



Revista Electrónica "Actualidades
Investigativas en Educación"

E-ISSN: 1409-4703

revista@inie.ucr.ac.cr

Universidad de Costa Rica
Costa Rica

Murillo Rivera, Maureen

Explorando el proceso de enseñanza y de aprendizaje en el área de la programación de
computadoras

Revista Electrónica "Actualidades Investigativas en Educación", vol. 6, núm. 1, enero-abril, 2006, p. 0

Universidad de Costa Rica

San Pedro de Montes de Oca, Costa Rica

Disponible en: <http://www.redalyc.org/articulo.oa?id=44760105>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



Universidad de Costa Rica
Facultad de Educación
Instituto de Investigación en Educación

ACTUALIDADES INVESTIGATIVAS EN EDUCACION



EXPLORANDO EL PROCESO DE ENSEÑANZA Y DE APRENDIZAJE EN EL ÁREA DE LA PROGRAMACIÓN DE COMPUTADORAS

EXPLORING THE TEACHING-LEARNING PROCESS
IN THE AREA OF COMPUTER PROGRAMMING

Maureen Murillo Rivera¹

Resumen: En este artículo se propone incorporar horas de laboratorio de programación para aquellos cursos que enfrentan por primera vez a los estudiantes con el mundo de la programación de computadoras. La propuesta está basada en un proyecto de investigación realizado en la Escuela de Ciencias de la Computación e Informática de la Universidad de Costa Rica, con el apoyo del Instituto de Investigación en Educación. De esta manera se realizó un proceso de análisis y se fundamentó con metodologías pedagógicas conocidas, principalmente relacionadas con la enseñanza de destrezas y habilidades. Se discute sobre cómo las sesiones de laboratorio de programación, en lugar de generar dependencia entre el profesor y el alumno, le facilitan al estudiante un proceso paulatino para el desarrollo real de sus conocimientos, culminando en el cumplimiento de los objetivos del curso.

Descriptores: ENSEÑANZA DE PROGRAMACIÓN DE COMPUTADORAS/ LABORATORIOS DE PROGRAMACIÓN/ INVESTIGACIÓN-ACCIÓN/

Abstract: This article deals with incorporating programming laboratory hours to support courses that introduce students to the world of computer programming. The proposal is the result of a research that took place in the Department of Computer Sciences and Informatics of the University of Costa Rica, with the support of the Research Institute of Education. The research analyzed a specific course that was based on known pedagogical methodologies to develop abilities and skills in the students. In this article it is discussed on how the sessions of programming laboratory, instead of generating dependency between the professor and the student, facilitate a gradual process to the student for the real development of his/her knowledge, culminating in the fulfillment of the objectives of the course.

Key words: TEACHING COMPUTING PROGRAMMING/ COMPUTER PROGRAMMING LABORATORIES/ ACTION RESEARCH/

Introducción

Es común encontrar, en los diferentes centros de enseñanza, que los cursos de programación de computadoras se realicen en un aula, tal vez con una computadora, en donde el profesor imparte los fundamentos teóricos de la programación e ilustra con ejemplos. Luego se asignan trabajos y proyectos prácticos a los estudiantes.

¹ Magister en Computación e Informática; Licenciatura y Bachillerato en Computación e Informática, todos los títulos de la Universidad de Costa Rica. Profesora e investigadora de la Escuela de Ciencias de la Computación e Informática de la Universidad de Costa Rica.

Correo electrónico: maureenm@cariari.ucr.ac.cr

Artículo recibido: 7 de julio, 2005

Aprobado: 27 de febrero, 2006

Este artículo expone el resultado de una investigación que consistió en la realización de una experiencia pedagógica que incorporó horas de laboratorio de programación en la metodología de un curso de programación que no las contemplaba. El objetivo de la investigación es mostrar la conveniencia de una etapa de práctica guiada por el docente, en el curso Programación 1, en donde los estudiantes son introducidos, por primera vez, a un mundo totalmente desconocido para ellos.

Esta investigación constituyó el estudio de un caso. Por lo tanto, no presenta la rigurosidad metodológica de una investigación tradicional. Los métodos utilizados estuvieron basados en los principios de la investigación-acción y de la investigación en el aula, que constituyen una estrategia fundamental para transformar los procesos de enseñanza y de aprendizaje. Este artículo expone la propuesta pedagógica que resulta de la investigación realizada en la Escuela de Ciencias de la Computación e Informática (ECCI) de la Universidad de Costa Rica (UCR).

Justificación teórica

2.1. Antecedentes

La Escuela de Ciencias de la Computación e Informática (ECCI) ha dado el primer paso de un proceso de revisión curricular de los cursos que integran su plan de estudio. Este interés institucional se ha plasmado en la conformación de una Comisión de Planes y Programas, la cual ha planteado un proceso de mejoramiento continuo.

Esta labor se logra mediante el desarrollo de una serie de actividades que pretenden mejorar los procesos de enseñanza y de aprendizaje. Las actividades se han organizado por áreas temáticas, para facilitar las discusiones relacionadas con los cursos. Una de estas áreas la conforman los cursos relacionados con programación, específicamente: Programación 1, Programación 2 y Estructuras de Datos (Comisión de Planes y Programas, 2004).

La revisión curricular de un curso es un proceso complejo e integral de análisis de los diversos aspectos que conllevan el diseño y la enseñanza de éste; por ejemplo, objetivos, contenidos, metodología de enseñanza y recursos disponibles. Esta revisión puede llevarse a cabo siguiendo uno de los siguientes enfoques: análisis que va de lo general a lo específico, o partiendo de aspectos específicos para luego integrarlos en una propuesta

general. En este sentido, debe impulsarse cualquier esfuerzo o experiencia innovadora dirigida a mejorar el curso.

Desde años atrás, el Centro de Asesoría Estudiantil (CASE) de Ingeniería, ha organizado los llamados "estudiaderos de matemática". Estos "estudiaderos" son sesiones de estudio guiadas por estudiantes-asistentes avanzados en la temática, los cuales dan apoyo a los estudiantes de cursos de matemática del programa de la ECCI, mediante la resolución de problemas o dudas relacionados con la materia.

Más recientemente, en el primer ciclo del 2003, se realizó un plan piloto para implementar "estudiaderos de programación", como un esfuerzo coordinado entre la ECCI y el CASE de Ingeniería. Las sesiones se realizaron en laboratorios de la Escuela y eran guiadas por tres tutores: un profesor y dos estudiantes. Estaban dirigidas hacia estudiantes de los cursos de Programación 1 y Programación 2.

En el informe de estos "estudiaderos" presentado por el CASE (Rodríguez, 2004), se explica que en los años 2000 y 2001 se dieron tutorías de programación teóricas. Tanto los tutores como los participantes indicaron que precisamente una de las principales limitaciones era la dificultad de aclarar dudas si no tenían acceso a una computadora en el momento. Esto se esperaba superar con la nueva modalidad de "estudiadero".

