Textos Docentes

Enseñanzas de Informática

Programación Didáctica del módulo

Programación en Lenguajes

Estructurados

Ciclo Formativo de Desarrollo de

Aplicaciones Informáticas

Juan Marcos Sacristán Donoso

ISBN: 84-689-4297-9

Registro: 05/73557

Índice

1	INTRODUCCIÓN	3
1.1	REFERENCIA LEGISLATIVA	3
1.2	NIVELES DE CONCRECIÓN CURRICULAR.....	5
1.3	DESCRIPCIÓN DEL ENTORNO.....	7
1.4	CARACTERÍSTICAS DEL ALUMNADO.	9
2	OBJETIVOS.....	11
2.1	OBJETIVOS GENERALES DEL CICLO FORMATIVO	11
2.2	OBJETIVOS DEL MÓDULO.....	14
2.3	OBJETIVOS RELACIONADOS CON TEMAS TRANSVERSALES	17
3	CONTENIDOS.....	18
3.1	CONTENIDOS RELACIONADOS CON LOS BLOQUES TEMÁTICOS DEL CURRÍCULO	18
	<i>Bloque 1: Metodología de la programación.....</i>	<i>22</i>
	UT1. Introducción a la programación estructurada.....	22
	UT2. Representación de algoritmos.....	23
	UT3. Desarrollo del software.....	23
	<i>Bloque 2: Programación en lenguajes estructurados de tercera generación.....</i>	<i>24</i>
	UT4. Lenguaje de programación C.....	24
	UT5. Sentencias de control de flujo.....	25
	UT6. Programación modular.....	25
	UT7. Estructuras estáticas de datos.....	26
	UT8. Métodos de búsqueda y ordenación.....	26
	UT9. Ficheros.....	27
	UT10. Programación dinámica.....	27
	UT11. Estructuras dinámicas de datos.....	28
	<i>Bloque 3: Herramientas de programación.....</i>	<i>28</i>
	UT12. Herramientas de programación.....	28
	<i>Bloque 4: Documentación del software.....</i>	<i>29</i>
	UT13. Documentación del software.....	29

<i>Bloque 5: Programación orientada a objetos.....</i>	<i>30</i>
UT14. Programación orientada a objetos.....	30
UT15. Programación en C++	31
3.2 CONTENIDOS SOBRE LOS TEMAS TRANSVERSALES	32
3.3 CONTENIDOS INTERDISCIPLINARES	33
<i>Relación entre los módulos SMR y PLE</i>	<i>33</i>
<i>Relación entre los módulos AAIG y PLE</i>	<i>34</i>
3.4 TEMPORALIZACIÓN DE LOS CONTENIDOS.....	35
4 METODOLOGÍA: ORIENTACIONES DIDÁCTICAS.....	37
5 CRITERIOS DE EVALUACIÓN	40
6 ATENCIÓN A ALUMNOS CON NECESIDADES EDUCATIVAS ESPECÍFICAS ..	49
7 MATERIALES Y RECURSOS DIDÁCTICOS.....	50

1 Introducción

El presente documento es la programación didáctica del módulo de **Programación en Lenguajes Estructurados** que se imparte en el primer curso del ciclo formativo de grado superior de **Desarrollo de Aplicaciones Informáticas**. Dicho módulo, tal como establece el Real Decreto de Título, se debe impartir por profesores del cuerpo de **Enseñanza Secundaria** de la especialidad de **Informática**. El autor, Juan Marcos Sacristán Donoso, es funcionario de carrera del cuerpo de profesores de Enseñanza Secundaria, especialidad Informática, con una amplia experiencia en lenguajes de programación, tanto de docente como de profesional. Por todo ello, el autor presenta esta programación didáctica como propuesta didáctica para el módulo mencionado.

1.1 Referencia legislativa

Para elaborar la programación didáctica se han tenido en cuenta las siguientes referencias legislativas.

- Ley Orgánica 1/90 de Ordenación General del Sistema Educativo.
- Ley Orgánica 10/2002 de Calidad de la Educación.
- R.D. 676/93, de 7 de mayo, por el que se establecen las directrices generales sobre los títulos y las correspondientes enseñanzas mínimas de Formación Profesional.
- Real Decreto 1661/94 por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Informáticas y las correspondientes enseñanzas mínimas.

- Real Decreto 1676/94 por el que se establece el currículo del ciclo formativo de grado superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Informáticas.
- Orden de 14 de Noviembre de 94 por la que se regula el proceso de evaluación y acreditación académica del alumnado que curse la formación profesional específica establecida en la LOGSE.
- R.D. 1744/98, de 31 de julio, sobre uso y supervisión de libros de texto y demás material curricular correspondiente a las enseñanzas de régimen general.
- Orden de 20 de diciembre de 94, de la Consellería de Educación y Ciencia, por la que se dictan instrucciones para el desarrollo de la educación en valores en las actividades educativas de los centros docentes (DOGV de 3 de marzo).
- Orden de 18 de junio de 99, de la Consellería de Cultura, Educación y Ciencia, por la que se regula la atención a la diversidad en la ESO (DOGV de 29 de junio).
- Decreto 39/98, de 31 de marzo, del Gobierno Valenciano, de ordenación de la educación para la atención del alumnado con necesidades educativas especiales (DOGV de 17 de abril).

Conviene señalar que la estructura de los R.D. de título y enseñanzas mínimas vienen señalados en el R.D. 676/1993, de 7 de Mayo, por el que se establecen las directrices generales sobre los títulos y las correspondientes enseñanzas mínimas de Formación Profesional. Sin embargo, el R.D. 676/1993 ha sido derogado recientemente por el R.D. 362/2004, de 5 de marzo, por el que se establece la ordenación general de la formación profesional específica,

de modo que el gobierno deberá legislar nuevos R.D. de título y enseñanzas mínimas. Pero mientras esto no ocurre, los R.D. de título y enseñanzas mínimas que se encuentran en vigor se basan en lo establecido por el R.D. 676/1993.

1.2 Niveles de concreción curricular

La LOGSE en su artículo IV establece que los elementos de currículo son: los objetivos, contenidos, métodos, criterios de evaluación y promoción de cada una de las etapas, ciclos, cursos y modalidades en que se organiza la práctica educativa.

El primer nivel de concreción curricular es definido por la Administración Educativa y tiene carácter obligatorio para todo el estado. En él se definen los objetivos generales de cada una de las etapas, así como los objetivos y contenidos generales de cada una de las áreas que configuran dichas etapas. Se incluyen además, consideraciones generales sobre metodología didáctica y criterios de evaluación. Este primer nivel de concreción curricular queda establecido por los reales decretos de enseñanzas mínimas y reales decretos de currículo, enriquecido por las aportaciones que realizan las comunidades autónomas con transferencias en educación.

