

PROGRAMACION DEL MODULO PROFESIONAL:
PROGRAMACIÓN

Ciclo: *Desarrollo de Aplicaciones Web*

CURSO 2016/17

I.E.S. Doménico Scarlatti, Aranjuez

INTRODUCCIÓN	5
RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN	7
CONTENIDOS DEL MÓDULO	11
CONTENIDOS GENERALES	11
CONTENIDOS MÍNIMOS	14
RELACIÓN SECUENCIADA DE LAS UNIDADES DE TRABAJO	16
ESPECIFICACIÓN Y TEMPORALIZACIÓN DE LAS UNIDADES DE TRABAJO	19
U.T.1: INTRODUCCIÓN A LA PROGRAMACIÓN:	19
U.T.2: INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS:	19
U.T.3: IDENTIFICACIÓN DE LOS ELEMENTOS DE UN PROGRAMA INFORMÁTICO:	20
U.T.4: UTILIZACIÓN DE OBJETOS:	20
U.T.5: USO DE ESTRUCTURAS DE CONTROL:	20
U.T.6: DESARROLLO DE CLASES:	21
U.T.7: APLICACIÓN DE LAS ESTRUCTURAS DE ALMACENAMIENTO:	22
U.T.8: UTILIZACIÓN AVANZADA DE CLASES:	22
U.T.9: CONTROL Y MANEJO DE EXCEPCIONES:	23
U.T.10: COLECCIONES DE DATOS:	23
U.T.11: LECTURA Y ESCRITURA DE INFORMACIÓN:	23
U.T.12: GESTIÓN DE BASES DE DATOS RELACIONALES:	24
U.T.13: MANTENIMIENTO DE LA PERSISTENCIA DE LOS OBJETOS:	25
TEMPORALIZACIÓN	27
METODOLOGÍA Y MATERIALES DIDÁCTICOS	28
SISTEMA DE EVALUACIÓN	29
SENTIDO DE LA EVALUACIÓN	29
CRITERIOS DE EVALUACIÓN	29
MOMENTOS DE LA EVALUACIÓN	29
INSTRUMENTOS O PRUEBAS DE EVALUACIÓN	30
CALIFICACIÓN	30

Programación - DAW

EVALUACIONES TRIMESTRALES

¡ERROR! MARCADOR NO DEFINIDO.

EVALUACIÓN ORDINARIA DE JUNIO	31
CALIFICACIÓN FINAL	32
EVALUACIÓN EXTRAORDINARIA DE SEPTIEMBRE	32
PÉRDIDA DE LA EVALUACIÓN CONTINUA	32
ALUMNOS CON EL MÓDULO PENDIENTE	32
RECUPERACIÓN	32
PROACCIÓN	33
OBJETIVOS CUANTITATIVOS	33
EVALUACIÓN DE LA PRÁCTICA DOCENTE	33
<u>PROCEDIMIENTO PARA QUE EL ALUMNO CONOZCA ESTA PROGRAMACIÓN.</u>	<u>34</u>
<u>BIBLIOGRAFÍA</u>	<u>36</u>

Introducción

Este módulo de 270 horas se encuadra en el primer curso del Ciclo formativo de grado superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web.

Este módulo profesional contiene parte de la formación necesaria para desempeñar:

1. La función de programación de aplicaciones de propósito general en lenguajes orientados a objetos.
2. La función de programación de aplicaciones de propósito general en lenguajes orientados a objetos incluye aspectos como:
3. El desarrollo de programas organizados en clases aplicando los principios de la programación orientada a objetos.
4. La utilización de interfaces para la interacción de la aplicación con el usuario.
5. La identificación, análisis e integración de librerías para incorporar funcionalidades específicas a los programas desarrollados.
6. El almacenamiento y recuperación de información en sistemas gestores de bases de datos relacionales y orientados a objetos.

Las actividades profesionales asociadas a esta función se aplican en el desarrollo y la adaptación de programas informáticos de propósito general en lenguajes orientados a objetos.

La formación del módulo contribuye a alcanzar los objetivos generales:

1. Interpretar el diseño lógico, verificando los parámetros establecidos para gestionar bases de datos.
2. Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
3. Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.

Las competencias profesionales que guiarán el aprendizaje en el módulo serán:

1. Configurar y explotar sistemas informáticos, adaptando la configuración lógica del sistema según las necesidades de uso y los criterios establecidos.
2. Desarrollar aplicaciones web con acceso a bases de datos utilizando lenguajes, objetos de acceso y herramientas de mapeo adecuados a las especificaciones.
3. Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.
4. Integrar componentes multimedia en el interface de una aplicación web, realizando el análisis de interactividad, accesibilidad y usabilidad de la aplicación.
5. Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación.
6. Realizar la gestión básica para la creación y funcionamiento de una pequeña empresa y tener iniciativa en su actividad profesional con sentido de la responsabilidad social.

