

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

<b>CURSO:</b>	<b>2023 – 2024</b>
<b>PROFESOR:</b>	<b>ENRIQUE MATAS CARVAJAL</b>

## ÍNDICE

1	INTRODUCCIÓN.....	3
2	MARCO LEGISLATIVO.....	3
3	REFERENTE CONTEXTUAL.....	4
4	OBJETIVOS GENERALES QUE SE DESARROLLAN EN EL CICLO.....	4
5	OBJETIVOS GENERALES QUE SE DESARROLLAN EN EL MÓDULO.....	4
6	COMPETENCIAS QUE SE DESARROLLAN EN EL MÓDULO.....	5
7	ADAPTACIÓN AL ENTORNO DE LA ECONOMÍA ANDALUZA.....	6
8	RESULTADOS DE APRENDIZAJE Y SUS CRITERIOS DE EVALUACIÓN.....	6
9	METODOLOGÍA GENERAL.....	8
10	CONTENIDOS BÁSICOS DEL MÓDULO.....	9
11	CONTENIDOS.....	11
12	TEMPORIZACIÓN.....	12
13	TABLA DE UNIDAD CON RESULTADO APRENDIZAJE.....	12
14	ESPECIFICACIÓN DE LAS UNIDADES DIDÁCTICAS.....	14
15	CONTENIDOS ACTITUDINALES.....	17
16	PROCESO DE EVALUACIÓN – CALIFICACIÓN.....	17
17	ADAPTACIONES CURRICULARES.....	20
18	RECURSOS DIDÁCTICOS.....	21
19	BIBLIOGRAFÍA 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 450/2010, de 16 de abril, del Ministerio de Educación, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se

fijan sus enseñanzas mínimas.

- Orden de 16 de junio de 2011, de la Consejería de Educación de la Junta de Andalucía, por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web

### 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

El presente título, según el real decreto 450/2010 y la orden de 16 de junio de 2011, tiene como **objetivos generales** 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.

r) Verificar los componentes software desarrollados, analizando las especificaciones, para completar un plan de pruebas.

## **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.

r) Realizar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.

t) Establecer vías eficaces de relación profesional y comunicación con sus superiores, compañeros y subordinados, respetando la autonomía y competencias de las distintas personas.

v) Gestionar su carrera profesional, analizando las oportunidades de empleo, autoempleo y de aprendizaje.

w) Mantener el espíritu de innovación y actualización en el ámbito de su

trabajo para adaptarse a los cambios tecnológicos y organizativos de su entorno profesional.

## 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"> <li>a) Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros.</li> <li>b) Se han clasificado los lenguajes de programación.</li> <li>c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.</li> <li>d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.</li> <li>e) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.</li> <li>f) Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.</li> <li>g) Se han identificado las fases de desarrollo de una aplicación informática.</li> </ul>
2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutables.	<ul style="list-style-type: none"> <li>a) Se han instalado entornos de desarrollo, propietarios y libres.</li> <li>b) Se han añadido y eliminado módulos en el entorno de desarrollo.</li> <li>c) Se ha personalizado y automatizado el entorno de desarrollo.</li> <li>d) Se ha configurado el sistema de actualización del entorno de desarrollo.</li> <li>e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.</li> <li>f) Se han generado ejecutables a partir de un</li> </ul>

	<p>mismo código fuente con varios entornos de desarrollo.</p> <p>g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.</p> <p>h) Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.</p>
<p>3. Verifica el funcionamiento de programas diseñando y realizando pruebas.</p>	<p>a) Se han identificado los diferentes tipos de pruebas.</p> <p>b) Se han definido casos de prueba.</p> <p>c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.</p> <p>d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.</p> <p>e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.</p> <p>f) Se ha documentado el plan de pruebas.</p> <p>g) Se han efectuado pruebas unitarias de clases y funciones.</p> <p>h) Se han efectuado pruebas de integración, de sistema y de aceptación.</p> <p>i) Se han implementado pruebas automáticas.</p> <p>j) Se han documentado las incidencias detectadas.</p> <p>k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software.</p> <p>l) Se han realizado medidas de calidad sobre el software desarrollado.</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 ha documentado el código fuente mediante comentarios.</p> <p>h) Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos.</p> <p>i) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.</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 ha instalado el módulo del entorno integrado de desarrollo que permite la</p>

