

## CUALIFICACIÓN PROFESIONAL:

### Programación de sistemas informáticos

Familia Profesional:	<b>Informática y Comunicaciones</b>
Nivel:	<b>3</b>
Código:	<b>IFC303_3</b>
Estado:	<b>BOE</b>
Publicación:	<b>RD 545/2023</b>
Referencia Normativa:	<b>RD 616/2020, Orden PRE/1636/2015, RD 1201/2007</b>

### Competencia general

Desarrollar componentes 'software' para la gestión, administración y supervisión de recursos en sistemas informáticos, partiendo de especificaciones, integrando los productos resultantes en librerías del sistema, gestionando los cambios y versiones, cumpliendo la normativa aplicable de protección de datos, seguridad, propiedad intelectual e industrial y la planificación de la actividad preventiva y los estándares de calidad.

### Unidades de competencia

- UC0490\_3:** GESTIONAR SERVICIOS EN EL SISTEMA INFORMÁTICO
- UC0964\_3:** Crear elementos 'software' para la gestión del sistema y sus recursos
- UC0227\_3:** DESARROLLAR COMPONENTES 'SOFTWARE' EN LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS
- UC0965\_3:** Desarrollar elementos 'software' con métodos y tecnologías orientados a componentes

### Entorno Profesional

#### Ámbito Profesional

Desarrolla su actividad profesional en el área de sistemas o de desarrollo dedicada a la programación y mantenimiento de sistemas informáticos, en entidades de naturaleza pública o privada, empresas de tamaño pequeño/mediano/grande o microempresas, tanto por cuenta propia como ajena, con independencia de su forma jurídica. Desarrolla su actividad dependiendo, en su caso, funcional y/o jerárquicamente de un superior. Puede tener personal a su cargo en ocasiones, por temporadas o de forma estable. En el desarrollo de la actividad profesional se aplican los principios de accesibilidad universal y diseño universal o diseño para todas las personas de acuerdo con la normativa aplicable.

#### Sectores Productivos

Se ubica sobre todo en el sector servicios, y principalmente en el subsector de la gestión de recursos informáticos o en cualquier otro sector productivo que utilice sistemas informáticos para su gestión formando parte del equipo de programación y mantenimiento.

#### Ocupaciones y puestos de trabajo relevantes

Los términos de la siguiente relación de ocupaciones y puestos de trabajo se utilizan con carácter genérico y omnicomprendivo de mujeres y hombres.

- Programadores de sistemas informáticos
- Programadores con lenguajes orientados a objetos

## Formación Asociada (630 horas)

### Módulos Formativos

- MF0490\_3:** GESTIÓN DE SERVICIOS EN EL SISTEMA INFORMÁTICO (90 horas)
- MF0964\_3:** Desarrollo de elementos 'software' para gestión de sistemas (180 horas)
- MF0227\_3:** PROGRAMACIÓN ORIENTADA A OBJETOS (240 horas)
- MF0965\_3:** Desarrollo de elementos 'software' con métodos y tecnologías orientadas a componentes (120 horas)

## UNIDAD DE COMPETENCIA 1

### GESTIONAR SERVICIOS EN EL SISTEMA INFORMÁTICO

Nivel: 3  
Código: UC0490\_3  
Estado: BOE

#### Realizaciones profesionales y criterios de realización

**RP1:** Gestionar la configuración del sistema para asegurar el rendimiento de los procesos según las necesidades de uso, considerando despliegues en arquitecturas dedicadas o distribuidas, con y sin virtualización y cumpliendo las directivas de la organización.

**CR1.1** Los procesos que intervienen en el sistema se identifican de forma que permitan evaluar parámetros de rendimiento, diferenciando los procesos que se encuentran repartidos en diferentes nodos, (si la arquitectura es distribuida) y/o si están asociados al 'software' de gestión de la virtualización, al hipervisor de los host físicos o a los propios servicios virtualizados (si se trata de un modelo virtualizado).

**CR1.2** Los parámetros que afectan a los componentes del sistema: memoria, procesador y periféricos, entre otros, se ajustan a las necesidades de uso asignándoles la configuración que maximice el rendimiento.

**CR1.3** Las prioridades de ejecución de los procesos se adecuan en función de las especificaciones del plan de explotación de la organización (tipo de proceso, usuario, perfil, entre otros).

**CR1.4** Las herramientas de monitorización se implantan, configurándolas y determinando los niveles de las alarmas.

**CR1.5** La conectividad y el ancho de banda que se necesita en arquitecturas distribuidas, se proporcionan según las especificaciones y/o manuales de fabricantes y de la organización.

**CR1.6** La distribución de la información en arquitecturas distribuidas se gestiona, siguiendo las especificaciones y/o manuales de fabricantes y de la organización, para maximizar el rendimiento del sistema.

**CR1.7** El 'software' de gestión de virtualización y el hipervisor, de los hosts físicos y los propios servicios virtualizados, en el caso de despliegues virtualizados, se gestiona, revisando la configuración y monitorizando el rendimiento, siguiendo las especificaciones y/o manuales de fabricantes y de la organización, y maximizando el rendimiento del sistema.

**RP2:** Administrar el almacenamiento según las necesidades de uso, considerando despliegues en arquitecturas dedicadas o distribuidas, con y sin virtualización y cumpliendo las directivas de la organización.

**CR2.1** Los dispositivos de almacenamiento se configuran para ser usados, asignando los parámetros propios del sistema operativo utilizado en el sistema informático.

**CR2.2** El almacenamiento se configura, teniendo en cuenta la posible necesidad de arquitecturas distribuidas que requieran distribución de la información, así como la necesidad de entornos virtualizados que requieren 'software' de gestión de virtualización, hipervisores y los propios servicios virtualizados.

**CR2.3** La estructura de almacenamiento se define, implantándose, atendiendo a las necesidades de los sistemas de archivos y a las especificaciones de uso de la organización.

**CR2.4** Los requerimientos de nomenclatura de objetos y restricciones de uso del almacenamiento se documentan, siguiendo el formato (tipo de documento, tamaño, maquetación, tipografía, entre otros) y otras indicaciones establecidas por la organización.

**CR2.5** El almacenamiento se integra para ofrecer un sistema funcional al usuario, siguiendo las especificaciones de la organización, con independencia del tipo de arquitectura (distribuida o dedicada) y de la existencia o no de capa de virtualización.

**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, asignando métodos de autenticación y perfiles, entre otros, para garantizar la seguridad e integridad del sistema.

**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 (dispositivos, espacio, número de conexiones, caudal/ancho de banda, entre otros) para los usuarios se limitan, usando las herramientas instaladas en el sistema, 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 servicios de comunicación se establecen con un sistema de calidad de servicio, garantizándose las comunicaciones de los mismos.

**CR4.2** Los dispositivos de comunicaciones se verifican en lo que respecta a su configuración y rendimiento, siguiendo las especificaciones de la organización.

**CR4.3** Los consumos de recursos de los servicios de comunicaciones se analizan, verificando que se encuentran dentro de los límites permitidos por las especificaciones.

**CR4.4** Las incidencias detectadas en los servicios de comunicaciones se documentan para informar a los responsables de la explotación del sistema y de la gestión de las mismas según los protocolos de la organización indicando, entre otros, el momento, la descripción y la solución aplicadas al problema.

## 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

Dispositivos de almacenamiento configurados y estructurados. Sistema configurado y operando. Rendimiento del sistema según los parámetros de explotación. Usuarios gestionados. Sistema seguro e íntegro en el acceso y utilización de servicios y recursos. Servicios de comunicaciones en funcionamiento.

### Información utilizada o generada

Normas externas de trabajo (normativa aplicable de protección de datos y publicación de la información). Normas internas de trabajo (plan de explotación de la organización; gráficas y análisis de rendimiento; listados de acceso y restricciones de usuarios; informe de incidencias; protocolo de actuación ante incidencias). Documentaciones técnicas (manuales de explotación del sistema operativo y de los dispositivos; manuales de las herramientas de monitorización utilizadas).

## UNIDAD DE COMPETENCIA 2

### Crear elementos 'software' para la gestión del sistema y sus recursos

Nivel: 3  
Código: UC0964\_3  
Estado: BOE

#### Realizaciones profesionales y criterios de realización

**RP1:** Preparar el entorno de desarrollo, instalando y configurando herramientas 'software' y de edición y depuración, seleccionando el lenguaje a utilizar para codificar tareas administrativas, según necesidades funcionales y metodología de trabajo especificadas por la entidad responsable de sistemas para el entorno o proyecto.

