

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 05-03

Programación 2: experiencias y propuestas

Javier Alliaume, Carlos Luna, Martín Pedemonte, Marcos Viera

Abril de 2005

Programación 2: experiencias y propuestas
Alliaume, Javier; Luna, Carlos; Pedemonte, Martín; Viera, Marcos

ISSN 0797-6410

Reporte Técnico RT 05-03

PEDECIBA

Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay, abril de 2005

PROGRAMACION 2: EXPERIENCIAS Y PROPUESTAS §

Alliaume, Javier¹; Luna, Carlos¹; Pedemonte, Martín¹; Viera, Marcos¹;

¹Instituto de Computación (InCo), Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

jalliaum@fing.edu.uy - cluna@fing.edu.uy - mpedemon@fing.edu.uy - viera@fing.edu.uy

Resumen: *En este trabajo presentamos experiencias y propuestas en torno al segundo curso de programación de la carrera de Ingeniero en Computación del Instituto de Computación (InCo).*

A partir del año 2003 el InCo decidió asignar un mayor número de recursos docentes al curso Programación 2 (P2) con el objetivo de "mejorar la calidad de enseñanza de la programación y el nivel general del curso". En este trabajo describimos las tareas realizadas a partir de 2003 por el equipo docente, en el contexto de masividad en el cual se desarrolla el curso. Puntos claves en este período han sido: la asignación de roles dentro del equipo docente; la potenciación de los recursos docentes en las tareas más relevantes; la profundización de tareas de programación tanto en trabajos prácticos como principalmente en los trabajos de laboratorio obligatorios; el desarrollo de materiales didácticos de apoyo; la mayor disponibilidad de materiales e información general del curso desde su página web; el apoyo constante a estudiantes en clases/consultas presenciales y a través de un e-group.

En función de esta experiencia se plantean posibles mejoras puntuales, sin embargo existen cambios estructurales más importantes que trascienden a la asignatura y vinculan a los cursos previo y posterior de programación (Programación 1 (P1) y Programación 3 (P3), respectivamente) y también al curso Lógica. En este sentido consideramos importante rescatar la figura de los encargados de materias. En particular para la materia programación, con la finalidad de realizar un análisis profundo de los objetivos y los contenidos de los distintos cursos de programación (en particular los primeros tres) y la vinculación de éstos con asignaturas de otras áreas.

1. Introducción

P2 es un curso de la carrera de Ingeniero en Computación del InCo que se sitúa en el semestre 3 y pertenece a la materia Programación. El objetivo general de la asignatura es capacitar al estudiante en el diseño de programas de tamaño mediano, utilizando técnicas consistentes en la formulación e implementación de abstracciones de datos. La asignatura se evalúa por medio de dos parciales y varios trabajos de laboratorio (en general: 2). Los trabajos de laboratorios son eliminatorios, es decir que el estudiante que no obtenga el nivel mínimo de suficiencia en su trabajo no podrá aprobar el curso independientemente del puntaje que obtenga en los parciales. El programa detallado del curso puede consultarse en el anexo.

P2 es un curso intermedio entre P1 (del semestre 2) y P3 (del semestre 4), que se desarrolla en condiciones de masividad. A partir del dictado correspondiente al año 2001, la inscripción ronda los 500 estudiantes. Desde el año 2003 el InCo decidió asignar un mayor número de recursos docentes a P2 con el objetivo de "*mejorar el curso*".

§ Presentado en el II Congreso de Enseñanza en Facultad de Ingeniería, Udelar, Montevideo, Uruguay, Octubre de 2004.

En este trabajo describimos las tareas realizadas a partir de 2003 por el equipo docente y presentamos propuestas a futuro para continuar el proceso iniciado en 2003. La organización del resto de este artículo es la siguiente. En la sección 2 mencionamos algunos de los cambios realizados en los últimos años, mostrando en la sección 3 los primeros resultados observados. Sin embargo existen cambios estructurales más importantes que podrían implementarse que trascienden a la asignatura y vinculan a los cursos previo y posterior de programación (P1 y P3, respectivamente) y también al curso Lógica. En la sección 4 presentamos propuestas en este sentido y finalmente las conclusiones de este trabajo se incluyen en la sección 5.