	<p>utilización de diagramas de clases.</p> <p>c) Se han identificado las herramientas para la elaboración de diagramas de clases.</p> <p>d) Se ha interpretado el significado de diagramas de clases.</p> <p>e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.</p> <p>f) Se ha generado código a partir de un diagrama de clases.</p> <p>g) 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 han interpretado diagramas de estados.</p> <p>f) Se han planteado diagramas de estados sencillos.</p> <p>g) Se ha interpretado el significado de diagramas de actividades.</p> <p>h) Se han elaborado diagramas de actividades sencillos.</p>

## 9 METODOLOGÍA GENERAL

Recogida en la programación del departamento y adaptada a la circular de 3 de septiembre de 2020, de la Viceconsejería de Educación y Deporte, relativa a medidas de flexibilización curricular y organizativas para el curso escolar 2020/2021. En base a la circular mencionada la docencia se imparte en el modelo de semipresencialidad, en el que para mantener la distancia de seguridad establecida por las autoridades sanitarias cada grupo se dividirá en dos turnos, de manera que cada turno tendrá 3 horas de clase presenciales diarias. Ocupándose las 3 horas restantes de su horario mediante tareas o actividades que el alumnado realizará en casa.

Para el seguimiento del trabajo a distancias se utilizará la plataforma Moodle propia del centro, <https://aula.iesjulioverne.es/>, y las herramientas propias de la suite Google Suite For Education (<https://edu.google.com/>) En ambos casos el alumnado accederá mediante un cuenta de correo creada desde el centro.

## 10 CONTENIDOS BÁSICOS DEL MÓDULO

### 1.- Reconocimiento de elementos del desarrollo de software:

- Conceptos de programa informático y de aplicación informática.
- Concepto de lenguaje de programación.
- Tipos de lenguajes de programación.
- Características de los lenguajes más difundidos.
- Código fuente, código objeto y código ejecutable; máquinas virtuales.
- Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.
  - Traductores de lenguajes.
  - Depuradores.
- Fases del desarrollo de una aplicación, análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras.

### 2.- Evaluación de entornos integrados de desarrollo:

- Funciones de un entorno de desarrollo.
- Componentes de un entorno de desarrollo.
- Instalación de un entorno de desarrollo.
- Mecanismo de actualización de un entorno de desarrollo.
- Uso básico de un entorno de desarrollo.
  - Edición de programas.
  - Generación de ejecutables.
- Entornos de desarrollo libres y comerciales más usuales.
- Uso de herramientas CASE en el desarrollo de software.

### 3.- Diseño y realización de pruebas:

- Tipos de pruebas, funcionales, estructurales, regresión.
- Procedimientos y casos de prueba.
- Pruebas de Código, cubrimiento, valores límite, clases de equivalencia.
- Herramientas de depuración de código.
- Planificación de Pruebas.

- Pruebas unitarias; herramientas.
- Pruebas de integración.
- Pruebas del sistema.
- Pruebas de aceptación.
- Automatización de pruebas.
- Calidad del software.
  - Normas y certificaciones.
  - Medidas de calidad del software.

#### 4.- Optimización y documentación:

- Refactorización.
  - Concepto y limitaciones.
  - Patrones de refactorización más usuales.
  - Refactorización y pruebas.
  - Herramientas de ayuda a la refactorización.
- Control de versiones.
  - Concepto y características.
  - Tipos.
  - Herramientas.
  - Repositorio.
- Documentación.
  - Uso de comentarios.
  - Alternativas.
  - Herramientas.

