


# Algoritmos, Estructuras de Datos y mousse de chocolate



**Jérémy Barbay**

*Profesor Asistente Departamento de Ciencias de la Computación, Universidad de Chile. Docteur en Informatique de l'université de Paris XI (2002).*

*Líneas de investigación: Diseño y Análisis de Algoritmos, Estructuras de Datos.*  
[jbarbay@dcc.uchile.cl](mailto:jbarbay@dcc.uchile.cl)

## PREPAREMOS UN MOUSSE DE CHOCOLATE

Una receta especifica el número de personas que se puede alimentar con ella (el output), la lista de ingredientes (la entrada) y la lista de instrucciones (el proceso en sí).

Una receta es al humano lo que los programas o algoritmos son para los computadores: se necesitan parámetros, al igual que el número  $n$  de porciones que se desea producir; incluye algunas operaciones condicionales (“*IF (...) THEN (...)*”) y operaciones lógicas; especifica procesos paralelos y secuenciales (e.g. “hacer en paralelo”); y se refiere a otras recetas, como por ejemplo la del “Baño María” (Figura 2).



### *Mousse de chocolate (para $n$ porciones)*

#### Implementos:

- Un recipiente para el “Baño María”.
- Un recipiente, con tapa para la mezcla.
- IF el recipiente más pequeño para el “Baño María”:
  - No tiene tapa o
  - No se puede poner en el refrigerador,
- THEN
  - Otro recipiente con tapa para poner la mezcla en el refrigerador.
- Espátula.

#### Ingredientes:

- $n + 1$  huevos,
- $n$  cucharadas de azúcar,
- $n * 25$  gramos de chocolate negro.

#### Preparación:

- Hacer en paralelo:
  - Poner el chocolate a derretir a “Baño María”.
  - Preparar la clara de los huevos:
    - Separar la yema de la clara de los huevos.
    - IF la yema cae sobre la clara,
    - THEN sacar con la cáscara de huevo.
    - Batir las claras hasta que estén firmes.
  - Cuando el chocolate esté líquido:
  - Sacar el recipiente del agua hirviendo.
  - Lentamente agregar el azúcar y la yema de los huevos, y revolver enérgicamente.
  - Agregar las claras batidas, incorporando rápidamente.
    - Comenzar con una cuchara.
    - Mezclar rápidamente hasta que la preparación esté uniforme.
    - Doblar la cantidad cada vez que se agregue más.
  - Dejar enfriar al menos dos horas en el refrigerador.
  - Servir acompañado de una fruta ácida o galletas.

### *Baño María*

- Derretir  $x$  a Baño María.
- Implementos:
  - Dos recipientes redondos para poner uno dentro de otro, con una diferencia de diámetro de 4 centímetros.
  - Una espátula o una cuchara de madera.
- **Ingredientes:**
  - $x$  para derretir
  - Agua
- **Preparación:**
  - Llenar el recipiente más grande hasta 1/3 de agua, de tal forma que el agua y el recipiente más pequeño (vacío) completen el recipiente más grande.
  - Hervir el agua.
  - Poner  $x$  en el recipiente más pequeño y luego poner en el agua hirviendo.
  - Disminuya la llama para que el agua siga hirviendo, sin que se escape del recipiente.
  - Revuelva  $x$  con frecuencia en el recipiente más grande.
  - Sacar cuando esté suficientemente líquido.

Figura 2

## ¿CUÁNTO TIEMPO DEMORA PREPARAR UN MOUSSE DE CHOCOLATE?

Esta pregunta es sobre el producto, no sobre la receta. Dado que el mousse de chocolate se sirve frío y que la receta menciona dos horas de tiempo de espera en el refrigerador, probablemente podemos deducir que cualquier receta de mousse



de chocolate en la que se use solamente el refrigerador para enfriar el producto, requiere de dos horas de preparación, pero eso sólo da una *cota inferior en el tiempo* requerido, y no es una información tan útil para alguien que desee saber con cuánto tiempo de anticipación debe empezar a cocinar antes de comer.