2. Implementando Cambios

A partir del año 2003 el InCo decidió duplicar el número de recursos docentes asignados al curso P2 con el objetivo de *"mejorar la calidad de enseñanza de la programación y el nivel general del curso"*. En esta sección describimos las actividades llevadas a cabo en este período.

Los objetivos generales han sido:

- Definir actividades y roles claves dentro del equipo docente.
- Potenciar los recursos docentes en las tareas más relevantes.
- Mejorar la calidad de enseñanza de la programación, profundizando las tareas de programación tanto en trabajos prácticos como principalmente en los trabajos de laboratorio obligatorios.

A partir de la forma de evaluación de la asignatura se consideró conveniente la conformación de dos equipos de trabajo complementarios. Uno encargado de las tareas relacionadas con los proyectos de laboratorio obligatorios y otro equipo especializado en el desarrollo de los trabajos prácticos. En consecuencia se definieron los siguientes roles dentro del equipo docente: encargado de curso; encargado de teóricos; encargado de prácticos; docente de práctico; encargado de laboratorio; docente de laboratorio.

Se optó por una forma de trabajo descentralizada en la que cada una de las áreas se organizó a partir de los equipos de trabajo. Cada equipo en forma independiente realizaba reuniones semanales, en las cuales participaban los encargados y los docentes del área para discutir sobre los aspectos concernientes a ésta. Adicionalmente, existían reuniones semanales de coordinación entre ambos equipos que permitían centralizar la información y la toma de decisiones críticas para el curso.

Algunas de las tareas específicas llevadas a cabo se concentraron en:

- Mejorar los materiales del curso:
 - Guías de teórico.
 - Letras de prácticos con ejercicios básicos y avanzados, para todos los temas del curso.
 - Materiales relacionados con los proyectos de laboratorio obligatorios: letras, especificaciones de los sistemas, ejemplos de uso y casos de prueba, errores frecuentes, entre otros.
 - Materiales complementarios: sobre el lenguaje de programación, el compilador y el entorno de programación usados.
- Mejorar el acceso a los materiales y el contacto con los docentes
 - Acceso a los materiales del curso a través de internet (página web del curso), incluyendo parciales y exámenes con soluciones, cartelera con novedades acerca del curso e información general del curso.
 - Clases y consultas presenciales semanales de práctico y laboratorio.

- Consultas de teórico, práctico y laboratorio (fundamentalmente) a través de un e-group, monitoreado diariamente por los encargados de cada actividad.
- Transformar a los trabajos de laboratorio obligatorios en el eje central del curso.
 - Incremento en la complejidad y porte de los trabajos de laboratorio, buscando cubrir la mayor cantidad posible de temas del curso e integrarlos en el desarrollo de un sistema de tamaño mediano, de acuerdo con el objetivo general del curso.
 - Influencia en la calificación final del estudiante, implementada mediante un ejercicio del segundo parcial cuya problemática es muy similar a alguna tratada en el laboratorio.
- Fomentar "buenas prácticas de programación" a través del desarrollo de soluciones de ejercicios seleccionados del práctico, con un enfoque didáctico.

3. Primeros Resultados

A continuación presentamos los resultados del curso correspondientes a las últimas seis ediciones.

Año	Inscriptos	Pierden (%)	Aprueban (%)	Exoneran (%)
1999	395	50,13	17,72	32,15
2000	405	50,62	22,22	27,16
2001	494	51,42	16,80	31,78
2002	481	49,29	35,75	14,96
2003	540	66,66	12,59	20,75
2004	502	63,74	13,55	22,71

Tabla 1: Estudiantes que Pierden, Aprueban o Exoneran el curso considerando las inscripciones de 1999 a 2004

Como se puede ver en la tabla 1 se ha producido un incremento en la cantidad de estudiantes que pierden el curso en los dos últimos años. Creemos que el nivel de exigencia del curso ha aumentado, debido fundamentalmente a los cambios que se operaron en los trabajos de laboratorio obligatorios (incremento del tamaño y complejidad). Adicionalmente, el curso parece haberse estabilizado en cuanto a sus resultados.

