



VNiVERSIDAD
D SALAMANCA

PACTÓMETRO

TRABAJO FINAL
INTERFACES GRÁFICAS DE
USUARIO

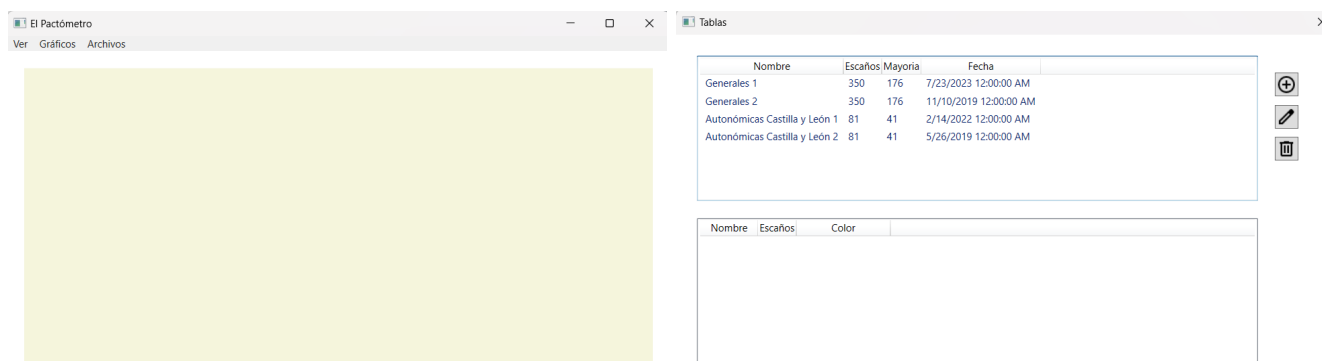
Rubén Angoso Berrocal

Índice

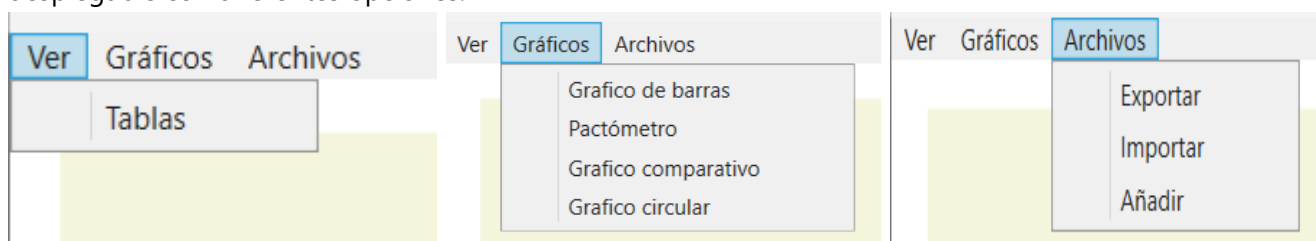
- [Enlace a Manual de usuario](#)
 - [Gráfico de barras](#)
 - [Pactómetro](#)
 - [Gráfico comparativo](#)
 - [Gráfico circular](#)
- [Enlace a Manual de programador](#)
 - [MainWindow](#)
 - [Tablas](#)
 - [EditEleccion y NewEleccion](#)
- [Enlace a Diagrama de clases](#)
- [Bibliografía](#)

Manual de usuario

Al iniciar la aplicación se abrirán estas dos ventanas:



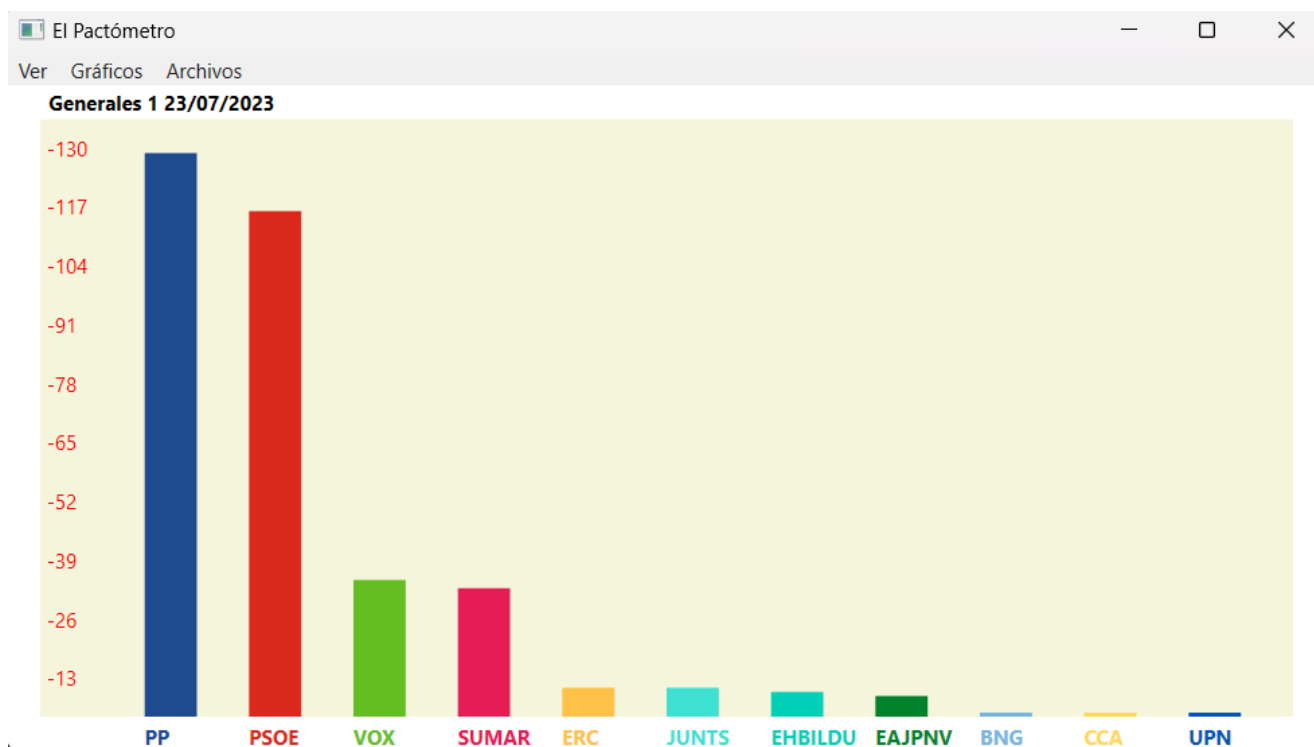
En la ventana secundaria aparecen precargadas 4 elecciones que podrán ser seleccionadas para ser dibujadas en la ventana principal. Además, hay 3 botones para añadir, editar y eliminar elecciones. En la ventana principal, se encuentran tres botones "Ver", "Gráficos" y "Archivos" que al pulsarlos muestran un desplegable con diferentes opciones:



"Ver" muestra u oculta la ventana secundaria. Cada botón de "Gráficos" corresponde uno de los tipos de gráficos que se pueden dibujar y "Archivos" permite exportar e importar las elecciones en formato CSV.

Gráfico de barras

Es el gráfico escogido por defecto, al seleccionar una elección aparecerá en la ventana principal los partidos de la elección seleccionada representada por rectángulos con su color característico y un texto indicando que partido es. También aparece una leyenda indicando el número de escaños para poder hacernos una idea de los escaños del partido sin necesidad de pasar el cursor por encima del rectángulo.



Pactómetro

Al seleccionar el tipo de gráfico "Pactómetro" aparecerá en una columna todos los partidos apilados (columna 2), al darle clic izquierdo sobre uno de los partidos, este se moverá a otra columna (columna 1) y al volver a darle clic se moverá a otra columna (columna 2). Si el número de escaños del partido conforma más del 5% de la elección aparecerá un texto al lado del rectángulo representante del partido.

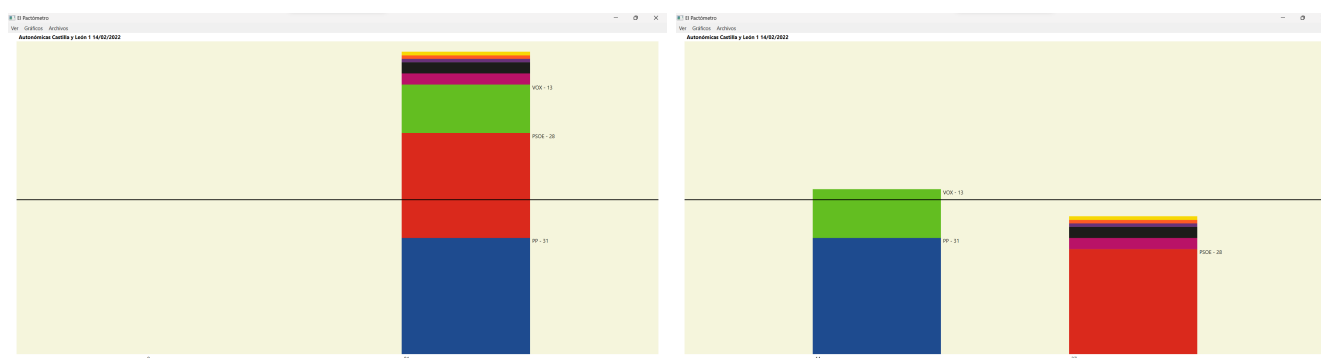


Gráfico comparativo

Al seleccionar el tipo de gráfico "Comparativo" aparecerán todas las elecciones comparadas entre ellas, para poder comparar elecciones generales y autonómicas no se compara el número de escaños del partido sino el porcentaje de escaños obtenidos respecto al total de la elección.

