

```
using BussinessLayer;
using Entities;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Gym
{
    public partial class Empleados : Form
    {
        #region Instancias
        //entidades
        private Entities.Empleados _empleados;
        private readonly Personas _personas;
        //realizar el registro de logueos
        private Registros_Logs _registroLogs;

        //clases internas
        private Restricciones _restricciones;
        private readonly MetodosGenerales _metodosGenerales;

        //Negocio
        private readonly BussinessEmpleados _bussinessEmpleados;
        private readonly BussinessPersonas _bussinessPersonas;
        private readonly BussinessJornadas _bussinessJornadas;

        #endregion

        #region variables
        private int personaLogueada;
        private bool camposObligatoriosVacios = true;
        private bool contenidoErroneo = false;
        private string clave = string.Empty;
        //Verificar coincidencias
        private bool personaRegistrada;
        private bool claveOK;
        private DataSet dsTablaEmpleados;
        private string buscar;
        private bool edicionEmpleado;
        private int motivoEdicion = 3;
        private bool chkTodos;
        private int persona_Jor;
        private bool cargarJornada;
        #endregion

        #region Load del form
        public Empleados(int idPersonaLog)
        {
            InitializeComponent();

            _restricciones = new Restricciones();
            _metodosGenerales = new MetodosGenerales();
        }
    }
}
```

```
_bussinessEmpleados = new BussinessEmpleados();
_bussinessPersonas = new BussinessPersonas();
_bussinessJornadas = new BussinessJornadas();

personaLogueada = idPersonaLog;

_empleados = new Entities.Empleados();
_personas = new Personas();
}
private void Empleados_Load(object sender, EventArgs e)
{
    Tipos_Documentos();
    Tipos_Sexos();
    Tipos_Empleados();
    Estados_Empleados();
    GetEmpleados();
}

#endregion

#region Métodos Encapsulados

private void CargarJornada()
{
    //Hacemos la solicitud para cargar el form de jornadas
    //enviándole el dato correspondiente para que cargue
    //las jornadas del empleado si es que hay, o directamente
    //haga la gestión que el usuario desee.
    bool empleado = true;
    bool darBaja;
    if (Convert.ToInt32(cmbEstados.SelectedValue) == 3)
    {
        darBaja = true;
    }
    else
    {
        darBaja = false;
    }
    Jornadas jn = new Jornadas(persona_Jor, empleado, darBaja);
    jn.ShowDialog();
}

private void BuscarEmpleado()
{
    //Asigno el ID de la fila, que está oculto
    //yield lo paso como argumento para buscar los datos de ese id en la bdd
    _personas.Persona_ID = Convert.ToInt32(dtgEmpleados.CurrentRow.Cells[0].Value);
    _bussinessPersonas.GetPersonaUnica(_personas);

    _metodosGenerales.personaID = _personas.Persona_ID;
    _empleados.Persona_ID = _metodosGenerales.personaID;
    _bussinessEmpleados.GetTipoEmpleado(_empleados);
    _metodosGenerales.empleadoID = _empleados.Empleado_ID;

    //Luego de traer todos los datos, se los asigno a cada textbox

    //Datos de Tabla Personas
    txtNombreEmpleado.Text = _personas.Nombre;
    txtApellidoEmpleado.Text = _personas.Apellido;
    cmbTipoDocumentoEmpleado.SelectedIndex = _personas.Tipo_Documento_ID;
    txtDocumentoEmpleado.Text = _personas.Nro_documento;
    cmbSexoEmpleado.SelectedIndex = _personas.Tipo_Sexo_ID;
```

```

txtTelefonoEmpleado.Text = _personas.Nro_Telefono;
txtAlternativoEmpleado.Text = _personas.Nro_Alternativo;
txtMailEmpleado.Text = _personas.Mail;
txtObservacionesEmpleado.Text = _personas.Observaciones;

//Datos de tabla Empleados
Tipos_Empleados();
cmbTipoEmpleado.SelectedIndex = _empleados.Tipo_Empleado_ID - 2;//xq los 2
primeros, están fuera de la búsqueda
txtUsuario.Text = _empleados.Usuario;
txtClave.Text = _empleados.Clave;
cmbEstados.SelectedIndex = _empleados.Estado_Empleado_ID;

//Le voy a cambiar el color al texto de los textbox tmb, sino se ve todo en
DimGray
foreach (Control cnt in this.gbEmpleado.Controls)
{
    if (cnt is TextBox box)
    {
        TextBox t;
        t = box;
        if (t.Text != "Usuario" && t.Text != "Clave")
        {
            t.ForeColor = Color.Black;
        }
    }
}
if (cmbTipoEmpleado.SelectedIndex != 0)
{
    txtUsuario.Enabled = false;
    txtClave.Enabled = false;
}
edicionEmpleado = true;
}
private void GetEmpleados()
{
    //Primero limpio el grid. Para que no haya errores.
