

PROGRAMACIÓN DIDÁCTICA.	
I.E.S. JULIO VERNE	DEPARTAMENTO DE INFORMÁTICA
C.F.G.S. DESARROLLO DE APLICACIONES WEB	1^{ER} CURSO
MÓDULO: <i>ENTORNOS DE DESARROLLO</i>	

CURSO:	2024 – 2025
PROFESOR:	FERNANDO CARBAJO CRUCES

ÍNDICE

INTRODUCCIÓN.	3
2MARCO LEGISLATIVO	3
3REFERENTE CONTEXTUAL	4
4OBJETIVOS GENERALES QUE SE DESARROLLAN EN EL CICLO	4
5OBJETIVOS GENERALES QUE SE DESARROLLAN EN EL MÓDULO	4
6COMPETENCIAS QUE SE DESARROLLAN EN EL MÓDULO.	5
7ADAPTACIÓN AL ENTORNO DE LA ECONOMÍA ANDALUZA.	6
8RESULTADOS DE APRENDIZAJE Y SUS CRITERIOS DE EVALUACIÓN	6
9METODOLOGÍA GENERAL	8
10CONTENIDOS BÁSICOS DEL MÓDULO	9
11CONTENIDOS.	11
12TEMPORIZACIÓN	12
13TABLA DE UNIDAD CON RESULTADO APRENDIZAJE	12
14ESPECIFICACIÓN DE LAS UNIDADES DIDÁCTICAS	14
15CONTENIDOS ACTITUDINALES	17
16PROCESO DE EVALUACIÓN – CALIFICACIÓN.	17
17ADAPTACIONES CURRICULARES.	20
18RECURSOS DIDÁCTICOS.	21
19BIBLIOGRAFÍA RECOMENDADA.	22

1 INTRODUCCIÓN.

El título de formación profesional de **Técnico Superior en Desarrollo de Aplicaciones Multiplataforma** tiene una duración de 2000 horas distribuidas en módulos que se desarrollarán durante dos cursos académicos.

La organización de los módulos de dicho título es la siguiente:

- I) Formación en centro educativo.
 - a) Módulos asociados a la competencia.
 - b) Módulos profesionales socioeconómicos.
 - c) Módulo profesional integrado.
- II) Formación en centro de trabajo.

Atendiendo a esa distribución, el módulo de **Entornos de Desarrollo** se enmarca dentro de "*formación en centro educativo*" y "*asociado a la competencia*".

La duración del mismo es de **96** horas lectivas impartidas durante el primer curso de los dos con los que cuenta el ciclo, repartidas en **3** horas semanales. La totalidad de las horas serán impartidas en el aula que dispone de equipamiento informático para cada alumno.

Este módulo será impartido por un profesor responsable de la asignatura sin apoyo por parte de otro profesor.

2 MARCO LEGISLATIVO

El marco legislativo que regula el ciclo formativo viene determinado por:

- Real Decreto 405/2023, de 29 de mayo, del Ministerio de Educación, 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.

3 REFERENTE CONTEXTUAL

El referente contextual viene determinado por el proyecto educativo del centro

4 OBJETIVOS GENERALES QUE SE DESARROLLAN EN EL CICLO

Recogidos en la programación del departamento

5 OBJETIVOS GENERALES QUE SE DESARROLLAN EN EL MÓDULO

Los objetivos generales, descritos en el real decreto 405/2023, que se desarrollan en el módulo, son los siguientes:

- 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, PDA y otros dispositivos móviles.
- 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.