En la siguiente tabla, se presentan los resultados para cada una de las instancias de evaluación del laboratorio. De los datos puede inferirse un importante abandono de la asignatura entre la primera y segunda instancia de entrega del laboratorio, porcentaje que no difiere mayoritariamente al de los años anteriores. Sin embargo, notamos una disminución del porcentaje de aprobación de la segunda instancia de evaluación del laboratorio con respecto a los años anteriores. Conviene destacar que en los dos últimos años perfeccionamos los procedimientos de detección de trabajos no individuales. En 2003 60 estudiantes y en 2004 10 estudiantes perdieron el curso por incumplir la cláusula de individualidad a la que están sujetos los trabajos de laboratorio.

Año	Primera Entrega				Segunda Entrega			
	Entregan	Aprueban	Pierden	No-Indiv.	Entregan	Aprueban	Pierden	No-Indiv.
2003	443	86,00%	6,77%	7,23%	280	65,71%	23,57%	10,72%
2004	399	92,24%	7,76%	0,00%	263	70,73%	25,47%	3,80%

Tabla 2: Resultados para cada una de las instancias del laboratorio de 2003 a 2004

Si bien hemos constatado un aumento en la cantidad de estudiantes que pierden el curso en los dos últimos años, respecto a los anteriores, de los estudiantes que aprobaron los laboratorios, un número inferior al 3% no lograron aprobar el curso o exonerar como lo muestra la tabla 3. Adicionalmente, el porcentaje de estudiantes que exonera la asignatura es considerablemente mayor al porcentaje de estudiantes que aprueban el curso. Estos

resultados reafirman que los trabajos de laboratorio se han transformado en el eje central del curso en concordancia con los objetivos planteados.

Año	Aprueban Lab.	Exoneran (%)	Aprueban (%)	Pierden (%)
2003	184	60,87	36,96	2,16
2004	186	61,29	36,56	2,15

Tabla 3: Estudiantes que Pierden, Aprueban o Exoneran el curso considerando los estudiantes que aprueban el laboratorio de 2003 a 2004

Un resultado importante respecto a la interacción con P3 es el hecho de que, al enfrentarse los estudiantes durante P2 con un trabajo de laboratorio de un porte considerable, se disminuyó el impacto que generaba este tipo de trabajos en P3. En P3 se cuenta además con el agravante del cambio de lenguaje, de uno fuertemente tipado (como Modula-2 o Pascal) a otro que no lo es (C/C++). Este cambio ha repercutido positivamente en P3 permitiendo que a partir del corriente año se aproveche la experiencia en programación de los estudiantes para enfatizar en la implementación de estructuras de datos y algoritmos avanzados.

4. La Materia Programación: Propuestas

Actualmente el curso P2 hace hincapié en tres bloques temáticos claves:

- (1) *Inducción y recursión*
- (2) *Punteros y estructuras de memoria dinámica*
- (3) *Tipos abstractos de datos*

Los temas *inducción y recursión* son esenciales para el curso P2, ya que los restantes temas se sustentan en éstos. Los temas *inducción y recursión* son abarcados también por el curso Lógica que está en el mismo semestre de la carrera (se desarrolla en paralelo). No parece natural mantener esta duplicación de esfuerzo, no solamente por un tema de recursos sino por un tema de coordinación. Al no existir coordinación entre lo dictado en Lógica y en P2 pueden haber diferencias sutiles de matices que impliquen una dificultad para los estudiantes que están aprendiendo la temática. Consideramos que en el corto plazo es fundamental que exista una coordinación entre el dictado de estos temas para ambas asignaturas. Aunque esto no representa una solución al problema y debería encararse una solución más de fondo al problema como por ejemplo incorporar estos conocimientos dentro de la asignatura Matemática Discreta o eventualmente del curso Lógica, en cuyo caso debería ser previo a P2.

