

La Universidad Española: Contenidos sobre Programación en los primeros cursos de las Titulaciones en Informática

Pedro J. Clemente y Pedro L. Pérez

Dpto. de Informática, Universidad de Extremadura.
Quercus Software Engineering Group. <http://quercusseg.unex.es>
e_mail: {jclemente, plperez}@unex.es

Resumen

Los planes de estudio establecen los conocimientos y habilidades que los alumnos deben desarrollar. En muchos casos, estos planes de estudio no están suficientemente adecuados a las necesidades actuales, y deben actualizarse y afrontar los cambios sufridos, no sólo a cambios tecnológicos, sino también cambios sociales y docentes. Sin embargo, no cabe duda que mantener actualizada la planificación de las asignaturas de informática resulta toda una tarea de investigación. El profesor debe documentarse sobre las últimas currícula internacionales, así como mantener una visión global de los contenidos que se imparten en otras universidades, nueva bibliografía, etc. Con el fin de ayudar en esta revisión de contenidos, en este artículo se presenta un estudio sobre los contenidos de programación que se imparten en la universidad española. Para ello se tiene en cuenta el número de créditos aplicados, los paradigmas y lenguajes de programación utilizados, así como los contenidos impartidos. Por último, se presenta una propuesta para la distribución de estos contenidos durante los primeros cursos de las Titulaciones de Informática.

1. Introducción

Cuando un profesor se plantea organizar los contenidos de una determinada materia, normalmente necesitaría más créditos de los que dispone. Por tanto, se debe ajustar los contenidos a los créditos de que dispone, y por tanto, se debe buscar un equilibrio entre contenidos y profundidad con que estos se imparten. En la toma de decisiones se deben tener en cuenta las recomendaciones curriculares más representativas (IS'97[1], CC91[2],

CC2001[3], MSIS'2000[9], IFIP'94[11], ICF'2000[12], etc.) las directrices que ofrece el MEC y aproximaciones curriculares nacionales, así como la situación en la universidad española de las asignaturas relacionadas con la materia.

La recopilación de información sobre los contenidos que se imparten en otras universidades y que pueden servir como referencia la hora de realizar una nueva planificación o plan de estudio resulta una labor tediosa. Sin embargo, es necesaria para establecer los contenidos que otros compañeros imparten en sus universidades.

En este trabajo se realiza un estudio sobre los contenidos en programación de las universidades españolas en los primeros cursos de las Titulaciones de Informática (sección 2). Posteriormente se presenta una propuesta que establece los contenidos a impartir en los primeros cursos de estas Titulaciones (sección 3). Esta propuesta incluye temporización, paradigmas y lenguajes de programación a utilizar para su desarrollo. Dicha propuesta está basada en el estudio de los planes de estudio y contenidos impartidos en las universidades españolas, las aproximaciones realizadas por diferentes currículas internacionales [1][9][11][12], centrándonos en las recomendaciones para los primeros cursos realizada por *Computing Curricula 2001* de ACM/IEEE [3], experiencia propia y de otros profesores en la materia.

2. La situación actual en la Universidad Española

En esta sección se presenta un estudio sobre los contenidos que se imparten sobre programación en la universidad española. Para la elaboración de este resumen se han tenido en cuenta las

asignaturas tanto troncales como obligatorias destinadas a impartir las directrices *Metodología y Tecnología de la Programación* y *Estructura de Datos y de la Información*, (esta última desde el punto de vista de *Estructuras de Datos y Algoritmos*, es decir, sin entrar en el tópico *Bases de Datos*)[10], en las universidades que imparten la titulación de Ingeniería Informática.

En primer lugar describiremos cuál es la distribución de créditos para los contenidos de primer y segundo curso, así como su distribución en asignaturas. Posteriormente, se analizará cuál es el primer paradigma de programación utilizado en las Universidades españolas, así como los lenguajes de programación utilizados. Por último, se presentan los contenidos impartidos tanto en primer y como en segundo curso¹.

