

Cualificación Profesional	PROGRAMACIÓN CON LENGUAJES ORIENTADOS A OBJETOS Y BASES DE DATOS RELACIONALES
Familia Profesional	Informática y Comunicaciones
Nivel	3
Código	IFC080_3
Versión	5
Situación	Publicada

Competencia general

Desarrollar aplicaciones informáticas sobre el diseño especificado utilizando lenguajes orientados a objetos y bases de datos relacionales.

Unidades de competencia

UC0223_3: Configurar y explotar sistemas informáticos.

UC0226_3: Programar bases de datos relacionales.

UC0227_3: Desarrollar componentes software en lenguajes de programación orientados a objetos.

Entorno Profesional

Ámbito Profesinal

Desarrolla su actividad profesional en los siguientes ámbitos:

- Empresas que tienen como objetivo de negocio la comercialización de servicios de análisis, diseño y construcción de aplicaciones informáticas.
- Como parte del equipo de sistemas informáticos de grandes organizaciones.

Sectores Productivos

Está presente en los siguientes tipos de empresas:

- Empresas de desarrollo de software.
- Empresas de consultoría técnica en sistemas de información.
- Empresas de servicios de teleatención y asistencia técnica.
- Empresas o entidades que utilizan sistemas informáticos para su gestión.

Ocupaciones y puestos de trabajo relevantes

Técnico en Data Mining (minería de datos)
Programador de aplicaciones de gestión.

Formación Asociada (610 horas)

Módulos Formativos

MF0223_3: Sistemas operativos y aplicaciones informáticas(150 h)

MF0226_3: Programación de bases de datos relacionales(210 h)

MF0227_3: Programación orientada a objetos(250 h)

UNIDAD DE COMPETENCIA 1 Configurar y explotar sistemas informáticos.

Nivel 3

Código UC0223_3

Realizaciones profesionales y criterios de realización

RP 1: Adaptar la configuración lógica del sistema para su explotación, según las necesidades de uso y dentro de las directivas de la organización.

CR 1.1 Los parámetros del sistema que afectan a la memoria, procesador y periféricos se ajustan a las necesidades de uso.

CR 1.2 Los dispositivos necesarios y sus ficheros de control se añaden o eliminan empleando para ello las utilidades del sistema operativo.

CR 1.3 Las conexiones lógicas del equipo se configuran para acceder a servicios remotos dentro o fuera de la organización.

CR 1.4 Los parámetros del sistema que afectan a la ergonomía o a la facilidad de uso se ajustan para mejorar las condiciones de trabajo del usuario, dentro de las directivas de la organización.

RP 2: Organizar la información en los sistemas de archivo del sistema operativo y mantener sus propiedades para facilitar el aprovechamiento de los recursos y asegurar el cumplimiento de las directivas de la organización.

CR 2.1 Las aplicaciones informáticas se organizan con una estructura y configuración que permitan su uso en óptimas condiciones.

CR 2.2 La información de usuario del sistema operativo se mantiene en estructuras organizadas de acuerdo con las posibilidades del propio sistema (ficheros, directorios, volúmenes, etc.) para facilitar el acceso a dicha información y mantener la homogeneidad en los diversos equipos de la organización.

CR 2.3 La estructura y configuración del sistema de archivos se conservan en disposición de uso para evitar fallos accidentales y compartir información.

CR 2.4 El espacio de almacenamiento de información se mantiene libre de informaciones inútiles u obsoletas para mejorar el rendimiento del sistema y aumentar su vida útil.

RP 3: Elaborar y transferir documentos mediante el uso de aplicaciones informáticas de propósito general.

CR 3.1 Las herramientas ofimáticas se utilizan con la destreza necesaria para auxiliar en las tareas de planificación y documentación de los trabajos.

CR 3.2 El intercambio de información con otras personas se realiza utilizando los sistemas de correo o mensajería electrónica para facilitar el flujo de información y reducir costes y tiempos cuando la naturaleza de dicho intercambio de información lo permita.

CR 3.3 Los servicios disponibles en Internet, u otras redes, se obtienen, mediante el correcto uso de las herramientas necesarias (navegación, foros, clientes ftp, etc.), para facilitar el acceso a información necesaria para el trabajo.

RP 4: Garantizar la integridad, disponibilidad y confidencialidad de la información de la que se es responsable y se encuentre almacenada en el sistema de archivos.

CR 4.1 la información almacenada (datos y software) puede devolverse a un estado que permita su utilización en cualquier momento mediante, entre otros medios, las copias de seguridad

CR 4.2 El acceso a la información se protege mediante el uso de claves y otras medidas de seguridad establecidas en la organización.

CR 4.3 Los medios de protección frente a desastres o accesos indebidos (antivirus, cortafuegos, proxys, sistemas de gestión de cambios, etc.) se implantan y utilizan en los sistemas de los que se es responsable.

CR 4.4 El sistema se mantiene libre de software no licenciado.

CR 4.5 Las normas internas de la organización y la legislación vigente sobre protección de datos se cumplen en los sistemas de los que se es responsable.

CR 4.6 Las incidencias se notifican al Administrador de sistemas para que realice las labores oportunas.

Contexto profesional

Medios de producción

Equipos informáticos y periféricos. Sistemas operativos y parámetros de configuración. Herramientas ofimáticas. Servicios de transferencia de ficheros y mensajería. Herramientas de backup. Cortafuegos antivirus y servidores proxy. Herramientas de gestión de cambios, incidencias y configuración.

Productos y resultados

Sistema informático en funcionamiento con un rendimiento óptimo y una utilización adecuada de sus recursos. Conexión en red adecuada dentro de una organización. Sistema operativo y aplicaciones configurados y parametrizados de acuerdo a las necesidades. Ficheros con información acorde a la naturaleza de la actividad profesional desarrollada (programas, guiones de consultas, documentos de texto, hojas de cálculo, etc.) almacenados en soporte físico adecuado. Copias de seguridad de la información según criterios de integridad, confidencialidad y disponibilidad.

Información utilizada o generada

Manuales de uso y funcionamiento de los sistemas informáticos. Manuales de funcionamiento del software asociado. Material de cursos de formación. Sistemas de ayuda de las aplicaciones informáticas Soportes técnicos de asistencia (telefónica, Internet, mensajería, foros, etc.)

UNIDAD DE COMPETENCIA 2 Programar bases de datos relacionales.

Nivel 3
Código UC0226_3

Realizaciones profesionales y criterios de realización

RP 1: Interpretar las estructuras de datos y el diseño de la base de datos de forma que pueda realizar correctamente las tareas de programación encomendadas, respetando las reglas de integridad y restricciones del sistema de información.

CR 1.1 El diseño lógico y la estructura de la base de datos se conocen al nivel necesario para establecer relaciones entre los elementos de datos.

CR 1.2 Las restricciones, reglas de integridad y semántica de los datos se identifican e interpretan para poder realizar correctamente las tareas de programación señaladas.

CR 1.3 El diseño físico y las particularidades de la implementación de la base de datos se conocen al nivel necesario para permitir la manipulación de los datos, identificando tipos de datos, índices, vistas y otras características implementadas.

RP 2: Manipular el contenido de bases de datos relacionales de forma interactiva.

CR 2.1 Las especificaciones recibidas se interpretan con corrección identificando los objetos de la base de datos que se van a manipular.

CR 2.2 Las herramientas de cliente de acceso a la base de datos se utilizan del modo adecuado para consultar la estructura de la base de datos y sus elementos (tablas, atributos, tipos de datos, relaciones, vistas, procedimientos almacenados, etc.).

CR 2.3 Las operaciones de manipulación de datos se construyen con corrección, de acuerdo a las especificaciones recibidas y utilizando un lenguaje de manipulación de datos o herramientas gráficas de acceso a datos.

CR 2.4 Las operaciones de manipulación de datos construidas se prueban en ambientes controlados y con información conocida para verificar que cumplen las especificaciones recibidas.

