

```
using BussinessLayer;
using Entities;
using System;
using System.Data;
using System.Drawing;
using System.Runtime.InteropServices;
using System.Windows.Forms;

namespace Gym
{
    public partial class Jornadas : Form
    {
        #region Instancias
        //Entidades
        private Jornadas_Empleados _jornadas_Empleados;
        private Entities.Jornadas_Planes _jornadasPlanes;
        private Entities.Empleados _empleados;

        //Capa de Negocio
        private readonly BussinessJornadas _bussinessJornadas;
        private readonly BussinessEmpleados _bussinessEmpleados;

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

        #endregion

        #region Variables
        //variables de tipo table.
        public DataTable DtJornadas = new DataTable();

        //otras.
        private string desde;
        private string hasta;
        private bool todosChk = false;
        private int check;
        private int contador = 0;
        private string formatoHora = "El formato de la hora es este: hh:mm";
        private int pTodosId = -1;
        private int pLunesId = -1;
        private int pMartesId = -1;
        private int pMiercolesId = -1;
        private int pJuevesId = -1;
        private int pViernesId = -1;
        private int pSabadoId = -1;
        private bool contenidoIncorrecto = false;
        private int empleado_ID;
        private bool cargarJornada;
        private int idJornada;
        private int contadorJornadas;
        private bool esEmpleado;
        private bool darleLaBaja;
        private string hora;

        #endregion

        #region Load
        public Jornadas(int jornadaID, bool empleado, bool darBaja)
        {
            InitializeComponent();
        }
    }
}
```

```

        _bussinessJornadas = new BussinessJornadas();
        _jornadas_Empleados = new Jornadas_Empleados();
        _metodosGenerales = new MetodosGenerales();
        _empleados = new Entities.Empleados();
        _bussinessEmpleados = new BussinessEmpleados();
        _restricciones = new Restricciones();
        _jornadasPlanes = new Entities.Jornadas_Planes();

        idJornada = jornadaID;
        esEmpleado = empleado;
        darleLaBaja = darBaja;

        DarDeBajaLaJornada();
    }
    #endregion

    #region Métodos encapsulados
    private void DarDeBajaLaJornada()
    {
        if (darleLaBaja)
        {
            if (esEmpleado)
            {
                _bussinessJornadas.EliminarJornadaEmpleado(idJornada);
            }
            else
            {
                _bussinessJornadas.EliminarJornadaPlan(idJornada);
            }
            MessageBox.Show("Las jornadas asociadas, se han eliminado también",
"Eliminación de jornadas");
            this.Close();
        }
        else
        {
            if (esEmpleado)
            {
                GetJornadaEmpleado();
            }
            else
            {
                EsJornadaDePlanes();
                GetJornadaPlan();
            }
        }
    }
    private void EsJornadaDePlanes()
    {
        foreach (Control ctrl in gbJornadaEmpleado.Controls)
        {
            if (ctrl is TextBox box)
            {
                TextBox t;
                t = box;
                t.ReadOnly = true;
            }
            if (ctrl is CheckBox chk)
            {
                CheckBox c;
                c = chk;
            }
        }
    }

```

```
        c.Enabled = false;
    }
    if (ctrl is Button btn)
    {
        Button b;
        b = btn;
        b.Enabled = false;
    }
}
btnAltaJornada.Enabled = false;
lblCancelar.Text = "Cerrar";
ActiveControl = lblCancelar;
}
private void GetJornadaPlan()
{
    _jornadasPlanes.Plan_ID = idJornada;
    //Una vez asignado, busco la jornada
    //Y la asigno a un datatable, para poder manejar el contenido.
    DtJornadas = _bussinessJornadas.GetJornadaPlan(_jornadasPlanes);

    if (DtJornadas.Rows.Count > 0)
    {
        cargarJornada = true;
        //cargo la jornada en los textbox correspondientes
        CargarJornada();
    }
    else
    {
        cargarJornada = false;
    }
}
private void GetJornadaEmpleado()
{
    _jornadas_Empleados.Empleado_ID = idJornada;
    //Una vez asignado, busco la jornada
    //Y la asigno a un datatable, para poder manejar el contenido.
