

Enfoques y herramientas en la enseñanza de un primer curso de computación (CS1)

De Giusti Armando¹, Madoz Cristina², Gorga Gladys³, Feierherd Guillermo⁴, Depetris Beatriz⁵

*Instituto de Investigación en Informática LIDI*⁶
Facultad de Informática. UNLP.

*Grupo de Investigación en Tecnologías Informáticas Aplicadas (GITIA)*⁷
Facultad de Ingeniería – Sede Ushuaia - UNPSJB

Resumen

Se presenta un análisis del enfoque clásico de los cursos CS1 (básicamente cursos introductorios de expresión de algoritmos e iniciación a la programación) y algunas herramientas utilizadas en su desarrollo.

En particular se discuten diferentes experiencias universitarias, analizando las ventajas del uso de herramientas visuales.

Por último se hace un breve análisis crítico de Visual Da Vinci (VDV), un ambiente desarrollado en la UNLP que se ha utilizado sistemáticamente durante los últimos 10 años en los cursos pre-universitarios y en las primeras etapas del curso de Algorítmica y Programación de varias Universidades del país.

En las conclusiones se marca la evolución y objetivos futuros para VDV, según el grupo de I/D de las Universidades de La Plata y Patagonia San Juan Bosco que trabajan en el tema.

Palabras Clave

Expresión de algoritmos. Introducción a la programación. Programación Visual. Ambientes de enseñanza-aprendizaje. Visual Da Vinci.

¹ Investigador Principal CONICET. Prof. Titular DE. III-LIDI. Fac de Informática. UNLP. degiusti@lidi.info.unlp.edu.ar

² Profesora Adjunta con DE III-LIDI – Fac. de Informática. UNLP. cmadoz@lidi.info.unlp.edu.ar

³ Profesora Adjunta con SD - Fac. de Informática. UNLP. ggorga@lidi.info.unlp.edu.ar

⁴ Profesor Asociado con SD – Fac. de Ingeniería. UNPSJB. feierherdge@ciudad.com.ar

⁵ Profesora Asociada con DE - Fac. de Ingeniería. UNPSJB. depetris@ciudad.com.ar

⁶ III-LIDI miembro del Instituto de Investigación en Ciencia y Tecnología Informática (IICyTI) - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina. TE/Fax +(54) (221) 422-7707. <http://lidi.info.unlp.edu.ar>

⁷ Grupo de Investigación en Tecnologías Informáticas Aplicadas (GITIA) – Facultad de Ingeniería de la Universidad Nacional de La Plata – Sede Ushuaia – Darwin y Canga (9410) – Ushuaia.

Introducción

Durante los últimos veinte años los primeros cursos de Algorítmica y Programación (CS1) han sido motivo de preocupación en la mayoría de las universidades que ofrecen carreras de Ciencias de la Computación/Informática en todo el mundo. [Gao00] [Lis03]

Si se trata de hacer un análisis de los contenidos curriculares en estos cursos, se puede afirmar que en un principio, el énfasis estuvo puesto básicamente en el aprendizaje de lenguajes de programación. [ACM68] [ACM78] [ACM84] [ACM91]

En los años sucesivos se fueron adaptando estos contenidos tratando de poner el acento en el análisis y diseño de algoritmos y su expresión como “programas” en diferentes niveles de lenguaje. [Coo03] [Bry01] [ACM01]

Estos cambios condujeron a que actualmente, los cursos de CS1 incluyan en su currícula términos relacionados con modelización y abstracción del mundo real, diseño top down, refinamientos sucesivos, tipos de datos abstractos, diseño de patrones, invariantes algorítmicos, corrección y eficiencia de algoritmos. [Gin00]

También es de hacer notar el importante auge y crecimiento de las tecnologías de comunicación e información y su utilización en los procesos educativos. En su análisis de “Teaching and Technology” Lee [Lee01] dice que nuestras Universidades recién ahora han comenzado a comprender que:

<i>What we hear, we forget</i>	(a)	<i>Lo que oímos, lo olvidamos</i>
<i>What we see, we remember</i>	(b)	<i>Lo que vemos, lo recordamos</i>
<i>What we do, we learn</i>	(c)	<i>Lo que hacemos, lo aprendemos</i>

Una contrapartida importante del fenómeno comunicacional de nuestros días es que debemos distinguir cuidadosamente entre *información y conocimiento* [Cab00]. Muchas veces los recursos tecnológicos facilitan el acceso a la información, pero el objetivo educativo en el nivel universitario debe ser la creación de conocimiento, es decir la formación del alumno en la ciencia informática.

