

**DEPARTAMENTO DE INFORMÁTICA Y
COMUNICACIONES.**

PROGRAMACIÓN DEL MÓDULO:

Programación

**Curso: 1º C.F.G.S
Desarrollo de Aplicaciones Multiplataforma**

(2024/2025)



1. JUSTIFICACIÓN.....	2
2. CONTEXTUALIZACIÓN.....	2
3. COMPETENCIAS DEL TÍTULO.....	2
4. CONTENIDOS.....	2
4.1 BLOQUES DE CONTENIDOS.....	2
4.2 UNIDADES DE TRABAJO.....	2
<i>UT1.- Conceptos básicos.....</i>	<i>4</i>
<i>UT2.- Condicionales.....</i>	<i>4</i>
<i>UT3.- Bucles.....</i>	<i>5</i>
<i>UT4.- Funciones.....</i>	<i>5</i>
<i>UT5.- Tablas.....</i>	<i>6</i>
<i>UT6.- Cadenas de caracteres.....</i>	<i>6</i>
<i>UT7.- Clases.....</i>	<i>7</i>
<i>UT8.- Herencia.....</i>	<i>7</i>
<i>UT9.- Interfaces.....</i>	<i>8</i>
<i>UT10.- Ficheros de texto.....</i>	<i>8</i>
<i>UT11.- Ficheros binarios.....</i>	<i>9</i>
<i>UT12.- Colecciones.....</i>	<i>9</i>
<i>UT13.- Conexión a bases de datos JDBC.....</i>	<i>10</i>
<i>UT14.- Bases de datos orientadas a Objetos.....</i>	<i>11</i>
5. OBJETIVOS ESPECÍFICOS DEL MÓDULO.....	11
6. METODOLOGÍA.....	12
6.1 PRINCIPIOS.....	12
6.2 AGRUPAMIENTOS.....	12
6.3 TIPOS DE ACTIVIDADES.....	12
6.4 ORGANIZACIÓN DEL AULA.....	12
6.5 ORGANIZACIÓN DEL TIEMPO.....	13
6.6 MATERIALES.....	14
6.7 USO DE LAS TIC.....	14
7. EVALUACIÓN Y PROMOCIÓN.....	14
7.1 RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN.....	14
7.2 PROMOCIÓN.....	18
7.3 CRITERIOS DE CALIFICACIÓN. PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN.....	18
7.3.1 CRITERIOS DE CALIFICACIÓN.....	18
7.3.2 INSTRUMENTOS DE EVALUACIÓN.....	19
7.3.3 PROCEDIMIENTOS DE EVALUACIÓN.....	20
7.4 CRITERIOS DE RECUPERACIÓN.....	20
7.5 INSTRUMENTOS DE EVALUACIÓN DE LA PROGRAMACIÓN Y DE LA PRÁCTICA DOCENTE.....	21
8. PROPUESTAS DE ATENCIÓN A LA DIVERSIDAD.....	21
9. CONCRECIÓN DE LOS PLANES Y PROGRAMACIÓN DE CONTENIDO PEDAGÓGICO DEL CENTRO.....	21
10. ACTIVIDADES EXTRAESCOLARES.....	21
10.1 ACTIVIDADES PARA ALUMNOS QUE SUPEREN LA PRIMERA EVALUACIÓN ORDINARIA.....	21
11. BIBLIOGRAFÍA.....	21

1. JUSTIFICACIÓN.

Desarrollado en la PGA del centro

2. CONTEXTUALIZACIÓN.

Desarrollado en la PGA del centro y en la Programación Didáctica del Departamento (PDD).

3. COMPETENCIAS DEL TÍTULO.

Las competencias profesionales, personales y sociales de este título están relacionadas en la PDD.

La formación del módulo de **Programación** contribuye a alcanzar los objetivos generales e), j), q) y w) del ciclo formativo y las competencias e), j), t) y w) del título.

4. CONTENIDOS.

Los contenidos son los elementos que permitirán alcanzar los objetivos y competencias indicados en el apartado anterior.

4.1 Bloques de contenidos

- Identificación de los elementos de un programa informático.
- Utilización de objetos.
- Uso de estructuras de control.
- Desarrollo de clases.
- Lectura y escritura de información.
- Aplicación de las estructuras de almacenamiento.
- Utilización avanzada de clases.
- Mantenimiento de la persistencia de los objetos.
- Gestión de bases de datos.