De manera similar, en construcción algorítmica se distingue entre la complejidad de un algoritmo particular (e.g. la receta de mousse de chocolate de la Figura 1) y la complejidad del problema en sí mismo (e.g. hacer un mousse de chocolate con cualquier otra receta). La evaluación de la segunda, por lo general, depende de los supuestos sobre los recursos disponibles (e.g. en nuestro ejemplo suponiendo que el mousse se enfría mediante un refrigerador, en lugar de una herramienta más sofisticada).

## ¿CUÁNTO TIEMPO TOMA SEGUIR ESTA RECETA?

La interrogación es un poco vaga: ¿es preguntar (1) cuánto tiempo antes de la

cena debemos empezar a cocinar, o (2) cuánto tiempo vamos a estar activamente siguiendo la receta, sin poder enfocarnos seriamente en otra cosa? ¡Ninguna pregunta es trivial de contestar! Además, el tiempo necesario para ejecutar una receta depende de muchos factores (e.g. podríamos estar derramando una gran cantidad de yema en la clara de los huevos, y toma bastante tiempo recuperarla por cada huevo). En la cocina, por lo general estamos contentos con una aproximación del tiempo necesario, con arreglo a suposiciones generales: suponiendo que la cantidad es razonable ( $4 \leq n \leq 8$  porciones); fundir el chocolate será el tiempo dominante en la parte paralela de la receta, tomando aproximadamente 15 minutos; la mezcla tardará aproximadamente 15 minutos más, mientras que dejar que se enfríe en el refrigerador tomará, como se especifica, 2 horas. Esto entrega respuesta a las dos interpretaciones de la pregunta: deberíamos empezar a cocinar 2,5 horas antes de la cena, y estaremos ocupados por 30 minutos.

Del mismo modo, el análisis de las construcciones algorítmicas requiere definir formalmente la propiedad que sea analizada, tal como la cantidad de *tiempo necesario para preprocesar los datos*, versus la cantidad de *tiempo esperada* por un usuario, *antes de recibir una respuesta a su consulta*. Cuando se trata de construcciones algorítmicas, también hacemos uso de aproximaciones al calcular el tiempo de ejecución (en particular, al *identificar el tiempo dominante de un proceso*); “supuestos generales” (lo cual puede ser problemático: en particular, es mucho más arriesgado suponer que  $n$  será pequeño, ya que la gente está mucho más dispuesta a que el computador trabaje, a que ellos mismos lo hagan); y diversas medidas del costo de la ejecución (e.g. el tiempo activo y el tiempo de espera de la receta). El contexto general de la investigación en análisis de algoritmos es *identificar formalmente los problemas* que los computadores debieran resolver; *definir nuevos algoritmos* para la solución de

esos problemas; y *analizar los algoritmos* de una manera que *ayude a decidir* qué algoritmos se deben utilizar en cada caso particular.

## ¿CUÁNTAS HERRAMIENTAS SE NECESITAN?

Si va a compartir la cocina con otras personas, o si preparará varias recetas al mismo tiempo, o si su cocina está mal equipada; debe comprobar con cuidado que tiene todo el equipo necesario. Para hacer su tarea más fácil, la receta debe incluir el equipo que se necesita para ejecutarla.

Del mismo modo, las construcciones algorítmicas tienen *requisitos de espacio*, por una razón muy parecida: el usuario podría compartir la máquina con otros, o ésta podría estar ejecutando varios programas al mismo tiempo, o ejecutarse en un dispositivo móvil con recursos limitados.

## ¿CUÁNTA ENERGÍA SE NECESITA?

Necesitamos energía para derretir el chocolate (quemador de gas), para batir la clara de los huevos (batidora eléctrica), para mezclar todo (con la mano) y para enfriar el producto (refrigerador). Muy a menudo el tiempo es el recurso crítico, pero en algún caso especial la energía también podría serlo (por ejemplo, si está en un pueblo de África Central, donde no hay refrigerador ni electricidad).

