

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

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

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

ANEXO IV

I. IDENTIFICACIÓN DEL CERTIFICADO DE PROFESIONALIDAD

Denominación: PROGRAMACIÓN CON LENGUAJES ORIENTADOS A OBJETOS Y BASES DE DATOS RELACIONALES

Código: IFCD0112

Familia Profesional: Informática y comunicaciones

Área profesional: Desarrollo

Nivel de cualificación profesional: 3

Cualificación profesional de referencia: IFC080_3 Programación con lenguajes orientados a objetos y bases de datos relacionales (Real Decreto 295/2004, de 20 de febrero).

Relación de unidades de competencia que configuran el certificado de profesionalidad

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.

Competencia general

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

Entorno profesional:

Ámbito profesional:

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 sectores donde se desarrollan procesos de:

- 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 relacionados:

Técnico en data mining (minería de datos).
Programador de aplicaciones de gestión.

Duración de la formación asociada: 710 horas

Relación de módulos formativos y de unidades formativas

MF0223_3: (Transversal) Sistemas operativos y aplicaciones informáticas (170 horas)

- UF1465: Computadores para bases de datos (60 horas)
- UF1466: Sistemas de almacenamiento (70 horas)
- UF1467: Aplicaciones microinformáticas e Internet para consulta y generación de documentación (40 horas)

MF0226_3: (Transversal) Programación de bases de datos relacionales (210 horas):

- UF2175: Diseño de bases de datos relacionales (50 horas)
- UF2176: Definición y manipulación de datos (80 horas)
- UF2177: Desarrollo de programas en el entorno de la base de datos (80 horas)

MF0227_3: Programación orientada a objetos (250 horas):

- UF2404: Principios de la programación orientada a objetos (90 horas)
- UF2405: Modelo de programación web y bases de datos (80 horas)
- UF2406: El ciclo de vida del desarrollo de aplicaciones (80 horas)

MP0501: Módulo de prácticas profesionales no laborales de Programación con lenguajes orientados a objetos y bases de datos relacionales (80 horas)

II. PERFIL PROFESIONAL DEL CERTIFICADO DE PROFESIONALIDAD

Unidad de competencia 1

Denominación: CONFIGURAR Y EXPLOTAR SISTEMAS INFORMÁTICOS

Nivel: 3

Código: UC0223_3

Realizaciones profesionales y criterios de realización

RP1: adaptar la configuración lógica del sistema para su explotación, según las necesidades de uso y dentro de las directivas de la organización.

CR1.1 Los parámetros del sistema que afectan a la memoria, procesador y periféricos se ajustan a las necesidades de uso.

CR1.2 Los dispositivos necesarios y sus ficheros de control se añaden o eliminan empleando para ello las utilidades del sistema operativo.

CR1.3 Las conexiones lógicas del equipo se configuran para acceder a servicios remotos dentro o fuera de la organización.

CR1.4 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.

RP2: 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.

CR2.1 Las aplicaciones informáticas se organizan con una estructura y configuración que permitan su uso en óptimas condiciones.

CR2.2 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.

CR2.3 La estructura y configuración del sistema de archivos se conservan en disposición de uso para evitar fallos accidentales y compartir información.

CR2.4 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.

RP3: Elaborar y transferir documentos mediante el uso de aplicaciones informáticas de propósito general.

CR3.1 Las herramientas ofimáticas se utilizan con la destreza necesaria para auxiliar en las tareas de planificación y documentación de los trabajos.

CR3.2 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.

CR3.3 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.

RP4: Garantizar la integridad, disponibilidad y confidencialidad de la información de la que se es responsable y se encuentre almacenada en el sistema de archivos.

CR4.1 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.

CR4.2 El acceso a la información se protege mediante el uso de claves y otras medidas de seguridad establecidas en la organización.

CR4.3 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.

CR4.4 El sistema se mantiene libre de software no licenciado.

CR4.5 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.

CR4.6 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

Denominación: PROGRAMAR BASES DE DATOS RELACIONALES

Nivel: 3

Código: UC0226_3

Realizaciones profesionales y criterios de realización

RP1: 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.

CR1.1 El diseño lógico y la estructura de la base de datos se conocen al nivel necesario para establecer relaciones entre os elementos de datos.

CR1.2 Las restricciones, reglas de integridad y semántica de los datos se identifican e interpretan para poder realizar correctamente las tareas de programación señaladas.

CR1.3 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.

RP2: Manipular el contenido de bases de datos relacionales de forma interactiva.

CR2.1 Las especificaciones recibidas se interpretan con corrección identificando los objetos de la base de datos que se van a manipular.

CR2.2 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.).

CR2.3 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.

CR2.4 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

CR3.1 Las especificaciones recibidas se interpretan con corrección identificando los objetos de la base de datos que se van a manipular.

CR3.2 La codificación se realiza en el lenguaje de programación propio del sistema de base de datos y siguiendo las especificaciones del diseño.

CR3.3 El código desarrollado debe finalizar las transacciones asegurando la integridad y consistencia de la base de datos en cualquier caso.

CR3.4 Las consultas se prueban en ambientes controlados y con información conocida.

CR3.5 Las consultas se optimizan utilizando las técnicas y herramientas disponibles.

CR3.6 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

Denominación: DESARROLLAR COMPONENTES SOFTWARE EN LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS

Nivel: 3

Código: UC0227_3

Realizaciones profesionales y criterios de realización

RP1: Implementar los componentes software encomendados de modo que cumplan las especificaciones del diseño y los niveles de calidad establecidos.

CR1.1 Los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software se identifican interpretando correctamente el diseño detallado.

CR1.2 Las inconsistencias del diseño que sean detectadas son transmitidas al responsable del diseño por los cauces establecidos.

CR1.3 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.

CR1.4 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.

CR1.5 El código implementa el control de errores especificado por las normas de calidad de la organización y por la especificación de requisitos.

CR1.6 Las herramientas de desarrollo se emplean para facilitar el proceso de generación del código.

CR1.7 Las herramientas de depuración se emplean para detectar y corregir errores del código desarrollado.

CR1.8 Los problemas encontrados en la implementación se resuelven consultando la documentación técnica y otras fuentes de información.

CR1.9 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.