Las líneas de actuación en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo versarán sobre:

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

Especificados en el *Real Decreto 686/2010, de 20 de mayo* del Boletín Oficial del Estado citado anteriormente.

Esta programación se ajustará a la legislación emitida por el Ministerio de Educación y por la Consejería de Educación de la Comunidad de Madrid:

1. *Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas.*
2. *Orden EDU/2887/2010, de 2 de noviembre, por la que se establece el currículo del ciclo formativo de Grado Superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web.*
3. *DECRETO 1/2011, de 13 de enero, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo del ciclo formativo de grado superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web.*

Resultados de aprendizaje y criterios de evaluación

Los objetivos didácticos son resultados que se espera que consigan los estudiantes como consecuencia de la formación. Concretan los componentes o elementos de competencias a adquirir en términos de resultados de aprendizaje: qué habilidades adquirir, qué actitudes desarrollar y qué conocimientos aprender. El elemento central es la habilidad que se espera desarrollar en los alumnos.

Los resultados de aprendizaje se derivan de los objetivos didácticos y facilitan la tarea del profesor en la programación y orienta el aprendizaje del alumno. Determinan qué deberán de ser capaces de hacer una vez acabado el aprendizaje, en qué condiciones deberán hacerlo y sugieren criterios que se pueden utilizar para valorar lo aprendido. Asimismo, guían la elección de las estrategias y actividades de aprendizaje, del método de enseñanza y del sistema de evaluación.

Las intenciones educativas, la formación profesional pretendida, las habilidades a adquirir, los contenidos a aprender y las actitudes a desarrollar, dan la medida de la diversidad de objetivos que deberían estar presentes en un programa.

En un contexto didáctico, todos los objetivos implican evaluación. Es decir, debe valorarse si se han logrado o no las pretensiones que se tenían al comenzar el proceso formativo.

De esta forma, los resultados de aprendizaje especificados en la legislación vigente para el módulo profesional tratado y sus correspondientes criterios de evaluación son:

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

Criterios de evaluación:

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

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

Criterios de evaluación:

- a. Se han identificado los fundamentos de la programación orientada a objetos.
 - b. Se han escrito programas simples.
 - c. Se han instanciado objetos a partir de clases predefinidas.
 - d. Se han utilizado métodos y propiedades de los objetos.
 - e. Se han escrito llamadas a métodos estáticos.
 - f. Se han utilizado parámetros en la llamada a métodos.
 - g. Se han incorporado y utilizado librerías de objetos.
 - h. Se han utilizado constructores.
 - i. Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.
3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje

Criterios de evaluación:

- a. Se ha escrito y probado código que haga uso de estructuras de selección.
 - b. Se han utilizado estructuras de repetición. ☐
 - c. Se han reconocido las posibilidades de las sentencias de salto. ☐
 - d. Se ha escrito código utilizando control de excepciones.
 - e. Se han creado programas ejecutables utilizando diferentes estructuras de control. ☐
 - f. Se han probado y depurado los programas. ☐
 - g. Se ha comentado y documentado el código.
4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

Criterios de evaluación:

- a. Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
 - b. Se han definido clases.
 - c. Se han definido propiedades y métodos.
 - d. Se han creado constructores.
 - e. Se han desarrollado programas que instancien y utilicen objetos de las clases ☐ creadas anteriormente.
 - f. Se han utilizado mecanismos para controlar la visibilidad de las clases y de ☐ sus miembros.
 - g. Se han definido y utilizado clases heredadas. ☐
 - h. Se han creado y utilizado métodos estáticos. ☐
 - i. Se han definido y utilizado interfaces.
 - j. Se han creado y utilizado conjuntos y librerías de clases.
5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Criterios de evaluación:

- a. Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
 - b. Se han aplicado formatos en la visualización de la información.
 - c. Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
 - d. Se han utilizado ficheros para almacenar y recuperar información.
 - e. Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
 - f. Se han utilizado las herramientas del entorno de desarrollo para crear interfaces 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.
6. Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.