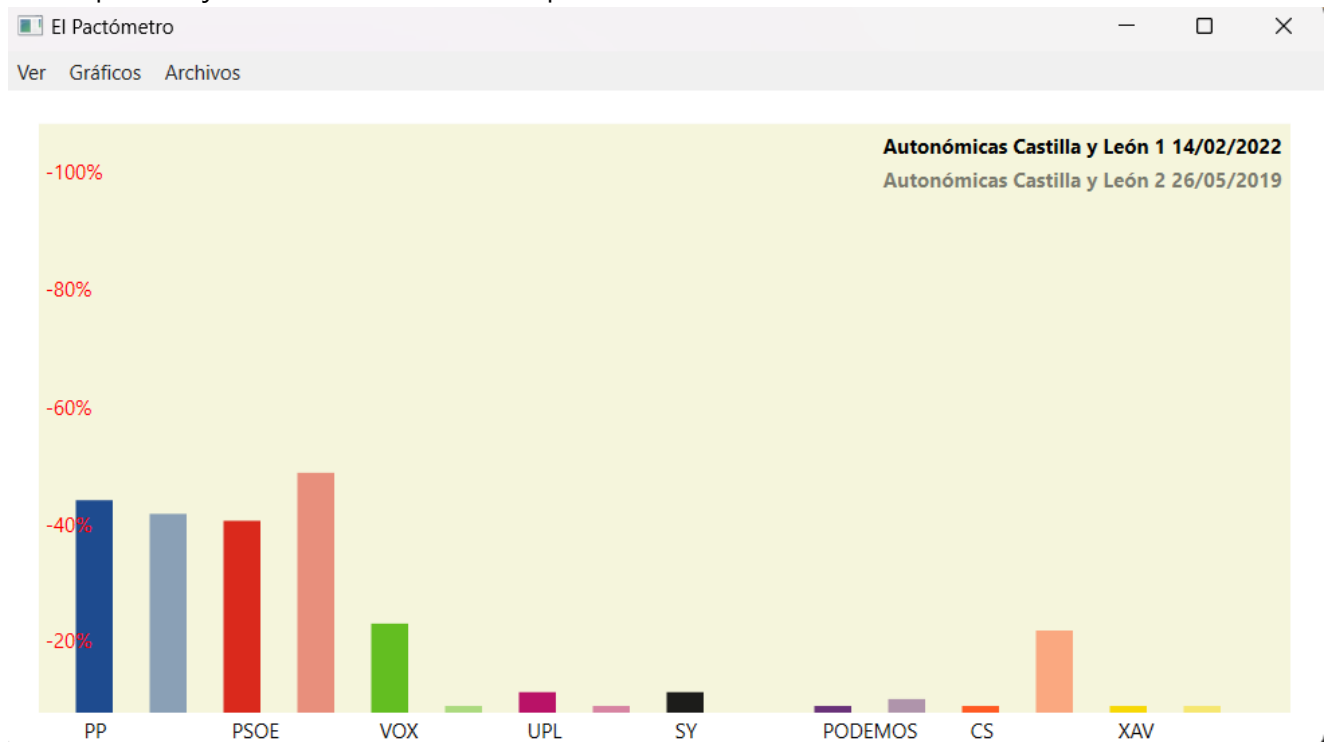
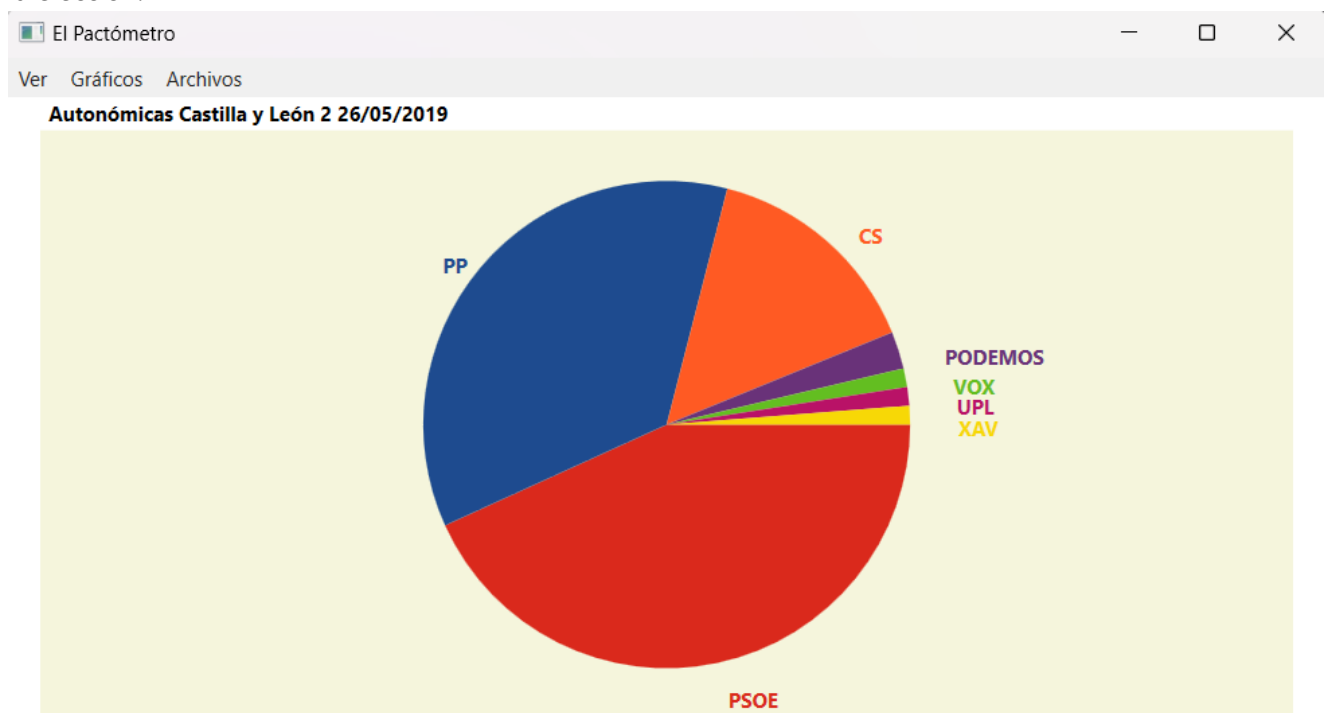