2.1. Los Créditos destinados a programación

La primera situación que nos encontramos es la existencia de aproximadamente el mismo número de créditos en primer y segundo curso relacionados con programación. Concretamente, de media se ofertan 16,6 créditos para asignaturas de primer curso, y una media de 15 créditos para asignaturas de segundo curso. En ambos casos, suele existir una distribución pareja: dos asignaturas cuatrimestrales en primer curso y dos asignaturas cuatrimestrales en segundo curso.

Sin embargo, existen picos sobre estos datos, como los proporcionados por la Universidad de Oviedo que oferta 15 créditos en primer curso y 24 créditos en segundo curso, con lo cual su oferta global de contenidos incluye contenidos de forma más extensa (Programación dirigida por eventos o programación en entorno Windows en primer curso, o por ejemplo, estudio detallado de metodología orientado a objetos y programación funcional en segundo curso).

En cuanto a la distribución de contenidos teóricos y prácticos: En primer curso, de los 16,6 créditos disponibles 9,2 corresponden a créditos teóricos y 7,4 a créditos prácticos. En

cuanto a segundo curso, de los 15 créditos se distribuyen en 9 teóricos y 6 prácticos.

Desde nuestro punto de vista, el número de créditos es insuficiente para proveer al alumno de los contenidos requeridos actualmente[3]. Por ello, se debería incrementar el número de créditos medio para hacer frente a los nuevos contenidos que demanda la sociedad, y que no estaban recogidos en los reales decretos que regulan las Titulaciones en Informática[10], como pueden ser los contenidos sobre Programación Orientada a Objetos.

2.2. El paradigma de programación

Desde el punto de vista del paradigma de programación utilizado en las diferentes asignaturas podemos apreciar en la Figura 1 la distribución correspondiente a las asignaturas de primer curso y en la Figura 2 segundo curso.

Vamos a centrar nuestra atención en los paradigmas impartidos en primer curso, y en este sentido, es el paradigma imperativo el predominante (66%), aunque también se aprecia que el paradigma orientado a objetos tiene una amplia aceptación (27%). Es importante recordar que como media en primer curso las universidades ofertan dos asignaturas, y es bastante común que en la primera de ellas se imparta el paradigma imperativo y en la segunda una aproximación orientada a objetos. Pocas universidades imparten como primer paradigma el orientado a objetos, tal y como ocurre, por ejemplo, en la Universidad Carlos III de Madrid.

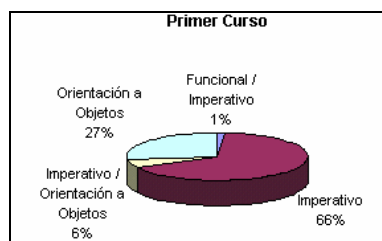


Figura 1. Distribución de los paradigmas de programación en Primer Curso

En el primer curso, el paradigma imperativo se impone en el primer cuatrimestre. En el segundo cuatrimestre cada vez más

¹ Los datos que forman este estudio han sido recopilados a finales de 2002.

universidades adoptan una aproximación orientada a objetos.

Analizando la situación para el segundo curso se observa un avance importante del paradigma orientado a objetos (en el 60% de las asignaturas). Casi el total de las universidades imparte este paradigma en alguna de las asignaturas de segundo curso.

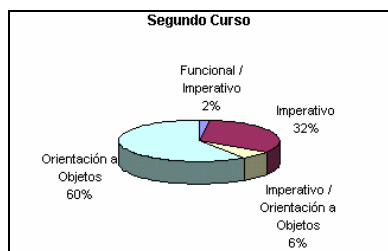


Figura 2. Distribución de los paradigmas de programación en Segundo Curso

Como conclusión destaca que en la Universidad Española el primer paradigma que se introduce es el paradigma imperativo. Posteriormente, en muchos casos en el segundo cuatrimestre, se introduce el paradigma orientado a objetos. En segundo curso el paradigma predominante es el orientado a objetos, siguiendo la estructura que se propone desde el enfoque *Primero Imperativo* en CC2001[3], donde los alumnos reciben formación sobre el paradigma imperativo inicialmente para pasar a estudiar el paradigma orientado a objetos posteriormente.

2.3. El Lenguaje de Programación

Un elemento de controversia en la comunidad docente en *Metodología de la Programación y Estructuras de Datos y Algoritmos* es el lenguaje de programación utilizado. Por ello, hemos analizado los lenguajes de programación utilizados tanto en primer, como en segundo curso.

