

¿Cómo se fomenta el interés por la Programación de Computadores?

Por EDUTEKA · 2007-05-01

Entrevista a Guillermo Londoño, Director de Ingeniería de Sistemas, Universidad Icesi, quien hace recomendaciones para iniciar de la mejor manera a estudiantes de Educación Media en el fascinante mundo de la programación. Describe además el Taller de lenguaje KPL que le permite cumplir con este propósito.

¿CÓMO SE FOMENTA EL INTERÉS

POR LA PROGRAMACIÓN DE COMPUTADORES?



[Guillermo Londoño Acosta](#),

Director del programa de Ingeniería de Sistemas

de la Universidad Icesi.

EduTEKA entrevistó a [Guillermo Londoño Acosta](mailto:glondono@icesi.edu.co), <mailto:glondono@icesi.edu.co>

Director del programa de Ingeniería de Sistemas de la Universidad Icesi, y profesor en el área de Modelado de Sistemas de Información. El profesor Londoño ha demostrado gran interés por acercarse a las Instituciones Educativas de Básica y Media, desde la

Universidad, tanto para ayudarles a mejorar la competencia Informática de sus egresados, como para implementar estrategias que promuevan el interés por el estudio de las Ingenierías. Fruto de la primera inquietud es el “[Modelo Curricular](https://eduteka.icesi.edu.co/curriculo2/CurriculoInteractivo.php)” <https://eduteka.icesi.edu.co/curriculo2/CurriculoInteractivo.php> construido colectivamente por docentes de Informática de 13 instituciones educativas, oficiales y privadas, para enseñar esta asignatura, con la coordinación conjunta de la Fundación Gabriel Piedrahita Uribe y de la Universidad Icesi.

EDUTEKA (E): *¿Podría contarnos cuál es el panorama actual de la programación de computadores en las Instituciones Educativas de Básica y Media?*

GUILLERMO LONDOÑO ACOSTA (GLA): Los adolescentes están perdiendo interés por la programación de computadores, la consideran una temática complicada y poco motivadora. Es necesario iniciar en la programación a las nuevas generaciones de una manera fácil, divertida y muy gráfica, con herramientas que oculten la complejidad y permitan conocer y aprender a utilizar esos elementos básicos.

En los años 80 los jóvenes tenían la oportunidad de conocer el mundo de la programación de computadores, a través de un lenguaje de programación para aficionados, fácil de aprender y de utilizar, de propósito general y con características avanzadas pero simples para los principiantes; aunque los expertos lo criticaban por su lentitud y sencillez. Este lenguaje llamado BASIC, se adecuaba muy bien a las computadoras personales de la época, y era el lenguaje estándar usado por los adolescentes como mecanismo para introducirse en el mundo de la programación. Un adolescente podía experimentar y conocer, de forma autodidacta, los elementos básicos de la programación.

En los años 90, la complejidad de los computadores personales y la llegada de los ambientes gráficos, exigían nuevas características a los lenguajes de programación, que BASIC no tenía. Lenguajes ya no para aficionados como C, C++ y VisualBasic permitían desarrollar aplicaciones complejas, que explotaban toda la potencialidad de las computadoras personales y de las interfaces gráficas de usuario. Los computadores personales se convirtieron en herramientas usadas principalmente para ejecutar aplicaciones escritas por otros; la programación empezó a ser menos

importante para la creciente mayoría de usuarios y de poco interés para los adolescentes.

Hoy en día, lenguajes como Java, C# y VBasic.NET, con sus populares entornos de desarrollo Eclipse, JDeveloper, Visual Studio.NET, etc., no son adecuados para que un adolescente, estudiante de secundaria, se inicie en el mundo de la programación. Actualmente la carencia de un lenguaje básico estándar, ha conducido a la eliminación de la enseñanza de la programación en muchos colegios, y aquellos que lo hacen con lenguajes complejos como C, Java, etc. están logrando que muchos jóvenes pierdan todo interés por la programación.

E: ¿Esta situación qué efectos tiene en los estudiantes que desean programar?

GLA: Nuestros adolescentes se han convertido sólo en usuarios de paquetes como Office; algunos piensan que un Ingeniero de Sistemas se dedica a estudiar Office y otros paquetes en profundidad, perciben la informática como el uso de aplicaciones, pues desconocen el poder de la programación para comunicarse con los computadores y lograr que hagan lo que uno quiere.

La diferencia entre dar instrucciones a una persona y dar instrucciones a un computador está en la complejidad de cada instrucción y en el lenguaje utilizado. El computador no puede suponer, ni imaginar nada; las instrucciones que le damos son más elementales, precisas, en un estricto orden y sometidas a reglas de un lenguaje más restrictivo que los lenguajes utilizados por los humanos. No estamos acostumbrados a dar instrucciones en forma tan lógica, precisa, ordenada y con un lenguaje tan restrictivo; este es un excelente ejercicio mental para un adolescente que desee aprender a programar.

