



aprenderaprogramar.com

Diseño de algoritmos en programación: del pseudocódigo al programa. (CU00224A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

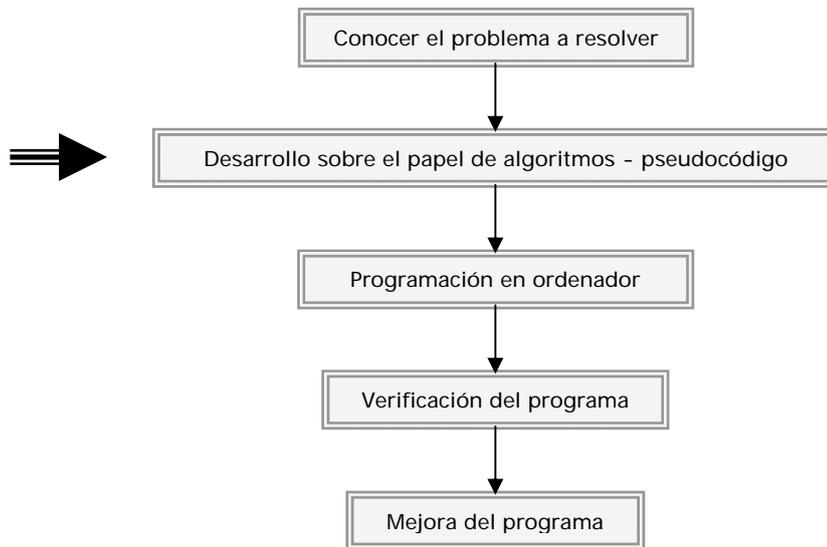
Autor: Mario R. Rancel

Resumen: Entrega nº 23 del Curso Bases de la programación Nivel II

24

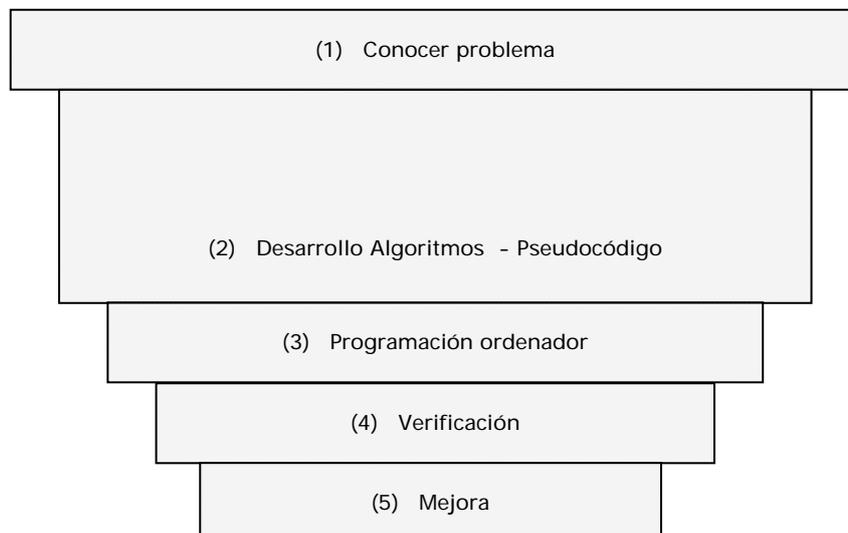
INTRODUCCIÓN AL DISEÑO DE ALGORITMOS EN PROGRAMACIÓN

Recordemos ahora el esquema de pasos lógicos para realizar un programa que es habitual estudiar cuando empezamos a hablar de fundamentos de programación, y que nos permite saber dónde estamos.

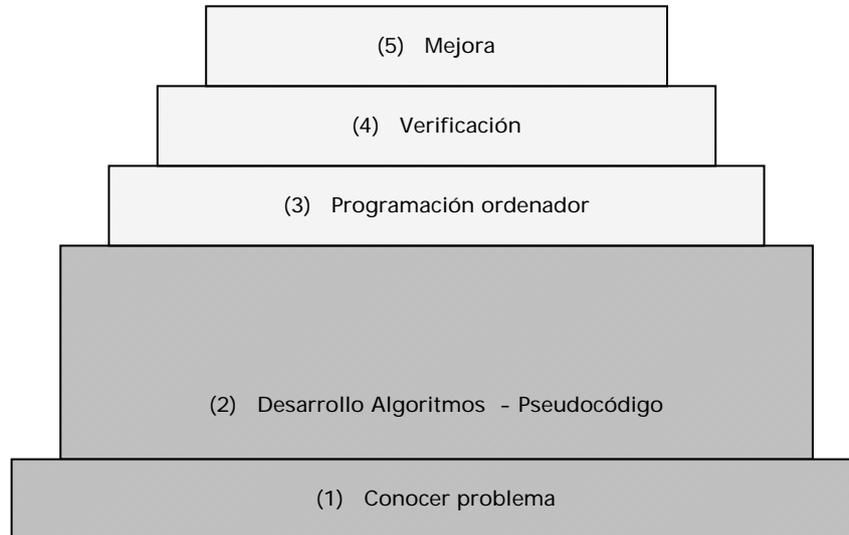


Si comparamos esta situación con la que constatábamos en etapas anteriores comprobaremos que ¡No nos hemos movido! Bueno, esto de “no nos hemos movido” suena un poco decepcionante y no es demasiado justo. Si recordamos la analogía con la construcción de un edificio, vendría a significar que no hemos empezado a construir el edificio. En cambio, tenemos el proyecto muy avanzado, lo cual es sin duda una satisfacción.