El segundo nivel de concreción curricular es competencia de los equipos docentes en los centros, y son conocidos como proyectos curriculares. En ellos se establecen, a partir de los reales decretos de enseñanzas mínimas y de currículo, una contextualización de los objetivos y una secuenciación y organización tanto de los objetivos como de los contenidos de cada uno de los cursos de cada uno de los ciclos de una etapa. Recogen también criterios de

evaluación y promoción que no necesariamente son iguales en todos los centros.

El tercer nivel de concreción curricular son las Programaciones Docentes, donde se establece una secuencia de Unidades Didácticas para cada una de las materias de cada uno de los cursos de un ciclo.

A continuación, mostramos una tabla resumen con los tres niveles de concreción curricular presentes en la formación profesional específica.

PRIMER NIVEL DE CONCRECIÓN
DEFINICIÓN: marco común en el que se formulan un conjunto de prescripciones y orientaciones sobre la intencionalidad de la Formación Profesional y las estrategias adecuadas.
CARÁCTER: abierto, flexible, orientador y prescriptivo.
RESPONSABLE: las Administraciones educativas.
DOCUMENTOS: <ul style="list-style-type: none">• R.D. de título.• R.D. de currículo.
SEGUNDO NIVEL DE CONCRECIÓN
DEFINICIÓN: conjunto de decisiones articuladas que materializan el Currículo en propuestas de intervención didáctica adecuadas a un contexto específico.
CARÁCTER: general y orientativo para un centro concreto.
RESPONSABLE: <ul style="list-style-type: none">• El Centro educativo.• El Departamento didáctico.• El Equipo Docente.

DOCUMENTOS: <ul style="list-style-type: none"> • Proyecto Educativo de Centro. • Proyecto Curricular de Ciclo Formativo.
TERCER NIVEL DE CONCRECIÓN
DEFINICIÓN: conjunto de unidades de trabajo ordenadas y secuenciadas para los módulos profesionales de cada ciclo formativo.
CARÁCTER: planificador del proceso de enseñanza-aprendizaje para un determinado grupo de alumnos/as.
RESPONSABLE: <ul style="list-style-type: none"> • El Departamento didáctico. • El Equipo Docente. • El profesor/a concreto de un módulo profesional.
DOCUMENTOS: Programación de aula.

Por tanto, esta programación didáctica se enmarca en el segundo nivel de concreción curricular, y formará parte del Proyecto Curricular del Ciclo Formativo. Su finalidad es la de adaptar el currículo prescriptivo en el centro, es decir ajusta el currículo a las características de un centro concreto.

1.3 Descripción del entorno

En esta programación didáctica nos vamos a situar en el contexto del Instituto de Educación Secundaria (I.E.S.) Mare Nostrum. El I.E.S. Mare Nostrum está ubicado en el barrio de San Gabriel de Alicante. Se trata de un centro creado en 1990, que se dedicaba inicialmente a la Formación Profesional, y disponiendo de profesorado especializado con experiencia

demostrada en la enseñanza tanto en el campo informático como en el administrativo y comercial.

La ciudad de Alicante es la capital de la provincia de Alicante, y su mayor fuente de ingresos es el turismo. Sin embargo, algunas empresas han aprovechado el hecho de tener cerca la Universidad de Alicante, lo que les ha permitido disponer de jóvenes prometedores que han diseñado e implementado tecnologías novedosas. Un ejemplo es la empresa Trymedia Systems, que se dedica a la protección de sistemas informáticos. Es muy importante conocer las posibilidades de inserción laboral de nuestros estudiantes, ya que ayuda a determinar los aprendizajes prioritarios y útiles de cara a su futuro laboral. Por todo lo dicho, salvo algún caso aislado como Trymedia Systems, lo más probable es que nuestros alumnos acaben ocupando puestos en empresas no dedicadas a la informática, pero con la necesidad de tener algún profesional de la informática que pueda administrar sus sistemas informáticos, y desarrollar sus aplicaciones informáticas.

Otro aspecto importante a destacar es la existencia de dos lenguas oficiales en la Comunidad Valenciana, que son el valenciano y el castellano. Esto se refleja, tal y como especifica la convocatoria del procedimiento selectivo de ingreso al cuerpo de profesores enseñanza secundaria, en que los docentes deben acreditar unos conocimientos mínimos de las dos lenguas oficiales. También es importante formar adecuadamente a los alumnos en las dos lenguas, facilitando su futura integración e inserción sociolaboral.

La oferta de transporte en Alicante es bastante amplia, aunque menor que otras provincias de la comunidad. Actualmente, dispone de autobuses urbanos, y también una línea especial que comunica directamente con la

Universidad (ubicada en el campus de San Vicente del Raspeig, a unos diez kilómetros). También se está desarrollando un proyecto de tranvía para la ciudad, aunque actualmente el recorrido es mínimo. Se espera que con el transporte urbano se consiga una mayor fluidez en el tráfico, ya que algunas zonas de Alicante se caracterizan por atascos a prácticamente cualquier hora del día. Las posibilidades de transporte interurbano son variadas: por un lado está la estación de tren de RENFE en el centro de la ciudad, y la estación del Ferrocarril de la Generalitat Valenciana. También está la estación de autobuses, y finalmente el puerto de Alicante con algunos trayectos por mar. Todo ello propicia que puedan venir alumnos de muchos sitios. Además, cabe destacar que la línea de cercanías de RENFE que comunica Alicante con Murcia dispone de una parada en el barrio de San Gabriel, muy cerca del I.E.S. Mare Nostrum.

Cualquier descripción de Alicante sería incompleta sin hablar de las hogueras de San Juan, posiblemente la fiesta más importante a nivel local. Se celebra todos los años la última semana de Junio. La oferta cultural de Alicante es bastante amplia, aunque desconocida por mucha gente. Destacan con mucha fuerza algunos centros que proponen multitud de actividades, ayuda y asesoramiento, especialmente dirigidos a jóvenes, y que son el centro cultural de la CAM, la fundación Bancaja, y el Centro 14.

1.4 Características del alumnado.

En la mayoría de los casos, nos vamos a encontrar con alumnos cuya mayor motivación es conseguir una rápida inserción laboral. Normalmente se tratará de alumnos con el título de bachiller y que han preferido cursar un ciclo

formativo de grado superior en vez de ir a la Universidad. Sin embargo, conviene no olvidar otros tipos de alumnos:

- Por un lado, aquellos alumnos que iniciaron estudios universitarios pero que han abandonado y han decidido refugiarse en la formación profesional. Suelen ser alumnos problemáticos.
- Por otro lado, trabajadores que han decidido cursar un ciclo formativo para mejorar su cualificación profesional. Se trata de personas muy motivadas, pero que pueden presentar circunstancias familiares complicadas, además de que normalmente han perdido el hábito del estudio.