Aprender a programar permite también desarrollar la capacidad de abstracción para entender un problema, diseñar una solución algorítmica e implementarla en una computadora. Este mecanismo de solución de problemas es algo de mucho valor en el desarrollo intelectual de un adolescente; no importa si eventualmente decide llegar a ser un ingeniero de sistemas o no; estas habilidades y conocimiento lo beneficiarán por el resto de su vida, pues tendrá que desenvolverse en un mundo inmerso en la

tecnología. Por otra parte, el conocimiento de los elementos básicos de la programación debe ser del dominio de todo adolescente que desee explotar la potencialidad de herramientas de productividad personal como la Hoja de Cálculo (Microsoft Excel, Calc, etc).

E: ¿Podría comentarnos cuáles son los problemas más frecuentes que han encontrado ustedes en la universidad con los estudiantes que ingresan a Ingeniería de Sistemas y a quienes les han enseñado en el colegio Algoritmos y Programación?

GLA: Muchos colegios en sus cursos de programación tienen como objetivo aprender un lenguaje y construir programas, no ven la programación como un medio para que el estudiante desarrolle su capacidad de pensamiento algorítmico. Por lo general se concentran en las herramientas, sintaxis de un lenguaje, y utilizan la orientación a procedimientos [PRO] como paradigma de programación.

Los cursos de programación en la Universidad Icesi, enfatizan más la construcción de algoritmos y utilizan el paradigma de programación orientado a objetos [OOP]. Este cambio de objetivo y de paradigma hace que los estudiantes que han estudiado programación en el colegio tengan mayores dificultades en la universidad con los cursos de algoritmos y programación.

E: De acuerdo con su experiencia, ¿qué recomendaciones haría usted para iniciar a los estudiantes de Básica Secundaria y de Media en el mundo de la programación?

GLA: Los adolescentes hace 20 años disponían de tecnologías en 2D basadas en texto y gráficos y para ellos tenía mucho sentido la frase “Una imagen vale más que mil palabras”. Hoy en día, los jóvenes utilizan diversos medios para obtener información, aprender y entretenerse; los adolescentes de hoy tienen una mente muy gráfica y usan tecnología basada en texto, imágenes, sonido, animación e interactividad. Pienso que la frase más adecuada para los jóvenes de hoy es: una animación interactiva vale más que mil imágenes.

Desafortunadamente, seguimos enseñando los principios básicos de la programación utilizando el texto como único medio, lo cual es frustrante y poco motivador para una

mente gráfica como la de los jóvenes. Hemos tenido en los últimos años pocas innovaciones en la enseñanza y el aprendizaje de estos principios, los profesores en los colegios y en las universidades son reacios a cambiar su forma de enseñar algoritmos y programación. Afortunadamente ya están apareciendo entornos de programación modernos como Alice [ALICE], desarrollado en Carnegie Mellon, o como KPL [KPL] desarrollado bajo la dirección de Jonah Stagner y sus colegas, Jon Schwartz, antiguo director de programación de Microsoft, y Walt Morrison, antes ingeniero de NCR.

Recomiendo a los profesores, tanto de los colegios como de las universidades, empezar a conocer y a utilizar estos entornos gráficos con sonido, animación, interactividad y en 3D como herramientas para introducir a sus estudiantes al mundo de los algoritmos y la programación.

E: ¿Podría explicarnos brevemente en qué consiste el KPL?

GLA: KPL [KPL] es un entorno de programación moderno compuesto por un editor, un compilador, un depurador, un entorno gráfico de ejecución y una biblioteca de estructuras, funciones y procedimientos que permiten manipular fácilmente imágenes, sonidos y dispositivos periféricos como el ratón y el teclado. La versión 2 de KPL, liberada a finales del 2006 y llamada *Phrogram*, fue desarrollada sobre el framework “.NET” [NET], incluye no solo un potente motor 2D como el de la versión original, sino que introduce un motor 3D basado en DirectX 9.0c [DIRECTX], el cual permite el uso de controles y “joysticks” para el manejo de los juegos. Y en un futuro cercano, los juegos desarrollados en KPL2 se podrán ejecutar en el Xbox 360 [XBOX].



El material del Taller está publicado en <http://www.icesi.edu.co/colegios/kpl/index.htm>

E: ¿De dónde surgió la idea de realizar en Icesi un Taller de KPL y de generar un sitio Web para apoyar este Taller?

GLA: En el mes de diciembre de 2005 invitamos estudiantes de grados 10° y 11°, de varios colegios de Cali, a la charla “Aprender Algoritmos Desarrollando Juegos de Video”. Esta invitación tuvo tanta acogida, que decidimos construir un material que permitiera a los estudiantes de los colegios, conocer los elementos básicos de la programación construyendo juegos de video.

