

Organización para un Curso de Programación en un Contexto de Masividad

Resultados tras Experiencia de 4 Años

Carlos Luna, Martín Pedemonte, Marcos Viera, Eduardo Fraschini

Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

Resumen

Enseñar programación en un curso masivo es una tarea compleja. Este trabajo presenta la experiencia realizada por un grupo de docentes para mejorar el nivel de conocimiento transmitido en un curso de las características mencionadas. Se definieron roles claves para cada integrante del equipo; se profundizó en la utilización de trabajos de laboratorio obligatorios para la enseñanza de programación, haciéndolos eje central del curso; y se potenció la utilización de distintos medios informáticos, como una página *web* y un *newsgroup*, para, en particular, lograr atender a un alto porcentaje de estudiantes no presenciales. Los resultados, tras una experiencia de 4 años, extraídos de la evaluación final del curso y de otros posteriores, resultan alentadores para seguir trabajando en la misma dirección y expandir la propuesta a otros cursos de características similares.

Palabras clave: Organización/Gestión de Cursos de Programación, Enseñanza de Programación, Enseñanza en Contextos de Masividad.

1. Introducción

Según el diccionario de la Real Academia Española la palabra *masificar* significa “hacer multitudinario algo que no lo era”. El continuo crecimiento de la matrícula para la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la Universidad de la República (Uruguay), experimentado en los últimos años, ha transformado a los cursos iniciales de programación en masivos. Por tanto, resulta necesaria la adopción de medidas tendientes a la adaptación a la nueva realidad de un sistema que originalmente no fue pensado para ese contexto y que, además, no cuenta con recursos suficientes para el mismo. Este tipo de transformaciones abarcan, en general, aspectos tan disímiles como: la estrategia educativa a seguir, la evaluación de los conocimientos adquiridos por parte de los estudiantes, la organización y gestión de los cursos, entre otros [4].

En el contexto previo y habiendo detectado, en estudiantes de los últimos años de la carrera, importantes carencias de conocimientos y capacidades para llevar a cabo tareas de programación de sistemas de mediano y gran porte, el Instituto de Computación de la Facultad de Ingeniería (InCo de aquí en más) decidió abordar el problema e instrumentar cambios. Una medida clave en este sentido fue la asignación de un mayor número de recursos docentes para la mejora de algunos cursos masivos de programación considerados críticos. Este trabajo presenta la experiencia realizada por el equipo docente de la asignatura (indistintamente curso de aquí en más) Programación 2 a partir de 2003, que fue el año a partir del cual se comenzaron a instrumentar los cambios promocionados por el InCo. Algunos primeros resultados sobre esta experiencia fueron presentados con anterioridad en un evento local de la Facultad de Ingeniería de la Universidad de la República (Uruguay) [1] y están reportados en [2]. Posteriormente, en CIESC'2006 [5] se presentaron los resultados de la experiencia detallada en este artículo, para los tres primeros años (2003, 2004 y 2005).

El artículo se centra en la descripción y el análisis de una organización apropiada para un curso básico de programación en un contexto de masividad. El objetivo perseguido es el de obtener una mejora en la gestión de los recursos que repercute positivamente en la calidad del conocimiento adquirido por los estudiantes. La organización del resto del artículo se detalla a continuación. La sección 2 presenta las características de la asignatura Programación 2 en cuanto a contenidos y su ubicación en la carrera. La sección 3 describe el diagnóstico inicial realizado por el equipo docente a fines de 2002. La sección 4 introduce las principales líneas de acción establecidas y los cambios realizados a partir de 2003. La sección 5 presenta los resultados observados en la propia asignatura y en asignaturas posteriores de la carrera. Luego, la sección 6 analiza trabajos relacionados en el contexto local y, finalmente la sección 7 exhibe las conclusiones de este trabajo. Se incluye como apéndice el programa de la asignatura Programación 2.

2. La Asignatura Programación 2

El plan de estudios de la carrera de Ingeniería en Computación del InCo, vigente desde 1997, establece el cumplimiento de requisitos mínimos de créditos por Materias, que corresponden a “grandes áreas temáticas ligadas a un sector de la ciencia o de la técnica” [8].

