

Introducción al pensamiento algorítmico

Por el Dr. Wayne Brown – Edición (Brown, 2015) y Traducción Jose David Maldonado

Contenido

Introducción al pensamiento algorítmico	1
¿Qué es un algoritmo?	1
¿Qué es el pensamiento algorítmico?	2
¿Cómo se comunican los algoritmos?.....	3
¿Qué es un diagrama de flujo?.....	4
Diagramas de flujo para algoritmos/programación informática	4
Simbología	5

¿Qué es un algoritmo?

Un algoritmo es un conjunto de instrucciones precisas, paso a paso, para resolver una tarea. Un algoritmo no resuelve una tarea; ofrece una serie de pasos que, si se ejecutan correctamente, darán como resultado la solución a dicha tarea. Usamos algoritmos todos los días, pero a menudo no pensamos explícitamente en los individuales del algoritmo. Por ejemplo, Arrancar un carro, ponerse la ropa, iniciar sesión en su computador o seguir una receta para cocinar un plato de comida, todas estas tareas cotidianas se logran usando un algoritmo, una serie de acciones paso a paso.

Para que un algoritmo sea válido, cada paso (o instrucción) debe ser:

- Sin ambigüedades: la instrucción solo se puede interpretar de una manera única.
- ejecutable: la persona o dispositivo que ejecuta la instrucción debe saber cómo llevar a cabo la instrucción sin ninguna información adicional.
- ordenado: los pasos de un algoritmo deben ordenarse en una secuencia adecuada para realizar correctamente la tarea.

Un ejemplo de algoritmo se muestra en los siete pasos a continuación. Este algoritmo explica cómo hacer pastel de limón congelado (que es el postre favorito de este autor).

- Paso 1: Coloque nuestra lata de leche evaporada en un tazón para mezclar.
- Paso 2: Batir la leche evaporada hasta que se formen picos.
- Paso 3: Agregue 1 taza de azúcar y 1/2 taza de jugo de limón.
- Paso 4: Si desea un sabor más a limón, añada 1 cucharadita de ralladura de limón.
- Paso 5: Espolvoree galletas trituradas en el fondo de un molde.

Paso 6: Vierta la mezcla de leche en el molde.

Paso 7: Deje en el congelador hasta que se congele.

Antes de terminar este ejemplo, pregúntese si los pasos de este algoritmo cumplen con los criterios enumerados anteriormente. Es decir, ¿cada paso es inequívoco, ejecutable y está en el orden correcto?

Esta lectura explica el concepto de algoritmos, algunas propiedades importantes de los algoritmos y cómo se crean, ejecutan y validan los algoritmos.

¿Qué es el pensamiento algorítmico?

El *pensamiento algorítmico* es la capacidad de **comprender, ejecutar, evaluar y crear** algoritmos. Vamos a discutir cada una de estas ideas por separado.

Para ser un pensador algorítmico necesitas la habilidad de entender y ejecutar algoritmos. A algunas personas les resulta fácil seguir una serie de instrucciones precisas, mientras que a otras les resulta muy desafiante. Algunas personas parecen carecer de la paciencia y la diligencia necesarias para seguir un plan paso a paso. Sin embargo, es una habilidad valiosa que todas las personas deben dominar. El pensamiento algorítmico requiere paciencia porque cada instrucción debe ser ejecutada en su secuencia correcta sin saltar por delante o "pasando por alto" algunos de las de instrucciones. Además, el pensamiento algorítmico requiere diligencia y perseverancia. A menudo es tedioso seguir los pasos de un algoritmo complejo y las personas a veces no completan un algoritmo porque simplemente se "dan por vencidos".

