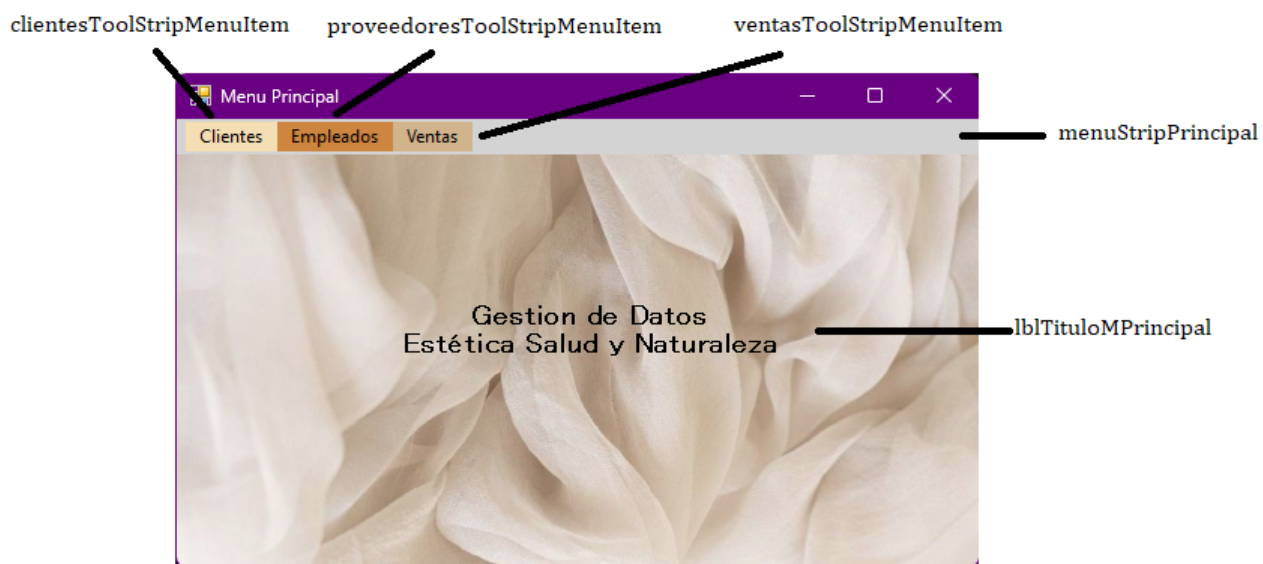


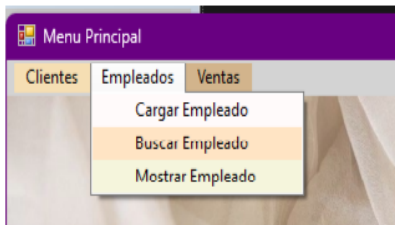
Desarrollo Segundo Proyecto Parcial

Interfaces:

frmMenuPrincipal



registrarClienteToolStripMenuItem
 buscarClienteToolStripMenuItem
 mostrarClienteToolStripMenuItem



registrarEmpleadoToolStripMenuItem
 buscarEmpleadoToolStripMenuItem
 mostrarEmpleadoToolStripMenuItem



ventasRegistradasToolStripMenuItem
 mostrarVentasToolStripMenuItem

frmCargarCliente

Cargar Cliente

Ingrese los datos del cliente a registrar

lblApellidoCliente **Apellido** txtApellidoCli
 lblNombreCliente **Nombre** txtNombreCli
 lblDniCliente **DNI** txtDniCli
 lblEdadCliente **Edad** txtEdadCli
 lblProvCliente **Provincia** cmbProvincias
 lblDepartamentoCliente **Departamento** cmbDepartamentos
 lblDomicilioCliente **Domicilio** txtDomicilioCli

btnCargar
 btnVolver

小红书号: 113056087

frmBuscarCliente

Buscar Cliente

Ingrese el DNI de el cliente a buscar

lblDniBuscado **DNI** txtDniBuscado
 btnBuscar

Datos:

lblApellidoEnc **Apellido** lblApellido
 lblNombreEnc **Nombre** lblNombre
 lblEdadEnc **Edad** lblEdad
 lblProvinciaEnc **Provincia** lblProvincia
 lblDepartamentoEnc **Departamento** lblDepartamento
 lblDomicilioEnc **Domicilio** lblDomicilio

btnVolver

frmMostarCliente

The screenshot shows a Windows form titled "Clientes Registrados". It features a table with the following columns: DNI, Apellido, Nombre, Provincia, Departamento, and Domicilio. The first row of the table is highlighted in blue. Below the table, there is a button labeled "VOLVER ATRAS".

