

UD4. Evaluación de Entornos Integrados de Desarrollo

Raúl Reyes Mangano
Patricia Vegas
Adrián Seoane

1º CFGS Desarrollo de Aplicaciones Multiplataforma
Entornos de desarrollo



Contextualización

R.A.2: Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.

- a) Se han instalado entornos de desarrollo, propietarios y libres.*
- d) Se ha configurado el sistema de actualización del entorno de desarrollo.*
- g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.*
- b) Se han añadido y eliminado módulos en el entorno de desarrollo.
- c) Se ha personalizado y automatizado el entorno de desarrollo.
- e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
- f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.
- h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.

Índice

1. Funciones de un entorno de desarrollo
2. Componentes
3. Tipos
4. Instalación
5. Mecanismos de actualización
6. Uso
7. Herramientas CASE

Funciones de un entorno de desarrollo

ENTORNO DE DESARROLLO o IDE

(Integrated Development Environment):

Programa informático que tiene el objetivo de asistir al programador en el diseño y codificación de un software mediante la inclusión de múltiples herramientas destinadas para dichas tareas.

Funciones:

- ❑ Las tareas esenciales de la fase de codificación:
 - ❑ Edición: creación y modificación del código fuente.
 - ❑ Proceso/Ejecución del programa
 - ❑ Interpretación directa del código fuente
 - ❑ Compilación (a código máquina) - enlazado y ejecución
 - ❑ Compilación (a código intermedio) - interpretación

Funciones de un entorno de desarrollo

ENTORNO DE DESARROLLO o IDE

(Integrated Development Environment):

Programa informático que tiene el objetivo de asistir al programador en el diseño y codificación de un software mediante la inclusión de múltiples herramientas destinadas para dichas tareas.

Otras funciones:

- ☐ Examinar (hojear) el código fuente
- ☐ Analizar consistencia, calidad, etc.
- ☐ Ejecutar en modo depuración
- ☐ Ejecución automática de pruebas
- ☐ Control de versiones
- ☐ Generar documentación
- ☐ Reformar (refactorizar) código

Componentes de un entorno de desarrollo

- ❑ Un editor de texto
- ❑ Un compilador
- ❑ Un intérprete
- ❑ Un depurador
- ❑ Posibilidad de ofrecer un sistema de control de versiones
- ❑ Factibilidad para ayuda en la construcción de interfaces gráficas de usuario

Tipos de entornos de desarrollo

❑ **Entornos centrados en un lenguaje**

- ❑ Son específicos para un lenguaje de programación en particular.
- ❑ El editor tiene una fuerte orientación al lenguaje.
- ❑ Son fáciles de usar.
- ❑ En ciertas ocasiones son poco flexibles a la interoperación con otros productos o la ampliación de sus funciones.
- ❑ Ejemplos: DEV-C++, VISUAL C++, DELPHI

Tipos de entornos de desarrollo

❑ **Orientados a la estructura**

- ❑ Podrían considerarse incluidos en el tipo anterior, ya que suelen ser específicos para un lenguaje de programación; pero difieren en que el editor de código fuente no es un editor de texto, sino un editor de estructuras o un editor sintáctico. Se basan en representar internamente el código fuente como una estructura.
- ❑ Estos lenguajes estuvieron de moda en los años 80; en la actualidad los lenguajes de marcado (XML) pueden ser una buena forma de representar la estructura del código fuente.
- ❑ Ejemplos: Mentor, Gandalf, Alice PASCAL

Tipos de entornos de desarrollo

❑ Entornos multilenguaje

- ❑ Hay aplicaciones que combinan piezas de código fuente escritas con diferentes lenguajes de programación. Algunas posibilidades de combinación son:
 - ❑ Entornos genéricos: no se combinan lenguajes en un mismo programa, hay varios programas, cada uno en su propio lenguaje. Ejemplo: Eclipse, Visual Studio Code
 - ❑ Entornos específicos: se usan para una combinación correcta de lenguajes. Vienen a ser como los entornos orientados al lenguaje, pero admiten más de un lenguaje de programación. Ejemplos: GPS, que combina ADA y C++.

Instalación de un entorno de desarrollo

Factores a la hora de elegir nuestro entorno de desarrollo:

❑ Sistema Operativo:

Saber en qué sistema operativo vamos a trabajar, y más aún, para qué sistema operativo vamos a desarrollar nuestro software.

