

CICLO SUPERIOR DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROGRAMACIÓN DIDÁCTICA ANUAL

Parte específica del módulo:

0485. Programación

**Departamento de Familia Profesional de
Informática y Comunicaciones**

Curso: 2022 / 2023

Nivel: primero

Turno: mañanas

Profesor: David Munuera Sánchez

CONTENIDO

CONTENIDO.....	2
CARACTERÍSTICAS GENERALES DEL CICLO DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA	5
DESCRIPCIÓN DEL MÓDULO.....	5
UBICACIÓN, OBJETIVOS, CONTENIDOS Y DISTRIBUCIÓN TEMPORAL DEL MÓDULO.....	5
UBICACIÓN, DISTRIBUCIÓN TEMPORAL Y CARACTERÍSTICAS 	5
OBJETIVOS GENERALES DEL MÓDULO Y RESULTADOS DE APRENDIZAJE ASOCIADOS.....	5
RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN 	6
OBJETIVOS GENERALES DEL MÓDULO ASOCIADOS A RESULTADOS DE APRENDIZAJE.....	8
COMPETENCIAS PROFESIONALES, PERSONALES Y SOCIALES ASOCIADAS A RESULTADOS DE APRENDIZAJE.....	9
CONTENIDOS Y UNIDADES DE TRABAJO.....	11
PLANIFICACIÓN DE UT's.....	12
UNIDADES DE TRABAJO.....	13
METODOLOGÍA.....	26
CRITERIOS.....	26
ASPECTOS CONCRETOS.....	26
AGRUPAMIENTO DEL ALUMNADO.....	27
MATERIALES, RECURSOS, ESPACIO DOCENTE.....	27
MATERIALES Y RECURSOS DIDÁCTICOS.....	27
BIBLIOGRAFÍA.....	27
DISTRIBUCIÓN DEL ESPACIO Y EL TIEMPO DOCENTE.....	28
MEDIDAS PARA ESTIMULAR EL INTERÉS Y EL HÁBITO DE LECTURA Y LA CAPACIDAD DEL ALUMNADO PARA EXPRESARSE CORRECTAMENTE.....	28

CRITERIOS, PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN.....28

CRITERIOS DE CALIFICACIÓN.....28

CÁLCULO DE LA CALIFICACIÓN FINAL.....30

EXAMEN DE CONVOCATORIA ORDINARIA.....30

EXAMEN DE CONVOCATORIA EXTRAORDINARIA.....31

RECUPERACIÓN DE CONTENIDOS.....31

ALUMN@S MATRICULAD@S EN 2º CON EL MÓDULO SUSPENSO.....32

TIPO DE RECUPERACIÓN.....32

FECHAS DE ENTREGA DE TRABAJOS O REALIZACIÓN DE EXÁMENES PARCIALES.....32

PROCESO DE RECUPERACIÓN.....32

CRITERIOS DE RECUPERACIÓN.....32

ATENCIÓN A LA DIVERSIDAD DEL ALUMNADO EN LOS CICLOS FORMATIVOS.....33

PREVENCIÓN DE RIESGOS LABORALES.....33

ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES33

USO DE LAS TIC's.....33

REFERENCIAS LEGALES.....33

CARACTERÍSTICAS GENERALES DEL CICLO DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Se relacionan en la parte general de la programación del ciclo formativo.

DESCRIPCIÓN DEL MÓDULO

El módulo profesional de *Programación* se centra en el estudio de los fundamentos de la programación de ordenadores desde el paradigma de la Orientación a Objetos siendo básico para los módulos de segundo curso.

A pesar del carácter introductorio del módulo al desarrollo de aplicaciones; entraña dificultades de diversa índole debido a la necesaria comprensión de conceptos bastante abstractos que además deben llevarse a un nivel práctico y de aplicación profesional. Un/a programador/a debe desarrollar capacidades y mentalidad analíticas junto con importantes habilidades de síntesis de soluciones; y eso es simplemente difícil y requiere bastante tiempo.

La adecuada y correcta orientación profesional previa, junto a una motivación personal, son esenciales para abordar las dificultades inherentes de este módulo.

UBICACIÓN, OBJETIVOS, CONTENIDOS Y DISTRIBUCIÓN TEMPORAL DEL MÓDULO

UBICACIÓN, DISTRIBUCIÓN TEMPORAL Y CARACTERÍSTICAS

El módulo profesional de *Programación* se encuadra en el primer curso del Ciclo Superior de *Desarrollo de Aplicaciones Multiplataforma*, correspondiente al título de *Técnico Superior en Desarrollo de Aplicaciones Multiplataforma*. Se desarrolla, contando con 230 horas de currículo equivalentes a 14 créditos ECTS, en 7 horas semanales de clase.

Las características de este módulo son:

1. Incluye los aspectos básicos de la competencia profesional definida para el título.
2. Es común, con el correspondiente, en el ciclo de *Desarrollo de Aplicaciones Web*.
3. Es básico para los módulos de segundo curso.
4. Se complementa con el módulo de *Entornos de Desarrollo*, también de primero, con el se requiere que esté coordinado.

OBJETIVOS GENERALES DEL MÓDULO Y RESULTADOS DE APRENDIZAJE ASOCIADOS

La legislación vigente (ME - Título DAM) establece los siguientes objetivos generales para el ciclo formativo. En el módulo de **Programación** se identifican los siguientes objetivos generales y resultados de aprendizaje asociados:

- e) Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.*
- l) Valorar y emplear herramientas específicas, atendiendo a la estructura de los contenidos, para crear tutoriales, manuales de usuario y otros documentos asociados a una aplicación.*
- m) Seleccionar y emplear técnicas y herramientas, evaluando la utilidad de los asistentes de instalación generados, para empaquetar aplicaciones.*
- r) Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.*
- s) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.*

El resto de objetivos generales se consideran asociados a otros módulos o bien tienen carácter transversal para todo el ciclo.

RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN

Los resultados de aprendizaje, para el módulo de **Programación**, establecidos en la legislación vigente (ME - Título DAM) *anexo 1*, son los siguientes:

RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

Criterios de evaluación:

- a) Se han identificado los bloques que componen la estructura de un programa informático.*
- b) Se han creado proyectos de desarrollo de aplicaciones*
- c) Se han utilizado entornos integrados de desarrollo.*
- d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.*
- e) Se ha modificado el código de un programa para crear y utilizar variables.*
- f) Se han creado y utilizado constantes y literales.*
- g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.*
- h) Se ha comprobado el funcionamiento de las conversiones de tipos explícitos e implícitos.*
- i) Se han introducido comentarios en el código.*

RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

Criterios de evaluación:

- a) Se han identificado los fundamentos de la programación orientada a objetos.*
- b) Se han escrito programas simples.*

- c) Se han instanciado objetos a partir de clases predefinidas.*
- d) Se han utilizado métodos y propiedades de los objetos.*
- e) Se han escrito llamadas a métodos estáticos.*
- f) Se han utilizado parámetros en la llamada a métodos.*
- g) Se han incorporado y utilizado librerías de objetos.*
- h) Se han utilizado constructores.*
- i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.*

RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

Criterios de evaluación:

- a) Se ha escrito y probado código que haga uso de estructuras de selección.*
- b) Se han utilizado estructuras de repetición.*
- c) Se han reconocido las posibilidades de las sentencias de salto.*
- d) Se ha escrito código utilizando control de excepciones.*
- e) Se han creado programas ejecutables utilizando diferentes estructuras de control.*
- f) Se han probado y depurado los programas.*
- g) Se ha comentado y documentado el código.*

RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

Criterios de evaluación:

- a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.*
- b) Se han definido clases.*
- c) Se han definido propiedades y métodos.*
- d) Se han creado constructores.*
- e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.*
- f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.*
- g) Se han definido y utilizado clases heredadas.*
- h) Se han creado y utilizado métodos estáticos.*
- i) Se han definido y utilizado interfaces.*
- j) Se han creado y utilizado conjuntos y librerías de clases.*

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Criterios de evaluación:

- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.*
Se han aplicado formatos en la visualización de la información.
- b) Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.*
- c) Se han utilizado ficheros para almacenar y recuperar información.*
- d) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.*
- e) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.*
- f) Se han programado controladores de eventos.*

g) Se han escrito programas que utilicen interfaces gráficas para la entrada y salida de información.

RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Criterios de evaluación:

- a) Se han escrito programas que utilicen arrays.*
- b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.*
- c) Se han utilizado listas para almacenar y procesar información.*
- d) Se han utilizado iteradores para recorrer los elementos de las listas.*
- e) Se han reconocido las características y ventajas de cada una de la colecciones de datos disponibles.*
- f) Se han creado clases y métodos genéricos.*
- g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.*
- h) Se han identificado las clases relacionadas con el tratamiento de documentos XML.*
- i) Se han realizado programas que realicen manipulaciones sobre documentos XML.*

RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

Criterios de evaluación:

- a) Se han identificado los conceptos de herencia, superclase y subclase.*
- b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.*
- c) Se ha reconocido la incidencia de los constructores en la herencia.*
- d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.*
- e) Se han diseñado y aplicado jerarquías de clases.*
- f) Se han probado y depurado las jerarquías de clases.*
- g) Se han realizado programas que implementen y utilicen jerarquías de clases.*
- h) Se ha comentado y documentado el código.*

RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

Criterios de evaluación:

- a) Se han identificado las características de las bases de datos orientadas a objetos.*
- b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.*
- c) Se han instalado sistemas gestores de bases de datos orientados a objetos.*
- d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.*
- e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.*
- f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.*

- g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.*
- h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.*

RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

Criterios de evaluación:

- a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.*
- b) Se han programado conexiones con bases de datos.*
- c) Se ha escrito código para almacenar información en bases de datos.*
- d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.*
- e) Se han efectuado borrados y modificaciones sobre la información almacenada.*
- f) Se han creado aplicaciones que ejecuten consultas sobre bases de datos.*
- g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.*

OBJETIVOS GENERALES DEL MÓDULO ASOCIADOS A RESULTADOS DE APRENDIZAJE

RESULTADO APRENDIZAJE	OBJETIVO GENERAL
<i>RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.</i>	<i>e) Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i>
<i>RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.</i>	<i>e) Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i>
<i>RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</i>	<i>e) Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i> <i>r) Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i>
<i>RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</i>	<i>e) Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i> <i>l) Valorar y emplear herramientas específicas, atendiendo a la estructura de los contenidos, para crear tutoriales, manuales de usuario y otros documentos asociados a una aplicación.</i> <i>m) Seleccionar y emplear técnicas y herramientas, evaluando la utilidad de los asistentes de instalación generados, para empaquetar aplicaciones.</i> <i>r) Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i> <i>s) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.</i>

<p>RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.</p>	<p>e) <i>Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i></p> <p>r) <i>Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i></p> <p>s) <i>Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.</i></p>
<p>RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.</p>	<p>e) <i>Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i></p> <p>r) <i>Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i></p> <p>s) <i>Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.</i></p>
<p>RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.</p>	<p>e) <i>Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i></p> <p>l) <i>Valorar y emplear herramientas específicas, atendiendo a la estructura de los contenidos, para crear tutoriales, manuales de usuario y otros documentos asociados a una aplicación.</i></p> <p>m) <i>Seleccionar y emplear técnicas y herramientas, evaluando la utilidad de los asistentes de instalación generados, para empaquetar aplicaciones.</i></p> <p>r) <i>Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i></p> <p>s) <i>Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.</i></p>
<p>RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.</p>	<p>e) <i>Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i></p> <p>r) <i>Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i></p>
<p>RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.</p>	<p>e) <i>Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.</i></p> <p>r) <i>Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.</i></p>

COMPETENCIAS PROFESIONALES, PERSONALES Y SOCIALES ASOCIADAS A RESULTADOS DE APRENDIZAJE

RESULTADO APRENDIZAJE	COMPETENCIA
<p>RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.</p>	<p>e) <i>Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</i></p>

<p>RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p>
<p>RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</p>	<p>r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.</p>
<p>RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p> <p>h) Desarrollar interfaces gráficos de usuario interactivos y con la usabilidad adecuada, empleando componentes visuales estándar o implementando componentes visuales específicos.</p> <p>l) Crear tutoriales, manuales de usuario, de instalación, de configuración y de administración, empleando herramientas específicas.</p> <p>m) Empaquetar aplicaciones para su distribución preparando paquetes auto instalables con asistentes incorporados.</p> <p>r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.</p> <p>s) Desplegar y distribuir aplicaciones en distintos ámbitos de implantación verificando su comportamiento y realizando las modificaciones necesarias.</p>
<p>RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p> <p>h) Desarrollar interfaces gráficos de usuario interactivos y con la usabilidad adecuada, empleando componentes visuales estándar o implementando componentes visuales específicos.</p>
<p>RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p> <p>r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.</p>
<p>RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p> <p>h) Desarrollar interfaces gráficos de usuario interactivos y con la usabilidad adecuada, empleando componentes visuales estándar o implementando componentes visuales específicos.</p> <p>l) Crear tutoriales, manuales de usuario, de instalación, de configuración y de administración, empleando herramientas específicas.</p> <p>m) Empaquetar aplicaciones para su distribución preparando paquetes auto instalables con asistentes incorporados.</p> <p>r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.</p> <p>s) Desplegar y distribuir aplicaciones en distintos</p>

	<p>ámbitos de implantación verificando su comportamiento y realizando las modificaciones necesarias.</p>
<p>RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p> <p>r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.</p>
<p>RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.</p>	<p>e) Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías (sic) y herramientas adecuados a las especificaciones.</p> <p>r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.</p>

CONTENIDOS Y UNIDADES DE TRABAJO

Los contenidos se organizan en ocho U.T. distribuidas en tres bloques que completan 230 horas durante un curso. Cada U.T. desarrolla los contenidos de tipo conceptual, procedimental, y actitudinal; necesarios para cubrir las realidades de aprendizaje, objetivos generales de ciclo asociados en el análisis y los específicamente establecidos para cada unidad.

Bloque 1: INICIACIÓN A LA PROGRAMACIÓN

Constituido por la UT1. Tienen carácter de introducción a las técnicas, fases y fundamentos metodológicos y de estructuración. Establece las bases clásicas para comprender cómo ha evolucionado la programación y hace posible la transición hacia las verdaderas técnicas y fundamentos de la programación de ordenadores actual.

U.T. 1 FUNDAMENTOS DE LA PROGRAMACIÓN ESTRUCTURADA

Bloque 2: INTRODUCCIÓN AL DESARROLLO ORIENTADO A OBJETOS Y LAS ESTRUCTURAS DE DATOS

Constituido por la UT2, UT3 y UT4. Tienen carácter conceptual y procedimental de nivel básico. Introduce las técnicas y paradigmas de la programación orientada a objetos. Hace una aproximación al desarrollo de programas y las estructuras de datos desde una perspectiva orientada a objetos.

U.T. 2 INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

U.T. 3 ARQUITECTURA DE UNA APLICACIÓN : PARTES, MÓDULOS Y COMPONENTES

U.T. 4 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS

Bloque 3: DESARROLLO BÁSICO DE APLICACIONES

Constituido por la UT5, UT6, UT7 y UT8. Tienen carácter conceptual y procedimental, profundiza en los conceptos y aspectos procedimentales ya planteados en el bloque anterior e introduce nociones propias de ámbitos de programación más especializados y avanzados.

U.T. 5 ENCAPSULACIÓN, HERENCIA Y POLIMORFISMO

U.T. 6 TRATAMIENTO DE EXCEPCIONES

U.T. 7 PERSISTENCIA BÁSICA Y ACCESO A DATOS

U.T. 8 DESARROLLO BÁSICO DE INTERFACES DE USUARIO

De acuerdo a la modificación publicada en el BORM, se aplica la Orden de la Consejería de Educación de 10 de septiembre de 2022 por la que se modifican los currículos de los ciclos formativos de formación profesional de la familia profesional de Informática y comunicaciones en el ámbito de la Región de la Comunidad Autónoma de la Región de Murcia.