Al evaluar los resultados de la experiencia del "estudiadero" de programación, el CASE indica (Rodríguez, 2004, p.4):

El estudiadero es un recurso muy importante de cara a la metodología y programa tradicional de estos cursos, los cuales según manifiestan los estudiantes que consultan y según perciben los tutores, al carecer de laboratorio vuelven el ritmo del curso más rápido que lo que el estudiante mismo puede absorber y se enfrenta solo al hacer las tareas con que sólo en apariencia había comprendido la materia cuando el o la docente explicó, con lo cual las tareas adquieren un grado de dificultad que solos no pueden enfrentar. El estudiadero en este sentido fue un buen recurso también.

Los participantes indican que el estudiadero les ayudó a una mayor integración de la teoría con la práctica y en ese sentido a una mayor comprensión de la teoría vista en clase y de los problemas. Les ayudó a perder el miedo a los exámenes, mejorar las

técnicas de estudio, superar las notas de los exámenes y sentirse más seguros en general.

En general, el informe concluye que el resultado de la experiencia fue positivo, y que incluso sirvió para generar motivación en los estudiantes por el curso y la materia.

Otro de los esfuerzos por apoyar la enseñanza de la programación de computadoras, es el material didáctico en formato digital producido en el 2004, para uso en los cursos de programación, patrocinado por la Cámara Costarricense de Tecnologías de Información y Comunicación (CAMTIC) y financiado por el Proyecto Prosoftware.

Este material es producto de un interés interinstitucional de promover la vinculación del sector industrial de tecnologías de información del país con las instituciones académicas (Casasola, 2004a; Casasola, 2004b). Su objetivo es servir de guía o complemento en la enseñanza de la programación, tomando como base el paradigma de orientación a objetos.

El diseño de contenido estuvo a cargo del profesor de la ECCI Edgar Casasola Murillo. Contó con la participación de profesores de los siguientes centros educativos: CENFOTEC, Colegio Universitario de Alajuela, Colegio Universitario de Cartago, Colegio Universitario de Puntarenas, Instituto Tecnológico de Costa Rica, Universidad de Costa Rica, Universidad Estatal a Distancia, Universidad Latina y Universidad Nacional.

Una de las iniciativas más recientes, es la emprendida por el estudiante Juan de Dios Murillo de la maestría académica en Computación e Informática de la UCR, quien trabaja en la formulación de una propuesta de tesis, bajo la dirección del profesor Vladimir Lara.

La propuesta gira alrededor de la idea de desarrollar una aplicación de software, que sirva de apoyo en el aprendizaje de los principios de análisis y diseño de programas orientados a objetos. El software le permitiría al estudiante abstraerse del código de un lenguaje de programación específico para concentrarse en el buen diseño de una solución computacional. La aplicación estaría constituida de tres capas: la gráfica que le permite al estudiante el diseño de la solución; la capa que representa el diseño en un lenguaje formal de representación (UML-Unified Modeling Language), y la que genera el código en el lenguaje de programación Java.

Por otro lado, el proyecto de investigación "Incorporación de un laboratorio de programación en el curso de Programación 1, como una experiencia de enseñanza y aprendizaje", a cargo de la profesora de la ECCI Maureen Murillo Rivera y dentro del cual se enmarca esta propuesta, desarrolló el estudio de un caso en el que se incluyeron horas de laboratorio en el curso, como un espacio didáctico (Murillo, 2004).

Para ello se realizaron, semanalmente, sesiones de laboratorio (12 en total) de 2 horas cada una en un cuarto con computadoras, como complemento a las 4 horas semanales formales del curso. Eran dirigidas por la profesora con la ayuda de un asistente. Su objetivo era evaluar esta innovación educativa como mejora metodológica en el aprendizaje de los estudiantes y determinar los beneficios de la implementación de esta metodología en el primer curso de programación.

Algunos de los logros alcanzados con la realización de este proyecto son (Murillo, 2004):

1. Generó motivación en los estudiantes, la cual pudo notarse con las siguientes observaciones:
 - a. De todos los estudiantes a los que se les ofreció integrarse en las sesiones de laboratorio, un 80% optó inicialmente por participar en la experiencia.
 - b. La asistencia de los estudiantes a las sesiones fue muy alta.
 - c. Se percibía el interés de los estudiantes en los temas tratados, al consultar acerca de la temática y al aplicar lo aprendido en los laboratorios en las tareas programadas, a pesar de no ser requisito.
 - d. Una vez finalizados los laboratorios, estudiantes próximos a matricular Programación 1 expresaron su interés en que se repitiera la experiencia.
2. Constituyó una fuente de apoyo diferenciado oportuno. Los laboratorios constituyeron una forma de promover la adquisición de conocimientos, más allá de los previstos en el programa oficial del curso para estudiantes destacados. Al mismo tiempo, le permitían a la profesora ofrecer una mayor atención a aquellos estudiantes que presentaban algún problema con temas particulares del curso. En ambos casos, la ayuda de la profesora y de los asistentes estaba disponible en el momento en que surgían las dudas.

3. Permitió realizar una autoevaluación que se dio en dos sentidos: tanto para el estudiante como para la profesora. Con la metodología actual del curso (sin laboratorios), el estudiante, al realizar la práctica por su cuenta, cree estarlo haciendo bien, hasta que recibe las observaciones del profesor sobre las tareas y exámenes. Los laboratorios le permiten al alumno detectar deficiencias a tiempo, al medir su propio desempeño en las actividades planteadas. De igual forma, en ocasiones el docente no es consciente de que la forma en que expuso algún tema no fue la más adecuada para el aprendizaje de los estudiantes. La interacción con los estudiantes durante los laboratorios le permitió darse cuenta de las deficiencias de la enseñanza que tuvo a cargo.
4. Sirvió como medio de divulgación de conocimientos. Se publicaron los laboratorios en una página web, permitiendo el aprendizaje virtual y a distancia, incluso para aquellos casos en que el estudiante no puede asistir a una sesión. Otro beneficio que se obtiene con esta forma de publicación, es el aprendizaje "fuera de tiempo" en relación con los tiempos previstos para el proceso de enseñanza-aprendizaje y en una población ampliada. Por ejemplo, luego de 6 meses de finalizados los laboratorios, se conoció del caso de un estudiante avanzado de carrera que utilizó los laboratorios (que seguían disponibles vía Internet) para conocer cómo se manipulan archivos por medio de interfaces gráficas de usuario, usando el lenguaje de programación Java. La información publicada en Internet con propósitos educativos promueve el autoaprendizaje, siempre que siga los lineamientos pedagógicos adecuados. Además de que facilita el avance individualizado, según el ritmo de aprendizaje de cada estudiante.

Los resultados de esta experiencia fueron sometidos a discusión entre los profesores del curso, y se obtuvieron críticas y aportes valiosos para la investigación. Actualmente, seis profesores imparten el curso de Programación 1, de los cuales cuatro han realizado sesiones de laboratorio complementarias al curso por iniciativa propia.