En la Figura 3 podemos observar la distribución de los lenguajes de programación utilizados en primer curso. Podemos observar que aparecen destacados los lenguajes de programación *híbridos* (C++, Pascal). Esta

situación se debe a las decisiones en cuanto al paradigma de programación utilizado. Hemos comentado en la sección anterior que se impone el paradigma imperativo en el primer cuatrimestre de primer curso, y que comienza a introducirse el paradigma orientado a objetos en el segundo cuatrimestre. Por tanto, es habitual utilizar un lenguaje híbrido para el desarrollo de las asignaturas, ya que el alumno no necesita cambiar de lenguaje para pasar del paradigma imperativo al orientado a objetos[5]. ¿Por qué C++ es el lenguaje más utilizado? Entendemos que además de tratarse un lenguaje híbrido, que podemos utilizar para ilustrar el paradigma imperativo y orientado a objetos, se trata de uno de los lenguajes de programación más utilizados en la empresa y esta situación motiva también a los alumnos a aprenderlo.

Las universidades que apuestan por impartir como primer paradigma el paradigma orientado a objetos, por ejemplo, las Universidades de Almería y Carlos III de Madrid, imparten Java (un lenguaje orientado a objetos calificado como *puro*). Como se ha comentado, esta no es la situación habitual. Por tanto, como podemos ver en la Figura 3, el porcentaje de asignaturas que imparten Java en primer curso está motivado por el sesgo orientado a objetos que sufren las asignaturas en el segundo cuatrimestre.

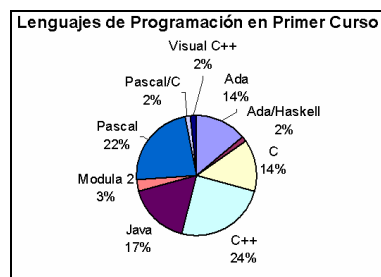


Figura 3. Distribución de los lenguajes de programación en Primer Curso

Los lenguajes de programación predominantes utilizados en segundo curso (ver Figura 4), donde se imparte el paradigma orientado a objetos, son C++(32%) y Java (28%). Podemos observar cómo C++ sigue manteniendo la hegemonía, seguido por Java, un

lenguaje creado desde sus orígenes orientado a objetos e implantado fuertemente en Internet, lo cual ofrece un gran atractivo al alumno.

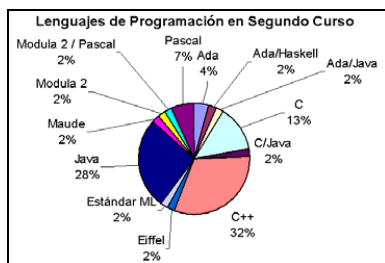


Figura 4. Distribución de los lenguajes de programación en Segundo Curso

2.4. Contenidos Primer Curso

En cuanto a los contenidos de primer curso, la primera consideración importante es que ambas directrices, Metodología y Tecnología de la Programación y Estructura de Datos y de la Información (esta última desde el punto de vista de Estructuras de Datos y Algoritmos)[10] y sus tópicos se imparten normalmente tanto en primer curso como en segundo curso, es decir, las asignaturas de primer curso imparten nociones de metodología de la programación y estructuras de datos, y las asignaturas de segundo curso profundizan y amplían los contenidos tanto de metodología de programación como de estructuras de datos y algoritmos.

Vamos a describir cuales con los contenidos que habitualmente se desarrollan, según la información obtenida sobre las asignaturas que se imparten en las universidades españolas. Los contenidos incluyen su correspondiente referencia en Computing Currícula 2001[3], lo cual muestra inicialmente la adaptación de contenidos a está currícula, considerada como la más completa.

