

el suficiente para un mínimo de 15 alumnos y deberá incrementarse, en su caso, para atender a número superior.

En el caso de que la formación se dirija a personas con discapacidad se realizarán las adaptaciones y los ajustes razonables para asegurar su participación en condiciones de igualdad.

ANEXO VI

I. IDENTIFICACIÓN DEL CERTIFICADO DE PROFESIONALIDAD

Denominación: PROGRAMACIÓN DE SISTEMAS INFORMÁTICOS

Código: IFCT0609

Familia profesional: Informática y Comunicaciones

Área profesional: Sistemas y telemática

Nivel de cualificación profesional: 3

Cualificación profesional de referencia:

IFC303_3 Programación de sistemas informáticos (RD 1201/2007, de 14 de septiembre)

Relación de unidades de competencia que configuran el certificado de profesionalidad:

UC0490_3: Gestionar servicios en el sistema informático.

UC0964_3: Crear elementos software para la gestión del sistema y sus recursos

UC0965_3: Desarrollar elementos software con tecnologías de programación basada en componentes.

Competencia general:

Desarrollar componentes software a partir de unas especificaciones concretas, proporcionando funciones de administración y supervisión del sistema operativo, para la gestión de los recursos de un sistema informático y la interacción con otros sistemas utilizando tecnologías de desarrollo orientadas a objetos y a componentes.

Entorno Profesional:

Ámbito profesional:

Desarrolla su actividad profesional tanto por cuenta propia, como por cuenta ajena en empresas o entidades públicas o privadas de cualquier tamaño, que dispongan de equipos informáticos para su gestión, en el área de sistemas o de desarrollo del departamento de informática.

Sectores productivos:

Se ubica sobre todo en el sector servicios, y principalmente en los siguientes tipos de empresas: empresas o entidades que utilizan sistemas informáticos para su gestión; empresas que tienen como objetivo de negocio la comercialización de servicios de

análisis, diseño y construcción de aplicaciones informáticas; grandes organizaciones, siendo parte del equipo de programación y mantenimiento de sistemas informáticos.

Ocupaciones o puestos de trabajo relacionados:

2711.1019 Analista de Sistemas, nivel superior
2712.1030 Analista Programador, nivel medio
2712.1012 Analista de Aplicaciones, nivel medio
3820.1017 Programador de Aplicaciones Informáticas
Programador de sistemas.
Programador de componentes.

Duración de la formación asociada: 590 horas

Relación de módulos formativos y de unidades formativas:

MF0490_3: (Transversal) Gestión de servicios en el sistema informático. (90 horas)
MF0964_3: Desarrollo de elementos software para gestión de sistemas. (210 horas)

- UF1286: Desarrollo y optimización de componentes software para tareas administrativas de sistemas. (90 horas)
- UF1287: Desarrollo de componentes software para el manejo de dispositivos (drivers). (60 horas)
- UF1288: Desarrollo de componentes software para servicios de comunicaciones. (60 horas)

MF0965_3: Desarrollo de software basado en tecnologías orientadas a componentes. (210 horas)

- UF1289: Diseño de elementos software con tecnologías basadas en componentes (90 horas)
- UF1290: Implementación e integración de elementos software con tecnologías basadas en componentes (90 horas)
- UF1291: Despliegue y puesta en funcionamiento de componentes software (30 horas)

MP0274: Módulo de prácticas profesionales no laborales de Programación de Sistemas informáticos (80 horas)

II. PERFIL PROFESIONAL DEL CERTIFICADO DE PROFESIONALIDAD

Unidad de competencia 1

Denominación: GESTIONAR SERVICIOS EN EL SISTEMA INFORMÁTICO

Nivel: 3

Código: UC0490_3

Realizaciones profesionales y criterios de realización

RP1: Gestionar la configuración del sistema para asegurar el rendimiento de los procesos según las necesidades de uso y dentro de las directivas de la organización.

CR1.1 Los procesos que intervienen en el sistema son identificados para evaluar parámetros de rendimiento.

CR1.2 Los parámetros que afectan a los componentes del sistema: memoria, procesador y periféricos, entre otros, se ajustan a las necesidades de uso.

CR1.3 Las prioridades de ejecución de los procesos se adecuan en función de las especificaciones del plan de explotación de la organización.

CR1.4 Las herramientas de monitorización se implantan y configuran determinando los niveles de las alarmas en función del plan de explotación de la organización.

RP2: Administrar los dispositivos de almacenamiento según las necesidades de uso y dentro de las directivas de la organización.

CR2.1 Los dispositivos de almacenamiento se configuran para ser usados en los distintos sistemas operativos utilizados en el sistema informático.

CR2.2 La estructura de almacenamiento se define y se implanta atendiendo a las necesidades de los distintos sistemas de archivos y a las especificaciones de uso de la organización.

CR2.3 Los requerimientos de nomenclatura de objetos y restricciones de uso de cada dispositivo de almacenamiento se documentan adecuadamente.

CR2.4 Los dispositivos de almacenamiento se integran para ofrecer un sistema funcional al usuario según las especificaciones de la organización.

RP3: Gestionar las tareas de usuarios para garantizar los accesos al sistema y la disponibilidad de los recursos según especificaciones de explotación del sistema informático.

CR3.1 El acceso de los usuarios al sistema informático se configura para garantizar la seguridad e integridad del sistema según las especificaciones de la organización.

CR3.2 El acceso de los usuarios a los recursos se administra mediante la asignación de permisos en función de las necesidades de la organización.

CR3.3 Los recursos disponibles para los usuarios se limitan con las herramientas adecuadas en base a lo especificado en las normas de uso de la organización.

RP4: Gestionar los servicios de red para asegurar la comunicación entre sistemas informáticos según necesidades de explotación.

CR4.1 Los dispositivos de comunicaciones son verificados en lo que respecta a su configuración y rendimiento según las especificaciones de la organización.

CR4.2 Los servicios de comunicaciones son identificados en el sistema con sus procesos correspondientes para analizar los consumos de recursos y verificar que están dentro de lo permitido por las especificaciones del plan de explotación de la organización.

CR4.3 Las incidencias en los servicios de comunicaciones se detectan y se documentan para informar a los responsables de la explotación del sistema y de la gestión de las comunicaciones según los protocolos de la organización.

Contexto profesional

Medios de producción

Sistemas operativos. Herramientas de administración de usuarios y gestión de permisos a recursos. Herramientas de control de rendimiento. Herramientas de monitorización de procesos. Herramientas de monitorización de uso de memoria. Herramientas de monitorización de gestión de dispositivos de almacenamiento. Herramientas de gestión de usuarios.

Productos y resultados

Sistema operando correctamente. Rendimiento del sistema adecuado a los parámetros de explotación. Sistema seguro e íntegro en el acceso y utilización de recursos. Servicios de comunicaciones en funcionamiento.

Información utilizada o generada

Manuales de explotación del sistema operativo y de los dispositivos. Plan de explotación de la organización. Manuales de las herramientas de monitorización utilizadas. Gráficas y análisis de rendimiento. Listados de acceso y restricciones de usuarios. Informe de incidencias. Protocolo de actuación ante incidencias.

