

# Unidad 3 - Capítulo 1 - Laboratorio 2

## 1. Crear una ventana con grilla en la capa de presentación de escritorio

### Objetivos

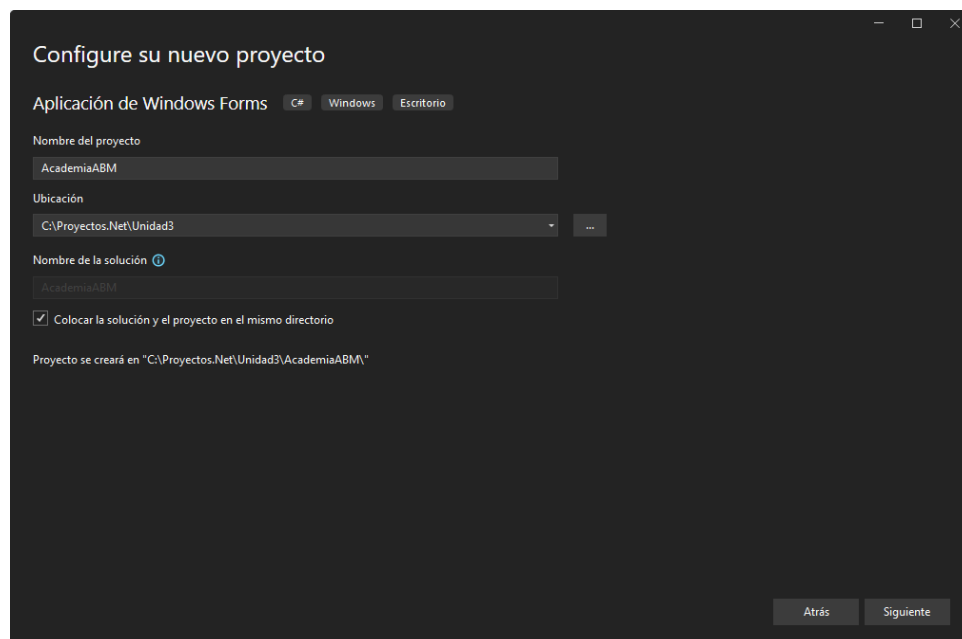
Crear en capa de presentación para una aplicación de escritorio un listado de usuarios utilizando un control grilla.