Los principales temas que los alumnos encuentran en las asignaturas de primer curso relacionadas con las directrices MEC estudiadas son:

- **Conceptos Fundamentales:** Introducción a los conceptos fundamentales de las metodologías de programación, breve historia de los lenguajes de programación (*SP1 en CC2001*). Recorrido por los diferentes paradigmas de programación: procedurales, orientados a objetos, funcionales, declarativos. Proceso de compilación, interpretación. Diferencias coste/portabilidad. (*PL1 de CC2001*).
- **Algoritmos y Resolución básica de problemas:** Estrategias de resolución de problemas, algoritmos y su representación. Metodologías de diseño de algoritmos, diseño descendente. (*PF1 y PF2 de CC2001*).
- **Programación Estructurada:** Se imparten las diferentes estructuras de control de flujo: secuenciales, de selección e iteración (*PF1 de CC2001*).
- **Tipos de Datos y Acciones Elementales:** Elementos básicos y tipos elementales: variables, tipos, expresiones, asignaciones, acciones elementales. (*PL4 de CC2001*).
- **Programación Modular:** Mecanismos de abstracción básicos: Funciones, procedimientos, módulos de los lenguajes de programación. Recursividad. Paso de parámetros tanto por valor como por referencia. Variables y ámbito de las mismas (*PL5 de CC2001*).
- **Análisis de algoritmos:** El estudio de la eficiencia de los algoritmos se estudia inicialmente en las asignaturas de primero. Por ejemplo, en la Universidad Autónoma de Barcelona o en la Universidad de Extremadura se estudia sólo complejidad en algoritmos iterativos. Se pospone para segundo curso el estudio profundo del análisis de algoritmos. (*AL1 de CC2001*).
- **Tipos de Datos definidos por el usuario: Vectores, Matrices, Registros.** En la totalidad de los planes de estudio se imparten estos conceptos y estructuras fundamentales. (*PF3 de CC2001*)
- **Algoritmos de ordenación y búsqueda:** Se trata de algoritmos de manipulación de estructuras de datos lineales (búsqueda secuencial y binaria) (*AL3 de CC2001*)
- **Tipos abstractos de datos.** En la gran parte de los planes de estudio consultados se

- imparten los Tipos Abstractos de Datos fundamentales: TAD Pila, TAD Cola, TAD Lista, e incluso TAD Árbol, aunque no es lo habitual. (AL2 en CC1991)
- *Punteros, gestión dinámica de memoria, estructuras dinámicas, listas enlazadas.* Conceptos ampliamente extendidos en los planes de estudio y en los contenidos que se imparten en primer curso. (PF3 en CC2001)
 - *Introducción a la Programación Orientada a Objetos:* La introducción del paradigma orientado a objetos es significativa. Sin embargo, sólo en algunas universidades se introduce los conceptos de herencia y polimorfismo, tal y como ocurre en la Universidad de Castilla la Mancha donde se ofrece una asignatura específica de Programación Orientada a Objetos en el segundo cuatrimestre, donde se imparte el lenguaje Java. Lo mismo ocurre en las aproximaciones de la Universidad Carlos III y en la Universidad de Almería donde el primer paradigma que se estudia es el orientado a objetos. Sin embargo, es bastante frecuente encontramos aproximaciones *centrada en objetos*[4], es decir, donde se realiza una aproximación a la programación orientada a objetos, describiendo, clases, objetos, métodos, atributos, características de encapsulación, pero dejando los conceptos de herencia y polimorfismo para profundizar en las asignaturas de segundo curso. (PL6 de CC2001)
 - *Ficheros.* Introducción al almacenamiento en memoria secundaria. Se tratan los conceptos introductorios, justificación, acciones básicas sobre ficheros, clasificación (secuenciales & acceso directo), algoritmos de ordenación (mergesort y quicksort) y fusión. (PF3 en CC2001)
 - *El proceso de desarrollo software.* Desde las asignaturas de primer curso se introduce la necesidad de una metodología para el desarrollo software. Este proceso se introduce en función del paradigma utilizado (Ciclo de vida del software estructurado, orientado a objetos, etc.). (SE1 en CC2001)
 - *Depuración, Prueba de programas y Documentación.* (SE6 en CC2001)
 - *Especificación Formal.* Sólo en algunas universidades. Aunque se introduce la necesidad de especificar los TADs, sin embargo, se utiliza una notación no formal. (DS3 en CC2001)
 - *Algoritmos numéricos:* El estudio de algoritmos numéricos se estudia en algunas universidades, como la universidad de Vigo, donde el número de créditos para primer curso es superior a la media (Vigo 27 Créditos para primer curso) (AL3 en CC2001).

