## 3.5. Controles de selección

Los controles de selección son mucho menos permisivos que los controles de edición, pues permiten seleccionar entre una o varias opciones de entre una **lista finita** de elementos. No obstante, este comportamiento puede tener un gran valor, por otro lado, cuando se pretende acelerar la información de un formulario reduciendo las posibilidades de error en las operaciones de categorización o de edición repetitiva.

### 3.5.1. RadioButton

El control **RadioButton** representa una opción de selección **exclusiva** que puede marcarse o no. Es posible recuperar este estado mediante la propiedad **IsCheched**. Cuando el control está marcado, es imposible desmarcarlo haciendo clic de nuevo en él. Para ello, es preciso marcar algún otro control RadioButton que pertenezca al mismo grupo. El grupo al que pertenece el control se define mediante la propiedad **GroupName**.

Title="MainWindow" Height="350" Width="400">

<Grid>

<StackPanel Orientation="Vertical" Margin="40">

<RadioButton GroupName="Opción" Content="Dam 1 diurno" Margin="10"/>

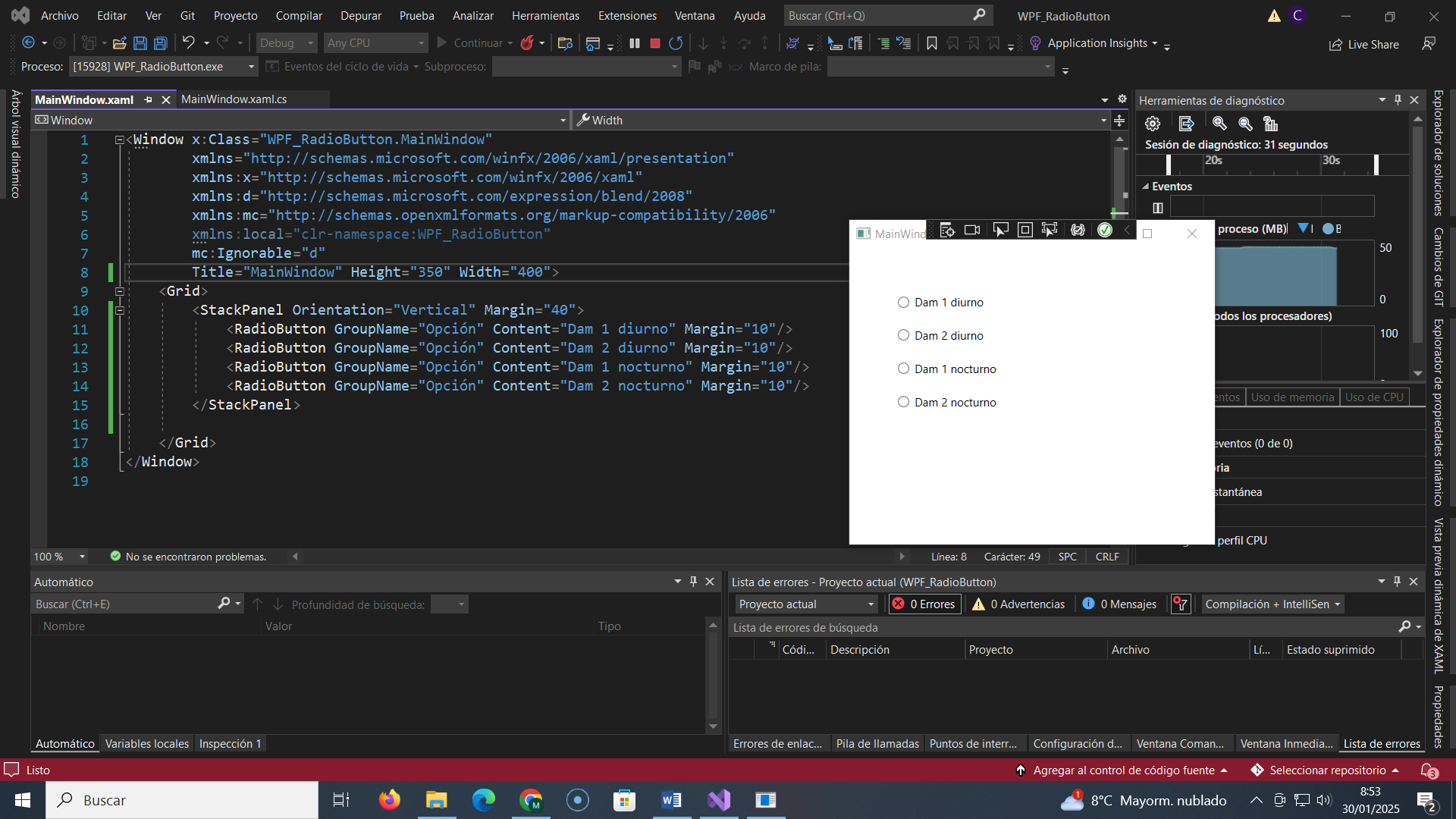
<RadioButton GroupName="Opción" Content="Dam 2 diurno" Margin="10"/>

