

## Normalización

La normalización vectorial, aunque tiene un nombre que puede asustar bastante, es algo sencillo. Consiste en que dado un arreglo con valores, se genera un nuevo arreglo donde todos los valores están entre cero y uno.

Para lograrlo cada valor se divide por el módulo, y el módulo se calcula como la raíz de la suma de cada uno de los elementos al cuadrado.

o sea:  $modulo([1, 2, 3]) = raiz(1^2 + 2^2 + 3^2)$

Necesitamos:

- Un método para calcular el módulo de un arreglo
- Dividir cada uno de los elementos del arreglo por el módulo y guardarlos en un nuevo arreglo.

Podemos verificar el resultado dado que la suma de todos los valores al cuadrado debe ser uno.

### Creando el método módulo, recibe un array y devuelve el módulo del arreglo

```
def modulo(array)
  n = array.count
  suma = 0
  n.times do |i|
    suma = suma + array[i]**2
  end
  Math.sqrt(suma)
end

modulo([1,2,3])
```

3.7416573867739413

## Recorremos el arreglo y a cada elemento lo dividimos por el módulo

```
def modulo(array)
  n = array.count
  suma = 0
  n.times do |i|
    suma = suma + array[i]**2
  end
  Math.sqrt(suma)
end

def normalizar(array)
  n = array.count
  m = modulo(array)
  array_normalizado = []
  n.times do |i|
    array_normalizado.push array[i] / m
  end
  array_normalizado
end

normalizar([1,2,3])
```

```
[0.2672612419124244, 0.5345224838248488, 0.8017837257372732]
```

## Verificamos el resultado

```
# Dijimos que la suma de cada uno de los números al
# cuadrado debe ser uno, esto lo podemos probar
# iterando el arreglo.

array = [0.2672612419124244, 0.5345224838248488, 0.8017837257372732]
suma = 0
n = array.count
n.times do |i|
  suma += array[i] ** 2
end

puts suma
```

```
1.0
```