Finalmente, se destaca que en la reforma curricular del 2000 del plan de estudio de la ECCI, se incorporaron cursos de laboratorio para los siguientes cursos: Diseño de Circuitos Digitales, Sistemas Operativos, Bases de Datos 1, Redes de Computadoras 1, Ingeniería de Software 1 e Ingeniería de Software 2 (ECCI, 1999).

2.2. Visión pedagógica

El curso Programación 1 tiene como objetivo general *"proveer al estudiante la formación básica en programación para su adecuado desempeño en los cursos subsiguientes del área de programación, fomentando en el estudiante habilidades generales para la resolución de problemas de programación"* (ECCI, 1999, p. 24).

A su vez,

la programación de las aplicaciones descansa en el empleo de lenguajes computacionales que permiten escribir las instrucciones que ejecutará el computador. La secuencia de instrucciones forma la lógica del programa. Tal secuencia obedece al algoritmo empleado para resolver el problema que plantea el programa. (ECCI, 1999, p. 4)

La naturaleza del curso, al proveer las bases para los procesos de enseñanza y de aprendizaje en los cursos posteriores de la carrera, hace determinante que se busque continuamente una formación de calidad, que le permita al estudiante desarrollar destrezas de resolución de problemas, mediante el uso del computador.

Tradicionalmente, este curso se ha impartido en 4 horas semanales de teoría. Las horas de teoría son impartidas en un aula provista de una pizarra y de un computador. Cada profesor tiene su propia metodología de enseñanza y desarrolla una dinámica diferente en sus lecciones. Sin embargo, en el nivel general, puede decirse que las clases de teoría consisten de explicaciones de conceptos, desarrollo de ejemplos en pizarra o en el computador por parte del profesor y el desarrollo de ejercicios por parte de los estudiantes en sus cuadernos o en la pizarra.

Adicionalmente, el profesor suele asignar tareas cortas y tareas programadas que el estudiante debe desarrollar fuera de las horas de clase. El estudiante, además, puede asistir a consulta con su profesor en horas específicas para la resolución de dudas.

La propuesta pedagógica planteada en este artículo, sugiere introducir en este proceso de enseñanza y de aprendizaje, horas de laboratorio adicionales a las horas de teoría, en donde

bajo la guía del profesor, los estudiantes realizan ejercicios o prácticas directamente en el computador.

La orientación teórica educativa en la que se basó esta propuesta es la constructivista. Bajo la concepción constructivista del aprendizaje, el objetivo de la educación es promover los procesos de crecimiento personal del alumno en el marco de la cultura del grupo al que pertenece. Para que estos aprendizajes se produzcan satisfactoriamente, es necesario que se suministre una ayuda específica, mediante la participación del alumno en actividades intencionales, planificadas y sistemáticas, que logren promover, guiar y orientar dicho aprendizaje en el estudiante (Díaz-Barriga y Hernández, 2002).

Esta visión pedagógica se desglosa en:

1. La sociedad
2. El sujeto educativo
3. Los procesos de enseñanza y de aprendizaje
4. La tecnología

A continuación, se comentan estas perspectivas relacionadas con la propuesta de laboratorios para el curso Programación 1.

2.2.1. Visión de la sociedad

Actualmente vivimos en la sociedad del conocimiento. Ante tanta cantidad de información disponible, lo fundamental es aprender a aprender y, por lo tanto, los docentes deben enfocar sus esfuerzos en enseñar a aprender. Para lograrlo debe prestarse cuidado al modelo pedagógico utilizado en la enseñanza.

La "pedagogía informacional" es un nuevo enfoque educativo, ante el cual los docentes y estudiantes deben asumir un nuevo rol de mediaciones entre la experiencia humana y la información existente y, sobre todo, caer en la cuenta que la información debe ser punto de partida y de llegada en el proceso de enseñanza aprendizaje (Picardo, 2002).

Teniendo presente esta necesidad, no es suficiente conformarse con que el estudiante adquiera los conocimientos previstos en un curso. Cobra especial importancia cómo los

aprende, es decir, el proceso por el cual pasa para alcanzar las expectativas del curso. El docente debe preguntarse:

- *¿Asimiló el estudiante los conocimientos previstos en el programa del curso?*

Generalmente, esta es la preocupación básica de los docentes. En un curso como Programación 1 esta pregunta se responde luego de la revisión de las soluciones desarrolladas por los estudiantes a problemas asignados por el profesor. Estas soluciones constituyen el producto final de un proceso de aprendizaje.

- *¿Adquirió las destrezas adecuadas de aprendizaje en la temática del curso?*

En este punto lo importante es el proceso de desarrollo paulatino de las habilidades. Por ejemplo, en el curso de Programación 1 una solución a un problema puede obtenerse después de un proceso cíclico de "prueba y error", hasta obtener una solución que satisfaga las normas de evaluación. Incluso podría construirse una solución correcta, pero basada en premisas falsas o distorsionadas. Esta metodología no contribuye a la necesidad de "aprender a aprender". Sin querer violar el estilo de aprendizaje de cada individuo, la aplicación de técnicas de abstracción, deducción, razonamiento y modelaje para obtener soluciones, favorece un proceso de aprendizaje metódico que el estudiante podrá aplicar en un futuro, cuando enfrente nuevos conceptos e información.

"Aprender a aprender" requiere que la persona sea capaz de enfrentarse solo a problemas, que busque alternativas de solución y que aplique la que mejor se adecúe a las circunstancias. Esto deja claro que los estudiantes deben entrenarse en la exploración y búsqueda de soluciones en forma independiente, sin tener el profesor a su lado continuamente.

Sin embargo, esto no lo puede lograr sin haber sido guiado en sus pasos iniciales, para que el proceso de aprendizaje sea el correcto. Para que sea oportuna, esta guía debe darse en el momento en que el estudiante está desarrollando una solución, que es cuando surgen y se detectan dudas y fallos en la comprensión de las bases teóricas y prácticas de la materia.

Por lo tanto, el curso de Programación 1 requiere de un balance adecuado entre guiar al estudiante en un mundo totalmente nuevo y ajeno para él hasta ese momento, y fomentar su

autonomía en la búsqueda de soluciones. Este objetivo puede alcanzarse, mediante la combinación de clases teóricas², asignación de tareas cortas y programadas, y horas de laboratorio supervisadas. La asignación de tareas promueve la autonomía del estudiante y los laboratorios proveen la guía básica que necesita.

2.2.2. Visión del sujeto educativo

En la Propuesta de Reforma al Plan de Estudios del Programa de Bachillerato en Computación e Informática, vigente desde el 2000 hasta la actualidad, se plantea como objetivo de la carrera *"ofrecerle a la sociedad un profesional integral capaz de desarrollarse de manera eficaz y eficiente en los campos identificados en la propuesta del currículo"* (ECCL, 1999, p.7). Este objetivo nos sugiere un profesional con una formación de calidad, que no sólo alcance los mínimos esperados en su desempeño, sino que sobresalga en aspectos de eficacia y eficiencia.

