

## **1Actividad 2: Práctica Desarrollo de aplicaciones en .Net**

Mario Andres Moreno romero y Camilo Alejandro Moreno Piñeros  
Junio 2024.

Fundación Universitaria Internacional de la Rioja  
Nombre del departamento.  
Plataformas de Desarrollo de Software

## **Tabla de Contenidos**

FARMACIA UNIR.....	1
VENTANAS DE FUNCIONAMIENTO PROCESO CORRECTO .....	1
Gestión de errores .....	1
BREAKPOINT. ....	3

## FARMACIA UNIR

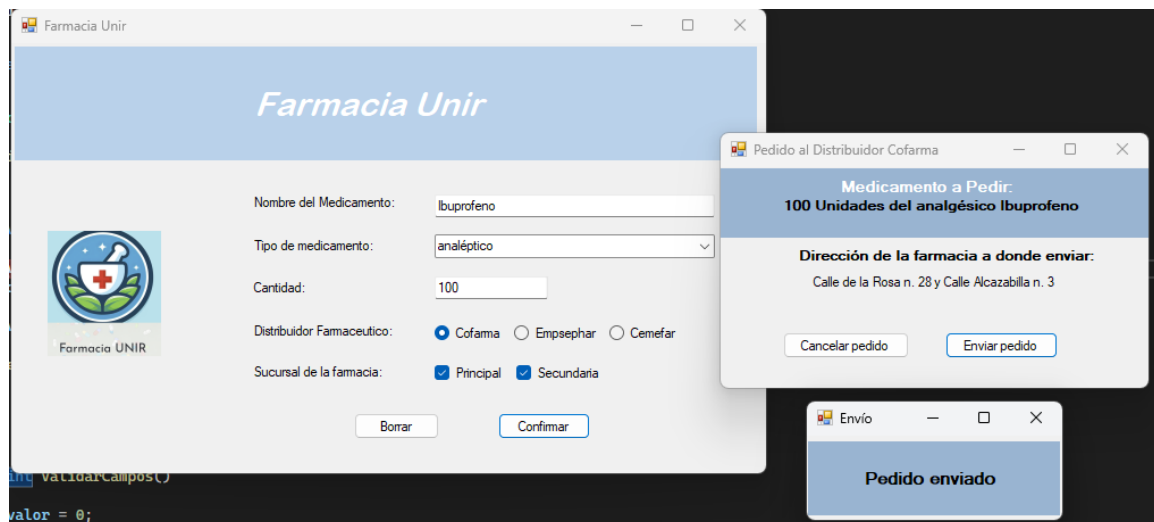
Repositorio : <https://github.com/maaiacos/UnirPlataformasDesarrollo>

### VENTANAS DE FUNCIONAMIENTO PROCESO CORRECTO

En este espacio se presenta la ejecución de todo el flujo correcto de la aplicación en ejecución, validando todos los datos de entrada y el envío de información entre formularios. El formulario principal contiene todos los campos solicitados, validación de errores al ingreso de datos para tipo de dato, también tiene las acciones en los botones confirmar y borrar.

El segundo formulario contiene el resumen solicitado en cada punto, mostrando los datos enviados desde el primer formulario.

Por último, cuando se da click en el botón enviar pedido del segundo formulario muestra una nueva pantalla con el mensaje Pedido enviado.



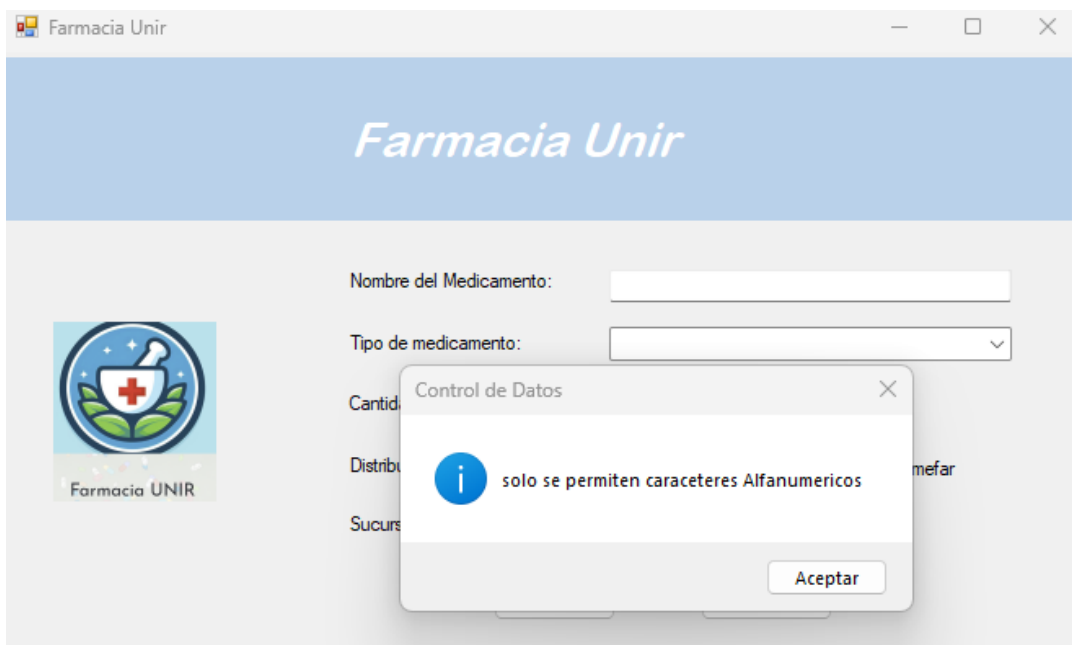
### Gestión de errores

Inicialmente se valida que en los campos solo se puedan ingresar caracteres adecuados para el comportamiento del sistema y de acuerdo con la información que debe recibir cada uno.