Programación 2 es el segundo curso de la materia Programación. Debido a los requisitos de créditos establecidos para esta materia y a la actual oferta de cursos disponibles, es una asignatura obligatoria para la obtención del título de Ingeniero en Computación. Programación 2 se desarrolla en el tercer semestre de la carrera y es un curso intermedio entre Programación 1 y Programación 3, que se dictan durante el segundo y cuarto semestre, respectivamente.

El curso de Programación 2 hace hincapié en tres bloques temáticos claves: Inducción y recursión; Punteros y estructuras de memoria dinámica; y Tipos abstractos de datos. Programación 1 es un curso introductorio a los aspectos esenciales de la programación imperativa, en el cual se abordan entre otras temáticas: Instrucciones de control; Procedimientos y funciones; y Alcance de variables [9]. En Programación 3 se estudian estructuras de datos y algoritmos para la resolución de problemas computacionales con un fuerte hincapié en las técnicas de formulación e implementación de abstracciones de datos. En particular, se destacan como temas centrales de este curso: Análisis de algoritmos; Algoritmos de ordenación; Algoritmos en grafos; y Técnicas de diseño de algoritmos [10].

El objetivo general de la asignatura Programación 2 es capacitar al estudiante en el diseño y la implementación de programas de mediano porte, para lo que se utilizan técnicas consistentes en la formulación e implementación de abstracciones de datos. La dedicación horaria que se estima debe tener un estudiante semanalmente, durante las 15 semanas del curso, contempla: 2 horas de asistencia a clases teóricas, 2 horas de asistencia a clases prácticas y 8 horas de trabajo individual, tanto sea en los aspectos teóricos, prácticos, como de laboratorio del curso. La asignatura se evalúa a través de dos pruebas parciales y de la realización de dos trabajos de laboratorio. En realidad, se trata en general de un trabajo de laboratorio, que debe realizarse de manera individual, pero con dos entregas, una intermedia y otra final. Los trabajos de laboratorios son eliminatorios, es decir que el estudiante que no obtenga el nivel mínimo de suficiencia exigido en sus trabajos no podrá aprobar el curso, independientemente del puntaje que obtenga en las pruebas parciales. A partir de los resultados de las pruebas parciales, se clasifica a los estudiantes en tres franjas: *exoneración*, el estudiante ha aprobado la asignatura; *suficiencia*, el

estudiante ha adquirido el derecho de rendir el examen de la asignatura en tres oportunidades; e *insuficiencia*, el estudiante debe recurrir la asignatura. De aquí en más diremos que un estudiante *exonera*, *aprueba* o *pierde* el curso si su calificación corresponde a las siguientes franjas, respectivamente: *exoneración*, *suficiencia*, *insuficiencia*. Por más detalles acerca del curso referirse al apéndice del artículo.

3. Diagnóstico Inicial

La asignatura Programación 2 se desarrolla en condiciones de masividad. A partir del dictado correspondiente al año 2001 la inscripción ronda los 500 estudiantes, con un porcentaje de entre el 40% y el 50% de estudiantes no presenciales. Previo al comienzo del dictado del año 2003 el nuevo equipo docente realizó un diagnóstico sobre las que se consideraban las principales carencias que la asignatura tenía.

El equipo docente se organizaba de la siguiente manera: un docente de alta dedicación, que impartía las clases teóricas, y un grupo de docentes asistentes de baja o mediana dedicación que atendían consultas de práctico y de laboratorio. En primer lugar, la actividad docente resultaba *desenfocada*, cada docente debía impartir el práctico y discutir aspectos vinculados al laboratorio en la misma clase. En la práctica resultaba muy difícil que se dominaran ambos temas y esto repercutía negativamente en la calidad del curso brindado. Al no haber docentes encargados del diseño e implementación del laboratorio, existían detalles importantes que se pasaban por alto. Las reuniones de coordinación resultaban poco ejecutivas, dado que requerían la participación de todos los docentes del curso para unificar los ejercicios que debían ser resueltos en clase, para abordar las posibles dudas surgidas con respecto a detalles de diseño o implementación del laboratorio por parte de los alumnos, y para las tareas propias de gestión del curso.

Por otra parte, los ejercicios de práctico se encontraban poco jerarquizados, algo desactualizados y no abordaban en forma exhaustiva las temáticas del curso. Tampoco existían soluciones internas a disposición de los docentes sobre los ejercicios ni se publicaban sistemáticamente soluciones o materiales complementarios para apoyo a los estudiantes.