<RadioButton GroupName="Opción" Content="Dam 1 nocturno" Margin="10"/>

<RadioButton GroupName="Opción" Content="Dam 2 nocturno" Margin="10"/>

</StackPanel>

</Grid>



Este control se utiliza con poca frecuencia cuando se muestran más de **cinco opciones**. En este caso, es más adecuado utilizar el control **ComboBox**, mucho más práctico que una pila de RadioButton, y mucho más legible.

### 3.5.2 CheckBox

El control **CheckBox** permite, como el RadioButton, seleccionar o no un valor, pero su modo de selección no es exclusivo: es posible marcar varios CheckBox al mismo tiempo. El estado de cada CheckBox se define mediante el valor de su propiedad **IsChecked**.

<Grid>

<StackPanel Orientation="Vertical">

<CheckBox Content="Opción nº 1"/>

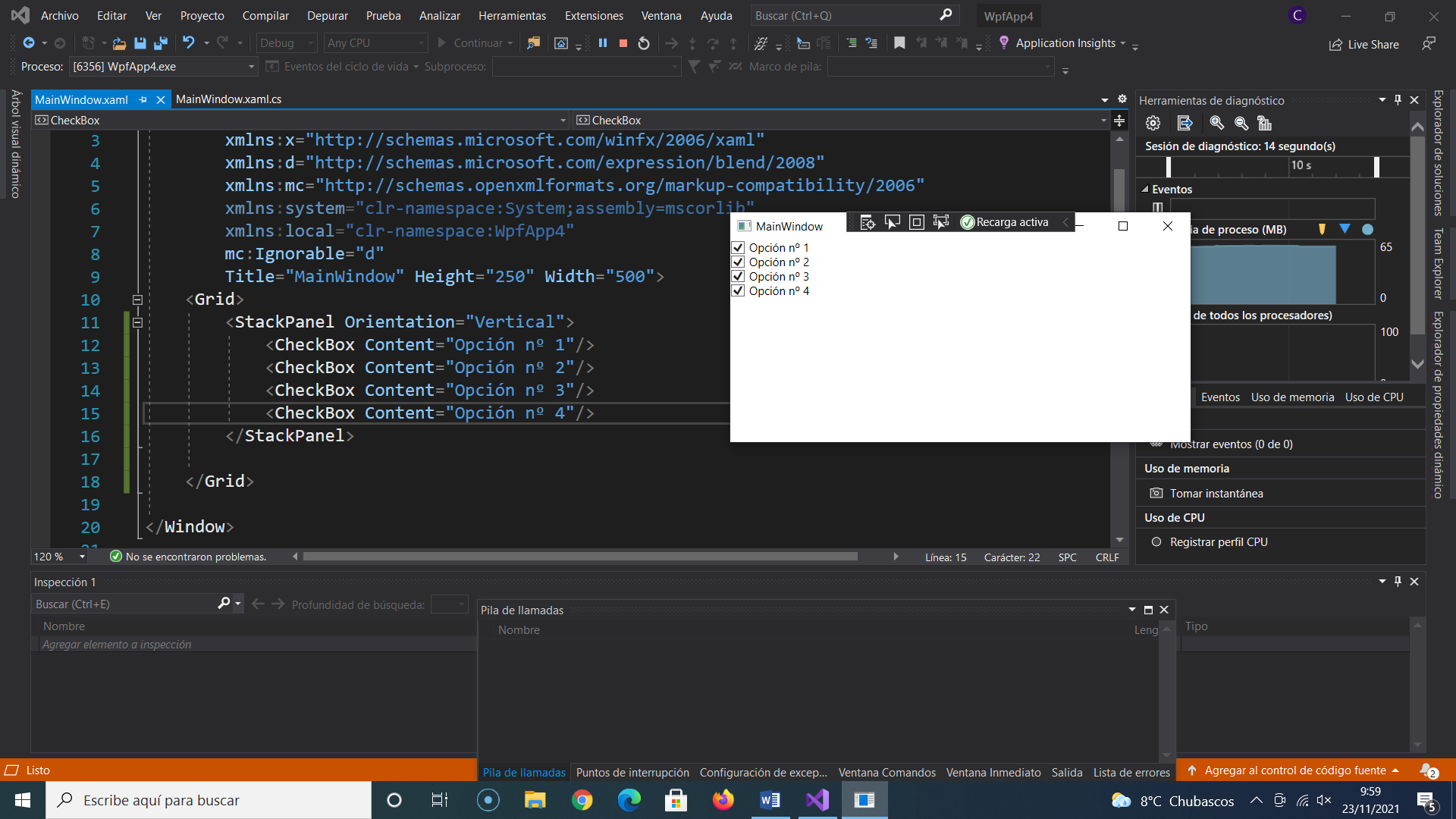
<CheckBox Content="Opción nº 2"/>

<CheckBox Content="Opción nº 3"/>

<CheckBox Content="Opción nº 4"/>

</StackPanel>

</Grid>



### 3.5.3. ComboBox

Este control proporciona una lista desplegable de elementos donde solamente es posible seleccionar uno. Este control resulta práctico para realizar selecciones únicas debido a lo sencillo que resulta su uso y su tamaño mínimo.

Este control deriva del tipo ItemsControl, lo cual implica que su uso sea algo similar. Los datos que se muestran se proveen mediante propiedades Items o ItemsSource. Esta segunda propiedad se alimenta mediante una expresión de tipo binding.