Labels pointing to the form elements:

- lblTituloCR: Points to the title "Clientes Registrados".
- dtvClientes: Points to the data table.
- btnVolver: Points to the "VOLVER ATRAS" button.

frmCargarEmpleado

The screenshot shows a Windows form titled "Cargar Empleado". It contains a title "Ingrese los datos del Empleado a registrar" and several input fields for employee data: Apellido, Nombre, DNI, Edad, and Domicilio. At the bottom, there are two buttons: "CARGAR" and "VOLVER ATRAS".

Labels pointing to the form elements:

- lblTituloBE: Points to the title "Ingrese los datos del Empleado a registrar".
- lblApellidoEmp: Points to the "Apellido" label.
- txtApellidoEmp: Points to the text input field for the last name.
- lblNombreEmp: Points to the "Nombre" label.
- txtNombreEmp: Points to the text input field for the first name.
- lblDniEmp: Points to the "DNI" label.
- txtDniEmp: Points to the text input field for the DNI.
- lblEdadEmp: Points to the "Edad" label.
- txtEdadEmp: Points to the text input field for the age.
- lblDomicilioEmp: Points to the "Domicilio" label.
- txtDomicilioEmp: Points to the text input field for the address.
- btnCargar: Points to the "CARGAR" button.
- btnVolver: Points to the "VOLVER ATRAS" button.

frmBuscarEmpleado

Buscar Empleado

Ingrese el DNI de el Empleado a buscar

DNI **BUSCAR**

Datos:

Apellido **Nombre** **Edad** **Domicilio**

VOLVER ATRAS

frmMostrarEmpleado

Empleados Registrados

	DNI	Apellido	Nombre	Edad	Domicilio
▶	0			0	
	0			0	
	0			0	
*					

VOLVER ATRAS

frmCargarVenta

Cargar Ventas

Carga de Ventas Realizadas

lblCodVenta Código de Venta txtCodVenta

lblDatos **Datos del Cliente y Vendedor**

lblDniCli txtDniClienteB DNI del Cliente DNI del Empleado lblDniEmp txtDniEmpB

lblApCli lblApCliEnc Apellido del Cliente Apellido del Empleado lblApEmp lblApEmpEnc

lblNomCli lblNombCliEnc Nombre del Cliente Nombre del Empleado lblNomEmp lblNombEmpEnc

lblDatosSO **Datos del Servicio Ofrecido**

mrcServicios Servicio Seleccionado

chkManicura chkMasajes ☐ Manicura ☐ Masaje

chkPedicura chkMascarillas ☐ Pedicura ☐ Mascarilla

chkPestañas chkDepilacion ☐ Pestañas ☐ Depilacion

Método de Pago

☐ Débito ☐ Crédito

☐ Efectivo ☐ Transferencia

mrcMetodosDePago

optDebito optCredito

optEfvo optTransf

lblPagos

lblCombos

lblMP Monto a Abonar

lblMontoAbonar \$

btnLimpiar

CALCULAR

btnCalcular

lblHabilitado

btnFinalizar

btnVolver VOLVER ATRAS

frmMostrarVentas

Registro de Ventas

lblTituloRV **Últimas Ventas Registradas**

dtvVentas

	Empleado	Cliente	Cód. Venta	Monto Venta
▶				0
				0
				0
*				

VOLVER ATRAS

btnVolver

Pseudocódigo:

Inicio Programa pryGestionDeDatos

//VARIABLES GLOBALES

//frmCargarCliente

```
variable publica estatica tipo String[] VNombres = new String[3];
variable publica estatica tipo String[] VApellidos = new String[3];
variable publica estatica tipo Int32[] VEdad = new Int32[3];
variable publica estatica tipo Int32[] VDni = new Int32[3];
variable publica estatica tipo String[] VProvincias = new String[3];
variable publica estatica tipo String[] VDepartamentos = new String[3];
variable publica estatica tipo String[] VDomicilio = new String[3];
variable publica estatica tipo Int32 i = 0;
```

//frmBuscarCliente

```
variable publica estatica tipo Int32 i = 0;
```

//frmCargarEmpleado

```
variable publica estatica tipo String[] VNombresEmp = new String[3];
variable publica estatica tipo String[] VApellidosEmp = new String[3];
variable publica estatica tipo Int32[] VDniEmp = new Int32[3];
variable publica estatica tipo Int32[] VEdadEmp = new Int32[3];
variable publica estatica tipo String[] VDomicilioEmp = new String[3];
variable publica estatica tipo Int32 i = 0;
```