### Duración Aproximada

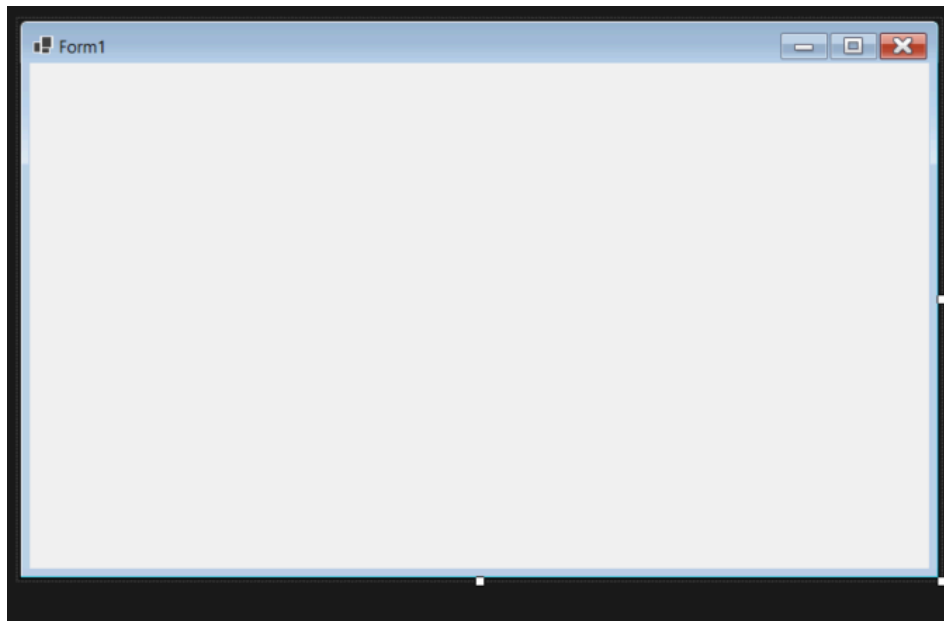
30 minutos

### Pasos

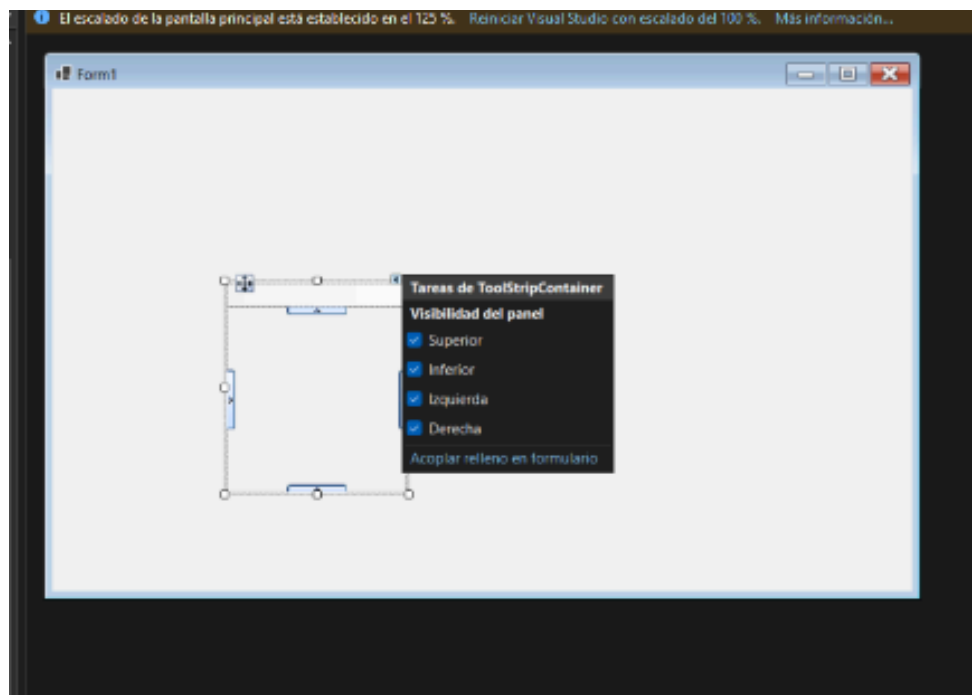
- 1) Abra el Visual Studio 2022 y elija crear un nuevo proyecto.
- 2) Elija un proyecto de la categoría C# -> Aplicación de Windows Forms.
  - a) Completar los siguientes datos:
    - Nombre del Proyecto (Name): AcademiaABM



- Ubicación de la solución (Location): C:\Proyectos\Unidades\Unidad03
- 3) En el explorador de soluciones, dentro del proyecto hacemos doble clic sobre el Form1.cs y se abrirá el diseñador de formularios mostrándonos un Windows Form como el siguiente:



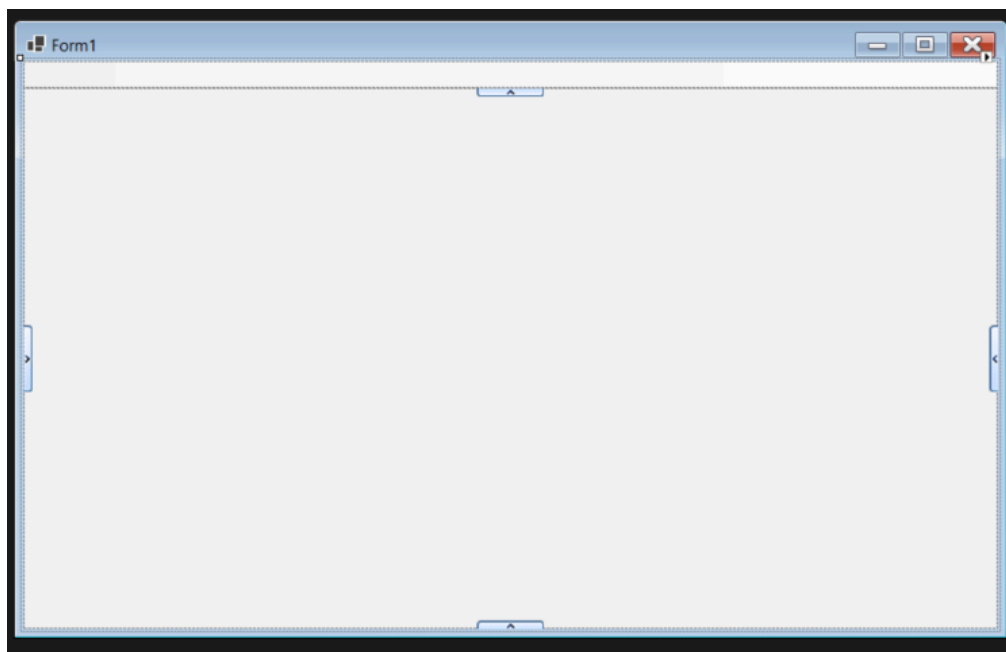
- 4) Luego arrastraremos desde el Cuadro de herramientas o Toolbox un ToolStripContainer que nos permitirá agregar una barra de herramientas y contenido sin que estos se superpongan.