Si bien la información fundamental del curso estaba disponible a través de su página web, existían carencias importantes. En lo que concierne al laboratorio, solamente estaban disponibles las letras con la descripción general de los problemas, las posibles aclaraciones o asunciones generales y los resultados. En lo que respecta a material de práctico, únicamente estaban disponibles las letras de los ejercicios. Si bien existían letras de parciales y exámenes de años

anteriores, así como soluciones de los mismos, no siempre eran publicados.

4. Cambios Implementados

En el año 2003 se duplicó el número de recursos docentes asignados al curso Programación 2 con el objetivo de *mejorar la calidad de enseñanza de la programación y el nivel general del curso*. El equipo docente del curso pasó de 6 a 12 integrantes. En esta sección se describen las actividades llevadas a cabo desde 2003 a la fecha.

A partir del diagnóstico realizado y de la nueva disponibilidad de recursos docentes, se plantearon los siguientes objetivos generales:

1. *Definir actividades y roles claves dentro del equipo docente, potenciando los recursos docentes en las tareas más relevantes y especializando sus funciones para lograr mayor eficiencia.*

2. *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.*

3. *Potenciar la utilización de los medios informáticos disponibles: página web y newsgroup, para mejorar la relación de enseñanza/aprendizaje, fundamentalmente, sobre el conjunto de los estudiantes no presenciales, que abarca aproximadamente al 50% del total.*

4.1. Sobre el Objetivo 1

Para el cumplimiento del primer objetivo se consideró conveniente la conformación de dos equipos de trabajo complementarios. Uno encargado de 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 docentes mencionados. Cada uno de éstos en forma independiente realizaba reuniones semanales, en las cuales participaban los encargados y docentes del área para discutir sobre aspectos concernientes a la misma. Adicionalmente, se organizaron reuniones semanales de coordinación entre los encargados de ambos equipos y el encargado del curso, que permitían centralizar la información y la toma de decisiones críticas para el curso. De esta manera se lograron vincular los intereses y las distintas realidades de los dos equipos en una versión *integral* del curso hacia los estudiantes.

La organización descentralizada y las especializaciones docentes propuestas permitieron que cada docente, a pesar de su baja dedicación horaria, tuviera un conocimiento detallado de las temáticas a desarrollar y de la situación de su grupo de trabajo. También se consiguió una mayor ejecutividad de los grupos docentes y de las reuniones de planificación.

4.2. Sobre el Objetivo 2

Para el cumplimiento del segundo objetivo se abordaron, fundamentalmente, dos líneas de trabajo. La primera correspondió a la mejora de los materiales disponibles del curso y a la generación de nuevos materiales de apoyo. La segunda línea se centró en convertir a los trabajos de laboratorio obligatorios en *el eje central del curso*, por considerar que es a partir de la realización de estos trabajos que el estudiante adquiere la verdadera capacidad para realizar proyectos de mediano porte. Es éste, por otra parte, el objetivo principal del curso que no se estaba cumpliendo según datos del InCo referidos en la sección 1.

En lo que respecta a los materiales del curso, se dedicó un esfuerzo considerable a la generación de nuevos materiales para el laboratorio. Se agregaron a las imprescindibles letras que describen los problemas, documentos que especifican completamente los sistemas; ejemplos de uso de las aplicaciones y casos de prueba; errores frecuentes; y materiales complementarios sobre el lenguaje de programación, el compilador, el entorno de programación usado y las técnicas elementales para el rastreo y la corrección de errores de la aplicación. En relación a los materiales de práctico, se realizó una jerarquización de las letras de práctico incorporando ejercicios de nivel básico y avanzado para toda la temática abordada por el curso. Se fomentó el uso de *buenas prácticas de programación* a través del desarrollo de soluciones de ejercicios seleccionados del práctico, con un enfoque didáctico. Finalmente, se realizó una actualización integral de las guías de teórico.

