

DATOS IDENTIFICATIVOS DE LA UNIDAD FORMATIVA

UNIDAD FORMATIVA	EL CICLO DE VIDA DEL DESARROLLO DE APLICACIONES.	DURACIÓN	80
		Condicionada	
Código	UF2406		
Familia profesional	INFORMÁTICA Y COMUNICACIONES		
Área Profesional	Desarrollo.		
Certificado de profesionalidad	PROGRAMACIÓN CON LENGUAJES ORIENTADOS A OBJETOS Y BASES DE DATOS RELACIONALES	Nivel	3
Módulo formativo	Programación orientada a objetos.	Duración	250
Resto de unidades formativas que completan el módulo	Principios de la programación orientada a objetos	Duración	90
	Modelo de programación web y bases de datos		80

Apartado A: REFERENTE DE COMPETENCIA

Esta unidad formativa se corresponde con la RP1 en lo referido a las especificaciones de diseño y niveles de calidad, RP3 y RP5.

Apartado B: ESPECIFICACIÓN DE LAS CAPACIDADES Y CONTENIDOS

Capacidades y criterios de evaluación:

C1: Manejar las herramientas de ingeniería de software.

CE1.1 Enumerar y comparar los modelos de ingeniería software, indicando los conceptos principales en los que se basan, su ámbito de uso y cómo se estructuran.

CE1.2 Describir las fases, en cada modelo, del proceso de ingeniería de software, indicando para cada una:

- Datos de partida (entradas).
- Datos finales (salidas).
- Funciones realizadas en la fase.
- Documentación generada.
- Trazabilidad.

CE1.3 Describir en detalle los conceptos fundamentales de una metodología de ingeniería software basada en la orientación a objetos.

CE1.4 Identificar en las herramientas de desarrollo orientada a objetos disponibles, los diferentes componentes y describir cómo se implementan los conceptos de la metodología y las distintas fases del proceso de ingeniería de software, haciendo especial hincapié en los procesos de:

- Diseño.
- Codificación.
- Pruebas unitarias.
- Documentación.
- Evaluación de la calidad y métricas.
- Gestión de la Configuración.
- Cambios.

CE1.5 En un supuesto práctico utilizar las herramientas de desarrollo, en el caso de que sea posible, para:

- Extraer la información relativa al diseño de una clase.
- Codificar la clase.
- Realizar su depuración.
- Incorporar la definición y la implementación de las pruebas unitarias.
- Trazar las pruebas con las clases.
- Configurar la herramienta para la realización de baterías de pruebas automáticas.
- Elaborar documentación mediante las plantillas incorporadas en la herramienta.
- Realizar modificaciones a las plantillas incorporadas para la elaboración de la documentación.
- Generar informes de calidad y métricas e interpretar los resultados.
- Incorporar cambios.

C2: Verificar la corrección de las clases desarrolladas mediante la realización de pruebas.