CR1.10 La codificación de cada clase se realiza manteniendo los principios de máxima cohesión y mínimo acoplamiento.

CR1.11 Las correcciones necesarias como consecuencia de los errores o discrepancias encontrados en las pruebas se codifican en los componentes software afectados.

CR1.12 Los cambios en la implementación como consecuencia de cambios en las especificaciones se codifican adecuando la estructura del código y manteniendo su calidad.

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

CR2.2 La herramienta de programación se adapta según el interfaz de objetos de acceso a datos seleccionado (integración de librerías, configuración de rutas, etc.).

CR2.3 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.

CR2.4 Las operaciones de manipulación de datos necesarias para el funcionamiento de la aplicación se ejecutan por medio del interfaz de programación.

CR2.5 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.

RP3: Probar los componentes software desarrollados para asegurar que cumplen las especificaciones recibidas

CR3.1 El conjunto de caminos básicos de una unidad de código se identifica para la preparación de las pruebas de caja blanca.

CR3.2 Cada unidad de código se prueba mediante la realización de pruebas estructurales o de caja blanca para verificar su funcionamiento.

CR3.3 Cada unidad de código se prueba mediante la realización de pruebas funcionales o de caja negra para verificar que cumple las especificaciones.

CR3.4 Las unidades de código se prueban de manera aislada y/ o en grupos de unidades considerando el nivel más bajo.

CR3.5 Las pruebas de regresión necesarias como consecuencia de cambios en las especificaciones se realizan para asegurar la funcionalidad previa.

CR3.6 Los componentes software desarrollados se prueban en ambientes controlados y con información conocida.

CR3.7 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.

CR3.8 Los escenarios posibles se prueban mediante la interacción de los objetos implicados.

CR3.9 Todos los métodos se prueban como parte de alguna secuencia.

CR3.10 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.

RP4: Utilizar los componentes orientados a objeto como base en el desarrollo de aplicaciones para el modelo de programación web.

CR4.1 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.

CR4.2 El código orientado a objeto se llama desde componentes software de aceptación de solicitudes de cliente para efectuar la operación solicitada.

CR4.3 El código orientado a objeto se descarga a clientes ligeros para colaborar en el funcionamiento de la capa de presentación integrándose en dicha capa.

CR4.4 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.

RP5: Elaborar la documentación del código desarrollado según los estándares de la organización.

CR5.1 La documentación correspondiente a cada clase se redacta de acuerdo con las normas y modelos incluidos en el diseño de la aplicación.

CR5.2 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.

CR5.3 El código contiene los comentarios necesarios para la mejor comprensión del software así como identificadores con nombres autoexplicativos (mnemotécnicos), tabulaciones, etc.

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

CR5.5 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.).

III. FORMACIÓN DEL CERTIFICADO DE PROFESIONALIDAD**MÓDULO FORMATIVO 1**

Denominación: SISTEMAS OPERATIVOS Y APLICACIONES INFORMÁTICAS

Código: MF0223_3

Nivel de cualificación profesional: 3

Asociado a la Unidad de Competencia: UC0223_3: Configurar y explotar sistemas informáticos.

Duración: 170 horas

Unidad formativa 1

Denominación: COMPUTADORES PARA BASES DE DATOS

Código: UF1465

Duración: 60 horas

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

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.

Contenidos

1. Definición e identificación de la estructura y componentes principales de un computador de propósito general atendiendo a su función y utilidad

- Procesador:
 - Identificación los distintos tipos de procesadores (CISC y RISC), así como sus diferentes partes (Set de Instrucciones, Registros, Contador, Unidad aritmético-Lógica, Interrupciones):
 - Enumeración y clasificación de los conjuntos de instrucciones principales de un procesador de la familia 8086.
 - Ventajas e inconvenientes de los modelos CISC y RISC.
- Memorias RAM y xPROM:
 - Descripción de los modelos de memorias RAM actuales y diferencias DDR, SDRAM.
 - Descripción de modelos de memorias xPROM.
 - Definición de memorias caches.
 - Niveles de caché.
 - Determinación de su importancia en el rendimiento del sistema.
 - Valoración de las memorias por velocidad y coste de unidad de almacenamiento.
- Interfaces de entrada/salida:
 - Dispositivos externos.
 - Enumeración general de tipos de interfaces más comunes.
 - Discos:
 - Definición y descripción.
 - Identificación y localización de las partes de discos mecánicos.
 - Clasificación de discos por su interfaz de Entrada/Salida.
 - Ventajas e inconvenientes de los distintos tipos de discos.
 - Análisis del disco SCSI y su interfaz.
 - Análisis del disco de estado sólido y sus ventajas e inconvenientes como medio de almacenamiento sustitutivo de discos mecánicos.
 - Análisis del disco SATA y comparación con los discos PATA.
- Familias y tipos de procesadores:
 - Clasificación de las familias de los procesadores basándose en su compatibilidad con el microprocesador 8086 de Intel (CISC y RISC).
 - Clasificación y modelos de procesador de 64 bits.
 - Ventajas e inconvenientes de las diferentes soluciones de los principales fabricantes.
 - Historia de los procesadores Intel desde el modelo 8086 a la familia Pentium.
 - Comparación con AMD.
 - Procesador Power. Ventajas y tipos de periféricos.
 - Clasificación por periféricos de entrada o de salida.
 - Introducción al lenguaje ensamblador.
 - Desarrollo de un supuesto práctico de identificación y ensamblado de un computador, según las pautas que se indiquen.
 - Sistemas operativos.

- 2. Funciones y objetivos de los sistemas operativos y manejo de la memoria**
 - El sistema operativos como interfaz usuario/computados.
 - El sistema operativo como administrador de recursos.
 - Facilidad de evolución de un sistema operativo.
 - Requerimientos de la gestión de memoria (reubicación, protección, compartición, organización lógica y física).
 - Concepto de memoria virtual.
 - Concepto de paginación.
 - Incidencia de la paginación en el rendimiento del sistema.
 - Descripción de la gestión de memoria en sistemas Linux, Windows y Solares.

- 3. Sistemas de archivo**
 - Archivos.
 - Directorios.
 - Implementación de sistemas de archivos.
 - Ejemplos y comparación de sistemas de archivos.
 - Sistemas de archivos con journaling.
 - Seguridad del sistema de archivos.