    //Sería como si hiciera un refresh.
    //Saco lo que haya y luego vuelvo a traer los datos
    dtgvEmpleados.Rows.Clear();

    dsTablaEmpleados = _bussinessEmpleados.GetEmpleados(buscar);
    buscar = string.Empty;

    if (dsTablaEmpleados.Tables[0].Rows.Count > 0)
    {
        foreach (DataRow dr in dsTablaEmpleados.Tables[0].Rows)
        {
            dtgvEmpleados.Rows.Add(dr[0].ToString(), dr[1], dr[2], dr[3], dr[4],
dr[5]);
        }
    }
}
private void ABMEmpleado()
{
    switch (motivoEdicion)
    {
        case 0:
            //Edición de todos los datos del empleado
            //En caso de que no sea un alta, sino modificación,
            //se va a realizar desde acá
            EditarPersona();

```

```

        EditarEmpleado();
        persona_Jor = Convert.ToInt32(dtgvEmpleados.CurrentRow.Cells[0].Value);

        break;
    case 1:
        //Edición solo de la jornada
        persona_Jor = Convert.ToInt32(dtgvEmpleados.CurrentRow.Cells[0].Value);

        break;
    case 2:
        //Blanqueo solo de usuario y clave.

        break;
    default:
        //Alta empleado
        AltaPersona();
        AltaEmpleado();
        _bussinessEmpleados.GetLastEmpleadoID(_empleados);
        persona_Jor = _empleados.Empleado_ID;
        break;
    }
    dsTablaEmpleados.Clear();
    GetEmpleados();
    ResetControls();
}

private void EditarPersona()
{
    //Es la edición de los datos personales
    //junto con los datos de empleado, excepto las claves
    int Persona_ID = _personas.Persona_ID;
    string Nombre = Convert.ToString(txtNombreEmpleado.Text);
    string Apellido = Convert.ToString(txtApellidoEmpleado.Text);
    int Tipo_Documento_ID = Convert.ToInt32(cmbTipoDocumentoEmpleado.SelectedValue);
    string Nro_documento = Convert.ToString(txtDocumentoEmpleado.Text);
    int Tipo_Sexo_ID = Convert.ToInt32(cmbSexoEmpleado.SelectedValue);
    string Nro_Telefono = Convert.ToString(txtTelefonoEmpleado.Text);
    string Nro_Alternativo, Mail, Observaciones;

    //Envío todo el objeto creado a la cada de negocios para ser enviado
    //a la bdd.

    //Acá verifico si el usuario ingresó datos nuevos o no a los campos
    //que reciben nulos.
