

DEPARTAMENTO DE INFORMÁTICA  
**MÓDULO: Entornos de Desarrollo**  
PROGRAMACIÓN DOCENTE  
CURSO 2022-2023

# 1 Contenido

<b>2</b>	<b>INTRODUCCIÓN.....</b>	<b>3</b>
<b>3</b>	<b>OBJETIVOS .....</b>	<b>4</b>
<b>4</b>	<b>CONTENIDOS.....</b>	<b>6</b>
4.1	UNIDADES DE TRABAJO.....	6
<b>UT 1.</b>	<b>Desarrollo de Software.....</b>	<b>6</b>
<b>UT 2.</b>	<b>Instalación y uso de entornos de desarrollo.....</b>	<b>6</b>
<b>UT 3.</b>	<b>Sistemas de control de versiones.....</b>	<b>7</b>
<b>UT 4.</b>	<b>Elaboración de diagramas de clases.....</b>	<b>8</b>
<b>UT 5.</b>	<b>Elaboración de diagramas de comportamiento.....</b>	<b>8</b>
<b>UT 6.</b>	<b>Diseño y realización de pruebas. ....</b>	<b>10</b>
<b>UT 7.</b>	<b>Optimización y documentación. ....</b>	<b>10</b>
4.2	CONTENIDOS MÍNIMOS.....	11
<b>UT 1.</b>	<b>Desarrollo de Software.....</b>	<b>11</b>
<b>UT 2.</b>	<b>Instalación y Uso de Entornos de Desarrollo.....</b>	<b>11</b>
<b>UT 3.</b>	<b>Sistemas de Control de Versiones.....</b>	<b>12</b>
<b>UT 4.</b>	<b>Elaboración de Diagramas de Clases.....</b>	<b>12</b>
<b>UT 5.</b>	<b>Elaboración de Diagramas de Comportamiento.....</b>	<b>12</b>
<b>UT 6.</b>	<b>Diseño y Realización de Pruebas.....</b>	<b>12</b>
<b>UT 7.</b>	<b>Optimización y Documentación.....</b>	<b>12</b>
4.3	TEMPORALIZACIÓN.....	13
<b>5</b>	<b>METODOLOGÍA .....</b>	<b>13</b>
<b>6</b>	<b>EVALUACIÓN.....</b>	<b>15</b>
6.1	CRITERIOS DE EVALUACIÓN .....	15
6.2	INSTRUMENTOS DE EVALUACIÓN.....	16
6.3	CRITERIOS DE CALIFICACIÓN .....	17
6.4	EVALUACIÓN FINAL.....	18
6.5	RECUPERACIÓN ORDINARIA .....	18
6.6	EVALUACIÓN EXTRAORDINARIA .....	19
6.7	EVALUACIÓN DE PENDIENTES.....	19
6.8	PROGRAMACIÓN DE RECUPERACIÓN PARA ALUMNOS CON IMPOSIBILIDAD DE APLICACIÓN DEL PROCESO DE EVALUACIÓN ORDINARIA.....	19
<b>7</b>	<b>ATENCIÓN A LA DIVERSIDAD.....</b>	<b>19</b>
<b>8</b>	<b>ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES.....</b>	<b>20</b>
<b>9</b>	<b>INDICADORES DE LOGRO Y PROCEDIMIENTO DE EVALUACIÓN DE LA APLICACIÓN Y DESARROLLO DE LA PROGRAMACIÓN DOCENTE .....</b>	<b>20</b>

## 2 INTRODUCCIÓN

El contexto legislativo viene definido en primer lugar y de forma general por la *Ley Orgánica 2/2006, de 3 de mayo, de Educación*, LOE (BOE 106, 4/5/2006, 17158-17207).

La formación profesional se regula mediante el *Real Decreto 1538/2006, de 15 de diciembre, por el que se establece la ordenación general de la formación profesional del sistema educativo* (BOE 3, 3/1/2007, 182-193).

El título y las enseñanzas mínimas se regulan mediante el *Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas* (BOE 143, 12/06/2010).

DECRETO 184/2012, de 8 de agosto, establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en *Desarrollo de Aplicaciones Web* en el Principado de Asturias.

