Programación del Módulo Profesional

“Programación”

Del Ciclo Formativo de Grado Superior

“Desarrollo de Aplicaciones Multiplataforma”

Contenido

1 OBJETIVOS GENERALES DEL MÓDULO PROFESIONAL. 3

2 UNIDADES DE COMPETENCIA ASOCIADAS AL MÓDULO PROFESIONAL 3

3 CONTENIDOS Y SECUENCIACIÓN 4

4 CONTENIDOS MÍNIMOS 9

5 METODOLOGÍA Y ESTRATEGIAS DIDÁCTICAS 9

6 RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN 12

7 PROCEDIMIENTOS E INSTRUMENTOS DE EVALUACIÓN 12

8 LOS CRITERIOS DE CALIFICACIÓN 16

9 RECURSOS DIDÁCTICOS 17

10 ATENCIÓN A LA DIVERSIDAD Y ADAPTACIONES CURRICULARES 19

11 UTILIZACIÓN DE LAS TIC 19

12 ACTIVIDADES DE RECUPERACIÓN DE MÓDULOS PROFESIONALES PENDIENTES 19

13 PROCEDIMIENTO PARA QUE EL ALUMNADO Y SUS FAMILIAS CONOZCAN LOS CONTENIDOS, CRITERIOS DE EVALUACIÓN, LOS CRITERIOS DE CALIFICACIÓN, LOS PROCEDIMIENTOS Y LOS INSTRUMENTOS DE EVALUACIÓN 19

14 ACTIVIDADES COMPLEMENTARIAS Y EXTRAESCOLARES 19

15 REFERENCIAS 20

1 Objetivos generales del módulo profesional.

Los objetivos generales del módulo de Programación son los fijados en el Real Decreto 450/2010, de 16 de abril, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y se fijan sus enseñanzas mínimas.

- Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.
- Seleccionar y emplear técnicas, lenguajes y entornos de desarrollo, evaluando sus posibilidades, para desarrollar aplicaciones en teléfonos, PDA y otros dispositivos móviles.
- Seleccionar y emplear lenguajes y herramientas, atendiendo a los requerimientos, para desarrollar componentes personalizados en sistemas ERP-CRM.
- Identificar los cambios tecnológicos, organizativos, económicos y laborales en su actividad, analizando sus implicaciones en el ámbito de trabajo, para mantener el espíritu de innovación.

2 Unidades de competencia asociadas al módulo profesional.

El objetivo fundamental del módulo profesional **Programación** se corresponde con las siguientes Unidades de Competencia del Catálogo Nacional de Cualificaciones Profesionales (R.D. 450/2010, de 16 de abril).

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

- Configurar y explotar sistemas informáticos, adaptando la configuración lógica del sistema según las necesidades de uso y los criterios establecidos.
- 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.
- Participar en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento y la educación empleando técnicas, motores y entornos de desarrollo específicos.
- Desarrollar aplicaciones para teléfonos, PDA y otros dispositivos móviles empleando técnicas y entornos de desarrollo específicos.

- Establecer vías eficaces de relación profesional y comunicación con sus superiores, compañeros y subordinados, respetando la autonomía y competencias de las distintas personas.
- 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 Contenidos y secuenciación

Los Contenidos del módulo de Programación son los fijados en el DECRETO 3/2011, 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 Multiplataforma.

La duración del módulo a lo largo del año es de 270 horas, repartidas en 8 horas semanales.

