

**CICLO SUPERIOR
DESARROLLO DE
APLICACIONES
WEB/DESARROLLO DE
APLICACIONES
MULTIPLATAFORMA**

PROGRAMACIÓN DIDÁCTICA ANUAL

Parte específica del módulo:

0487. Entornos de Desarrollo

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

Turno: mañana/tarde

Profesor mañana y tarde: David Munuera Sánchez

Contenido

Sumario

CARACTERÍSTICAS GENERALES DEL CICLO DE DESARROLLO DE APLICACIONES WEB/DESARROLLO DE APLICACIONES MULTIPLATAFORMA.....	4
DESCRIPCIÓN DEL MÓDULO.....	4
UBICACIÓN, OBJETIVOS, CONTENIDOS Y DISTRIBUCIÓN TEMPORAL DEL MÓDULO.....	4
UBICACIÓN, DISTRIBUCIÓN TEMPORAL Y CARACTERÍSTICAS.....	4
OBJETIVOS GENERALES DEL MÓDULO Y RESULTADOS DE APRENDIZAJE ASOCIADOS.....	4
RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN.....	5
OBJETIVOS GENERALES DEL MÓDULO ASOCIADOS A RESULTADOS DE APRENDIZAJE.....	7
COMPETENCIAS PROFESIONALES, PERSONALES Y SOCIALES ASOCIADAS A RESULTADOS DE APRENDIZAJE.....	8
CONTENIDOS Y UNIDADES DE TRABAJO.....	9
PLANIFICACIÓN DE UT.....	9
UNIDADES DE TRABAJO.....	10
METODOLOGÍA.....	18
CRITERIOS.....	18
ASPECTOS CONCRETOS.....	19
AGRUPAMIENTO DEL ALUMNADO.....	20
MATERIALES, RECURSOS, ESPACIO DOCENTE.....	20
MATERIALES Y RECURSOS DIDÁCTICOS.....	20
BIBLIOGRAFÍA.....	20
DISTRIBUCIÓN DEL ESPACIO Y EL TIEMPO DOCENTE.....	21

MEDIDAS PARA ESTIMULAR EL INTERÉS Y EL HÁBITO DE LECTURA Y LA CAPACIDAD DEL ALUMNADO PARA EXPRESARSE CORRECTAMENTE.....	21
CRITERIOS, PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN.....	21
CRITERIOS DE CALIFICACIÓN.....	22
CONSIDERACIONES.....	22
CÁLCULO DE LA CALIFICACIÓN FINAL.....	23
RECUPERACIÓN DE CONTENIDOS.....	23
ALUMNADO MATRICULADO EN 2º CURSO CON EL MÓDULO SUSPENSO.....	24
TIPO DE RECUPERACIÓN.....	24
FECHAS DE ENTREGA DE TRABAJOS O REALIZACIÓN DE EXÁMENES PARCIALES	24
PROCESO DE RECUPERACIÓN.....	24
CRITERIOS DE RECUPERACIÓN.....	24
ATENCIÓN A LA DIVERSIDAD DEL ALUMNADO EN LOS CICLOS FORMATIVOS.....	25
PREVENCIÓN DE RIESGOS LABORALES.....	25
ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES... 	25
USO DE LAS TIC´s.....	25
REFERENCIAS LEGALES.....	25

CARACTERÍSTICAS GENERALES DEL CICLO DE DESARROLLO DE APLICACIONES WEB/DESARROLLO DE APLICACIONES MULTIPLATAFORMA

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

DESCRIPCIÓN DEL MÓDULO

El módulo profesional de *Entornos de Desarrollo* se centra en el estudio de las metodologías, técnicas, procedimientos y herramientas necesarias para el desarrollo de aplicaciones.

El carácter introductorio del módulo a las técnicas de modelado con UML hace que sea importante para desarrollar las necesarias capacidades analíticas y de síntesis necesarias para la programación.

UBICACIÓN, OBJETIVOS, CONTENIDOS Y DISTRIBUCIÓN TEMPORAL DEL MÓDULO

UBICACIÓN, DISTRIBUCIÓN TEMPORAL Y CARACTERÍSTICAS

El módulo profesional de *Entornos de Desarrollo* se encuadra en el primer curso del Ciclo Superior de *Desarrollo de Aplicaciones Web/Desarrollo de Aplicaciones Multiplataforma*, correspondiente al título de *Técnico Superior en Desarrollo de Aplicaciones Web/Técnico Superior en Desarrollo de Aplicaciones Multiplataforma*. Se desarrolla, contando con 90 horas de currículo equivalentes a 6 créditos ECTS, a razón de 3 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, para los dos ciclos enunciados
3. Es básico para los módulos de segundo curso.
4. Se complementa con el módulo de *Programación*, también de primero, con el se requiere que esté coordinado.

