

O ENTORNO DE PROGRAMACIÓN

Ferramentas Básicas

Para programar unicamente é necesario:

- Un editor de textos para poder introducir o *código fonte*⁵ do programa. Debe ser un editor de textos simple, que non inserte caracteres de marca no documento (tipos de letra, negriñas...etc). En Linux serve un editor simple como *vim* ou *nano* (consola) ou ***gedit*** (entorno gráfico). En Windows, é moi útil como editor simple o **Notepad++**, ou **Sublime Text**. Tanto vim, como gedit ou Notepad++, ou Sublime Text soportan resaltado de sintaxe.
- Un compilador ou un intérprete da linguaxe utilizada

Aínda que potencialmente calquera linguaxe pode ser compilada (con un compilador) ou interpretada (cun intérprete), normalmente as linguaxes están deseñadas para unha das dúas aproximacións. Así, de modo xeral:

- As linguaxes, C, C++, Pascal son compiladas.
- As linguaxes de Shell (bash, PowerShell, dash, zsh...) e mais PHP, Python, Perl son interpretadas.
- As linguaxes Java, .NET (C#) son linguaxes p-code (pseudocompiladas).

Contornos Integrados de Desenvolvemento (IDE)

Aínda que cun editor de textos simple é suficiente, existen editores que incorporan numerosas utilidades para a programación como son:

- Resaltado de Sintaxe: Os editores dos IDE incorporan códigos de cores para resaltar as palabras e estruturas propias da linguaxe, o que fai máis cómoda a edición
- Axuda en Liña: Moitos IDE nos proporcionan axuda da linguaxe dende o propio entorno, o que o fai especialmente útil a hora de acceder á *librería estándar*⁶ da linguaxe.
- Refactorizador: É moi común que durante a escritura dunha aplicación sexa necesario cambiar de nome algún elemento, sendo necesario cambialo en todos os sitios onde aparece. O refactorizador do IDE encárgase, entre outras, de facilitar este tipo de tarefas evitando erros e permitindo cambiar o que realmente se necesita.
- Optimizacións de código: Moitos IDE son capaces de suxerir modos alternativos de realizar certas operacións.
- Depurador: O depurador nos permite executar a aplicación instrución por instrución, examinando si o desexamos o valor das variables. Isto é moi útil para a corrección de erros lóxicos da aplicación.

5 Se coñece como *código fonte* o código tal e como o escribe o programador nunha linguaxe de programación interpretada. Para que este código poda ser executado será necesario ou ben compilalo a *código máquina* utilizando un compilador da linguaxe, ou ben utilizar un intérprete da linguaxe.

6 A *librería estándar* dunha linguaxe está composta polo conxunto de funcións ou obxectos que forman parte da propia distribución do compilador da linguaxe, e polo tanto, da propia linguaxe. ***Unha función é un algoritmo codificado para realizar unha función simple concreta, como pode ser calcular unha raíz cadrada, unha entrada de datos por teclado, unha visualización en pantalla ou ler dun ficheiro do disco.***

- Xestión de proxectos: Unha aplicación pode estar composta de numerosos ficheiros que deben ser enlazados xuntos. Este tipo de contornos se encarga de xestionalo, e incluso integrar asistentes para o deseño das interfaces gráficas.
- Control de Versións: A maioría dos IDE inclúen a posibilidade de uso de sistemas de control de versións (Git/Subversion) para manter un control dos cambios e compartir o desenvolvemento da aplicación entre varios programadores.

Inicialmente os IDE estaban desenvolto para linguaxes específicas, pero a tendencia de hoxe en día é a de incorporar o compilador / intérprete / depurador en forma de “plugin”, de modo que un mesmo IDE pode utilizarse en varias linguaxes. Os IDE máis estendidos hoxe en día son:

- **Netbeans**: De Oracle, o seu obxectivo principal é a programación en linguaxe JAVA aínda que é multilinguaxe e multiplataforma (Unix / Linux / Windows / Mac)
- **Eclipse**: Da fundación Apache, é un IDE multilinguaxe e se emprega moito para programar en linguaxe JAVA. Multiplataforma (Unix / Linux / Windows / Mac)
- **VisualStudio**: De Microsoft, para moitos o mellor IDE do mercado. É multilinguaxe, pero funciona só baixo Windows.
- **Code::Blocks**: Este IDE está enfocado principalmente a linguaxe C/C++, aínda que soporta outros como D ou FORTRAN. Está creado con WxWidgets, e é Multiplataforma (Unix / Linux / Windows / Mac)
- **DevCpp**: Este IDE está enfocado á linguaxe C/C++ baixo Windows.

Entorno Aconsellado (GNU/Linux)

Probablemente Linux sexa hoxe en día a mellor plataforma para o desenvolvemento de aplicacións, xa que proporciona multitude de linguaxes, editores e librerías de modo “estándar” o que fai a instalación das plataformas de desenvolvemento algo moi simple e rápido.

Por estas razóns, se recomenda para este curso o uso de GNU/Linux (aínda que non é obrigatorio). Unha posibilidade para os usuarios de MS/Windows é a instalación de un sistema GNU/Linux baixo VirtualBox. O sistema aconsellado é Debian, aínda que calqueira distribución é válida. Como editor para o desenvolvemento, Debian dispón de modo estándar de **gedit** (gnome) ou **pluma** (MATE), aínda que existen moitos outros editores válidos.

Como IDE para aplicacións máis complexas, utilizaremos **NetBeans**, xa que é multiplataforma e nos permite realizar os traballos en Windows e Linux de xeito idéntico.

Entorno Alternativo (Windows)

Si se desexa desenvolver o curso baixo MS/Windows se recomenda a instalación de **Notepad++** ou **Sublime Text** como editor de textos para programas simples, e de **NetBeans** para proxectos maiores.

O maior problema do traballo con Windows é a necesidade de descargar e instalar os compiladores e intérpretes necesarios para probar os exemplos con distintas linguaxes, como **Python**, **C** ou **PHP**.

Para probar os exemplos en **C**, se recomenda **Code::Blocks**, que tamén está dispoñible en Linux.