Se estiman **20 horas para evaluación**, por lo que se prevé la siguiente distribución de tiempos:

PRIMER TRIMESTRE	**74**
U.T. 1 INTRODUCCIÓN A LA PROGRAMACIÓN .	8
U.T. 2 ESTRUCTURA Y ELEMENTOS DE UN PROGRAMA .	16
U.T. 3 LECTURA Y ESCRITURA DE INFORMACIÓN. STREAMS.	8
U.T. 4 ESTRUCTURAS DE CONTROL: CONDICIONALES E ITERATIVAS.	24
U.T. 5 INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS .	18
SEGUNDO TRIMESTRE	**80**
U.T. 6 ESTRUCTURAS DE DATOS. ARRAYS Y CADENAS DE CARACTERES	32
U.T.7 DESARROLLO DE CLASES E INSTANCIACIÓN DE OBJETOS.	24
U.T.8 UTILIZACIÓN AVANZADA DE CLASES.	24
TERCER TRIMESTRE	**96**
U.T.9 COLECCIONES DE DATOS: LISTAS, PILAS COLAS	24
U.T.10 CONTROL Y MANEJO DE EXCEPCIONES.	8
U.T.11 ESTRUCTURAS DE DATOS EXTERNAS. FICHEROS.	30
U.T.12 INTERFACES GRÁFICOS DE USUARIO. ACCESO A BASES DE DATOS RELACIONALES	24
U.T.13 MANTENIMIENTO DE LA PERSISTENCIA DE LOS OBJETOS.	10

Antes de especificar los contenidos de cada Unidad, es necesario decir que podrían verse alterados debido a las necesidades de adaptación y que se abordarán en una parte significativa de una manera práctica.

U.T. 1. Introducción a la programación	**8 horas**
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. - Introducción al Lenguaje Java. Entorno de programación.	

U.T. 2 Estructura y Elementos de un Programa.	**16 horas**
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 explicitas (casting). - Comentarios.	

U.T. 3. Lectura y escritura de la información. Streams	**8 horas**
Contenidos	
- 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.

U.T. 4. Estructuras de control	24 horas
Contenidos	

- Sentencias Condicionales
  - Alternativa simple: IF
  - Alternativa doble: IF-ELSE
  - Anidación de sentencias condicionales.
  - Alternativa múltiple: SWITCH
- Sentencias Iterativas
  - WHILE
  - DO-WHILE
  - FOR
- Sentencias de Salida de un bucle.
  - BREAK
  - CONTINUE

U.T. 5. Introducción a la programación orientada a objetos	18 horas
Contenidos	

- P.O.O. Conceptos, propiedades.
- Clases. Características.
- atributos ,métodos y visibilidad
- Objetos. Estado, comportamiento e identidad
- Comunicación entre objetos. Envío de mensajes.

U.T. 6. Estructuras de datos. Arrays y Cadenas de caracteres	32 horas
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.

U.T. 7. Desarrollo de clases e instanciación de objetos	**24 horas**
Contenidos	

- Concepto de clase.
- Estructura y miembros de una clase.
- Creación de atributos. Declaración e inicialización.
- Creación de métodos. Declaración, argumentos y valores de retorno.
- Sobrecarga de métodos.
- Visibilidad. Modificadores de clase, de atributos y de métodos.
- Paso de parámetros. Paso por valor y paso por referencia.
- Métodos estáticos.
- Características de los objetos.
- Constructores.
- Instanciación de objetos. Declaración y creación.
- Librerías y paquetes de clases. Utilización y creación.
- Documentación sobre librerías y paquetes de clases.

U.T. 8. Utilización avanzada de clases	**24 horas**
Contenidos	

- Herencia.

Concepto y tipos (simple y múltiple).

Superclases y subclases.

Constructores y herencia.

Clases y métodos abstractos y finales.

Interfaces. Clases abstractas vs. Interfaces.

- Polimorfismo.

Concepto.

Polimorfismo en tiempo de compilación (sobrecarga)

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.

U.T. 9 Colecciones de datos: Listas, Pilas y Colas	**24 horas**
Contenidos	

- 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.

U.T.10. Control y manejo de excepciones	**8 horas**
Contenidos	

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

U.T. 11 Estructuras externas de datos	**30 horas**
Contenidos	

- 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.