RP 3: Programar módulos de manipulación de la base de datos

CR 3.1 Las especificaciones recibidas se interpretan con corrección identificando los objetos de la base de datos que se van a manipular.

CR 3.2 La codificación se realiza en el lenguaje de programación propio del sistema de base de datos y siguiendo las especificaciones del diseño.

CR 3.3 El código desarrollado debe finalizar las transacciones asegurando la integridad y consistencia de la base de datos en cualquier caso.

CR 3.4 Las consultas se prueban en ambientes controlados y con información conocida.

CR 3.5 Las consultas se optimizan utilizando las técnicas y herramientas disponibles.

CR 3.6 Las estructuras de almacenamiento temporal necesarias son manipuladas de acuerdo con las normas de diseño de la base de datos.

Contexto profesional

Medios de producción

- Equipos informáticos y periféricos.
- Herramientas ofimáticas.
- Sistemas gestores de bases de datos.
- Diccionarios de datos (catálogo, tablas de sistema, etc.).
- Lenguajes de manipulación de datos.
- Lenguajes estructurados.
- Lenguajes orientados a objetos.
- Lenguajes 4GL.
- Herramientas de control de cambios.
- Herramientas de depuración.
- Sistemas de documentación de elementos de programación.

Productos y resultados

- Sistema informático con una configuración adecuada para el acceso a las bases de datos.

- Entorno de programación adaptado para la utilización de objetos de acceso a datos.
- Ficheros almacenados en soporte físico con información acorde a la naturaleza de la actividad profesional desarrollada (programas, guiones de consultas, documentos de texto, hojas de cálculo, etc.).
- Consultas para la manipulación de la base de datos de forma interactiva.
- Aplicaciones que manipulan la base de datos a través de código embebido.
- Conexiones lógicas disponibles para permitir el acceso a clientes.
- Mecanismos adecuados para la recuperación de transacciones.
- Programas de prueba.
- Procedimientos y casos de prueba.
- Documentación asociada al código desarrollado.

Información utilizada o generada

- Manuales de funcionamiento del software.
 - Material de cursos de formación.
 - Manuales de operación de los SGBD.
 - Diseño lógico y físico de las BBDD.
 - Legislación vigente acerca de protección de datos y confidencialidad de la información.
 - Ayuda en línea de las aplicaciones.
 - Soportes técnicos de asistencia (telefónica, Internet, mensajería, foros, etc.).

UNIDAD DE COMPETENCIA 3 Desarrollar componentes software en lenguajes de programación orientados a objetos.

Nivel 3
Código UC0227_3

Realizaciones profesionales y criterios de realización

RP 1: Implementar los componentes software encomendados de modo que cumplan las especificaciones del diseño y los niveles de calidad establecidos.

CR 1.1 Los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software se identifican interpretando correctamente el diseño detallado.

CR 1.2 Las inconsistencias del diseño que sean detectadas son transmitidas al responsable del diseño por los cauces establecidos.

CR 1.3 Las clases definidas, sus atributos, relaciones, métodos e interfaces se codifican utilizando el lenguaje de programación elegido y las librerías de clases definidas, de acuerdo con las especificaciones recibidas.

CR 1.4 La implementación se realiza siguiendo técnicas metodológicas orientadas a objetos, y de acuerdo con las normas de programación y calidad del software vigentes en la organización.

CR 1.5 El código implementa el control de errores especificado por las normas de calidad de la organización y por la especificación de requisitos.

CR 1.6 Las herramientas de desarrollo se emplean para facilitar el proceso de generación del código.

CR 1.7 Las herramientas de depuración se emplean para detectar y corregir errores del código desarrollado.

CR 1.8 Los problemas encontrados en la implementación se resuelven consultando la documentación técnica y otras fuentes de información.

CR 1.9 La documentación técnica se interpreta correctamente tanto si está editada en castellano o las lenguas oficiales de las Comunidades Autónomas como si lo está en el idioma extranjero de uso más frecuente en el sector.

CR 1.10 La codificación de cada clase se realiza manteniendo los principios de máxima cohesión y mínimo acoplamiento.

CR 1.11 Las correcciones necesarias como consecuencia de los errores o discrepancias encontrados en las pruebas se codifican en los componentes software afectados.

CR 1.12 Los cambios en la implementación como consecuencia de cambios en las especificaciones se codifican adecuando la estructura del código y manteniendo su calidad.

RP 2: 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.

CR 2.1 El interfaz de programación de acceso a datos (OLE DB, JDBC, DAO, RDO, ADO, ADO.NET, etc.) se selecciona según las necesidades de la aplicación.

CR 2.2 La herramienta de programación se adapta según el interfaz de objetos de acceso a datos seleccionado (integración de librerías, configuración de rutas, etc.).

CR 2.3 El interfaz de programación se utiliza para iniciar, configurar y cerrar una sesión con la base de datos, o bien se utiliza una de las sesiones abiertas del grupo (pool) si está disponible.

CR 2.4 Las operaciones de manipulación de datos necesarias para el funcionamiento de la aplicación se ejecutan por medio del interfaz de programación.

CR 2.5 El interfaz de programación se utiliza para garantizar que los accesos a la base de datos finalizan las transacciones asegurando su integridad y consistencia en cualquier caso.

RP 3: Probar los componentes software desarrollados para asegurar que cumplen las especificaciones recibidas

CR 3.1 El conjunto de caminos básicos de una unidad de código se identifica para la preparación de las pruebas de caja blanca.

CR 3.2 Cada unidad de código se prueba mediante la realización de pruebas estructurales o de caja blanca para verificar su funcionamiento.

CR 3.3 Cada unidad de código se prueba mediante la realización de pruebas funcionales o de caja negra para verificar que cumple las especificaciones.

CR 3.4 Las unidades de código se prueban de manera aislada y/ o en grupos de unidades considerando el nivel más bajo.

CR 3.5 Las pruebas de regresión necesarias como consecuencia de cambios en las especificaciones se realizan para asegurar la funcionalidad previa.

CR 3.6 Los componentes software desarrollados se prueban en ambientes controlados y con información conocida.

CR 3.7 Las pruebas se codifican para verificar que el paso de mensajes y/ o eventos a través del interfaz mantiene coherente el estado de los objetos de la clase.

CR 3.8 Los escenarios posibles se prueban mediante la interacción de los objetos implicados.

CR 3.9 Todos los métodos se prueban como parte de alguna secuencia.

CR 3.10 Los casos de prueba definidos en el diseño (entradas, resultados esperados, recursos necesarios y criterios de evaluación de dichas pruebas) se efectúan utilizando las herramientas de pruebas especificadas.

RP 4: Utilizar los componentes orientados a objeto como base en el desarrollo de aplicaciones para el modelo de programación web.

CR 4.1 El código orientado a objeto se utiliza en los componentes software de una aplicación web para aportar la información necesaria para la preparación del interfaz para el cliente.

CR 4.2 El código orientado a objeto se llama desde componentes software de aceptación de solicitudes de cliente para efectuar la operación solicitada.

CR 4.3 El código orientado a objeto se descarga a clientes ligeros para colaborar en el funcionamiento de la capa de presentación integrándose en dicha capa.

CR 4.4 Los objetos de una aplicación orientada a objetos en el modelo de programación web viajan desde ordenadores clientes a servidores o entre servidores para realizar las funciones que les son encomendadas según protocolos y estándares definidos.

RP 5: Elaborar la documentación del código desarrollado según los estándares de la organización.

CR 5.1 La documentación correspondiente a cada clase se redacta de acuerdo con las normas y modelos incluidos en el diseño de la aplicación.

CR 5.2 La documentación de una clase desarrollada se genera mediante las herramientas de producción automática de documentación a partir del texto incluido en el componente software, si se dispone de tales herramientas.

CR 5.3 El código contiene los comentarios necesarios para la mejor comprensión del software así como identificadores con nombres autoexplicativos (mnemotécnicos), tabulaciones, etc.

CR 5.4 La documentación del código se mantiene coherente con sus sucesivas modificaciones.

