

# VISUAL STUDIO 2019

---

HELLO WORLD

# Conociendo la interfaz

---

Todo el que empieza a aprender un lenguaje de programación empieza con el “Hola mundo”.

No vamos a ser menos, vamos a estudiar cómo se hace en el entorno de programación Visual Studio en modo Consola y modo Gráfico.

# Hello World (Módo Gráfico)

---




Cuando cree una aplicación en Visual Studio, primero creará un proyecto y una solución. Para este ejemplo, creará un proyecto de Windows Formtion (WPF).

1. Abra Visual Studio 2019.
2. En la pantalla **Crear un nuevo proyecto**, busque "WPF", elija **Aplicación de WPF (.NET Framework)** y haga clic en **Siguiente**.

# Visual Studio 2019

## Crear un proyecto

### Plantillas de proyecto recientes


-  WPF App (.NET) C#
-  Windows Forms App (.NET) Visual Basic
-  Aplicación de consola (.NET Core) C#


Borrar todo


Todos los lenguajes


Todas las plataformas

Todos los tipos de proy...

 **WPF App (.NET)**  
Aplicación cliente de Windows Presentation Foundation.  
C# XAML Windows Escritorio

 **WPF Custom Control Library (.NET)**  
Biblioteca de controles personalizados de Windows Presentation Foundation.  
C# XAML Windows Escritorio Biblioteca

 **WPF User Control Library (.NET)**  
Biblioteca de controles de usuario de Windows Presentation Foundation.  
C# XAML Windows Escritorio Biblioteca

 **WPF Library (.NET)**  
Aplicación cliente de Windows Presentation Foundation.  
C# XAML Windows Escritorio Biblioteca

¿No encuentra lo que busca?

[Instalar más herramientas y características](#)

Atrás

Siguiente

# Visual Studio 2019

---

En la pantalla siguiente, asígnele al proyecto el nombre **HelloWPFApp** y haga clic en **Crear**.

Visual Studio crea el proyecto HelloWPFApp y la solución, y el **Explorador de soluciones** muestra los distintos archivos. **WPF Designer** muestra una vista de diseño y una vista XAML de *MainWindow.xaml* en una vista en dos paneles. Puede deslizar el divisor para mostrar más o menos de cualquiera de las vistas. Puede elegir ver solo la vista visual o solo la vista XAML. Los elementos siguientes aparecen en el **Explorador de soluciones**:

Explorador de soluciones



Buscar en Explorador de soluciones (Ctrl+`)



Solución "HelloWorldSemi" (1 de 1 proyecto)



**C# HelloWorldSemi**



Dependencias



App.xaml

C#

AssemblyInfo.cs



MainWindow.xaml



C#

MainWindow.xaml.cs

# Visual Studio 2019

---

Después de crear el proyecto, puede personalizarlo. Mediante la ventana Propiedades , (que se encuentra en el menú Ver ), puede mostrar y cambiar las opciones de los elementos de proyecto, controles y otros elementos de una aplicación.

# Cambiar el nombre de MainWindow.xaml

---

Asignemos a MainWindow un nombre más específico. En el Explorador de soluciones, haga clic con el botón derecho en MainWindow.xaml y seleccione Cambiar nombre. Cambie el nombre del archivo por Greetings.xaml



# Diseñar la interfaz de usuario (IU)

---

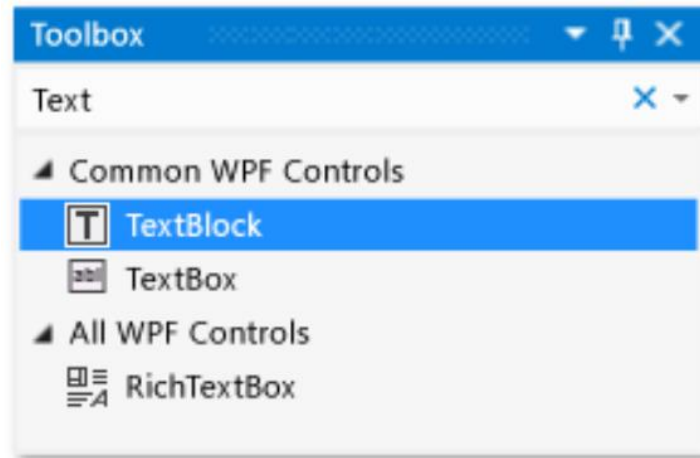
Si el diseñador no está abierto, seleccione Greetings.xaml en el Explorador de soluciones y presione Mayús+F7 para abrir el diseñador.

Agregaremos tres tipos de controles a esta aplicación: un control TextBlock, dos controles RadioButton y un control Button.

# Agregar un control TextBlock

---

1. Presione **Ctrl+Q** para activar el cuadro de búsqueda y escriba **cuadro de herramientas**. Elija **Ver > Cuadro de herramientas** en la lista de resultados.
2. En el **Cuadro de herramientas**, expanda el nodo **Controles WPF comunes** para ver el control TextBlock.

