

UD2. PROGRAMACIÓ DE INTERFACES. PROGRAMACIÓ EN C#



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Módulo: Desarrollo de interfaces. Curso 2023-24

Objetivos

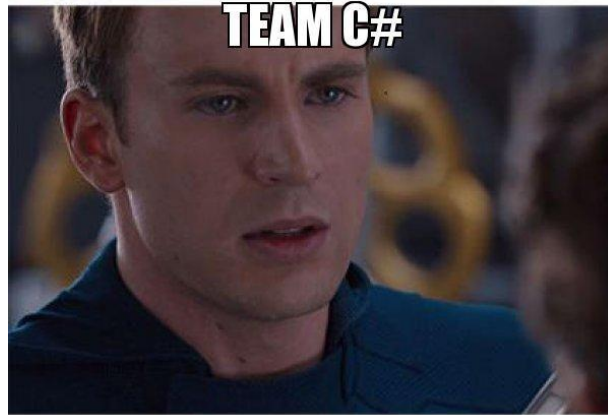
- .Reconocer la importancia de un lenguaje de programación (C#) para la implementación de una interfaz.
- .Identificar las diferencias y similitudes entre el lenguaje Java y C#
- .Analizar los distintos entornos de desarrollo y en particular el IDE Visual Studio.

De Java a .Net

1. Tras un primer año de introducción a la programación en lenguaje Java, en este segundo curso se continuará la programación en dicho lenguaje en algunas de las asignaturas. Sin embargo en otras vamos a completar el perfil programador del alumno introduciéndonos en otra gran tecnología de programación ampliamente extendida hoy en día que es el Framework .Net de Microsoft.
2. Además MS está realizando un esfuerzo y uniéndose con la comunidad de software libre para convertir este Framework en realmente multiplataforma a través de proyectos como Mono y Xamarin.

Como ya conocemos un lenguaje, unas librerías y un entorno de desarrollo no partiremos de cero en esta nueva herramienta. Nos basaremos en lo aprendido el año pasado para facilitar en gran medida el salto de Java a C# que será el lenguaje .Net que trabajaremos (Además de ser el más parecido a Java, lo que es una ventaja).

Introduciremos en este capítulo en IDE Visual Studio Community y luego las bases del lenguaje C# de una forma muy ágil partir de las mismas bases de Java.



El IDE Visual Studio

En el mundo de la programación .Net existen varios IDEs, nosotros concretamente vamos a trabajar con el Visual Studio de Microsoft. Si accedemos a la web de MS para descargarlo:

<https://www.visualstudio.com/es-es/downloads/download-visual-studio-vs.aspx>

encontramos las siguientes posibilidades:

- Community Edition: Versión gratuita que permite desarrollar aplicaciones .Net en distintos lenguajes y distintos sistemas operativos. Realmente en sistemas Unix (linux y macOS) es una adaptación del proyecto Xamarin.
- Enterprise: Versión de pago con herramientas avanzadas para el desarrollo. Existe una versión intermedia (Professional) de pago pero más barata y con menos herramientas pensada para equipos pequeños.
- Code: Editor muy potente que tiene posibilidades de reconocimiento sintáctico, control de versiones, depuración y admite extensiones. Es multiplataforma y lo encontramos para Linux, Mac OS y Windows.

En <https://www.visualstudio.com/es/vs/compare/?rr=https%3A%2F%2Fwww.google.es%2F> se pueden ver esquemáticamente las diferencias.

Se puede ver una comparativa de las distintas versiones de Visual Studio y su evolución histórica aquí:

http://en.wikipedia.org/wiki/Microsoft_Visual_Studio#Editions_feature_grid

Como curiosidad en la siguiente Web se pueden ver los precios de la herramienta completa:

<https://www.microsoft.com/es-es/store/collections/visualstudio>

Uso básico

Veamos mediante un ejemplo guiado cómo es el ciclo de creación y depuración de una aplicación en Visual Studio .

- Ejecuta la aplicación Visual Studio. Tras ejecutar el IDE nos sale una ventana principal en la que nos permite crear un proyecto nuevo o abrir uno reciente. Por supuesto esto se puede hacer también desde el menú de la aplicación.
- Crea un nuevo proyecto. Al crear un nuevo proyecto nos sale la posibilidad de varios tipos de proyecto. Esto configura de forma automática el entorno para que tenga acceso a las librerías que podamos necesitar. Inicialmente trabajaremos con dos de estas posibilidades: Aplicación de consola para realizar programas ejecutables en línea de comandos (sin gráficos) y Aplicación de Windows Forms para realizar aplicaciones de escritorio. Más adelante veremos otras posibilidades según nos vayan haciendo falta.