CR 5.5 Los procedimientos de realización de las pruebas unitarias y de regresión así como los resultados de las mismas se documentan convenientemente.

Contexto profesional

Medios de producción

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

Productos y resultados

- Código fuente de la aplicación.
 - Código ejecutable de la aplicación.
 - Programas de prueba.
 - Procedimientos y casos de prueba.
 - Documentación asociada al código.

Información utilizada o generada

- 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 las aplicaciones.

- Soportes técnicos de asistencia (telefónica, Internet, mensajería, foros, etc.).

MÓDULO FORMATIVO	1 Sistemas operativos y aplicaciones informáticas
Nivel	3
Código	MF0223_3
Asociado a la UC	Configurar y explotar sistemas informáticos.
Duración horas	150

Capacidades y criterios de evaluación

- C1: Diferenciar los componentes principales de un ordenador indicando sus funciones y características técnicas.**
- CE1.1 Explicar los componentes principales de un ordenador o servidor de propósito general sobre la base de su función y utilidad.
 - CE1.2 Enumerar y describir los elementos de la placa base de un ordenador reconociendo sus funciones principales.
 - CE1.3 Clasificar los tipos de procesadores principales atendiendo a su familia tecnológica, evolución histórica y características más relevantes.
 - CE1.4 Clasificar y explicar los periféricos y componentes de entrada/salida principales de un ordenador señalando la función que desarrollan en el conjunto del sistema.
 - CE1.5 Enumerar y clasificar los comandos principales del conjunto de instrucciones de bajo nivel de un procesador sobre la base de la función que ejecutan.
 - CE1.6 En una serie de supuestos prácticos de configuración de sistemas microinformáticos debidamente caracterizados mediante diagrama de conexiones y documentación técnica:
 - Identificar la placa base y reconocer: el procesador, los bancos de memoria, localizar los discos y unidades de disquete y CD / DVD, localizar los conectores de entrada /salida y clasificarlos por tipo.
- C2: Analizar las funciones principales de un sistema operativo multiusuario y multitarea, reconociendo y clasificando los diferentes tipos de sistemas operativos existentes.**
- CE2.1 Explicar los conceptos de núcleo, núcleo virtual e intérprete de comandos de un sistema operativo.
 - CE2.2 Explicar los diferentes modos de direccionar y almacenar los archivos y sistemas de archivo de un sistema operativo y de estructurar los permisos de lectura y edición.
 - CE2.3 Analizar la función de la memoria en el proceso de tareas del ordenador e identificar los conceptos relacionados con ella: memoria central y expandida, memoria virtual y paginación e intercambio.
 - CE2.4 Enumerar las diferentes políticas de reparto de tiempo de procesador implementadas en los sistemas operativos, identificando el impacto de cada una de ellas en los tipos de procesos.
 - CE2.5 Reconocer y explicar las funciones de los cambios de contexto, semáforos, planificador de trabajos y manejadores de interrupciones en el funcionamiento de los sistemas operativos multiusuario y multitarea.
 - CE2.6 Explicar los diferentes mecanismos de entrada/salida que maneja un sistema operativo en función del manejo de recursos.
 - CE2.7 Clasificar los sistemas operativos y arquitecturas por las diferentes formas que históricamente se han empleado.
 - CE2.8 A partir de un supuesto práctico de configuración de un sistema informático multiusuario y multiproceso:
 - Instalar diferentes sistemas operativos en la máquina identificando los hitos importantes del proceso.
 - Configurar las áreas de paginación e intercambio de memoria y reconocer su impacto en el sistema
 - Provocar e interpretar los bloqueos de recursos y su impacto en el comportamiento del sistema.
 - Crear y organizar archivos y sistemas de archivos.
- C3: Distinguir y analizar las variables de configuración de un sistema operativo, especificando su efecto sobre el comportamiento del sistema.**
- CE3.1 Enumerar y explicar los diferentes tipos de dispositivos lógicos usados para la instalación de servicios y aplicaciones.
 - CE3.2 Reconocer y explicar los principales parámetros de configuración del núcleo de un sistema operativo y su impacto sobre el comportamiento del sistema.
 - CE3.3 Analizar los servicios principales que se ejecutan en un sistema operativo y su influencia y competencia en la gestión de recursos.
 - CE3.4 Describir las diferentes maneras de monitorizar y ajustar los componentes de un sistema operativo y analizar tendencias a partir del estado de carga.
 - CE3.5 Correlacionar alarmas enviadas por el sistema de monitorización previamente implementado y definir eventos para su resolución.

CE3.6 A partir de un supuesto práctico por documentación técnica de la instalación y configuración del sistema operativo:

- Confeccionar la estructura de archivos y sistemas de archivo con los permisos de usuario.
- Detallar los procesos arrancados en la máquina.

- Detallar el estado de carga de: Ocupación en disco, uso de memoria, identificar las redes definidas en el sistema, instalar y compilar diferentes manejadores de dispositivo de componentes hardware, arrancar monitores del sistema y analizar los datos en tiempo real y en modo agregado.

C4: Reconocer y describir codificaciones y nomenclaturas de elementos informáticos de acuerdo con los criterios de estandarización más extendidos.

CE4.1 Describir y aplicar la normativa referente a la nomenclatura y clasificación de ficheros y sus contenedores requerida para facilitar la salvaguarda y administración de los datos del sistema.

CE4.2 Describir y emplear normativas de nomenclatura estandarizada de máquinas, servicios y aplicaciones requerida para facilitar las tareas de administración.

CE4.3 Reconocer y aplicar las políticas de migración y archivado de ficheros que se han de utilizar en la gestión de almacenamiento del sistema en función de su necesidad de proceso posterior y de la eficiencia de uso de recursos.

CE4.4 En varios supuestos y casos prácticos debidamente caracterizados de servidores conectados a diferentes redes de comunicaciones TCP/IP:

- Generar un mapa de direcciones IP de redes y servidores
- Definir e implantar un servidor de nombres (DNS)

CE4.5 A partir de un supuesto práctico convenientemente caracterizado de arquitectura de sistemas de archivo:

- Analizar y explicar la estructura implementada
- Identificar las fechas de creación, vigencia y última modificación de un conjunto característico de archivos
- Identificar los usuarios autorizados para abrir y modificar un conjunto característico de archivos
- Aplicar diferentes políticas de migración de datos analizando su influencia en la disponibilidad de espacio y en el tiempo de ejecución de procesos

C5: Distinguir los diferentes tipos de almacenamiento usados en los sistemas operativos multiusuario indicando su estructura, características y modos de operación.

CE5.1 Enumerar y clasificar los diferentes sistemas de almacenamiento en función de su capacidad, características de rendimiento y compatibilidad con los sistemas operativos más extendidos.

CE5.2 Describir y clasificar los mecanismos de protección y recuperación física de la información en función de su modo de funcionamiento y rendimiento.

CE5.3 Enumerar y analizar las agrupaciones de volúmenes, volúmenes lógicos y tipos de formato que se definen e implementan en cada sistema operativo y gestor de volúmenes.

CE5.4 Escoger y emplear las herramientas de gestión de volúmenes lógicos que se usan para la administración de almacenamiento sobre la base de su modo de funcionamiento y por su compatibilidad con los diferentes sistemas operativos.

CE5.5 Explicar como funciona y que valor aporta para el sistema operativo el acceso en paralelo a múltiples volúmenes físicos.

CE5.6 Enumerar los sistemas de almacenamiento en cinta y cartucho y clasificarlos por tipo de soporte, por su gestión manual o automática y por su uso en los sistemas operativos y aplicaciones.

CE5.7 En varios supuestos y casos prácticos debidamente caracterizados de sistemas con almacenamiento externo e interno y librerías de cintas:

- Documentar un mapa físico / lógico de capacidades que defina: Volúmenes físicos con su capacidad, dirección y modo de acceso, unidades de cinta y sus etiquetas, protecciones de paridad implementada y número de accesos a cada volumen.
- Definir volúmenes lógicos y sistemas de archivo con diferentes tamaños y estructura.
- Instalar y configurar un sistema de balanceo de accesos tolerante a fallos.
- Definir acceso en paralelo a sistemas de archivo y analizar el impacto en el rendimiento del sistemas usando las herramientas de monitorización del sistema operativo.
- Implementar con el gestor de volúmenes lógicos el espejado de volúmenes por software y analizar su utilidad para la recuperación del sistema operativo.