    DtJornadas = _bussinessJornadas.GetJornadaEmpleado(_jornadas_Empleados);

    if (DtJornadas.Rows.Count > 0)
    {
        cargarJornada = true;
        //cargo la jornada en los textbox correspondientes
        CargarJornada();
    }
    else
    {
        cargarJornada = false;
    }
}

private void CargarJornada()
{
    foreach (DataRow dr in DtJornadas.Rows)
    {
        //Vamos a switchear cada vuelta para que vaya
        //colocando los datos cada vuelta donde corresponda.
        switch (dr[2].ToString())
        {
            case "Todos":
                chkTodos.Checked = true;
            break;
        }
    }
}
```

```
//Este es el Id de la jornada que se carga
pTodosId = Convert.ToInt32(dr[0]);
txtDesdeLunes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
txtHastaLunes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));
txtDesdeLunes.ForeColor = Color.Black;
txtHastaLunes.ForeColor = Color.Black;
VerificarChk();
break;
case "Lunes":
    txtDesdeLunes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
Convert.ToString(dr[3]);
    txtHastaLunes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));
    txtDesdeLunes.ForeColor = Color.Black;
    txtHastaLunes.ForeColor = Color.Black;
    pLunesId = Convert.ToInt32(dr[0]);
    chkLunes.Checked = true;
    break;
case "Martes":
    txtDesdeMartes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
    txtHastaMartes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));
    txtDesdeMartes.ForeColor = Color.Black;
    txtHastaMartes.ForeColor = Color.Black;
    pMartesId = Convert.ToInt32(dr[0]);
    chkMartes.Checked = true;
    break;
case "Miercoles":
    txtDesdeMiercoles.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
    txtHastaMiercoles.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));
    txtDesdeMiercoles.ForeColor = Color.Black;
    txtHastaMiercoles.ForeColor = Color.Black;
    pMiercolesId = Convert.ToInt32(dr[0]);
    chkMiercoles.Checked = true;
    break;
case "Jueves":
    txtDesdeJueves.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
    txtHastaJueves.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));
    txtDesdeJueves.ForeColor = Color.Black;
    txtHastaJueves.ForeColor = Color.Black;
    pJuevesId = Convert.ToInt32(dr[0]);
    chkJueves.Checked = true;
    break;
case "Viernes":
    txtDesdeViernes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
    txtHastaViernes.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));
    txtDesdeViernes.ForeColor = Color.Black;
    txtHastaViernes.ForeColor = Color.Black;
    pViernesId = Convert.ToInt32(dr[0]);
    chkViernes.Checked = true;
    break;
case "Sabado":
    txtDesdeSabado.Text =
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[3].ToString()));
```

```
                txtHastaSabado.Text =  
Convert.ToString(_metodosGenerales.ConvertirfechaAhora(dr[4].ToString()));  
                txtDesdeSabado.ForeColor = Color.Black;  
                txtHastaSabado.ForeColor = Color.Black;  
                pSabadoId = Convert.ToInt32(dr[0]);  
                chkSabado.Checked = true;  
                break;  
            }  
        }  
    }  
    private void VerificarChk()  
    {  
        switch (check)  
        {  
            case 0:  
                ActivarTodoChk();  
                break;  
            case 1:  
                ChkDias();  
                break;  
            case 2:  
                ChkDias();  
                break;  
            case 3:  
                ChkDias();  
                break;  
            case 4:  
                ChkDias();  
                break;  
            case 5:  
                ChkDias();  
                break;  
            case 6:  
                ChkDias();  
                break;  
        }  
        contador = 0;  
    }  
    private void ChkDias()  
    {  
        RevisarTodosChk();  
        if (chkTodos.Checked)  
        {  
            chkTodos.Checked = false;  
            todosChk = false;  
        }  
        else  
        {  
            if (contador > 5 && !chkTodos.Checked)  
            {  
                ActivarTodoChk();  
            }  
        }  
    }  
    private void ActivarTodoChk()  
    {  
        if (!chkTodos.Checked && contador < 7 && !todosChk)  
        {  
            foreach (Control chk in gbJornadaEmpleado.Controls)  
            {
```

```
        if (chk is CheckBox box)
        {
            CheckBox c;
            c = box;
            c.Checked = true;
        }
    }
    todosChk = true;
}
else
{
    if (todosChk)
    {