Virtualizar o emular → Último recurso

Si desarrollamos aplicaciones para Linux es inusual desarrollar la aplicación en Windows → **Compilador del IDE integrado**

Instalación de un entorno de desarrollo

Factores a la hora de elegir nuestro entorno de desarrollo:

❑ **Lenguaje de programación:**

Un IDE puede soportar uno o varios lenguajes de programación, por lo que saber en qué lenguaje de programación vamos a codificar nuestro software y qué lenguajes nos ofrecen los distintos IDE es una información valiosa que hay que tener en cuenta.

Este criterio va de la mano con el sistema operativo, ya que si quisiéramos desarrollar en Visual Basic bajo un sistema operativo Linux no sería Visual Studio nuestra opción, sino que tendríamos que utilizar Gambas.

Instalación de un entorno de desarrollo

Factores a la hora de elegir nuestro entorno de desarrollo:

❑ **Framework (Plataforma de Trabajo):**

Soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

Si vamos a desarrollar con Visual Basic con la plataforma .NET bajo Linux, tendríamos que usar Mono Develop en lugar de Visual Studio.

“.Net” ofrece el “Visual Studio .net” que le permite a los desarrolladores construir aplicaciones, y su motor es el “.Net framework” que permite ejecutar dichas aplicaciones. Este motor es un componente que se instala sobre el sistema operativo y que ahora viene incluido en la mayoría de los sistemas operativos de Microsoft.

Instalación de un entorno de desarrollo

Factores a la hora de elegir nuestro entorno de desarrollo:

❑ **Framework (Plataforma de Trabajo):**

Framework puede ser algo tan grande como “.NET” o Java (también es un framework), pero también el concepto se aplica a ámbitos más específicos, por ejemplo dentro de Java en el ámbito específico de aplicaciones Web tenemos los frameworks: Struts, “Java Server Faces”, o Spring.

Estos frameworks de Java en la práctica son conjuntos de librerías (API's) para desarrollar aplicaciones Web, más librerías para su ejecución (o motor), y más un conjunto de herramientas para facilitar esta tarea (debuggers, ambientes de desarrollo como Eclipse, etc).

Instalación de un entorno de desarrollo

Factores a la hora de elegir nuestro entorno de desarrollo:

❑ **Herramientas y Disponibilidad:**

Las diferentes herramientas de las que disponen los IDE son el último criterio de selección, seguramente nos encontremos con varios IDE que cumplen los requisitos de lenguaje y sistema operativo, pero no todos tienen las mismas funciones, por lo que saber cuáles son esas herramientas es un dato sumamente importante en nuestra decisión.

- Trabajo Colaborativo, se necesita un control de versiones:
 - ❑ Team Server Foundation (TFS), en Visual Studio.
 - ❑ Subversion (SVN), se puede usar indistintamente Netbeans, Eclipse o IntelliJ IDEA entre otros.
- Archivos de ayuda y documentación: deberíamos de buscar un IDE que tuviese dicha funcionalidad (o un plugin para ese IDE).
- Refactorizaciones: número de refactorizaciones automáticas que te proporcionan los distintos IDE.

Tarea 4.1. Funciones y componentes

Con esta tarea evaluaremos los siguientes criterios de evaluación:

- b) Se han añadido y eliminado módulos en el entorno de desarrollo.
- c) Se ha personalizado y automatizado el entorno de desarrollo.

Nota: 5% de la calificación final del módulo.

Instala, si aún no lo has hecho, Visual Studio Code y explica en qué consiste y cómo podemos acceder a las siguientes características:

- ☐ **Extensiones (plugins). Qué son y cómo se utilizan.**
 - ☐ **Instala alguna extensión para HTML, CSS o diagramas UML**
- ☐ **Abre un terminal en la carpeta del proyecto. Explica como hacerlo con una extensión.**
- ☐ **Investiga cómo generar Snippets**

Tarea 4.2. Generar un archivo ejecutable

Con esta tarea evaluaremos los siguientes criterios de evaluación:

- e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
- f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.

Nota: 5% de la calificación final del módulo.

- ❑ **Realiza una presentación en la que expliques cómo podemos crear un ejecutable a partir del código fuente de una aplicación desarrollada en Java.**