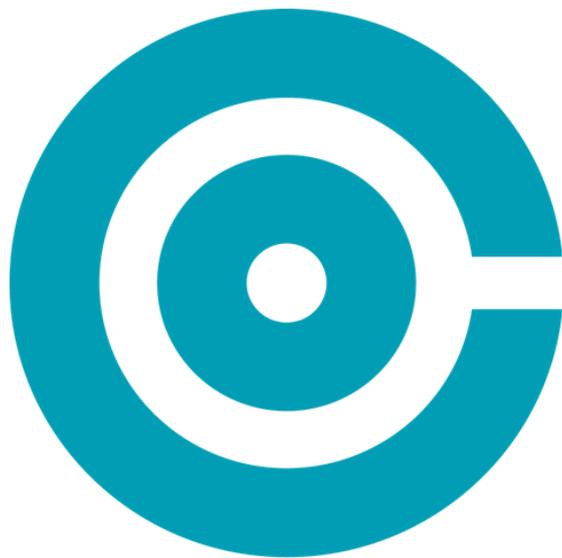


PROGRAMACIÓN DIDÁCTICA

CURSO 2023-2024

MÓDULO PROGRAMACIÓN 1º



**IES
CO
MER
CIO**

FAMILIA PROFESIONAL INFORMÁTICA Y COMUNICACIONES

**CICLO FORMATIVO DE GRADO SUPERIOR: DESARROLLO DE APLICACIONES
MULTIPLATAFORMA, DESARROLLO DE APLICACIONES WEB**

CURSO: 1º

PROFESORES:

DIURNO: DANIEL ESPINOSA AZOFRA

VESPERTINO: ANA PÉREZ SEIJAS, CARLOS SÁENZ

ÍNDICE

1.	INTRODUCCIÓN	2
2.	OBJETIVOS.....	3
2.1	Competencia general del Título.....	3
2.2	Cualificaciones profesionales y unidades de competencia.....	3
2.3	Competencias profesionales, personales y sociales del módulo	3
2.4	Objetivos generales del ciclo que contribuye a alcanzar el módulo.....	3
2.5	Objetivos del módulo	3
3.	CONTENIDOS Y DISTRIBUCIÓN TEMPORAL.....	3
3.1	Contenidos básicos	5
3.2	Contenidos actitudinales.....	7
3.3	Distribución temporal	8
4.	UNIDADES DIDÁCTICAS	9
5.	METODOLOGÍA.....	9
5.1	Materiales y recursos didácticos	28
6.	EVALUACIÓN.....	29
6.1	Criterios de evaluación	29
6.2	Instrumentos y procedimientos de evaluación	32
6.3	Criterios de calificación	33
6.4	Actividades de refuerzo o recuperación.....	36
6.5	Criterios de recuperación.....	33
6.6	Recuperación de módulos pendientes.....	37
7.	ATENCIÓN AL ALUMNADO CON NECESIDADES ESPECÍFICAS DE APOYO EDUCATIVO.....	38
8.	ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES	39
9.	NECESIDADES Y PROPUESTAS DE FORMACIÓN DEL PROFESORADO.....	40
10.	BIBLIOGRAFÍA DE REFERENCIA	41

1. INTRODUCCIÓN

El módulo de Programación, se encuadra dentro de las especificaciones del título de Técnico Superior de Desarrollo de Aplicaciones Multiplataforma y Web, integrado en la Familia Profesional de Informática y Comunicaciones, recogida en el Real Decreto 1147/2011, de 29 de julio, publicado en el BOE con fecha viernes 21 de octubre de 2011.

Referente europeo: CINE-5b (Clasificación Internacional Normalizada de la Educación).

Este módulo tiene una duración total de 245 horas a razón de 8 horas semanales.

Además de todas las referencias legislativas anteriormente mencionadas también se ha respetado, para el desarrollo de la presente programación didáctica, el Proyecto Educativo de Centro en el que se aplica y su Programación General Anual.

2. OBJETIVOS

2.1 Competencia general del Título

La competencia general de este título consiste en desarrollar, implantar, documentar y mantener aplicaciones informáticas multiplataforma, utilizando tecnologías y entornos de desarrollo específicos, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de «usabilidad» y calidad exigidas en los estándares establecidos.

