

Módulo 6 - Algoritmos I - Teoría

Contenido:

1. [Introducción](#)
2. [Hola Algoritmo](#)

1. Introducción

Definición

Cuando estemos desarrollando nuestros programas, muchas veces tendremos que enfrentarnos a resolver problemas matemáticos, estadísticos, etc... Una buena base en algoritmia nos proporcionará herramientas necesarias para aportar mejores soluciones a esos problemas.

¿Y qué es esto de la algoritmia?

Un algoritmo es por definición, un **conjunto ordenado de operaciones matemáticas** que nos permite hacer un cálculo para **hallar la solución de todos los problemas de un mismo tipo**. Es decir, es un conjunto de instrucciones que siempre sigue el siguiente patrón:

1. Precisión: Tiene que explicar de manera única e inequívoca cada paso o instrucción.
2. Finito: El número de instrucciones a ejecutar debe ser limitada.
3. Definición: Los mismos datos de entrada siempre deben proporcionar la misma información de salida.
4. Entrada: El número de elementos de entrada puede ser cero o más.
5. Resolución: Siempre debe producir un resultado, que será el o los datos de salida.

En pocas palabras, **un algoritmo sirve para resolver paso a paso un problema**. De hecho, si lo piensas, un algoritmo es incluso la base de los manuales de usuario, folletos de instrucciones, etc...

Partes de un algoritmo

Todo algoritmo debe constar de las siguientes partes:

- Entrada: El ingreso de los datos que el algoritmo necesita para operar.
- Proceso: Se trata de la operación lógica formal que el algoritmo realizará con lo recibido en la entrada.
- Salida: Los resultados obtenidos del proceso sobre la entrada, una vez terminada la ejecución del algoritmo.

Por ejemplo, necesitamos construir el algoritmo de la suma de dos números.

```
function suma(number1, number2) {  
  var suma = number1 + number2;  
  return suma;  
}
```

Entrada: `numberA` y `numberB`. Proceso: sentencia `var suma = number1 + number2;`, donde se suman los dos números y se almacenan en la variable `suma`. Salida: variable `suma`.

Lo que comentábamos anteriormente sobre que dado una misma entrada, siempre debe producir la misma salida, es un concepto importante de los algoritmos. Deben estar correctamente definidos para que los resultados sean los esperados, siempre.

Cómo hacer un algoritmo

Y después de ver las partes en las que se compone un algoritmo, vamos a ver cómo podemos crear un algoritmo. Lo único que debemos tener claro es **qué necesitamos y cómo lo podemos resolver**.

Para ello, debemos realizar una serie de pasos necesarios y sus derivaciones para poder llegar al resultado esperado del problema. Los pasos para definir y concretar nuestro algoritmo son los siguientes:

- **Primer paso: Análisis preliminar del problema.**
Lo primero que tenemos que hacer, antes de comenzar con el desarrollo de cualquier algoritmo es llevar a cabo un análisis de la situación o problema.
- **Segundo paso: Definir los requerimientos.**
En este paso se debe definir exactamente el problema que tenemos que solucionar y desglosarlo, incluyendo todas las derivaciones que puedan surgir.
- **Tercer paso: La identificación de los apartados.**
En este paso, identificar claramente los apartados en los que se puede descomponer es tan importante como la identificación de los requerimientos. Esto es así debido a que identificar correctamente los apartados nos va servir para simplificar considerablemente la puesta en marcha de los pasos del algoritmo correcto para nuestro problema, y que hemos identificado en el paso anterior.
- **Cuarto paso: La creación del algoritmo.**
En este punto debemos asegurarnos que el algoritmo cumpla con todos los requerimientos adecuados para llevar a cabo la función encomendada.
- **Quinto paso: La implementación del algoritmo.**
La implementación de los algoritmos se debe llevar a cabo traduciéndolo a un lenguaje de programación con el propósito de que cualquier computadora pueda interpretar sus instrucciones y enviar a su hardware la información necesaria para poder completar los pasos correspondientes, y de esta manera obtener el resultado esperado.

Pongamos en práctica la teoría. Empecemos con un algoritmo sencillo.

Imaginemos que queremos hacer el algoritmo de una dispensadora de bebidas calientes. Vamos a definirlo en pasos:

- **Primer paso: Análisis preliminar del problema.** Queremos dispensar una bebida caliente para el usuario.
- **Segundo paso: Definir los requerimientos.** Debemos tener en cuenta la selección del usuario, si existe bebida disponible y si el usuario introduce el dinero correspondiente.
- **Tercer paso: La identificación de los apartados.** Como apartados tenemos "solicitar la selección", "comprobar que existe disponibilidad", "pedir al usuario el importe", "comprobar que el importe es correcto" y por último, "dispensar la bebida".
- **Cuarto paso: La creación del algoritmo.**

Comienzo

- Paso 1 Escribir "Introduzca el número de selección"
- Paso 2 Leer "Selección"
- Paso 3 Si existe disponibilidad entonces
- Paso 4 Escribir "Introduzca el importe correspondiente"
- Paso 5 En caso contrario, "si no"
- Paso 6 Avisar del faltante y volver al paso 1
- Paso 7 Recoger dinero y comprobar
- Paso 8 Si es correcto entonces
- Paso 9 Dispensar bebida
- Paso 10 En caso contrario, "si no"
- Paso 11 Volver al paso 4

Final

2. Hola Algoritmo

Hagamos nuestro primer algoritmo básico. El problema sería el siguiente:

- Dada una lista de productos donde se describe la cantidad y el precio unitario de cada producto, aplica un descuento del 5% para aquellos productos que superen 5 unidades, y un descuento adicional del 5% si el precio total de un producto es igual o supera 50€.

Entrada

Listado de productos con número de unidades y precio unitario. Un ejemplo podría ser:

```
var productList = [
  {
    product: "Jamón Ibérico",
    units: 1,
    price: 9.99,
  },
  {
    product: "Queso Tostado",
    units: 6,
    price: 4.45,
  },
  {
    product: "Vino Tinto",
    units: 12,
    price: 8.15,
  },
];
```

Procesado

Veamos, deberemos iterar por cada producto en la lista y chequear el número de unidades así como su precio total. Si se cumplen alguna de las 2 condiciones dadas, el precio total para ese producto debe reducirse según el descuento obtenido. Por ejemplo:

```
var getDiscount = (units, cost) => (units > 5 ? 5 : 0) + (cost >= 50 ? 5 : 0);

var productCost = product => {
  var productCost = product.units * product.price;
  return productCost * (1 - getDiscount(product.units, productCost) / 100);
};

var totalCost = productList => {
  var totalCost = 0;
  for (var product of productList) {
    totalCost += productCost(product);
  }
  return totalCost;
};
```

Salida

El precio final de la lista de la compra con el descuento aplicado. Si ejecutamos nuestro algoritmo para la lista de artículos propuesta, deberíamos obtener un total de:

123.375€

Donde el vino tinto ha sido rebajado un 10% y el queso un 5%, suponiendo un ahorro total de 11,115€ sobre el precio sin descuento (134,49€).