Otro tema que resulta esencial es el estudio de *punteros*, el cual debe ser desarrollado en su totalidad en el curso debido a que no es abordado en P1. Esto, sumado a lo expresado anteriormente, le da un quiebre temático al curso en dos partes, una en la cual se brindan conocimientos iniciales (*inducción, recursión y punteros*) y otra en donde se estudian estructuras de datos y tipos abstractos de datos. Proponemos que el tema *punteros* se incluya, por lo menos en forma introductoria, en P1. Este hecho, si bien implica una vuelta al anterior plan, permitiría que P2 se concentrara, en relación a este tema, en el diseño, la implementación y el análisis más profundos de estructuras de memoria dinámica (lineales y arborescentes). En el seminario sobre temas de programación que se realiza en el *InCo* se analizó esta posibilidad favorablemente.

En base a las dos propuestas anteriores el curso P2 podría dedicar más tiempo a la resolución de problemas con el uso de tipos abstractos de datos e incluir conceptos de análisis de algoritmos que no son considerados, pero que resultan sumamente necesarios. La idea es transformar al curso P2 en un curso de *estructuras de datos y algoritmos*, que incluya una introducción al análisis de algoritmos. P1 sería un *curso básico de programación* que consideraría memoria dinámica además de estructuras de memoria estática. P3 se

transformaría en un curso de *estructuras de datos y algoritmos avanzados* y abarcaría el estudio de técnicas de análisis y diseño de algoritmos, como temas más relevantes.

Como se deduce de los puntos planteados anteriormente existe una fuerte interrelación de los contenidos de P2 con los de otras asignaturas de Programación (P1 y P3) e inclusive con los de otras asignaturas del propio instituto (Lógica) y de otros institutos (Matemática Discreta). Es necesario asegurar la coherencia de contenidos y la coordinación entre temáticas afines. En este sentido, consideramos importante rescatar la figura de los encargados de materias. En particular para la materia programación, con la finalidad de realizar un análisis profundo de los objetivos y los contenidos de los distintos cursos de programación (sobre todo de los primeros tres) y la vinculación de éstos con asignaturas de otras áreas, como la Matemática.

5. Conclusiones

Los cambios realizados a partir de 2003 no han logrado mejorar las estadísticas de aprobación/exoneración de P2. Estamos convencidos de que las mejoras obtenidas son más bien cualitativas. Esta observación se basa fundamentalmente en el análisis con el responsable y docentes del curso P3. La confirmación de esta hipótesis la obtendremos a medida que se generen más datos con respecto al desempeño de los estudiantes en las asignaturas de programación posteriores de la carrera.

Actualmente y tras dos años de trabajo se ha logrado una puesta a punto del curso P2, generándose una cantidad significativa de material. Los cambios que se han abordado dependían directamente de la asignatura y del equipo docente encargado de su dictado. Sin embargo existen cambios estructurales más importantes que podrían implementarse que trascienden a la asignatura. Como resultado de la experiencia realizada se formularon propuestas que atienden a la estructura e interrelación de las asignaturas de programación y a la vinculación de éstas con asignaturas de otras áreas.

Este trabajo ha pretendido ser el puntapié inicial para una discusión que resulte en una revisión de los contenidos de las asignaturas básicas de programación viéndolas como un todo. Actualmente en el ámbito del seminario sobre temas de programación que se realiza en el *InCo* se ha comenzado esta discusión.

Anexo: Programa del curso Programación 2

Nombre de la Asignatura: Programación 2.

Materia: Programación.

Créditos: 12.

Objetivos de la Asignatura

Generales

Capacitar al estudiante en el diseño de programas de tamaño mediano, utilizando técnicas consistentes en la formulación e implementación de abstracciones de datos.

Particulares

- Que el estudiante conozca los conceptos de tipo abstracto de datos y módulo de programa, y los ponga en uso en el diseño de programas.
- Que el estudiante conozca un lenguaje de programación en el que los conceptos de módulo y tipo abstracto de datos sean expresables de manera natural.
- Que el estudiante conozca y ejercite las técnicas de composición de datos y programas por recurrencia.
- Que el estudiante conozca y tenga experiencia en la utilización de punteros y estructuras de datos dinámicas.
- Que el estudiante conozca, implemente y utilice abstracciones de datos básicas relacionadas con distintos tipos de estructuras lineales y arborescentes.

Metodología de enseñanza

La enseñanza estará centrada en clases prácticas y el desarrollo de trabajos de laboratorio. Estas actividades serán acompañados de exposiciones teóricas de presentación de los temas del curso.