Esta reforma curricular especifica una serie de actitudes, habilidades, destrezas y conocimientos básicos que debe poseer el profesional graduado de la carrera. Cada curso que conforma el plan y sus contenidos, aporta en la consecución de estos aspectos. No obstante, se destacan las siguientes habilidades y destrezas que los laboratorios de programación podrían contribuir a reforzar:

1. *Aprender por sí mismo (ser autodidacta)*: lejos de crear dependencia del estudiante hacia el profesor, si los laboratorios son bien planificados y combinados con otras estrategias pedagógicas, fortalecen el proceso de "aprender a aprender".
2. *Experimentar, instalar y mantener la tecnología informática nueva, tanto hardware como software*: sesiones de laboratorio de programación guiadas le permiten al estudiante experimentar y explorar el mundo de la programación; pero a diferencia de realizarlo en solitario o con compañeros, lo realiza en forma guiada y significativa. Los laboratorios son una etapa del proceso de exploración, el cual es continuado por la experimentación autónoma.

² En este artículo se entiende por "clase teórica de programación" aquella en la que el profesor explica conceptos, desarrolla ejemplos (en computadora o en pizarra) y los estudiantes desarrollan ejercicios "en papel" (sin computadora).

3. *Plantear una solución computacional óptima para un problema particular. Esto incluye la identificación, análisis, formulación, planteamiento e implementación de la solución:* los laboratorios pueden complementar las clases teóricas y ampliar el tiempo de análisis y discusión de posibles soluciones a problemas.
4. *Capaz de trabajar bajo presión:* en un nivel básico, los laboratorios constituyen sesiones de trabajo limitadas en el tiempo, las cuales introducen al estudiante en el desempeño bajo presión. Esto los prepara en el corto plazo para la realización de los exámenes del curso.

El estudio de perfiles académicos y profesionales, realizado por Prosoftware en el 2003, coincide con las destrezas anteriores, al mencionar las habilidades genéricas consideradas como importantes para los programadores de aplicaciones. Algunas de ellas son: capacidad de pensamiento analítico y lógico, capacidad de identificar, analizar y resolver problemas, y capacidad para trabajar y tomar decisiones bajo presión (Mata, Matarrita y Araya, 2003).

2.2.3. Visión de los procesos de enseñanza y de aprendizaje

El aprendizaje de la programación de computadoras pertenece a lo que se conoce, como saber hacer o saber procedimental. El saber hacer es aquel conocimiento que se refiere a la ejecución de procedimientos, estrategias, técnicas, habilidades, destrezas, métodos, etc. A diferencia del saber qué, que es de tipo declarativo y teórico, el saber procedimental es de tipo práctico (Dede, 2000).

Se ha establecido que un aprendizaje de este tipo ocurre en las siguientes etapas (Díaz-Barriga y Hernández, 2002, p. 54-55):

1. *La apropiación de datos relevantes respecto a la tarea y sus condiciones.* En esta etapa se enfatiza el conocimiento declarativo, proporcionándole al aprendiz la información factual relacionada con los procedimientos a desarrollar, explicándole las propiedades y condiciones para su realización, así como las reglas generales de aplicación.
2. *La actuación o ejecución del procedimiento.* En esta etapa el estudiante procede por prueba y error, mientras el docente lo va corrigiendo mediante sesiones de

práctica con retroalimentación. En esta fase se manejan dos aspectos: el declarativo y el procedimental. Debe culminar con la fijación del procedimiento.

3. *La automatización del procedimiento.* Esto se logra mediante la ejecución continua de los procedimientos en situaciones adecuadas. Una persona que ha automatizado un procedimiento muestra facilidad, ajuste, unidad y ritmo continuo cuando lo ejecuta.
4. *El perfeccionamiento indefinido del procedimiento.* Esta etapa no tiene final. Marca claramente la diferencia entre un experto (el que domina el procedimiento) y el novato (el que se inicia en su aprendizaje).

Al hacer una relación de estas etapas con el curso Programación 1, la primera fase se lleva a cabo durante las clases teóricas, en donde el profesor expone conceptos y ejemplifica los procesos relacionados.

La segunda fase puede iniciarse, dependiendo de la metodología del profesor, en las mismas clases teóricas. En este caso, los estudiantes realizarían ejercicios en papel o en pizarra bajo la supervisión del docente. Esto resulta de utilidad principalmente en la discusión de diseño de programas y algoritmos. Sin embargo, no puede darse por completa esta fase en este punto ya que como se sabe, el traducir el diseño o algoritmo al lenguaje de programación directamente en la computadora debe considerar una mayor cantidad de detalles técnicos. Durante la implementación del programa en la computadora surgen gran cantidad de dudas, y muchas de ellas están relacionadas con una comprensión incompleta de los conceptos teóricos.

Esta segunda fase es muy importante, ya que en la adquisición de una destreza, tal como programar, no sólo es necesario plantearle al alumno el desarrollo ideal de los procesos de programación, también es indispensable confrontarlo con los errores típicos, las rutas erróneas y las alternativas u opciones de aplicación y solución de problemas, cuando éstos se presentan. Un proceso de enseñanza de este tipo consume mucho tiempo, posiblemente más del que se dispone con las lecciones teóricas de un semestre, cuando se deben cubrir todos los temas establecidos en el programa oficial del curso.

Los autores Díaz-Barriga y Hernández (2002, p. 55) comentan que:

Una crítica importante hacia la forma en que habitualmente se enseñan los procedimientos en la escuela es que no se llega más allá de la fase uno, o si acaso se introduce al alumno a la fase dos. Parece que la creencia errónea más arraigada al respecto es que es posible ejecutar un procedimiento simplemente a partir de proporcionar la información 'teórica' o las 'reglas' que nos dicen cómo hacerlo.

Las sesiones de laboratorio complementarias a las clases teóricas de programación, le permiten al estudiante trabajar en contextos de práctica auténticos, con el profesor que supervisa el proceso de automatización del procedimiento e intenta su perfeccionamiento. Además, estas sesiones aumentan la disponibilidad de tiempo para llevar a cabo las actividades pedagógicas necesarias que culminen con un aprendizaje exitoso.

La tercera etapa de automatización del procedimiento, se iniciaría con las mismas sesiones de laboratorio. Sin embargo, su principal desarrollo se da con la realización por cuenta del estudiante de ejercicios, tareas cortas y tareas programadas, de ser necesario, contando con la ayuda del profesor en sus horas de consulta.

La cuarta etapa de perfeccionamiento indefinido, ocurre en el transcurso de los estudios de carrera del estudiante e, incluso, durante su desempeño profesional.

En resumen, las sesiones de laboratorio que se proponen, tratan de fortalecer la relación teoría-práctica propia del aprendizaje de la programación. Cabe destacar que, con estas sesiones, no se busca sustituir las actividades pedagógicas que tradicionalmente se han aplicado en el curso ni el objetivo que con ellas se persigue, tal como las tareas programadas. Sigue siendo vital que el estudiante adquiera costumbre y habilidad de aprender, investigar, analizar y proponer soluciones en forma autónoma³. No obstante, para ello debe garantizarse que se le han brindado las herramientas necesarias. Los laboratorios complementan y, a la vez, cubren un vacío en el proceso de enseñanza y de aprendizaje que hasta ahora se ha seguido.

