer

   Private Sub tsmi\_medicamentos\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsmi\_medicamentos.Click

       Medicamentos.Show()

   End Sub

   Private Sub tsmi\_proveedores\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsmi\_proveedores.Click

       Proveedores.Show()

   End Sub

   Private Sub tspi\_ventas\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tspi\_ventas.Click

       Ventas.Show()

   End Sub

   Private Sub MenuPrincipal\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       Me.WindowState = FormWindowState.Maximized

   End Sub

   Private Sub tsmi\_CerrarSesion\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsmi\_CerrarSesion.Click

       Form\_Login.Show()

       Me.Close()

   End Sub

   Private Sub tsmi\_Salir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsmi\_Salir.Click

       Application.Exit()

   End Sub

   Private Sub tspi\_nueva\_facturacion\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tspi\_nueva\_facturacion.Click

       Facturacion.Show()

   End Sub

   Private Sub EfectuarRenovaciónToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EfectuarRenovaciónToolStripMenuItem.Click

       If Form\_Login.tipouser = "A" Then

           GestionBD.Show()

       Else

           MessageBox.Show("No cuentas con los permisos suficientes para acceder a esta característica")

       End If

   End Sub

End Class

Se usó una plantilla por defecto de menú que es lo común ver en cualquier aplicación de escritorio, entonces gráficamente se editan los apartados que tienen cada uno y después se vinculan los formularios correspondientes que es este caso esta todos.

**VENTANA MODIFICAR REGISTRO DE UN MEDICAMENTO**

Public Class ModifMedic

   Dim conexion As conexion = New conexion

   Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

       Dim actualizar As String = "IDMed = '" + txtm\_id\_medim.Text + "', Nombre = '" + txtm\_nombre.Text + "', IDProv = '" + txtm\_id\_prov.Text + "', Precio\_publico = '" + txtm\_prec\_pub.Text + "', Stock = '" + txtm\_stock.Text + "', Sustancia = '" + txtm\_sustancia.Text + "', Gramaje = '" + txtm\_gramaje.Text + "', Presentacion = '" + txtm\_presentacion.Text + "'"

       If (conexion.Actualizar("medicamentos", actualizar, "IDMed = " + txtm\_id\_medim.Text)) Then

           MessageBox.Show("DATOS ACTUALIZADOS CORRECTAMENTE")

           Medicamentos.mostrarMedicamentos()

           Me.Close()

       Else

           MessageBox.Show("ERROR AL ACTUALIZAR")

       End If

       Me.Close()

   End Sub

   Private Sub txtm\_id\_medim\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtm\_id\_medim.TextChanged

   End Sub

   Private Sub ModifMedic\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

   End Sub

End Class

Proceso de agregado en la base de datos. Se tendrán que llenar los campos como el ID de medicamentos, nombre, ID de proveedor, precio público, precio del proveedor, cantidad en stock que se va a sumando por default de acuerdo a las cantidades que ingreses, sustancia, gramaje y presentación del medicamento (grageas, jarabe, polvo, etc)

NOTAS:

Es posible eliminar registros por su número de clave

También es posible modificar un registro mediante reingresar todos sus datos nuevamente, utilizando el número de clave.

**VENTANA MODIFICAR REGISTRO DE UN PROVEEDOR**

Imports MySql.Data

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class ModifProveed

   Dim conexion As conexion = New conexion

   Private Sub btn\_modif\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_modif.Click

       Dim actualizar As String = "IDProv = '" + txtm\_clave.Text + "', Nombre = '" + txtm\_nombre.Text + "', Direccion = '" + txtm\_direccion.Text + "', Laboratorio = '" + txtm\_laboratorio.Text + "'"

       If (conexion.Actualizar("proveedores", actualizar, "IDProv = " + txtm\_clave.Text)) Then

           MessageBox.Show("DATOS ACTUALIZADOS CORRECTAMENTE")

           Proveedores.mostrarDatos()

           Me.Close()

       Else

           MessageBox.Show("ERROR AL ACTUALIZAR")

