

PROGRAMACIÓN DEL MÓDULO PROFESIONAL

PROGRAMACIÓN

CICLO FORMATIVO DE GRADO SUPERIOR

Desarrollo de Aplicaciones Web

Desarrollo de Aplicaciones Multiplataforma

Código Modulo Programación: 0485

ÍNDICE

<u>1.INTRODUCCIÓN.....</u>	<u>4</u>
<u>2.VINCULACIÓN CON EL CURRÍCULO OFICIAL.....</u>	<u>4</u>
<u>2.1.Introducción.....</u>	<u>4</u>
<u>2.2.Perfil profesional del título.....</u>	<u>4</u>
<u>2.3.Competencia general de título.....</u>	<u>5</u>
<u>2.4.Unidades de competencia del módulo.....</u>	<u>5</u>
<u>UC0964_3: Crear elementos software para la gestión del sistema y sus recursos.....</u>	<u>6</u>
<u>2.5.Ojetivos generales asociados al módulo.....</u>	<u>6</u>
<u>2.6.Competencias profesionales, personales y sociales del módulo.....</u>	<u>8</u>
<u>3.CRITERIOS DE EVALUACIÓN Y RESULTADOS DE APRENDIZAJE.....</u>	<u>9</u>
<u>4.CONTENIDOS: SECUENCIACIÓN Y DISTRIBUCIÓN TEMPORAL EN UNIDADES DE TRABAJO.....</u>	<u>12</u>
<u>4.1.Introducción.....</u>	<u>12</u>
<u>4.2.Contenidos mínimos.....</u>	<u>12</u>
<u>4.3.Unidades de trabajo.....</u>	<u>14</u>
<u>Unidad de trabajo 1. - Introducción a la Programación.....</u>	<u>14</u>
<u>Unidad de trabajo 2. - Identificación de los elementos de un programa informático.....</u>	<u>15</u>
<u>Unidad de trabajo 3. - Introducción a la orientación a objetos.....</u>	<u>16</u>
<u>Unidad de trabajo 4. - Uso de estructuras de control.....</u>	<u>16</u>
<u>Unidad de trabajo 5. - Desarrollo de clases y creación de objetos.....</u>	<u>17</u>
<u>Unidad de trabajo 6. - Aplicación de las estructuras de almacenamiento.....</u>	<u>18</u>
<u>Unidad de trabajo 7. - Utilización avanzada de clases.....</u>	<u>19</u>
<u>Unidad de trabajo 8. - Control y manejo de excepciones.....</u>	<u>20</u>
<u>Unidad de trabajo 9. - Colecciones de datos.....</u>	<u>20</u>
<u>Unidad de trabajo 10. - Lectura y escritura de información.....</u>	<u>21</u>
<u>Unidad de trabajo 11. - Interfaces Gráficas.....</u>	<u>22</u>
<u>Unidad de trabajo 12. - Gestión de bases de datos relacionales.....</u>	<u>22</u>
<u>4.4.Secuenciación y distribución temporal.....</u>	<u>23</u>
<u>5.METODOLOGÍA Y RECURSOS DIDACTICOS.....</u>	<u>24</u>
<u>5.1 MATERIAL DIDACTICO Y RECURSOS.....</u>	<u>26</u>

5.2 MATERIAL BIBLIOGRÁFICO.....	27
6.PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN.....	28
7.CRITERIOS DE CALIFICACIÓN.....	31
8.PROCEDIMIENTO DE RECUPERACIÓN DE EVALUACIONES PENDIENTES....	33
9.PROCEDIMIENTOS Y ACTIVIDADES DE RECUPERACIÓN PARA LOS ALUMNOS CON MÓDULOS PENDIENTES DE CURSOS ANTERIORES.....	33
10.EVALUACIÓN EXTRAORDINARIA.....	34
11.CRITERIOS Y PROCEDIMIENTOS PREVISTOS PARA LA ORGANIZAR LA ATENCIÓN A LA DIVERSIDAD DE LOS ALUMNOS.....	34
12.PROCEDIMIENTO PARA QUE EL ALUMNADO Y, EN SU CASO LAS FAMILIAS (PORQUE EL ALUMNO SEA MENOR DE EDAD), CONOZCAN LOS OBJETIVOS.	35
13.COORDINACION ENTRE PROFESORES QUE IMPARTAN EL MISMO CICLO.	35
14.ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES.....	36
15.EVALUACIÓN DE LA PRACTICA DOCENTE.....	36
16.ANEXO.....	36

1. INTRODUCCIÓN

El presente documento constituye una programación didáctica relativa al módulo **Programación (duración 270 horas)** encuadrado en el **primer curso del Ciclo Formativo de Grado Superior Desarrollo de Aplicaciones Multiplataforma** y el **primer curso del Ciclo Desarrollo de Aplicaciones Web**.

En esta programación quedan recogidas las normas derivadas del reglamento de régimen interior del centro educativo en lo relativo a la pérdida de evaluación ordinaria a los módulos por acumulación de faltas y la no superación del módulo de FCT.

2. VINCULACIÓN CON EL CURRÍCULO OFICIAL

2.1. Introducción

Los siguientes decretos constituyen la principal fuente para la elaboración de la programación didáctica:

- El **REAL DECRETO 450/2010**, de 16 de Abril , por el que se establece el **título de Técnico en Desarrollo de Aplicaciones Multiplataforma** y se fijan sus **enseñanzas mínimas**.
- El **REAL DECRETO 686/2010**, de 20 de Mayo , por el que se establece el **título de Técnico en Desarrollo de Aplicaciones Web** y se fijan sus enseñanzas **mínimas**.
- **DECRETO 3/2011, de 13 de enero**, del Consejo de Gobierno por el que se establece para la Comunidad de Madrid el currículo de ciclo formativo de grado superior correspondiente al título de **Técnico Superior en Desarrollo de Aplicaciones Multiplataforma**.
- **DECRETO 1/2011, de 13 de enero**, del Consejo de Gobierno por el que se establece para la Comunidad de Madrid el currículo de ciclo formativo de grado superior correspondiente al título de **Técnico Superior en Desarrollo de Aplicaciones Web**.

2.2. Perfil profesional del título.

El **perfil profesional** de los títulos de **Técnico en Desarrollo de Aplicaciones Multiplataforma** y de **Técnico en Desarrollo de Aplicaciones Web**, quedan determinados por su **competencia general**, sus **competencias profesionales, personales y sociales**, y por la relación de **cualificaciones y, en su caso, unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales incluidas en el título**.

2.3. Competencia general de título

La competencia general describe el rol o roles esencial/es de trabajo que se han identificado para cada figura profesional en forma de funciones y objetivos que deben ser alcanzados. La competencia general de este título consiste en:

Para el título de **Técnico Superior en Desarrollo de Aplicaciones Multiplataforma**:

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.

Para el título de **Técnico Superior en Desarrollo de Aplicaciones Web**:

La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.

2.4. Unidades de competencia del módulo

Las Cualificaciones profesionales completas que incluye el título determinan que el perfil que pretende conseguir el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Multiplataforma** en el alumno se corresponde con las siguientes unidades de competencia y cualificaciones profesionales asociadas al ciclo:

a) Programación en lenguajes estructurados de aplicaciones de gestión IFC155_3 (R.D. 1087/2005, de 16 de septiembre), que comprende las siguientes unidades de competencia:

UC0226_3: Programar bases de datos relacionales.

UC0494_3: Desarrollar componentes software en lenguajes de programación estructurada.

b) Programación con lenguajes orientados a objetos y bases de datos relacionales IFC080_3 (R.D. 295/2004, de 20 de febrero), que comprende las siguientes unidades de competencia:

UC0226_3: Programar bases de datos relacionales.