También conviene mencionar que, dadas sus facilidades de acceso y oferta de estudios, muchos alumnos de distintas zonas acuden al I.E.S. Mare Nostrum, lo que incrementa la pluralidad del alumnado en base a características sociales, económicas, etc. Por todo ello, el centro ha ido ampliando su oferta en horarios y niveles, y también ha ido adaptando sus enseñanzas y métodos.

2 Objetivos

2.1 Objetivos generales del ciclo formativo

Los objetivos generales del ciclo formativo (OGC) se especifican en el apartado 3.1 del RD 1661/94 por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Informáticas y las correspondientes enseñanzas mínimas:

- OGC1. Seleccionar y utilizar fuentes y recursos de información de la organización e información técnica, interpretando y relacionando su contenido con las características particulares de la organización y el sistema informático actual para el diseño y desarrollo de aplicaciones informáticas.*
- OGC2. Configurar y utilizar los recursos de sistemas informáticos aislados o interconectados en red para la realización y verificación de pruebas y el desarrollo de programas.*
- OGC3. Documentar las aplicaciones de un proyecto informático, utilizando el lenguaje profesional con precisión y pulcritud.*
- OGC4. Evaluar el funcionamiento de una aplicación mediante la realización de pruebas.*
- OGC5. Diseñar bases de datos a partir del modelo conceptual y desarrollar aplicaciones, diseñando servicios de presentación, módulos de programación e interfaces Inter.-módulos, aplicando una metodología de desarrollo.*

- OGC6. *Proponer cambios en la composición y configuración de los sistemas informáticos, así como en las aplicaciones informáticas existentes en el sistema para resolver los problemas técnicos de la implantación de nuevas aplicaciones y responder al rendimiento y prestaciones demandadas.*
- OGC7. *Organizar los recursos y el tiempo disponible, estableciendo prioridades en sus actividades, actuando con orden y meticulosidad y valorando la trascendencia de su trabajo.*
- OGC8. *Establecer un clima positivo de relación y colaboración con el entorno, valorando la comunicación como uno de los aspectos más esenciales en su profesión.*
- OGC9. *Utilizar con autonomía las estrategias y procedimientos característicos y saber hacer propios de su sector, para tomar decisiones frente a problemas concretos o supuestos prácticos, en función de datos e informaciones conocidos, valorando los resultados previsibles que de su actuación pudieran derivarse.*
- OGC10. *Aplicar procedimientos de seguridad, protección, confidencialidad y conservación de la documentación e información en los medios y equipos de oficina e informáticos para garantizar la integridad, el uso, acceso y consulta de los mismos.*
- OGC11. *Valorar la incidencia de las nuevas tecnologías de comunicación y transmisión de información en los procesos administrativos y de gestión de empresas.*

- OGC12. *Desarrollar la iniciativa, el sentido de la responsabilidad, la identidad y la madurez profesional que permitan mejorar la calidad del trabajo y motivar hacia el perfeccionamiento profesional.*
- OGC13. *Valorar la importancia de la comunicación profesional, así como las normas y procedimientos de organización en las relaciones laborales, tanto de carácter formal como informal y su repercusión en la actividad e imagen de la empresa u organismo.*
- OGC14. *Interpretar el marco legal, económico y organizativo que regula y condiciona la actividad informática, identificando los derechos y las obligaciones que se derivan de las relaciones laborales, adquiriendo la capacidad de seguir y mejorar los procedimientos establecidos y de actuar proponiendo soluciones a las anomalías que pueden presentarse en los mismos.*
- OGC15. *Seleccionar y valorar críticamente las diversas fuentes de información relacionadas con su profesión, que le permitan el desarrollo de su capacidad de autoaprendizaje y posibiliten la evolución y adaptación de sus capacidades profesionales a los cambios tecnológicos y organizativos del sector.*

2.2 Objetivos del módulo

El módulo de programación en lenguajes estructurados está asociado a la unidad de competencia 3: *Elaborar, adaptar y probar programas en lenguajes de programación estructurada y de cuarta generación.*

En el apartado 3.2 del Real Decreto de Título se especifican las capacidades terminales (CT) de cada módulo. Las capacidades terminales del ciclo formativo son un reflejo de las realizaciones del perfil profesional asociado, y describen lo que una persona debe saber hacer en una determinada situación de trabajo. Para nuestro módulo tenemos:

CT1. *Elaborar programas utilizando lenguajes estructurados, cumpliendo con las especificaciones establecidas en el diseño.*

CT2. *Evaluar el funcionamiento de las aplicaciones mediante la realización de pruebas de los diferentes módulos de programación.*

CT3. *Elaborar la documentación completa relativa a las aplicaciones desarrolladas.*

CT4. *Adaptar aplicaciones a partir de nuevos requerimientos establecidos en el diseño.*

Podemos extraer los objetivos del módulo a partir de las capacidades terminales, haciendo una concreción en función de las características del alumnado y del centro, y fijándonos también en los criterios de evaluación asociados a cada capacidad terminal. Proponemos como objetivos del módulo (OM):

- OM1. Conocer los conceptos básicos sobre la programación, independientemente del lenguaje empleado.
- OM2. Conocer y aplicar las técnicas de la programación estructurada.
- OM3. Diseñar algoritmos utilizando las técnicas de representación básicas.
- OM4. Construir programas fiables y con buen estilo.
- OM5. Utilizar la abstracción en el proceso de desarrollo del software.
- OM6. Establecer procedimientos de prueba que permitan verificar los programas desarrollados.
- OM7. Utilizar herramientas de programación para facilitar y mejorar el proceso de desarrollo del software.
- OM8. Elaborar la documentación de los programas desarrollados.
- OM9. Conocer los conceptos básicos de la programación orientada a objetos y a eventos en entornos gráficos o programación visual.
- OM10. Respetar el ciclo de vida del software, siguiendo los pasos de desarrollo de software.

A continuación mostramos en una tabla la relación entre los objetivos del módulo y las capacidades terminales, es decir, qué capacidades terminales se desarrollan en cada objetivo del módulo.