**Entornos de Desarrollo** es un módulo que se imparte en el primer curso del ciclo formativo de grado superior **Desarrollo de Aplicaciones Web**. Tiene una duración de 90 horas, con una distribución de 3 horas semanales.

Las características del ciclo se resumen en esta tabla:

TÍTULO	<b>Técnico Superior en Desarrollo de Aplicaciones Web</b>
NORMA	Real Decreto 686/2010, de 20 de mayo (BOE de 12/06/2010)
NIVEL	Formación profesional de Grado Superior
DURACIÓN TOTAL	2000 horas
FAMILIA PROFESIONAL	Informática y Comunicaciones
REFERENTE EUROPEO	CINE-5b (Clasificación Internacional Normalizada de la Educación)
CÓDIGO	IFC303LOE
DENOMINACIÓN	Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web

Esta programación didáctica es considerada un documento de referencia de carácter dinámico, es decir, susceptible de ser actualizada con la finalidad de mejorar el proceso de enseñanza-aprendizaje para conseguir los objetivos terminales del perfil profesional que nos compete. Será aconsejable analizar los posibles errores, actualizar los contenidos en función de la evolución de los sectores productivos del mercado, en definitiva, evaluar la programación didáctica para proponer modificaciones que permitan la mejora continua.

### 3 OBJETIVOS

**Este módulo contribuye a alcanzar los objetivos generales del ciclo:**

- d) Ajustar parámetros analizando la configuración para gestionar servidores de aplicaciones.
- 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.

**Y las competencias profesionales, personales y sociales:**

- d) Gestionar bases de datos, interpretando su diseño lógico y verificando integridad, consistencia, seguridad y accesibilidad de los datos
- f) Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.
- h) Desarrollar componentes multimedia para su integración en aplicaciones web, empleando herramientas específicas y siguiendo las especificaciones establecidas.
- i) Integrar componentes multimedia en el interface de una aplicación web, realizando el análisis de interactividad, accesibilidad y usabilidad de la aplicación
- j) Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación.

**Los resultados de aprendizaje asociados a este módulo son:**

#### **R.A.-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.

#### **R.A.-2**

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

#### **R.A.-3**

Verifica el funcionamiento de programas, diseñando y realizando pruebas

**R.A.-4**

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

**R.A.-5**

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

**R.A.-6**

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

Las líneas de actuación en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo están relacionados con:

- La interpretación de documentación técnica.
- La instalación, configuración y personalización de diversos entornos de desarrollo.
- La utilización de distintos entornos de desarrollo para la edición y prueba de aplicaciones.
- La utilización de herramientas de depuración, optimización y documentación de aplicaciones.
- La generación de diagramas técnicos.
- La elaboración de la documentación interna de la aplicación.

## 4 CONTENIDOS

### 4.1 UNIDADES DE TRABAJO

#### **UT 1. DESARROLLO DE SOFTWARE.**

Se relaciona con el resultado de aprendizaje: R.A.-1

##### **CONTENIDOS**

- Concepto de programa informático y características.
- Concepto de lenguajes de programación.
- Tipos de lenguajes de programación:
  - Según nivel de abstracción.
  - Según la forma de ejecución.
  - Según el paradigma de la programación.
- Características de los lenguajes más difundidos.
- Fases del desarrollo de una aplicación: análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento.
- Código fuente, código objeto y código ejecutable; máquinas virtuales.
- Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.
- Herramientas utilizadas en programación. Ejemplos: Pseudocódigo y Diagramas de flujo.

##### **ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE**

- Explicar el tema con ayuda de presentaciones digitales y fotocopias.
- Elaborar mapas conceptuales de los temas desarrollados con ayuda de herramientas como Cmaptools.
- Buscar documentación en la Web.
- Reconocer las fases de un código.
- Clasificar los distintos lenguajes de programación.
- Elaboración de diagramas de flujo para la representación de algoritmos sencillos.

#### **UT 2. INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO.**

Se relaciona con el resultado de aprendizaje: R.A.-2

##### **CONTENIDOS**

- Funciones de un entorno de desarrollo.
- Instalación de un entorno de desarrollo.
  - Instalación entorno Eclipse.