Criterios de Evaluación:

- a. Se han escrito programas que utilicen arrays. ☐
 - b. Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados. ☐
 - c. Se han utilizado listas para almacenar y procesar información. ☐
 - d. Se han utilizado iteradores para recorrer los elementos de las listas. ☐
 - e. Se han reconocido las características y ventajas de cada una de la colecciones de datos disponibles. ☐
 - f. Se han creado clases y métodos genéricos. ☐
 - g. Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.
 - h. Se han identificado las clases relacionadas con el tratamiento de documentos XML. ☐
 - i. Se han realizado programas que realicen manipulaciones sobre documentos XML.
7. Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

Criterios de evaluación:

- a. Se han identificado los conceptos de herencia, superclase y subclase.
 - b. Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
 - c. Se ha reconocido la incidencia de los constructores en la herencia.
 - d. Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.
 - e. Se han diseñado y aplicado jerarquías de clases.
 - f. Se han probado y depurado las jerarquías de clases.
 - g. Se han realizado programas que implementen y utilicen jerarquías de
 - h. clases.
 - i. Se ha comentado y documentado el código.
8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

Criterios de evaluación:

- a. Se han identificado las características de las bases de datos orientadas a objetos.
 - b. Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
 - c. Se han instalado sistemas gestores de bases de datos orientados a objetos.
 - d. Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
 - e. Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
 - f. Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
 - g. Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
 - h. Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.
9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos.

Criterios de evaluación:

- a. Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
- b. Se han programado conexiones con bases de datos.
- c. Se ha escrito código para almacenar información en bases de datos.
- d. Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
- e. Se han efectuado borrados y modificaciones sobre la información almacenada.
- f. Se han creado aplicaciones que ejecuten consultas sobre bases de datos.
- g. Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

Contenidos del módulo

Contenidos generales

Los contenidos generales definidos en el DECRETO 1/2011, de 13 de enero, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo del ciclo formativo de grado superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web son:

Introducción a la programación:

- Datos, algoritmos y programas.
- Paradigmas de programación.
- Lenguajes de programación.
- Herramientas y entornos para el desarrollo de programas.
- Errores y calidad de los programas.

Introducción a la orientación a objetos:

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

Identificación de los elementos de un programa informático:

- 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.

☐ Utilización de objetos:

- 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. ☐

Uso de estructuras de control:

- Estructuras de selección.

- Estructuras de repetición.
- Estructuras de salto.

Desarrollo de clases:

- Concepto de clase.
- Estructura y miembros de una clase.
- Creación de atributos. Declaración e inicialización.
- Creación de métodos. Declaración, argumentos y valores de retorno.
- Creación de 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.
- Utilización de clases y objetos.
- Utilización de clases heredadas.
- Librerías y paquetes de clases. Utilización y creación.
- Documentación sobre librerías y paquetes de clases.

Aplicación de las estructuras de almacenamiento:

- 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.

Utilización avanzada de clases:

- Relaciones entre clases. Composición de clases.
- Herencia. Concepto y tipos (simple y múltiple).
- Superclases y subclases.
- Constructores y herencia.
- Modificadores en clases, atributos y métodos.
- Sobreescritura de métodos.
- Clases y métodos abstractos y finales.
- Interfaces. Clases abstractas vs. Interfaces.
- Polimorfismo:
 - Concepto.
 - Polimorfismo en tiempo de compilación (sobrecarga) y polimorfismo en tiempo de ejecución (ligadura dinámica).
 - Comprobación estática y dinámica de tipos.
- Conversiones de tipos entre objetos (casting).
- Clases y tipos genéricos o parametrizados. □

Control y manejo de excepciones:

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

Colecciones de datos:

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

Lectura y escritura de información:

- 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
- Interfaces gráficos de usuario simples. Concepto de evento. Creación de controladores de eventos.

Gestión de bases de datos relacionales:

- 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.

Mantenimiento de la persistencia de los objetos:

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

- Tipos de datos objeto; atributos y métodos.
- Tipos de datos colección.

Contenidos mínimos

Los contenidos básicos definidos en el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas son:

1. 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.

2. Utilización de objetos:

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

3. Uso de estructuras de control:

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

- Control de excepciones.

4. Desarrollo de clases:

- Concepto de clase.
- Estructura y miembros de una clase.
- Creación de atributos.
- Creación de métodos.
- Creación de constructores.
- Utilización de clases y objetos.
- Utilización de clases heredadas.

5. 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. □

6. Aplicación de las estructuras de almacenamiento:

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

7. ☐ Utilización avanzada de clases:

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

8. Mantenimiento de la persistencia de los objetos:

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

9. ☐ Gestión de bases de datos relacionales:

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

Relación secuenciada de las unidades de trabajo

Teniendo en cuenta toda la información anterior así como las orientaciones pedagógicas detalladas en el *Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas*:

Este módulo profesional contiene parte de la formación necesaria para desempeñar la función de programación de aplicaciones de propósito general en lenguajes orientados a objetos.

