

RESUMEN-Alternativas para la enseñanza de pseudocódigo y diagrama de flujo

Landa Merino Humberto Brandon

Este artículo de Ana del Prado y Nancy Lamas, publicado en 2014 en una revista iberoamericana, trata sobre cómo mejorar la enseñanza de Fundamentos de Informática en la Universidad Nacional de Catamarca, Argentina. La idea principal es que los alumnos primero aprendan a resolver problemas con pseudocódigo y diagramas de flujo, antes de meterse de lleno en un lenguaje de programación. Los autores resaltan que lo más importante son los algoritmos, no la computadora ni la sintaxis, y que empezar enseñando código es un error común.

En la introducción mencionan el problema: en la Tecnicatura Industrial, muchos estudiantes no logran diseñar algoritmos correctamente. Para resolver esto proponen usar software libre como PSeInt (para pseudocódigo) y FreeDFD (para diagramas de flujo), además de integrar actividades y autoevaluaciones con la plataforma Moodle. Citan a Winslow (1996), quien dice que programar implica entender el problema, diseñar la solución, traducirla a código y luego depurar.

Las herramientas analizadas son:

- PSeInt: pensado para principiantes, usa pseudolenguaje en español, tiene ayudas como autocompletado, genera diagramas de flujo y funciona en Windows, Linux y Mac. Permite detectar errores de sintaxis, seguir el flujo paso a paso y cuenta con manuales y foros.
- FreeDFD: creado en Colombia, se enfoca en diagramas de flujo gráficos. Permite ejecutar y depurar algoritmos, manejar tipos de datos (reales, cadenas, arreglos), pero solo corre en Windows (o con Wine en Linux). Tiene ejemplos, aunque es menos completo que PSeInt.

Para compararlas usaron 10 criterios como usabilidad, multiplataforma, detección de errores, facilidad de instalación, entre otros. Con el ejemplo de "Obtener los primeros 100 números pares", comprobaron que PSeInt ofrece más ventajas, ya que es más amigable, motiva a los estudiantes al mostrar el proceso paso a paso, y es multiplataforma. En cambio, FreeDFD tiene limitaciones como no mostrar la traza completa del algoritmo, no tener todas las estructuras cíclicas y presentar problemas de diseño visual.

Pros de PSeInt: evita instrucciones incorrectas, ayuda a corregir errores antes de entregar, motiva porque es interactivo.

Contras: los diagramas no siempre usan notación estándar y aún pueden aparecer errores de sintaxis si no se usa bien.

Pros de FreeDFD: permite expresiones más complejas y también detecta errores.

Contras: interfaz menos práctica, no muestra bien todo el flujo, y depende casi exclusivamente de Windows.

Ellos recomiendan SelInt, ya que favorece el desarrollo del pensamiento lógico y algorítmico sin depender de un lenguaje específico. Además, al combinarlo con Moodle, se fomenta el aprendizaje cooperativo y más organizado. En pocas palabras, enseñar lógica antes que sintaxis forma mejores programadores.

Porque elegí el texto:

Elegí este texto porque me identifico con la problemática que plantea: muchas veces en la escuela nos enseñan directamente a programar en un lenguaje, pero no nos enseñan a pensar como programadores. A mí me pasó que sabía usar comandos, pero no entendía cómo estructurar un algoritmo desde cero. Este trabajo me pareció útil porque propone una solución práctica con herramientas que podemos usar como estudiantes para entrenar la lógica antes de meternos con lenguajes más complejos.