- Instalación Plugins.
- Uso básico de un entorno de desarrollo.
  - Paquetes.
  - Proyectos.
  - Clases.
- Edición de programas.
  - Vistas.
- Generación de programas ejecutables.
  - Compilar y detectar errores.
  - Ejecución.
- Exportación e Importación.

### ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE

- Explicar el tema con ayuda de presentaciones digitales.
- Elaborar mapas conceptuales de los temas desarrollados con ayuda de herramientas como CmapTools.
- Utilizar entorno de desarrollo libre Eclipse.
- Adición y eliminación de módulos.
- Exportación e Importación con menú y con jar JAVA.
- Compilación y ejecución con menú y con javac y java JAVA. Generación de ejecutables.

### UT 3. SISTEMAS DE CONTROL DE VERSIONES

Se relaciona con el resultado de aprendizaje: **R.A.-4**

### CONTENIDOS

- Fundamentos de un Sistema de Control de Versiones
- Fundamentos de GIT
- Instalación de GIT
- Comandos de GIT
- Gestión de log
- Crear alias para los comandos
- Commits
- Repositorios
- Ramas, uniones, conflictos y tags
- Git Stash y Git Rebase
- GitHub, Git Remote, Push & Pull

### ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE

- Utilizar el sistema de Gestión de Versiones GIT para realizar prácticas de cada uno de los puntos tratados en el temario
- Creación de una cuenta en Github
- Instalación de plugins de IDEs para facilitar la operación con Git desde el propio entorno
- Gestión de los repositorios siguiendo las buenas prácticas propuestas

### UT 4. ELABORACIÓN DE DIAGRAMAS DE CLASES.

Se relaciona con el resultado de aprendizaje: **R.A.-5**

#### CONTENIDOS

- Características principales del lenguaje UML.
- Historia de UML.
- Principales diagramas UML.
- Clases. Atributos, métodos y visibilidad.
- Objetos. «Instanciación».
- Relaciones. Herencia, composición, agregación y especialización.
- Notación de los diagramas de clases.

### ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE

- Explicar el tema con ayuda de presentaciones digitales, fotocopias, etc.
- Utilizar herramientas gráficas para representar el diseño UML: DIA , StarUml y Eclipse.
- Instalación del módulo IDE para trabajar en Eclipse.
- Conocer los elementos fundamentales de la POO.
- Identificar las distintas clases de un enunciado de un ejercicio.
- Identificar las distintas relaciones entre las clases de un enunciado de un ejercicio.
- Generar código a partir de diagramas de clases.
- Generar diagramas de clases mediante ingeniería inversa.

### UT 5. ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO.

Se relaciona con el resultado de aprendizaje: **R.A.-6**

#### CONTENIDOS

- Tipos de diagramas de comportamiento. Campos de aplicación.
- Diagramas de casos de uso.

- Actores,
- Escenario y casos de uso
- Relación de comunicación: comunica, inclusión, extensión, herencia
- Diagramas de secuencia.
  - Línea de vida de un objeto.
  - Activación, tiempo.
  - Envío de mensajes.
- Diagramas de colaboración.
  - Objetos.
  - Mensajes.
- Diagramas de comunicación.
- Diagramas de estado.
  - Estados.
  - Transiciones.
  - Sucesos.
  - Acciones.
- Diagramas de Actividad.
  - Actividades.
  - Flujo de control.
  - Decisiones.
  - Rutas concurrentes.

#### **ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE**

- Explicar el tema con ayuda de presentaciones digitales.
- Elaborar mapas conceptuales de los temas desarrollados con ayuda de herramientas como CmapTools.
- Utilizar herramientas gráficas para representar el diseño UML: DIA , StarUml y Eclipse.
- Interpretación de diagramas de interacción.
- Elaboración de diagramas de interacción.
- Interpretación de diagramas de estado.
- Elaboración de diagramas de estado.
- Interpretación de diagramas de actividades.
- Elaboración de diagramas de actividades.
- Interpretación de diagramas de secuencia.
- Elaboración de diagramas de secuencia.
- Interpretación de diagramas de comportamiento.
- Elaboración de diagramas de comportamiento.