#### 5.- Elaboración de diagramas de clases:

- Notación de los diagramas de clases.
  - Clases. Atributos, métodos y visibilidad.
  - Objetos. Instanciación.
  - Relaciones. Herencia, composición, agregación, asociación y uso.
- Herramientas para la elaboración de diagramas de clases. Instalación.
- Generación de código a partir de diagramas de clases.
- Generación de diagramas de clases a partir de código.

## 6.- Elaboración de diagramas de comportamiento:

- Tipos. Campo de aplicación.
- Diagramas de casos de uso. Actores, escenario, relación de comunicación.
- Diagramas de interacción.
  - Diagramas de secuencia. Línea de vida de un objeto, activación, envío de mensajes.
  - Diagramas de colaboración. Objetos, mensajes.
- Diagramas de estados. Estados, eventos, señales, transiciones.
- Diagramas de actividades. Actividades, transiciones, decisiones y combinaciones.

## **11 CONTENIDOS.**

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

UD01: Desarrollo de software.

UD02: Entornos integrados 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. Atendiendo a este último punto, se ha incluido el tema de Lenguaje de script de cliente para que los alumnos tengan la base para afrontar las asignaturas del segundo curso.

## 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, 2.h
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, 3.j, 3.k, 3.l
UD04: Optimización y documentación.	4	4.a, 4.b, 4.c, 4.e, 4.f, 4.g, 4.h, 4.i
UD05: Elaboración de	5	5.a, 5.b, 5.c, 5.d, 5.e, 5.f, 5.g

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

#### 15 ESPECIFICACIÓN DE LAS UNIDADES DIDÁCTICAS

<b>UNIDAD DIDÁCTICA 1. Desarrollo de software.</b>	
<p>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.</p>	
<b>Criterios de evaluación:</b> 1.a, 1.b, 1.c, 1.d, 1.e, 1.f, 1.g	<b>N.º sesiones:</b> 10
Contenidos	Metodología
<ol style="list-style-type: none"> <li>1. Software. Tipos de software.</li> <li>2. Lenguajes de programación. Clasificación y tipos.</li> <li>3. Proceso de obtención de ejecutables. Compiladores e intérpretes.</li> <li>4. Ciclo de desarrollo de software. Modelo en cascada, modelo en espiral y metodologías ágiles.</li> </ol>	<ul style="list-style-type: none"> <li>● Exposiciones teóricas.</li> <li>● Búsquedas en internet de información relativa al tema.</li> <li>● Realización de ejercicios de resumen y síntesis.</li> </ul>
Observaciones	

<b>UNIDAD DIDÁCTICA 2. Entornos integrados de desarrollo.</b>	
<p>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.</p>	
<b>Criterios de evaluación:</b> 2.a, 2.b, 2.c, 2.d, 2.e, 2.f, 2.g, 2.h	<b>N.º sesiones:</b> 20
Contenidos	Metodología
<ol style="list-style-type: none"> <li>1. Entornos integrados de desarrollo. Definición, componentes y funciones.</li> <li>2. IDEs libres vs comerciales.</li> <li>3. Instalación y configuración de un entorno integrado de desarrollo.</li> <li>4. Herramientas CASE.</li> </ol>	<ul style="list-style-type: none"> <li>● Exposiciones teóricas.</li> <li>● Ejercicios y supuestos prácticos.</li> <li>● Prácticas de instalación, configuración y uso Colab, Netbeans, Visual Code e IntelliJ</li> </ul>
Observaciones:	

<b>UNIDAD DIDÁCTICA 3: Diseño y realización de pruebas</b>	
<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><b>Criterios de evaluación:</b> 3.a, 3.b, 3.c, 3.d, 3.e, 3.f, 3.g, 3.h, 3.i, 3.j, 3.k, 3.l</p>	<p><b>N.º sesiones:</b> 20</p>
Contenidos	Metodología
<ol style="list-style-type: none"> <li>1. Depuración de código fuente.</li> <li>2. Análisis estático de código fuente.</li> <li>3. Ciclos de pruebas del software.</li> <li>4. Pruebas unitarias. Diseño y ejecución.</li> <li>5. Pruebas funcionales. Diseño y ejecución.</li> </ol>	<ul style="list-style-type: none"> <li>● Exposiciones teóricas.</li> <li>● Ejercicios y supuestos prácticos.</li> <li>● Prácticas de depuración de código con NetBeans .</li> <li>● Prácticas de validación estática de código con PMP, Checkstyle y FindBug.</li> <li>● Prácticas de realización de pruebas unitarias con JUnit.</li> </ul>
Observaciones	