U.T. 12. Interfaces Gráficos de usuario. Acceso a bases de datos relacionales	**24 horas**
Contenidos	

- Interfaces Gráficos de usuario.
  - Concepto de evento.
  - Creación de controladores de eventos.
- Bases de datos relacionales.
  - Establecimiento de conexiones.
  - Recuperación de información.
  - Manipulación de la información.
  - Ejecución de consultas sobre la base de datos.

U.T. 13 Mantenimiento de la persistencia de los objetos.	**10 horas**
Contenidos	

- *Bases de datos orientadas a objetos. Características.*
- *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.*

4 Contenidos mínimos

Se entienden como contenidos básicos los que se detallan en el Real Decreto por el que se establece el título. A saber:

Identificación de los elementos de un programa informático:

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

Utilización de objetos:

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

Uso de estructuras de control:

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

Desarrollo de clases:

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

Lectura y escritura de información:

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

Aplicación de las estructuras de almacenamiento:

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

- Listas.

Utilización avanzada de clases:

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

Mantenimiento de la persistencia de los objetos:

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

Gestión de bases de datos relacionales:

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

5 Metodología y estrategias didácticas

Metodología

El profesor facilitará a través de la aplicación moodle apuntes, actividades, prácticas y enlaces a páginas web y videos relativos a los contenidos expuestos en la programación. Explicará a los alumnos la mayor parte de los contenidos tanto teóricos como prácticos del módulo. El resto deberán ser autoaprendidos por ellos, de forma guiada o autónoma.

Las actividades de enseñanza/aprendizaje consistirán principalmente en la codificación en el ordenador de pequeños programas relativos a cada unidad temática. Estás se realizarán de forma individual o en grupo de 2 alumnos dependiendo del número de ordenadores disponibles en el aula.

Los alumnos utilizarán también la aplicación moodle para subir las actividades y programas que indique el profesor.

Estrategia

Se busca que los alumnos:

a) Disfruten del aprendizaje. Para lo cual es preciso hacerlo dinámico y participativo.

b) Asuman la responsabilidad de su propio aprendizaje. Para ello, se les orientará para que se impliquen y que desarrollen su autonomía.

6 Resultados de aprendizaje y Criterios de evaluación

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.

   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 clases.

   h) 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.

d) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.

e) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.

f) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.

g) 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 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.

7 Procedimientos e instrumentos de evaluación

La evaluación se realizará agrupando las unidades temáticas por evaluaciones trimestrales.

Los instrumentos de evaluación serán:

- **Prueba específica de evaluación**. Examen escrito y/o práctico relativo a todos los contenidos impartidos en esa evaluación. Se realizará al finalizar cada evaluación.
- **Actividades de enseñanza/aprendizaje**. Trabajos, prácticas, cuestionarios, pruebas parciales escritas o prácticas realizadas durante el periodo correspondiente a cada evaluación.

8 Los criterios de calificación

La evaluación del módulo se hará conforme a la siguiente tabla.

Convocatoria	Peso Actividades de enseñanza/aprendizaje	Peso prueba específica de evaluación	Recuperación
Evaluación 1	20%	80%	Sí
Evaluación 2	20%	80%	Sí
Evaluación 3	20%	80%	No

La calificación de cada evaluación parcial se hará del siguiente modo:

- **Actividades de enseñanza/aprendizaje:** Serán evaluadas con un valor numérico comprendido entre 0 y 10 o con un APTO o NO APTO.
- **Prueba específica de evaluación:** Tendrá una nota numérica entre 0 y 10. Se considera aprobado si es igual o mayor que 5.

Como resultado de la aplicación de los porcentajes presentados en la tabla anterior se obtendrá una nota con una precisión de un decimal, que se redondeará al entero más cercano; no obstante, en los futuros cálculos en los que se utilicen estos resultados del alumno, se empleará la nota previa al redondeo.

A efectos de redondeo, los decimales inferiores a 0,5 se redondearán al entero más bajo. Los superiores o iguales a 0,5 al entero más alto. Esta regla tiene dos excepciones: la franja entre 4 y 5 se redondeará siempre a 4 y las notas inferiores a 1 se redondearán a 1.