       End If

   End Sub

   Private Sub ModifProveed\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

   End Sub

End Class

**VENTANA PROVEEDORES**

Imports MySql.Data

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class Proveedores

   Public agregprov As Boolean = "False"

   Dim conexion As conexion = New conexion()

   Private Sub Proveedores\_Load(ByVal sender As Object, ByVal e As System.EventArgs)

   End Sub

   Public Sub mostrarDatos()

       conexion.Consulta("select \* from proveedores", "proveedores")

       dgVProveedores.DataSource = conexion.ds.Tables("proveedores")

   End Sub

   Public Sub mostrarConsultaProv(ByVal atributo, ByVal valor)

       conexion.Consulta("select \* from proveedores where " + atributo + "= '" + valor + "'", "proveedores")

       dgVProveedores.DataSource = conexion.ds.Tables("proveedores")

   End Sub

   Private Sub dgVProveedores\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)

       Try

           Dim dgv As Integer = dgVProveedores.CurrentRow.Index

       Catch ex As Exception

       End Try

   End Sub

   Private Sub Label1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_insertar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Public Sub actualizacion()

       mostrarDatos()

   End Sub

   Private Sub btn\_elimproveed\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub txts\_clave\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub lbl\_direccion\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_modif\_prov\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub Label4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_b\_clave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_b\_nombre\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub txtc\_clave\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_b\_direccion\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_b\_laboratorio\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub btn\_mostrar\_todo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

       mostrarDatos()

   End Sub

   Private Sub dgVProveedores\_CellContentClick\_1(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgVProveedores.CellContentClick

   End Sub

   Private Sub Proveedores\_Load\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       MessageBox.Show(conexion.dirIP)

       conexion.Conectar(conexion.dirIP)

       mostrarDatos()

   End Sub

   Private Sub btn\_insertar\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_insertar.Click

       Dim addproveed As String = "insert into proveedores values(" + txt\_id\_prov.Text + ", '" + txt\_nombre.Text + "', '" + txt\_direccion.Text + "', '" + txt\_laboratorio.Text + "'); "

       If (conexion.Insertar(addproveed)) Then

           mostrarDatos()

           MessageBox.Show("DATOS AGREGADOS EXITOSAMENTE")

       Else

           MessageBox.Show("ERROR")

       End If

   End Sub

   Private Sub btn\_elimproveed\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_elimproveed.Click

       If (conexion.Eliminar("proveedores", "IDProv= " + txt\_clave\_prov.Text)) Then

           MessageBox.Show("Datos eliminados exitosamente")

           mostrarDatos()

           If (conexion.ResetID("proveedores")) Then

               MessageBox.Show("ID CAMBIADO EXITOSAMENTE")

           End If

       Else

           MessageBox.Show("ERROR")

       End If

   End Sub

   Private Sub btn\_modif\_prov\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_modif\_prov.Click

       ModifProveed.Show()

   End Sub

   Private Sub btn\_mostrar\_todo\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_mostrar\_todo.Click

   End Sub

   Private Sub btn\_b\_clave\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_b\_clave.Click

       Dim valor As String = txtc\_clave.Text

       mostrarConsultaProv("IDProv", valor)

   End Sub

   Private Sub btn\_b\_nombre\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_b\_nombre.Click

       Dim valor As String = txtc\_nombre.Text

       mostrarConsultaProv("Nombre", valor)

   End Sub

   Private Sub btn\_b\_direccion\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_b\_direccion.Click

       Dim valor As String = txtc\_direccion.Text

       mostrarConsultaProv("Direccion", valor)

   End Sub

   Private Sub btn\_b\_laboratorio\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_b\_laboratorio.Click

       Dim valor As String = txtc\_laboratorio.Text

       mostrarConsultaProv("Laboratorio", valor)

   End Sub

End Class

**VENTANA VENTAS**

El sistema de Ventas, funciona agregando el dato del ID del medicamento. Y la cantidad de medicamentos de esa ID de medicamento (así si se llevan 3 paracetamol, en el rubro del ID se debería ingresar 00000001, y en la cantidad 3). En caso de no llenar el rubro de Cantidad, el sistema asumirá que sólo se trata de 1 medicamento de esa ID.