6 COMPETENCIAS QUE SE DESARROLLAN EN EL MÓDULO.

La formación del módulo contribuye a alcanzar las competencias profesionales, personales y sociales de este título que se relacionan a continuación:

- d) Gestionar entornos de desarrollo adaptando su configuración en cada caso para permitir el desarrollo y despliegue de aplicaciones.
- f) Desarrollar aplicaciones implementando un sistema completo de formularios e informes que permitan gestionar de forma integral la información almacenada.
- h) Desarrollar interfaces gráficas de usuario interactivos y con la usabilidad adecuada, empleando componentes visuales estándar o implementando componentes visuales específicos.
- i) Participar en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento y la educación empleando técnicas, motores y entornos de desarrollo específicos.
- j) Desarrollar aplicaciones para teléfonos, PDA y otros dispositivos móviles empleando técnicas y entornos de desarrollo específicos.
- m). Seleccionar y emplear técnicas y herramientas, evaluando la utilidad de los asistentes de instalación generados, para empaquetar aplicaciones.
- n). Analizar y aplicar técnicas y librerías específicas, simulando diferentes escenarios, para desarrollar aplicaciones capaces de ofrecer servicios en red

7 ADAPTACIÓN AL ENTORNO DE LA ECONOMÍA ANDALUZA.

El presente módulo tiene una vigencia máxima en el entorno andaluz. Puede afirmarse esto debido a que el perfil de técnico especialista capacitado para el procesado de información independientemente del método utilizado para su publicación es muy demandado en la actualidad en Andalucía debido a que el perfil de las empresas es mayoritariamente pequeña y mediana empresa.

8 RESULTADOS DE APRENDIZAJE Y SUS CRITERIOS DE EVALUACIÓN

Resultados de aprendizaje	Criterios de evaluación
1. 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.	<ul style="list-style-type: none"> 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.
2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.	<ul style="list-style-type: none"> 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.
3. Verifica el funcionamiento de programas diseñando y realizando pruebas.	<ul style="list-style-type: none"> 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.

	<p>h). Se han documentado las incidencias detectadas.</p> <p>i). Se han utilizado dobles de prueba para aislar los componentes durante las pruebas.</p>
<p>4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.</p>	<p>a). Se han identificado los patrones de refactorización más usuales.</p> <p>b). Se han elaborado las pruebas asociadas a la refactorización.</p> <p>c). Se ha revisado el código fuente usando un analizador de código.</p> <p>d). Se han identificado las posibilidades de configuración de un analizador de código.</p> <p>e). Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.</p> <p>f). Se ha realizado el control de versiones integrado en el entorno de desarrollo.</p> <p>g). Se han utilizado herramientas del entorno de desarrollo para documentar las clases.</p> <p>h). Se han utilizado repositorios remotos para el desarrollo de código colaborativo.</p> <p>i). Se han utilizado herramientas para la integración continua del código.</p>
<p>5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.</p>	<p>a). Se han identificado los conceptos básicos de la programación orientada a objetos.</p> <p>b). Se han utilizado herramientas para la elaboración de diagramas de clases.</p> <p>c). Se ha interpretado el significado de diagramas de clases.</p> <p>d). Se han trazado diagramas de clases a partir de las especificaciones de las mismas.</p> <p>e). Se ha generado código a partir de un diagrama de clases.</p> <p>f). Se ha generado un diagrama de clases mediante ingeniería inversa.</p>
<p>6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.</p>	<p>a). Se han identificado los distintos tipos de diagramas de comportamiento.</p> <p>b). Se ha reconocido el significado de los diagramas de casos de uso.</p> <p>c). Se han interpretado diagramas de interacción.</p> <p>d). Se han elaborado diagramas de interacción sencillos.</p> <p>e). Se ha interpretado el significado de diagramas de actividades.</p> <p>f). Se han elaborado diagramas de actividades sencillos.</p> <p>g). Se han interpretado diagramas de estados.</p> <p>h). Se han planteado diagramas de estados sencillos</p>

9 METODOLOGÍA GENERAL

La metodología didáctica tiene como finalidad contribuir al logro de la competencia general del ciclo así como la consecución de los resultados del aprendizaje propios del módulo profesional. Existen dos dimensiones de la metodología didáctica: las estrategias didácticas y las actividades. Las estrategias didácticas hacen referencia al tipo de las actividades que se desarrollan en el aula y al modo de organizarlas o secuenciarlas.

