

Física computacional en el nivel medio: ¿una asignatura pendiente?

Hugo D. Navone ^{1,2,3} - Pablo A. Turner^{1,2}

¹Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario;

²Instituto de Física de Rosario (CONICET-UNR); ³Observatorio Astronómico Municipal de Rosario
hnavone@fceia.unr.edu.ar

La presencia constante de las Tecnologías de la Información y de la Comunicación en el mundo del trabajo y en el uso del tiempo libre es innegable. Tampoco pasa desapercibido su rol en la construcción del conocimiento científico y tecnológico. Sin embargo, no se observa claramente su correspondiente influencia en los procesos de enseñanza-aprendizaje de las ciencias, al menos de aquellas que, como la Física, tienen un alto nivel de formalización. Resulta notable observar cómo los alumnos que llegan a la Universidad están familiarizados con el uso de computadoras y cómo, a su vez, muy pocos han utilizado estas máquinas para la simulación y análisis de sistemas físicos. Si bien varios autores han propuesto actividades en este sentido, por diversas razones no han perdurado ni se han profundizado. La Física Computacional propone recurrir a la computadora, no sólo “como herramienta”, sino también “como laboratorio” en la construcción del conocimiento científico. Se constituye hoy en un abordaje complementario al teórico y al experimental y, como tal, debiera ser incluido adecuadamente en la escuela media. En este trabajo se re-explora esta temática desde la perspectiva propuesta, recabando sus antecedentes disciplinares más importantes y se bosqueja un posible plan de acción para su implementación.

Palabras clave: Física, Computación, Física Computacional, Escuela Media

The presence of the Information and Communication Technologies in the world of work and in the spare time activities is undeniable. Neither goes unnoticed its role in the construction in the scientific and technological knowledge. Nevertheless, its influence cannot be clearly seen in the teaching-learning process of sciences, at least no in those which, as physics, have a high level of formalization. It is worth while noticing how students who enter University are familiarized with the use of computers and how, at the same time, very few have used them for simulation and analysis of physics systems. Even though several authors have proposed activities in this matter, for various reasons have not lasted nor been thoroughly studied. Computational Physics proposes to use computers, not only as a tool, but also as a laboratory in the construction of scientific knowledge. Today it constitutes a complement to the theoretical and experimental approaches, and as such, it should be adequately included in high school. In the present work we re-analyze this thematic under the proposed perspective, collecting its most important disciplinary precedence and we sketch a possible plan for its implementation.

Keywords: Physics, Computation, Computational Physics, High School

Introducción

El hombre se ha caracterizado por ser un constante creador de herramientas, aparatos, máquinas y dispositivos que han permitido y le permiten la expansión de sus capacidades físicas e intelectuales: poleas, grúas, imprentas, teléfonos, barcos, aviones, televisores, computadoras, Internet, son sólo algunos ejemplos. Pero de todas las máquinas que el hombre ha inventado, quizás la que mayor impacto ha tenido en todos los ámbitos y, muy

particularmente, en la construcción del conocimiento científico y tecnológico, ha sido y es la computadora, ingenio que ha posibilitado el desarrollo de las Tecnologías de la Información y de la Comunicación (TICs).

Es posible buscar algunos hitos históricos que, de alguna manera, dan cuenta de la estrecha relación entre ciencias y máquinas en términos generales, y entre física y computadoras en particular. De todos ellos, quizás los más ilustrativos sean aquéllos que no sólo mues-

tran la presencia de esta relación sino que, además, ponen de manifiesto sus dimensiones más interesantes, las que a su vez importan en el contexto de este artículo por ser motivadoras para la introducción del trabajo con computadoras en la enseñanza y en el aprendizaje de la física.

La interpretación más común que surge a partir de estudiar la relación entre ciencias y máquinas a lo largo de la Historia de la Ciencia, consiste en considerar a la computadora como un recurso al cual se accede casi naturalmente para ser usado “como herramienta” o “máquina-herramienta” en la obtención de algún producto. Este producto puede ser un cálculo, la elaboración de un documento, el resultado de una consulta a una base de datos o un gráfico, por mencionar sólo algunos. Sin embargo, esta visión que nos es casi natural resulta ser parcial y, conceptualmente, bastante limitada, ya que nos sitúa ante una máquina que, de alguna manera, sólo parece contribuir a expandir nuestras capacidades físicas, ocultando -o menospreciando incluso- su rol en las actividades creativas del hombre de hoy y, muy particularmente, del investigador.

Quizás los primeros en exponer dicho rol de la computadora hayan sido, en 1953, E. Fermi, J. Pasta y S. Ulam (Jacovkis, 2004) quienes inventaron el concepto de “experimento numérico” o “experimento por computadora” al diseñar simulaciones para entender mejor el concepto de entropía. Esta expresión sigue siendo utilizada actualmente en la descripción de aquellos experimentos que se llevan a cabo sobre modelos computacionales de sistemas físicos para lograr entender su comportamiento, explicar su estructura o predecir su evolución.

Si bien este trabajo tiene importancia por haber inaugurado una nueva manera de ver al rol de la computadora en la construcción del conocimiento científico, fue Lorenz en 1963 quien puso de manifiesto la verdadera magnitud del mismo al tratar de modelar numéricamente la dinámica de fluidos en la atmósfera (Jacovkis, 2004). Hasta ese momento –salvo algunas excepciones que no lograron trascender con el correr de la historia- el pensamiento general de los físicos era que, de alguna

manera, los sistemas debían ser estables, esto es, que pequeños cambios en los datos debían provocar cambios razonablemente pequeños en los resultados. Justamente, Lorenz comprobó mediante “experimentos numéricos” que esto no era siempre así. En su modelo, por cierto para nada complejo, pequeños cambios en los datos conducían a resultados totalmente distintos e inesperados. Estos hechos dieron comienzo a lo que hoy conocemos como Teoría del Caos; esto es, experimentos realizados con una computadora permitieron detectar un tipo de fenómeno físico desconocido hasta entonces o que, al menos, no había sido evidenciado convenientemente. La máquina, entonces, pasa a ser un elemento determinante para el estudio de tales sistemas que no sólo aparecen en el ámbito de la meteorología o de la física, sino que abundan en muchos otros campos del saber (astrofísica, química, biología, economía, entre otros).