Unidad de competencia 2

Denominación: CREAR ELEMENTOS SOFTWARE PARA LA GESTIÓN DEL SISTEMA Y SUS RECURSOS

Nivel: 3

Código: UC0964_3

Realizaciones profesionales y criterios de realización

RP1: Desarrollar componentes software que implementen servicios y herramientas de gestión del sistema operativo, utilizando lenguajes orientados a la programación de sistemas, para soportar tareas administrativas según necesidades funcionales dadas.

CR1.1 Las especificaciones técnicas del servicio o herramienta de gestión a implementar, se analizan para identificar los recursos para el desarrollo del componente según necesidades funcionales detectadas.

CR1.2 Los diagramas y la documentación previa al desarrollo se realizan, para ser utilizados como soporte de la creación de los componentes según especificaciones de la organización.

CR1.3 El desarrollo del código del componente se realiza, asistido por el uso de herramientas editoras y depuradoras para optimizar los rendimientos según especificaciones de la organización.

CR1.4 Los componentes software que implementan los servicios y herramientas de gestión se programan, para dar soporte a las funciones definidas de acuerdo a las especificaciones técnicas del diseño suministrado.

CR 1.5 El plan de prueba se elabora con el fin de comprobar la funcionalidad de los componentes desarrollados, según especificaciones y criterios de calidad establecidos.

CR1.6 Los componentes software de servicios y de herramientas desarrollados se prueban y depuran, para corregir los errores utilizando las herramientas de depuración del entorno de programación según los criterios de calidad establecidos.

CR1.7 La documentación de los componentes software de servicios y herramientas de gestión se realiza, para cumplimentar el registro de la información producida siguiendo los patrones, normativa y procedimientos especificados en el diseño.

CR1.8 La documentación técnica específica asociada, se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

RP2: Codificar y utilizar funciones de las librerías del sistema en el desarrollo de componentes software, para optimizar los desarrollos según especificaciones técnicas y funcionales.

CR2.1 Las funciones documentadas de las librerías del sistema se identifican y catalogan, para facilitar la localización de la información de las mismas según necesidades de desarrollo.

CR2.2 Las funciones de las librerías del sistema se utilizan en la elaboración de nuevos componentes software, para mejorar los rendimientos de los desarrollos mediante la reutilización del código escrito y probado, según las especificaciones técnicas de cada función y los requisitos de invocación de las mismas.

CR2.3 Los componentes software se desarrollan con los requisitos especificados, para ser incluidos en librerías para su posterior uso y distribución, según necesidades y especificaciones técnicas.

CR2.4 Las pruebas funcionales y estructurales del componente realizado se planifican y se realizan, para comprobar y asegurar los objetivos del desarrollo según especificaciones técnicas y de calidad de la organización.

CR2.5 La documentación de las librerías del sistema operativo desarrolladas se realiza, para cumplimentar las necesidades de registro siguiendo los patrones, normativa y procedimientos especificados en el diseño.

RP3: Elaborar componentes software utilizando lenguajes orientados a la programación de sistemas, según especificaciones establecidas para manejar dispositivos hardware.

CR3.1 La documentación técnica con las especificaciones de los dispositivos hardware se interpreta, para identificar las características y los parámetros de la programación del manejador de dispositivo, de acuerdo al diseño suministrado.

CR3.2 La documentación técnica de las herramientas software a utilizar y del sistema operativo se interpreta, para identificar las características y los parámetros de la programación del manejador de dispositivo de acuerdo al diseño suministrado.

CR3.3 Las herramientas de programación se utilizan para desarrollar y depurar los posibles errores del código desarrollado, según criterios de calidad de la organización.

CR3.4 Las pruebas del manejador del dispositivo elaborado se planifican y se realizan en los posibles escenarios en los que puede ser implantado, para asegurar su funcionalidad y la ausencia de conflictos con el resto de los componentes del sistema, según especificaciones técnicas y normativa de calidad de la organización.

CR3.5 La documentación técnica y de usuario del manejador desarrollado, se confecciona según los parámetros y la normativa de la organización.

CR3.6 La documentación técnica específica asociada, se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

RP4: Confeccionar componentes software que implementen servicios de comunicaciones, para enlazar distintos sistemas según estándares de desarrollo.

CR4.1 Las especificaciones del servicio se interpretan para discriminar los elementos que intervendrán en el desarrollo del componente como puertos de comunicaciones entre los sistemas y protocolos estándares seleccionados, entre otros, según especificaciones y necesidades del servicio.

CR4.2 El desarrollo del componente se realiza en entornos cliente/servidor, para implementar las funcionalidades del servicio de comunicaciones según especificaciones técnicas y funcionales aportadas.

CR4.3 La codificación del componente se realiza utilizando herramientas de programación y depuración, para optimizar la fase de desarrollo según especificaciones de la organización.

CR4.4 El componente se somete a baterías de pruebas en réplicas de los posibles escenarios de su implantación posterior, para asegurar la funcionalidad e integridad según criterios de calidad y seguridad de la organización.

CR4.5 La documentación del desarrollo y pruebas realizadas se confecciona siguiendo los patrones, normativa y procedimientos especificados en el diseño.

Contexto profesional

Medios de producción

Cortafuegos antivirus y servidores proxy.
Entornos integrados de desarrollo.
Equipos informáticos, periféricos y dispositivos hardware.
Herramientas de control de cambios.
Herramientas de prueba.
Herramientas de depuración.
Herramientas de desarrollo o entornos integrados (IDE).
Herramientas de distribución de aplicaciones.
Herramientas de documentación de elementos de programación.
Herramientas ofimáticas.
Lenguajes 4GL.
Lenguajes estructurados.
Lenguajes orientados a objetos.
Lenguajes de programación concurrentes.

Productos y resultados

Código ejecutable y código fuente del software desarrollado.
Procedimientos y casos de prueba desarrollados.
Programas de prueba.
Sistema operativo y aplicaciones configurados y parametrizados de acuerdo a las necesidades.

Información utilizada o generada

Documentación sobre los casos y datos de prueba desarrollados.
Documentación técnica del diseño del software a desarrollar.
Documentación técnica y de usuario del software desarrollado.
Manuales de funcionamiento del software.
Manuales de interfaces de programación (API) del sistema operativo.
Manuales de la herramienta de programación empleada.
Manuales de uso del sistema operativo.
Manuales del entorno de programación (IDE).
Manuales del lenguaje de programación empleado.
Manuales técnicos del dispositivo hardware a programar.
Normas corporativas de desarrollo de software, de pruebas, de control de calidad.
Sistemas de ayuda de las aplicaciones informáticas.
Soportes técnicos para asistencia (telefónica, Internet, mensajería y foros, entre otros).