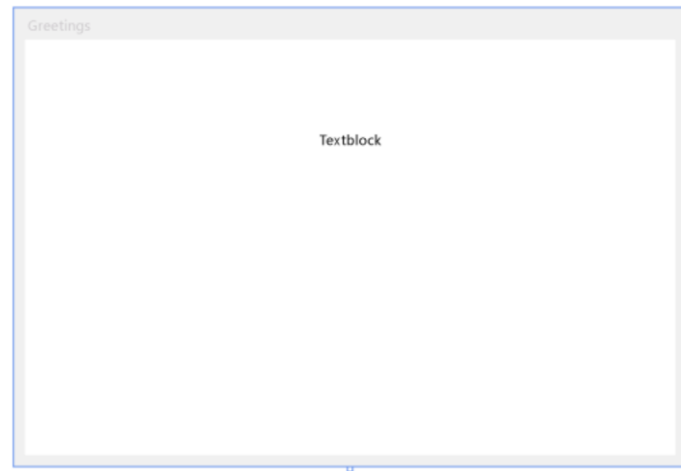


# Agregar un control TextBlock

---

3. Agregue un control TextBlock a la superficie de diseño. Para ello, elija el elemento **TextBlock** y arrástrelo a la ventana en la superficie de diseño. Centre el control cerca de la parte superior de la ventana. En Visual Studio 2019 y versiones posteriores, puede usar las guías de color rojo para centrar el control.

La ventana debería ser similar a la siguiente ilustración:



# Agregar un control TextBlock

---

El marcado XAML debe tener un aspecto similar al siguiente ejemplo:

XAML

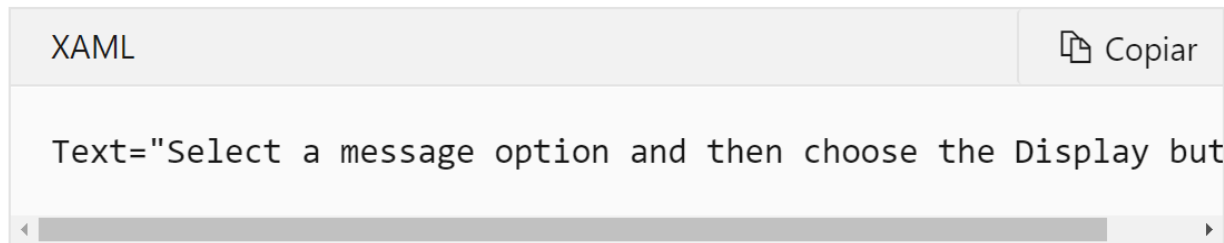
Copiar

```
<TextBlock HorizontalAlignment="Left" Margin="381,100,0,0" TextWrapping="Wrap" Text="TextBlock" FontSize="24" Foreground="Black"/>
```

# Personalizar el texto en el bloque de texto

---

1. En la vista XAML, busque el marcado de TextBlock y cambie el atributo de texto:

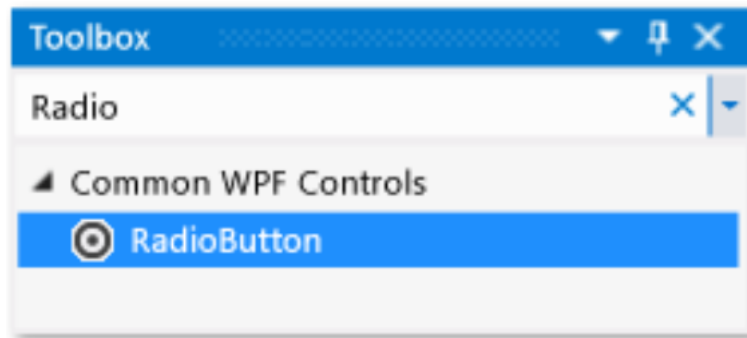


2. Vuelva a centrar TextBlock si es necesario y guarde los cambios. Para ello, presione Ctrl+S o use el elemento de menú Archivo.
3. Después, agregará dos controles RadioButton al formulario.