## **UT 6. DISEÑO Y REALIZACIÓN DE PRUEBAS.**

Se relaciona con el resultado de aprendizaje: **R.A.-3**

### **CONTENIDOS**

- Planificación de Pruebas.
- Técnicas de diseño de casos de prueba: Caja blanca y caja negra.
- Pruebas de código:
  - Prueba del camino básico.
  - Partición o clase de equivalencia.
  - Análisis de valores límite.
- Tipos de pruebas: funcionales, estructurales, regresión, etc.
- Procedimientos y casos de prueba.
- Pruebas unitarias con Junit.
  - Creación, preparación y ejecución de clases de prueba.
  - Pruebas parametrizadas.
  - Suite de pruebas.
- Herramientas de depuración:
  - Puntos de ruptura y seguimiento.
  - Examen y modificación de variables.

### **ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE**

- Explicar el tema con ayuda de presentaciones digitales y fotocopias.
- Elaborar mapas conceptuales de los temas desarrollados con ayuda de herramientas como Cmaptools.
- Utilizar herramientas de un entorno de desarrollo para la depuración.
- Utilizar herramientas de un entorno de desarrollo para puntos de ruptura y seguimiento.
- Utilizar herramientas de un entorno de desarrollo para modificar el comportamiento de un programa en tiempo de ejecución.
- Identificar las distintas pruebas.
- Definir nuevos casos de pruebas: unitarias de clases y funciones.
- Trabajar con pruebas automáticas.
- Documentación de las incidencias detectadas.

## **UT 7. OPTIMIZACIÓN Y DOCUMENTACIÓN.**

Se relaciona con el resultado de aprendizaje: **R.A.-4**

## CONTENIDOS

- Refactorización.
  - Concepto.
  - Limitaciones.
  - Patrones de refactorización más usuales.
  - Refactorización y pruebas.
  - Herramientas de ayuda a la refactorización.
- Documentación.
  - Uso de comentarios.
  - Generación automática de documentación.

## ACTIVIDADES DE ENSEÑANZA-APRENDIZAJE

- Explicar el tema con ayuda de presentaciones digitales y fotocopias.
- Elaborar mapas conceptuales de los temas desarrollados con ayuda de herramientas como Cmaptools.
- Utilizar herramientas de entorno de desarrollo.
- Identificación de los patrones de refactorización más frecuentes.
- Elaboración de pruebas asociadas a la refactorización.
- Revisión del código fuente mediante un analizador de código.
- Identificación de posibilidades de configuración de un analizador de código.
- Aplicación de patrones de refactorización en un entorno de desarrollo.
- Documentación de clases mediante un entorno de desarrollo.

## 4.2 CONTENIDOS MÍNIMOS

### UT 1. DESARROLLO DE SOFTWARE.

- Reconocer las fases de desarrollo del software.
- Clasificar los distintos lenguajes de programación.
- Representar mediante diagrama de flujo estructuras básicas de programación como alternativas o bucles.

### UT 2. INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO.

- Uso básico de un entorno de desarrollo.
  - Paquetes.
  - Proyectos.
  - Clases.
- Edición de programas.
  - Vistas.
- Generación de programas ejecutables.
  - Compilar y detectar errores.
  - Ejecución.
- Exportación e Importación.

### **UT 3. SISTEMAS DE CONTROL DE VERSIONES.**

- Fundamentos de GIT
- Instalación de GIT
  - Comandos de GIT
- Commits
- Git Stash y Git Rebase
- GitHub, Git Remote, Push & Pull

### **UT 4. ELABORACIÓN DE DIAGRAMAS DE CLASES.**

- Identificar las distintas clases en un enunciado de un ejercicio.
- Identificar las distintas relaciones entre las clases en un enunciado de un ejercicio.
- Generar código a partir de diagramas de clases.
- Generar diagramas de clases mediante ingeniería inversa.

### **UT 5. ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO.**

- Interpretación de diagramas de interacción.
- Elaboración de diagramas de interacción.
- Interpretación de diagramas de estado.
- Elaboración de diagramas de estado.
- Interpretación de diagramas de actividades.
- Elaboración de diagramas de actividades.
- Interpretación de diagramas de secuencia.
- Elaboración de diagramas de secuencia.
- Interpretación de diagramas de comportamiento.
- Elaboración de diagramas de comportamiento.