Unidad de competencia 3

Denominación: DESARROLLAR ELEMENTOS SOFTWARE CON TECNOLOGÍAS DE PROGRAMACIÓN BASADA EN COMPONENTES

Nivel: 3

Código: UC0965_3

Realizaciones profesionales y criterios de realización

RP1: Realizar el diseño del componente software, para su posterior desarrollo según la tecnología de componentes especificada.

Contexto profesional

Medios de producción

Entornos integrados de desarrollo.
Equipos informáticos y periféricos de comunicaciones.
Herramientas de control de cambios.
Herramientas de depuración.
Herramientas de desarrollo o entornos integrados (IDE).
Herramientas de distribución de aplicaciones.
Herramientas de documentación de elementos de programación.
Herramientas de gestión de cambios, incidencias y configuración.
Herramientas de prueba.
Herramientas ofimáticas.
Lenguajes 4GL.
Lenguajes de manipulación de datos.
Lenguajes estructurados.
Lenguajes orientados a objetos.
Servicios de transferencia de ficheros y mensajería.
Sistemas operativos y parámetros de configuración.

Productos y resultados

Código ejecutable y código fuente del software desarrollado.
Paquete de instalación y/o despliegue del software desarrollado.
Procedimientos y casos de prueba.
Programas de prueba realizados.
Sistema informático en funcionamiento con un rendimiento óptimo y una utilización adecuada de sus recursos.
Sistema operativo y aplicaciones configurados y parametrizados de acuerdo a las necesidades.

Información utilizada o generada

Documentación sobre los casos y datos de prueba desarrollados.
Documentación técnica del diseño del software a desarrollar.
Documentación técnica y de usuario del software desarrollado.
Manuales de interfaces de programación (API) del sistema operativo.
Manuales de uso del sistema operativo. Manuales de uso y funcionamiento de los sistemas informáticos.
Manuales del entorno de programación (IDE).
Manuales del lenguaje de programación.
Manuales del lenguaje de programación empleado.
Manuales técnicos de los dispositivos de comunicaciones sobre los que se vaya a programar.
Normas corporativas de desarrollo de software, de pruebas, de control de calidad.
Soportes técnicos de asistencia (telefónica, Internet, mensajería y foros, entre otros).

III. FORMACIÓN DEL CERTIFICADO DE PROFESIONALIDAD

MÓDULO FORMATIVO 1

Denominación: GESTIÓN DE SERVICIOS EN EL SISTEMA INFORMÁTICO

Código: MF0490_3

Nivel de cualificación profesional: 3

Asociado a la Unidad de Competencia:

UC0490_3: Gestionar servicios en el sistema informático

Duración: 90 horas

Capacidades y criterios de evaluación

C1: Analizar los procesos del sistema con objeto de asegurar un rendimiento adecuado a los parámetros especificados en el plan de explotación.

CE1.1 Identificar los procesos del sistema y los parámetros que los caracterizan (procesos padre, estado del proceso, consumo de recursos, prioridades y usuarios afectados entre otros) para determinar su influencia en el rendimiento del sistema.

CE1.2 Describir cada una de las herramientas provistas por el sistema para la gestión de procesos con objeto de permitir la intervención en el rendimiento general del sistema.

CE1.3 Explicar técnicas de monitorización y herramientas destinadas a evaluar el rendimiento del sistema.

CE1.4 En un supuesto práctico en el que se cuenta con un sistema informático con una carga de procesos debidamente caracterizada:

- Utilizar las herramientas del sistema para identificar cuantos procesos activos existen y las características particulares de alguno de ellos.
- Realizar las operaciones de activación, desactivación y modificación de prioridad entre otras con un proceso utilizando las herramientas del sistema.
- Monitorizar el rendimiento del sistema mediante herramientas específicas y definir alarmas, que indiquen situaciones de riesgo.

C2: Aplicar procedimientos de administración a dispositivos de almacenamiento para ofrecer al usuario un sistema de registro de la información íntegro, seguro y disponible.

CE2.1 Identificar los distintos sistemas de archivo utilizables en un dispositivo de almacenamiento dado para optimizar los procesos de registro y acceso a los mismos.

CE2.2 Explicar las características de los sistemas de archivo en función de los dispositivos de almacenamiento y sistemas operativos empleados.

CE2.3 Describir la estructura general de almacenamiento en el sistema informático asociando los dispositivos con los distintos sistemas de archivos existentes.

CE2.4 En un supuesto práctico en el que se dispone de un sistema de almacenamiento de la información con varios dispositivos:

- Realizar el particionamiento, en los casos que sea necesario, y la generación de la infraestructura de los sistemas de archivo a instalar en cada dispositivo.
- Implementar la estructura general de almacenamiento integrando todos los dispositivos y sus correspondientes sistemas de archivos.
- Documentar los requerimientos y restricciones de cada sistema de archivos implantado.

C3: Administrar el acceso al sistema y a los recursos para verificar el uso adecuado y seguro de los mismos.

CE3.1 Identificar las posibilidades de acceso al sistema distinguiendo los accesos remotos de los accesos locales.

CE3.2 Describir las herramientas que se utilizan en la gestión de permisos a usuarios para el uso de los recursos del sistema.

CE3.3 En un supuesto práctico en el que se cuenta con derecho de administración de usuarios:

- Identificar los posibles accesos de un usuario al sistema.
- Modificar los permisos de utilización de un recurso del sistema a un usuario.
- Definir limitaciones de uso de un recurso del sistema a los usuarios.

C4: Evaluar el uso y rendimiento de los servicios de comunicaciones para mantenerlos dentro de los parámetros especificados.

CE4.1 Explicar los parámetros de configuración y funcionamiento de los dispositivos de comunicaciones para asegurar su funcionalidad dentro del sistema.

CE4.2 Relacionar los servicios de comunicaciones activos en el sistema con los dispositivos utilizados por ellos con objeto de analizar y evaluar el rendimiento.

CE4.3 En un supuesto práctico en el que tomamos un sistema informático conectado con el exterior por medio de varias líneas de comunicaciones:

- Identificar los dispositivos de comunicaciones y describir sus características.
- Verificar el estado de los servicios de comunicaciones.
- Evaluar el rendimiento de los servicios de comunicaciones.
- Detectar y documentar las incidencias producidas en el sistema.