**CR1.1** Los recursos a utilizar en el desarrollo de los componentes 'software' a implementar tales como memoria, espacio de almacenamiento, entorno de desarrollo, entorno de pruebas, entre otros, se determinan planificando la ejecución de las tareas, analizando las especificaciones técnicas y las necesidades funcionales detectadas o reportadas por la clientela, usuarios del sistema y responsables de administración del mismo.

**CR1.2** Las herramientas a utilizar en la fase de desarrollo tales como edición y generación del código, creación de los módulos ejecutables, control de versiones, depuración y pruebas, documentación y empaquetado se seleccionan, previo análisis de sus características, para escoger las que se adaptan al desarrollo a realizar.

**CR1.3** El entorno de desarrollo se instala, previa selección del mismo, siguiendo las instrucciones proporcionadas por el fabricante, configurando elementos tales como licencias, paneles o vistas, complementos ('plugin') u otros, para dar soporte al desarrollo y depuración de los programas a desarrollar.

**RP2:** Desarrollar componentes 'software' que implementen servicios y herramientas de gestión del sistema, utilizando lenguajes o herramientas orientados a la programación de sistemas, para soportar tareas administrativas según necesidades funcionales y metodología de trabajo especificadas por la entidad responsable de sistemas para el entorno o proyecto.

**CR2.1** La diagramación y documentación previa al desarrollo del componente se elabora especificando los requisitos necesarios para su inclusión en librerías de modo que se facilite su reutilización.

**CR2.2** Los componentes 'software' de alto o bajo nivel para gestión del sistema tales como acceso a memoria, E/S, comunicaciones, almacenamiento, entre otros se codifican en función de las especificaciones técnicas del diseño, haciendo uso de herramientas de programación y depuración y reutilizando y/o integrando funciones de librería del sistema o servicios de la nube.

**CR2.3** Las herramientas de control de versiones centralizadas o distribuidas se utilizan, creando, bloqueando elementos para modificación, abriendo ramas ('fork' o 'branch') y publicando el código desarrollado o modificado para garantizar el control de cambios en el 'software', facilitando el desarrollo, la adición de nuevas funcionalidades y la corrección de errores.

**CR2.4** La documentación de los componentes 'software' de servicios y herramientas de gestión se elabora, incluyendo requerimientos e instrucciones de uso, siguiendo los formatos, normativa y procedimientos especificados por la entidad responsable de programación de sistemas.

**CR2.5** El componente desarrollado se publica en librerías del sistema una vez superadas las pruebas, incluyéndolo, identificándolo y catalogándolo, para facilitar la localización según necesidades de desarrollo y para su reutilización en el desarrollo de nuevos componentes.

**RP3:** Probar componentes 'software' desarrollados para gestión de sistemas, ejecutando el plan de pruebas para comprobar su funcionalidad, depurando los errores y documentando el proceso, utilizando lenguajes o herramientas de prueba y depuración, según estándares de desarrollo, cumpliendo la metodología de trabajo de la entidad responsable de sistemas para el entorno o proyecto y garantizando la autenticación, integridad, disponibilidad y confidencialidad.

**CR3.1** El plan de pruebas se elabora para confirmar la funcionalidad de los componentes desarrollados, incluyendo las personas responsables de su ejecución, la secuencia de pruebas que indique el proyecto y las acciones a realizar para cada prueba, según criterios de calidad.

**CR3.2** Las pruebas incluidas en el plan se ejecutan, documentando los resultados de las mismas incluyendo datos de la persona o entidad que la ejecuta, prueba realizada, resultado esperado y resultado obtenido, según el plan de pruebas.

**CR3.3** Los componentes 'software' se depuran analizando los resultados de las pruebas para corregir los errores, utilizando las herramientas que proporcione el entorno de programación según criterios de calidad.

## Contexto profesional

### Medios de producción

Equipos y/o dispositivos informáticos, periféricos y dispositivos conectables. 'Software' de base. Entornos integrados de desarrollo (IDE). Herramientas de depuración. Herramientas de control de versiones. Herramientas de prueba. Herramientas de despliegue de aplicaciones. Herramientas de documentación de elementos de programación. Herramientas ofimáticas. Lenguajes de desarrollo orientados a sistemas. Servicios en la nube.

### Productos y resultados

Componentes 'software' de servicios y herramientas de gestión del sistema operativo elaborados. Funciones de librería del sistema integradas en el 'software'. Componentes 'software' para manejo de dispositivos 'hardware' elaborados. Componentes 'software' de servicios de red elaborados.

### Información utilizada o generada

Normas externas de trabajo (normativa aplicable de protección de datos, normativa aplicable de seguridad informática y propiedad intelectual e industrial, normativa aplicable de planificación de la acción preventiva -ergonomía-). Normas internas de trabajo (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; normas corporativas de desarrollo de 'software', de pruebas, de control de calidad). Documentación técnica (manuales de funcionamiento del 'software'; manuales de interfaces de programación -API- del sistema operativo; manuales o ayudas de la herramienta de programación empleada; manuales o ayudas de uso del sistema operativo; manuales o ayudas del entorno de programación -IDE-; manuales del lenguaje de programación empleado; soportes

técnicos para asistencia -telefónica, Internet, mensajería y foros, entre otros-; manuales o ayudas de servicios en la nube).



## UNIDAD DE COMPETENCIA 3

### DESARROLLAR COMPONENTES 'SOFTWARE' EN LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS

Nivel: 3  
Código: UC0227\_3  
Estado: BOE

#### Realizaciones profesionales y criterios de realización

**RP1:** Implementar los elementos 'software' encomendados de modo que cumplan las especificaciones del diseño y los niveles de calidad establecidos por la organización para obtener los componentes orientados a objetos base de aplicaciones en diferentes tecnologías.

**CR1.1** Los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del 'software' se determinan, interpretando las especificaciones establecidas en el diseño.

**CR1.2** Las inconsistencias del diseño detectadas se registran para su transmisión a la persona responsable por los cauces establecidos por la organización para garantizar la trazabilidad.

**CR1.3** Las clases definidas, sus atributos, relaciones, métodos e interfaces se codifican utilizando el lenguaje de programación elegido, las técnicas metodológicas orientadas a objetos, las especificaciones recibidas, las normas de calidad y las pautas de programación de la organización.

**CR1.4** El control de errores se implementa en el código de acuerdo a las normas de calidad de la organización.

**CR1.5** Las herramientas de desarrollo y depuración se emplean para facilitar el proceso de generación del código, detectando y corrigiendo errores en el código.

**CR1.6** Los problemas encontrados en la implementación se resuelven, consultando la documentación técnica, editada en castellano, en las lenguas cooficiales de las Comunidades Autónomas o en el idioma extranjero de uso más frecuente en el sector.

**CR1.7** Los componentes de terceros se utilizan, seleccionándolos según las necesidades, interpretando su documentación e implementando su uso, para reutilizar código y reducir el tiempo de desarrollo.

**CR1.8** Los componentes orientados a objeto se utilizan como base en el desarrollo de aplicaciones en diferentes tecnologías como programación web, dispositivos móviles, servicios en la nube, aprendizaje automático, u otras.

**RP2:** Manipular bases de datos a través de interfaces para integrar el lenguaje de programación con el lenguaje de acceso a datos en la construcción de una aplicación.

**CR2.1** El conector de base de datos -con acceso directo, con mapeo objeto-relacional o el acceso directo a ficheros- se emplea en el programa, seleccionando una interfaz de programación de aplicaciones.

**CR2.2** El inicio, configuración y cierre de una sesión en la base de datos se lleva a cabo con la interfaz de programación o bien utilizar una sesión abierta del grupo -pool- disponible.

**CR2.3** Las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación se ejecutan por medio de la interfaz de programación, garantizando la seguridad de la base de datos.

**CR2.4** El interfaz de programación se utiliza para garantizar que los accesos a la base de datos finalizan las transacciones, asegurando su integridad y consistencia.

**RP3:** Probar los componentes 'software' desarrollados para asegurar que cumplen las especificaciones definidas en el proyecto, garantizando su funcionamiento en producción.

**CR3.1** El plan de pruebas se ejecuta, siguiendo los protocolos y los criterios de calidad definidos en la organización.

**CR3.2** Las pruebas automatizadas se integran, ejecutándolas dentro de los procesos de integración continua y control de calidad.

**CR3.3** Los datos de prueba de cada unidad funcional se concretan de forma que contemplen una variedad de posibilidades, incluyendo casos límite y conflictivos.

**CR3.4** Las pruebas de cada unidad funcional se detallan de forma que cubran los caminos básicos que pueden surgir en su ejecución.