### **UT 6. DISEÑO Y REALIZACIÓN DE PRUEBAS.**

- Identificar los distintos tipos de pruebas.
- Definir casos de prueba siguiendo los criterios establecidos.
- Conocer los conceptos de análisis de valores límite y clases de equivalencia.
- Conocer el concepto de Test Driven Development y hacer ejemplos sencillos basados en este enfoque.
- Trabajar con pruebas automáticas.
- Documentación de las incidencias detectadas.

### **UT 7. OPTIMIZACIÓN Y DOCUMENTACIÓN.**

- Identificación de los patrones de refactorización más frecuentes.
- Elaboración de pruebas asociadas a la refactorización.
- Revisión del código fuente mediante un analizador de código.
- Identificación de posibilidades de configuración de un analizador de código.
- Aplicación de patrones de refactorización en un entorno de desarrollo.
- Documentación de clases mediante un entorno de desarrollo.

### 4.3 TEMPORALIZACIÓN

EVALUACIÓN	UNIDAD DE TRABAJO	HORAS
PRIMERA	UT1. Desarrollo de Software	8
	UT2. Instalación y Uso de Entornos de Desarrollo	14
	UT3. Sistemas de control de versiones	12
SEGUNDA	UT4. Elaboración de Diagramas de Clases	18
	UT5. Elaboración de Diagramas de Comportamiento	10
TERCERA	UT5. Elaboración de Diagramas de Comportamiento	12
	UT6. Diseño y Realización de Pruebas	14
	UT7. Optimización y Documentación	8

## 5 METODOLOGÍA

Las clases se imparten con una introducción conceptual y una secuencia estructurada de prácticas acordes con cada unidad de trabajo. En general, a la hora de trabajar los contenidos procedimentales, se utilizará la metodología “Aprender haciendo”, es decir, se realizarán actividades eminentemente prácticas y de carácter individual o por grupos.

El aprendizaje estará basado en proyectos o en problemas, dado que los trabajos o proyectos en este módulo pueden reproducir perfectamente una situación real de una empresa. Este tipo de metodologías aumentan la motivación del alumnado, fomentan el trabajo en equipo y la investigación, y les ayuda a relacionar mejor la teoría y la práctica.

El profesor deberá realizar un seguimiento cercano e individualizado del proceso de aprendizaje de cada alumno, realizando anotaciones sistemáticas de avances y dificultades, y asesorando y guiando al alumno en su proceso de aprendizaje. La utilización de un aula de moodle como apoyo a las clases presenciales ayudará a llevar un registro sistemático de la actividad de cada alumno.

Finalmente, se debe remarcar, en todo momento, que aquellos contenidos relacionados con las actitudes y comportamientos que caracterizan al profesional, especialmente los relativos al orden, rigor, seguridad, respeto, colaboración, atención y asistencia, deben ser tratados con el mayor énfasis posible, ya que, de ello depende que otros profesionales puedan desempeñar su trabajo.

### RECURSOS DIDÁCTICOS

- Se usará como recurso fundamental el siguiente libro:  
 Editorial: RA-MA  
 Título: “Entornos de desarrollo”.  
 Autor: Carlos Casado Iglesias.
- Se usará como recurso complementario el siguiente libro:  
 Editorial: Síntesis  
 Título: “Entornos de desarrollo”.  
 Autor: Juan Carlos Moreno Pérez.

- El alumno tomará sus propias notas sobre las explicaciones que el profesor realiza.
- Recomendación de lecturas auxiliares y de apoyo.
- Un cañón proyector.
- Conexión a Internet.
- El aula dispone de 30 ordenadores, dotados del software necesario:
  - Sistemas operativos: Windows 10.
  - Entornos de Desarrollo
    - Libre: Eclipse, Netbeans
- Programas auxiliares (lector PDF, compresor, antivirus, etc.)

En caso de enfermedad el alumnado podrá consultar información del módulo a través del Microsoft Teams.

En esta plataforma se irán colgando apuntes, actividades para realizar, etc.

## 6 EVALUACIÓN

### 6.1 CRITERIOS DE EVALUACIÓN

**Resultado de Aprendizaje 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.

- a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador y periféricos, entre otros.
- b) Se han identificado las fases de desarrollo de una aplicación informática.
- c) Se han diferenciado los conceptos de código fuente, código objeto y código 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.
- f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.

