



# RJ CODE MODERN UI

Custom themes, styles, forms and controls for WinForm + Source code



## GUÍA RÁPIDA

## PROGRAMMING TUTORIALS

Simple, easy, fast and fun learning

## Contenido

1.	Introducción .....	1
1.1.	Documentación .....	1
1.2.	Comentarios .....	2
1.3.	Demostración .....	2
2.	¿Cómo implementar el proyecto en mi proyecto existente? .....	3
3.	¿Cómo utilizar los formularios personalizados? .....	4
3.1.	Crear formulario principal .....	4
3.2.	Crear formularios secundarios .....	6
3.3.	¿Cómo cambiar las propiedades?.....	7
3.4.	¿Cómo cambiar el ancho del formulario secundario? .....	7
4.	¿Cómo utilizar los controles personalizados?.....	8
4.1.	Ejemplo 1- Diseñar el formulario principal .....	9
4.2.	Ejemplo 2- Diseñar un formulario secundario .....	11
5.	¿Cómo modificar los componentes del proyecto? .....	12

## 1. Introducción

Hola colega ☺, te saluda RJ Code. Antes que nada, quiero agradecerte por adquirir el proyecto, realmente espero que esto agregue un poco más a sus habilidades y que obtenga nuevos conocimientos.

Bueno, **el proyecto en sí NO es una plantilla. El objetivo principal del proyecto es enseñar cómo hacer formularios y controles personalizados** para construir interfaces de usuario modernas y elegantes, por eso lo categorizo como un tutorial escrito. Para ello hay **3 componentes de ayuda**: la documentación, los comentarios y la demostración.

### 1.1. Documentación

Cabe mencionar que no se refiere a la documentación del ciclo de vida de un software, sino que es una documentación sencilla para el usuario final (Para usted).

La documentación **describe cada uno de los componentes del proyecto**, esto facilita la interacción con el código fuente, para que puedas conocer la funcionalidad de cada uno de los campos, propiedades y métodos, o poder **ubicar rápidamente una clase, método o propiedad** en un contexto determinado, pudiendo así **modificar o agregar nuevas funciones o propiedades de apariencia**.

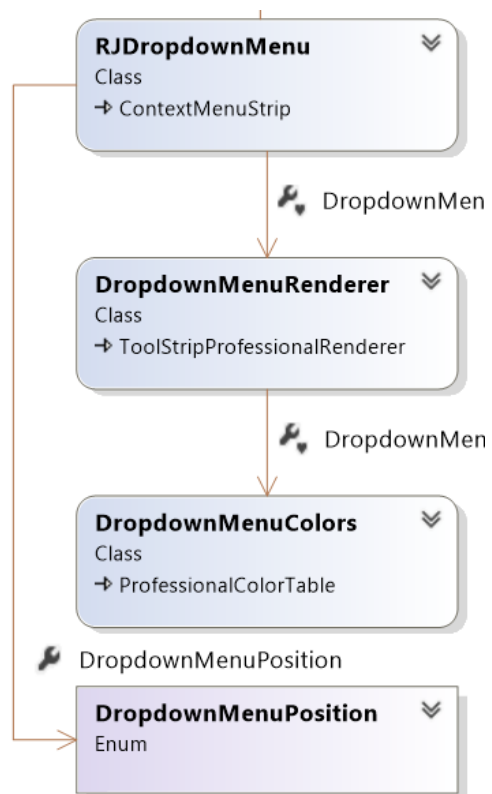
El truco está en **observar los diagramas de clases**, porque con esto ya sabrás **de qué clase hereda un control o formulario**, además de saber **qué campos, propiedades y métodos implementa**. Por ejemplo, supongamos que tenemos el control de **menú desplegable** (RJDropDownMenu):

#### - RJDropDownMenu – Diagrama de clases contraído

Con esto podemos determinar que:

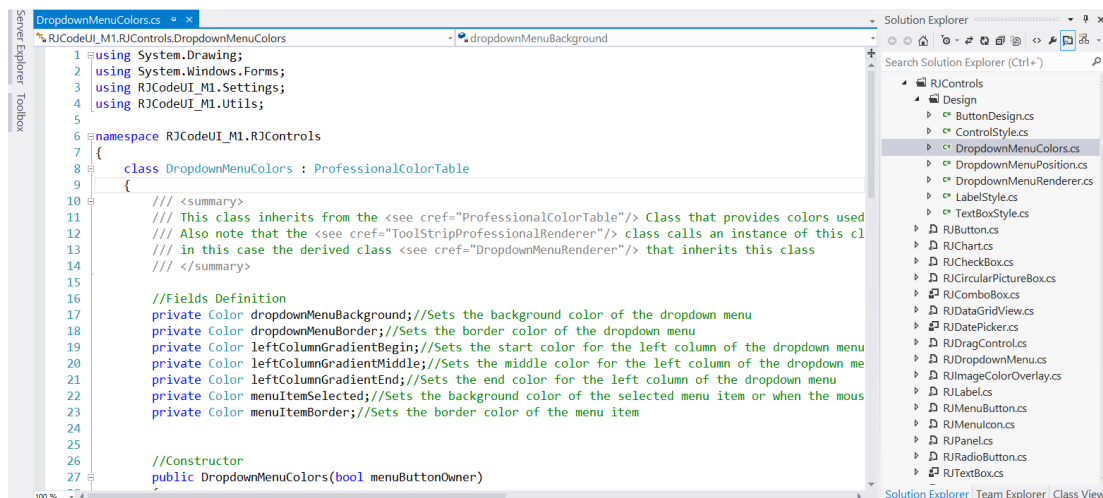
- ✓ El control **RJDropDownMenu** hereda de la **clase ContextMenuStrip** (Control existente de Windorm Form), por lo tanto conserva todas sus funcionalidades.
- ✓ El control **RJDropDownMenu** depende de la **clase DropdowMenuRenderer y DropdowMenuColors** para configurar sus propiedades de apariencia.
- ✓ La clase **DropdowMenuRenderer** hereda de la **clase ToolStripProfessionalRenderer**.
- ✓ La clase **DropdowMenuColors** hereda de la **clase ProfessionalColorTable**.

También están los **diagramas de clases expandido** para conocer los campos, propiedades y métodos de una clase. Entonces, si deseas modificar el color de borde, simplemente localiza la propiedad **ColorBorder** o algo relacionado con ello.



## 1.2. Comentarios

Los comentarios **describen o especifican la función** de un campo, propiedad, método o clase. Como se observa en la siguiente captura de pantalla de la clase **DropdownMenuColors**.



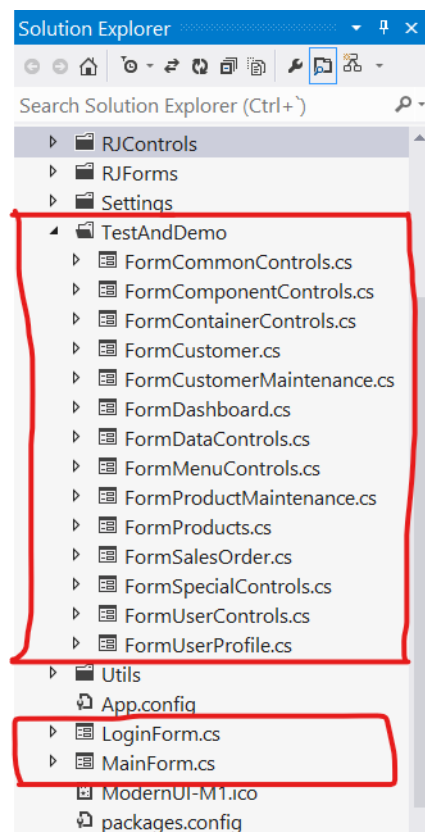
Por lo tanto, **no es tan necesario leer la descripción de los campos, propiedades y métodos en la documentación, ya que a menudo son similares.**

## 1.3. Demostración

La demostración consiste básicamente en todos los formularios ya diseñados en el proyecto que prueban y demuestran cómo usar los controles y formularios personalizados (La mayoría de estos formularios agrupan y describen un control en específico).

El formulario **LoginForm**, **MainForm** y todos los formularios de la carpeta **TestAndDemo** son ejemplos de demostración.