        //Si el chk que dice "Seleccionar todos" está seleccionado
        //Entonces todos los otros chk se van a seleccionar
        //Caso contrario, se les quitará el check
        foreach (Control chk in gbJornadaEmpleado.Controls)
        {
            if (chk is CheckBox box)
            {
                CheckBox c;
                c = box;
                //Deshabilito el check en cada checkbox
                c.Checked = false;
            }
        }
        todosChk = false;
    }
    else
    {
        foreach (Control chk in gbJornadaEmpleado.Controls)
        {
            if (chk is CheckBox box)
            {
                CheckBox c;
                c = box;
                c.Checked = true;
            }
        }
        todosChk = true;
    }
}
}

private void RevisarTodosChk()
{
    foreach (Control chk in gbJornadaEmpleado.Controls)
    {
        if (chk is CheckBox box)
        {
            CheckBox c;
            c = box;
            if (c.Checked)
            {
                contador++;
            }
        }
    }
}

private void AsignacionHoras()
{

```

```
int contadorDesde = desde.Length;
int contadorHasta = hasta.Length;

if (contadorDesde <= 5 && contadorHasta <= 5)
{
    _jornadas_Empleados.Desde_Hora = _metodosGenerales.ConvertirHoraAfecha(desde);
    _jornadas_Empleados.Hasta_Hora = _metodosGenerales.ConvertirHoraAfecha(hasta);
}
else
{
    MessageBox.Show("El campo de horas, está mal registrado. Desde registrarse con  
este formato: ##:##", "Formato incorrecto", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void GuardarJornada()
{
    foreach (Control chk in this.gbJornadaEmpleado.Controls)
    {
        if (chk is CheckBox box)
        {
            CheckBox c;
            c = box;
            //Y luego de cada checkbox le asignamos el horario.
            //Lunes
            if (c.Checked)
            {
                switch (c.Text)
                {
                    case "Lunes":
                        _jornadas_Empleados.Dia = "Lunes";
                        _jornadas_Empleados.Jornada_Empleado_ID = pLunesId;
                        desde = txtDesdeLunes.Text.ToString();
                        hasta = txtHastaLunes.Text.ToString();
                        AsignacionHoras();
                        break;
                    case "Martes":
                        _jornadas_Empleados.Dia = "Martes";
                        _jornadas_Empleados.Jornada_Empleado_ID = pMartesId;
                        desde = txtDesdeMartes.Text.ToString();
                        hasta = txtHastaMartes.Text.ToString();
                        AsignacionHoras();
                        break;
                    case "Miercoles":
                        _jornadas_Empleados.Dia = "Miercoles";
                        _jornadas_Empleados.Jornada_Empleado_ID = pMiercolesId;
                        desde = txtDesdeMiercoles.Text.ToString();
                        hasta = txtHastaMiercoles.Text.ToString();
                        AsignacionHoras();
                        break;
                    case "Jueves":
                        _jornadas_Empleados.Dia = "Jueves";
                        _jornadas_Empleados.Jornada_Empleado_ID = pJuevesId;
                        desde = txtDesdeJueves.Text.ToString();
                        hasta = txtHastaJueves.Text.ToString();
                        AsignacionHoras();
                        break;
                    case "Viernes":
                        _jornadas_Empleados.Dia = "Viernes";
                        _jornadas_Empleados.Jornada_Empleado_ID = pViernesId;
                        desde = txtDesdeViernes.Text.ToString();
```

```

        hasta = txtHastaViernes.Text.ToString();
        AsignacionHoras();
        break;
    case "Sabado":
        _jornadas_Empleados.Dia = "Sabado";
        _jornadas_Empleados.Jornada_Empleado_ID = pSabadoId;
        desde = txtDesdeSabado.Text.ToString();
        hasta = txtHastaSabado.Text.ToString();
        AsignacionHoras();
        break;
    }
    _jornadas_Empleados.Estado = "A";
    _jornadas_Empleados.Empleado_ID = idJornada;
    if (_jornadas_Empleados.Jornada_Empleado_ID != -1)
    {
        _bussinessJornadas.EditarJornadaEmpleado(_jornadas_Empleados);
    }
    else
    {
        _bussinessJornadas.AltaJornadaEmpleados(_jornadas_Empleados);
    }
    desde = string.Empty;
    hasta = string.Empty;
}
}
}
}
private void JornadasManagement()
{
    //Acá editamos
    if (cargarJornada)
    {
        GuardarJornada();
    }
    else
    {
        //Acá creamos:

        //Ahora tenemos 2 opciones.