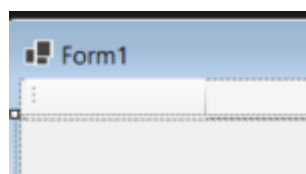
- 5) Teniendo seleccionado el ToolStripContainer vamos a la pestaña de la derecha llamada Propiedades y vemos que se despliega la lista de todas las propiedades que podemos editar desde el diseñador de formularios. A este objeto le modificaremos las propiedades Name y Dock como se visualiza a continuación:

[-] <b>Design</b>	
(Name)	tlUsuarios
GenerateMember	True
Locked	False
Modifiers	Private
[-] <b>Diseño</b>	
Anchor	Top, Left
AutoScroll	False
[-] AutoScrollMargin	0; 0
[-] AutoScrollMinSize	0; 0
AutoSize	False
AutoSizeMode	GrowOnly
ColumnCount	2
Columns	(Collection)
Dock	Fill

La propiedad name nos permite definir con qué nombre haremos referencia al control dentro de nuestro código y la propiedad Dock nos permitirá determinar cómo se ajusta el panel dentro del control que lo contiene (en este caso el formulario). Ahora se verá de esta forma:

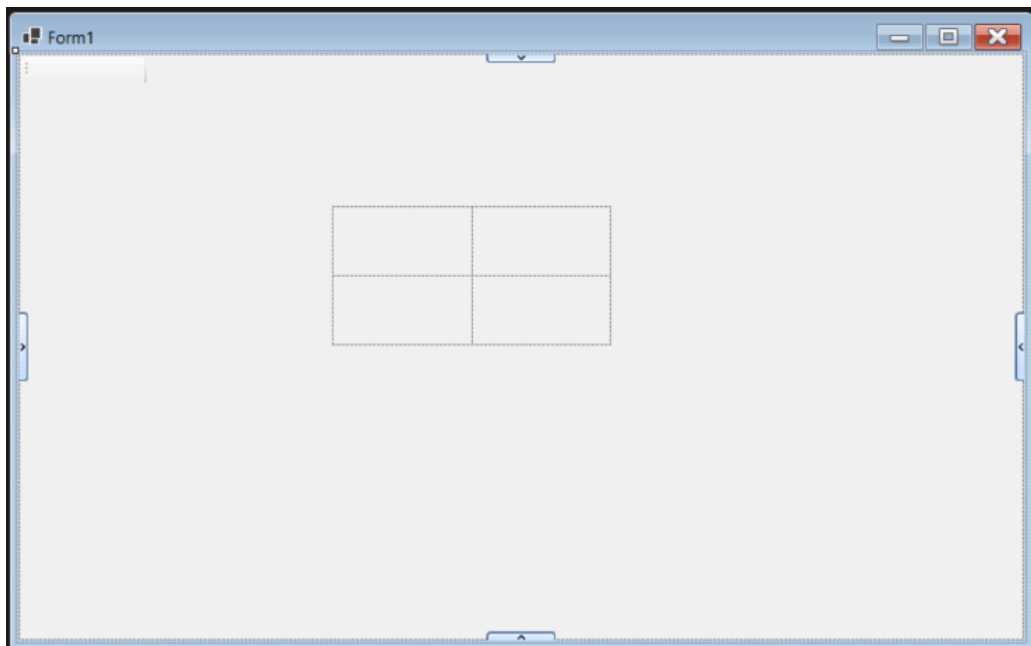


6) Arrastraremos un ToolStrip a la parte superior de tcAcademia para que se vea así:



7) Al toolStrip1 le modificaremos su (Name) por tsUsuarios. Luego continuaremos editando este elemento.