    //Si no agregaron datos, los mandamos como nulos
    //caso contrario, se envía lo que haya en su correspondiente Textbox
    if (txtAlternativoEmpleado.Text != "Alternativo")
    {
        Nro_Alternativo = Convert.ToString(txtAlternativoEmpleado.Text);
    }
    else
    {
        Nro_Alternativo = string.Empty;
    }
    if (txtMailEmpleado.Text != "Mail")
    {
        Mail = Convert.ToString(txtMailEmpleado.Text);
    }
    else
    {

```

```
        Mail = string.Empty;
    }
    if (txtObservacionesEmpleado.Text != "Observaciones y/o consideraciones")
    {
        Observaciones = Convert.ToString(txtObservacionesEmpleado.Text);
    }
    else
    {
        Observaciones = string.Empty;
    }
    _metodosGenerales.EditarPersona(Persona_ID, Nombre,
                                    Apellido,
                                    Tipo_Documento_ID,
                                    Nro_documento,
                                    Tipo_Sexo_ID,
                                    Nro_Telefono,
                                    Nro_Alternativo,
                                    Mail,
                                    Observaciones);
}

private void AltaPersona()
{
    string Nombre = Convert.ToString(txtNombreEmpleado.Text);
    string Apellido = Convert.ToString(txtApellidoEmpleado.Text);
    int Tipo_Documento_ID = Convert.ToInt32(cmbTipoDocumentoEmpleado.SelectedValue);
    string Nro_documento = Convert.ToString(txtDocumentoEmpleado.Text);
    int Tipo_Sexo_ID = Convert.ToInt32(cmbSexoEmpleado.SelectedValue);
    string Nro_Telefono = Convert.ToString(txtTelefonoEmpleado.Text);
    DateTime FechaAlta = DateTime.Now;
    string Nro_Alternativo, Mail, Observaciones;

    //Envío todo el objeto creado a la cada de negocios para ser enviado
    //a la bdd.
    if (txtAlternativoEmpleado.Text != "Alternativo")
    {
        Nro_Alternativo = Convert.ToString(txtAlternativoEmpleado.Text);
    }
    else
    {
        Nro_Alternativo = string.Empty;
    }

    if (txtMailEmpleado.Text != "Mail")
    {
        Mail = Convert.ToString(txtMailEmpleado.Text);
    }
    else
    {
        Mail = string.Empty;
    }

    if (txtObservacionesEmpleado.Text != "Observaciones y/o consideraciones")
    {
        Observaciones = Convert.ToString(txtObservacionesEmpleado.Text);
    }
    else
    {
        Observaciones = string.Empty;
    }
    _metodosGenerales.AltaPersona(Nombre,
```

```
        Apellido,
        Tipo_Documento_ID,
        Nro_documento,
        Tipo_Sexo_ID,
        Nro_Telefono,
        Nro_Alternativo,
        Mail,
        Observaciones,
        FechaAlta);
    }

    private void EditarEmpleado()
    {
        //Vamos a verificar si se ha modificado el combobox del tipo de empleado
        //cosa de saber si hace falta o no que tenga una clave.
        _empleados.Empleado_ID = _metodosGenerales.empleadoID;
        _empleados.Persona_ID = _metodosGenerales.personaID;
        _empleados.Estado_Empleado_ID = Convert.ToInt32(cmbEstados.SelectedValue);
        if (cmbTipoEmpleado.SelectedIndex == 0)
        {
            _empleados.Tipo_Empleado_ID = Convert.ToInt32(cmbTipoEmpleado.SelectedValue);
            _empleados.Usuario = txtUsuario.Text.ToString();
            _empleados.Clave = clave;
        }
        else
        {
            //Caso contrario, si el perfil no requiere clave, se envían los campos vacíos.
            _empleados.Usuario = string.Empty;
            _empleados.Clave = string.Empty;
        }

        //Mando todo a la capa de negocio
        _bussinessEmpleados.EditarEmpleado(_empleados);
    }

    private void AltaEmpleado()
    {
        _empleados.Persona_ID = _metodosGenerales.persona_ID;
        _empleados.Tipo_Empleado_ID = Convert.ToInt32(cmbTipoEmpleado.SelectedValue);
        _empleados.Estado_Empleado_ID = Convert.ToInt32(cmbEstados.SelectedValue);
        if (cmbTipoEmpleado.SelectedIndex == 0)
        {
            _empleados.Tipo_Empleado_ID = Convert.ToInt32(cmbTipoEmpleado.SelectedValue);
            _empleados.Usuario = txtUsuario.Text.ToString();
            _empleados.Clave = clave;
        }
        else
        {
            //Caso contrario, si el perfil no requiere clave, se envían los campos vacíos.
            _empleados.Usuario = string.Empty;
            _empleados.Clave = string.Empty;
        }
        //Envío toda la data a la capa de negocio para ser mandada a la bdd.