Actualmente, es claro que el impacto de la computadora en el desarrollo de la Física y de otras disciplinas relacionadas es muy relevante. Resulta imposible imaginar hoy líneas de investigación científica en física en donde la computadora no esté presente de alguna manera; ya sea desde su uso como máquina-herramienta comenzando por su intervención en la producción y comunicación de documentos científicos o en la gestión de información, pasando por aquellas temáticas que requieren el uso de cálculo simbólico, para seguir luego hacia aplicaciones basadas en cálculo numérico formuladas a partir de primeros principios, llegando hasta problemáticas en donde el abordaje sólo puede ser algorítmico o bien, en donde directamente no es posible recurrir a principios fundamentales, utilizándolas entonces, para descubrir regularidades y patrones (algunos autores usan el término “conocimiento”) inmersos en grandes bases de datos científicos (minería de datos). Pero a pesar de esta fuerte presencia de las computadoras en el proceso de investigación científica en física y en otras disciplinas, estando su uso ligado a una nueva forma con que los investigadores cuentan para interpelar, explorar y conjeturar acerca de los fenómenos naturales, otro es el panorama que se observa

en el ámbito educativo medio y, aún, en el nivel terciario.

Teniendo en cuenta lo expuesto, es posible inferir que el sistema educativo todavía no habría logrado una apropiación adecuada de la máquina desde el punto de vista aquí propuesto, de forma similar a la que es utilizada en la actividad científica y, muy particularmente, en el ámbito de las ciencias naturales.

Antecedentes y estado de situación

A lo largo de los años han surgido propuestas en dirección hacia el uso de la computadora en la enseñanza de las ciencias pero por diversas razones no se han logrado establecer e implementar efectivamente en nuestro sistema educativo. Quizás, una de las obras más cercanas al punto de vista aquí planteado sea la de Hurley (1986) quien propone la realización de proyectos y experimentos de física usando el lenguaje de programación LOGO. Este autor sugiere problemas de física resueltos con procedimientos en LOGO abarcando temas como: vectores, caída libre, movimiento de proyectiles, órbitas planetarias, oscilador armónico, leyes de la óptica geométrica, decaimiento radiactivo y dinámica de poblaciones. La propuesta es muy interesante y amplia, los problemas elegidos son pertinentes para el nivel medio y están bien documentados, pero en opinión de los autores la perspectiva desde la cual son propuestos no logra mostrar adecuadamente el rol de la computadora en la construcción del conocimiento en física. Además, al presuponer conocimientos de programación en general, y de programación LOGO en particular, la obra se torna un tanto árida para los docentes de física que deseen implementarla en el aula.

De la misma manera, y ahora más en el terreno de la matemática -aunque no privativa de ella como remarca el autor- podemos también citar los trabajos de Vitale (1988, 1990, 1991, 1992) quien propone conducir a esta disciplina hacia el terreno de la experimentación a través de la programación en un lenguaje formal como LOGO. Este autor destaca la necesidad de introducir el abordaje computacional para la resolución de problemas desde el comienzo de la escuela secundaria,

pero explicita que los mismos pueden ser trasladados a todos los niveles de enseñanza con una adecuada adaptación. En particular, se ocupa de abordar situaciones problemáticas en donde los profesores conozcan sólo un poco más que los alumnos, esto es, sin que de antemano cuenten con todas las respuestas. La idea es que el docente también sea un sujeto participativo del proceso de “experimentación matemática” propuesto. Este criterio, por supuesto, también es trasladable a otras disciplinas que hagan uso del recurso informático, como la física. Algunas de las problemáticas que propone Vitale son: evolución poblacional y equilibrio dinámico en sistemas predador-presa, diseño de autómatas celulares simples, representación de árboles y procesos aleatorios.

Las obras de Hurley y de Vitale se han tomado como exponentes de una línea de trabajo tendiente a incorporar a la computadora en la enseñanza y aprendizaje de las ciencias. Ambos implican un cierto nivel de complejidad y de elaboración, mostrando los siguientes elementos en común: a) se propone el uso del ordenador como “herramienta para pensar” a través del diseño de algoritmos que traten de simular el comportamiento de sistemas físicos o matemáticos y b) utilizan el mismo lenguaje de programación para la formalización de ideas, el LOGO.

Ahora bien, según podemos constatar actualmente, por lo menos en la zona de influencia de la Universidad Nacional de Rosario -aunque es posible que esta situación se replique en otros lugares de nuestro país-, ni este tipo de propuestas ni otras de similares características han logrado incorporarse efectivamente en la escuela media. Más aún, las evidencias indican que no existe un uso de la computadora en el nivel secundario dirigido a simular, analizar y entender los sistemas físicos. A través de entrevistas y encuestas realizadas a estudiantes universitarios de la carrera de Licenciatura en Física en el período 2003-2008 nos ha sido posible detectar que si bien más del 90% de los alumnos posee una computadora en su casa, que todos manifiestan conocer un sistema operativo (esto es, saber cómo se “opera” una máquina de este

tipo) y que aproximadamente el 85% asistieron a cursos de informática en el nivel medio, sólo el 45% manifiesta conocer algún lenguaje de programación¹ y sólo en casos muy aislados se ha utilizado a la computadora en el análisis de algún problema de ciencias.

Para completar este panorama y tratar de obtener una visión más certera acerca del estado de situación en esta temática, hemos realizado un recorrido por algunos libros de física dirigidos al nivel de enseñanza media tratando de explorar la existencia de manifestaciones acerca de la relación entre esta ciencia y las computadoras. Las obras analizadas fueron Maiztegui y Sábado (1997a, 1997b), Aristegui *et al.* (2005a, 2005b), Calderón *et al.* (2001), Rela y Sztrajman (1999a, 1999b), Cerdeira *et al.* (2001) y, Rubinstein y Tignanelli (2004a, 2004b).

El libro de Aristegui *et al.* (2005a) contiene en la sección *Recursos para el trabajo científico* el tema *La computación en la ciencia y en la técnica* (pp. 305-309) en donde se presenta a la computadora como “*un instrumento esencial para la investigación en las ciencias naturales*”. Brevemente, los autores tratan de abordar el rol del ordenador en ciencias y en tecnología mediante algunos ejemplos, y luego desarrollan una aplicación que consiste en simular el movimiento de una partícula en un fluido viscoso. Para ello utilizan el lenguaje de programación BASIC, listándose el código con algunas aclaraciones acerca del mismo. Si bien la propuesta es muy interesante, desde la perspectiva de nuestro trabajo es importante tener en cuenta los siguientes aspectos: a) se destaca a la computación en el desarrollo de las ciencias, pero su inclusión en la obra es muy breve, quedando reducida a un plano auxiliar y poco atractivo para el docente no formado en estas temáticas; b) parece acentuarse más la relación entre computadoras y tecnología que entre computadoras y ciencias; c) la máquina aparece nuevamente más bien como una máquina-herramienta que como una máquina-laboratorio, es decir la experimentación numérica no se hace explícita y d) se utiliza un lenguaje de programación que no es muy adecuado para trabajar en el ámbito educativo, existiendo una opción mucho

mejor como LOGO de acceso totalmente libre y gratuito².