**CR3.5** Los casos de prueba que involucran varios objetos se definen teniendo en cuenta los escenarios de interacción posibles para verificar que los objetos mantienen coherente su estado tras el intercambio de mensajes.

**CR3.6** Las pruebas funcionales de alto nivel se efectúan, siguiendo los procesos definidos tanto de ejecución como documentación de forma que aseguren el cumplimiento de las especificaciones.

**CR3.7** Las pruebas de puesta en producción se desarrollan para asegurar que los componentes soportan la carga de trabajo del entorno y su integración con servicios externos a los mismos - pasarelas de pago, servicios de internet u otros-.

**RP4:** Elaborar la documentación del código desarrollado según los estándares de la organización para garantizar su futuro mantenimiento.

**CR4.1** La documentación -manuales, guías entre otras- para desarrolladores, administradores o usuarios se redacta de acuerdo a las especificaciones del proyecto.

**CR4.2** La documentación correspondiente a cada componente se redacta de acuerdo con las normas y modelos incluidos en el diseño de la aplicación.

**CR4.3** Los comentarios, identificadores con nombres autoexplicativos -mnemotécnicos-, tabulaciones u otros, se añaden al código, documentándolo para la mejor comprensión del programa-'software'-.

**CR4.4** La documentación de un componente desarrollado se genera mediante las herramientas de producción automática de documentación a partir del texto incluido en el componente 'software'.

**CR4.5** Los procedimientos de ejecución de las pruebas unitarias y de regresión, así como los resultados de las mismas se documentan de acuerdo a los estándares definidos por la organización.

**RP5:** Gestionar los cambios en el código desarrollado para asegurar el cumplimiento de las normas de la organización y los niveles de calidad establecidos durante la fase de desarrollo y la de mantenimiento.

**CR5.1** Los cambios en la implementación como consecuencia de cambios en las especificaciones se codifican, adecuando la estructura del código y manteniendo su calidad.

**CR5.2** Las correcciones efectuadas como consecuencia de los errores o discrepancias encontrados en las pruebas se codifican en los componentes 'software' afectados.

**CR5.3** Las pruebas de regresión como consecuencia de cambios en las especificaciones se efectúan de acuerdo a los procedimientos establecidos para asegurar la funcionalidad previa.

**CR5.4** La documentación del código se mantiene coherente con sus sucesivas modificaciones.

## Contexto profesional

### Medios de producción

Equipos informáticos y periféricos. Sistemas operativos y parámetros de configuración. Herramientas ofimáticas. Paradigmas de programación. Lenguajes de programación orientados a objetos. Herramientas de control de cambios. Herramientas de compilación. Herramientas de depuración. Herramientas de prueba. Entornos integrados de desarrollo.

### Productos y resultados

Código fuente de la aplicación. Código ejecutable de la aplicación. Bases de datos manipuladas a través de interfaces. Componentes 'software' desarrollado y probado. Documentación del código desarrollado elaborada. Cambios en el código desarrollado gestionados.

### Información utilizada o generada

Normativa relativa a la planificación de la actividad preventiva. Normativa sobre prevención de riesgos laborales. Normas internas de trabajo, reglamentación y estándares de calidad. Manuales de uso y funcionamiento de los sistemas informáticos. Manuales de funcionamiento del 'software'. Material de cursos de formación. Manuales del lenguaje de programación empleado. Manuales de la herramienta de programación empleada. Ayuda en línea de bibliotecas y aplicaciones. Soportes técnicos de asistencia de telefonía, Internet, mensajería, foros u otros. Documentación del código desarrollado.

## UNIDAD DE COMPETENCIA 4

### Desarrollar elementos 'software' con métodos y tecnologías orientados a componentes

Nivel: 3  
Código: UC0965\_3  
Estado: BOE

#### Realizaciones profesionales y criterios de realización

**RP1:** Instalar herramientas de programación, previa selección en su caso, analizando las tecnologías disponibles, escogiendo aquellas que se ajusten a las especificaciones del producto o aplicación basada en componentes, configurándolas según los requisitos técnicos, para usarlas en su desarrollo posterior.

**CR1.1** La tecnología a utilizar se evalúa, analizando si permite o no, elementos tales como paletas, lienzo o contenedor, editores para configurar y especializar componentes, buscadores, repositorios de componentes, acceso a intérpretes, compiladores y depuradores, herramientas de gestión y control de proyectos, entre otros.

**CR1.2** El entorno de desarrollo se selecciona, escogiendo aquel que ofrezca, de entre todos los analizados, la mejor relación coste/beneficio en función de las facilidades que proporcionan y las limitaciones tecnológicas para el producto o aplicación a desarrollar.

**CR1.3** El entorno de desarrollo integrado (IDE), complementos adicionales y los 'framework' asociados a las tecnologías basadas en componentes se instalan, configurándolos de modo que permitan minimizar el tiempo de desarrollo y maximizar la seguridad y control de la aplicación o producto.

**CR1.4** La documentación de la instalación se elabora, siguiendo los modelos internos establecidos por la organización, recogiendo las configuraciones y/o acciones aplicadas y archivándola para su control, trazabilidad y uso posterior.

**RP2:** Desarrollar 'software' usando el paradigma de programación orientada a componentes, empleando un entorno de desarrollo específico de esta tecnología, para su integración en una arquitectura definida, según el diseño y especificaciones resultantes del análisis.

**CR2.1** El componente se codifica haciendo uso de los lenguajes soportados por la arquitectura utilizada, utilizando herramientas de programación y depuración.

**CR2.2** La interfaz del componente se programa con herramientas y lenguajes específicos, según los estándares de definición de interfaces de la arquitectura, para implementar la vía de comunicaciones con el resto de componentes.

**CR2.3** El componente se somete a pruebas de análisis del rendimiento, pruebas funcionales y pruebas de seguridad para comprobar la funcionalidad del sistema de seguridad, siguiendo los procedimientos establecidos por la organización.

**CR2.4** La documentación del desarrollo y pruebas se elabora, siguiendo los modelos internos establecidos por la organización, recogiendo las configuraciones y/o acciones aplicadas y archivándola para su control, trazabilidad y uso posterior.

**CR2.5** Las herramientas de control de versiones se utilizan para garantizar el control de cambios en el 'software', facilitando el desarrollo, las pruebas y el retorno a versiones previas de las modificaciones relativas a nuevas funcionalidades y corrección de errores.

**RP3:** Desplegar los componentes desarrollados, implantándolos para su integración en el sistema, según especificaciones técnicas y criterios de calidad y seguridad de la organización responsable del desarrollo.

**CR3.1** La estructura del componente se prueba, verificando que se comunica con el resto y que no produce conflictos.

**CR3.2** El despliegue se ejecuta, según requisitos del componente desarrollado, usando las facilidades y utilidades que proporcionen el entorno, para asegurar la implantación del mismo con la provisión de sus funcionalidades y la ausencia de conflictos.

**CR3.3** El rendimiento de los componentes desarrollados se monitoriza, usando las utilidades del entorno para asegurar y garantizar su integración en el sistema.

**CR3.4** La documentación del despliegue e implantación se elabora, siguiendo los modelos internos establecidos por la organización, recogiendo las configuraciones y/o acciones aplicadas y archivándola para su control, trazabilidad y uso posterior.

## Contexto profesional

### Medios de producción

Entornos integrados de desarrollo. Equipos informáticos y periféricos de comunicaciones. Herramientas de control de versiones. Herramientas de depuración. Herramientas de desarrollo o entornos integrados (IDE). 'Framework'. 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. Herramientas de diagramación y modelización. Compiladores e/o intérpretes de lenguajes utilizados. Servicios de transferencia de ficheros y mensajería.

### Productos y resultados

Herramienta de programación seleccionadas e instaladas. Componentes 'software' desarrollados. Componentes 'software' desplegados e integrados en el sistema.

### Información utilizada o generada

Normas externas de trabajo (normativa aplicable de seguridad, propiedad intelectual e industrial y de protección de datos). Normas internas de trabajo (documentación de instalación y configuración, Inventario de 'hardware'; plan sobre prevención de riesgos laborales -ergonomía-; normas corporativas de desarrollo de 'software', de pruebas, de control de calidad; manuales de uso y funcionamiento del 'software' desarrollado). Documentación técnica (Manuales de interfaces de programación -API- del sistema operativo; manuales del entorno de programación -IDE-, complementos y 'framework'; catálogos de productos 'software', proveedores y precios; manuales del lenguaje de programación; manuales técnicos de los dispositivos de comunicaciones sobre los que se vaya a programar).