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

Asimismo, Las Cualificaciones profesionales incompletas que incluye el título determinan que el perfil que pretende conseguir el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones**

Multiplataforma en el alumno se corresponde con las siguientes unidades de competencia y cualificaciones profesionales asociadas al ciclo:

UC0964_3: Crear elementos software para la gestión del sistema y sus recursos

Las Cualificaciones profesionales completas que incluye el título determinan que el perfil que pretende conseguir el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Web** en el alumno se corresponde con las siguientes unidades de competencia y cualificaciones profesionales asociadas al ciclo:

Desarrollo de aplicaciones con tecnologías Web IFC154_3 (Real Decreto 1087/2005, de 16 de septiembre), que comprende las siguientes unidades de competencia:

UC0491_3 Desarrollar elementos software en el entorno cliente.

UC0492_3 Desarrollar elementos software en el entorno servidor.

UC0493_3 Implementar, verificar y documentar aplicaciones web en entornos internet, intranet y extranet.

Asimismo, Las Cualificaciones profesionales incompletas que incluye el título determinan que el perfil que pretende conseguir el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Web** en el alumno se corresponde con las siguientes unidades de competencia y cualificaciones profesionales asociadas al ciclo:

UC0226_3 Programar bases de datos relacionales.

2.5.Ojetivos generales asociados al módulo

La formación del módulo contribuye a alcanzar los objetivos generales:

Para el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Multiplataforma**:

- Instalar y configurar módulos y complementos, evaluando su funcionalidad, para gestionar entornos de desarrollo.
- Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.
- Gestionar la información almacenada, planificando e implementando sistemas de formularios e informes para desarrollar aplicaciones de gestión.
- Seleccionar y utilizar herramientas específicas, lenguajes y librerías, evaluando sus posibilidades y siguiendo un manual de estilo, para manipular e integrar en aplicaciones multiplataforma contenidos gráficos y componentes multimedia.
- Emplear herramientas de desarrollo, lenguajes y componentes visuales, siguiendo las especificaciones y verificando interactividad

y usabilidad, para desarrollar interfaces gráficos de usuario en aplicaciones multiplataforma.

- Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
- Valorar y emplear herramientas específicas, atendiendo a la estructura de los contenidos, para crear tutoriales, manuales de usuario y otros documentos asociados a una aplicación.
- Seleccionar y emplear técnicas y herramientas, evaluando la utilidad de los asistentes de instalación generados, para empaquetar aplicaciones.
- Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y organización de trabajo y de la vida personal.

Para el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Web**:

- Seleccionar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- Generar componentes de acceso a datos, cumpliendo las especificaciones, para integrar contenidos en la lógica de una aplicación web.
- Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
- Utilizar herramientas y lenguajes específicos, cumpliendo las especificaciones, para desarrollar e integrar componentes software en el entorno del servidor web
- Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
- Utilizar herramientas específicas, cumpliendo los estándares establecidos, para elaborar y mantener la documentación de los procesos.
- Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y organización de trabajo y de la vida personal.

Por otra parte, en este módulo, se orientarán las actividades del alumno para llevar a cabo los objetivos descritos en el *Proyecto Educativo del Centro* en lo que respecta a convivencia, integración, trabajo en equipo y respeto mutuo entre los integrantes de la comunidad docente.

2.6. Competencias profesionales, personales y sociales del módulo

La formación del módulo contribuye a alcanzar las siguientes competencias del título:

Para ambos módulos:

- Establecer vías eficaces de relación profesional y comunicación con sus superiores, compañeros y subordinados, respetando la autonomía y competencias de las distintas personas.
- En consonancia al acuerdo del consejo escolar en su reunión del 30 de junio de 2016 se podrá decidir, de forma justificada, qué el alumno no promocionará a la FCT, o al periodo de prácticas, si se considera que no se han alcanzado las competencias personales y sociales que permitan al alumno su integración total en el ámbito laboral durante sus prácticas.

Para el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Multiplataforma**:

- Gestionar entornos de desarrollo adaptando su configuración en cada caso para permitir el desarrollo y despliegue de aplicaciones.
- Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías y herramientas adecuados a las especificaciones.
- Desarrollar aplicaciones implementando un sistema completo de formularios e informes que permitan gestionar de forma integral la información almacenada.
- Integrar contenidos gráficos y componentes multimedia en aplicaciones multiplataforma, empleando herramientas específicas y cumpliendo los requerimientos establecidos.
- Desarrollar interfaces gráficos de usuario interactivos y con la usabilidad adecuada, empleando componentes visuales estándar o implementando componentes visuales específicos.
- Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados según las especificaciones.

- Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones.
- Gestionar su carrera profesional, analizando las oportunidades de empleo, autoempleo y de aprendizaje.
- Mantener el espíritu de innovación y actualización en el ámbito de su trabajo para adaptarse a los cambios tecnológicos y organizativos de su entorno profesional.

Para el módulo **Programación** del título **Técnico en Desarrollo de Aplicaciones Web**:

- Desarrollar aplicaciones web con acceso a bases de datos utilizando lenguajes, objetos de acceso y herramientas de mapeo adecuados a las especificaciones.
- Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.
- Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados según las especificaciones.
- Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones.
- Gestionar su carrera profesional, analizando las oportunidades de empleo, autoempleo y de aprendizaje.
- Mantener el espíritu de innovación y actualización en el ámbito de su trabajo para adaptarse a los cambios tecnológicos y organizativos de su entorno profesional.

3. CRITERIOS DE EVALUACIÓN Y RESULTADOS DE APRENDIZAJE

Los criterios de evaluación se relacionan con los objetivos específicos de este módulo (resultados del aprendizaje) de la siguiente forma:

Resultados de aprendizaje	Unidades	Criterios de evaluación
1 Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de	1	<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

programación utilizado.		<p>utilizar variables.</p> <ul style="list-style-type: none"> •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 explícitas e implícitas. •Se han introducido comentarios en el código.
2. Identifica y usa los principales elementos de un programa informático.	2	<ul style="list-style-type: none"> •Se han instalado entornos de desarrollo, propietarios y libres. •Se han añadido y eliminado módulos en el entorno de desarrollo. •Se ha personalizado y automatizado el entorno de desarrollo. •Se ha configurado el sistema de actualización del entorno de desarrollo. •Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo. •Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo. •Se han identificado las características comunes y específicas de diversos entornos de desarrollo. Verifica el funcionamiento de programas, diseñando y realizando pruebas
3. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.	3	<ul style="list-style-type: none"> •Se han identificado los fundamentos de la programación orientada a objetos. •Se han escrito programas simples. •Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.
4. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.	4	<ul style="list-style-type: none"> •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.
5. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos y programas utilizando objetos y analizando y aplicando los principios de la programación orientada a objetos.	5	<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