Las estrategias didácticas empleadas serán las siguientes:

- Clases expositivas. Tendrán un carácter complementario, limitándose a la exposición y explicación de los conceptos esenciales de cada tema apoyándose en la proyección de esquemas y diagramas que hagan más fácil y ameno el proceso de enseñanza y aprendizaje.
- Realización de trabajos prácticos. Estos trabajos servirán para fijar y aplicar los conocimientos, resolver las dudas que aparezcan y para introducir las técnicas y procedimientos explicados en clase, así como a la utilización de herramientas.

Las actividades hacen referencia a las tareas realizadas por los alumnos/as con la finalidad de adquirir determinados aprendizajes.

Respecto a las actividades de enseñanza-aprendizaje para el alumnado podemos clasificarlas en:

- Resolución de supuestos prácticos: Se desarrollará actividades de las distintas unidades de trabajo, para conseguir así su asimilación de lo expuesto o explicado con anterioridad.
- Investigación: Analizar, buscar información y encontrar las posibles soluciones a un problema.
- Trabajo en grupo: resolver supuestos y compartir la información. Estos grupos serán preferentemente mixtos y en su construcción participarán

ellos mismos. De esta manera se potencia el esfuerzo del alumno/a por hacerse entender y entender a los demás, en definitiva se favorecen los procesos de socialización y crítica.

Actualmente existe un consenso en torno a una concepción constructivista del aprendizaje escolar, que parte de que toda persona posee una serie de conocimientos sobre el mundo que le rodea, conocimientos que pueden ser erróneos o parciales pero útiles para actuar e intervenir en la realidad. Por lo tanto adquirir nuevos conocimientos implica sustituir o modificar conocimientos. Desde esta perspectiva el aprendizaje significativo supone una relación sustantiva entre la nueva información y la disponible en la mente del sujeto dando lugar a una revisión, modificación o enriquecimiento de los esquemas previos.

En el proceso de enseñanza-aprendizaje pretendemos fomentar la iniciativa del alumno/a, sí como el desarrollo de las capacidades de comprensión, análisis, relación, búsqueda y manejo de información. Se motivará además al alumnado para que el propio alumnado intuya que sería útil la existencia de esa herramienta o concepto para mejorar el modo en que se podría resolver el problema planteado. De este modo el propio alumnado está de alguna manera anticipando lo que va a venir a continuación.

1. Se introduce ese nuevo concepto o herramienta en los ejemplos que se están desarrollando para facilitar la resolución de un problema, provocando que sea el propio alumnado quien lo pida o necesite.
2. Se procura que el alumnado experimente con esas nuevas herramientas o conceptos de manera intuitiva e incompleta para resolver nuevos problemas más complejos y así descubra la necesidad de una mayor sistematización y documentación en la explicación de esas nuevas herramientas y conceptos. De este modo el alumnado se encontrará motivado y preparado para asimilar y aplicar esos nuevos conocimientos sin que le parezcan ajenos, extraños o innecesarios.
3. Se lleva a cabo una exposición más sistemática y demostrativa de esos conceptos o herramientas para sistematizar, completar y documentar lo que

el alumnado ya ha aprendido a utilizar de manera práctica e intuitiva.

4. Se plantean diversos problemas y ejercicios donde el alumnado podrá poner en práctica todo lo aprendido tanto de manera intuitiva como sistemática.

