

Instalación y uso de entornos

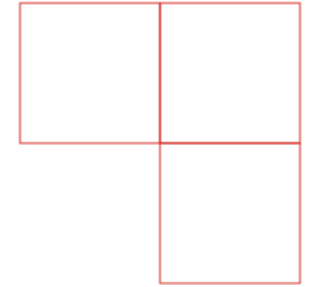
UF 2.1

ENTORNOS DE
DESARROLLO

Ve más allá

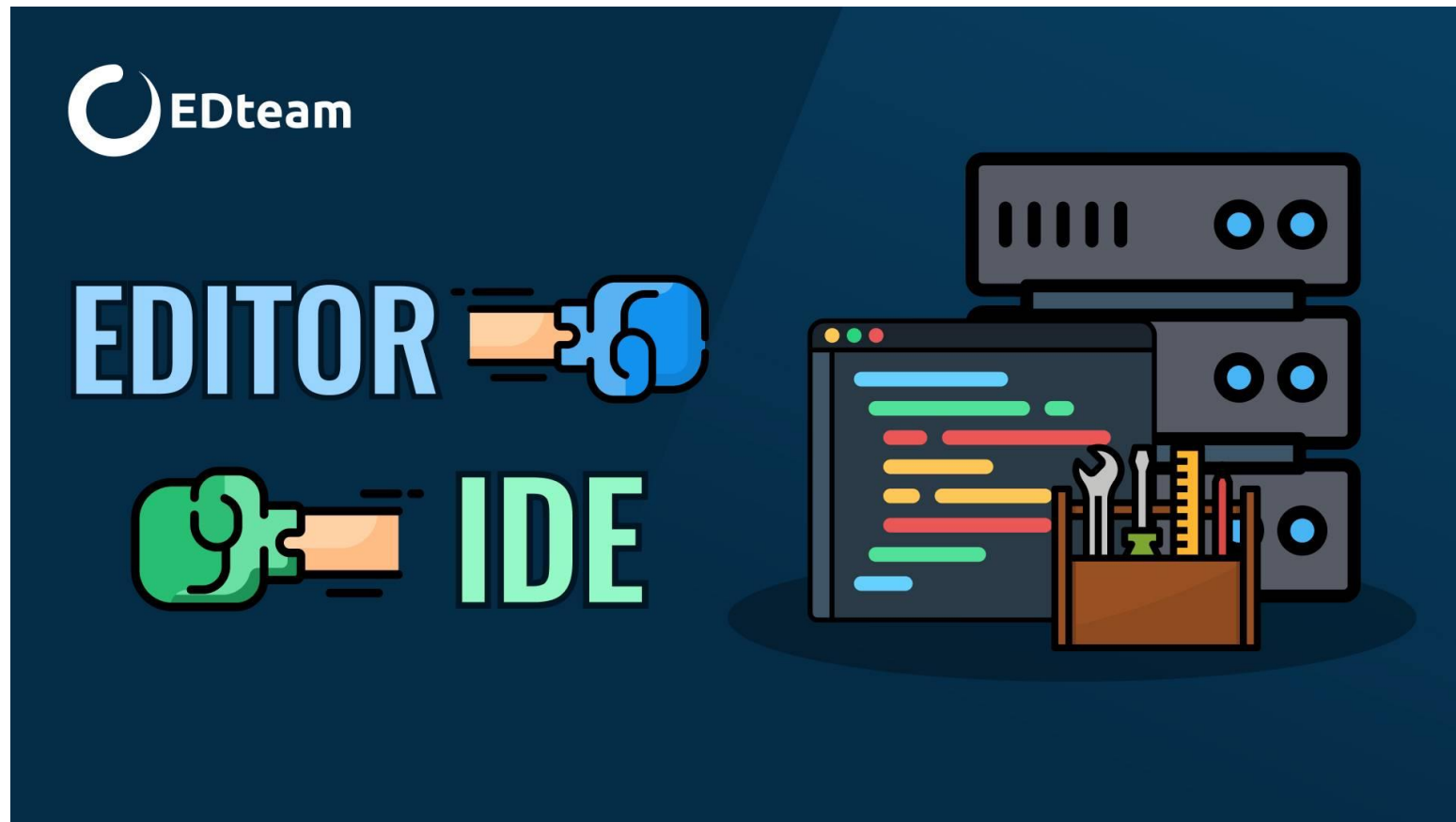
CONTENIDO

- Funciones de un entorno de desarrollo
- Ventajas de utilizar un IDE
- Tipos de entornos de desarrollo
- Características de un entorno de desarrollo



