Algoritmos y programación educativa

Guía de Algoritmos y Programación para Docentes

2023-08-30

Guía de Algoritmos y Programación para Docentes

> *En el 2007 se publicó la primera edición de la presente Guía para Docentes que tuvo una acogida muy positiva reflejada en sus cerca de 500.000 descargas; se decidió por esto actualizarla. En esta segunda edición se reforzó la importancia de programar computadores como medio para ayudar a los estudiantes a desarrollar pensamiento algorítmico, creatividad y habilidades para solucionar problemas. Otras mejoras fueron: incluir un apartado dedicado a creatividad y presentar todos los ejemplos con su respectivo código tanto en MicroMundos como en Scratch. Le invitamos a descargar el PDF.*

En el 2007 se publicó la primera edición de la presente Guía para Docentes que tuvo una acogida muy positiva reflejada en sus cerca de 500.000 descargas; se decidió por esto actualizarla. En esta segunda edición se reforzó la importancia de programar computadores como medio para ayudar a los estudiantes a desarrollar pensamiento algorítmico, creatividad y habilidades para solucionar problemas. Otras mejoras fueron: incluir un apartado dedicado a creatividad y presentar todos los ejemplos con su respectivo código tanto en MicroMundos como en Scratch. Le invitamos a descargar el PDF.

**ALGORITMOS Y PROGRAMACIÓN

EN LA EDUCACIÓN ESCOLAR**

|

https://eduteka.icesi.edu.co/pdfdir/AlgoritmosProgramacion.pdf

|

Guía de Algoritmos y Programación para docentes

-

Descargue la versión PDF (4.1MB, 96 Páginas) *

-

Visualice la versión navegable en pantalla

|

|

http://es.calameo.com/read/000170621dc582e35fd10

|

Programación de Computadores con Scratch

Cuaderno de Trabajo para estudiantes

(Tercera Edición)

-

Para ir al documento haga clic aquí.

|

- Para visualizar el documento usted debe tener instalado Acrobat Reader.

Si aún no lo tiene, haga clic aquí.

SOLUCIÓN DE PROBLEMAS MEDIANTE LA PROGRAMACIÓN

Existe actualmente un consenso general dentro de la comunidad educativa mundial sobre la necesidad de superar el tipo de enseñanza basada en la transmisión de contenidos para apuntarle en su lugar al desarrollo de capacidades. Investigaciones y estudios recientes proponen diversos conjuntos de habilidades que la educación debe fomentar para que los estudiantes puedan tener éxito en el mundo digital y globalizado en el que van a vivir [1]. Este planteamiento exige, sin dilaciones, implementar estrategias que contribuyan efectivamente en el desarrollo de esas habilidades planteadas como fundamentales para la educación en el Siglo XXI [1].

En la mayoría de conjuntos de habilidades propuestos figura la destreza para solucionar problemas; por esta razón, se requiere seleccionar estrategias efectivas para ayudar a que los estudiantes la adquieran. Para atender esta necesidad, la programación de computadores constituye una buena alternativa, siempre y cuando se la enfoque al logro de esta habilidad y no a la formación de programadores. Es importante insistir en esta orientación debido a que las metodologías utilizadas en educación básica [2] para realizar cursos de Algoritmos y Programación, son heredadas de la educación superior y muchos de los docentes que las utilizan se dedican a enseñar los vericuetos de lenguajes orientados a objetos, la mayoría de las veces, bajo el paradigma de la programación estructurada [3]. Hablar hoy de aprender a diseñar y construir aplicaciones (programas) complejas, implica una labor titánica que está fuera del alcance de la educación básica [4].

Por esta razón, en la educación básica es recomendable utilizar ambientes de programación como Logo [5] ya que son fáciles de utilizar y permiten realizar procedimientos [6] basados en estructuras básicas (secuencial, decisión y repetición), pero siempre conducentes a solucionar problemas. Solo en los grados de Educación Media [2] sería aconsejable introducir a los estudiantes a la programación orientada a objetos [4] mediante entornos de programación visuales y amigables como Alice [7, 8] o Processing [9].

Desde el punto de vista educativo, la programación de computadores posibilita no solo activar una amplia variedad de estilos de aprendizaje [10] sino desarrollar el Pensamiento Computacional. Adicionalmente, compromete a los estudiantes en la consideración de varios aspectos importantes para la solución de problemas: decidir

sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo y, monitorear sus propios pensamientos (metacognición) y estrategias de solución. Este último, es un aspecto que deben desarrollar desde edades tempranas. No debemos olvidar que solucionar problemas con ayuda del computador puede convertirse en una excelente herramienta para adquirir la costumbre de enfrentar problemas de manera rigurosa y sistemática, aun, cuando no se utilice un computador para solucionarlo.