Gráfico circular

Al seleccionar el tipo de gráfico "Circular" aparecerán los partidos de la elección seleccionada formando un triángulo dentro de un círculo representando el porcentaje de escaños obtenidos respecto al total de la elección.



Añadir elección

En esta ventana podremos darle valores a la elección, si alguno de los campos está vacío, es incorrecto o la suma de escaños no sea correcta, se notificará para que el usuario pueda corregirlo y no creará la elección hasta que no se corrija.

Nueva Eleccion

Elección

Tipo

Nombre

Partidos

Nombre	Escaños	Color
--------	---------	-------

Fecha

Seleccione una fecha15

Partido

Nombre

Escaños

Color

Añadir

Añadir

Cancelar

Editar elección

Similar a la ventana de añadir elección, pero esta recibe los datos de la elección seleccionada para poderlos editar. Al igual que al añadir elección, si alguno de los campos está vacío, es incorrecto o la suma de escaños no sea correcta, se notificará para que el usuario pueda corregirlo y no editará la elección hasta que no se corrija.

Editar Eleccion

Elección

Tipo

Autonómicas

Lugar

Castilla y León

Nombre

1

Partidos

Nombre	Escaños	Color
PP	31	
PSOE	28	
VOX	13	

Fecha

14/02/202215

Partido

Nombre

Escaños

Color

Añadir

Editar

Cancelar

Borrar elección

Se le preguntará al usuario si está seguro de eliminar la elección, en caso afirmativo, esta se eliminará, si no, esta se mantendrá intacta.

Tablas

Nombre	Escaños	Mayoria	Fecha
Generales 1	350	176	7/23/2023 12:00:00 AM
Generales 2	350	176	11/10/2019 12:00:00 AM
Autonómicas Castilla y León 1	81	41	2/14/2022 12:00:00 AM
Autonómicas Castilla y León 2	81	41	5/26/2019 12:00:00 AM

+

✎

🗑

Nombre	Escaños	Color
PP	136	
PSOE	122	
VOX	33	
SUMAR	31	
ERC	7	
JUNTS	7	

Borrar eleccion

?

¿Quieres borrar la elección seleccionada?

Sí

No

Manual del programador

MainWindow

Existen dos clases principales en este trabajo, "Partido" con los atributos `nombre`, `escaños`, `color` y `posPactometro`, `posPactometro` estará instanciado a 2 (con posibilidad de editarlo en la gráfica del pactómetro) para que en primera instancia se coloquen en la segunda columna de ese tipo de gráfica. La clase "Elección" consta de los atributos `nombre`, `partidos`(que es un `ObservableCollection<Partido>`), `escaños`, `mayoría` y `fecha`

En la ventana principal se crea un `ObservableCollection<Eleccion>` que contendrá todas las elecciones que se creen y se añadirán al principio 4 elecciones predeterminadas por el enunciado gracias a la función `CargaDeDatos()`

Tenemos una variable global `Eleccion eleccionSeleccionada`; la cual almacenará la elección seleccionada en la ventana secundaria, Tablas.

