



Conciencia Tecnológica

ISSN: 1405-5597

contec@mail.ita.mx

Instituto Tecnológico de Aguascalientes  
México

Pinales Delgado, Francisco Javier; Patlán Balandrán, Félix  
Eficiencia de las metodologías de programación  
Conciencia Tecnológica, núm. 18, diciembre, 2001, pp. 24-26  
Instituto Tecnológico de Aguascalientes  
Aguascalientes, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=94401808>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica  
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

## Eficiencia de las Metodologías de Programación

Francisco Javier Pinales Delgado.  
e-mail: [fpinales@correo.uaa.mx](mailto:fpinales@correo.uaa.mx)

Félix Patlán Balandrán  
e-mail: [fpatlan@correo.uaa.mx](mailto:fpatlan@correo.uaa.mx)

**Centro de Estudios Tecnológicos industrial y de servicios No. 155**  
Academia de Programación  
Av. Perseo S/N.  
Fracc. Jesús Terán  
Aguascalientes, Ags.

### Resumen.

Durante quince años de actividad docente impartiendo materias del área de programación, hemos visto como muchos jóvenes inician una carrera en la especialidad de computación, con la ilusión de conquistar el mundo utilizando una computadora. Sin embargo, se enfrentan a una realidad: No cuentan con los conocimientos, aptitudes y habilidades requeridas para una carrera de esta área. Por lo tanto, se ven obligados a abandonar la carrera en los primeros semestres. Esto originó realizar un estudio, con el fin de diagnosticar los beneficios que se pudieran tener al utilizar una metodología de programación determinada en el proceso enseñanza-aprendizaje, que permitiera a los estudiantes desarrollar una lógica apropiada para la solución de problemas en computadora. Las herramientas que se aplicaron son: **a) pseudocódigo b) diagramas de flujo c) diagramas de Nassi Shneiderman (N/S)**. Con los resultados obtenidos podemos concluir que sin duda alguna la metodología utilizada es muy importante y la elección muchas de las veces dependerá del gusto y dominio del instructor, pero es más importante presentar al alumno alternativas de cómo solucionar los problemas mediante diferentes metodologías de programación.

**Palabras Clave:** Pseudocódigo, Diagramas de flujo, Diagramas de Nassi Shneiderman.

### Introducción.

En la actualidad resulta muy común el interés por parte de los alumnos ingresar a los CETis y CBTis y estudiar la especialidad de computación, para posteriormente seguir una carrera en el área de Sistemas de Información, o bien, en Ciencias de la Computación. Esto debido al auge que se le ha dado a la computación en todas las áreas del conocimiento humano. Como consecuencia que el escoger computación resulte muy atractivo para la mayoría de las personas, las cuales han visto las facilidades y beneficios que se tienen con el uso de una herramienta como es la computadora, la que permite desarrollar una gran cantidad de trabajos con mínimo esfuerzo y de gran calidad.

Aunado a lo anterior, para muchas personas el uso de la computadora se ha vuelto una necesidad, lo cual motiva a muchas de ellas a estimular a sus hijos, parientes o amigos a que realicen una carrera relacionada con la computación, dejando de lado la consideración que para cursar este tipo de carreras se requiere de ciertos conocimientos, aptitudes y habilidades por parte de los estudiantes, que les permita desarrollarse y sobresalir dentro del área de la informática.

En estos planteles las materias de programación de los primeros semestres en la carrera de computación, presentan un alto índice de reprobación, y por consiguiente, la deserción de los educandos. Por lo anterior, es necesario buscar los mecanismos didácticos que ayuden a los educandos a desarrollar una lógica adecuada para el planteamiento de problemas de programación, el análisis y diseño de su tentativa solución. Esto se puede lograr posiblemente por medio del uso de una metodología adecuada, aplicada en el proceso de enseñanza-aprendizaje para este tipo de materias. De esta manera se formarán los cimientos apropiados para que el alumno continúe una carrera profesional en el área de computación con un buen soporte de lógica de programación, evitando con esto, cualquier tropiezo en el logro de su meta.