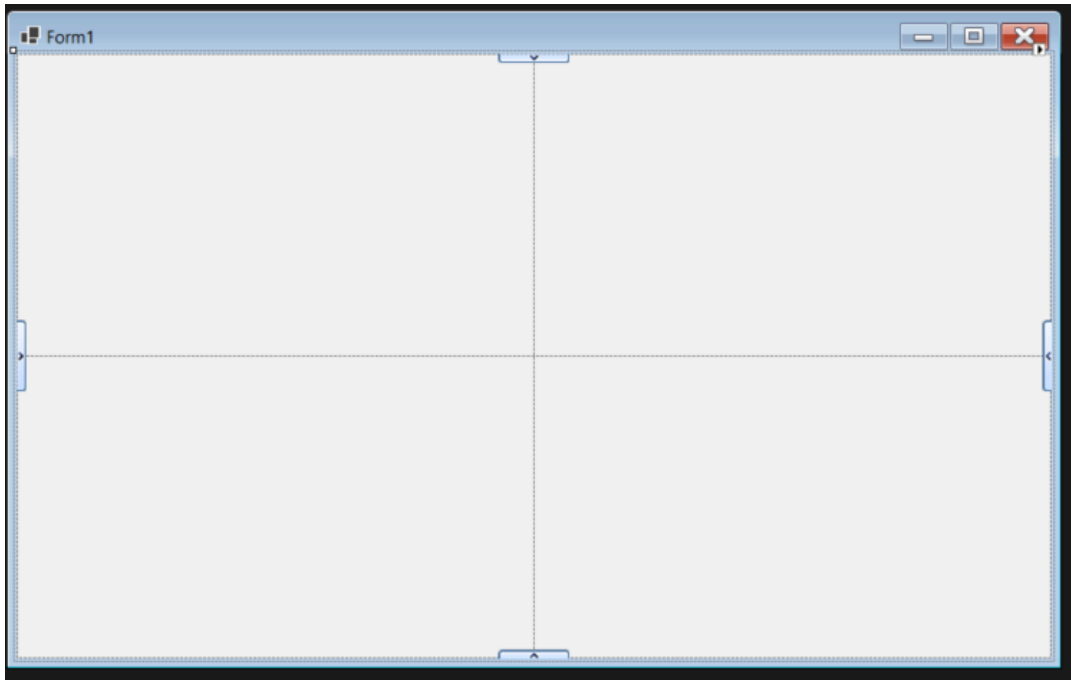
8) Arrastraremos un TableLayoutPanel dentro del tcUsuarios



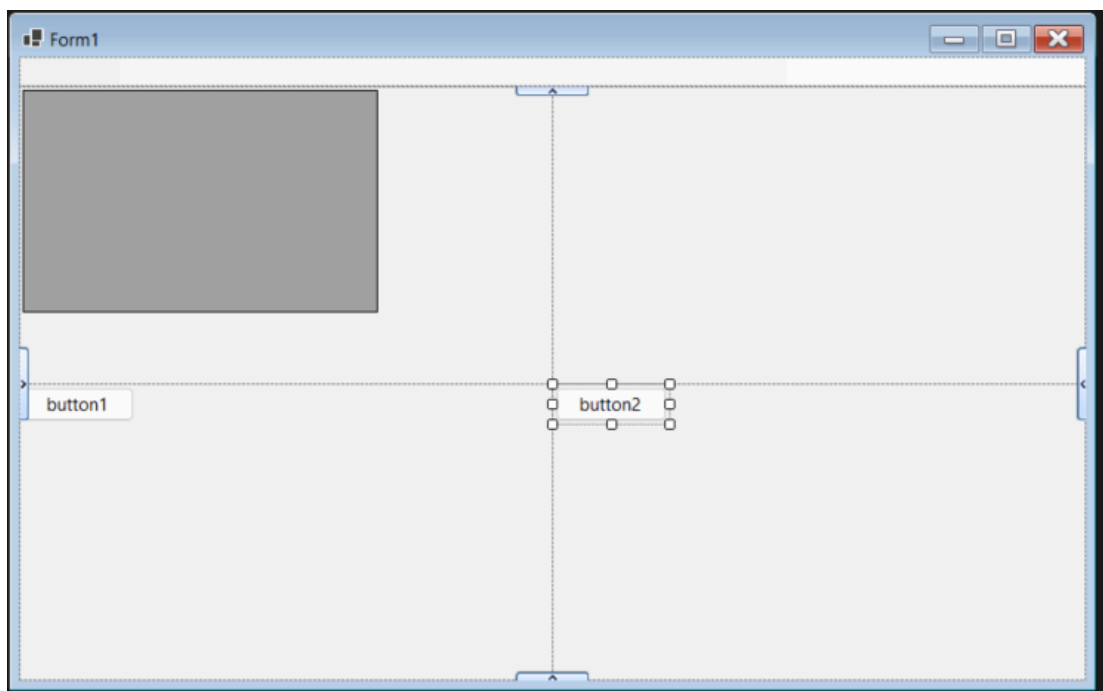
9) A este objeto le modificaremos las propiedades Name y Dock como se visualiza a continuación:

<b>Design</b>	
(Name)	tIAlumnos
GenerateMember	True
Locked	False
Modifiers	Private
<b>Diseño</b>	
Anchor	Top, Left
AutoScroll	False
AutoScrollMargin	0; 0
AutoScrollMinSize	0; 0
AutoSize	False
AutoSizeMode	GrowOnly
ColumnCount	2
Columns	(Collection)
Dock	Fill

10) Entonces se verá así:



- 11) Ahora arrastraremos una DataGridView a la zona superior izquierda y dos Button, uno a cada una de las regiones inferiores.



