

DATOS IDENTIFICATIVOS DE LA UNIDAD FORMATIVA

UNIDAD FORMATIVA	DESARROLLO Y OPTIMIZACIÓN DE COMPONENTES SOFTWARE PARA TAREAS ADMINISTRATIVAS DE SISTEMAS	Duración	90
		Específica	
Código	UF1286		
Familia profesional	INFORMÁTICA Y COMUNICACIONES		
Área Profesional	Sistemas y telemática		
Certificado de profesionalidad	Programación de sistemas informáticos	Nivel	3
Módulo formativo	Desarrollo de elementos software para gestión de sistemas	Duración	210
Resto de unidades formativas que completan el módulo	Desarrollo de componentes software para el manejo de dispositivos	Duración	60
	Desarrollo de componentes software para servicios de comunicaciones	Duración	60

Apartado A: REFERENTE DE COMPETENCIA

Esta unidad formativa se corresponde con la RP1 y RP2 de la UC0964_3 CREAR ELEMENTOS SOFTWARE PARA LA GESTIÓN DEL SISTEMA Y SUS RECURSOS.

Apartado B: ESPECIFICACIÓN DE LAS CAPACIDADES Y CONTENIDOS

Capacidades y criterios de evaluación

C1: Analizar las especificaciones técnicas proporcionadas para el desarrollo a realizar, comprobando su validez y optimización.

CE1.1 Clasificar las principales arquitecturas de sistemas operativos (monolítico, modular, microkernel, sistemas distribuidos) y sus características, para identificar las necesidades de gestión y administración de dichos sistemas según especificaciones técnicas.

CE1.2 Explicar los apartados de un análisis de requisitos, detallando los diagramas básicos utilizados para la especificación funcional y de datos según metodologías y estándares de diseño.

CE1.3 Analizar las especificaciones técnicas del servicio o herramienta de gestión que se desea implementar, para identificar los recursos requeridos del componente según las especificaciones funcionales dadas.

CE1.4 Realizar los diagramas previos a la fase de desarrollo siguiendo las especificaciones técnicas y criterios de calidad especificados.

CE1.5 En un supuesto práctico, para interpretar un análisis de requisitos dado para el desarrollo de un elemento software, teniendo en cuenta las necesidades de administración de los recursos del sistema informático:

- Definir los módulos software a realizar a partir de las especificaciones técnicas y catálogo de requisitos.
- Establecer las relaciones entre módulos determinando entradas, salidas y flujos de datos según el diseño funcional y las especificaciones del sistema.
- Determinar las estructuras necesarias para representar la información especificada en los requisitos.
- Crear las estructuras necesarias para realizar el seguimiento de requisitos durante la codificación del elemento software según unos criterios de calidad especificados.

C2: Desarrollar elementos software destinados a la gestión de los recursos del sistema, mediante herramientas y lenguajes de programación de sistemas.

CE2.1 Enumerar las herramientas y lenguajes estructurados, orientados al desarrollo de programas para la gestión de recursos del sistema, detallando sus características.

CE2.2 Describir las técnicas de funcionamiento y principios de los sistemas de memoria, detallando su organización en jerarquías, para desarrollar elementos software que las utilicen según las especificaciones funcionales aportadas.

CE2.3 Clasificar las arquitecturas de entrada/salida, de buses y de microprocesadores en sistemas, explicando las técnicas y procesos funcionales utilizados para el desarrollo de los elementos software, según unas especificaciones funcionales dadas.

CE2.4 Distinguir las funciones de las librerías del sistema para la elaboración de nuevos componentes software, reutilizando el código ya desarrollado e implementando nuevos elementos en dichas librerías u obteniendo componentes aislados, según unos criterios de optimización y calidad especificados.

CE2.5 Describir los tipos de pruebas, tanto funcionales como estructurales, y los procesos de depuración a los que debe ser sometido un componente desarrollado y las herramientas utilizadas, para verificar su funcionalidad e integración con el resto de componentes del sistema, según unos criterios de calidad especificados.