2.2 Cualificaciones profesionales y unidades de competencia

Cualificación profesional: Programación con lenguajes orientados a objetos y bases de datos relacionales IFC080_3 (R.D. 295/2004, de 20 de febrero).

Unidad de competencia: UC0227_3 Desarrollar componentes software en lenguajes de programación orientados a objetos.

2.3 Competencias profesionales, personales y sociales del módulo

Las competencias a), e), f), i), j), t) y w) del título.

2.4 Objetivos generales del ciclo que contribuye a alcanzar el módulo

La formación del módulo contribuye a alcanzar los objetivos generales e), j), q) y w) del ciclo formativo.

2.5 Objetivos del módulo

- La interpretación y aplicación de los principios de la programación orientada a objetos.
- La evaluación, selección y utilización de herramientas y lenguajes de programación orientados a objetos
- La utilización de las características específicas de lenguajes y entornos de programación en el desarrollo de aplicaciones informáticas.
- La identificación de las funcionalidades aportadas por los sistemas gestores de bases de datos y su incorporación a los programas desarrollados.

- La documentación de los programas desarrollados.

3. CONTENIDOS Y DISTRIBUCIÓN TEMPORAL

3.1 Contenidos básicos

Identificación de los elementos de un programa informático:

- Estructura y bloques fundamentales.
- Variables.
- Tipos de datos.
- Literales.
- Constantes.
- Operadores y expresiones.
- Conversiones de tipo.
- Comentarios.

Utilización de objetos:

- Características de los objetos.
- Instanciación de objetos.
- Utilización de métodos.
- Utilización de propiedades.
- Utilización de métodos estáticos.
- Constructores.
- Destrucción de objetos y liberación de memoria.

Uso de estructuras de control:

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.
- Control de excepciones.

Desarrollo de clases:

- Concepto de clase.
- Estructura y miembros de una clase.

- Creación de atributos.
- Creación de métodos.
- Creación de constructores.
- Utilización de clases y objetos.
- Utilización de clases heredadas.

Lectura y escritura de información:

- Tipos de flujos. Flujos de bytes y de caracteres.
- Clases relativas a flujos.
- Utilización de flujos.
- Entrada desde teclado.
- Salida a pantalla.
- Ficheros de datos. Registros.
- Apertura y cierre de ficheros. Modos de acceso.
- Escritura y lectura de información en ficheros.
- Utilización de los sistemas de ficheros.
- Creación y eliminación de ficheros y directorios.
- Interfaces.
- Concepto de evento.
- Creación de controladores de eventos.

Aplicación de las estructuras de almacenamiento:

- Estructuras.
- Creación de arrays.
- Arrays multidimensionales.
- Cadenas de caracteres.
- Listas.

Utilización avanzada de clases:

- Composición de clases.
- Herencia.

- Superclases y subclases.
- Clases y métodos abstractos y finales.
- Sobreescritura de métodos.
- Constructores y herencia.

Mantenimiento de la persistencia de los objetos:

- Bases de datos orientadas a objetos.
- Características de las bases de datos orientadas a objetos.
- Instalación del gestor de bases de datos.
- Creación de bases de datos.
- Mecanismos de consulta.
- El lenguaje de consultas: sintaxis, expresiones, operadores.
- Recuperación, modificación y borrado de información.
- Tipos de datos objeto; atributos y métodos.
- Tipos de datos colección.

Gestión de bases de datos relacionales:

- Establecimiento de conexiones.
- Recuperación de información.
- Manipulación de la información.
- Ejecución de consultas sobre la base de datos.

3.2 Contenidos actitudinales

Los contenidos actitudinales serán comunes a todas las unidades, estos son:

- Aceptación de las normas de comportamiento y trabajo establecidas.
- Participación activa en los debates y en la formación de grupos de trabajos.
- Presentación limpia/ordenada en los trabajos y pruebas realizados.
- Respeto por las opiniones, ideas y conductas de los compañeros y compañeras.
- Iniciativa personal.
- Orden.
- Limpieza.