        _bussinessEmpleados.AltaEmpleado(_empleados);
    }

    private void EncriptarClaveClave()
    {
        //Encriptamos la clave con un HASH256
        //Y la asignamos al campo de la entidad correspondiente
        clave = EncriptamientoSHA256.GetSHA256(txtClave.Text.ToString());
    }
}
```

```

private void VerificarClave()
{
    if (cmbTipoEmpleado.SelectedIndex == 0)
    {
        if (txtUsuario.Text != "Usuario" && txtClave.Text != "Clave")
        {
            //Si está todo OK.
            //Vamos a encriptar la clave y consultar si es una edición o alta.
            EncriptarClaveClave();
            //Tambien asigno la clave sin encriptación a otra variable.
            string claveSinCypT = txtClave.Text;

            if (motivoEdicion == 0)
            {
                //Si es una edición, hay que comparar si la clave nueva es != a la de
                if (claveSinCypT == _empleados.Clave)
                {
                    _empleados.Clave = claveSinCypT;
                }
                else
                {
                    _empleados.Clave = clave;
                }
            }
            else
            {
                _empleados.Clave = clave;
            }
            claveOK = true;
        }
        else
        {
            MessageBox.Show("El usuario y/o la clave están mal registrados. Por favor, asigne usuario y clave validos.", "Usuario y/o clave mal registrados");
            claveOK = false;
        }
    }
}

private void Tipos_Documentos()
{
    //llamo al método para traer los tipos de documentos y los asigno al combobox
    _metodosGenerales.Bring_Tipos_Documentos();
    cmbTipoDocumentoEmpleado.DataSource = _metodosGenerales.DtTipos_Documentos;
    cmbTipoDocumentoEmpleado.DisplayMember = "Tipo";
    cmbTipoDocumentoEmpleado.ValueMember = "Tipo_documento_ID";
}

private void Tipos_Sexos()
{
    //llamo al método para traer los tipos de sexos y los asigno al combobox
    _metodosGenerales.Bring_Tipos_Sexos();
    cmbSexoEmpleado.DataSource = _metodosGenerales.DtTipos_Sexos;
    cmbSexoEmpleado.DisplayMember = "Sexo"; //Pero solo esta columna es la que muestro
    cmbSexoEmpleado.ValueMember = "Tipo_Sexo_ID";
}

private void Tipos_Empleados()
{
    //llamo al método para traer los tipos de Empleados y los asigno al combobox
    _metodosGenerales.Bring_Tipos_Empleados();
    cmbTipoEmpleado.DataSource = _metodosGenerales.DtTipos_Empleados;
    cmbTipoEmpleado.DisplayMember = "Tipo"; //Pero solo esta columna es la que muestro

```

```

        cmbTipoEmpleado.ValueMember = "Tipo_Empleado_ID";
    }
    private void Estados_Empleados()
    {
        //llamo al método para traer todos los estados disponibles en la base de datos.
        _metodosGenerales.GetEstadosEmpleados();
        cmbEstados.DataSource = _metodosGenerales.DtEstados_Empleados;
        cmbEstados.Tag = "Estado inicial del empleado";
        cmbEstados.DisplayMember = "Estado_Empleado"; //Pero solo esta columna es la que
muestro
        cmbEstados.ValueMember = "Estado_Empleado_ID";
    }
    private void ValidarCamposVacios()
    {
        if (string.IsNullOrEmpty(txtNombreEmpleado.Text) ||
            string.IsNullOrEmpty(txtApellidoEmpleado.Text) ||
            string.IsNullOrEmpty(txtDocumentoEmpleado.Text) ||
            string.IsNullOrEmpty(txtTelefonoEmpleado.Text))
        {
            camposObligatoriosVacios = true;
        }
        else
        {
            camposObligatoriosVacios = false;
        }
    }
    private void ValidarContenido()
    {
        foreach (Control txt in this.gbEmpleado.Controls)
        {
            if (txt is TextBox)
            {
                if (txt.Text == "Nombre" ||
                    txt.Text == "Apellido" ||
                    txt.Text == "Documento" ||
                    txt.Text == "Teléfono")
                {
                    MessageBox.Show("Existen campos sin completar. Por favor, rellene
todos los campos obligatorios", "Campos obligatorios sin rellenar.", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                    contenidoErroneo = true;
                    break;
                }
                else
                {
                    contenidoErroneo = false;
                }
            }
        }
    }
    private void ResetControls()
    {
        //luego el groupbox general, lo volvemos al estado inicial.