C6: Usar los principales tipos de herramientas ofimáticas y los servicios y aplicaciones asociados a Internet.

CE6.1 Enumerar y explicar las funciones principales de los procesadores de texto, hojas de cálculo y edición de presentaciones.

CE6.2 Relacionar los principales servicios asociados a Internet y clasificarlos sobre la base de su función y especificidad.

CE6.3 Aplicar las funciones de las herramientas ofimáticas y servicios Internet a la elaboración de documentación técnica debidamente estructurada y estandarizada para facilitar la comprensión y el control de versiones.

CE6.4 Utilizar los servicios de transferencia de ficheros para el intercambio de información con los servicios de soporte que los fabricantes de tecnologías de la información publican en Internet.

CE6.5 A partir de un supuesto práctico convenientemente caracterizado de sistema microinformático con posibilidad de conexión a Internet:

- Elaborar documentos de texto, hojas de cálculo y presentaciones a partir de una serie de modelos entregados y que requieren el uso de diferentes funcionalidades de las herramientas en dificultad creciente.
- Configurar el equipo para su acceso a Internet a partir de las especificaciones del Proveedor de Servicios.
- Encontrar y extraer documentación técnica y aplicaciones de diferentes proveedores de servicios en Internet a partir de una relación de situaciones planteadas.

C7: Aplicar técnicas y procedimientos relacionados con la seguridad de sistemas, redes de comunicaciones y

datos.

CE7.1 Explicar los conceptos fundamentales de las políticas de seguridad y protección de datos y su relación en la recuperación y continuidad de servicios y aplicaciones.

CE7.2 Explicar las diferencias entre copias de seguridad físicas y lógicas y su influencia en los sistemas operativos, sistemas de ficheros y bases de datos.

CE7.3 Identificar las principales arquitecturas de alta disponibilidad de sistemas y componentes y analizar sus ventajas y debilidades en función de cada caso.

CE7.4 Explicar el modo de funcionamiento de los cortafuegos, antivirus y proxys en las arquitecturas de redes de comunicaciones.

CE7.5 Reconocer las técnicas y procedimientos operativos empleados para garantizar la seguridad en los accesos de usuario a los servicios y aplicaciones con especial interés en las arquitecturas relacionadas con Internet.

CE7.6 En varios supuestos y casos prácticos debidamente caracterizados de sistemas informáticos conectados a redes de comunicaciones.

- Implementar copias de seguridad de ficheros y bases de datos.
- Recuperar aplicaciones que usen bases de datos a partir de copias de seguridad físicas e incrementales y especificaciones de continuidad de las mismas.
- Instalar y configurar cortafuegos en los servidores que sólo permitan el acceso desde los clientes y protocolos especificados.

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

C3 respecto a:

- Descripción de las diferentes maneras de monitorizar y ajustar los componentes de un sistema operativo y analizar las tendencias a partir del estado de carga
- Correlación de alarmas enviadas por el sistema de monitorización previamente implementado y definición de eventos para su resolución
- Realización de diferentes funciones relacionadas con instalación y configuración del sistema operativo.

C4 respecto a:

- Reconocimiento y aplicación de las políticas de migración y archivado de ficheros que se han de utilizar en la gestión de almacenamiento del sistema,
- Generación de mapas de direcciones IP de redes y servidores y definición e implantación de un servidor de nombres
- Diferentes supuestos relacionados con la arquitectura de sistemas de archivo

C5 Respecto a la caracterización de sistemas de almacenamiento.

C7 Respecto a la aplicación de técnicas y procedimientos relacionados con la seguridad informática.

Contenidos

Conceptos de ordenadores y servidores de propósito general

La estructura y componentes principales:

- Procesador (Set de Instrucciones, Registros, Contador, Unidad Aritmético-Lógica, Interrupciones).
- Memorias RAM y xPROM.
- Interfaces de entrada/salida:
- Discos.

Familias y tipos de procesadores. Evolución histórica.

Tipos de periféricos.

Sistemas operativos

Conceptos Generales. Tipos de clasificación

Principales funciones:

- Manejo de la memoria. Memoria Virtual y paginación.

- Políticas de reparto de tiempo de proceso.
- Entrada/salida. Manejadores de interrupciones y dispositivos.
- Bloqueo de recursos.
- Sistemas de archivo.
- Multiproceso y multiusuario.
- Organización de usuarios.

Particionamiento lógico y núcleos virtuales.

Técnicas de configuración y ajuste de sistemas

Rendimiento de los sistemas.

Consumo de recursos y competencia.

Modelos predictivos y análisis de tendencias.

Planes de pruebas preproducción.

Organización y gestión de la información

Sistemas de archivo:

- Nomenclatura y codificación.
- Jerarquías de almacenamiento.
- Migraciones y archivado de datos.

Volúmenes lógicos y físicos:

- Particionamiento.
- Sistemas NAS y SAN.
- Gestión de volúmenes lógicos.
- Acceso paralelo.
- Protección RAID.

Políticas de Salvaguarda:

- Salvaguarda física y lógica.
- Conceptos de Alta Disponibilidad. Cluster y balanceo de carga.
- Integridad de datos y recuperación de servicio.
- Custodia de ficheros de seguridad.

Políticas de Seguridad:

- Acceso restringido por cuentas de usuario. Propiedad de la información.
- Identificador único de acceso.
- Protección antivirus.
- Auditorias de seguridad.
- Cortafuegos y servidores proxy.

Aplicaciones microinformáticas e Internet

Procesadores de Texto, Hojas de Cálculo y Edición de Presentaciones:

- Manejo y conocimiento a nivel de usuario.
- Técnicas de elaboración de documentación técnica.
- Formatos de documento. Estructura de la información.

Uso de Internet:

- Conocimiento de www. Navegadores.
- Sistemas de correo electrónico, chat y foros.
- Transferencia de ficheros.

Requisitos básicos del contexto formativo

Espacios e instalaciones:

Aula de informática de 45 m2.

Perfil profesional del formador:

1. Dominio de los conocimientos y las técnicas relacionadas con configurar y explotar sistemas, en lengua propia y extranjera, que se acreditará mediante una de las formas siguientes:

- Formación académica de Licenciado o Ingeniero y de otras de igual nivel relacionadas con este 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 Programación de bases de datos relacionales
Nivel	3
Código	MF0226_3
Asociado a la UC	Programar bases de datos relacionales.
Duración horas	210