<https://docs.microsoft.com/es-es/xamarin/android/get-started/>

<https://blogs.msdn.microsoft.com/esmsdn/2017/05/09/guia-para-principiantes-de-xamarin/>

<https://thatcsharpguy.github.io/post/xamarin-como-empiezo/>

- Filtra por Windows y Desktop.
- Selecciona Aplicación de Windows Forms en C# (Ojo porque existe también en VB) para este ejemplo. Dale como nombre “NavegadorWeb”.
- Pincha en el Explorador de Soluciones en Form1.cs. Se desplegará y podrás seleccionar en la barra de herramientas superior o en el menú contextual “Ver Código”. Veremos, explicaremos y analizaremos más adelante todo el código generado por el diseñador.
- A la izquierda está oculto el “Cuadro de herramientas”. Pincha sobre él y déjalo fijo si te resulta más cómodo mediante el icono de la chincheta. Si se tiene seleccionado el diseño del formulario, en este cuadro se verán los distintos componentes que se pueden colocar en el formulario. Prueba a arrastrar algún botón o caja de edición.

- Coloca una etiqueta (Label), una caja de texto (TextBox) y un botón (Button) en el formulario. Cada uno de estos elementos como ya sabes es un objeto con sus propiedades.
- Cambia el campo Text de la etiqueta (URL) y el del botón (Allá voy!).
- Coloca también un componente WebBrowser. Ajústalo de forma que ocupe la parte inferior del formulario y cambia la propiedad Anchor de forma que incorpore los valores Top, Left, Right y Bottom. Esto hace que al cambiar el tamaño del formulario también cambie el del componente pues se mantiene la distancia a cada punto de anclaje.
- Ejecuta pulsando el icono Play o F5 o en el menú Depurar → Iniciar Depuración. Verás que se ejecuta nuestro diseño aunque por ahora no haga nada: Hemos situado los componentes pero no hemos programado los Eventos a los que tienen que responder.

- Haz doble clic sobre el botón. Aparece la pantalla de código y en concreto aparece el método siguiente:

```
private void button1_Click(object sender, EventArgs e)
{
}
```

- pruébalo añadiendo esta línea de código:
webBrowser1.Navigate(textBox1.Text);
Ejecútalo y prueba que funciona correctamente.

- Vamos a ver la depuración. Para ello modificamos ligeramente el código anterior para hacerlo un poco más largo con el objetivo de establecer un punto de ruptura y ejecutar paso a paso. Sustituye el código anterior por el siguiente:

```
string url= textBox1.Text;

if (url.Trim() == "")

{

    url = "www.duckduckgo.com"; }

webBrowser1.Navigate(url);

this.Text = "Estamos en: " + url;
```

- Realiza doble clic sobre el margen izquierdo de la ventana de edición de código. Aparecerá un círculo granate que indica que ahí se ha puesto un punto de ruptura.
- Ejecuta de nuevo. En cuanto establezcas la URL y pulses el botón aparecerá el código. Una flecha amarilla en el margen indica la línea en la que estás. En la parte inferior se ven las variables locales.

Si quieres ver tus variables u objetos puedes verlos en la ventana de inspección.

Para ejecutar paso a paso se usa F11 y paso a paso sin entrar en los procedimientos con F10. Puedes ver otras teclas y posibilidades en el menú Depuración.

- Finalmente para obtener la ayuda sobre algún componente (como los de Java son muy amplios en cuanto a métodos, propiedades, eventos, etc...) Lo mejor es situarse sobre él con un simple clic y pulsar F1 lo que te lleva al MSDN que es la biblioteca de ayuda de Visual Studio .Net.

Otros entornos/Frameworks

Debido a la popularización de .Net y de las grandes posibilidades que tiene han surgido otros entornos de desarrollo que conviene tener en cuenta:

- MonoDevelop: Basado en el Proyecto Mono. Es software libre. Es multiplataforma (Linux, Mac y Windows) y no requiere del Framework de Microsoft ya que tiene el suyo propio. Por supuesto es compatible con el de MS. De hecho Xamarin está basado en Mono.

<http://monodevelop.com/>

- SharpDevelop: Es exclusivamente un entorno de desarrollo, de hecho es necesario tener instalado el Framework .Net para poder programar. Es gratuito, libre y sólo para plataformas Windows.

<http://www.icsharpcode.net/OpenSource/SD/>

- Xamarin: No es exactamente un entorno si no que es una implementación del Framework .Net multiplataforma para móviles Android e iOS. Tanto Visual Studio como los otros IDEs mencionados permiten desarrollo tanto con el Framework de MS como con Xamarin.

<https://www.xamarin.com>

Code Snippets (Fragmentos de código).

Al igual que en otros IDEs había la posibilidad de escribir código de forma rápida en VS existen los Code Snippets por si alguien quiere escribir de forma rápida ciertos fragmentos usando XML.

Por ejemplo *cw* y doble *TAB* escribe *Console.WriteLine();*

Los existentes puedes verlos en Tools→Code Snippets Manager. Si quieres diseñar los tuyos propios puedes leer más en:

<https://docs.microsoft.com/es-es/visualstudio/ide/walkthrough-creating-a-code-snippet?view=vs-2017>

<https://docs.microsoft.com/es-es/visualstudio/ide/code-snippets-schema-reference?view=vs-2017>