# Agregar botones de radio

---

1. En el Cuadro de herramientas, busque el control RadioButton.

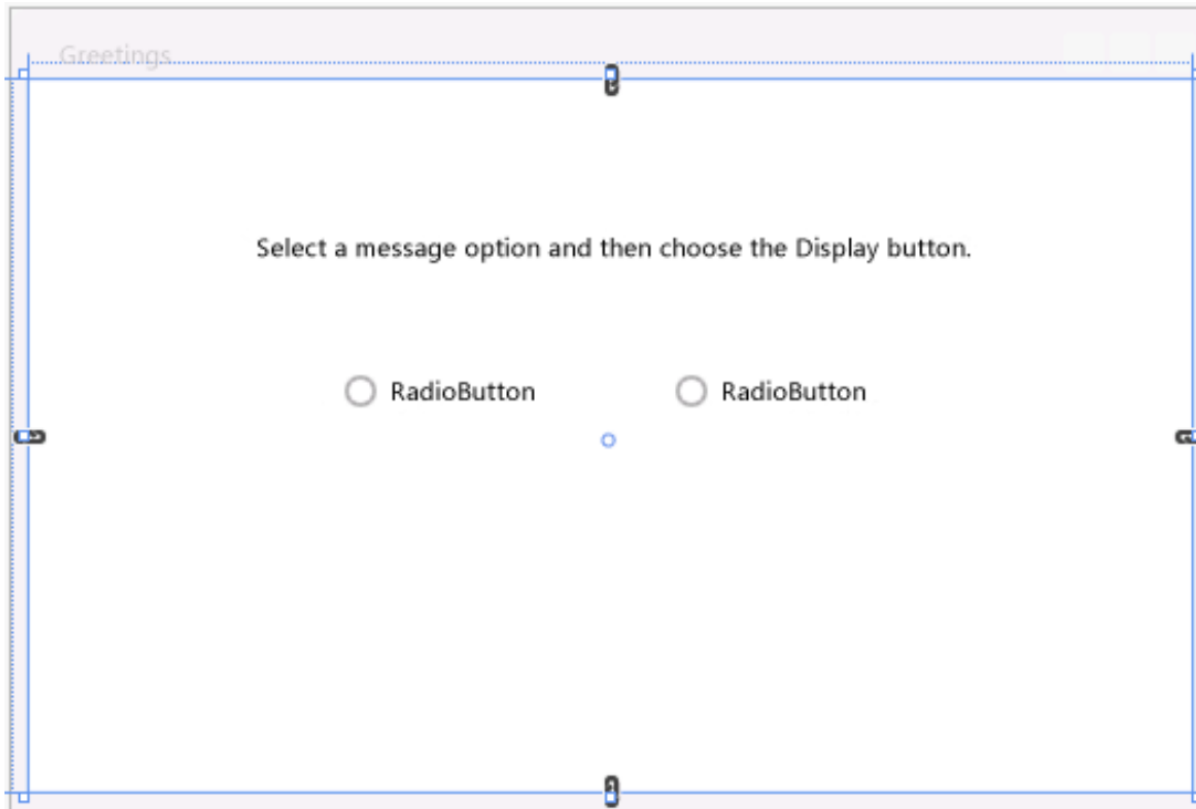


2. Agregue dos controles RadioButton a la superficie de diseño. Para ello, elija el elemento RadioButton y arrástrelo a la ventana en la superficie de diseño. Mueva los botones (para hacerlo, selecciónelos y use las teclas de dirección) de modo que aparezcan uno junto al otro bajo el control TextBlock. Use las guías de color rojo para alinear los controles. vuelva a centrar TextBlock si es necesario y guarde los cambios. Para ello, presione Ctrl+S o use el elemento de menú Archivo.

# Agregar botones de radio

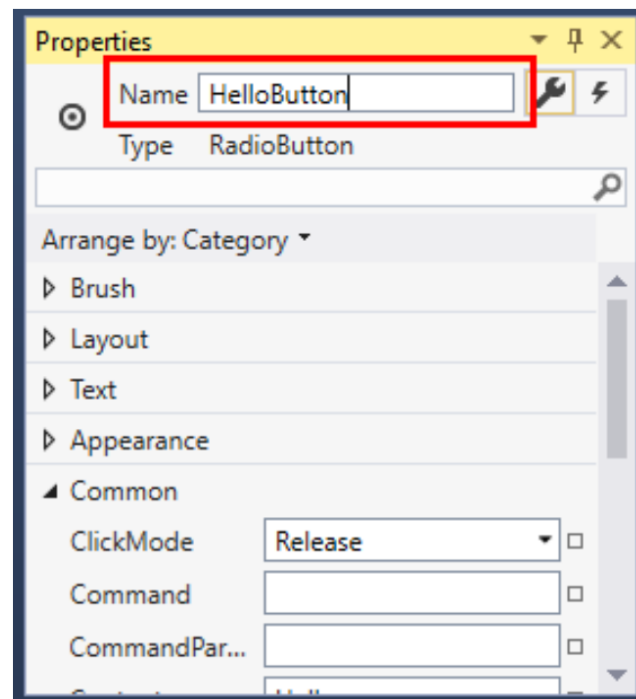
---

La ventana debe ser similar a la que se muestra a continuación:



# Agregar botones de radio

3. En la ventana Propiedades del control RadioButton izquierdo, cambie la propiedad Name (la propiedad situada en la parte superior de la ventana Propiedades ) a `HelloButton`.





# Agregar botones de radio

---

4. En la ventana **Propiedades** del control RadioButton derecho, cambie la propiedad **Name** a `GoodbyeButton` y después guarde los cambios.

Ahora puede agregar el texto para mostrar de cada control RadioButton. El procedimiento siguiente actualiza la propiedad **Content** de un control RadioButton.

# Agregar el texto para mostrar de cada botón de radio

---

Cambie el atributo **Content** de los controles `HelloButton` y `GoodbyeButton` por `"Hello"` y `"Goodbye"` en el código XAML. El marcado XAML ahora debería ser similar al del ejemplo siguiente:

XAML Copiar


```
<Grid>
    <TextBlock HorizontalAlignment="Left" Margin="252,47,0,0" Tex
    <RadioButton x:Name="HelloButton" Content="Hello" HorizontalA
    <RadioButton x:Name="GoodbyeButton" Content="Goodbye" Horizor
</Grid>
```

# Establecer que un botón de radio esté activado de forma predeterminada

---

En este paso haremos que HelloButton esté activado de forma predeterminada, de modo que siempre esté seleccionado uno de los dos botones de radio.

En la vista XAML, busque el marcado de HelloButton y agregue un atributo **IsChecked**:

XAML	 Copiar
<code>IsChecked="True"</code>	

El último elemento de la interfaz de usuario que agregará es un control Button.

