

## UNA PROPUESTA PARA ASIGNATURAS DE INTRODUCCIÓN A LA PROGRAMACIÓN.

Aguilar, Jesús S.<sup>1</sup>, González, Mariano<sup>1</sup>, Riquelme, José C.<sup>1</sup>.

<sup>1</sup>*Departamento de Lenguajes y S.I.  
Facultad de Informática. U. de Sevilla  
e-mail: {aguilar,mariano,riquelme}@lsi.us.es*

**RESUMEN:** En este artículo se presentan las asignaturas de Introducción a la Programación que se imparten en las titulaciones de Ingeniero en Informática de la Universidad de Sevilla. Los rasgos más destacados son: introducción a la POO y la abstracción de datos, el uso de C y C++ en el laboratorio y una metodología basada en diapositivas, código para experimentos y el acceso a toda la documentación a través de la web.

### 1.- INTRODUCCIÓN.

Las asignaturas que inician al alumno en la programación durante el primer curso de las tres titulaciones en Ingeniería Informática se denominan Introducción a la Programación I e Introducción a la Programación II. Ambas son cuatrimestrales con 7.5 créditos, es decir, 5 horas de docencia a la semana y un temario común para la titulación de Ingeniero Superior e Ingenieros Técnicos. Estas 5 horas se reparten en 3 horas de clase en el aula convencional y que podríamos denominar clases de Teoría y 2 horas en el Laboratorio o clases de Prácticas. Estos laboratorios constan de una red local de ordenadores personales donde los alumnos trabajan en parejas. Desde este laboratorio los alumnos tienen acceso restringido a Internet, concretamente sólo a las páginas Web de la Facultad y de los departamentos con docencia en ésta.

El número de matriculados en ambas asignaturas durante el pasado curso 1999-2000 es superior a 1000 divididos en 8 grupos. Para el presente curso se prevén un total de 11 grupos, lo que se traduce en un importante esfuerzo en la coordinación y puesta en común por el profesorado, no sólo entre los distintos grupos sino en la imprescindible conexión teoría-laboratorio para que el alumno consolide en el laboratorio lo explicado en el aula.

Cada grupo tiene unos 125 alumnos que para las clases de Prácticas se dividen en dos grupos que reciben docencia con profesores distintos en aulas de Laboratorio diferentes. Durante el curso pasado en cada cuatrimestre había cuatro profesores impartiendo clases de Teoría y otros cuatro impartiendo prácticas.

Los contenidos de las asignaturas son los siguientes

1. Conceptos básicos	5. Secuencias
2. Tipos y expresiones	6. Tablas y registros
3. Notación Algorítmica	7. Ficheros
4. Funciones	

Tabla I. Programa de IP1

1. Recursividad	6. TAD Vector Dinámico
2. Tipos Abstractos de Datos	7. TAD Lista
3. Intr. a la Programación OO	8. TAD Pila
4. TAD Fecha	9. TAD Cola
5. TAD Cadena	10. TAD Árbol

Tabla II. Programa de IP2

Como se puede observar, el temario de IP1 es un temario clásico de primer curso de programación donde quizás lo único resaltable sea la utilización del lenguaje C en las clases de Laboratorio, en lugar de los más extendidos Modula o Pascal.

El temario de IP2 es menos "tradicional" para una asignatura de primer curso, en el sentido de que incorpora una introducción a la Programación Orientada a Objetos basada en la implementación de los tipos abstractos de datos clásicos (Lista, Pila, Cola, Árbol) mediante el uso de los conceptos de clases y tipos genéricos (*templates*) en C++.

## 2.- INTRODUCCIÓN A LA PROGRAMACIÓN I.

### a) Clases de Teoría.

Los alumnos denominan clases de teoría a aquellas que se imparten en el aula convencional. Los profesores pretendemos tres líneas de actuación que se repiten en el mismo orden y de forma circular a lo largo del cuatrimestre, adaptándose a cada unidad específicamente:

- Presentación de un lenguaje algorítmico simple (que denominamos LEA), compuesto por un conjunto de tipos de datos básicos e instrucciones y estructuras de control elementales.
- Presentación del lenguaje de especificación formal que ayude a la comprensión de la interfaz de algoritmos mediante el concurso de precondiciones y poscondiciones (Pre/Pos).
- Presentación de los esquemas algorítmicos asociados.

Así para cada tema de los enumerados en la tabla I se presentan los tipos de datos con la signatura y las condiciones de definición para cada operación asociada a ellos. Posteriormente, se describe el lenguaje de especificación (operadores, dominios y cuantificadores) y se presentan las instrucciones, junto con sus propiedades, verificando su corrección. De esta forma el alumno dispone de un lenguaje imperativo completo y potente y, además, puede describir el comportamiento de cada algoritmo con las especificaciones Pre/Pos. A continuación se presentan los esquemas algorítmicos genéricos que se usarán para la implementación de los problemas. Estos esquemas (recorrido, búsqueda, filtro y ordenación) se comportan como

unidades independientes y su composición adecuada sirve para resolver los problemas al final de cada unidad.