PROPUESTA DE LA FUNDACIÓN GABRIEL PIEDRAHITA URIBE

Bajo este enfoque, desde el año 2004, se ha llevado a cabo un curso de Algoritmos y Programación con estudiantes de grados 4° y 5° en el Instituto Nuestra Señora de la Asunción (INSA). Durante este tiempo se han realizado ajustes buscando la mejor forma tanto de secuenciar los contenidos como de abordar la solución de problemas, en este caso, con el área de Matemáticas. Como resultado de esta experiencia, se ha evidenciado que cuando los estudiantes “programadores” resuelven problemas matemáticos retadores en un ambiente como Logo [5], este pone a prueba la real comprensión que tienen respecto a los conceptos matemáticos involucrados en las soluciones. Esto implicó que en INSA durante el desarrollo del proyecto, se asegurara que los estudiantes de primaria desarrollaran previamente competencias en comprensión lectora y en los temas básicos de matemáticas con los que iban a trabajar.

El presente trabajo, fruto de varios años de labor, reflexión, consulta y afinamiento, que ponemos a su disposición, consta de una **Guía dirigida a docentes** que tienen a su cargo el área de Informática en los grados 3° a 9° y de un **Cuaderno de Trabajo** enfocado a estudiantes de grados 4° y 5° (primaria). Estos recogen la experiencia con

la que se ha logrado integrar en el INSA la programación de computadores al desarrollo de habilidades para solucionar problemas. La **Guía** esta compuesta por cuatro unidades y aunque en INSA se utiliza con estudiantes de grados 4° y 5°, puede trabajarse con estudiantes de secundaria [2], basta con que el docente ajuste los ejemplos y ejercicios a las capacidades de sus estudiantes.

En la primera unidad se introduce un enfoque para educación básica orientado hacia el desarrollo de habilidades de pensamiento con ayuda del computador. En ella, se plantean las cuatro fases que componen el ciclo de programación que a su vez concuerdan con las operaciones mentales descritas por Polya [11] para resolver problemas matemáticos: Analizar el problema (entender el problema), Diseñar una solución algorítmica (trazar un plan), Traducir el algoritmo a un lenguaje de programación (ejecutar el plan) y Depurar el programa (revisar). Cabe destacar el marcado énfasis que se hace en la primera fase: Analizar el problema hasta lograr la mejor comprensión posible de este. Para ello, se lo debe formular claramente, especificar los resultados que se desean obtener, identificar la información disponible (datos), determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles (materia prima) en la información requerida (resultados solicitados). Al análisis de problemas, los textos universitarios consultados, aunque reconocen su importancia, omiten explicar la forma de trabajarlo en el aula y a esto le dedican únicamente unas pocas páginas.

La segunda unidad se concentra en los conceptos básicos requeridos para llevar a cabo la fase dos: Diseñar soluciones algorítmicas que permitan resolver problemas mediante pasos sucesivos y organizados en secuencia lógica (pensamiento algorítmico). Se tratan temas como ¿qué es un algoritmo?, formas comunes de representarlos (seudocódigo y diagrama de flujo) y conceptos básicos de programación (variable, constante, identificador, palabra reservada, contador, acumulador, tipos de datos, operadores y expresiones).

La tercera unidad atiende las fases dos y tres del ciclo de programación: diseñar un algoritmo y traducirlo a un lenguaje de programación. En esta se exponen las tres estructuras de control básicas conocidas como secuencial, iterativa (repetición) y condicional (decisión, selección). Adicionalmente, se explican los fundamentos de

programación tanto con Scratch, como en el *área de procedimientos* [6] de MicroMundos (versión de Logo desarrollada por LCSI [12]).

La última unidad está dedicada a la cuarta fase del ciclo de programación: Depurar procedimientos. Para un estudiante resulta muy difícil elaborar procedimientos [6] perfectos en los primeros intentos y la dificultad aumenta a medida que los problemas se vuelven más complejos. Después de traducir el algoritmo a un lenguaje de programación, el procedimiento resultante se debe probar y se deben validar los resultados (revisión). Este proceso se conoce como depuración y desde el punto de vista educativo estimula en los estudiantes la curiosidad, la perspectiva y la comunicación, además de promover valores como responsabilidad, fortaleza, laboriosidad, paciencia y perseverancia.