CE2.6 Clasificar las herramientas utilizadas en las fases de desarrollo: generación del código, creación de los módulos ejecutables, control de versiones, depuración y pruebas, documentación y empaquetado para su distribución para optimizar la generación y asegurar la calidad de los productos del desarrollo, según unos criterios de seguridad y calidad especificada.

CE2.7 En varios supuestos prácticos para desarrollar elementos software para la gestión de los recursos del sistema, dadas unas especificaciones técnicas:

- Diseñar los algoritmos asociados al elemento software utilizando técnicas de desarrollo estructurado.

- Codificar los módulos software a partir de los algoritmos diseñados utilizando herramientas y lenguajes estructurados.
- Codificar estructuras de datos utilizando las funcionalidades proporcionadas por el lenguaje estructurado.
- Utilizar los recursos y librerías disponibles en las herramientas de desarrollo para realizar la codificación de los algoritmos.
- Diseñar y codificar los manejadores de errores necesarios para garantizar el óptimo funcionamiento del módulo software.
- Diseñar y codificar componentes que permitan el acceso concurrente a los recursos del sistema.
- Depurar los módulos desarrollados utilizando las herramientas disponibles.
- Diseñar y aplicar baterías de pruebas sobre los módulos desarrollados para comprobar su correcto funcionamiento y documentar los resultados obtenidos.
- Documentar los módulos desarrollados para facilitar su revisión y futuras modificaciones y ampliaciones.
- Realizar la implantación de los módulos documentando el proceso y las incidencias detectadas.

CE2.8 Interpretar la documentación técnica asociada a las herramientas y lenguajes de programación, incluso si está editada en la lengua extranjera de uso más frecuente en el sector, utilizándola de ayuda en el desarrollo.

Contenidos

1. Descripción de los servicios, estructura y administración de Sistemas Operativos

- o Definición y conceptos básicos sobre Sistemas Operativos:
 - Descripción de los servicios básicos ofrecidos por un Sistema Operativo
 - Gestión de memoria. Memoria virtual
 - Ejecución de programas y gestión de procesos
 - Gestión del almacenamiento. Sistemas de Archivos
 - Gestión de dispositivos de entrada/salida
 - Gestión de red
 - Gestión de errores
 - Gestión de la seguridad
 - Auditoría (logs del sistema)
 - Procesos de arranque (boot) y finalización del sistema (shutdown)
- o Características estructurales de los Sistemas Operativos:
 - Sistemas monolíticos
 - Microkernels
 - Sistemas modulares y por capas
 - Máquinas virtuales
 - Sistemas distribuidos
- o Herramientas administrativas de uso común en Sistemas Operativos:
 - Interfaces de usuario gráficos
 - Intérpretes de comandos

2. Programación de sistemas operativos. Lenguajes y librerías de uso común

- o Las llamadas al sistema (System Calls):
 - Definición
 - Uso directo y mediante Application Programming Interfaces (APIs)
 - Principales tipos de llamadas al sistema:
 - o Control de procesos
 - o Gestión de ficheros
 - o Gestión de dispositivos
 - o Información del sistema
 - o Comunicaciones
 - Descripción y uso de las APIs estándar de uso común para llamadas a sistema:
 - o Win32 API (Sistemas Windows)
 - o POSIX API (Sistemas Unix, Linux, Mac)
 - o Java API (Multiplataforma)
 - o Programas de utilidades y comandos del sistema:
 - Principales tipos:
 - o Operaciones con ficheros y directorios
 - o Funciones de estado
 - o Edición y manipulación de ficheros
 - o Soporte para lenguajes de programación (compiladores, enlazadores, ensambladores, intérpretes, etc.)
 - o Ejecución de programas
 - o Comunicaciones, mensajería, intercambio remoto de archivos, etc.
 - Uso de utilidades y comandos mediante lenguajes de script de uso común
 - o Windows scripting