Mediante esta metodología se ofrece un lenguaje consistente y no ambiguo, del cual los alumnos conocen el uso de una determinada operación asociada a un tipo de datos y la condiciones en las que puede aplicarse. De esta forma se esbozan subliminalmente los conceptos de abstracción de datos y encapsulamiento, conceptos fundamentales en la próxima IP2.

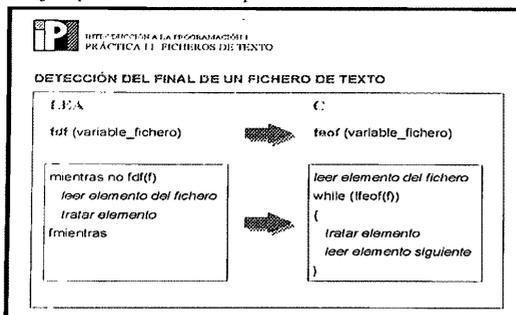
**b) Clases de Laboratorio.**

En las clases de laboratorio se imparten las prácticas, a razón de una por semana. Las clases se dividen en dos partes: durante la primera, que tiene una duración de entre 30 y 45 minutos, el profesor explica la traducción a lenguaje C de los esquemas algorítmicos estudiados en teoría correspondientes a la práctica de la semana. No se trata, pues, de estudiar el lenguaje C en su totalidad, sino solamente aquellas expresiones que son "traducción" directa del lenguaje LEA. Por ejemplo, se dedica poco tiempo a explicar los formatos de escritura de la función *printf*, no se estudian el operador ternario *?* y los operadores *+=* y similares, sólo se permite el uso del operador *++* detrás de una variable, etc.

1. Entorno de trabajo	7. Módulos
2. Introducción al lenguaje C	8. Tablas
3. Estructuras de control selectivas	9. Cadenas de caracteres
4. Estructuras de control repetitivas	10. Estructuras
5. Funciones (I)	11. Ficheros de texto
6. Funciones (II)	12. Ficheros binarios

**Tabla III:** listado de las prácticas de IP1

Para la impartición de esta docencia el profesor se apoya en una presentación realizada en Powerpoint que se proyecta en la pantalla del aula. Los alumnos pueden seguir la exposición en la pantalla o en sus propios ordenadores, descargando la presentación que se encuentra disponible en la página web de la asignatura. Esto les permite además preparar la clase con antelación y repasar los conceptos explicados a posteriori. La figura 1 muestra una diapositiva de ejemplo de una de las presentaciones.



**Figura 1:** diapositiva de una presentación de las clases de laboratorio de IP1

La segunda parte de la clase de laboratorio se dedica a la realización de una serie de experimentos y ejercicios por parte de los alumnos. Los experimentos son ejercicios semirresueltos donde se plantea al alumno un problema concreto sobre alguno de los conceptos estudiados en la práctica. Si es preciso se proporciona al alumno el código del experimento para que se centre en el problema y no pierda el tiempo tecleando. Por su

parte, los ejercicios enuncian un problema que el alumno debe resolver en su totalidad, planteando en primer lugar el esquema algorítmico y traduciéndolo posteriormente a lenguaje C. Los alumnos cuentan en todo momento con la asistencia del profesor en el aula para resolver sus posibles dudas.

6.2 Complete los huecos del siguiente programa para que muestre en pantalla el mensaje

```
El código es 12 y el tipo es B

#include <stdio.h>
typedef int * Pint;
typedef char * Pchar;
void inicia (Pint, Pchar);

void main (void)
{
    int codigo;
    char tipo;
    inicia ( __ codigo, __ tipo);
    printf ("\nEl código es
           %d",codigo);
    printf (" y el tipo es
           %c",tipo);
}
void inicia ( __ c, __ t)
{
    __ c = 12;
```

Figura 2: Ejemplo de experimento

La figura 2 muestra un ejemplo de un tipo de experimento consistente en completar el código de un programa que debe realizar una tarea concreta. Todos los experimentos y ejercicios que se realizan en el aula, junto con otros ejercicios adicionales y un resumen de los conceptos teóricos correspondientes, están disponibles en los enunciados de las prácticas, que se encuentran a disposición de los alumnos tanto en la copistería del centro como en la página web con una semana de antelación. Esto permite que los alumnos puedan preparar los esquemas algorítmicos necesarios previos a la resolución en C de los ejercicios, y de esta manera aprovechar mucho mejor el tiempo en el aula. Por otra parte, una vez impartida la práctica se publican las soluciones de los experimentos y ejercicios para que el alumno pueda compararlas con las propuestas por él, así como obtener las de

aquellos ejercicios que no haya podido terminar en el aula.