        // 1 es que se haya seleccionado el check "todos"
        if (todosChk)
        {
            _jornadas_Empleados.Empleado_ID = idJornada;
            _jornadas_Empleados.Dia = "Todos";
            _jornadas_Empleados.Estado = "A";
            desde = txtDesdeLunes.Text.ToString();
            hasta = txtHastaLunes.Text.ToString();
            AsignacionHoras();

            _bussinessJornadas.AltaJornadaEmpleados(_jornadas_Empleados);
        }
        else
        {
            //Y la otra es que se hayan seleccionado varios dias
            //Vamos a verificar los checkbox que se hayan seleccionado
            GuardarJornada();
        }
    }
}
}
private void EliminarJornada(int jornadaID)

```



```

{
    _bussinessJornadas.EliminarJornadaEmpleado(jornadaID);
}

private void RevisarContenido()
{
    if (chkTodos.Checked)
    {
        if (txtDesdeLunes.Text != "hh:mm" ||
!string.IsNullOrEmpty(txtDesdeLunes.Text))
        {
            contenidoIncorrecto = false;
        }
        else
        {
            contenidoIncorrecto = true;
        }
    }
    else
    {
        foreach (Control ctrl in gbJornadaEmpleado.Controls)
        {
            if (ctrl is TextBox txt)
            {
                //Si el contenido de txt es igual a "hh:mm" o es nulo o está vacío
                if (txt.Text == "hh:mm" || string.IsNullOrEmpty(txt.Text))
                {
                    //entonces voy a ver si el checkbox que le corresponde está
seleccionado

                    foreach (Control ctl in gbJornadaEmpleado.Controls)
                    {
                        if (ctl is CheckBox chk)
                        {
                            //si está seleccionado y es el que corresponde..
                            if (chk.Checked && txt.Name.Contains(chk.Text))
                            {
                                //entonces el contenido está mal registrado y hay que
registrar bien

                                //el contenido de la hora en cada cuadro.
                                contenidoIncorrecto = true;
                                break;
                            }
                        }
                    }
                    if (contenidoIncorrecto)
                    {
                        break;
                    }
                }
                else
                {
                    contenidoIncorrecto = false;
                }
            }
        }
    }
}

private void VerificarFormatoHora()
{
    foreach (Control ctrl in gbJornadaEmpleado.Controls)
    {

```

```
        if (ctrl is TextBox box)
        {
            TextBox t;
            t = box;
            if (t.Text.Length < 5)
            {
                contenidoIncorrecto = true; ;
                break;
            }
            else
            {
                contenidoIncorrecto = false;
            }
        }
    }
}

#endregion

#region Focus TextBox

//Lunes
private void txtDesdeLunes_Enter(object sender, EventArgs e)
{
    if (txtDesdeLunes.Text == "hh:mm")
    {
        txtDesdeLunes.Text = string.Empty;
        txtDesdeLunes.ForeColor = Color.Black;
    }
}

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

    if (string.IsNullOrEmpty(txtDesdeLunes.Text))
    {
        txtDesdeLunes.Text = "hh:mm";
        txtDesdeLunes.ForeColor = Color.DimGray;
    }
}

private void txtHastaLunes_Enter(object sender, EventArgs e)
{
    if (txtHastaLunes.Text == "hh:mm")
    {
        txtHastaLunes.Text = string.Empty;
        txtHastaLunes.ForeColor = Color.Black;
    }
}

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

    if (string.IsNullOrEmpty(txtHastaLunes.Text))
    {
        txtHastaLunes.Text = "hh:mm";
        txtHastaLunes.ForeColor = Color.DimGray;
    }