Por otra parte, los Anexos de la **Guía para docentes** están compuestos por los resúmenes de comandos de MicroMundos y de Scratch que se utilizan en los diferentes ejemplos, un esquema de los temas que se tratan y un plan de trabajo con la secuencia de contenidos para llevar al aula esta Guía, apoyada por una propuesta curricular. Propuesta completa planteada para llevarse a cabo durante un año escolar y secuenciada de manera que obedezca al orden descrito en el Anexo 3, fruto de la más reciente experiencia en INSA. Su objetivo final o Alcance es buscar que el estudiante utilice metodologías y estructuras secuenciales, iterativas y condicionales para analizar problemas, diseñar algoritmos, traducir algoritmos a un lenguaje de programación y depurar procedimientos sencillos con el fin de solucionar problemas. El objetivo no es que los estudiantes elaboren programas complejos sino que se concentren en la elaboración de procedimientos utilizando apropiadamente las capacidades mentales que esto implica y en el proceso desarrollándolas.

En resumen, el principal objetivo de esta propuesta no consiste en lograr que los estudiantes se conviertan en hábiles programadores, consiste más bien en darles la oportunidad de *enseñarle* al computador cómo realizar algo mediante procedimientos. En este orden de ideas, los ambientes de aprendizaje enriquecidos con computadores apoyan muy bien el supuesto de que la mejor manera de aprender es enseñar (muchos docentes afirman que solo cuando han tenido que explicar un tema a otros es cuando verdaderamente lo han entendido). En estos ambientes, los estudiantes

tratan de enseñar al computador lo que ellos deben aprender; autores como Arthur Luehrmann y Seymour Papert resaltan que el hecho de tratar de enseñar mejora procesos cognitivos y ayuda a desarrollar habilidades de expresión y de solución de problemas.

Una forma de lograr el manejo de estructuras y conceptos básicos de la programación es mediante instrucción acompañada de ejemplos abundantes, ejercicios retadores y actividades que requieran la aplicación de lo aprendido. Esta publicación también incluye un **Cuaderno de Trabajo** el cual es una ayuda didáctica que sirve de guía para enseñar Scratch en el aula mediante una serie de actividades dirigidas a estudiantes de grados 4° y 5°, cuyo grado de dificultad es incremental. Incluye ejercicios adicionales, algunos de ellos para realizarse antes de las actividades y los otros para trabajarse a su término; estos últimos sirven de refuerzo. La filosofía de este cuaderno se enmarca tanto en la Espiral de la Creatividad del Dr. Mitchel Resnick, como en la metodología para solución de problemas de George Polya.

Por último y como propuesta de solución a los escollos que se presentaron en INSA, vale la pena aclarar que la secuencia óptima para presentar los temas a los estudiantes es el que se encuentra en el Cuaderno de Trabajo. De la misma forma como algunos expertos aconsejan secuenciar la enseñanza de las matemáticas, la sucesión de contenidos de Programación de Computadores se debe planear de acuerdo a una estructura helicoidal, en la que los distintos temas se retomen en distintas ocasiones a lo largo del proceso de aprendizaje, de forma que el estudiante pueda comprender e interiorizar progresivamente dichos contenidos.

|

https://eduteka.icesi.edu.co/pdfdir/AlgoritmosProgramacion.pdf

|

Guía de Algoritmos y Programación para docentes

- Descargue la versión PDF (4.1MB, 96 Páginas) *
- Visualice la versión navegable en pantalla

|

|

http://es.calameo.com/read/000170621dc582e35fd10

|

Programación de Computadores con Scratch

Cuaderno de Trabajo para estudiantes

(Tercera Edición)

-

Para ir al documento haga clic aquí.

|

- Para visualizar el documento usted debe tener instalado Acrobat Reader.

Si aún no lo tiene, haga clic aquí.

OTROS ENLACES DE INTERÉS:

- Currículo de Algoritmos y Programación
- Currículo de programación con Scratch
- Currículo Diagramas de Flujo
- Manual de MicroMundos Pro elaborado por el Proyecto Teddi (PDF, 564KB, 19 Páginas)
- MicroMundos (LCSI)
- Paraíso de Logo

- Logo Foundation (ingles)
- FSM Logo
- Descarga de MSW Logo (gratuito)
- Programar y crecer con MicroMundos (Costa Rica)
- Todo Logo
- Logo in Your Browser
- Light-Bot

NOTAS:

[1] Ver los documentos: “Seis elementos fundamentales para la educación en el siglo XXI” , “Estándares educativos en TIC para estudiantes”.

[2] En Colombia, la Educación Básica se divide en Primaria (grados 1° a 5°) y Secundaria (grados 6° a 9°). La Educación Media corresponde a los grados 10° y 11°.