### c) Evaluación.

La evaluación de la asignatura se realiza, lógicamente, también por separado, con dos partes claramente diferenciadas. En el examen que podríamos denominar de Teoría se evalúa la utilización de los esquemas algorítmicos en la resolución de determinados problemas sobre ficheros y tablas principalmente. Los conocimientos impartidos en las clases de laboratorio se evalúan en una parte del examen escrito. Esta parte consiste en una batería de preguntas, que pueden ser incluso de tipo test, sobre la sintaxis del lenguaje C. De esta forma, en la evaluación de las prácticas sólo se valoran los conocimientos del lenguaje de programación, separándolos claramente de la habilidad para diseñar algoritmos o para utilizar esquemas algorítmicos.

La figura 3 muestra una muestra de un examen de prácticas de tipo test, extraído de la página web de la asignatura.

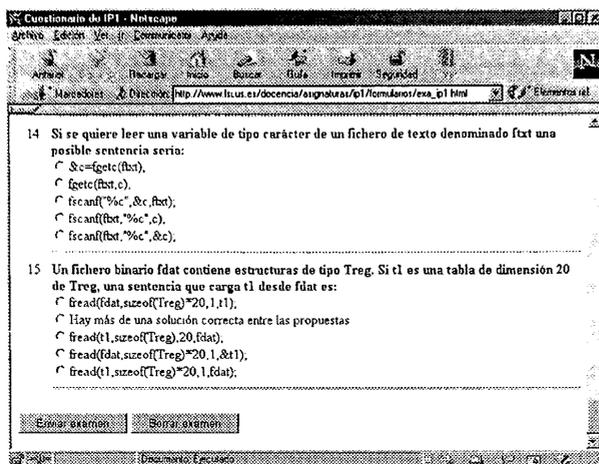


Figura 3: ejemplo de cuestionario sobre lenguaje C

### 3.- INTRODUCCIÓN A LA PROGRAMACIÓN II.

#### a) Clases de Teoría.

Las clases de Teoría de este cuatrimestre se dividen temporalmente en tres partes: el primer tema es un estudio clásico de los esquemas recursivos y la resolución de problemas clásicos: factorial, fibonacci, búsqueda binaria, torres de Hanoi, *quicksort*, etc. La segunda parte de la asignatura consiste en una introducción al concepto de tipo abstracto de datos y la programación orientada a objetos. De la POO se resaltan los aspectos de abstracción y encapsulamiento posponiendo la herencia y el polimorfismo para cursos posteriores. Con respecto a la abstracción se modelan los TAD Complejo y Fecha como ejemplos, para posteriormente incluir el concepto de genericidad de los TAD para que sirvan de plantilla. Así podemos utilizar Arbol de Lista de Cadena. El encapsulamiento permite que el alumno sólo utilice la interfaz de los métodos para resolver problemas. Así en las clases de teoría se desconoce el código de las operaciones del TAD, que se estudia en las clases de laboratorio. Por último, la mayor parte de la asignatura (75% del total) consiste en la introducción de los distintos TAD estudiados en la asignatura. Cada uno de ellos se introduce mediante una especificación de sus atributos y métodos en notación pre/pos, y en la realización de ejercicios basados en la interfaz proporcionada. Estos ejercicios se desarrollan en un lenguaje algorítmico similar a LEA pero donde aparecen objetos invocando a métodos de la Clase.

#### b) Clases de Laboratorio.

Básicamente, el desarrollo de las clases de prácticas es similar al de IP1. Es decir, el profesor explica la implementación que se ha realizado del TAD correspondiente y que se proporciona al alumno en la red local de forma completa o, normalmente, sin algunos de los métodos de la Clase. Posteriormente el alumno dispone de más de una hora para la implementación de una serie de experimentos, ejercicios semiresueltos y propuestos para su realización en los boletines correspondientes.

1. Entorno de Programación	7. TAD Lista
2. Recursividad	8. TAD Pila
3. TAD Fecha	9. TAD Cola
4. TAD Cadena	10. TAD Árbol
5. TAD Vector Dinámico	11. Repaso
6. TAD Polinomio	

**Tabla IV:** listado de las prácticas de IP2

La implementación de los distintos TAD se realiza mediante la definición de Clases en C++ con plantillas o *templates*, y de forma, que un TAD es utilizado en sucesivas prácticas. Así, por ejemplo, el TAD Fecha y el TAD Cadena se usan como atributos de un TAD Persona que se utiliza como base para el TAD Cola. Asimismo, las Clases Lista y Vector Dinámico se usan como bases para la implementación de las clases Pila y Cola.