Cada alumno recibirá o deberá dedicar:

2 hs. semanales de exposiciones teóricas,

2 hs. semanales de clases prácticas y

8 hs. semanales de dedicación domiciliaria y trabajo en laboratorio

durante las 15 semanas de curso.

Temario

1. Tipos abstractos de datos.

Motivación, concepto y terminología. Tipos abstractos de datos como concepto de diseño de programas. Especificación e implementación de tipos abstractos de datos. Módulos de programas. Los conceptos de módulo y tipo abstracto de datos en lenguajes de programación. El lenguaje MODULA-2.

2. Recurrencia.

Tipos de datos recurrentes: naturales, listas, arboles. Funciones y procedimientos recurrentes. Técnicas de composición.

3. Implementación de recurrencias.

Implementación de estructuras de datos recurrentes: punteros. Concepto y manipulación de Punteros. Implementaciones de listas y árboles. Implementación de procedimientos recurrentes.

4. Diseño de programas mediante abstracción de datos.

Refinamiento de procedimientos y datos. Casos de estudio.

5. Abstracciones de datos básicas.

Estructuras lineales: listas, pilas, colas, dobles colas.

Estructuras arborescentes: árboles generales o finitarios, árboles n-arios, árboles binarios, árboles binarios de búsqueda.

Otros tipos de abstracciones.

Bibliografía

Básica:

- Programming in Modula-2. N. Wirth. Springer-Verlag, 1985. ISBN 3540150781.
- Abstract Data Types in Modula-2. R. Harrison. Wiley, 1989. ISBN 0471922307

Complementaria:

- A Second Course in Computer Science with Modula-2. D. Mc Cracken, W. Salmon. Wiley, 1987. ISBN 0471631116.
- Estructuras de Datos y Algoritmos. Mark Allen Weiss. Addison-Wesley Iberoamericana, 1995. ISBN 0-201-62571-7
- Estructuras de Datos y Algoritmos. Aho, Hopcroft y Ullman. Addison Wesley Iberoamericana, 1988. ISBN 0-201-64024-4
- Programación con Pascal. Konvalina J., Wileman S. - Prentice Hall, 1986

Previaturas

Programación 1 (curso).

Conocimientos previos exigidos y recomendados

Son requeridos los conocimientos adquiridos al cursar la asignatura Programación 1 de la carrera de Ingeniería en Computación.

Anexo: Evaluación

La asignatura se evaluará por medio de dos parciales y trabajos de laboratorio. El nivel mínimo de suficiencia en los trabajos de laboratorio es eliminatorio, ya que esta parte del trabajo del curso no puede ser evaluada mediante exámenes. Por otra parte, dependiendo de las condiciones de dictado del curso, el trabajo de laboratorio se evalúa según las opciones aprobado/no aprobado, o con puntaje diferenciado en el caso de aprobación. En este último caso, el puntaje del laboratorio se integraría al puntaje total del curso, prorrateándose en los de las pruebas parciales.

En todos los casos de los resultados obtenidos surgen tres posibilidades:

- Exoneración del examen final.
- Suficiencia en el curso; el estudiante queda habilitado a rendir el examen.
- Insuficiencia en el curso; el estudiante reprueba el curso.

Se presenta a continuación el esquema de evaluación para el caso en que el laboratorio presente sólo los niveles aprobado/no aprobado. Dado que en esta hipótesis en general no es posible implementar un seguimiento del trabajo de laboratorio de cada estudiante en forma individual, la evaluación de dicho trabajo se realiza en dos instancias: mediante la corrección de las tareas propuestas y mediante preguntas asociadas a éstas en las pruebas parciales (éstas serán identificadas como tales en dichas pruebas).

Exoneración. El estudiante debe cumplir los siguientes requisitos:

- llegar al nivel mínimo en los trabajos de laboratorio.
- reunir al menos el 60% del puntaje de parciales.
- obtener al menos el 25% en cada prueba parcial.

Suficiencia. El estudiante llega al nivel mínimo en los trabajos de laboratorio (medido como se indica en el punto anterior) y al 25% del puntaje total.

Insuficiencia. El estudiante no obtiene los puntajes de ninguna de las franjas anteriores.