**Resultado de Aprendizaje 2.** Evalúa entornos integrados de desarrollo, analizando, sus características para editar código fuente y generar ejecutables.

- 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.

**Resultado de Aprendizaje 3.** Verifica el funcionamiento de programas, diseñando y realizando pruebas:

- 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.

**Resultado de Aprendizaje 4.** Optimiza código empleando las herramientas disponibles en el entorno de desarrollo:

- 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.

**Resultado de Aprendizaje 5.** Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno:

- Se han identificado los conceptos básicos de la programación orientada a objetos.
- Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.
- Se han identificado las herramientas para la elaboración de diagramas de clases.
- Se ha interpretado el significado de diagramas de clases.
- Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
- Se ha generado código a partir de un diagrama de clases.
- Se ha generado un diagrama de clases mediante ingeniería inversa.

**Resultado de Aprendizaje 6.** Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno:

- Se han identificado los distintos tipos de diagramas de comportamiento.
- Se ha reconocido el significado de los diagramas de casos de uso.
- Se han interpretado diagramas de interacción.
- Se han elaborado diagramas de interacción sencillos.
- Se ha interpretado el significado de diagramas de actividades.
- Se han elaborado diagramas de actividades sencillos.
- Se han interpretado diagramas de estados.
- Se han planteado diagramas de estados sencillos.

## 6.2 INSTRUMENTOS DE EVALUACIÓN

La evaluación de los aprendizajes se realizará tomando como referencia los objetivos generales y los criterios de evaluación programados en el módulo.

Los criterios de evaluación establecen el nivel aceptable de consecución de los objetivos correspondientes, señalando los contenidos mínimos que deben alcanzarse en el proceso de enseñanza-aprendizaje.

En el proceso de evaluación debemos tener en cuenta que:

- En el desarrollo de las unidades de trabajo, se llevará a cabo un proceso de evaluación continua, lo que requiere la asistencia regular del alumno a las clases y la realización a tiempo de las actividades encomendadas por el profesor.
- El profesor llevará a cabo, mediante la plataforma Aula Virtuales y/o Microsoft Teams la subida de los recursos necesarios para el desarrollo de las distintas unidades de trabajo, así como el control de las actividades a realizar por el alumnado.
- A lo largo del curso el alumnado almacenará en un medio digital sus apuntes, trabajos, actividades y ejercicios que vayan realizando.
- En el desarrollo de los aprendizajes, cuando se lleven a cabo actividades y trabajos en grupo, se calificarán los mismos, evaluándose, en su caso, tanto la calidad de los trabajos como la claridad de las exposiciones, el interés y la participación en las actividades, teniéndose en cuenta también la coordinación del alumnado en el grupo y el diálogo con los demás grupos.
- También es importante la resolución de ejercicios y la realización de trabajos y actividades individuales con el fin de conocer y evaluar el grado de comprensión con que se van adquiriendo individualmente los conocimientos, y para poner de manifiesto las deficiencias o errores en la comprensión de los conceptos y procesos.

La evaluación continua se realiza, en consecuencia, valorando los siguientes instrumentos:

**A. OBSERVACION SISTEMÁTICA.**

Se evaluará la actitud que manifiesta el alumno a lo largo del curso. El profesor registrará la evolución de la actitud de cada alumno basándose en los siguientes indicadores:

- Participación e interés.
- Trabajo en clase y en casa.
- Iniciativa y autonomía.
- Uso adecuado de los medios y recursos.

**B. ANÁLISIS DE PRODUCCIONES DE LOS ALUMNOS.**

- Trabajos y actividades, tanto individuales como en grupo y resolución de ejercicios.
- Repositorio digital de documentos, apuntes y prácticas.

**C. REALIZACIÓN DE PRUEBAS ESPECÍFICAS.**

Puede tratarse de pruebas escritas o realizadas en un equipo informático, según la unidad de trabajo a evaluar.