- 12) Ahora hacemos clic sobre el botón de la izquierda (button1) y modificamos las propiedades Name y Text de la siguiente forma:

<b>Design</b>	
(Name)	btnActualizar
GenerateMember	True
Locked	False
Modifiers	Private
<b>Diseño</b>	

Text	Actualizar
------	------------

- 13) Hacemos clic en el botón de la derecha (button2) y en la ventana de propiedades las modificamos de la siguiente forma:

<b>Design</b>	
(Name)	btnSalir
GenerateMember	True
Locked	False
Modifiers	Private
<b>Diseño</b>	

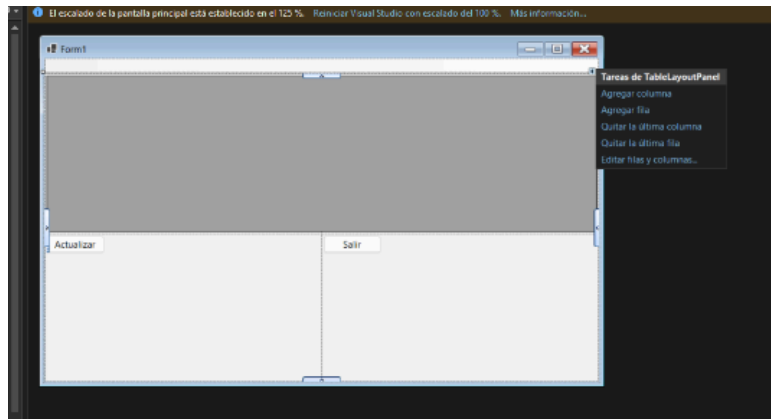
Text	Salir
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True
UseVisualStyleBackColor	True

- 14) Luego hacemos clic sobre la grilla y modificamos las propiedades Name, Dock y ColumnSpan para que queden de esta forma:

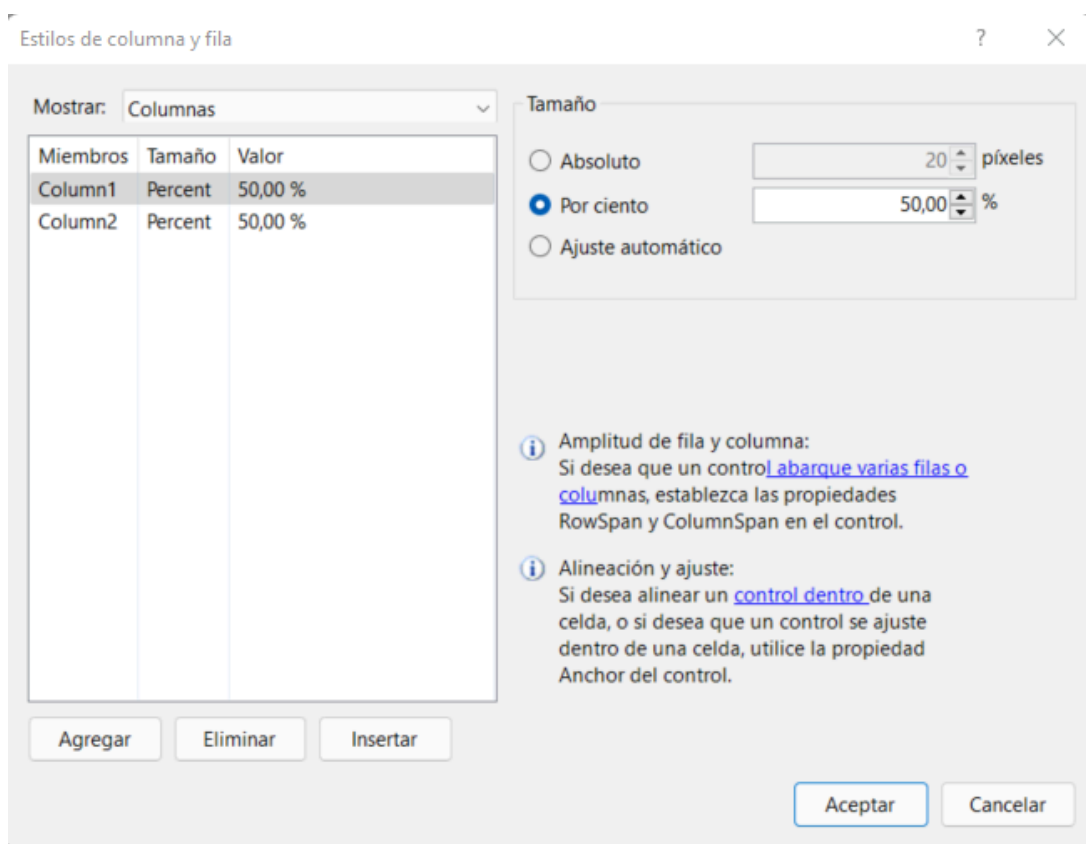
<b>Design</b>	
(Name)	dgvUsuarios
GenerateMember	True
Locked	False
Modifiers	Private
<b>Diseño</b>	
Anchor	Top, Left
AutoSizeColumnsMode	None
AutoSizeRowsMode	None
Cell	0,0
Column	0
ColumnSpan	2
Dock	Fill