2.5. Contenidos Segundo Curso

En este apartado vamos a destacar los principales temas que los alumnos encuentran en las asignaturas dedicadas a programación de segundo curso:

- *Análisis de Algoritmos:* Se trata de uno de los aspectos fundamentales para segundo curso en las universidades españolas. Se estudia la notación asintótica, cálculo de la eficiencia de algoritmos tanto en iterativos y recursivos. Este estudio se plasma con el cálculo de la eficiencia de diferentes algoritmos de búsqueda y ordenación. (AL1 en CC2001)
- *Especificación y Verificación formal:* Se imparte en profundidad en varias universidades. Por ejemplo en las asignaturas *Programación metódica* en la Politécnica de Cataluña, *Metodología de la programación* en las Universidades del País Vasco y Zaragoza, o *Programación II* en la Universidad Nacional de Educación a Distancia. (DS3 en CC2001)
- *Tipos Abstractos de Datos:* La especificación de tipos abstractos de datos, bien con especificación formal e informal es un contenido típico en la universidad española, posiblemente influenciada por CC1991[2]. Por ejemplo, encontramos Tipos Abstractos de Datos en la asignatura Algoritmos y Estructuras de Datos en la Universidad de Murcia, o incluso como asignatura específica, véase la asignatura Tipos Abstractos de Datos en la Universidad de Málaga. (AL2 en CC1991)

- *Estructuras de Datos*: Estructuras de datos lineales (Conjuntos, Pilas, Colas y Listas)(PF3 en CC2001), Árboles y Grafos (DS5 en CC2001)(PF3 en CC2001). Por ejemplo, en la Universidad Autónoma de Barcelona se imparte una asignatura completa para el estudio de Grafos (Grafs i complexitat). Para estas estructuras se estudia su especificación, representación, diferentes implementaciones, algoritmos de ordenación y búsqueda (pila, colas y listas), algoritmos de coste mínimo (grafos), árboles de expansión mínima (grafos). (AL3 de CC2001)
- *Tablas*: Estudio de Tablas, su especificación e implementación. Estudio de Tablas Hash y los técnicas de resolución de colisiones (PF3 en CC2001)
- *Técnicas de Diseño de Algoritmos*: El estudio de las principales técnicas de diseño de algoritmos se encuentra en la mayoría de las universidades. Los tópicos incluidos en este estudio son: Programación Dinámica, Algoritmos Voraces, Algoritmos Probabilísticos, Algoritmos heurísticos, así como Ramificación y Poda. En algunas universidades disponen de asignaturas específicas donde este tema es el núcleo de la misma. Por ejemplo, la asignatura *Teoría de Algoritmos* de la Universidad de Granada, la asignatura *Análisis y Diseño de Algoritmos* en la Universidad de Sevilla o *Metodología de Programación* en la Universidad de Castilla la Mancha. (AL2 en CC2001)
- *Programación Orientada a Objetos*: En segundo curso se imponen el paradigma orientado a objetos, tal y como hemos visto en las secciones anteriores. Este paradigma se estudia en profundidad en las asignaturas de segundo curso, completando en su caso los conceptos de orientación a objetos que los alumnos han recibido en primer curso (PL6 en CC2001).
- *Metodología de Programación Orientada a Objetos*: En algunos casos, se ofertan asignaturas como *Laboratorio de Tecnología Orientada a Objetos* de la Universidad de Málaga o *Metodología de Programación II* de la Universidad de Oviedo donde se incluyen metodologías de

análisis y diseño orientado a objetos, técnicas de depuración y documentación, utilización de librerías y API y programación dirigida por eventos (SE1, PL6, SE2 en CC2001).

En esta sección hemos realizado un estudio sobre los contenidos que comúnmente se imparten en las asignaturas de programación dentro de la Universidad Española. En la siguiente sección se presenta una distribución de estos contenidos para 1^{er} y 2^o curso.