		<ul style="list-style-type: none"> •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.
6. Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.	6	<ul style="list-style-type: none"> •Se han escrito programas que utilicen arrays. •Se han realizado programas que realicen las operaciones de ordenación y búsqueda en un array. •Se han escrito programas que manejen cadenas de caracteres. •Se han escrito programas que realicen las diferentes operaciones sobre las cadenas de caracteres. •Se han escrito programas que utilicen expresiones regulares.
7.Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación	7	<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.
8.Escribe programas que incorporan control de excepciones.	8	<ul style="list-style-type: none"> •Se han identificado el concepto de excepción. •Se han utilizado jerarquías de excepciones. •Se ha entendido el manejo de las excepciones en los diferentes casos posibles.
9. Desarrolla programas en los que se implementan colecciones y sus operaciones mediante clases y métodos.	9	<ul style="list-style-type: none"> •Se han identificado los distintos tipos de colecciones y sus diferentes utilidades. •Se conocen las jerarquías de colecciones. •Se han identificado las operaciones que se pueden llevar a cabo con cada colección. •Se ha entendido como usar clases y métodos genéricos para manejar las colecciones. •Se utilizan de forma adecuada las colecciones incorporadas en Java
10- Desarrolla programas en los que se implementan diferentes tipos de flujos, se manejan ficheros	10	<ul style="list-style-type: none"> •Se han identificado los distintos tipos de flujos existentes en Java. •Se han identificado las clases relativas a flujos y sus jerarquías. •Se han utilizado flujos. •Se han realizados programas en los que intervenga la entrada salida a través del teclado y la pantalla. •Se han realizado programas de manejo de ficheros.
11.Desarrolla programas bajo interfaz gráfica	11	<ul style="list-style-type: none"> •Se han identificado las características y jerarquías de las interfaces gráficas en Java. •Se han creado programas bajo interfaz gráfica.
12.Gestiona información	12	<ul style="list-style-type: none"> •Se han identificado las características y métodos de

almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos.		<p>acceso a sistemas gestores de bases de datos relacionales.</p> <ul style="list-style-type: none"> •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.
---	--	--

4. CONTENIDOS: SECUENCIACIÓN Y DISTRIBUCIÓN TEMPORAL EN UNIDADES DE TRABAJO

4.1.Introducción

La propuesta de programación está constituida por una relación secuenciada de unidades de trabajo. Los elementos curriculares que se tratan en las mismas y su distribución temporal se han estructurado con el objetivo de que los alumnos desarrollen una serie de destrezas, técnicas y habilidades que permitan su incorporación al mundo laboral. Se han tenido en cuenta los elementos curriculares obtenidos anteriormente, la orientación práctica de los ciclos formativos de formación profesional y el nivel de desarrollo de los alumnos así como su grado de conocimiento previo.

En cada unidad de trabajo se proponen una serie de actividades de enseñanza aprendizaje las cuales podrán ser realizadas en parte o en su totalidad según la evolución de los alumnos a lo largo del curso.

4.2.Contenidos mínimos

Los contenidos básicos para este módulo vienen indicados en:

El **REAL DECRETO 450/2010**, de 16 de Abril , por el que se establece el **título de Técnico en Desarrollo de Aplicaciones Multiplataforma** y se fijan sus **enseñanzas mínimas**.

El **REAL DECRETO 686/2010**, de 20 de Mayo , por el que se establece el **título de Técnico en Desarrollo de Aplicaciones Web** y se fijan sus enseñanzas **mínimas**.

En ambos casos son los mismos:

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.

Desarrollo de clases y Utilización de objetos :

- 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.
- 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.

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.
- Expresiones Regulares.
- Listas.
- Colecciones en Java.

Utilización avanzada de clases:

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

Interfaces Gráficos

- Conceptos y definición.
- Desarrollo de Aplicaciones gráficas.

Gestión de bases de datos relacionales:

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

4.3. Unidades de trabajo

Unidad de trabajo 1. - Introducción a la Programación

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Identificar los elementos que forman parte de los diferentes bloques de un programa.
- Entender el concepto de algoritmo
- Realizar algoritmos en lenguaje humano sobre tareas normales del mundo real.
- Valorar los diferentes tipos de lenguajes de programación en función de sus características y la adecuación de su uso en función del tipo de proyecto a abordar.
- Distinguir las diferentes etapas por las que pasa una aplicación durante su desarrollo , entendiendo que tareas se realizan en cada etapa.
- Introducirse en el manejo de alguna herramienta de desarrollo.
- Identificar los distintos tipos de errores que se pueden encontrar en un programa y como afectan al software desarrollado.

Criterios de evaluación

- 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.

Contenidos

- Datos, algoritmos y programas.
- Paradigmas de programación.

- Lenguajes de programación.
- Herramientas y entornos para el desarrollo de programas.
- Errores y calidad de los programas.

Unidad de trabajo 2. - Identificación de los elementos de un programa informático

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Realizar programas en pseudocódigo utilizando variables, constantes, etc y operaciones de entrada y salida.
- Realizar programas en Java en los que se usen los diferentes tipos de variables básicas.
- Realizar programas en Java en los que se usen constantes y literales.
- Realizar programas en Java en los que entre en juego la escritura de diferentes tipos de expresiones.
- Realizar programas en Java en los que entre en juego la conversión entre distintos tipos de datos.
- Comentar el código realizado en Java
- Probar y depurar el código realizado.
- Realizar pequeñas documentaciones sobre alguno de los programas.

Criterios de evaluación

- 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 explícitas e implícitas.
- Se han introducido comentarios en el código.
- Se han probado y depurado los programas.
- Se ha comentado y documentado el código.

Contenidos

- Estructura y bloques fundamentales.
- Identificadores.
- Palabras reservadas.
- Variables. Declaración, inicialización y utilización. Almacenamiento en memoria.
- Tipos de datos.
- Literales.
- Constantes.

- Operadores y expresiones. Precedencia de operadores
- Conversiones de tipo. Implícitas y explícitas (casting).
- Comentarios.

Unidad de trabajo 3. - Introducción a la orientación a objetos

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Identificar los elementos fundamentales de la P.O.O
- Escribir programas sencillos según esta metodología

Criterios de evaluación

- Se han identificado los fundamentos de la programación orientada a objetos.
- Se han escrito programas simples.
- Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

Contenidos

- Clases. Atributos, métodos y visibilidad.
- Objetos. Estado, comportamiento e identidad. Mensajes.
- Encapsulado. Visibilidad.
- Relaciones entre clases.
- Principios básicos de la orientación a objetos.

Unidad de trabajo 4. - Uso de estructuras de control

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Escribir código en el que intervengan estructuras de selección.
- Escribir código en el que intervengan estructuras de repetición..
- Escribir código en el que intervengan sentencias de salto.
- Escribir código en el que se realice control de excepciones.
- Crear ejecutables en los que intervengan este tipo de sentencias.
- Probar, depurar y comentar el código anterior.

Criterios 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.

Contenidos

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.

Unidad de trabajo 5. - Desarrollo de clases y creación de objetos

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Definir clases comprendiendo su sintaxis, estructura y componentes típicos.
- Crear propiedades ,métodos y constructores para una clase.
- Codificar programas que instancien y utilicen objetos de las clases creadas con anterioridad.
- Establecer mecanismos que controlen la visibilidad de las clases creadas.
- Definir y utilizar clases heredadas.
- Crear objetos a partir de clases ya existentes
- Utilizar métodos de los objetos
- Escribir llamadas a métodos estáticos pasando parámetros.
- Incorporar a los programas librerías de objetos.
- Utilizar constructores.
- Realizar estas tareas mediante el entorno de desarrollo.

Criterios de evaluación

- Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- Se han definido clases.
- Se han definido propiedades y métodos.
- Se han instanciado objetos de clases desarrolladas.
- Se han creado y utilizado constructores adecuados.
- Se han instanciado objetos con diferentes 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 maneja paso de parámetros por valor y por referencia.
- Se han definido y utilizado interfaces.

- Se han creado y utilizado conjuntos y librerías de clases.

Contenidos

- Concepto de clase y objeto.
- Estructura y miembros de una clase.
- Creación de atributos. Declaración e inicialización.
- Creación de métodos. Declaración, argumentos y valores de retorno.
- Instanciación de objetos a partir de clases desarrolladas.
- Creación de constructores.
- Instanciación de objetos con diferentes constructores.
- Ámbito de atributos y variables.
- Sobrecarga de métodos.
- Visibilidad. Modificadores de clase, de atributos y de métodos.
- Paso de parámetros. Paso por valor y paso por referencia.
- Características de los objetos.
- Constructores.
- Instanciación de objetos. Declaración y creación.
- Utilización de métodos. Parámetros y valores de retorno.
- Utilización de propiedades.
- Utilización de métodos estáticos.
- Almacenamiento en memoria. Tipos básicos vs. objetos.
- Destrucción de objetos y liberación de memoria.