<ComboBox>

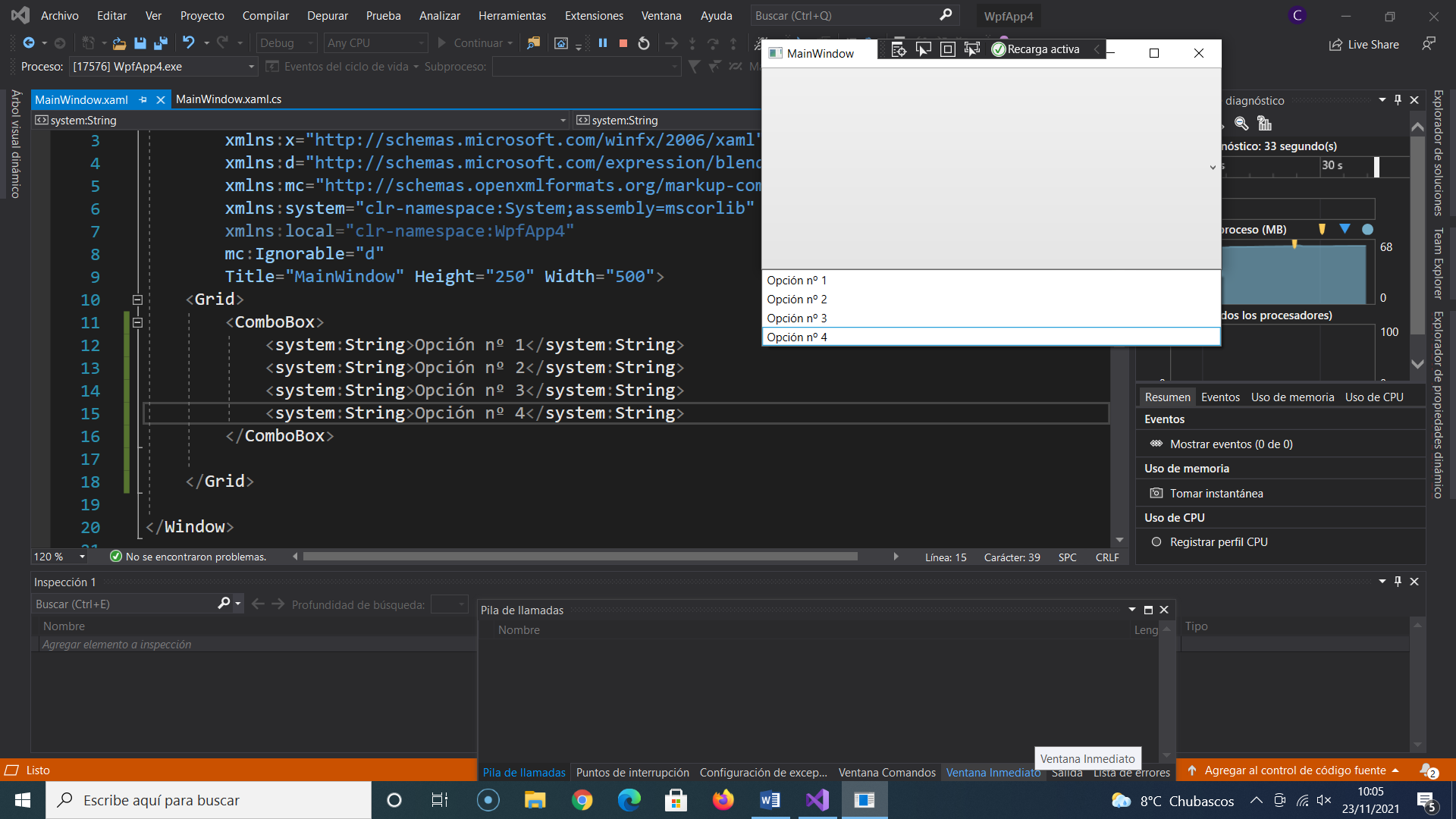
<system:String>Opción nº 1</system:String>

<system:String>Opción nº 2</system:String>

<system:String>Opción nº 3</system:String>

<system:String>Opción nº 4</system:String>

</ComboBox>



### 3.5.4. ListBox

El control ListBox muestra una lista de elementos en la que es posible seleccionar uno o varios. El modo de selección se define mediante el valor de la propiedad SelectionMode del control: Single, que permite seleccionar un único valor, o bien Multiple y Extended, que permiten realizar una selección múltiple. La diferencia entre estos dos últimos modos es la siguiente:

* Multiple: la selección o deselección de un elemento se realiza mediante un clic.
* Extended: es necesario mantener pulsada la tecla [Ctrl] durante las operaciones de selección y deselección.

La lista de datos contenidos en el control se provee mediante sus propiedades Items o ItemsSource, de la misma manera que con el tipo ItemsControl. Como con los demás controles que permiten mostrar colecciones, es posible personalizar la representación gráfica mediante la propiedad ItemTemplate con un objeto de tipo DataTemplate.