        txtNombreEmpleado.Text = "Nombre";
        txtApellidoEmpleado.Text = "Apellido";
        txtDocumentoEmpleado.Text = "Documento";
        txtTelefonoEmpleado.Text = "Teléfono";
        txtAlternativoEmpleado.Text = "Alternativo";
        txtMailEmpleado.Text = "Mail";
        txtObservacionesEmpleado.Text = "Observaciones y/o consideraciones";
        txtUsuario.Text = "Usuario";
        txtClave.Text = "Clave";
    }

```



```
cmbTipoDocumentoEmpleado.SelectedItem = 0;
cmbSexoEmpleado.SelectedItem = 0;
cmbTipoEmpleado.SelectedItem = 0;
foreach (Control txt in this.gbEmpleado.Controls)
{
    if (txt is TextBox box)
    {
        txt.ForeColor = Color.DimGray;
    }
}

private void HabilitarClave()
{
    if (cmbTipoEmpleado.SelectedIndex == 0)
    {
        txtUsuario.Enabled = true;
        txtClave.Enabled = true;
    }
    else
    {
        txtUsuario.Enabled = false;
        txtClave.Enabled = false;
    }
}

public void FormIncompleto(bool formIncompleto)
{
    foreach (Control cntl in this.gbEmpleado.Controls)
    {
        if (cntl is TextBox box)
        {
            TextBox t;
            t = box;
            if (t.Text == string.Empty || t.ForeColor == Color.DimGray)
            {
                DialogResult result = MessageBox.Show("Hay campos sin completar.
¿Quiere salir de todas formas?",
                "Campos incompletos", MessageBoxButtons.YesNo,
                MessageBoxIcon.Warning);
                if (result == DialogResult.Yes)
                {
                    formIncompleto = false;
                }
                else
                {
                    formIncompleto = true;
                }
                break;
            }
        }
        else
        {
            formIncompleto = false;
        }
    }
}

private void EditarDatos()
{
    BuscarEmpleado();
    HabilitarClave();
}
```

```
}

private void EditarLogin()
{
    _empleados.Usuario = txtUsuario.Text.ToString();
    EncriptarClaveClave();
    _bussinessEmpleados.EditarClave(_empleados);
}

private void VerificarCoincidencias()
{
    if (motivoEdicion != 0)
    {
        int id = _empleados.Persona_ID;
        string documento = txtDocumentoEmpleado.Text.ToString();
        _bussinessPersonas.BuscarCoincidencias(id, documento, _personas);
        if (_personas.Persona_ID == id && _personas.Nro_documento == documento)
        {
            personaRegistrada = true;
        }
        else
        {
            personaRegistrada = false;
        }
    }
}

#endregion

#region Eventos de Foco en Textbox

private void txtNombreEmpleado_Enter(object sender, System.EventArgs e)
{
    //Primero preguntams si el texto dice Nombre
    //Al principio siempre lo va a decir, porque así
    //inicializa el form. Por lo que lo vamos a vaciar
    //y cambiar el color de la letra.