Public Class Ventas

   Dim conexion As conexion = New conexion()

   Dim hora, fecha As String

   Dim nombreMed, IDMed As String

   Dim precio As Integer

   Dim u As Integer = 0

   Dim m As Integer

   Dim fila, ult, ind As Integer

   Dim sum\_monto, cambio, pago As Decimal

   Dim vboton1 As Boolean = False

   Dim vboton2 As Boolean = False

   Private Sub Label2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lbl\_id\_ticket.Click

   End Sub

   Private Sub btn\_fechayhora\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_fechayhora.Click

       vboton1 = True

       hora = TimeOfDay

       hora = Now.ToString("HH:mm:ss")

       fecha = DateString

       fecha = Now.ToString("yyyy/MM/dd")

       lbl\_fecha.Text = fecha

       lbl\_hora.Text = hora

       If vboton1 = True And vboton2 = True Then

           btn\_registrarcompra.Visible = True

       End If

   End Sub

   Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_id\_medic.TextChanged

   End Sub

   Public Sub obtenerMedim()

       For cont = 1 To m

           dgVCompraActual.Rows.Add()

           IDMed = dgVBDMedicamentos.Rows(0).Cells(0).Value.ToString

           nombreMed = dgVBDMedicamentos.Rows(0).Cells(1).Value.ToString

           precio = dgVBDMedicamentos.Rows(0).Cells(3).Value

           dgVCompraActual.Rows(u).Cells(0).Value = u

           dgVCompraActual.Rows(u).Cells(1).Value = IDMed

           dgVCompraActual.Rows(u).Cells(2).Value = nombreMed

           dgVCompraActual.Rows(u).Cells(3).Value = precio

           u += 1

       Next

   End Sub

   Private Sub dgVCompraActual\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgVCompraActual.CellContentClick

   End Sub

   Private Sub btn\_ingresar\_prod\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_ingresar\_prod.Click

       If txt\_cantidad.Text = "" Then

           m = 1

       Else

           Try

               m = Int(txt\_cantidad.Text)

           Catch ex As Exception

               MessageBox.Show("El campo cantidad debe tener de valor UN NÚMERO")

           End Try

       End If

       Try

           obtenerMedim()

       Catch ex As Exception

           MessageBox.Show("LA ID INGRESADA NO EXISTE")

       End Try

   End Sub

   Private Sub btn\_revisar\_prod\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_revisar\_prod.Click

       conexion.Consulta("select \* from medicamentos where IDMed = '" + txt\_id\_medic.Text + "'", "medicamentos")

       dgVBDMedicamentos.DataSource = conexion.ds.Tables("medicamentos")

   End Sub

   Private Sub Label10\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label10.Click

   End Sub

   Private Sub btn\_Seleccionar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Seleccionar.Click

       fila = dgVCompraActual.CurrentRow.Cells(0).Value

       btn\_Eliminar.Visible = True

       btn\_Cancelar.Visible = True

       lbl\_eliminar.Visible = True

       btn\_Seleccionar.Visible = False

   End Sub

   Private Sub btn\_Eliminar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Eliminar.Click

       ult = dgVCompraActual.RowCount - 2

       For cont = fila To ult

           dgVCompraActual.Rows(cont).Cells(0).Value = cont

           dgVCompraActual.Rows(cont).Cells(1).Value = ""

           dgVCompraActual.Rows(cont).Cells(2).Value = ""

           dgVCompraActual.Rows(cont).Cells(3).Value = ""

           dgVCompraActual.Rows(cont).Cells(1).Value = dgVCompraActual.Rows(cont + 1).Cells(1).Value

           dgVCompraActual.Rows(cont).Cells(2).Value = dgVCompraActual.Rows(cont + 1).Cells(2).Value

           dgVCompraActual.Rows(cont).Cells(3).Value = dgVCompraActual.Rows(cont + 1).Cells(3).Value