- Trabajo en grupo
- Creatividad.
- Análisis.
- Autoaprendizaje.
- Capacidad para organizar y planificar

3.3 Distribución temporal

La temporalización puede verse modificada en virtud del seguimiento de la materia por parte de los alumnos y de la práctica docente en sí, no obstante, y de forma aproximada se recoge a continuación:

1ª Evaluación

U.D.0.- Presentación del módulo (1 hora)

U.D.1.- Identificación de los elementos de un programa informático (20 horas)

U.D.2.- Uso de estructuras de control (30 horas)

U.D.3.- Utilización de objetos. (30 horas)

2ª Evaluación

U.D.4.- Desarrollo de clases (45 horas)

U.D.5.- Aplicación de las estructuras de almacenamiento (45 horas)

3ª Evaluación

U.D.6.- Utilización avanzada de clases (40 horas)

U.D.7.- Lectura y escritura de información (10 horas)

U.D.8.- Mantenimiento de la persistencia de los objetos (10 horas)

U.D.9.- Gestión de base de datos relacionales (14 horas)

4. UNIDADES DIDÁCTICAS

U.D.0.- Presentación del módulo
Objetivos a alcanzar
El objetivo de esta unidad es dar al alumno una visión amplia de los contenidos del módulo así como familiarizarse con su equipo de trabajo y sus compañeros.
Contenidos a desarrollar
<ul style="list-style-type: none">• Planificación global de desarrollo del módulo, así como conocer a los miembros del grupo.• Criterios que serán considerados y aplicados por el profesor en la gestión del proceso formativo.• Derechos y obligaciones como estudiante, en relación con el módulo.• Principales interrelaciones que se dan entre las unidades didácticas del módulo y entre este y los demás que lo constituyen.• Identificación de los propios conocimientos en relación con los que se deben alcanzar en el módulo
Recursos a utilizar
Los materiales y recursos didácticos descritos en la sección 5.1.
Actividades de enseñanza-aprendizaje y de evaluación
<ul style="list-style-type: none">• Presentación de alumnos y profesor.• Presentación de los elementos que componen la programación• Presentación de los criterios y normas que guiarán la gestión del proceso formativo.

- Identificación de los conocimientos previos de los alumnos y de las alumnas en relación con el módulo profesional a cursar.

U.D.1.- Identificación de los elementos de un programa informático	
Objetivos a alcanzar	
El objetivo de esta unidad didáctica es entender los elementos básicos que conforman un programa informático. Los cuales van desde los diferentes tipos de variables, como la utilización de IDEs para la generación de código fuente.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> Estructura y bloques fundamentales. Proyectos. Utilización de los entornos integrados de desarrollo. Variables. Tipos de datos: 	<ul style="list-style-type: none"> Literales: Constantes. Operadores y expresiones. Conversiones de tipo. Comentarios.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> Se han identificado los bloques que componen la estructura de un programa informático. Se han creado proyectos de desarrollo de aplicaciones. Se han utilizado entornos integrados de desarrollo. Se han identificado los distintos tipos de variables y la utilidad específica de cada uno. Se ha modificado el código de un programa para crear y utilizar variables. Se han creado y utilizado constantes y literales. 	

- Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- Se ha comprobado el funcionamiento de las conversiones de tipo explícita e implícita.
- Se han introducido comentarios en el código.

U.D.2.- Uso de estructuras de control

Objetivos a alcanzar

El objetivo de esta unidad didáctica es mostrar al alumno las diferentes estructuras de control que nos ofrecen los lenguajes de programación. Estructuras de repetición, selección, salto, etc.

Contenidos a desarrollar

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.
- Control de excepciones.
- Prueba y depuración.
- Documentación.

Recursos a utilizar

Los materiales y recursos didácticos descritos en la sección 5.1.

Actividades de enseñanza-aprendizaje y de evaluación

- Se ha escrito y probado código que haga uso de estructuras de selección.
- Se han utilizado estructuras de repetición.
- Se han reconocido las posibilidades de las sentencias de salto.
- Se ha escrito código utilizando control de excepciones.
- Se han creado programas ejecutables utilizando diferentes estructuras de control.
- Se han probado y depurado los programas.