Al setear el Dock en Fill vemos que la grilla ocupa todo el espacio que le queda asignado dentro del TableLayoutPanel. Por defecto es 1 fila (RowSpan) y 1 columna (ColumnSpan). Sin embargo como pretendemos que la grilla ocupe todo el ancho indicamos que ocupe 2 columnas de ancho y la propiedad Dock en Fill hará que esta rellene ambas columnas de la fila.

- 15) Ahora modificaremos las columnas y filas del TableLayoutPanel (tlUsuario) para mejorar la estética y permitir que al redimensionar el formulario se siga viendo correctamente la pantalla. Hacemos clic sobre el tlUsuario y en la parte superior derecha de tlUsuarios se encuentra un pequeño cuadrado blanco con una punta flecha negra. Esto se llama smart tag hacemos clic en él (cuidado que no estemos haciendo clic sobre el smart tag de la grilla, para ello tener en cuenta que al desplegar el smart tag diga Tareas de TableLayoutPanel en negrita) y se desplegarán varias opciones.

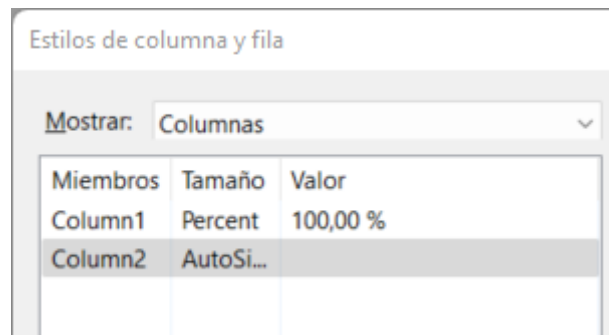


16) Entonces hacemos clic sobre Editar filas y columnas... y se abre la siguiente ventana:

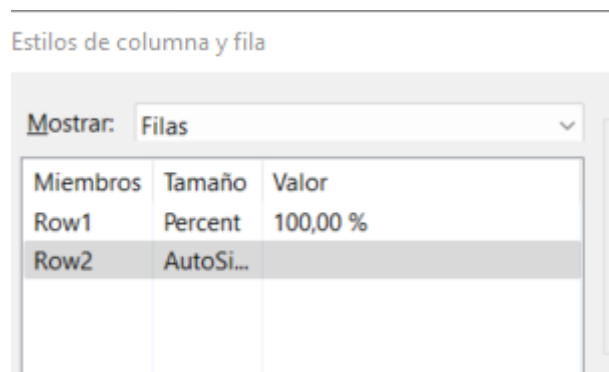


17) Lo que pretendemos es que la segunda columna se adapte automáticamente al tamaño de los controles que contenga (en este caso el botón Salir y que la columna 1 se expanda para cubrir todo el ancho. Para ello, teniendo seleccionada Column1 en la lista de la izquierda vamos al sector de la derecha y donde dice Porcentaje escribimos 100%. Luego hacemos clic en Column2 y en la parte de la derecha elegimos ajuste automático.