Contenidos

1. Gestión de la seguridad y normativas en sistemas informáticos

- Norma ISO 27002 Código de buenas practicas para la gestión de la seguridad de la información
- Metodología ITIL Librería de infraestructuras de las tecnologías de la información
- Ley orgánica de protección de datos de carácter personal.
- Normativas mas frecuentemente utilizadas para la gestión de la seguridad física

2. Análisis de los procesos de sistemas informáticos

- Identificación de procesos de negocio soportados por sistemas de información
- Características fundamentales de los procesos electrónicos:
 - o Estados de un proceso,
 - o Manejo de señales, su administración y los cambios en las prioridades
- Determinación de los sistemas de información que soportan los procesos de negocio y los activos y servicios utilizados por los mismos
- Análisis de las funcionalidades de sistema operativo para la monitorización de los procesos y servicios
- Técnicas utilizadas para la gestión del consumo de recursos

3. Demostración de sistemas informáticos de almacenamiento

- Tipos de dispositivos de almacenamiento más frecuentes
- Características de los sistemas de archivo disponibles
- Organización y estructura general de almacenamiento
- Herramientas del sistema para gestión de dispositivos de almacenamiento

4. Utilización de métricas e indicadores de monitorización de rendimiento de sistemas informáticos

- Criterios para establecer el marco general de uso de métricas e indicadores para la monitorización de los sistemas de información

- Identificación de los objetos para los cuales es necesario obtener indicadores
- Aspectos a definir para la selección y definición de indicadores
- Establecimiento de los umbrales de rendimiento de los sistemas de información
- Recolección y análisis de los datos aportados por los indicadores
- Consolidación de indicadores bajo un cuadro de mandos de rendimiento de sistemas de información unificado

5. **Confección del proceso de monitorización de sistemas y comunicaciones**

- Identificación de los dispositivos de comunicaciones
- Análisis de los protocolos y servicios de comunicaciones
- Principales parámetros de configuración y funcionamiento de los equipos de comunicaciones
- Procesos de monitorización y respuesta
- Herramientas de monitorización de uso de puertos y servicios tipo Sniffer
- Herramientas de monitorización de sistemas y servicios tipo Hobbit, Nagios o Cacti
- Sistemas de gestión de información y eventos de seguridad (SIM/SEM)
- Gestión de registros de elementos de red y filtrado (router, switch, firewall, IDS/IPS, etc.)

6. **Selección del sistema de registro de en función de los requerimientos de la organización**

- Determinación del nivel de registros necesarios, los periodos de retención y las necesidades de almacenamiento
- Análisis de los requerimientos legales en referencia al registro
- Selección de medidas de salvaguarda para cubrir los requerimientos de seguridad del sistema de registros
- Asignación de responsabilidades para la gestión del registro
- Alternativas de almacenamiento para los registros del sistemas y sus características de rendimiento, escalabilidad, confidencialidad, integridad y disponibilidad
- Guía para la selección del sistema de almacenamiento y custodia de registros

7. **Administración del control de accesos adecuados de los sistemas de información**

- Análisis de los requerimientos de acceso de los distintos sistemas de información y recursos compartidos
- Principios comúnmente aceptados para el control de accesos y de los distintos tipos de acceso locales y remotos
- Requerimientos legales en referencia al control de accesos y asignación de privilegios
- Perfiles de de acceso en relación con los roles funcionales del personal de la organización
- Herramientas de directorio activo y servidores LDAP en general
- Herramientas de sistemas de gestión de identidades y autorizaciones (IAM)
- Herramientas de Sistemas de punto único de autenticación Single Sign On (SSO)

Orientaciones metodológicas

Formación a distancia:

Módulo formativo	Número de horas totales del módulo	N.º de horas máximas susceptibles de formación a distancia
Módulo formativo - MF0490_3	90	40

Criterios de acceso para los alumnos

Serán los establecidos en el Real Decreto que regula el certificado de profesionalidad de la familia profesional al que acompaña este anexo

MÓDULO FORMATIVO 2

Denominación: DESARROLLO DE ELEMENTOS SOFTWARE PARA GESTIÓN DE SISTEMAS

Código: MF0964_3

Nivel de cualificación profesional: 3

Asociado a la Unidad de Competencia:

UC0964_3 Crear elementos software para la gestión del sistema y sus recursos

Duración: 210 horas

UNIDAD FORMATIVA 1

Denominación: DESARROLLO Y OPTIMIZACIÓN DE COMPONENTES SOFTWARE PARA TAREAS ADMINISTRATIVAS DE SISTEMAS

Código: UF1286

Duración: 90 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP1 y RP2.

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 especificados.

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

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

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

- Las llamadas al sistema (System Calls):
 - o Definición
 - o Uso directo y mediante Application Programming Interfaces (APIs)
 - o Principales tipos de llamadas al sistema:
 - Control de procesos
 - Gestión de ficheros
 - Gestión de dispositivos
 - Información del sistema
 - Comunicaciones
 - o Descripción y uso de las APIs estándar de uso común para llamadas a sistema:
 - Win32 API (Sistemas Windows)
 - POSIX API (Sistemas Unix, Linux, Mac)
 - Java API (Multiplataforma)
- Programas de utilidades y comandos del sistema:
 - o Principales tipos:
 - Operaciones con ficheros y directorios
 - Funciones de estado

- Edición y manipulación de ficheros
- Soporte para lenguajes de programación (compiladores, enlazadores, ensambladores, intérpretes, etc.)
- Ejecución de programas
- Comunicaciones, mensajería, intercambio remoto de archivos, etc.
- o Uso de utilidades y comandos mediante lenguajes de script de uso común
 - Windows scripting
 - Linux/Unix scripting

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

- Modelos del ciclo de vida del software.
 - o En cascada (waterfall)
 - o Iterativo
 - o Incremental
 - o En V
 - o Basado en componentes (CBSE)
 - o Desarrollo rápido (RAD)
 - o Ventajas e inconvenientes. Pautas para la selección de la metodología más adecuada.
- Descripción de las fases en el ciclo de vida del software:
 - o 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
 - o 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
 - o 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
 - o 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:
 - o Principios de calidad del software
 - o 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
 - o 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:
 - o Identificación de los componentes necesarios según las especificaciones
 - o Análisis de los componentes reutilizables
 - o Análisis de la integración de los componentes en la arquitectura del sistema
 - o 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:
 - o 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
 - o 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:
 - o Reutilización de código.
 - Uso de librerías del sistema
 - Llamadas a utilidades y aplicaciones del sistema
 - o 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.
 - o Técnicas de optimización
- Control de calidad del desarrollo del software de gestión de sistemas:
 - o Métricas aplicables
 - o Verificación de requisitos
 - o Proceso de mejora continua
- Herramientas de uso común para el desarrollo de software de sistemas:
 - o Editores orientados a lenguajes de programación
 - o Compiladores y enlazadores
 - o 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

UNIDAD FORMATIVA 2

Denominación: DESARROLLO DE COMPONENTES SOFTWARE PARA EL MANEJO DE DISPOSITIVOS (DRIVERS)

Código: UF1287

Duración: 60 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP3.

Capacidades y criterios de evaluación

C1: Utilizar las técnicas y estándares utilizadas en el desarrollo, distribución e implantación de manejadores de dispositivos (drivers), para la integración de periféricos en el sistema informático según especificaciones técnicas y funcionales dadas.

CE1.1 Interpretar la documentación técnica de las herramientas software a utilizar y del sistema operativo donde se implementará el manejador de dispositivo, incluso si está editada en la lengua extranjera de uso más frecuente en el sector, utilizándola de ayuda, para identificar las características y los parámetros para la programación del manejador de dispositivo de acuerdo al diseño suministrado.

CE1.2 Utilizar las técnicas, estándares y herramientas de programación para desarrollar el manejador de dispositivo y depurar los posibles errores en el código desarrollado, según especificaciones técnicas de las herramientas y necesidades funcionales dadas.