Si consideramos el esquema anterior no sólo con sus pasos, sino también con los volúmenes de trabajo correspondientes, podríamos tener algo así:



Si invertimos la figura para poner como base los pasos iniciales (el sombreado indica el nivel de desarrollo en que nos encontraríamos si venimos siguiendo los cursos de bases de la programación de aprenderaprogramar.com):



Visto así la cuestión es un poco más alentadora: estamos terminando la definición de nuestra base, cuya solidez y tamaño siempre será deseable seguir aumentando en el futuro, para a continuación ver cómo plasmar nuestros desarrollos en un ordenador.

Titulamos el epígrafe como *“Diseño de algoritmos”*. ¿No hemos hablado hasta ahora de diseño de algoritmos? Por supuesto que sí. Llevamos mucho tiempo hablando de diseño de algoritmos. Ya decíamos cuando se introdujo el concepto de algoritmo que este término casi se confunde con el de programa. Del mismo modo, el concepto de diseño de algoritmos casi se confunde con el de programación. No podemos entonces aquí, en unas pocas páginas, tratar de condensar las supuestas pautas para el correcto diseño de algoritmos. Llevamos trabajando en ello todo el tiempo a lo largo del curso. Cuando hablábamos de eficiencia nos referíamos a diseño, cuando hablábamos del nombre de variables nos referíamos a diseño, cuando hablábamos de estructuras, de control de flujo, de módulos, en fin... todo. Hablábamos de diseño.

¿Cuál es el objetivo ahora de proponer un apartado como diseño de algoritmos?

Pues por un lado, el servir de alto en el camino, de reflexión sobre distintas cosas de las que se ha hablado. Y por otro, introducir nuevas herramientas o enfoques que quizás más que aumentar nuestro bagaje de recursos, lo que hacen es darle una mayor potencialidad al mismo. El motor será el mismo, pero vamos a desarrollar *“modificaciones”* para multiplicar su potencia.

PROGRAMAR, RESOLVER PROBLEMAS. RESOLVER PROBLEMAS, PROGRAMAR.

No es casualidad el que dediquemos tiempo en nuestros cursos a “Conocer el problema a resolver”. Una de las identidades que en muchas ocasiones podemos establecer es:

Programar = Resolver problemas

El concepto de resolución de problemas tiene sus orígenes probablemente en los albores de la humanidad y del pensamiento. Es algo mucho más antiguo que la informática, y aún que las matemáticas, que ya de por sí tienen orígenes remotos. Las técnicas y el entrenamiento mental para resolver problemas son aspectos del conocimiento, en términos genéricos, de indudable importancia. Sin embargo, es raro que en la universidad o en los centros de enseñanzas medias se dedique tiempo suficiente a ello, lo cual sería deseable.

El programador habrá de aplicar conocimientos específicos del “saber informático” que posea, pero también estrategias y enfoques que son válidos para cualquier disciplina.

El filósofo francés René Descartes (1596 – 1650) dedicó parte de su tiempo al estudio del conocimiento y a la búsqueda de métodos para su desarrollo. En su “Reglas para la dirección del espíritu” escribía (Regulae ad directionem ingenii. Reglas para la dirección del espíritu. René Descartes. Traducción de Juan Manuel Navarro Cordón. Ed. Alianza. Madrid 2003.):

Regla II: Conviene ocuparse tan sólo de aquellos objetos, sobre los que nuestros espíritus parezcan ser suficientes para obtener un conocimiento cierto e indudable.

Regla IV: El método es necesario para la investigación de la verdad de las cosas.

Dice en la regla IV: “Los mortales están poseídos por una curiosidad tan ciega que con frecuencia conducen sus espíritus por vías desconocidas, sin motivo alguno de esperanza, sino tan sólo para tantear si se encuentra allí lo que buscan: como alguien que ardiese en tan estúpido deseo de encontrar un tesoro, que vagase continuamente por las calles, tratando de encontrar por casualidad alguno [...] No niego que algunas veces vagan tan felizmente que encuentran algo de verdad [...]

Así pues entiendo por método reglas ciertas y fáciles, mediante las cuales el que las observe exactamente no tomará nunca nada falso por verdadero.”

Regla V: Todo el método consiste en el orden y disposición de aquellas cosas a las que se ha de dirigir la mirada de la mente a fin de que descubramos alguna verdad. Y la observaremos exactamente si reducimos gradualmente las proposiciones complicadas y oscuras a otras más simples, y si después intentamos ascender por los mismos grados desde la intuición de las más simples hasta el conocimiento de todas las demás.