//frmBuscarCliente

```
variable publica estatica tipo Int32 i = 0;
```

//frmCargarVentas

```
variable publica estatica tipo String[] VNombresCliVentas = new String[3];
variable publica estatica tipo String[] VApellidosCliVentas = new String[3];
variable publica estatica tipo String[] VNombresEmpVentas = new String[3];
variable publica estatica tipo String[] VApellidosEmpVentas = new String[3];
variable publica estatica tipo Int32[] VMontos = new Int32[3];
variable publica estatica tipo String[] VCodVenta = new String[3];
variable publica estatica tipo Int32 i = 0;
```

//INTERFACES

Inicio Interfaz frmMenuPrincipal

 lblTituloMPrincipal, "Gestion de Datos Estética Salud y Naturaleza"

 Menu Strip Menu Principal

 Tool Strip Menu Clientes

 Tool Strip Item Cargar Cliente

 Tool Strip Item Buscar Cliente

 Tool Strip Item Mostrar Cliente

 Tool Strip Menu Empleados

 Tool Strip Item Cargar Empleado

Tool Strip Item Buscar Empleado
 Tool Strip Item Mostrar Empleado
 Tool Strip Menu Ventas
 Tool Strip Item Cargar Ventas
 Tool Strip Item Mostrar Ventas

Fin interfaz

Inicio interfaz frmCargarCliente

lblTituloRegCliente, "Ingrese los datos del cliente a registrar".
 lblApellidoCliente, "Apellido".
 txtApellidoCli.
 lblNombreCli, "Nombre"
 txtNombreCliente
 lblDniCliente, "DNI"
 txtDniCliente
 lblEdadCliente, "Edad"
 txtEdadCliente
 lblProvCliente, "Provincia"
 cmbProvincias
 lblDepartamentoCliente, "Departamento"
 cmbDepartamentos
 lblDomicilioCliente, "Domicilio"
 txtDomicilioCli
 btnCargar, "CARGAR"
 btnVolver, "VOLVER ATRAS"

Fin Interfaz

Inicio Interfaz frmBuscarCliente

lblTituloBC, "Ingrese el DNI de el cliente a buscar"
 lblDniBuscado, "DNI"
 txtDniBuscado
 btnBuscar, "BUSCAR"
 lblDatosEnc, "Datos"
 lblApellidoEnc, "Apellido"
 lblApellido
 lblNombreEnc, "Nombre"
 lblNombre
 lblEdadEnc, "Edad"
 lblEdad
 lblProvinciaEnc, "Provincia"
 lblProvincia
 lblDepartamentoEnc, "Departamento"
 lblDepartamento
 lblDomicilioEnc, "Domicilio"
 lblDomicilio
 btnVolver, "VOLVER ATRAS"

Fin Interfaz

Inicio Interfaz frmMostrarCliente

lblTituloCR, "Clientes Registrados"
 dtvClientes: "DNI, Apellido, Nombre, Provincia, Departamento, Domicilio"
 btnVolver, "VOLVER ATRAS"

Fin Interfaz

Inicio Interfa frmCargarEmpleado

lblTituloBE, "Ingrese los datos del Empleado a Registrar"
 lblApellidoEmp, "Apellido"
 txtApellidoEmp
 lblNombreEmp, "Nombre"
 txtNombreEmp
 lblDniEmp, "DNI"
 txtDniEmp
 lblEdadEmp, "Edad"
 txtEdadEmp
 lblDomicilioEmp, "Domicilio"
 txtDomicilioEmp
 btnCargar, "CARGAR"
 btnVolver, "VOLVER ATRAS"

Fin Interfaz

Inicio Interfaz frmBuscarEmpleado

lblTituloBE, "Ingrese el DNI de el Empleado a buscar"
 lblDniBusc, "DNI"
 txtDniBuscadoEmp
 btnBuscar, "BUSCAR"
 lblDatosEnc, "Datos:"
 lblApellidoEnc, "Apellido"
 lblApellidoEmp
 lblNombreEnc, "Nombre"
 lblNombreEmp
 lblEdadEnc, "Edad"
 lblEdadEmp
 lblDomicilioEnc, "Domicilio"
 lblDomicilioEmp
 btnVolver, "VOLVER ATRAS"

Fin Interfaz

Inicio Interfaz frmMostrarEmpleado

lblTituloME, "Empleados Registrados"
 dtvEmpleados, "DNI, Apellido, Nombre, Edad, Domicilio"
 btnVolver, "VOLVER ATRAS"