La función de programación de aplicaciones de propósito general en lenguajes orientados a objetos incluye aspectos como:

El desarrollo de programas organizados en clases aplicando los principios de la programación orientada a objetos. □ La utilización de interfaces para la interacción de la aplicación con el usuario.

La identificación, análisis e integración de librerías para incorporar funcionalidades específicas a los programas desarrollados.

El almacenamiento y recuperación de información en sistemas gestores de bases de datos relacionales y orientados a objetos.

Las actividades profesionales asociadas a esta función se aplican en el desarrollo y la adaptación de programas informáticos de propósito general en lenguajes orientados a objetos.

Las líneas de actuación en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo versarán sobre:

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

Todo ello queda concretado en la siguiente relación secuenciada de unidades de trabajo:

U.T.1: Introducción a la programación.

U.T.2: Introducción a la orientación a objetos.

U.T.3: Identificación de los elementos de un programa informático.

U.T.4: Utilización de objetos.

U.T.5: Uso de estructuras de control.

U.T.6: Desarrollo de clases.

U.T.7: Aplicación de las estructuras de almacenamiento.

U.T.8: Utilización avanzada de clases.

U.T.9: Control y manejo de excepciones.

U.T.10: Colecciones de datos.

U.T.11: Lectura y escritura de información.

U.T.12: Gestión de bases de datos relacionales.

U.T.13: Mantenimiento de la persistencia de los objetos.

Especificación y temporalización de las Unidades de Trabajo

En este apartado se detalla la división de contenidos en Unidades de Trabajo así como la temporalización que se aplicará en la docencia directa en el aula.

U.T.1: Introducción a la programación:

- Datos, algoritmos y programas.
- Paradigmas de programación.
- Lenguajes de programación.
- Herramientas y entornos para el desarrollo de programas.
- Errores y calidad de los programas.

Que se corresponde con el resultado de aprendizaje 1: *Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.*

Criterios de evaluación:

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

U.T.2: Introducción a la orientación a objetos:

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

Que se corresponde con el resultado de aprendizaje 2: *Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.*

Criterios de evaluación:

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

U.T.3: Identificación de los elementos de un programa informático:

- 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.

Que al igual que la U.T.1 se corresponde con el resultado de aprendizaje 1: *Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.*

U.T.4: Utilización de objetos:

- 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. □

Que al igual que la U.T.2 se corresponde con el resultado de aprendizaje 2: *Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.*

U.T.5: Uso de estructuras de control:

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

Que se corresponde con el resultado de aprendizaje 3: *Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje*

Criterios de evaluación:

- a. Se ha escrito y probado código que haga uso de estructuras de selección.
- b. Se han utilizado estructuras de repetición. ☐
- c. Se han reconocido las posibilidades de las sentencias de salto. ☐
- d. Se ha escrito código utilizando control de excepciones.
- e. Se han creado programas ejecutables utilizando diferentes estructuras de control. ☐
- f. Se han probado y depurado los programas. ☐
- g. Se ha comentado y documentado el código.

U.T.6: Desarrollo de clases:

- Concepto de clase.
- Estructura y miembros de una clase.
- Creación de atributos. Declaración e inicialización.
- Creación de métodos. Declaración, argumentos y valores de retorno.
- Creación de 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.
- Utilización de clases y objetos.
- Utilización de clases heredadas.
- Librerías y paquetes de clases. Utilización y creación.
- Documentación sobre librerías y paquetes de clases.

Que se corresponde con el resultado de aprendizaje 4: *Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.*

Criterios de evaluación:

- a. Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- b. Se han definido clases.
- c. Se han definido propiedades y métodos.
- d. Se han creado constructores.
- e. Se han desarrollado programas que instancien y utilicen objetos de las clases ☐ creadas anteriormente.
- f. Se han utilizado mecanismos para controlar la visibilidad de las clases y de ☐ sus miembros.
- g. Se han definido y utilizado clases heredadas. ☐
- h. Se han creado y utilizado métodos estáticos. ☐
- i. Se han definido y utilizado interfaces. ☐
- j. Se han creado y utilizado conjuntos y librerías de clases.

U.T.7: Aplicación de las estructuras de almacenamiento:

- 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.

Que se corresponde con el resultado de aprendizaje 6: *Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.*

Criterios de Evaluación:

- a. Se han escrito programas que utilicen arrays. ☐
- b. Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados. ☐
- c. Se han utilizado listas para almacenar y procesar información. ☐
- d. Se han utilizado iteradores para recorrer los elementos de las listas. ☐
- e. Se han reconocido las características y ventajas de cada una de la colecciones de datos disponibles. ☐
- f. Se han creado clases y métodos genéricos. ☐
- g. Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.
- h. Se han identificado las clases relacionadas con el tratamiento de documentos XML. ☐
- i. Se han realizado programas que realicen manipulaciones sobre documentos XML.

