

Departamento **INFORMÁTICA**  
Área, materia o módulo **Entornos de Desarrollo**  
Curso y nivel **1º DAM**

## CRITERIOS DE EVALUACIÓN

1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.
  - a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros.
  - b) Se han clasificado los lenguajes de programación.
  - c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.
  - d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
  - e) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.
  - f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.
  - g) Se han identificado las fases de desarrollo de una aplicación informática.
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.
  - b) Se han añadido y eliminado módulos en el entorno de desarrollo.
  - c) Se ha personalizado y automatizado el entorno de desarrollo.
  - d) Se ha configurado el sistema de actualización del 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.
  - g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.
  - h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.
3. Verifica el funcionamiento de programas diseñando y realizando pruebas.
  - a) Se han identificado los diferentes tipos de pruebas.
  - b) Se han definido casos de prueba.
  - c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
  - d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
  - e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
  - f) Se ha documentado el plan de pruebas.
  - g) Se han efectuado pruebas unitarias de clases y funciones.
  - h) Se han efectuado pruebas de integración, de sistema y de aceptación.
  - i) Se han implementado pruebas automáticas.
  - j) Se han documentado las incidencias detectadas.
  - k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software.
  - l) Se han realizado medidas de calidad sobre el software desarrollado.
4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.
  - a) Se han identificado los patrones de refactorización más usuales.
  - b) Se han elaborado las pruebas asociadas a la refactorización.
  - c) Se ha revisado el código fuente usando un analizador de código.
  - d) Se han identificado las posibilidades de configuración de un analizador de código.

- e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
  - f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.
  - g) Se ha documentado el código fuente mediante comentarios.
  - h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos.
  - i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.
5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.
- a) Se han identificado los conceptos básicos de la programación orientada a objetos.
  - b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.
  - c) Se han identificado las herramientas para la elaboración de diagramas de clases.
  - d) Se ha interpretado el significado de diagramas de clases.
  - e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
  - f) Se ha generado código a partir de un diagrama de clases.
  - g) Se ha generado un diagrama de clases mediante ingeniería inversa.
6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.
- a) Se han identificado los distintos tipos de diagramas de comportamiento.
  - b) Se ha reconocido el significado de los diagramas de casos de uso.
  - c) Se han interpretado diagramas de interacción.
  - d) Se han elaborado diagramas de interacción sencillos.
  - e) Se han interpretado diagramas de estados.
  - f) Se han planteado diagramas de estados sencillos.
  - g) Se ha interpretado el significado de diagramas de actividades.
  - h) Se han elaborado diagramas de actividades sencillos.

## **INSTRUMENTOS DE EVALUACIÓN**

- Observación diaria
- Ejercicios prácticos
- Pruebas teóricas
- Pruebas prácticas

EL/LA JEFE/A DEL DEPARTAMENTO