Capacidades y criterios de evaluación

- C1: Comprender y aplicar los fundamentos conceptuales y las técnicas de las bases de datos relacionales.**
- CE1.1 Describir los fundamentos y objetivos del modelo relacional.
 - CE1.2 Enumerar y describir los principales elementos del modelo de datos relacional: relaciones/tablas, atributos, claves principales, claves ajenas, índices, vistas.
 - CE1.3 Enumerar los tipos de restricciones asociados a las claves.
 - CE1.4 Explicar el concepto de dependencia funcional y enumerar los tipos existentes.
 - CE1.5 Explicar los objetivos de la teoría de la normalización y describir las diferentes formas normales: 1FN, 2FN, 3FN, 4FN y 5FN.
 - CE1.6 Explicar las razones por las que se procede a la desnormalización de los modelos de datos.
 - CE1.7 En un supuesto práctico de estudio de un diseño lógico de una base de datos relacional:
 - Identificar las tablas, claves primarias y ajenas, índices y vistas.
 - Reconocer el grado de normalización de las tablas de la base de datos.
 - Justificar las posibles desnormalizaciones del modelo.
 - Reconocer el dominio de los atributos de las tablas indicando el rango o conjunto de valores que pueden tomar.
 - Indicar las restricciones de integridad asociadas a cada una de las claves primarias.
 - Indicar las restricciones de integridad asociadas a las claves ajenas, describiendo en cada caso cómo se comportan los borrados o modificaciones realizados sobre las mismas (restricción de la acción, propagación de la acción, anulación de las claves en registros relacionados).
- C2: Determinar los elementos de la base de datos que se han de manipular, mediante la interpretación del diseño de la base de datos y el análisis de los requisitos de usuario.**
- CE2.1 Explicar el concepto de diccionario de datos y su estructura (tablas y variables auxiliares para la manipulación del mismo).
 - CE2.2 Enumerar las herramientas del sistema de bases de datos para la consulta y manipulación del diccionario de datos.
 - CE2.3 Enumerar los principales modelos para la obtención de esquemas conceptuales de la base de datos.
 - CE2.4 Describir la simbología asociada al modelo conceptual entidad-relación.
 - CE2.5 Explicar la necesidad del control de calidad dentro del ciclo de vida de un proyecto.
 - CE2.6 Enumerar las principales estrategias para realizar el seguimiento de los requisitos de usuario, concretando las específicas para la fase de desarrollo de software.
 - CE2.7 En un supuesto práctico de estudio del diseño de la BBDD y de los requisitos de usuario:
 - Identificar las funcionalidades a desarrollar a partir de los requisitos de usuario.
 - Identificar los elementos de la BBDD a manipular para cada funcionalidad y localizarlos en el esquema conceptual.
 - Utilizar el diccionario de datos para observar las particularidades de los elementos de la BBDD a manipular.
 - Documentar los elementos de la BBDD que van a ser utilizados para cada funcionalidad para facilitar el seguimiento de los requisitos de usuario.
 - Identificar las necesidades de definición de nuevos elementos en la BBDD: tablas auxiliares, vistas, índices.
 - Documentar los nuevos elementos de la BBDD para su posterior creación.
- C3: Formular consultas de manipulación y definición de datos, a partir del diseño de la BBDD y de los requisitos de usuario**
- CE3.1 Explicar los fundamentos del álgebra y cálculo relacional y enumerar y diferenciar los lenguajes asociados a la base de datos.
 - CE3.2 Explicar el tipo de consultas (de selección, de actualización, de inserción, de borrado) que se pueden realizar utilizando el lenguaje DML.
 - CE3.3 Explicar el tipo de elementos que se pueden crear y manipular utilizando el lenguaje DDL.
 - CE3.4 Describir la sintaxis de un lenguaje de consultas relacional..
 - CE3.5 Explicar el concepto de vista y describir su utilidad.
 - CE3.6 Indicar las extensiones del lenguaje de consultas relacional para especificar restricciones de integridad, para definir control de acceso a los elementos de la BBDD y para controlar la ejecución de las transacciones.
 - CE3.7 Enumerar y describir las herramientas de la BBDD para realizar formulaciones de manipulación y definición de datos de forma interactiva.

CE3.8 Describir las herramientas de la base de datos para la optimización de consultas.

CE3.9 En un supuesto práctico de realización de formulaciones de manipulación de datos, a partir del diseño de la base de datos y de los requisitos de usuario:

- Seleccionar el lenguaje adecuado para realizar la codificación.
- Seleccionar la herramienta de la BBDD adecuada para la ejecución interactiva de la formulación codificada.
- Utilizar el lenguaje DML para construir la formulación de manipulación de datos.
- Probar la formulación de manipulación en un entorno controlado que interfiera lo mínimo posible con el sistema.
- Utilizar las facilidades del lenguaje de consultas relacional para el control de la ejecución de las transacciones, garantizando la integridad de los datos de la BBDD.
- Analizar los resultados obtenidos en la ejecución y realizar las modificaciones necesarias en el código para corregir posibles fallos de funcionamiento.
- Optimizar las consultas codificadas utilizando las herramientas de la base de datos
- Documentar el código realizado y las pruebas para facilitar el seguimiento de los requisitos.

CE3.10 En un supuesto práctico de realización de formulaciones de definición de datos, a partir del diseño de la base de datos y de los requisitos de usuario:

- Seleccionar el lenguaje adecuado para realizar la codificación.
- Seleccionar la herramienta de la BBDD adecuada para la ejecución interactiva de la formulación codificada.
- Utilizar el lenguaje DDL para construir la formulación de definición de datos.
- Comprobar que los elementos creados cumplen las especificaciones del diseño.

C4: Formular consultas utilizando el lenguaje de programación de la base de datos, a partir del diseño de la base de datos y de los requisitos de usuario.

CE4.1 Enumerar y describir los entornos de desarrollo integrados disponibles en el sistema de gestión de bases de datos.

CE4.2 Enumerar los lenguajes de programación disponibles en los entornos de desarrollo.

CE4.3 Describir la sintaxis de un lenguaje de programación disponible en un entorno integrado en la base de datos. Detallar las características generales del mismo: tipos de variables, tipos de datos, estructuras de control, librerías de funciones.

CE4.4 Enumerar y describir las posibles herramientas para el desarrollo de entornos gráficos de usuario integradas en el ámbito de la base de datos.

CE4.5 Enumerar y describir las utilidades para la depuración y control de código disponibles en el entorno de la base de datos.

CE4.6 Enumerar y describir las técnicas para el control de la ejecución de las transacciones.

CE4.7 Describir las herramientas de la base de datos para la optimización de consultas.

CE4.8 En un supuesto práctico de desarrollo de programas en el entorno de la base de datos, a partir del diseño de la misma y de los requisitos de usuario:

- Seleccionar el entorno de desarrollo adecuado según las necesidades del diseño.
- Seleccionar el lenguaje de programación adecuado según las necesidades de diseño.
- Codificar los módulos utilizando técnicas de programación según las especificaciones del diseño y los requisitos del usuario.
- Utilizar las herramientas para el desarrollo de entornos gráficos de usuario según los requisitos de usuario y el diseño de la base de datos.
- Seleccionar la técnica de control de transacciones más adecuada y utilizarla para garantizar la integridad de los datos de la BBDD.
- Probar los módulos desarrollados en ambientes controlados y que no interfieran con el funcionamiento normal del sistema.
- Analizar los resultados de las pruebas y realizar las modificaciones del código oportunas para solucionar los posibles errores de funcionamiento.
- Optimizar las consultas utilizadas en los módulos utilizando las herramientas de la base de datos.
- Documentar los módulos desarrollados y las baterías de pruebas realizadas para facilitar el seguimiento de los requisitos de usuario.

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

C3 respecto a la formulación de manipulación y de definición de datos.

C4 respecto al desarrollo de programas en el entorno de la base de datos.

Contenidos

El ciclo de vida de un proyecto

- Conceptos generales acerca del análisis de aplicaciones.
- Conceptos generales acerca del diseño de aplicaciones.
- Conceptos generales del control de calidad:
 - . Control de calidad de las especificaciones funcionales.
 - . Seguimiento de los requisitos de usuario.

Introducción a las bases de datos

- Evolución histórica de las bases de datos.
- Ventajas e inconvenientes de las bases de datos.

Fundamentos del modelo relacional:

- Estructura del modelo relacional:
 - . El concepto de relación. Propiedades de las relaciones.
 - . Atributos y dominio de los atributos.
 - . Claves: claves candidatas, claves primarias, claves alternativas, claves ajenas.
- Restricciones de integridad: integridad de las entidades, integridad referencial.
- Teoría de normalización:
 - . El proceso de normalización. Tipos de dependencias funcionales.
 - . Primera forma normal (1FN).
 - . Segunda forma normal (2FN).
 - . Tercera forma normal (3FN).
 - . Otras formas normales (4FN, 5FN).
 - . Desnormalización.
- Operaciones en el modelo relacional:
 - . Álgebra relacional: operaciones primitivas: selección, proyección, producto, unión y diferencia y otras operaciones: intersección, join, y división.
 - . Cálculo relacional: cálculo relacional de dominios y cálculo relacional de tuplas.
 - . Transformación de consultas entre álgebra y cálculo relacional.

