

# EFFECTOS DE LA REFORMA DEL PLAN DE ESTUDIOS EN LAS ASIGNATURAS DE ALGORITMOS Y ESTRUCTURAS DE DATOS.

C. Herrero<sup>1</sup>, P. Rosso<sup>1</sup>

<sup>1</sup> *Dpto. de Sistemas Informáticos y Computación (DSIC)  
U. Politécnica de Valencia,  
e-mail: [cherrero,proso@dsic.upv.es](mailto:cherrero,proso@dsic.upv.es)*

**Resumen:** En la Universidad Politécnica de Valencia se han detectado ciertas carencias y defectos en asignaturas fundamentales en estudios de informática como pueda ser las de Algoritmos y estructuras de datos. Dichas carencias son un efecto secundario de la aplicación del plan de estudios que ahora se extingue. El presente trabajo es un esbozo de las modificaciones que en el nuevo plan se piensan llevar a cabo para corregir dichos defectos y carencias.

## 1.- INTRODUCCIÓN

Los estudios de informática tanto en la Escuela como en la Facultad de Informática de la Universidad Politécnica de Valencia (UPV), tienen entre sus ejes fundamentales el estudio de algoritmos y estructuras de datos. Este eje se vertebra con tres asignaturas que llevan el mismo nombre (Algoritmos y estructuras de Datos I, II y III) y que se consideran como partes independientes de un todo. En concreto la asignatura de Algoritmos y estructuras de Datos II (AD2) es una de las asignaturas troncales en las tres titulaciones de informática de la UPV y sobre ella vamos a centrar el presente estudio. AD2 tiene una carga lectiva de seis créditos, que se dividen en cuatro de teoría y problemas y dos en prácticas en el laboratorio. Se trata de una asignatura cuatrimestral puente entre AD1 y AD3 también cuatrimestrales. De cara a los nuevos planes de estudios, se pretende modificar esta situación mediante la descomposición de estas asignaturas cuatrimestrales para reconvertirlas en dos anuales sin modificar

substancialmente el número total de créditos aunque sí la distribución temporal y la relación teoría / prácticas.

En las secciones siguientes presentamos un esbozo de la posible reorganización de la materia de la antigua asignatura AD2, así como posibles modificaciones de la misma (lenguaje, notación, metodología etc.) con vistas a su integración en el nuevo plan de estudios. La mejora de la asignatura debe estar orientada a mejorar la calidad de la enseñanza y reducir así las tasas de fracaso. Asimismo estudiamos las posibles causas de ese fracaso así como un estudio comparativo de los lenguajes utilizados en las asignaturas de programación en otras universidades.

## **2.- LAS ACTUALES ASIGNATURAS DE PROGRAMACIÓN**

Las asignaturas AD1 y AD2 son troncales en las tres titulaciones de Informática en esta Universidad: Ingeniería Técnica de Informática de Gestión y de Sistemas (ITIG, ITIS) e Ingeniería Informática (II). Estas asignaturas conforman el núcleo básico en materia de diseño y análisis de algoritmos y de estructuras de datos, y tienen su continuación en la asignatura obligatoria de universidad en las tres titulaciones AD3 y en la titulación superior en la asignatura Algorítmica.

Las asignaturas de programación AD1 y AD2 se imparten en el primer curso en las tres titulaciones y en segundo la de AD3; AD1 se cursa en el primer cuatrimestre, AD2 en el segundo [1] y AD3 en el primer cuatrimestre de segundo curso.

Los contenidos de las asignaturas de programación AD1 que se imparten en el primer cuatrimestre del primer curso son, básicamente:

**AD1:** Estudio de los datos compuestos de tipos tupla y vector; técnicas para la resolución de problemas: serialización, análisis por casos, diseño descendente, uso de esquemas; diseño iterativo guiado por la noción de invariante; algoritmos de recorrido y búsqueda; algoritmos básicos de ordenación.

La asignatura AD2 se imparte en el segundo cuatrimestre y básicamente comprende el estudio de la complejidad de los algoritmos, el estudio formal de la recursión y una introducción a los tipos lineales:

**AD2:** Análisis de la eficiencia de los algoritmos; inducción en el diseño de los algoritmos y diseño recursivo; diseño orientado a los datos; tipos lineales básicos: pila, cola y lista.

El objetivo común de estas asignaturas es la capacitación del alumno para la especificación, descripción, validación, análisis e implementación de soluciones algorítmicas eficientes a problemas específicos. En concreto, la asignatura AD2 supone una evolución en la formación hacia la madurez del alumno en los temas específicos del diseño y análisis de algoritmos y estructuras de datos [1].