Para transformar a los trabajos de laboratorio obligatorios en el eje central del curso resultaba indispensable incrementar la complejidad y el porte de los mismos, buscando cubrir la mayor cantidad posible de temas centrales del curso e integrarlos en el desarrollo de un sistema de tamaño medio. Asimismo, a fin de motivar a los estudiantes para la realización de estos sistemas, se optó por considerar problemáticas de mayor interés que las que se venían abordando en años anteriores. Por ejemplo, la implementación de simuladores de relativa complejidad para: un administrador de archivos de un sistema operativo, un manejador de base de datos o una planilla de cálculo. A pesar de la importancia que tienen en las asignaturas de programación los trabajos de laboratorio, la realización

de estos trabajos no repercutía en la nota que obtenía el estudiante al terminar el curso, ya que en el caso de Programación 2 los trabajos de laboratorio no llevaban calificación. Se consideró necesario revertir esta situación para que la realización de estos trabajos tuviera influencia en la calificación final del estudiante. Para esto, se decidió incluir un ejercicio en el segundo parcial que abordara una problemática muy similar a alguna tratada en el laboratorio, de tal manera que el estudiante que resolviera correctamente este ejercicio, en una instancia individual de evaluación, prácticamente obtuviera la condición de *suficiencia* del curso.

4.3. Sobre el Objetivo 3

Si bien no se descuidó el contacto directo entre los docentes y los alumnos mediante las clases presenciales de teórico y las clases y consultas presenciales semanales de práctico y de laboratorio, la alta proporción de estudiantes/docentes hizo imprescindible aumentar las vías de intercambio entre los mismos. Para esto se instrumentó un newsgroup para la realización de consultas de teórico, práctico y laboratorio, monitoreado diariamente por docentes integrantes de cada uno de los equipos asignados a tales fines. Para permitir un mejor acceso de los estudiantes a los materiales del curso, se sistematizó la publicación de todo el material a través de Internet (página web: <http://www.fing.edu.uy/inco/cursos/prog2/>), incluyendo los nuevos materiales generados, las letras de parciales y exámenes con soluciones, una cartelera con novedades e información general del curso.

5. Resultados Obtenidos

La presentación y el análisis de los resultados observados desde 2003 a la fecha se aborda desde dos ópticas, la interna o propia del curso y la correspondiente a la proyección hacia los cursos siguientes. El análisis se centra en los resultados de los cursos, sin considerar resultados de exámenes. Conviene aclarar que no se han detectado cambios en el porcentaje de aprobación de exámenes con respecto al promedio histórico.

5.1. Resultados en Programación 2

A continuación se presentan los resultados del curso correspondientes a las ocho últimas ediciones. Se indica la cantidad de inscriptos cada año como referencia, pero los porcentajes son calculados solamente sobre los estudiantes que efectivamente tuvieron actividad durante el curso, es decir, aquellos que por lo menos realizaron una de las entregas de laboratorio o se presentaron a la toma de al menos uno de los parciales. Conviene remarcar que no existen cambios significativos en los años considerados en el porcentaje

de estudiantes que no registran actividad, siendo éstos en promedio un 20 % de los inscriptos al curso, con la excepción del dictado correspondiente al año 2005 en el cual este valor se elevó al 30 %.

Año	Total	Pierden (%)	Aprueban (%)	Exoneran (%)	Ap. + Ex.(%)
1999	395	37,66	22,15	40,19	62,34
2000	405	40,48	26,78	32,74	59,52
2001	494	40,45	20,59	38,96	59,55
2002	481	39,90	42,36	17,74	60,10
2003	540	59,36	15,35	25,28	40,63
2004	502	54,39	17,04	28,57	45,61
2005	528	40,75	29,19	30,06	59,25
2006	502	35,06	30,13	34,81	64,94

Tabla 1: Estudiantes que Perdieron, Aprobaron y Exoneraron Programación 2 entre 1999 y 2006.

Como se puede apreciar en la tabla 1 se ha producido un incremento en la cantidad de estudiantes que pierden el curso desde el año 2003. Sin embargo, este resultado inicial se ha ido mitigando con los sucesivos dictados, tal cual lo ilustra la figura 1.

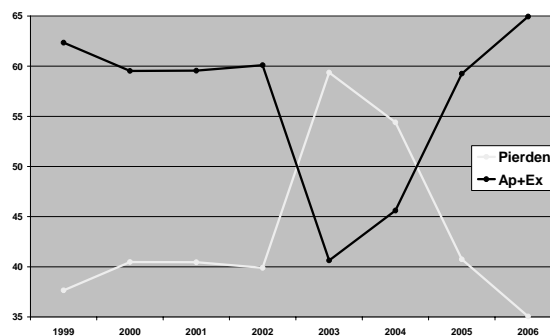


Figura 1: Grafica de los porcentajes de Pérdida y Aprobación+Exoneración entre 1999 y 2006.