Todo el proceso de enseñanza-aprendizaje estará guiado por la realización práctica de los contenidos de cada unidad, tras una exposición teórica. Para afianzar los conceptos, el alumnado tendrá que realizar trabajos y actividades, algunos de los cuales serán de carácter obligatorio, y otros de carácter opcional. Se procurará que existan prácticas individuales, pero también grupales, ya que tan importante es alcanzar los resultados de aprendizaje del módulo, como lo es alcanzar las soft skills (competencias personales y sociales). Se dispone de un ordenador personal para cada alumno o alumna. El alumnado es responsable de salvaguardar las actividades y trabajos que va realizando a lo largo de todo el curso. El profesorado solicitará al alumnado las actividades y trabajos realizados para su posterior evaluación. El alumnado deberá utilizar algún soporte de almacenamiento propio (memoria USB, conexión con la nube o similar) para salvaguardar el trabajo de cara a continuar con él en sesiones posteriores o para la entrega al profesorado. De esta forma se evita que el alumnado pierda el trabajo si el equipo en el que trabaja excepcionalmente se estropee o si otro alumno u otra alumna lo borra accidentalmente. Se fomentará la realización de trabajos de investigación en los que habrá que contrastar informaciones de diferentes fuentes (material proporcionado en clase, documentación técnica, y búsquedas por Internet). También se fomentará el intercambio de información entre los miembros del grupo. Tanto para la realización de actividades como para la realización de trabajos de investigación, se dejará tiempo suficiente en clase para su realización. Se hará un seguimiento en clase del trabajo del alumnado, con el fin de evaluar apropiadamente la adquisición de los resultados de aprendizaje. Así mismo, se fomentará que el alumnado haga exposiciones de uno o varios de los trabajos que haya realizado con el mismo fin anterior. Se incluirán exposiciones orales para mejorar la destreza comunicativa. Los trabajos o actividades en grupo se fundamentarán en los principios del aprendizaje cooperativo, buscando la implicación en el desarrollo de la actividad

práctica o trabajo, sin perjuicio de la distribución de roles entre los participantes. A modo de síntesis y sin perjuicio del necesario rigor conceptual, se tendrá siempre presente la consideración de que lo importante es desarrollar las capacidades para abordar realizaciones prácticas similares a aquellas que se va a tener que afrontar en la vida profesional, una vez concluida la etapa formativa.

10 CONTENIDOS BÁSICOS DEL MÓDULO

1.- Desarrollo de software:

- Concepto de programa informático.
- Código fuente, código objeto y código ejecutable; tecnologías de virtualización.
- Tipos de lenguajes de programación. Paradigmas.
- Características de los lenguajes más difundidos.
- Fases del desarrollo de una aplicación: análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras.
- Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.
- Metodologías ágiles. Técnicas. Características.

2.- Instalación y uso de entornos de desarrollo:

- Funciones de un entorno de desarrollo.
- Instalación de un entorno de desarrollo.
- Uso básico de un entorno de desarrollo.
- Personalización del entorno de desarrollo: temas, estilos de codificación, módulos y extensiones, entre otras.
- Edición de programas.
- Generación de ejecutables en distintos entornos.
- Herramientas y automatización

3.- Diseño y realización de pruebas:

- Planificación de Pruebas.
- Tipos de pruebas: Funcionales, estructurales y regresión, entre otras.
- Procedimientos y casos de prueba.
- Pruebas de Código: Cubrimiento, valores límite y clases de equivalencia, entre otras.
- Pruebas unitarias; herramientas de automatización.
- Documentación de las incidencias.
- Dobles de prueba. Tipos. Características

4.- Optimización y documentación:

- Refactorización. Concepto. Limitaciones. Patrones de refactorización más usuales. refactorización y pruebas. Herramientas de ayuda a la refactorización.
- Analizadores de código.
- Control de versiones. Estructura de las herramientas de control de versiones.

5.- Repositorio. Herramientas de control de versiones. Uso integrado en el entorno de desarrollo:

- Repositorios remotos.
- Documentación. Uso de comentarios. Alternativas.
- Integración continua. Herramientas

6.- Elaboración de diagramas de clases:

- Clases. Atributos, métodos y visibilidad.
- Objetos. Instanciación.
- Relaciones. Asociación, navegabilidad y multiplicidad. Herencia, composición, agregación. Realización y dependencia.
- Notación de los diagramas de clases.
- Herramientas.
- Generación automática de código. Ingeniería inversa.

7.- Elaboración de diagramas de comportamiento:

- Tipos. Campo de aplicación.
- Diagrama de casos de uso. Actores, escenario, relación de comunicación.
- Diagrama de secuencia. Línea de vida de un objeto, activación, envío de mensajes.
- Diagrama de comunicación. Objetos, mensajes.
- Diagrama de actividad. Diagrama de estados.

11 CONTENIDOS.

Las unidades didácticas en las que se distribuye la asignatura son las siguientes:

UD01: Desarrollo de software.