### **3.- CARENCIAS Y DEFECTOS: ESBOZO DE UNA NUEVA PROPUESTA**

El plan de estudio vigente obliga a una excesiva fragmentación de la materia en distintas asignaturas, lo que dificulta enormemente por un lado la organización de la docencia y la enseñanza de las asignaturas de programación, y por el otro el proceso de aprendizaje por parte de los alumnos. El proceso de aprender va más allá de lo de entender: se necesita un tiempo de sedimentación para permitir al alumno asimilar los conceptos explicados.

Las asignaturas cuatrimestrales con muchas horas no facilitan la asimilación de la materia ya que no hay suficiente tiempo para la redundancia. Los ejemplos, problemas y ejercicios sobre los mismos conceptos vistos en teoría son una repetición de ideas fundamental para el proceso de aprendizaje del alumno. Asignaturas con una excesiva fragmentación no favorecen la redundancia y por tanto el aprendizaje. Además a efectos prácticos, el ruido introducido por asignaturas cuatrimestrales claramente secuenciales como las que se trata, causa tanto o más daño al aprovechamiento del alumno como las derivadas de la falta de tiempo.

El ruido introducido puede resumirse en los lapsos de tiempo para el asentamiento del alumno en los nuevos grupos, la reorganización de horarios y la redistribución de las prácticas. Ninguno de estos efectos indeseables se producen en la asignatura anual, dado que desde el principio del año académico está clara la distribución temporal (e incluso física) de las distintas clases de teoría, problemas y prácticas.

Para el nuevo plan de estudio se han propuesto dos asignaturas anuales troncales de doce créditos cada una, divididos en nueve de teoría y tres de prácticas, que sustituirían las asignaturas del primer curso AD1, AD2 y la del segundo AD3 [2].

La asignatura de Programación se impartiría el primer curso. Básicamente comprende: el diseño de algoritmos; el estudio de un lenguaje de programación; las técnicas básicas de diseño de algoritmos; el diseño de programas: descomposición modular y documentación; las técnicas de verificación y prueba de programas [3], [4], [5], [6]. (Figura 1)

<p><i>Tema 1.- Problemas, algoritmos y programas.</i> Introducción, ejemplos preliminares. Notaciones para expresar algoritmos. El lenguaje de programación C. Ejemplos.</p> <p><i>Tema 2.- Definición de problemas y algoritmos</i> Datos y resultados. Variables y tipos de datos. Estados, especificación. Ejemplos.</p> <p><i>Tema 3.- Instrucciones</i> Expresiones, evaluación. Instrucciones de entrada/salida. La asignación. Secuencias de instrucciones. Instrucciones condicionales. Ejemplos.</p> <p><i>Tema 4.- Instrucciones iterativas</i> Introducción. La instrucción 'mientras' (con cond. Continuación). Propiedad invariante. Terminación. Otras instrucciones iterativas. Algoritmos numéricos elementales. Ejemplos.</p> <p><i>Tema 5.- Subprogramas</i> Introducción. Parámetros, subprogramas. Definición y uso de subprogramas. Diseño descendente. Ejemplos.</p> <p><i>Tema 6.- Estructuras y vectores</i> Estructuras. Vectores unidimensionales y multidimensionales. Tratamiento secuencial. Problemas de recorrido de vectores. Problemas de búsqueda en vectores. Algoritmos numéricos elementales con vectores.</p> <p><i>Tema 7.- Ficheros</i> Introducción. Ficheros secuenciales, ficheros de texto. Problemas de recorrido y búsqueda en ficheros. Ficheros directos. Ejemplos.</p> <p><i>Tema 8.- Diseño recursivo</i> Introducción. Inducción y recursión. Diseño recursivo. Etapas. Ejemplos.</p> <p><i>Tema 9.- Asignación dinámica de memoria</i> El tipo puntero. Referenciación y dereferenciación. Estructuras lineales enlazadas mediante punteros. Problemas de tratamiento secuencial. Ejemplos.</p>
---

Figura 1.