PLANIFICACIÓN DE UT's

Es difícil desarrollar las competencias asociadas a este módulo, de forma completamente secuencial; es más factible de forma sistémica o conjunta, poniendo el énfasis en determinados contenidos dentro de cada unidad de trabajo, pero sin perder la visión del conjunto.

PRIMER TRIMESTRE (88 HORAS)

U.T. 1 FUNDAMENTOS DE LA PROGRAMACIÓN ESTRUCTURADA

U.T. 2 INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

U.T. 3 ARQUITECTURA DE UNA APLICACIÓN : PARTES, MÓDULOS Y COMPONENTES (I)

U.T. 4 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS (I)

SEGUNDO TRIMESTRE (88 HORAS)

U.T. 3 ARQUITECTURA DE UNA APLICACIÓN : PARTES, MÓDULOS Y COMPONENTES (II)

U.T. 4 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS (II)

U.T. 5 ENCAPSULACIÓN, HERENCIA Y POLIMORFISMO (I)

U.T. 7 TRATAMIENTO DE EXCEPCIONES (I)

TERCER TRIMESTRE (54 HORAS)

U.T. 5 ENCAPSULACIÓN, HERENCIA Y POLIMORFISMO (II)

U.T. 6 TRATAMIENTO DE EXCEPCIONES (II)

U.T. 7 PERSISTENCIA BÁSICA Y ACCESO A DATOS

U.T. 8 DESARROLLO BÁSICO DE INTERFACES DE USUARIO

UNIDADES DE TRABAJO

U.T. 1 FUNDAMENTOS DE LA PROGRAMACIÓN ESTRUCTURADA	56 horas
REALIZACIONES DE APRENDIZAJE	
<p>RA1: RECONOCE LA ESTRUCTURA DE UN PROGRAMA INFORMÁTICO, IDENTIFICANDO Y RELACIONANDO LOS ELEMENTOS PROPIOS DEL LENGUAJE DE PROGRAMACIÓN UTILIZADO.</p> <p>RA2. ESCRIBE Y PRUEBA PROGRAMAS SENCILLOS, RECONOCIENDO Y APLICANDO LOS FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.</p> <p>RA3. ESCRIBE Y DEPURA CÓDIGO, ANALIZANDO Y UTILIZANDO LAS ESTRUCTURAS DE CONTROL DEL LENGUAJE.</p>	
OBJETIVOS DIDÁCTICOS	
<ol style="list-style-type: none"> 1. En relación con tareas específicas propuestas como ejercicios de aplicación de los conceptos básicos, utilizar las herramientas disponibles en el entorno de trabajo para escribir pequeños programas y comprender la estructura de un programa de ordenador, siendo capaces de explicar y reproducir y modificar su funcionamiento. 2. En relación con tareas específicas propuestas como ejercicios de aplicación de los conceptos básicos, utilizar las herramientas disponibles en el entorno de trabajo para escribir pequeños programas estructurados con varios métodos y comprender su funcionamiento, siendo capaces de explicarlos reproducirlos y modificarlos. 	
MÍNIMOS	
<ul style="list-style-type: none"> • Dado un programa simple, identificar cada uno de los elementos del lenguaje que aparecen. • Dado un enunciado, escribir de forma autónoma, en el entorno de programación habitual, programas sencillos que implementan algoritmos con estructuras de control alternativas y repetitivas con dos niveles de anidamiento. • Dado un enunciado, escribir de forma autónoma, en el entorno de programación habitual, programas básicos estructurados con varios métodos que implementan algoritmos típicos. 	
CONCEPTOS BASE	
<ul style="list-style-type: none"> • CONCEPTOS PREVIOS Y TERMINOLOGÍA • SISTEMAS INFORMÁTICOS • DESARROLLO DE APLICACIONES • LOS DATOS • PARADIGMAS DE LA PROGRAMACIÓN • METODOLOGÍAS • LENGUAJES • ENTORNOS OPERATIVOS DE PROGRAMACIÓN 	<ul style="list-style-type: none"> • LA PROGRAMACIÓN ESTRUCTURADA • PRINCIPIO DE LA ESTRUCTURACIÓN • ESTRUCTURAS DE CONTROL DE FLUJO • ESTRUCTURAS DE SELECCIÓN • ESTRUCTURAS ITERATIVAS • LOS MÉTODOS • RECURSIVIDAD BÁSICA • ASERCIONES
PROCEDIMIENTOS	ACTITUDES

- | | |
|--|---|
| <ul style="list-style-type: none"> • Búsqueda y organización de la información necesaria para definir un problema. • Análisis básico para la correcta interpretación de enunciados y problemas sencillos. • Utilización de libros, manuales y documentos desde la perspectiva de la necesidad concreta para resolver un problema. • Síntesis de soluciones básicas de ejercicios clasificados con tres niveles de dificultad, partiendo de explicaciones, demostraciones y ejemplos. Todo ello con herramientas y entornos profesionales de programación. • Documentación técnica de los programas. | <ul style="list-style-type: none"> • Manifestación de acuerdo, compromiso, participación y colaboración en el proceso de aprendizaje. • Motivación y predisposición para adquirir y refinar el pensamiento analítico-crítico necesario en la comprensión de problemas de gestión resolubles con un programa de ordenador. • Valoración de la importancia de desarrollar hábitos, rutinas y estilos de trabajo propios de los profesionales de la programación. |
|--|---|

CRITERIOS DE EVALUACIÓN

RA1

- a) Se han identificado los bloques que componen la estructura de un programa informático.
- b) Se han creado proyectos de desarrollo de aplicaciones.
- c) Se han utilizado entornos integrados de desarrollo.
- d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- e) Se ha modificado el código de un programa para crear y utilizar variables.
- f) Se han creado y utilizado constantes y literales.
- g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- h) Se ha comprobado el funcionamiento de las conversiones de tipos explícitos e implícitos.
- i) Se han introducido comentarios en el código.

RA2

- ~~a) Se han identificado los fundamentos de la programación orientada a objetos.~~
- b) Se han escrito programas simples.
- c) Se han instanciado objetos a partir de clases predefinidas.
- d) Se han utilizado métodos y propiedades de los objetos.
- e) Se han escrito llamadas a métodos estáticos.
- f) Se han utilizado parámetros en la llamada a métodos.
- g) Se han incorporado y utilizado librerías de objetos.
- ~~h) Se han utilizado constructores.~~
- i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

RA3

- a) Se ha escrito y probado código que haga uso de estructuras de selección.
- b) Se han utilizado estructuras de repetición.
- c) Se han reconocido las posibilidades de las sentencias de salto.
- ~~d) Se ha escrito código utilizando control de excepciones.~~
- e) Se han creado programas ejecutables utilizando diferentes estructuras de control.
- f) Se han probado y depurado los programas.
- g) Se ha comentado y documentado el código.

Notas

**U.T. 2 INTRODUCCIÓN A LA
PROGRAMACIÓN ORIENTADA A OBJETOS**

28 horas

REALIZACIONES DE APRENDIZAJE

RA2. ESCRIBE Y PRUEBA PROGRAMAS SENCILLOS, RECONOCIENDO Y APLICANDO LOS FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.

RA4. DESARROLLA PROGRAMAS ORGANIZADOS EN CLASES ANALIZANDO Y APLICANDO LOS PRINCIPIOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.

OBJETIVOS DIDÁCTICOS

1. En supuestos básicos, implementar algoritmos que resuelva el comportamiento previsto de una clase realizando las pruebas necesarias.
2. Teniendo en cuenta los modelos elaborados para un proyecto de software, con lenguajes y herramientas estándares, diseñar y desarrollar un prototipo de solución básica aplicando principios de POO.
3. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Explicar y describir los cuatro principios fundamentales de la orientación a objetos.
- Realizar un programa que crea y utiliza objetos a partir de clases disponibles en librerías.
- Realizar un programa que crea y utiliza objetos a partir de clases básicas definidas por el programador.
- Realizar un programa que utiliza clases definidas por el programador aplicando encapsulación básica para disponer de constructores típicos, protección de atributos.