### 6.3 CRITERIOS DE CALIFICACIÓN

#### EVALUACIÓN TRIMESTRAL

La calificación de los alumnos en cada evaluación se calculará de acuerdo con los siguientes porcentajes:

**A. OBSERVACIÓN SISTEMÁTICA ..... 10%**

Valoración de la actitud observada en el alumnado. Se tendrá en cuenta:

- **La participación e interés.** Se trata de valorar la participación del alumnado en clase, sus intervenciones y explicaciones sobre actividades y ejercicios propuestos teniéndose en cuenta, también, su interés y dedicación.
- **El trabajo en clase y en casa.** El alumno se esfuerza durante las clases y en casa en la realización de las actividades planteadas por parte del profesor.
- **La iniciativa y autonomía.** El alumno tiene espíritu emprendedor, manifiesta interés por ampliar sus conocimientos y buscar soluciones para resolver nuevos problemas o de mayor complejidad.
- **Uso adecuado de los medios y recursos.** Se valorará la utilización adecuada de los recursos materiales del centro por parte del alumno, con responsabilidad. El uso y cuidado de su equipo informático, así como el de sus compañeros y compañeras, el uso apropiado del software, solo el permitido, el mantenimiento de la configuración del equipo y el orden del puesto de estudio.

Una actitud del alumno *negativa* se calificará con 0, una actitud *aceptable* con 5, una actitud *positiva* con 7,5 y una actitud *excelente* se calificará con 10.

**B. PRODUCCIONES DE LOS ALUMNOS ..... 20%**

Este apartado se refiere a las actividades o ejercicios prácticos individuales realizados en clase o de forma telemática, que el profesor considere oportuno evaluar durante el trimestre.

- Las actividades se puntuarán entre 0 y 10 puntos.
- Estarán sujetas a plazos de entrega con fechas límite.
- Aquellas actividades no entregadas serán calificadas con una puntuación de 0.
- Se tendrá en cuenta la idoneidad de las respuestas, su calidad de redacción y presentación.
- Se podrá requerir en cualquier momento la defensa de dichas actividades para garantizar que han sido realizadas por el alumno o alumna en cuestión.
- Se contemplará la posibilidad de entregas fuera de plazo a modo de recuperación, fijando nuevos plazos y fechas límite, en cuyo caso la calificación no será superior a 5 puntos.

La puntuación obtenida en este apartado será la media de las notas obtenidas en cada actividad, pudiendo ser una media ponderada según el grado de dificultad de estas.

**C. PRUEBAS TEÓRICO-PRÁCTICAS INDIVIDUALES:.....70%**

Estas pruebas, que podrán ser sobre parte de una unidad de trabajo, una o varias, tendrán una calificación de entre 0 y 10.

Consistirán en la realización de una o varias actividades relacionadas con los contenidos del trimestre, pudiendo ser de tipo test, cuestiones teóricas y/o uno o varios supuestos prácticos.

La puntuación obtenida en este apartado será la media aritmética de las notas obtenidas en cada prueba.

## 6.4 EVALUACIÓN FINAL

La calificación final del alumno se calculará como la media de las calificaciones obtenidas en las tres evaluaciones trimestrales. Para que la evaluación final sea positiva la nota media obtenida ha de ser igual o superior a 5 puntos y tener las tres evaluaciones aprobadas.

**Nota\_Evaluación\_Final= Redondear (Nota media de las evaluaciones trimestrales)**

## 6.5 RECUPERACIÓN ORDINARIA

Los alumnos que no superen alguna evaluación, por haber obtenido en ella una nota inferior a 5, deberán realizar una recuperación de la misma, que consistirá en:

**Prueba final:** Consistirá en la realización de una prueba teórico-práctica basada en los contenidos mínimos correspondientes a las Unidades de Trabajo de cada una de las evaluaciones que el alumno tenga pendientes. Será necesaria una calificación de al menos un 5 para considerar aprobado el módulo.

La nota obtenida en la prueba sustituirá a la nota de las evaluaciones pendientes para aplicarla a la media en la evaluación final.

## 6.6 EVALUACIÓN EXTRAORDINARIA

Recuperación extraordinaria de junio. A estas pruebas se someterán los alumnos que no hayan obtenido una calificación mínima de 5 en el examen final de junio y se basarán en los contenidos impartidos durante el curso, realizándose uno o varios exámenes teóricos y/o prácticos. No se exigirá la presentación de actividades para evaluarlas, aunque el alumno tendrá acceso a todas las realizadas durante el curso

La prueba será calificada de 0 a 10 puntos de acuerdo con la puntuación especificada en la misma, siendo necesaria una calificación de al menos un 5 para considerar aprobado el módulo. La nota obtenida en la prueba será la que figure en la evaluación extraordinaria de junio.