4.2 Unidades de Trabajo

Teniendo en cuenta los contenidos que deben desarrollarse a lo largo del módulo, seguidamente se establecen las unidades de trabajo:

UNIDADES DE TRABAJO	RESULTADOS DE APRENDIZAJE ASOCIADO (RA)
U.T.1. CONCEPTOS BÁSICOS	RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
U.T.2. CONDICIONALES	RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
U.T.3. BUCLES	
U.T.4. FUNCIONES	RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
U.T.5. TABLAS.	RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
U.T.6. CADENAS DE CARACTERES	
U.T.7. CLASES	RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
U.T.8. HERENCIA	RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
U.T.9. INTERFACES	
U.T.10. FICHEROS DE TEXTO	RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases. (BÁSICO)
U.T.11. FICHEROS BINARIOS.	
U.T.12. COLECCIONES	RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos. (BÁSICO)
U.T.13. BASES DE DATOS RELACIONALES.	RA9. Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.
U.T.14. BASES DE DATOS ORIENTADAS A OBJETOS.	RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

UT1.- Conceptos básicos.

Objetivos

- Identificar los bloques que componen la estructura de un programa informático.
- Crear proyectos de desarrollo de aplicaciones
- Utilizar entornos integrados de desarrollo.
- Identificar los distintos tipos de variables y la utilidad específica de cada uno.
- Modificar el código de un programa para crear y utilizar variables.
- Crear y utilizar constantes y literales.
- Clasificar, reconocer y utilizar en expresiones los operadores del lenguaje.
- Comprobar el funcionamiento de las conversiones de tipo explícitas e implícitas.
- Introducir comentarios en el código.

Contenidos:

- Algoritmo.
- Lenguajes de programación.
- Programa principal.
- Palabras reservadas.
- Concepto de variable. Identificadores.
- Tipos primitivos.
- Constantes.
- Comentarios.
- API de Java.
- Operaciones básicas. Operadores.
- Conversiones de tipos.

Resultados de aprendizaje

- **RA1.** Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

UT2.- Condicionales.

Objetivos

- Escribir y probar código que haga uso de estructuras de selección.
- Reconocer las posibilidades de las sentencias de salto.
- Escribir código utilizando control de excepciones.
- Crear programas ejecutables utilizando diferentes estructuras de selección.
- Probar y depurar los programas.
- Comentar y documentar el código.

Contenidos:

- Expresiones lógicas.
- Condicional simple: if.
- Condicional doble: if-else.
- Operador ternario.
- Anidación de condicionales.
- Condicional múltiple: switch.

Resultados de aprendizaje

- **RA3.** Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

UT3.- Bucles.

Objetivos

- Escribir y probar código que haga uso de estructuras de repetición.
- Escribir código utilizando control de excepciones.
- Crear programas ejecutables utilizando diferentes bucles.
- Probar y depurar los programas.
- Comentar y documentar el código.

Contenidos:

- Bucle while.
- Bucle do-while.
- Bucles for.
- Salidas anticipadas. Break y continue.
- Bucles anidados.

Resultados de aprendizaje

- **RA3.** Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

UT4.- Funciones.

Objetivos

- Asimilar el concepto de función o método.
- Usar parámetros de entrada a la función.
- Escribir programas que usen correctamente funciones que reciban y devuelvan valores.
- Usar adecuadamente el valor devuelto por la función.
- Usar funciones recursivas sencillas.

Contenidos

- Conceptos básicos.

- Ámbito de las variables.
- Paso de información a una función.
- Valor devuelto por una función.
- Sobrecarga de funciones.
- Recursividad.

Resultados de aprendizaje

- **RA1.** Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

UT5.- Tablas.

Objetivos

- Identificar los fundamentos de la programación orientada a objetos.
- Conocer las tablas (o arrays) de distinto tipo y longitud.
- Escribir programas que hagan uso de tablas.
- Conocer y Usar la API de Java relacionada con las tablas.
- Utilizar el entorno integrado de desarrollo en la creación y compilación de programas que usen tablas.

Contenidos

- Variables escalares versus tablas.
- Indices.
- Construcción de tablas.
- Uso de tablas.
- Tablas como parámetros en las funciones.
- Operaciones con tablas: la clase Arrays.
- Tablas n-dimensionales.