CONCEPTOS BASE

- | | |
|---|--|
| <ul style="list-style-type: none"> • PARADIGMA ORIENTADO A OBJETOS • CLASES Y OBJETOS • ABSTRACCIÓN • ENCAPSULACIÓN | <ul style="list-style-type: none"> • DISEÑO DE CLASES • ATRIBUTOS Y ESTADO DE UN OBJETO • MÉTODOS Y COMPORTAMIENTO DE UNA CLASE • HERENCIA • POLIMORFISMO |
|---|--|

PROCEDIMIENTOS

- Diseño y desarrollo de prototipos de programas OO.
- Búsqueda, organización, estudio y valoración de la información necesaria para el refinamiento y ampliación de los prototipos.
- Investigación y refinamiento de diseños y prototipos con POO.
- Elaboración de documentación técnica del proyecto de software.
- Interpretación de modelos de procesos y de datos.
- Diseño y desarrollo de prototipos de programas.
- Búsqueda, organización y estudio de la información necesaria para la implementación de algoritmos
- Investigación y refinamiento de diseños y prototipos.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Utilización de la terminología adecuada.
- Implicación en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.
- Propuesta e investigación nuevas características de POO para los modelos y prototipos diseñados y desarrollados.
- Utilización de la terminología adecuada.
- Integración en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.

CRITERIOS DE EVALUACIÓN

RA2

- a) *Se han identificado los fundamentos de la programación orientada a objetos.*
- b) *Se han escrito programas simples.*
- c) *Se han instanciado objetos a partir de clases predefinidas.*
- d) *Se han utilizado métodos y propiedades de los objetos.*
- e) *Se han escrito llamadas a métodos estáticos.*
- f) *Se han utilizado parámetros en la llamada a métodos.*
- g) *Se han incorporado y utilizado librerías de objetos.*
- h) *Se han utilizado constructores.*
- i) *Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.*

RA4

- a) *Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.*
- b) *Se han definido clases.*
- c) *Se han definido propiedades y métodos.*
- d) *Se han creado constructores.*
- e) *Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.*
- f) *Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.*
- ~~g) *Se han definido y utilizado clases heredadas.*~~
- h) *Se han creado y utilizado métodos estáticos.*
- ~~i) *Se han definido y utilizado interfaces.*~~
- j) *Se han creado y utilizado conjuntos y librerías de clases.*

Notas

**U.T. 3 ARQUITECTURA DE UNA
 APLICACIÓN : PARTES, MÓDULOS Y
 COMPONENTES**

21 horas

REALIZACIONES DE APRENDIZAJE

RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

OBJETIVOS DIDÁCTICOS

1. En un proyecto de software, identificar las tareas llevadas a cabo en el proceso que se pretende solucionar; definiendo un modelo básico que represente las tareas esenciales requeridas.
2. Coherentemente con las especificaciones de un proyecto de software, elegir y definir estructuras de datos necesarios para la resolución del problema; definiendo un modelo datos que represente todas las necesidades establecidas.
3. Teniendo en cuenta los modelos elaborados para un proyecto de software, con herramientas estándar, diseñar y desarrollar un prototipo de solución básica aplicando principios estructurados.
4. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Refactorizar un programa organizado en tres capas creando las clases necesarias; una de ellas se especializa en el almacenamiento de datos en un fichero, otra en la interacción con el usuario y otra implementa la lógica principal del programa.

CONCEPTOS BASE

- | | |
|--|---|
| <ul style="list-style-type: none"> • MODELOS Y ARQUITECTURAS DE APLICACIONES • PROTOTIPOS • CLASES Y PAQUETES • BIBLIOTECAS DE COMPONENTES • ATRIBUTOS • MÉTODOS | <ul style="list-style-type: none"> • ACOPLAMIENTO • INTEGRACIÓN • PRUEBAS, DEPURACIÓN Y DOCUMENTACIÓN DE LA APLICACIÓN |
|--|---|

PROCEDIMIENTOS

- Elaboración de modelos de procesos y de datos.
- Diseño y desarrollo de prototipos de programas.
- Búsqueda, organización y estudio de la información necesaria para la elaboración de modelos.
- Investigación y refinamiento de diseños y prototipos.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Utilización de la terminología adecuada.
- Integración en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.

CRITERIOS DE EVALUACIÓN

RA4

- a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- b) Se han definido clases.
- c) Se han definido propiedades y métodos.
- d) Se han creado constructores.
- e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- g) Se han definido y utilizado clases heredadas.
- h) Se han creado y utilizado métodos estáticos.
- ~~i) Se han definido y utilizado interfaces.~~
- j) Se han creado y utilizado conjuntos y librerías de clases.

RA5

- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
Se han aplicado formatos en la visualización de la información.
- b) Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
- c) Se han utilizado ficheros para almacenar y recuperar información.
- d) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- ~~e) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.~~
- ~~f) Se han programado controladores de eventos.~~
- ~~g) Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.~~

Notas

U.T. 4 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS

21 horas

REALIZACIONES DE APRENDIZAJE

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

OBJETIVOS DIDÁCTICOS

1. En un proyecto de software, identificar las tareas llevadas a cabo en el proceso que se pretende solucionar; definiendo un modelo básico que represente las tareas esenciales requeridas.
2. Coherentemente con las especificaciones de un proyecto de software, elegir y definir estructuras de datos necesarios para la resolución del problema; definiendo un modelo datos que represente todas las necesidades establecidas.
3. Teniendo en cuenta los modelos elaborados para un proyecto de software, con herramientas estándar, diseñar y desarrollar un prototipo de solución básica aplicando principios estructurados.
4. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Realizar un programa que utiliza arrays en algoritmos típicos.
- Realizar un programa que utiliza cadenas de caracteres para almacenar información que se procesa con expresiones regulares.
- Realizar un programa que utiliza control del formato de salida al mostrar la información.
- Implementar estructuras dinámicas basadas en listas enlazadas y sus operaciones típicas.
- Realizar un programa organizado en tres capas en la que una de ellas resuelve el almacenamiento de datos con estructuras dinámicas de la librería estándar del lenguaje.
- Realizar un programa que en una de sus capas implementa altas/bajas/modificaciones de objetos almacenados en colecciones que persisten en fichero.

CONCEPTOS BASE

- | | |
|---|--|
| <ul style="list-style-type: none"> • LOS DATOS SIMPLES Y PRIMITIVOS • OBJETOS, PUNTEROS Y REFERENCIAS • LOS DATOS ESTRUCTURADOS ESTÁTICOS • ESTRUCTURAS DE DATOS DINÁMICAS • CADENAS DE CARACTERES. EXPRESIONES REGULARES • GENERICIDAD | <ul style="list-style-type: none"> • ARRAYS DINÁMICOS • DATOS EXTERNOS (FICHEROS) • PERSISTENCIA DE OBJETOS • CONJUNTOS Y DICCIONARIOS • OPERACIONES AGREGADAS: FILTRADO, REDUCCIÓN Y RECOLECCIÓN |
|---|--|

PROCEDIMIENTOS

- Elaboración de modelos de procesos y de datos.
- Diseño y desarrollo de prototipos de programas.
- Búsqueda, organización y estudio de la información necesaria para la elaboración de modelos.
- Investigación y refinamiento de diseños y prototipos.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Utilización de la terminología adecuada.
- Integración en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.

CRITERIOS DE EVALUACIÓN

RA5

- a) *Se ha utilizado la consola para realizar operaciones de entrada y salida de información.*
- Se han aplicado formatos en la visualización de la información.*
- b) *Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.*
- c) *Se han utilizado ficheros para almacenar y recuperar información.*
- d) *Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.*
- ~~e) *Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.*~~
- ~~f) *Se han programado controladores de eventos.*~~
- ~~g) *Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.*~~

RA6

- a) *Se han escrito programas que utilicen arrays.*
- b) *Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.*
- c) *Se han utilizado listas para almacenar y procesar información.*
- d) *Se han utilizado iteradores para recorrer los elementos de las listas.*
- e) *Se han reconocido las características y ventajas de cada una de la colecciones de datos disponibles.*
- f) *Se han creado clases y métodos genéricos.*
- g) *Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.*
- h) *Se han identificado las clases relacionadas con el tratamiento de documentos XML.*
- i) *Se han realizado programas que realicen manipulaciones sobre documentos XML.*

Notas

U.T. 5 ENCAPSULACIÓN, HERENCIA Y POLIMORFISMO

28 horas

REALIZACIONES DE APRENDIZAJE

RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

OBJETIVOS DIDÁCTICOS

1. Teniendo en cuenta los modelos elaborados para un proyecto de software, con lenguajes y herramientas estándares, diseñar y desarrollar un prototipo de solución óptimo aplicando todos los principios de POO.
2. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Realizar un programa que implementa y utiliza una jerarquía de clases, teniendo en cuenta el efecto sobre los constructores.
- Realizar un programa que implementa y utiliza una jerarquía de clases con clases abstractas e interfaces, teniendo en cuenta la redefinición de métodos.