- Se ha comentado y documentado el código.

U.D.3.- Utilización de objetos.	
Objetivos a alcanzar	
El objetivo de esta unidad es que el alumno comprenda la utilización de los objetos, así como sus propiedades y métodos.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> • Características de los objetos. • Instanciación de objetos. • Utilización de métodos. • Utilización de propiedades. • Modo consola: entrada y salida de información. 	<ul style="list-style-type: none"> • Utilización de métodos estáticos. • Parámetros y valores devueltos. • Librerías de objetos. • Constructores. • Destrucción de objetos y liberación de memoria.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> • Se han identificado los fundamentos de la programación orientada a objetos. • Se han escrito programas simples. • Se han instanciado objetos a partir de clases predefinidas. • Se han utilizado métodos y propiedades de los objetos. • Se han escrito llamadas a métodos estáticos. • Se han utilizado parámetros en la llamada a métodos. • Se han incorporado y utilizado librerías de objetos. 	

- Se han utilizado constructores.
- Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

U.D.4.- Desarrollo de clases	
Objetivos a alcanzar	
El objetivo de esta unidad didáctica es entender la programación orientada a objetos a través del desarrollo de clases.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> • Concepto de clase. • Estructura y miembros de una clase. • Creación de atributos. • Creación de métodos. • Creación de constructores. 	<ul style="list-style-type: none"> • Encapsulación y visibilidad. • Utilización de clases y objetos. • Utilización de clases heredadas: • Empaquetados de clases.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> • Se ha reconocido la sintaxis, estructura y componentes típicos de una clase. • Se han definido clases. • Se han definido propiedades y métodos. • Se han creado constructores. • Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente. • Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros. 	

- Se han definido y utilizado clases heredadas.
- Se han creado y utilizado métodos estáticos.
- Se han definido y utilizado interfaces.
- Se han creado y utilizado conjuntos y librerías de clases.

U.D.5.- Utilización avanzada de clases	
Objetivos a alcanzar	
En esta unidad didáctica se profundizará en conceptos de la programación orientada a objetos, como es la herencia, la sobrescritura, etc.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> Composición de clases. Herencia. Superclases y subclases. Clases y métodos abstractos y finales. 	<ul style="list-style-type: none"> Sobrescritura de métodos. Constructores y herencia. Acceso a métodos de la superclase. Polimorfismo.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> Se han identificado los conceptos de herencia, superclase y subclase. Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos. Se ha reconocido la incidencia de los constructores en la herencia. Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase. Se han diseñado y aplicado jerarquías de clases. Se han probado y depurado las jerarquías de clases. Se han realizado programas que implementen y utilicen jerarquías de clases. Se ha comentado y documentado el código. 	

U.D.6.- Aplicación de las estructuras de almacenamiento

Objetivos a alcanzar

En esta unidad didáctica se introducirán diferentes estructuras de almacenamiento de información ampliamente utilizadas en programación.

Contenidos a desarrollar

- | | |
|---|---|
| <ul style="list-style-type: none">• Estructuras.• Creación de arrays.• Inicialización.• Arrays multidimensionales. | <ul style="list-style-type: none">• Cadenas de caracteres.• Listas.• Colecciones. |
|---|---|

Recursos a utilizar

Los materiales y recursos didácticos descritos en la sección 5.1.

Actividades de enseñanza-aprendizaje y de evaluación

- Se han escrito programas que utilicen arrays.
- Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
- Se han utilizado listas para almacenar y procesar información.
- Se han utilizado iteradores para recorrer los elementos de las listas.
- Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.
- Se han creado clases y métodos genéricos.
- Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.
- Se han identificado las clases relacionadas con el tratamiento de documentos XML.

- Se han realizado programas que realicen manipulaciones sobre documentos XML.

