

DESARROLLO DE APLICACIONES EN .NET (C#)

Juan Carlos Muñoz Castañeda
Julio 2023.

Fundación Universitaria Internacional de la Rioja
Especialización en Ingeniería de Software
Plataformas de Desarrollo de Software

Tabla de Contenidos

INTRODUCCION	3
METODO MAIN	4
GESTION DE ERRORES	6
BORRAR DATOS	9
MODO DEPURACION	10
CONCLUSION	11
BIBLIOGRAFÍA	12

INTRODUCCION

C# creado por Microsoft en 1999 (Tokio, 16), llamado al inicio como COOL (C Object Oriented Language) como su Sigla lo indica, Lenguaje C orientado a Objetos, el cual corre o se ejecuta junto con el Framework .Net también creado por Microsoft (AWS, s.f.) que permite la creación de aplicaciones de escritorio, web y móviles (Microsoft, 04).

C# es preferiblemente utilizado con el IDE desarrollado por su creador (Microsoft) conocido como Visual Studio, el cual maneja plantillas predeterminadas y en el mismo se puede editar código no solo de C# , sino de otros lenguajes como F#, VB, Python, entre otros.

Igualmente al ser un producto de Microsoft permite la integración con herramientas de la misma marca tales como GitHub, Azure Etc. (Microsoft, s.f.).

La presente aplicación esta desarrollada en C# usando el Framework .Net y el Editor de Código Visual Studio. La separación de los ficheros del proyecto es la escogida por defecto por el editor en donde separa el código de la interfaz de la lógica de la misma.

El repositorio en donde lo aloje en GitHub <https://github.com/juancmu/Pharmacy-CS.git> .

METODO MAIN

La aplicación tiene un fichero denominado Program.cs que hace el llamado al primer formulario MainForm en donde se encuentra también parte de la lógica de la aplicación:

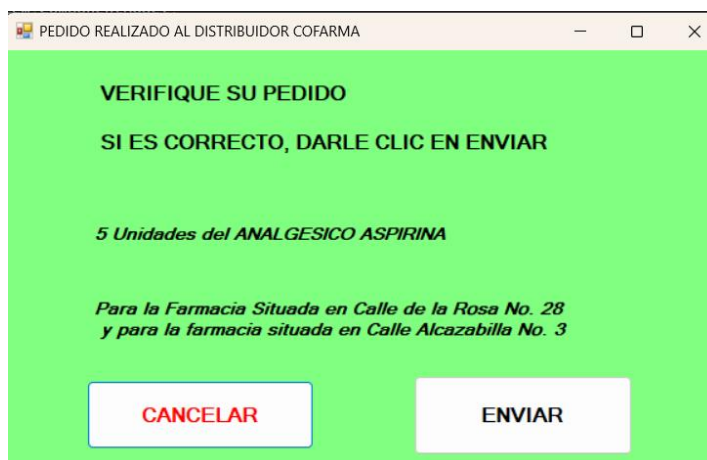
```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new MainForm());
}
```

Figura 1. Formulario Principal de la Aplicación



Una vez recibido los datos pasando los validadores , estos son desplegados en una nueva ventana llamada `Confirm_For`:

Figura 2. Formulario de respuesta



Igualmente conforme a lo requerido, una vez se verifique el pedido al decirle “Enviar”, el programa muestra un mensaje a un label dentro del mismo formulario y un mensaje por consola.

Figura 3. Confirmación del pedido y mensaje por consola.



GESTION DE ERRORES

Se realizaron comprobaciones en cada campo del formulario principal, las cuales se describen en cada elemento, una vez capturada el error o la inconsistencia, se envía al usuario un formulario indicando los errores, apareciendo también una etiqueta en el formulario principal al lado de cada campo que tiene la inconsistencia, estas son recogidas en un arreglo y solo muestra la inconsistencia de los campos con observaciones:

1. **Nombre del medicamento:** se realizaron comprobación no permitir el ingreso de caracteres especiales o números y también que el campo no estuviese vacío. La validación sobre no permitir caracteres especiales o números se realiza a medida que el usuario intenta escribirlas y arrojando un aviso a una etiqueta del mismo formulario mientras el campo tiene el enfoque, mediante el siguiente código:

```
public void onlyCharacters(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar >= 33 && e.KeyChar <= 64 || e.KeyChar >= 91 && e.KeyChar
<= 96 || e.KeyChar >= 123 && e.KeyChar <= 255)
    {
        this.error1.Text = "Favor ingresar solo letras";
        e.Handled = true;
        return;
    }
    else
    {
        this.error1.Text = "";
    }
}
```

Figura 4. Mensaje en formulario , controla caracteres especiales y numeros.