Desde el enfoque constructivista, la enseñanza de la programación puede basarse en la siguiente estrategia general: el traspaso progresivo del control y responsabilidad en el manejo de la competencia procedimental, mediante la participación guiada y con la asistencia continua, pero paulatinamente decreciente del profesor, la cual ocurre al mismo tiempo que se genera la creciente mejora en el manejo del procedimiento por parte del alumno (Díaz-Barriga y Hernández, 2002). En esta estrategia se destaca la idea del aprendizaje paulatino, lo que, a su vez, requiere de tiempo. Además, sugiere que el número de laboratorios realizados puede ir en disminución, conforme avanza el período lectivo. En todo este proceso de enseñanza y de aprendizaje, el papel del profesor adquiere dos dimensiones: como docente y como investigador.

En el nivel docente, el profesor tiene como responsabilidad ayudar a que el estudiante logre el desarrollo de habilidades y la adquisición de destrezas en la solución de problemas, mediante la programación de computadoras. Para lograrlo, el docente deberá promover un aprendizaje paulatino, ya que la adquisición de destrezas requiere de tiempo.

El diseño y la realización de las sesiones de laboratorio que se proponen para fortalecer este proceso, dependen de la metodología, del criterio y del estilo de enseñanza del profesor que imparte el curso. Por lo tanto, cada profesor deberá convertirse en un investigador dentro del aula, capaz de observar, analizar e innovar el proceso educativo.

Dentro del enfoque de la pedagogía informacional, expuesta en la sección 2.2.1 de la visión de la sociedad, el docente se debe transformar en un "pedagogo investigador", quien debe propiciar aprendizajes significativos en una verdadera mediación entre: 1) la experiencia de los estudiantes; 2) la información existente; y 3) la producción colectiva de nueva información (Picardo, 2002).

Con la observación y el análisis que el profesor realiza durante el laboratorio, el docente puede variar y redirigir el proceso educativo que tenía planeado originalmente. La retroalimentación que obtiene de las sesiones prácticas, permite detectar necesidades específicas de los alumnos y, de esta forma, el profesor podrá mejorar la orientación y el

³ En este artículo, el trabajo autónomo implica la poca participación del docente en el mismo; no excluye el trabajo en grupo con otros estudiantes.

énfasis de sus clases para que el proceso se vuelva más significativo para el estudiante. La observación y el análisis dentro del aula son definitivamente una forma de investigación, que posiblemente ya estemos desarrollando, pero de la cual podemos obtener aún más ventajas.

2.2.4. Visión de la tecnología

Este apartado comprende la visión de la tecnología, como andamiaje en el proceso educativo.

Los 'andamios' son elementos auxiliares externos que brindan un soporte cognitivo y social a quienes son neófitos en un trabajo o un dominio del conocimiento dado, tal como un andamio sostiene a los obreros y los materiales durante la construcción de un edificio. (Dede, 2000, p. 40)

El concepto de tecnología como andamiaje se integra eficazmente dentro de la concepción constructivista de la educación. Desde el punto de vista del constructivismo, los tres elementos fundamentales de toda situación de aprendizaje son: contenidos (qué aprende), procesos (cómo aprende) y condiciones (entorno que facilita el aprendizaje y experiencias del alumno). Se ha concluido que las nuevas tecnologías de la información y la comunicación (NTIC) amplían la capacidad de interacción personal con estos elementos (Picardo, 2002).

El ofrecimiento de sesiones de laboratorio en el curso Programación 1, incorpora la tecnología como andamiaje. El trabajo que desarrolla un estudiante en un laboratorio de programación, constituye un recurso pedagógico de exploración, independientemente de que el objeto de estudio también sea tecnológico. Le permite involucrarse directamente en el momento en que la temática es abordada y puede ser parte activa y productiva en la clase que el profesor dirige. En este sentido, la computadora se convierte en una herramienta de apoyo en el desarrollo del proceso de enseñanza y aprendizaje.

Resulta interesante la opinión que el autor Picardo expresa con respecto a las NTIC, ya que destaca su función como herramienta de investigación del proceso educativo (Picardo, 2002, p.14):

Pero más que un medio didáctico, las NTIC representan un nuevo escenario para comprender el fenómeno educativo, que implica una nueva cultura organizacional y pedagógica.

Un ejemplo de esta funcionalidad que se le puede dar a la tecnología, en el caso concreto de los laboratorios de programación, lo constituye el haber observado y palpado el proceso de razonamiento que aplicaban los estudiantes, durante los laboratorios que se realizaron como parte del proyecto de investigación que motiva esta propuesta.

El poder interactuar con el estudiante, en el momento mismo en que diseña o programa una solución, permitió descubrir algunos detalles acerca de cómo el estudiante interpretó lo explicado por la profesora durante la clase teórica, lo que, a su vez, sacó a la luz deficiencias en la metodología o forma en que se expusieron los conceptos. La detección de fallas en el proceso es el primer paso para un mejoramiento continuo y, en este caso, la incorporación de la tecnología en el aula facilitó la identificación de algunas deficiencias.

En conclusión, la tecnología puede utilizarse como herramienta didáctica en beneficio del estudiante y como herramienta para el profesor en la investigación educativa.

Estrategias metodológicas propuestas

Las estrategias deben considerarse como procedimientos de carácter heurístico y flexible (Díaz-Barriga y Hernández, 2002), por lo que las que se plantean en esta sección constituyen una guía para el docente. El esquema básico que se presenta a continuación, propuesto por Coll y Valls (1992), sirve de base para trazar las estrategias metodológicas del curso Programación 1, tomando en cuenta que implica el aprendizaje de una destreza.

Este esquema identifica tres pasos básicos que ocurren en el lapso entre el desconocimiento del proceso por parte del estudiante hasta su uso autónomo y autorregulado (Díaz-Barriga y Hernández, 2002, p. 260-261):

1. *Presentación de estrategias*: exposición y ejecución del procedimiento por parte del profesor.
2. *Práctica guiada*: ejecución guiada del procedimiento por parte del estudiante y/o compartida con el profesor.

3. *Práctica independiente*: ejecución independiente y autorregulada del procedimiento por parte del alumno.

3.1. Integración de los laboratorios en la dinámica del curso

Reutilizando la terminología de Coll y Valls (1992), se propone dividir el desarrollo del curso Programación 1 en tres espacios metodológicos:

1. *Presentación de estrategias*
2. *Práctica guiada*
3. *Práctica independiente*

La intensidad propuesta de cada espacio en el transcurso del ciclo lectivo, se ilustra en la figura 1.