Como consecuencia de todo lo expresado, los docentes encargados de estos cursos se han visto obligados a realizar tareas de análisis y estudio de distintos paradigmas, enfoques y herramientas, de modo de encontrar propuestas que mejoren el proceso de enseñanza y aprendizaje de los contenidos curriculares que se desea que el alumno incorpore. [Dan01] [Pro00]

El aporte de este trabajo trata de ser una reflexión sobre los enfoques y herramientas utilizados en los cursos introductorios de algorítmica y programación, poniendo énfasis en algunas experiencias desarrolladas en Universidades y en particular en la utilización de Visual Da Vinci como ambiente visual para enseñanza-aprendizaje de conceptos iniciales de programación.

Objetivos de un primer curso CS1

Se entiende por objetivo lo que se quiere lograr, es decir, un estado al cual se desea llegar. Todo objetivo tiene tres elementos: intención, medida y plazo [Cat99].

En nuestro caso el “objetivo” es la transformación (entendiendo “transformación” como la incorporación de fundamentos, conocimientos metodológicos e instrumentos concretos) de las capacidades del alumno en relación con la resolución de problemas mediante su modelización y expresión algorítmica correcta.

La formulación de los objetivos sirve para:

- Fijar la situación actual: el hecho de determinar un estado final a *alcanzar*, obliga a fijar una situación actual .
- Determinar el estado final a alcanzar por medio de evaluaciones.
- Determinar las estrategias a emplear: dado que se tiene una situación actual y un estado final.
- Medir los resultados: mediante evaluaciones de diferente alcance.

En particular, los objetivos de un primer curso de computación en la UNLP y en la UNPSJB comprenden básicamente temas relacionados con la Algorítmica y la Programación (CS1).

Los mismos se pueden sintetizar en los siguientes cuatro puntos:

- ❖ Analizar problemas resolubles con computadora, poniendo énfasis en la modelización, abstracción de funciones y en la modularización de los mismos. Obtener una expresión sintética y precisa de los problemas, con una documentación de una metodología de trabajo realizada por el alumno.
- ❖ Estudio, expresión simbólica, implementación y evaluación de algoritmos, orientando los mismos a la resolución de las partes en que se descomponen los problemas.
- ❖ Introducción de las nociones de estructuras de datos, tipos de datos, y abstracción de datos.
- ❖ Combinar los elementos mencionados anteriormente a fin de que el alumno complete el ciclo del problema a su solución con computadora, con un método de refinamientos sucesivos en el que se traten simultáneamente las estructuras de control y los datos y en el que se introduzcan los conceptos de corrección y de eficiencia.

Enfoques de un curso CS1

Normalmente los enfoques de un primer curso de algorítmica y programación se han centrado en la expresión de algoritmos, utilizando diferentes paradigmas y dentro de cada uno de ellos diferentes herramientas (desde expresiones formales, lenguaje natural o lenguajes de programación reales o simbólicos) [Fag00] [Gai98] [Kae01] [Pat81]

En este trabajo no insistiremos en la discusión de *cual es el primer paradigma a enseñar en un curso de programación*, sino en algunas herramientas utilizadas en el ámbito universitario y en particular en la UNLP y la UNPSJB. Haremos hincapié en la reflexión de Lee citada previamente para defender un enfoque centrado en la utilización de herramientas visuales (What we see...) y en la ejecución efectiva de algoritmos “verificables” por el alumno (What we do...). De todos modos, nuestro análisis y conclusiones son el resultado de una larga experiencia, que no descarta otros enfoques y nuevas herramientas metodológicas y tecnológicas que pueden emplearse con los mismos fines.

Formación inicial y expectativas de los alumnos que llegan a un curso CS1

Como ya hemos dicho la enseñanza de la programación tiene como objetivo desarrollar la capacidad para la resolución de problemas y la expresión de las soluciones (algoritmos) mediante alguna herramienta de expresión simbólica, que puede ser un lenguaje concreto de programación. Esto requiere que el alumno alcance alguna forma de representación mental del mismo, lo convierta en un modelo computable, analice alternativas de solución y decida la secuencia de ejecución más eficaz para cumplir una función específica del “programa-solución”.