- 4. Identificación y descripción de los conceptos de multiproceso y multiusuario.**
 - Hardware de multiprocesador.
 - Tipos de sistemas operativos para multiprocesador.
 - Multicomputadoras.
 - Explicación de la organización de usuarios. Descripción de los diferentes modelos de organización.

- 5. Particionamiento lógico y núcleos virtuales.**
 - Concepto de virtualización.
 - Historia de la virtualización.
 - Descripción y comparación de las diferentes implementaciones de virtualización. (virtualización completa, paravirtualización...).

- 6. Aplicación de las técnicas de configuración y ajuste de sistemas**
 - Rendimiento de los sistemas. Enumeración, descripción e interpretación de las principales herramientas para observar el consumo de recursos en sistemas en memoria, CPU y disco en Windows, Linux y Solaris.
 - Ejemplos de resolución de situaciones de alto consumo de recursos y competencia en sistemas Windows, Linux y Solaris.
 - Enumeración y descripción los principales procesos de servicios que se ejecutan en los sistemas operativos Windows, Linux y Solaris y su efecto sobre el conjunto del sistema.
 - Descripción de diferentes sistemas de accounting que permitan establecer modelos predictivos y análisis de tendencias en los sistemas operativos Windows, Linux y Solaris.
 - Planes de pruebas de preproducción. Descripción de diferentes herramientas para realizar pruebas de carga que afecten a CPU, Memoria y Entrada/Salida en los sistemas operativos Windows, Linux y Solaris.
 - Elaboración de un plan de pruebas para el sistema operativo.

Unidad formativa 2**Denominación:** SISTEMAS DE ALMACENAMIENTO**Código:** UF1466

Duración: 70 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP2 y RP4

Capacidades y criterios de evaluación

C1: Reconocer y describir codificaciones y nomenclaturas de elementos informáticos de acuerdo con los criterios de estandarización más extendidos.

CE1.1 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.

CE1.2 Describir y emplear normativas de nomenclatura estandarizada de máquinas, servicios y aplicaciones requerida para facilitar las tareas de administración.

CE1.3 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.

CE1.4 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)

CE1.5 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.

C2: Distinguir los diferentes tipos de almacenamiento usados en los sistemas operativos multiusuario indicando su estructura, características y modos de operación.

CE2.1 Enumerar y clasificar los diferentes sistemas de almacenamiento en función de su capacidad, características de rendimiento y compatibilidad con los sistemas operativos más extendidos.

CE2.2 Describir y clasificar los mecanismos de protección y recuperación física de la información en función de su modo de funcionamiento y rendimiento.

CE2.3 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.

CE2.4 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.

CE2.5 Explicar como funciona y que valor aporta para el sistema operativo el acceso en paralelo a múltiples volúmenes físicos.

CE2.6 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.

CE2.7 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 sistema 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.

C3: Aplicar técnicas y procedimientos relacionados con la seguridad de sistemas, redes de comunicaciones y datos.

CE3.1 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.

CE3.2 Explicar las diferencias entre copias de seguridad físicas y lógicas y su influencia en los sistemas operativos, sistemas de ficheros y bases de datos.

CE3.3 Identificar las principales arquitecturas de alta disponibilidad de sistemas y componentes y analizar sus ventajas y debilidades en función de cada caso.

CE3.4 Explicar el modo de funcionamiento de los cortafuegos, antivirus y proxys en las arquitecturas de redes de comunicaciones.

CE3.5 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.

CE3.6 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.

Contenidos

1. Organización y gestión de la información

- Sistemas de archivo:
 - Nomenclatura y codificación.
 - Jerarquías de almacenamiento.
 - Migraciones y archivado de datos.
- Volúmenes lógicos y físicos:
 - Concepto de particionamiento.
 - Concepto de tabla de particiones y MBR.
 - Descripción de sistemas de almacenamiento NAS y SAN. Comparación y aplicaciones. Comparación de los sistemas SAN iSCSI, FC y FCoE.
 - Gestión de volúmenes lógicos. El sistema de gestión de volúmenes LVM. Guía básica de uso de LVM.
 - Acceso paralelo.
 - Protección RAID. Comparación de los diferentes niveles de protección RAID. Mención de la opción de controladoras RAID software o hardware: RAID 0, RAID 1, RAID 5 (Recuperación de discos grandes con RAID 5) y RAID 6.
 - Análisis de las políticas de Salvaguarda:
 - Los puntos únicos de fallo, concepto e identificación.
 - Tipos de copias de seguridad y calendarización de copias.

- Salvaguarda física y lógica.
- Salvaguarda a nivel de bloque y fichero.
- Conceptos de Alta Disponibilidad. Diferencias entre cluster, grid y balanceo de carga.
- Integridad de datos y recuperación de servicio. Guía mínima para elaborar un plan de continuidad de negocio. Conceptos de RTO (Recovery Point Objective) y RTO (Recovery Time Objective).
- Custodia de ficheros de seguridad. Problemática de la salvaguarda y almacenamiento de datos confidenciales. Algunas implicaciones Ley Orgánica de Protección de Datos (LOPD).
- Análisis de las políticas de Seguridad:
 - Acceso restringido por cuentas de usuario. Propiedad de la información.
 - Identificador único de acceso. Sistemas de Single Sign On (SSO).
 - Protección antivirus.
 - Auditorias de seguridad.

2. Desarrollo de diferentes supuestos prácticos, debidamente caracterizados, en los que se analicen

- El efecto de las posibles decisiones de particionamiento y acceso a disco así como la implementación de una política de salvaguarda de datos.
- La política de nomenclatura de los diferentes sistemas y el desarrollo de un mapa de red para documentarlo.
- Distintos sistemas de ficheros para estudiar la nomenclatura seleccionada y los datos de acceso y modificación de los ficheros, así como los permisos de los usuarios de acceso a los mismos.
- La migración de datos entre diferentes sistemas.

Unidad formativa 3

Denominación: APLICACIONES MICROINFORMÁTICAS E INTERNET PARA CONSULTA Y GENERACIÓN DE DOCUMENTACIÓN

Código: UF1467

Duración: 40 horas

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

Capacidades y criterios de evaluación

C1: Usar los principales tipos de herramientas ofimáticas y los servicios y aplicaciones asociados a Internet.