CE1.3 Realizar las pruebas del manejador del dispositivo elaborado en los posibles escenarios en los que puede ser implantado, para asegurar su funcionalidad y la ausencia de conflictos con el resto de los elementos del sistema según especificaciones técnicas y normativa de calidad dadas.

CE1.4 Confeccionar la documentación técnica y de usuario del manejador desarrollado según unos parámetros y una normativa dadas.

CE1.5 En un supuesto práctico para modificar un manejador de dispositivo en el que se va a realizar un cambio en su diseño, de acuerdo a unas especificaciones funcionales dadas:

- Modificar el código para incorporar los cambios necesarios siguiendo los criterios de calidad especificados.
- Comprobar, mediante la realización de pruebas, que la modificación ha sido incorporada con éxito y que no ha alterado la funcionalidad del resto de manejadores, en particular, y del sistema en general.

- Realizar la actualización de toda la documentación que se vea afectada por el cambio propuesto.
- Diseñar el manejador del dispositivo utilizando técnicas de desarrollo estructurado y los estándares adecuados para el tipo de dispositivo.
- Codificar manejadores de dispositivos utilizando la herramienta seleccionada, aplicando el diseño previamente obtenido y utilizando lenguajes estructurados.
- Implantar el manejador del dispositivo documentando el código generado, los resultados obtenidos y las incidencias detectados.

Contenidos

1. El núcleo del sistema operativo

- Arquitectura general del núcleo
- Subsistemas del núcleo:
 - o Gestión de procesos
 - o Gestión de memoria
 - o Sistemas de ficheros
 - o Control de dispositivos
 - o Comunicaciones
- Aspectos de seguridad sobre el desarrollo de elementos del núcleo
- Consideraciones sobre compatibilidad de versiones del núcleo

2. Programación de controladores de dispositivo

- Funcionamiento general de un controlador de dispositivo
- Principales tipos de controladores de dispositivo:
 - o Carácter
 - o Bloque
 - o Paquete
- Técnicas básicas de programación de controladores de dispositivos
 - o Estructuras básicas de datos de dispositivos
 - o Gestión de errores de dispositivos
 - o Gestión de memoria de dispositivos
 - o Control de interrupciones
 - o Gestión de puertos de entrada y salida
 - o Uso de Acceso directo a memoria (DMA) y buses
- Técnicas de depuración y prueba:
 - o Impresión de trazas
 - o Monitorización de errores
 - o Técnicas específicas de depuración de controladores en sistemas operativos de uso común:
 - Windows
 - Unix
 - o Aplicación de estándares de calidad del software al desarrollo de controladores de dispositivos
- Compilación y carga de controladores de dispositivos
- Distribución de controladores de dispositivo
- Particularidades en el desarrollo de dispositivos en sistemas operativos de uso común:
 - o Sistemas Windows
 - o Sistemas Unix
 - o Modos de instalación de controladores de dispositivo en sistemas operativos de uso común. Dispositivos Plug & Play:
 - Instalación de dispositivos en Windows
 - Instalación de dispositivos en Sistemas Unix

- Herramientas:
 - o Entornos de desarrollo de controladores de dispositivo en sistemas operativos de uso común
 - o Herramientas de depuración y verificación de controladores de dispositivos
- Documentación de manejadores de dispositivo:
 - o Elaboración de especificaciones técnicas siguiendo directrices específicas de sistemas operativos de uso común
 - o Elaboración de manual de instalación
 - o Elaboración de manual de uso

UNIDAD FORMATIVA 3

Denominación: DESARROLLO DE COMPONENTES SOFTWARE PARA SERVICIOS DE COMUNICACIONES

Código: UF1288

Duración: 60 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP4.

Capacidades y criterios de evaluación

C1: Implementar servicios de comunicaciones entre sistemas aplicando las técnicas y estándares de desarrollo de elementos software, de acuerdo a unas especificaciones técnicas y funcionales dadas.

CE1.1 Clasificar las arquitecturas de servicios de comunicaciones para distinguir servicios prestados en entornos cliente/servidor de entornos entre iguales (punto a punto).

CE1.2 Describir los protocolos y puertos utilizados para la comunicación entre sistemas, teniendo en cuenta el soporte que ofrecen a los servicios de comunicaciones.

CE1.3 Identificar las principales API's (Application Program Interface) y librerías y su uso para el desarrollo mediante programación estructurada.

CE1.4 Realizar la codificación del componente utilizando herramientas de programación y depuración adecuadas para optimizar la fase de desarrollo según unas especificaciones técnicas dadas.

CE1.5 Someter al componente a baterías de pruebas en réplicas de los posibles escenarios de su implantación posterior, para verificar la ausencia de conflictos y su integración con el resto de componentes del sistema, según unos criterios de calidad y seguridad dados.

CE1.6 Clasificar los estándares definidos para el desarrollo de servicios de comunicaciones entre sistemas según diferentes criterios: organizaciones de estandarización, tipos de servicios y protocolos soportados, entre otros.

CE1.7 Enumerar los principales problemas de seguridad en el ámbito de las comunicaciones y describir las estrategias a aplicar, para el desarrollo de componentes que implementen servicios seguros según estándares y especificaciones dadas.

CE1.8 En varios supuestos prácticos donde se van a desarrollar componentes para el establecimiento de servicios de comunicaciones entre sistemas, dadas unas especificaciones técnicas:

- Seleccionar la herramienta adecuada para el desarrollo de los componentes de comunicaciones.

- Diseñar el componente utilizando técnicas de desarrollo estructurado y los estándares definidos.
- Codificar el elemento software utilizando la herramienta seleccionada, aplicando el diseño previamente obtenido y utilizando lenguajes estructurados.
- Depurar y probar el componente garantizando su óptimo funcionamiento.
- Diseñar baterías de posibles ataques contra el servicio y probarlas para detectar posibles vulnerabilidades.
- Implantar los componentes para verificar el servicio de comunicaciones documentando los resultados e incidencias detectados.
- Documentar el código desarrollado, las pruebas realizadas y el resultado de los procesos de implantación de los componentes.