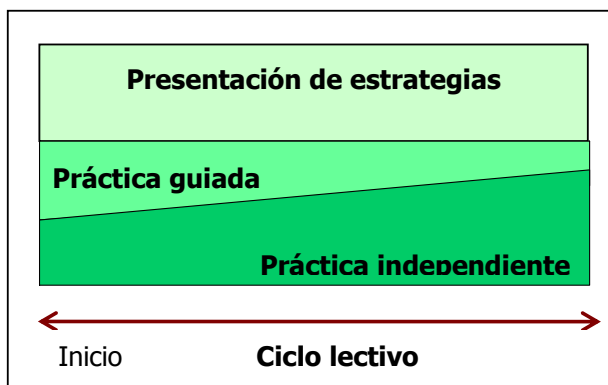


Figura 1. Intensidad de los espacios metodológicos del curso Programación 1

3.1.1. *Presentación de estrategias*

Este espacio lo constituyen las clases teóricas, entendidas en este artículo como aquellas en las que el profesor explica conceptos, desarrolla ejemplos (en computadora o en pizarra) y, en ocasiones, los estudiantes resuelven ejercicios "en papel" (sin computadora).

Este espacio se desarrolla en el horario de clase definido oficialmente para el curso (4 horas semanales divididas en 2 días) y está a cargo del profesor del curso.

Específicamente, las actividades principales por realizar en este espacio son:

1. Presentación de los temas, los conceptos y los contenidos especificados en el programa del curso. Éstos pueden organizarse en las siguientes áreas:
 - Análisis de problemas
 - Diseño de programas
 - Resolución algorítmica de problemas
 - Patrones de programación
 - Sintaxis del lenguaje de programación
2. Ejemplificación de los temas y procedimientos, mediante los cuales el profesor muestra a los estudiantes la aplicación de los conceptos explicados en un problema práctico.
3. Sesiones cortas de ejercicios realizados por los alumnos, para que el estudiante tenga un primer acercamiento con lo expuesto por el profesor. Generalmente, los ejercicios se enmarcan dentro de uno o varios de estos tópicos:
 - Análisis de problemas
 - Diseño de programas
 - Escritura de algoritmos y programas

El espacio de presentación de estrategias es constante en tiempo, esfuerzo e intensidad a lo largo de todo el período lectivo, y constituye el centro alrededor del cual gira el proceso de enseñanza-aprendizaje en el curso.

3.1.2. Práctica guiada

Este espacio metodológico tiene lugar principalmente en las sesiones de laboratorio, las cuales se realizarían en una sala con computadoras, en donde se dispone de una computadora para cada uno o dos estudiantes.

Los laboratorios tienen como objetivo principal que el estudiante ejercite en un contexto real (el computador con el lenguaje de programación) lo expuesto por el profesor, donde este último supervisa el proceso de automatización del procedimiento y guía su mejoramiento.

Se sugiere realizar sesiones semanales de 2 horas, que estén a cargo del profesor del curso con el apoyo del asistente. Dependiendo de la temática y de circunstancias especiales,

algunos de los laboratorios podrían ser dirigidos por el asistente. Sin embargo, la responsabilidad principal recae en el profesor, ya que el asistente no tiene la experiencia docente ni el conocimiento contextual académico necesario para supervisar el proceso de aprendizaje de los estudiantes.

Los laboratorios se llevarían a cabo en un horario adicional al definido formalmente para el curso, disminuyendo su frecuencia conforme avanza el semestre y el estudiante va adquiriendo mayor autonomía y responsabilidad. En este sentido, la asistencia del profesor será continua pero decreciente en el tiempo.

En esta propuesta se sugiere que los laboratorios sean de asistencia voluntaria por parte del estudiante, con base en los siguientes razonamientos:

1. Para que la asistencia sea obligatoria, debería incluirse en el plan de estudio de la carrera un curso formal de "Laboratorio de Programación 1". Dado que la intensidad de este espacio metodológico decrece conforme avanza el semestre, es posible que en el último mes del ciclo lectivo se realicen muy pocas o ninguna sesión de laboratorio. Por lo tanto, no se justificaría un curso formal de laboratorio.
2. Promueve una participación del estudiante voluntaria y por motivación en el mejoramiento de su propio aprendizaje.
3. Constituye una alternativa intermedia entre la no realización de laboratorios que tradicionalmente se daba en el curso e incorporar en la carrera un curso formal de laboratorio de programación. Esto permite que los profesores que lo consideren conveniente adopten esta estrategia y los que no la comparten sigan con su metodología tradicional. A su vez, su implementación optativa mostrará con el tiempo la conveniencia o inconveniencia de esta propuesta.
4. Dado su carácter optativo, le brinda al profesor flexibilidad para definirlos en la forma que considere más favorable para su curso. Por ejemplo, facilita que dos profesores impartan un laboratorio en forma conjunta, lo que le aporta diversidad a los estudiantes.

Aún cuando la realización de los laboratorios sea sugerida para el profesor, es importante ir creando una "cultura metodológica", en donde se considere relevante su realización como parte de las actividades del curso. Es deseable que la implementación de sesiones de práctica guiadas se convierta en una experiencia común entre los profesores que imparten el curso.

El objetivo de cada laboratorio variará, dependiendo de las necesidades de cada grupo y del criterio del profesor. Se pueden identificar los siguientes tipos:

1. Dirigidos al aprendizaje del ambiente de desarrollo (IDE) y sus herramientas. Por ejemplo: utilización básica, manejo de proyectos, depurador, generador de documentación.
2. Dirigidos a ejercitar las estrategias y los conceptos presentados por el profesor en la clase teórica, relativos a un tema de los especificados en el programa oficial del curso.
3. Dirigidos a introducir un tema nuevo complementario que no se encuentra especificado en el programa oficial del curso. Por ejemplo: interfaces de usuario gráficas; manejo de archivos; programación por eventos; ambientes de desarrollo alternativos; otros tipos de aplicaciones (applets o aplicaciones independientes); manejo de excepciones.
4. Dirigidos a realizar práctica y resolución de dudas de temas que presentan una mayor dificultad para los estudiantes.
5. Dirigidos a realizar algún tipo de evaluación para medir el desempeño de los estudiantes.
6. Dirigidos a algún otro aspecto deseable en la formación del estudiante. Por ejemplo: promover el trabajo de los estudiantes en equipo y fomentar el aprendizaje independiente de parte del alumno (autoaprendizaje).

Una misma sesión de laboratorio puede presentar características de varias categorías. Por ejemplo, la realización de un laboratorio para poner en práctica la manipulación básica de arreglos, introduciendo al mismo tiempo el uso del depurador para la detección y corrección de errores.

3.1.3. Práctica independiente

La práctica independiente consiste en la realización por cuenta del estudiante de ejercicios, tareas cortas y tareas programadas, de ser necesario contando con la ayuda del profesor en sus horas de consulta. El objetivo de esta práctica es que el estudiante adquiera autonomía en el análisis de problemas, diseño de una solución y su implementación mediante un programa computacional.

Este espacio metodológico es llamado "independiente" ya que se basa en el trabajo del estudiante, ya sea solo o en equipo con otros compañeros, sin la supervisión constante del docente.