OBJETIVOS GENERALES DEL MÓDULO Y RESULTADOS DE APRENDIZAJE ASOCIADOS

En el módulo de *Entornos de Desarrollo* se identifican los siguientes objetivos generales y resultados de aprendizaje asociados para el Ciclo Superior de **Desarrollo de Aplicaciones Web**:

- d)** Ajustar parámetros analizando la configuración para gestionar servidores en distintos ámbitos de implantación.
- e)** Interpretar el diseño lógico, verificando los parámetros establecidos para gestionar bases de datos.

- h)** Generar componentes de acceso a datos, cumpliendo las especificaciones, para integrar contenidos en la lógica de una aplicación web.
- i)** Utilizar lenguajes de marcas y estándares web, asumiendo el manual de estilo, para desarrollar interfaces en aplicaciones web.
- j)** Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
- r)** Analizar y utilizar los recursos y oportunidades de aprendizaje relacionadas con la evolución científica, tecnológica y organizativa del sector y las tecnologías de la información y la comunicación, para mantener el espíritu de actualización y adaptarse a nuevas situaciones laborales y personales.

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

En el módulo de *Entornos de Desarrollo* se identifican los siguientes objetivos generales y resultados de aprendizaje asociados para el Ciclo Superior de **Desarrollo de Aplicaciones Multiplataforma:**

- d)** Instalar y configurar módulos y complementos, evaluando su funcionalidad, para gestionar entornos de desarrollo.
- e)** Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.
- h)** Emplear herramientas de desarrollo, lenguajes y componentes visuales, siguiendo las especificaciones y verificando interactividad y usabilidad, para desarrollar interfaces gráficos de usuario en aplicaciones multiplataforma.
- i)** Seleccionar y emplear técnicas, motores y entornos de desarrollo, evaluando sus posibilidades, para participar en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento.
- j)** Seleccionar y emplear técnicas, lenguajes y entornos de desarrollo, evaluando sus posibilidades, para desarrollar aplicaciones en teléfonos móviles, tabletas y otros dispositivos inteligentes.
- ñ)** Analizar y aplicar técnicas y librerías de programación, evaluando su funcionalidad para desarrollar aplicaciones multiproceso y multihilo.
- o)** Reconocer la estructura de los sistemas ERP-CRM, identificando la utilidad de cada uno de sus módulos, para participar en su implantación.

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 *Entornos de Desarrollo*, establecidos en la legislación vigente (ME - Título DAW/DAM) *anexo 1*, son los siguientes:

RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.

Criterios de evaluación:

- a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador, periféricos, entre otros.
- b) Se han identificado las fases de desarrollo de una aplicación informática.
- c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.
- d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
- e) Se han clasificado los lenguajes de programación, identificando sus características.
- f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en el desarrollo de software.
- g) Se han identificado las características y escenarios de uso de las metodologías ágiles de desarrollo de software.

RA2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.

Criterios de evaluación:

- a) Se han instalado entornos de desarrollo, propietarios y libres.
- b) Se han añadido y eliminado módulos en el entorno de desarrollo.
- c) Se ha personalizado y automatizado el entorno de desarrollo.
- d) Se ha configurado el sistema de actualización del entorno de desarrollo.
- e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
- f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.
- g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.

RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas.

Criterios de evaluación:

- a) Se han identificado los diferentes tipos de pruebas.
- b) Se han definido casos de prueba.
- c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
- d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
- e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
- f) Se han efectuado pruebas unitarias de clases y funciones.
- g) Se han implementado pruebas automáticas.
- h) Se han documentado las incidencias detectadas.
- i) Se han utilizado dobles de prueba para aislar los componentes durante las pruebas.

RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.

Criterios de evaluación:

- a) Se han identificado los patrones de refactorización más usuales.
- b) Se han elaborado las pruebas asociadas a la refactorización.
- c) Se ha revisado el código fuente usando un analizador de código.
- d) Se han identificado las posibilidades de configuración de un analizador de código.
- e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
- f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.
- g) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.
- h) Se han utilizado repositorios remotos para el desarrollo de código colaborativo.
- i) Se han utilizado herramientas para la integración continua del código.

RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

Criterios de evaluación:

- a) Se han identificado los conceptos básicos de la programación orientada a objetos.
- b) Se han utilizado herramientas para la elaboración de diagramas de clases.
- c) Se ha interpretado el significado de diagramas de clases.
- d) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
- e) Se ha generado código a partir de un diagrama de clases.
- f) Se ha generado un diagrama de clases mediante ingeniería inversa.

RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

Criterios de evaluación:

- a) Se han identificado los distintos tipos de diagramas de comportamiento.
- b) Se ha reconocido el significado de los diagramas de casos de uso.
- c) Se han interpretado diagramas de interacción.
- d) Se han elaborado diagramas de interacción sencillos.
- e) Se ha interpretado el significado de diagramas de actividades.
- f) Se han elaborado diagramas de actividades sencillos.
- g) Se han interpretado diagramas de estados.
- h) Se han planteado diagramas de estados sencillos.