## MÓDULO FORMATIVO 1

### GESTIÓN DE SERVICIOS EN EL SISTEMA INFORMÁTICO

Nivel:	3
Código:	MF0490_3
Asociado a la UC:	UC0490_3 - GESTIONAR SERVICIOS EN EL SISTEMA INFORMÁTICO
Duración (horas):	90
Estado:	BOE

#### Capacidades y criterios de evaluación

- C1:** Analizar procesos del sistema, asegurando un rendimiento acorde a los parámetros especificados en el plan de explotación considerando despliegues en arquitecturas dedicadas o distribuidas, con y sin capa de virtualización.
- CE1.1** Identificar procesos del sistema, analizando 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 explicando sus características y funciones.
  - CE1.3** Explicar técnicas de monitorización y herramientas destinadas a evaluar el rendimiento del sistema, indicando qué parámetros se miden y qué funciones se controlan.
  - CE1.4** En un supuesto práctico de análisis del rendimiento de un sistema informático con una carga de procesos concreta:
    - Utilizar herramientas del sistema, monitorizando sus parámetros para identificar cuantos procesos activos existen y las características particulares de alguno de ellos.
    - Realizar 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 del almacenamiento para ofrecer al usuario un sistema de registro de la información íntegro, confidencial y disponible.
- CE2.1** Identificar 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 un sistema de archivo, en función de la arquitectura 'hardware' (dedicada o distribuida), los dispositivos de almacenamiento y sistemas operativos empleados.
  - CE2.3** Describir la estructura general de almacenamiento asociando, para cada nodo o sistema informático final, los dispositivos con los sistemas de archivos existentes.
  - CE2.4** Describir la distribución del almacenamiento en nodos, dispositivos y sistemas de archivo, comprobando que se garantice la funcionalidad y el rendimiento del conjunto.
  - CE2.5** En un supuesto práctico de aplicación de procedimientos de administración de almacenamiento de la información con varios dispositivos:

- Particionar los dispositivos, en los casos que se requiera distribuir la información de manera separada, generando la infraestructura de los sistemas de archivo a instalar.
- Distribuir la información en diferentes nodos, integrándolos en un sistema de almacenamiento común, garantizando las comunicaciones y el rendimiento cuando la distribución del almacenamiento sea un requisito de implementación.
- Implementar la estructura general de almacenamiento, integrando todos los nodos, dispositivos y sus correspondientes sistemas de archivos.
- Documentar los requerimientos y restricciones de cada sistema de archivos implantado, indicando la restricción o el requerimiento y el tipo de dispositivo afectado.
- Aplicar los puntos anteriores sobre sistemas virtualizados.

**C3:** Administrar accesos al sistema y a los recursos para asegurarlos, restringiendo su uso en función del perfil de acceso.

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

**CE3.2** Describir 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 de administración del acceso al sistema en el que se cuenta con derecho de administración de usuarios:

- Identificar los posibles accesos de un usuario al sistema, monitorizando mediante visionado de 'log' o usando herramienta 'software'.
- Modificar los permisos de utilización de un recurso del sistema a un usuario, estableciendo otros que se hayan solicitado.
- Definir limitaciones de uso de un recurso del sistema a los usuarios, verificando dicha limitación simulando el acceso.

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

**CE4.1** Explicar parámetros de configuración y funcionamiento de los dispositivos de comunicaciones, indicando los servicios afectados por cada uno para asegurar su funcionalidad dentro del sistema.

**CE4.2** Relacionar servicios de comunicaciones activos en el sistema con los dispositivos utilizados por ellos, analizando y evaluando el rendimiento.

**CE4.3** En un supuesto práctico de evaluación de uso y rendimiento de un sistema informático conectado con el exterior por medio de varias líneas de comunicaciones:

- Identificar los dispositivos de comunicaciones, describiendo sus características.
- Verificar el estado de los servicios de comunicaciones, comprobando su funcionalidad.
- Evaluar el rendimiento de los servicios de comunicaciones, midiendo los parámetros de conectividad y caudal.
- Detectar las incidencias producidas en el sistema, documentando las que se produzcan.

## Capacidades cuya adquisición debe ser completada en un entorno real de trabajo

C1 respecto a CE1.4; C2 respecto a CE2.5; C3 respecto a CE3.3; C4 respecto a CE4.3.

### Otras Capacidades:

Mantener el área de trabajo con el grado apropiado de orden y limpieza.

Demostrar cierto grado de autonomía en la resolución de contingencias relacionadas con su actividad.

Demostrar creatividad en el desarrollo del trabajo que realiza.  
Interpretar y ejecutar instrucciones de trabajo.  
Demostrar resistencia al estrés, estabilidad de ánimo y control de impulsos.  
Valorar el talento y el rendimiento profesional con independencia del sexo.  
Mostrar una actitud de respeto hacia los compañeros, procedimientos y normas de la empresa.  
Cumplir las medidas que favorezcan el principio de igualdad de trato y de oportunidades entre hombres y mujeres.  
Aplicar de forma efectiva el principio de igualdad de trato y no discriminación en las condiciones de trabajo entre mujeres y hombres.

## Contenidos

### 1 Procesos en el sistema informático

Estados de un proceso.  
Manejo de señales entre procesos.  
Administración de procesos.  
Cambio de prioridades.  
Monitorización de procesos.  
Gestión del consumo de recursos.

### 2 Almacenamiento de información en la gestión de servicios

Dispositivos de almacenamiento.  
Sistemas de archivo.  
Estructura general de almacenamiento.  
Herramientas del sistema para gestión del almacenamiento.

### 3 Gestión de usuarios en la gestión de servicios

Acceso al sistema.  
Permisos y acceso a los recursos.  
Limitaciones de uso de recursos.

### 4 Servicios de comunicaciones en la gestión de servicios

Dispositivos de comunicaciones.  
Protocolos de comunicaciones.  
Servicios de comunicaciones.  
Rendimientos de los servicios de comunicaciones.

## Parámetros de contexto de la formación

### Espacios e instalaciones

Los talleres e instalaciones darán respuesta a las necesidades formativas de acuerdo con el contexto profesional establecido en la unidad de competencia asociada, teniendo en cuenta la normativa aplicable del sector productivo, prevención de riesgos laborales, accesibilidad universal, igualdad de género y protección medioambiental. Se considerará con carácter orientativo como espacios de uso:

- Instalación de 2 m<sup>2</sup> por alumno o alumna.

### Perfil profesional del formador o formadora:

1. Dominio de los conocimientos y las técnicas relacionados con la gestión de servicios en el sistema informático, que se acreditará mediante una de las dos formas siguientes:



- Formación académica de nivel 2 (Marco Español de Cualificaciones para la Educación Superior), Ingeniero Técnico, Diplomado o de otras de superior nivel relacionadas con el campo profesional.
  - Experiencia profesional de un mínimo de 2 años en el campo de las competencias relacionadas con este módulo formativo.
2. Competencia pedagógica acreditada de acuerdo con lo que establezcan las Administraciones competentes.

## MÓDULO FORMATIVO 2

### Desarrollo de elementos 'software' para gestión de sistemas

Nivel:	3
Código:	MF0964_3
Asociado a la UC:	UC0964_3 - Crear elementos 'software' para la gestión del sistema y sus recursos
Duración (horas):	180
Estado:	BOE

### Capacidades y criterios de evaluación

**C1:** Aplicar procedimientos de instalación y configuración de herramientas y entornos de desarrollo para programas de gestión de sistemas, seleccionando el lenguaje a utilizar, para codificar tareas administrativas.

**CE1.1** Describir arquitecturas de sistemas operativos tales como monolítico, modular, 'microkernel' y sistemas distribuidos entre otros, explicando sus características, para identificar las necesidades de gestión y administración.

**CE1.2** Describir los recursos del sistema tales como tipos de almacenamiento, memoria y servicios de comunicaciones, arquitecturas de entrada/salida, de buses y de microprocesadores, entre otros, explicando sus características, limitaciones y ventajas.

**CE1.3** Clasificar herramientas a utilizar en la fase de desarrollo tales como edición y generación del código, creación de los módulos ejecutables, control de versiones, depuración y pruebas, complementos ('plugin'), documentación y empaquetado, analizando sus características, para escoger las que se adaptan a cada desarrollo.

**CE1.4** Identificar los tipos de licencia de aplicaciones, clasificándolas y determinando los pasos a seguir para su habilitación e instalación.

**CE1.5** En un supuesto práctico de aplicación de procedimientos de instalación y configuración de entornos de desarrollo para programas de gestión de sistemas, seleccionando el lenguaje a utilizar para codificar tareas administrativas:

- Los recursos a utilizar en el desarrollo de los componentes 'software' a implementar tales como memoria, espacio de almacenamiento, entorno de desarrollo, entorno de pruebas, entre otros, planificando la ejecución de las tareas, analizando las especificaciones técnicas y las necesidades funcionales detectadas o reportadas por la clientela, usuarios del sistema y responsables de administración del mismo.