- CE2.1 Enumerar y describir las herramientas y utilidades más comunes para la depuración de programas.
 - CE2.2 Enumerar y describir los tipos de pruebas posibles que se pueden dar en el proceso de desarrollo de aplicaciones, distinguiendo especialmente aquellas que son responsabilidad del programador.
 - CE2.3 Enumerar y describir los tipos de pruebas que se deben realizar a una clase para verificar su corrección.
 - CE2.4 Utilizar las características proporcionadas por un entorno de desarrollo para realizar la depuración de un programa, mediante:
 - Ejecución paso a paso.
 - Establecimiento de puntos de parada (condicionales o incondicionales).
 - Monitorización de variables.
 - CE2.5 En supuestos prácticos, documentar una estrategia de pruebas completa a una clase a partir de su documentación de diseño, de forma que se asegure el óptimo funcionamiento en aspectos como:
 - Coherencia en el estado de los objetos.
 - Todos los escenarios posibles.
 - Rendimiento.
 - Casos límite.
 - Situaciones excepcionales.
 - CE2.6 En supuestos prácticos, a partir de una clase y la estrategia definida de pruebas:
 - Establecer el conjunto de secuencias y estados iniciales de los objetos que intervienen en la secuencias.
 - Establecer el criterio de evaluación de los resultados.
 - Automatizar el proceso mediante programas de prueba.
 - CE2.7 Realizar pruebas a una clase mediante herramientas de prueba.
- C3: Elaborar la documentación completa relativa a las clases desarrolladas y pruebas realizadas.
- CE3.1 Describir la información que debe acompañar a una clase desarrollada.
 - CE3.2 Describir los criterios fundamentales para la inclusión de cabeceras y comentarios en el código.
 - CE3.3 Proponer índices (plantillas) para los documentos de diseño y explicar el contenido de cada uno de los apartados.
 - CE3.4 Proponer índices (plantillas) para los documentos de pruebas (planificación y resultados) y explicar el contenido de cada uno de los apartados.
 - CE3.5 Proponer índices (plantillas) para la documentación de operación y mantenimiento (manuales técnicos) y explicar el contenido de cada uno de los apartados.
 - CE3.6 Aplicar unos criterios de normalización establecidos para incluir cabeceras y comentarios en el código.
 - CE3.7 En un supuesto práctico, a partir de una clase desarrollada, elaborar la documentación de diseño de la clase de acuerdo a un índice establecido.
 - CE3.8 En un supuesto práctico, a partir de una clase y conjunto de pruebas, elaborar la documentación de pruebas (planificación y resultados) de acuerdo a un índice establecido.
 - CE3.9 En un supuesto práctico, a partir de la documentación de desarrollo y pruebas, elaborar la documentación de operación y mantenimiento de acuerdo a un índice establecido.
 - CE3.10 Realizar la documentación de las clases mediante herramientas de documentación automática.
- C4: Realizar modificaciones de clases existentes por cambios en las especificaciones.
- CE4.1 Enumerar y describir los conceptos fundamentales de la gestión de la configuración del software desarrollado por una organización.
 - CE4.2 Enumerar y describir los conceptos fundamentales de la gestión de la configuración de la documentación.
 - CE4.3 Describir los pasos y precauciones fundamentales en el proceso de modificación de clases existentes.
 - CE4.4 En un supuesto práctico de modificación de un documento, y de acuerdo con un procedimiento:
 - Realizar la modificación en el documento.
 - Marcar las hojas cambiadas con la modificación y el código de revisión.
 - Elaborar la propuesta de cambio de acuerdo al formato especificado.
 - Editar la nueva edición o revisión del documento.
 - CE4.5 En un supuesto práctico de modificación de una clase por un cambio en su diseño y de acuerdo a un procedimiento.
 - Modificar el código para incorporar el cambio siguiendo los criterios de calidad establecidos.
 - Incluir en el lugar de la modificación y en la cabecera del componente software los datos del histórico y la explicación de la modificación.
 - Modificar, si es necesario, los programas de pruebas asociados siguiendo los criterios de calidad establecidos.
 - Comprobar, mediante la realización de pruebas, que la modificación ha sido incorporada con éxito y que no ha alterado la funcionalidad del resto de la clase.
 - Realizar la actualización de toda la documentación que se vea afectada por el cambio propuesto.
 - CE6.6 Utilizar herramientas para la gestión de la configuración y las versiones del software.
- C5: Desarrollar interfaces de usuario en lenguajes de programación orientados a objeto, a partir del diseño detallado.
- CE5.1 Interpretar diagramas de casos de uso y analizar las necesidades y peticiones de usuarios.
 - CE5.2 Identificar y describir las clases básicas que se usan para el interfaz hombre / máquina.

CE5.3 Explicar el concepto de evento y de programación orientada a eventos y su implementación en los lenguajes orientados a objetos en relación con las clases necesarias para el desarrollo de la interfaz.

CE5.4 Identificar los recursos multimedia que pueden incluirse en una ventana de interfaz de usuario.

CE5.5 Realizar el diseño de las ventanas correspondientes a la interfaz de usuario, aplicando criterios de ergonomía, eficacia y posible utilización por discapacitados, en la comunicación de información.

CE5.6 En un supuesto práctico de construcción de un interfaz de usuario:

- Identificar los elementos básicos que constituyen la interfaz, en la documentación de diseño proporcionada.
- Asociar a cada elemento identificado la clase adecuada de la librería de clases disponibles.
- Construir la interfaz con una herramienta de diseño de interfaz gráfica.
- Identificar y describir el código fuente generado por la herramienta de interfaz gráfica.