<ListBox SelectionMode="Multiple" Height="100" Width="150">

<ListBox.Items>

<system:String>Opción nº 1</system:String>

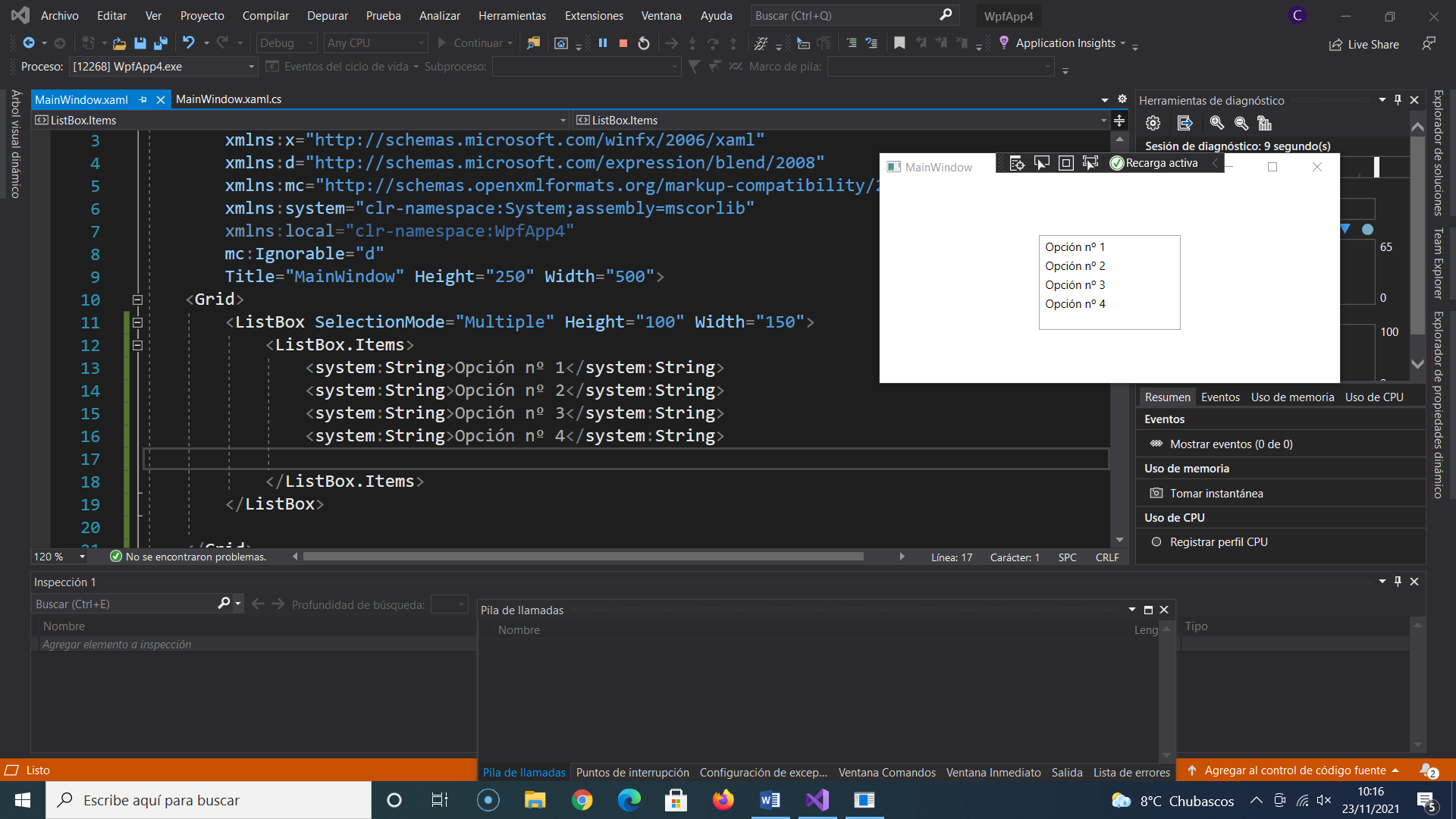
<system:String>Opción nº 2</system:String>

<system:String>Opción nº 3</system:String>

<system:String>Opción nº 4</system:String>

</ListBox.Items>

</ListBox>



### 3.5.5. ListView

El control ListView que deriva del tipo ItemsControl, permite mostrar una lista de elementos en una cuadrícula. Esta se derive mediante la propiedad View de tipo ViewBase. La única clase del framework .NET que hereda de este tipo es la clase GridView, por lo que será un objeto de este tipo el que pasaremos a la propiedad View.