U.D.7.- Lectura y escritura de información	
Objetivos a alcanzar	
En esta unidad didáctica se abordarán diferentes técnicas para la persistencia y lectura de información a través de la programación.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> • Concepto de flujo. • Tipos de flujos. Flujos de bytes y de caracteres. • Flujos predefinidos. • Clases relativas a flujos. • Utilización de flujos. • Entrada desde teclado. • Salida a pantalla. • Aplicaciones del almacenamiento de información en ficheros. • Ficheros de datos. Registros. 	<ul style="list-style-type: none"> • Escritura y lectura de información de ficheros. • Utilización de los sistemas de ficheros. • Creación y eliminación de ficheros y directorios. • Creación de interfaces gráficas de usuario utilizando asistentes y herramientas del entorno integrado. • Interfaces. • Concepto de evento. • Creación de controladores de eventos. • Apertura y cierre de ficheros. Modos de acceso.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> • Se ha utilizado la consola para realizar operaciones de entrada y salida de información. 	

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

U.D.8.- Mantenimiento de la persistencia de los objetos	
Objetivos a alcanzar	
En esta unidad didáctica se profundizará en la noción de lectura y persistencia de información utilizando gestores de bases de datos de forma programática.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> Bases de datos orientadas a objetos. Características de las bases de datos orientadas a objetos. Instalación del gestor de bases de datos. Creación de bases de datos. Tipos de datos básicos y estructurados. El lenguaje de definición de objetos. 	<ul style="list-style-type: none"> Mecanismos de consulta. El lenguaje de consultas: sintaxis, expresiones, operadores. Recuperación, modificación y borrado de información. Tipos de datos objeto; atributos y métodos. Herencia. Constructores. Tipos de datos colección.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> Se han identificado las características de las bases de datos orientadas a objetos. Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos. Se han instalado sistemas gestores de bases de datos orientados a objetos. 	

- Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
- Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

U.D.9.- Gestión de base de datos relacionales	
Objetivos a alcanzar	
En esta unidad didáctica se hará hincapié en la integración de bases de datos con la lógica de negocio de las aplicaciones.	
Contenidos a desarrollar	
<ul style="list-style-type: none"> • Establecimiento de conexiones. • Recuperación de información. • Utilización de asistentes. 	<ul style="list-style-type: none"> • Manipulación de la información. • Mecanismos de actualización de la base de datos. • Ejecución de consultas sobre la base de datos.
Recursos a utilizar	
Los materiales y recursos didácticos descritos en la sección 5.1.	
Actividades de enseñanza-aprendizaje y de evaluación	
<ul style="list-style-type: none"> • Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales. • Se han programado conexiones con bases de datos. • Se ha escrito código para almacenar información en bases de datos. • Se han creado programas para recuperar y mostrar información almacenada en bases de datos. • Se han efectuado borrados y modificaciones sobre la información almacenada. • Se han creado aplicaciones que ejecuten consultas sobre bases de datos. 	

- Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

5. METODOLOGÍA

La naturaleza de este módulo es práctica, por lo que se busca fomentar la autonomía del alumno, planteándole retos a superar con un mínimo de indicaciones, pero aportándole la documentación necesaria para que busque e investigue lo necesario para resolverlos. Esto tiene como objetivo enseñarles a afrontar los retos que les puedan surgir en su vida laboral de forma autosuficiente y eficiente.

Debido a que el alumnado puede orientar las actividades desde distintos puntos de vista y ser concebidos de diferente forma, se buscará fomentar el debate del grupo como estrategia de aprendizaje. El debate, no solo servirá para aclarar conceptos y llegar a un consenso, sino para favorecer el trabajo en grupo.

5.1 Materiales y recursos didácticos

Libro:

- *Programación orientada a objetos con Java: una introducción práctica usando BlueJ.* Barnes, David J. y Kolling, Michael. Pearson Education.
- *Programación.* Jimenez Marín, Alfonso y Pérez Montes, Francisco Manuel. Paraninfo.

Plataforma virtual

- <https://aulavirtual-educacion.larioja.org>

6. EVALUACIÓN

6.1 Criterios de evaluación

Los criterios de evaluación asociados a cada resultado de aprendizaje son:

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

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

- Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
 - a) Se ha escrito y probado código que haga uso de estructuras de selección.
 - b) Se han utilizado estructuras de repetición.