- Linux/Unix scripting

3. El ciclo de vida del software de gestión de sistemas

- 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.
- Descripción de las fases en el ciclo de vida del software:
 - Análisis y especificación de requisitos
 - Tipos de requisitos : funcionales/ no funcionales, de usuario, de interfaz, de seguridad y de rendimiento
 - Modelos para el análisis de requisitos
 - Documentación de requisitos
 - Validación de requisitos
 - Gestión de requisitos
 - Diseño:
 - Modelos para el diseño de sistemas : contexto y arquitectura, procesos, datos, objetos, interfaces de usuario, componentes y despliegues
 - Diagramas de diseño: diagramas de entidad-relación, diagramas de flujo, diagramas de contexto y UML. Diagramas UML de uso común en diseño de sistemas
 - Documentación: herramientas de generación de documentación y documentación el código
 - Implementación. Conceptos generales de desarrollo de software:
 - Principios básicos del desarrollo de software
 - Técnicas de desarrollo de software : basadas en prototipos, basadas en componentes, métodos de desarrollo rápido y otras técnicas de desarrollo
 - Validación, verificación y pruebas:
 - Validación y verificación de sistemas: planificación, métodos formales de verificación y métodos automatizados de análisis
 - Pruebas de software: tipos, diseño de pruebas, ámbito de aplicación, automatización de pruebas, herramientas y 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

4. Desarrollo del software de gestión de sistemas

- Análisis de especificaciones para el desarrollo de software de gestión de sistemas:
 - Identificación de los componentes necesarios según las especificaciones
 - Análisis de los componentes reutilizables
 - Análisis de la integración de los componentes en la arquitectura del sistema
 - Identificación de los modelos funcionales y de datos de los componentes
- Técnicas de programación presentes en lenguajes de uso común aplicables al desarrollo de software de gestión de sistemas:
 - Programación estructurada:
 - Tipos primitivos y estructurados
 - Variables. Ámbito de utilización
 - Operadores aritméticos y lógicos
 - Estructuras de control. Bucles, condicionales y selectores
 - Funciones y procedimientos. Parámetros por valor y referencia.
 - Recursividad
 - Programación de elementos básicos: cadenas, fechas y ficheros.
 - Conversiones de tipos
 - Manejo de errores (excepciones)
 - Lenguajes estructurados de uso común
 - Programación orientada a objetos:

- Clases y objetos
- Herencia, polimorfismo y sobrecarga dinámica de métodos
- Propiedades: selectores (get), modificadores (set) y referencias (let)
- Lenguajes orientados a objetos de uso común
- Técnicas de programación de software de gestión de sistemas:
 - Reutilización de código.
 - Uso de librerías del sistema
 - Llamadas a utilidades y aplicaciones del sistema
 - Técnicas específicas aplicables a los servicios básicos del sistema:
 - Programación de la gestión de los procesos: multitarea, control de bloqueos(deadlock) y comunicación entre procesos
 - Programación de la gestión de memoria: jerarquías de memoria, paginación de memoria, segmentación de memoria, intercambio(swapping), compartición de memoria, seguridad y memoria virtual
 - Programación de los sistemas de archivos: acceso a archivos y directorios, atributos y mecanismos de protección
 - Programación de los sistemas de entrada y salida: gestión de interrupciones, acceso directo a memoria (DMA) , puertos de entrada/salida y asignación de memoria
 - Programación de la seguridad: control de variables, control de desbordamiento de búferes , aserciones, precondiciones y post-condiciones.
 - Técnicas de optimización
- Control de calidad del desarrollo del software de gestión de sistemas:
 - Métricas aplicables
 - Verificación de requisitos
 - Proceso de mejora continua
- Herramientas de uso común para el desarrollo de software de sistemas:
 - 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
 - Despliegue de software:
 - Gestores y repositorios de paquetes. Versionado y control de dependencias
 - Distribución de software
 - Gestores de actualización de software
 - De control de versiones
 - Entornos integrados de desarrollo (IDE) de uso común:
 - Específicos de sistemas Windows
 - Específicos de sistemas Unix
 - Multiplataforma.

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.