CE1.1 Enumerar y explicar las funciones principales de los procesadores de texto, hojas de cálculo y edición de presentaciones.

CE1.2 Relacionar los principales servicios asociados a Internet y clasificarlos sobre la base de su función y especificidad.

CE1.3 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.

CE1.4 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.

CE1.5 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.

Contenidos

1. 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 estándar. Estructura de la información y metadatos en los documentos.
 - El Wiki como herramienta de escritura colaborativa.
- Uso de Internet:
 - Conocimiento de www. Navegadores.
 - Sistemas de correo electrónico, chat y foros. Reglas de conducta aplicar en los foros, chat y correo electrónico.
 - Transferencia de ficheros. Explicación de las técnicas de transferencias de ficheros a nivel de usuario y discriminando las que aportan elementos de seguridad tanto para identificación como cifrado.
 - Proyectos de software libre en la web. Identificación de los sitios para encontrarlos, dónde bajar el software y como contactar con la comunidad.
 - Sistemas de control de versiones.

Orientaciones metodológicas

Para acceder a la unidad formativa UF2 debe haberse superado la unidad formativa UF1.

La unidad formativa UF3 se puede programar de manera independiente.

Criterios de acceso para los alumnos

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

MÓDULO FORMATIVO 2

Denominación: PROGRAMACIÓN DE BASES DE DATOS RELACIONALES

Código: MF0226_3

Nivel de calificación profesional: 3

Asociado a la Unidad de Competencia: UC0226_3: Programar bases de datos relacionales.

Duración: 210 horas

Unidad formativa 1

Denominación: DISEÑO DE BASES DE DATOS RELACIONALES

Código: UF2175

Duración: 50 horas

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

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 del 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.

Contenidos

1. Introducción a las bases de datos

- Evolución histórica de las bases de datos.
- Ventajas e inconvenientes de las bases de datos.
- Conceptos generales:
 - Concepto de bases de datos.
 - Objetivos de los sistemas de bases de datos:
 - Redundancia e inconsistencia de datos.
 - Dificultad para tener acceso a los datos.
 - Aislamiento de los datos.
 - Anomalías del acceso concurrente.
 - Problemas de seguridad.
 - Problemas de integridad.
 - Administración de los datos y administración de bases de datos.
 - Niveles de Arquitectura: interno, conceptual y externo.
 - Modelos de datos. Clasificación.
 - Independencia de los datos.
 - Lenguaje de definición de datos.
 - Lenguaje de manejo de bases de datos. Tipos.
 - El Sistema de Gestión de la Base de Datos (DBMS).Funciones.
 - El Administrador de la base de datos (DBA).Funciones.
 - Usuarios de las bases de datos.
 - Estructura general de la base de datos. Componentes funcionales.
 - Arquitectura de sistemas de bases de datos.

2. Modelos conceptuales de bases de datos

- El modelo entidad-relación:
 - Entidades.
 - Interrelaciones: Cardinalidad, Rol y Grado.
 - Dominios y valores.
 - Atributos.
 - Propiedades identificatorias.
 - Diagramas entidad-relación. Simbología.
- El modelo entidad-relación extendido.
- Restricciones de integridad:
 - Restricciones inherentes.
 - Restricciones explícitas.

3. El modelo relacional

- Evolución del modelo relacional.
- Estructura del modelo relacional:
 - El concepto de relación. Propiedades de las relaciones.

- Atributos y dominio de los atributos.
- Tupla, grado y cardinalidad.
- Relaciones y tablas.
- Claves en el modelo relacional:
 - Claves candidatas.
 - Claves primarias.
 - Claves alternativas
 - Claves ajenas.
- Restricciones de integridad:
 - Valor “Null” en el modelo.
 - Integridad de las entidades.
 - Integridad referencial.
- Teoría de la 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. Razones para la desnormalización.

4. El ciclo de vida de un proyecto

- El ciclo de vida de una base de datos:
 - Estudio previo y plan de trabajo. Actividades.
 - Concepción de la BD y selección del equipo físico y lógicos:
 - Conceptos generales acerca del análisis de aplicaciones.
 - Concepción de la base de datos.
 - Selección del equipo físico y lógicos necesarios.
 - Diseño y carga:
 - Conceptos generales acerca del diseño de aplicaciones.
 - Diseño lógico.
 - Diseño físico.
 - Carga y optimización de la base de datos.
- Conceptos generales del control de calidad:
 - Control de calidad de las especificaciones funcionales.
 - Seguimiento de los requisitos de usuario.

5. Creación y diseño de bases de datos

- Enfoques de diseño:
 - Diseños incorrectos. Causas.
 - Enfoque de análisis. Ventajas y desventajas.
 - Enfoque de síntesis. Ventajas y desventajas.
- Metodologías de diseño:
 - Concepto.
 - Diseños conceptual, lógico y físico.
 - Entradas y salidas del proceso.
- Estudio del diseño lógico de una base de datos relacional.
- El Diccionario de Datos: concepto y estructura.
- Estudio del diseño de la BBDD y de los requisitos de usuario.

Unidad formativa 2

Denominación: DEFINICIÓN Y MANIPULACIÓN DE DATOS

Código: UF2176

Duración: 80 horas

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

Capacidades y criterios de evaluación

C1: Formular consultas de manipulación y definición de datos, a partir del diseño de la BBDD y de los requisitos de usuario.

CE1.1 Explicar los fundamentos del álgebra y cálculo relacional y enumerar y diferenciar los lenguajes asociados a la base de datos.

CE1.2 Explicar el tipo de consultas (de selección, de actualización, de inserción, de borrado) que se pueden realizar utilizando el lenguaje DML.

CE1.3 Explicar el tipo de elementos que se pueden crear y manipular utilizando el lenguaje DDL.

CE1.4 Describir la sintaxis de un lenguaje de consultas relacional.

CE1.5 Explicar el concepto de vista y describir su utilidad.

CE1.6 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.

CE1.7 Enumerar y describir las herramientas de la BBDD para realizar formulaciones de manipulación y definición de datos de forma interactiva.

CE1.8 Describir las herramientas de la base de datos para la optimización de consultas.

CE1.9 Realizar 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.

CE1.10 Realizar 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.