- c) Se han reconocido las posibilidades de las sentencias de salto.
 - d) Se ha escrito código utilizando control de excepciones.
 - e) Se han creado programas ejecutables utilizando diferentes estructuras de control.
 - f) Se han probado y depurado los programas.
 - g) Se ha comentado y documentado el código.
- Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
- a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
 - b) Se han definido clases.
 - c) Se han definido propiedades y métodos.
 - d) Se han creado constructores.
 - e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
 - f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
 - g) Se han definido y utilizado clases heredadas.
 - h) Se han creado y utilizado métodos estáticos.
 - i) Se han definido y utilizado interfaces.
 - j) Se han creado y utilizado conjuntos y librerías de clases.
- Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
 - b) Se han aplicado formatos en la visualización de la información.
 - c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.
 - d) Se han utilizado ficheros para almacenar y recuperar información.
 - e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
 - f) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficas de usuario simples.
 - g) Se han programado controladores de eventos.

- h) Se han escrito programas que utilicen interfaces gráficas para la entrada y salida de información.
- Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.
 - a) Se han escrito programas que utilicen arrays.
 - b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
 - c) Se han utilizado listas para almacenar y procesar información.
 - d) Se han utilizado iteradores para recorrer los elementos de las listas.
 - e) Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.
 - f) Se han creado clases y métodos genéricos.
 - g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.
 - h) Se han identificado las clases relacionadas con el tratamiento de documentos XML.
 - i) Se han realizado programas que realicen manipulaciones sobre documentos XML.
- Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
 - a) Se han identificado los conceptos de herencia, superclase y subclase.
 - b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
 - c) Se ha reconocido la incidencia de los constructores en la herencia. Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.
 - d) Se han diseñado y aplicado jerarquías de clases.
 - e) Se han probado y depurado las jerarquías de clases.
 - f) Se han realizado programas que implementen y utilicen jerarquías de clases.
 - g) Se ha comentado y documentado el código.
- Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
 - a) Se han identificado las características de las bases de datos orientadas a objetos.
 - b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.

- c) Se han instalado sistemas gestores de bases de datos orientados a objetos.
 - d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
 - e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
 - f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
 - g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
 - h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.
- Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.
- a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
 - b) Se han programado conexiones con bases de datos.
 - c) Se ha escrito código para almacenar información en bases de datos.
 - d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
 - e) Se han efectuado borrados y modificaciones sobre la información almacenada.
 - f) Se han creado aplicaciones que ejecuten consultas sobre bases de datos.
 - g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

6.2 Instrumentos y procedimientos de evaluación

La evaluación del alumno se realizará considerando los resultados de la evaluación continua en las diferentes actividades de evaluación realizadas. Concretamente, por cada una de las unidades didácticas citadas anteriormente habrá por lo menos una actividad evaluable. Además, habrá un examen (teórico y/o práctico) por cada una de las evaluaciones.

Los alumnos tienen dos formas de obtener calificación en el módulo:

- A lo largo del curso, se evaluará de forma continua, asignando a cada instrumento de evaluación un porcentaje para la calificación final.

- Recuperación. Se realizarán recuperaciones en las evaluaciones ordinarias (primera y segunda) para aquellos alumnos que no han superado el curso o que han perdido el derecho de evaluación continua. En estas evaluaciones, el alumnado tendrá que examinarse del contenido completo del módulo.

6.3 Criterios de calificación

La calificación del alumno se realizará considerando los resultados de la evaluación continua en las diferentes actividades de evaluación realizadas como exámenes, ejercicios, prácticas o proyectos.

La **calificación de cada evaluación** se realizará con la siguiente cuantificación:

Exámenes	75 %
Actividades	25 %

Para obtener la calificación de cada evaluación se realizará la media ponderada de los resultados obtenidos en cada uno de los exámenes y prácticas realizadas hasta el momento de la evaluación. La nota de evaluación resultará del truncamiento de esta media ponderada

De acuerdo con la normativa vigente la calificación del módulo profesional es numérica entre 1 y 10, sin decimales. Se considerarán positivas las calificaciones iguales o superiores a cinco puntos y negativas las restantes.

Serán calificados con un 0 aquellos exámenes y/o pruebas en los que se detecte que ha habido copia entre diferentes compañeros y/o se hayan utilizado medios no autorizados por el profesor.