	OM1	OM2	OM3	OM4	OM5	OM6	OM7	OM8	OM9	OM10
CT1	X	X	X	X	X		X		X	X
CT2						X	X			X
CT3							X	X		
CT4	X	X	X	X	X		X		X	X

También mostramos la relación entre los objetivos generales del ciclo formativo y las capacidades terminales, es decir, qué capacidades terminales se desarrollan en cada objetivo del ciclo formativo

	GC 1	GC 2	GC 3	GC 4	GC 5	GC 6	GC 7	GC 8	GC 9	GC 10	GC 11	GC 12	GC 13	GC 14	GC 15
CT 1	X	X			X										
CT 2		X		X											
CT 3			X							X					
CT 4	X	X			X	X									

2.3 Objetivos relacionados con temas transversales

En el currículo de la nueva Formación Profesional Específica aparece un nuevo concepto general que se denomina Temas Transversales, los cuales deben impregnar la práctica educativa y estar presentes explícitamente en los diferentes módulos profesionales.

Los temas que se consideran transversales en el currículo son:

- Educación Moral y Cívica.
- Educación para la paz.
- Educación para la salud.
- Educación para la igualdad de oportunidades entre los sexos.
- Educación del consumidor.
- Educación medioambiental.
- Educación vial.

Establecemos para nuestro módulo los siguientes objetivos adicionales relacionados con dichos temas transversales:

OM11. Resolver los conflictos mediante el diálogo, siendo transigentes y aceptando las ideas de los demás como bases de una convivencia en paz, así como respetar a todas las personas sea cual sea su condición social, sexual, racial o sus creencias.

OM12. Conocer y respetar las principales normas de ergonomía en el puesto de trabajo.

3 Contenidos

3.1 *Contenidos relacionados con los bloques temáticos del currículo*

A la hora de especificar los contenidos, distinguimos tres tipos.

- **Conceptuales:** representan el saber. Son los conocimientos de base necesarios.
- **Procedimentales:** representan el saber hacer. Son las habilidades y destrezas necesarias para desempeñar el puesto de trabajo.
- **Actitudinales:** representan el saber estar y actuar. Son las actitudes y características psicológicas favorables para desempeñar el puesto de trabajo.

La mayor parte de los contenidos debe ser de tipo procedimental, ya que la formación profesional tiene un claro referente ocupacional y práctico. Los contenidos se programan de cara a que el alumno adquiera las capacidades terminales.

En el Real Decreto 1676/1994 se establece el currículo del título de Técnico Superior en Desarrollo de Aplicaciones Informáticas. En dicho currículo se describen los contenidos de cada módulo, siendo los del módulo de programación en lenguajes estructurados:

- *Metodología de la programación.*
 - *Datos y algoritmos.*
 - *Datos: tipos y características.*
 - *Operaciones: tipos y características.*
 - *Estructuras básicas de tratamiento.*
 - *Algoritmos.*
 - *Programación estructurada.*

- *Programación modular.*
- *Recursividad.*
- *Estructuras de datos.*
 - *Estructuras estáticas de datos.*
 - *Organización secuencial.*
 - *Organización directa.*
 - *Organización secuencial-indexada.*
 - *Estructuras dinámicas de datos.*
- *Programación en lenguajes estructurados de tercera generación.*
 - *Lenguaje C.*
 - *Estructuras de control.*
 - *Funciones.*
 - *Estructura modular de un programa en C.*
 - *Librerías.*
 - *Desarrollo de programas.*
- *Introducción a la programación orientada a objetos.*
 - *Elementos, características y ventajas de la programación orientada a objetos*
 - *Objetos. Estructura, encapsulamiento, ocultación.*
 - *Relaciones entre objetos. Propiedades propias y heredadas.*
 - *Métodos.*
 - *Herencia: simple y múltiple.*
 - *Mensajes. Ligamento dinámico.*
 - *Lenguajes de programación que soportan la OOP.*

- *Utilidades para desarrollo y prueba de programas no incluidos en el entorno de programación.*
 - *Editores*
 - *Traductores*
- *Documentación de programas*
 - *Descripción de estructuras de datos utilizadas*
 - *Descripción de algoritmos*
 - *Código fuente*

A partir de estos contenidos podemos identificar los siguientes bloques de contenidos (BC):

BC1. Metodología de la programación.

BC2. Programación en lenguajes estructurados de tercera generación.

BC3. Herramientas de programación.

BC4. Documentación del software.

BC5. Programación orientada a objetos.

Con esta secuenciación de bloques de contenidos conseguimos que el alumno vaya adquiriendo de forma progresivo los conocimientos, habilidades y actitudes necesarias.

Cabe destacar que el último bloque es el de programación orientada a objetos, puesto que se trata de un contenido introducido en este módulo para luego ser desarrollado en otros módulos del ciclo formativo.

Otra decisión ha sido incorporar los contenidos de estructuras de datos dentro del bloque de programación en lenguajes estructurados de tercera generación. El criterio tomado ha sido organizar los contenidos de forma que se den los mínimos conocimientos de base para que al alumno pueda empezar a

programar trabajando directamente de la máquina, aumentando su motivación y favoreciendo el aprendizaje práctico.

Comentar también que aunque los reales decretos son de carácter prescriptivo, se pueden añadir otros contenidos siempre que sean adecuados para la formación y futura inserción laboral del alumnado. Un detalle a considerar podría ser el lenguaje de programación a emplear. Sin embargo, en este caso, las características del entorno hacen que el uso del lenguaje C, especificado en el Real Decreto de Currículo, sea una buena decisión, por lo que no es necesario enseñar otros lenguajes como Java o Pascal. Por otro lado, aunque el Real Decreto de Currículo no especifica ningún lenguaje concreto orientado a objetos, en esta programación se ha optado por C++, tanto por coherencia (pasar de C a C++) como por las características del entorno, con una gran demanda de programadores en lenguaje C/C++. Además, la propia naturaleza de ambos lenguajes hace que sea muy fácil aprender un nuevo lenguaje, mientras que al revés (sabiendo por ejemplo Pascal, y luego aprender C) suele ser más difícil.

Se ha elegido la plataforma Windows de entre todas las disponibles dado que es la plataforma de uso mayoritario en el entorno, por lo que es conveniente que nuestros alumnos se acostumbren a las herramientas que usarán cuando estén trabajando. Se ha elegido Visual C++ como implementación concreta de C++, ya que permite enseñar de forma sencilla conceptos de programación orientada a objetos y programación por eventos, además de ser una de las implementaciones usadas mayoritariamente en el entorno.

Veamos ahora cada bloque de contenidos en detalle, con la secuencia numerada de unidades de trabajo y los contenidos conceptuales y procedimentales a desarrollar en cada unidad.

Bloque 1: Metodología de la programación.

En este bloque se desarrollarán los contenidos básicos de metodología de la programación, centrándose en la programación estructurada.

