ED04.- Optimización y documentación.

Orientaciones Alumnado

Esta es la cuarta unidad del módulo. En ella aprenderás técnicas que ayudan a mejorar la calidad del software que se está desarrollando. Conocerás diferentes técnicas de refactorización y análisis estático de código, que formando parte del entorno de desarrollo que utilizas, mejoran el código.

Dado el trabajo en equipo que supone el trabajo de programación, conocerás y utilizarás herramientas de control de versiones de tu entorno de desarrollo, para gestionar los proyectos y ficheros que vas realizando y modificando.

Por último, deberás valorar la importancia de la documentación en el software, insertando comentarios siguiendo las pautas que marca el lenguaje de programación en el que se desarrollan.

Datos generales de la Unidad de Trabajo

Nombre completo del MP	Entornos de desarrollo.	**Siglas MP**	ED
N° y título de la UT	04.- Optimización y documentación.		
índice o tabla de contenidos	1.- Refactorización. 1.1.- Introducción. 1.2.- Patrones de refactorización más habituales. 1.3.- Analizadores de código. 2.- Control de versiones. 2.1.- Introducción. 2.2.- Tipos de herramientas de control de versiones. 2.3.- Estructura de las herramientas de control de versiones. 2.4.- Herramientas de control de versiones. 2.5.- Planificación de la gestión de configuraciones. 2.6.- Gestión del cambio. 2.7.- GIT. 2.7.1.- Introducción. 2.7.2.- Terminología. 2.7.3.- Escenario de trabajo con GIT. 2.7.4.- Flujo de trabajo. 2.7.5.- Resumen de comandos GIT. 2.7.6.- Gitk. 3.- Documentación. 3.1.- Introducción. 3.2.- Uso de comentarios. 3.3.- Javadoc.		
Objetivos	✓ Conocer los patrones más habituales de refactorización de código		

	que se utilizan, y su utilización mediante el entorno de desarrollo donde se esté programando. ✓ Conocer el concepto y fin de un analizador de código, instalando y usando alguno en el entorno de desarrollo. ✓ Realizar la documentación en código, del proceso de implementación y refactorización de código.	
Temporalización (estimación)	**Tiempo necesario para estudiar los contenidos (h)**	6
	Tiempo necesario para completar la tarea (h)	8
	Tiempo necesario para completar el examen (h)	1
	N° de días que se recomienda dedicar a esta unidad	9
	La temporalización anterior no deja de ser una estimación media, ya que el tiempo a invertir va a depender mucho de las circunstancias personales de cada cual.	
Consejos y recomendaciones	✓ La primera parte de la unidad es muy teórica, por lo que hay que prestar atención a los conceptos y procedimientos que se presentan. ✓ Para poder entender el concepto de refactorización, debes tener ya, alguna soltura en la implementación de código, y en el uso del entorno de desarrollo. ✓ Debes tener soltura en la búsqueda e instalación de complementos de software libre en su entorno de desarrollo. ✓ Debes de haber realizado alguna aplicación que suponga una continua revisión, creando diferentes versiones, para que utilice herramientas como GIT o Subversion, para la automatización de las diferentes versiones generadas.	