3. Propuesta de distribución de contenidos en 1^{er} y 2^o curso

A la hora de realizar la planificación de los planes de estudio es importante establecer de forma adecuada requisitos y correquisitos. Con este fin, las asignaturas donde se imparten los contenidos sobre programación deben formar bloques muy bien diferenciados, donde se determine adecuadamente sus requisitos, y los conocimientos que se proveen tanto desde el punto de vista teórico como práctico. En este sentido deben evitarse situaciones como la siguiente: planificar asignaturas que comparten temario durante un mismo cuatrimestre, donde una de ellas imparte los temas teóricos que deben ser aplicados en las clases prácticas de una segunda asignatura. Esta situación, aparece en algunos los planes de estudio de la universidad española, donde se mezclan asignaturas anuales con cuatrimestrales, asignaturas eminentemente teóricas con asignaturas totalmente prácticas, lo cual dificulta una clara división de contenidos y plantea problemas de planificación.

Desde nuestro punto de vista, una distribución de asignaturas cuatrimestral, donde sea factible realizar una clasificación de requisitos y correquisitos, unido a una relación teoría/práctica equilibrada en las asignaturas permitirá a los alumnos asimilar paulatinamente los contenidos sobre la materia. En esta relación se deben evitar los temarios de contenidos excesivos donde únicamente se ofrezcan *pincladas* y no se abordan los temas en profundidad.

La propuesta que a continuación se presenta está desarrollada teniendo en cuenta las Directrices del MECD, currículas

internacionales y así como recomendaciones nacionales en la materia[5][6][7][8], análisis de los planes de estudio y estudio de la materia en otras universidades (ver sección 2), experiencia propia de la materia y de otros profesores.

Para conseguir los objetivos planteados se presentan las siguientes asignaturas troncales: *Programación I y Estructuras de Datos I* para primer curso, y *Programación II y Estructuras de Datos II* para segundo curso. En segundo curso se añade una obligatoria *Software Orientado a Objetos*. Esta nueva obligatoria responde a la necesidad de ajustar los contenidos y créditos disponibles para el bloque de programación, donde la POO requiere un espacio para su estudio específico.

La distribución de contenidos y créditos puede apreciarse en Tabla 1. Debemos notar que cada asignatura dispone de un conjunto de créditos prácticos mediante los cuales el alumno debe familiarizarse y asimilar los contenidos impartidos en las clases teóricas.

Curso	Cuat.	Asignatura	Créditos
1	1	Programación I	7,5 (4,5T+3P)
1	2	Estructuras de Datos I	7,5(4,5T+3P)
2	1	Programación II	7,5(4,5T+3P)
2	1	Software Orientado a Objetos	6(4,5T+1,5P)
2	2	Estructuras de Datos II	7,5(4,5T+3P)

Tabla 1. Distribución de asignaturas en 1^{er} y 2^o Curso

En esta propuesta se introduce al alumno en programación mediante el paradigma imperativo, para introducir posteriormente los conceptos iniciales de programación orientada a objetos, que será el paradigma que se utilice a partir del segundo cuatrimestre de primer curso. En cuanto al lenguaje de programación utilizado, se opta por C++ por los motivos descritos en la sección 2, y que llevan a este lenguaje a ser el más utilizado en la universidad española. Sin embargo, en segundo curso, se

propone un segundo lenguaje de programación: Java, que nos permite una aproximación más completa de la POO, al menos, en cuanto a características específicas de POO como pueden ser las interfaces. A continuación se detalla brevemente los contenidos a impartir en cada una de las asignaturas:

3.1. Programación I

Se trata de la primera asignatura sobre programación que estudiarán los futuros titulados. Para ello se realiza una aproximación a la programación imperativa, eficiencia de algoritmos, estructuras fundamentales (registros y vectores), para terminar con una introducción a la programación orientada a objetos. Podemos establecer tres módulos bien diferenciados:

Un primer módulo de *introducción a la programación imperativa*. Este módulo pretende ofrecer una visión global sobre la programación imperativa al alumno. Los contenidos de este módulo corresponden a: Conceptos Fundamentales, Algoritmos y Resolución básica de Problemas, Programación Estructurada, Tipos de Datos y Acciones Elementales, Programación Modular, Depuración y Prueba de Programas, Documentación. Conceptos de eficiencia de los algoritmos, donde se estudia el cálculo de la eficiencia en algoritmos iterativos.