UD02: Instalación y uso de entornos de desarrollo.

UD03: Diseño y realización de pruebas.

UD04: Optimización y documentación.

UD05: Elaboración de diagramas de clases.

UD06: Elaboración de diagramas de comportamiento.

Tanto la temporización como la división en unidades didácticas, ha sido realizado atendiendo a distintos factores, como son la experiencia obtenida de años anteriores, la carga teórica y práctica de las distintas unidades, así como las capacidades profesionales que se desarrollan en cada una de las unidades. Además de las capacidades profesionales que necesitan los alumnos para afrontar con éxito el segundo curso del ciclo.

12 TEMPORIZACIÓN

UD01: Desarrollo de software.	10 h.
UD02: Entornos integrados de desarrollo.	20 h.
UD03: Diseño y realización de pruebas.	20 h.
UD04: Optimización y documentación.	18 h.
UD05: Elaboración de diagramas de clases.	14 h.
UD06: Elaboración de diagramas de comportamiento.	14 h.
TOTAL	96 h

13 TABLA DE UNIDAD CON RESULTADO APRENDIZAJE

UNIDAD	RESULTADOS APRENDIZAJE	CRITERIOS DE EVALUACIÓN
UD01: Desarrollo de software.	1	1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g
UD02: Entornos integrados de desarrollo.	2	2.a, 2.b, 2.c, 2.d, 2.e, 2.f, 2.g
UD03: Diseño y realización de pruebas	3	3.a, 3.b, 3.c, 3.d, 3.e, 3.f, 3.g, 3.h, 3.i
UD04: Optimización y documentación.	4	4.a, 4.b, 4.c, 4.d, 4.e, 4.f, 4.g, 4.h, 4.i
UD05: Elaboración de diagramas de clases.	5	5.a, 5.b, 5.c, 5.d, 5.e, 5.f
UD06: Elaboración de diagramas de comportamiento	6	6.a, 6.b, 6.c, 6.d, 6.e, 6.f, 6.g, 6.h

14 SECUENCIACIÓN

La secuenciación se ha cambiado sobre el orden establecido por defecto, atendiendo a buscar una mejor coordinación con otras materias del curso. La secuenciación queda por tanto del siguiente modo:

UD01: Desarrollo de software.	1 ev
UD02: Entornos integrados de desarrollo.	1 ev
UD05: Elaboración de diagramas de clases.	2 ev
UD06: Elaboración de diagramas de comportamiento.	2 ev
UD04: Optimización y documentación.	3 ev
UD03: Diseño y realización de pruebas.	3 ev

15 ESPECIFICACIÓN DE LAS UNIDADES DIDÁCTICAS**UNIDAD DIDÁCTICA 1. Desarrollo de software.**

Descripción: En esta unidad se dan a conocer los conceptos básicos sobre desarrollo de software. Se describen los tipos de software según su uso, los lenguajes de programación y su clasificación, así como el proceso de obtención de aplicaciones a partir de código fuente. La unidad finaliza con la introducción de metodologías de desarrollo.

Criterios de evaluación: 1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g

N.º sesiones: 10

Contenidos	Metodología
<ol style="list-style-type: none"> 1. Concepto de programa informático. 2. Código fuente, código objeto y código ejecutable; tecnologías de virtualización. 3. Tipos de lenguajes de programación. Paradigmas. 4. Características de los lenguajes más difundidos. 5. Fases del desarrollo de una aplicación: análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras. 6. Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas. 7. Metodologías ágiles. Técnicas. Características 	<ul style="list-style-type: none"> ● Exposiciones teóricas. ● Búsquedas en internet de información relativa al tema. ● Realización de ejercicios de resumen y síntesis.
Observaciones	

UNIDAD DIDÁCTICA 2. Entornos integrados de desarrollo.

Descripción: En esta unidad didáctica el alumnado tendrá su primer contacto con los entornos integrados de desarrollo (IDE), revisando sus funciones y componentes. Se evaluarán distintos IDEs, tanto libres como comerciales.