Editor de texto vs. IDE

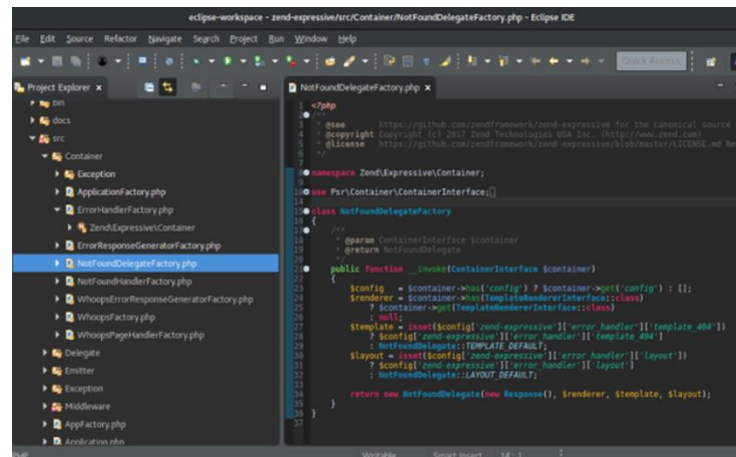
¿Qué es un IDE? ¿Y un editor de texto?



Funciones de un entorno de desarrollo

¿Qué es un IDE?

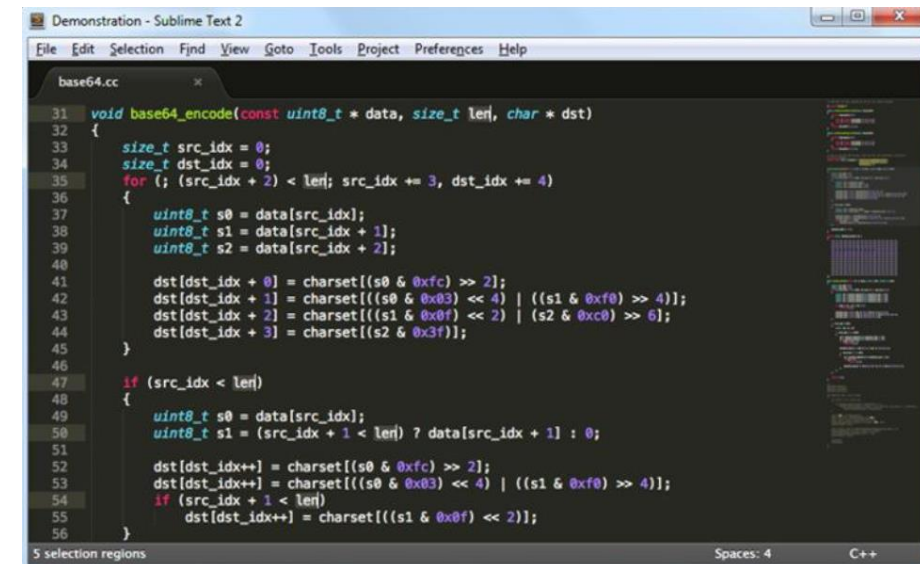
- Un IDE es una aplicación que nos ayuda en la tarea de creación de proyectos de software.
- Está pensada para facilitar la escritura del mismo por medio de asistentes y de realizar comprobaciones para que todo esté correcto. Es decir, es un **editor de código, compilador, depurador y constructor de interfaz gráfica (GUI)**.
- Nos proporcionan un entorno de trabajo amigable para la mayoría de los lenguajes existentes, además de que nos permite escribir el código de una forma sencilla resaltando la sintaxis.
- Suelen poseer un compilador o intérprete, un depurador, entre otras funcionalidades.



Funciones de un entorno de desarrollo

Editor de texto

- El editor de texto es la parte que nos permite escribir el código fuente del programa.
- Es capaz de reconocer, resaltar y cambiarlos colores de las variables, las cadenas de caracteres, las palabras reservadas, las instrucciones, el inicio y fin de los corchetes...
- De esta manera, el código fuente será mucho más visual y cómodo; además se podrán reconocer los errores a simple vista.
- No solo nos corregirá en caso de detectar un error, también mostrará sugerencias e indicaciones del posible fallo cometido.



```
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
```

Funciones de un entorno de desarrollo

Compilador

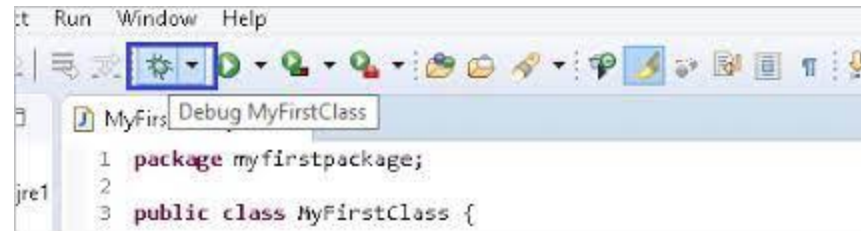
- Se encarga de “traducir” el código fuente de nuestro programa en el lenguaje máquina.
- El IDE se encarga precisamente de esta acción con el fin de que podamos ir realizando pruebas a medida que vamos creando nuestros desarrollos.
- Normalmente, tras la compilación se genera un **archivo ejecutable** que puede usarse para correr el programa las veces que sea necesario.