Resultados de aprendizaje

- **RA2.** Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

UT6.- Cadenas de caracteres.

Objetivos

- Utilizar el tipo primitivo char.
- Conocer la clase Character.
- Conocer la clase String
- Instanciar objetos a partir de las clases predefinidas.
- Utilizar métodos y propiedades de los objetos.
- Escribir llamadas a métodos estáticos.
- Utilizar parámetros en la llamada a métodos.
- Conocer y Usar la API de Java relacionada con las clases predefinidas anteriores.

Contenidos

- Tipo primitivo char.

- Clase Character.
- Clase String.
- Cadenas y tablas de caracteres.

Resultados de aprendizaje

- **RA2.** Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

UT7.- Clases.

Objetivos

- Reconocer la sintaxis, estructura y componentes típicos de una clase.
- Definir clases.
- Definir propiedades y métodos.
- Crear constructores.
- Desarrollar programas que instancien y utilicen objetos de las clases creadas anteriormente.
- Utilizar mecanismos para controlar la visibilidad de las clases y de sus miembros.

Contenidos

- Definición de una clase.
- Crear clases en el entorno integrado.
- Atributos.
- Objetos.
- Métodos.
- Atributos y métodos estáticos.
- Constructores.
- Paquetes.
- Modificadores de acceso.
- Enumerados.

Resultados de aprendizaje

- **RA4.** Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

UT8. Herencia.

Objetivos

- Identificar los conceptos de herencia, superclase, subclase y polimorfismo.
- Utilizar modificadores para bloquear y forzar la herencia de clases y métodos.
- Reconocer la incidencia de los constructores en la herencia.
- Crear clases heredadas que sobrescriban la implementación de métodos de la superclase.
- Diseñar y aplicar jerarquías de clases.
- Probar y depurar las jerarquías de clases.

- Realizar programas que implementen y utilicen jerarquías de clases.
- Comentar y documentar el código.

Contenidos

- Subclase y Superclase.
- Modificador de acceso para herencia.
- Redefinición de miembros heredados.
- La clase Object.
- Clases abstractas.

Resultados de aprendizaje

- **RA7.** Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

UT9. Interfaces.

Objetivos

- Conocer la necesidad de las interfaces y distinguirlas de las clases abstractas.
- Definir métodos abstractos en una interfaz.
- Implementar una o más interfaces en una clase.
- Diseñar interfaces para operaciones específicas en clases diversas.

Contenidos

- Concepto de interfaz.
- Sintaxis general.
- Atributos de una interfaz.
- Métodos implementados en una interfaz.
- Herencia.
- Variables de tipo interfaz.
- Clases anónimas.
- Algunas interfaces de la API de Java

Resultados de aprendizaje

- **RA7.** Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

UT10.- Ficheros de texto.

Objetivos

- Conocer el concepto de excepción y sus tipos.
- Manipular excepciones.
- Saber el concepto de flujo de entrada y de salida.
- Crear flujos de entrada de texto.

- Conocer y aplicar funciones de lectura en flujos de entrada de texto.
- Crear flujos de salida de texto.
- Conocer y aplicar funciones de escritura en flujos de salida de texto.
- Cerrar flujos de entrada y de salida.
- Utilizar ficheros de texto para almacenar y recuperar información.
- Crear programas que utilicen diversos métodos de acceso al contenido de los ficheros

Contenidos

- Excepciones.
- Flujos de entrada de texto.
- Scanner y flujos de entrada.
- Flujos de salida de texto.

Resultados de aprendizaje

- **RA5.** Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

UT11.- Ficheros binarios.

Objetivos

- Utilizar ficheros binarios para almacenar y recuperar información.
- Escribir datos primitivos y objetos en ficheros binarios.
- Leer datos primitivos y objetos en ficheros binarios.
- Conocer las excepciones que se pueden producir durante la lectura y escritura del fichero.
- Identificar las excepciones que se pueden arrojar durante la apertura y cierre del fichero.
- Realizar operaciones de altas, bajas y modificaciones en ficheros binarios.

Contenidos

- Flujos de salida binarios.
- Flujos de entrada binarios.
- Ficheros de tipos primitivos.
- Ficheros de objetos.

Resultados de aprendizaje

- **RA5.** Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

UT12.- Colecciones.

Objetivos

- Crear clases y métodos genéricos.
- Conocer la interfaz Collection.
- Reconocer las librerías de clases relacionadas con tipos de datos avanzados.