Unidad de trabajo 6. - Aplicación de las estructuras de almacenamiento

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Realizar programas que utilicen arrays como estructuras de almacenamiento.
- Realizar programas que realicen operaciones sobre arrays: ordenación y búsqueda.
- Realizar programas que utilicen cadenas de caracteres como estructuras de almacenamiento.
- Realizar programas que realicen operaciones sobre cadenas de caracteres: copia, comparación, concatenación, extracción de subcadenas, etc.
- Realizar programas que manejen Expresiones Regulares.

Criterios de evaluación

- Se han escrito programas que utilicen arrays.
- Se han realizado programas que realicen las operaciones de ordenación y búsqueda en un array.
- Se han escrito programas que manejen cadenas de caracteres.
- Se han escrito programas que realicen las diferentes operaciones sobre las cadenas de caracteres
- Se han escrito programas que manejen expresiones regulares.

Contenidos

- Estructuras.
- Arrays unidimensionales y multidimensionales:
- Declaración.
- Creación de arrays unidimensionales y multidimensionales.
- Inicialización
- Acceso a elementos.
- Recorridos, búsquedas y ordenaciones.
- Cadenas de caracteres:
- Declaración.
- Creación de cadenas de caracteres.
- Inicialización
- Operaciones. Acceso a elementos, conversiones, concatenación.
- Expresiones Regulares.

Unidad de trabajo 7. - Utilización avanzada de clases

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Realizar programas que contemplen los conceptos de herencia, superclase y subclase.
- Realizar programas que utilicen modificadores que bloqueen y fuercen la herencia de clases y métodos.
- Realizar programas que utilicen clases que sobrescriban los métodos implementados en la superclase.
- Diseñar, aplicar, probar y depurar jerarquías de clases.
- Realizar programas que utilicen jerarquías de clases.
- Comentar y documentar el código realizado sobre los conceptos antes expuestos.

Criterios de evaluación

- 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.

Contenidos

- Composición de clases.
- Herencia.
- Superclases y subclases.
- Clases y métodos abstractos y finales.

- Sobreescritura de métodos.
- Constructores y herencia.

Unidad de trabajo 8. - Control y manejo de excepciones

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Realizar programas que incluyan excepciones.
- Diseñar jerarquías de excepciones.
- Realizar programas que utilicen esas jerarquías.
- Realizar programas que contemplen todos los posibles casos del manejo de excepciones.

Criterios de evaluación

- Se han identificado el concepto de excepción.
- Se han utilizado jerarquías de excepciones.
- Se ha entendido el manejo de las excepciones en los diferentes casos posibles.

Contenidos

- Excepciones. Concepto.
- Jerarquías de excepciones.
- Manejo de excepciones:
- Captura de excepciones.
- Propagar excepciones.
- Lanzar excepciones.
- Crear clases de excepciones.

Unidad de trabajo 9. - Colecciones de datos

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Identificar cada colección y sus características.
- Identificar y entender las operaciones que pueden ser llevadas a cabo con cada tipo de colecciones.
- Implementar mediante clases los distintos tipos de colecciones y sus operaciones mediante métodos.
- Uso de las colecciones incorporadas en Java.

Criterios de evaluación

- Se han identificado los distintos tipos de colecciones y sus diferentes utilidades.
- Se conocen las jerarquías de colecciones.

- Se han identificado las operaciones que se pueden llevar a cabo con cada colección.
- Se ha entendido como usar clases y métodos genéricos para manejar las colecciones.
- Se conocen y utilizan adecuadamente las colecciones incorporadas en Java.

Contenidos

- Tipos de colecciones (listas, pilas, colas, tablas...).
- Jerarquías de colecciones.
- Operaciones con colecciones. Acceso a elementos y recorridos.
- Uso de clases y métodos genéricos.
- Colecciones incorporadas en Java.

Unidad de trabajo 10. - Lectura y escritura de información

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Identificar los distintos tipos de flujos existentes en Java.
- Identificar las clases relativas a flujos en Java y sus jerarquías.
- Realizar programas en los que se implementen distintos tipos de flujos.
- Realizar programas en los que se realice manejo de ficheros.

Criterios de evaluación

- Se han identificado los distintos tipos de flujos existentes en Java.
- Se han identificado las clases relativas a flujos y sus jerarquías.
- Se han utilizado flujos.
- Se han realizados programas en los que intervenga la entrada salida a través del teclado y la pantalla.
- Se han realizado programas de manejo de ficheros.

Contenidos

- Flujos (streams):
 - Tipos de flujos. Flujos de bytes y de caracteres.
 - Clases relativas a flujos. Jerarquías de clases.
 - Utilización de flujos.
- Entrada/salida estándar:
 - Entrada desde teclado.
 - Salida a pantalla.
- Almacenamiento de información en ficheros:
- Ficheros de datos. Registros.
- Apertura y cierre de ficheros. Modos de acceso.
- Escritura y lectura de información en ficheros.
- Almacenamiento de objetos en ficheros. Persistencia. Serialización.
- Utilización de los sistemas de ficheros.
- Creación y eliminación de ficheros y directorios

Unidad de trabajo 11. - Interfaces Gráficas.

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Identificar los diferentes elementos en el desarrollo de programas utilizando interfaces gráficas de usuario.
- Uso de librerías para el desarrollo de GUIs.
- Realización de programas en entornos gráficos.

Criterios de evaluación

- Se han identificado los diferentes elementos en el desarrollo de GUIs.
- Se han completado programas en entornos gráficos.
- Se han utilizado los asistentes y herramientas del entorno integrado.

Contenidos

- Interfaces
- Librerías Java para GUIs.
- Gestión de Eventos.
- Generación de programas en entorno gráfico.

Unidad de trabajo 12. - Gestión de bases de datos relacionales.

Resultados de aprendizaje

Los indicados en la tabla anterior para esta unidad de trabajo

Objetivos

- Identificar las características y métodos de acceso a SGBDR.
- Programar conexiones con diferentes bases de datos.
- Realizar programas que lleven a cabo el almacenamiento de información en bases de datos.
- Realizar programas que lleven a cabo la recuperación y salida de los datos contenidos en una base de datos.
- Realizar programas que lleven a cabo el borrado y modificación de los datos de una base de datos.

Criterios de evaluación

- 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.

Contenidos

- Interfaces de programación de acceso a bases de datos.
- Establecimiento de conexiones.
- Recuperación de información.
- Manipulación de la información.
- Ejecución de consultas sobre la base de datos.

4.4. Secuenciación y distribución temporal

Este modulo se imparte en el primer curso del ciclo formativo (tanto para el título de Técnico Superior de Desarrollo de Aplicaciones Multiplataforma, como para el título de Técnico Superior de Desarrollo de Aplicaciones Web), y tiene una duración de 270 horas lectivas, a razón de 8 horas a la semana¹.