Criterios de evaluación: 2.a, 2.b, 2.c, 2.d, 2.e, 2.f, 2.g

N.º sesiones: 20

Contenidos	Metodología
<ol style="list-style-type: none"> 1. Funciones de un entorno de desarrollo. 2. Instalación de un entorno de desarrollo. 	<ul style="list-style-type: none"> ● Exposiciones teóricas. ● Ejercicios y supuestos

<ol style="list-style-type: none"> 3. Uso básico de un entorno de desarrollo. 4. Personalización del entorno de desarrollo: temas, estilos de codificación, módulos y extensiones, entre otras. 5. Edición de programas. 6. Generación de ejecutables en distintos entornos. 7. Herramientas y automatización 	<p>prácticos.</p> <ul style="list-style-type: none"> ● Prácticas de instalación, configuración y uso: Netbeans, Visual Code e IntelliJ
Observaciones:	

UNIDAD DIDÁCTICA 3: Diseño y realización de pruebas	
<p>Descripción: En esta unidad se introducirán las técnicas de validación de software necesarias en el desarrollo de aplicaciones. El alumnado aprenderá a depurar código, validar la calidad técnica de su código fuente y a realizar pruebas tanto unitarias como funcionales.</p>	
<p>Criterios de evaluación: 3.a, 3.b, 3.c, 3.d, 3.e, 3.f, 3.g, 3.h, 3.i</p>	<p>N.º sesiones: 20</p>
Contenidos	Metodología
<ol style="list-style-type: none"> 1. Planificación de Pruebas. 2. Tipos de pruebas: Funcionales, estructurales y regresión, entre otras. 3. Procedimientos y casos de prueba. 4. Pruebas de Código: Cubrimiento, valores límite y clases de equivalencia, entre otras. 5. Pruebas unitarias; herramientas de automatización. 6. Documentación de las incidencias. 7. Dobles de prueba. Tipos. Características. 	<ul style="list-style-type: none"> ● Exposiciones teóricas. ● Ejercicios y supuestos prácticos. ● Prácticas de depuración de código con NetBeans . ● Prácticas de validación estática de código con PMP, Checkstyle y FindBug. ● Prácticas de realización de pruebas unitarias con JUnit.
Observaciones	

UNIDAD DIDÁCTICA 4: Optimización y documentación	
Descripción: En esta unidad el alumnado aprenderá patrones y técnicas de optimización de código fuente. Además, se revisarán herramientas para garantizar el control de versiones y la generación de documentación de una aplicación software.	
Criterios de evaluación: 4.a, 4.b, 4.c, 4.d, 4.e, 4.f, 4.g, 4.h, 4.i	N.º sesiones: 18
Contenidos	Metodología
1. Refactorización. Concepto. Limitaciones. Patrones de refactorización más usuales. Refactorización y pruebas. Herramientas de ayuda a la refactorización. 2. Analizadores de código. 3. Control de versiones. Estructura de las herramientas de control de versiones. 4. Repositorio. Herramientas de control de versiones. Uso integrado en el entorno de desarrollo. 5. Repositorios remotos. 6. Documentación. Uso de comentarios. Alternativas. 7. Integración continua. Herramientas.	<ul style="list-style-type: none"> ● Exposiciones teóricas. ● Ejercicios y supuestos prácticos. ● Prácticas de refactorización de código fuente. ● Prácticas de aplicación de patrones de diseño. ● Prácticas de documentación con JavaDoc. ● Prácticas con GitHub, Docker...
Observaciones:	

UNIDAD DIDÁCTICA 5: Elaboración de diagramas de clases	
Descripción: En esta unidad se introducirá el lenguaje de modelo UML, y en concreto los diagramas de clases para la representación estática de nuestra aplicación.	
Criterios de evaluación: 5.a, 5.b, 5.c, 5.d, 5.e, 5.f	N.º sesiones: 14
Contenidos	Metodología
1. Clases. Atributos, métodos y visibilidad. 2. Objetos. Instanciación. 3. Relaciones. Asociación, navegabilidad y multiplicidad. Herencia,	<ul style="list-style-type: none"> ● Exposiciones teóricas. ● Ejercicios y supuestos prácticos. ● Prácticas de creación de