- Utilizar listas para almacenar y procesar información.
- Conocer la interfaz List sus métodos e implementaciones.
- Conocer la interfaz Set sus métodos e implementaciones.
- Conocer la interfaz Map sus métodos e implementaciones.

Contenidos

- Tipos parametrizados o genéricos.
- Interfaz Collection.
- Interfaz List.
- Interfaz Set.
- Conversión entre colecciones.
- Clase Collections.
- Interfaz Map.

Resultados de aprendizaje

- **RA6.** Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

UT13.- Bases de datos relacionales.

Objetivos

- Identificar las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
- Programar conexiones con bases de datos.
- Escribir código para almacenar información en bases de datos.
- Crear programas para recuperar y mostrar información almacenada en bases de datos.
- Efectuar borrados y modificaciones sobre la información almacenada.
- Crear aplicaciones que ejecuten consultas sobre bases de datos.
- Crear aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

Contenidos

- API JDBC.
- Driver.
- Conexión.
- Ejecución de sentencias.
- Clase ResultSet.
- SQL Injection.
- Sentencias parametrizadas.
- Operaciones CRUD.
- Objeto de acceso a datos.

Resultados de aprendizaje

- **RA9.** Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.

UT14.- Bases de datos orientadas a Objetos.

Objetivos

- Identificar las características de las bases de datos orientadas a objetos.
- Analizar su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
- Instalar sistemas gestores de bases de datos orientados a objetos.
- Clasificar y analizar los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- Crear bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- Programar aplicaciones que almacenen objetos en las bases de datos creadas.
- Realizar programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- Realizar programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

Contenidos

- Características de las bases de datos orientadas a objetos.
- Creación de la base de datos.
- Almacenamiento de objetos.
- Recuperación y actualización de objetos.
- Eliminación de objetos.
- Consultas.

Resultados de aprendizaje

- **RA8.** Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

5. OBJETIVOS ESPECÍFICOS DEL MÓDULO.

Los **RESULTADOS DE APRENDIZAJE PARA EL MÓDULO de Programación** son:

1. **RA1** - Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
2. **RA2** - Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
3. **RA3** - Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
4. **RA4** - Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
5. **RA5** - Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
6. **RA6** - Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

7. **RA7** - Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
8. **RA8** - Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
9. **RA9** - Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.

Los OBJETIVOS A CUMPLIR, basados en los Resultados de Aprendizaje, **para el módulo de Programación son:**

1. Reconocer la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
2. Escribir y probar programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
3. Escribir y depurar código, analizando y utilizando las estructuras de control del lenguaje.
4. Desarrollar programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
5. Realizar operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
6. Escribir programas que manipulen información seleccionando y utilizando tipos avanzados de datos.
7. Desarrollar programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
8. Utilizar bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
9. Gestionar información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.

6. METODOLOGÍA

En este apartado se va a establecer como enseñar o como conseguir los objetivos marcados, utilizando los contenidos propios del módulo. Se detallará el método a seguir en el proceso de enseñanza-aprendizaje.

6.1 Principios

Desarrollado en el punto 6.1.1 Principios Presencial de la PDD.

6.2 Agrupamientos

Especificados en la PDD.

6.3 Tipos de actividades

Desarrollado en el punto 6.3.1 Tipos de actividades Presencial de la PDD.

6.4 Organización del aula

Especificados en la PDD.

6.5 Organización del tiempo

Estamos ante un módulo de **186 horas**. La temporalización aproximada es la siguiente:

UNIDADES DE TRABAJO	PERIODOS LECTIVOS	RESULTADOS DE APRENDIZAJE ASOCIADO (RA)
U.T.1. CONCEPTOS BÁSICOS	17	RA1
U.T.2. CONDICIONALES	14	RA3.
U.T.3. BUCLES	15	
U.T.4. FUNCIONES	14	RA1
U.T.5. TABLAS.	16	RA2.
U.T.6. CADENAS DE CARACTERES	18	
U.T.7. CLASES	22	RA4.
U.T.8. HERENCIA	12	RA7.
U.T.9. INTERFACES	3	
U.T.10. FICHEROS DE TEXTO	5	RA5.
U.T.11. FICHEROS BINARIOS.	12	
U.T.12. COLECCIONES	18	RA6.
U.T.13. BASES DE DATOS RELACIONALES.	10	RA9. FORMACIÓN EN EMPRESA
U.T.14. BASES DE DATOS ORIENTADAS A OBJETOS.	10	RA8. FORMACIÓN EN EMPRESA

Cuando haya problemas de tiempo las unidades de trabajo 13 y 14 (RA9 y RA8) podrán no impartirse ya que estas unidades se trabajan más ampliamente en el módulo de Acceso a Datos del segundo curso.