# Agregar el control Button

---

1. En el **Cuadro de herramientas**, busque el control **Botón** y, después, agréguelo a la superficie de diseño en los controles RadioButton. Para ello, arrástrelo al formulario de la vista de diseño. Si usa Visual Studio 2019 o una versión posterior, una línea de color rojo le ayuda a centrar el control.
2. En la vista XAML, cambie el valor de **Content** del control de botón de `Content="Button"` a `Content="Display"` y, después, guarde los cambios.

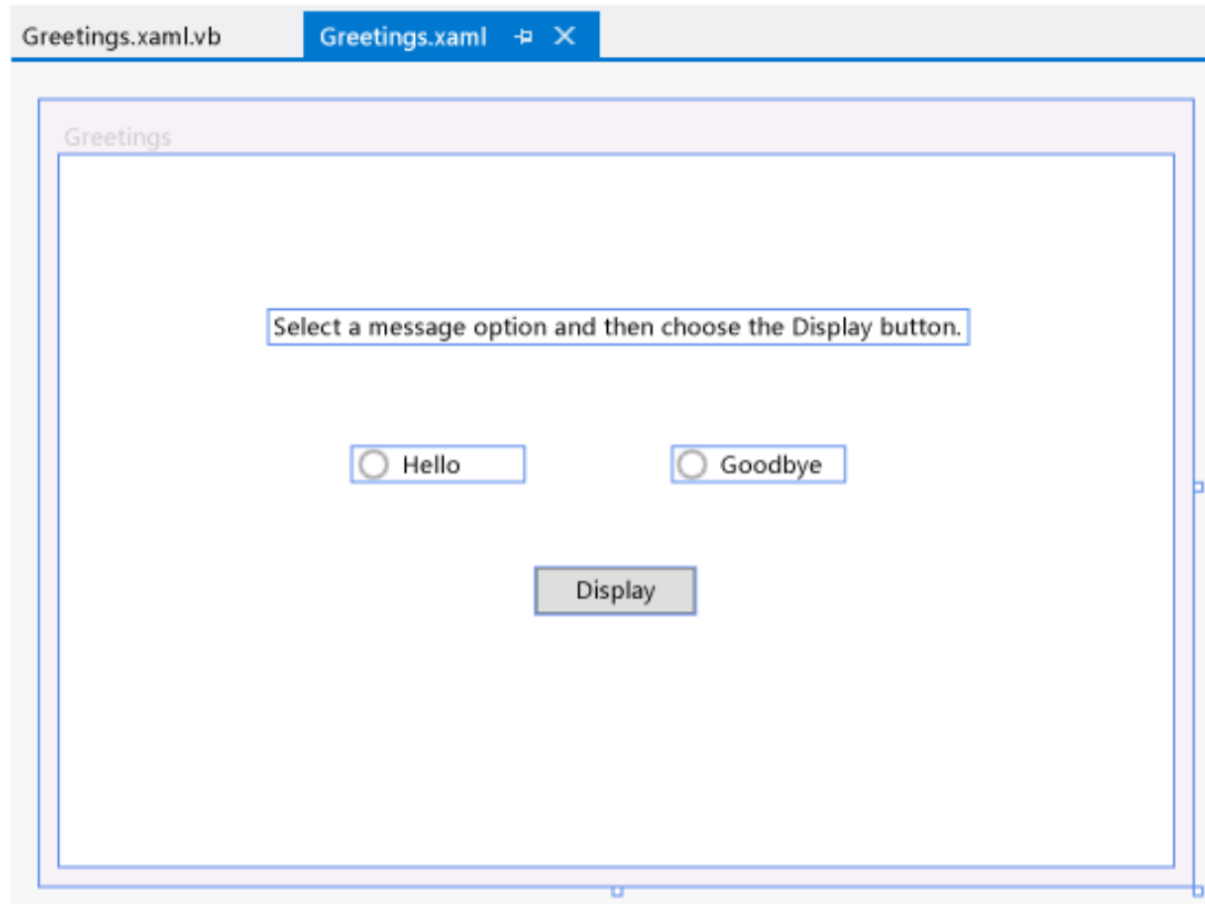
El marcado debería ser similar al del ejemplo siguiente: `<Button`

`Content="Display" HorizontalAlignment="Left"`

`VerticalAlignment="Top" Width="75" Margin="215,204,0,0"/>`

La ventana debería parecerse a la de la siguiente ilustración.