También tenemos de manera global 4 variables booleanas `Normal = true`; `Pactometro = false`; `Comparativo = false`; `Circular = false`; así se sabrá que gráfico dibujar, se cambiarán al llamar a los eventos `Grafico_De_Barras(object sender, RoutedEventArgs e)`, `Pactómetro(object sender, RoutedEventArgs e)`, `Grafico_Comparativo(object sender, RoutedEventArgs e)` y `Grafico_Circular(object sender, RoutedEventArgs e)` que pondrán a true su variable correspondiente y a false las otras dos restantes además de redibujar el gráfico si la variable `eleccionSeleccionada` es distinta de null.

A la hora de dibujar los gráficos se recorre con un foreach la variable `eleccionSelector` por cada partido que crea un rectángulo y un label con el nombre del partido y lo añade al canvas de la siguiente manera:

```
int j = 2;
foreach (Partido partido in el.Partidos) {
    // RECTANGULO
    Rectangle r = new Rectangle();
    r.Height = (((altocanvas - 20) * partido.Escaños) / max));
    r.Width = anchocanvas / ((el.Partidos.Count + 1) * 2);

    // Convierte el nombre del color a un objeto Brush
    Brush colorBrush = (Brush)new
    BrushConverter().ConvertFromString(partido.Color);
    r.Fill = colorBrush;
    r.ToolTip = partido.Escaños == 1 ? $"{partido.Nombre} - {partido.Escaños}
    escaño" : $"{partido.Nombre} - {partido.Escaños} escaños";

    CanvaFondo.Children.Add(r);

    Canvas.SetBottom(r, 0);
    Canvas.SetLeft(r, j * anchocanvas / ((el.Partidos.Count + 1) * 2));
```



```

// TEXTO
Label l = new Label();
l.Content = partido.Nombre.Length >= 10 ? partido.Nombre.Substring(0, 10) :
partido.Nombre.ToString();
l.Foreground = colorBrush;
l.FontWeight = FontWeights.Bold;

CanvaFondo.Children.Add(l);

Canvas.SetBottom(l, -25);
Canvas.SetLeft(l, (j * anchocanva / ((el.Partidos.Count + 1) * 2)) -5);

j += 2;
}

```

En los demás tipos de gráficos se realizan diferentes cambios para adaptarlo a las necesidades, pero siguen la misma mecánica de creación de los rectángulos en un foreach.

La ventana principal también contiene un evento Closing que cierra todas las ventanas en caso de que esta se cierre.

```

private void MainWindow_Closing(object sender,
System.ComponentModel.CancelEventArgs e) {
    // Obtener todas las ventanas abiertas
    foreach (Window window in App.Current.Windows) {
        if (window != this) {
            // Manejar el evento Closing de la ventana t por separado
            if (window is Tablas t) {
                t.Closing += TablasClosingMain;
                t.Close();
            } else {
                window.Close();
            }
        }
    }
    e.Cancel = false;
}

```

En el MainWindow también se encuentra el evento de exportar las elecciones.

```

private void ExportarCSV_Click(object sender, RoutedEventArgs e) {
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Archivos CSV (*.csv)|*.csv";
    saveFileDialog.Title = "Selecciona donde guardar el archivo CSV";

    if (saveFileDialog.ShowDialog() == true) {
        string filePath = saveFileDialog.FileName;
        try {

```

```

        using (var writer = new StreamWriter(filePath)) {
            foreach (var eleccion in listaElecciones) {
                // Se asume que el formato de las fechas se puede convertir
                directamente a una cadena
                string partidosComoCadena =
                ObtenerPartidosComoCadena(eleccion.Partidos);
                writer.WriteLine($"{eleccion.Nombre},{eleccion.Escaños},
                {eleccion.Fecha},{eleccion.Mayoria},{partidosComoCadena}");
            }
        }

        MessageBox.Show("Datos exportados correctamente.", "Exportar CSV",
        MessageBoxButton.OK, MessageBoxImage.Information);
    } catch (Exception ex) {
        MessageBox.Show($"Error al exportar datos: {ex.Message}", "Error",
        MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

private string ObtenerPartidosComoCadena(ObservableCollection<Partido>
partidos) {
    if (partidos == null || partidos.Count == 0) {
        return string.Empty;
    }

    //Utiliza StringBuilder para poder hacer el return de un string
    StringBuilder sb = new StringBuilder();

    foreach (var partido in partidos) {
        sb.Append($"{partido.Nombre},{partido.Escaños},{partido.Color},");
    }

    // Elimina la última ","
    sb.Length--;

    return sb.ToString();
}

```

En el MainWindow también se encuentra el evento de importar las elecciones, eliminando las ya existentes.

```

private void ImportarCSV_Click(object sender, RoutedEventArgs e) {
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Archivos CSV (*.csv)|*.csv";
    openFileDialog.Title = "Selecciona un archivo CSV";

    if (openFileDialog.ShowDialog() == true) {
        string filePath = openFileDialog.FileName;
    }
}