Las fechas de las evaluaciones se encuentran recogidas en el apartado 6.5.1 de la PDD.

6.6 Materiales

Desarrollado en el apartado 6.6.1 de la PDD.

Libro de texto tanto para el profesor como para el alumno:

Programación. Edición 2021

ISBN: 9788428342865

Autores: Alfonso Jiménez Marín, Francisco Manuel Pérez Montes

Editorial Paraninfo

6.7 Uso de las TIC

Referenciarse al correspondiente apartado de la PDD.

7. EVALUACIÓN Y PROMOCIÓN.

La evaluación será continua, formativa y sumativa, considerándose además de las pruebas objetivas (exámenes, tareas o trabajos), el trabajo en clase, el progreso, el interés por el módulo, la atención, etc.

7.1 Resultados de aprendizaje y Criterios de evaluación

Los **RESULTADOS DE APRENDIZAJE** (RRAA) son tratados como los objetivos propios del módulo y los **CRITERIOS DE EVALUACIÓN** (CCEE) concretan dichos resultados de aprendizaje y serán el elemento fundamental para poder evaluar el grado de consecución de los objetivos.

El grado de relevancia que queramos otorgar a cada RA se verá reflejado con la aplicación de distintos porcentajes en el cálculo de la nota final del módulo.

En la siguiente tabla se muestran:

- los **RESULTADOS DE APRENDIZAJE** (RRAA) del módulo con sus **CRITERIOS DE EVALUACIÓN** (CCEE) asociados,
- el instrumento para evaluar cada RA,
- el número de la unidad de trabajo donde se trabajará y evaluará
- el porcentaje de la nota que representa el RA con respecto de la nota final,
- los CCEE **BÁSICOS** que se utilizarán para impartir los **contenidos prioritarios o mínimos**.

Los RRAA son tratados como los objetivos propios del módulo y los CCEE concretan dichos resultados de aprendizaje y serán el elemento fundamental para poder evaluar el grado de consecución de los objetivos.

CATEGORÍA	RESULTADOS DE APRENDIZAJE (RRAA) (Criterios de evaluación - CCEE)	INSTRUMENTO DE EVALUACIÓN	UT donde se trabajará
BÁSICO: B O VACÍO T - TEÓRICO			
RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.		9%	
B / T	a) Se han identificado los bloques que componen la estructura de un programa informático.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT.1 UT.4
	b) Se han creado proyectos de desarrollo de aplicaciones		
	c) Se han utilizado entornos integrados de desarrollo.		
B / T	d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.		
B	e) Se ha modificado el código de un programa para crear y utilizar variables.		
B / T	f) Se han creado y utilizado constantes y literales.		
B / T	g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.		
T	h) Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.		
B/T	i) Se han introducido comentarios en el código.		
B / T	j) Se han creado y utilizado procedimientos y funciones.		
B / T	k) Se ha utilizado el paso de parámetros en procedimientos y funciones.		
RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.		16%	
B / T	a) Se han identificado los fundamentos de la programación orientada a objetos.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT.5 UT.6
B	b) Se han escrito programas simples.		
B	c) Se han instanciado objetos a partir de clases predefinidas.		
B	d) Se han utilizado métodos y propiedades de los objetos.		
B	e) Se han escrito llamadas a métodos estáticos.		
B	f) Se han utilizado parámetros en la llamada a métodos.		
B	g) Se han incorporado y utilizado librerías de objetos.		
B	h) Se han utilizado constructores y destructores de objetos.		
	i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.		
RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.		16%	
B	a) Se ha escrito y probado código que haga uso de estructuras de selección.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT.2 UT.3 UT.4
B / T	b) Se han utilizado estructuras de repetición.		
B	c) Se han reconocido las posibilidades de las sentencias de salto.		
B / T	d) Se ha escrito código utilizando control de excepciones.		
B	e) Se han creado programas ejecutables utilizando diferentes estructuras de control.		
	f) Se han probado y depurado los programas.		