A continuación se plantea el calendario de ejecución de las unidades de trabajo descritas en el apartado anterior:

Unidad de Trabajo	Duración prevista (hrs.)
1. INTRODUCCIÓN A LA PROGRAMACIÓN	22
2. IDENTIFICACIÓN DE LOS ELEMENTOS DE UN PROGRAMA INFORMÁTICO	18
3. INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS	8
4. USO DE ESTRUCTURAS DE CONTROL	22
5. DESARROLLO DE CLASES Y CREACIÓN DE OBJETOS	52
6. APLICACIÓN DE LAS ESTRUCTURAS DE ALMACENAMIENTO	30
7. UTILIZACIÓN AVANZADA DE CLASES	20
8. CONTROL Y MANEJO DE EXCEPCIONES	18
9. COLECCIONES DE DATOS	20
10. LECTURA Y ESCRITURA DE INFORMACIÓN	20
11. INTERFACES GRÁFICAS	20
12. GESTIÓN DE BASES DE DATOS RELACIONALES Y	20

¹Se propone una distribución basada en el currículo de la Comunidad de Madrid que habrá que adaptar para otras comunidades autónomas.

PERSISTENCIA	
TOTAL HORAS	270

El orden de las unidades de trabajo y su temporización podría variar a lo largo del curso dependiendo del criterio del profesor y según las necesidades y evolución del grupo.

5. METODOLOGÍA Y RECURSOS DIDACTICOS

La metodología didáctica de las enseñanzas de formación profesional debe integrar los aspectos científicos, tecnológicos y organizativos que en cada caso correspondan, con el fin de que el alumnado adquiriera una visión global de los procesos productivos propios de la actividad profesional correspondiente. Además, favorecerá que el alumno/a se forme como ser **autónomo** planteándose interrogantes, participando y asumiendo responsabilidades y, por tanto, que desarrolle la *capacidad para aprender por sí mismo*.

La metodología, a su vez, debe conseguir ser motivadora de futuros aprendizajes y debe ayudar a comprender al alumno que el aprendizaje es algo que nunca se acaba ya que los cambios en el sistema productivo, los adelantos tecnológicos, los nuevos descubrimientos y la propia posición del individuo le debe convertir en un sujeto “deseante de más conocimiento”. Para ello es básico orientar la enseñanza hacia unos aprendizajes que relacionen los contenidos teóricos con la práctica. Y es el profesor el que ha de actuar como guía proporcionando los recursos necesarios y planificando las situaciones para que se puedan llevar a cabo los aprendizajes. En ese sentido, las decisiones metodológicas que se adopten deben tener en cuenta que *aprender es asimilar significados nuevos en un proceso interactivo entre el propio alumno, los contenidos, el profesor y los compañeros o compañeras*. De ahí, la importancia de **trabajar en equipo**.

En este punto hay que considerar que la mayoría de los contenidos del módulo son de tipo procedimental y, por lo tanto, las actividades que se desarrollen a lo largo del curso deben estar orientadas a que el alumno desarrolle este tipo de capacidades, estando el profesor asumiendo un papel de animador, organizador y orientador de las diferentes tareas a realizar.

Al comienzo del curso se realizará una presentación general de objetivos, contenidos y actividades y procedimientos de evaluación y calificación.

Cada unidad de trabajo quedará centrada en el contexto del módulo y, si es posible, se intentará relacionar con otros módulos. A continuación, se iniciará la unidad mediante una explicación teórica de los conceptos necesarios y sencillos supuestos que permiten encauzar las tareas de aprendizaje de tipo esencialmente procedimental, dejándose determinados

conceptos y explicaciones para aquellas otras actividades específicas que los necesiten.

Las actividades propuestas tenderán a desarrollar la autonomía en el aprendizaje de los alumnos, para que, en la medida de lo posible, el alumno sepa enfrentarse a los problemas que puedan surgirles en otras situaciones y en su futuro trabajo. Por todo ello, las actividades pueden encuadrarse en los distintos tipos de actuaciones:

- a) Resolución de supuestos prácticos sencillos.
- b) Desarrollo de ejercicios más complejos.
- c) Documentación clara y exhaustiva de los trabajos realizados.
- d) Acceso a fuentes de documentación:
- e) Exposición de trabajos y soluciones a problemas:
 - Exposición y explicación de trabajos de investigación.
 - Presentación de las soluciones encontradas en las actividades propuestas (en especial las que tienen que ver con la resolución de problemas a través de la codificación de algoritmos con lenguajes de programación).

El trabajo diario en el aula se llevará a cabo en dos fases:

- En una primera fase el tema será presentado, desde un punto de vista teórico, por el profesor, haciendo intervenir al alumno mediante preguntas para detectar si el nivel de partida es el apropiado y si la asimilación es correcta. Posteriormente se resolverán en el aula ejercicios prácticos, de forma individual y en equipo, según sea más conveniente en cada caso.
- En una segunda fase y también de forma práctica, se procederá a la resolución y puesta en marcha en el ordenador de los ejercicios resueltos y de otros propuestos. En esta fase se efectuará también una puesta en común (por parte de los alumnos) de las conclusiones a las que han llegado, de los problemas que les han surgido y de los resultados obtenidos. Además, se propondrán ejercicios a los alumnos, para su resolución en casa.

Se utilizará el cañón para ayudarse en la exposición de los diferentes contenidos. Se procurará que alguna unidad vaya acompañada de un trabajo de investigación, que permita ampliar y/o asentar conocimientos relativos a la unidad explicada. Se instará a algún grupo de alumnos a que realice la exposición del trabajo.

5.1 MATERIAL DIDACTICO Y RECURSOS

Los medios que deben estar al alcance del alumno para el correcto seguimiento del módulo, son los siguientes:

a) Tecnológicos:

- El alumno dispondrá de un ordenador en la mayoría de las horas semanales.

- Es responsabilidad del alumno el mantenimiento personal de las necesarias copias de seguridad de sus trabajos, ya sea en un servidor local, en dispositivos extraíbles o en un servidor remoto, no siendo disculpa alguna la pérdida de datos por su negligencia. El profesor le indicará la forma adecuada en cada caso de mantener dichas copias.
- Es responsabilidad del alumno mantener su puesto de trabajo en condiciones de funcionamiento correcto, rellenando en cada sesión de trabajo el parte de incidencias o falta de ellas, habilitado para tal fin, informando de cualquier anomalía, desperfecto, pérdida deliberada o fortuita de información provocada por terceras personas, etc. haciéndosele responsable de los desperfectos que no denuncie a su debido tiempo, provocados o no por el alumno, pudiendo ser sancionado, si el equipo docente así lo considera oportuno, tipificándolo como falta grave.
- El aula taller dispondrá de un sistema en red.

b) Didácticos.

- No utilizaremos un libro de texto formal asociado al módulo. Sin embargo, de forma puntual se podría proponer una bibliografía básica con algún libro que siga un guión de contenidos afín a los contenidos anteriormente expuestos, así como algún libro de referencia para algún punto concreto, o bien manuales y/o tutoriales disponibles en Internet.
- La clase es básicamente de tipo práctico, por lo que es fundamental que el alumno maneje la documentación de la herramienta que se esté tratando de forma que sepa interpretarla correctamente, investigue soluciones válidas a los problemas propuestos.
- El alumno tomará apuntes en clase, que reflejen lo principal de las explicaciones teóricas dadas por el profesor. Éste es uno de los mejores ejercicios de comprensión de los conceptos expuestos en clase, y su posible aclaración en casos difíciles o de dudosa comprensión.
- El profesor también tendrá a su disposición, dentro del aula taller, medios audiovisuales, y ordenador de aula con pantalla proyectada, que se utilizará cuando lo crea necesario.
- El alumno dispondrá de la ayuda en las herramientas utilizadas en el desarrollo de aplicaciones, que será objeto de evaluación permanente.

- Se podrá proveer de una herramienta de gestión de repositorios (github, gitlab) como apoyo a laboratorios, contenidos teóricos, etc.