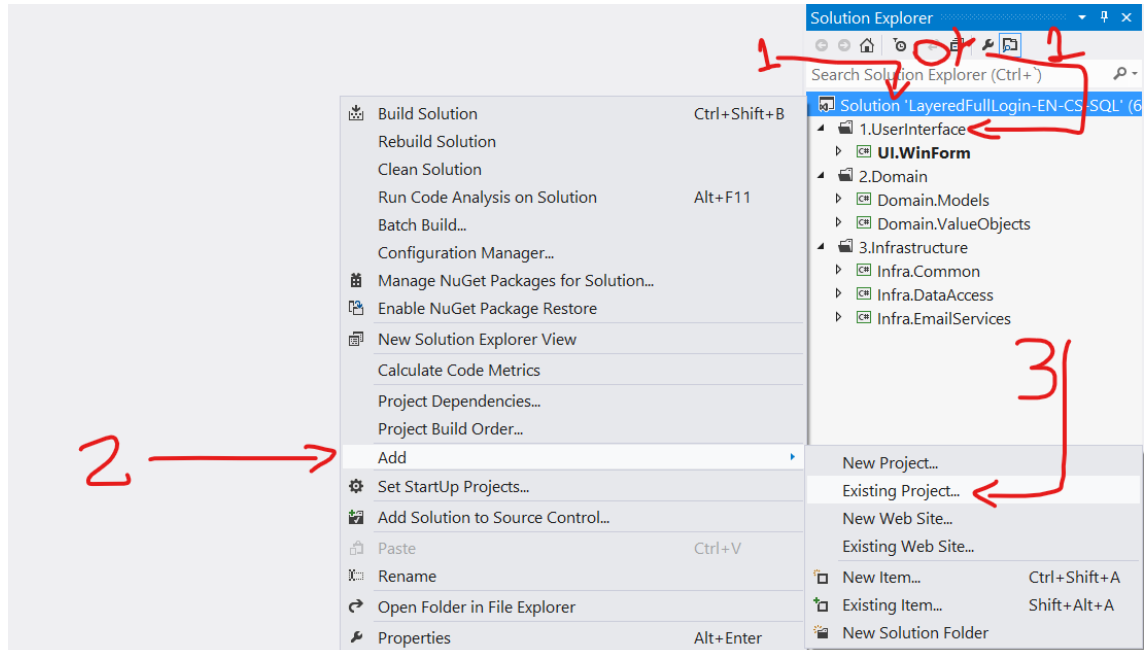
- ✓ El formulario **LoginForm** demuestra el uso de la clase base **RJBaseForm**.
- ✓ El formulario **MainForm** demuestra el uso de la clase base **RJMainForm**.
- ✓ Los formularios de la carpeta **TestAndDemo** demuestran el uso de la clase base **RJChildForm**.
- ✓ **Todos los formularios anteriores agregan controles personalizados** para demostrar el uso de estos.



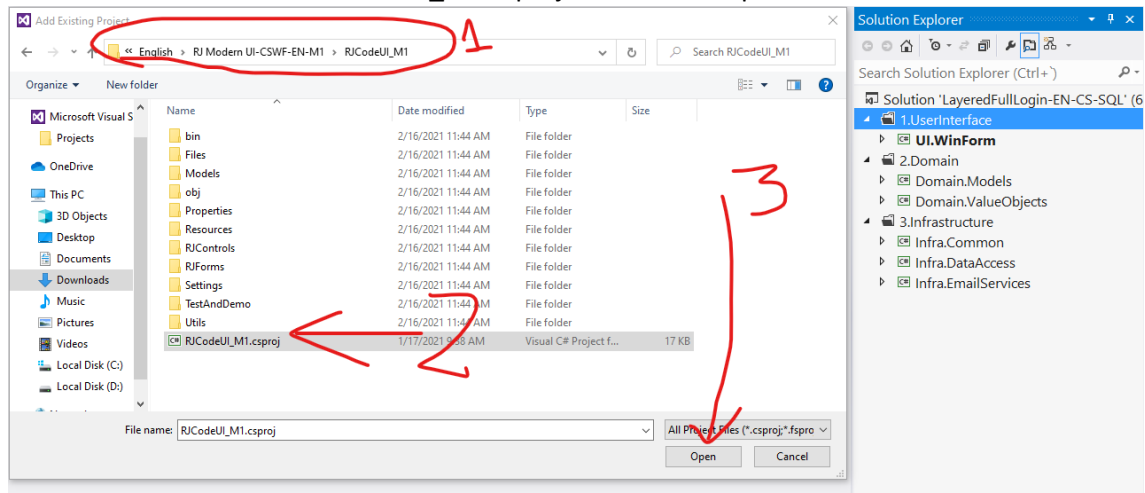
## 2. ¿Cómo implementar el proyecto en mi proyecto existente?

Puedes agregar fácilmente el proyecto RJ Code Moden UI-M1 a su proyecto existente, siguiendo los pasos a continuación:

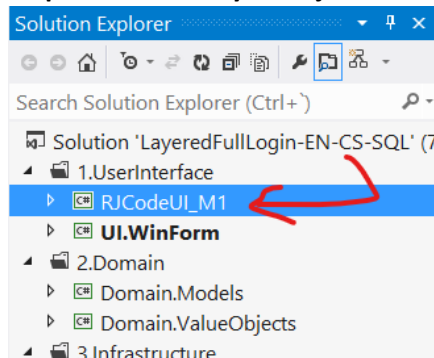
1. **Clic derecho** sobre la **solución o carpeta de soluciones de su proyecto** de Visual Studio  
-> Desplácese sobre la opción **Agregar** -> Seleccione la opción **Proyecto existente**.



2. **Localizamos el archivo RJCodeUI\_M1.csproj** en el paquete descargado ->  
**Seleccionamos el archivo RJCodeUI\_M1.csproj** -> Finalmente pulsamos en **Abrir**.



3. **Ya puedes utilizar y trabajar sobre el proyecto RJ Code Moden UI-M1** (No olvide de establecer el proyecto como **proyecto de inicio**).



### RECOMENDACIÓN:

**No es posible actualizar la apariencia de tus formularios y controles existentes con el proyecto RJCodeUI\_M1. Por lo tanto, te recomiendo rehacer toda la capa de Interfaz de Usuario (Presentación) de tu proyecto en el proyecto RJCodeUI\_M1. Como se explica más adelante.**

### 3. ¿Cómo utilizar los formularios personalizados?

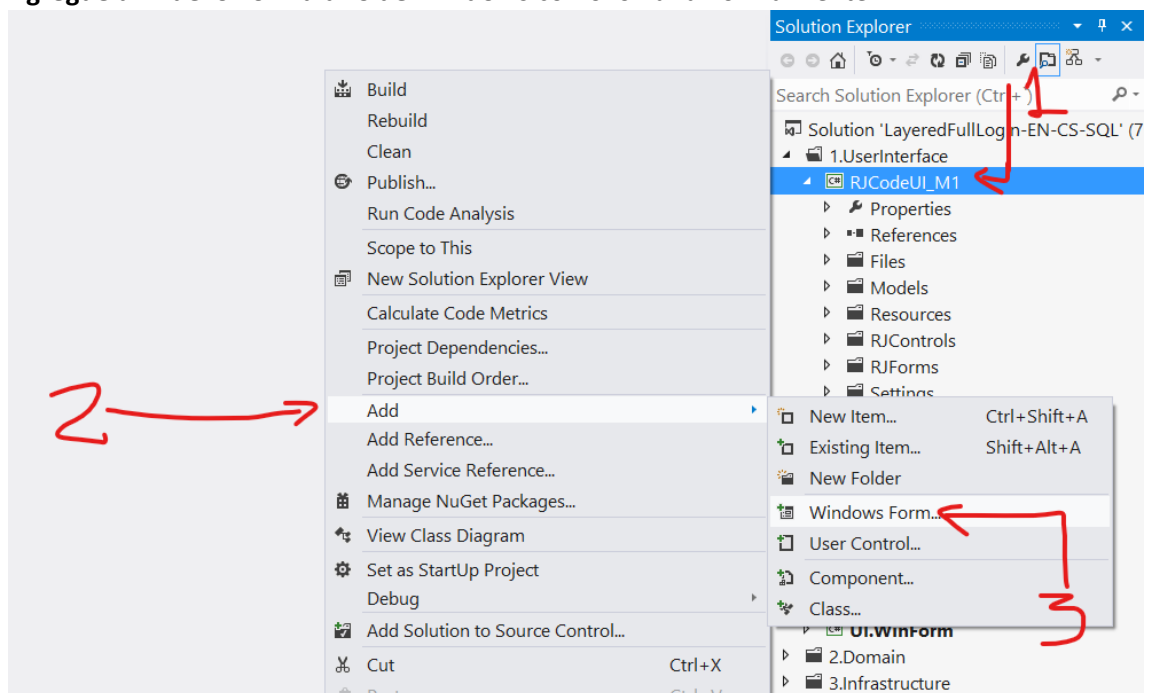
Como se indicó anteriormente, **debe rehacer completamente la capa de interfaz de usuario** de su proyecto existente en el proyecto **RJCodeUI\_M1** que agregó. **No recomiendo editar los formularios de prueba y demostración** del proyecto (Esto incluye LoginForm, MainForm y todos los formularios de la carpeta TestAndDemo), ya que estos te ayudan a entender e interactuar con el código fuente. Por lo tanto, te **recomiendo crear nuevos formularios**.