Del mismo modo, los recursos energéticos de las construcciones algorítmicas son críticos cuando se destinan a *equipos integrados o nanorobots*, donde la energía es un recurso tan crítico como el tiempo; y los teraservidores masivamente paralelos de Google o Amazon,

en los que consumo de energía es un recurso aún más crítico.

## ¿SE PUEDE PREPARAR EN EL FONDO DEL MAR? ¿O EN LA LUNA?

La temperatura de ebullición del agua se ve afectada por la presión: en un entorno en el que hierve a una temperatura más baja, no se derrite el chocolate. Batir la clara de los huevos es una acción mecánica que es afectada por la gravedad; en un entorno con menos gravedad los huevos batidos serán más livianos y el resultado de la receta distinta.

Del mismo modo, el rendimiento de las construcciones algorítmicas se ve afectado por condiciones externas. *Computadores enviados al espacio* tienen mayores tasas de error debido al aumento de las radiaciones en comparación con la superficie de la tierra, lo que se corrige a través de la autoestabilización de algoritmos. Más cerca de la tierra, en *equipos con varios tipos diferentes de memoria* (e.g. una barata, grande y lenta; una rápida y pequeña), los algoritmos están diseñados para optimizar el uso de aquellas.

## ¿SE PUEDE USAR LA RECETA PARA ALIMENTAR A UN MILLÓN DE PERSONAS?

La receta se expresa en función del número  $n$  de las porciones, que generalmente se supone es más bien pequeño, alias cuatro a ocho, ¡aunque sólo sea por el tamaño limitado de los recipientes en los cuales se preparan los ingredientes y el producto! Para preparar más porciones de mousse de

chocolate, uno puede utilizar cuencos más grandes, repetir la receta varias veces, ejecutar la misma receta varias veces en paralelo, o una combinación de estas soluciones (e.g. usar recipientes tan grandes como razonable sea y repetir la receta). En particular, si se tarda  $t$  horas y  $e$  energía en preparar  $n$  porciones, la preparación de  $2n$  porciones puede ser optimizada para tardar menos de  $2t$  horas y gastar menos de  $2e$  de energía (e.g. no lavar los recipientes entre dos recetas ahorra tiempo).

Del mismo modo, el comportamiento de un algoritmo en entradas de tamaño creciente es objeto de muchos estudios. En algunos casos, el tiempo de ejecución de un algoritmo aumenta más lentamente que el tamaño de su entrada (e.g. *buscar en un arreglo ordenado* de tamaño  $n=128$  a través de comparaciones requiere  $1+\log_2 n=8$  comparaciones, mientras que *buscar en una matriz ordenada* de tamaño  $n=2*128=256$  toma  $1+\log_2 n=9<2*8$  comparaciones), y, a veces, crece más rápido (e.g. *clasificar  $n=128$  elementos* a través de comparaciones requiere  $n \log_2 n=128 * 7$  comparaciones, mien-

tras que clasificar  $n=128 * 2 =256$  elementos a través de comparaciones requieren  $\log_2 n =256 * 8 > 2 * 128 * 7$ .

## ¿PUEDES HACERME UNA RECETA DETALLADA CON FOTOS?

Eso no tomará mucho más tiempo que una simple ejecución de la receta: sólo mantener la cámara cerca y publicar las fotos en orden.

De manera similar, los algoritmos diseñados para producir una salida particular (e.g. ordenar un arreglo) implícitamente describen un objeto a través de su proceso (e.g. codificar una permutación), y es poco el trabajo adicional requerido para almacenar ese objeto (e.g. anotando el resultado de cada comparación realizada por el algoritmo de ordenamiento). En algunos casos particulares (e.g. *merge sort*), esta codificación ha resultado ser útil (e.g. estructuras de datos comprimidas para permutaciones, más pequeñas y más rápidas que las previamente conocidas). BITS



Mousse de chocolate casero.