La siguiente tabla muestra **la distribución temporal de las unidades** y el peso de los **exámenes** programados a lo largo del curso.

EVALUACIÓN	UNIDADES DIDÁCTICAS	HORAS	EXÁMENES PROGRAMADOS	PESO SOBRE EL TOTAL DE EXÁMENES	PESO SOBRE LA NOTA FINAL
1ª	UD0	1	Examen evaluación 1ª	20%	15%
	UD1	20			
	UD2	30			
	UD3	30			
2ª	UD4	45	Examen evaluación 2ª	20%	15%
	UD5	45			
3ª	UD6	40	Examen evaluación 3ª	60%	45%
	UD7	10			
	UD8	10			
	UD9	14			
TOTAL		245		100%	75%

La siguiente tabla muestra **el peso de las prácticas** realizadas a lo largo del curso.

EJERCICIOS PRÁCTICAS	Y	PESO SOBRE EL TOTAL DE PRÁCTICAS	PESO SOBRE LA NOTA FINAL
UD0		0%	0%
UD1		7%	1.75%
UD2		7%	1.75%
UD3		7%	1.75%
UD4		10%	2.5%
UD5		10%	2.5%
UD6		14%	3.5%
UD7		15%	3.75%
UD8		15%	3.75%
UD9		15%	3.75%
TOTAL		100 %	25 %

Si en una unidad didáctica hay más de una práctica se realizará la media aritmética entre todas las prácticas realizadas en cada unidad.

Otros aspectos que considerar sobre las prácticas son:

- Las prácticas entregadas fuera de plazo se calificarán con un 0.
- Aquellos proyectos, trabajos, prácticas o ejercicios de diferentes alumnos en los que haya una manifiesta similitud entre ellos o con otros de años anteriores o de Internet serán calificados con un 0.
- Podrá requerirse la exposición oral de las prácticas por parte de los alumnos.

6.4 Criterios de recuperación

El profesor debe facilitar el éxito de sus alumnos, por lo que deben establecerse unos criterios para permitir la recuperación de las partes no superadas. Al ser la evaluación individualizada, las actividades de recuperación podrán variar en función del alumno y de los conocimientos y capacidades que sean objeto de recuperación.

Los procedimientos de recuperación son detección de las carencias del alumno, realización de tareas específicas que refuercen la carencia detectada y reevaluación de los conocimientos y/o capacidades no superadas.

La reevaluación de los conocimientos y/o capacidades no superadas se realizará:

- En las **evaluaciones ordinarias** (primera y segunda), que se llevarán a cabo en el mes de junio y serán realizadas por:
 - **Alumnos que no hayan superado la materia a lo largo del curso.**

Los alumnos realizarán un examen de todos los contenidos del módulo. La nota final del módulo se calculará con los porcentajes establecidos en los criterios de calificación.

CALIFICACIÓN	1ª ORDINARIA	2ª ORDINARIA
Exámenes	100%	100%

- **Alumnos que han perdido el derecho a evaluación continua o que no hayan superado las actividades.** Se llevará a cabo la evaluación de toda la materia impartida en la asignatura en un examen una vez acabada la docencia. En este tipo de evaluación, la nota de los exámenes corresponderá con la calificación obtenida en el examen final. Este examen podrá tener contenidos de las prácticas realizadas durante el curso.

CALIFICACIÓN	1ª ORDINARIA	2ª ORDINARIA
Examen	100%	100%

6.5 Actividades de refuerzo o recuperación

El examen final ordinario valorará los conocimientos del alumnado, sin embargo, se facilitará un conjunto de tareas al alumnado para la preparación de los mismos en caso de haber suspendido el módulo.

Estas tareas no serán evaluadas ni obligatorias, pero servirán para indicar al alumnado los contenidos que se abordarán en estos exámenes finales.

6.6 Recuperación de módulos pendientes

Para la recuperación de módulos pendientes se realizará un plan de recuperación específico para los alumnos con las actividades y prácticas que el profesor considere necesario con el fin de que pueda aprobar un examen teórico-práctico con el contenido del módulo en la que debe alcanzarse un mínimo de 5 para aprobar.

