DATOS IDENTIFICATIVOS DE LA UNIDAD FORMATIVA

UNIDAD FORMATIVA	DISEÑO DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES	Duración Específica	90
Código	UF1289		
Familia profesional	INFORMÁTICA Y COMUNICACIONES		
Área Profesional	Sistemas y telemática		
Certificado de profesionalidad	Programación de sistemas informáticos	Nivel	3
Módulo formativo	Desarrollo de software basado en tecnologías orientadas a componentes	Duración	210
Resto de unidades formativas que completan el módulo	Implementación e integración de elementos software con tecnologías basadas en componentes	Duración	90
	Despliegue y puesta en funcionamiento de componentes software		30

Apartado A: **REFERENTE DE COMPETENCIA**

Esta unidad formativa se corresponde con la RP1 de la UC0965_3 DESARROLLAR ELEMENTOS SOFTWARE CON TECNOLOGÍAS DE PROGRAMACIÓN BASADA EN COMPONENTES.

Apartado B: **ESPECIFICACIÓN DE LAS CAPACIDADES Y CONTENIDOS**

Capacidades y criterios de evaluación

C1: Identificar las características y arquitecturas de las tecnologías de desarrollo, orientadas a componentes para la creación y modificación de elementos software integrados en estos entornos, según estándares y normalizaciones existentes.

CE1.1 Describir las técnicas y métodos de desarrollo involucrados en el paradigma del desarrollo, orientado a componentes para la confección y modificación elementos software, según los estándares de esta tecnología.

CE1.2 Clasificar las herramientas y lenguajes orientados a objetos utilizados en el desarrollo orientado a componentes, describiendo sus características para identificar las que son específicas para la creación o modificación de los elementos software, según las especificaciones funcionales dadas.

CE1.3 Clasificar los estándares de modelos de componentes, describiendo las pasarelas para interoperar entre componentes heterogéneos, para realizar las tareas de integración de los elementos desarrollados según especificaciones funcionales y técnicas.

CE1.4 Identificar las técnicas de diagramación y documentación para el desarrollo de software basado en tecnologías orientadas a componentes, según estándares de diseño de metodologías orientadas a componentes.

CE1.5 En un caso práctico para desarrollar componentes dentro de una arquitectura dada y contando con unas especificaciones funcionales precisas:

- Realizar la diagramación y documentación previa al desarrollo del componente, para optimizar los procesos de creación del componente según especificaciones recibidas.
- Identificar los diferentes interfaces y técnicas utilizadas para la intercomunicación de componentes, para poder aplicarlas al desarrollo de nuevos componentes.
- Definir los interfaces del componente software a desarrollar para la intercomunicación con el resto de componentes del sistema, según especificaciones técnicas de la arquitectura de componentes y necesidades funcionales.
- Diseñar la estructura del componente utilizando los estándares de creación de componentes, según especificaciones técnicas de la arquitectura utilizada y necesidades funcionales.
- Confeccionar la documentación del diseño realizado siguiendo los patrones, normativa y procedimientos especificados.

Contenidos

1. La orientación a objetos

- Principios de la orientación a objetos. Comparación con la programación estructurada:
  - Ocultación de información (information hiding)
  - El tipo abstracto de datos (ADT). Encapsulado de datos.
  - Paso de mensajes
- Conceptos básicos de orientación a objetos:
  - Clases:
    - Atributos, variables de estado y variables de clase
    - Métodos. Requisitos e invariantes.
    - Gestión de excepciones
    - Agregación de clases

- Objetos:
  - Creación y destrucción de objetos
  - Llamada a métodos de un objeto
  - Visibilidad y uso de las variables de estado
  - Referencias a objetos
  - Persistencia de objetos
  - Optimización de memoria y recolección de basura (garbage collection)
- Herencia:
  - Concepto de herencia. Superclases y subclases.
  - Herencia múltiple
  - Clases abstractas
  - Tipos de herencia: herencia de implementación, herencia de interfaces y de tipos y otros tipos de herencia
  - Polimorfismo y enlace dinámico (dynamic binding)
  - Directrices para el uso correcto de la herencia
- Modularidad:
  - Librerías de clases. Ámbito de utilización de nombres
  - Ventajas de la utilización de modulos o paquetes
- Genericidad y sobrecarga:
  - Concepto de genericidad
  - Concepto de Sobrecarga. Tipos de sobrecarga
  - Comparación entre genericidad y sobrecarga

- Desarrollo orientado a objetos:
  - Lenguajes de desarrollo orientado a objetos de uso común
  - Herramientas de desarrollo
- Lenguajes de modelización en el desarrollo orientado a objetos:
  - El lenguaje unificado de modelado (UML)
  - Diagramas para la modelización de sistemas orientados a objetos

2. La orientación a componentes

- Fundamentos conceptuales:
  - Definición de componente
  - Comparación entre componentes y objetos
  - Módulos
  - Interfaces:
    - Tipos de interfaces
    - Versionado de interfaces
    - Interfaces como contratos
  - Escalado de componentes
  - Estado de componentes
- Arquitecturas de componentes:
  - Basadas en objetos. Composición y uso de objetos
  - Multicapa
  - Basadas en middleware
  - Basadas en objetos distribuidos
- Diseño de componentes:
  - Principios de diseño de componentes:
    - Dependencias no cíclicas
    - Principio “open/closed”
    - Reusabilidad
    - Configurabilidad
    - Abstracción
    - Dependencias
  - Técnicas de reusabilidad:
    - Patrones
    - Librerías
    - Interfaces
    - Protocolos y esquemas de mensajes
    - Uso de lenguajes de programación
    - Estructuras y jerarquías de estructuras
    - Arquitecturas de sistemas

- Modelo de componente:
  - Especificación de servicios: transacciones, seguridad, persistencia y acceso remoto
  - Especificación de Interface
  - Especificación de la implementación
  - Especificación de las unidades de despliegue (modulos)
- Modelos de integración de componentes:
  - Referencias e identidad de objetos, componentes e interfaces
  - Servicios de localización
  - Modelos de intercambio: objetos distribuidos, capa intermedia (Middleware) e interacción e integración mediante servicios web
  - Comparación entre métodos de intercambio en las principales infraestructuras de componentes: OMG: CORBA, OMA, Java: JavaBeans, EJBs y Microsoft: COM, OLE/ActiveX, .NET
- Diagramación y documentación de componentes:
  - Modelo de información: diagramas conceptuales, diagramas de arquitectura de componentes y diagramas de despliegue.
  - Modelo dinámico: diagramas de interacción y de actividad, diagramas de casos de uso y diagramas de estado.

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: Programación de sistemas informáticos.