OBJETIVOS GENERALES DEL MÓDULO ASOCIADOS A RESULTADOS DE APRENDIZAJE

RESULTADO APRENDIZAJE	OBJETIVO GENERAL
RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.	DAM: d, e, h, i, j, ñ, o DAW: d, e, h, i, j, r

RA2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.	DAM: d, e, h, i, j, ñ, o DAW: d, e, h, i, j, r
RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas.	DAM: d, e, h, i, j, ñ, o DAW: d, e, h, i, j, r
RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.	DAM: d, e, h, i, j, ñ, o DAW: d, e, h, i, j, r
RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.	DAM: d, e, h, i, j, ñ, o DAW: d, e, h, i, j, r
RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.	DAM: d, e, h, i, j, ñ, o DAW: d, e, h, i, j, r

COMPETENCIAS PROFESIONALES, PERSONALES Y SOCIALES ASOCIADAS A RESULTADOS DE APRENDIZAJE

RESULTADO APRENDIZAJE	COMPETENCIA
RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.	j) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación.
RA2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.	j) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación.
RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas.	j) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación. m) Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones. n) Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones. ñ) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su comportamiento y realizando modificaciones.
RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.	j) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación. m) Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones. n) Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones. ñ) Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su

	<i>comportamiento y realizando modificaciones.</i>
<i>RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.</i>	<i>n) Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones.</i>
<i>RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.</i>	<i>i) Integrar componentes multimedia en el interface de una aplicación web, realizando el análisis de interactividad, accesibilidad y usabilidad de la aplicación.</i>

CONTENIDOS Y UNIDADES DE TRABAJO

Los contenidos se organizan en seis U.T. distribuidas en tres bloques que completan **90 horas** durante un curso. Cada U.T. desarrolla los contenidos de tipo conceptual, procedimental, y Actitudinal necesarios para cubrir las realizaciones de aprendizaje, objetivos generales de ciclo asociados en el análisis y los específicamente establecidos para unidad.

PLANIFICACIÓN DE UT

La distribución secuencial de contenidos se realizará a través de la secuenciación de las distintas UT en cada trimestre/evaluación:

PRIMER TRIMESTRE (30 HORAS)

U.T. 0 PRESENTACIÓN 2 HORAS

U.T. 1 INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO (20 HORAS)

U.T. 2 DESARROLLO DE SOFTWARE (8 HORAS)

SEGUNDO TRIMESTRE (30 HORAS)

U.T. 3 DISEÑO Y REALIZACIÓN DE PRUEBAS (15 HORAS)

U.T. 4 OPTIMIZACIÓN Y DOCUMENTACIÓN (15 HORAS)

TERCER TRIMESTRE (30 HORAS)

U.T. 5 ELABORACIÓN DE DIAGRAMAS DE CLASES (15 HORAS)

U.T. 6 ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO (15 HORAS)

UNIDADES DE TRABAJO

UT0. Presentación del módulo (2 horas)

U.T. 1 INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO	20 horas
REALIZACIONES DE APRENDIZAJE	
RA2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.	
OBJETIVOS DIDÁCTICOS	
<i>1. Dadas las necesidades para un proyecto, elegir, las herramientas y entornos necesarios para todas las fases del proyecto, valorando las ventajas e inconvenientes.</i> <i>2. En el entorno del aula, instalar y configurar un entorno de programación; siendo capaz de valorarlo y personalizarlo.</i>	
MÍNIMOS	
<ul style="list-style-type: none">• Añadido y eliminar módulos en el entorno de desarrollo.• Generar ejecutables a partir de código fuente.• Configurar y personalizar básicamente un entorno integrado de programación.	
CONCEPTOS BASE	
<ul style="list-style-type: none">• ENTORNOS INTEGRADOS DE DESARROLLO.• MÓDULOS Y EXTENSIONES.• Herramientas para la automatización del proceso de construcción de ejecutables.	
PROCEDIMIENTOS	ACTITUDES

- Identificación de las características y funciones de los entornos de desarrollo y de las opciones del mercado más importantes (libres y propietarias).
- Instalación y personalización de un entorno de desarrollo con los módulos más habituales.
- Uso de un entorno de desarrollo para la edición y generación de ejecutables.
- Automatización del proceso de construcción de ejecutables.
- Generación de ejecutables mediante código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
- Generación de ejecutables de un mismo código fuente en diversos entornos de desarrollo.
- Configuración de un entorno de desarrollo para el desarrollo de un proyecto de forma colaborativa.

- 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.
- Rigor en la evaluación de las herramientas.

CRITERIOS DE EVALUACIÓN

RA2

- a) Se han instalado entornos de desarrollo, propietarios y libres.*
- b) Se han añadido y eliminado módulos en el entorno de desarrollo.*
- c) Se ha personalizado y automatizado el entorno de desarrollo.*
- d) Se ha configurado el sistema de actualización del entorno de desarrollo.*
- e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.*
- f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.*
- g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.*

Notas

U.T. 2 DESARROLLO DE SOFTWARE

8 horas

REALIZACIONES DE APRENDIZAJE

RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.