El Pensamiento algorítmico también requiere la capacidad de evaluar algoritmos. Esto implica determinar si un algoritmo realmente resuelve una tarea determinada. Esto puede ser muy desafiante. Por ejemplo, una "lista de verificación previa" es un algoritmo para preparar un avión para el despegue. Suponga que se le dio el trabajo de determinar si una nueva "lista de verificación de previa" para el F35 es correcta es decir que verifica todos los sistemas en la aeronave adecuadamente y que es completa, es decir no deja de lado ninguna verificación importante. Estamos de acuerdo que este es un trabajo importante (ya que las vidas de los pilotos dependen de hacerlo bien) y que no es fácil obtener la lista de verificación correcta y completa. (Como nota al margen, una forma de crear la lista de verificación previa es determinar la falla en cada nuevo accidente aéreo y agregar un nuevo control a la lista para evitar futuros choques. Pero el costo en vidas y dinero es demasiado alto. Necesitamos obtener el algoritmo correcto y completo antes de que el primer avión tome el vuelo para evitar una sola pérdida)

Y finalmente, el *pensamiento algorítmico* incluye la capacidad de **crear** Nuevos algoritmos. Este es probablemente el aspecto más desafiante del pensamiento algorítmico. Dada una tarea, ¿puede crear una serie de instrucciones precisas, paso a paso, que siempre resuelvan la tarea correctamente? Obviamente, la complejidad de la tarea tiene un gran impacto en la complejidad de un algoritmo que llevará a cabo la tarea. Las tareas simples normalmente se pueden realizar con algoritmos simples, mientras que las tareas complejas generalmente requieren algoritmos más complejos.

Un aspecto crítico de la creación de algoritmos es el "ejecutor" de un algoritmo. Por ejemplo, el "ejecutor" de una lista de verificación de verificación previa es un piloto. La persona promedio en la calle no pudo ejecutar la lista de verificación previa correctamente porque no entenderían las instrucciones. Las tres características indicadas anteriormente sobre una instrucción de un algoritmo (es decir, una instrucción debe ser inequívoca, ejecutable y ordenada) deben ser coherentes con el conocimiento y la experiencia del "ejecutor". Cuando cree un algoritmo, asegúrese de que cada instrucción pueda ser interpretada y ejecutada sin ambigüedades por el "ejecutor".

Vivimos en la "era de la información", donde muchas de las tareas que las personas necesitan (o desean) realizar pueden ser realizadas por las computadoras. Sin embargo, las computadoras actualmente no tienen una comprensión o conocimiento real, solo pueden realizar tareas que las personas han desarrollado algoritmos para resolver. Por lo tanto, para crear algoritmos que las computadoras puedan ejecutar, debe comprender lo que una computadora es capaz de ejecutar y ser capaz de escribir una serie de instrucciones inequívocas que una computadora pueda ejecutar con éxito. Para el resto de este documento restringiremos nuestra discusión al desarrollo de algoritmos que se pueden ejecutar en las computadoras.

¿Cómo se comunican los algoritmos?

Cada paso (instrucción) de un algoritmo debe indicarse con precisión. Es muy difícil explicar los algoritmos utilizando solo el lenguaje verbal. Por ejemplo, vuelva a leer el algoritmo de ejemplo para hacer "pastel de limón congelado" y luego responda las siguientes preguntas:

- ¿La leche evaporada viene en envases de diferentes tamaños?
- ¿Cuánta leche es "1 lata"?
- ¿Importa qué tipos de batidores se usan para batir la leche evaporada?
- ¿Es "cucharadita" una cuchara pequeña o una cuchara de mesa?

Si intentó implementar la receta para hacer una tarta de limón congelado, es muy posible que descubra otras ambigüedades en las instrucciones. Para eliminar estos "problemas de comunicación", necesitamos un lenguaje preciso para especificar algoritmos, especialmente porque nuestro "ejecutor" es una computadora tonta. Un lenguaje preciso para indicar las

instrucciones de un algoritmo que puede ser ejecutado por una computadora se llama "lenguaje de programación".

Muchos lenguajes de programación se han desarrollado a lo largo de los años y cada lenguaje tiene sus propias características y beneficios especiales. Probablemente haya oído hablar de algunos de estos lenguajes, como C, C ++, Pascal, BASIC, Java, Perl, Python y Ada. Requiere un tiempo considerable para aprender un nuevo lenguaje de programación y no es un objetivo de este curso enseñarle a programar computadoras. Queremos que los estudiantes se conviertan en mejores solucionadores de problemas convirtiéndose en pensadores algorítmicos. Una de las formas comunes de expresar algoritmos sin usar un lenguaje de programación se denomina diagrama de flujo [1]. Los diagramas de flujo proporcionan una descripción visual de un proceso paso a paso.