Nombre del Medicamento	<input type="text"/>	Favor ingresar solo letras
---------------------------	----------------------	----------------------------

Figura 5. Muestra de formulario indicando las inconsistencia en los datos y en el formulario principal

The screenshot shows a web application titled "ACTIVIDAD 2 PLATAFORMAS DE DESARROLLO DE SOFTWARE - ESTUDIANTE: JUAN CARLOS MUNOZ". The main form is titled "SELECCIONE EL MEDICAMENTO:" and contains the following fields and controls:

- Nombre del Medicamento:** A text input field with a red error message "Ingrese el nombre del medicamento".
- Tipo de Medicamento:** A dropdown menu with a red error message "Seleccione el tipo de medicamento".
- Cantidad del Medicamento:** A text input field with a red error message "Ingrese la cantidad".
- Distribuidor:** Three radio buttons labeled "COFARMA", "EMPSEPHAR", and "CEMEFAR". A red error message "Favor seleccionar el Distribuidor" is displayed.
- Sucursal de Envio:** Two checkboxes labeled "PRINCIPAL" and "SECUNDARIA". A red error message "Debe seleccionar al menos una sucursal" is displayed.
- Buttons:** "BORRAR" and "CONFIRMAR".

A summary window titled "Hay Inconsistencias en su pedido, revisar:" is open, listing the following errors:

- El nombre del medicamento esta vacio
- No ha seleccionado tipo de medicamento
- No ha ingresado la cantidad de medicamento
- No ha Seleccionado el distribuidor
- Debe seleccionar al menos una sucursal

A "Corregir" button is located at the bottom of the summary window.

2. Tipo de medicamento: la ser de tipo ComboBox, el valor por defecto esta vacío, por lo que hace la verificación que el campo algún tipo se encuentre seleccionado (ver figura 5).
3. Cantidad: este campo tipo texto tiene dos tipo de validaciones:
 - a. La primera que se ejecuta mientras el usuario ingresa la información impidiendo que pueda ingresar caracteres que no sean números, mediante el siguiente código:

```
public void onlyinteger(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar >= 32 && e.KeyChar <= 47 || e.KeyChar >= 58 && e.KeyChar
<= 255)
    {
        this.error3.Text = "Favor ingresar solo Numeros";
        e.Handled = true;
        return;
    }
    else
    {
        this.error3.Text = "";
    }
}
```

Figura 6. Mensaje en formulario , controla caracteres especiales y letras.

This close-up shows the "Cantidad del Medicamento" label and a text input field. To the right of the field, a red error message reads "Favor ingresar solo Numeros".

- b. La segunda validación se realiza igual que las anteriores al confirmar el envío del formulario para el campo no debe estar vacío (ver figura 5 o 6).
4. Distribuidor: igualmente se verifica que un de los RadioButtons este seleccionado, excluyendo la múltiple selección agregando los RadioButton en un GroupBox, enviando el mensaje con la ventana emergente y el formulario principal (ver figura 5).
5. Sucursal: Se verifica la selección de al menos uno de los dos CheckBox (ver figura 5).

Los errores son recogidos, mostrados en la ventana emergente y formulario principal, para que el usuario pueda retornar y corregir los mismos, una vez el usuario comienza a ingresar valores, los mensajes en rojo dentro del formulario principal al lado de cada input van desapareciendo a la par que va corrigiendo:

Figura 7. El mensaje de inconsistencia desaparece cuando es corregido.

SELECCIONE EL MEDICAMENTO:

Nombre del Medicamento:

Tipo de Medicamento:

Cantidad del Medicamento:

Distribuidor:

☐ COFARMA ☐ EMPSEPHAR ☐ CEMEFAR

Sucursal de Envio:

☐ PRINCIPAL ☐ SECUNDARIA

Favor seleccionar el Distribuidor

Debe seleccionar al menos una sucursal

BORRAR DATOS

El usuario al ejecutar el botón de borrar, el formulario principal retorna los valores por defecto de cada input y en caso de haber mensajes de errores anteriores, estos también son borrados mediante el siguiente código:

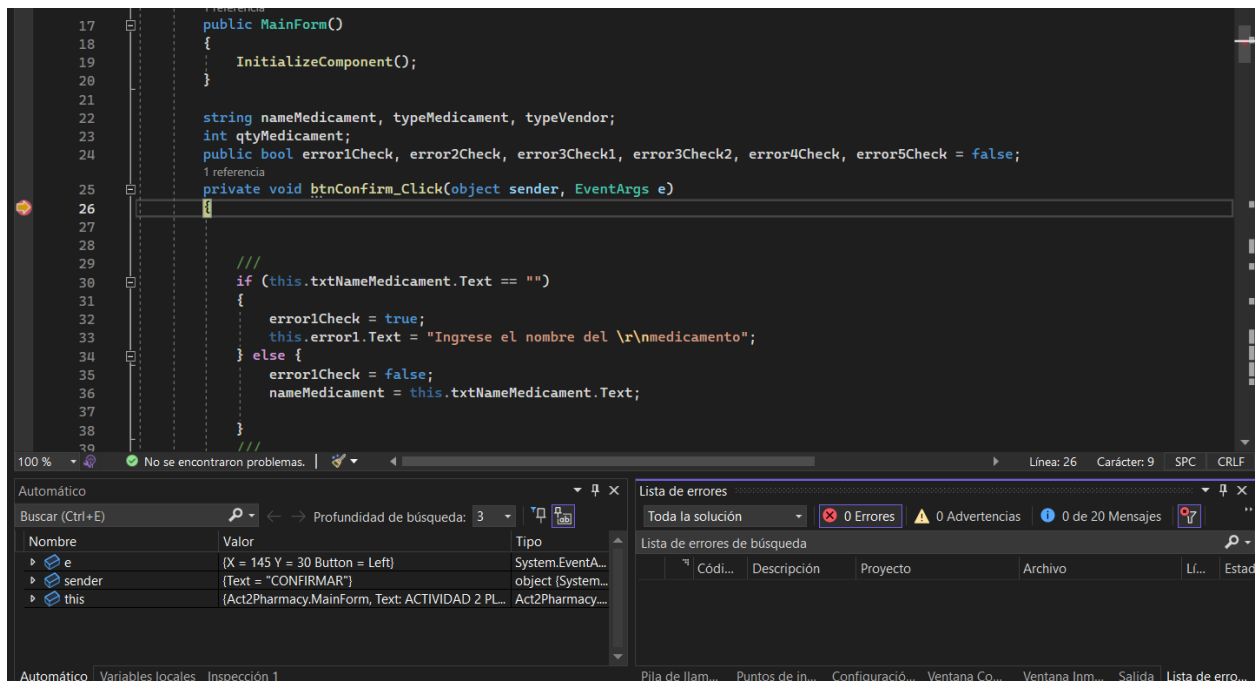
```
private void btnErase_Click(object sender, EventArgs e)
{
    this.txtNameMedicament.Text = "";
    this.cmbBoxTypeMedicament.SelectedIndex = -1;
    this.txtQtyMedicament.Text = "";
    this.rBtnCofarma.Checked = false;
    this.rBtnEmpsephar.Checked = false;
    this.rBtnCemefar.Checked = false;
    this.chkBoxSecondVendor.Checked = false;
    this.chkBoxMainVendor.Checked = false;
    this.error1.Text = "";
    this.error2.Text = "";
    this.error3.Text = "";
    this.error4.Text = "";
    this.error5.Text = "";
}
```

La primera parte como se evidencia, retorna los valores de cada input a su valor por defecto y los campos donde se muestran las inconsistencias de cada campo.

MODO DEPURACION

Conforme a lo indicado en los requerimientos de la actividad, se coloca un BreakPoint en el botón de “Confirmar” de la ventana principal y se ejecuto en modo de depuración, arrojando el siguiente resultado:

Figura 8. Screen de la ejecución en modo de depuración



CONCLUSION

La sintaxis entre JAVA y C# es bastante parecida y la implementación de elementos de los formularios es bastante similar entre Eclipse y Visual Studio, igual que JAVA, C# también tiene fuerte tipado hace se tenga mayor control sobre lo escrito siendo menos verboso que JAVA.

Visual Studio realiza la separación del código entre el diseño y la lógica, ayudando al desarrollador a tener un código mas ordenado.

BIBLIOGRAFÍA

- AWS. (s.f.). Obtenido de <https://aws.amazon.com/es/what-is/net/#:~:text=NET%20es%20una%20plataforma%20de,escalable%20y%20de%20alto%20rendimiento.>
- Microsoft. (s.f.). Obtenido de <https://visualstudio.microsoft.com/es/vs/>
- Microsoft. (2023 de 05 de 04). Obtenido de <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- Tokio. (2018 de 07 de 16). *Tokio School*. Obtenido de <https://www.tokioschool.com/noticias/c-que-es/#:~:text=Andr%C3%A9s%20Hejlsberg%20decidi%C3%B3%20formar%20un,de%20programaci%C3%B3n%20orientado%20a%20objetivos.>