CONCEPTOS BASE

- PARADIGMA ORIENTADO A OBJETOS
- HERENCIA
- POLIMORFISMO
- RELACIONES DE COMPOSICIÓN, AGREGACIÓN Y ESPECIALIZACIÓN
- INTERFACES

PROCEDIMIENTOS

- Diseño y desarrollo de prototipos de programas OO.
- Refinamiento y ampliación de los modelos y prototipos.
- Refinamiento de diseños y prototipos con POO.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Utilización de la terminología adecuada.
- Implicación en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.
- Propuesta e investigación nuevas características de POO para los modelos y prototipos diseñados y desarrollados.

CRITERIOS DE EVALUACIÓN

RA7

- a) Se han identificado los conceptos de herencia, superclase y subclase.*
- b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.*
- c) Se ha reconocido la incidencia de los constructores en la herencia.*
- d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.*
- e) Se han diseñado y aplicado jerarquías de clases.*
- f) Se han probado y depurado las jerarquías de clases.*
- g) Se han realizado programas que implementen y utilicen jerarquías de clases.*
- h) Se ha comentado y documentado el código.*

Notas

U.T. 6 TRATAMIENTO DE EXCEPCIONES

35 horas

REALIZACIONES DE APRENDIZAJE

RA3. ESCRIBE Y DEPURA CÓDIGO, ANALIZANDO Y UTILIZANDO LAS ESTRUCTURAS DE CONTROL DEL LENGUAJE.

OBJETIVOS DIDÁCTICOS

1. Teniendo en cuenta los modelos y prototipo desarrollado en un proyecto de software, con lenguajes y herramientas de POO, incorporar características de tratamiento de excepciones al prototipo de solución; aplicando los cambios y refinando el prototipo.
2. Teniendo en cuenta los modelos y prototipo desarrollado en un proyecto de software, con lenguajes y herramientas de POO, incorporar características de multitarea al prototipo de solución; aplicando los cambios y refinando el prototipo.
3. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Realizar un programa que utiliza excepciones de las previstas en el lenguaje para gestionar los errores.
- Realizar un programa que implemente nuevas excepciones específicas para gestionar los errores.

CONCEPTOS BASE

- TRATAMIENTO DE EXCEPCIONES
 - LANZAMIENTO
 - MANEJADOR
 - JERARQUÍA
 - Ventajas

PROCEDIMIENTOS

- Modificación, refinamiento y depuración de prototipos de programas OO.
- Búsqueda, organización, estudio y valoración de la información necesaria para el refinamiento y ampliación de los prototipos.
- Investigación y refinamiento de diseños y prototipos con POO.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Utilización de la terminología adecuada.
- Implicación en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejoras.
- Propuesta e investigación nuevas características de POO para los modelos y prototipos diseñados y desarrollados.

CRITERIOS DE EVALUACIÓN

RA3

- ~~a) Se ha escrito y probado código que haga uso de estructuras de selección.~~
- ~~b) Se han utilizado estructuras de repetición.~~
- ~~c) Se han reconocido las posibilidades de las sentencias de salto.~~
- ~~d) Se ha escrito código utilizando control de excepciones.~~
- ~~e) Se han creado programas ejecutables utilizando diferentes estructuras de control.~~
- ~~f) Se han probado y depurado los programas.~~
- ~~g) Se ha comentado y documentado el código.~~

Notas

U.T.7 PERSISTENCIA BÁSICA Y ACCESO A DATOS

28 horas

REALIZACIONES DE APRENDIZAJE

RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

OBJETIVOS DIDÁCTICOS

1. Teniendo en cuenta los modelos y prototipo desarrollado en un proyecto de software, con lenguajes y herramientas de POO, incorporar características de funcionamiento en red y acceso a bases de datos remotas al prototipo de solución; aplicando los cambios y refinando el prototipo.
2. Teniendo en cuenta los modelos y el prototipo desarrollado en un proyecto de software, con lenguajes y herramientas de POO, diseñar y desarrollar un nuevo prototipo en versión applet; aplicando los cambios y refinando el nuevo prototipo.
3. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Realizar un programa que implementa persistencia de datos sobre una base de datos orientada a objetos.
- Realizar un programa que implementa persistencia de datos sobre una base de datos SQL.

CONCEPTOS BASE

- ARQUITECTURA CLIENTE / SERVIDOR
- LA ABSTRACCIÓN SOCKET
- SERVICIOS
- PROGRAMACIÓN PARA SERVICIOS BASADOS EN WEB
- Acceso a bases de datos

PROCEDIMIENTOS

- Modificación, refinamiento y depuración de prototipos de programas OO.
- Búsqueda, organización, estudio y valoración de la información necesaria para el refinamiento, modificación y ampliación de los prototipos.
- Investigación y refinamiento de diseños y prototipos con POO.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Utilización de la terminología adecuada.
- Integración en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.
- Propuesta e investigación nuevas características de POO para los modelos y prototipos diseñados y desarrollados.

CRITERIOS DE EVALUACIÓN

RA8

- a) *Se han identificado las características de las bases de datos orientadas a objetos.*
- b) *Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.*
- c) *Se han instalado sistemas gestores de bases de datos orientados a objetos.*
- d) *Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.*
- e) *Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.*
- f) *Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.*
- g) *Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.*
- h) *Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.*

RA9

- a) *Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.*
- b) *Se han programado conexiones con bases de datos.*
- c) *Se ha escrito código para almacenar información en bases de datos.*
- d) *Se han creado programas para recuperar y mostrar información almacenada en bases de datos.*
- e) *Se han efectuado borrados y modificaciones sobre la información almacenada.*
- f) *Se han creado aplicaciones que ejecuten consultas sobre bases de datos.*
- g) *Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.*

Notas

U.T.8 DESARROLLO BÁSICO DE INTERFACES DE USUARIO

28 horas

REALIZACIONES DE APRENDIZAJE

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

OBJETIVOS DIDÁCTICOS

1. Teniendo en cuenta los modelos y especificaciones de un proyecto de software, con lenguajes de POO y herramientas RAD, diseñar y desarrollar prototipos de programas; siendo capaz de aplicar y seguir criterios metodológicos preestablecidos y justificados.
2. En relación con una tarea específica dentro de un proyecto de software, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para el seguimiento del proyecto.

MÍNIMOS

- Realizar un programa que implementa un interfaz gráfico de usuario utilizando componentes estándar del lenguaje.

CONCEPTOS BASE

- ARQUITECTURAS DE APLICACIONES
- PROGRAMACIÓN DIRIGIDA POR EVENTOS
- LOS COMPONENTES VISUALES
- Desarrollo de aplicaciones completas en entornos visuales y oo

PROCEDIMIENTOS

- Búsqueda, organización, estudio y valoración de la información necesaria para el diseño de visual de aplicaciones.
- Diseño y desarrollo visual de aplicaciones.
- Búsqueda, organización, estudio y valoración de la información necesaria para el refinamiento, modificación y ampliación de los prototipos.
- Investigación y refinamiento de diseños y prototipos con POO y programación visual.
- Elaboración de documentación técnica del proyecto de software.

ACTITUDES

- Manifestación de acuerdo y compromiso de participación y colaboración.
- Utilización de la terminología adecuada.
- Integración en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.
- Propuesta e investigación nuevas características de POO para los modelos y prototipos diseñados y desarrollados.