Consideramos que el nivel de exigencia del curso aumentó, debido principalmente a los cambios que se operaron en los trabajos de laboratorio obligatorios (incremento de tamaño y complejidad).

Parece razonable inferir que el cambio inicial presentó un salto en cuanto al nivel de dificultad que no pudo ser acompañado por los estudiantes. Posteriormente, se observan mejoras progresivas en los resultados, sin bajar el nivel de exigencia del curso. Una hipótesis al respecto es que al ser conocido por los estudiantes el

nivel de dificultad del curso, éstos comenzaron a afrontarlo de manera diferente.

Año	Ent.	Aprob. (%)	Pierden (%)	No-Ind (%)
2003	443	86,00	6,77	7,23
2004	399	92,24	7,76	0,00
2005	346	80,93	14,45	4,62
2006	385	89,35	9,61	1,04

Tabla 2: Resultados del primer laboratorio de 2003 a 2006.

Año	Ent.	Aprueban (%)	Pierden (%)	No-Ind (%)
2003	280	65,71	23,57	10,72
2004	263	70,73	25,47	3,80
2005	245	84,08	15,10	0,82
2006	313	80,51	16,93	2,56

Tabla 3: Resultados del segundo laboratorio de 2003 a 2006.

En las tablas 2 y 3 se presentan los resultados para cada una de las instancias de evaluación del laboratorio de los 4 últimos años. De los datos puede inferirse un importante abandono de la asignatura entre la primera y segunda instancia de entrega, porcentaje que no difiere mayoritariamente del de años anteriores. Sin embargo, notamos una disminución del porcentaje de aprobación de la segunda instancia de evaluación con respecto al de años previos. Conviene destacar que en los últimos años perfeccionamos los procedimientos de corrección de las tareas y de detección de trabajos no individuales. 60 estudiantes en el año 2003, 10 en 2004, 18 en 2005 y 12 en 2006 perdieron el curso por incumplir la cláusula de individualidad a la que están sujetos los trabajos de laboratorio.

Año	Aprueban Lab.	Exoneran (%)	Aprueban (%)	Pierden (%)
2003	184	60,87	36,96	2,16
2004	186	61,29	36,56	2,15
2005	206	50,49	49,03	0,48
2006	252	53,17	46,03	0,80

Tabla 4: Estudiantes que Perdieron, Aprobaron y Exoneraron el curso entre 2003 y 2006, considerando sólo la población que aprobó las tareas de laboratorio.

Si bien se observa un aumento en la cantidad de estudiantes que pierden el curso en los cuatro últimos años, respecto a los anteriores, de los estudiantes que aprobaron las tareas de laboratorio, un número inferior al 3% no logró aprobar o exonerar el curso, como lo muestra la tabla 4. Adicionalmente, el porcentaje de estudiantes que exoneró la asignatura es, en general, mayor al porcentaje de los que la aprobaron. En el año 2005 los resultados se apartaron un poco de esta regla, lo cual llevó al equipo docente a replantearse la estructura de los parciales para corregir esta desviación. El efecto de este cambio ha comenzado a visualizarse en los resultados obtenidos en el año 2006 y confiamos en que esta tendencia se mantenga durante el presente año.

El análisis previo reafirma que los trabajos de laboratorio se han transformado en el eje central del curso, en concordancia con los objetivos planteados.

5.2. Resultados Observados en Cursos Posteriores

Es una tarea compleja realizar un análisis preciso de los efectos provocados, por los cambios implementados en Programación 2, en las asignaturas siguientes de la carrera, fundamentalmente porque las otras asignaturas no son departamentos estancos, en los cuales no se producen cambios. Programación 3 y Programación 4 fueron las asignaturas elegidas para el análisis. Se tuvieron en cuenta los años considerados significativos para comparar por los propios docentes encargados. Programación 4 es el curso temáticamente posterior a Programación 3, donde se desarrollan los fundamentos de la programación orientada a objetos. Por más detalles acerca de este curso ver [10].