## 6.7 EVALUACIÓN DE PENDIENTES

Los alumnos/as que estén matriculados en segundo curso y tienen pendiente el módulo realizarán una prueba en el mes de febrero en primera convocatoria y/o en el mes de junio en segunda convocatoria. Esta prueba será única y estará basada en los contenidos mínimos de todas y cada una de las unidades temáticas del módulo.

## 6.8 PROGRAMACIÓN DE RECUPERACIÓN PARA ALUMNOS CON IMPOSIBILIDAD DE APLICACIÓN DEL PROCESO DE EVALUACIÓN ORDINARIA

A los alumnos que hayan superado el 15% de horas lectivas en faltas de asistencia en este módulo, en un trimestre determinado, no se les podrá aplicar los criterios de evaluación ordinaria y tendrán que ser evaluados de forma extraordinaria.

Realizarán una prueba específica en el trimestre en el que no se les pueda aplicar la evaluación continua, basada en los criterios de evaluación de las unidades temáticas que no haya superado hasta el momento. En este caso,

$$\text{Nota\_Evaluación} = \text{Redondear } (0,90 * \text{Nota pruebas Objetivas} + 0,10 * \text{Nota Trabajos})$$

La nota de evaluación ha de ser igual o superior a 5 puntos

## 7 ATENCIÓN A LA DIVERSIDAD

Las siguientes actuaciones permiten atender las diferencias individuales del alumnado:

- Se dará prioridad a los contenidos procedimentales frente a los conceptuales.
- Se hará una clara diferenciación entre conceptos básicos y otros más avanzados que dificulten el proceso de enseñanza-aprendizaje.
- Se graduará la dificultad de las tareas (actividades de refuerzo, pruebas de evaluación, etc.) que se propongan, de forma que todo el alumnado pueda encontrar espacios de respuesta acordes a sus capacidades.
- Se preferirá el trabajo en grupo para la realización de las actividades en el aula, fomentando el apoyo y la colaboración mutua.
- Se propondrán actividades complementarias, tanto durante el desarrollo de los contenidos como en las fases de realización de los trabajos, afines a las actividades que se estén realizando.
- Se adecuarán tiempos y espacios, en la medida de lo posible, a las características físicas y/o intelectuales de cada alumno para facilitar el desempeño de su actividad.
- Se reducirán al mínimo indispensable las tareas para casa para evitar sobreesfuerzos y estrés en el alumnado.
- Se mantendrá un repositorio digital de la asignatura para facilitar al alumno el acceso a los materiales, la realización de tareas, la colaboración con los compañeros y la comunicación con el profesor.
- Asegurarse de que las instrucciones que se dan al grupo han sido comprendidas realmente por el alumno. A menudo es preciso dirigirlas personalmente de forma explícita, oralmente o por escrito, con frases directas, cortas y simples.
- Proporcionarles la información por escrito. La información escrita demanda menos conocimiento social que la interacción verbal y les proporciona un recuerdo visual estable.
- Dar más tiempo para terminar sus tareas y en las pruebas.
- Establecer pautas metodológicas a la hora de desarrollar el temario.
- Sentarse adelante.

## 8 Actividades complementarias y extraescolares

No se tiene prevista ninguna actividad.

## 9 Indicadores de logro y procedimiento de evaluación de la aplicación y desarrollo de la programación docente

El procedimiento de evaluación de la programación será el que el propio centro determine en la Programación General Anual.

Se incluirán en un cuestionario específico los siguientes indicadores de logro:

- Resultados de la evaluación del curso en cada una de las materias, por curso y grupo.
- Adecuación de los materiales, recursos didácticos, y distribución, en su caso, de espacios y tiempos a la secuenciación de contenidos y criterios de evaluación asociados.
- Contribución de los métodos pedagógicos y medidas de atención a la diversidad aplicadas a la mejora de los resultados obtenidos.
- Valoración de actividades complementarias organizadas por el Departamento o con participación del mismo.