**[Distancia euclídea](http://www.solveet.com/exercises/Distancia-euclidea/71) en**[**Python**](http://www.solveet.com/python-solutions)

La solución sería una sola línea, pero aporto unos cuantos tests para comprobar que la solución es correcta. Para solucionarlo utilizo 'list comprehensions' de python y la función 'zip', que me ayuda a procesar los dos arrays de una sola tacada. La línea sería:

**sqrt(sum([(v1[i]-v2[i])\*\*2 for i in range(len(v1))]))**

## [¿Cómo se puede calcular la distancia euclidiana con numpy?](http://www.iteramos.com/pregunta/17663/como-se-puede-calcular-la-distancia-euclidiana-con-numpy)

Esta fórmula para calcular la distancia de euclidiana con tres variables seria de la siguiente manera:

Las variable: (var1a, var2a, var3a) (var1b, var2b, var3b)

dist = sqrt((var1a-var2b)^2 + (var2a-var2b)^2 + (var3a-var3b)^2)

¿Cuál es la mejor manera de hacer esto con Numpy, o con Python en general? Entonces serian de estas dos formas :

a = numpy.array((xa ,ya, za))

b = numpy.array((xb, yb, zb))

[**Distancia euclídea**](http://www.solveet.com/exercises/Distancia-euclidea/71)**en**[**Python**](http://www.solveet.com/python-solutions)

Mientras que la distancia de **Manhattan se calcula de la siguiente manera: esto es solo para dos variables.**

**def manjathan(v1,v2):**

**return abs(sum(|(v1[i]-v2[i]) for i in range(len(v1))|))**