Los alumnos que realizan un primer curso CS1 en general presentan algunas características que conllevan dificultades en el aprendizaje y concreción de los objetivos definidos previamente:

- falta de entrenamiento en pensar y expresar ideas.
- dificultad en la comprensión y asimilación de contenidos.
- dificultad de aprendizaje de temas básicos que impliquen algún grado de abstracción.
- falta de orientación vocacional adecuada.
- gran disparidad de conocimientos y formación previa.
- escasa valoración por el trabajo sistemático.
- cierta metodología de “prueba y error” asociada con el empleo de computadoras.

Normalmente estos alumnos no cuentan con conocimientos básicos de la Ciencia Informática y tienen algún grado de información dada por la escuela, la familia, grupos de pares, medios de comunicación, cursos voluntarios, etc. Todos estos aspectos conducen a formación inicial muy diversa por parte del alumnado. Por otra parte, también se deben considerar las expectativas que traen estos alumnos. Así como la formación inicial, las expectativas resultan muy diversas. Entre otras se pueden mencionar el interés personal y la alternativa de una rápida salida laboral [Gon02].

Las razones mencionadas relacionadas tanto con la formación inicial como con las expectativas de los alumnos, generan un contexto que hace muy difícil alcanzar los objetivos propuestos en la currícula, ya que los contenidos curriculares exigen un determinado tiempo para adquirir y comprender conocimientos básicos para usarlos y recrearlos adecuadamente. En la mayoría de los casos, este tiempo no coincide con el tiempo en el que el alumno espera cumplir con sus expectativas. [Inf94] [Inf95]

Por otra parte las aptitudes requeridas para tener éxito en una carrera universitaria en Informática que se desprenden de diversos estudios [Inf02] [Inf03] no siempre son las que el alumno asocia con su elección de “estudiar Informática”.

En este contexto los cursos CS1 de nuestras Universidades suelen ser una especie de “filtro” de aptitudes y vocación para el estudio de una carrera de Informática. Sería muy deseable que este planteo de análisis y conocimiento por parte del alumno de sus posibilidades de tener éxito en la carrera universitaria se instale en etapas más tempranas, pero esto claramente requiere una articulación y transformación de la Escuela Media. [UNLP01]

Herramientas Visuales en el aprendizaje de la programación

Los diferentes paradigmas utilizados históricamente se han combinado con la gran cantidad de lenguajes de programación disponibles [Ghe91] para ser utilizados en cursos CS1. Si bien el paradigma imperativo y la programación estructurada son los enfoques más tradicionales, hoy hay numerosas experiencias (la mayoría de ellas centradas en la programación orientada a objetos) que cambian el instrumento a utilizar en los cursos CS1, aunque mantienen los objetivos citados previamente.

De todos modos, un aspecto común es que los métodos “clásicos” parten de lenguajes textuales por naturaleza. En otras palabras, todo algoritmo expresado en un lenguaje de programación textual está compuesto de frases formadas por cadenas lineales de caracteres, los cuales son los símbolos de un alfabeto determinado.

Esta linealidad puede verse como una limitación de los lenguajes textuales, y se debe en gran medida a que la primera forma de comunicación entre computadoras y humanos que se pensó, estaba muy ligada a la forma de comunicación entre humanos. Es decir, a través de un lenguaje compuesto por frases lineales.

Por otro lado, refleja las limitaciones propias de los dispositivos de entrada/salida disponibles (los cuales, seguramente fueron diseñados bajo las mismas influencias).

Además, en general, es comparativamente más sencillo verificar la sintáxis de una notación textual.

Ahora volveremos por un momento a la reflexión de Lee, para discutir las ventajas de las herramientas visuales en el desarrollo de un curso CS1.

Hay una gran cantidad de definiciones acerca de qué es un lenguaje visual. Algunos autores hablan de lenguajes visuales, otros de programación visual, y aún otros de lenguajes de programación visual. [Cha90] [Gol90b] [Mye90] [Pan93]

Si bien todos se refieren más o menos al mismo concepto, hay quienes prefieren hacer notar algunas pequeñas diferencias entre una denominación y otra.

La programación visual es comúnmente definida como el uso de expresiones visuales (tales como gráficos, dibujos, animaciones o íconos) en el proceso de programación. Esas expresiones visuales pueden ser usadas en ambientes de programación como interfaces gráficas para lenguajes de programación textuales; ellos pueden ser usados para formar la sintaxis de nuevos lenguajes de programación visual conduciendo a nuevos paradigmas tales como la programación por demostración; o pueden ser utilizados en la representación gráfica del comportamiento o estructura de un programa [Mci92a].