Contenidos

1. Lenguajes relacionales

- Tipos de lenguajes relacionales.
- Operaciones en el modelo relacional:
 - Álgebra relacional:

- Clasificación de operadores.
- Denominación de atributos.
- Relaciones derivadas.
- Operaciones primitivas: selección, proyección, producto cartesiano, unión y diferencia.
- Otras operaciones: intersección, join, división, etc.
- Cálculo relacional:
 - Cálculo relacional orientado a dominios
 - Cálculo relacional orientado a tuplas.
 - Transformación de consultas entre álgebra y cálculo relacional.
- Lenguajes comerciales: SQL (Structured Query Language), QBE (Query By Example):
 - Orígenes y evolución del SQL.
 - Características del SQL.
 - Sistemas de Gestión de bases de datos con soporte SQL.

6. El lenguaje de manipulación de la base de datos

- El lenguaje de definición de datos (DDL):
 - Tipos de datos del lenguaje.
 - Creación, modificación y borrado de tablas.
 - Creación, modificación y borrado de vistas.
 - Creación, modificación y borrado de índices.
 - Especificación de restricciones de integridad.
- El lenguaje de manipulación de datos (DML):
 - Construcción de consultas de selección: Agregación, Subconsultas, Unión, Intersección, Diferencia.
 - 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.
- El lenguaje de control de datos (DCL):
 - 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.
 - Control de las transacciones.
 - Privilegios: autorizaciones y desautorizaciones.
- Procesamiento y optimización de consultas:
 - Procesamiento de una consulta.
 - Tipos de optimización: basada en reglas, basada en costes, otros.
 - Herramientas de la BBDD para la optimización de consultas.

Unidad formativa 3

Denominación: DESARROLLO DE PROGRAMAS EN EL ENTORNO DE LA BASE DE DATOS

Código: UF2177

Duración: 80 horas

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

Capacidades y criterios de evaluación

C1: 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.

CE1.1 Enumerar y describir los entornos de desarrollo integrados disponibles en el sistema de gestión de bases de datos.

CE1.2 Enumerar los lenguajes de programación disponibles en los entornos de desarrollo.

CE1.3 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.

CE1.4 Enumerar y describir las posibles herramientas para el desarrollo de entornos gráficos de usuario integradas en el ámbito de la base de datos.

CE1.5 Enumerar y describir las utilidades para la depuración y control de código disponibles en el entorno de la base de datos.

CE1.6 Enumerar y describir las técnicas para el control de la ejecución de las transacciones.

CE1.7 Describir las herramientas de la base de datos para la optimización de consultas.

CE1.8 Desarrollar 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.

Contenidos**1. Lenguajes de programación de bases de datos**

- Entornos de desarrollo:
 - Qué es un entorno de desarrollo.

- Componentes.
- Lenguajes que soportan.
- 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.
- Programación de módulos de manipulación de la base de datos: paquetes, procedimientos y funciones.
- Herramientas de depuración y control de código.
- Herramientas gráficas de desarrollo integradas en la base de datos:
 - Creación de formularios.
 - Creación de informes.
- Técnicas para el control de la ejecución de transacciones.
- Optimización de consultas.

Orientaciones metodológicas

Las unidades formativas deberán superarse de forma correlativa.

Criterios de acceso para los alumnos

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

MÓDULO FORMATIVO 3

Denominación: PROGRAMACIÓN ORIENTADA A OBJETOS

Código: MF0227_3

Nivel de cualificación profesional: 3

Asociado a la Unidad de Competencia: UC0227_3 Desarrollar componentes software en lenguajes de programación orientados a objetos.

Duración: 250 horas

Unidad formativa 1

Denominación: PRINCIPIOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

Código: UF2404

Duración: 90 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP1 en lo referido a la implementación de los componentes de software.

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

Contenidos

1. 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.
- Análisis del proceso de construcción de software: Modularidad.
- Distinción del concepto de módulo en el paradigma orientado a objetos.
- Identificación de 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.
 - Identificación del comportamiento de un objeto: Concepto de mensaje.

2. Clases y objetos

- Distinguir el concepto de clase y sus atributos, métodos y mecanismo de encapsulación:
 - Relación entre interfaz y clase.
 - Distinción de los tipos de datos y clases.
- Análisis de los objetos: Estado, comportamiento e identidad:
 - Análisis de mensajes.
 - Tipos de métodos y su clasificación: Métodos de acceso, de selección o consulta, de construcción, de destrucción.
- Uso de objetos como instancias de clase. Instancia actual (this, self, current).
- Identificación del concepto de programa en el paradigma orientado a objetos. POO = Objetos + Mensajes.

3. Generalización/Especialización: herencia

- Descripción del concepto de 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.
- Distinción de la 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.

4. Relaciones entre clases

- Distinción entre Agregación/Composición.
- Distinción entre Generalización / Especialización.
- Identificación de asociaciones.

5. Análisis del polimorfismo

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

6. Técnicas de programación estructurada

- Identificación de elementos básicos: constantes, variables, operadores y expresiones.
- Análisis de estructuras de control: Secuencial, condicional y de repetición.
- Distinción entre 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.
- Demostración de llamadas a funciones y procedimientos.
- Empleo de llamadas a funciones y procedimientos incluidos en las clases:
 - Llamadas calificadas y no calificadas (instancia actual).
 - Paso de parámetros.
 - Los atributos de la clase.

7. Estructura de la información

- Enumeración de 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:
 - Uso de la gestión automática de memoria.
 - Construcción y destrucción de objetos.
 - Objetos inalcanzables.
 - Recolección de «basura».
 - Métodos constructores y destructores.

8. Lenguajes de programación orientados a objetos

- Análisis del 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.
- Librerías de clases:
 - Definición de su estructura.
 - Creación y utilización.

9. Implementación del paradigma utilizando un lenguaje de programación orientado a objetos

- Elección del lenguaje.

- Enumeración de los 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.
- Empleo de excepciones.
- Gestión de eventos:
 - Eventos, fuentes y auditores de eventos.
 - Tipos de eventos. Mecanismos de gestión de eventos.
 - Librerías de clases asociadas.
- Empleo de hilos:
 - Fundamentos.
 - Creación.
 - Prioridad.
 - Comunicación.
 - Sincronización.
 - Estados.
 - Creación y ejecución de hilos en el lenguaje.
 - Librerías de clases asociadas.
 - Programación multihilo.
- Definición y análisis de programación en red:
 - Aplicaciones cliente servidor.
 - Sockets.
- Acceso a bases de datos desde las aplicaciones. Librerías de clases asociadas.