OBJETIVOS DIDÁCTICOS

- 1. Dadas las necesidades concretas de información para un trabajo, describir los aspectos fundamentales de la programación de ordenadores, siendo capaz de buscar información y explicar la relación entre el problema planteado y los aspectos descritos.*
- 2. En relación con tareas propuestas, utilizar las herramientas disponibles en el sistema para documentar las actividades; organizando esa documentación con criterios técnicos justificados para su seguimiento.*

MÍNIMOS

- Explicar la relación de los programas con la memoria, el procesador y los periféricos.
- Describir poniendo ejemplos de las fases de desarrollo de una aplicación informática.
- Explicar la diferencia entre código fuente y código máquina.
- Explicar las características del bytecode y cómo se ejecuta en máquinas virtuales.
- Clasificar y describir los distintos lenguajes de programación.
- Instalar, configurar básicamente un entorno integrado de programación.

CONCEPTOS BASE

- | | |
|--|--|
| <ul style="list-style-type: none">•CONCEPTO DE “PROGRAMA INFORMÁTICO”.•INGENIERÍA DEL SOFTWARE.•FASES DEL PROCESO DE DESARROLLO DE SOFTWARE: ANÁLISIS, DISEÑO, CODIFICACIÓN, PRUEBAS, DOCUMENTACIÓN, EXPLOTACIÓN Y MANTENIMIENTO, ENTRE OTRAS.•METODOLOGÍAS DE DESARROLLO.•CÓDIGO FUENTE, CÓDIGO OBJETO Y CÓDIGO EJECUTABLE; MÁQUINAS VIRTUALES.•RELACIÓN DEL SOFTWARE CON LOS COMPONENTES DEL SISTEMA.•ARQUITECTURAS DE SOFTWARE. | <ul style="list-style-type: none">•PATRONES DE DESARROLLO.•REQUERIMIENTOS OPERACIONALES: FIABILIDAD, DISPONIBILIDAD, EXTENSIBILIDAD, ESCALABILIDAD, SEGURIDAD, MANEJABILIDAD, ACCESIBILIDAD.•ROLES QUE INTERACTÚAN EN EL PROCESO DE DESARROLLO DEL SOFTWARE: ANALISTA, PROGRAMADOR, DISEÑADOR, ARQUITECTO, ENTRE OTROS.•DESARROLLO A CAPAS: CLIENTELA, PRESENTACIÓN, NEGOCIO, INTEGRACIÓN, PERSISTENCIA.•Entornos de producción, entornos de desarrollo. |
|--|--|

PROCEDIMIENTOS

- Identificación de las herramientas utilizadas para el desarrollo de software.
- Clasificación de los lenguajes de programación e identificación de las características de los más difundidos.
- Obtención de código ejecutable a través del código fuente: herramientas implicadas. Ejecución en máquinas virtuales.
- Identificación de las principales metodologías de desarrollo de software. Características y ámbitos de aplicación.
- Identificación de la importancia del desarrollo a capas.

ACTITUDES

- Interés por este tipo de lenguajes y su aplicación.
- Valoración de la importancia y las ventajas de la generación metódica de programas.
- Disposición a la planificación de las propias tareas y a la autoevaluación de lo conseguido.
- Interés por la producción (exploración) de soluciones técnicas ante problemas que se presenten y, también, como elemento de mejora del proceso.

CRITERIOS DE EVALUACIÓN

RA1

- a) *Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador, periféricos, entre otros.*
- b) *Se han identificado las fases de desarrollo de una aplicación informática.*
- c) *Se han diferenciado los conceptos de código fuente, objeto y ejecutable.*
- d) *Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.*
- e) *Se han clasificado los lenguajes de programación, identificando sus características.*
- f) *Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en el desarrollo de software.*
- g) *Se han identificado las características y escenarios de uso de las metodologías ágiles de desarrollo de software.*

Notas

U.T. 3 DISEÑO Y REALIZACIÓN DE PRUEBAS

15 horas

REALIZACIONES DE APRENDIZAJE

RA3. Verifica el funcionamiento de programas diseñando y realizando pruebas.

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, preparar casos de prueba sobre el prototipo; comprobando el cumplimiento de requisitos.*
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

- Explicar la diferencia entre los distintos tipos de pruebas.
- Crear casos de prueba para programas sencillos.
- Utilizar básicamente el debugger con un caso de prueba preparado previamente
- Diseñar casos de prueba para métodos.
- Diseñar casos de prueba unitaria para clases.
- Pruebas unitarias
- Pruebas de integración
- Pruebas de regresión

CONCEPTOS BASE

- TIPOS DE PRUEBA: UNITARIAS, FUNCIONALES, INTEGRALES O ESTRUCTURALES, REGRESIÓN, ENTRE OTRAS.
- CASOS DE PRUEBA. ESTRUCTURA DE LOS CASOS DE PRUEBA.
- PRUEBAS DE CÓDIGO: CUBRIMIENTO, VALORES LÍMITE, CLASES EQUIVALENTE, ENTRE OTRAS.
- DEPURACIÓN DEL CÓDIGO.
- HERRAMIENTAS DE DEPURACIÓN.