U.T.8: Utilización avanzada de clases:

- Relaciones entre clases. Composición de clases.
- Herencia. Concepto y tipos (simple y múltiple).
- Superclases y subclases.
- Constructores y herencia.
- Modificadores en clases, atributos y métodos.
- Sobreescritura de métodos.
- Clases y métodos abstractos y finales.
- Interfaces. Clases abstractas vs. Interfaces.
- Polimorfismo:
 - Concepto.
 - Polimorfismo en tiempo de compilación (sobrecarga) y polimorfismo en tiempo de ejecución (ligadura dinámica).

- Comprobación estática y dinámica de tipos.
- Conversiones de tipos entre objetos (casting).
- Clases y tipos genéricos o parametrizados. ☐

Que se corresponde con el resultado de aprendizaje 7: *Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.*

Criterios de evaluación:

- a. Se han identificado los conceptos de herencia, superclase y subclase.
- b. Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
- c. Se ha reconocido la incidencia de los constructores en la herencia.
- d. Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.
- e. Se han diseñado y aplicado jerarquías de clases.
- f. Se han probado y depurado las jerarquías de clases.
- g. Se han realizado programas que implementen y utilicen jerarquías de
- h. clases.
- i. Se ha comentado y documentado el código.

U.T.9: Control y manejo de excepciones:

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

Que no se corresponde directamente con ningún resultado de aprendizaje.

U.T.10: Colecciones de datos:

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

Que al igual que la U.T.7 se corresponde con el resultado de aprendizaje 6.

U.T.11: Lectura y escritura de información:

- 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
- Interfaces gráficos de usuario simples. Concepto de evento. Creación de controladores de eventos.

Que se corresponde con el resultado de aprendizaje 5: *Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.*

Criterios de evaluación:

- a. Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- b. Se han aplicado formatos en la visualización de la información.
- c. Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
- d. Se han utilizado ficheros para almacenar y recuperar información.
- e. Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- f. Se han utilizado las herramientas del entorno de desarrollo para crear interfaces 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.

U.T.12: Gestión de bases de datos relacionales:

- 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.

Que se corresponde con el resultado de aprendizaje 8: *Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.*

Criterios de evaluación:

- a. Se han identificado las características de las bases de datos orientadas a objetos.
- b. Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
- c. Se han instalado sistemas gestores de bases de datos orientados a objetos.
- d. Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- e. Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- f. Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
- g. Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- h. Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

U.T.13: Mantenimiento de la persistencia de los objetos:

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

Que se corresponde con el resultado de aprendizaje 9: *Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos.*

Criterios de evaluación:

- a. Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
- b. Se han programado conexiones con bases de datos.
- c. Se ha escrito código para almacenar información en bases de datos.
- d. Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
- e. Se han efectuado borrados y modificaciones sobre la información almacenada.

- f. Se han creado aplicaciones que ejecuten consultas sobre bases de datos.
- g. Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

Temporalización

Las unidades de trabajo indicadas quedarán encuadradas en las tres evaluaciones de la siguiente manera:

Primera evaluación

U.T.1: Introducción a la programación.

U.T.2: Introducción a la orientación a objetos.

U.T.3: Identificación de los elementos de un programa informático.

U.T.4: Utilización de objetos.

U.T.5: Uso de estructuras de control.

Segunda evaluación

U.T.6: Desarrollo de clases.

U.T.7: Aplicación de las estructuras de almacenamiento.

U.T.8: Utilización avanzada de clases.

U.T.9: Control y manejo de excepciones.

U.T.10: Colecciones de datos.

Tercera evaluación

U.T.11: Lectura y escritura de información.

U.T.12: Gestión de bases de datos relacionales.

U.T.13: Mantenimiento de la persistencia de los objetos.

Metodología y materiales didácticos

Las clases tendrán una parte teórica donde se impartirán los conceptos necesarios para el desarrollo del módulo y una parte práctica donde se propondrán ejercicios a los alumnos para que resuelvan en el ordenador. Los alumnos contarán con la ayuda del profesor para resolver las posibles dudas.

Como material didáctico se emplearán apuntes confeccionados por el profesor y la documentación existente en Internet así como ejemplos de programas. También se utilizará el cañón de vídeo conectado al ordenador del profesor para las explicaciones de entorno gráfico y la proyección de ejemplos.

Las prácticas en el aula se harán con ordenadores en un entorno Windows. Desarrollando en Netbeans.