5.2 MATERIAL BIBLIOGRÁFICO

Tal y como se menciona en el apartado anterior:

No utilizaremos un libro de texto formal asociado al módulo. Sin embargo, de forma puntual se podría proponer una bibliografía básica o recursos de internet que siga un guión de contenidos afín a los contenidos anteriormente expuestos, así como algún libro de referencia para algún punto concreto, o bien manuales y/o tutoriales o artículos disponibles en Internet.

Se podrá indicar a los alumnos bibliografía complementaria en cada una de las unidades didácticas.

La bibliografía que se utilice como material o para diseñar actividades prácticas se notificará a los alumnos en clase.

Bibliografía General para Java :

Enlaces y recursos web :

[Java™ Platform, Standard Edition (SE) 8. API Specification]

(<https://docs.oracle.com/javase/8/docs/api/overview-summary.html>)

[Curso Java uc3m]

(<http://ocw.uc3m.es/historico/programacion-java>)

[Curso DAM de Java]

(<http://tazasdejava.blogspot.com.es/p/curso-java.html>)

[Curso Java Gratuito]

(https://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188)

[Curso Universidad de Navarra en pdf]

(<http://www4.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/Java/Java2.pdf>).

Tutorial de [Java muy bueno]

(<https://www.tutorialspoint.com/java/index.htm>). _Ingles_

[IDE para aprender Java **Bluej**]

(<https://www.bluej.org/>)

- IDE sencillo y amigable para los primeros pasos de Java.

- [Video Tutorials](<https://www.bluej.org/doc/video-tutorials.html>)

[From **Bluej** to NetBeans**](<https://youtu.be/dNAbhoO7QFY>).

- Poner Subtítulos en Español si es necesario.

- Muestra la transición e Interoperatividad entre Bluej y NetBeans.

[Píldoras Informáticas](https://www.pildorasinformaticas.es/). Sitio con mucho material para informática.

- [Vídeos de Píldoras Informáticas](https://www.youtube.com/user/pildorasinformaticas/featured) generales y específicos para [Java](https://youtu.be/coK4jM5wvko).

<<https://beginnersbook.com/>>

- Sitio muy bien montado, sencillo y con muchos ejemplos.
- Java con diversos aspectos y otras cosas.

Catálogo de Ejemplos de Java.

<<http://www.java2s.com/Code/Java/CatalogJava.htm>>

Herramienta de repositorios de tipo git : www.github.com

Otra herramienta : www.gitlab.com

Bibliografía Clásica :

David Camacho, José M^a Valls, Jesús García, José M. Molina, Enrique Bueno. Programación, algoritmos y ejercicios resueltos en Java. Pearson/Prentice Hall (2003)

Curso básico de Java que cubre la mayoría de temas de la asignatura

Bruce Eckel. Piensa en Java. 2ª ed. Prentice Hall, (2002)

Mucho más extenso que el anterior, su contenido excede el del curso, pero puede ser muy útil como libro para saber más sobre el tema. La versión inglesa se puede descargar libremente desde <http://www.mindviewinc.com/downloads/TIJ-3rd-edition4.0.zip>

Paul S. Wang. Java con programación orientada a objetos y aplicaciones en la www. Thomson International, (1999)

Algo antiguo, pero interesante para ver las aplicaciones web de Java

Javier García de Jalón, José Ignacio Rodríguez, Iñigo Mingo, Aitor Imaz, Alfonso Brazález, Alberto Larzabal, Jesús Calleja, Jon García. Aprenda Java como si estuviera en primero. Universidad de Navarra (2000)

Manual de Java muy sencillo de entender, muy recomendable para empezar a entender el lenguaje.

Deitel, Paul, Deitel, Harvey. "Java. Cómo programar". Pearson, 10ed.2016

Liang Y., Daniel. "Introduction to Java Programming and Data Structures". Pearson. 11ed. 2019

6. PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN

La evaluación del módulo tendrá en cuenta la legislación vigente sobre el tema, en concreto la **ORDEN 2694/2009, de 9 de junio**, que dice en su **art. 24, apartado 2**: *“La evaluación de estas enseñanzas tendrá por objeto valorar los avances de los alumnos en relación con la competencia general del título y con los objetivos generales del ciclo formativo. Para aplicarla, se tendrán en cuenta los siguientes referentes:*

- a) Para los módulos profesionales de formación en el centro educativo: Los resultados de aprendizaje y los criterios de evaluación especificados en los Reales Decretos que establecen los títulos y las*

correspondientes enseñanzas mínimas y los contenidos curriculares aplicables en la Comunidad de Madrid para cada ciclo formativo...”

Se plantea una evaluación trimestral, global y tan individualizada como sea posible. Algunos aspectos que hay que considerar en la evaluación son:

- actitud grupal e individual
- trabajo individual y en equipo
- expresión oral y escrita
- presentación
- rigor, conceptos y procedimientos.
- Pruebas y actividades realizadas.

También se ha de proporcionar oportunidades para la auto-evaluación por parte del alumno mediante la entrega de numerosos ejercicios. Para ello es imprescindible, por parte del alumnado, la asistencia regular a las clases y actividades programadas, por lo que se controlarán la puntualidad y las faltas de asistencia a clase.

Para poder evaluar se llevarán a cabo las siguientes acciones:

- Una o varias **pruebas** por cada una de las tres evaluaciones que podrá tener una parte de contenido teórico y otra parte de contenido práctico. Dicha/s prueba/s será práctica en tanto en cuanto lo permitan los contenidos de las unidades didácticas incluidas en cada una de ellas, pero en cualquier caso se podrá incluir una parte de teoría. La parte práctica, se realiza por escrito o en ordenador, a criterio del profesor. En particular, se podrán realizar pruebas en ordenador, con tiempo limitado, de forma que la agilidad y precisión en la realización de dichas pruebas sea un factor fundamental a la hora de calificar al alumno.
- Un conjunto de prácticas por cada una de las tres evaluaciones. Para estimular el hábito del trabajo diario en los alumnos, se plantearán ejercicios, que se realizarán de forma individual y se resolverán en clase durante los días siguientes. Para estimular el hábito de trabajo en grupo, se podrán plantear tareas prácticas individuales y/o de grupo. Las prácticas que el profesor indique como obligatorias, deberán ser entregadas dentro del plazo y forma establecida por el profesor.
- **Observación directa del profesor**, sobre la iniciativa, interés, grado de participación y socialización, y adaptación a las normas de comportamiento y sobre los hábitos de trabajo: rigor, limpieza y puntualidad de las presentaciones, y adaptación a las normas de presentación de los trabajos solicitados.

- Una prueba de recuperación por cada evaluación suspensa que tenga el alumno. La recuperación de la 3ª evaluación quedará inmersa en la prueba final ordinaria mencionada más adelante. La nota de una recuperación tendrá como máximo carácter de suficiencia, es decir, 5.
- Un **proyecto globalizador** (con carácter excepcional, sólo a alumnos que no hayan superado la convocatoria ordinaria y a aquellos alumnos indicados por el profesor). Según criterio del profesor, y como complemento a las actividades de recuperación, se podrá proponer al alumno la realización de un proyecto globalizador. Podrá tener carácter individual o de pequeño grupo (se establecerá a criterio del profesor), que recogerá conceptos y procedimientos impartidos durante el curso, y que será necesario entregar en tiempo y forma indicado por el profesor así como y obtener calificación positiva para optar al resto de convocatorias.
- Una **prueba individual final ordinaria y/o extraordinaria** del módulo, que realizarán los alumnos que no hayan superado las pruebas de evaluación. Esta prueba final se basará en los contenidos mínimos o básicos del módulo que marca esta programación, realizándose por escrito o en ordenador, a criterio del profesor.