UT1. Introducción a la programación estructurada.

En esta unidad presentaremos las características básicas de la programación estructurada, introduciendo al alumno en el módulo.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Procesos interactivos y por lotes.• Programación.• Lenguaje fuente.• Lenguaje máquina.• Programación estructurada.	<ul style="list-style-type: none">• Diferenciar la programación de procesos interactivos y por lotes.• Distinguir entre lenguaje fuente y lenguaje máquina.• Describir las características de la programación estructurada

UT2. Representación de algoritmos.

En esta unidad se introducirá al alumno el concepto de algoritmo, presentando las técnicas básicas de representación.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Definición de algoritmo.• Elementos de un algoritmo<ul style="list-style-type: none">○ Instrucciones○ Datos• Diagramas de flujo.• Pseudocódigo.	<ul style="list-style-type: none">• Diseñar algoritmos para resolver problemas concretos.• Usar la abstracción como herramienta de resolución de problemas.• Representar algoritmos con las técnicas básicas.• Elegir el tipo de dato idóneo según el caso.

UT3. Desarrollo del software.

Esta unidad presenta el ciclo de vida del software y explica los pasos de desarrollo del software. Después se centra en el paso de validación, que es fundamental para conseguir software fiable.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Ciclo de vida del software.• Pasos a seguir en el desarrollo del software.• Validación del software.	<ul style="list-style-type: none">• Planificar los pasos a seguir en el desarrollo del software.• Diseñar pruebas del software, mejorando la fiabilidad.

Bloque 2: Programación en lenguajes estructurados de tercera generación.

En este bloque se desarrollan los contenidos necesarios para que alumno pueda implementar programas en lenguajes estructurados de tercera generación, concretamente en el lenguaje C.

UT4. Lenguaje de programación C.

En esta unidad se introduce al alumno el lenguaje C, describiendo sus características básicas.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Tipos de datos.• Estructura de un programa.• Variables y constantes.• Estructura secuencial.Asignación. Sentencias de E/S.	<ul style="list-style-type: none">• Distinguir entre variables y constantes.• Elegir el tipo de dato idóneo según el caso.• Implementar algoritmos secuenciales en C.• Estructurar los programas de forma clara, consiguiendo un buen estilo.

UT5. Sentencias de control de flujo.

En esta unidad se explican las sentencias de control de flujo que se pueden utilizar en el lenguaje C.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Sentencias simples y compuestas.• Estructuras de selección.• Estructuras iterativas.	<ul style="list-style-type: none">• Implementación de algoritmos con estructuras de selección en C.• Implementación de algoritmos con estructuras iterativas en C.

UT6. Programación modular.

En esta unidad se explican los conceptos de la programación modular, aplicándolos al lenguaje C.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Módulos y funciones.• Paso de parámetros: por valor y por referencia.• Recursividad.	<ul style="list-style-type: none">• División de código en funciones.• Elección del tipo de paso de parámetros en función del caso.• Diseño e implementación de algoritmos recursivos.

UT7. Estructuras estáticas de datos.

En esta unidad se explican las estructuras estáticas de datos disponibles en C.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Tipos enumerados.• Vectores.• Arrays multidimensionales.• Matrices.• Cadenas de caracteres.• Registros.	<ul style="list-style-type: none">• Elección de la estructura de datos más adecuada según el caso.• Uso de las distintas estructuras de datos en C.

UT8. Métodos de búsqueda y ordenación.

En esta unidad se explican métodos de búsqueda y ordenación.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Búsqueda un vector: secuencial y dicotómica.• Métodos de ordenación: burbuja, selección, inserción.	<ul style="list-style-type: none">• Comparación de los métodos de búsqueda y ordenación, eligiendo el idóneo según el caso.• Implementación en C de los métodos de búsqueda y ordenación.

UT9. Ficheros.

En esta unidad se explica el concepto de fichero, y el uso de ficheros en

C.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Ficheros en C.• Modos de acceso.• Tipos de fichero según su contenido.	<ul style="list-style-type: none">• Identificación del tipo de fichero.• Elección del modo de acceso más adecuado según el caso.• Manipulación de ficheros en C.

UT10. Programación dinámica.

En esta unidad se introduce al alumno en la programación dinámica, centrándose en el lenguaje C.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Definición de puntero.• Operaciones con punteros.• Creación y liberación de memoria dinámica.	<ul style="list-style-type: none">• Manejar punteros en C.• Usar memoria dinámica en C.• Detectar el uso erróneo de punteros y/o memoria dinámica, evitando errores críticos en los programas implementados.

UT11. Estructuras dinámicas de datos.

En esta unidad se explican las estructuras dinámicas de datos, y su implementación en C.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Pilas.• Colas.• Listas.	<ul style="list-style-type: none">• Elección de la estructura dinámica más adecuada según el caso.• Implementación de estructuras dinámicas de datos en C.

Bloque 3: Herramientas de programación.

Este bloque consta de una única unidad didáctica.

UT12. Herramientas de programación.

En esta unidad didáctica se explican diversas herramientas de programación que ayudan al programador en diversas fases del desarrollo.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Editores.• Traductores.• Enlazadores.• Librerías de funciones.• Depuradores.• Sistemas de ayuda.	<ul style="list-style-type: none">• Identificación de las distintas herramientas de programación disponibles en el sistema.• Uso de las herramientas de programación.

Bloque 4: Documentación del software.

Este bloque consta de una única unidad didáctica.

UT13. Documentación del software.

En esta unidad se explican los pasos a seguir para documentar el software.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Documentación del código fuente.• Documentación de pruebas.• Guía de uso.• Guía de instalación.	<ul style="list-style-type: none">• Documentar el software de forma completa, estructurada y legible.• Uso de herramientas de generación de documentación.

Bloque 5: Programación orientada a objetos.

En este bloque se desarrollan los contenidos necesarios para que el alumno pueda implementar programas en lenguajes orientados a objetos, concretamente en el lenguaje C++.

UT14. Programación orientada a objetos.

En esta unidad se explican los conceptos fundamentales de la programación orientada a objetos.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Clases y objetos.• Herencia• Polimorfismo	<ul style="list-style-type: none">• Diseñar una solución software orientada a objetos especificando clases y objetos.• Aplicar los conceptos básicos de la programación orientada a objetos.

UT15. Programación en C++

En esta unidad se introduce al alumno el lenguaje C++ (entorno Visual C++), describiendo sus características básicas.

Conceptos	Procedimientos
<ul style="list-style-type: none">• Modelo de trabajo: orientado a objetos y eventos.• Entorno de trabajo.• Diseño de formularios.• Controles más habituales.• Menús.	<ul style="list-style-type: none">• Diseñar una interfaz de usuario en Visual C++.• Implementar clases y objetos en C++.