[3] La programación estructurada surgió a finales de los años sesenta como un paradigma de programación que no solamente daba lugar a programas fiables y eficientes, sino que además facilitaba su comprensión posterior. Consiste en una técnica de diseño y modelado de aplicaciones informáticas por medio de la cual la lógica de un programa se dispone de manera funcional y jerárquica en módulos lógicos. Además, los procedimientos y los datos se encuentran separados. El principal inconveniente de este método de programación, es que se obtiene un bloque único de programa cuyo manejo puede resultar problemático si llega a ser demasiado grande. Para solucionar este inconveniente se propone la programación modular, que permite definir módulos interdependientes programados que se pueden compilar por separado.

[4] La programación orientada a objetos (POO u OOP según su sigla en inglés), se basa en que una situación puede ser moldeada, y los problemas que se presenten en su interior resueltos, mediante la identificación de los objetos que intervienen en ella y la forma como se comunican entre sí. Es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La POO expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar. De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos). A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos (de una misma clase o de clases distintas), y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos).

[5] En palabras de Seymour Papert creador de este lenguaje de programación, "Logo es un lenguaje de programación más una filosofía de educación" y esta última se caracteriza con suma frecuencia como "constructivismo" o "aprendizaje a través del descubrimiento". Por su parte, Micromundos es una poderosa herramienta multimedia basada en Logo, dirigida a niños de 6 a 12 años, que permite crear proyectos incorporando películas, fotos, sonidos, gráficos, textos y animación. Esto lo convierte en una poderosa herramienta de trabajo para realizar proyectos educativos basados en el constructivismo.

[6] En programación se define “Procedimiento” como una subrutina o subprograma en forma de algoritmo que permite resolver una tarea específica. En MicroMundos Pro, los procedimientos contienen instrucciones que se inician con el comando “para” y que el computador ejecuta automáticamente, una tras otra, hasta encontrar el comando “fin”. Esto demanda que los estudiantes planifiquen, formulen hipótesis y anticipen qué sucederá antes de determinar los comandos que conforman un procedimiento

para poderlo escribir, ejecutar y comprobar si produce el resultado esperado. De esta manera, Logo promueve lo que Piaget denominó “la conquista de la difícil conducta de la reflexión” que se inicia a partir de los siete u ocho años cuando el niño deja de actuar por impulso y empieza a pensar antes de proceder.

[7] El Proyecto Alice está enfocado en proveer una herramienta gratuita para iniciar a los estudiantes de educación básica secundaria y media en la programación de computadores. http://www.alice.org/

[8] Ver entrevista a Guillermo Londoño, director del Programa de Ingeniería de Sistemas de la Universidad Icesi, y sus comentarios acerca de la implementación de un Taller de KPL dirigido a estudiantes de grados 10° y 11°.

[9] Processing es un lenguaje de programación gratuito y de código abierto que permite trabajar con imagen, animación y sonido. Muy utilizado por estudiantes, artistas, diseñadores, arquitectos, investigadores y aficionados a la programación tanto en ambientes de aprendizaje como en la elaboración de prototipos.

[10] ver el artículo de Gary Stager “En pro de los computadores”.

[11] Los aportes de George Polya (1887 - 1985) a las matemáticas incluyen más de 250 documentos y tres libros que promueven un acercamiento al conocimiento y al desarrollo de estrategias para la solución de problemas. Su famoso libro “Cómo Plantear y Resolver Problemas”, traducido a 15 idiomas, presenta su método de cuatro pasos junto con estrategias específicas útiles en la solución de problemas. Según Polya, cuando se resuelven problemas, intervienen cuatro operaciones mentales: 1) Entender el problema, 2) Trazar un plan, 3) Ejecutar el plan (resolver) y 4) Revisar.

[12] LCSI, empresa ubicada en Montreal (Canadá), desarrolla programas para la enseñanza en el método educativo-constructivo del lenguaje Logo como MicroMundos. LCSI reúne un grupo interdisciplinario que incluye diseñadores, maestros, ingenieros de sistemas y escritores para que desarrollen productos basados en la filosofía del aprendizaje constructivista de Logo. Es miembro primordial de este grupo el Dr. Seymour Papert, fundador del Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusetts y principal creador del lenguaje Logo.

CRÉDITOS:

Guía de Algoritmos y Programación para docentes y Cuaderno de Trabajo para programación con Scratch elaborados especialmente para EDUTEKA por Juan Carlos López García, Editor de Eduteka.

*Publicación de este documento en EDUTEKA: Mayo 01 de 2007.

Última modificación de este documento: Febrero 01 de 2010.*

Descargado desde eduteka.co · edukatic.co/articulos/ver.php?id=904