Hay muchos argumentos en favor de la programación visual. Generalmente están centrados en el hecho de que los humanos procesan, por su naturaleza, más fácil y rápidamente las imágenes que el texto. Es decir, adquieren más información en menos tiempo descubriendo las relaciones gráficas de una imagen que leyendo las letras de un texto [Rae85].

Algunos de esos argumentos son enumerados a continuación:

- ❖ El texto es estrictamente secuencial, mientras que las imágenes permiten acceso no secuencial a cualquier contenido de interés para el aprendiente.
- ❖ Las imágenes permiten una visión global o detallada indistintamente, dependiendo de lo que espere encontrar el observador.
- ❖ El sistema sensorial humano está "diseñado" para el procesamiento de imágenes en tiempo real. Por esa razón, las imágenes pueden ser accedidas y decodificadas más rápidamente.
- ❖ El texto es unidimensional por naturaleza, mientras que las imágenes son multidimensionales.
- ❖ Las imágenes proveen un lenguaje mucho más rico a través de sus propiedades visuales, como los colores, las formas, los tamaños. Esto conduce a una codificación de la información significativamente más compacta que la del texto.
- ❖ Las imágenes pueden capturar más fácilmente una idea abstracta.
- ❖ El texto es una secuencia de palabras y símbolos de puntuación. A su vez, cada palabra es una secuencia de letras. Y cada letra no es otra cosa que una pequeña imagen que representa sencillamente a un único símbolo de todo el texto.

Es común el uso de imágenes o diagramas en la especificación de algoritmos y estructuras de datos que un programador o analista debe comunicar a otros.

Todos estos elementos han conducido a una utilización creciente de herramientas visuales interactivas en el proceso de enseñanza y aprendizaje de los cursos CS1. Algunas experiencias se analizan posteriormente en este artículo.

Estrategias para alcanzar los objetivos considerando el contexto de los alumnos

Dado que se abre una brecha profunda entre los objetivos propuestos para el curso y las expectativas y posibilidades de los alumnos, resulta necesario estudiar diferentes estrategias que favorezcan el proceso educativo.

La problemática mencionada está generalizada en las distintas Universidades con carreras en Informática. Dicha situación nos lleva a remitirnos a lo expresado por Gimeno Sacristán en su artículo "Maestros, formación práctica y transformación escolar": "... *cuando se produce un fracaso escolar, este debe verse como una inadecuación de los alumnos a las condiciones escolares, o a una cultura o a unas exigencias impuestas que ellos no pueden superar*". Cualquiera sea la causa de esa inadecuación, los docentes deben buscar distintas estrategias para poder solucionar estos inconvenientes, entre las que cabe el uso de recursos atractivos y de interés para los alumnos y así evitar la *alta deserción* que se da en los primeros cursos de algorítmica y programación.

Se puede considerar que las estrategias de aprendizaje son procedimientos que presentan rasgos propios tales como aplicación controlada (es decir, no automática), están compuestas de otros elementos más simples que constituyen técnicas o destrezas y suponen un uso selectivo de los propios recursos y capacidades disponibles. Su uso eficaz depende en buena medida del dominio de las técnicas que la componen.

Cuando se opta por una estrategia, se debe disponer de recursos alternativos entre los cuales se decide utilizar, en función de las demandas de las tareas del proceso educativo en cuestión, aquel que se considera más apto. Sin una variedad de recursos no es posible actuar estratégicamente.

Uno de los aspectos a tener en cuenta está asociado con las estrategias planteadas para la presentación de los materiales didácticos. Estos materiales en general, se pueden definir como el conjunto de recursos que intervienen en el acto didáctico, facilitando los procesos de enseñanza y aprendizaje. Sus fines centrales persiguen facilitar la comunicación entre el docente y el estudiante para favorecer a través de la intuición y el razonamiento un acercamiento comprensivo de las ideas a través de los sentidos. [Eis92]

En el proceso educativo, dada las características de las carreras de informática y el perfil de los alumnos inscriptos se considera adecuado brindar los contenidos de un primer curso de algorítmica y programación a través de estrategias relacionadas con materiales multimediales, materiales hipermediales, herramientas visuales, software educativo, entre otras, que nos ayuden a lograr el objetivo principal que es que el alumno logre construir estructuras mentales cada vez más complejas y que desarrolle su poder de abstracción

Algunas experiencias y herramientas

El diseño de metodologías y el desarrollo de herramientas que permitan incorporar la Informática en el proceso de enseñanza aprendizaje se han constituido en temas de investigación y aplicación permanente dentro de las Ciencias de la Educación y de la Computación, tal como surge de un elemental análisis de la actividad de ambos campos.