    //En cso de que no diga literalmente Nombre
    //Se va a dejar como está, ya que puede ser
    //un texto que el usuraio escribió

    if (txtNombreEmpleado.Text == "Nombre")
    {
        txtNombreEmpleado.Text = string.Empty;
        txtNombreEmpleado.ForeColor = Color.Black;
    }
}

private void txtNombreEmpleado_Leave(object sender, System.EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtNombreEmpleado.Text))
    {
        txtNombreEmpleado.Text = "Nombre";
        txtNombreEmpleado.ForeColor = Color.DimGray;
    }
}

/*Así igual para los siguientes eventos dentro de esta región*/
private void txtApellidoEmpleado_Enter(object sender, System.EventArgs e)
```

```
{
    if (txtApellidoEmpleado.Text == "Apellido")
    {
        txtApellidoEmpleado.Text = string.Empty;
        txtApellidoEmpleado.ForeColor = Color.Black;
    }
}

private void txtApellidoEmpleado_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtApellidoEmpleado.Text))
    {
        txtApellidoEmpleado.Text = "Apellido";
        txtApellidoEmpleado.ForeColor = Color.DimGray;
    }
}

private void txtDocumentoEmpleado_Enter(object sender, System.EventArgs e)
{
    if (txtDocumentoEmpleado.Text == "Documento")
    {
        txtDocumentoEmpleado.Text = string.Empty;
        txtDocumentoEmpleado.ForeColor = Color.Black;
    }
}

private void txtDocumentoEmpleado_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtDocumentoEmpleado.Text))
    {
        txtDocumentoEmpleado.Text = "Documento";
        txtDocumentoEmpleado.ForeColor = Color.DimGray;
    }
    else
    {
        VerificarCoincidencias();
    }
}

private void txtTelefonoEmpleado_Enter(object sender, System.EventArgs e)
{
    if (txtTelefonoEmpleado.Text == "Teléfono")
    {
        txtTelefonoEmpleado.Text = string.Empty;
        txtTelefonoEmpleado.ForeColor = Color.Black;
    }
}

private void txtTelefonoEmpleado_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtTelefonoEmpleado.Text))
    {
        txtTelefonoEmpleado.Text = "Teléfono";
        txtTelefonoEmpleado.ForeColor = Color.DimGray;
    }
}

private void txtAlternativoEmpleado_Enter(object sender, System.EventArgs e)
{
    if (txtAlternativoEmpleado.Text == "Alternativo")
    {
        txtAlternativoEmpleado.Text = string.Empty;
        txtAlternativoEmpleado.ForeColor = Color.Black;
    }
}
```

```
private void txtAlternativoEmpleado_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtAlternativoEmpleado.Text))
    {
        txtAlternativoEmpleado.Text = "Alternativo";
        txtAlternativoEmpleado.ForeColor = Color.DimGray;
    }
}

private void txtMailEmpleado_Enter(object sender, System.EventArgs e)
{
    if (txtMailEmpleado.Text == "Mail")
    {
        txtMailEmpleado.Text = string.Empty;
        txtMailEmpleado.ForeColor = Color.Black;
    }
}

private void txtMailEmpleado_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtMailEmpleado.Text))
    {
        txtMailEmpleado.Text = "Mail";
        txtMailEmpleado.ForeColor = Color.DimGray;
    }
}

private void txtObservacionesEmpleado_Enter(object sender, System.EventArgs e)
{
    if (txtObservacionesEmpleado.Text == "Observaciones y/o consideraciones")
    {
        txtObservacionesEmpleado.Text = string.Empty;
        txtObservacionesEmpleado.ForeColor = Color.Black;
    }
}

private void txtObservacionesEmpleado_Leave(object sender, System.EventArgs e)
{
    if (string.IsNullOrEmpty(txtObservacionesEmpleado.Text))
    {
        txtObservacionesEmpleado.Text = "Observaciones y/o consideraciones";
        txtObservacionesEmpleado.ForeColor = Color.DimGray;
    }
}

private void txtBuscarEmpleado_Enter(object sender, EventArgs e)
{
    if (txtBuscarEmpleado.Text == "Buscar")
    {
        txtBuscarEmpleado.Text = string.Empty;
        txtBuscarEmpleado.ForeColor = Color.Black;
    }
}

private void txtBuscarEmpleado_Leave(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtBuscarEmpleado.Text))
    {
        txtBuscarEmpleado.Text = "Buscar";
        txtBuscarEmpleado.ForeColor = Color.DimGray;
    }
}
```

```
}  
private void txtUsuario_Enter(object sender, EventArgs e)  
{  
    if (txtUsuario.Text == "Usuario")  
    {  
        txtUsuario.Text = string.Empty;  
        txtUsuario.ForeColor = Color.Black;  
    }  
}  
private void txtUsuario_Leave(object sender, EventArgs e)  
{  
    if (string.IsNullOrEmpty(txtUsuario.Text))  
    {  
        txtUsuario.Text = "Usuario";  