7. ATENCIÓN AL ALUMNADO CON NECESIDADES ESPECÍFICAS DE APOYO EDUCATIVO

Esta etapa educativa debe atender las necesidades educativas de los alumnos y alumnas, tanto de los que requieren un refuerzo porque presentan ciertas dificultades en el aprendizaje como de aquellos cuyo nivel esté por encima del habitual.

Escalonar el acceso al conocimiento y graduar los aprendizajes constituye un medio para lograr responder a la diversidad del alumnado, de manera que se puedan valorar progresos parciales. Representa también un factor importante el hecho de que los alumnos y alumnas sepan qué es lo que se espera de ellos.

De los objetivos generales del módulo, se tendrá en cuenta que, la adquisición de las capacidades presentará diversos grados, en función de esta diversidad del alumnado.

Por último, será el profesor o profesora el que adopte la decisión de que objetivos, contenidos, metodología, actividades, instrumentos y criterios de evaluación adaptará según las características del alumnado de los grupos que imparta. Se podrá contemplar de la siguiente forma:

Desarrollando cuestiones de diagnóstico previo, al inicio de cada unidad didáctica, para detectar el nivel de conocimientos y de motivación del alumnado que permita valorar al profesor el punto de partida y las estrategias que se van a seguir. Conocer el nivel del que partimos nos permitirá saber qué alumnos y alumnas requieren unos conocimientos previos antes de comenzar la unidad, de modo que puedan abarcarla sin dificultades.

Incluyendo actividades de diferente grado de dificultad, bien sean de contenidos mínimos, de ampliación o de refuerzo o profundización, permitiendo que el profesor seleccione las más oportunas atendiendo a las capacidades y al interés de los alumnos y alumnas.

8. ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES

Se fomentará la visita al centro de profesionales que trabajen en el sector informático y que puedan contar su experiencia dentro del desarrollo software. Concretamente se buscará aportar experiencia dentro del uso de metodologías ágiles para el desarrollo de aplicaciones.

Fomento de la lectura

Se buscará el fomentar la lectura de información técnica relacionada con la informática. Para ello se motivará al alumnado a leer artículos de actualidad. También, se procurará acceder a la documentación técnica y oficial de las herramientas utilizadas, cuyo entendimiento y desarrollo forma parte de los criterios de evaluación del módulo.

Fomento de la participación en iniciativas de diferentes organizaciones

- *INCIBE*: ofrece una cartera amplísima de cursos formativos en materia de ciberseguridad. Además, ofrece numerosas competiciones tanto para alumnos de FP como para alumnos Universitarios. Por ejemplo, es colaborador de la “National Cyberleag GC”. También destaca iniciativas recientes como la competición “Academia Hacker” donde los equipos inscritos resuelven retos de tipología Captura La Bandera.
- *Ministerio de Educación y Formación Profesional (MEFP)*: entre los campeonatos con mayor repercusión social que organiza el MEFP están las SpainSkills. Esta competición nacional no solo es en el ámbito de la informática, sino que cubre todas las familias de la Formación Profesional. A su vez, La Comunidad Autónoma de La Rioja celebra pruebas regionales para escoger a sus alumnos que acudirán a la fase nacional. En la web <http://rojaskills.es/> puede encontrarse toda la información sobre esta competición.
- *Facultad de Informática de la UCM*: una de las competiciones de más éxito dentro de la familia de informática en Formación Profesional es el concurso ProgramaMe. En dicho concurso, equipos de alumnos de Formación Profesional de la familia de informática abordan diferentes retos de programación.

9. NECESIDADES Y PROPUESTAS DE FORMACIÓN DEL PROFESORADO

Las necesidades de formación del profesorado, en este curso pasan por realizar algún curso de actualización, sobre todo en versiones más actuales del lenguaje Java. También sería recomendable que los docentes pudiéramos realizar algún curso de formación sobre el diseño de interfaces mediante JavaFX.

10. BIBLIOGRAFÍA DE REFERENCIA

Manual:

- El tutorial de BlueJ <http://www.bluej.org/doc/tutorial.html>
- *Programación*. Jimenez Marín, Alfonso y Pérez Montes, Francisco Manuel. Paraninfo.