Fin Interfaz

Inicio Interfaz frmCargarVentas

lblTituloVR, "Carga de Ventas Realizadas"
 lblCodVenta, "Código de Venta"
 txtCodVenta
 lblDatos, "Datos del Cliente y Vendedor"
 lblDniCli, "DNI del Cliente"
 txtDniClienteB
 lblDniEmp, "DNI del Empleado"
 txtDniEmpB
 lblApCli, "Apellido del Cliente"
 lblApCLiEnc
 lblApEmp, "Apellido del Empleado"
 lblApEmpEnc


```

lblNomCli, "Nombre del Cliente"
lblNombCliEnc
lblNomEmp, "Nombre del Empleado"
lblNombEmpEnc
lblDatosSO, "Datos del Servicio Ofrecido"
Inicio marco mrcServicios, "Servicio Seleccionado"
    chkManicura, "Manicura"
    chkMasajes, "Masaje"
    chkPedicura, "Pedicura"
    chkMascarillas, "Mascarilla"
    chkPestañas, "Pestañas"
    chkDepilacion, "Depilacion"
fin Marco
Inicio Marco mrcMetodosDePago
    optDebito, "Debito"
    optCrédito, "Crédito"
    optEfvo, "Efectivo"
    optTransf, "Transferencia"
Fin Marco
lblCombos
lblPagos
lblIMP, "Monto a Abonar"
lblMontoAbonar, "$"
btnLimpiar, "LIMPIAR"
btnCalcular, "CALCULAR"
lblHabilitado
btnVolver, "VOLVER ATRAS"
btnFinalizar, "FINALIZAR"

```

Fin Interfaz

```

Inicio Interfaz frmMostrarVentas
    lblTituloRV, "Últimas Ventas Realizadas"
    dtvVentas, "Empleado, Cliente, Cód.Venta, Monto Venta"
    btnVolver, "VOLVER ATRAS"

```

Fin Interfaz

// PROCEDIMIENTOS

//frmMenuPrincipal

```

Inicio Procedimiento ToolStripItemCargarCliente
    this.Hide();
    frmCargarCliente frmCargarCliente = new frmCargarCliente();
    frmCargarCliente.Show();
Fin Procedimiento

```

```

Inicio Procedimiento ToolStripItemBuscarCliente
    this.Hide();
    frmBuscarCliente frmBuscarCliente = new frmBuscarCliente();
    frmBuscarCliente.Show();

```

Fin Procedimiento

Inicio Procedimiento ToolStripItemMostrarCliente

```
this.Hide();
frmMostrarCliente frmMostrarCliente = new frmMostrarCliente();
frmMostrarCliente.Show();
```

Fin Procedimiento

Inicio Procedimiento ToolStripItemCargarEmpleado

```
this.Hide();
frmCargarEmpleado frmCargarEmpleado = new frmCargarEmpleado();
frmCargarEmpleado.Show();
```

Fin Procedimiento

Inicio Procedimiento ToolStripItemBuscarEmpleado

```
this.Hide();
frmBuscarEmpleado frmBuscarEmpleado = new frmBuscarEmpleado();
frmBuscarEmpleado.Show();
```

Fin Procedimiento

Inicio Procedimiento ToolStripItemMostrarEmpleado

```
this.Hide();
frmMostrarEmpleado frmMostrarEmpleado = new frmMostrarEmpleado();
frmMostrarEmpleado.Show();
```

Fin Procedimiento

Inicio Procedimiento ToolStripItemCargarVentas

```
this.Hide();
frmCargarVentas frmCargarVentas = new frmCargarVentas();
frmCargarVentas.Show();
```

Fin Procedimiento

Inicio Procedimiento ToolStripItemMostrarVentas

```
this.Hide();
frmMostrarVentas frmMostrarVentas = new frmMostrarVentas();
frmMostrarVentas.Show();
```