<p><i>Tema 1: La eficiencia de los algoritmos</i></p> <p>Noción de complejidad, talla e instancia. Unidades de medida. Complejidad asintótica. Problemas: ordenación, búsqueda, otros.</p> <p><i>Tema 2: Estrategias algorítmicas: Divide y Vencerás y Voraz</i></p> <p>La aproximación Divide y Vencerás. Principio de Reducción. Análisis de la corrección y la eficiencia de los algoritmos DyV. Problemas: Ordenación Rápida, Selección, Multiplicación de Matrices, otros. La estrategia voraz. Ámbito de aplicación y esquema general. Problemas: Mochila con Fraccionamiento, Planificación de trabajos, otros.</p> <p><i>Tema 3: Tipos abstractos de datos. Tipos lineales</i></p> <p>Tipos Abstractos de Datos (TAD's). Definición y especificación de tipos lineales: pilas, listas, colas. Implementaciones de los tipos lineales: pilas, listas, colas.</p> <p><i>Tema 4: Grafos y árboles</i></p> <p>Conceptos generales: grafos, árboles, árboles binarios. Implementaciones. El tipo árbol binario. Especificación e implementaciones. Recorridos.</p> <p><i>Tema 5: Conjuntos</i></p> <p>Conceptos Generales. Tablas de Dispersión. Árboles Parcialmente Ordenados (Heaps). Árboles Binarios de Búsqueda. B-árboles. Conjuntos Disjuntos (Mfsets)</p> <p><i>Tema 6: Algoritmos sobre grafos.</i></p> <p>Algoritmos de Recorrido. Test de aciclicidad y ordenación topológica de un grafo. Obtención del árbol de extensión de coste mínimo. El problema de los caminos más cortos en un grafo.</p> <p><i>Tema 7: Algoritmos de retroceso</i></p> <p>Esquema general. Exploración total del árbol. Poda del árbol de retroceso. Problemas: ocho reinas, el caballo de ajedrez, etc.</p>
--

Figura 2.

#### **4.- PRÁCTICAS: UNA DISCUSIÓN PERMANENTE ¿PASCAL O C?**

Las prácticas de la asignatura son fundamentales para la consecución de los objetivos de la asignatura. Los contenidos específicos de AD2 y de las otras asignaturas implicadas en la enseñanza de la programación se basan en la formalización de los problemas, el diseño sistemático de algoritmos y en la relevancia de la abstracción.

En el informe elaborado por el Proyecto de Innovación Docente de la UPV coordinado por los autores se ha tratado una posible reorganización de las asignaturas sobre todo en función del lenguaje de programación a utilizar,

atendiendo al conjunto de las opciones de la carrera, y a lo aportado en otras universidades. Con este proyecto pretendemos ante todo elaborar diferente material didáctico para la mejora del aprendizaje (guías didácticas y boletines de prácticas), en base al nuevo lenguaje.

Acerca de la elección del lenguaje de programación a enseñar en los cursos introductorios de programación, no hay consenso en las universidades.

En el informe "19<sup>th</sup> edition of the list of the languages used in the FIRST course for Computer Science majors at Colleges and Universitys" realizado por el profesor Richard J. Reid de la Michigan University ([ftp.cps.msu.edu:pub/arch/CS1\\_Language\\_List](ftp.cps.msu.edu:pub/arch/CS1_Language_List))[10], sintetiza los datos enviados por 538 universidades americanas y europeas sobre el primer lenguaje de programación que se enseña en los cursos introductorios. En este estudio se muestra que entre el Pascal y el C/C++ se cubren más del 50% de las universidades. La utilización del Pascal si bien ha descendido sigue siendo predominante junto con el C/C++.

## 5.- BIBLIOGRAFÍA

- [1] Casanova A., Castro M.J., Fuentes V., Galiano I., García C., Marqués F., Prieto N., La Enseñanza de la Programación en los Estudios de Informática de la UPV, IV Jornadas Sobre L'Ensenyament Universitari de la Informàtica JENUI98, Universitat Ramon Llull, 1998.
- [2] Castro M.J., Marqués F., Sánchez J. A. et. al., Una Competición de Algoritmos y Estructuras de Datos, IV Jornadas Sobre L'Ensenyament Universitari de la Informàtica JENUI98, Universitat Ramon Llull, 1998.
- [3] Aho, Hopcroft, Ullman, Estructuras de Datos y Algoritmos", Addison Wesley Iberoamericana, 1988
- [4] Balcázar J.L., Programación Metódica, McGraw Hill, 1993.
- [5] Castro J., Cucker F., Messeguer X., Rubio A., Solano L., Valles B., Curs de Programació, McGraw-Hill. 1992.
- [6] Scholl P.C., Algorítmica y Representación de Datos. Tomo 2: Recursividad y Árboles, Ed. Masson, 1986.
- [7] Brassard G., Bratley P., Fundamentos de Algorítmica, Prentice-Hall, 1997.
- [8] Cormen T. H. , Leiserson C. E., Rivest R. L., Introduction to Algorithms, MIT Press, 1990.
- [9] Weiss M.A., Estructuras de Datos y Algoritmos, Addison-Wesley, 1995.
- [10] Sánchez, J.A., Proyecto Docente, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica Valencia, 1999