CRITERIOS DE EVALUACIÓN

RA5

- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.*
- b) Se han aplicado formatos en la visualización de la información.*
- c) Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.*
- d) Se han utilizado ficheros para almacenar y recuperar información.*
- e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.*
- f) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.*
- g) Se han programado controladores de eventos.*
- h) Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.*

Notas

METODOLOGÍA

CRITERIOS

Se trata en la parte general de la programación del ciclo formativo.

ASPECTOS CONCRETOS

La estrategia metodológica del módulo, tiene en cuenta el enfoque educativo de Aprendizaje Orientado a Proyectos (AOP). El trabajo a partir de proyectos, bajo el enfoque educativo de AOP, se fundamenta pedagógicamente en los siguientes principios:

Los proyectos se enfocan en problemas que inducen a los estudiantes a enfrentarse a los conceptos y principios básicos de una o varias disciplinas. Pueden encaminarse a un tema en particular o a formar enlaces entre una o más disciplinas.

Los proyectos implican a los estudiantes en un proceso de investigación creadora. Las actividades han de permitir la transformación y construcción de conocimientos que conducen a la adquisición de nuevos conocimientos, nuevas capacidades para mejorar las habilidades del alumnado.

Los proyectos son dirigidos, en gran medida, por los mismos estudiantes implicando una mayor autonomía y participación activa en el proceso de toma de decisiones; adquiriendo así mayores responsabilidades que en el proceso de enseñanza tradicional. Cada proyecto no tiene un resultado predeterminado.

Los proyectos abordan situaciones razonablemente reales y no simuladas.

Las actividades didácticas se organizan en torno a unas necesidades o requisitos de proyecto, no a los contenidos curriculares, como ocurre tradicionalmente con frecuencia. En este sentido los contenidos curriculares deben reorganizarse y vincularse al proyecto a través de un diseño adecuado de los objetivos y los requisitos. En AOP el elemento organizador no es directamente el currículo sino el proyecto.

Esta estrategia implica las siguientes ventajas didácticas:

- Vincular los contenidos de un módulo en uno o varios proyectos propicia la visión global y justificada de los mismos; hace *“pensar globalmente y actuar localmente”*.
- Fomenta de forma natural el trabajo colaborativo para el análisis de tareas relacionadas con la elaboración del proyecto, favorece el intercambio de experiencias, estimula habilidades sociales y cognitivas en el grupo; facilitando el aprendizaje de otros y con otros. Para este fin son esenciales los medios de comunicación y las TIC aplicadas a la docencia que deben estar disponibles en el aula.
- La elaboración del proyecto promueve la capacidad en los participantes de aprender a aprender, partiendo siempre de las potencialidades y necesidades individuales de cada uno.
- Las actividades formativas previstas buscan generar tanto la motivación por comprobar lo aprendido como las dudas y problemas reales que se encuentran en el quehacer diario y que posteriormente se convierten en revisiones y mejoras del proceso de aprendizaje.

Como conclusión básica se puede decir que el desarrollo o adquisición de cualquier competencia profesional implica necesariamente actitudes, valores, conocimientos, capacidades y habilidades inherentes e inseparables entre sí (naturaleza sistémica). Los proyectos permiten una de las mejores aproxima-

ciones –holísticas– para el desarrollo integral de competencias; permiten hacer la consideración:

“la mejor forma de aprender a hacer algo, por ejemplo desarrollar un programa de ordenador, es realizar las mismas actividades que se harían para demostrar que ya se saben hacer”

Es una especie de juego de roles en donde se pretende actuar como si ya se fuera competente en una determinada tarea. A la vez que se adquieren los conocimientos, capacidades y habilidades claves de la tarea; se refuerzan y se avanza.

El enfoque de AOP se aplica en el curso virtual previsto para el módulo **Programación** en la plataforma Moodle del centro.

AGRUPAMIENTO DEL ALUMNADO

Se procurará en todo momento que cada alumno disponga de su propio ordenador para trabajar, dado que según muchas experiencias, es muy negativo que dos o más personas compartan un mismo puesto de trabajo.

En el caso de que el número de alumnos sea superior al de ordenadores en el aula, se propondrá un desdoble del grupo, si los medios materiales lo permiten.

Si el número de alumnos en el aula es elevado, se transmitirá a la Jefatura de Estudios la necesidad de un profesor de apoyo, para procurar una atención personalizada que, cuando esta circunstancia se da, es imposible de conseguir. De esta manera los alumnos con dificultades en el aprendizaje se verían muy beneficiados.

MATERIALES, RECURSOS, ESPACIO DOCENTE

MATERIALES Y RECURSOS DIDÁCTICOS

- Curso Moodle utilizado para los módulos de **Programación**: <http://moodle.iescierva.net/course/view.php?id=400>
- Aula de Informática cableada Gbit.
- Servidores de red con recursos didácticos para la docencia y apoyo para las actividades: Git y otros.
- Puestos de trabajo con Linux o Windows.
- Software para modelado UML y entornos integrados de desarrollo de aplicaciones IDE.
- Software de ofimática para elaborar trabajos y documentación.
- Software para diagramación y elaborar mapas conceptuales.
- Software para control y presentaciones en red del aula.
- Cuentas de usuario de acceso a la red de trabajo.
- Libros de consulta y documentación electrónica.
- Sistemas audiovisuales de apoyo, como video, cañón de proyección y pantalla.

- Acceso a Internet, otros servicios en Intranet y red local.

BIBLIOGRAFÍA

Se usarán habitualmente contenidos disponibles en Moodle y en la red local del Departamento de Informática o en Internet; por lo que no se propone el uso de libro de texto específico obligatorio; tan sólo se va a hacer referencia a un libro que cubriría todos los contenidos programados y que en parte será utilizado con carácter opcional juntos a otros recursos y manuales:

- Apuntes disponibles en el curso Moodle **Programación y Entornos de Desarrollo**: <http://moodle.iescierva.net/mod/page/view.php?id=20273>
- Otros recursos, manuales descargables y enlaces externos disponibles en la **BIBLIOTECA DE RECURSOS Y MANUALES**: <http://moodle.iescierva.net/mod/page/view.php?id=20273> del curso Moodle **Programación y Entornos de Desarrollo**.

DISTRIBUCIÓN DEL ESPACIO Y EL TIEMPO DOCENTE

Se velará por el cumplimiento de las normas de uso de los ordenadores y talleres del Departamento.

Se intentará que las horas de este módulo en un mismo día sean consecutivas, para evitar la pérdida de tiempo que se produce durante los arranques y paradas de los sistemas.

MEDIDAS PARA ESTIMULAR EL INTERÉS Y EL HÁBITO DE LECTURA Y LA CAPACIDAD DEL ALUMNADO PARA EXPRESARSE CORRECTAMENTE

Este apartado se detalla en la parte general de la programación del ciclo formativo.

CRITERIOS, PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN

Los apartados bajo este epígrafe no especificados expresamente, se detallan en la parte general de la programación del ciclo formativo.

CRITERIOS DE CALIFICACIÓN

Como criterio general se establece que cualquier calificación global del módulo, final o parcial, se obtendrá por la aportación, la acumulación y el promedio de las calificaciones correspondientes a distintas actividades establecidas específicamente, o no, para la evaluación que se hayan informado debidamente.

Los criterios de evaluación/valoración, indicadores detallados y tipos de pruebas serán siempre conocidos y difundidos por el profesor responsable del módulo en el curso de Moodle asociado, en concreto en la sección de *guía de*

evaluación. Estos criterios están establecidos y dispuestos en cada actividad evaluada.