Fin Procedimiento

//frmCargarCliente

Inicio Procedimiento Load(object sender, EventArgs e)

```
CargarNombresProvincias();
```

fin procedimiento

//Uso de API

Inicio Procedimiento de Tarea asíncrona pública CargarNombresProvincias()

```
HttpClient client = new HttpClient();
```

```
try
```

```
    HttpResponseMessage response = await
```

```
client.GetAsync("https://infra.datos.gob.ar/catalog/modernizacion/dataset/7/distribution/7.2/download/provincias.json");
```

```

        si (response.IsSuccessStatusCode)

            string json = await response.Content.ReadAsStringAsync();
            dynamic data =
Newtonsoft.Json.JsonConvert.DeserializeObject(json);
            dynamic provincias = data.provincias;
            cmbProvincias.Items.Clear();

            foreach (var provincia in provincias)

                string nombre = provincia.nombre;
                cmbProvincias.Items.Add(nombre);

        si no

            MessageBox.Show("La solicitud a la API Provincias ha fallado.");
        fin si

    catch (Exception ex)

        MessageBox.Show("Se ha producido un error: " + ex.Message);

    finally

        client.Dispose();

```

Fin Procedimiento

Inicio Procedimiento de Tarea asíncrona pública CargarDepartamentos()

```

        si (cmbProvincias.Items.Count > 0 && cmbProvincias.SelectedIndex != -1) entonces
            string provinciaSeleccionada = cmbProvincias.SelectedItem.ToString();
            HttpClient client = new HttpClient();
            try
                HttpResponseMessage response = await
client.GetAsync("https://infra.datos.gob.ar/catalog/modernizacion/dataset/7/distribution/7.3/download/departamentos.json");

                si (response.IsSuccessStatusCode) entonces
                    string json = await response.Content.ReadAsStringAsync();
                    dynamic data = JsonConvert.DeserializeObject(json);
                    dynamic departamentos = data.departamentos;
                    cmbDepartamentos.Items.Clear();
                Fin si
                foreach (var departamento in departamentos)

                    string nombre = departamento.nombre;
                    string provincia = departamento.provincia.nombre;

```

```

        si (provincia.Equals(provinciaSeleccionada,
StringComparison.OrdinalIgnoreCase)) entonces

            cmbDepartamentos.Items.Add(nombre);
            si no

                MessageBox.Show("La solicitud a la API Departamentos ha
fallado.");
            fin si

            catch (Exception ex)

                MessageBox.Show("Se ha producido un error: " + ex.Message);
            finally

                client.Dispose();
            fin si

Fin Procedimiento

Inicio Procedimiento de Tarea asíncrona pública cmbProvincias_SelectedIndexChanged

await CargarDepartamentos();

Fin Procedimiento

//Botones
Inicio Procedimiento bntCargar == "Click"
    si (i < VNombres.Length) entonces

        VNombres[i] = txtNombreCli.Text;
        VApellidos[i] = txtApellidoCli.Text;
        VEdad[i] = Convert.ToInt32(txtEdadCli.Text);
        VDni[i] = Convert.ToInt32(txtDniCli.Text);
        VProvincias[i] = Convert.ToString(cmbProvincias.SelectedItem);
        VDepartamentos[i] = Convert.ToString(cmbDepartamentos.SelectedItem);
        VDomicilio[i] = txtDomicilioCli.Text;
        MessageBox.Show("DATO GUARDADO");
        txtNombreCli.Clear();
        txtApellidoCli.Clear();
        txtEdadCli.Clear();
        txtDniCli.Clear();
        txtDomicilioCli.Clear();
        cmbProvincias.Text = "";
        cmbDepartamentos.Text = "";
        i++;
    si no

        MessageBox.Show("NO ENTRAN MAS DATOS");
        txtNombreCli.Clear();
        txtApellidoCli.Clear();
        txtEdadCli.Clear();
        txtDniCli.Clear();
        txtDomicilioCli.Clear();
        cmbProvincias.Text = "";

```

```

        cmbDepartamentos.Text = "";
    Fin si

    CargarNombresProvincias();
    CargarDepartamentos();
Fin Procedimiento

Inicio Procedimiento btnVolver == "Click"
    this.Close();
    frmMenuPrincipal principal = new frmMenuPrincipal();
    principal.Show();
Fin Procedimiento

//frmBuscarCliente

Inicio Procedimiento btnBuscar == "Click"

    variable de tipo Int32 datoBuscado = Convert.ToInt32(txtDniBuscado.Text);
    variable de tipo booleano datoEncontrado = false;

    para (int i = 0; i < frmCargarCliente.VDni.Length; i++) entonces

        si (datoBuscado == frmCargarCliente.VDni[i]) entonces

            lblApellido.Text = frmCargarCliente.VApellidos[i];
            lblNombre.Text = frmCargarCliente.VNombres[i];
            lblEdad.Text = Convert.ToString(frmCargarCliente.VEdad[i]);
            lblDomicilio.Text = frmCargarCliente.VDomicilio[i];
            lblProvincia.Text = Convert.ToString(frmCargarCliente.VProvincias[i]);
            lblDepartamento.Text = frmCargarCliente.VDepartamentos[i];
            datoEncontrado = true;
            break;
        fin si

    fin Para

    si (!datoEncontrado) entonces
        MessageBox.Show("El DNI ingresado no coincide con los Clientes
Registrados");

        lblNombre.Text = "";
        lblApellido.Text = "";
        lblEdad.Text = "";
        lblDomicilio.Text = "";
        lblProvincia.Text = "";
        lblDepartamento.Text = "";

    Fin si
Fin Procedimiento

Inicio Procedimiento btnVolver == "Click"
    this.Close();
    frmMenuPrincipal principal = new frmMenuPrincipal();
    principal.Show();
Fin Procedimiento