El lenguaje de manipulación de la base de datos

- Tipos de lenguajes de manipulación relacionales.
- El lenguaje de definición de datos (DDL):
 - . Tipos de datos del lenguaje.
 - . Creación y borrado de tablas.
 - . Creación y borrado de índices.
- El lenguaje de manipulación de datos (DML):
 - . Construcción de consultas de selección
 - . Construcción de consultas de inserción
 - . Construcción de consultas de modificación
 - . Construcción de consultas de borrado

- Cláusulas del lenguaje para la agrupación y ordenación de las consultas.
- Capacidades aritméticas, lógicas y de comparación del lenguaje.
- Funciones agregadas del lenguaje.
- Tratamiento de valores nulos.
- Construcción de consultas anidadas.
- Unión, intersección y diferencia de consultas.
- Consultas de tablas cruzadas.
- Otras cláusulas del lenguaje.
- Extensiones del lenguaje:
 - . Creación, manipulación y borrado de vistas
 - . Especificación de restricciones de integridad
 - . Instrucciones de autorización
 - . Control de las transacciones
- Propiedades de las transacciones: atomicidad, consistencia, aislamiento y permanencia:
 - . Estados de una transacción: activa, parcialmente comprometida, fallida, abortada y comprometida.
 - . Consultas y almacenamiento de estructuras en XML.
 - . Estructura del diccionario de datos.
- Herramientas de la BBDD para la optimización de consultas.

Modelos conceptuales de bases de datos

- El modelo entidad-relación:
 - . Entidades, relaciones y atributos
 - . Diagramas entidad-relación
- El modelo entidad-relación extendido

Lenguajes de programación de bases de datos

- Entornos de desarrollo en el entorno de la base de datos.
- La sintaxis del lenguaje de programación: variables, tipos de datos, estructuras de control, librerías de funciones.
- Herramientas de depuración y control de código.
- Facilidades para el desarrollo de entornos gráficos.

Requisitos básicos del contexto formativo

Espacios e instalaciones:

Aula de informática con una superficie de 45 m2.

Perfil profesional del formador:

1. Dominio de los conocimientos y las técnicas relacionadas con programar bases de datos relacionales, en lengua propia y extranjera, que se acreditará mediante una de las formas siguientes:

- Formación académica de Licenciado o Ingeniero y de otras de igual nivel relacionadas con este 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	Desarrollar componentes software en lenguajes de programación orientados a objetos.
Duración horas	250

Capacidades y criterios de evaluación

C1: Dominar los conceptos fundamentales del paradigma Orientado a Objetos.

CE1.1 Explicar las características del ciclo de desarrollo del software bajo el paradigma de orientación a objetos, distinguiendo la programación orientada a objetos como una fase dentro del mismo.

CE1.2 Describir y enumerar las características de una clase: atributos, métodos y mecanismo de encapsulación, identificando la interfaz de la clase y lo que representa.

CE1.3 Describir y enumerar las características que definen un objeto, distinguiendo las diferencias entre los conceptos de objeto y clase.

CE1.4 Describir la estructura y el significado de los mensajes y su relación con el comportamiento de los objetos.

CE1.5 Explicar las características fundamentales que tienen que estar presentes en una relación entre dos clases para que pueda ser calificada como relación de herencia.

CE1.6 Describir el mecanismo de herencia múltiple y los problemas que presenta en el proceso de desarrollo de software.

CE1.7 Explicar el concepto de polimorfismo y enumerar y describir las características que introduce en el proceso de desarrollo del software.

CE1.8 En un supuesto práctico, a partir de una documentación típica de diseño detallado, identificar las clases establecidas, los atributos y las relaciones.

C2: Desarrollar clases aplicando los fundamentos del paradigma Orientado a Objetos.

CE2.1 Enumerar y describir los principales criterios de calidad del software y los principales factores evaluados por las métricas orientadas a objetos.

CE2.2 Enumerar y describir los mecanismos de gestión de memoria utilizados en la creación y destrucción de los objetos.

CE2.3 Describir los mecanismos existentes para realizar la implementación de las relaciones entre clases.(Clases contenedores, objetos colección, etc).

CE2.4 Explicar la utilización de los objetos "super" y "this" ("current", "self" u otros), en relación con el acceso a los atributos definidos en una clase, desde una subclase o desde el código de la propia clase.

CE2.5 Clasificar los diferentes lenguajes de programación, identificando y reconociendo en los mismos las principales características del paradigma orientado a objetos: Clases, objetos, herencia y polimorfismo.

CE2.6 Distinguir y utilizar las características proporcionadas por un entorno de desarrollo asociado a un lenguaje Orientado a Objetos.

CE2.7 Distinguir las estructuras de datos más habituales (listas, pilas, árboles, grafos, etc) y los posibles mecanismos de construcción en los lenguajes orientados a objetos.

CE2.8 Distinguir las librerías de clases estándares del lenguaje de programación conociendo la utilidad de cada una de ellas y la forma básica de uso.

CE2.9 En un supuesto práctico, construir las clases que representan las estructuras de datos en un lenguaje orientado a objetos.

CE2.10 En un supuesto práctico, en el que se pide realizar la programación de una clase con un lenguaje orientado a objetos y desde una documentación a nivel de diseño detallado:

- 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

C3: Aplicar los conceptos básicos del modelo de programación web.

- CE3.1 Enumerar y describir los componentes del modelo multicapa de programación web. (Cliente ligero, servidores web, servidores de aplicaciones, servidores de base de datos).
- CE3.2 Enumerar y describir la función de los protocolos y tecnologías habituales. (TCP/IP, http, HTML, XML, XSL, SOAP).
- CE3.3 Enumerar las características básicas de los modelos de programación ampliamente utilizados (J2EE y .NET).
- CE3.4 Describir las capas lógicas de una aplicación web (Presentación, Aplicación y Datos).
- CE3.5 Describir las características básicas del lenguaje de presentación (HTML).
- CE3.6 Describir las características básicas de los lenguajes de scripting en cliente (JavaScript, VBScript).
- CE3.7 Describir el funcionamiento de una sesión de aplicación en el modelo de programación web.
- CE3.8 Aplicar las características básicas de los lenguajes orientados a objetos a la recepción de solicitudes y preparación de la capa de presentación (JSP, ASP, Servlets, PHP).

C4: Verificar la corrección de las clases desarrolladas mediante la realización de pruebas.

- CE4.1 Enumerar y describir las herramientas y utilidades más comunes para la depuración de programas.
- CE4.2 Enumerar y describir los tipos de pruebas posibles que se pueden dar en el proceso de desarrollo de aplicaciones, distinguiendo especialmente aquellas que son responsabilidad del programador.
- CE4.3 Enumerar y describir los tipos de pruebas que se deben realizar a una clase para verificar su corrección.
- CE4.4 Utilizar las características proporcionadas por un entorno de desarrollo para realizar la depuración de un programa, mediante:
 - Ejecución paso a paso.
 - Establecimiento de puntos de parada (condicionales o incondicionales).
 - Monitorización de variables.
- CE4.5 En supuestos prácticos, documentar una estrategia de pruebas completa a una clase a partir de su documentación de diseño, de forma que se asegure el óptimo funcionamiento en aspectos como:
 - Coherencia en el estado de los objetos.
 - Todos los escenarios posibles.
 - Rendimiento.
 - Casos límite.
 - Situaciones excepcionales.
- CE4.6 En supuestos prácticos, a partir de una clase y la estrategia definida de pruebas:
 - Establecer el conjunto de secuencias y estados iniciales de los objetos que intervienen en la secuencias.
 - Establecer el criterio de evaluación de los resultados.
 - Automatizar el proceso mediante programas de prueba.
- CE4.7 Realizar pruebas a una clase mediante herramientas de prueba.

C5: Elaborar la documentación completa relativa a las clases desarrolladas y pruebas realizadas.