Contenidos

1. Programación concurrente

- Programación de procesos e hilos de ejecución:
 - o Gestión de procesos
 - o Hilos y sincronización
- Programación de eventos asíncronos:
 - o Señales
 - o Temporizadores
- Mecanismos de comunicación entre procesos:
 - o Tuberías (pipes)
 - o Semáforos
 - o Compartición de memoria
 - o Mensajes
- Sincronización:
 - o Funciones de sincronización entre hilos
 - o Problemas de sincronización. Bloqueos (Deadlocks)
- Acceso a dispositivos:
 - o Funciones de lectura y escritura
 - o Puertos de entrada y salida

2. Fundamentos de comunicaciones

- Modelos de programación en red:
 - o El modelo cliente/servidor
 - o El modelo de objetos distribuidos
 - o Modelos basados en mensajes. Introducción a los Servicios web
- El nivel físico:
 - o Dispositivos físicos
 - o Protocolos de nivel físico
- El nivel de enlace:
 - o Redes Ethernet
 - o Direcciones físicas
- El nivel de transporte:
 - o El protocolo TCP/IP
 - o Esquemas de direccionamiento
 - o El nivel de transporte. Protocolos TCP y UDP. Otros protocolos de uso común.
 - o Puertos
 - o Servicios de red básicos

3. Programación de servicios de comunicaciones

- Aplicaciones y utilidades de comunicaciones. Estándares de comunicaciones:
 - o Organismos de estandarización de comunicaciones

- Comunicaciones en sistemas operativos de uso común
- Tipos de servicios de comunicaciones
- Protocolos de comunicaciones de uso común
- Estándares de comunicaciones inalámbricas
- Librerías de comunicaciones de uso común:
 - APIs para entornos Windows
 - APIs para entornos Unix
- Programación de componentes de comunicaciones:
 - Programación de sockets:
 - Funciones básicas
 - Ejemplos de utilización. Sockets TCP y UDP
 - Programación cliente/servidor mediante sockets
 - Programación de manejadores de protocolos
- Técnicas de depuración de servicios de comunicaciones:
 - Directrices para el diseño de pruebas
 - Exploración de vulnerabilidades y puertos
 - Revisión de logs
 - Otras técnicas de depuración
 - Herramientas de prueba y depuración de servicios de comunicaciones
- Rendimiento en las comunicaciones:
 - Calidad de servicio IP
 - Control del ancho de banda
 - Herramientas de monitorización de redes

4. Seguridad en las comunicaciones

- Principios de seguridad en las comunicaciones:
 - Mecanismos de seguridad
 - Principales vulnerabilidades y amenazas
- Herramientas para la gestión de la seguridad en red. Scanners
- Seguridad IP
- Seguridad en el nivel de aplicación. El protocolo SSL
- Seguridad en redes inalámbricas

Orientaciones metodológicas

Formación a distancia:

Unidades formativas	Duración total en horas de las unidades formativas	N.º de horas máximas susceptibles de formación a distancia
Unidad formativa 1 – UF1286	90	40
Unidad formativa 2 – UF1287	60	30
Unidad formativa 3 – UF1288	60	30

Secuencia:

Para acceder a las Unidades formativas 2 y 3 debe haberse superado la Unidad formativa 1

Criterios de acceso para los alumnos

Serán los establecidos en el artículo cuatro del Real Decreto que regula el certificado de profesionalidad de la familia profesional al que acompaña este anexo.

MÓDULO FORMATIVO 3

Denominación: DESARROLLO DE SOFTWARE BASADO EN TECNOLOGÍAS ORIENTADAS A COMPONENTES

Código: MF0965_3

Nivel de cualificación profesional: 3

Asociado a la Unidad de Competencia:

UC0965_3 Desarrollar elementos software con tecnologías de programación basada en componentes

Duración: 210 horas

UNIDAD FORMATIVA 1

Denominación: DISEÑO DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES

Código: UF1289

Duración: 90 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP1.

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:
 - o Ocultación de información (information hiding)
 - o El tipo abstracto de datos (ADT). Encapsulado de datos.
 - o Paso de mensajes
- Conceptos básicos de orientación a objetos:
 - o Clases:
 - Atributos, variables de estado y variables de clase
 - Métodos. Requisitos e invariantes.
 - Gestión de excepciones
 - Agregación de clases
 - o 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)
 - o 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
 - o Modularidad:
 - Librerías de clases. Ámbito de utilización de nombres
 - Ventajas de la utilización de módulos o paquetes
 - o Genericidad y sobrecarga:
 - Concepto de genericidad
 - Concepto de Sobrecarga. Tipos de sobrecarga
 - Comparación entre genericidad y sobrecarga
- Desarrollo orientado a objetos:
 - o Lenguajes de desarrollo orientado a objetos de uso común
 - o Herramientas de desarrollo
- Lenguajes de modelización en el desarrollo orientado a objetos:
 - o El lenguaje unificado de modelado (UML)
 - o Diagramas para la modelización de sistemas orientados a objetos

2. La orientación a componentes

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

UNIDAD FORMATIVA 2

Denominación: IMPLEMENTACIÓN E INTEGRACIÓN DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES

Código: UF1290

Duración: 90 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP2.

Capacidades y criterios de evaluación

C1: Construir elementos software a partir de las especificaciones de necesidades y con las condiciones de desarrollo de la tecnología de componentes utilizada.

CE1.1 Explicar los enfoques de desarrollo e implementación para la creación de componentes aplicando el principio de reutilización.

CE1.2 Describir el proceso de adaptación de un componente existente para incluirlo en la arquitectura en la que se quiere reutilizar, según especificaciones técnicas de la tecnología de componentes utilizada.

CE1.3 Enunciar las características del proceso de diseño de un nuevo componente para incluirlo en la arquitectura en la que se quiere utilizar, garantizando su futura reutilización.

CE1.4 Clasificar las herramientas de programación y depuración para optimizar la fase de desarrollo de los componentes según unas especificaciones dadas.

CE1.5 Identificar los elementos y parámetros de la interfaz del componente, para su desarrollo con herramientas y lenguajes específicos, para implementar la vía de comunicaciones con el resto de componentes según los estándares de definición de interfaces de la arquitectura.

CE1.6 Realizar los procesos de instalación del componente, comprobando que ejecuta las acciones requeridas y su disponibilidad para las aplicaciones que lo invoquen, según especificaciones técnicas de la arquitectura.

CE1.7 En un caso práctico para desarrollar componentes software reutilizables, dadas unas especificaciones funcionales y técnicas:

- Diseñar el nuevo componente para que cumpla las especificaciones funcionales dadas.
- Comprobar que la funcionalidad del componente diseñado puede ser extendida para futuras reutilizaciones.
- Implementar el componente utilizando herramientas y lenguajes orientados a objeto.
- Depurar y probar el componente desarrollado utilizando las herramientas disponibles.
- Documentar el componente y sus interfaces para facilitar su futura reutilización.

CE1.8 En un caso práctico para desarrollar software reutilizando componentes, dadas unas especificaciones funcionales y técnicas:

- Utilizar repositorios de componentes para localizar aquellos que se ajusten a las especificaciones dadas y puedan ser reutilizados.
- Diseñar las modificaciones que se van a realizar sobre el componente existente para que cumpla las especificaciones dadas.
- Modificar el componente utilizando herramientas y lenguajes orientados a objeto.
- Depurar y probar el componente modificado utilizando las herramientas disponibles.
- Documentar las modificaciones realizadas sobre el componente y sus interfaces para facilitar su futura reutilización.