}
}
```

```
//Martes
private void txtDesdeMartes_Enter(object sender, EventArgs e)
{
    if (txtDesdeMartes.Text == "hh:mm")
    {
        txtDesdeMartes.Text = string.Empty;
        txtDesdeMartes.ForeColor = Color.Black;
    }
}
private void txtDesdeMartes_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtDesdeMartes.Text))
    {
        txtDesdeMartes.Text = "hh:mm";
        txtDesdeMartes.ForeColor = Color.DimGray;
    }
}
private void txtHastaMartes_Enter(object sender, EventArgs e)
{
    if (txtHastaMartes.Text == "hh:mm")
    {
        txtHastaMartes.Text = string.Empty;
        txtHastaMartes.ForeColor = Color.Black;
    }
}
private void txtHastaMartes_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtHastaMartes.Text))
    {
        txtHastaMartes.Text = "hh:mm";
        txtHastaMartes.ForeColor = Color.DimGray;
    }
}

//Miércoles
private void txtDesdeMiercoles_Enter(object sender, EventArgs e)
{
    if (txtDesdeMiercoles.Text == "hh:mm")
    {
        txtDesdeMiercoles.Text = string.Empty;
        txtDesdeMiercoles.ForeColor = Color.Black;
    }
}
private void txtDesdeMiercoles_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtDesdeMiercoles.Text))
    {
        txtDesdeMiercoles.Text = "hh:mm";
        txtDesdeMiercoles.ForeColor = Color.DimGray;
    }
}
private void txtHastaMiercoles_Enter(object sender, EventArgs e)
```

```
{
    if (txtHastaMiercoles.Text == "hh:mm")
    {
        txtHastaMiercoles.Text = string.Empty;
        txtHastaMiercoles.ForeColor = Color.Black;
    }
}

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

    if (string.IsNullOrEmpty(txtHastaMiercoles.Text))
    {
        txtHastaMiercoles.Text = "hh:mm";
        txtHastaMiercoles.ForeColor = Color.DimGray;
    }
}

//Jueves
private void txtDesdeJueves_Enter(object sender, EventArgs e)
{
    if (txtDesdeJueves.Text == "hh:mm")
    {
        txtDesdeJueves.Text = string.Empty;
        txtDesdeJueves.ForeColor = Color.Black;
    }
}

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

    if (string.IsNullOrEmpty(txtDesdeJueves.Text))
    {
        txtDesdeJueves.Text = "hh:mm";
        txtDesdeJueves.ForeColor = Color.DimGray;
    }
}

private void txtHastaJueves_Enter(object sender, EventArgs e)
{
    if (txtHastaJueves.Text == "hh:mm")
    {
        txtHastaJueves.Text = string.Empty;
        txtHastaJueves.ForeColor = Color.Black;
    }
}

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

    if (string.IsNullOrEmpty(txtHastaJueves.Text))
    {
        txtHastaJueves.Text = "hh:mm";
        txtHastaJueves.ForeColor = Color.DimGray;
    }
}

//Viernes
private void txtDesdeViernes_Enter(object sender, EventArgs e)
```

```
{
    if (txtDesdeViernes.Text == "hh:mm")
    {
        txtDesdeViernes.Text = string.Empty;
        txtDesdeViernes.ForeColor = Color.Black;
    }
}
private void txtDesdeViernes_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtDesdeViernes.Text))
    {
        txtDesdeViernes.Text = "hh:mm";
        txtDesdeViernes.ForeColor = Color.DimGray;
    }
}
private void txtHastaViernes_Enter(object sender, EventArgs e)
{
    if (txtHastaViernes.Text == "hh:mm")
    {
        txtHastaViernes.Text = string.Empty;
        txtHastaViernes.ForeColor = Color.Black;
    }
}
private void txtHastaViernes_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtHastaViernes.Text))
    {
        txtHastaViernes.Text = "hh:mm";
        txtHastaViernes.ForeColor = Color.DimGray;
    }
}

//Sábado
private void txtDesdeSabado_Enter(object sender, EventArgs e)
{
    if (txtDesdeSabado.Text == "hh:mm")
    {
        txtDesdeSabado.Text = string.Empty;
        txtDesdeSabado.ForeColor = Color.Black;
    }
}
private void txtDesdeSabado_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtDesdeSabado.Text))