Un segundo módulo donde se presenta el concepto de *Tipo Abstracto de Datos* como herramienta de abstracción. En este módulo se describe el tratamiento de *vectores* introduciendo los aspectos básicos de su manejo, así como los algoritmos típicos de ordenación y búsqueda, que pueden ser utilizados como banco de prueba de los conceptos sobre eficiencia.

Un tercer módulo donde se presenta la *programación orientada a objetos*. Donde los alumnos estudiarán los conceptos básicos y las principales características de la programación orientada a objetos, sin entrar en detalles sobre herencia y polimorfismo. Se debe incidir sobre la capacidad de abstracción del mundo real mediante objetos.

3.2. Estructuras de Datos I

Esta asignatura de segundo cuatrimestre permite introducir al alumno en las primeras estructuras

de datos, concretamente estructuras lineales, implementadas de forma estática y dinámica, así como la gestión de datos persistentes en ficheros.

Inicialmente debe realizarse un breve resumen sobre conceptos básicos de POO introducidos en la asignatura requisito (Programación I). A continuación, se estudiarían *Estructuras Lineales*. En este módulo se estudia la implementación de las estructuras lineales más usuales (Pilas, Colas, Listas, Conjuntos).

El estudio de la *gestión dinámica de memoria* (punteros, variables dinámicas, reserva y liberación de memoria) nos permitirá introducir al alumno las implementaciones dinámicas de las estructuras de datos lineales estudiadas. Por último es conveniente introducir al alumno en el almacenamiento persistente de datos (módulo sobre *ficheros*). Posiblemente este tema sea de los que más interés despierta en los alumnos, ya que abre todo un mundo de posibilidades para sus aplicaciones. Desde el punto de vista teórico, se deben explicar la necesidad de almacenamiento persistente de datos, los conceptos y operaciones básicas, clasificación de ficheros (secuencial y acceso directo) (texto y binario), algoritmos básicos de inserción y borrado, así como fusión y ordenación de ficheros.

3.3. Programación II

Esta asignatura tiene como objetivo introducir al alumno en el análisis de algoritmos, las técnicas más usuales de diseño de algoritmos, así como introducir la especificación y verificación formal (esta asignatura tiene como requisito Programación I).

En un primer módulo sobre *análisis de algoritmos* se presentan los conceptos necesarios para analizar los algoritmos desde el punto de vista de la eficiencia. Dichos conceptos son los que posteriormente deben ser utilizados para razonar sobre la complejidad computacional de los algoritmos construidos. En un segundo módulo presentan las principales *técnicas de diseño de algoritmos* (Divide y vencerás, Algoritmos Voraces, Vuelta atrás o backtracking, Ramificación y poda, Programación dinámica). Se debe razonar sobre su aplicabilidad y adecuación a la hora de

resolver un problema. Las diferentes estrategias pueden presentarse con su esquema general y un conjunto de ejemplos de aplicación de cada una de las técnicas. Por último, un módulo sobre *especificación algebraica y verificación formal* se ofrece una visión de la programación desde el punto de vista diferente, que permita a los alumnos familiarizarse con la verificación y derivación formal de programas.

3.4. Software Orientado a Objetos

En esta asignatura se extienden los conceptos sobre programación orientada a objetos presentando las características de herencia y polimorfismo, y los conceptos asociados tipos de herencia, ligadura tardía, métodos virtuales, etc. En esta asignatura se debe presentar una introducción a alguna metodología para el desarrollo de aplicaciones software orientado a objetos (diagramas de clases, diagramas de interacción, etc.)².

En un curso de estas características es importante que el alumno no adquiera la visión de OO ofrecida por un lenguaje en particular, sino que comprendan los conceptos que subyacen a la OO de una forma independiente del lenguaje[7].

Como complemento a las clases teóricas se introduce el lenguaje de programación Java. Que será el lenguaje de programación utilizado en las clases prácticas, si embargo, y como se ha comentado se recomienda la exposición de los conceptos sobre orientación a objetos mediante ejemplos en varios lenguajes, y no centrar los conceptos en la implementación concreta realizada por un lenguaje de programación.