CE1.9 Interpretar la documentación técnica asociada a las herramientas 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. Desarrollo de componentes

- Lenguajes de desarrollo de componentes.:
 - o Comparativa con lenguajes orientados a objetos
 - o Lenguajes orientados a componentes:
 - Descripción de interfaces
 - Ensamblado
 - Descripción de arquitectura
- Requisitos principales del desarrollo orientado a componentes:
 - Modularidad
 - Despliegue independiente
 - Reemplazabilidad
 - Seguridad
 - Separación entre interfaz e implementación
- Infraestructuras (frameworks) de componentes:
 - o Modelos de infraestructuras de componentes:
 - Orientados a conexión
 - Orientados a contexto
 - Orientados a aspectos
 - o Descripción de las infraestructuras de componentes de uso común:
 - OMG: CORBA, OMA
 - Java: JavaBeans, EJBs
 - Microsoft: COM, OLE/ActiveX, .NET
- Métodos de desarrollo de componentes:
 - o Uso de lenguajes orientados a objetos
 - o Selección de infraestructuras de componentes
- Construcción de software mediante componentes:
 - Definición de interfaces. Lenguajes de descripción de interfaces
 - Reutilización de componentes.
 - Técnicas de ensamblado en infraestructuras de uso común
- Técnicas específicas de desarrollo:
 - o Componentes en la capa de servidor web. Páginas dinámicas
 - o Componentes en la capa de servidor de aplicaciones.
 - o Componentes en la capa de aplicación cliente:
 - Componentes de interfaz gráfico
 - Componentes orientados a documento
 - o Componentes en la capa de servicios web
 - o Componentes para dispositivos móviles
- Herramientas para el desarrollo de componentes:
 - o Entornos integrados de desarrollo de componentes
 - o Configuración e instalación de herramientas de uso común:
 - Entorno Java
 - Entorno .NET
 - o Gestión del ciclo de vida en el desarrollo de componentes mediante herramientas de uso común:
 - Uso de repositorios de componentes. Registro de componentes
 - Reutilización de componentes para la construcción de sistemas software
 - Definición de metadatos de componente. Descriptores de interfaces
 - Modelo de seguridad

- Instalación de componentes
- Depuración y prueba de componentes

2. Componentes distribuidos

- o Programación distribuida en infraestructuras de uso común:
 - Programación multihilo (multithreading)
 - Comunicaciones síncronas y asíncronas
- o Modelos de intercambio:
 - Llamadas a procedimientos remotos
 - Orientados a mensajes
 - Orientados a recursos

UNIDAD FORMATIVA 3

Denominación: DESPLIEGUE Y PUESTA EN FUNCIONAMIENTO DE COMPONENTES SOFTWARE

Código: UF1291

Duración: 30 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP3.

Capacidades y criterios de evaluación

C1: Aplicar los procedimientos de despliegue e integración del componente en un entorno de tecnología orientada a componentes, según especificaciones técnicas de la arquitectura.

CE1.1 Describir las técnicas disponibles para realizar el proceso de búsqueda de componentes, que satisfagan los requisitos impuestos en el diseño inicial.

CE1.2 Clasificar los métodos de evaluación y selección de componentes, basándose en una serie de requisitos impuestos por las especificaciones iniciales de diseño.

CE1.3 Realizar las pruebas estructurales para verificar que el componente seleccionado se comunica con el resto de componentes y que no produce conflictos, según criterios de calidad y seguridad dados, y del diseño preeliminar.

CE1.4 Definir los procedimientos para el despliegue y adaptación para realizar la implantación del elemento software, según requisitos del componente desarrollado y siguiendo criterios de calidad, seguridad y especificaciones de la arquitectura.

CE1.5 Describir los procesos de integración, configuración e interconexión de los componentes seleccionados, para que permitan construir la aplicación final según los criterios de calidad y seguridad especificados en el diseño inicial.

CE1.6 Monitorizar el rendimiento de los componentes desarrollados o seleccionados para asegurar su integración en el sistema, de acuerdo a criterios de calidad y seguridad.

CE1.7 Realizar la documentación del despliegue y la implantación siguiendo los patrones, normativa y procedimientos especificados.

CE1.8 En un supuesto práctico debidamente caracterizado, utilizar las herramientas de desarrollo para realizar el desarrollo y despliegue de un componente software según unas especificaciones funcionales y técnicas dadas:

- Extraer la información relativa al diseño de un componente.
- Codificar el componente según especificaciones funcionales y técnicas.
- Realizar las pruebas estructurales según normativa y criterios de calidad establecidos.

- Incorporar el despliegue, adaptación, configuración e integración del componente según especificaciones técnicas y de implantación del desarrollo.
- Trazar las pruebas del componente según normativas de calidad y seguridad dadas.
- Configurar la herramienta para la realización de baterías de pruebas automáticas según normativa y criterios de calidad dados.
- Elaborar documentación mediante las plantillas facilitadas o incorporadas en la propia herramienta.
- Generar informes de calidad y métricas, e interpretar los resultados.

Contenidos

1. Despliegue de componentes

- Modelos de despliegue:
 - o Diseño sin repositorio:
 - Diseño y ejecución sin despliegue
 - Ejemplos: UML
 - o Diseño con repositorio sólo para el depósito de componentes:
 - Tipos de contenedores
 - Ejemplos: EJBs, .NET, CCM, Servicios web
 - o Despliegue con repositorio:
 - Composición y depósito de componentes
 - Ejemplo: JavaBean
 - o Diseño con repositorio:
 - Tipos de conectores
 - Ejemplos: Koala

2. Selección de componentes

- Tipos:
 - o Componentes comerciales:
 - Sin posibilidad de modificaciones (COTS)
 - Con posibilidad de adaptaciones (MOTS)
 - o Componentes de fuente abierta
 - o Ventajas e inconvenientes
- Métodos de personalización de componentes:
 - o Parametrización
 - o Uso de extensiones (plugins)
- Criterios de selección de componentes reutilizables:
 - o Adaptabilidad
 - o Auditabilidad
 - o Estandarización
 - o Características de concurrencia
 - o Rendimiento
 - o Consumo de recursos
 - o Seguridad
 - o Características de mantenimiento y actualización
- Proceso de selección de componentes:
 - o Evaluación de componentes según requisitos
 - o Diseño y codificación (código de enlace):
 - Enlace de componentes con otros sistemas
 - Integración
 - Configuración
 - o Diseño de pruebas
 - o Detección de fallos

- Mantenimiento y gestión de configuraciones
- Actualización de componentes
- Métodos de selección de uso común:
 - CAP (COTS Acquisition Process)
 - RUP (Rational Unified Process)

3. Control de calidad de componentes

- Métodos de evaluación de calidad de componentes. Estándares de calidad
- Categorías y métricas de evaluación
- Proceso de validación y medición de calidad:
 - Pruebas de conformidad a requisitos funcionales
 - Pruebas de integración con otros sistemas
 - Pruebas de aspectos no funcionales:
 - Rendimiento
 - Seguridad
 - Integración
- Documentación de componentes
- Descripción funcional
- Descripción de aspectos no funcionales
- Descripción del proceso de instalación y despliegue:
 - Descripción del empaquetamiento (packaging)
 - Requisitos de implantación
 - Parametrización y ajuste

Orientaciones metodológicas

Formación a distancia:

Unidades formativas	Duración total en horas de las unidades formativas	N.º de horas máximas susceptibles de formación a distancia
Unidad formativa 1 – UF1289	90	40
Unidad formativa 2 – UF1290	90	40
Unidad formativa 3 – UF1291	30	20

Secuencia:

Para acceder a las Unidad formativa 2 debe haberse superado la Unidad formativa 1

Criterios de acceso para los alumnos

Serán los establecidos en el artículo 4 del Real Decreto que regula en certificado de profesionalidad de la familia profesional al que acompaña este anexo.