Los contenidos actitudinales a desarrollar en todas las unidades de trabajo son:

- Valoración de la importancia del trabajo en grupo.
- Respeto al trabajo, ideas y opiniones de los demás.
- Cumplimiento de las normas básicas de ergonomía en el puesto de trabajo.
- Gusto por el trabajo bien hecho y bien presentado, en tiempo y forma.
- Responsabilidad tanto en el trabajo en grupo como a nivel individual.
- Observación de las normas de uso de los recursos disponibles en el puesto de trabajo.
- Respeto a las personas que conforman la comunidad educativa.
- Verificación y contraste de la información obtenida a través de los medios proporcionados por las tecnologías de la información y la comunicación.

3.2 Contenidos sobre los temas transversales

En el apartado 1.3 de esta programación hemos especificado los objetivos del módulo (OM11 y OM12) en relación con los temas transversales. En este apartado vamos a desarrollar los contenidos asociados a dichos temas, así como su secuenciación. Dichos contenidos se van a secuenciar de manera transversal a lo largo de todo el módulo, de forma que a la finalización del mismo se hayan desarrollado completamente.

➤ OM11

- Contenidos actitudinales
 - Valoración de la importancia del trabajo en grupo.
 - Respeto al trabajo, ideas y opiniones de los demás.
 - Respeto a las personas que conforman la comunidad educativa.

- OM12
 - Contenidos conceptuales
 - Características de los puestos de trabajo con pantallas de visualización de datos.
 - Problemas oculares y visuales.
 - Trastornos músculo-esqueléticos.
 - Normas ergonómicas.
 - Actitudinales
 - Cumplimiento de las normas básicas de ergonomía en el puesto de trabajo.

3.3 Contenidos interdisciplinares

El Real Decreto de currículo establece que en el primer curso del ciclo formativo de Desarrollo en Aplicaciones Informáticas se impartirán los siguientes módulos:

- Sistemas informáticos multiusuario y en red (SMR).
- Análisis y diseño detallado de aplicaciones informáticas de gestión (AAIG).
- Programación en lenguajes estructurados (PLE).

Relación entre los módulos SMR y PLE

Por un lado, en el módulo SMR se desarrollan contenidos sobre el sistema operativo. Sería interesante que los profesores de ambos módulos se pusieran de acuerdo para desarrollar dichos contenidos de SMR antes del bloque de contenidos 3 (BC3) de PLE, ya que dicho bloque desarrolla los contenidos de las herramientas de programación disponibles en el sistema.

Por otro lado, en el módulo SMR se desarrollan contenidos sobre programación de scripts. Aunque en su caso más sencillo se trate de programación por lotes (ficheros BAT de DOS), existen implementaciones avanzadas (como los guiones shell de UNIX) que presentan características de los lenguajes estructurados. Por tanto, los profesores de ambos módulos se deben poner de acuerdo para desarrollar antes los contenidos del bloque de contenidos 1 (BC1) de PLE, ya que dicho bloque desarrolla los contenidos básicos sobre programación en lenguajes estructurados.

Relación entre los módulos AAIG y PLE

Por un lado, en el módulo AAIG se desarrollan contenidos sobre el ciclo de vida del software y los pasos del desarrollo. Si entendemos la programación como un paso concreto dentro del desarrollo del software, vemos que sería conveniente que se desarrollaran dichos contenidos en AAIG antes del bloque de contenidos 1 (BC1) de PLE, ya que dicho bloque desarrolla contenidos relacionados con el ciclo de vida del software y los pasos del desarrollo.

Por otro lado, en el módulo AAIG se desarrollan contenidos sobre el diseño modular. Considerando que la programación modular es una concreción del diseño modular, vemos que sería conveniente que se desarrollaran dichos contenidos en AAIG antes del bloque de contenidos 2 (BC2) de PLE, ya que dicho bloque de contenidos desarrolla contenidos sobre la programación modular.

En el módulo AAIG también se desarrollan contenidos sobre la documentación en general. Dichos contenidos se deben ver antes del bloque de contenidos 4 (BC4) de PLE, ya que dicho bloque desarrolla contenidos sobre la documentación de programas.

Finalmente, en el módulo AAIG se desarrollan contenidos sobre análisis y diseño orientado a objetos. Sabiendo que después del análisis y el diseño, viene la implementación, parece lógico coordinar ambos módulos de forma que se den dichos contenidos en AAIG antes del bloque de contenidos 5 (BC5) de PLE, ya que dicho bloque desarrolla contenidos sobre la programación orientada a objetos.

3.4 Temporalización de los contenidos

El Real Decreto de currículo establece que para nuestro módulo disponemos de 380 horas, que se distribuirán a razón de 12 horas semanales durante los tres trimestres del curso. La asignación de horas para las distintas unidades de trabajo queda como sigue.

- BC1
 - UT1: 12 horas
 - UT2: 24 horas
 - UT3: 24 horas

- BC2
 - UT4: 24 horas
 - UT5: 24 horas
 - UT6: 24 horas
 - UT7: 24 horas
 - UT8: 24 horas
 - UT9: 24 horas
 - UT10: 24 horas
 - UT11: 24 horas

- BC3
 - UT12: 24 horas
- BC4
 - UT14: 24 horas
- BC5
 - UT14: 36 horas
 - UT15: 36 horas

La suma de horas da 372, siendo las 8 horas restantes para la presentación del módulo (1 hora) y para posibles actividades extraescolares y complementarias (7 horas).

4 Metodología: orientaciones didácticas

En primer lugar, se debe mantener siempre el referente ocupacional en todas las enseñanzas impartidas. Esto quiere decir que se debe tener presente que el objetivo del ciclo formativo es preparar alumnos como futuros profesionales. Por tanto, se debe hacer hincapié en los contenidos procedimentales, si bien también es cierto que será necesario tener unos contenidos conceptuales de base. También es importante inculcar una serie de actitudes positivas y deseables en el trabajo, a través de los contenidos actitudinales.

Otro aspecto a tener en cuenta es que hay que fomentar el aprendizaje significativo. Los aprendizajes significativos constituyen información relevante para el alumno y que por tanto es fácil de recordar y asimilar. Para ir creando progresivamente una red de conocimientos en el alumno, debemos partir siempre de sus conocimientos previos. A la hora de introducir un nuevo concepto a los alumnos, hay que tratar de encontrar un punto de referencia y de interés que sirva como enganche y además motive el aprendizaje a fondo.