En la siguiente tabla se presentan los resultados para el curso de Programación 3. Se indica la cantidad de inscriptos cada año; el porcentaje de estudiantes activos, es decir los que realizaron alguna actividad durante el curso; y los porcentajes de reprobación, aprobación y exoneración, calculados sobre los estudiantes activos.

Año	Total	Activos (%)	Pierden (%)	Aprueban (%)	Exon. (%)
2001	333	84,08	42,86	26,78	30,36
2002	409	88,51	47,51	42,27	10,22
2003	352	83,52	35,03	54,42	10,55
2004	334	96,71	19,19	56,35	24,46
2005	293	95,22	28,31	54,84	16,85
2006	350	88,86	27,00	49,20	23,80

Tabla 5: Estudiantes Activos que Perdieron, Aprobaron y Exoneraron el curso Programación 3 entre 2001 y 2006.

Se registró un aumento en el porcentaje de estudiantes que presentan actividad y una disminución significativa en el porcentaje de estudiantes que pierden el curso con respecto al dictado en los años previos a los cambios instrumentados (2001 y 2002). Consideramos que se constata en la práctica que al enfrentarse los estudiantes durante Programación 2 con un trabajo de laboratorio de un porte considerable, se disminuyó el impacto que generaba este tipo de trabajos en Programación 3. En este curso se cuenta además con el agravante del cambio de lenguaje de programación. Se pasa de uno fuertemente tipado (como Modula-2 o Pascal) a otro que no lo es (C/C++). El cambio observado, debido al trabajo realizado en Programación 2, ha repercutido positivamente en Programación 3, permitiendo que se aproveche la experiencia en programación de los estudiantes para profundizar en la implementación de estructuras de datos y algoritmos avanzados.

En la siguiente tabla se presentan los resultados para el curso de Programación 4. Se indica la cantidad de inscriptos cada año; y los porcentajes de reprobación, aprobación y exoneración calculados sobre los estudiantes activos. Se consideran sólo los años relevantes, de 2003 a 2006. Cabe aclarar que es 2004 el año que corresponde al primero de la transición realizada en Programación 2.

Año	Total	Activos (%)	Pierden (%)	Aprueban (%)	Exon. (%)
2003	269	81,78	54,09	35,00	10,91
2004	373	100,00	28,69	57,37	13,94
2005	356	100,00	29,21	55,90	14,89
2006	384	94,01	14,96	56,51	28,53

Tabla 6: Estudiantes Activos que Perdieron, Aprobaron y Exoneraron el curso Programación 4 entre 2003 a 2006.

Las estadísticas marcan una disminución significativa en el porcentaje de estudiantes que pierden el curso con respecto a los años previos al año en que se implementaron los cambios en Programación 2. Al haber disminuido el impacto que se generaba en los trabajos de laboratorio en Programación 3 y permitir que ahora dicho curso pueda concentrarse en temas propios de la asignatura, parece estar observándose un efecto de proyección hacia los cursos superiores. Resulta razonable pensar que al llegar los estudiantes con prácticas y conocimientos mejor asimilados desde cursos previos, aprovechan en mejor medida los cursos más avanzados y se obtienen mejores resultados. Por último, conviene destacar también que existe la percepción por parte de los docentes implicados en el

desarrollo de las asignaturas Programación 3 y Programación 4 que se han obtenido logros desde un punto de vista *cualitativo* en la formación con la cual llegan los estudiantes a sus cursos.

6. Trabajos Locales Relacionados

El presente trabajo se enmarca en el contexto de las políticas planteadas por la Facultad de Ingeniería de la Universidad de la República para abordar la problemática de la enseñanza en contextos de masividad. En particular, consiste en la puesta en práctica de algunas de las medidas propuestas en [4].

Existen experiencias recientes de adopción de nuevas tecnologías de la información y la comunicación (NTICs) para la mejora de los cursos en el contexto local, respondiendo principalmente a dos problemáticas [6]. La primera obedece a la imposibilidad de los estudiantes de asistir periódicamente a clases, con lo cual es necesario adoptar medidas tendientes a adaptar los cursos para permitir sus desarrollos en modalidad de educación a distancia o semi presencial [7]. La segunda problemática corresponde a la realidad que surge en los cursos masivos donde existe poco contacto entre docentes y estudiantes. En este último caso se pretende complementar el desarrollo habitual del curso, aumentando las vías de comunicación existentes entre ambos actores [3]. La realidad del curso y las acciones llevadas adelante por el equipo docente de Programación 2 se enmarcan en esta última dirección. Sin embargo, a diferencia de las experiencias referidas, nuestro trabajo aborda también el problema de la organización y gestión de los recursos docentes disponibles, como un aspecto clave.