Cada una de las obras de Rela y Sztrajman (1999a, 1999b) son acompañadas con un disquete que contiene *Ejercicios para la computadora*. Se trata de un conjunto de programas desarrollados inicialmente en GW-Basic³, y para los cuales se propone su ejecución en QBASIC⁴, con resoluciones numéricas de diversos problemas de física. Según los autores, los programas que se presentan “*se han mostrado útiles para ver con algún detalle procedimientos numéricos de resolución de problemas físicos*”. Además, se les propone a los estudiantes que introduzcan modificaciones en los programas para que “*puedan experimentar los efectos de cada cambio aunque no tengan noción de qué es, exactamente, lo que están haciendo*”. Aunque el planteo es atrayente y está dirigido, de alguna manera, a la realización de “experimentos”, no se trata de un abordaje desde el marco conceptual de la Física Computacional tal cual se propone en este trabajo, en donde la resolución de un problema se construye con los alumnos hasta lograr un modelo computacional para luego experimentar con él. Es decir, se presentan problemas casi cerrados, ya diseñados por los autores, y la participación que se sugiere a los alumnos y docentes está centrada en interactuar con el programa, pudiendo ser poco significativa en el proceso de enseñanza-aprendizaje. En este sentido, la propuesta parece caer más bien en el terreno de la “enseñanza asistida por computadoras”, limitando toda la riqueza y potencialidad que posee el abordaje computacional. En la obra tampoco se menciona como lenguaje de programación posible al LOGO, si bien aparecen mencionados otros lenguajes formales.

Los demás textos analizados no hacen uso del recurso computacional para abordar las diversas temáticas que presentan.

Llegado a este punto, resulta pertinente tratar de realizar un breve resumen con el objeto de relevar el estado de situación descrito, plantear interrogantes adecuados y conjeturar explicaciones posibles con la mirada puesta en la elaboración de un posible plan de trabajo que permita incorporar a la Física Computa-

cional en la enseñanza-aprendizaje de la física en el nivel medio.

Vimos que en el ámbito de la investigación científica, y muy particularmente en la física, la computación desempeña un rol fundamental en la construcción del conocimiento, convirtiéndose en un dispositivo que los científicos utilizan para interpelar a la naturaleza, explorando y proponiendo explicaciones acerca de muy variados fenómenos y de muy diversa complejidad. Si bien esto es innegable, advertimos luego que aún existiendo proyectos interesantes para incorporar a la computadora en la enseñanza-aprendizaje de las ciencias (Hurley, 1986; Vitale, 1988, 1990, 1991, 1992) en concordancia con su presencia en el desarrollo científico y tecnológico, esto no se ha dado convenientemente o directamente está ausente. Para afirmar esto presentamos evidencias recabadas a partir de encuestas y entrevistas realizadas a estudiantes universitarios de Licenciatura en Física de nuestra universidad y, también, relevamos parte de la bibliografía recomendada para física de nivel medio. Concluimos que la ausencia del abordaje computacional –y más precisamente desde el marco conceptual que brinda la Física Computacional– como complemento para la resolución de problemáticas de la física y de disciplinas relacionadas en el sistema educativo medio es al menos preocupante, dada toda la potencialidad educativa y formativa que el mismo implica.

Si bien no resulta fácil indagar sobre las posibles causas que llevan a esta situación, trataremos al menos de puntualizar –en base a nuestra experiencia– algunos aspectos que nos parecen importantes, previniendo al lector que este análisis requeriría de un estudio exhaustivo que está fuera del alcance de este trabajo.

Obstáculos y explicaciones posibles

En nuestra opinión uno de los factores determinantes de la ausencia de un enfoque computacional en la resolución de problemas de física en el nivel medio es, justamente, el desconocimiento del rol que la computadora desempeña en la construcción del conocimiento científico. Probablemente perdure una

visión de máquina-herramienta, utilitarista, que hace que la computadora sea relegada a un segundo o tercer plano o a una mera curiosidad en la enseñanza-aprendizaje de las ciencias. En este sentido, es interesante recordar que el enfoque experimental en física (Física Experimental) se encuentra sometido a un proceso de características similares, aunque los factores que se encuentran en juego sean diferentes.

También podemos decir que hay una marcada ausencia del enfoque computacional en los libros de texto de física y, presumimos, que el mismo también lo está en las carreras de formación docente o al menos, si está presente, no logra ser trasladado al aula como hemos podido constatar.

Todo esto deriva en una paradoja interesante: por un lado existen recursos computacionales en muchas de las escuelas medias, en estos establecimientos también se dan clases de Informática, la mayoría de los alumnos tienen algún contacto con esta tecnología independientemente de su nivel social y, generalmente, a todos los adolescentes les resulta atractiva, pero sin embargo este recurso no es aprovechado convenientemente en las clases de ciencias.

Ante esta situación podemos postular lo siguiente: es posible que no existan recursos humanos capacitados que puedan afrontar proyectos de este tipo ya sea porque no reconocen su importancia, porque sean reticentes al uso de esta tecnología, porque no cuentan con material de referencia adecuado o porque no existen planes de capacitación docente ni políticas institucionales en este sentido. Sumado a ello, es probable que la inclusión de la disciplina Informática entorpezca el proceso de incorporación de la computadora en la enseñanza de las ciencias, al transformarse ésta en una disciplina aislada dejando que todo el trabajo con estas máquinas recaiga sobre el docente de esta asignatura, como bien puntualiza Vitale (1991).

Además, las dificultades que los adultos naturalmente tenemos ante el manejo de las nuevas tecnologías –que son vistas como “máquinas de los jóvenes”–, nuestro propio miedo a equivocarnos y a no poder “contro-

lar” todo el trabajo que se pueda proyectar sobre estos dispositivos, operan como verdaderos obstáculos cognitivos. En este sentido, quizás sea interesante destacar que la estructura funcional del ordenador debiera ser considerada por alumnos y docentes como un “objeto de conocimiento” (Vitale, 1990, 1991) y afianzarse desde esta perspectiva para vencer todas las resistencias ante este ingenio de apariencia sumamente compleja.