Sistema de Evaluación

Sentido de la evaluación

El profesor, además de estimular, orientar y enseñar, ha de evaluar la calidad de los resultados obtenidos, para conocer la eficacia del esfuerzo realizado.

La evaluación se orienta a la calidad de los procesos de enseñanza/aprendizaje y tiene un carácter formativo; será educativa en la medida que pueda considerarse estimulante y orientadora. Cada miembro de la Comunidad educativa, especialmente los alumnos, han de capacitarse para hacer un diagnóstico válido de sus posibilidades y limitaciones, de su comportamiento, analizando las funciones y tareas que realiza y conociendo su propio progreso. (López-Barajas Zayas, 1995)

Criterios de evaluación

Los criterios de evaluación aparecen especificados junto al resultado de aprendizaje correspondiente tanto en la enumeración de los resultados de aprendizaje como en la especificación de las unidades de trabajo, por lo que no se repetirán aquí.

Los mínimos exigibles son los enumerados en el apartado citado en el párrafo anterior ya que son los especificados por la Comunidad Autónoma de Madrid.

Momentos de la evaluación

La evaluación del módulo formativo considerará:

Evaluación continua y formativa: Se tratará de llevar un seguimiento lo más intenso posible del proceso de aprendizaje seguido por cada alumno. De esta manera, será factible proponer, en el momento más adecuado, las actividades de refuerzo necesarias en cada caso para poder resolver los problemas detectados en el aprendizaje individual. La evaluación continua también permitirá al profesor detectar y modificar enfoques (objetivos, métodos, formas de enseñar y motivar) que no resulten acertados en el ejercicio de su práctica docente, reajustando en lo necesario la programación.

Evaluación final: También es necesaria, y se realizará al final de cada evaluación, y al final del curso. Permitirá obtener una visión global de los logros hallados. Habrá de realizarse de acuerdo a criterios didácticamente correctos: la evaluación está sometida más a criterio que a norma. En palabras de López-Barajas Zayas (1995) «estamos en una sociedad competitiva. Y, aun no estándolo algún día, es necesario conocer la situación del alumno respecto de su «clase». La evaluación sujeta a norma (grupo normativo) no se contrapone a la evaluación sujeta a criterio, ya que sirve de complemento informativo.»

Instrumentos o pruebas de evaluación

La evaluación será continua e individualizada, y la observación sistemática será un instrumento de evaluación habitual. Dada la complejidad de la evaluación, se utilizarán distintas técnicas para realizarla, ya que evaluar los aspectos cuantitativos y cualitativos de rendimiento con una sola forma resultaría siempre insuficiente. Los procedimientos de análisis irán desde los más estructurados (tests) a los menos estructurados («notas u observaciones de clase»). La interpretación de los datos y los análisis debe ser holista (totalista), debe considerarse en su conjunto; la percepción ha de ser de los aspectos cualitativos y fundamentales. (López-Barajas Zayas, 1995).

Se observará la actitud en clase (hacia los compañeros, profesores, y material; hacia la materia propia del módulo; etc.). Se analizarán los trabajos realizados en clase y en casa, los partes de faltas (la asistencia y la puntualidad) y posibles apercebimientos, cuestionarios individuales, preguntas abiertas para trabajo en grupo, trabajos entregados (manuales, guías, ...), cuaderno del alumno (compleción, presentación, limpieza, ...), pruebas objetivas, ...

Cerca del final de cada evaluación, habrá una prueba específica de evaluación.

Calificación

«A tenor de lo dispuesto en el artículo 27 de la *ORDEN 2694/2009, de 9 de junio* de la Comunidad de Madrid publicada en el B.O.C.M. del día 22 de junio de las calificaciones de los módulos profesionales que componen el ciclo formativo se expresarán en valores numéricos de 1 a 10, sin decimales. Se considerarán positivas las iguales o superiores a 5 y negativas las restantes.»

Bloques de Contenidos y Evaluaciones Trimestrales

La naturaleza del módulo hace que el contenido sea incremental como el de cualquier lenguaje, por ello se compone de un único bloque global y la evaluación se corresponderá con una evaluación continua.

Por ello la influencia de cada evaluación en la calificación final es como sigue:

	1ª Evaluación	2ª Evaluación	3ª Evaluación
Calificación final	20%	40%	40%

Al igual que la nota final del módulo, la de cada evaluación trimestral oscilará entre 1 y 10, dependiendo del grado de consecución de las actividades de enseñanza/aprendizaje, de la prueba específica de evaluación y de la actitud en clase. Si la calificación es 5 o superior a 5 se considerará aprobada la evaluación trimestral.