```

```
// frmMostrarCliente
```

```

    Inicio Procedimiento LOAD
        Para (int i = 0; i < frmCargarCliente.VDni.Length; i++) entonces
            dtvClientes.Rows.Add(frmCargarCliente.VDni[i] , frmCargarCliente.VApellidos[i],
frmCargarCliente.VNombres[i] ,
frmCargarCliente.VProvincias[i],frmCargarCliente.VDepartamentos[i],
frmCargarCliente.VDomicilio[i]);
        Fin Para
    Fin Procedimiento

```

```
//frmCargarEmpleado
```

```

    Inicio Procedimiento btnCargar == "Click"

        si (i < VNombresEmp.Length) entonces

            VNombresEmp[i] = txtNombreEmp.Text;
            VApellidosEmp[i] = txtApellidoEmp.Text;
            VEdadEmp[i] = Convert.ToInt32(txtEdadEmp.Text);
            VDniEmp[i] = Convert.ToInt32(txtDniEmp.Text);
            VDomicilioEmp[i] = txtDomicilioEmp.Text;
            MessageBox.Show("DATO GUARDADO");
            txtNombreEmp.Clear();
            txtApellidoEmp.Clear();
            txtEdadEmp.Clear();
            txtDniEmp.Clear();
            txtDomicilioEmp.Clear();
            i++;

        si no
            MessageBox.Show("NO ENTRAN MAS DATOS");
            txtNombreEmp.Clear();
            txtApellidoEmp.Clear();
            txtEdadEmp.Clear();
            txtDniEmp.Clear();
            txtDomicilioEmp.Clear();

        Fin si
    Fin Procedimiento

    Inicio Procedimiento btnVolver == "Click"
        this.Close();
        frmMenuPrincipal principal = new frmMenuPrincipal();
        principal.Show();
    Fin Procedimiento

```

```
//frmBuscarCliente
```

```

    Inicio Procedimiento btnCargar == "Click"

```

```

para (int i = 0; i < frmCargarEmpleado.VDniEmp.Length; i++) entonces

    si (datoBuscado == frmCargarEmpleado.VDniEmp[i])
        lblApellidoEmp.Text = frmCargarEmpleado.VApellidosEmp[i];
        lblNombreEmp.Text = frmCargarEmpleado.VNombresEmp[i];
        lblEdadEmp.Text =
Convert.ToString(frmCargarEmpleado.VEdadEmp[i]);
        lblDomicilioEmp.Text = frmCargarEmpleado.VDomicilioEmp[i];
        datoEncontrado = true;
        break;
    fin si
fin para

si (!datoEncontrado)
    MessageBox.Show("El DNI ingresado no coincide con los Clientes
Registrados");
    lblNombreEmp.Text = "";
    lblApellidoEmp.Text = "";
    lblEdadEmp.Text = "";
    lblDomicilioEmp.Text = "";
Fin si
Fin Procedimiento

Inicio Procedimiento btnVolver == "Click"
    this.Close();
    frmMenuPrincipal principal = new frmMenuPrincipal();
    principal.Show();
Fin Procedimiento

//frmMostrarEmpleado

Inicio Procedimiento LOAD
    para (int i = 0; i < frmCargarEmpleado.VDniEmp.Length; i++) entonces
        dtvEmpleados.Rows.Add(frmCargarEmpleado.VDniEmp[i],
frmCargarEmpleado.VApellidosEmp[i], frmCargarEmpleado.VNombresEmp[i],
frmCargarEmpleado.VEdadEmp[i], frmCargarEmpleado.VDomicilioEmp[i]);
    Fin para
Fin Procedimiento

Inicio Procedimiento btnVolver == "Click"
    this.Close();
    frmMenuPrincipal principal = new frmMenuPrincipal();
    principal.Show();
Fin Procedimiento

//frmCargarVentas
variable tipo Double montoAPagar = 0;

Inicio Procedimiento btnCalcular == "Click"
    variable tipo Double manicura = 2000;
    variable tipo Double pedicura = 2500;
    variable tipo Double masajes = 3000;