En junio habrá una prueba final ordinaria a la que tendrán que presentarse los alumnos que no hayan superado alguna de las 3 evaluaciones. Dicha prueba, una por evaluación, consistirá en un examen que estará dividido en pruebas teórico-prácticas en formato de papel y/o ordenador para aquellos alumnos que tengan evaluaciones pendientes siendo necesario obtener una calificación de 5 para poder superar cada evaluación en cada una de las evaluaciones pendientes.

Se convocará una prueba final extraordinaria a aquellos alumnos que no hayan aprobado todas las evaluaciones debiendo examinarse de la materia completa. La prueba extraordinaria consistirá en una prueba global que evaluará los contenidos de todo el curso siendo necesario obtener una calificación de 5 para poder superar el módulo.

Los instrumentos que se van a utilizar para llevar a cabo las evaluaciones son:

- Observación directa del profesor: intervenciones, motivación, asistencia, actitud e interés en el trabajo diario.
- Valoración de los trabajos realizados: ejercicios indicados en clase, ejercicios propuestos, tareas, proyectos si los hubiera, etc.
- Pruebas orales y escritas sobre los distintos contenidos.

- Realización de prácticas: éstas deberán ser realizadas con aprovechamiento durante cada uno de los días y horas en el aula o fuera del aula marcados para ello. Por su especial importancia y por motivos metodológicos y de seguimiento de los avances del alumno, estas prácticas no podrán sustituirse por examen escrito alguno.
- Al finalizar cada trimestre se realizara una prueba teorico-práctica objetiva bien escrita o bien a ordenador en la que el alumno deberá demostrar la consecución de los conocimientos mínimos exigidos para ese bloque de Unidades de Trabajo.

7. CRITERIOS DE CALIFICACIÓN

Las infracciones al RRI del centro serán corregidas y sancionadas de acuerdo al mismo.

- El proceso de evaluación será continuo y por lo tanto requiere la asistencia regular de los alumnos a clase y a las actividades programadas para el módulo.
- Se considera excluido del derecho de presentación a exámenes parciales y evaluación por trimestre al alumno cuyas faltas, no justificadas debidamente, superen el 15% del total de horas del módulo.
- Los alumnos que pierdan el derecho a evaluación continua deberán presentarse al examen final ordinario completo.
- La evaluación de los aprendizajes de los alumnos se realizará tomando como referencia las capacidades terminales y los criterios de evaluación del módulo, teniendo en cuenta los resultados mínimos que deben ser alcanzados en el proceso de aprendizaje.
- La evaluación se realizará valorando:

5% : Asistencia y evaluación de Competencias Personales, Profesionales y Sociales.

10% : Resolución de casos prácticos en clase, ya sea de forma individual o colectiva y entrega de trabajos propuestos por el profesor, en la fecha por él indicada.

85% : Exámenes individuales teórico-prácticos.

- Se realizará al menos un examen teórico-práctico en cada uno de los 3 trimestres de que consta el curso. Cada uno de dichos exámenes versará sobre los contenidos impartidos en el trimestre correspondiente y se puntuará de 0 a 10, siendo necesario tener una nota de 5 para que se tengan en cuenta el resto de los apartados del párrafo anterior.

- La nota final de los alumnos que hayan superado los tres trimestres según las normas antes expuestas, se obtendrá con una proporción del 20%, 40% y 40% para la 1ª, 2ª y 3ª evaluación respectivamente. Para optar a hacer dicha media se deben haber aprobado de forma individual cada una de las tres evaluaciones, si alguna evaluación no hubiera sido superada no se da por aprobado el curso.
- Para el primer y segundo trimestre, en caso de no aprobar la prueba parcial, se planteará una recuperación pudiendo obtenerse como máximo un 5. Para el tercer trimestre, la prueba de recuperación se planteará integrada en el examen final ordinario de junio, debiendo examinarse de dicha parte y pudiendo obtener en ella como máximo un 5.
- Aquellos alumnos que no hayan superado alguno de los trimestres deberán realizar el examen final ordinario (junio) con la materia que les corresponda. Asimismo podrán realizar ese examen final aquellos alumnos que hayan perdido el derecho de presentación a exámenes parciales y evaluación por trimestre.
- Para los alumnos que hayan tenido que hacer el examen final de junio (convocatoria ordinaria) íntegro, la calificación final será la obtenida en dicho examen, siendo condición indispensable para aprobar haber obtenido al menos un 5 en cada una de las partes, y en el caso de que tengan alguna evaluación aprobada previamente, la nota media será calculada promediando la/s evaluación/es aprobada/s con el resto de la/s parte/s examinada/s en el examen final de junio, donde la máxima nota obtenible en la parte evaluada de dicho examen será de un 5.
- En el caso en que el alumno no haya superado el módulo tras la sesión ordinaria de evaluación, deberá presentarse a la **prueba extraordinaria** con toda la materia expuesta en clase durante el curso. Dicha prueba extraordinaria tendrá que ser superada con una nota igual ó superior a un cinco. La nota de esta recuperación será la obtenida en el examen, siendo necesario al menos la calificación de 5 para aprobar.
- En cada prueba, tanto de carácter ordinario como extraordinario, preferentemente se especificará el valor de cada pregunta y para que dicha pregunta se acumule a la nota de la prueba, se debe responder en su totalidad a las exigencias en ella planteadas, teniendo en cuenta que si fuera de carácter práctico, la respuesta, por parte del alumno, debe satisfacer completamente los requerimientos planteados, ya que de lo contrario no se acumularán los puntos correspondientes a dicha pregunta en la prueba.
- Aquel o aquellos alumnos que copien o ayuden a copiar en alguna de las pruebas realizadas serán calificados en la evaluación correspondiente con un uno.

8. PROCEDIMIENTO DE RECUPERACIÓN DE EVALUACIONES PENDIENTES

Tal y como se menciona en el apartado anterior:

- Para el primer y segundo trimestre, en caso de no aprobar la prueba parcial, se planteará una recuperación pudiendo obtenerse como máximo un 5. Para el tercer trimestre, la prueba de recuperación se planteará integrada en examen final ordinario de junio en los términos que se indican en secciones anteriores.
- Aquellos alumnos que no hayan superado alguno de los trimestres deberán realizar el examen final ordinario (junio) con la materia que les corresponda. Asimismo podrán realizar ese examen final aquellos alumnos que hayan perdido el derecho de presentación a exámenes parciales y evaluación por trimestre.
- Para los alumnos que hayan tenido que hacer el examen final de junio (convocatoria ordinaria) íntegro, la calificación final será la obtenida en dicho examen, siendo condición indispensable para aprobar haber obtenido al menos un 5 en cada una de las partes, y en el caso de que tengan alguna evaluación aprobada previamente, la nota media será calculada promediando la/s evaluación/es aprobada/s con el resto de la/s parte/s examinada/s en el examen final de junio, donde la máxima nota obtenible en la parte evaluada de dicho examen será de un 5.
- En el caso en que el alumno no haya superado el módulo tras la sesión ordinaria de evaluación, deberá presentarse a la **prueba extraordinaria** con toda la materia expuesta en clase durante el curso. Dicha prueba extraordinaria tendrá que ser superada con una nota igual ó superior a un cinco. La nota de esta recuperación será la obtenida en el examen, siendo necesario al menos la calificación de 5 para aprobar. La máxima nota posible para éste examen extraordinario se establece en un 7.