PROCEDIMIENTOS	ACTITUDES
<ul style="list-style-type: none"> • Creación de casos de prueba. • Identificación de herramientas de depuración ofrecidas por entornos de desarrollo. • Utilización de herramientas de depuración para definir puntos de ruptura y seguimiento. • Utilización de herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución. • Ejecución de pruebas unitarias de clases y funciones. • Implementación de pruebas automáticas. • Documentación de incidencias. • Puntos de ruptura. • Pruebas unitarias: herramientas. • Pruebas automáticas. • Elaboración de documentación técnica del proyecto de software. 	<ul style="list-style-type: none"> • Rigor y responsabilidad en el proceso de depuración y prueba de las aplicaciones y en la elaboración de la documentación. • Reconocimiento del potencial de las nuevas tecnologías como elementos de consulta y apoyo. • Autonomía para desenvolverse en medios que no son los habituales con confianza y adoptando las medidas necesarias. • Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.

CRITERIOS DE EVALUACIÓN

RA3

- a) Se han identificado los diferentes tipos de pruebas.*
- b) Se han definido casos de prueba.*
- c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.*
- d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.*
- e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.*
- f) Se han efectuado pruebas unitarias de clases y funciones.*
- g) Se han implementado pruebas automáticas.*
- h) Se han documentado las incidencias detectadas.*
- i) Se han utilizado dobles de prueba para aislar los componentes durante las pruebas.*

Notas

REALIZACIONES DE APRENDIZAJE

RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.

OBJETIVOS DIDÁCTICOS

- 1. En un proyecto de software, identificar los diferentes aspectos de calidad del software y refactorizar el código para mejorar su diseño; realizando pruebas de equivalencia en el comportamiento del programa.*
- 2. En un proyecto de software, utilizar y aprovechar un sistema de gestión de versiones para apreciar la evolución; sacando conclusiones sobre las mejoras introducidas en el programa.*
- 3. En un proyecto de software, generar documentación automatizada; valorando la eficacia en su actualización.*

MÍNIMOS

- Identificar y describir los patrones típicos de mejora del código.
- Refactorizar programas sencillos mejorando su diseño.
- Aplicar patrones de refactorización utilizando los asistentes del entorno de desarrollo.
- Utilizar un sistema de versiones para apreciar la evolución en la calidad de un programa.
- Generar documentación automatizada de un programa con las herramientas del entorno de desarrollo.

CONCEPTOS BASE

- REFACTORIZACIÓN. CONCEPTO. LIMITACIONES.
- PATRONES DE REFACTORIZACIÓN MÁS HABITUALES.
- HERRAMIENTAS DE AYUDA A LA REFACTORIZACIÓN.
- ANALIZADOR DE CÓDIGO. CALIDAD DEL SOFTWARE
- CONTROL DE VERSIONES. HERRAMIENTAS.
- DOCUMENTACIÓN DEL CÓDIGO. USO DE COMENTARIOS. ALTERNATIVAS.

PROCEDIMIENTOS

ACTITUDES

- Elaboración de pruebas asociadas a la refactorización.
- Revisión del código fuente usando un analizador de código.
- Identificación de las posibilidades de configuración de un analizador de código.
- Aplicación de patrones de refactorización utilizando las herramientas proporcionadas por el entorno de desarrollo.
- Aplicación del control de versiones integrado en el entorno de desarrollo.
- Documentación de las clases mediante el uso de herramientas
- proporcionadas por el entorno de desarrollo.
- Investigación y refinamiento de diseños y prototipos.

- Utilización de la terminología adecuada.
- Integración en la metodología de trabajo diario, en su defecto, proposición de alternativas constructivas, viables y aceptadas por todos.
- Rigor y responsabilidad en el proceso de depuración y prueba de las aplicaciones y en la elaboración de la documentación.
- Actitud ordenada y metódica durante la realización de las tareas y perseverancia ante las dificultades.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.

CRITERIOS DE EVALUACIÓN

RA4

- a) Se han identificado los patrones de refactorización más usuales.*
- b) Se han elaborado las pruebas asociadas a la refactorización.*
- c) Se ha revisado el código fuente usando un analizador de código.*
- d) Se han identificado las posibilidades de configuración de un analizador de código.*
- e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.*
- f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.*
- g) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.*
- h) Se han utilizado repositorios remotos para el desarrollo de código colaborativo.*
- i) Se han utilizado herramientas para la integración continua del código.*

Notas

U.T. 5 ELABORACIÓN DE DIAGRAMAS DE CLASES

15 horas

REALIZACIONES DE APRENDIZAJE

RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas específicas.