Para el campo nombre de medicamento se realiza la validación que solo permita el ingreso de caracteres alfanuméricos y en el campo cantidad del medicamento valida ingreso numérico, para esto en el evento keypress del campo se realiza la invocación de

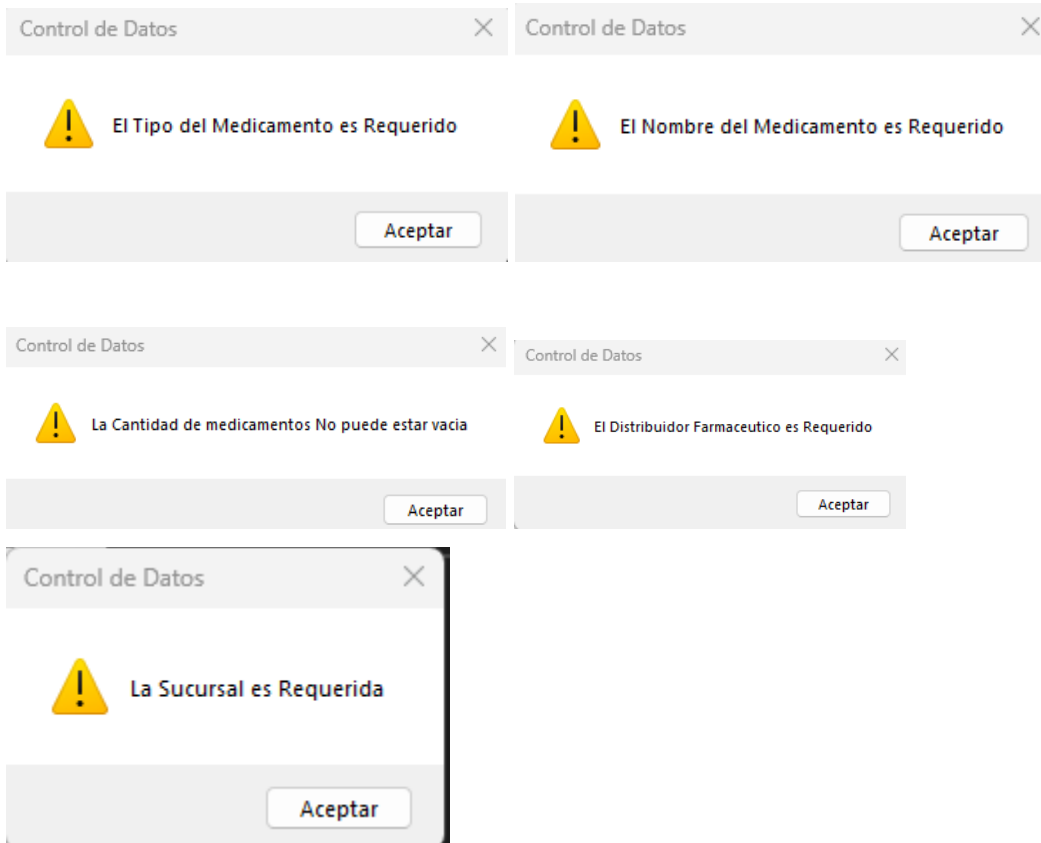
un método creado que al recibir el carácter digitado valida con las funciones isDigit, isControl, Isletter.

En el caso de que el carácter digitado no corresponda a uno válido se maneja la salida de errores usando el MessageBox, adicionando un mensaje predeterminado, un Título para el error Control de Datos, un botón para aceptar el mensaje y un icono de Información.



El siguiente bloque de control de errores se utilizó en el botón confirmar, en este botón se valida que todos los campos se encuentren diligenciados.

En el caso de alguno de los campos no se encuentre diligenciado la salida de errores usando el MessageBox, adicionando un mensaje predeterminado para cada campo, un Título para el error Control de Datos, un botón para aceptar el mensaje y un icono de warning.



## **BREAKPOINT.**

Al dar click en el botón confirmar de la primera ventana del aplicativo se agregó un breakpoint en la asignación de unas de las variables de trabajo que se usan.

El punto de control se puede agregar dando click con el botón izquierdo del mouse sobre la barra lateral izquierda en el formulario.

```
20 }
21
22 1 referencia
private void btnConfirmar_Click(object sender, EventArgs e)
23 {
24     int validar = 0; ≤ 623.018 ms transcurridos
25     validar = ValidarCampos();
26
27     if (validar == 0)
28     {
29         enviarPedido();
30     }
31 }
32
33 1 referencia
private int ValidarCampos()
34 {
```

Visual Studio Code interface showing a C# code file named `Form1.cs` in the `Farmacia_Unir` namespace. The code is in the `Debug` state, running on `Any CPU`. The process is `Farmacia_Unir.exe` (PID 36696), and the sub-process is `Subproceso principal` (PID 116296). The stack frame is `Farmacia_Unir.Form1.btnConfirmar_Click`.

```
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11 using System.Windows.Forms.VisualStyles;
12
13 namespace Farmacia_Unir
14 {
15     public partial class Form1 : Form
16     {
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         private void btnConfirmar_Click(object sender, EventArgs e)
23         {
24             int validar = 0;
25             validar = ValidarCampos();
26         }
27
28         private int ValidarCampos()
29         {
30             int valor = 0;
31
32             try
33             {
34                 if (string.IsNullOrWhiteSpace(textNomMed.Text))
35                 {
36                     // ...
37                 }
38             }
39             catch { }
40         }
41     }
42 }
```

A context menu is open over line 24, showing options:

- Insertar punto de interrupción condicional
- Insertar punto de seguimiento
- Insertar punto de interrupción temporal
- Insertar Punto de ruptura dependiente