¿Qué es un diagrama de flujo?

Un diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender. Los diagramas de flujo emplean rectángulos, óvalos, diamantes y otras numerosas figuras para definir el tipo de paso, junto con flechas conectoras que establecen el flujo y la secuencia. Pueden variar desde diagramas simples y dibujados a mano hasta diagramas exhaustivos creados por computadora que describen múltiples pasos y rutas.

Diagramas de flujo para algoritmos/programación informática

Como una representación visual del flujo de datos, los diagramas de flujo son útiles para escribir un programa o algoritmo y explicárselo

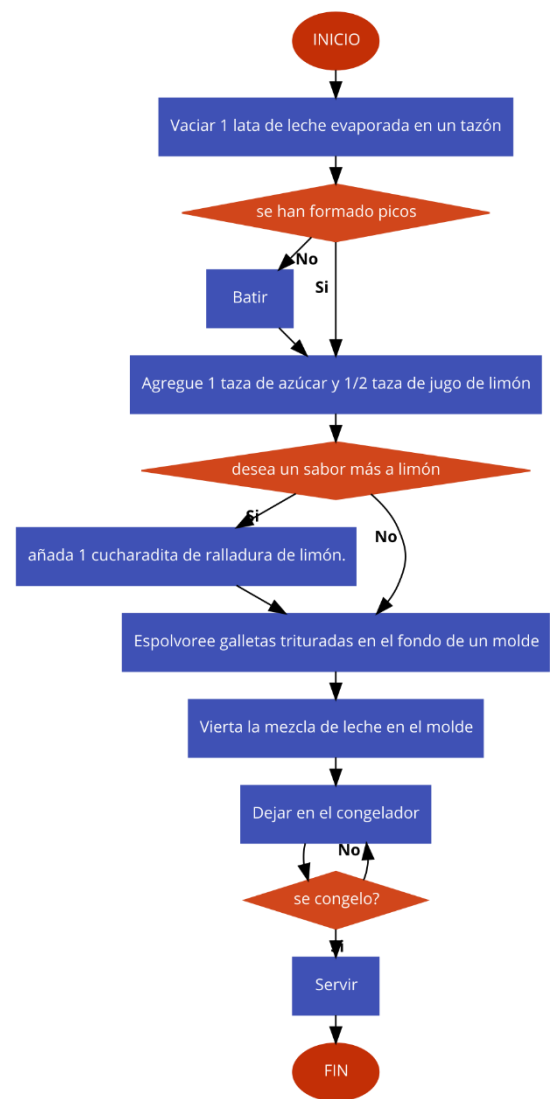



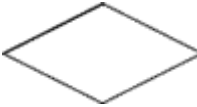

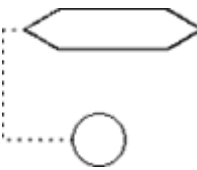




Ilustración 1- Diagrama de Flujo del pastel del limón



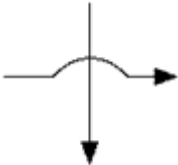

a otros o colaborar con otros en el mismo. Puedes usar un diagrama de flujo para explicar detalladamente la lógica detrás de un programa antes de empezar a codificar el proceso automatizado. Puede ayudar a organizar una perspectiva general y ofrecer una guía cuando llega el momento de codificar.

A menudo, los programadores pueden escribir un pseudocódigo, una combinación de lenguaje natural y lenguaje informático que puede ser leído por personas. Esto puede permitir más detalle que el diagrama de flujo y servir como reemplazo del diagrama de flujo o como el próximo paso del código mismo.