           If cont = ult Then

               u = cont

           End If

       Next

       ind = dgVCompraActual.Rows(ult).Index

       dgVCompraActual.Rows.Remove(dgVCompraActual.Rows(ind))

       btn\_Cancelar.Visible = False

       btn\_Eliminar.Visible = False

       lbl\_eliminar.Visible = False

       btn\_Seleccionar.Visible = True

       dgVCompraActual.Refresh()

   End Sub

   Private Sub Ventas\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       btn\_registrarcompra.Visible = False

       dgVVentas.Visible = False

       btn\_Eliminar.Visible = False

       btn\_Cancelar.Visible = False

       lbl\_eliminar.Visible = False

   End Sub

   Private Sub lbl\_eliminar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lbl\_eliminar.Click

   End Sub

   Private Sub btn\_Cancelar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Cancelar.Click

       btn\_Cancelar.Visible = False

       btn\_Eliminar.Visible = False

       lbl\_eliminar.Visible = False

       btn\_Seleccionar.Visible = True

   End Sub

   Private Sub btn\_monto\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_monto.Click

       Dim cant As Integer = dgVCompraActual.RowCount - 2

       sum\_monto = 0

       For o = 0 To cant

           sum\_monto += dgVCompraActual.Rows(o).Cells(3).Value

       Next

       Try

           lbl\_monto.Text = sum\_monto.ToString

           pago = txt\_pago.Text

           cambio = pago - sum\_monto

           If cambio < 0 Then

               MessageBox.Show("El valor del pago debe ser IGUAL O MAYOR al monto total")

               Exit Sub

           End If

           lbl\_cambio.Text = cambio

           vboton2 = True

       Catch ex As Exception

           MessageBox.Show("Ingresa el PAGO (dinero recibido del cliente)")

       End Try

       If vboton1 = True And vboton2 = True Then

           btn\_registrarcompra.Visible = True

       End If

   End Sub

   Private Sub btn\_registrarcompra\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_registrarcompra.Click

       Try

           Dim ind, ufila As Integer

           Dim idticket, IDMed, nombreMed, precio, ultfila As String

           conexion.Insertar("INSERT INTO ventas values (NULL, '" + sum\_monto.ToString + "', '" + pago.ToString + "', '" + cambio.ToString + "', '" + lbl\_fecha.Text + "', '" + lbl\_hora.Text + "')")

           conexion.Consulta("select IDTicket from ventas", "ventas")

           dgVVentas.DataSource = conexion.ds.Tables("ventas")

           ultfila = dgVVentas.RowCount - 2

           idticket = dgVVentas.Rows(ultfila).Cells(0).Value.ToString

           Dim cant As Integer = dgVCompraActual.RowCount - 2

           For o = 0 To cant

               IDMed = dgVCompraActual.Rows(o).Cells(1).Value.ToString()

               nombreMed = dgVCompraActual.Rows(o).Cells(2).Value.ToString()

               precio = dgVCompraActual.Rows(o).Cells(3).Value.ToString()

               conexion.Insertar("INSERT INTO prod\_vendidos values (NULL, '" + idticket + "', '" + IDMed + "', '" + nombreMed + "', '" + precio + "')")

           Next

           vboton1 = False

           vboton2 = False

           btn\_registrarcompra.Visible = False

           ufila = dgVCompraActual.RowCount - 2

           For n = 0 To ufila

               ind = dgVCompraActual.Rows(ult).Index

               dgVCompraActual.Rows.Remove(dgVCompraActual.Rows(ind))

           Next

           u = 0

       Catch ex As Exception

           MessageBox.Show("Al menos debe haber un artículo para hacer una compra")

       End Try

   End Sub

   Private Sub btn\_Cambio\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

   End Sub

   Private Sub dgVVentas\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgVVentas.CellContentClick

   End Sub

End Class