**Para usar los formularios personalizados base** (RJBaseForm, RJMainForm y RJChildForm), **simplemente herede uno de ellos**, como se muestran a continuación.

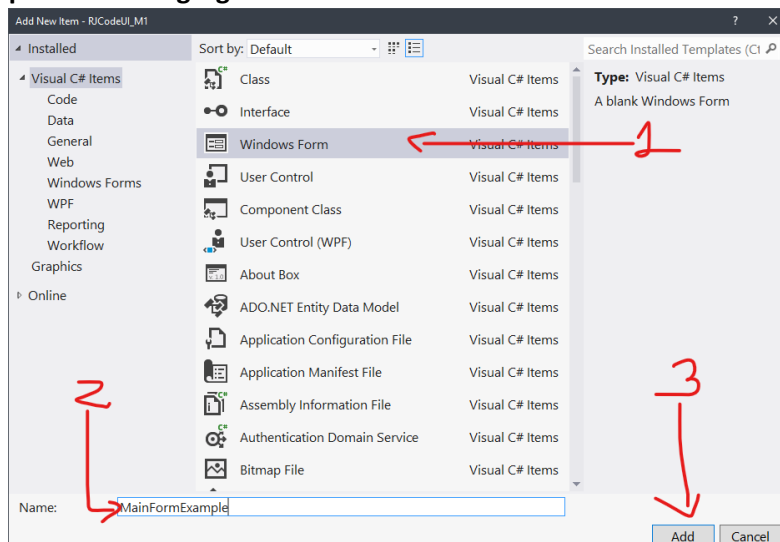
#### 3.1. Crear formulario principal

Para crear un formulario principal **herede la clase RJMainForm**, siguiendo los pasos a continuación:

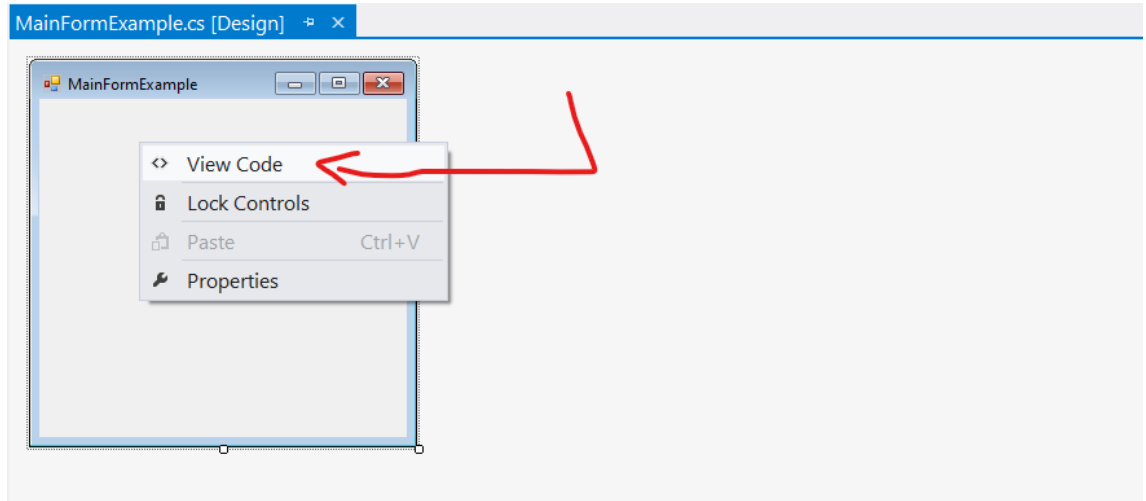
1. **Agregue un nuevo Formulario de Windows** como lo haría normalmente.



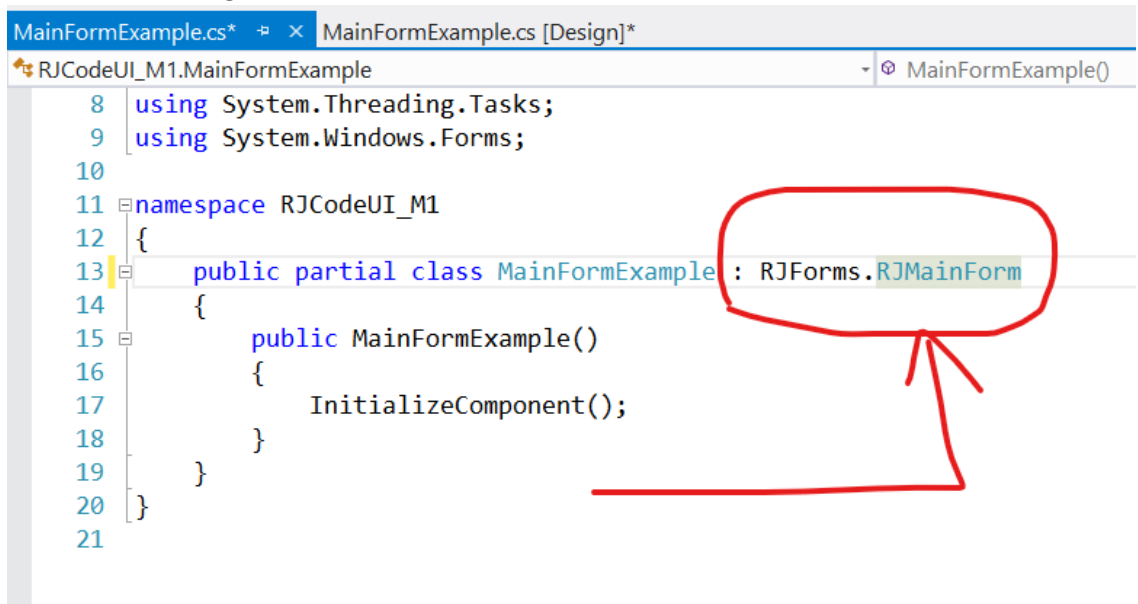
2. **Coloque un nombre para el formulario** (En mi caso Ejemplo Formulario Principal) y **presione en Agregar**.