Funciones de un entorno de desarrollo

Intérpretes y depurador

- Los **intérpretes** se diferencian de los compiladores en que solo realizan la traducción a medida que se va ejecutando el código. El código vuelve a traducirse cada vez que el programa se ejecuta, sin guardar el resultado de esa interpretación.
- La traducción puede ser más lenta, porque se realiza por cada línea del programa, pero necesita menos memoria.
- El **depurador (Debugger)** es una herramienta que nos permite comprobar el funcionamiento de nuestro código, examinando instrucción a instrucción la ejecución de un programa y los cambios que se produzcan en las variables de este o en los registros del procesador.
- Permite detener el programa en un punto de ruptura para examinar los caminos o el valor de los datos que tienen en cada momento.



Ventajas de utilizar un IDE

- La **curva de aprendizaje es muy baja**; al disponer de herramientas visuales, facilita la comprensión en la realización de tareas.
- Formatea el código para ayudarnos a realizar un código ordenado, realizando las tabulaciones, incorporando llaves...
- **Funciones de refactorización** para renombrar variables, funciones...
- Nos muestra **warnings y errores** de sintaxis en pantalla de algo que no va a funcionar al interpretar o compilar.
- En lenguajes interpretados, como PHP o Javascript, el IDE va ejecutando el código línea a línea como si se tratara del entorno de producción (se conoce como máquina virtual).
- **Crea proyectos** y ayuda a mantener todos los archivos del programa localizados.