7. Conclusiones

En un contexto de masividad y escasez de recursos asignados a la enseñanza resulta vital la organización del equipo docente de manera de obtener un mejor aprovechamiento del mismo. Este trabajo propone y experimenta con una estructura descentralizada de organización con fuerte especialización de los recursos docentes en sus roles y tareas asignadas. De esta manera se logra que, a pesar de su baja dedicación horaria, cada docente tenga un conocimiento detallado de las temáticas a desarrollar y de la situación de su grupo de trabajo. También se consigue una mayor ejecutividad de los grupos docentes y de las reuniones de planificación. Tenemos la convicción de que esta especialización de las funciones docentes incide positivamente en la calidad de la enseñanza de la programación y en el nivel general del curso.

Actualmente y tras 4 años de trabajo se ha logrado una puesta a punto del curso Programación 2, generándose

una cantidad significativa de material y mejorándose el existente. Consideramos que ésta es una medida paliativa fundamental al problema de la baja cantidad de tiempo que los docentes pueden dedicar a cada estudiante. Para enfrentar este tema, el *newsgroup* ha sido también una herramienta muy importante ya que permite aumentar el intercambio de ideas entre docentes y estudiantes, y fundamentalmente entre los mismos estudiantes. Asimismo, la utilización de distintos medios informáticos con todos los materiales del curso para, en particular, llegar a un alto porcentaje de estudiantes no presenciales.

Estimamos conveniente que los trabajos de laboratorio tengan una incidencia determinante en cursos básicos de programación e influyan directamente en sus resultados. En la experiencia desarrollada en Programación 2 el aumento en la complejidad del laboratorio redundó en un aumento de la complejidad del curso, por lo que los cambios realizados a partir de 2003 no han logrado mejorar sustancialmente sus estadísticas de *suficiencia/exoneración*. Originalmente se observó una retracción en los niveles de aprobación que ha sido paleada paulatinamente hasta retomar los niveles históricos. Sin embargo, estamos convencidos de que las mejoras obtenidas son de carácter *cualitativo*. Esta observación se basa fundamentalmente en el análisis de los resultados obtenidos en los cursos siguientes (en particular, Programación 3 y Programación 4), y en la percepción de sus docentes, que observan una mejora en la preparación de los estudiantes que ingresan a los mismos, tanto a nivel de conocimientos como de capacidades.

Los resultados de esta experiencia resultan alentadores para seguir trabajando en la misma dirección y expandir la propuesta a otros cursos de características similares. Algunos cambios ya se han comenzado a instrumentar en asignaturas posteriores de la carrera de Ingeniería en Computación del InCo.

Agradecimientos

A Javier Alliaume que formó parte del equipo que implementó los cambios. Al resto del equipo docente de Programación 2 de los años 2003 al 2006 que colaboraron en la formulación de propuestas y día a día implementaron los cambios convenidos. A Daniel Calegari y a Elena Balboa por brindarnos la información necesaria para abordar la sección en la que se presentan los resultados de los cursos.

Referencias

[1] Alliaume, J; Luna, C; Pedemonte, M; Viera M. Programación 2: Experiencias y Propuestas. Segundo Congreso de Enseñanza en la Facultad de Ingeniería, Octubre

de 2004, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay.

[2] Alliaume, J; Luna, C; Pedemonte, M; Viera, M. Programación 2: Experiencias y Propuestas. Reporte Técnico 05-03, Abril de 2005, PEDECIBA Informática, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay. Disponible en línea en <http://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0503.pdf>. Consultado en Junio de 2007.

[3] Fernández, F.; González, J.; Míguez, M.; Peré, N.; Rodés, V. Elaboración de Material de Apoyo y Evaluación del Curso de Computación 1. COBENGE 2004, Congreso Brasileiro de Enseñanza de la Ingeniería, Setiembre de 2004, Brasilia, Brasil.