Las estrategias metodológicas a seguir por los profesores de la asignatura son:

- Método expositivo e interrogativo: consiste en el sistema clásico de enseñanza en que se imparten nuevos conocimientos, con el añadido de preguntar directamente a los alumnos para indagar en sus conocimientos previos, creando un vínculo entre los dos tipos de conocimientos.
- Método de aprendizaje por descubrimiento: consiste en proponer un problema a los alumnos, y que a través de unas indicaciones básicas sobre referencias bibliográficas y de consulta, sean capaces de encontrar la solución. Al haber intentado solucionar el problema, los alumnos que no lo consigan sí entenderán mejor la solución del profesor, y estarán más motivados.
- Método de aprendizaje por proyectos: consiste en asignar proyectos de trabajo individuales o en grupo, en que los alumnos deben alcanzar unos objetivos.

Con todo ello, planteamos los siguientes grupos de actividades que se concretarán en cada unidad de trabajo.

- Actividades de introducción y motivación: para presentar un tema nuevo y captar el interés de los alumnos.
- Actividades de desarrollo: para profundizar en los contenidos de un tema.
- Actividades de recuperación: para ayudar a los alumnos con un ritmo más lento de lo normal a alcanzar los mínimos exigibles.
- Actividades de ampliación: para que los alumnos con un ritmo más rápido de lo normal puedan profundizar en su aprendizaje, manteniendo el interés en clase.

A lo largo del módulo también se pueden desarrollar algunas de las siguientes actividades adicionales.

- Actividades extraescolares: visitas a empresas de programación para familiarizarse con el futuro entorno de trabajo de los alumnos.
- Actividades complementarias: charlas o debates impartidas por empresas o profesionales, para compartir su experiencia con los alumnos.

5 Criterios de evaluación

Se utilizará un sistema de evaluación continua en el ámbito de un sistema docente y académico de tipo presencial, por lo que la asistencia a clase será obligatoria.

Dado que el enfoque de la metodología didáctica a emplear es fundamentalmente procedimental, la evaluación dará mucha importancia a la realización de prácticas en el aula de informática y a la presentación de trabajos y ejercicios resueltos por parte de los alumnos.

Sin embargo, deberán realizarse determinados exámenes para comprobar el grado de asimilación de los contenidos conceptuales.

Para poder superar el nivel mínimo que requieren los objetivos de formación, y por tanto, aprobar el módulo profesional, los alumnos deberán:

- Superar todos los exámenes y controles escritos o en máquina realizados a lo largo del curso.
- Realizar, entregar y superar todos los ejercicios prácticos, trabajos, prácticas, supuestos teórico-prácticos, etc. y cualquier otro elemento evaluador de tipo procedimental, individual o en grupo, que sean establecidos por el profesor y que se desarrollen a lo largo del curso.

El Real Decreto de Título establece los criterios de evaluación (CE) para cada una de las capacidades terminales de cada módulo. Dichos criterios de evaluación permiten comprobar si los alumnos han adquirido las capacidades terminales. En el caso del módulo PLE tenemos.

- CT1
 - CE1.1: Diferenciar la programación de procesos interactivos y por lotes.
 - CE1.2: Describir la utilidad de las librerías y de los enlazadores de los sistemas operativos y depuradores, así como su forma de empleo.
 - CE1.3: Evaluar la importación de la claridad y legibilidad de los programas para facilitar el mantenimiento y el trabajo en equipo.
 - CE1.4: Elegir y definir estructuras de datos necesarios para la resolución del problema en un lenguaje estructurado.
 - CE1.5: Aplicar una metodología de desarrollo estructurado para el diseño de algoritmos.
 - CE1.6: Codificar programas en un lenguaje estructurado a partir de los algoritmos diseñados.
 - CE1.7: Aplicar estrategias de programación modular y de programación orientada a objetos, utilizando módulos, unidades o paquetes de desarrollo de programas.
 - CE1.8: A partir de un supuesto práctico o de una situación real en el aula

- Codificar un módulo de programación en un lenguaje estructurado de tercera generación.
 - Documentar el código de un módulo de programación con comentarios significativos, concisos y legibles.
- CT2
 - CE2.1: Describir tipologías de errores en los tratamientos.
 - CE2.2: A partir de un supuesto o de una situación real realizada en el aula
 - Aplicar estándares de control de calidad a partir de las especificaciones establecidas en el diseño y de las prestaciones esperadas por el usuario de la aplicación.
 - Realizar pruebas para cada módulo de una aplicación y pruebas de integración.
 - Comprobar que el acceso y el tratamiento de los datos cumplen las directrices de la administración de datos del sistema y las especificaciones establecidas en el diseño.
 - Comprobar que los formatos de entrada y salida de la aplicación son los esperados.
 - Comprobar que la funcionalidad de los datos es la esperada.
 - Provocar y verificar los diversos tratamientos de error.

- Medir los rendimientos de la aplicación y evaluar la eficiencia de las prestaciones de la aplicación y el consumo de recursos.
- CT3
 - CE3.1: Sobre una aplicación desarrollada en el aula
 - Documentar y describir las estructuras de datos utilizadas.
 - Elaborar matrices de referencia cruzadas que relacionen procesos y almacenamientos de datos.
 - Redactar guías de uso de las aplicaciones.
- CT4
 - CE4.1: Identificar los datos y módulos de programación afectados por la modificación de los requerimientos.
 - CE4.2: Utilizar matrices de referencias cruzadas entre módulos y datos para conocer las implicaciones que sobre éstos producen las nuevas modificaciones.
 - CE4.3: Probar que los nuevos datos y módulos no producen pérdidas de eficiencia y eficacia en las prestaciones ni en los rendimientos de la aplicación y satisfacen los nuevos requerimientos funcionales.
 - CE4.4: Documentar los cambios realizados sobre los datos, módulos y estructuras de datos y control de la aplicación.
 - CE4.5: Mantener librerías de módulos reutilizables.

A partir de los criterios de evaluación asociados a cada capacidad terminal, podemos especificar los siguientes criterios de evaluación asociados a los objetivos del módulo.

- OM1
 - Diferenciar la programación de procesos interactivos y por lotes.
 - Describir las características genéricas de la programación.
 - Explicar las distintas clasificaciones de los lenguajes de programación.
- OM2
 - Identificar los elementos básicos de un programa: algoritmo y datos.
 - Describir las características específicas de la programación estructurada
 - Clasificar las instrucciones típicas de los lenguajes estructurados según su función.
 - Describir las estructuras de datos típicas que maneja un lenguaje estructurado, su utilidad y ámbito de aplicación.
 - Sobre un problema de programación propuesto
 - Elegir las estructuras más adecuadas para representar y manejar los datos del problema.
 - Describir los algoritmos de tratamiento de las estructuras para la resolución del problema.
 - Diseñar un algoritmo para resolver el problema.
 - Implementar el algoritmo en lenguaje C.