Este mismo problema se presenta también a nivel superior, en las materias que tienen que ver con el diseño de programas para computadora [1]. También está presente y palpable en la industria ya que al realizar estudios en el Estado de Aguascalientes en "Factores Críticos en el área de Informática", arrojaron que el factor: Mejoramiento de la calidad de desarrollo

de software, se encuentra en cuarto lugar de importancia, ya que tiene que ver con el uso de una metodología de programación adecuada [2]. El problema no es ajeno para otros países, como España, que han dedicado gran cantidad de tiempo en la búsqueda de una solución al problema mediante el diseño de nuevas metodologías [3].

Por tal motivo, es necesario tratar de establecer qué o cuáles metodologías son las óptimas en el proceso de enseñanza aprendizaje para el diseño de programas para computadora. Sin duda alguna existen muchas herramientas, pero dentro de las más conocidas se encuentran: Árboles de decisión, pseudocódigo, diagramas de flujo y diagramas de Nassi Shneiderman (N/S), entre otras.

A continuación se muestran los resultados obtenidos de un estudio de diagnóstico sobre cuál es el impacto en el aprendizaje y aprovechamiento en alumnos que se les presentaron diferentes herramientas de programación para la solución y diseño de problemas por computadora. Las herramientas que se utilizaron fueron: a) pseudocódigo, b) diagramas de flujo y c) diagramas de Nassi Shneiderman (N/S) por ser de las más conocidas.

Donde Algoritmo, se puede definir como: secuencia de pasos que describe la solución de un problema dado, de forma clara y detallada. Al Diagrama de flujo se le puede definir como: Secuencia de pasos que describe gráficamente la solución de un problema dado. Y los Diagramas N/S, al igual que un diagrama de flujo, solo que con las flechas omitidas y cajas o bloques contiguos [4].

### Desarrollo Del Estudio.

Para obtener los resultados del estudio realizado, se hizo lo siguiente:

- 1) Se formó un grupo con una muestra de cuarenta alumnos voluntarios de nuevo ingreso a la carrera de Licenciatura en Informática de la Universidad Autónoma de Aguascalientes, en el semestre Agosto-Diciembre de 1999. Suponiendo que la realización del experimento sería tomado por los participantes con más seriedad por ser alumnos de nivel licenciatura, que si la muestra hubiera sido formada con alumnos de nuevo ingreso a un nivel medio superior. Mediante una encuesta determinaron las condiciones iniciales del grupo en lo que respecta a conocimientos previos de las herramientas de programación, obteniendo los siguientes resultados:
  - a. En cuanto al dominio de alguna de las herramientas, la mayoría de los alumnos contestó que no tenía los conocimientos suficientes de ninguna de las herramientas presentadas, aunque había algunos alumnos con conocimientos muy

generales o no eran lo suficientemente sólidos para considerarlos como óptimos. En la tabla 1 se muestran los resultados obtenidos.

DOMINIO DE ALGUNA METODOLOGIA	% ALUMNOS.
MUY POCO O NADA	80%
REGULAR	16%
MUY BIEN	4%

Tabla 1) Nivel de dominio de herramientas de programación por los alumnos encuestados.

- b. En lo que respecta a herramientas conocidas por los alumnos encuestados se obtuvieron los resultados mostrados en la Tabla 2.

HERRAMIENTAS CONOCIDAS	% ALUMNOS
PSEUDOCÓDIGO	36%
DIAGRAMAS DE FLUJO	40%
DIAGRAMAS N/S	15%
OTRAS	2%

Tabla 2) Porcentajes de alumnos que conocían alguna de las herramientas de programación.

