



Criterios de Evaluación

Entornos de Desarrollo

1º DAW

I.E.S. Bernaldo de Quirós
Departamento de Informática
2024-2025



Resultados de aprendizaje	Criterios de evaluación	Ponderación
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.	<ul style="list-style-type: none">a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador, periféricos, entre otros.b) Se han identificado las fases de desarrollo de una aplicación informática.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 han clasificado los lenguajes de programación, identificando sus características.f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en el desarrollo de software.g) Se han identificado las características y escenarios de uso de las metodologías ágiles de desarrollo de software.	15%
2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.	<ul style="list-style-type: none">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.	15%



3. Verifica el funcionamiento de programas diseñando y realizando pruebas.	<ul style="list-style-type: none">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 han efectuado pruebas unitarias de clases y funciones.g) Se han implementado pruebas automáticas.h) Se han documentado las incidencias detectadas.i) Se han utilizado dobles de prueba para aislar los componentes durante las pruebas.	20%
4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.	<ul style="list-style-type: none">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 han utilizado herramientas del entorno de desarrollo para documentar las clases.h) Se han utilizado repositorios remotos para el desarrollo de código colaborativo.i) Se han utilizado herramientas para la integración continua del código	10%
5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando herramientas	<ul style="list-style-type: none">a) Se han identificado los conceptos básicos de la programación orientada a objetos.b) Se han utilizado herramientas para la elaboración de diagramas de clases.c) Se ha interpretado el significado de diagramas de clases.	20%



específicas.	d) Se han trazado diagramas de clases a partir de las especificaciones de las mismas. e) Se ha generado código a partir de un diagrama de clases. f) 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 herramientas específicas.	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 ha interpretado el significado de diagramas de actividades. f) Se han elaborado diagramas de actividades sencillos. g) Se han interpretado diagramas de estados. h) Se han planteado diagramas de estados sencillos.	20%