Un párrafo aparte merece el análisis del no uso del lenguaje de programación LOGO en la escuela media, ya sea tanto en el área de Informática, como en aquellos intentos que proponen el abordaje computacional de algún problema en el área de física, como se manifiesta en las obras de Aristegui *et al.* (2005a) y de Rela y Sztrajman (1999a, 1999b). LOGO es un lenguaje de programación de alto nivel que ha sido especialmente diseñado para su uso en el ámbito educativo por un equipo liderado por Seymour Papert (1987, 1997, 1995) en el Instituto Tecnológico de Massachussets (MIT). Es un lenguaje de fácil aprendizaje que permite interactuar muy rápidamente con la computadora. Posee un conjunto de instrucciones básicas (primitivas) relativamente pequeño que, a su vez, pueden ser utilizadas para generar otras instrucciones más complejas. En este sentido, podemos decir que el LOGO es un lenguaje orientado a procedimientos. Las primitivas del Lenguaje Formal (LF) LOGO se acercan al Lenguaje Natural (LN), existiendo versiones muy potentes desarrolladas en español, hecho que simplifica notablemente su introducción en edades muy variadas. A su vez, si bien LOGO es muy accesible inicialmente, es un LF complejo que no tiene limitaciones para su uso en cualquier ámbito educativo y con diversos propósitos. El LOGO es un lenguaje que acompaña el desarrollo de los alumnos, puede ser usado en todas las edades y en muy diversas situaciones de aprendizaje. En particular, resulta una herramienta muy adecuada para el trabajo en programación con grupos de adultos.

En LOGO se dispone de un puntero o cursor que permite graficar rápidamente en la pantalla de la computadora lo que el usuario indique sin necesidad de recurrir a complejas

instrucciones. Este cursor recibe el nombre de “tortuga” y en muchas versiones de este lenguaje orientadas a niños pequeños adquiere la forma de este animalito. Es por ello que muchos jóvenes recuerdan al LOGO como el “lenguaje de la tortuguita”, según surge de algunas entrevistas realizadas a alumnos universitarios de nuestra Licenciatura en Física. Quizás uno de los puntos más importantes que se establece como obstáculo para el uso del LOGO en la enseñanza media sea justamente éste, el de estar asociado con un “lenguaje para niños pequeños”, tanto por alumnos como por docentes.

A esta inconveniencia, se le suma un desconocimiento generalizado de todas las potencialidades del LOGO como lenguaje de programación y, fundamentalmente, como herramienta educativa para la enseñanza-aprendizaje de las ciencias. Además, conspira la poca disponibilidad de recursos educativos orientados en esta dirección, tales como bibliografía y cursos de capacitación docente que logren integrar las experiencias y propuestas que hoy se encuentran demasiado dispersas para poder ser implementadas en el aula.

Un lenguaje, todos los lenguajes

Si bien hemos destacado las bondades del LOGO como lenguaje de programación para formalizar el abordaje computacional de problemáticas de las ciencias naturales, creemos importante trabajar desde una perspectiva un poco más general. El lenguaje LOGO posee muchas virtudes que lo hacen especialmente apto para su aplicación en el ámbito educativo, principalmente el hecho de haber sido creado para cumplir este rol a partir de fundamentos pedagógicos dirigidos a entender la problemática del aprendizaje en general, y del aprendizaje con computadoras, en particular (Papert, 1987). Sin embargo, creemos que es pertinente para el nivel medio hacer un esfuerzo por introducir a la programación tratando de abstraernos de un lenguaje en particular, develando los elementos fundamentales que tienen en común todos los lenguajes formales orientados a procedimientos.

En este artículo hemos usado las palabras máquinas, computadoras, ordenadores y,

también, programas, modelos computacionales y modelos numéricos casi en forma equivalente. Cuando nos referimos a la relación máquinas y ciencias o computadoras y ciencias, estamos siempre diciendo implícitamente que en el concepto “máquinas” o “computadoras” también interviene la idea de programa (software) y no sólo la parte tangible o física de estos dispositivos (hardware). Usamos todos estos conceptos como uno sólo, no por error o descuido, sino porque en realidad no hay necesidad de distinción alguna para el tipo de abordaje que estamos proponiendo. Sí, en cambio, es necesario distinguir entre componentes físicas y lógicas, es decir entre hardware y software, cuando examinamos detenidamente al ordenador como un objeto de estudio y pretendemos distinguir e incorporar todas las funcionalidades disponibles en este ingenio.

Al respecto, Niklaus Wirth⁵ explicita que programar es la *“actividad que consiste en idear y proyectar una nueva máquina (cuyo funcionamiento se realiza por intermedio de otra máquina preexistente, de carácter universal)”* (Wirth, 1984). Es esta concepción, justamente, la que nos permite amalgamar las denominaciones anteriores asociadas con máquinas y programas, y movernos libremente de unas a otras en el contexto de este trabajo.

En base a esta definición la idea es, entonces, identificar aquellos elementos o mecanismos esenciales presentes en los lenguajes de programación orientados a procedimientos que nos permitan construir “máquinas-laboratorio” para estudiar sistemas físicos mediante la realización de experimentos numéricos.

El conocimiento de estos elementos fundamentales y su identificación en un determinado lenguaje formal posibilitará, a su vez, el aprendizaje de otros lenguajes. Se trata entonces de adquirir la habilidad de manipular lenguajes formales, evitando quedar “anclados” a uno en particular por más potente que éste sea. En opinión de los autores, esta es una estrategia muy adecuada para trabajar la temática en el nivel medio, ya sea en el área de informática o en el contexto de una disciplina

en particular, como física o matemática, en donde se intente resolver problemas usando computadoras.

La Física Computacional como estrategia para el estudio, análisis y comprensión de los sistemas físicos se basa en la construcción de “algoritmos” (Marro, 2000) los que luego son formalizados usando un lenguaje de programación para que puedan ser procesados por una computadora.

Podemos definir en pocas palabras a un algoritmo como un conjunto finito de pasos cuya ejecución lleva a la solución de un problema dado en un tiempo razonable.