La Informática en general, y los recursos tecnológicos como los sistemas distribuidos, las redes y la multimedia, ofrecen alternativas y herramientas para favorecer los procesos de aprendizaje, autoaprendizaje, aprendizaje a distancia con guía docente y aprendizaje cooperativo a distancia. Por otra parte la utilización misma de la tecnología informática puede dar un índice indirecto de las potencialidades e intereses del “usuario” por la misma.

Numerosas experiencias destacan el potencial educativo de este enfoque, tanto en cuanto a la posibilidad de facilitar los procesos de aprendizaje, adecuándolos a las necesidades y a las potencialidades individuales, como para servir, en particular, a la evaluación y autoevaluación de aptitudes e intereses por la disciplina Informática.

Para alcanzar los objetivos propuestos por CS1 resulta necesario realizar un análisis exhaustivo de algunas herramientas existentes, utilizadas en el ámbito académico.

A continuación se presentan alternativas para introducir los conceptos de un primer curso de Algorítmica y Programación.

En cuánto a los materiales multimediales e hipermediales, dado que ha tenido un auge muy importante en los últimos años, se considera de gran utilidad para la motivación, la interacción y la definición vocacional de los alumnos. [Bur94] [Sha92] [Ber94]

Sistemas Multimediales e Hipermediales

Los aspectos más interesantes que se pueden destacar de los sistemas multimediales e hipermediales desde el punto de vista educativo están relacionados con la integración *natural* de múltiples medios estimulando el *interés* del alumno, con el tratamiento de información *no secuencial*, con la *adaptación* de las capacidades del alumno, con la posibilidad de fuerte *interacción* con las herramientas, con facilidades para la actividad *creativa y la combinación de múltiples medios*.

Una de las alternativas está relacionada con la utilización de Mapas conceptuales Hipermediales.

Los mapas conceptuales (MC) representan un desarrollo muy interesante correspondiente al área de Representación de Conocimiento en sistemas de software relacionados con la educación. En la búsqueda de soluciones a los problemas de aprendizaje, se ha encontrado en los MC una potente herramienta para lograr aprendizajes significativos. La aplicación de tecnología informática hipermedial facilita la construcción de los MC. Principalmente favorece el trabajo de jerarquización de conceptos, enriqueciendo de esta manera su valor pedagógico. En base a esto se han creado los Mapas Conceptuales Hipermediales (MCH), una metodología de desarrollo y una plataforma específica que permite la creación, mantenimiento, lectura e interconexión de estos mapas.[Señ99]

Existe un proyecto desarrollado en la UNPSJB que tiene como objetivo presentar una herramienta computacional de enseñanza y aprendizaje que permite navegar entre distintos mapas conceptuales. Estos representan las estructuras de datos que se estudian en las asignaturas de los dos primeros años de carreras en Informática.

Esta posibilidad de recorrer y conectar las estructuras posibilita la construcción del conocimiento global de las mismas, lo que permite una visión de conjunto entre ellas y al mismo tiempo resalta las particularidades de cada estructura [Ouv02]

Herramientas visuales

En cuánto a las herramientas visuales, se espera que las mismas permitan escribir soluciones pensada por los alumnos para un problema dado y *visualizar la ejecución* de las soluciones, de manera tal que puedan verificar la corrección de las mismas.

La UNS utiliza en sus primer curso un programa llamado Cubik , que es un editor que permite el diseño de Algoritmos Estructurados, y provee un traductor que toma los algoritmos editados y los transforma en un programa Pascal.

Existen también alternativas que están relacionadas con el *uso de robots*, como la de Karel el Robot, desarrollado por Richard Pattis y conocido/citado por la mayoría de los docentes de cursos de Algorítmica y Programación, cuyo enfoque ha sido incorporado como una parte vital en la curricula de CS1.[Pat81][Bec01]

Esta experiencia ha favorecido el desarrollo de cursos atractivos tanto para los docentes como para los estudiantes y donde los estudiantes han entendido los conceptos fundamentales integrando las ideas de “*ver y hacer*”.

Otra experiencia que se puede mencionar es la presentada por Barry Fagin que utiliza Robótica basada en Ada para la enseñanza de las Ciencias de la Computación [Fag00]. Esta propuesta combina Robótica y Ada para la enseñanza de los conceptos básicos de computación a través de la presentación de numerosos ejemplos.