Simbología

Para la construcción de los Diagramas de Flujo se utilizarán los siguientes símbolos:

	Inicio/Final Se utiliza para indicar el inicio y el final de un diagrama; del Inicio sólo puede salir una línea de flujo y al Final sólo debe llegar una línea.		Decisión Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino del diagrama u otro.
	Entrada General Entrada/Salida de datos en General (en esta guía, solo la usaremos para la Entrada).		Iteración Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces.
	Entrada por teclado Instrucción de entrada de datos por teclado. Indica que el computador debe esperar a que el usuario teclee un dato que se guardará en una variable o constante.		Salida Impresa Indica la presentación de uno o varios resultados en forma impresa.
	Llamada a subrutina Indica la llamada a una subrutina o		Salida en Pantalla Instrucción de presentación de mensajes o

	procedimiento determinado.		resultados en pantalla.
	Acción/Proceso General Indica una acción o instrucción general que debe realizar el computador (cambios de valores de variables, asignaciones, operaciones aritméticas, etc).		Conector Indica el enlace de dos partes de un diagrama dentro de la misma página.
	Flujo Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones.		Conector Indica el enlace de dos partes de un diagrama en páginas diferentes.

Pasos para la construcción de un Diagrama de Flujo

Paso 1: Definir claramente la utilización del Diagrama de Flujo y el resultado que se espera obtener de la sesión de trabajo.

- En primer lugar, es necesario clarificar el objetivo de la construcción del Diagrama de Flujo y escribirlo de forma que sea visible para los participantes durante toda la sesión.
- Esta clarificación permitirá definir el grado de detalle y la estructura que se requieren en el diagrama para poder alcanzar dicho objetivo.

Paso 2: Definir los límites del proceso en estudio.

La mejor forma de definir y clarificar dicha definición de los límites del proceso es decidir cuáles son el primer y último pasos del Diagrama de Flujo.

- El primer paso es la respuesta a la pregunta:**
¿Qué nos indica que empieza el proceso?
- El último paso debe contestar a la pregunta:**
¿Cómo sabemos que el proceso ha terminado?

Escribir estos pasos expresándolos de forma clara y concisa

Paso 3: Identificar y documentar los pasos del proceso.

Las preguntas a realizar para la identificación y documentación de los pasos del proceso son las siguientes:

- ¿Existen entradas significativas asociadas con este paso en materia de información, etc?.
- Señalar estas entradas, por medio de los símbolos apropiados, en el diagrama.
- ¿Existen resultados significativos como consecuencia de este paso, tales como información, etc?.
- Señalar estos resultados, por medio de los símbolos apropiados, en el diagrama.
- Una vez realizado este paso, ¿cuál son las actividades inmediatamente siguientes que debemos realizar?.
- Señalar estas actividades, mediante el símbolo apropiado, en el diagrama.

Partiendo del primer paso, realizar este proceso hasta alcanzar el último
Debe describirse en lenguaje verbal e identificar cada uno de los puntos claves

Paso 4: Realizar el trabajo adecuado para los puntos de decisión o bifurcación.

Cuando se llega a un paso en el que existe un punto de decisión o de bifurcación:

- a) Escribir la decisión o alternativa de acuerdo con la simbología utilizada e identificar los posibles caminos a seguir mediante la notación adecuada. En general, cuando se trata de una toma de decisión, se incluye dentro del símbolo una pregunta y la notación de las dos ramas posibles correspondientes se identifican con la notación SI/NO.
- b) Escoger la rama más natural o frecuente de la bifurcación y desarrollarla, según lo dispuesto en el "Paso 3", hasta completarla.
- c) Retroceder hasta la bifurcación y desarrollar el resto de las ramas de igual modo.

Paso 5: Revisar el diagrama completo.

Comprobar que no se han omitido pasos, pequeños bucles, etc. y que el proceso tiene una secuencia lógica.

En caso de que existan dudas sobre parte del proceso representado, realizar una observación directa del proceso o contactar con expertos de ese área para su aclaración.

El resultado final de este paso es el Diagrama de Flujo del proceso en estudio.

Referencias

Brown, W. (Abril de 2015). *Introduction to Algorithmic Thinking*. Obtenido de
<https://raptor.martincarlisle.com/Introduction%20to%20Algorithmic%20Thinking.doc>

[1] <http://en.wikipedia.org/wiki/Flowchart>