Se ha demostrado que para diseñar cualquier algoritmo bastan tres principios de control: ejecución secuencial, ejecución repetitiva y ejecución condicional (Wirth, 1984). La ejecución secuencial especifica algo que todos aceptamos, casi sin darnos cuenta, y es que de no mediar ninguna instrucción en contrario, las órdenes especificadas en un algoritmo se ejecutan una tras otra, esto es, “secuencialmente”. La ejecución repetitiva nos dice que se cuenta con alguna instrucción que permite “repetir” un conjunto de órdenes o acciones. Mientras que la ejecución condicional controla la realización de conjuntos de instrucciones siempre y cuando se cumpla una “condición” estipulada en el algoritmo.

Todo lenguaje orientado a procedimientos debe contener estructuras o instrucciones basadas en estos principios y, entonces, la especificación de la secuencia de sentencias que componen un algoritmo resulta independiente de la sintaxis particular de un lenguaje de programación dado.

Además de estas estructuras, todo lenguaje hace uso de objetos constantes y variables. Los objetos constantes son los que no cambian a medida que se ejecuta un algoritmo, mientras que los variables son aquellos cuyo valor puede ser modificado en cualquier momento. A su vez, necesitamos contar con las siguientes funcionalidades: asignar valores a variables; ingresar datos al programa y mostrar los resultados que éste produce (instrucciones de entrada y salida); escribir expresiones aritméticas, relacionales (predicados simples) y lógicas (predicados compuestos); dis-

poner de funciones internas definidas en el lenguaje (las funciones matemáticas más usuales, por ejemplo); definir estructuras de datos o datos estructurados (como arreglos y listas); construir nuestros propios procedimientos o “subalgoritmos”, y usar archivos para leer datos y/o almacenar resultados.

Identificar todos estos elementos en un lenguaje de programación equivale a conocerlo para poder utilizarlo en la formalización de algoritmos. Trabajando desde este punto de vista, el aprendizaje de un lenguaje de programación en realidad nos permite desarrollar la habilidad de poder aprender cualquier otro.

Ahora bien, como nuestro objetivo es conjeturar y argumentar acerca de la posibilidad de trabajar en Física Computacional en el nivel medio usando LOGO, mostraremos algunos de los elementos de este lenguaje de programación y, también, ciertas particularidades que lo hacen especialmente apto para el trabajo en física, a la vez que desarrollamos una aplicación concreta.

Una aplicación, todas las aplicaciones

Cuando un jugador de fútbol ejecuta un tiro con pelota detenida le imprime al balón una cierta velocidad inicial con un determinado ángulo respecto del plano horizontal (nivel del campo de juego). Bien sabemos que en ausencia de rozamiento (en el vacío) es posible demostrar que la trayectoria de la pelota es una parábola. Mientras el “esférico” asciende el módulo de su velocidad va disminuyendo hasta alcanzar un valor mínimo en el punto más alto de su trayectoria (vértice de la parábola), y luego comienza a aumentar a medida que va descendiendo hasta tocar el suelo (alcance de la pelota).

Para que haya una variación de la velocidad debe estar actuando una fuerza. Esta fuerza es el peso, que le da a la pelota una aceleración en dirección vertical y con sentido hacia abajo. La aceleración que está actuando en cada instante es la aceleración gravitatoria o simplemente gravedad (9.8 m/s^2). Cuando la pelota está subiendo, la aceleración hace que la velocidad vaya disminuyendo paulatinamente y cuando está descendiendo la acelera-

ción provoca un aumento progresivo de la velocidad.

Este problema tiene solución analítica y consideramos necesario construir este tipo de resolución en primera instancia para luego comenzar a trabajar en su modelado computacional. La ventaja de realizar los dos abordajes reside, justamente, en la posibilidad de comparar ambos enfoques y en mostrar los alcances y limitaciones de cada uno de ellos. En este artículo, presentaremos la resolución computacional de este problema a los efectos de introducir algunos elementos comunes a todos los lenguajes de programación por intermedio del lenguaje LOGO, para luego comentar algunas potencialidades del mismo y mostrar las posibilidades que brinda el trabajo en Física Computacional en el nivel medio usando como ejemplo este problema y sus derivados.

Para diseñar un algoritmo de cálculo que permita obtener la trayectoria de la pelota sólo basta considerar al movimiento como “dividido” en pequeños intervalitos de tiempo (en donde el móvil recorre una pequeña distancia) durante los cuales la velocidad se puede considerar constante. Transcurrido cada intervalito de tiempo, la velocidad se corrige como producto de las fuerzas (aceleraciones) que intervienen en el movimiento (sólo gravedad, cuando se considera un lanzamiento en el vacío). Una vez corregidas, las velocidades se toman nuevamente como constantes para el próximo intervalo temporal. De esta manera es posible ir obteniendo las posiciones en el plano (x,y) de la pelota o sea, “su trayectoria”, a partir del cálculo de las velocidades que son “corregidas” en cada intervalo de tiempo por las aceleraciones involucradas en el movimiento. La trayectoria de la pelota finaliza cuando alcanza el suelo, esto es, cuando su coordenada vertical es nula o bien, también podemos decir que el movimiento se mantiene o se realiza siempre y cuando su coordenada vertical sea mayor a cero.

Teniendo esto en mente, sólo falta formalizar el algoritmo usando un lenguaje de programación de alto nivel; en nuestro caso, el LOGO:

```

para saque :V :alpha
haz "dt 0.01
haz "g 9.8
haz "Vx :V * cos :alpha
haz "Vy :V * sen :alpha
ponxy (coorx + :Vx * :dt) (coory + :Vy * :dt)
mientras [ coory > 0 ] [ haz "Vy :Vy - :g * :dt ponxy (coorx + :Vx * :dt) (coory + :Vy * :dt) ]
fin