<b>UNIDAD DIDÁCTICA 4: Optimización y documentación</b>	
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.	
<b>Criterios de evaluación:</b> 4.a, 4.b, 4.c, 4.e, 4.f, 4.g, 4.h, 4.i	<b>N.º sesiones:</b> 18
Contenidos	Metodología
1. Refactorización. Definición y aplicación. 2. Patrones. Definición y uso en problemas generales. 3. Control de versiones: Git y GitHub 4. Documentación de aplicaciones. Documentación técnicas, documentación funcional y documentación de usuario.	<ul style="list-style-type: none"> <li>● Exposiciones teóricas.</li> <li>● Ejercicios y supuestos prácticos.</li> <li>● Prácticas de refactorización de código fuente.</li> <li>● Prácticas de aplicación de patrones de diseño.</li> <li>● Prácticas de documentación con JavaDoc.</li> </ul>
Observaciones:	

<b>UNIDAD DIDÁCTICA 5: Elaboración de diagramas de clases</b>	
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.	
<b>Criterios de evaluación:</b> 5.a, 5.b, 5.c, 5.d, 5.e, 5.f, 5.g	<b>N.º sesiones:</b> 14
Contenidos	Metodología
1. Lenguaje de modelado unificado (UML). 2. Diagramas de clases. Definición y componentes. 3. Relaciones. Herencia, composición, agregación, asociación y uso	<ul style="list-style-type: none"> <li>● Exposiciones teóricas.</li> <li>● Ejercicios y supuestos prácticos.</li> <li>● Prácticas de creación de modelos de clases</li> </ul>
Observaciones	

<b>UNIDAD DIDÁCTICA 6: Elaboración de diagramas de comportamiento</b>	
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.	
<b>Criterios de evaluación:</b> 6.a, 6.b, 6.c, 6.d, 6.e, 6.f, 6.g, 6.h	<b>N.º sesiones:</b> 14
<b>Contenidos</b>	<b>Metodología</b>
<ol style="list-style-type: none"> <li>1. Diagramas de casos de uso. Actores, escenario, relación de comunicación.</li> <li>2. Diagramas de interacción.</li> <li>3. Diagramas de secuencia. Línea de vida de un objeto, activación, envío de mensajes.</li> <li>4. Diagramas de colaboración. Objetos, mensajes.</li> <li>5. Diagramas de estados. Estados, eventos, señales, transiciones.</li> <li>6. Diagramas de actividades. Actividades, transiciones, decisiones y combinaciones.</li> </ol>	<ul style="list-style-type: none"> <li>● Exposiciones teóricas.</li> <li>● Ejercicios y supuestos prácticos.</li> <li>● Prácticas de creación de modelos de comportamiento con Eclipse Papyrus.</li> </ul>
<b>Observaciones:</b>	

**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.
- 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 la media aritmética de las notas de los resultados de aprendizaje 1, 6, 5
- **La nota de la segunda evaluación** será la media aritmética de las notas de los resultados de aprendizaje 2, 4
- **La nota de la tercera evaluación** será la nota del RA 3

- **La nota final del curso** vendrá determinada por la nota de todos los resultados de aprendizaje.
- Opciones para aprobar cada resultado de aprendizaje:
  - o 1ª oportunidad: cuando se realice el examen asociado al resultado de aprendizaje..
  - o En una recuperación de los RA no superados en el siguiente trimestre.
  - o 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:**

No se ha detectado ningún alumno

## 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<sup>a</sup> 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