OBJETIVOS DIDÁCTICOS

- 1. En un proyecto de software, realizar el modelo de datos utilizando diagramas de clases de UML; identificando y explicando claramente las relaciones existentes entre las entidades.*

MÍNIMOS

- Describir los cuatro principios básicos en los que se fundamenta la Orientación a Objetos.
- Realizar con una herramienta UML un diagrama de clases a partir de la especificación de las clases implicadas.
- Obtener un diagrama de clases a partir del código fuente de las clases.

CONCEPTOS BASE

- | | |
|---|--|
| <ul style="list-style-type: none"> • PROGRAMACIÓN ORIENTADA A OBJETOS. • UML. • CLASES. ATRIBUTOS, MÉTODOS Y VISIBILIDAD. • OBJETOS, INSTANCIACIÓN. • DIAGRAMAS DE CLASE. NOTACIÓN DE LOS DIAGRAMAS DE CLASE. • PROPIEDADES. TIPOS. | <ul style="list-style-type: none"> • OPERACIONES. TIPOS. • INTERFACES. • RELACIONES ENTRE CLASES: HERENCIA, AGREGACIÓN, ASOCIACIÓN, INSTANCIACIÓN. • INGENIERÍA INVERSA. • GENERACIÓN AUTOMÁTICA DE CÓDIGO. |
|---|--|

PROCEDIMIENTOS

- Identificación de los diagramas de clase y su funcionalidad.
- Instalación del módulo del entorno de desarrollo que permite el diagramado de clases.
- Identificación de las funcionalidades y herramientas del módulo de diagramado de clases.
- Generación de diagramas de clase a partir de especificaciones.
- Generación automática de código partiendo de los diagramas de clase.
- Generación de diagramas de clase mediante ingeniería inversa.
- Elaboración de modelos de datos.
- Búsqueda, organización y estudio de la información necesaria para la elaboración de modelos.
- Investigación y refinamiento de diseños. 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.
- Curiosidad por descubrir la evolución en los lenguajes de programación.
- Disposición e iniciativa ante nuevas tareas y para la innovación.
- Autonomía para desenvolverse en medios que no son los habituales con confianza y adoptando las medidas necesarias.
- Valoración positiva de los recursos y medios empleados en las actividades, en su defecto, propuesta justificada de mejora.

CRITERIOS DE EVALUACIÓN

RA5

- a) Se han identificado los conceptos básicos de la programación orientada a objetos.*
- b) Se han utilizado herramientas para la elaboración de diagramas de clases.*
- c) Se ha interpretado el significado de diagramas de clases.*
- d) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.*
- e) Se ha generado código a partir de un diagrama de clases.*
- f) Se ha generado un diagrama de clases mediante ingeniería inversa.*

Notas

U.T. 6 ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO

15 horas

REALIZACIONES DE APRENDIZAJE

RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas específicas.

OBJETIVOS DIDÁCTICOS

1. En un proyecto de software, realizar el modelo de comportamiento utilizando diagramas de casos de uso, diagramas de actividad, diagramas de secuencia y de estado de UML; identificando y explicando claramente los detalles reflejados.

MÍNIMOS

- Comprender la interpretación de los diagramas de casos de uso
- Realizar diagramas de casos de uso dado un enunciado.
- Interpretar y explicar un diagrama de secuencia.
- Interpretar y explicar un diagrama de actividad.
- Realizar diagramas de secuencia a partir del código fuente de un programa.
- Realizar diagramas de actividad a partir del código fuente de un programa.

CONCEPTOS BASE

- UML.
- DIAGRAMAS DE COMPORTAMIENTO. TIPOS Y CAMPO DE APLICACIÓN.
- DIAGRAMAS DE CASOS DE USO. ACTORES, ESCENARIO, RELACIÓN DE COMUNICACIÓN.
- DIAGRAMAS DE ACTIVIDAD. ESTADO DE ACCIÓN, ESTADO DE ACTIVIDAD, TRANSICIÓN, BIFURCACIÓN, DIVISIÓN Y UNIÓN.
- Diagramas de secuencia. Línea de vida de un objeto, activación, envío de mensajes.
- Diagramas de colaboración. Objetos, mensajes.

PROCEDIMIENTOS

- Instalación del módulo del entorno de desarrollo que permite la generación de diagramas de comportamiento.
- Interpretación y elaboración de diagramas de casos de uso.
- Interpretación y elaboración de diagramas de interacción sencillos.
- Interpretación y elaboración de diagramas de actividad sencillos.
- Interpretación y elaboración de diagramas de estados sencillos.
- 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.
- Curiosidad por descubrir la evolución en los lenguajes de programación.
- Disposición e iniciativa ante nuevas tareas y para la innovación.
- Actitud positiva y de confianza en la propia capacidad para progresar.
- Propuesta e investigación nuevas características de POO para los modelos y prototipos diseñados y desarrollados.