```

En el cuadro anterior se muestra la definición de un **procedimiento LOGO** mediante las primitivas del lenguaje **para** y **fin**, esto es, el nombre del procedimiento que acabamos de diseñar es **saque** y las primitivas mencionadas determinan el comienzo y el final de la secuencia de instrucciones LOGO que resuelven el problema. El nombre del procedimiento –**saque** en este caso– es elegido por el programador. Los valores **0.01**, **9.8** (gravedad) y **0** son objetos constantes, es decir, su valor no puede cambiar mientras se ejecuta el algoritmo (procedimiento). En cambio, los objetos **V**, **alpha**, **dt**, **g**, **Vx**, **Vy** son objetos variables definidos por el programador, es decir, su valor puede cambiar mientras se ejecuta el procedimiento. A su vez, el LOGO, como cualquier otro lenguaje de programación tiene una sentencia de asignación de valores a variables. Esta sentencia es **haz** y en ella se especifica primero el nombre de la variable indicado con comillas iniciales y luego, separado por un espacio, el valor a asignar (que puede ser una constante, como 9.8 en nuestro caso, o bien el resultado de la evaluación de una expresión). El LOGO distingue el nombre de las variables –utilizando comillas para especificarlos, por ejemplo “**g**– de los valores de las variables, para los que usa dos puntos antes de su denominación, por ejemplo **:g**. El módulo de la velocidad inicial está dado por **:V** y el valor del ángulo inicial por **:alpha**. El valor de la componente horizontal de la velocidad está dado por **:Vx** y el de la componente vertical por **:Vy**. Los valores iniciales de las componentes de la velocidad se calculan mediante las siguientes expresiones **:V * cos :alpha** y **:V * sen :alpha**, respectivamente. En estas expresiones aparece el operador aritmético de multiplicación simbolizado por ***** y las funciones

internas de la biblioteca matemática del LOGO **cos** y **sen** utilizadas para calcular el coseno y el seno del ángulo **:alpha**. Las primitivas **coorx** y **coory** proporcionan la posición (x,y) del cursor (tortuga) en la pantalla de graficación y, por tanto, pueden ser utilizadas para ir actualizando la posición a medida que el móvil (pelota) se desplaza luego de transcurrir un intervalo de tiempo **:dt**. Las posiciones hacia las que se dirige el móvil se especifican mediante la primitiva **ponxy**, a la cual se le deben indicar las nuevas coordenadas del mismo. La instrucción **ponxy (coorx + :Vx * :dt) (coory + :Vy * :dt)** le dice al cursor que se mueva desde la posición en donde se encontraba (**coorx**, **coory**) hasta la próxima posición que se calcula como (**coorx + :Vx * :dt**, **coory + :Vy * :dt**), asumiendo que la componente vertical de la velocidad se mantiene constante durante el intervalo de tiempo **:dt**. Hay que tener en cuenta que cada vez que el cursor se desplaza lo hace dibujando y, por lo tanto, va trazando una aproximación de la trayectoria ideal del móvil o parábola de tiro. Considerando todo esto, la repetición del movimiento intervalito a intervalito de tiempo es controlada y desplegada por la siguiente instrucción LOGO:

```

mientras [coory > 0] [haz "Vy :Vy - :g * :dt
ponxy (coorx + :Vx * :dt) (coory + :Vy * :dt)]

```

La lectura de esta instrucción en lenguaje natural es la siguiente: **mientras** la coordenada vertical sea positiva (**coory > 0**), modificar la componente vertical de la velocidad debido a la aceleración de la gravedad durante el intervalo de tiempo estipulado (**haz "Vy :Vy - :g * :dt**), y mover el cursor hacia la nueva posición suponiendo constantes las componentes calcu-

ladas de la velocidad (V_x no cambia en este caso).

El procedimiento se ejecuta invocándolo por su nombre junto con los argumentos de entrada de la siguiente manera: **saque 20 45**, en donde V asume una velocidad inicial de 20 m/s y el ángulo α que se especifica es de 45° (por ejemplo). El resultado de este procedimiento es el dibujo en la pantalla del ordenador de la trayectoria del móvil (una parábola en este caso).

De esta manera, con elementos muy simples, hemos conseguido un modelo computacional del tiro oblicuo a la vez que investigamos conceptos de lenguajes de programación, formalización de algoritmos y programación LOGO. Llegado a este punto, resulta interesante re-explorar la física del tiro oblicuo. En primer lugar, se puede estudiar con los alumnos toda la potencialidad de la solución analítica frente a la “aproximación” computacional desarrollada. La solución analítica de este problema nos brinda una descripción completa del fenómeno físico en las condiciones en que éste se desarrolla, obteniendo la función que describe la trayectoria del móvil, así como también diversos parámetros que la caracterizan como: altura máxima, alcance máximo, determinación del ángulo de máximo alcance, entre otros. Un trabajo para realizar con los alumnos es proponerles que en el mismo procedimiento LOGO calculen el resultado analítico y todas las cantidades mencionadas, representándolas gráficamente para poder realizar comparaciones entre una y otra solución. Esta tarea, muy simple, comienza a abrir las puertas hacia el desarrollo de “experimentos numéricos”.

Llegada esta instancia, la solución numérica pudiera aparecer como artesanal y hasta anacrónica, exigiendo la necesidad de “ensuciarse las manos” para extraer un resultado que es sólo semejante al analítico y, por lo tanto, “imperfecto”. Es así como los alumnos se encuentran con una trayectoria que, en lugar de ser una curva suave como la analítica, va “de a saltos” o parece un “serrucho”, según sus propias palabras⁶. Para obtener una curva más suave es necesario jugar con el valor del parámetro Δt (intervalo temporal) y

esto genera cierta desconfianza e inseguridad en los participantes frente a la potencia de la solución analítica.

Sin embargo, mientras esta última llega hasta allí, el modelo numérico no tiene límites en cuanto a la descripción de problemas mucho más complejos. Realizando pequeñas modificaciones en el algoritmo de base ya diseñado podremos calcular trayectorias para muy variadas condiciones físicas en donde la solución analítica es muy difícil de obtener o bien es demasiado simplificada para describir el sistema o, directamente, no existe. Ahora, estaríamos en condiciones de pensar este problema cuando existe un medio de diferentes características, sería posible simular el movimiento de una pelota en el aire y, si queremos complicar más este experimento, se podría simular qué sucede con la trayectoria si la pelota se pateo con “efecto” (presencia de rotación). Aparece en escena la masa del móvil, su forma y las características del medio, así como también una velocidad de rotación. En este punto es interesante notar que el medio no necesariamente debe ser real, la posibilidad de trabajar con un modelo computacional también permite jugar con medios artificiales “inventados” por los propios alumnos y discutir acerca de las trayectorias resultantes.

Para completar esta idea, vamos a simular el movimiento de una pelota en el aire, considerando que la misma es lanzada sin rotación⁷, esto es sin “efecto”⁸. Ahora, la fuerza que produce la variación de la velocidad es la suma del peso más la fuerza de arrastre debida a la presencia del aire. Simplificando el problema, pero sin perder generalidad, la masa y la forma del móvil, y las características del medio (aire) entran en un coeficiente que, multiplicado por la velocidad, define la aceleración de arrastre para cada componente (el valor de este coeficiente puede ser estimado a partir de las características del móvil y del medio).