En las sesiones de Laboratorio se hace sobre todo especial hincapié en los problemas de implementación, pues los esquemas algorítmicos son estudiados en las clases de Teoría. De esta manera, los experimentos se diseñan para que los alumnos comprueben la diferencia entre codificar un constructor copia o utilizar el que proporciona el compilador, pasar un parámetro por referencia o por valor, uso o no de la cláusula *const* en parámetros o métodos, etc. Los ejercicios semiresueltos suelen consistir en la implementación de algunos métodos del TAD correspondiente que han sido especificados en las clases de Teoría. Estos ejercicios se proporcionan en ficheros fuentes donde los alumnos sólo tienen que escribir algunas líneas de código para completarlo.

Por último, se proponen otros problemas que el alumno debe resolver previamente en LEA (la mayoría ya se han estudiado en las clases de Teoría) y posteriormente implementar en C++. Los comentarios a los experimentos propuestos y las soluciones de todos los ejercicios son publicados la semana siguiente. Si los alumnos lo desean pueden utilizar las sesiones de Laboratorio para discutir o corregir las propuestas de resolución de prácticas anteriores.

#### 4.- UTILIZACIÓN DE NUEVAS TECNOLOGÍAS.

Las nuevas tecnologías ofrecen una serie de posibilidades que pueden mejorar muchos aspectos de la enseñanza y hacer más dinámica la comunicación entre el alumno y el profesor. Actualmente se están aprovechando algunas de estas posibilidades, como son:

- Uso de presentaciones multimedia en las clases de laboratorio para mostrar los conceptos de una forma didáctica y a la vez amena para los alumnos
- Potenciación de la página web para proporcionar información de interés a los alumnos:
  - Documentación: enunciados, presentaciones y soluciones de las prácticas, apuntes y problemas resueltos de teoría, enunciados y soluciones de exámenes de convocatorias anteriores
  - Calificaciones de los exámenes
  - Recursos de interés en la red
  - Cuestionarios de autoevaluación sobre las prácticas, mediante los cuales el alumno puede comprobar su nivel de conocimientos
- Acceso a tutorías por correo electrónico

Otras posibilidades que se piensan implantar en un futuro próximo son las siguientes:

- Mejora de las presentaciones para incluir más animaciones

- Realización de cuestionarios más flexibles mediante el uso de bases de datos y tecnología ASP
- Creación de una página de FAQs para consulta de los alumnos
- Creación de una lista de correo para distribuir rápidamente información de interés a los alumnos
- Publicación en la web de apuntes de teoría con simulaciones animadas de conceptos de interés o ejercicios resueltos

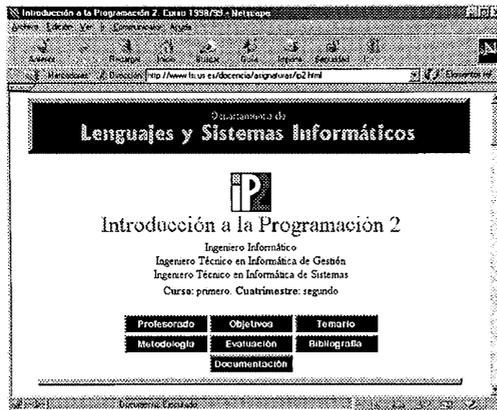


Figura 4: página web de la asignatura IP2

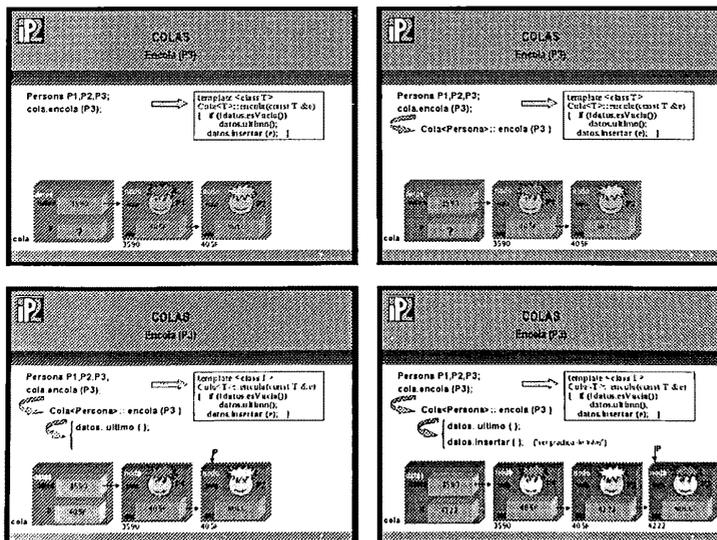


Figura 5: dispositivos de una presentación de las clases de laboratorio de IP1