- Aplicar técnicas de modularización.
- Diseño de algoritmos recursivos, y su implementación en C.
- OM3
 - Diseñar algoritmos para resolver problemas concretos.
 - Representar algoritmos mediante pseudocódigo y diagramas de flujo.
 - Traducir algoritmos de pseudocódigo a diagrama de flujo y viceversa.
- OM4
 - Distinguir entre la salida esperada y la salida real de un algoritmo.
 - Verificar la corrección de un programa, mejorando su fiabilidad.
 - Incluir código de comprobación en los programas.
 - Construir programas con un buen estilo, mejorando su legibilidad.
- OM5
 - Abstracter los problemas a resolver de las características específicas de los sistemas informáticos, obteniendo soluciones genéricas.
- OM6
 - Diseñar pruebas para cada módulo implementado.
 - Especificar los datos de prueba para un programa.

- Diseñar pruebas de integración de los distintos módulos de un proyecto software.
- OM7
 - Describir los distintos tipos de herramientas de programación y su uso.
 - Utilizar herramientas de programación en situaciones concretas, justificando su elección y uso.
 - Depurar el programa fuente.
 - Especificar los pasos a seguir para obtener el programa ejecutable a partir del programa fuente, nombrando las distintas herramientas de programación que intervienen en cada paso.
- OM8
 - Codificar un programa fuente en el lenguaje con comentarios significativos y concisos.
 - Elaborar la documentación de un programa, indicando si es posible el uso de alguna herramienta generación automática.
 - Guía de usuario.
 - Guía de instalación.
 - Documentación de pruebas.
- OM9
 - Describir las características de la programación orientada a objetos.

- Describir las características de la programación por eventos.
- Sobre un problema de programación propuesto
 - Diseñar una interfaz de usuario utilizando Visual C++.
 - Implementar las clases y objetos en C++.
- OM10
 - Describir el ciclo de vida del software.
 - Enumerar los pasos del desarrollo del software
- OM11
 - Realizar trabajos en grupo de forma satisfactoria, integrándose plenamente en él.
 - Contrastar y debatir las soluciones obtenidas en la resolución de actividades con las obtenidas por el resto de grupos y/o alumnos del curso.
- OM12
 - Enumerar los problemas potenciales sobre la salud derivados de la no observación de las principales normas de ergonomía en el puesto de trabajo.
 - Mantener una posición ergonómica correcta en el puesto de trabajo.

Una vez descritos los criterios de evaluación para los objetivos del módulo (qué evaluar), debemos indicar cuándo y cómo evaluar el proceso de aprendizaje de los alumnos.

Respecto a cuándo evaluar se harán las siguientes evaluaciones.

- Evaluación inicial: al principio de cada unidad de trabajo haremos una evaluación inicial diagnóstica que nos permitirá medir los conocimientos y habilidades previos de los alumnos.
- Evaluación parcial: al final de cada unidad de trabajo haremos una evaluación para medir los conocimientos y habilidades adquiridos por los alumnos.
- Evaluación final: al final del módulo haremos una evaluación final para medir en conjunto todos los conocimientos y habilidades adquiridos por los alumnos.

Respecto a cómo evaluar, se emplearán los siguientes instrumentos de evaluación.

La observación sistemática del proceso de aprendizaje a través del seguimiento directo de las actividades, precisando, en cada caso, lo que se pretende observar y los períodos de tiempo para la recogida de información. Utilizaremos para ello guías y fichas de observación que recogerán de forma concreta los elementos observables.	10%
El seguimiento y análisis de las producciones de los alumnos, individual o de grupo, a través de cuadernos de trabajo, presentaciones, entrevistas, trabajos monográficos, cuestionarios, etc.	40%
Los debates y presentaciones que permitirán observar y obtener información sobre aspectos actitudinales, de integración y actuación social.	10%
La aplicación de pruebas específicas, orales o escritas, abiertas o cerradas para la evaluación de determinados contenidos.	40%

6 Atención a alumnos con necesidades educativas específicas

Por un lado, se plantearán actividades de ampliación y recuperación para aquellos alumnos cuyo ritmo de aprendizaje sea más rápido o más lento de lo normal.

Por otro lado, están los alumnos que presentan discapacidades. Para estos casos, lo que debemos hacer es acudir al departamento de Orientación para ver qué medidas podemos tomar. De cara a posibles adaptaciones curriculares, no debemos olvidar que el título de técnico superior en Desarrollo de Aplicaciones Informáticas habilita al alumno para realizar un trabajo, por lo que se deben alcanzar las capacidades terminales de cada módulo. En todo caso, sí se pueden hacer adaptaciones en la metodología en función de las necesidades específicas del alumno.

7 Materiales y Recursos Didácticos

- Aula de informática, con entorno de red, sistema operativo Windows, compilador C y Visual C++ instalado en cada ordenador.
- Pizarra.
- Retroproyector para mostrar la salida del ordenador del profesor a los alumnos.
- Apuntes de clase elaborados por el profesor.
- Libros de consulta.
 - Programación en C, de “Bryan S. Gottfried”.
 - Este libro cubre todos los aspectos importantes del lenguaje C, incluyendo temas tan específicos como la mezcla con código de otros lenguajes. Sin duda, se trata de una obra fundamental para cualquier persona que programe en C.
 - Fundamentos de Programación, de “Armando Suárez Cueto” y otros.
 - Este libro es un buen punto de partida para alguien que quiere empezar a programar desde cero. En los primeros temas se introducen los conceptos básicos de la lógica booleana, y la representación de algoritmos mediante diagramas de flujo y pseudocódigo. Después comenta los aspectos más importantes de C.

- C++ Primer, de “Stanley B. Lippman”.
 - Este libro describe de forma exhaustiva todas las prestaciones de C++. Es interesante descubrir todo lo que ofrece el lenguaje, más allá de la implementación de las características propias de la programación orientada a objetos.
- C++ versión 3.0, de “Antonio Corbí”.
 - Este libro permite realizar una fácil y rápida transición de C a C++, describiendo los cambios más importantes y las novedades más significativas.
- Introducción a la programación orientada a objetos, de “Timothy Budd”.
 - Este libro es muy recomendable para entender los conceptos de la programación orientada a objetos, independientemente del lenguaje de programación utilizado.
- Enlaces web de interés
 - www.programacion.net
 - www.microsoft.com/msdn
 - www.google.com