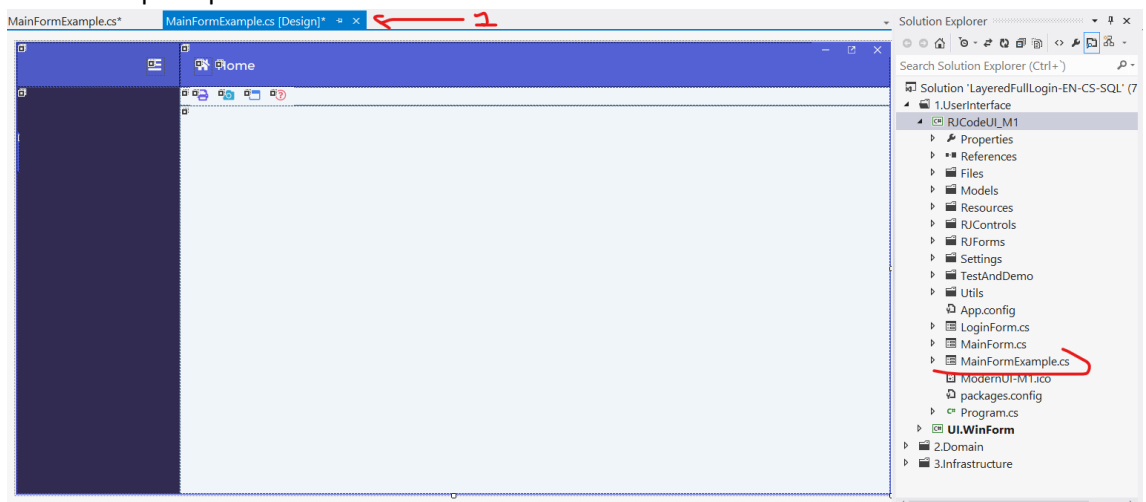
3. **Dirígete al código del formulario agregado (Clic derecho-> Ver código).**



4. Una vez en el código, **herede la clase RJMainForm.**



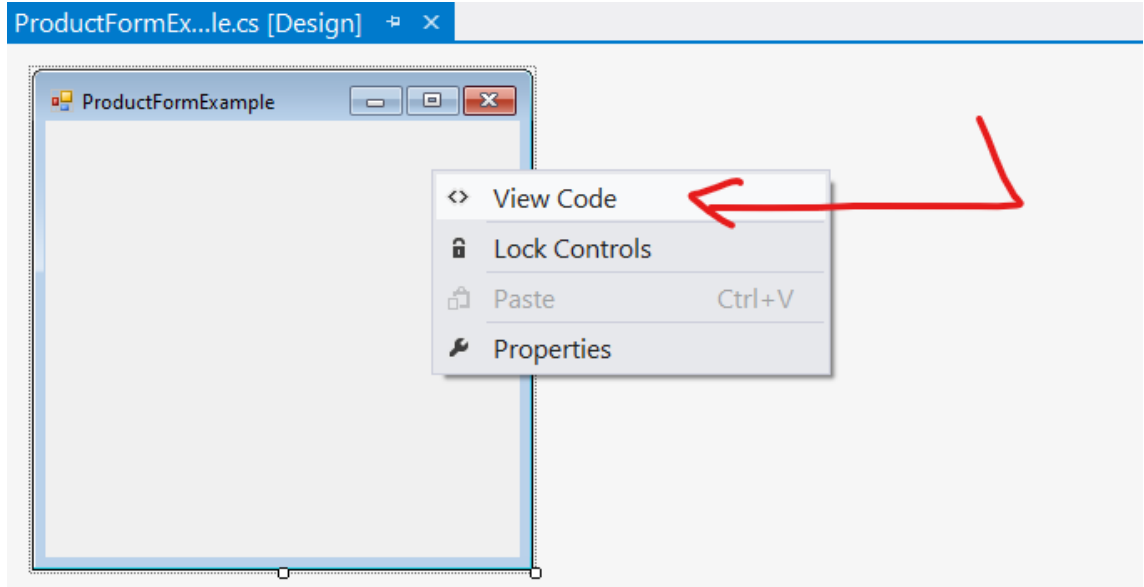
5. Por último, **vaya al diseñador de formularios y tendrá la apariencia aplicada del formulario principal como se muestra a continuación.**



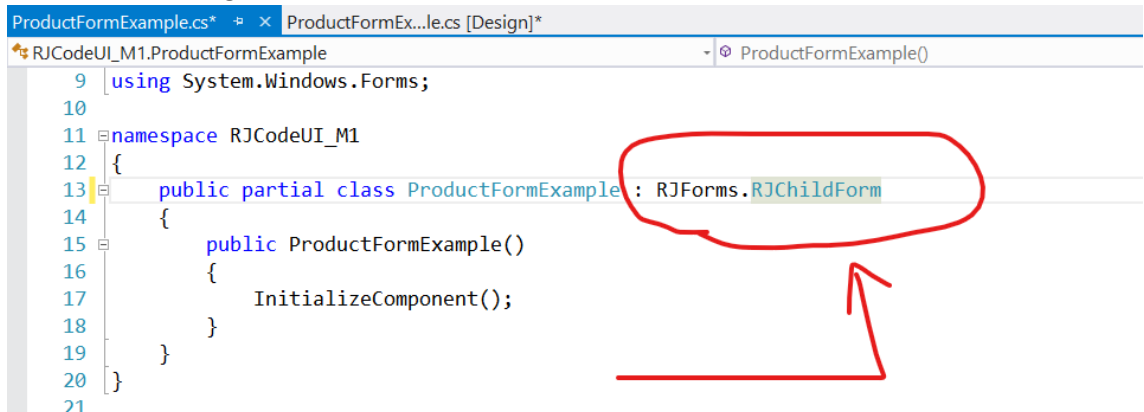
### 3.2. Crear formularios secundarios

Para crear un formulario secundario **herede la clase RJChildForm**, siguiendo los pasos a continuación:

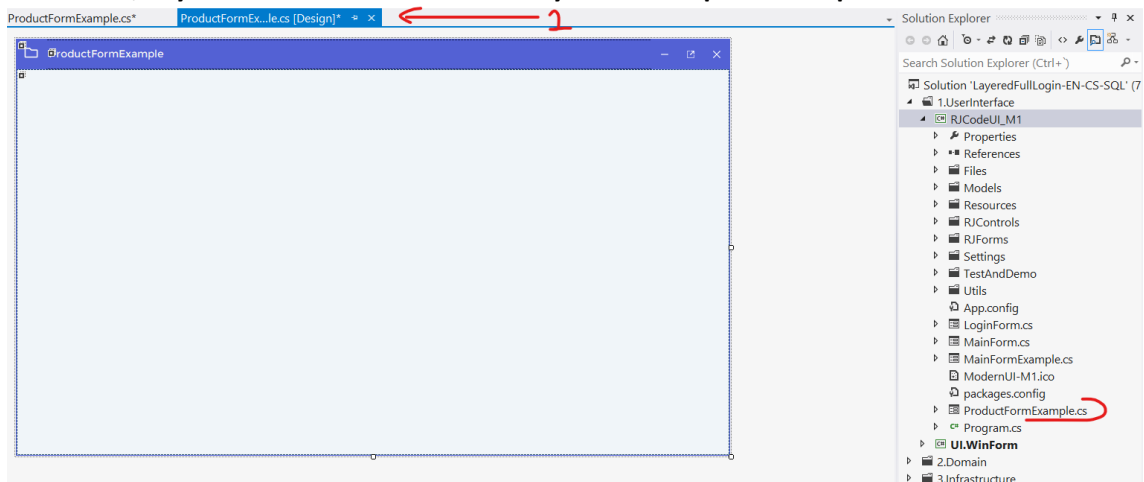
1. **Agregue un nuevo Formulario de Windows** y vaya al **código del formulario** como se hizo anteriormente (Paso 1- 3).