    {
        txtDesdeSabado.Text = "hh:mm";
        txtDesdeSabado.ForeColor = Color.DimGray;
    }
}
private void txtHastaSabado_Enter(object sender, EventArgs e)
{
    if (txtHastaSabado.Text == "hh:mm")
```

```

    {
        txtHastaSabado.Text = string.Empty;
        txtHastaSabado.ForeColor = Color.Black;
    }
}
private void txtHastaSabado_Leave(object sender, EventArgs e)
{
    //Por el lado contrario, cuando se quita el foco
    //del control, lo volvemos a su estado y color inicial

    if (string.IsNullOrEmpty(txtHastaSabado.Text))
    {
        txtHastaSabado.Text = "hh:mm";
        txtHastaSabado.ForeColor = Color.DimGray;
    }
}
#endregion

#region Clicks
private void chkTodos_Click(object sender, EventArgs e)
{
    check = 0;
    VerificarChk();
}

private void lblCancelar_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    this.Close();
}

private void btnAltaJornada_Click(object sender, EventArgs e)
{
    RevisarContenido();
    VerificarFormatoHora();
    if (contenidoIncorrecto)
    {
        MessageBox.Show("Si selecciona un día, el horario correspondiente " +
            "debe ser registrado de manera correcta.", "Datos erróneos",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        JornadasManagement();
        this.Close();
    }
}

#endregion

#region Arrastrar Form

//Se importan archivos DLL para poder realizar la captura de la acción del mouse
//y poder arrastrar los formularios desde donde se realiza la captura

[DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
private extern static void ReleaseCapture();
[DllImport("user32.DLL", EntryPoint = "SendMessage")]
private extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int
lParam);
private void Jornadas_MouseDown(object sender, MouseEventArgs e)

```

```
{
    ReleaseCapture();
    SendMessage(this.Handle, 0x112, 0xf012, 0);
}

#endregion

#region KeyPress en Textbox
private void txtHastaSabado_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtDesdeSabado_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtDesdeViernes_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtHastaViernes_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtDesdeJueves_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtHastaJueves_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtDesdeMiercoles_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtHastaMiercoles_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtDesdeMartes_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtHastaMartes_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private void txtDesdeLunes_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}
```

```
}

private void txtHastaLunes_KeyPress(object sender, KeyPressEventArgs e)
{
    _restricciones.Horarios(e);
}

private string AgregarColon(string texto)
{
    //verificamos si el texto ingresado hasta el momento tiene 2 caracteres
    //si es así, se le agregan los 2 puntos, para evitar ponerlo a cada momento
    //cuando están poniendo los horarios.
    if (texto.Length == 2)
    {
        texto += ":";
    }

    return texto;
}

private void txtDesdeLunes_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtDesdeLunes.Text;
    txtDesdeLunes.Text = AgregarColon(hora);
    txtDesdeLunes.Select(txtDesdeLunes.Text.Length, 0);
}

private void txtHastaLunes_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtHastaLunes.Text;
    txtHastaLunes.Text = AgregarColon(hora);
    txtHastaLunes.Select(txtHastaLunes.Text.Length, 0);
}

private void txtDesdeMartes_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtDesdeMartes.Text;
    txtDesdeMartes.Text = AgregarColon(hora);
    txtDesdeMartes.Select(txtDesdeMartes.Text.Length, 0);
}