[4] La Problemática de la Masividad en la Enseñanza en la Facultad de Ingeniería. Aprobado por el Consejo de la Facultad de Ingeniería en su sesión del 08/03/2001. Disponible en línea en <http://www.fing.edu.uy/institucion/masiv.htm>. Consultado en Junio de 2007.

[5] Luna, C; Pedemonte, M; Viera, M; Frascchini, E. Organización para un Curso de Programación en un Contexto de Masividad. CIESC'2006, Agosto de 2006, Santiago de Chile.

[6] Peré, N. and Miguez, M. Experiencias en la implementación de Nuevas Tecnologías en cursos de grado de la Facultad de Ingeniería de la Universidad de la República, Uruguay. Docencia Universitaria e Innovación, Editora Universidad de Barcelona, Julio de 2002, Tarragona, España.

[7] Peré, N; Rodés, V; Viera, O; Pedemonte, M. Grupo de trabajo en educación a distancia. Curso de modelado y optimización con GAMS. COBENGE 2002, Congreso Brasileiro de Enseñanza de la Ingeniería, Setiembre de 2002, San Pablo, Brasil.

[8] Plan de la Carrera de Ingeniería en Computación. Aprobado por el Consejo de la Facultad de Ingeniería en su sesión del 02/07/97. Aprobado por el Consejo Directivo Central en su sesión del 11/11/97. Disponible en línea en <http://www.fing.edu.uy/servadm/secretaria/comisiones/claustro/computacion.html>. Consultado en Junio de 2007.

[9] Programa de la Asignatura Programación 1. Disponible en línea en: <http://www.fing.edu.uy/inco/cursos/prog1/pm/field.php/Main/Descripci%f3n>. Consultado en Junio de 2007.

[10] Programa de la Asignatura Programación 3. Disponible en línea en <http://www.fing.edu.uy/inco/cursos/prog3/>. Consultado en Junio de 2007.

[11] Programa de la Asignatura Programación 4. Disponible en línea en <http://www.fing.edu.uy/inco/cursos/prog4/>. Consultado en Junio de 2007.

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, árboles. 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.

- Estructuras de Datos y Algoritmos. Mark Allen Weiss. Addison-Wesley Iberoamericana, 1995.

- Estructuras de Datos y Algoritmos. Aho, Hopcroft y Ullman. Addison Wesley Iberoamericana, 1988.

- 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.

Dirección de Contacto de los Autores:

Carlos Luna

Julio Herrera y Reissig 565
11300 Montevideo

Uruguay

e-mail: cluna@fing.edu.uy

<http://www.fing.edu.uy/~cluna/>

Martín Pedemonte

Julio Herrera y Reissig 565
11300 Montevideo

Uruguay

e-mail: mpedemon@fing.edu.uy

Marcos Viera

Julio Herrera y Reissig 565
11300 Montevideo

Uruguay

e-mail: mviera@fing.edu.uy

<http://www.fing.edu.uy/~mviera/>

Eduardo Fraschini

Julio Herrera y Reissig 565
11300 Montevideo

Uruguay

e-mail: efraschi@fing.edu.uy

Carlos Luna es Magister en Informática, docente e investigador del grupo de Métodos Formales del Instituto de Computación de la Facultad de Ingeniería de la Udelar, Uruguay. Es responsable del curso Programación 2. Por información adicional consultar el curriculum vitae accesible desde su página web.

Martín Pedemonte es Ingeniero en Computación, docente e investigador del grupo de Computación de Alta Performance del Centro de Cálculo del Instituto de Computación de la Facultad de Ingeniería de la Udelar, Uruguay. Se desempeñó hasta 2006 como responsable de las actividades prácticas del curso Programación 2.

Marcos Viera es Ingeniero en Computación, docente e investigador del Laboratorio de Ciencias de la Computación de la Facultad de Ingeniería de la Udelar, Uruguay. Es responsable de las actividades de laboratorio del curso Programación 2. Por información adicional consultar el curriculum vitae accesible desde su página web.

Eduardo Fraschini es Ingeniero en Computación, docente e investigador del grupo de Lenguaje Natural del Instituto de Computación de la Facultad de Ingeniería de la Udelar, Uruguay. Se desempeñó hasta 2006 como docente del curso Programación 2.