Otra alternativa está relacionada con la **visualización de software**. Este recurso utiliza gráficos de computadora y de animación interactiva para ayudar a ilustrar y presentar programas, procesos y algoritmos. Se basa en el uso de diseños gráficos, de animación, de sonido, de video que tienen como característica sobresaliente la interacción entre el usuario y la computadora, apuntando a una mayor comprensión y a un uso efectivo del software.

La propuesta presentada por la UNS, plantea el diseño de un sistema para la visualización de algoritmos de búsqueda y modificaciones de distintas estructuras de datos. El mismo ha sido pensado para trabajar sobre las siguientes estructuras: tabla de hash abierta, árboles binario de búsqueda, AVL, 2-3 y parcialmente ordenados [Mor02].

La Facultad de Informática de la UNLP cuenta con una herramienta visual incorporada al curso desde hace algunos años, que es el ambiente Visual Da Vinci, que puede ser utilizada tanto por alumnos como por docentes sin conocimientos previos de informática. La misma tiene como objetivo facilitar la introducción de los conceptos de tipos de datos simples y las estructuras de control del paradigma procedural.[Cha97] . Cabe aclarar que la misma herramienta ha sido introducida a cursos iniciales de enseñanza de la programación en otras universidades, al respecto desde hace tres años se ha implementado en un curso introductorio en la Universidad Nacional de la Patagonia San Juan Bosco en la Sede Ushuaia.

Por último pueden mencionarse productos de software educativo genéricos que se adaptan o personalizan para un curso CS1. Este enfoque se puede observar desde tres puntos de vista diferentes: como tutores (enseñanza asistida por computadora) como aprendices y como herramienta. [Sch97]. Existen numerosos productos de software educativos disponibles en Internet que resultan de utilidad para introducir los distintos conceptos, aunque en la mayoría de los casos se acercan más a la *información que al conocimiento*. [Mar03]

La utilización de Visual Da Vinci. Un análisis crítico.

Como ya se mencionó anteriormente las **herramientas visuales**, permiten escribir soluciones pensadas por los alumnos para un problema dado y *visualizar la ejecución* de las mismas, de modo de realizar una verificación experimental.

En particular, las herramientas visuales que emplean el **uso de robots** para impartir contenidos de CS1 resulta muy atractivo y beneficioso para los alumnos dado que:

- ✓ Muchos problemas pueden ser enunciados utilizando una imagen que representa la situación inicial, otra que representa la situación final y algunas líneas de texto. Esto lleva

en general a que los alumnos tengan pocas dudas respecto de la especificación del problema.

- ✓ La animación de la solución propuesta proporciona un *feedback* visual que permite medir la corrección de los algoritmos planteados. Si el programa del alumno no da como resultado la misma imagen que la enunciación del problema, se puede afirmar que debe haber un error de programación (por supuesto, el hecho de que parece igual no significa que se haya hecho correctamente).
- ✓ Los alumnos pueden darse cuenta donde el programa ha producido un error, por ejemplo, observando simplemente la animación. Debido a que el cerebro humano está altamente optimizado para procesar un *input* visual, esto resulta más fácil y más rápido que, se podría decir, *escanear* una lista de números a partir de un programa en la pantalla.

Estos aspectos mencionados anteriormente determinan que el uso de robots resulta atractivo y motivador tanto para los docentes como para los alumnos en la resolución de problemas planteados en un curso CS1. Es fundamental la simplicidad de la clase de problemas a tratar y la relación directa con las estructuras de control básicas, para poder establecer recorridos.

En nuestra Facultad, los alumnos tienen un primer curso de algoritmo y programación donde se utiliza la herramienta Visual Da Vinci que ha sido largamente expuesta en publicaciones anteriores. [Deg01]

Citaremos a continuación brevemente las ventajas y desventajas de la utilización de este ambiente:

- ✓ El alumno siente una gran motivación por resolver problemas concretos que se relacionan con la aplicación directa de los conceptos teóricos.
- El alumno siente una atracción por resolver los problemas por “prueba y error” poniendo la verificación experimental antes que el análisis cuidadoso del *que y el como*.
- ✓ Es muy natural pasar de un ambiente limitado (por la clase de problemas a tratar y las estructuras de datos que se manejan) a un lenguaje de programación “real” tipo Pascal.
- La clase de problemas a tratar no exige demasiada abstracción.
- ✓ Resulta natural trabajar los conceptos de modularización y derivados de ellos los de datos locales y globales, así como argumentos y parámetros.
- El ambiente tiene defectos propios de un desarrollo de software (típicamente los errores son poco explicados).
- ✓ El alumno quiere ir rápidamente a herramientas “profesionales” más complejas, lo que dificultaría su maduración a partir de los conceptos básicos. En este sentido Visual Da Vinci funciona como un “puente” que establece límites.
- Sería deseable tener una herramienta de menor nivel para la educación pre-universitaria (en la Escuela Media) que vaya evolucionando. En este sentido VDV no tiene grados de flexibilidad.