# Agregar el control Button



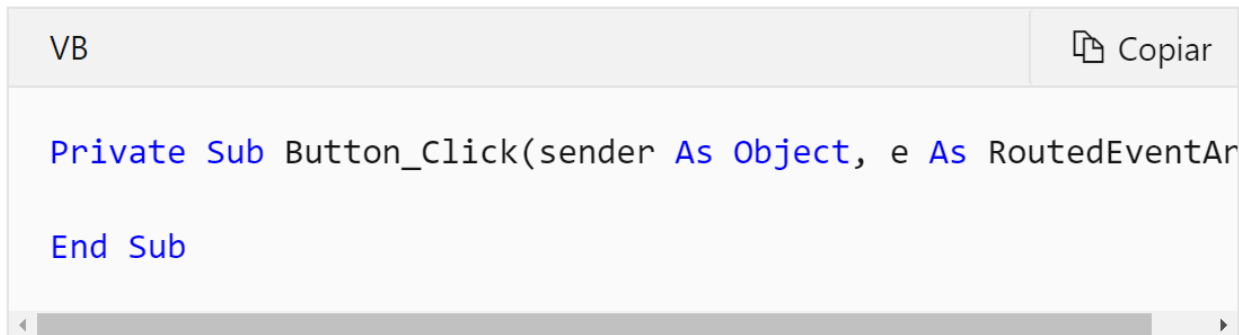
# Agregar código al botón Mostrar

---

Cuando se ejecuta esta aplicación, aparece un cuadro de mensaje después de que un usuario elija un botón de radio y, luego, el botón **Display**. Aparecerá un cuadro de mensaje para Hello y otro para Goodbye. Para crear este comportamiento, debe agregar código al evento `Button_Click` en *Greetings.xaml.vb* o *Greetings.xaml.cs*.

1. En la superficie de diseño, haga doble clic en el botón **Mostrar**.

Se abre *Greetings.xaml.vb* con el cursor en el evento `Button_Click`.



```
VB [Copiar]

Private Sub Button_Click(sender As Object, e As RoutedEventArgs)

End Sub
```

# Agregar código al botón Mostrar

---

2. Escriba el siguiente código:

VB

 Copiar

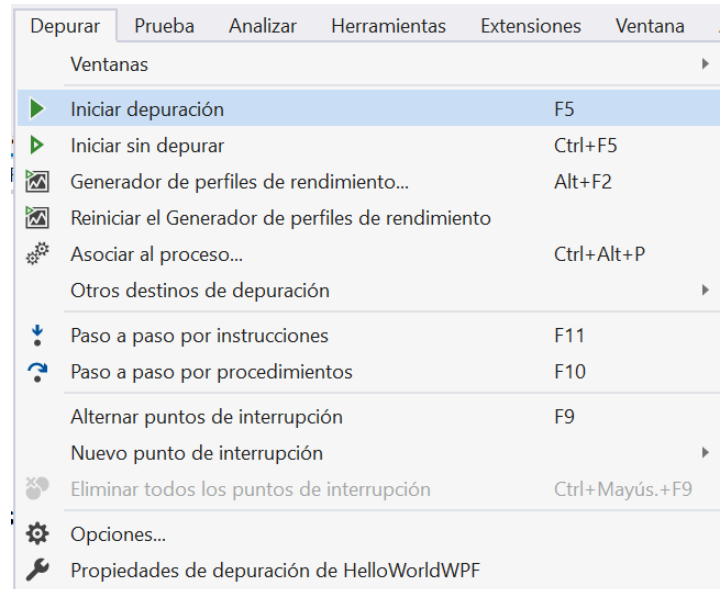
```
If HelloButton.Checked = True Then  
    MessageBox.Show("Hello.")  
ElseIf GoodbyeButton.Checked = True Then  
    MessageBox.Show("Goodbye.")  
End If
```

3. Guarde la aplicación.

# Depurar y probar la aplicación

A continuación, depuraremos la aplicación para buscar errores y probar que los dos cuadros de mensaje aparecen correctamente.

Las siguientes instrucciones indican cómo compilar e iniciar el depurador:

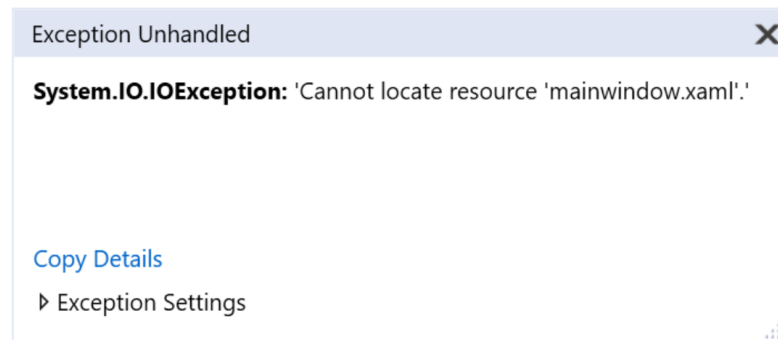




# Depurar y probar la aplicación

---

En este paso, encontraremos el error que causamos anteriormente al cambiar el nombre del archivo XAML de la ventana principal.



Elegimos el botón aceptar y detenemos la depuración.

Recordemos que hemos cambiado el nombre de MainWindow.xaml a Greetings.xaml al comienzo de este tutorial, pero el código todavía hace referencia a MainWindow.xaml como inicio de la aplicación, por lo que el proyecto no puede iniciarse.