Los elementos que se muestran en el control se pasan, como ocurre con el tipo ItemsControl, mediante las propiedades Items o ItemsSource

El tipo de datos utilizado en este ejemplo se define de la siguiente manera:

### 3.5.6. TreeView

El control TreeView ofrece la posibilidad de mostrar datos de manera jerárquica. Cada uno de los objetos puede tener uno o varios hijos que pueden, ellos mismos, ser padres de otros nodos, y así sucesivamente.

WPF permite describir esta jerarquía de manera sencilla y clara. Basta, en efecto, con considerar que cada nodo de nuestro árbol es un objeto de tipo TreeViewItem que puede contener uno o varios objetos del mismo tipo.

El texto correspondiente a cada uno de los nodos se define mediante la propiedad Header de cada objeto TreeViewItem. Esta propiedad es de tipo object, lo que significa, indirectamente, que no es obligatorio darle valor mediante una cadena de caracteres, sino que puede contener también, imágenes, texto, una opción a marcar, etc.

Ejemplo:

<TreeView>

<TreeViewItem Header="América">

<TreeViewItem Header="América del Norte">

<TreeViewItem Header="Canadá"/>

<TreeViewItem Header="Estados Unidos"/>

</TreeViewItem>

<TreeViewItem Header="América Central">

<TreeViewItem Header="Méjico"/>

<TreeViewItem Header="Panámá"/>

<TreeViewItem Header="Honduras"/>

</TreeViewItem>

</TreeViewItem>

<TreeViewItem Header="Europa">

<TreeViewItem Header="Unión Europea">

<TreeViewItem Header="Zona Euro">

<TreeViewItem Header="España"/>

<TreeViewItem Header="Francia"/>

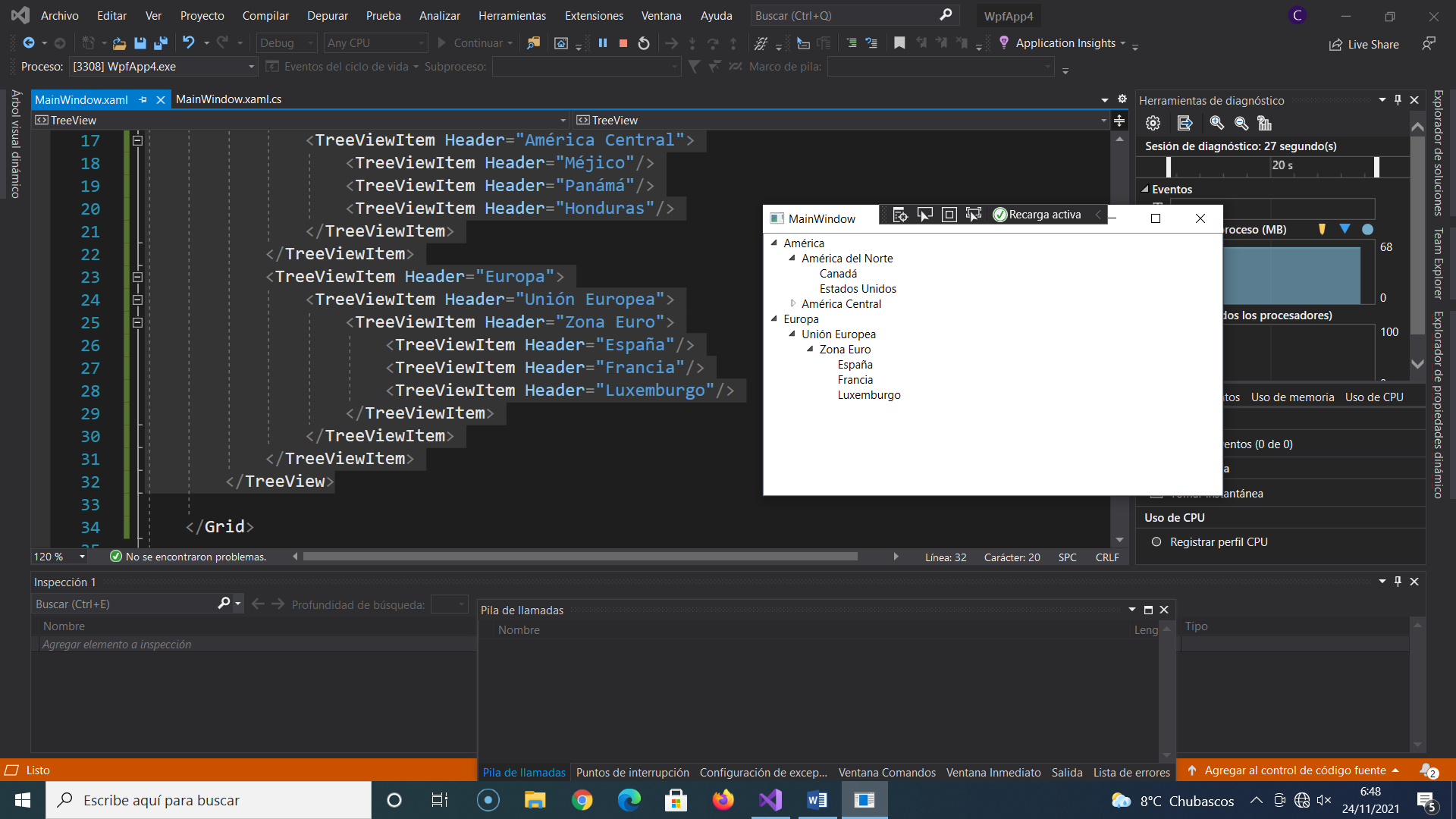
<TreeViewItem Header="Luxemburgo"/>

</TreeViewItem>

</TreeViewItem>

</TreeViewItem>

</TreeView>



3.5.7. Slider