B	g) Se ha comentado y documentado el código.		
	h) Se han creado excepciones.		
	i) Se han utilizado aserciones para la detección y corrección de errores durante la fase de desarrollo.		
RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.		15%	
B / T	a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT.7
B / T	b) Se han definido clases.		
B / T	c) Se han definido propiedades y métodos.		
B / T	d) Se han creado constructores.		
B	e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.		
B	f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.		
B / T	g) Se han definido y utilizado clases heredadas.		
	h) Se han creado y utilizado métodos estáticos.		
B	i) Se han creado y utilizado conjuntos y librerías de clases.		
RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.		12%	
B	a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT. 10 UT.11
B	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.		
B	d) Se han utilizado ficheros para almacenar y recuperar información.		
B	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 gráficos de usuario simples.		
	g) Se han programado controladores de eventos.		
	h) Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.		
RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos:		12%	
B	a) Se han escrito programas que utilicen arrays.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT. 12
T	b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.		
B	c) Se han utilizado listas para almacenar y procesar información.		
B	d) Se han utilizado iteradores para recorrer los elementos de las listas.		
B / T	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 escritos en diferentes lenguajes de intercambio de datos.		
	i) Se han realizado programas que realicen manipulaciones sobre documentos escritos en diferentes lenguajes de intercambio de datos.		
	j) Se han utilizado operaciones agregadas para el manejo de información almacenada en colecciones.		
RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.		10%	
B / T	a) Se han identificado los conceptos de herencia, superclase, subclase y polimorfismo.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT. 8 UT. 9
B	b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.		
T	c) Se ha reconocido la incidencia de los constructores en la herencia.		
B / T	d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.		
B / T	e) Se han diseñado y aplicado jerarquías de clases.		
	f) Se han probado y depurado las jerarquías de clases.		
B	g) Se han realizado programas que implementen y utilicen jerarquías de clases.		
	h) Se ha comentado y documentado el código.		
	i) Se han identificado y evaluado los escenarios de uso de interfaces.		
	j) Se han identificado y evaluado los escenarios de utilización de la herencia y la composición.		
RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.		5%	
B / T	a) Se han identificado las características de las bases de datos orientadas a objetos.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT. 13
B / T	b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.		
B	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.		
B	e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.		
B	f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.		
B	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.		
RA9. Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.		5%	
T	a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos.	Se realizará una o varias pruebas escritas a mano (en papel) o a ordenador para evaluar los criterios.	UT. 14
B	b) Se han programado conexiones con bases de datos.		
B	c) Se ha escrito código para almacenar información en bases de datos.		
B	d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.		

B	e) Se han efectuado borrados y modificaciones sobre la información almacenada.		
B	f) Se han creado aplicaciones que muestren la información almacenada en bases de datos.		
B	g) Se han creado aplicaciones para gestionar la información presente en bases de datos.		

Se han categorizado algunos Criterios de Evaluación como **BÁSICOS**. Esto permitirá mayor objetividad de cara a:

- **IMPARTIR EL CONTENIDO DE LAS UNIDADES** en base a ellos cuando por cuestiones de viabilidad temporal sea imposible evaluarlos todos.
- **REALIZAR LAS RECUPERACIONES**, cuando por cuestiones de viabilidad temporal sea imposible evaluarlos todos.

7.2 Promoción

Desarrollado en la PDD.

7.3 Criterios de calificación. Procedimientos e instrumentos de evaluación

7.3.1 Criterios de calificación

- Cada RA tendrá un peso en función de su relevancia.
- Cada RA se evaluará y calificará independientemente de los demás con un valor de 0 a 10, considerándose que el RA ha sido logrado si la calificación es mayor o igual a 5.
- Para calificar cada RA se usarán los instrumentos de evaluación indicados anteriormente.

La calificación de cada RA se muestra en la siguiente tabla:

RESULTADOS DE APRENDIZAJE	Porcentaje de la nota
1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado. (B)	9%
2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos. (B)	16%
3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje. (B)	16%
4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos. (B)	15%
5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases. (B)	12%
6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos. (B)	12%
7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación. (B)	10%
8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.	5%
9. Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.	5%

- Por cada RA se realizará una o varias pruebas. La nota será la nota media de las pruebas.
- Para que un RA se considere superado, el alumno deberá tener una calificación promedio ≥ 5 .
- Los RRAA etiquetados con una (B) (Básico) constituyen los mínimos que debe saber hacer, conocer y comprender el alumno.