Conclusiones y Líneas de Trabajo actuales

Si partimos de la realidad que dan los elevados índices de deserción que se observan en la educación universitaria en informática, cuyas consecuencias socioeconómicas son por todos conocidas, y considerando que estos niveles de deserción se atribuyen, entre otros factores, a una mala orientación previa y al desconocimiento por parte de los ingresantes de las características de las tareas que se desarrollan en el campo y de las competencias requeridas para llevarlas a cabo, entendemos que todos los esfuerzos que se hagan por desarrollar enfoques y herramientas para la enseñanza temprana de los conceptos básicos (típicos de un curso CS1) son positivos.

En este sentido las ideas de las que nació el desarrollo de Visual Da Vinci siguen vigentes, aunque en su utilización sistemática se observan ventajas y desventajas.

Las líneas de trabajo actuales (más allá de las mejoras de VDV propias del mantenimiento de un sistema de software relativamente complejo) son:

- 1- Desarrollar todo un soporte en WEB para la orientación pre-universitaria no presencial, orientada a los alumnos y docentes del polimodal. En este soporte se prevén demostraciones interactivas utilizando VDV, que complementan las actividades de autoaprendizaje y tutoría.
- 2- Desarrollar tres niveles para Visual Da Vinci, el primero para la introducción de conceptos en la Escuela Media, el segundo (similar al actual) para los cursos introductorios de algorítmica y programación y el tercero para iniciar a los alumnos en los conceptos de concurrencia, utilizando el ambiente con múltiples robots.

Bibliografía

- [ACM68] ACM Curriculum Committee on Computer Science. "Curriculum '68: Recommendations for Academic Programs in Computer Science", Communications of the ACM, Vol.11, No.3. March 1968.
- [ACM78] ACM Curriculum Committee on Computer Science. "Curriculum '78: Recommendations for Academic Programs in Computer Science", Communications of the ACM, Vol.22, No.3. March 1979.
- [ACM84] ACM Curriculum Committee on Computer Science. "Curriculum '84: Recommendations for Academic Programs in Computer Science", Communications of the ACM, Vol.27, No.10. October 1984.
- [ACM91] ACM/IEEE-CS Joint Curriculum Task Force Report. "Computing Curricula 1991", Communications of the ACM, Vol.34, No.6. June 1991.
- [ACM01] ACM/IEEE-CS Joint Curriculum Task Force Report. "Computing Curricula 2001"
- [Bec01] Becker Weber, B. Teaching CS1 with Karel the Robot in Java. Conference Proceedings Thirty-second SIGCSE Technical Symposium on Computer Science Education. 2001.
- [Ber94] Bertone, R. Multimedia aplicada a orientación vocacional. ICIE 94. Bs. As. 1994.
- [Bry01] Bryant, R. Introducing Computer Systems from a Programmer's Perspective. ACM SIGCSE BULLETIN, Volume 33, Number 1. March 2001.
- [Bur94] Burger, J. Multimedia. Addison Wesley. 1994.
- [Cab00] Cabero, B. Nuevas Tecnologías aplicadas a la Educación. Editorial Síntesis. Madrid. 2000.
- [Cat99] Cataldo, Z. Revisión de marcos teóricos educativos para el diseño y uso de programas didácticos. ICIE99, Bs. As. 1999.
- [Cha90] Chang, S. Visual Languages and Visual Programming. Plenum Press. 1990
- [Cha97] Champredonde, R. Visual Tool from programming teaching. ICIE . Bs. As. 1997. Tesina de grado. Facultad de Informática. UNLP.1997.
- [Coo03] Cooper, S. Teaching Objects-first In Introductory Computer Science. ACM SIGCSE BULLETIN, Volume 35, Number 1. March 2003.