Dice en la regla V: “[...] Esta regla ha de ser seguida por el que ha de emprender el conocimiento de las cosas no menos que el hilo de Theseo por quien ha de entrar en el laberinto [...] Muchos [...] con frecuencia examinan las cuestiones más difíciles tan desordenadamente, que me parecen obrar del mismo modo que si intentaran llegar de un solo salto desde la parte más baja de un edificio hasta la más alta [...]

Regla VII: Para completar la ciencia es preciso recorrer en un movimiento continuo e ininterrumpido del pensamiento todas y cada una de las cosas que conciernen a nuestro propósito, y abarcarlas en una enumeración suficientemente ordenada.

Regla X: Para que el espíritu se vuelva sagaz debe ejercitarse en buscar las mismas cosas que ya han sido descubiertas por otros, y en recorrer con método incluso los más insignificantes artificios de los hombres, pero sobre todo aquellos que explican el orden o lo suponen.

Regla XIII: Si entendemos perfectamente una cuestión, debemos abstraerla de todo concepto superfluo, reducirla a la mayor simplicidad y dividirla en las partes más pequeñas que se pueda enumerándolas.

Dice la regla XIII: “[...] Dada una cuestión cualquiera, es preciso esforzarse ante todo por comprender [...] lo que se busca.

Pues frecuentemente algunos de tal modo se apresuran en investigar los problemas, que aplican a su solución un espíritu ligero [...]

Son tan ineptos como un criado que, enviado a algún sitio por su amo [...] se apresurase a correr sin haber recibido aún las órdenes y no sabiendo adónde se le mandaba ir.”

Regla XVI: En cuanto a las cosas que no requieren la atención presente de la mente, incluso si son necesarias para la conclusión, es mejor designarlas por medio de signos muy breves que por figuras completas: pues así la memoria no podrá fallar, mientras que además el pensamiento no se distraerá en retenerlas, cuando se dedique a deducir otras.

Dice la regla XVI: [...] los calculistas acostumbran a designar cada una de las magnitudes por medio de varias unidades o por medio de algún número, y que nosotros en cambio en este lugar hacemos abstracción de los números mismos* [...]

[...] Una vez que la hemos buscado expresada en términos generales, conviene someterla a los números dados [...] Todo esto lo distinguimos nosotros, que buscamos un conocimiento evidente y distinto de las cosas, pero no los calculistas, que se quedan satisfechos con tal que se les presente la suma buscada, aun cuando no se den cuenta de qué modo ésta dependa de los datos, en lo cual sólo, sin embargo, consiste propiamente la ciencia.

* Se refiere al uso de letras en vez de números.

Hemos traído este texto de Descartes a colación porque es sorprendentemente actual y aplicable a la programación, incluso teniendo en cuenta que por esa época aún faltaban siglos para el nacimiento de la informática. Lo errático del tanteo sin reflexión previa, la necesidad de dividir el problema en partes más simples... ¡Nada de eso es nuevo! La programación tiene entre sus bases la resolución de problemas y la actividad mental humana: es ciencia, arte, filosofía... quizás todas y cada una de las vertientes del pensamiento humano. La búsqueda de soluciones a problemas es uno de los motores que mueve al hombre hasta acercarlo al apasionamiento, al reto. Programar es apasionante, una partida de ajedrez contra el ordenador, o mejor dicho, contra nosotros mismos... un reto. Es difícil comprender la pasión que puede despertar una partida de ajedrez y cómo un jugador puede sentirse tan dentro de ella como para sentir dolor, rabia o placer en función del desarrollo de los movimientos. No es una mala estrategia para aproximarse a la programación, para no entenderla como una actividad intelectual

desesperante y compleja; considerarla un reto nos ayudará a tener en cuenta que no debemos precipitarnos, a dedicarle tiempo, método y a sentirnos satisfechos con nuestros desarrollos. Dará igual la escasa trascendencia de lo que vayamos a programar. Si lo asumimos como un reto personal, la motivación será suficiente.

Estudiar la creación de algoritmos es intelectualmente más interesante que estudiar la creación de programas usando un lenguaje concreto, precisamente porque el carácter de reto es mucho más nítido. Desarrollar la ingeniería del motor de un coche es más complicado y apasionante que el trabajo mecánico de montarlo. Sin desdeñar que tanto una cosa como otra son necesarias para alcanzar el objetivo final.

Si en la programación interviene la resolución de problemas (la actividad mental humana) no es de extrañar que se vea afectada por factores variopintos. Entre ellos la capacidad intrínseca del programador, sus conocimientos, su entrenamiento, sus hábitos o costumbres, sus ideas o creencias, sus inclinaciones personales, su estado emocional, su actitud, su autoconfianza y autoestima, etc. Y todo ello a su vez se ve influido por factores externos como la cultura, el ambiente, las modas, las convenciones impuestas por los estudiosos o la comunidad de programadores. El estudio de todos estos factores es sin duda interesante: conocer nos hace libres, nos permite tener conclusiones propias. Dado que abarcar tantos elementos de estudio excede nuestros objetivos, vamos a centrarnos en algunos aspectos del conocimiento humano que nos serán de utilidad como programadores. Y empezaremos hablando de la abstracción de los problemas y la aprehensión de los problemas.

Próxima entrega: CU00225A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60