- Seleccionar herramientas a utilizar en la fase de desarrollo tales como edición y generación del código, creación de los módulos ejecutables, control de versiones, depuración y pruebas, documentación y empaquetado se seleccionan, previo análisis de sus características, para escoger las que se adaptan al desarrollo a realizar.

- Instalar un entorno desarrollo, previa selección del mismo, siguiendo las instrucciones proporcionadas por el fabricante, configurando elementos tales como paneles o vistas, complementos ('plugin') u otros, para dar soporte al desarrollo y depuración de los programas a desarrollar.

**C2:** Aplicar procedimientos de desarrollo de componentes 'software' que implementen servicios y herramientas de gestión del sistema, utilizando

lenguajes o herramientas orientados a la programación de sistemas, para soportar tareas administrativas.

**CE2.1** Explicar los apartados de un análisis de requisitos, detallando los diagramas utilizados para la especificación funcional y de datos según metodologías y estándares de diseño.

**CE2.2** Describir diagramas de especificación funcional y de datos previos a la fase de desarrollo, explicando sus componentes y características.

**CE2.3** Clasificar los lenguajes de programación orientados a la gestión de sistemas, en función del paradigma, modularidad, facilidad de desarrollo y consumo de recursos.

**CE2.4** Distinguir las funciones de las librerías del sistema para la elaboración de nuevos componentes 'software' tales como comunicaciones, memoria, drivers, E/S en general y almacenamiento, entre otros, reutilizando el código ya desarrollado.

**CE2.5** Describir como incorporar nuevos elementos en las librerías del sistema, según unos criterios de optimización y calidad especificados.

**CE2.6** En un supuesto práctico, de aplicación de procedimientos de desarrollo de componentes 'software' que implementen servicios y herramientas de gestión del sistema, utilizando lenguajes o herramientas orientados a la programación de sistemas, para soportar tareas administrativas:

- Definir los módulos 'software' a realizar a partir de las especificaciones técnicas y el 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 de datos y de control necesarias para representar la información especificada en los requisitos.
- Gestionar el desarrollo usando una herramienta de control de versiones, creando, bloqueando elementos para modificación, abriendo ramas ('fork' o 'branch') y publicando el código desarrollado o modificado para garantizar el control de cambios en el 'software', facilitando el desarrollo, la adición de nuevas funcionalidades y la corrección de errores.
- Elaborar la documentación de los componentes 'software', incluyendo requerimientos e instrucciones de uso, siguiendo un formato.
- Publicar el componente desarrollado en librerías del sistema una vez superadas las pruebas, incluyéndolo, identificándolo y catalogándolo, para facilitar la localización según necesidades de desarrollo y para su reutilización en el desarrollo de nuevos componentes.

**C3:** Aplicar procedimientos de prueba de componentes 'software' desarrollados para gestión de sistemas, ejecutando el plan de pruebas para comprobar su funcionalidad, depurando los errores y documentando el proceso, utilizando lenguajes o herramientas de prueba y depuración, según estándares de desarrollo y garantizando la autenticación, integridad, disponibilidad y confidencialidad.

**CE3.1** Explicar la estructura de un plan de pruebas, describiendo sus apartados y los procedimientos para aplicarlo.

**CE3.2** 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.

**CE3.3** En un supuesto práctico de diseño y codificación de elementos 'software' para la gestión de los recursos del sistema, dadas unas especificaciones técnicas:

- Elaborar un plan de pruebas para confirmar la funcionalidad de los componentes desarrollados, incluyendo hipotéticas personas responsables de su ejecución, la secuencia de pruebas que indique el proyecto y las acciones a realizar para cada prueba, según criterios de calidad.
- Ejecutar las pruebas incluidas en el plan, documentando los resultados de las mismas incluyendo datos de la persona o entidad que la ejecuta, prueba realizada, resultado esperado y resultado obtenido, según el plan de pruebas.
- Depurar los componentes 'software', analizando los resultados de las pruebas para corregir los errores, utilizando las herramientas que proporcione el entorno de programación según criterios de calidad.

## Capacidades cuya adquisición debe ser completada en un entorno real de trabajo

C1 respecto a CE1.5; C2 respecto a CE2.6 y C3 respecto a CE3.3.

### Otras Capacidades:

Demostrar creatividad en el desarrollo del trabajo que realiza.

Proponerse objetivos retadores que supongan un nivel de rendimiento y eficacia superior al alcanzado previamente.

Demostrar cierto grado de autonomía en la resolución de contingencias relacionadas con su actividad.

Emplear tiempo y esfuerzo en ampliar conocimientos e información complementaria para utilizarlos en su trabajo.

Aprender nuevos conceptos o procedimientos y aprovechar eficazmente la formación utilizando los conocimientos adquiridos.

Mostrar una actitud de respeto hacia los compañeros, procedimientos y normas de la empresa.

Cumplir las medidas que favorezcan el principio de igualdad de trato y de oportunidades entre hombres y mujeres.

## Contenidos

### 1 Arquitecturas de sistemas operativos para la programación de sistemas

Arquitecturas de sistemas operativos y sus características. Monolítico, modular, 'microkernel' y sistemas distribuidos

Gestión de los recursos.

Arquitectura de los subsistemas del equipo informático.

Gestión de buses y puertos de los sistemas operativos.

Protocolos de comunicación. Dispositivos físicos y puertos de comunicación. Servicios de comunicaciones.

Librerías y funciones del sistema.

### 2 Análisis de requisitos del desarrollo de programas de sistemas

El ciclo de desarrollo del 'software' en gestión de sistemas.

Análisis, diseño y programación para la gestión de sistemas.

Modelos de procesos: nomenclatura y representación.

Modelos de datos: nomenclatura y representación.

Herramientas de diagramación y documentación.

Métricas y estándares de calidad para desarrollo de 'software'.

### 3 Herramientas y entornos de desarrollo

Entornos integrados de desarrollo.

Editores.

Compiladores.

Depuración.

Control de versiones.

Tipos de licencia de aplicaciones. Clasificación y taxonomías. Uso libre, uso temporal, en desarrollo (beta), acuerdos corporativos de uso de aplicaciones, licencias mediante código, licencias mediante mochilas. 'Software' libre y 'copyright'.

#### 4 Desarrollo de elementos 'software' para la gestión del sistema y sus recursos

Técnicas y herramientas de programación orientadas a la programación de sistemas: tipos y estructuras de datos, operadores, estructuras lógicas, sentencias, recursividad, programación concurrente, excepciones.

Desarrollo de librerías de bajo nivel. Acceso a memoria, comunicaciones, operaciones de E/S, directa o por bloques con DMA. Drivers de sistema.

Llamadas a funciones del sistema. Gestión de memoria, comunicaciones, almacenamiento y E/S en general.

Programación en red. Aplicaciones cliente/servidor.

Técnicas y sistemas de distribución y despliegue de 'software'. Publicación en librerías.

Herramientas de compresión y empaquetamiento de 'software'.

#### 5 Pruebas y documentación de elementos 'software' para la gestión del sistema y sus recursos

Planificación y diseño de pruebas.

Técnicas y herramientas de depuración de 'software'.

Pruebas y verificación del desarrollo.

Técnicas y herramientas de documentación del 'software' desarrollado.

### Parámetros de contexto de la formación

#### Espacios e instalaciones

Los talleres e instalaciones darán respuesta a las necesidades formativas de acuerdo con el contexto profesional establecido en la unidad de competencia asociada, teniendo en cuenta la normativa aplicable del sector productivo, prevención de riesgos laborales, accesibilidad universal, igualdad de género y protección medioambiental. Se considerará con carácter orientativo como espacios de uso:

- Instalación de 2 m<sup>2</sup> por alumno o alumna.

#### Perfil profesional del formador o formadora:

1. Dominio de los conocimientos y las técnicas relacionados con la creación de elementos 'software' para la gestión del sistema y sus recursos, que se acreditará mediante una de las dos formas siguientes:

- Formación académica de nivel 2 (Marco Español de Cualificaciones para la Educación Superior), Ingeniero Técnico, Diplomado o de otras de superior nivel relacionadas con el campo profesional.

- Experiencia profesional de un mínimo de 2 años en el campo de las competencias relacionadas con este módulo formativo.