CONSIDERACIONES

1. La asistencia o seguimiento de las actividades es obligatorio siendo determinante para la valoración del trabajo diario en clase.
2. Todas las actividades docentes que se organicen tienen carácter obligatorio para todos los alumnos/as, y como ya se ha indicado, son objeto de evaluación.
3. Las actividades que impliquen algún producto sometido a plazo de entrega deberán presentarse en tiempo y forma; en caso contrario, si se admite su entrega fuera de plazo, obtendrán una nota máxima del 50% de la ponderación total prevista.
4. No serán aceptados los trabajos retrasados; una vez que éstos hayan sido puestos en común, revisados o resueltos en clase; considerándose, en ese caso, que se renuncia explícitamente a aportarlos como evidencias para una evaluación positiva, con la consiguiente merma o perjuicio en la calificación resultante.
5. Los cuestionarios, controles, exámenes y en general cualquier prueba objetiva individual o en grupo se realizará siempre en convocatoria única, planificada y previamente anunciada al menos con dos días de tiempo. Se realizarán con las siguientes condiciones:
 1. La periodicidad será acorde y coincidente con el final de cada una de las prácticas previstas.
 2. La segunda convocatoria es automática para los no presentados; actuando de convocatoria única, a efecto de notas, si la falta está justificada. Es opcional, bajo petición expresa, para quienes quieran mejorar o recuperar resultados de notas. A efectos de notas, siempre se calculará media simple entre las dos convocatorias.
 3. En las pruebas objetivas; I@s alumn@s no presentad@s por causas de fuerza mayor, justificadas documentalmente y admitida su validez/veracidad, obtendrán cero, por no presentado, pero no se les computará a efectos del cálculo de medias en el libro de calificaciones del curso en Moodle.
 4. En pruebas objetivas; I@s alumn@s no presentados por causas injustificadas o por justificación no admitida, obtendrán cero por no presentado, que se computará a efectos del cálculo de medias en el libro de calificaciones del curso en Moodle.
 6. En caso de observarse que la participación y aportación en los trabajos de grupo de algún miembro del grupo no es la adecuada y equiparable a la del resto de colaboradores, se indagará en las causas a fin de tomar las medidas oportunas, llegándose incluso a invalidar su participación en la actividad a efectos de notas y evaluación. La situación descrita repercutirá negativamente en la valoración de actitud, debiéndose rehacer individualmente la totalidad de las tareas y actividades de grupo.

MODALIDAD SEMIPRESENCIAL O TOTALMENTE A DISTANCIA

Se seguirá el mismo proceso que en la modalidad presencial normal anteriormente descrita. Se podrá solicitar al alumnado la defensa de los ejercicios y prácticas mediante una entrevista personal online o telefónica en la que el profesor podrá comprobar si los objetivos previstos se han cumplido.

PONDERACIÓN DE CONTENIDOS

Se seguirán los aspectos y apartados establecidos en el **modelo2** de la parte general de la programación del ciclo formativo.

Actividad / ponderación		Aspectos / ponderación				
		10%	10%	60%	60%	30%
		Actitud, comportamiento y responsabilidad. Asistencia / puntualidad. Colaboración, participación e intervención en el aula.	Trabajo diario y realización completa de ejercicios y prácticas.	Aplicación de los contenidos estudiados.	Puesta en práctica de las habilidades y procedimientos de trabajo desarrollados en clase.	Conocimiento y comprensión de conceptos.
Prácticas	35%	X	X	X	X	X
Supuestos prácticos	45%	X	X	X	X	X
Exámenes teóricos	20%			X	X	X

CÁLCULO DE LA CALIFICACIÓN FINAL

La calificación final será calculada según el **método2** indicado en la parte general de la programación del ciclo. Se tendrán en cuenta las siguientes consideraciones:

1. **Las notas de evaluación**, al final de cada trimestre, **serán** fiel reflejo del estado actualizado de **notas acumuladas en el libro de calificaciones** de cada alumn@ en la plataforma Moodle, en la fecha que se tenga que informar.
2. Las calificaciones en **las evaluaciones parciales** (1ª, 2ª) y final serán calculadas a partir de **todas las notas acumuladas desde principio de curso hasta la fecha de la correspondiente evaluación**; serán ponderadas detalladamente en el libro de calificaciones, con el acuerdo del alumnado, y de manera que mantengan coherencia con la tabla de ponderación de contenidos establecida de forma general en esta programación.
3. Se considerarán **aprobados por curso** todos aquellos casos que, al final del último trimestre, obtengan **un promedio final ponderado, en el libro de calificaciones, superior o igual a 5.**
4. Si **no se ha obtenido el aprobado por curso** se **deberá realizar el examen de junio** (primera convocatoria). **Si la nota del examen es igual o mayor de 5 se aprueba la convocatoria.** Para que la calificación numérica efectiva en la convocatoria supere el 5, será porque

así lo haga la **media simple entre la nota del examen y la nota final obtenida por curso.**

5. Si **no se ha obtenido el aprobado** en la convocatoria de junio se deberá realizar el examen **de septiembre** (segunda convocatoria). **Si la nota del examen es igual o mayor de 5 se aprueba la convocatoria.** Para que la calificación numérica efectiva en la convocatoria supere el 5, será necesario que así lo haga la media simple entre **la nota del examen y la nota final obtenida por curso o bien las tareas de recuperación supletorias propuestas.**

EXAMEN DE CONVOCATORIA ORDINARIA

El examen de la convocatoria ordinaria abarcará la totalidad de los contenidos del módulo con especial insistencia en los contenidos y aspectos relacionados con los objetivos de cada RA del curso y con un enfoque aplicado y práctico en forma de pequeño proyecto similar al realizado durante el curso.

Constará de dos partes:

- **Un supuesto de aplicación o proyecto breve** en el que dadas unas especificaciones se deberá completar una solución técnica debidamente justificada. Todo ello realizado en un puesto de trabajo como el utilizado durante el curso en el módulo y durante un tiempo limitado, pero suficiente para un nivel medio de capacidad. La finalidad es demostrar las capacidades y habilidades procedimentales y de actitud mínimas necesarias para las competencias básicas de un programador de aplicaciones informáticas. La **ponderación es de 2/3 de la nota del examen.**
- **Un cuestionario objetivo de tipo test** sobre los aspectos conceptuales, tecnológicos o aplicados de forma simple, que se han puestos en juego durante todo el curso. La **ponderación es de 1/3 de la nota del examen.**

Una vez aprobado el examen de esta convocatoria, al menos con un 5, en la evaluación se tiene en cuenta la actitud, el trabajo en clase durante el curso y los trabajos individuales realizados para la recuperación. Para poder obtener más de 5 en la evaluación, **se deberán aportar y completar íntegramente todas las actividades de carácter individual previstas en el curso, siguiendo los criterios establecidos en el apartado: *Recuperación de Contenidos.***

Puesto que los trabajos y actividades de recuperación puede que no se hayan realizado durante las clases presencialmente por el alumnado implicado, **se establecerán**, si se considera necesario, **cuestionarios** sobre los diversos aspectos teóricos y aplicados de los trabajos realizados **que se responderán de forma oral o escrita;** con el fin de demostrar las capacidades y habilidades mínimas necesarias, y su autoría.

EXAMEN DE CONVOCATORIA EXTRAORDINARIA

El examen de la convocatoria extraordinaria, abarcará la totalidad de los contenidos del módulo con especial insistencia en los contenidos y aspectos relacionados con los objetivos de cada UT y con un enfoque aplicado y práctico en forma de pequeño proyecto similar al realizado durante el curso.

Constará de dos partes:

- **Un supuesto de aplicación o proyecto breve** en el que dadas unas especificaciones se deberá completar una solución técnica debidamente justificada. Todo ello realizado en un puesto de trabajo como el utilizado durante el curso en el módulo y durante un tiempo limitado, pero suficiente para un nivel medio de capacidad. La finalidad es demostrar las capacidades y habilidades procedimentales y de actitud mínimas necesarias para las competencias básicas de un programador de aplicaciones informáticas. La **ponderación es de 2/3 de la nota del examen.**
- **Un cuestionario objetivo de tipo test** sobre los aspectos conceptuales, tecnológicos o aplicados de forma simple, que se han puestos en juego durante todo el curso. La **ponderación es de 1/3 de la nota del examen.**

Una vez aprobado el examen de esta convocatoria, al menos con un 5, en la evaluación se tiene en cuenta la actitud y los trabajos individuales realizados para la recuperación. Para poder obtener más de 5 en la evaluación, **se deberán aportar y completar íntegramente todas las actividades de carácter individual previstas en el curso, siguiendo las consideraciones del apartado: *Recuperación de Contenidos.***

Puesto que los trabajos y actividades de recuperación puede que no se hayan realizado durante las clases presencialmente por el alumnado implicado, **se establecerán**, si se considera necesario, **cuestionarios** sobre los diversos aspectos teóricos y aplicados de los trabajos realizados **que se responderán de forma oral o escrita;** con el fin de demostrar las capacidades y habilidades mínimas necesarias, y su autoría.