2. Una vez en el código, **herede la clase RJChildForm**.



3. Por último, **vaya al diseñador de formularios** y tendrá la apariencia aplicada.

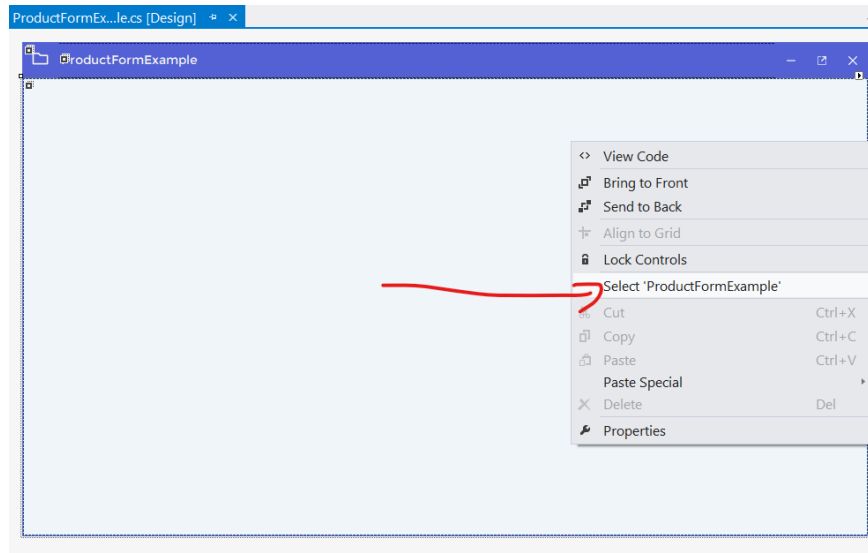




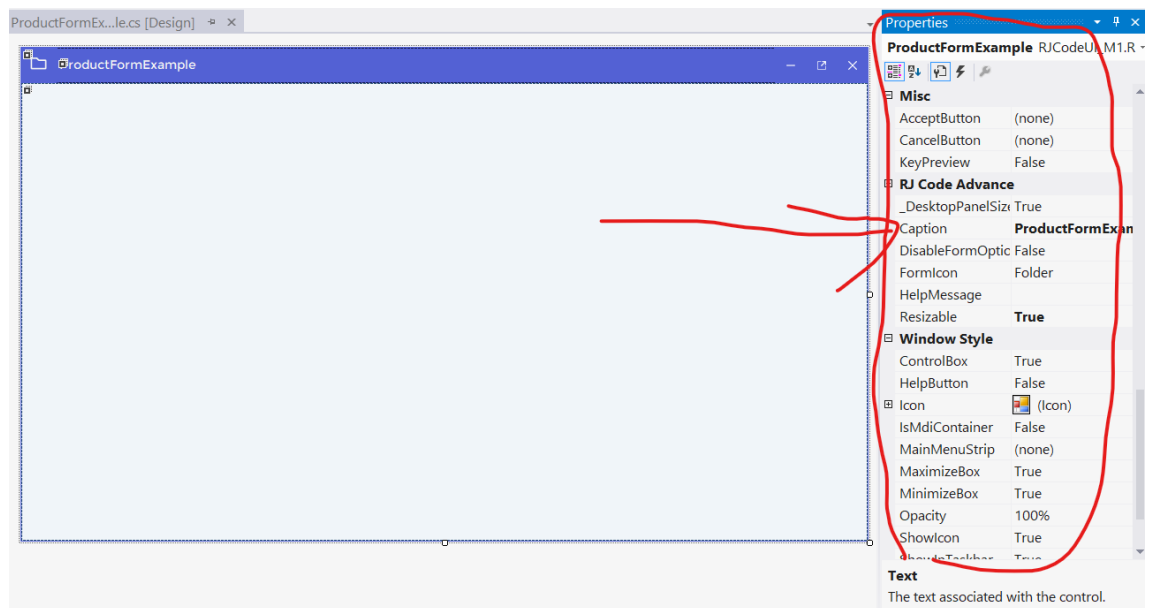
### 3.3. ¿Cómo cambiar las propiedades?

Debido al hecho de que los formularios base RJMainForm y RJChildForm ya tienen controles agregados para la apariencia, **no es posible seleccionar directamente el formulario** y cambiar las propiedades. Para poder hacerlo tienes que hacerlo de la siguiente manera:

1. **Clic derecho sobre el formulario** o el espacio del diseñador-> Presionar en **Seleccionar 'Nombre de formulario'**.



2. Ya puedes cambiar las propiedades del formulario desde el cuadro de herramientas.

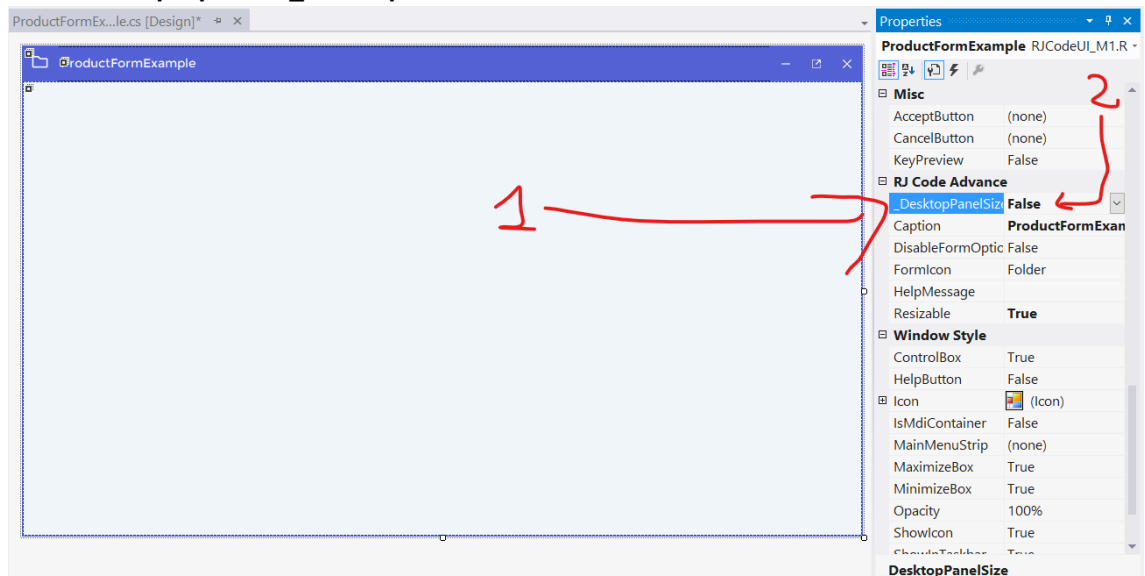


### 3.4. ¿Cómo cambiar el ancho del formulario secundario?

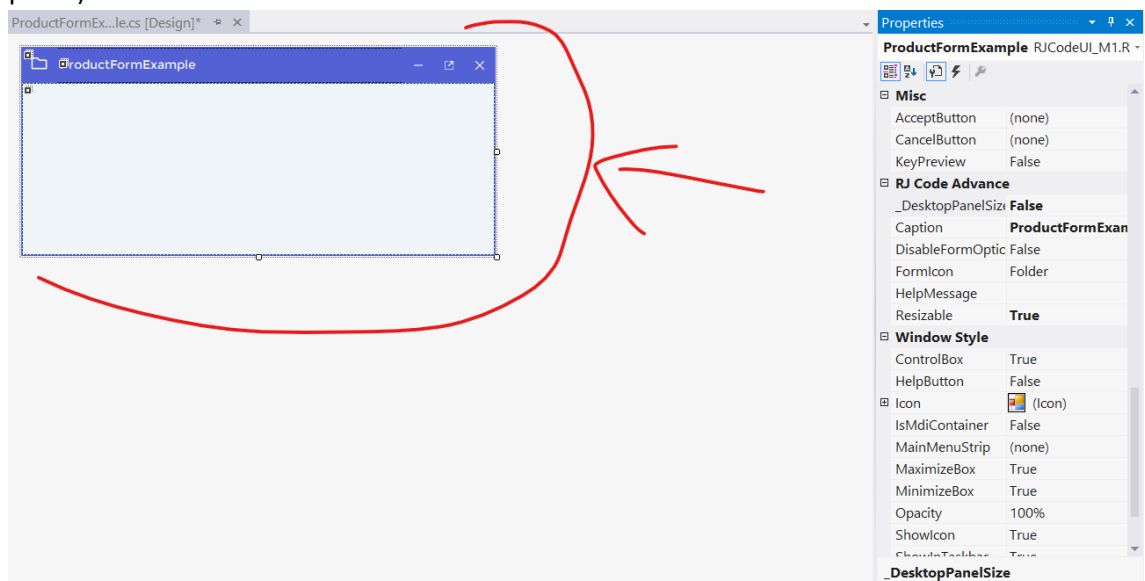
**Por defecto**, el tamaño del formulario secundario es el mismo que el tamaño del panel escritorio del formulario principal y **no es posible cambiar el ancho, solo permite cambiar el alto**, debido a que ello te facilita diseñar el formulario de manera más elegante y exacta para el escritorio del formulario principal.

Sin embargo, en muchas ocasiones no necesitamos abrir el formulario secundario en el escritorio del formulario principal y deseamos un tamaño diferente.

1. Para cambiar el ancho del formulario dirígete a las **propiedades del formulario** y establece la **propiedad \_DesktopPanelSize en FALSO**.



2. Ahora **ya puedes cambiar el ancho del formulario a placer** (Si solamente deseas cambiar el alto del formulario no es necesario (tampoco lo recomiendo) realizar estos pasos).