2. Competencia pedagógica acreditada de acuerdo con lo que establezcan las Administraciones competentes.

## MÓDULO FORMATIVO 3

### PROGRAMACIÓN ORIENTADA A OBJETOS

Nivel:	3
Código:	MF0227_3
Asociado a la UC:	UC0227_3 - DESARROLLAR COMPONENTES 'SOFTWARE' EN LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS
Duración (horas):	240
Estado:	BOE

### Capacidades y criterios de evaluación

**C1:** Implementar elementos 'software', cumpliendo especificaciones del diseño y niveles de calidad para obtener los componentes orientados a objetos base de aplicaciones en diferentes tecnologías.

**CE1.1** Definir los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del 'software', interpretando las especificaciones establecidas en el diseño.

**CE1.2** Determinar las inconsistencias del diseño detectadas, registrándolas de acuerdo al procedimiento establecido para garantizar su trazabilidad.

**CE1.3** Codificar las clases definidas, sus atributos, relaciones, métodos e interfaces utilizando el lenguaje de programación elegido, las técnicas metodológicas orientadas a objetos, las especificaciones recibidas, las normas de calidad y las pautas de programación.

**CE1.4** Implementar el control de errores en el código de acuerdo a las normas de calidad.

**CE1.5** Emplear herramientas de desarrollo y depuración, facilitando el proceso de generación del código, detectando y corrigiendo errores en el código.

**CE1.6** Resolver los problemas encontrados en la implementación, consultando la documentación técnica, editada en castellano, en las lenguas cooficiales de las Comunidades Autónomas o en el idioma extranjero de uso más frecuente en el sector.

**CE1.7** Utilizar los componentes de terceros, seleccionándolos según las necesidades, interpretando su documentación e implementando su uso, reutilizando código y optimizando el tiempo de desarrollo.

**CE1.8** En un supuesto práctico, a partir de una documentación típica de diseño detallado, identificar los componentes orientados a objeto, utilizándolos como base en el desarrollo de aplicaciones en diferentes tecnologías como programación web, dispositivos móviles, servicios en la nube, aprendizaje automático, u otras:

- Diseñar un algoritmo para cada operación definida en la clase, aplicando técnicas de programación estructurada y modular.
- Elegir la estructura de datos más adecuada para cada atributo.
- Codificar cada atributo utilizando los tipos base proporcionados por el lenguaje, si es el caso, y las librerías de clases existentes.
- Codificar los métodos de acceso a los atributos siguiendo los criterios de calidad que se establezcan.
- Codificar los métodos constructores, utilizando la sobrecarga si es necesario, siguiendo los criterios de calidad que se establezcan.

- Codificar los métodos, como función o procedimiento, teniendo en cuenta la interfaz de la clase y los algoritmos diseñados y siguiendo los criterios de calidad que se establezcan.
- Incluir las relaciones de especialización / generalización, agregación / composición y/o de asociación con el resto de las clases descritas en el diseño en la construcción de la clase.
- Incluir el código para el tratamiento de casos de error y excepciones de usuario.
- Usar las librerías de clases existentes para incorporar accesos a bases de datos, interfaces gráficas y otras librerías.

**C2:** Aplicar técnicas de manipulación de bases de datos a través de interfaces para integrar el lenguaje de acceso a datos en la construcción de una aplicación.

**CE2.1** Determinar el conector de base de datos -con acceso directo, con mapeo objeto-relacional o el acceso directo a ficheros- a emplear en el programa, seleccionando una interfaz de programación de aplicaciones.

**CE2.2** Iniciar, configurar y cerrar sesión en la base de datos con la interfaz de programación o bien utilizar una sesión abierta del grupo -'pool'- disponible.

**CE2.3** Ejecutar las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación por medio de la interfaz de programación, garantizando la seguridad de la base de datos.

**CE2.4** Utilizar la interfaz de programación, garantizando que los accesos a la base de datos finalizan las transacciones, asegurando su integridad y consistencia.

**CE2.5** En un supuesto práctico, de construcción de una aplicación con manipulación de bases de datos a través de interfaces, integrando el lenguaje de programación con el lenguaje de acceso a datos, a partir de un diseño:

- Seleccionar la tecnología de conexión adecuada a las especificaciones del diseño, el entorno de trabajo y las características del lenguaje orientado a objetos que se vaya a emplear.
- Cargar el controlador, en su caso, de conexión de la base de datos.
- Ejecutar la conexión entre el programa y la base de datos, utilizando los componentes 'software' de la librería de acceso a la base de datos.
- Desarrollar los componentes 'software' encomendados de modo que cumplan las especificaciones del diseño, ejecutando consultas simples en la base de datos y disponiendo los resultados para su tratamiento, siguiendo las normas de calidad de la organización.
- Comprobar que los componentes desarrollados funcionan de acuerdo a las especificaciones del diseño.
- Redactar la documentación de los componentes desarrollados.

**C3:** Verificar componentes 'software' desarrollados, asegurando que cumplen especificaciones definidas en el proyecto, garantizando su funcionamiento en producción.

**CE3.1** Ejecutar el plan de pruebas, siguiendo los protocolos y los criterios de calidad definidos en la organización.

**CE3.2** Integrar las pruebas automatizadas, ejecutándolas dentro de los procesos de integración continua y control de calidad.

**CE3.3** Definir los datos de prueba de cada unidad funcional, contemplando una variedad de posibilidades, incluyendo casos límite y conflictivos.

**CE3.4** Determinar las pruebas de cada unidad funcional, cubriendo las pruebas de los caminos básicos que puedan surgir en su ejecución.

**CE3.5** Definir los casos de prueba que involucran varios objetos, teniendo en cuenta los escenarios de interacción posibles, verificando que los objetos mantienen coherente su estado tras el intercambio de mensajes.

**CE3.6** Ejecutar las pruebas funcionales de alto nivel, siguiendo los procesos definidos tanto de ejecución como documentación, asegurando el cumplimiento de las especificaciones.

**CE3.7** Ejecutar las pruebas de puesta en producción, asegurando que los componentes soportan la carga de trabajo del entorno y su integración con servicios externos a los mismos - pasarelas de pago, servicios de internet u otros-.

**CE3.8** En un supuesto práctico, ejecutar una estrategia de pruebas completa para los componentes 'software' desarrollados, asegurando que cumplen las especificaciones definidas en el proyecto garantizando su funcionamiento en producción en aspectos como:

- Coherencia en el estado de los componentes.
- Contemplar todos los escenarios posibles.
- Utilizar 'software' de automatización de pruebas.
- Rendimiento óptimo.
- Casos límite.
- Situaciones excepcionales.

**C4:** Aplicar técnicas de elaboración de documentación del código desarrollado según estándares, garantizando su futuro mantenimiento.

**CE4.1** Redactar la documentación -manuales, guías entre otras- para desarrolladores, administradores o usuarios de acuerdo a las especificaciones del proyecto.

**CE4.2** Redactar la documentación correspondiente a cada componente, cumpliendo las normas y modelos incluidos en el diseño de la aplicación.

**CE4.3** Añadir comentarios, identificadores con nombres autoexplicativos -mnemotécnicos-, tabulaciones u otros al código, documentándolo para la mejor comprensión del programa - 'software'-.

**CE4.4** Generar la documentación de una clase desarrollada, utilizando herramientas de producción automática de documentación a partir del texto incluido en el componente 'software'.

**CE4.5** Documentar los procedimientos de ejecución de las pruebas unitarias y de regresión, así como los resultados de las mismas de acuerdo a los estándares definidos por la organización.

**CE4.6** En un supuesto práctico, a partir del código desarrollado según los estándares de la organización, elaborar la documentación, garantizando su fase de mantenimiento:

- Guías y manuales para desarrolladores, administradores o usuarios.
- Documentación interna de los componentes 'software'.
- Documentación del código.
- Generación automática de la documentación.
- Documentación de la planificación de las pruebas realizadas y sus resultados.

**C5:** Aplicar técnicas de gestión de cambios en un código desarrollado para asegurar el cumplimiento de normas y niveles de calidad establecidos durante la fase de desarrollo y mantenimiento.

**CE5.1** Codificar los cambios en la implementación como consecuencia de cambios en las especificaciones de diseño, adecuando la estructura del código y manteniendo su calidad.

**CE5.2** Codificar las correcciones efectuadas como consecuencia de los errores o discrepancias encontrados en las pruebas en los componentes 'software' afectados.



**CE5.3** Aplicar técnicas de pruebas de regresión como consecuencia de cambios en las especificaciones, asegurando la funcionalidad previa.

**CE5.4** Mantener la documentación del código coherente con sus sucesivas modificaciones.

**CE5.5** En un supuesto práctico de modificación de una clase por un cambio en su diseño y de acuerdo a un procedimiento.