Unidad formativa 2

Denominación: MODELO DE PROGRAMACIÓN WEB Y BASES DE DATOS

Código: UF2405

Duración: 80 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP2 y RP4.

Capacidades y criterios de evaluación

C1: Aplicar los conceptos básicos del modelo de programación web.

CE1.1 Enumerar y describir los componentes del modelo multicapa de programación web. (Cliente ligero, servidores web, servidores de aplicaciones, servidores de base de datos).

CE1.2 Enumerar y describir la función de los protocolos y tecnologías habituales. (TCP/IP, http, HTML, XML, XSL, SOAP).

CE1.3 Enumerar las características básicas de los modelos de programación ampliamente utilizados (J2EE y .NET).

CE1.4 Describir las capas lógicas de una aplicación web (Presentación, Aplicación y Datos).

CE1.5 Describir las características básicas del lenguaje de presentación (HTML).

CE1.6 Describir las características básicas de los lenguajes de scripting en cliente (JavaScript, VBScript).

CE1.7 Describir el funcionamiento de una sesión de aplicación en el modelo de programación web.

CE1.8 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).

C2: Realizar conexiones con bases de datos relacionales.

CE2.1 Enumerar y describir las diferentes tecnologías de conexión con la BBDD desde las aplicaciones.

CE2.2 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.

CE2.3 Enumerar y describir las clases que proporcionan los medios adecuados para efectuar consultas, actualizaciones, acceder y operar con una base de datos relacional.

CE2.4 Describir los procedimientos para realizar dichas consultas (abrir y cerrar conexiones, ejecutar comandos, recoger sus resultados y utilizarlos).

CE2.5 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.

CE2.6 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.

Contenidos

1. Introducción al desarrollo de aplicaciones en el modelo de programación web

- Análisis de la arquitectura web: Cliente ligero, servidor web, servidor de aplicaciones, servidor de datos.
- Enumeración de protocolos y tecnologías habituales.
- Análisis de los modelos de programación estándares de facto.

- Uso de componentes orientados a objeto como base en el desarrollo de aplicaciones en el modelo de programación web.
- 2. Arquitectura multicapa (n-tier)**
 - Análisis de la arquitectura multicapa:
 - Distinción y estudio del modelo de tres capas en web: presentación, aplicación y datos.
 - Diseño de arquitecturas de aplicación basadas en el modelo multicapa
 - Análisis del concepto de lógica de negocio y significado de la capa lógica
- 3. La capa de presentación**
 - Descripción de la capa de presentación: El lenguaje de hipertexto.
 - Descripción de la capa de presentación avanzada: Lenguajes de scripting y lenguaje de hipertexto dinámico.
 - Análisis de lenguajes orientados a la preparación de la capa de presentación y a la ejecución de solicitudes desde clientes ligeros web. (JSP, Servlets, ASP, PHP).
- 4. Diseño de bases de datos relacionales**
 - Definición de bases de datos relacionales.
 - Diseño de bases de datos en varios niveles.
 - Análisis de los distintos tipos de relaciones y su implementación en base de datos.
 - Descripción del lenguaje de acceso a base de datos
 - Descripción de correlaciones entre el modelo relacional y modelo orientado a objetos.
 - Nociones sobre el almacenamiento de objetos en las bases de datos relacionales.
- 5. Acceso a bases de datos relacionales: capa de acceso a datos**
 - Análisis del API de acceso a la base de datos.
 - Nivel controlador.
 - Interfaz de acceso a la base de datos (driver).
 - Análisis del 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.
- 6. Lenguajes de definición de datos**
 - Conceptos básicos, nociones y estándares
 - Lenguaje de definición de datos (DDL SQL) y aplicación en SGBD actuales
 - Discriminación de los elementos existentes en el estándar SQL-92 de otros elementos existentes en bases de datos comerciales
 - Sentencias de creación: CREATE:
 - Bases de datos
 - Tablas
 - Vistas
 - Disparadores o Triggers
 - Procedimientos
 - Sentencias de modificación: ALTER:
 - Bases de datos
 - Tablas
 - Vistas
 - Disparadores o Triggers

- Procedimientos
- Sentencias de borrado: DROP, TRUNCATE:
 - Bases de datos
 - Tablas
 - Vistas
 - Disparadores o Triggers
 - Procedimientos

7. Manipulación de los datos

- Lenguaje de manipulación de datos (DML SQL)
- Consultas de datos: SELECT
- Inserción de datos: INSERT
- Modificación de datos: UPDATE
- Eliminación de datos: DELETE
- Agregación de conjuntos de datos para consulta: JOIN, UNION
- Subconsultas

Unidad formativa 3

Denominación: EL CICLO DE VIDA DEL DESARROLLO DE APLICACIONES

Código: UF2406

Duración: 80 horas

Referente de competencia: Esta unidad formativa se corresponde con la RP1 en lo referido a las especificaciones de diseño y niveles de calidad, RP3 y RP5.

Capacidades y criterios de evaluación

C1: Manejar las herramientas de ingeniería de software.

CE1.1 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.

CE1.2 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.

CE1.3 Describir en detalle los conceptos fundamentales de una metodología de ingeniería software basada en la orientación a objetos.

CE1.4 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.

CE1.5 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.

C2: Verificar la corrección de las clases desarrolladas mediante la realización de pruebas.

CE2.1 Enumerar y describir las herramientas y utilidades más comunes para la depuración de programas.

CE2.2 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.

CE2.3 Enumerar y describir los tipos de pruebas que se deben realizar a una clase para verificar su corrección.

CE2.4 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.

CE2.5 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.

CE2.6 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.

CE2.7 Realizar pruebas a una clase mediante herramientas de prueba.

C3: Elaborar la documentación completa relativa a las clases desarrolladas y pruebas realizadas.

CE3.1 Describir la información que debe acompañar a una clase desarrollada.

CE3.2 Describir los criterios fundamentales para la inclusión de cabeceras y comentarios en el código.