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

- Actividades de enseñanza/aprendizaje 30 %

- Prueba específica de evaluación 70 %

1. Actividades de enseñanza/aprendizaje: Son las actividades que el profesor declare como imprescindibles de cada evaluación. El profesor indicará el carácter individual o grupal de las mismas. Deberán ser realizadas y se entregarán de la forma que indique el profesor para cada una de ellas. Generalmente podrá solicitarse que el alumno corrija la práctica en clase o que entregue el resultado al profesor. Cada práctica será evaluada de 0 a 10 y al final de la evaluación, se calculará la nota media de todas ellas que contribuirá a nota de la evaluación, mediante el porcentaje antes descrito. En las evaluaciones, en las que esté presente la resolución del proyecto integrador, el 30% de peso en la calificación de las actividades de enseñanza/aprendizaje tendrá dos componentes: la calificación del proyecto valdría un 20% de la nota de la evaluación, y el resto de actividades de enseñanza/aprendizaje un 10%. El proyecto integrador del módulo **deberá entregarse y aprobarse** para poder aprobar el módulo profesional. En aquellas actividades que requieran del trabajo en grupo, el profesor podrá evaluar por medio oral o escrito la implicación y participación de cada uno de los miembros.
2. Prueba específica de evaluación: Es una prueba que se llevará a cabo de forma escrita o a ordenador, en el que se probará el conocimiento del alumno acerca de la materia impartida en esa evaluación. Deberá tener **una calificación superior a 4 (independientemente de los otros criterios)** por el alumno para que pueda aprobar la evaluación.

NOTA: El departamento considera muy importante que el alumno aprenda a trabajar correctamente por lo que cualquier práctica o examen que no esté correctamente sangrado (indentado) o que no compile no podrá en ningún caso obtener una calificación superior a 4.

Para superar el módulo es necesario que el alumno no tenga actitudes contrarias a las normas de convivencia.

Evaluación ordinaria de Junio

Los alumnos que, llegado junio, no hayan obtenido evaluación positiva en alguna de las evaluaciones trimestrales deberán someterse a una prueba específica de evaluación de todo el módulo.

En el caso de que un alumno no haya superado **la primera evaluación** pero obtenga calificación positiva en la segunda y la tercera, se considerará recuperada la primera evaluación con un 5 como nota.

En el caso de que un alumno no haya superado **la segunda evaluación** pero obtenga calificación positiva en la primera y la tercera, se considerará recuperada la segunda evaluación con un 5 como nota siempre que obtenga al menos un 7 en la tercera evaluación.

Calificación Final

La calificación final del módulo será:

En el caso de haber aprobado todas las evaluaciones trimestrales la nota final se obtendrá de la media ponderada de las notas de las evaluaciones de la siguiente forma:

- Primera evaluación 20%
- Segunda Evaluación 40%
- Tercera Evaluación 40%

En el caso de haberse presentado a la convocatoria ordinaria de junio para evaluarse de todo el módulo la calificación será la obtenida en el examen correspondiente redondeada.

Evaluación extraordinaria de Septiembre

En caso de no aprobar el módulo por ninguno de los procedimientos ordinarios establecidos, el alumno podrá presentarse a un examen extraordinario en septiembre. La calificación final del alumno será la nota del examen redondeada.

Pérdida de la evaluación continua

En caso de que algún alumno pierda la evaluación continua a tenor de lo establecido en PGA del centro, el alumno deberá presentarse al examen final de junio y la calificación final será la nota del examen redondeada.

Alumnos con el módulo pendiente

Los alumnos con el módulo pendiente podrán presentarse a un examen de recuperación del módulo en enero (convocatoria ordinaria) y en caso de no superarlo tendrá otra oportunidad en junio (convocatoria extraordinaria).

Recuperación

«La orientación es ayuda oportuna, y en su momento adecuado para que el alumno pueda superar «lagunas» de conocimientos, carencias personales, o mejorar actitudes ante el trabajo y la vida de relación interpersonal o social.» (López-Barajas Zayas, 1995)

El profesor habrá de facilitar, identificando las variables alterables, el éxito de sus alumnos, porque el éxito permite enfrentar con decisión las nuevas tareas. Entendemos por recuperación o retroinformación el conjunto de actividades y su proceso, que se establece en la programación de aula para los alumnos que tienen dificultades para avanzar. Al ser la evaluación continua e individualizada, las actividades de recuperación podrán variar en función del alumno y de los conocimientos y capacidades que sean objeto de recuperación. Dicho esto, se estipulan no obstante unos criterios básicos de recuperación para las unidades didácticas: detección de las carencias del alumno y de los motivos de las mismas por si fuera preciso modificar el enfoque didáctico, y realización de tareas específicas que ayuden al alumno en la carencia detectada.