La programación se puede utilizar para resolver problemas matemáticos, o problemas relacionados con el manejo de información en una organización, o para animar imágenes en la pantalla del computador al construir aplicaciones educativas o juegos de video, o en muchos otros campos. Todos estos dominios de aplicación requieren los mismos elementos básicos de programación. Empleamos los juegos de video para aprender a programar, puesto que este es un tema de mucho interés para los adolescentes; además, les permite ver resultados gráficos e inmediatos de sus programas y estimula su creatividad, a la vez que se les enseña elementos básicos como el concepto de variable, operador, tipo de datos, instrucción de selección y de repetición, arreglo, estructura, método, función y clase.

E: ¿Cuántos de estos Talleres se han realizado hasta ahora y qué logros se han alcanzado con ellos?

GLA: Hemos realizado 6 talleres con estudiantes de grado 10 y 11, cinco abiertos a los colegios en general y uno dirigido sólo a los estudiantes de grado 11 del colegio Jefferson de la ciudad de Cali. Este último fue un taller de 40 horas, dictado en las salas de la Universidad Icesi durante un período de 2 meses, en el que los estudiantes asistían 3 horas por semana.

E: De acuerdo con su experiencia y con los Talleres antes mencionados, ¿cuál es el lenguaje de programación apropiado para realizar este tipo de actividades en el colegio?

GLA: El lenguaje debe tener una sintaxis bastante simple, como en KPL [KPL], o el entorno de programación debe permitir programar arrastrando y pegando, como en Alice [ALICE], para evitar desviar la atención del estudiante a la continua corrección de

errores de sintaxis. El estudiante debe concentrarse en el diseño algorítmico de la solución al problema y no en la complejidad del lenguaje.

NOTAS DEL EDITOR:

[PRO] Programación orientada a procedimientos: Paradigma utilizado para analizar un problema y diseñar una solución mediante la descomposición en funciones. La función o procedimiento es la unidad básica de análisis, diseño y construcción de los programas.

[OOP] Programación orientada a objetos: Paradigma de programación utilizado para analizar un problema y diseñar una solución mediante la identificación y clasificación de los objetos del dominio el problema. Las clases de objetos son las unidades básicas de análisis, diseño y construcción de los programas.

[ALICE] 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/>

[KPL] El Lenguaje de Programación para Niños (KPL, por sus siglas en inglés Kid's Programming Language) permite a los niños aprender a programar un computador de forma sencilla. KPL se puede descargar gratuitamente; es divertido y facilita la programación de videojuegos con gráficas y sonidos.

<http://www.kidsprogramminglanguage.com/espanol/index.php>

[NET] “.NET” es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software independiente del sistema operativo y que permita un rápido desarrollo de aplicaciones, en respuesta al creciente mercado de los negocios en entornos Web y a la competencia de la plataforma Java de Sun Microsystems. Con “.NET”, Microsoft intenta desarrollar una estrategia que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de productividad. La idea fundamental de Microsoft .NET es un cambio de enfoque en lo que es la informática, pasando de un mundo de aplicaciones, sitios Web y dispositivos aislados a una infinidad de computadores, dispositivos, transacciones y servicios que se conectan directamente y trabajan en conjunto para ofrecer soluciones más amplias y ricas en contenido. <http://www.microsoft.com/latam/net/introduccion/>

[DIRECTX] DirectX es una colección de Interfaces de Programación de Aplicaciones (Application Programming Interface – API por su sigla en inglés) creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows. Las bibliotecas de DirectX eran distribuidas originalmente por los desarrolladores de juegos con cada programa, pero más tarde Microsoft empezó a distribuir gratuitamente el conjunto de herramientas de desarrollo de DirectX, cuya última versión de es la 10. <http://directxworld.altervista.org/>

[XBOX] Xbox es un sistema de videojuegos diseñado y desarrollado por la empresa Microsoft. La primera unidad de esta consola fue puesta en venta el 15 de noviembre de 2001 y utiliza una versión reducida del sistema operativo Windows 2000, también desarrollado por Microsoft. <http://www.xbox.com/es-ES/>

CRÉDITOS:

Entrevista concedida especialmente a EDUTEKA por Guillermo Londoño Acosta, director del programa de Ingeniería de Sistemas de la Universidad Icesi, y profesor en el área de Modelado de Sistemas de Información.

El profesor Londoño es Físico de la Universidad del Valle; magíster en Física del Instituto Venezolano de Investigaciones Científicas (IVIC), Caracas, Venezuela; profesional asociado a investigación del Instituto Venezolano de Investigaciones Científicas (IVIC), Caracas, Venezuela; asistente de investigación de la Universidad de Wisconsin, Milwaukee, Estados Unidos; y magíster en Ingeniería de Sistemas de la Universidad del Valle. Además, realizó cursos de especialización en Ingeniería del Software en la Universidad Politécnica de Madrid.

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

Última modificación de este documento: Mayo 01 de 2007.*