

## Lógica de programación (2): Elementos básicos de un programa

By [Israel Capetillo](#)11:45Básico, Desarrollador, Lógica de Programación, ProgramaciónSin comentarios

¿Listos para arrancar con el aprendizaje de la lógica de programación? Pues bien, lo primero que un buen programador debe hacer es reconocer a detalle cuáles son los **datos de entrada** que necesita, cuáles son los **procesos** que debe realizar con esos datos de entrada y cuáles son los **datos de salida** que se obtendrán como resultado de ese procesamiento.

Esto representa un cambio significativo (y definitivo) en nuestro rol de usuarios a programadores. Y es que cuando somos usuarios, el procesamiento de los datos es como una caja negra. Como usuarios de un programa, sólo nos interesa introducir datos, hacer algunos clics y dejar que la computadora haga su magia para mostrarnos ese reporte que necesitamos, o para añadir un amigo en Facebook. Sin embargo, como programadores, nuestra labor será convertir esa caja negra en una transparente, no sólo para comprender el código detrás de la magia, sino para crear esa magia nosotros mismos.



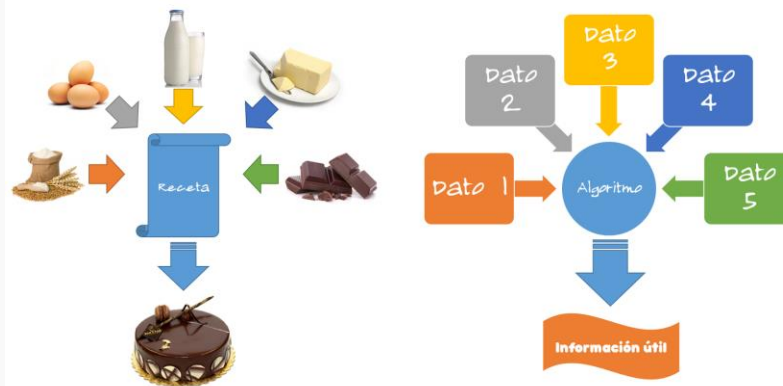
Ni magia ni aliens. La computadora funciona con programas escritos por personas...

---

Esa "caja transparente" es el algoritmo que como programadores debemos crear para convertir las entradas esperadas en las salidas deseadas. Cabe

mentonar que estas salidas tienen que ser **información útil** ya sea para los usuarios finales o para algún otro programa que requiera esa información como entrada para procesarla.

Si comparamos un programa con la receta para preparar un pastel de chocolate (que, como establecimos en la primera lección de este curso, ambos son algoritmos), las entradas serían todos los ingredientes necesarios, la receta misma sería el algoritmo y la salida sería nuestro flamante y delicioso pastel de chocolate listo para consumirse.



(Clic en la imagen para agrandar)

En una computadora, las entradas pueden venir de una gran variedad de dispositivos: el teclado, el mouse, un lector de código de barras, una memoria USB, etc., mientras que los datos de salida también se pueden mostrar en una amplia variedad de formas: en la pantalla, impresos en papel, en forma de sonidos en las bocinas, en forma de alguna acción de otro dispositivo (como encender o apagar la Wi-Fi del celular), etc.

Un buen programador sabe no sólo procesar los datos de entrada para generar las salidas, sino también el origen de las primeras y el destino de las segundas.

Por ahora nosotros nos preocuparemos sólo en generar las instrucciones para procesar datos, independientemente de su origen o su destino después del procesamiento.

¿Recuerdan nuestro primer algoritmo de ejemplo en la lección anterior?

**Algoritmo** `Calcular_Nomina`

```
1.      Leer      horas_trabajadas,      tarifa_por_hora,
tasa_impuesto
2.      Calcular   PagoBruto      =      horas_trabajadas      *
tarifa_por_hora
3.      Calcular   Impuestos = PagoBruto * tasa_impuesto
4.      Calcular   PagoNeto = PagoBruto - Impuestos
5.      Visualizar PagoBruto, Impuestos, PagoNeto
```

En ese algoritmo, los datos de entrada se leen en la instrucción número 1 y son: horas\_trabajadas, tarifa\_por\_hora y tasa\_impuesto. Estos datos son procesados por las instrucciones 2,3 y 4 para generar los datos de salida que son visualizados en el paso 5. Como ven, en este caso no nos preocupamos ni de dónde vienen las entradas ni dónde serán mostrados los resultados finales.

En la próxima lección comenzaremos a adentrarnos en el diseño del algoritmo conociendo los diferentes tipos de instrucciones que podemos utilizar para darle forma a nuestro proceso. ¡Hasta la próxima!

## ACTIVIDADES DE APRENDIZAJE

- En la lección anterior escribiste 2 algoritmos: uno para cambiar el pañal a un bebé y otro para calcular el monto a pagar por el consumo en una taquería. Define ahora cuáles serían las entradas, las instrucciones de procesamiento y las salidas esperadas para cada uno de esos algoritmos.

## CONCEPTOS CLAVE

- Datos de entrada
- Proceso
- Datos de salida