CE5.7 En un supuesto práctico de construcción de una aplicación con interfaz de usuario, a partir del código generado por una herramienta de interfaz gráfica:

- Incluir el código necesario para incorporar otros recursos multimedia a partir de las librerías de clases disponibles y siguiendo los criterios de calidad que se establezcan en un lenguaje orientado a objetos.
- Realizar la programación de las clases necesarias para conectar la interfaz de usuario con la aplicación, siguiendo los criterios de calidad que se establezcan y utilizando las características proporcionadas por un lenguaje de programación orientado a objetos.

CE5.8 En un supuesto práctico, sobre una interfaz de usuario desarrollada y con la documentación correspondiente al diseño detallado:

- Evaluar que los servicios de presentación implementados cumplen con las necesidades definidas por el usuario y utilizan de forma óptima los recursos del sistema.
- Establecer criterios de validación de los servicios de presentación.
- Describir diversos tipos de errores en función de la interfaz.
- Comprobar que los formatos de entrada y salida son los esperados conforme a las especificaciones de diseño.
- Verificar que la realización de operaciones indebidas no altera la fiabilidad del sistema.
- Elaborar la documentación de la interfaz.
- Redactar la guía de usuario correspondiente.

Contenidos:

1. Proceso de ingeniería del software

- Distinción de las fases del proceso de ingeniería software: especificación, diseño, construcción y pruebas unitarias, validación, implantación y mantenimiento.
- Análisis de los modelos del proceso de ingeniería: modelo en cascada, desarrollo evolutivo, desarrollos formales, etc.
- Identificación de requisitos: concepto, evolución y trazabilidad.
- Análisis de metodologías de desarrollo orientadas a objeto.
- Resolución de un caso práctico de metodologías de desarrollo que utilizan UML.
- Definición del concepto de herramientas CASE:
 - Herramientas de ingeniería software
 - Entornos de desarrollo
 - Herramientas de prueba
 - Herramientas de gestión de la configuración
 - Herramientas para métricas.

2. Planificación y seguimiento

- Realización de estimaciones
- Planificaciones: modelos de diagramado. Diagrama de Gantt
- Análisis del proceso del seguimiento: Reuniones e Informes

3. Diagramado

- Identificación de los principios básicos de UML.
- Empleo de diagramas de uso.

4. Desarrollo de la GUI

- Análisis del modelo de componentes y eventos.
- Identificación de elementos de la GUI.
- Presentación del diseño orientado al usuario. Nociones de usabilidad.
- Empleo de herramientas de interfaz gráfica

5. Calidad en el desarrollo del software

- Enumeración de criterios de calidad.
- Análisis de métricas y estándares de calidad.

6. Pruebas

- Identificación de tipos de pruebas.
- Análisis de pruebas de defectos: Pruebas de caja negra. Pruebas estructurales. Pruebas de trayectorias. Pruebas de integración. Pruebas de interfaces:
 - Preparación de los datos de prueba.
 - Casos de pruebas.
 - Codificar las pruebas.
 - Definir procesos de pruebas.
 - Ejecución de pruebas.
 - Generación de informes de las pruebas.

7. Excepciones

- Definición. Fuentes de excepciones. Tratamiento de excepciones. Prevención de fallos. Excepciones definidas y lanzadas por el programador.
- Uso de las excepciones tratadas como objetos.

8. Documentación

- Como producir un documento.
- Estructura del documento.
- Generación automática de documentación.

Apartado C: REQUISITOS Y CONDICIONES

Deberá cumplir alguno de los requisitos siguientes:

- Estar en posesión del título de Bachiller.
- Estar en posesión de algún certificado de profesionalidad de nivel 3.
- Estar en posesión de un certificado de profesionalidad de nivel 2 de la misma familia y área profesional.
- Cumplir el requisito académico de acceso a los ciclos formativos de grado superior o haber superado las correspondientes pruebas de acceso a ciclos de grado superior.
- Tener superada la prueba de acceso a la universidad para mayores de 25 años y/o de 45 años.
- Tener, de acuerdo con la normativa que se establezca, los conocimientos formativos o profesionales suficientes que permitan cursar con aprovechamiento la formación.

En relación con las exigencias de los formadores o de las formadoras, instalaciones y equipamientos se atenderá las exigencias solicitadas para el propio certificado de profesionalidad.