Este control permite seleccionar un valor numérico comprendido entre dos extremos. El usuario realiza su selección deslizando un cursor a lo largo de una línea. Este tipo de controles se utiliza, a menudo, en aplicaciones multimedia para gestionar el volumen del sonido o la situación de la lectura de un archivo de audio o de vídeo.

Los extremos mínimo y máximo se definen asignando valor a las propiedades Minimun y Maximun, mientras que el intervalo entre los valores seleccionables se define mediante la propiedad smallChange. Si esta propiedad no posee ningún valor, es posible seleccionar cualquier valor decimal soportado por el tipo double.

La propiedad LargeChange permite definir el número de unidades que debe recorrer el cursor hacia adelante o hacia atrás cuando el usuario haga clic en la línea, pero no deslice el cursor.

### 3.5.8. Calendar

Este control permite navegar de manera visual en un calendario para seleccionar una fecha, de manera similar al calendario integrado en la barra de tareas de Windows.

La fechaseleccionada se representa mediante la propiedad SelectedDate.

### 3.5.9. DatePicker

La selección de una fecha resulta particularmente complicada de implementar, y por este motivo el framework .NET provee el control DatePicker. Este control está compuesto por tres partes: un campo que permite introducir texto, un calendario que puede estar visible u oculto y por último un botón que permite mostrar el calendario y, por tanto, introducir de manera directa una fecha y confirmar que los datos que recibe se corresponden con una fecha válida. La selección de fecha mediante el calendario permite evitar cualquier problema.

La propiedad SelectedDate permite acceder en lectura y escritura al valor seleccionado o introducido en el control.