- Modificar el código para incorporar el cambio de acuerdo a las normas de calidad de la organización.
- Documentar los cambios realizados.
- Planificar y ejecutar las pruebas de regresión de acuerdo a las normas de calidad de la organización.
- Actualizar la documentación con las sucesivas modificaciones.
- Utilizar herramientas de gestión de cambios.

## Capacidades cuya adquisición debe ser completada en un entorno real de trabajo

C1 respecto a CE1.8; C2 respecto a CE2.5; C3 respecto a CE3.8; C4 respecto a CE4.6 y C5 respecto a CE5.5.

### Otras Capacidades:

Responsabilizarse del trabajo que desarrolla y del cumplimiento de los objetivos.

Proponerse objetivos retadores que supongan un nivel de rendimiento y eficacia superior al alcanzado previamente.

Demostrar cierto grado de autonomía en la resolución de contingencias relacionadas con su actividad.

Aprender nuevos conceptos o procedimientos y aprovechar eficazmente la formación utilizando los conocimientos adquiridos.

Adaptarse a la organización, a sus cambios organizativos y tecnológicos, así como a situaciones o contextos nuevos.

Adoptar códigos de conducta tendentes a transmitir el contenido del principio de igualdad.

## Contenidos

### 1 Metodología de la programación orientada a objetos vinculada al desarrollo de componentes 'software' en lenguajes de programación orientada a objetos

Ciclo de desarrollo del 'software' bajo el paradigma de orientación a objetos: Análisis, diseño y programación orientada a objetos.

Proceso de construcción de 'software': Modularidad. Módulo en el paradigma orientado a objetos.

Objetos como abstracciones de las entidades del mundo real que se quiere modelar.

Descripción de objetos: Conjunto de datos que definen un objeto y conjunto de comportamientos que pueden solicitarse a los objetos.

Comportamiento de un objeto: Concepto de mensaje.

### 2 Desarrollo de clases y objetos

Clase: atributos, métodos y mecanismo de encapsulación. Interfaz de la clase. Clases y tipos de datos.

Clasificación jerárquica de las clases. Clase raíz. Clases abstractas. Métodos virtuales. Redefinición de métodos.

Polimorfismo. Tipos: Polimorfismo en tiempo de compilación (sobrecarga) y polimorfismo en tiempo de ejecución (ligadura dinámica). Objetos polimórficos. Comprobación estática y dinámica de tipos.

Herencia. Herencia simple y múltiple. Reglas y características que definen una relación de herencia. Transmisión de atributos y métodos. Acceso a los atributos de una clase y acoplamiento entre las clases. Utilización de objetos. Creación de objetos en la herencia.

Objetos: Estado, comportamiento e identidad. Mensajes. Tipos de métodos. Clasificación: Métodos de acceso, de selección o consulta, de construcción y de destrucción.

Objetos como instancias de clase. Instancia actual.

Concepto de programa en el paradigma orientado a objetos. POO = Objetos + Mensajes.

Lenguajes de programación orientado a objetos, características. Lenguajes de programación basados en objetos y lenguajes de programación que utilizan objetos

### 3 Programación estructurada vinculada al desarrollo de componentes 'software' en lenguajes de programación orientada a objetos

Elementos: constantes, variables, operadores y expresiones.

Tipos de datos: Datos simples, numéricos (enteros y reales), lógicos, carácter, cadena de caracteres, puntero o referencia a memoria; Datos estructurados, arrays, listas enlazadas, pilas y colas.

Estructuras. Ficheros. Otras estructuras complejas: tablas hash e introducción a los árboles y grafos.

Estructuras de control. Secuencial, condicional y de repetición.

Funciones y procedimientos. Interfaz. Paso de parámetros: por valor y por referencia. Parámetros actuales y formales. Funciones: valor de retorno. Procedimientos. Ámbito de las variables.

Almacenamiento de las variables.

Llamadas a funciones y procedimientos. Llamadas a funciones y procedimientos incluidos en las clases. Llamadas calificadas y no calificadas (instancia actual). Paso de parámetros. Los atributos de la clase.

Librerías. Estructura. Creación y utilización. Librerías de clases.

Reutilización del 'software'.

### 4 Acceso a bases de datos y otras estructuras vinculado al desarrollo de componentes 'software' en lenguajes de programación orientada a objetos

Objetos de la base de datos.

Integridad, consistencia y seguridad de los datos.

Conexiones para el acceso a datos. Establecimiento, configuración y cierre de la conexión. Objetos de acceso a datos.

Herramientas de acceso a datos proporcionadas por el entorno de programación.

Herramientas de mapeado objeto-relacional (ORM)

Sentencias del lenguaje de programación orientado a objetos para operar sobre las bases de datos.

Integración de los objetos de la base de datos en el lenguaje de programación orientado a objetos.

Integración de los tipos de datos propios del lenguaje de acceso a base de datos en el lenguaje de programación de la aplicación.

Procedimientos almacenados.

Transacciones distribuidas.

Seguridad.

Inyección de SQL.

### 5 Técnicas de pruebas de 'software'

Criterios de calidad. Métricas y estándares de calidad.

Objetivos de las pruebas. Tipos de pruebas.

Planificación de las pruebas: escenarios (datos -consultas, inserciones, borrados-, del sistema, de plataforma), casos de prueba.

Proceso de pruebas. Pruebas de integración, de interfaces, de rendimiento y de seguridad.

Herramientas para automatización de pruebas e integración continua.  
Gestión de errores.  
Gestión de cambios, concepto y pruebas de regresión.  
Análisis de resultados.  
Documentación de pruebas.

## 6 Mecanismos de gestión de memoria y tratamiento de errores

Gestión automática de memoria.  
Construcción y destrucción de objetos. Objetos inalcanzables. Recolección de "basura".  
Métodos constructores y destructores.  
Definición de excepción. Fuentes de excepciones. Tratamiento de excepciones. Prevención de fallos. Excepciones definidas y lanzadas por el programador.  
Excepciones tratadas como objetos.

## 7 Técnicas de documentación

Documentación de una aplicación, características, tipos: documentación técnica; guía de uso de la aplicación.  
Herramientas de documentación: características.  
Herramientas para generación de ayudas.  
Normativa aplicable y planificación de la actividad preventiva relativa a ergonomía, disposiciones mínimas de seguridad y salud en lugares de trabajo, entre otros.

## Parámetros de contexto de la formación

### Espacios e instalaciones

Los talleres e instalaciones darán respuesta a las necesidades formativas de acuerdo con el contexto profesional establecido en la unidad de competencia asociada, teniendo en cuenta la normativa aplicable del sector productivo, prevención de riesgos laborales, accesibilidad universal, igualdad de género y protección medioambiental. Se considerará con carácter orientativo como espacios de uso:

- Taller de 2 m<sup>2</sup> por alumno o alumna.
- Instalación de 4 m<sup>2</sup> por alumno o alumna.

### Perfil profesional del formador o formadora:

1. Dominio de los conocimientos y las técnicas relacionadas con desarrollar componentes 'software' en lenguajes orientados a objetos, en lengua propia y extranjera, que se acreditará mediante una de las dos formas siguientes:

- Formación académica de nivel 2 (Marco Español de Cualificaciones para la Educación Superior) o de otras de superior nivel relacionadas con el campo profesional.
- Experiencia profesional de un mínimo de 2 años en el campo de las competencias relacionadas con este módulo formativo.

2. Competencia pedagógica acreditada de acuerdo con lo que establezcan las Administraciones competentes.

## MÓDULO FORMATIVO 4

### Desarrollo de elementos 'software' con métodos y tecnologías orientadas a componentes

Nivel:	3
Código:	MF0965_3
Asociado a la UC:	UC0965_3 - Desarrollar elementos 'software' con métodos y tecnologías orientados a componentes
Duración (horas):	120
Estado:	BOE

### Capacidades y criterios de evaluación

**C1:** Aplicar técnicas de instalación de herramientas, previa selección en su caso, analizando las tecnologías disponibles, escogiendo aquellas que se ajusten a las especificaciones del producto o aplicación basada en componentes, configurándolas según los requisitos técnicos, para usarlas en su desarrollo posterior.

**CE1.1** Describir técnicas y métodos involucrados en el paradigma del desarrollo orientado a componentes, explicando sus objetivos y enumerando los medios a utilizar para el fin que se persigue.

**CE1.2** Clasificar herramientas, elementos de las mismas, 'framework' 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'.