- [**Dan01**] Dann, W. Using Visualization To Teach Novices Recursion. ACM SIGCSE BULLETIN, Volume 33, Number 3. September 2001.
- [**Deg01**] De Giusti A. y otros. “Algoritmos Datos y Programas”. Prentice Hall 2001.
- [**Fag00**] Fagin, B. Using Ada-Based Robotics to Teach Computer Science. Conference Proceedings The 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. ITiCSE 2000.
- [**Gao00**] Gaona, A. The Relevance of Design in CS1. ACM SIGCSE BULLETIN, Volume 32, Number 2. June 2000.
- [**Gai98**] Gail, A. What Abilities Are Necessary for Success in Computer Science? ACM SIGCSE BULLETIN, Volume 30, Number 4. December 1998.
- [**Ghe91**] Ghezzi, M. Fundamentals of Software Engineering. Prentice Hall. 1991.
- [**Gin00**] Ginat D. Colorful examples for elaborating exploration of regularities in High-School CS1. The 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. ITiCSE 2000.
- [**Gol90b**] Golin, E. The Specification of Visual Language Syntax. Visual Languages and Computing. Volumen1. Número 2. 1990
- [**Gon02**] Gonzalez A. Hacia un perfil del ingresante de Informática. La articulación entre sus expectativas y la realidad. Seminario aspectos sociopolíticos universitarios. Carrera docente. UNLP. 2002.
- [**Inf94**] Informe de resultados de la Comisión de Enseñanza (90-95). Fac. Ciencias Exactas, UNLP.
- [**Inf95**] Informe de resultados de las Universidades Nacionales. Secretaría de Políticas Universitarias. Ministerio de Educación de la Nación. 1995.
- [**Inf02**] Informe de resultados de la Comisión de Ingreso (2002). Fac. de Informática de la UNLP.
- [**Inf03**] Informe de resultados de la Comisión de Ingreso (2003). Fac. de Informática de la UNLP.
- [**Kae01**] Karen R. “Teaching Programming from a Distance : Problems and a Proposed Solution.” ACM SIGCSE BULLETIN, Volume 33, Number 4. December 2001.
- [**Lee01**] Lee, J. Teaching and Learning in the 21st Century: The development of “future CS faculty” ACM SIGCSE BULLETIN, Volume 33, Number 2. June 2001.
- [**Lis03**] Lister, R. Introductory Programming, Criterion-Referencing, and Bloom. ACM SIGCSE BULLETIN, Volume 35, Number 1. March 2003.
- [**Mar03**] Martín María V. InterNet en las Escuelas: Mitos y Realidades. Revista Tramas de la Comunicación y la Cultura. Facultad de Periodismo UNLP. Año 2 – Num. 14 – Junio 2003
- [**Mci92a**] McIntyre W. A Visual Method for Generating Iconic Programming Environment. Rensselaer Polytechnic Institute. 1992. Tesis Doctoral.
- [**Mor02**] Moroni, N. SVED: Sistema de Visualización de Algoritmos. UNS. CACIC. Bs. As. 2002.
- [**Mye90**] Myers, B. Taxonomies of Visual Programming and Program Visualization. Visual Languages and Computing. Volumen 1. Número 1. 1990
- [**Ouv02**] Ouviña, P. Mapas conceptuales : una herramienta para el aprendizaje de estructuras de datos. Fac. de Ingeniería. UNPSJB. CACIC. Bs. As. 2002.
- [**Pat81**] Pattis, R. Karel the Robot : A Gentle Introduction to the Art of Programming. Wiley & Sons. 1981.
- [**Pan93**] Pandey, R. Is it Easier to Write Matrix Manipulation Programs Visually or Textually? An Empirical Study. IEEE Symposium on Visual Languages. 1993
- [**Pro00**] Proulx, V. Programming Patterns and Design Patterns in the Introductory Computer Science Course. ACM SIGCSE BULLETIN, Volume 32, Number 1. March 2000.
- [**Rae85**] Raeder G. A Survey of Current Graphical Programming Techniques. IEEE Computer. Volumen 18. Número 8. 1985
- [**Sch97**] Schunk Dale, H. Teorías de la educación. Prentice Hall. 1997.
- [**Señ99**] Señas, P. Mapas Conceptuales Hipermediales como Herramienta para la Representación de Conocimiento en Agentes Inteligentes Pedagógicos. UNS. 1999.
- [**Sha92**] Shaddock, P. Creaciones Multimedia. Waite Group Press. 1994.
- [**UNLP01**] Ordenanza de Articulación con la Escuela Media. UNLP. 2001