4. composición, agregación. Realización y dependencia. 5. Notación de los diagramas de clases. 6. Herramientas. 7. Generación automática de código. Ingeniería inversa	modelos de clases.
Observaciones	

UNIDAD DIDÁCTICA 6: Elaboración de diagramas de comportamiento	
Descripción: En esta unidad el alumnado continuará trabajando con el lenguaje de modelado UML, concretamente con los distintos tipos de diagramas de comportamiento que servirán para modelar dinámicamente un sistema software.	
Criterios de evaluación: 6.a, 6.b, 6.c, 6.d, 6.e, 6.f, 6.g, 6.h	N.º sesiones: 14
Contenidos	Metodología
1. Tipos. Campo de aplicación. 2. Diagrama de casos de uso. Actores, escenario, relación de comunicación. 3. Diagrama de secuencia. Línea de vida de un objeto, activación, envío de mensajes. 4. Diagrama de comunicación. Objetos, mensajes. 5. Diagrama de actividad. Diagrama de estados.	<ul style="list-style-type: none"> ● Exposiciones teóricas. ● Ejercicios y supuestos prácticos. ● Prácticas de creación de modelos de comportamiento con Eclipse Papyrus.
Observaciones:	

16 CONTENIDOS ACTITUDINALES

Recogidos en la programación del departamento.

17 PROCESO DE EVALUACIÓN – CALIFICACIÓN.

Se usarán los siguientes instrumentos de evaluación:

- Práctica: Trabajo individual práctico, en el que el alumnado desarrolla sobre los ordenadores los contenidos explicados en clase, con acceso a cualquier tipo de fuente de documentación. Estos ejercicios se realizarán dentro del aula y tendrán una calificación individual asociada al resultado de aprendizaje correspondiente.
- Cuestionario: Prueba teórica de carácter individual. Durante la prueba el alumnado no podrá consultar ninguna fuente de documentación, ni podrá contar con la ayuda de ninguna persona.
- Examen: Prueba teórico-práctica de carácter individual. Durante la prueba el alumnado no podrá consultar ninguna fuente de documentación, ni podrá contar con la ayuda de ninguna persona.
- Trabajo de aula: Diálogo y observación diaria sobre los procedimientos.

El proceso de evaluación se realiza por resultado de aprendizaje. Debiendo ser superado cada resultado de aprendizaje por separado, y teniendo en cuenta que los contenidos asociados a cada uno de ellos no tienen relación con el resto, por lo que la superación de un resultado de aprendizaje concreto no implica la superación de otros.

La relación entre los resultados de aprendizaje y las unidades didácticas es la siguiente:

Resultado de aprendizaje	Unidades didácticas
1	1
2	2
3	3
4	4
5	5
6	6

Siendo la separación de resultados de aprendizaje por trimestre la siguiente:

Evaluación	Resultado de aprendizaje	Unidad didáctica
1ª	1, 6, 5	1, 6, 5
2ª	2, 4	2, 4
3ª	3	3

Criterios de calificación generales:

- Un resultado de aprendizaje estará aprobado o superado si su nota es igual o superior a cinco (5).
- Los resultados de aprendizaje aprobados se guardan hasta el final del curso.
- **La nota de la primera evaluación será:**
 - Si se han superado los resultados de aprendizaje 1, 5, 6, la nota de la 1ª evaluación será la media aritmética de las notas de dichos resultados de aprendizaje.
 - Si no se ha superado alguno de los resultados de aprendizaje (1, 5 o 6), la nota de la 1ª evaluación será la media aritmética de las notas de dichos RRAA, siempre que esta sea igual o inferior a cuatro (4). En otro caso, la nota de la 1ª evaluación será igual a cuatro (4).
- **La nota de la segunda evaluación será:**
 - Si se han superado los resultados de aprendizaje 2 y 4, la nota de la 2ª evaluación será la media aritmética de las notas de dichos resultados de aprendizaje.
 - Si no se ha superado alguno de los resultados de aprendizaje (2 o 4), la nota de la 2ª evaluación será la media aritmética de las notas de dichos módulos, siempre que esta sea igual o inferior a cuatro (4). En otro caso, la nota de la 2ª evaluación será igual a cuatro (4).
- **La nota de la tercera evaluación será:**
 - Si se han superado los resultados de aprendizaje 3, la nota de la 3ª evaluación será la media aritmética de las notas de dichos resultados de aprendizaje.
 - Si no se ha superado alguno de los resultados de aprendizaje (3), la nota de la 3ª evaluación será la media aritmética de las notas de dichos módulos, siempre que esta sea igual o inferior a cuatro (4). En otro caso, la nota de la 3ª evaluación será igual a cuatro (4).
- **La nota final del curso** vendrá determinada por la nota de todos los resultados de aprendizaje:

- Si se han superado todos los resultados de aprendizaje, la nota final será la media aritmética de las notas de los resultados de aprendizaje.
- Si no se han superado alguno de los resultados de aprendizaje, la nota final será la media aritmética de las notas de los módulos, siempre que esta sea igual o inferior a cuatro (4). En otro caso, la nota final será igual a cuatro (4).
- Opciones para aprobar cada resultado de aprendizaje:
 - 1ª oportunidad: cuando se realice el examen asociado al resultado de aprendizaje, y se hayan realizado todas las prácticas correspondientes.
 - 2ª oportunidad: cuando se realice el siguiente examen marcado en el calendario. Será necesario realizar la entrega de las prácticas correspondientes al resultado de aprendizaje que se desea recuperar en caso de no haberlo hecho previamente.
 - Recuperación final: en el mes de junio se establecerán las pruebas necesarias para que el alumnado pueda recuperar todos los resultados de aprendizaje.
- El alumnado que desee subir nota podrá hacer uso de los momentos de recuperación, 2ª oportunidad y recuperación final. En cualquier caso se les guardarán las notas aprobadas y no podrán bajar nota.

18 ADAPTACIONES CURRICULARES.

Recogidas en la programación del departamento.

Casos Particulares:

- Alumna con problemas de audición. Cuenta con apoyo de profesional de interprete de signos. Se adoptarán medidas de flexibilización temporal y tutoría entre iguales.
- Alumno con lesiones de origen cerebral. Cuenta con un monitor de apoyo. Se adoptarán medidas de flexibilización temporal y tutoría entre iguales.
- Alumno con sobredotación intelectual. Se le hará una adaptación curricular

dirigida a alumnos con altas capacidades.

- Alumno con talento complejo. Se le planteará una metodología de aprendizaje por proyectos y se tomarán medidas de flexibilización temporal y espacial.

19 RECURSOS DIDÁCTICOS.

En el caso de este módulo los recursos los dividimos en dos tipos: Humanos y Materiales.

a) Recursos Humanos: El módulo cuenta exclusivamente con un profesor titular

b) Recursos Materiales: Se pueden inventariar los siguientes:

- Un aula taller, donde se ubican todas las clases del grupo.
- Un proyector de vídeo/SVGA.
- 20 ordenadores de sobremesa para el alumnado.
- 1 ordenador de sobremesa para el profesor.
- Una impresora láser.
- Una LAN que integra todos los ordenadores del aula.
- Red WiFi de soporte conectada a la red del centro.
- Plataforma Moodle de apoyo.
- Ubuntu 18.04 y Microsoft Windows 10.
- Software Colab, NetBeans, IntelliJ, Visual Code, GIT, Git-Hub, Junit, Selenium, UML para Netbeans, Trello

20 BIBLIOGRAFÍA RECOMENDADA.

- Carlos Casado Iglesias
Entornos de Desarrollo
Ed. Ra-Ma
ISBN: 978-84-9964-169-0

- Alicia Ramos Martín, M^a Jesús Ramos Martín
Entornos de Desarrollo
Ed. Garceta
ISBN: 978-84-1545-297-3

- Craig Larman.
UML y Patrones
Ed. Prentice Hall
ISBN: 978-84-2053-438-1