```

```

        try {
            // Eliminar todas las elecciones cuando la apertura del fichero es
correcta
            listaElecciones.Clear();
            var lines = File.ReadLines(filePath);

            foreach (var line in lines) {
                string[] values = line.Split(',');

                if (values.Length >= 4) {
                    Eleccion eleccion = new Eleccion(values[0], new
ObservableCollection<Partido>(), DateTime.Parse(values[2]));

                    for (int i = 4; i < values.Length; i += 3) {
                        Partido partido = new Partido(values[i],
int.Parse(values[i + 1]), values[i + 2]);
                        eleccion.Partidos.Add(partido);
                    }
                    listaElecciones.Add(eleccion);
                } else {
                    MessageBox.Show("Formato de línea incorrecto en el archivo
CSV", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
                }
            }
        } catch (Exception ex) {
            MessageBox.Show($"Error al importar el archivo CSV: {ex.Message}",
"Error", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }
}

```

En el MainWindow también se encuentra el evento de añadir las elecciones, a diferencia de importar, las elecciones existentes no se eliminan.

```

private void AñadirCSV_Click(object sender, RoutedEventArgs e) {
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Archivos CSV (*.csv)|*.csv";
    openFileDialog.Title = "Selecciona un archivo CSV";

    if (openFileDialog.ShowDialog() == true) {
        string filePath = openFileDialog.FileName;

        try {
            var lines = File.ReadLines(filePath);

            foreach (var line in lines) {
                string[] values = line.Split(',');

                if (values.Length >= 4) {
                    Eleccion eleccion = new Eleccion(values[0], new

```

```

ObservableCollection<Partido>(), DateTime.Parse(values[2]));

        for (int i = 4; i < values.Length; i += 3) {
            Partido partido = new Partido(values[i],
int.Parse(values[i + 1]), values[i + 2]);
            eleccion.Partidos.Add(partido);
        }
        listaElecciones.Add(eleccion);
    } else {
        MessageBox.Show("Formato de línea incorrecto en el archivo
CSV", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
} catch (Exception ex) {
    MessageBox.Show($"Error al importar el archivo CSV: {ex.Message}",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}

```

Tablas

Esta ventana tiene un evento y una variable global `public event EventHandler<EleccionSeleccionadaEventArgs> EleccionSeleccionada;` `ObservableCollection<Eleccion> listaElecciones;` la variable es la lista de todas las elecciones, se la pasa la ventana principal al crear esta y la instancia de la siguiente manera

```

public Tablas(ObservableCollection<Eleccion> listaElecciones) {
    InitializeComponent();
    this.listaElecciones = listaElecciones;
    eleccionesListView.ItemsSource = listaElecciones;
}

```

el evento detecta si se ha seleccionado una elección para poder editarla, borrar o que la ventana principal dibuje el gráfico.

```

public class EleccionSeleccionadaEventArgs : EventArgs {
    public Eleccion EleccionSeleccionada { get; }

    public EleccionSeleccionadaEventArgs(Eleccion eleccionSeleccionada) {
        EleccionSeleccionada = eleccionSeleccionada;
    }
}

```

El botón "Add" crea una nueva ventana la cual creará y añadirá una nueva elección a la lista. El botón "Edit" crea una nueva ventana y edita la elección seleccionada en caso de existir (que esté seleccionada). El

botón "Delete" comprueba si hay una elección seleccionada y en caso afirmativo, pregunta al usuario si desea eliminarla, si la respuesta es afirmativa, se eliminará de la lista.

```
private void Add_Eleccion(object sender, RoutedEventArgs e) {
    NewEleccion nEl = new NewEleccion();
    nEl.ShowDialog();
    if (nEl.DialogResult == true) {
        listaElecciones.Add(nEl.AddElecciones);
    }
}

private void Del_Eleccion(object sender, RoutedEventArgs e) {
    if (eleccionesListView.SelectedItem is Eleccion selectedEleccion) {
        if (MessageBox.Show("¿Quieres borrar la elección seleccionada?",
            "Borrar eleccion", MessageBoxButton.YesNo, MessageBoxImage.Question) ==
            MessageBoxResult.Yes) {
            listaElecciones.Remove(selectedEleccion);
        }
    } else {
        MessageBox.Show("No se ha seleccionado ninguna eleccion", "",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

private void Editar_Eleccion(object sender, RoutedEventArgs e) {
    if (eleccionesListView.SelectedItem is Eleccion selectedEleccion) {
        if (MessageBox.Show("¿Quieres editar la elección seleccionada?",
            "Editar elección", MessageBoxButton.YesNo, MessageBoxImage.Question) ==
            MessageBoxResult.Yes) {
            EditEleccion nEl = new EditEleccion(selectedEleccion);
            nEl.ShowDialog();

            // Verificar si la edición fue exitosa en la ventana de edición
            if (nEl.DialogResult == true) {
                // Realizar acciones adicionales si es necesario
                MessageBox.Show("Edición exitosa", "", MessageBoxButton.OK,
                    MessageBoxImage.Information);
            } else {
                // La edición no fue exitosa o se canceló
            }
        }
    } else {
        MessageBox.Show("No se ha seleccionado ninguna elección", "",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
```

