

# Programación: producto de dos vectores por componentes

**Objetivos.** Programar la función que calcule el producto de dos vectores por componentes.

**Requisitos.** Programación de funciones, ciclos `for`, entradas de vectores.

**1. Producto de dos vectores por componentes.** Dados  $a, b \in \mathbb{R}^n$ , denotemos por  $a \odot b$  su *producto por componentes*:

$$a \odot b = \left[ a_j b_j \right]_{j=1}^n.$$

En el lenguaje de MATLAB el producto  $a \odot b$  se puede calcular como `a .* b`.

**2. Problema: producto punto de dos vectores.** En algún lenguaje de programación escribir una función que calcule el producto por componentes de dos vectores dados.

Entrada:  $a, b \in \mathbb{R}^n$ .

Condición que debe cumplir la entrada: los vectores dados son de la misma longitud. No tiene que verificar esta condición.

Salida: el vector  $\left[ a_j b_j \right]_{j=1}^n$ .

Solución en el lenguaje de MATLAB (guardar en el archivo `componentwiseproduct.m`):

```
function p = componentwiseproduct(a, b),
    n = length(a);
    p = zeros(n, 1);
    for j = 1 : n,
        p(j) = a(j) * b(j);
    end
end
```

**3. Análisis de complejidad.** Calcular el número de operaciones de multiplicación en el algoritmo anterior. La respuesta es un polinomio de la variable  $n$ .

**4. Pruebas con vectores pequeños.** En algún lenguaje de programación escribir un programa que llame a la función del ejercicio anterior, aplicándola a los vectores

$$a = \begin{bmatrix} 3 \\ -4 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 5 \\ -2 \end{bmatrix}.$$

Solución en MATLAB (guardar en el archivo `testcomponentwiseproduct1.m`):

```
function [] = testcomponentwiseproduct1(),
    a = [3; -4; 1]; b = [2; 5; -2];
    display(componentwiseproduct(a, b));
    display(a .* b);
end
```

**5. Pruebas con vectores grandes aleatorios.** En algún lenguaje de programación escribir un programa que genere vectores de tamaños grandes:  $n = 10^4$ , luego  $n = 10^5$ , luego  $n = 10^6$ , aplique a estos vectores la función del Ejercicio 2 y mida el tiempo de ejecución. Solución en MATLAB (guardar en el archivo `testcomponentwiseproduct2.m`):

```
function [] = componentwiseproduct2(),
    for n = [100000, 1000000, 10000000],
        display(n);
        a = rand(n, 1); b = rand(n, 1);
        tic;
        p = componentwiseproduct(a, b);
        t = toc;
        display(t);
    end
end
```

¿Cómo se cambia el tiempo de ejecución al multiplicar  $n$  por 10?