CE3.3 Proponer índices (plantillas) para los documentos de diseño y explicar el contenido de cada uno de los apartados.

CE3.4 Proponer índices (plantillas) para los documentos de pruebas (planificación y resultados) y explicar el contenido de cada uno de los apartados.

CE3.5 Proponer índices (plantillas) para la documentación de operación y mantenimiento (manuales técnicos) y explicar el contenido de cada uno de los apartados.

CE3.6 Aplicar unos criterios de normalización establecidos para incluir cabeceras y comentarios en el código.

- CE3.7 En un supuesto práctico, a partir de una clase desarrollada, elaborar la documentación de diseño de la clase de acuerdo a un índice establecido.
- CE3.8 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.
- CE3.9 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.
- CE3.10 Realizar la documentación de las clases mediante herramientas de documentación automática.
- C4: Realizar modificaciones de clases existentes por cambios en las especificaciones.
- CE4.1 Enumerar y describir los conceptos fundamentales de la gestión de la configuración del software desarrollado por una organización.
- CE4.2 Enumerar y describir los conceptos fundamentales de la gestión de la configuración de la documentación.
- CE4.3 Describir los pasos y precauciones fundamentales en el proceso de modificación de clases existentes.
- CE4.4 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.
- CE4.5 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.
- C5: Desarrollar interfaces de usuario en lenguajes de programación orientados a objeto, a partir del diseño detallado.
- CE5.1 Interpretar diagramas de casos de uso y analizar las necesidades y peticiones de usuarios.
- CE5.2 Identificar y describir las clases básicas que se usan para el interfaz hombre / máquina.
- CE5.3 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.
- CE5.4 Identificar los recursos multimedia que pueden incluirse en una ventana de interfaz de usuario.
- CE5.5 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.
- CE5.6 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.

CE5.7 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.

CE5.8 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.

Contenidos

1. Proceso de ingeniería del software

- Distinción de las fases del proceso de ingeniería software: especificación, diseño, construcción y pruebas unitarias, validación, implantación y mantenimiento.
- Análisis de los modelos del proceso de ingeniería: modelo en cascada, desarrollo evolutivo, desarrollos formales, etc.
- Identificación de requisitos: concepto, evolución y trazabilidad.
- Análisis de metodologías de desarrollo orientadas a objeto.
- Resolución de un caso práctico de metodologías de desarrollo que utilizan UML.
- Definición del concepto de herramientas CASE:
 - Herramientas de ingeniería software
 - Entornos de desarrollo
 - Herramientas de prueba
 - Herramientas de gestión de la configuración
 - Herramientas para métricas.

2. Planificación y seguimiento

- Realización de estimaciones
- Planificaciones: modelos de diagramado. Diagrama de Gantt
- Análisis del proceso del seguimiento: Reuniones e Informes

3. Diagramado

- Identificación de los principios básicos de UML.
- Empleo de diagramas de uso.

4. Desarrollo de la GUI

- Análisis del modelo de componentes y eventos.
- Identificación de elementos de la GUI.
- Presentación del diseño orientado al usuario. Nociones de usabilidad.
- Empleo de herramientas de interfaz gráfica

5. Calidad en el desarrollo del software

- Enumeración de criterios de calidad.
- Análisis de métricas y estándares de calidad.

6. Pruebas

- Identificación de tipos de pruebas.
- Análisis de 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.

7. Excepciones

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

8. Documentación

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

Orientaciones metodológicas

Para acceder a la unidad formativa UF2 debe haberse superado la unidad formativa UF1.

La unidad formativa UF3 se puede programar de manera independiente.

Criterios de acceso para los alumnos

Serán los establecidos en el artículo 4 del real decreto que regula el certificado de profesionalidad de la familia Profesional al que acompaña este anexo.

MÓDULO DE PRÁCTICAS PROFESIONALES NO LABORALES DE PROGRAMACIÓN CON LENGUAJES ORIENTADOS A OBJETOS Y BASES DE DATOS RELACIONALES

Código: MP0501

Duración: 80 horas

Capacidades y criterios de evaluación

C1: Reconocer y describir codificaciones y nomenclaturas de elementos informáticos de acuerdo con los criterios de estandarización más extendidos.

CE1.1 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.

- CE1.2 Describir y emplear normativas de nomenclatura estandarizada de máquinas, servicios y aplicaciones requerida para facilitar las tareas de administración.
- CE1.3 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.
- C2: Dominar los conceptos fundamentales del paradigma Orientado a Objetos.
- CE2.1 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.
- CE2.2 Describir y enumerar las características de una clase: atributos, métodos y mecanismo de encapsulación, identificando la interfaz de la clase y lo que representa.
- CE2.3 Describir y enumerar las características que definen un objeto, distinguiendo las diferencias entre los conceptos de objeto y clase.
- CE2.4 Describir la estructura y el significado de los mensajes y su relación con el comportamiento de los objetos.
- CE2.5 Explicar las características fundamentales que tienen que estar presentes en una relación entre dos clases para que pueda ser calificada como relación de herencia.
- CE2.6 Describir el mecanismo de herencia múltiple y los problemas que presenta en el proceso de desarrollo de software.
- CE2.7 Explicar el concepto de polimorfismo y enumerar y describir las características que introduce en el proceso de desarrollo del software.
- C3: Aplicar los conceptos básicos del modelo de programación web.
- CE3.1 Describir las capas lógicas de una aplicación web (Presentación, Aplicación y Datos).
- CE3.2 Describir el funcionamiento de una sesión de aplicación en el modelo de programación web.
- CE3.3 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: Elaborar la documentación completa relativa a las clases desarrolladas y pruebas realizadas.
- CE4.1 Describir la información que debe acompañar a una clase desarrollada.
- CE4.2 Describir los criterios fundamentales para la inclusión de cabeceras y comentarios en el código.
- CE4.3 Proponer índices (plantillas) para los documentos de diseño y explicar el contenido de cada uno de los apartados.
- CE4.4 Proponer índices (plantillas) para los documentos de pruebas (planificación y resultados) y explicar el contenido de cada uno de los apartados.
- CE4.5 Proponer índices (plantillas) para la documentación de operación y mantenimiento (manuales técnicos) y explicar el contenido de cada uno de los apartados.
- CE4.6 Aplicar unos criterios de normalización establecidos para incluir cabeceras y comentarios en el código.
- CE4.7 Realizar la documentación de las clases mediante herramientas de documentación automática.
- C5: Realizar conexiones con bases de datos relacionales.
- CE5.1 Enumerar y describir las clases que proporcionan los medios adecuados para efectuar consultas, actualizaciones, acceder y operar con una base de datos relacional.