Las faltas de ortografía cometidas en todo tipo de escritos (ejercicios, prácticas, exámenes, etc.) se penalizarán, hasta un máximo de un punto, con arreglo al siguiente baremo:

- Cada error en el empleo de las grafías: 0'2 puntos.
- *Cada error de acentuación o puntuación: 0'1 puntos*

Será necesario, para que el alumno pueda aprobar la evaluación, que supere la prueba específica al menos con un 5’0, y que haya entregado todas las actividades que el profesor haya declarado como imprescindibles en esa evaluación. En caso contrario su nota máxima será un 4’0 independientemente del resultado de la ponderación anterior.

La prueba de recuperación de evaluación consistirá en un examen similar al realizado en la prueba específica de evaluación y se calificará siguiendo los mismos criterios. En el tercer trimestre no se realizará esta prueba por proximidad con el examen final de evaluación.

Los alumnos que no hayan aprobado alguna evaluación podrán recuperarla en el examen final de evaluación ordinaria. Cada alumno se examinará de la evaluación o evaluaciones que tenga pendientes, debiendo obtener al menos un 5,0 en cada una de ellas y que haya entregado todas las actividades que el profesor haya declarado como imprescindibles en cada evaluación.

La calificación final del módulo consistirá en la media aritmética de las calificaciones obtenidas en las tres evaluaciones trimestrales, siempre que estén todas aprobadas.

Los alumnos que no superen el módulo en convocatoria ordinaria realizarán un examen final de evaluación extraordinaria sobre los criterios de evaluación mínimos incluidos en la programación. Cada alumno recibirá un informe de evaluación donde se detallarán los contenidos sobre los que versará el examen. El examen se calificará con un valor numérico comprendido entre 0 y 10, y se considerará aprobado si este valor es mayor o igual a 5. Se aplicarán los mismos criterios de redondeo establecidos anteriormente.

9 Recursos didácticos

En el aula:

Recursos comunes: Pizarra blanca, rotuladores de pizarra blanca y videoproyector. Además, los alumnos deberán disponer de un cuaderno de teoría y de ejercicios donde vayan recogiendo ordenadamente su información de las clases, así como un medio de almacenamiento persistente (disco externo o memoria USB) para almacenar los trabajos que vayan realizando, sin tener que depender exclusivamente de los discos duros de los equipos.
Además del correo electrónico los profesores podrán utilizar la plataforma educativa Moodle para la distribución de apuntes y ejercicios a los alumnos.

Recursos de infraestructura informática: Al menos habrá un PC en el aula para cada dos alumnos, y otro para el profesor. En cada puesto informático del aula (incluyendo el PC del profesor) estará instalado un sistema operativo Windows, la máquina virtual de Java y algunos entornos de desarrollo para Java. Todos estos ordenadores estarán conectados por una red a través de un concentrador o de un conmutador, y tendrán acceso controlado a la red Internet.

Los alumnos dispondrán de una cuenta de usuario en el servidor de dominio de la clase, con un directorio asociado en el que depositarán los ficheros que se necesiten conservar en el aula.

Bibliografía y recursos en línea:

No se usará libro de texto, aportando el profesor parte de los apuntes y recomendando el uso de algunos libros de los citados a continuación, y determinadas páginas de Internet.

Bibliografía

FERNÁNDEZ, J., OLIVER, F. J., y SÁNCHEZ, J. L. (1992): *Algoritmos. Problemas resueltos y comentados*. Madrid. Paraninfo.

ALONSO, M. D., y RUMEU, S. (1992): *Metodología de la Programación: Programación Estructurada*. Madrid. Paraninfo.

ALONSO, F., y SEGOVIA, F. J.: *Entornos y Metodologías de Programación*. Madrid. Paraninfo.

PIATTINI, M. G., y GARCÍA, F. O. (2002): *Calidad en el Desarrollo y Mantenimiento del Software*. Madrid. Ra-Ma.