RECUPERACIÓN DE CONTENIDOS

La recuperación de contenidos se resolverá para cada realización de aprendizaje (RA) con el fin de llegar al nivel de aprobado o mejorar, en cada una de ellas. Para ello se realizarán, a propuesta de al menos un interesado, durante el tercer trimestre, cuestionarios y exámenes de repaso. Es importante indicar que en este proceso no se excluyen o sustituyen las notas previas; se promedian dentro de la correspondiente RA resultando un mecanismo bastante personalizado y variable en los resultados, porque depende de la historia previa de notas.

Hay que tener en cuenta la siguiente consideración:

- Para los alumnos cuyas faltas de asistencia se consideren justificadas o cuya incorporación al centro se produzca una vez iniciado el curso, **se aplicarán exactamente los mismos criterios que para el resto del alumnado y deberán realizar íntegramente todas las actividades individuales programadas atrasadas a la vez que las corrientes. Las actividades de carácter colectivo y de colaboración -muy importantes en el enfoque didáctico seguido en este módulo- podrán ser compensadas con trabajos alternativos propuestos específicamente.**

ALUMN@S MATRICULAD@S EN 2º CON EL MÓDULO SUSPENSO

Tipo de recuperación

En el curso PROGRAMACIÓN en Moodle, sección: RECUPERACIÓN PENDIENTES estarán planificadas las siguientes actividades:

1. PRACTICAS Y EXAMENES DE AUTOEVALUACIÓN
 - [5.1.1. \(P\) Repaso: Ejercicios sobre elementos de un programa java](#)
 - [5.1.2. \(P\) Repaso: Ejercicios sobre fundamentos de la POO y diseño de programas](#)
 - Cuestionarios de autoevaluación.
 - Exámenes prácticos de autoevaluación.
1. EXAMEN FINAL semejante al de junio y septiembre, con dos partes:
 - Cuestionario de tipo test.
 - Supuesto práctico.

Fechas de entrega de trabajos o realización de exámenes parciales

Las prácticas propuestas tienen carácter de entrenamiento y tendrán configuradas fechas de entrega repartida en los dos trimestres antes del examen final; así como los cuestionarios y exámenes de autoevaluación.

Proceso de Recuperación

- Realización de las dos prácticas de repaso y entrenamiento a lo largo de los dos trimestres.
- Realización de todos los cuestionarios y exámenes prácticos de autoevaluación intercalados y coordinados con las dos prácticas de entrenamiento propuestas.
- Realización del EXAMEN FINAL en la fecha fijada por Jefatura de Estudios.

Criterios de recuperación

1. Si La nota de examen es igual o mayor de **5 se aprueba la convocatoria.**
2. Los aspectos y criterios de evaluación corresponden íntegramente con los establecidos para la convocatoria de septiembre en la programación didáctica en vigor durante la realización del examen.
3. El examen constará de dos partes:
 - Un cuestionario objetivo de tipo test sobre los aspectos conceptuales, tecnológicos o aplicados de forma básica, que se han puestos en juego durante todo el curso. **Determina 1/3 de la nota del examen.**

- Un supuesto de aplicación o proyecto breve en el que dadas unas especificaciones se deberá completar una solución técnica debidamente justificada. Todo ello realizado en un puesto de trabajo como el utilizado durante el curso en el módulo y durante un tiempo limitado, pero suficiente para un nivel medio de capacidad. La finalidad es demostrar las capacidades y habilidades procedimentales y de actitud mínimas necesarias para las competencias básicas de un programador de aplicaciones informáticas. **Determina 2/3 de la nota del examen.**

ATENCIÓN A LA DIVERSIDAD DEL ALUMNADO EN LOS CICLOS FORMATIVOS

Este apartado se detalla en la parte general de la programación del ciclo formativo.

PREVENCIÓN DE RIESGOS LABORALES

Este apartado se detalla en la parte general de la programación del ciclo formativo.

ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES

Este apartado se detalla en la parte general de la programación del ciclo formativo.

USO DE LAS TIC's

Este apartado se detalla en la parte general de la programación del ciclo formativo.

REFERENCIAS LEGALES

CARM - Cuadro horario DAM. (Borrador). *Cuadro horario - Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.* [En línea] <http://www.carm.es/web/servlet/integra.servlets.Blob?ARCHIVO=02.TABLA%20HORAS%20%20aplic%20multiplat%20mod.pdf&TABLA=ARCHIVOS&CAMPOCLAVE=IDARCHIVO&VALORCLAVE=72131&CAMPOIMAGEN=ARCHIVO&IDTIPO=60&RASTRO=null>.

CARM - Curriculum DAM. (Borrador). *Técnico Superior en Desarrollo Aplicaciones Multiplataforma.* [En línea] <http://www.carm.es/web/servlet/integra.servlets.Blob?ARCHIVO=07.Curr.Des.Aplic%20Multiplat%202011%20.pdf&TABLA=ARCHIVOS&CAMPOCLAVE=IDARCHIVO&VALORCLAVE=72129&CAMPOIMAGEN=ARCHIVO&IDTIPO=60&RASTRO=null>.

INCUAL - IFC080_3. Catálogo Nacional de Cualificaciones Profesionales. *IFC080_3 Programación con lenguajes orientados a objetos y bases de datos relacionales.* [En línea] http://www.educacion.gob.es/educa/incual/pdf/Publicacion/IFC080_3OK.pdf.

INCUAL - IFC155_3. Catálogo Nacional de Cualificaciones Profesionales. *IFC155_3 Programación en lenguajes estructurados de aplicaciones de gestión.* [En línea] http://www.educacion.gob.es/educa/incual/pdf/Publicacion/IFC155_3OK.pdf.

INCUAL - IFC303_3. Catálogo Nacional de Cualificaciones Profesionales. *BDC - IFC303_3 Programación de sistemas informáticos.* [En línea] http://www.educacion.gob.es/educa/incual/pdf/BDC/IFC303_3.pdf.

INCUAL - IFC363_3. Catálogo Nacional de Cualificaciones Profesionales. *BDC - IFC363_3 Administración y programación en sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes.* [En línea] http://www.educacion.gob.es/educa/incual/pdf/BDC/IFC363_3.pdf.

IVAC - DCB DAM. (Borrador) Diseño Curricular Base. *Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.* [En línea] http://www.kei-ivac.com/cast/scfp/detalle-producto.jsp?prd_id=616®Doc=1.

IVAC - UD Programación - DAM. (borrador) Unidades didácticas. *Módulo: Programación - Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.* [En línea] http://www.kei-ivac.com/upload/productos/cast/662/t.s._desarrollo_aplicaciones_multiplataforma_mod.3_prog_transv.pdf.

ME - Currículum DAM. Orden EDU/2000/2010, de 13 de julio. *Técnico Superior en Desarrollo Aplicaciones Multiplataforma.* [En línea] <http://www.boe.es/boe/dias/2010/07/26/pdfs/BOE-A-2010-11888.pdf>.

ME - Título DAM. Real Decreto 450/2010 de 16 de abril. *Técnico Superior en Desarrollo Aplicaciones Multiplataforma.* [En línea] <http://www.boe.es/boe/dias/2010/05/20/pdfs/BOE-A-2010-8067.pdf>.

MTAS - Reglamento Servicios Prevención. Real Decreto 39/1997, de 17 de enero. *Nivel básico en prevención de riesgos laborales.* [En línea] <http://www.boe.es/boe/dias/1997/01/31/pdfs/A03031-03045.pdf>.

UNESCO - CINE 2011. UNESCO Institute for Statistics (UIS). *Clasificación Internacional Normalizada de la Educación CINE 2011.* [En línea] http://www.uis.unesco.org/Education/Documents/UNESCO_GC_36C-19_ISCED_SP.pdf.