private void txtHastaMartes_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtHastaMartes.Text;
    txtHastaMartes.Text = AgregarColon(hora);
    txtHastaMartes.Select(txtHastaMartes.Text.Length, 0);
}

private void txtDesdeMiercoles_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtDesdeMiercoles.Text;
    txtDesdeMiercoles.Text = AgregarColon(hora);
    txtDesdeMiercoles.Select(txtDesdeMiercoles.Text.Length, 0);
}

private void txtHastaMiercoles_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtHastaMiercoles.Text;
    txtHastaMiercoles.Text = AgregarColon(hora);
    txtHastaMiercoles.Select(txtHastaMiercoles.Text.Length, 0);
}
```



```
}

private void txtDesdeJueves_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtDesdeJueves.Text;
    txtDesdeJueves.Text = AgregarColon(hora);
    txtDesdeJueves.Select(txtDesdeJueves.Text.Length, 0);
}

private void txtHastaJueves_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtHastaJueves.Text;
    txtHastaJueves.Text = AgregarColon(hora);
    txtHastaJueves.Select(txtHastaJueves.Text.Length, 0);
}

private void txtDesdeViernes_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtDesdeViernes.Text;
    txtDesdeViernes.Text = AgregarColon(hora);
    txtDesdeViernes.Select(txtDesdeViernes.Text.Length, 0);
}

private void txtHastaViernes_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtHastaViernes.Text;
    txtHastaViernes.Text = AgregarColon(hora);
    txtHastaViernes.Select(txtHastaViernes.Text.Length, 0);
}

private void txtDesdeSabado_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtDesdeSabado.Text;
    txtDesdeSabado.Text = AgregarColon(hora);
    txtDesdeSabado.Select(txtDesdeSabado.Text.Length, 0);
}

private void txtHastaSabado_KeyUp(object sender, KeyEventArgs e)
{
    hora = txtHastaSabado.Text;
    txtHastaSabado.Text = AgregarColon(hora);
    txtHastaSabado.Select(txtHastaSabado.Text.Length, 0);
}

#endregion

#region Selección de checks
private void chkLunes_Click(object sender, EventArgs e)
{
    check = 1;
    VerificarChk();
}
private void chkMartes_Click(object sender, EventArgs e)
{
    check = 2;
    VerificarChk();
    if (chkMartes.Checked && !chkTodos.Checked)
    {
        txtDesdeMartes.Enabled = true;
        txtHastaMartes.Enabled = true;
    }
}
```

```
else
{
    txtDesdeMartes.Enabled = false;
    txtHastaMartes.Enabled = false;
}
}
private void chkMiercoles_Click(object sender, EventArgs e)
{
    check = 3;
    VerificarChk();
    if (chkMiercoles.Checked && !chkTodos.Checked)
    {
        txtDesdeMiercoles.Enabled = true;
        txtHastaMiercoles.Enabled = true;
    }
    else
    {
        txtDesdeMiercoles.Enabled = false;
        txtHastaMiercoles.Enabled = false;
    }
}
private void chkJueves_Click(object sender, EventArgs e)
{
    check = 4;
    VerificarChk();
    if (chkJueves.Checked && !chkTodos.Checked)
    {
        txtDesdeJueves.Enabled = true;
        txtHastaJueves.Enabled = true;
    }
    else
    {
        txtHastaJueves.Enabled = false;
        txtDesdeJueves.Enabled = false;
    }
}
private void chkViernes_Click(object sender, EventArgs e)
{
    check = 5;
    VerificarChk();
    if (chkViernes.Checked && !chkTodos.Checked)
    {
        txtDesdeViernes.Enabled = true;
        txtHastaViernes.Enabled = true;
    }
    else
    {
        txtDesdeViernes.Enabled = false;
        txtHastaViernes.Enabled = false;
    }
}
private void chkSabado_Click(object sender, EventArgs e)
{
    check = 6;
    VerificarChk();
    if (chkSabado.Checked && !chkTodos.Checked)
    {
        txtDesdeSabado.Enabled = true;
        txtHastaSabado.Enabled = true;
    }
    else
```

```

        {
            txtDesdeSabado.Enabled = false;
            txtHastaSabado.Enabled = false;
        }
    }

#endregion

#region Click botones de borrado de jornada.