9. PROCEDIMIENTOS Y ACTIVIDADES DE RECUPERACIÓN PARA LOS ALUMNOS CON MÓDULOS PENDIENTES DE CURSOS ANTERIORES

Si fuera necesario, para alumnos que hayan promocionado a segundo curso del ciclo formativo de grado superior de Desarrollo de Aplicaciones Multiplataforma o del ciclo formativo de grado superior de Desarrollo de Aplicaciones Web, se contempla la posibilidad de proporcionar clases de apoyo o refuerzo. No obstante, estos alumnos, si los hubiere, dispondrán de 2 convocatorias para poder superar el módulo. Se prevé que el examen correspondiente a la convocatoria ordinaria será en febrero, mientras que el examen de la convocatoria extraordinaria será en mayo. En todo caso se

pondrá en conocimiento de los alumnos interesados la fecha, hora y lugar en que se celebrarán dichas pruebas, con antelación suficiente.

10. EVALUACIÓN EXTRAORDINARIA

En el caso en que el alumno no haya superado el módulo tras la prueba ordinaria de evaluación, deberá presentarse a la prueba extraordinaria con toda la materia expuesta en clase durante el curso. Dicha prueba extraordinaria tendrá que ser superada con una nota igual ó superior a un cinco. Dado el carácter aglutinador e incremental de la materia, las pruebas final ordinaria y final extraordinaria pueden no distinguir explícitamente los trimestres a que corresponden.

Dicha prueba se realizará en junio siguiendo el modelo del examen final ordinario, pudiendo constar de contenido teórico y práctico. También se podrá contemplar de manera adicional como actividad, el desarrollo práctico de un proyecto, o bien solicitar las actividades que no haya llevado a cabo durante el curso o alguna actividad extraordinaria.

La calificación final se registrará por la misma norma que para la evaluación Ordinaria. Se establece un 7 como máxima nota para la prueba de evaluación extraordinaria.

La evaluación de los conocimientos de los alumnos y su calificación se realizará tomando como referencias los resultados de aprendizaje del módulo y los criterios de evaluación publicados en el Real Decreto El **REAL DECRETO 450/2010**, de 16 de Abril, por el que se establece el **título de Técnico en Desarrollo de Aplicaciones Multiplataforma** y se fijan sus **enseñanzas mínimas** y El **REAL DECRETO 686/2010**, de 20 de Mayo , por el que se establece el **título** de Técnico en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas **mínimas**.

Las fechas de realización de la pruebas será publicadas en el tablón del Centro y/o de Departamento, o bien a través de página web, con suficiente antelación para que el alumno pueda preparar la materia.

11. CRITERIOS Y PROCEDIMIENTOS PREVISTOS PARA LA ORGANIZAR LA ATENCIÓN A LA DIVERSIDAD DE LOS ALUMNOS

El profesor mediante sus referencias vigilará las características personales de cada alumno con el objetivo de conseguir que todos adquieran unos conocimientos mínimos del módulo con el que se trabaja.

Velando por este objetivo, el profesor podrá preparar prácticas especiales para aquellos alumnos que necesiten una mayor motivación, más profundización o insistencia en la materia sin olvidar al resto, proponiendo

alguna labor de estudio adicional o trabajos de investigación para los alumnos más aventajados, así :

- Se efectuarán explicaciones particulares y apoyo personal en los trabajos a aquellos alumnos que así lo requieran.
- Se propondrán actividades de ampliación para aquellos alumnos más aventajados

12. PROCEDIMIENTO PARA QUE EL ALUMNADO Y, EN SU CASO LAS FAMILIAS (PORQUE EL ALUMNO SEA MENOR DE EDAD), CONOZCAN LOS OBJETIVOS

Desde el contacto directo en el aula el profesor hará conocer a los alumnos/as información recogida en esta Programación como son los contenidos, los criterios de evaluación, los criterios de calificación y los procedimientos de evaluación.

El número de faltas de asistencia no justificadas que dan lugar a la anulación de la matrícula será notificada mediante correo al alumno/familias.

13. COORDINACION ENTRE PROFESORES QUE IMPARTAN EL MISMO CICLO

Se establecerá un día de reunión de Departamento que entre otras cosas, coordinará los diferentes aspectos pedagógicos y organizativos para el correcto funcionamiento del Ciclo, pudiendo proponerse reuniones específicas del equipo docente de un determinado ciclo y curso.

Se propone la misma Programación Didáctica para el módulo Programación de los ciclos de Desarrollo de Aplicaciones Multiplataforma (turno diurno) y Desarrollo de Aplicaciones Web (turno vespertino). Es por esto, por lo que se establecerán canales de comunicación como reuniones o vía e-mail entre los profesores de ambos turnos, que garantizarán que los contenidos impartidos y la temporalización sean similares dentro de lo posible. Estas reuniones se harán coincidir, preferentemente, con el día asignado para la reunión de Departamento.

En ambos casos se utilizarán los mismos criterios de calificación que han quedado expresados en esta programación.

14.ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES

Se contempla la posibilidad de realizar diferentes actividades relacionadas con el mundo de la programación del tipo de las realizadas en cursos anteriores como :

- “La Hora del Código”
- “Prográmame”
- Se está estudiando la posibilidad de realizar un viaje a Estrasburgo dentro del marco de las actividades de la Comunidad Europea.
- otras posibles vinculadas al ámbito informático o de interés.

15.EVALUACIÓN DE LA PRACTICA DOCENTE

Con el objeto de valorar el desarrollo óptimo de la programación, el grado de cumplimiento de los objetivos, la eficacia de la practica docente propia y los procesos de enseñanza del Modulo específico, se realizara una recogida de información, trimestralmente, a través de un cuestionario, así como la recepción de sugerencias por parte del alumnado y pequeño debate con conclusiones para el siguiente trimestre.

Con el objeto de que la programación pueda seguirse de forma clara y sin confusión por los profesores sustitutos, en caso de enfermedad, ausencias imprevistas, permisos retribuidos, etc. el profesor titular lleva un registro actualizado de los conocimientos impartidos en cada grupo, ejercicios realizados, así como detalles del proceso de evaluación diario. Bastara con que el profesor sustituto acceda a este cuaderno de registro para que pueda seguir la programación sin duda.

El seguimiento de la programación, se hará de forma continua, en las reuniones de departamento de las de las que se lleva acta mensual.

16.ANEXO

El alumno con un 15% o más de faltas del total de horas del módulo puede ocurrir en la pérdida de evaluación ordinaria así como otras sanciones de aplicación en el centro.

1º) Los equipos pedagógicos podrán decidir, de forma justificada, qué alumnos no promocionarán a la FCT, o al periodo de prácticas de cualquier modalidad, si consideran que no se han alcanzado las competencias personales y sociales que permitan al alumno su integración total en el ámbito laboral durante sus prácticas.

2º) Situaciones o actuaciones que darán lugar a la exclusión o calificación de “No apto” en la FCT o Prácticas en empresas. :

- No realizar el total de horas de FCT o prácticas del ciclo, incluidas las horas de tutoría en el Instituto. El total de faltas no podrá exceder de un determinado porcentaje respecto del número total de horas de FCT:
- Ausencias justificadas: 10%
- Ausencias no justificadas: 2%
- Suma de faltas justificadas y no justificadas: $\leq 10\%$
- Se considerarán justificadas las faltas por enfermedad o deber inexcusable de carácter público o privado. La justificación deberá estar debidamente acreditada y será valorada por el Profesor Tutor.
- Retrasos reiterados e injustificados.
- Incumplimiento del horario.
- Actitud negativa o indiferente en el desempeño de las tareas del puesto formativo.
- Negociación o pacto con la empresa de condiciones propias (permisos, cambios de horario...) sin el conocimiento y aprobación del Profesor Tutor.
- No presentar al Profesor Tutor las hojas semanales con las actividades desarrolladas y el visto bueno del Tutor de la Empresa en los plazos señalados.
- Informe negativo del tutor de la Empresa sobre su comportamiento y/o actitud.