CRITERIOS DE EVALUACIÓN

RA6

- a) Se han identificado los distintos tipos de diagramas de comportamiento.
- b) Se ha reconocido el significado de los diagramas de casos de uso.
- c) Se han interpretado diagramas de interacción.
- d) Se han elaborado diagramas de interacción sencillos.
- e) Se ha interpretado el significado de diagramas de actividades.
- f) Se han elaborado diagramas de actividades sencillos.
- g) Se han interpretado diagramas de estados.
- h) Se han planteado diagramas de estados sencillos.

Notas

METODOLOGÍA

CRITERIOS

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

ASPECTOS CONCRETOS

La estrategia metodológica del módulo, tiene en cuenta el enfoque educativo orientado a elementos prácticos y 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 o nuevas capacidades y habilidades en el 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 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 en el grupo habilidades sociales y cognitivas que facilitan el aprendizaje de otros y con otros. Para este fin son esenciales los medios de comunicación que brinda la presencia en el aula y las TIC.
- La elaboración del proyecto promueve la capacidad de 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 aproximaciones -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 **Entornos de Desarrollo** 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 **Entornos de Desarrollo**: <http://moodle.iescierva.net/course/view.php?id=422>

- Aula de Informática cableada en red.
- Servidores de red con recursos didácticos para la docencia y apoyo para las actividades: GitHub, Virtualización y otros.
- Puestos de trabajo con Linux o Windows.
- 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 y otros servicios en Intranet.

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>
 - **[Apuntes sobre requisitos del software](http://moodle.iescierva.net/pluginfile.php/28485/mod_page/content/69/Requisitos%20del%20software.pdf)**
http://moodle.iescierva.net/pluginfile.php/28485/mod_page/content/69/Requisitos%20del%20software.pdf
 - **[Introducción a UML](http://moodle.iescierva.net/pluginfile.php/28485/mod_page/content/69/IntroduccionUML.pdf)**
http://moodle.iescierva.net/pluginfile.php/28485/mod_page/content/69/IntroduccionUML.pdf
 - **[Manual de usuario del programa StarUML](#)**
 - **[Manual de Java. Anexo 6. Entorno de desarrollo Eclipse](https://docs.google.com/document/d/1_mj8yq9RMDxj8fROFcU63UkQcQKupN83KSrR4ysfQGE/edit?usp=sharing)**
https://docs.google.com/document/d/1_mj8yq9RMDxj8fROFcU63UkQcQKupN83KSrR4ysfQGE/edit?usp=sharing
- 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 se podrán establecer específicamente para cada actividad evaluada.

CONSIDERACIONES

- 1.La asistencia es obligatoria y es 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, puede ser 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 se aceptarán 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:

- a. La periodicidad será aproximadamente de dos o tres semanas; coincidiendo con la finalización de las prácticas previstas.
 - b. Estas pruebas nunca serán repetidas aunque pueden ser anuladas o aplazadas con la justificación adecuada y de común acuerdo entre el profesor y l@s alumn@s.
 - c. En pruebas objetivas; l@s alumn@s no presentad@s por causas de fuerza mayor, justificadas documentalmente y admitida su validez/veracidad, dispondrán de una segunda convocatoria adecuadamente coordinada.
 - d. En pruebas objetivas; a l@s alumn@s no presentados por causas injustificadas o por justificación no admitida, se les computará como cero a efectos del cálculo de medias en el libro de calificaciones del curso en Moodle. Queda a la decisión personal del alumn@, o del profesor si lo considera conveniente, el proponer algún trabajo personal alternativo que pueda compensar el cero obtenido y recuperar los aprendizajes previstos.
6. En caso de observarse que la participación y aportación en los trabajos de grupo de algún alumn@ 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, además de tener que demostrar la autoría.

PONDERACIÓN DE CONTENIDOS

Usaremos el Modelo 3 de la programación general con las siguientes concreciones.

N	Apartado	Ponderación
1	Trabajos y proyectos a realizar en clase/casa basados en enunciados prácticos de aplicación amplia de los contenidos y tareas con posible entrevista.	55%
2	Pruebas objetivas teórico-prácticas	45%

CÁLCULO DE LA CALIFICACIÓN FINAL

Usaremos el Modelo 1 usando una ponderación por defecto de 33%

RECUPERACIÓN DE CONTENIDOS

Será resuelta según el **método3** indicado en la parte general de la programación del ciclo. Teniéndose en cuenta las siguientes consideraciones:

- Para el alumnado 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 de carácter colectivo y de colaboración - muy importantes en el enfoque didáctico seguido en este módulo- serán compensadas con trabajos alternativos propuestos específicamente. El/la alumn@ implicad@ deberá elaborar y presentar un plan de trabajo para ponerse al día y realizar todas las actividades y trabajos incluidos en ese plan de recuperación.** -Está detallado cómo debe ser en la sección de *trabajos complementarios* en el curso de Moodle-. El profesor revisará la viabilidad del plan presentado; y sugerirá pautas, reajustes y orientaciones para lograrlo. Como es evidente requiere un esfuerzo suplementario que el/la alumn@ deberá asumir.