CE5.2 Describir los procedimientos para realizar dichas consultas (abrir y cerrar conexiones, ejecutar comandos, recoger sus resultados y utilizarlos).

C6: Manejar las herramientas de ingeniería de software.

CE6.1 Describir las fases, en cada modelo, del proceso de ingeniería de software.

CE6.2 Describir en detalle los conceptos fundamentales de una metodología de ingeniería software basada en la orientación a objetos.

C7: Formular consultas de manipulación y definición de datos, a partir del diseño de la BBDD y de los requisitos de usuario

CE7.1 Explicar el tipo de consultas (de selección, de actualización, de inserción, de borrado) que se pueden realizar utilizando el lenguaje DML.

CE7.2 Explicar el tipo de elementos que se pueden crear y manipular utilizando el lenguaje DDL.

CE7.3 Realizar formulaciones de manipulación de datos, a partir del diseño de la base de datos y de los requisitos de usuario:

CE7.4 Realizar formulaciones de definición de datos, a partir del diseño de la base de datos y de los requisitos de usuario

C8: Participar en los procesos de trabajo de la empresa, siguiendo las normas e instrucciones establecidas en el centro de trabajo.

CE8.1 Comportarse responsablemente tanto en las relaciones humanas como en los trabajos a realizar.

CE8.2 Respetar los procedimientos y normas del centro de trabajo.

CE8.3 Empezar con diligencia las tareas según las instrucciones recibidas, tratando de que se adecuen al ritmo de trabajo de la empresa.

CE8.4 Integrarse en los procesos de producción del centro de trabajo.

CE8.5 Utilizar los canales de comunicación establecidos.

CE8.6 Respetar en todo momento las medidas de prevención de riesgos, salud laboral y protección del medio ambiente.

Contenidos

1. Organización y gestión de la información

- Sistemas de archivo:
- Nomenclatura y codificación.
- Jerarquías de almacenamiento.
- Migraciones y archivado de datos.

2. Análisis y diseño de clases

- Identificación de clases.
- Modelado de clases con UML: diagrama de clases.

3. Implementación del modelo

- Implementación de clases.
- Implementación de interfaces.

4. Despliegue arquitectura web

- Diseño de la arquitectura lógica cliente-servidor.
- Implementación del servidor.
- Despliegue de clases en el servidor.
- Desarrollo de un cliente de pruebas.

5. Acceso a datos

- Diseño de la arquitectura lógica cliente-servidor.
- Implementación del servidor.

- Despliegue de clases en el servidor.
- Desarrollo de un cliente de pruebas.

6. Plan de pruebas

- Diseño del plan de pruebas.
- Pruebas unitarias.
- Pruebas de integración.
- Gestión de requisitos.
- Gestión de cambios.
- Change Log.
- Implementación de cambios: releases y actualización de documentación.

7. El lenguaje de manipulación de la base de datos.

- El lenguaje de definición de datos (DDL)
- Tipos de datos del lenguaje.
- Creación, modificación y borrado de tablas.
- Creación, modificación y borrado de vistas.
- Creación, modificación y borrado de índices.
- Especificación de restricciones de integridad.
- El lenguaje de manipulación de datos (DML).
- Construcción de consultas de selección: Agregación, Subconsultas, Unión, Intersección, Diferencia.
- 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.

8. Integración y comunicación en el centro de trabajo

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

I. PRESCRIPCIONES DE LOS FORMADORES

Módulos formativos	Acreditación requerida	*Experiencia profesional requerida en el ámbito de la unidad de competencia
MF0223_3 Sistemas operativos y aplicaciones informáticas.	Licenciado, Ingeniero, Arquitecto o el Título de Grado correspondiente u otros títulos equivalentes. Diplomado, Ingeniero Técnico, Arquitecto Técnico o el Título de Grado correspondiente u otros títulos equivalentes.	1 año

Módulos formativos	Acreditación requerida	*Experiencia profesional requerida en el ámbito de la unidad de competencia
MF0226_3 Programación de bases de datos relacionales.	Licenciado, Ingeniero, Arquitecto o el Título de Grado correspondiente u otros títulos equivalentes. Diplomado, Ingeniero Técnico, Arquitecto Técnico o el Título de Grado correspondiente u otros títulos equivalentes.	1 año
MF0227_3 Programación orientada a objetos.	Licenciado, Ingeniero, Arquitecto o el Título de Grado correspondiente u otros títulos equivalentes. Diplomado, Ingeniero Técnico, Arquitecto Técnico o el Título de Grado correspondiente u otros títulos equivalentes.	1 año

* En los últimos tres años

REQUISITOS MÍNIMOS DE ESPACIOS, INSTALACIONES Y EQUIPAMIENTO

Espacio formativo	Superficie m ² 15 alumnos	Superficie m ² 25 alumnos
Aula de informática.	45	75

Espacio formativo	M1	M2	M3
Aula de informática.	X	X	X

Espacio formativo	Equipamiento
Aula de informática	<ul style="list-style-type: none"> – Equipos informáticos y periféricos. PCs instalados en red y conexión a Internet. – Software ofimático, herramientas internet. – Software de seguridad y antivirus. – 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. – Sistemas gestores de bases de datos. – Diccionarios de datos (catálogo, tablas de sistema, etc.). – Sistemas de documentación de elementos de programación. – Lenguajes estructurados. – Herramientas de depuración. – Herramientas de distribución de aplicaciones. – Entornos de desarrollo de aplicaciones. – Bases de datos. – Software de manejo de bases de datos. – Herramientas de documentación. – Cañón de proyección. – Rotafolios. – Pizarra. – Material de aula. – Mesas y sillas para alumnos y mesa y silla para el formador – Mobiliario auxiliar para el equipamiento de aula. <p>* El equipamiento y el software correspondiente deberán estar actualizados.</p>

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

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

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

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