EditEleccion y NewEleccion

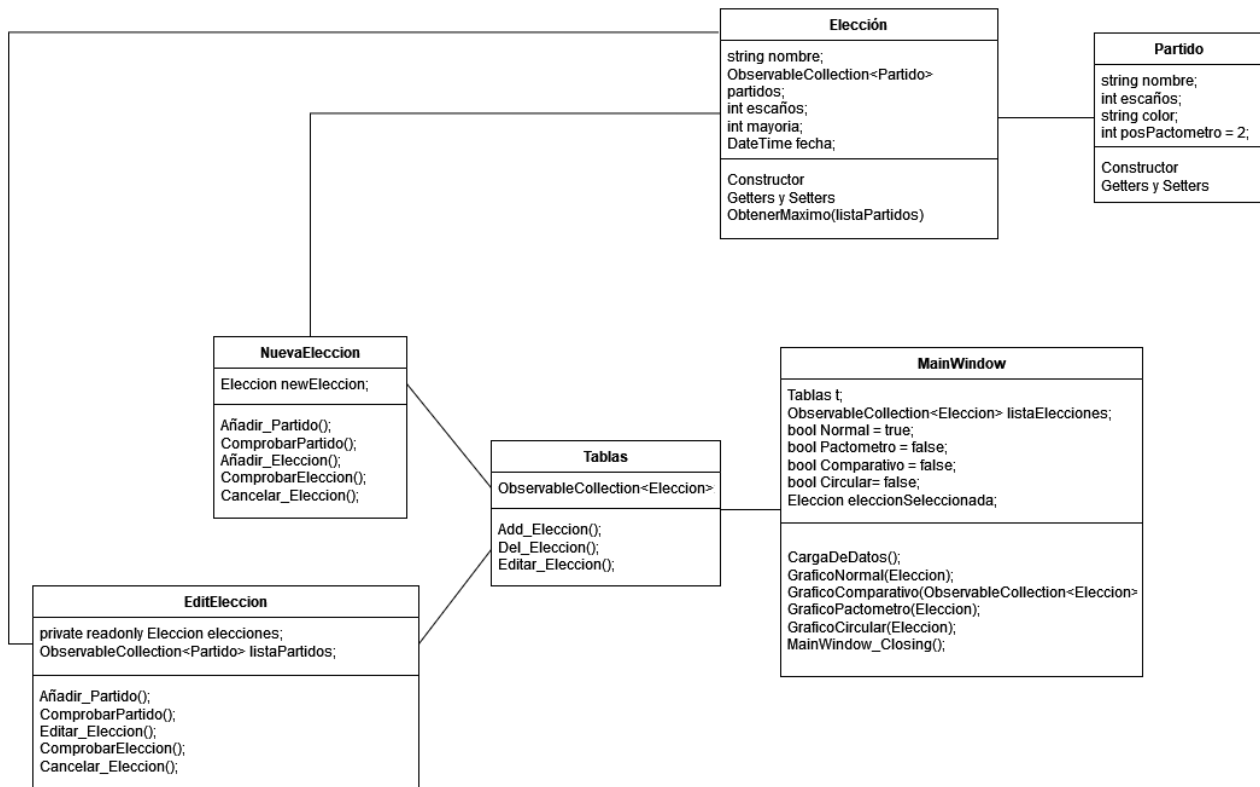
Comprueba si todos los datos están introducidos y si estos son válidos, en caso negativo se le notificará al usuario

```
//COMPROBAR ELECCION
private bool ComprobarEleccion() {
    bool comp = true;
    int sum = 0;
    if (TipoComboBox.SelectedItem != null) {
        ComboBoxItem selectedTipoItem =
(ComboBoxItem)TipoComboBox.SelectedItem;
        string tipo = selectedTipoItem.Content.ToString();
    } else {
        TipoComboBox.BorderBrush = Brushes.Red;
        comp = false;
    }
    foreach (Partido partido in listaPartidos) {
        sum += partido.Escaños;
    }
    if ((TipoComboBox.SelectedItem as ComboBoxItem)?.Content.ToString() ==
"Generales") {
        if (sum != 350) {
            MessageBox.Show("Los partidos no suman 350 escaños", "",
MessageBoxButton.OK, MessageBoxImage.Error);
            comp = false;
        }
    } else if ((TipoComboBox.SelectedItem as ComboBoxItem)?.Content.ToString()
== "Autonómicas") {
        if (sum != 81) {
            MessageBox.Show("Los partidos no suman 81 escaños", "",
MessageBoxButton.OK, MessageBoxImage.Error);
            comp = false;
        }
    }
    if (String.IsNullOrEmpty(NombreEleccion.Text)) {
        //Error
        NombreEleccion.BorderBrush = Brushes.Red;
        comp = false;
    }
    if (String.IsNullOrEmpty(introducirfecha.Text)) {
        introducirfecha.BorderBrush = Brushes.Red;
        comp = false;
    }
    return comp;
}
```