Proacción

A los alumnos que avancen rápidamente en la consecución de las metas establecidas se les propondrán actividades que les permitan seguir avanzando, para evitar de esta manera el posible aburrimiento, con la consiguiente desmotivación.

Objetivos cuantitativos

- Alcanzar un porcentaje de alumnos aprobados que supere el nivel del curso anterior y que como mínimo alcance el 50% de los alumnos que acuden habitualmente a clase durante todo el curso.
- Alcanzar el 80% de alumnos aprobados en los apartados de "Actividades de enseñanza/aprendizaje" y "Actitud y asistencia a clase".

Evaluación de la práctica docente

El equipo docente del ciclo formativo acuerda las siguientes actividades para evaluar la práctica docente en el ciclo:

- Lograr impartir y desarrollar ampliamente la totalidad de las unidades de trabajo que componen esta programación didáctica.
- Evaluar, a final de curso, la consecución de los objetivos cuantitativos planteados en la programación didáctica.
- Elaborar una serie de cuestionarios, por parte de los miembros del equipo docente del ciclo para recoger las inquietudes y propuestas de los alumnos del ciclo sobre algunos aspectos como metodología, desarrollo de las clases, utilización de las nuevas tecnologías en la práctica docente...
- Incorporar a la programación didáctica y la práctica docente de años sucesivos los cambios y mejoras derivadas de esta evaluación.

Procedimiento para que el alumno conozca esta programación.

La programación será leída y explicada por el profesor al principio de curso a los alumnos. En dicho momento se informará a los alumnos de la posibilidad de tener acceso a esta programación con solo solicitarlo.

Plan de atención a la diversidad.

Hay que tener en cuenta que esta etapa de formación, no es obligatoria y tiene como objetivo formar profesionalmente al alumno. Las adaptaciones curriculares, por lo tanto, deben ser **no significativas**, es decir no se pueden variar ni los contenidos ni los resultados de aprendizaje.

Las medidas tomadas serán relativas a cambios en la metodología, atención más personalizada, cambios en las actividades o recursos materiales y didácticos, e incluso de tiempo en la realización de determinadas tareas, etc. Todo ello en función de lo que se pretenda compensar.

Las unidades de trabajo podrán ser secuenciadas y desarrolladas en el tiempo de distinto modo, de acuerdo a las características del grupo clase.

En cuanto a la metodología se fomentará la experiencia directa, la cooperación entre alumnos, la participación, la motivación, la autonomía, etc.

Se adecuará la evaluación a los criterios de acuerdo con la realidad del grupo. De este modo, se determinarán las estrategias de evaluación formativas que permitan prevenir, o en su defecto, detectar lo antes posible y evitar la consolidación de problemas y dificultades de aprendizaje.

En caso de contar con uno o más alumnos con necesidades específicas de apoyo educativo, es decir, aquellos que parten de desigualdad de oportunidades, se realizarán las adaptaciones de acceso al currículo no significativas convenientes, tomando como referencia las pautas del departamento de Orientación que sirvan para establecer una alternativa de metodología de trabajo que garantice el cumplimiento de, al menos, los objetivos mínimos marcados en la presente programación. En este caso, se adaptará el apartado Medios y recursos en base a las directrices marcadas por el centro: supresión de barreras arquitectónicas, cambios en las condiciones de iluminación, adaptación de periféricos del ordenador, uso de sistemas complementarios de comunicación, etc.

Para aquellos alumnos que muestren un especial interés y superen con facilidad las capacidades del módulo, se deben tener planteadas una serie de actividades que profundicen en aquellos aspectos más destacados del módulo, a través de las unidades de trabajo.

Bibliografía

Juan Carlos Moreno, *Programación- Ciclos Formativos (GS)*, RA-MA

Schildt Herbert, *Fundamentos de Java 3ª Edición*, Mc Graw-Hill

Bruce Eckel, *Thinking in Java 6rd Edition*, <http://www.bruceeckel.com>

Doug Lowe y Barry Burd, *Java All-In-One Desk Reference For Dummies, 2nd Edition*, John Wiley & Sons

Holzner, Steven, *La Biblia De Java 2*, Anaya Multimedia 2000

Alcalde, E., García,M., *Metodología de la programación*, McGraw-Hill,1992.

Joyanes,L., *Fundamentos de Programación*, McGraw-Hill

Cay S. Horstmann & Gary Cornell, *Java 2 Fundamentos*, Prentice Hall