private void btnDellunes_Click(object sender, EventArgs e)
{
    if (pLunesId != -1)
    {
        EliminarJornada(pLunesId);
        pLunesId = -1;
    }
    else if (!string.IsNullOrEmpty(txtDesdeLunes.Text) && txtDesdeLunes.Text !=
"hh:mm")
    {
        txtDesdeLunes.Text = "hh:mm";
    }
    else if (!string.IsNullOrEmpty(txtHastaLunes.Text) && txtHastaLunes.Text !=
"hh:mm")
    {
        txtHastaLunes.Text = "hh:mm";
    }
    else if (chkTodos.Checked)
    {
        chkTodos.Checked = false;
        todosChk = false;
    }
    chkLunes.Checked = false;
}
private void btnDelMartes_Click(object sender, EventArgs e)
{
    if (pMartesId != -1)
    {
        EliminarJornada(pMartesId);
        pMartesId = -1;
    }
    else if (!string.IsNullOrEmpty(txtDesdeMartes.Text) && txtDesdeMartes.Text !=
"hh:mm")
    {
        txtDesdeMartes.Text = "hh:mm";
    }
    else if (!string.IsNullOrEmpty(txtHastaMartes.Text) && txtHastaMartes.Text !=
"hh:mm")
    {
        txtHastaMartes.Text = "hh:mm";
    }
    else if (chkTodos.Checked)
    {
        chkTodos.Checked = false;
        todosChk = false;
    }
    chkMartes.Checked = false;
}
private void btnDelMiercoles_Click(object sender, EventArgs e)
{
    if (pMiercolesId != -1)
    {
        EliminarJornada(pMiercolesId);
    }
}

```

```
        pMiercolesId = -1;
    }
    else if (!string.IsNullOrEmpty(txtDesdeMiercoles.Text) && txtDesdeMiercoles.Text
!= "hh:mm")
    {
        txtDesdeMiercoles.Text = "hh:mm";
    }
    else if (!string.IsNullOrEmpty(txtHastaMiercoles.Text) && txtHastaMiercoles.Text
!= "hh:mm")
    {
        txtHastaMiercoles.Text = "hh:mm";
    }
    else if (chkTodos.Checked)
    {
        chkTodos.Checked = false;
        todosChk = false;
    }
    chkMiercoles.Checked = false;
}
private void btnDelJueves_Click(object sender, EventArgs e)
{
    if (pJuevesId != -1)
    {
        EliminarJornada(pJuevesId);
        pJuevesId = -1;
    }
    else if (!string.IsNullOrEmpty(txtDesdeJueves.Text) && txtDesdeJueves.Text !=
"hh:mm")
    {
        txtDesdeJueves.Text = "hh:mm";
    }
    else if (!string.IsNullOrEmpty(txtHastaJueves.Text) && txtHastaJueves.Text !=
"hh:mm")
    {
        txtHastaJueves.Text = "hh:mm";
    }
    else if (chkTodos.Checked)
    {
        chkTodos.Checked = false;
        todosChk = false;
    }
    chkJueves.Checked = false;
}
private void btnDelViernes_Click(object sender, EventArgs e)
{
    if (pViernesId != -1)
    {
        EliminarJornada(pViernesId);
        pViernesId = -1;
    }
    else if (!string.IsNullOrEmpty(txtDesdeViernes.Text) && txtDesdeViernes.Text !=
"hh:mm")
    {
        txtDesdeViernes.Text = "hh:mm";
    }
    else if (!string.IsNullOrEmpty(txtHastaViernes.Text) && txtHastaViernes.Text !=
"hh:mm")
    {
        txtHastaViernes.Text = "hh:mm";
    }
    else if (chkTodos.Checked)
    {

```

```
        chkTodos.Checked = false;
        todosChk = false;
    }
    chkViernes.Checked = false;
}
private void btnDelSabado_Click(object sender, EventArgs e)
{
    if (pSabadoId != -1)
    {
        EliminarJornada(pSabadoId);
        pSabadoId = -1;
    }
    else if (!string.IsNullOrEmpty(txtDesdeSabado.Text) && txtDesdeSabado.Text !=
"hh:mm")
    {
        txtDesdeSabado.Text = "hh:mm";
    }
    else if (!string.IsNullOrEmpty(txtHastaSabado.Text) && txtHastaSabado.Text !=
"hh:mm")
    {
        txtHastaSabado.Text = "hh:mm";
    }
    else if (chkTodos.Checked)
    {
        chkTodos.Checked = false;
        todosChk = false;
    }
    chkSabado.Checked = false;
}

#endregion
    }
}
```