BOOCH, G. (1994): *Análisis y Diseño Orientado a Objetos con Aplicaciones (2ª Edición)*. EEUU. Addison-Wesley/Díaz de Santos.

FRANCISCO JAVIER CEBALLOS. Java 2 Curso de programación (2ª Edición). España, Ra-Ma.

GILLES BRASSARD & PAUL BRATLEY. *Fundamentos de* algoritmia. ISBN 848966000X

JAVIER GARCIA DE JALÓN y otros. *Aprenda Java como si estuviera en primero.* Tecnun.

Direcciones URL

http://www.gnu.org → Sitio web de GNU (GNU's Not Unix)

http://java.sun.com → Sitio web de Sun dedicada a las tecnologías Java

http://netbeans.org → La herramienta de desarrollo Netbeans

http://bluej.org → La herramienta BlueJ, de desarrollo con enfoque docente

http://www.tecnun.es/typo3/index.php?id=1559 -> Colección de libros y apuntes gratuitos acerca de C/C++/Java y Visual Basic

http://msdn.microsoft.com -> Microsoft Developer Network

Recursos de ilustración audiovisual: Se hará uso de esquemas, diagramas, tablas cronológicas, presentaciones informáticas, videos, etc. en formato electrónico. Se utilizará el vídeo-proyector instalado en el aula.

10 Atención a la diversidad y adaptaciones curriculares

En el caso en que este módulo sea cursado por un alumno con discapacidad, se realizará la adaptación de las actividades de formación, los criterios y los procedimientos de evaluación necesarios, de modo que se garantice su accesibilidad a las pruebas de evaluación; esta adaptación en ningún caso supondrá la supresión de objetivos, o resultados de aprendizaje que afecten a la competencia general del título. La adaptación curricular se archivará en el expediente del alumno.

11 Utilización de las TIC

Se utilizarán todos los recursos informáticos y audiovisuales disponibles, expresados en el apartado 9 de Recursos didácticos:

12 Actividades de recuperación de módulos profesionales pendientes

Los alumnos de 2º curso con el módulo de Programación suspenso serán examinados en Febrero/Marzo mediante un examen final de evaluación ordinaria, sobre los criterios de evaluación mínimos incluidos en la programación. El examen se calificará con un valor numérico comprendido entre 0 y 10, y se considerará aprobado si este valor es mayor o igual a 5. Se aplicarán los mismos criterios de redondeo establecidos anteriormente.

13 Procedimiento para que el alumnado y sus familias conozcan los contenidos, criterios de evaluación, los criterios de calificación, los procedimientos y los instrumentos de evaluación

Se publicará la presente programación en la página web del Centro (www.iesjovellanos.org).

14 Actividades complementarias y extraescolares

- Concurso Nacional de Programación para Ciclos Formativos "**ProgramaMe**".
- Visitar alguna empresa relaciona con el sector correspondiente al ciclo.
- Recibir la visita en el centro de alguna de las empresas colaboradoras en las FCT para su presentación a los alumnos.

15 Referencias

- **DECRETO 3/2011, 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 Multiplataforma. (BOCM 31 de enero de 2011)
- **Real Decreto 450/2010, de 16 de abril**, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y se fijan sus enseñanzas mínimas. (BOE 20 de mayo de 2010)
- **Orden 2694/2009, de 9 de junio**, por la que se regula el acceso, la matriculación, el proceso de evaluación y la acreditación académica de los alumnos que cursen en la Comunidad de Madrid la modalidad presencial de la formación profesional del sistema educativo establecida en la Ley Orgánica 2/2006, de 3 de mayo, de Educación. (BOCM lunes 22 de Junio de 2009). (Modificada por la Orden 11783/2012, de 11 de diciembre – BOCM de 04/01/2013).
- **Proyecto Educativo de Centro**. IES Gaspar Melchor de Jovellanos, Fuenlabrada.