3.5. Estructuras de Datos II

En este módulo se parte de los conceptos estudiados sobre eficiencia de algoritmos y sobre especificación y verificación formal impartidos en Programación II, así como los conceptos sobre orientación a objetos impartidos en Software Orientado a Objetos, y se presentan

² Sería recomendable que la metodología seleccionada se desarrollase completamente en una asignatura de tercer curso

las estructuras de datos básicas: Pilas, Colas, Listas, Conjuntos, Tablas, Árboles y Grafos. La exposición de cada estructura de datos puede comenzar con una introducción sobre la estructura de datos de estudio y su especificación algebraica, a continuación, presentar diversas alternativas de implementación, analizando la eficiencia de cada uno de los algoritmos de manipulación implementados. Especial interés requieren los temas dedicados a las estructuras arbóreas y grafos, dada la utilidad de las mismas y el amplio conjunto de algoritmos susceptibles de ser estudiados.

4. Conclusión

Se ha presentado un estudio sobre los contenidos relacionados con programación que se imparten en la universidad española. De este estudio destaca que en la universidad española el paradigma de programación que se presenta al alumno en primer curso es el imperativo, sin embargo, se observa una mayor implantación del paradigma orientado a objetos en el segundo cuatrimestre de primer curso y así como en el segundo curso. Dada estas características de cambio de paradigma de programación, los lenguajes de programación híbridos son los más utilizados. Por último, cabe destacar en cuanto a los contenidos impartidos (en primer y segundo curso) la coincidencia con la propuesta *Primero Imperativo* de Computing Curricula 2001[3] para estos cursos iniciales.

Por otro lado, se ha presentado una propuesta de plan de estudios para los contenidos sobre programación en los primeros cursos de las Titulaciones en Informática. Establecer un equilibrio entre los contenidos teóricos y prácticos de las asignaturas, dando la oportunidad a los alumnos de afrontar dichos contenidos bajo una constante componente práctica. Esta componente práctica ayuda a los alumnos en el desarrollo de las habilidades, aptitudes y conocimientos requeridos.

Referencias

[1] ACM-AIS-AITP Joint Task Force. *IS'97 Model Curriculum and Guidelines for*

Undergraduate Degree Programs in Information Systems, 1997.

- [2] ACM/IEEE. *Computing Curricula 1991*. Report of the ACM/IEEE-CS Joint Curriculum Task Force. IEEE Computer Society Press, 1991.
- [3] ACM/IEEE. *Computing Curricula 2001*. Computer Science. Final Report December, 15. The Joint Task Force on Computing Curricula, 2001.
- [4] Arnow D. y Weiss G. *Introduction to Programming using Java*. 2ª edición, Addison-Wesley, 2000.
- [5] Fernández Muñoz L., Peña R., Nava F. y Velásquez Iturbide A. *Análisis de las propuestas de la enseñanza en la programación orientada a objetos en los primeros cursos*. VIII JENUI, Cáceres, 2002.
- [6] García Molina, J.J. *¿Es conveniente la Orientación a Objetos en un primer curso de programación?* Novatica, noviembre-diciembre 2001.
- [7] García Molina, J.J.; Menáñez Tortosa, Marcos; Moros Valle, Begoña. *Una propuesta para organizar la Programación Orientada a Objetos*. VIII JENUI, Cáceres, 2002.
- [8] Gómez Albarrán, Mercedes. *Metodología basada en descomposición funcional y orientación a objetos en la introducción a la programación*. VIII JENUI, Cáceres, 2002.
- [9] *Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems (MSIS'2000)*. Association for Computing Machinery (ACM) y Association for Information Systems (AIS). 2000.
- [10] Reales Decretos 1459/1990, 1460/1990 y 1461/1990 (B.O.E. de 20 de noviembre de 1990). Establecen la creación de los títulos universitarios oficiales de II, ITIS e ITIG.
- [11] Unesco-IFIP. *A modular Curriculum in Computer Science*. The United Nations Educational Scientific and Cultural Organization, 1994.
- [12] UNESCO-IFIP (2000). *ICF-2000. Informatics Curriculum Framework 2000 for Higher Education*. International Federation for Information Processing. UNESCO, Paris.