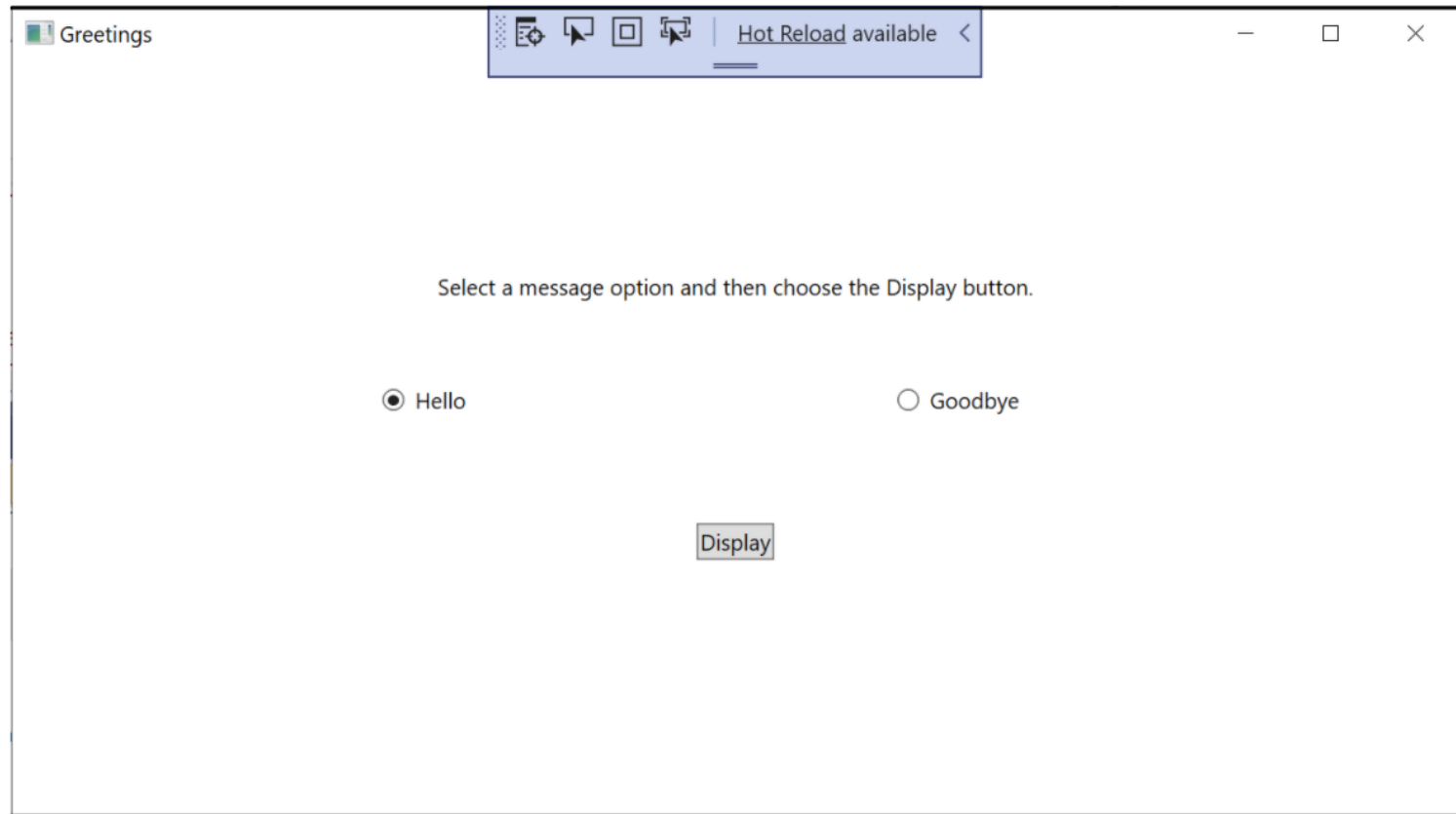
# Depurar y probar la aplicación

---

Para especificar Greetings.xaml como el URI de inicio debemos:

1. En el explorador de soluciones, abrimos el archivo App.xaml (En el proyecto de c-) en la vista XAML (no puede abrirse en la vista de Diseño). Para ello, seleccionamos el archivo y presionamos Entrar o hacemos doble clic en él.
2. Cambiamos *StartupURI*="MainWindow.xaml" a *StartupURI*="Greetings.xaml" y guardamos los cambios con Ctrl-s.
3. Volvemos a iniciar el depurador (presiona F5).
4. Deberíamos ver la ventana Greetings de la aplicación.

# Depurar y probar la aplicación



# Agregar puntos de interrupción

---

1. Abra *Greetings.xaml.vb* y seleccione la línea siguiente:

```
MessageBox.Show("Hello.")
```

2. Para agregar un punto de interrupción, presione **F9** o seleccione **Depurar** en el menú y, luego, **Alternar punto de interrupción**.

Aparece un círculo rojo al lado de la línea de código en el margen izquierdo de la ventana del editor.