        txtUsuario.ForeColor = Color.DimGray;  
    }  
}  
private void txtClave_Enter(object sender, EventArgs e)  
{  
    if (txtClave.Text == "Clave")  
    {  
        txtClave.Text = string.Empty;  
        txtClave.ForeColor = Color.Black;  
    }  
}  
private void txtClave_Leave(object sender, EventArgs e)  
{  
    if (string.IsNullOrEmpty(txtClave.Text))  
    {  
        txtClave.Text = "Clave";  
        txtClave.ForeColor = Color.DimGray;  
    }  
}  
  
#endregion  
  
#region Eventos KeyPress en Textbox  
  
private void txtNumDocumentoEmpleado_KeyPress(object sender, KeyPressEventArgs e)  
{  
    string strTexto = txtDocumentoEmpleado.Text;  
    _restricciones.SoloNumeros(e, strTexto);  
}  
private void txtTelefonoEmpleado_KeyPress(object sender, KeyPressEventArgs e)  
{  
    string strTexto = txtTelefonoEmpleado.Text;  
    _restricciones.SoloNumeros(e, strTexto);  
}  
  
private void txtAlternativoEmpleado_KeyPress(object sender, KeyPressEventArgs e)  
{  
    string strTexto = txtAlternativoEmpleado.Text;  
    _restricciones.SoloNumeros(e, strTexto);  
}  
  
private void txtBuscarEmpleado_KeyPress(object sender, KeyPressEventArgs e)  
{  
    if (e.KeyChar == Convert.ToChar(Keys.Enter))  
    {  
        buscar = txtBuscarEmpleado.Text;  
        GetEmpleados();  
    }  
}
```

```

    }
}
private void cmbTipoEmpleado_SelectedIndexChanged(object sender, EventArgs e)
{
    HabilitarClave();
}

#endregion

#region Alta Empleado
private void btnAltaEmpleado_Click(object sender, EventArgs e)
{
    //Valido los campos obligatorios
    ValidarCamposVacios();
    if (!camposObligatoriosVacios)
    {
        //Valido si el contenido de los campos es válido
        ValidarContenido();
        if (!contenidoErroneo)
        {
            //Verifico si hay clientes con los datos registrados
            VerificarCoincidencias();
            if (!personaRegistrada)
            {
                //Veo si el empleado es el que tiene acceso al programa
                if (Convert.ToInt32(cmbTipoEmpleado.SelectedValue) == 2)
                {
                    VerificarClave();
                    if (claveOK)
                    {
                        //Hacemos la edición de la persona.
                        ABMEmpleado();

                        //Luego verificamos si va con jornada.
                        if (chkJornadaEmpleados.Checked)
                        {
                            CargarJornada();
                        }
                        motivoEdicion = 3;
                    }
                }
            }
            else
            {
                ABMEmpleado();

                //Luego verificamos si va con jornada.
                if (chkJornadaEmpleados.Checked)
                {
                    CargarJornada();
                    motivoEdicion = 3;
                }
            }
        }
        else
        {
            //Vamos a mostrar un mensaje, en caso de que esta persona ya esté
            registrada.
            MessageBox.Show("Existe un registro idéntico del dni de esta persona.
            Solo pueden hacerse modificaciones", "Coincidencia encontrada", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
    }
}

```

```
    }
    else
    {
        MessageBox.Show("Campos obligatorios sin completar. Por favor, complete todos los campos requeridos", "Campos vacios", MessageBoxButtons.OK);
    }
}

#endregion

#region Edición de datos de empleado.
private void editarEmpleado_Click(object sender, EventArgs e)
{
    motivoEdicion = 0;
    if (edicionEmpleado)
    {
        DialogResult result = MessageBox.Show("Tiene una edición en proceso. " +
            "¿Desea cancelarla y editar otro empleado?", "Edición en proceso",
            MessageBoxButtons.OKCancel);
        if (result == DialogResult.OK)
        {
            EditarDatos();
        }
    }
    else
    {
        EditarDatos();
    }
}

private void editarClave_Click(object sender, EventArgs e)
{
    motivoEdicion = 2;
    foreach (Control ctrl in this.Controls)
    {
        if (ctrl is TextBox box)
        {
            TextBox t;
            t = box;
            if (t != txtUsuario && t != txtClave)
            {
                t.Enabled = false;
            }
        }
    }
    EditarDatos();
    EditarLogin();
}

private void editarJornada_Click(object sender, EventArgs e)
{
    motivoEdicion = 1;
    CargarJornada();
}

#endregion
}
```