Esta práctica la realiza el estudiante en sus horas de estudio y entre las actividades que abarca se destacan las siguientes:

1. Práctica mediante ejercicios de los temas vistos en clase en cualquiera de las siguientes fases:
 - Uso del ambiente de desarrollo
 - Análisis de problemas
 - Diseño de programas
 - Resolución algorítmica de problemas
 - Patrones de programación
 - Sintaxis del lenguaje de programación
2. Investigación de temas específicos. Por ejemplo:
 - Bibliotecas y comandos especializados del lenguaje de programación
 - Herramientas específicas del ambiente de desarrollo
 - Alternativas de solución a problemas específicos
3. Elaboración de trabajos asignados. Por ejemplo:

- Tareas cortas
- Tareas programadas

4. Consultas al profesor del curso sobre aspectos específicos.

Los tres espacios metodológicos explicados (presentación de estrategias, práctica guiada y práctica independiente) pueden compartir el espacio físico y temporal que habitualmente se dedica a otros espacios. Por ejemplo, en las clases teóricas puede combinarse o alternarse ocasionalmente la presentación de estrategias con la práctica guiada. Similarmente, las sesiones de laboratorio podrían incorporar la presentación de conceptos.

3.2. Sugerencias de implementación de laboratorios

3.2.1. Fases de los laboratorios

Es conveniente que cada laboratorio esté conectado en alguna medida con los otros espacios metodológicos del curso; es decir, con la presentación de estrategias y con la práctica independiente. Para ello se propone que la ejecución de cada laboratorio esté formada de las siguientes fases:

3.2.1.1. Actividades previas

Son aquellas actividades, que el estudiante debe realizar, antes de iniciar el laboratorio. Por ejemplo: estudiar la teoría expuesta y transcribir los ejemplos presentados por el profesor en la clase teórica, investigar alguna biblioteca de clases necesaria para la práctica dirigida, realizar trabajo de implementación previo de algún ejercicio que se desarrollará en el laboratorio.

Esta fase trata de evitar que el estudiante llegue "perdido" a la sesión práctica y que asuma cierta responsabilidad en su ejecución.

3.2.1.2. Actividades del laboratorio

Son las que realmente conforman la sesión práctica guiada. Deberá definirse el o los objetivos del laboratorio y las actividades que deberá realizar el estudiante para alcanzarlos.

Adicionalmente se sugiere incluir una sección de "extensión de ejercicio" para estudiantes que avancen más rápido que el resto de los compañeros. En esta sección, se solicita al alumno adicionar a los ejercicios del laboratorio mayor complejidad o características que le proveen mayor funcionalidad. También pueden especificarse otros ejercicios relacionados con la temática.

Es conveniente que la especificación de los laboratorios sea lo más guiada posible, para disminuir al mínimo las consultas relacionadas con la interpretación de los enunciados. Incluso pueden preverse posibles consultas de los estudiantes e incluir la información necesaria en el documento. Es muy recomendable contar con la ayuda del asistente del curso durante la sesión para la aclaración de dudas.

Se recomienda que la especificación y la publicación de los laboratorios se haga por un medio electrónico, por ejemplo, en una página web. Este formato amplía la ejecución de los laboratorios al eliminar las barreras de tiempo y espacio, de forma que en circunstancias especiales los estudiantes pueden realizar los laboratorios fuera de la sala de computadoras (por ejemplo, desde la casa) y en un horario conveniente.

3.2.1.3. Actividades posteriores

Son las que se realizan una vez finalizada la sesión de laboratorio. Es importante que el estudiante perciba la relevancia del trabajo realizado y que no se constituya en una actividad aislada del curso.

Como actividad posterior fundamental está la finalización por cuenta del estudiante de las actividades que no haya podido concluir durante la sesión. Además, en la siguiente clase teórica el profesor deberá comentar los resultados de la experiencia práctica y su importancia, dentro del desarrollo de los contenidos del curso.

Durante las clases posteriores, el docente puede integrar en los contenidos del curso lo abordado en el laboratorio y destacar la posibilidad de utilización de lo aprendido en los trabajos que se asignen.

Esta conformación en fases de las prácticas guiadas, favorece el papel protagónico del estudiante y evita la dependencia que podría generarse entre el estudiante y el profesor si los laboratorios no se diseñan adecuadamente. Asimismo, promueve un aprendizaje significativo en el estudiante.

3.2.2. Recomendaciones para el docente

Como se mencionó anteriormente, los laboratorios constituyen una etapa faltante actualmente en el proceso de enseñanza-aprendizaje. Por lo tanto, no debe generarse demasiadas expectativas, con respecto a los resultados producto de su implementación. Más bien, buscan mejorar la calidad de la enseñanza y facilitar el proceso de aprendizaje de los estudiantes, dentro de un contexto totalmente educativo, como lo es la universidad.

Es claro que son realmente útiles para aquellos estudiantes que verdaderamente deseen aprovecharlos. Esto queda reflejado en el interés de los estudiantes más destacados, quienes se muestran deseosos de aprender más allá de lo establecido en el programa oficial del curso. Para los estudiantes esforzados, constituyen un recurso muy valioso para fortalecer el desarrollo de sus destrezas.

Como es de esperarse, en el caso de estudiantes poco interesados, los laboratorios no harán ninguna diferencia. Por lo que se recomienda que, en su mayoría, las sesiones no sean de práctica masiva de ejercicios y resolución de dudas, ya que se corre el peligro de fomentar la sustitución de la práctica independiente que los estudiantes están obligados a realizar.

Para una implementación exitosa de los laboratorios, es indispensable que el docente observe y analice el desarrollo de la experiencia. Esto le permitirá, entre otras cosas:

- Conocer el proceso lógico y de razonamiento aplicado por los estudiantes en la resolución de problemas.
- Determinar los aspectos o temas que más se le dificultan.

- Identificar las características académicas de sus estudiantes, permitiendo, por ejemplo, que los estudiantes demuestren suficiencia en el curso.
- Adaptar el curso según las necesidades de sus estudiantes.
- Autoevaluar su papel como docente, ya que verá reflejado en el trabajo de los estudiantes el grado de asimilación de los temas.

Finalmente, se recomienda que la generación de material para los laboratorios se realice tomando en cuenta su posible reutilización y mejoramiento para otros semestres. Además, puede consultarse el documento "Informe de resultados de los laboratorios de Programación 1" (Murillo, 2004) para un mayor detalle del estudio del caso que motiva esta propuesta.

3.2.3. Recursos metodológicos adicionales

Como complemento de los laboratorios y de las actividades realizadas en el curso por los docentes, se sugieren los siguientes recursos instruccionales que se emplean en los procesos de enseñanza-aprendizaje de tipo procedimental (Díaz-Barriga y Hernández, 2002, p. 56), tal como lo es el área de programación:

- Repetición y ejercitación reflexiva.
- Observación crítica.
- Imitación de modelos apropiados.
- Retroalimentación oportuna, pertinente y profunda.
- Establecimiento del sentido de las tareas y del proceso en su conjunto, mediante la evocación de conocimientos y experiencias previos.
- Verbalización mientras se aprende.
- Actividad intensa del alumno, centrada en condiciones auténticas, lo más naturales y cercanas a las condiciones reales donde se aplica lo aprendido.
- Fomento de la metacognición: conocimiento, control y análisis de los propios comportamientos.