Para modelar este último problema el procedimiento LOGO anterior se modifica de la siguiente manera (se destacan en negritas las sentencias incorporadas):

```

para saque :V :alpha
haz "dt 0.01
haz "g 9.8
haz "Vx :V * cos :alpha
haz "Vy :V * sen :alpha
haz "frx 0.025
haz "fry 0.025
haz "fesc 12
ponxy (coorx + :Vx * :dt) (coory + :Vy * :dt)
mientras [ coory > 0 ] [haz "Vmod raizcuadrada(:Vx * :Vx + :Vy * :Vy) ~
    haz "Vx :Vx - :frx*:Vx*:Vmod*:dt ~
    haz "Vy :Vy - :g*:dt - :fry*:Vy*:Vmod*:dt ~
    ponxy (coorx + :fesc * :Vx * :dt) (coory + :fesc * :Vy * :dt) ]
fin

```

En este algoritmo, al igual que en el anterior, utilizamos una estructura de “repetición iterativa” -la estructura **mientras**- para construir la trayectoria de la pelota. El valor del módulo de la velocidad **:Vmod** se calcula en cada iteración usando la función **raizcuadrada** que pertenece a la biblioteca de funciones internas del lenguaje LOGO.

En el siguiente programa LOGO, que

resuelve el mismo problema, presentamos una estrategia de cálculo alternativa que hace uso de una “repetición recursiva”. Para ello, se diseña un procedimiento denominado **trayectoria** que se invoca a sí mismo con los valores de las componentes de la velocidad sucesivamente corregidas por las aceleraciones involucradas en cada intervalito de tiempo:

```

para saque :V :alpha
haz "Vx :V * cos :alpha
haz "Vy :V * sen :alpha
haz "g 9.8
haz "dt 0.01
haz "frx 0.025
haz "fry 0.025
haz "fesc 12
trayectoria :Vx :Vy
fin

```

```

para trayectoria :Vx :Vy
ponxy (coorx + :fesc*:Vx*:dt) (coory + :fesc*:Vy*:dt)
haz "Vmod raizcuadrada(:Vx*:Vx+:Vy*:Vy)
haz "Vx :Vx - :frx*:Vx*:Vmod*:dt
haz "Vy :Vy - :fry*:Vy*:Vmod*:dt - :g*:dt
si coory > 0 [trayectoria :Vx :Vy]
fin

```

El valor **:fesc** sólo se utiliza para modificar la escala de la representación gráfica a los efectos de hacer que la trayectoria se vea más claramente cuando se trabaja con velocidades y medios “cuasi-reales” (por ejemplo, 15 m/s para una velocidad inicial de un saque de arco y un coeficiente que representa la fricción de aproximadamente 0.025 m^{-1} para una pelota de fútbol convencional moviéndose en el aire y a nivel del mar).

Propuesta de un posible plan de trabajo

El rol que desempeña la Física Computacional en la construcción del conocimiento disciplinar, las propuestas educativas realizadas por diversos autores, la ausencia de este planteo en los libros de texto del nivel medio, la existencia de recursos computacionales en la escuela media, el especial interés que demuestran los alumnos por las TICs y el disponer de un LF como el LOGO, especialmente diseñado para educación, en español y de

libre disponibilidad, son factores que nos animan a bosquejar –basados también en experiencias desarrolladas en distintos ámbitos– una posible metodología de trabajo para tratar de incorporar este enfoque en la escuela media.

Nos parece adecuado dividir las actividades en dos espacios de enseñanza-aprendizaje y en tres tiempos educativos. Un tiempo de introducción, motivación y planificación de la temática computacional y de los problemas de física que se intentan abordar, en el aula; un tiempo de construcción e implementación de los modelos computacionales, en el laboratorio de informática pero en situación de taller; y un tiempo posterior, de regreso al aula, en donde se propone reflexionar y discutir sobre todo lo desarrollado, es decir, un tiempo de síntesis.

Para ello es necesario contar con material de apoyo adecuado para alumnos y docentes. Creemos que deberían proponerse Unidades Didácticas sobre cada tema a desarrollar que sean auto-contenidas, con Teoría y Problemas, disponibles para ser seguidas por los alumnos durante el desarrollo de los tres tiempos de enseñanza-aprendizaje. Estas Unidades conformarían el material didáctico de base para sustentar a los Proyectos de Física Computacional que serían propuestas de actividades con un grado de complejidad creciente, tal como la presentada en este artículo (Tiro Oblicuo en el Vacío y con Rozamiento).

En nuestra opinión, las Unidades Didácticas deberían abarcar los siguientes temas: U1) La computadora como objeto de estudio: recursos y funciones, sistemas operativos, conectividad, software libre y software propietario; U2) Representación interna de la información; U3) Errores: expresiones matemáticas y computacionales; U3) Algoritmos y lenguajes de programación en general y U4) Programación en lenguaje LOGO. Todas estas Unidades deberían proponer el desarrollo de algoritmos de distinto tipo y contener problemas de física.

En el caso de los Proyectos Computacionales, la idea es postular actividades abiertas que den lugar a la experimentación y a la exploración de conceptos y problemáticas de la física

a través del diseño de modelos computacionales. A modo de ejemplo, proponemos: PFC1) Suma de vectores; PFC2) Tiro oblicuo, sin rozamiento y con rozamiento; PFC3) Cálculo de órbitas: dos soles y un planeta (problema de 3 cuerpos); PFC4) Dinámica de la polución en un curso de agua; PFC5) Dinámica de poblaciones (problema predador-presa) y PFC6) Desarrollo de páginas web para mostrar, comunicar e intercambiar procedimientos LOGO que aborden los problemas de física desarrollados. Todos estos proyectos debieran ser trabajados tratando de fortalecer un enfoque interdisciplinar, promoviendo la vinculación de la física con los demás campos del saber.

Conclusiones

La Física Computacional propone, de alguna manera, una nueva estrategia para abordar el aprendizaje de la física –que puede pensarse como complementaria al enfoque teórico y experimental tradicional– al estar presente en el proceso de construcción del conocimiento disciplinar como un método para interpelar a la naturaleza.