- CE5.1 Describir la información que debe acompañar a una clase desarrollada.
- CE5.2 Describir los criterios fundamentales para la inclusión de cabeceras y comentarios en el código.
- CE5.3 Proponer índices (plantillas) para los documentos de diseño y explicar el contenido de cada uno de los apartados.
- CE5.4 Proponer índices (plantillas) para los documentos de pruebas (planificación y resultados) y explicar el contenido de cada uno de los apartados.
- CE5.5 Proponer índices (plantillas) para la documentación de operación y mantenimiento (manuales técnicos) y explicar el contenido de cada uno de los apartados.
- CE5.6 Aplicar unos criterios de normalización establecidos para incluir cabeceras y comentarios en el código.
- CE5.7 En un supuesto práctico, a partir de una clase desarrollada, elaborar la documentación de diseño de la clase de acuerdo a un índice establecido.
- CE5.8 En un supuesto práctico, a partir de una clase y conjunto de pruebas, elaborar la documentación de pruebas (planificación y resultados) de acuerdo a un índice establecido.
- CE5.9 En un supuesto práctico, a partir de la documentación de desarrollo y pruebas, elaborar la documentación de operación y mantenimiento de acuerdo a un índice establecido.
- CE5.10 Realizar la documentación de las clases mediante herramientas de documentación automática.

C6: Realizar modificaciones de clases existentes por cambios en las especificaciones.

- CE6.1 Enumerar y describir los conceptos fundamentales de la gestión de la configuración del software desarrollado por una organización.
- CE6.2 Enumerar y describir los conceptos fundamentales de la gestión de la configuración de la documentación.
- CE6.3 Describir los pasos y precauciones fundamentales en el proceso de modificación de clases existentes.
- CE6.4 En un supuesto práctico de modificación de un documento, y de acuerdo con un procedimiento:
 - Realizar la modificación en el documento.
 - Marcar las hojas cambiadas con la modificación y el código de revisión.
 - Elaborar la propuesta de cambio de acuerdo al formato especificado.
 - Editar la nueva edición o revisión del documento.

- CE6.5 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 siguiendo los criterios de calidad establecidos.
 - Incluir en el lugar de la modificación y en la cabecera del componente software los datos del histórico y la explicación de la modificación.
 - Modificar, si es necesario, los programas de pruebas asociados siguiendo los criterios de calidad establecidos.
 - Comprobar, mediante la realización de pruebas, que la modificación ha sido incorporada con éxito y que no ha alterado la funcionalidad del resto de la clase.
 - Realizar la actualización de toda la documentación que se vea afectada por el cambio propuesto.

CE6.6 Utilizar herramientas para la gestión de la configuración y las versiones del software.

C7: Realizar conexiones con bases de datos relacionales.

CE7.1 Enumerar y describir las diferentes tecnologías de conexión con la BBDD desde las aplicaciones.

CE7.2 Analizar las diferentes tecnologías de conexión y acceso a datos, determinando las que se deben utilizar para la manipulación del sistema de base de datos.

CE7.3 Enumerar y describir las clases que proporcionan los medios adecuados para efectuar consultas, actualizaciones, acceder y operar con una base de datos relacional.

CE7.4 Describir los procedimientos para realizar dichas consultas (abrir y cerrar conexiones, ejecutar comandos, recoger sus resultados y utilizarlos).

CE7.5 En supuestos prácticos debidamente caracterizados de incorporación de un acceso a una base de datos relacional desde una clase, a partir de un diseño:

- Seleccionar la tecnología de conexión más adecuada según los requisitos de diseño y el entorno de trabajo y las características del lenguaje orientado a objetos que se vaya a emplear.
- Cargar el controlador (en el caso que sea necesario) de conexión de la base de datos.
- Realizar la conexión entre el programa y la base de datos usando las clases de la librería de acceso a la base de datos.
- Desarrollar la clase, según la especificación, realizando consultas simples en la base de datos y disponiendo los resultados para su tratamiento por la aplicación, siguiendo los criterios de calidad que se establezcan.
- Comprobar el funcionamiento correcto de la clase desarrollada mediante pruebas.
- Realizar la documentación asociada a la clase desarrollada.

CE7.6 En supuestos prácticos debidamente caracterizados de modificación de una aplicación con acceso a una base de datos relacional desde una clase, a partir de un cambio en el diseño de la aplicación o de la base de datos:

- Modificar el código de la clase para incorporar el cambio siguiendo los criterios de calidad establecidos.
- Incluir en el lugar de la modificación y en la cabecera del componente software los datos del histórico y la explicación de la modificación.
- Modificar, si es necesario, los programas de pruebas asociados siguiendo los criterios de calidad establecidos.
- Comprobar, mediante la realización de pruebas, que la modificación ha sido incorporada con éxito y que no ha alterado la funcionalidad del resto de la clase.
- Realizar la actualización de toda la documentación que se vea afectada por el cambio propuesto.

C8: Desarrollar interfaces de usuario en lenguajes de programación orientados a objeto, a partir del diseño detallado.

CE8.1 Interpretar diagramas de ¿casos de uso¿ y analizar las necesidades y peticiones de usuarios.

CE8.2 Identificar y describir las clases básicas que se usan para el interfaz hombre / máquina.

CE8.3 Explicar el concepto de evento y de programación orientada a eventos y su implementación en los lenguajes orientados a objetos en relación con las clases necesarias para el desarrollo de la interfaz.

CE8.4 Identificar los recursos multimedia que pueden incluirse en una ventana de interfaz de usuario.

CE8.5 Realizar el diseño de las ventanas correspondientes a la interfaz de usuario, aplicando criterios de ergonomía, eficacia y posible utilización por discapacitados, en la comunicación de información.

CE8.6 En un supuesto práctico de construcción de un interfaz de usuario:

- Identificar los elementos básicos que constituyen la interfaz, en la documentación de diseño proporcionada.
- Asociar a cada elemento identificado la clase adecuada de la librería de clases disponibles.
- Construir la interfaz con una herramienta de diseño de interfaz gráfica.
- Identificar y describir el código fuente generado por la herramienta de interfaz gráfica.

CE8.7 En un supuesto práctico de construcción de una aplicación con interfaz de usuario, a partir del código generado por una herramienta de interfaz gráfica:

- Incluir el código necesario para incorporar otros recursos multimedia a partir de las librerías de clases disponibles y siguiendo los criterios de calidad que se establezcan en un lenguaje orientado a objetos.
- Realizar la programación de las clases necesarias para conectar la interfaz de usuario con la aplicación, siguiendo los criterios de calidad que se establezcan y utilizando las características proporcionadas por un lenguaje de programación orientado a objetos.

CE8.8 En un supuesto práctico, sobre una interfaz de usuario desarrollada y con la documentación correspondiente al diseño detallado:

- Evaluar que los servicios de presentación implementados cumplen con las necesidades definidas por el usuario y utilizan de forma óptima los recursos del sistema.
- Establecer criterios de validación de los servicios de presentación.
- Describir diversos tipos de errores en función de la interfaz.
- Comprobar que los formatos de entrada y salida son los esperados conforme a las especificaciones de diseño.
- Verificar que la realización de operaciones indebidas no altera la fiabilidad del sistema.
- Elaborar la documentación de la interfaz.
- Redactar la guía de usuario correspondiente.

C9: Manejar las herramientas de ingeniería de software.

CE9.1 Enumerar y comparar los modelos de ingeniería software, indicando los conceptos principales en los que se basan, su ámbito de uso y cómo se estructuran.

CE9.2 Describir las fases, en cada modelo, del proceso de ingeniería de software, indicando para cada una:

- Datos de partida (entradas).
- Datos finales (salidas).
- Funciones realizadas en la fase.
- Documentación generada.
- Trazabilidad.

CE9.3 Describir en detalle los conceptos fundamentales de una metodología de ingeniería software basada en la orientación a objetos.