```



```
variable tipo Double pestañas = 5000;
variable tipo Double mascarillas = 1500;
variable tipo Double depilacion = 3000;
```

```
Si (chkManicura.Checked) entonces
montoAPagar += manicura;
Fin si
```

```
Si (chkPedicura.Checked) entonces
montoAPagar += pedicura;
Fin si
```

```
Si (chkMasajes.Checked)
montoAPagar += masajes;
Fin si
```

```
Si (chkPestañas.Checked) entonces
    montoAPagar += pestañas;
Fin si
```

```
Si (chkMascarillas.Checked) entonces
    montoAPagar += mascarillas;
Fin si
```

```
Si (chkDepilacion.Checked) entonces
    montoAPagar += depilacion;
Fin si
```

```
// COMBOS
```

```
Si (chkManicura.Checked && chkPedicura.Checked) entonces
```

```
    lblCombos.Text = ("Combo Pink Activado (10% desc)");
    montoAPagar -= montoAPagar * 0.10;
```

```
Si (chkPestañas.Checked && chkMascarillas.Checked)
```

```
    lblCombos.Text = ("Combo GOLD Activado (20% desc)");
    montoAPagar -= montoAPagar * 0.20;
```

```
Fin si
```

```
Si No (chkMasajes.Checked && chkMascarillas.Checked) entonces
```

```
    lblCombos.Text = ("Combo Relax Activado (15% desc)");
    montoAPagar -= montoAPagar * 0.15;
```

```
Si no
```

```
    lblCombos.Text = ("No aplica ningun Combo");
```

```
Fin si
```

```
//MEDIOS DE PAGO
```

```
Según sea (true)
```

```

cuando sea var __ when optDebito.Checked:
    montoAPagar += montoAPagar * 0.10;
    lblPagos.Text = "Con este medio tiene un 10% de recargo";
    break;

cuando sea var __ when optCredito.Checked:
    montoAPagar += montoAPagar * 0.20;
    lblPagos.Text = "Con este medio tiene un 20% de recargo";
    break;

cuando sea var __ when optEfvo.Checked:
    montoAPagar -= montoAPagar * 0.10;
    lblPagos.Text = "Con este medio tiene un 10% de descuento";
    break;

cuando sea var __ when optTransf.Checked:
    montoAPagar += montoAPagar * 0.10;
    lblPagos.Text = "Con este medio tiene un 10% de recargo";
    break;

default:
    lblPagos.Text = "Ningún método de pago seleccionado";
    break;

```

```
Fin Según Sea
```

```

lblMontoAbonar.Text = "$" + montoAPagar;
btnCalcular.Enabled = false;
btnLimpiar.Enabled = true;

```

```

Int32 datoBuscadoCli = Convert.ToInt32(txtDniClienteB.Text);
Int32 datoBuscadoEmp = Convert.ToInt32(txtDniEmpB.Text);

```

```

variable tipo booleano datoEncontradoCli = false;
variable tipo booleano datoEncontradoEmp = false;

```

```
Para (int i = 0; i < VMontos.Length; i++) entonces
```

```
Si (datoBuscadoCli == frmCargarCliente.VDni[i])
```

```

    lblApCliEnc.Text = frmCargarCliente.VApellidos[i];
    lblNombCliEnc.Text = frmCargarCliente.VNombres[i];
    datoEncontradoCli = true;

```

```
Fin si
```

```
Si (datoBuscadoEmp == frmCargarEmpleado.VDniEmp[i])
```

```

    lblApEmpEnc.Text = frmCargarEmpleado.VApellidosEmp[i];
    lblNombEmpEnc.Text = frmCargarEmpleado.VNombresEmp[i];
    datoEncontradoEmp = true;

```

Fin si

Fin Para

Si (datoEncontradoEmp && datoEncontradoCli)

Si ((chkManicura.Checked || chkDepilacion.Checked || chkMasajes.Checked ||
chkPestañas.Checked || chkPedicura.Checked || chkMascarillas.Checked) && (optCredito.Checked ||
optDebito.Checked || optEfvo.Checked || optTransf.Checked)) entonces
 btnFinalizar.Enabled = true;
 lblHabilitado.Text = "Operacion Habilitada";