#### 4. ¿Cómo utilizar los controles personalizados?

Para usar los controles personalizados del proyecto, simplemente **abra la caja de herramientas** de Visual Studio y **comience a arrastrar los controles que necesita al formulario** como lo haría normalmente (En caso que no aparezca los controles compile el proyecto), a continuación se muestra algunos ejemplos.

#### 4.1. Ejemplo 1- Diseñar el formulario principal

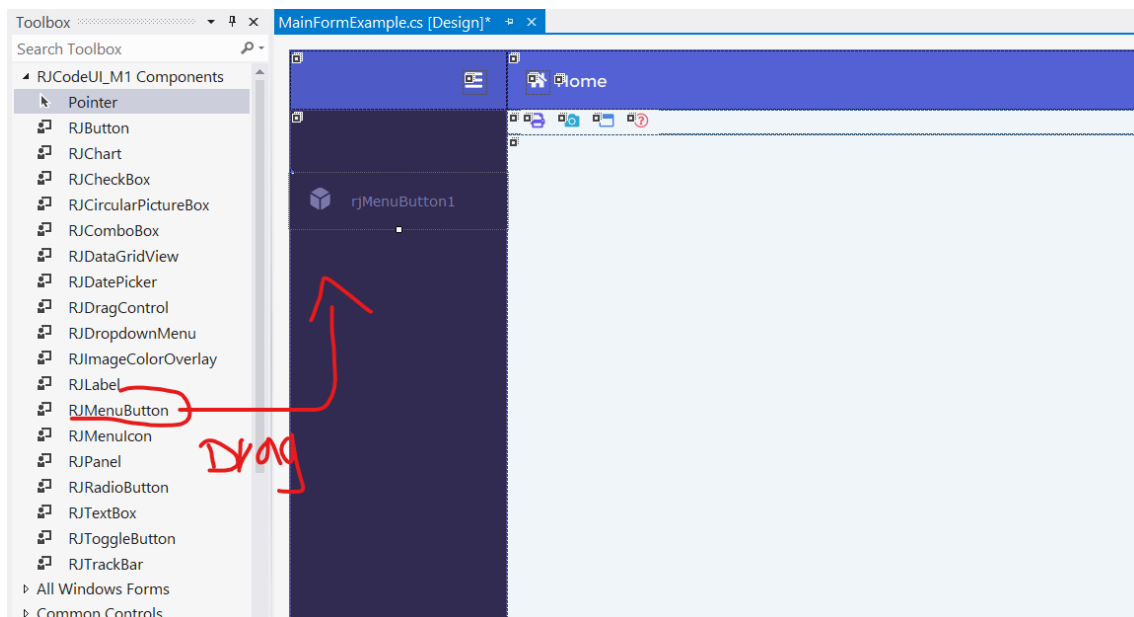
Cabe mencionar que en el proyecto hay controles específicos que puede agregar al formulario principal, estos son:

- ✓ **RJMenuButton** (Está optimizado solamente para ser utilizado en el menú lateral del formulario principal).
- ✓ **RJMenuIcon** (Por defecto está optimizado para ser utilizado en la barra de título del formulario principal, puede cambiar eso estableciendo la propiedad BackIcon en true).
- ✓ **RJDropDownMenu** (Menú desplegable que puede asociarlo con los controles anteriores: RJMenuButton y RJMenuIcon, no es necesario codificar para mostrarlo).
- ✓ **RJLabel** (Puede establecer la propiedad Style en BarCaption o BarText y usarlo en la barra de título del formulario principal).

Puede agregar otros controles, por ejemplo RJCircularPictureBox para mostrar la foto del usuario conectado en la aplicación.

#### Agregar un botón de menú (RJMenuButton)

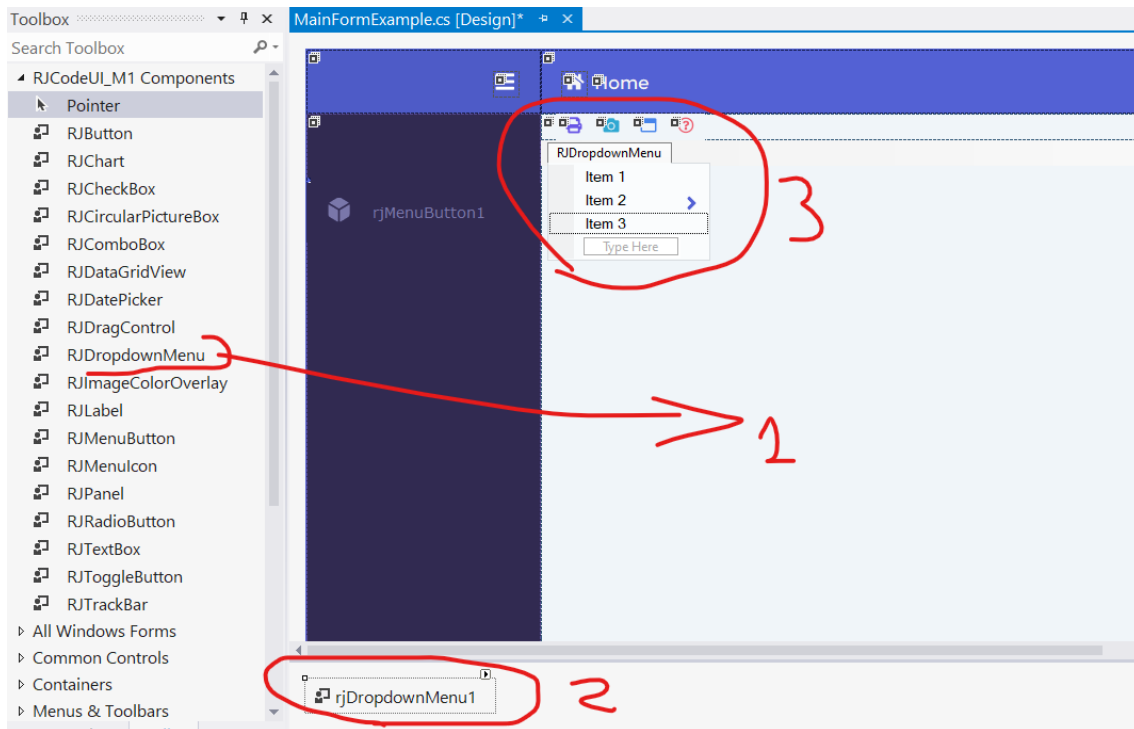
**Arrastre el control RJMenuButton** desde la caja de herramientas al **panel de menú lateral** de formulario principal como se muestra en la imagen.



Por ahora el control se comporta como un botón de menú normal, puedes convertirlo en un botón de menú desplegable, agregando el control RJDropDownMenu y asociarlo con el botón de menú, como se demuestra a continuación:

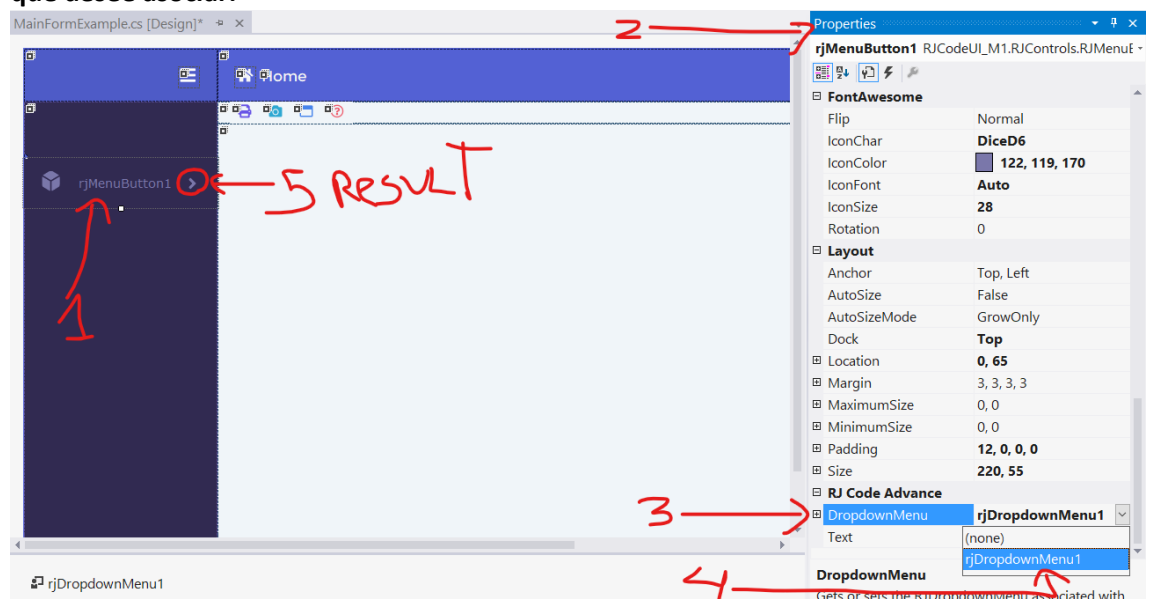
## Agregar un menú desplegable (RJDropDownMenu)

- 1) Arrastre el control RJDropDownMenu desde la caja de herramientas al formulario y agregue los elementos y sub-elementos que desee.