Su implementación en el ámbito educativo desarrolla habilidades y estilos de pensamiento que son necesarios tanto en física como en disciplinas relacionadas, trascendiendo el marco escolar hacia todo tipo de actividades. Estas habilidades están relacionadas con el mundo del trabajo en donde las TICs están permanentemente presentes, siendo ellas de especial interés para los adolescentes. Esta temática llama la atención de los participantes porque pone a prueba sus capacidades, materializándose en un juego y en un desafío. Por otra parte, el trabajo con computadoras hace que los alumnos reflexionen acerca de su propio pensamiento, siendo la programación LOGO una herramienta especialmente adecuada para ello. Todo esto sin perder de vista la importancia que tiene el trabajar inicialmente con independencia de un lenguaje de programación en particular, como ya puntualizáramos oportunamente.

En este artículo hemos analizado la situación actual en esta temática, tanto sea desde una revisión de los antecedentes como desde

las experiencias e implementaciones desarrolladas en distintos ámbitos educativos. Mostramos el diseño de un posible plan de acción basado en distintos espacios y tiempos de enseñanza-aprendizaje, indicando cuáles serían las Unidades Didácticas, sus posibles contenidos y algunos Proyectos de Física Computacional, todo ello recurriendo al LOGO como LF.

Los primeros ensayos de este plan, con un mayor nivel de complejidad, han sido volcados en un Taller de Física Computacional que actualmente se dicta en el segundo año de nuestra Licenciatura en Física, lo que nos ha

permitido realimentar la propuesta a partir de los comentarios y de las dificultades de los alumnos, y nos ha mostrado las habilidades que se promueven y su necesidad de implementación temprana, al menos en los últimos años de la escuela media.

Finalmente, es importante remarcar que este tipo de proyectos debieran estar acompañados por una estrategia especialmente diseñada para la formación de recursos humanos en este sentido y por políticas educativas e institucionales que posibiliten su introducción, fortalecimiento y perdurabilidad.

Notas

- ¹ Muchos de los alumnos aclaran que se acercaron a la programación por propia inquietud y no por haber sido éste un tema de la currícula escolar.
- ² En el portal educ.ar se encuentra disponible una versión libre del lenguaje de programación LOGO en español.
- ³ Intérprete para el desarrollo de aplicaciones en lenguaje BASIC.
- ⁴ QBASIC es un intérprete y compilador para el diseño de programas en lenguaje BASIC posterior al GW-Basic.
- ⁵ Niklaus Wirth diseñó el lenguaje PASCAL cuyo objetivo era facilitar la enseñanza de la programación.
- ⁶ Este problema se ha trabajado durante el desarrollo de talleres no formales de informática para alumnos de nivel medio y en ese ámbito se han recopilado las frases y opiniones citadas.
- ⁷ El ejemplo presentado está basado en los apuntes de *Mecánica* (Capítulo 4: Dinámica) del Prof. Julio Gratton (Universidad de Buenos Aires). Dicho material también incluye el problema de un móvil con rotación.
- ⁸ En este caso, la trayectoria seguirá siendo en un plano. Si tratamos de simular un movimiento con rotación obtendremos una trayectoria en el espacio agregándole demasiada complejidad al problema.

Referencias

- Aristegui A.; Baredes, C. F.; Dasso, J.; Delmonte, J. L.; Fernández, D. P.; Sobico, C. I. y Silva, A. M. (2005a). *Física I*, Buenos Aires: Santillana Polimodal.
- Aristegui A.; Baredes, C. F.; Dasso, J.; Delmonte, J. L.; Fernández, D. P.; Sobico, C. I. y Silva, A. M. (2005b). *Física II*, Buenos Aires: Santillana Polimodal.
- Calderón, S. E.; Lemarchand, G. A.; Naso, C. A.; Navas, D. R.; Negroti, P. F.; Rodríguez Usé, M. G.; Vázquez, S. M.; Codner, G. (2001). *Física Activa*. Buenos Aires: Puerto de Palos.
- Cordeira, S.; Cwi, M.; Ferrari, H.; Greco, M.; Marín, G.; Mollerach, R.; Ortí, E.; Tonina, A. y Turner, S. (2004). *Ciencias Naturales y Tecnología 9º EGB*, Buenos Aires: Aique.
- Hurley, J. P. (1986). *Física con Logo: Proyectos y experimentos*, Madrid: Anaya Multimedia.
- Jacovkis, P. (2004). Computadoras, modelización matemática y ciencia experimental. *Mecánica Computacional XXIII*, pp. 2747-2758.
- Maiztegui A. y Sábato, J. (1997a). *Física I*, Buenos Aires: Kapeluz.
- Maiztegui A. y Sábato, J. (1997b). *Física II*, Buenos Aires: Kapeluz.
- Marro, J. (2000). Física y Computación. *Reporte Interno del Instituto Carlos I de Física Teórica y Computacional*. Facultad de Ciencias, Universidad de Granada (España).

- Papert, S. (1987). *Desafío a la mente: Computadoras y Educación*, Buenos Aires: Ediciones Galápagos.
- Papert, S. (1995). *La máquina de los niños: Replantearse la educación en la era de los ordenadores*, Buenos Aires: Ediciones Paidós.
- Papert, S. (1997). *La familia conectada: Padres, hijos y computadoras*, Buenos Aires: Emecé Editores.
- Rela A. y Sztrajman J. (1999a). *Física I*, Buenos Aires: Aique.
- Rela A. y Sztrajman J. (1999b). *Física II*, Buenos Aires: Aique.
- Rubinstein, J. y Tignanelli H. (2004a). *Física I: La energía en los fenómenos físicos*, Buenos Aires: Estrada.
- Rubinstein, J. y Tignanelli H. (2004b). *Libro de actividades, Física I: La energía en los fenómenos físicos*, Buenos Aires: Estrada.
- Vitale, B. (1988). Psycho-cognitive aspects of dynamic model-building in LOGO: A simple population evolution and predator/prey program. *J. Educational Computing Research* 4(3), pp. 227-252.
- Vitale, B. (1990). Una aproximación psicopedagógica a la enseñanza de un lenguaje de programación en la escuela secundaria. *Infancia y Aprendizaje* 50, pp. 94-101.
- Vitale, B. (1991). Computador na escola: um brinquedo a mais?. *CIÊNCIA HOJE* 13(77), pp. 19-25.
- Vitale, B. (1992). Processes: A dynamical integration of computer science into mathematical education. En *Learning Mathematics and Logo* (Hoyles C. And Noss R. Eds.). Cambridge: The MIT Press.
- Wirth, N. (1984). Algoritmos y estructuras de datos. *Investigación y Ciencia* 98, pp. 24-35.