Si esta comprobación obtiene un resultado positivo se creará o editará la elección

```
private void Añadir/Editar_Eleccion(object sender, RoutedEventArgs e) {  
    if (ComprobarEleccion() == true) {  
        string tipo = (TipoComboBox.SelectedItem as  
ComboBoxItem)?.Content.ToString();  
        string nombre = tipo + " " + NombreEleccion.Text;  
        newEleccion = new Eleccion(nombre, listaPartidos,  
introducirfecha.SelectedDate.Value.Date);  
        DialogResult = true;  
    }  
}
```

Diagrama de clases



Bibliografía

Funcionamiento ComboBox --> <https://wpf-tutorial.com/es/73/controles-de-lista/the-combobox-control/>

Búsqueda de fallos, ComboBox con todos los colores automáticamente y regiones gráfico circular --> <https://chat.openai.com/>

Cuadros de mensaje --> <https://learn.microsoft.com/es-es/dotnet/desktop/wpf/windows/how-to-open-message-box?view=netdesktop-8.0>

Propiedades del Label --> <https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.label?view=windowsdesktop-8.0>

Colores de partidos predefinidos --> <https://htmlcolorcodes.com/es/>
<https://learn.microsoft.com/es-es/windows/powertoys/>
<https://learn.microsoft.com/es-es/dotnet/api/system.windows.media.brushes?view=windowsdesktop-8.0>
<https://learn.microsoft.com/es-es/dotnet/api/system.windows.media.brush?view=windowsdesktop-8.0>

SizeChanged --> <https://stackoverflow.com/questions/32668707/c-sharp-wpf-sizechanged-event-doesnt-update-width-and-height-when-maximized>

Diagrama de clases --> <https://www.youtube.com/watch?v=PBfRnXEdXCg>

Closing --> <https://learn.microsoft.com/es-es/dotnet/api/system.windows.window.closing?view=windowsdesktop-8.0>

Creación de gráficos (no utilizado al final) --> <https://www.youtube.com/watch?v=itwHg2l3gTw>

SaveFileDialog --> <https://wpf-tutorial.com/dialogs/the-savefiledialog/>
<https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-save-files-using-the-savefiledialog-component?view=netframeworkdesktop-4.8>
<https://www.youtube.com/watch?v=ldOeGUto6BM>

OpenFileDialog --> <https://wpf-tutorial.com/dialogs/the-openfiledialog/>
<https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-open-files-using-the-openfiledialog-component?view=netframeworkdesktop-4.8>

StringBuilder --> <https://github.com/microsoft/microsoft-ui-xaml/issues/4774>
<https://learn.microsoft.com/es-es/dotnet/standard/base-types/stringbuilder>

Canvas --> <https://learn.microsoft.com/es-es/uwp/api/windows.ui.xaml.controls.canvas?view=winrt-22621>
<https://wpf-tutorial.com/es/24/paneles/el-control-canvas/>

Rectángulos --> <https://learn.microsoft.com/es-es/dotnet/desktop/wpf/graphics-multimedia/how-to-draw-a-rectangle?view=netframeworkdesktop-4.8>

Imagen en un botón --> <https://learn.microsoft.com/es-es/dotnet/desktop/wpf/controls/how-to-create-a-button-that-has-an-image?view=netframeworkdesktop-4.8>

ColorPicker (no utilizado al final) --> https://www.youtube.com/watch?v=_YGihw8bAoc
<https://learn.microsoft.com/es-es/uwp/api/windows.ui.xaml.controls.colorpicker?view=winrt-22621>

Operador lambda --> <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/lambda-operator>

ToolTip --> <https://wpf-tutorial.com/es/21/conceptos-sobre-controles/control-tooltips/>
<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/controls/tooltip-overview?view=netframeworkdesktop-4.8>

Google Icons --> [https://fonts.google.com/icons?](https://fonts.google.com/icons?selected=Material+Symbols+Outlined:add_circle:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=add)
[https://fonts.google.com/icons?](https://fonts.google.com/icons?selected=Material+Symbols+Outlined:edit:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=edit)
[https://fonts.google.com/icons?](https://fonts.google.com/icons?selected=Material+Symbols+Outlined:delete:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=delete)
[https://fonts.google.com/icons?](https://fonts.google.com/icons?selected=Material+Symbols+Outlined:delete:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=delete)

Regex --> <https://stackoverflow.com/questions/15775938/wpf-textbox-validation-with-regex>

Gráfica circular --> https://www.youtube.com/watch?v=-_540TruS0g&t=249s

Escaños CCAA --> https://es.wikipedia.org/wiki/Anexo:Composici%C3%B3n_de_los_Parlamentos_Auton%C3%B3micos