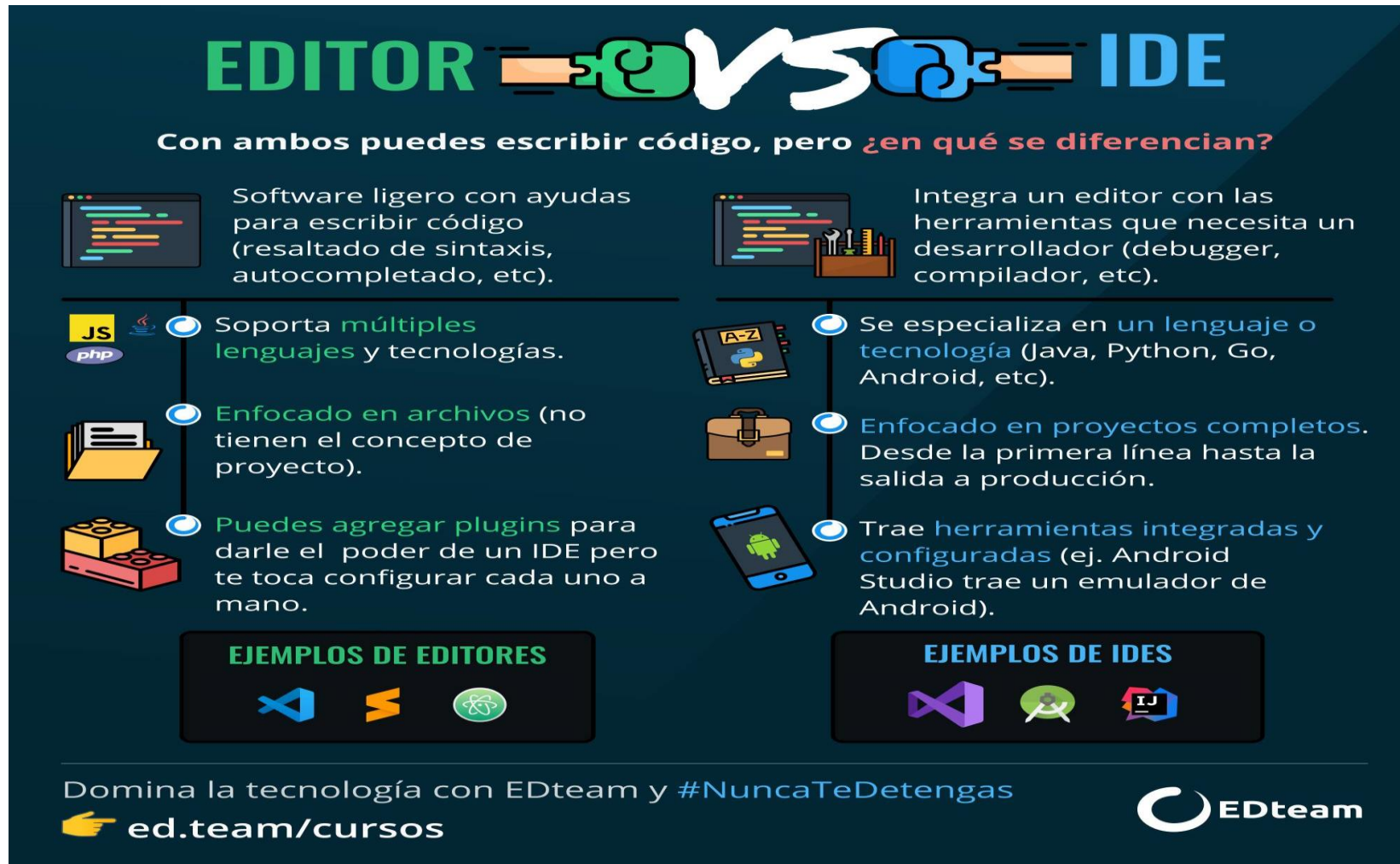


Tipos de entornos de desarrollo

- Casi todos los lenguajes (C++, PHP, Python, Java, C#, Delphi, VisualBasic...) tienen un entorno de desarrollo específico, VisualStudio de Microsoft.
- Aunque también podremos usar otros más genéricos que admiten lenguajes diferentes, como Eclipse o NetBeans, entre otros.



Tipos de entornos de desarrollo



Características de un entorno de desarrollo

Dependiendo del lenguaje de programación, los requerimientos que solicitamos a un entorno de desarrollo pueden ser diferentes, pero se considera que un buen entorno de desarrollo debe tener las siguientes **características**:

- Poder ser instalado en múltiples plataformas.
- Soporte para varios lenguajes de programación.
- Control de versiones.
- Asistentes de sintaxis.
- Extensiones y plugins adicionales.
- Integración con framework.
- Depurador.
- Importar y exportar proyectos.
- Manual de usuario y ayuda.



Instalación de entornos

Pasos para la instalación

- Descarga del IDE desde el sitio web oficial o el repositorio adecuado.
- Selección de la versión compatible con el sistema operativo en uso (Windows, macOS, Linux, etc.).
- Ejecución del instalador y seguimiento de las instrucciones de instalación.



Instalación de entornos

Consideraciones clave

- **Requisitos del Sistema:** Comprobar que el sistema cumple con los requisitos mínimos de hardware y software para el IDE en cuestión (CPU, RAM, espacio en disco, sistema operativo compatible, etc.).
- **Compatibilidad con Versiones Anteriores:** Asegurarse de que la versión del IDE que se va a instalar sea compatible con la versión de cualquier proyecto existente o las herramientas y librerías que se estén utilizando.
- **Configuraciones Adicionales:** Algunos IDEs pueden requerir configuraciones adicionales después de la instalación, como la configuración de rutas de acceso, variables de entorno u opciones de personalización.
- **Gestión de Dependencias y Extensiones:** Durante la instalación, es posible que se ofrezcan opciones para instalar extensiones, plugins o complementos adicionales. Se debe evaluar la necesidad de estas herramientas y seleccionarlas según el flujo de trabajo.
- **Actualizaciones y Mantenimiento:** Familiarizarse con el proceso de actualización del IDE y establecer prácticas para mantenerlo actualizado con las últimas versiones y parches de seguridad.



Instalación de entornos

Consideraciones clave

- **Respaldo de Configuraciones:** Realizar copias de seguridad de las configuraciones personalizadas y preferencias del IDE, si es posible, para facilitar la restauración en caso de fallos o migraciones de sistema.
- **Integración con Herramientas Externas:** Verificar si es necesario configurar la integración del IDE con herramientas externas, como compiladores, emuladores, control de versiones, para asegurar un flujo de trabajo fluido.
- **Compatibilidad con Proyectos Actuales:** Antes de la instalación, confirmar que la versión del IDE es compatible con los proyectos actuales, especialmente si se está migrando a una versión más reciente.
- **Seguridad:** Asegurarse de descargar el IDE desde fuentes confiables y oficiales para evitar posibles riesgos de seguridad. Además, se deben seguir las mejores prácticas de seguridad informática.





Universidad
Europea

Instalación y uso de entornos

CFGs

- Desarrollo de Aplicaciones Web
- Desarrollo de Aplicaciones Multiplataforma

Ve más allá