ALUMNADO MATRICULADO EN 2º CURSO CON EL MÓDULO SUSPENSO

Tipo de recuperación

En el curso ENTORNOS DE DESARROLLO en Moodle, sección: RECUPERACIÓN PENDIENTES estarán planificadas las siguientes actividades:

1. PRACTICAS Y EXÁMENES DE AUTOEVALUACIÓN
 - Prácticas semejantes a las realizadas en el curso.
2. EXAMEN FINAL semejante a los exámenes de convocatoria, 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.

Proceso de Recuperación

1. Realización de las dos prácticas de repaso y entrenamiento a lo largo de los dos trimestres.
2. Realización de todos los cuestionarios y exámenes prácticos de autoevaluación intercalados y coordinados con las dos prácticas de entrenamiento propuestas.
3. 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 extraordinaria de junio 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 desarrollado 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 Atención a la diversidad de la programación del ciclo formativo DAM y de DAW.

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

Real Decreto 405/2023, de 29 de mayo, por el que se actualizan los títulos de la formación profesional del sistema educativo de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y Técnico Superior en Desarrollo de Aplicaciones Web, de la familia profesional Informática y Comunicaciones, y se fijan sus enseñanzas mínimas.

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 - Cuadro horario DAW. (Borrador). *Cuadro horario - Técnico Superior en Desarrollo de Aplicaciones Web.* [En línea] <http://www.carm.es/web/servlet/integra.servlets.Blob?ARCHIVO=02.TABLA%20HORAS%20%20desar%20aplic%20web%20con%20mod.pdf&TABLA=ARCHIVOS&CAMPOCLAVE=IDARCHIVO&VALORCLAVE=72113&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>.

CARM - Curriculum DAW. (Borrador). *Técnico Superior en Desarrollo Aplicaciones Web.* [En línea] <http://www.carm.es/web/servlet/integra.servlets.Blob?ARCHIVO=06.Curr.Des.%20Aplic%20web%20.pdf&TABLA=ARCHIVOS&CAMPOCLAVE=IDARCHIVO&VALORCLAVE=72128&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 - IFC154_3. Catálogo Nacional de Cualificaciones Profesionales. *IFC154_3 Desarrollo de aplicaciones con tecnologías Web.* [En línea] http://www.educacion.gob.es/educa/incual/pdf/Publicacion/IFC154_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.

—. 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/detalleproducto.jsp?prd_id=616®Doc=1.

IVAC - DCB DAW. (Borrador) Diseño Curricular Base. *Técnico Superior en Desarrollo de Aplicaciones Web.* [En línea] http://www.kei-ivac.com/upload/productos/cast/641/dcb_tecnico_superior_aplicaciones_web.pdf.

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.

IVAC - UD Programación - DAW. (borrador) Unidades didácticas. *Módulo: Programación - Técnico Superior en Desarrollo de Aplicaciones Web.* [En línea] http://www.kei-ivac.com/upload/productos/cast/661/t.s._desarrollo_aplicaciones_web_mod.3_prog.pdf.

ME - Curriculum 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>.

ME - Título DAW. Real Decreto 450/2010 de 16 de abril. *Técnico Superior en Desarrollo Aplicaciones Web.* [En línea] <http://www.boe.es/boe/dias/2010/06/12/pdfs/BOE-A-2010-9269.pdf>.

— Real Decreto 686/2010 de 20 de mayo. *Técnico Superior en Desarrollo Aplicaciones Web.* [En línea] <http://www.boe.es/boe/dias/2010/06/12/pdfs/BOE-A-2010-9269.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.

ANEXO I: APOYO Y DESDOBLES TERCER TRIMESTRE

Este curso debido a la posibilidad de tener apoyo en el 3º trimestre por poder contar con un apoyo de profesores que reducen horario de los segundos cursos, se planifica sesiones de apoyo/desdoble del grupo con otro docente.

Dichos apoyos/desdobles, que serán de 1 o 2 sesiones a la semana, se centrarán principalmente en:

- **U.T. 5 ELABORACIÓN DE DIAGRAMAS DE CLASES**

El modelado supone un nivel de abstracción notable. Se aprovecha en estas clases prácticas para relacionarlo con la generación de código en lenguaje Java y profundizar en los conceptos de POO. Se trabaja en grupos de 2 alumnos y es necesario apoyo para poder tender a más de 12 grupos trabajando aplicaciones de cierta complejidad.

- **U.T. 6 ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO**

Esta UT sería la culminación del modelado intentando mostrar técnicas también de trabajo en equipo y metodologías denominadas Ágiles. El apoyo es pues necesario dado que cada grupo trabajará sobre una aplicación o miniproyecto diferente.

Estos apoyos o desdobles se podrán realizar para todo el alumnado en un mismo aula con dos docentes presenciales o de forma dividida en dos grupos, en caso de las actividades de refuerzo para alumnos con la 2ª evaluación pendiente.