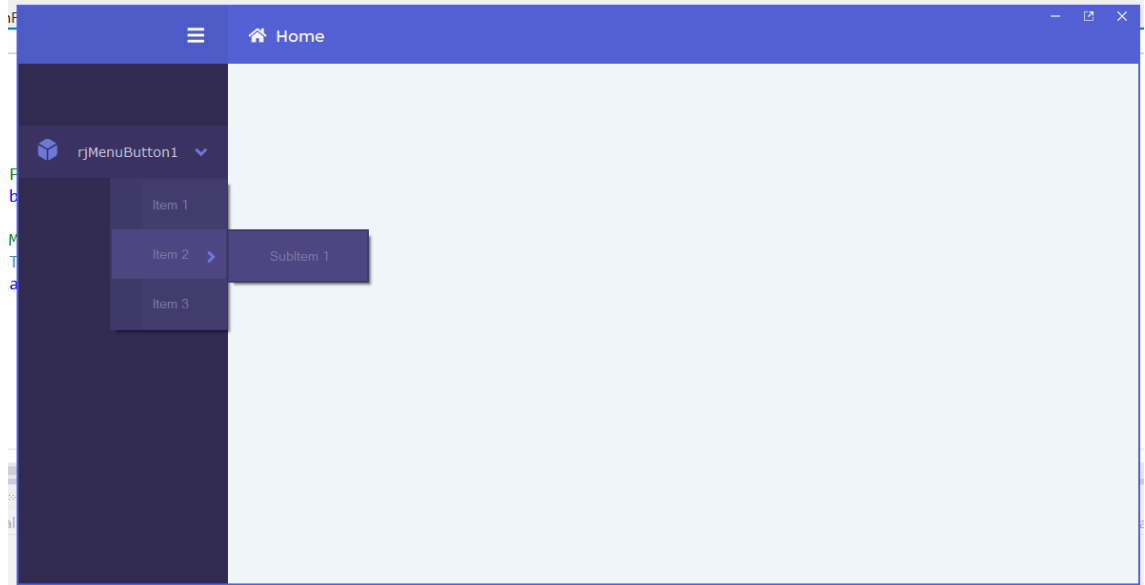


Es importante recordar que el control **RJDropDownMenu** hereda del control **ContextMenuStrip**, este control **es un componente**. Los componentes **generalmente no tienen una representación visual y no son elementos secundarios del formulario**. Por lo tanto, al agregar un componente al formulario estos **se ubican en la parte inferior** del espacio de trabajo del diseñador de formularios.

- 2) Asocie el control RJDropDownMenu con el control RJMenuButton o RJMenuIcon, para ello, realice lo siguiente: **Seleccione el botón de menú**-> Vaya a las **propiedades**-> Localice la **propiedad DropDownMenu**-> Finalmente **seleccione el menú desplegable que desee asociar**.

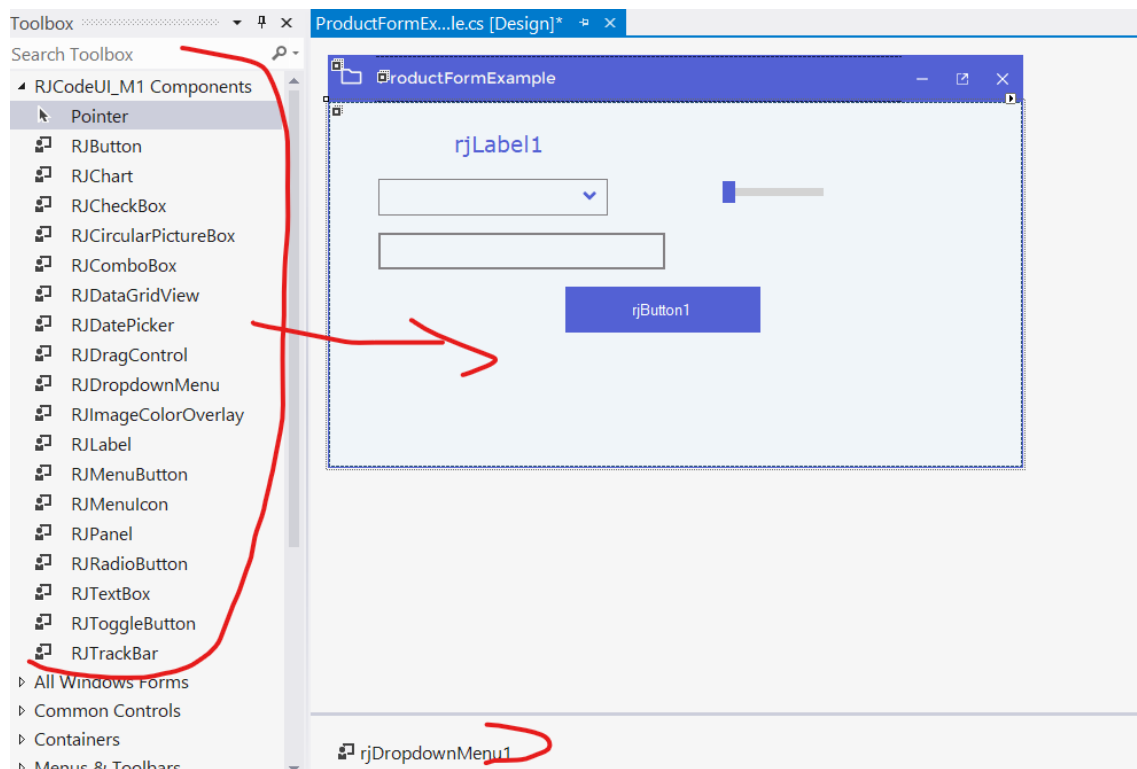


### 3) Resultado.



### 4.2. Ejemplo 2- Diseñar un formulario secundario

Como se hizo anteriormente, simplemente **arrastre los controles que necesite al formulario y configure las propiedades.**