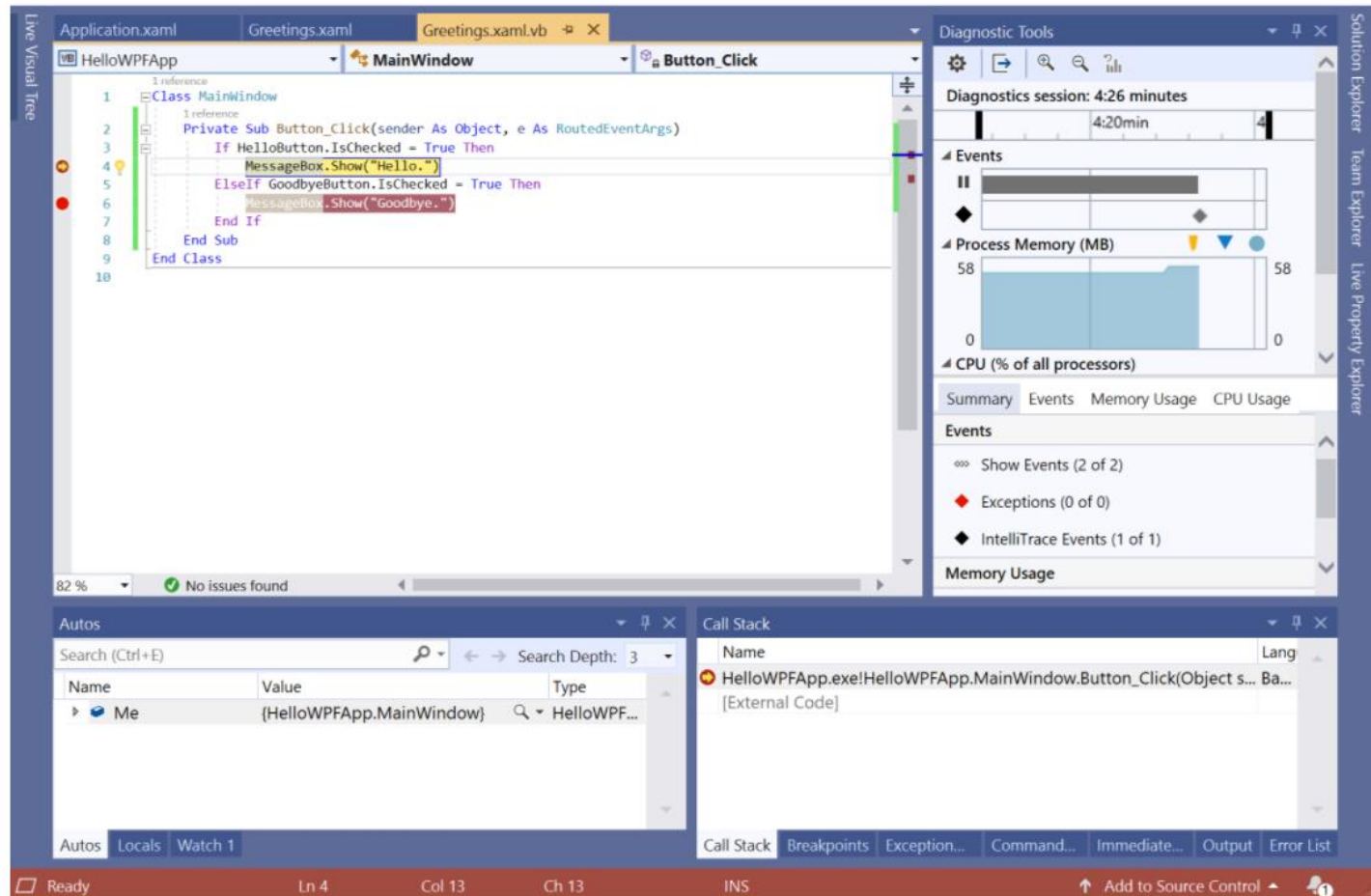
3. Seleccione la línea siguiente: `MessageBox.Show("Goodbye.")`.
4. Presione la tecla **F9** para agregar un punto de interrupción y, después, presione **F5** para iniciar la depuración.
5. En la ventana **Greetings**, elija el botón de radio **Hello** y después elija el botón **Mostrar**.

# Agregar puntos de interrupción

---

La línea `MessageBox.Show("Hello.")` se resalta en amarillo. En la parte inferior del IDE, las ventanas Automático, Variables locales e Inspección están acopladas juntas en el lado izquierdo, mientras que las ventanas Pila de llamadas, Puntos de interrupción, Configuración de excepción, Comando, Inmediato y Salida están acopladas juntas en el lado derecho.

# Agregar puntos de interrupción



# Agregar puntos de interrupción

---

6. En la barra de menús, elija **Depurar** > **Paso a paso para salir**.

La aplicación reanuda la ejecución y aparece un cuadro de mensaje con la palabra "Hello".

7. Elija el botón **Aceptar** en el cuadro de mensaje para cerrarlo.

8. En la ventana **Greetings** , elija el botón de radio **Goodbye** y después elija el botón **Mostrar** .

La línea `MessageBox.Show("Goodbye.")` se resalta en amarillo.

# Agregar puntos de interrupción

---

9. Presione la tecla **F5** para continuar con la depuración. Cuando aparezca el cuadro de mensaje, elija el botón **Aceptar** en el cuadro de mensaje para cerrarlo.
10. Cierre la ventana de la aplicación para detener la depuración.
11. En la barra de menús, elija **Depurar > Deshabilitar todos los puntos de interrupción**.