7.3.2 Instrumentos de evaluación

Para evaluar cada Resultado de Aprendizaje (RA) se realizará una prueba escrita a mano en papel o a ordenador.

Para las **pruebas escritas a mano en papel**:

- El contenido puede ser teórico y práctico.
- El **contenido teórico** versará sobre los CCEE marcados como teóricos (T).
- El **contenido práctico** consistirá en resolver uno o varios problemas escribiendo código Java (o pseudocódigo) y versará sobre los CCEE de carácter práctico del RA a evaluar. Un problema está bien si hace lo que se pide.

Para las **pruebas escritas a ordenador**:

- Estas pruebas consistirán en la entrega de un ejercicio o varios escritos en Java, libre de errores de compilación. El ejercicio se entregará según el formato y fecha indicado y **NO** deberá tener errores de compilación. Un ejercicio está bien si hace lo que se pide.
- Se utilizará la plataforma **EducamosCLM** para la entrega de ejercicios. En el caso de que no esté operativa, los ejercicios prácticos se entregarán en un pendrive o se enviarán por e-mail al profesor.
- Estas pruebas se realizarán si se dispone de ordenadores suficientes.

7.3.3 Procedimientos de evaluación

Al finalizar cada Unidad de Trabajo o las unidades de trabajo contenidas en un RA se realizarán las pruebas escritas a mano (en papel) o realizadas mediante ordenador.

Para las pruebas realizadas en el ordenador se utilizará la plataforma **EducamosCLM** para la entrega de las mismas en tiempo y forma. **NO** se recogerán entregas fuera de plazo.

7.4 Criterios de recuperación

Consideraciones:

- El alumno deberá **recuperar los RRAA no logrados**.
- Cuando se recurra a la entrega de prácticas o ejercicios cuya realización pueda albergar dudas de autoría y correcto aprendizaje, **se podrá exigir al alumno que explique y justifique su solución propuesta** y la calificación vendrá determinada por la adecuación de dicha defensa.

CANTIDAD Y MOMENTO DE LAS RECUPERACIONES:

- La recuperación de los RRAA no logrados se realizará en el **periodo extraordinario**, una vez finalizada la primera convocatoria ordinaria.
- Los alumnos **que no hayan superado el módulo en la primera convocatoria ordinaria** del curso (principios de Junio), tendrán que evaluarse de nuevo en la **segunda convocatoria** (finales de Junio) ordinaria del curso:
 - Se realizará un examen a mano en papel o a ordenador **que incluya los RRAA suspensos. Un ejercicio para todos los RRAA, o uno o varios por cada RA suspenso.**

PROCEDIMIENTO DE PÉRDIDA DE EVALUACIÓN CONTINUA:

Pruebas a realizar por la pérdida de evaluación continua:

- **Examen práctico** (a mano o mediante ordenador) para evaluar los CCE de carácter práctico de cada RA. Porcentaje de la nota 85%.
- **Examen teórico-práctico** para evaluar los criterios de orientación más teórica. Porcentaje de la nota 15%

7.5 Instrumentos de evaluación de la programación y de la práctica docente

Desarrollados en la PDD.

8. PROPUESTAS DE ATENCIÓN A LA DIVERSIDAD.

Desarrollado en la PDD.

9. CONCRECIÓN DE LOS PLANES Y PROGRAMACIÓN DE CONTENIDO PEDAGÓGICO DEL CENTRO.

Desarrollados en la PDD.

10. ACTIVIDADES EXTRAESCOLARES.

Desarrollado en la PDD.

10.1 Actividades para alumnos que superen la primera evaluación ordinaria.

Alumnos que hayan superado el curso en la primera evaluación ordinaria final:

- A estos alumnos se les propondrán actividades utilizando un lenguaje de programación distinto a Java. En este caso se realizarán sitios Web con PHP.

Alumnos que deben presentarse a la segunda evaluación ordinaria

- A estos alumnos se les proporcionará las actividades necesarias para practicar y poder recuperar los RRAA suspensos.

11. BIBLIOGRAFÍA.

Para el seguimiento del módulo ya se indicó en el apartado de 6.6. Materiales.