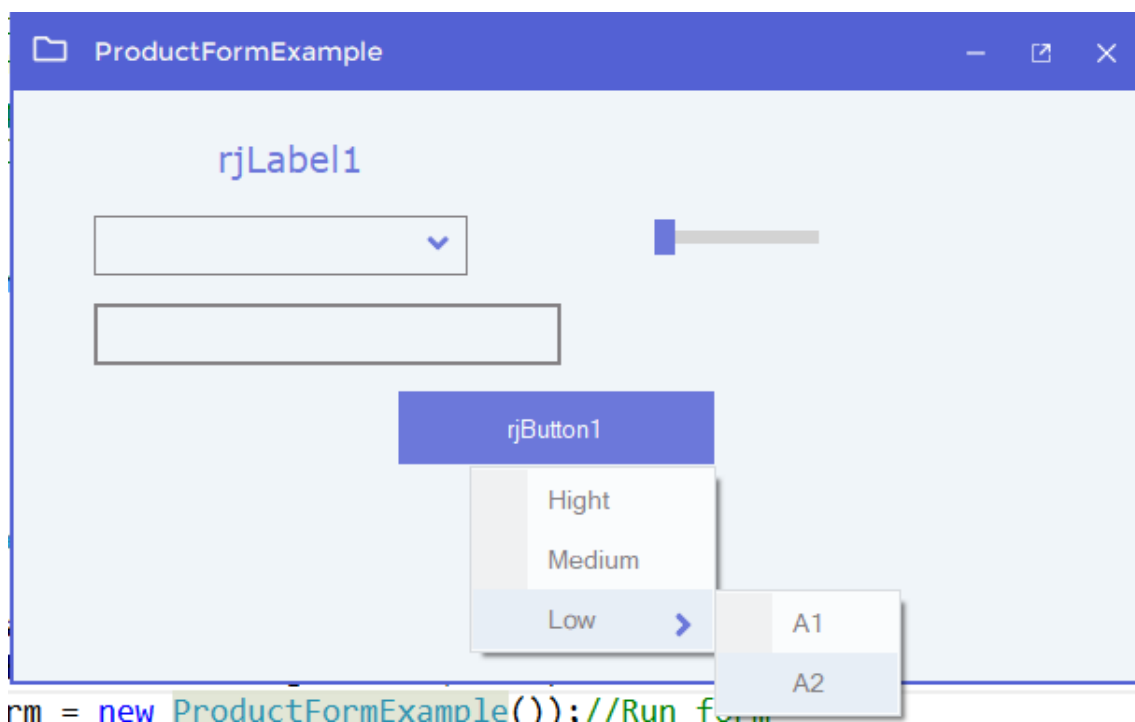
Tal vez te preguntes, **como usar el menú desplegable en los formularios secundarios.** Bueno vuelvo a recordarte que el control **RJDropDownMenu** hereda del control **ContextMenuStrip**, por lo tanto **utiliza el método Show()** del control en cualquier evento de cualquier control, como lo harías normalmente con el control **ContextMenuStrip**. Por ejemplo en el evento Clic del botón **rjBoton1**.

```
ProductFormExample.cs* x ProductFormEx...le.cs [Design]*
RJCodeUI_M1.ProductFormExample rjButton1_Click(object sender, EventArgs e)

9 using System.Windows.Forms;
10
11 namespace RJCodeUI_M1
12 {
13     public partial class ProductFormExample : RJForms.RJChildForm
14     {
15         public ProductFormExample()
16         {
17             InitializeComponent();
18         }
19
20         private void rjButton1_Click(object sender, EventArgs e)
21         {
22             rjDropDownMenu1.Show(rjButton1, DropdownMenuPosition.BottomRight);
23         }
24     }
25 }
```

Esta es la **demonstración del nuevo método Show()** que **facilita la configuración rápida** de la **posición del menú desplegable**. Sin embargo, aún puede usar los otros métodos Show () originales del control ContextMenuStrip.

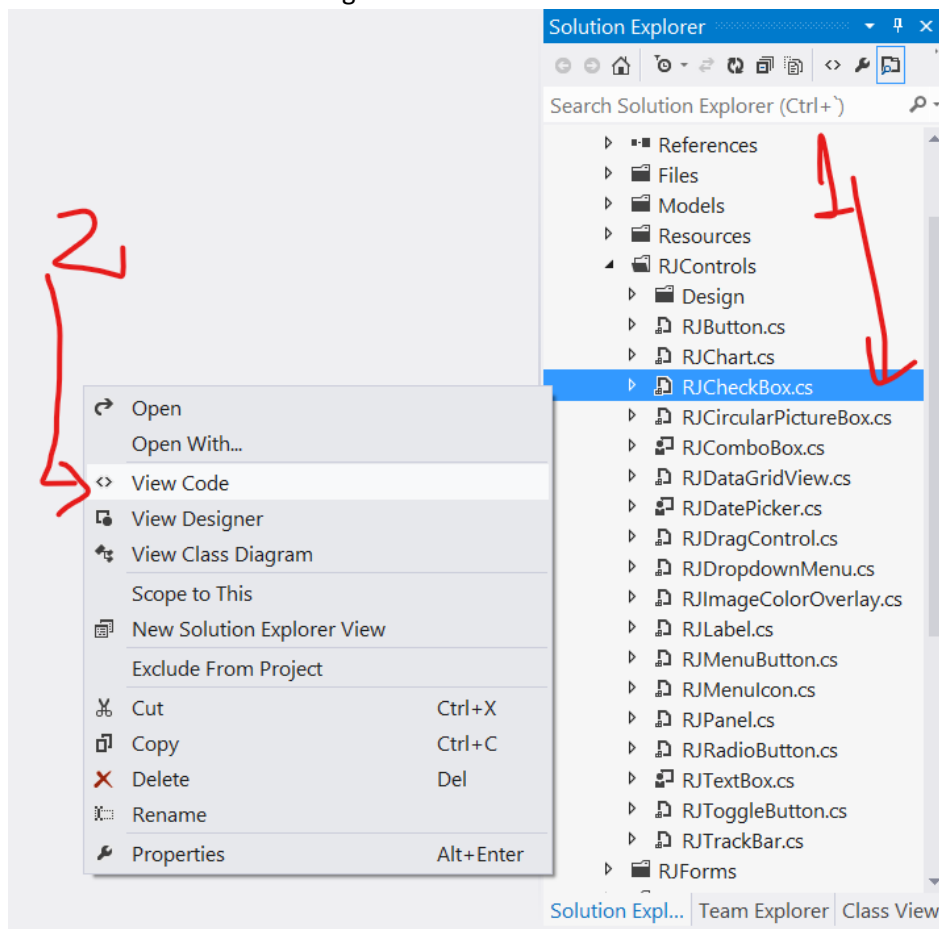
**Resultado:**



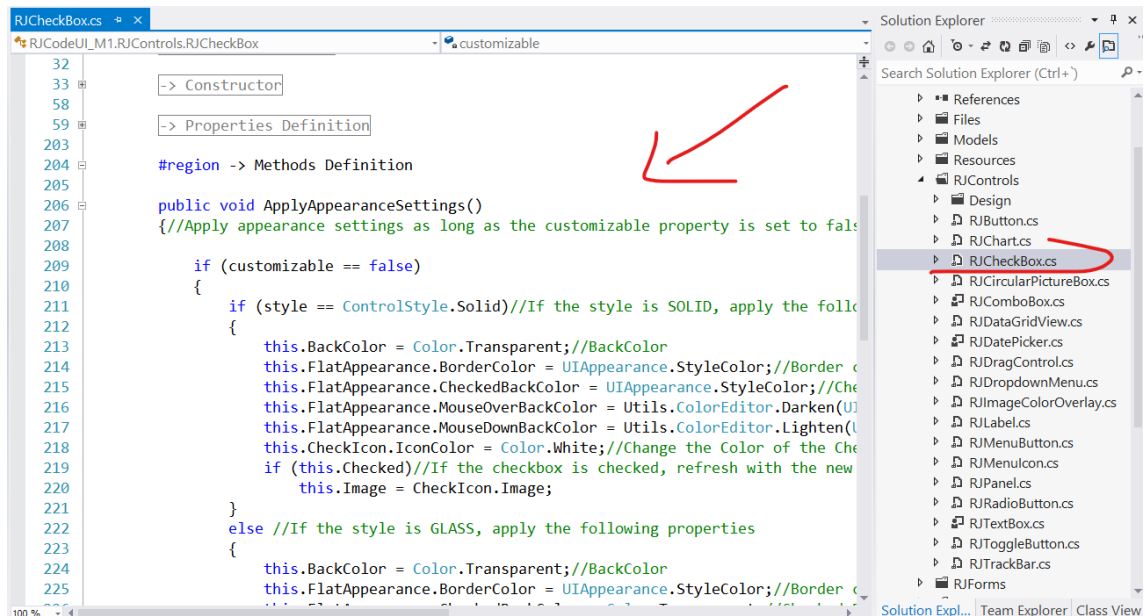
## 5. ¿Cómo modificar los componentes del proyecto?

Puede modificar o agregar propiedades y métodos de cualquier control o formulario personalizado, pero ello simplemente localice (Con ayuda de los diagrama de clases y demostración) la clase-> campo, propiedad o método que desee modificar.

- 1) Para realizar cambios, **abre el código del control o formulario**: Clic derecho sobre la clase-> Seleccionar Ver Código.



- 2) Modifique el código fuente.



**Cada vez que realices cambios en los controles o formularios, debes de Compilar/ Generar/ Construir el proyecto para aplicar los cambios en tiempo de diseño.**

Bueno eso es todo, espero que te haya ayudado con alguna de tus dudas.