namespace TRABAJO\_PROGRAMACIÓN

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void btnGuardar\_Click(object sender, EventArgs e)

{

BorrarMensajeError();

if (ValidarCampos())

{

MessageBox.Show("Datos Ingresados Correctamente");

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Form2 frm = new Form2();

frm.Show();

}

private void cboTipoMedicamentos\_SelectedIndexChanged(object sender, EventArgs e)

{

cboTipoMedicamentos.Items.Add("Excipientes");

cboTipoMedicamentos.Items.Add("Analgésicos");

cboTipoMedicamentos.Items.Add("Antiinflamatorios");

cboTipoMedicamentos.Items.Add("Laxantes y antidiarreicos");

cboTipoMedicamentos.Items.Add("Antiinfecciosos");

cboTipoMedicamentos.Items.Add("Antipiréticos");

cboTipoMedicamentos.Items.Add("Antitusivos y mucolíticos");

}

private void cboFormaFarmaceutica\_SelectedIndexChanged(object sender, EventArgs e)

{

cboFormaFarmaceutica.Items.Add("Comprimidos");

cboFormaFarmaceutica.Items.Add("Grageas");

cboFormaFarmaceutica.Items.Add("Cápsulas");

cboFormaFarmaceutica.Items.Add("Polvos");

cboFormaFarmaceutica.Items.Add("Píldoras");

cboFormaFarmaceutica.Items.Add("Supositorios");

}

private void cboViaAdministracion\_SelectedIndexChanged(object sender, EventArgs e)

{

cboViaAdministracion.Items.Add("Vía sublingual y oral");

cboViaAdministracion.Items.Add("vía bucal");

cboViaAdministracion.Items.Add("Vía rectal");

cboViaAdministracion.Items.Add("Vía vaginal");

cboViaAdministracion.Items.Add("Vía nasal");

cboViaAdministracion.Items.Add("Vía cutánea");

cboViaAdministracion.Items.Add("Vía transdérmica");

}

private void txtNombreProfesional\_TextChanged(object sender, EventArgs e)

{

}

private void txtNombreProfesional\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 32 && e.KeyChar <= 64) || (e.KeyChar >= 91 && e.KeyChar <= 96) || (e.KeyChar >= 123 && e.KeyChar <= 255))

{

MessageBox.Show("Solo se permite ingresar letras. Presiona OK para continuar.", "Alerta", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

return;

}

}

private void txtNumeroOrden\_TextChanged(object sender, EventArgs e)

{

}

private void txtNumeroOrden\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 32 && e.KeyChar <= 47) || (e.KeyChar >= 58 && e.KeyChar <= 255))

{

MessageBox.Show("Solo se permite ingresar números. Presiona OK para continuar", "Alerta", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

return;

}

}

private void txtNombreMedicamento\_TextChanged(object sender, EventArgs e)

{

}

private void txtNombreMedicamento\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 32 && e.KeyChar <= 64) || (e.KeyChar >= 91 && e.KeyChar <= 96) || (e.KeyChar >= 123 && e.KeyChar <= 255))

{

MessageBox.Show("Solo se permite ingresar letras. Presiona OK para continuar.", "Alerta", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

return;

}

}

private void txtPeriodoDuracion\_TextChanged(object sender, EventArgs e)

{

}

private void txtPeriodoDuracion\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 32 && e.KeyChar <= 64) || (e.KeyChar >= 91 && e.KeyChar <= 96) || (e.KeyChar >= 123 && e.KeyChar <= 255))

{

MessageBox.Show("Solo se permite ingresar letras. Presiona OK para continuar.", "Alerta", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

return;

}

}

private void txtCantidadTotal\_TextChanged(object sender, EventArgs e)

{

}

private void txtCantidadTotal\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 32 && e.KeyChar <= 47) || (e.KeyChar >= 58 && e.KeyChar <= 255))

{

MessageBox.Show("Solo se permite ingresar números. Presiona OK para continuar", "Alerta", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

return;

}

}

private bool ValidarCampos()

{

bool ok = true;

if (txtNombreProfesional.Text == "")

{

ok = false;

errorProvider1.SetError(txtNombreProfesional, "Ingresar Nombre Profesional");

}

if (dtpFechaPrescripcion.Text == "")

{

ok = false;

errorProvider1.SetError(dtpFechaPrescripcion, "Ingresar Fecha de Prescripción");

}

if (txtNumeroOrden.Text == "")

{

ok = false;

errorProvider1.SetError(txtNumeroOrden, "Ingresar Número de Orden");

}

if (cboTipoMedicamentos.Text == "")

{

ok = false;

errorProvider1.SetError(cboTipoMedicamentos, "Seleccionar Tipo de Medicamento");

}

if (txtNombreMedicamento.Text == "")

{

ok = false;

errorProvider1.SetError(txtNombreMedicamento, "Ingresar el Nombre del Medicamento");

}

if (cboFormaFarmaceutica.Text == "")

{

ok = false;

errorProvider1.SetError(cboFormaFarmaceutica, "Seleccionar la Forma Farmacéutica");

}

if (txtPeriodoDuracion.Text == "")

{

ok = false;

errorProvider1.SetError(txtPeriodoDuracion, "Ingresar Periodo de Duración");

}

if (cboViaAdministracion.Text == "")

{

ok = false;

errorProvider1.SetError(cboViaAdministracion, "Seleccionar Vía de Administración");

}

if (txtCantidadTotal.Text == "")

{

ok = false;

errorProvider1.SetError(txtCantidadTotal, "Ingresar Cantidad Total de Unidades Farmacéuticas");

}

if (dtpVigenciaPrescripcion.Text == "")

{

ok = false;

errorProvider1.SetError(dtpVigenciaPrescripcion, "Ingresar Vigencia de Prescripción");

}

return ok;

}

private void BorrarMensajeError()

{

errorProvider1.SetError(txtNombreProfesional, "");

errorProvider1.SetError(dtpFechaPrescripcion, "");

errorProvider1.SetError(txtNumeroOrden, "");

errorProvider1.SetError(cboTipoMedicamentos, "");

errorProvider1.SetError(txtNombreMedicamento, "");

errorProvider1.SetError(cboFormaFarmaceutica, "");

errorProvider1.SetError(txtPeriodoDuracion, "");

errorProvider1.SetError(cboViaAdministracion, "");

errorProvider1.SetError(txtCantidadTotal, "");

errorProvider1.SetError(dtpVigenciaPrescripcion, "");

}

}

}