CE9.4 Identificar en las herramientas de desarrollo orientada a objetos disponibles, los diferentes componentes y describir cómo se implementan los conceptos de la metodología y las distintas fases del proceso de ingeniería de software, haciendo especial hincapié en los procesos de:

- Diseño.
- Codificación.
- Pruebas unitarias.
- Documentación.
- Evaluación de la calidad y métricas.
- Gestión de la Configuración.
- Cambios.

CE9.5 En un supuesto práctico utilizar las herramientas de desarrollo, en el caso de que sea posible, para:

- Extraer la información relativa al diseño de una clase.
- Codificar la clase.
- Realizar su depuración.
- Incorporar la definición y la implementación de las pruebas unitarias.
- Trazar las pruebas con las clases.
- Configurar la herramienta para la realización de baterías de pruebas automáticas.
- Elaborar documentación mediante las plantillas incorporadas en la herramienta.
- Realizar modificaciones a las plantillas incorporadas para la elaboración de la documentación.
- Generar informes de calidad y métricas e interpretar los resultados.
- Incorporar cambios.

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

C2 respecto a la programación de una clase con un lenguaje orientado a objetos.

C3 respecto a la aplicación de los lenguajes orientados a objetos a la recepción de solicitudes y preparación de la capa de presentación.

C4 respecto a la verificación de las clases mediante realización de pruebas.

C5 respecto a la elaboración de la documentación relativa a las clases y pruebas establecidas.

C6 respecto a la modificación de documentos y clases por cambios de distinta índole.

C7 respecto a la conexiones de bases de datos orientadas a objetos con bases de datos relacionales.

C8 respecto a la construcción de una interfaz de usuario vinculada a una aplicación.

C9 respecto al manejo de las herramientas de ingeniería de software.

Contenidos

Introducción al paradigma orientado 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 comportamientos que pueden solicitarse a los objetos. Comportamiento de un objeto: Concepto de mensaje.

Clases y objetos

- Clase: atributos, métodos y mecanismo de encapsulación. Interfaz de la clase. Clases y tipos de datos.
- Objetos: Estado, comportamiento e identidad. Mensajes. Tipos de métodos. Clasificación: Métodos de acceso, de selección o consulta, de construcción, de destrucción.
- Objetos como instancias de clase. Instancia actual (this, self, current).
- Concepto de programa en el paradigma orientado a objetos. POO = Objetos + Mensajes.

Técnicas de programación estructurada:

- Elementos básicos: constantes, variables, operadores y expresiones.
- 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.

Estructura de la información

- 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.

Mecanismos de gestión de memoria

- Gestión automática de memoria.
- Construcción y destrucción de objetos. Objetos inalcanzables. Recolección de "basura".
- Métodos constructores y destructores.

Relaciones entre clases

- Agregación/Composición.
- Generalización / Especialización.
- Asociación.

Generalización/Especialización: herencia

- Herencia: Simple y múltiple. Relación de herencia: Características. Reglas y características que definen una relación de herencia: Regla "Es-un". Transmisión de atributos y métodos. Regla de especialización de la superclase en la subclase. Acceso a los atributos de una clase y acoplamiento entre las clases. Utilización de objetos this (current, self u otros) y super. Leyes de Demeter.
- Herencia múltiple. Problemas: Conflictos de nombres, herencia repetida. Soluciones.
- Creación de objetos en la herencia.
- Clasificación jerárquica de las clases. Clase raíz. Clases abstractas. Métodos virtuales. Redefinición de métodos.

Polimorfismo

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

Lenguajes de programación orientados a objetos

- Lenguaje de programación orientado a objetos y paradigma orientado a objetos. Lenguajes de programación orientados a objetos. Lenguajes de programación basados en objetos. Lenguajes de programación que utilizan objetos.
- Comparación entre los lenguajes de programación orientados a objetos más habituales. Características esenciales.

Excepciones

- Definición. Fuentes de excepciones. Tratamiento de excepciones. Prevención de fallos. Excepciones definidas y lanzadas por el programador.
- Las excepciones tratadas como objetos.

Librerías de clases

- Estructura.
- Creación y utilización.

Implementación del paradigma utilizando un lenguaje de programación orientado a objetos

- Elección del lenguaje.
- Tipos de aplicaciones.
- Herramientas de desarrollo.
- Tipos de datos y elementos básicos característicos del lenguaje. Instrucciones.
- Estudio y utilización de las clases básicas incluidas en la librería de clases.
- Definición de clases:
 - . Construcción de métodos. Sobrecarga.
 - . Construcción de atributos.
 - . Construcción de la interfaz de la clase.
 - . Construcción de clases incluyendo relaciones de Agregación /Composición y Asociación.
 - . Construcción de clases con herencia.
 - . Construcción de clases con herencia múltiple.
 - . Definición de clases abstractas.
 - . Construcción de clases con herencia incluyendo polimorfismo.
- Excepciones.
 - Gestión de eventos. Eventos, fuentes y auditores de eventos. Tipos de eventos. Mecanismos de gestión de eventos. Librerías de clases asociadas.
 - Hilos. Fundamentos. Creación. Prioridad. Comunicación. Sincronización. Estados. Creación y ejecución de hilos en el lenguaje. Librerías de clases asociadas. Programación multihilo.
 - Programación en red. Aplicaciones cliente y servidor. Sockets.
 - Acceso a bases de datos desde las aplicaciones. Librerías de clases asociadas.

Introducción al desarrollo de aplicaciones en el Modelo de programación web

- La arquitectura multicapa: Cliente ligero, servidor web, servidor de aplicaciones, servidor de datos.
- El modelo de tres capas en web: presentación, aplicación y datos.
- Protocolos y tecnologías habituales.
- Modelos de programación estándares de facto.
- La capa de presentación: El lenguaje de hipertexto.
- La capa de presentación avanzada: Lenguajes de scripting y lenguaje de hipertexto dinámico.

- Lenguajes orientados a la preparación de la capa de presentación y a la ejecución de solicitudes desde clientes ligeros web. (JSP, Servlets, ASP, PHP).
- Uso de componentes orientados a objeto como base en el desarrollo de aplicaciones en el modelo de programación web.

Acceso a bases de datos relacionales

- Bases de datos relacionales.
- Diseño de bases de datos en varios niveles.
- Lenguaje de acceso a base de datos.
- API de acceso a la base de datos.
 - . Nivel controlador.
 - . Interfaz de acceso a la base de datos (driver).
- Nivel aplicación:
 - . Establecimiento de la conexión con una base de datos.
 - . Operar sobre la base de datos. Sentencias del lenguaje de acceso a base de datos. Objetos que permiten ejecutar una consulta. Objetos que permiten manipular el resultado de una consulta.
 - . 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.

Calidad en el desarrollo del software

- Criterios de calidad.
- Métricas y estándares de calidad.

Documentación

- Como producir un documento.
- Estructura del documento.
- Generación automática de documentación.

Pruebas

- Tipos de pruebas.
- Pruebas de defectos: Pruebas de caja negra. Pruebas estructurales. Pruebas de trayectorias. Pruebas

de integración. Pruebas de interfaces. Preparación de los datos de prueba. Casos de pruebas. Codificar las pruebas. Definir procesos de pruebas. Ejecución de pruebas. Generación de informes de las pruebas.

Proceso de ingeniería del software

- Fases del proceso de ingeniería software: especificación, diseño, construcción y pruebas unitarias, validación, implantación y mantenimiento.
- Modelos del proceso de ingeniería: modelo en cascada, desarrollo evolutivo, desarrollos formales, etc.
- Requisitos: concepto, evolución y trazabilidad.
- Metodologías de desarrollo orientadas a objeto. Caso práctico: Metodologías de desarrollo que utilizan UML.
- Herramientas CASE: herramientas de ingeniería software, entornos de desarrollo, herramientas de prueba, de gestión de la configuración, herramientas para métricas.

Requisitos básicos del contexto formativo

Espacios e instalaciones:

Aula de informática con una superficie de 45 m².

Perfil profesional del formador:

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 formas siguientes:

- Formación académica de Licenciado o Ingeniero y de otras de igual nivel relacionadas con este 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.