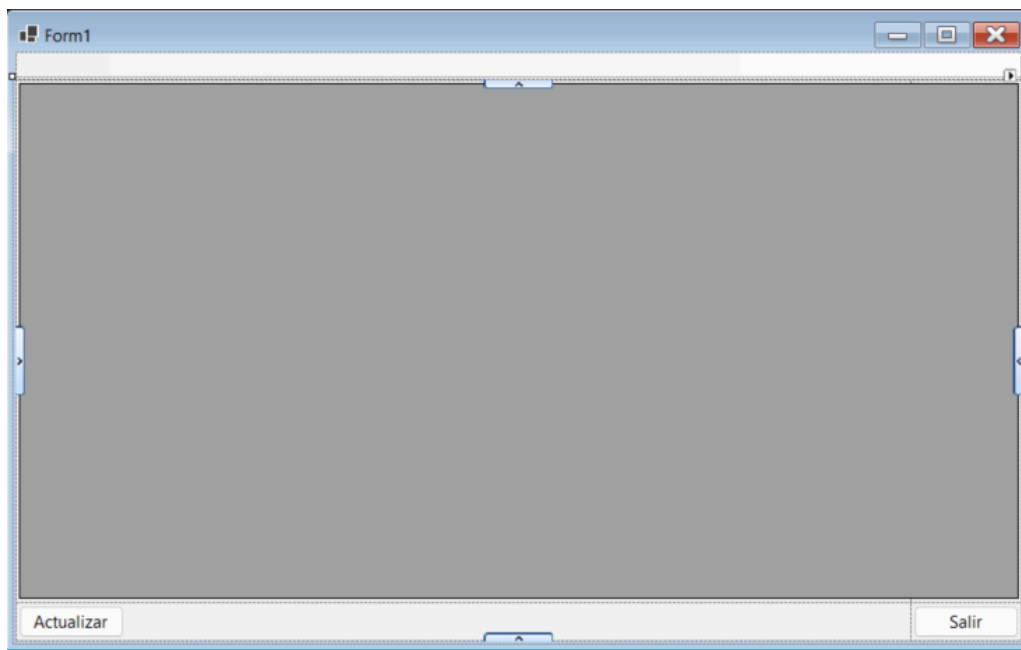
Debería quedar así:



- 18) Luego en el combo de arriba seleccionamos Filas y las modificaremos igual que las columnas para que queden así:

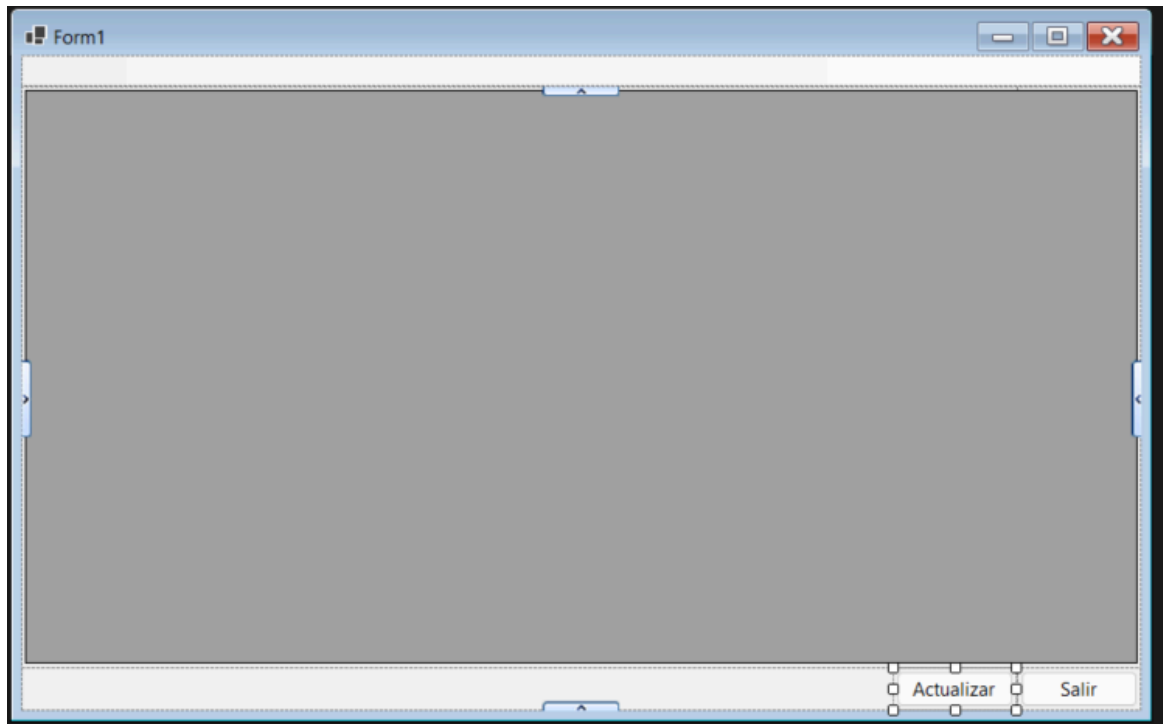


- 19) Ahora presionamos Aceptar y el formulario debería verse así:



- 20) Para que el botón Actualizar quede alineado a la derecha. Hacemos clic sobre actualizar. En la ventana de propiedades vamos a donde dice Anchor. Hacemos clic sobre la flecha que se despliega y dejamos que se quede fijado a la parte superior y a la derecha. Al presionar enter deberá decir Top, Right y verse así:

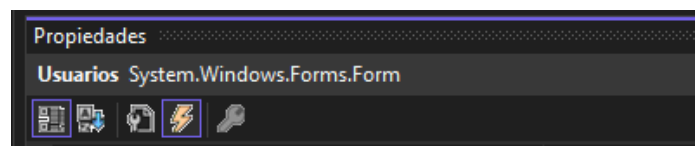




- 21) Ahora haremos clic con el botón derecho sobre el formulario y luego clic en Ver Código.
- 22) Como podemos observar el nombre de la clase es Form1. Entonces procederemos a cambiarlo. Primero nos aseguraremos que el proyecto compile.
- 23) Para renombrar el archivo hacemos clic con el botón derecho sobre el archivo en el explorador de soluciones y clic en Cambiar nombre y lo llamaremos Usuarios.cs (Cuidando de no olvidar la extensión del archivo).
- 24) A continuación crearemos el método public void Listar().

En este método crearemos una List<Usuario>, de la misma forma que lo hicimos en las unidades anteriores, dejando al menos 3 valores hardcoded que tenga la información que se debe mostrar luego en la grilla (detallado en punto 30)

- 25) Acto seguido volvemos al diseñador de formularios. Seleccionamos el formulario y en la ventana de propiedades cambiamos la propiedad Text por Usuarios. Y luego en la parte superior hacemos clic sobre el botón de eventos. Ahora en lugar de ver la lista de propiedades, estamos visualizando la listas de eventos del control seleccionado.



- 26) Buscamos el Evento Load y hacemos doble clic sobre él. Automáticamente generará el manejador del evento y el método Usuarios\_Load para manejarlo.
- 27) En el evento Load invocamos al método Listar().
- 28) Luego en el diseñador de formularios seleccionamos el botón actualizar y en la lista de eventos hacemos doble clic sobre el evento Click. Automáticamente se generará el manejador del evento

y el método Actualizar\_Click que será invocado por dicho manejador. En este método también invocamos a Listar().

**Aclaración:** Actualmente los sistemas son multiusuario y por lo tanto los datos pueden ser accedidos y modificados por varios usuarios en forma simultánea. Esto hace necesario que resulte conveniente contar con un mecanismo para que un usuario pueda actualizar la información que ve en pantalla, ya que la misma pudo haber sufrido cambios a mano de otro usuario.

Para esto pueden utilizarse diversos mecanismos. Desde los más simples como cerrar la ventana y volverla a abrir, hacerlo mediante un botón de actualizar o que el sistema lo haga en forma automática ya sea por tiempo o al ocurrir un evento. Aquí se eligió este mecanismo por ser fácil de implementar y comprender.

29) Luego en el diseñador de formularios en el botón Salir agregamos el evento Click y en el código del mismo escribimos this.Close();

30) Investigar cómo modificar las propiedades para para que grilla quede de la siguiente forma:

- Sea de solo lectura
- No permita agregar ni eliminar filas manualmente
- No agregue columnas automáticamente (propiedad AutoGenerateColumns)
- Cuento con las siguientes columnas:

Nombre Columna	Cabecera	Origen de datos	Tipo Columna
id	ID	ID	Texto
nombre	Nombre	Nombre	Texto
apellido	Apellido	Apellido	Texto
usuario	Usuario	Usuario	Texto
email	EMail	EMail	Texto
habilitado	Habilitado	Habilitado	Check

31) Hacemos clic sobre el ToolStrip tsUsuarios. Aparecerá un botón con una lista desplegable, allí elegimos Button

32) Aparecerá un botón con un ícono genérico. En el mismo modificar las propiedades Name y ToolTipText por tsbNuevo y Nuevo respectivamente

ToolTipText es el texto de ayuda que aparece sobre el control cuando ponemos el mouse sobre él.

33) Reemplazar la imagen por otra acorde a la acción utilizando la propiedad Image

34) Repetir la operación para los botones Editar y Eliminar.