MÓDULO DE PRÁCTICAS PROFESIONALES NO LABORALES DE PROGRAMACIÓN DE SISTEMAS INFORMÁTICOS

Código: MP0274

Duración: 80 horas

Capacidades y criterios de evaluación

C1: Colaborar en el desarrollo y análisis de sistemas informáticos.

CE1.1 Analizar los requisitos de desarrollo de los sistemas corporativos.

CE1.2 Evaluar el análisis y diseño de los sistemas conforme a los requisitos establecidos.

CE1.3 Verificar el uso y reutilización de componentes software en la realización de los sistemas corporativos.

CE1.4 Apoyar en el desarrollo y mantenimiento de los sistemas software

CE1.5 Colaborar en la realización de documentación del software

CE1.6 Investigar nuevas herramientas o actualizaciones de las existentes para mejorar la productividad en el desarrollo

CE1.7 Facilitar la coordinación entre los grupos de diseño, desarrollo, instalación y despliegue de software

C2: Auditar la calidad y seguridad de los sistemas software

CE2.1 Clasificar los sistemas software según su criticidad y valor para la empresa

CE2.2 Proporcionar apoyo en la realización de auditorías de software para la verificación y mejora de de la calidad y seguridad del software

CE2.3 Realizar pruebas según normativa y criterios de calidad establecidos en la empresa

CE2.4 Proporcionar asistencia en la aplicación de las medidas de mejora de la calidad y seguridad del software corporativo

C3: Participar en los procesos de trabajo de la empresa, siguiendo las normas e instrucciones establecidas en el centro de trabajo.

CE3.1 Comportarse responsablemente tanto en las relaciones humanas como en los trabajos a realizar.

CE3.2 Respetar los procedimientos y normas del centro de trabajo.

CE3.3 Empezar con diligencia las tareas según las instrucciones recibidas, tratando de que se adecuen al ritmo de trabajo de la empresa.

CE3.4 Integrarse en los procesos de producción del centro de trabajo.

CE3.5 Utilizar los canales de comunicación establecidos.

CE3.6 Respetar en todo momento las medidas de prevención de riesgos, salud laboral y protección del medio ambiente.

Contenidos

1. Análisis del ciclo de vida de software

- Definición de requisitos funcionales
- Metodologías de diseño
- Lenguajes y herramientas de desarrollo corporativos
- Procedimientos de depuración y prueba de software
- Procesos de instalación y despliegue
- Normalización y reutilización de componentes corporativos
- Criterios para la actualización del software
- Normas de documentación

2. Desarrollo y reutilización de componentes corporativos

- Utilización de herramientas de desarrollo en las distintas fases del ciclo de vida software según las normas corporativas
- Realización de diagramas de diseño
- Validación de requisitos
- Coordinación entre diferentes equipos o programadores para la construcción de sistemas software
- Realización de pruebas y validación de requisitos

3. Auditorías de calidad y seguridad

- Aplicación de la normativa de calidad
- Realización de planes de auditoría
- Revisión de la seguridad del software
- Análisis del rendimiento del software
- Evaluación del nivel de integración y optimización en la construcción del software
- Identificación de aspectos de mejora
- Realización de informes

4. Integración y comunicación en el centro de trabajo

- Comportamiento responsable en el centro de trabajo.
- Respeto a los procedimientos y normas del centro de trabajo.
- Interpretación y ejecución con diligencia las instrucciones recibidas.
- Reconocimiento del proceso productivo de la organización.
- Utilización de los canales de comunicación establecidos en el centro de trabajo.
- Adecuación al ritmo de trabajo de la empresa.
- Seguimiento de las normativas de prevención de riesgos, salud laboral y protección del medio ambiente.

IV. PRESCRIPCIONES DE LOS FORMADORES

Módulos Formativos	Acreditación requerida	Experiencia profesional requerida en el ámbito de la unidad de competencia	
		Si se cuenta con acreditación	Si no se cuenta con acreditación
MF0490_3: Gestionar servicios en el sistema informático	<ul style="list-style-type: none"> • Licenciado, Ingeniero, Arquitecto o el título de grado correspondiente u otros títulos equivalentes. • Diplomado, Ingeniero Técnico, Arquitecto Técnico o el título de grado correspondiente u otros títulos equivalentes. 	2 años	4 años
MF0964_3: Desarrollo de elementos software para gestión de sistemas	<ul style="list-style-type: none"> • Licenciado, Ingeniero, Arquitecto o el título de grado correspondiente u otros títulos equivalentes. • Diplomado, Ingeniero Técnico, Arquitecto Técnico o el título de grado correspondiente u otros títulos equivalentes. 	1 año	3 años
MF0965_3: Desarrollo de software basado en tecnologías orientadas a componentes	<ul style="list-style-type: none"> • Licenciado, Ingeniero, Arquitecto o el título de grado correspondiente u otros títulos equivalentes. • Diplomado, Ingeniero Técnico, Arquitecto Técnico o el título de grado correspondiente u otros títulos equivalentes. 	1 año	3 años

V. REQUISITOS MÍNIMOS DE ESPACIOS, INSTALACIONES Y EQUIPAMIENTO

Espacio Formativo	Superficie m ² 15 alumnos	Superficie m ² 25 alumnos
Aula de gestión	45	60

Espacio Formativo	M1	M2	M3
Aula de gestión	X	X	X

Espacio Formativo	Equipamiento
Aula de gestión	<ul style="list-style-type: none"> - Equipos audiovisuales - PCs instalados en red, con conexión a Internet - PC con funciones de servidor - Cañón de proyección - Software específico de la especialidad - Pizarras para escribir con rotulador - Rotafolios - Material de aula - Mesa y silla para formador - Mesas y sillas para alumnos

No debe interpretarse que los diversos espacios formativos identificados deban diferenciarse necesariamente mediante cerramientos.

Las instalaciones y equipamientos deberán cumplir con la normativa industrial e higiénico sanitaria correspondiente y responderán a medidas de accesibilidad universal y seguridad de los participantes.

El número de unidades que se deben disponer de los utensilios, máquinas y herramientas que se especifican en el equipamiento de los espacios formativos, será el suficiente para un mínimo de 15 alumnos y deberá incrementarse, en su caso, para atender a número superior.

En el caso de que la formación se dirija a personas con discapacidad se realizarán las adaptaciones y los ajustes razonables para asegurar su participación en condiciones de igualdad.