El objetivo de mencionar estas técnicas no es profundizar en ellas, simplemente que sirvan de ejemplo de que el proceso de enseñanza-aprendizaje debe ser sometido a análisis y a mejoramiento continuo, y que muchas de las opciones metodológicas ya han sido estudiadas y podemos sacar provecho de ellas.

Conclusiones

Así como existen formalmente "las horas de consulta de los profesores" para dar apoyo a los estudiantes que así lo requieran (esto incluye no sólo a aquellos con problemas académicos sino a cualquiera que desee un apoyo extra de parte del profesor), los laboratorios constituyen un espacio pedagógico abierto para aquellos estudiantes que deseen sacar provecho de ellos.

La implementación de prácticas guiadas no requiere alterar ni cuestionar los contenidos temáticos ni el enfoque actual del curso. Simplemente, la realización de laboratorios opcionales (en su mayoría no remediales), aporta calidad a la enseñanza de la programación, promoviendo en el estudiante ir más allá de la obligatoriedad de la enseñanza y buscar, por iniciativa propia y no del docente, ampliar o fortalecer sus conocimientos.

En definitiva, los laboratorios promueven el aprendizaje paulatino que lleva a un desarrollo de habilidades y destrezas más sólido. Esto es indispensable para los cursos posteriores de la carrera y, sobre todo, para un buen inicio en la autonomía del estudiante en el aprendizaje continuo.

Finalmente, se plantean tres limitaciones relacionadas con la implementación de laboratorios y se comentan algunas recomendaciones:

1. Recurso tecnológico

La realización de laboratorios en todos los grupos del curso, requerirá la utilización del equipo computacional disponible para los estudiantes. Esto limita en tiempo de uso de las computadoras al resto de estudiantes de la Unidad Académica que no están cursando Programación 1. Esta limitación puede ser significativa, si se toma en cuenta que en un semestre puede haber hasta 7 grupos del curso.

Tomando como base la intensidad de los espacios metodológicos sugerida en la figura 1, se recomienda que los laboratorios se realicen en forma frecuente al inicio del semestre, que es cuando el estudiante requiere de una mayor guía de parte del profesor, y disminuyan al final del período. Esto coincide favorablemente con la demanda creciente durante el semestre, de uso de los laboratorios de parte de la población estudiantil en general, conforme se intensifica su carga de trabajo.

2. Recurso humano

La realización de laboratorios requiere de una cantidad significativa de horas de trabajo de docentes y de asistentes que apoyen durante las sesiones. En el caso de otros cursos que realizan sesiones prácticas, estas horas son parte de la carga docente del profesor, ya que constituyen cursos formales dentro del plan de carrera.

Al implementarse opcionalmente los laboratorios de programación, esta opción no es viable. Para contar con la disponibilidad de los docentes, se sugiere motivar y concientizar acerca de las ventajas de realizar prácticas guiadas dentro del proceso de enseñanza y de aprendizaje. Además, las horas de laboratorio pueden cubrir parcialmente las horas de consulta que el profesor ofrece a los estudiantes.

3. Realización de otros casos de estudio

Es recomendable que se realice una experiencia similar a la realizada en el proyecto de investigación que motiva esta propuesta, que permita estudiar otro caso de implementación de laboratorios. Esto permitiría profundizar en las ventajas y las desventajas de esta metodología de enseñanza, analizando una experiencia en circunstancias diferentes (otra población estudiantil, otros docentes, etc.).

Como consideración final, las horas de laboratorios de programación, si se implementan adecuadamente, en lugar de generar dependencia de los alumnos hacia el profesor, promueven un aprendizaje real que contempla todas las etapas de enseñanza-aprendizaje por las que debe pasar un estudiante en un campo como la programación de computadoras. Esto ayudará a que, al final del semestre, el alumno haya desarrollado en forma sólida y exitosa sus destrezas, y esté totalmente preparado para enfrentar los siguientes cursos de la carrera.

Agradecimientos

Agradezco los valiosos aportes de la Dra. Jacqueline García y del Dr. Carlos Vargas, profesores investigadores de la Universidad de Costa Rica que colaboraron con la investigación dentro de la cual se enmarca este artículo. Igualmente se agradecen los comentarios realizados por varios profesores de la Escuela de Ciencias de la Computación e

Informática y el apoyo del Instituto de Investigación en Educación, ambas unidades de la UCR.

Referencias

- Casasola, Edgar. (2004a). **Curso de Programación I: programación basada en objetos**. San José, C.R.: Cámara Costarricense de Tecnologías de Información y Comunicación (CAMTIC), Proyecto Prosoftware.
- Casasola, Edgar. (2004b). **Curso de Programación II: programación orientada a objetos**. San José, C.R.: Cámara Costarricense de Tecnologías de Información y Comunicación (CAMTIC), Proyecto Prosoftware.
- Comisión de Planes y Programas. (2004). **Primer Informe de Labores**. San José, C.R.: Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica.
- Dede, Chris (comp.). (2000). **Aprendiendo con tecnología**. Argentina: Editorial Paidós SAICF.
- Díaz-Barriga, Frida y Hernández, Gerardo. (2002). **Estrategias Docentes para una Aprendizaje Significativo: una interpretación constructivista**. 2^a. ed. México: McGraw-Hill Interamericana.
- Escuela de Ciencias de la Computación e Informática (ECCI). (1999). **Propuesta de Reforma al Plan de Estudios del Programa de Bachillerato en Computación e Informática**. San José, C.R.: Facultad de Ingeniería, Universidad de Costa Rica.
- Mata, Francisco; Matarrita, Rosaura y Araya, Eduardo. (2003). **Estudio para el Fortalecimiento de los Centros de Enseñanza en Computación e Informática y la Actualización Curricular**. San José, C.R.: Programa de Apoyo a la Competitividad del Sector Software, BID-PROCOMER-CAPROSOFT-FUNCENAT.
- Murillo, Maureen. (2004). **Informe: Resultados de los Laboratorios de Programación 1**. San José, C.R.: Proyecto de investigación "Incorporación de un laboratorio de programación en el curso de Programación 1 de la Escuela de Ciencias de la Computación e Informática, como una experiencia de enseñanza y aprendizaje", N° 326-A3-119, Escuela de Ciencias de la Computación e Informática, Instituto de Investigación en Educación, Universidad de Costa Rica.
- Picardo Joao, Oscar. (2002). Pedagogía Informacional: Enseñar a aprender en la Sociedad del Conocimiento. **Revista Iberoamericana de Ciencia, Tecnología, Sociedad e Innovación**. (3). Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura.
- Rodríguez, Ma. Elena. (2004). **Informe del estudiadero de programación para Computación en el I ciclo del 2003**. San José, C.R.: CASE de Ingeniería, Oficina de Orientación, Universidad de Costa Rica.