

DATOS IDENTIFICATIVOS DE LA UNIDAD FORMATIVA

UNIDAD FORMATIVA	DESARROLLO DE APLICACIONES WEB EN EL ENTORNO SERVIDOR.	DURACIÓN	90
		Específica	
Código	UF1844		
Familia profesional	INFORMÁTICA Y COMUNICACIONES		
Área Profesional	Desarrollo.		
Certificado de profesionalidad	DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB.	Nivel	3
Módulo formativo	Programación web en el entorno servidor.	Duración	240
Resto de unidades formativas que completan el módulo	Acceso a datos en aplicaciones web del entorno servidor.	Duración	90
	Desarrollo de aplicaciones web distribuidas.		60

Apartado A: REFERENTE DE COMPETENCIA

Esta unidad formativa se corresponde con la RP1.

Apartado B: ESPECIFICACIÓN DE LAS CAPACIDADES Y CONTENIDOS

Capacidades y criterios de evaluación

C1: Crear componentes software con tecnologías de desarrollo orientadas a objetos.

CE1.1 En un supuesto práctico en el que se pide gestionar componentes software en el entorno del servidor mediante herramientas de desarrollo y lenguajes de programación contando con documentación de diseño detallado:

- Crear y archivar componentes software.
- Modificar y eliminar componentes software.
- Depurar y verificar los componentes software elaborados.

CE1.2 Crear objetos, clases y métodos adecuados a la funcionalidad del componente software a desarrollar utilizando lenguajes de programación orientados a objetos.

CE1.3 Formular estructuras de datos y flujo de control mediante lenguajes de programación orientados a objetos según la funcionalidad del componente software a desarrollar.

CE1.4 Documentar el componente software desarrollado.

CE1.5 En un supuesto práctico en el que se pide construir componentes de software mediante herramientas de desarrollo y lenguajes de programación orientados a objetos a partir de documentación de diseño detallado:

- Integrar componentes software de control del contenido de los documentos ubicados en el servidor para ser utilizados en el entorno del cliente tipo servlet.
- Integrar gestión de ficheros en el componente software a desarrollar.
- Integrar gestión de errores en el componente software a desarrollar.
- Utilizar variables de servidor en el componente software a desarrollar para proporcionar acceso a las propiedades del servidor.
- Integrar seguimiento de sesiones de usuario y propiedades de la aplicación web a desarrollar en el componente software a construir.
- Crear componentes software con la funcionalidad de aplicación de cliente para ser utilizado en el entorno cliente tipo applet.
- Crear componentes software que puedan ofrecer su funcionalidad a otros componentes software del mismo servidor u otros servidores de la red.

Contenidos

1. El proceso del desarrollo de software.

- Modelos del ciclo de vida del software .
 - En cascada (waterfall).
 - Iterativo.
 - Incremental.
 - En V.
 - Basado en componentes (CBSE).

- Desarrollo rápido (RAD).
- Ventajas e inconvenientes. Pautas para la selección de la metodología más adecuada.
- Análisis y especificación de requisitos.
 - Tipos de requisitos.
 - Modelos para el análisis de requisitos.
 - Documentación de requisitos.
 - Validación de requisitos.
 - Gestión de requisitos.
- Diseño.
 - Modelos para el diseño de sistemas.
 - Diagramas de diseño. El estándar UML.
 - Documentación.
- Implementación. Conceptos generales de desarrollo de software.
 - Principios básicos del desarrollo de software.
 - Técnicas de desarrollo de software.
- Validación y verificación de sistemas.
 - Planificación.
 - Métodos formales de verificación.
 - Métodos automatizados de análisis.
- Pruebas de software.
 - Tipos.
 - Pruebas funcionales (BBT).
 - Pruebas estructurales (WBT).
 - Comparativa. Pautas de utilización.
 - Diseño de pruebas.
 - Ámbitos de aplicación.
 - Pruebas de Sistemas.
 - Pruebas de componentes.
 - Automatización de pruebas. Herramientas.
 - Estándares sobre pruebas de software.
- Calidad del software.
 - Principios de calidad del software.
 - Métricas y calidad del software.
 - Concepto de métrica y su importancia en la medición de la calidad.
 - Principales métricas en las fases del ciclo de vida software.
 - Estándares para la descripción de los factores de Calidad.
 - ISO-9126.
 - Otros estándares. Comparativa.
- Herramientas de uso común para el desarrollo de software
 - Editores orientados a lenguajes de programación.
 - Compiladores y enlazadores.
 - Generadores de programas.
 - Depuradores.
 - De prueba y validación de software.
 - Optimizadores de código.
 - Empaquetadores.
 - Generadores de documentación de software.
 - Gestores y repositorios de paquetes. Versionado y control de dependencias.
 - De distribución de software.
 - Gestores de actualización de software.
 - De control de versiones.
 - Entornos integrados de desarrollo (IDE) de uso común.
- Gestión de proyectos de desarrollo de software.
 - Planificación de proyectos.
 - Control de proyectos.
 - Ejecución de proyectos.
 - Herramientas de uso común para la gestión de proyectos.

2. 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.
- Clases de objetos.
 - 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.
 - 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 módulos 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.
 - Uso del lenguaje unificado de modelado (UML) en el desarrollo orientado a objetos.
 - Diagramas para la modelización de sistemas orientados a objetos.

3. Arquitecturas web.

- Concepto de arquitectura web.
- El modelo de capas.
- Plataformas para el desarrollo en las capas servidor.
- Herramientas de desarrollo orientadas a servidor de aplicaciones web.
 - Tipos de herramientas.
 - Extensibilidad. Instalación de módulos.
 - Técnicas de configuración de los entornos de desarrollo, preproducción y producción.
 - Funcionalidades de depuración.

4. Lenguajes de programación de aplicaciones web en el lado servidor.

- Características de los lenguajes de programación web en servidor.
- Tipos y características de los lenguajes de uso común.
 - Interpretados orientados a servidor.
 - Lenguajes de cliente interpretados en servidor.
 - Lenguajes compilados.
- Criterios en la elección de un lenguaje de programación web en servidor. Ventajas e inconvenientes.
- Características generales.
 - Tipos de datos.
 - Clases.
 - Operadores básicos. Manipulación de cadenas de caracteres.

- Estructuras de control. Bucles y condicionales.
- Módulos o paquetes.
- Herencia.
- Gestión de bibliotecas (libraries).
- Gestión de la configuración.
 - Configuración de descriptores.
 - Configuración de ficheros.
- Gestión de la seguridad.
 - Conceptos de identificación, autenticación y autorización.
 - Técnicas para la gestión de sesiones.
- Gestión de errores.
 - Técnicas de recuperación de errores.
 - Programación de excepciones.
- Transacciones y persistencia.
 - Acceso a bases de datos. Conectores.
 - Estándares para el acceso a bases de datos.
 - Gestión de la configuración de acceso a bases de datos.
 - Acceso a directorios y otras fuentes de datos.
 - Programación de transacciones.
- Componentes en servidor. Ventajas e inconvenientes en el uso de contenedores de componentes.
- Modelos de desarrollo. El modelo vista controlador.
 - Modelo: programación de acceso a datos.
 - Vista: Desarrollo de aplicaciones en cliente. Eventos e interfaz de usuario.
 - Programación del controlador.
 - Documentación del software. Inclusión en código fuente. Generadores 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.