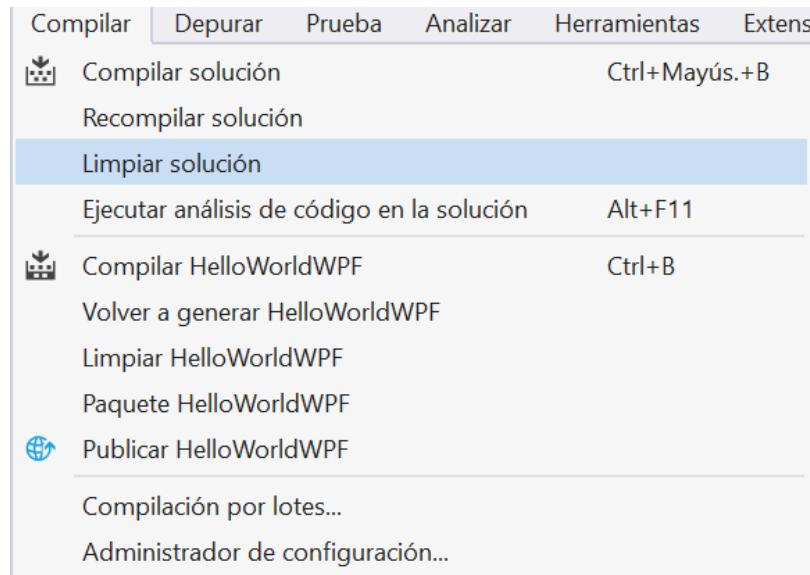


# Compilar una versión de lanzamiento de la aplicación

---

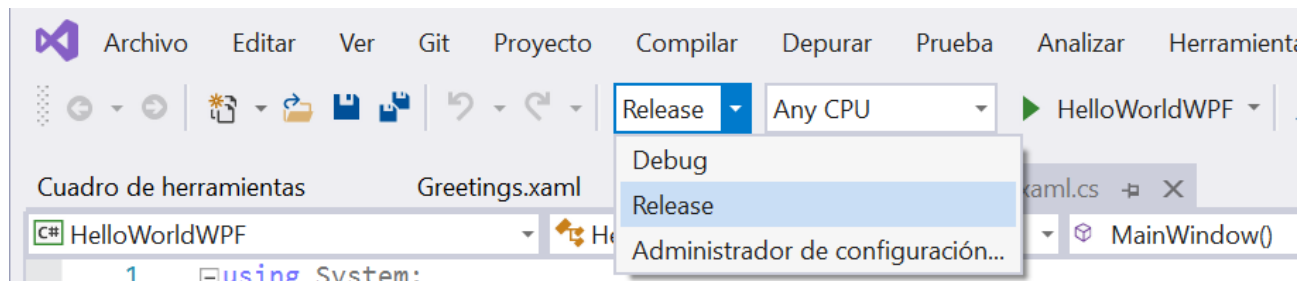
Ahora que ha comprobado que todo funciona, puede preparar una versión de lanzamiento de la aplicación.

1. En el menú principal, seleccione **Compilar > Limpiar solución** para eliminar los archivos intermedios y de salida que se han creado durante las compilaciones anteriores. Esto no es necesario, pero se limpian las salidas de compilación de depuración.



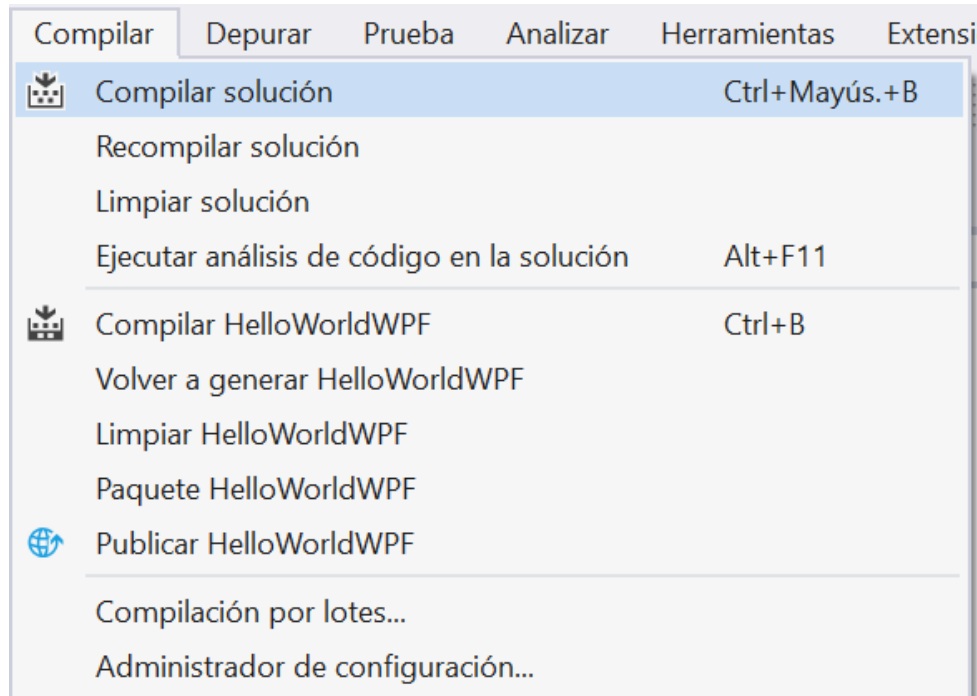
# Compilar una versión de lanzamiento de la aplicación

2. Cambie la configuración de compilación de HelloWorldWPF de **Depurar** a **Liberar** mediante el control de lista desplegable en la barra de herramientas (dice "Depurar" actualmente)



# Compilar una versión de lanzamiento de la aplicación

3. Compile la solución seleccionando **Compilar > Compilar solución** o presiona F6



# Fin

---

Enhorabuena por completar este tutorial.

Puede encontrar el `.exe` creado en el directorio de soluciones y proyectos  
(`...\HelloWPFApp\HelloWPFApp\bin\Release`).>