**CE1.3** En un supuesto práctico de aplicación de técnicas de instalación de herramientas de programación, previa selección en su caso, analizando las tecnologías disponibles, escogiendo aquellas que se ajusten a las especificaciones del producto o aplicación basada en componentes, configurándolas según los requisitos técnicos, para usarlas en su desarrollo posterior:

- Evaluar varias tecnologías a utilizar, analizando si permiten o no, elementos tales como paletas, lienzo o contenedor, editores para configurar y especializar componentes, buscadores, repositorios de componentes, acceso a intérpretes, compiladores y depuradores, herramientas de gestión y control de proyectos, entre otros.

- Seleccionar un entorno de desarrollo se selecciona, escogiendo aquel que ofrezca, de entre todos los analizados, un coste/beneficio óptimo en función de las facilidades que proporcionan y las limitaciones tecnológicas para el producto o aplicación a desarrollar.

- Instalar un entorno de desarrollo integrado (IDE), complementos adicionales y los 'framework' asociados a las tecnologías basadas en componentes, configurándolos de modo que permitan minimizar el tiempo de desarrollo y maximizar la seguridad y control de la aplicación o producto.

- Elaborar la documentación de la instalación, siguiendo unos modelos, recogiendo las configuraciones y/o acciones aplicadas y archivándola para su control, trazabilidad y uso posterior.

**CE1.4** En un supuesto práctico de diseño de 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 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, normas y procedimientos especificados.

**C2:** Aplicar técnicas de desarrollo de 'software', usando el paradigma de programación orientada a componentes, empleando un entorno de desarrollo específico de esta tecnología, para su integración en una arquitectura definida, según un diseño y especificaciones de análisis.

**CE2.1** Identificar técnicas de diagramación y documentación de desarrollo orientado a componentes, según estándares de diseño de metodologías orientadas a componentes.

**CE2.2** Explicar los enfoques de desarrollo e implementación para la creación de componentes, aplicando el principio de reutilización.

**CE2.3** 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.

**CE2.4** Detallar el funcionamiento de las herramientas de programación y depuración para optimizar la fase de desarrollo de los componentes según unas especificaciones.

**CE2.5** 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.

**CE2.6** Aplicar técnicas de prueba de componentes, comprobando que ejecuta las acciones requeridas y su disponibilidad para las aplicaciones que lo invoquen, según especificaciones de la arquitectura.

**CE2.7** En un supuesto práctico de aplicación de técnicas de desarrollo de 'software' usando el paradigma de programación orientada a componentes, empleando un entorno de desarrollo específico de esta tecnología, para su integración en una arquitectura definida, según un diseño y especificaciones de análisis:

- Comprobar que la funcionalidad de un componente diseñado puede ser extendida para futuras reutilizaciones, modificando en otro caso el mismo para lograrlo.
- Desarrollar un componente utilizando herramientas y lenguajes orientados a objeto y según las especificaciones de diseño resultantes de la comprobación anterior.
- Depurar y probar el componente desarrollado utilizando las herramientas disponibles.
- Documentar el componente y sus interfaces siguiendo unos modelos, recogiendo las configuraciones y/o acciones aplicadas y archivándola para su control, trazabilidad y uso posterior.

**CE2.8** En un supuesto práctico de desarrollo de '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, recogiendo los trabajos realizados para facilitar su futura reutilización.
- Utilizar una herramienta de control de versiones para garantizar el control de cambios en el 'software', facilitando el desarrollo, las pruebas y el retorno a versiones previas de las modificaciones relativas a nuevas funcionalidades y corrección de errores.

**C3:** Aplicar procedimientos de despliegue e integración de componentes en un entorno de tecnología orientada a componentes, según especificaciones técnicas de una arquitectura y criterios de calidad y seguridad.

**CE3.1** Clasificar estándares de modelos de componentes, describiendo las pasarelas para interoperar entre componentes heterogéneos, para realizar las tareas de integración de los elementos desarrollados según unas especificaciones funcionales y técnicas.

**CE3.2** Describir las técnicas para realizar el proceso de búsqueda de componentes, de manera que se satisfagan unos requisitos.

**CE3.3** Clasificar métodos de evaluación y selección de componentes, describiendo los pasos a seguir en cada caso.

**CE3.4** Explicar los procedimientos de pruebas estructurales para verificar que un componente seleccionado se comunica con el resto y que no produce conflictos, detallando los pasos que se siguen y los criterios que se aplican.

**CE3.5** Definir procedimientos para el despliegue y adaptación para la implantación del elemento 'software', según requisitos del componente desarrollado y de las herramientas a utilizar.

**CE3.6** Describir los procesos de integración, configuración e interconexión de los componentes que constituyen una aplicación final, explicando la sistemática a aplicar.

**CE3.7** Aplicar sistemas de monitorización del rendimiento de unos componentes desarrollados o seleccionados para asegurar su integración en el sistema, conectando las alertas sobre el sistema a supervisar mediante configuración de la herramienta utilizada.

**CE3.8** Documentar el despliegue y la implantación siguiendo los patrones, normas y procedimientos especificados.

**CE3.9** En un supuesto práctico de aplicación de procedimientos de despliegue e integración de componentes en un entorno de tecnología orientada a componentes, según especificaciones técnicas de una arquitectura y criterios de calidad y seguridad:

- Probar la estructura de un componente, verificando que se comunica con el resto y que no produce conflictos.
- Ejecutar el despliegue, según requisitos del componente desarrollado, usando las facilidades y utilidades que proporciona el entorno, para asegurar la implantación del mismo con la provisión de sus funcionalidades y la ausencia de conflictos.
- Monitorizar el rendimiento de los componentes desarrollados, usando las utilidades del entorno para asegurar y garantizar su integración en el sistema.
- Elaborar la documentación del despliegue e implantación, siguiendo los modelos internos establecidos por la organización, recogiendo las configuraciones y/o acciones aplicadas y archivándola para su control, trazabilidad y uso posterior.

## Capacidades cuya adquisición debe ser completada en un entorno real de trabajo

C1 respecto a CE1.3 y CE1.4; C2 respecto a CE2.7 y CE2.8; C3 respecto a CE3.9.

### Otras Capacidades:

Responsabilizarse del trabajo que desarrolla y del cumplimiento de los objetivos.

Proponerse objetivos retadores que supongan un nivel de rendimiento y eficacia superior al alcanzado previamente.

Demostrar cierto grado de autonomía en la resolución de contingencias relacionadas con su actividad.

Aprender nuevos conceptos o procedimientos y aprovechar eficazmente la formación utilizando los conocimientos adquiridos.

Adaptarse a la organización, a sus cambios organizativos y tecnológicos, así como a situaciones o contextos nuevos.

Adoptar códigos de conducta tendentes a transmitir el contenido del principio de igualdad.

## Contenidos

### 1 El paradigma de la orientación a componentes

Programación orientada a componentes frente a programación orientada a objetos.

Diagramas de componentes.

Componentes e Interfaces. Estados del componente.

### 2 Arquitecturas de componentes y desarrollo

Entornos de desarrollo específicos. Complementos. 'Framework'.

Arquitecturas actuales orientadas a componentes.

Enfoques de desarrollo e implementación. Reutilización de código. Dependencias del contexto.

Búsqueda en repositorios. Evaluación y selección de componentes reutilizables.

Especificación e implementación del componente.

Lenguajes de descripción de interfaces (IDL).

Técnicas y herramientas de prueba y depuración de componentes aislados.

### 3 Programación distribuida

Integración y despliegue de componentes.

Técnicas y herramientas de pruebas de integración.

Monitorización del sistema.

Llamadas a procedimientos remotos (RPC).

Servidores de aplicaciones.

Técnicas y herramientas de documentación del 'software' desarrollado.

Herramientas de compresión y empaquetamiento de 'software'.

Sistemas de control de versiones (CVS).

## Parámetros de contexto de la formación

### Espacios e instalaciones

Los talleres e instalaciones darán respuesta a las necesidades formativas de acuerdo con el contexto profesional establecido en la unidad de competencia asociada, teniendo en cuenta la normativa aplicable del sector productivo, prevención de riesgos laborales, accesibilidad universal, igualdad de género y protección medioambiental. Se considerará con carácter orientativo como espacios de uso:

- Instalación de 2 m<sup>2</sup> por alumno o alumna.

### Perfil profesional del formador o formadora:

1. Dominio de los conocimientos y las técnicas relacionados con el desarrollo de elementos 'software' con métodos y tecnologías orientados a componentes, que se acreditará mediante una de las dos formas siguientes:

- Formación académica de nivel 2 (Marco Español de Cualificaciones para la Educación Superior) o de otras de superior nivel relacionadas con el campo profesional.

- Experiencia profesional de un mínimo de 2 años en el campo de las competencias relacionadas con este módulo formativo.

2. Competencia pedagógica acreditada de acuerdo con lo que establezcan las Administraciones competentes.