- 2) Se capacitaron a todos los alumnos de la muestra sobre definiciones, y reglas a seguir, en cada una de las herramientas del diseño de programas que fueron elegidas para el análisis.
- 3) Se trabajo en talleres para la solución de problemas utilizando las tres herramientas, primero por separado cada una, y posteriormente, combinándolas, en ocasiones mediante la inducción de resolverlo mediante alguna metodología en especial y en otras con la elegida por el estudiante.
- 4) Después de realizar la solución de un gran número de problemas, utilizando las tres herramientas se aplicó un examen de diagnóstico, el que contemplaba evaluar dos aspectos: a) La facilidad para interpretar un problema resuelto con alguna metodología; b) la solución de un problema. Para esto se utilizaron problemas con el mismo grado de dificultad para cada una de las herramientas obteniendo los resultados mostrados en la Tabla 3, al final del artículo.

Como se pudo observar en la Tabla 3, los alumnos pudieron “interpretar” los problemas de manera más eficiente, utilizando la herramienta de Diagramas de Flujo, con menos eficiencia y de manera semejante las otras dos herramientas utilizadas. El porcentaje más alto en que fallaron los alumnos fue en la herramienta de pseudocódigo, pero de manera no muy significativa. En lo que respecta al “como resolver” el problema, se puede ver que para diagramas de flujo también se tiene mejor desempeño, pero con un menor margen de diferencia con respecto a las otras herramientas empleadas.

Donde el elegir una metodología dependerá posiblemente en la mayoría de los casos del dominio que tenga el instructor y de sus conocimientos. Ya que es probable que muchos de los instructores que imparten las materias que tienen que ver con la enseñanza de una metodología de programación, no dominen más de dos herramientas y sean las que normalmente transmiten, limitando de alguna manera el “pensamiento creativo”, que deben desarrollar los estudiantes y profesionales de computación.

### Conclusiones.

Después de ver los resultados obtenidos en el experimento de diagnóstico, podemos concluir que, los logros obtenidos son bastante satisfactorios dado que el porcentaje no aceptable más alto es de un 18.33%, lo cual indica que con el entrenamiento adquirido, un poco más del 80% de los alumnos obtuvo la lógica necesaria para resolver los problemas planteados de manera eficiente, donde se puede notar que con las herramientas gráficas se tiene un ligero mejor desempeño, que utilizando pseudocódigo.

El hecho de que durante el entrenamiento, un problema se realizara de tres maneras diferentes, permitió a los alumnos afianzar el conocimiento adquirido con una metodología y resolviendo el problema utilizando otra, nos conduce a pensar que más que el escoger una específica, es más conveniente presentarle a los alumnos diferentes alternativas de solución, y de diferentes casos que les permita

aprender a identificar “el qué hacer” para solucionar un problema y después identificar “el cómo hacerlo”.

HERRAMIENTA	% DE EFICIENCIA EN CADA UNA DE LAS HERRAMIENTAS					
	INTERPRETAR			RESOLVER		
	EXCELENTE	REGULAR	MAL	EXCELENTE	REGULAR	MAL
PSEUDOCODIGO	33.33	44.17	18.33	75.00	8.33	12.50
DIAGRAMAS DE FLUJO	50.00	45.83	4.17	87.50	4.17	8.33
DIAGRAMS N/S	33.33	50.00	12.50	75.00	8.33	12.50

Tabla 3. Porcentajes de eficiencia en interpretación y resolución de problemas planteados en cada una de las herramientas de diseño de algoritmos, utilizadas por los alumnos de la muestra.

### Referencias.

- [1] Universidad Autónoma de Aguascalientes (1995), Evaluación 1995, Dpto. de Informática y Estadística.
- [2] Patlán y Macías (1988), “Identificación de Factores Críticos en el Área de Informática en el Estado de Aguascalientes Para Definir el Perfil de Carreras Profesionales”. UAA.
- [3] Joyanes, Rodríguez, y Fernández, (1997), Fundamentos de Programación, Mc. Graw Hill, Madrid España.
- [4] Joyanes Aguilar, (1993), Metodología de la Programación, Diagramas de Flujo algoritmos y programación estructurada, Mc Graw Hill, México D.F.