Si No

 MessageBox.Show("Debe seleccionar Servicio y Metodo de Pago");

Fin Si

Si (!datoEncontradoEmp || !datoEncontradoCli)

 lblHabilitado.Text = "El empleado y cliente deben estar registrados
previamente";

Fin Si

Fin Procedimiento

Inicio Procedimiento btnLimpiar == "Click"

 btnCalcular.Enabled = true;
 montoAPagar = 0;
 lblMontoAbonar.Text = "";
 txtDniClienteB.Clear();
 txtDniEmpB.Clear();
 lblNombEmpEnc.Text = "";
 lblApEmpEnc.Text = "";
 lblApCliEnc.Text = "";
 lblNombCliEnc.Text = "";
 txtCodVenta.Clear();
 lblMontoAbonar.Text = "";
 txtCodVenta.Clear();
 chkDepilacion.Checked = false;
 chkManicura.Checked = false;
 chkMasajes.Checked = false;
 chkPedicura.Checked = false;
 chkPestañas.Checked = false;
 chkMascarillas.Checked = false;
 optCredito.Checked = false;
 optDebito.Checked = false;
 optEfvo.Checked = false;
 optTransf.Checked = false;
 btnLimpiar.Enabled = false;
 btnFinalizar.Enabled = false;

Fin Procedimiento

Inicio Procedimiento LOAD

Para (int i = 0; i < VMontos.Length; i++) entonces

```
VNombresEmpVentas[i] = frmCargarEmpleado.VNombresEmp[i];
VApellidosEmpVentas[i] = frmCargarEmpleado.VApellidosEmp[i];
VNombresCliVentas[i] = frmCargarCliente.VNombres[i];
VApellidosCliVentas[i] = frmCargarCliente.VApellidos[i];
```

Fin Para

Fin Procedimiento

Inicio Procedimiento btnVolver == "Click"

```
this.Close();
frmMenuPrincipal principal = new frmMenuPrincipal();
principal.Show();
```

Fin Procedimiento

Inicio Procedimiento btnFinalizar == "Click"

```
Si (i < VCodVenta.Length) entonces
VMontos[i] = Convert.ToInt32(montoAPagar);
VCodVenta[i] = txtCodVenta.Text;
VNombresCliVentas[i] = lblNombCliEnc.Text;
VApellidosCliVentas[i] = lblApCliEnc.Text;
VNombresEmpVentas[i] = lblNombEmpEnc.Text;
VApellidosEmpVentas[i] = lblApEmpEnc.Text;
```

```
lblNombEmpEnc.Text = "";
lblApEmpEnc.Text = "";
lblApCliEnc.Text = "";
lblNombCliEnc.Text = "";
txtDniClienteB.Clear();
txtDniEmpB.Clear();
montoAPagar = 0;
lblMontoAbonar.Text = "";
txtCodVenta.Clear();
MessageBox.Show("DATOS CARGADOS");
i++;
chkDepilacion.Checked = false;
chkManicura.Checked = false;
chkMasajes.Checked = false;
chkPedicura.Checked = false;
chkPestañas.Checked = false;
chkMascarillas.Checked = false;
optCredito.Checked = false;
optDebito.Checked = false;
optEfvo.Checked = false;
optTransf.Checked = false;
```

```
btnFinalizar.Enabled = false;
```

```
Si no
```

```
txtDniClienteB.Clear();
txtDniEmpB.Clear();
txtCodVenta.Clear();
lblMontoAbonar.Text = "";
txtCodVenta.Clear();
MessageBox.Show("NO ENTRAN MAS DATOS");
Fin si
```

```
Fin Procedimiento
```

```
//frmMostrarVentas
```

```
Inicio Procedimiento LOAD
```

```
Para (int i = 0; i < frmCargarVentas.VCodVenta.Length; i++) entonces
    dtvVentas.Rows.Add(frmCargarVentas.VApellidosCliVentas[i]+ " "+
        frmCargarVentas.VNombresCliVentas[i] ,frmCargarVentas.VApellidosEmpVentas[i]+ " " +
        frmCargarVentas.VNombresEmpVentas[i] , frmCargarVentas.VCodVenta[i],
        frmCargarVentas.VMontos[i]);
```

```
Fin Para
```

```
Fin Procedimiento
```

```
Inicio Procedimiento btnVolver="Click"
```

```
this.Close();
frmMenuPrincipal principal = new frmMenuPrincipal();
principal.Show();
```

```
Fin Procedimiento
```

```
Fin Programa pryGestiondeDatos
```