

**IP Honores  
N1-HT1**

Nombre completo: \_\_\_\_\_

El *Teorema de Pitágoras* establece que, en un triángulo rectángulo, la hipotenusa  $c$  se puede calcular como:

$$c^2 = a^2 + b^2$$

Si tomamos los puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ , podemos ver que forman un triángulo rectángulo con los catetos:

$$\begin{aligned} a &= x_2 - x_1 \\ b &= y_2 - y_1 \end{aligned}$$

Aplicamos el teorema:

$$d^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

Finalmente, despejamos  $d$  tomando la raíz cuadrada, y así obtenemos la fórmula para calcular la distancia entre los dos puntos en el plano 2D:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Se requiere una función que permita hacer este cálculo. Para ello, aplique la metodología del curso:

## 1. Análisis

1.1 Leer y comprender el enunciado.

1.2 Identificar el problema.

1.3 Especificar el problema:

- ¿Cuál es el propósito de la función?

*Calcula la distancia entre los dos puntos en el plano 2D.*

- ¿Cuál será el nombre la función?

- ¿Qué dato(s) de entrada se requiere?

- ¿Qué salida (resultado) producirá?

- ¿Qué restricciones le aplican a los datos de entrada?

- ¿Qué restricciones le aplican a la salida o resultado producido?

1.4 Escriba la documentación\* de la función:

""""  
  
\_\_\_\_\_  
  
Args:  
  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
  
Returns:  
  
\_\_\_\_\_  
  
""""

\*En este caso, nombres de parámetro como por ejemplo x1, x2, y1 y y2, son adecuados.

## 2. Diseño

2.1 Formulación de ejemplos.

- Escriba al menos 4 ejemplos, usando casos significativos y no redundantes.

- *Para  $x1$ ,  $x2$ ,  $y1$  y  $y2$  iguales a 0, el resultado esperado sería 0.0*

## 2.2 Diseño del algoritmo:

¿Qué algoritmo planea usar para calcular la distancia entre los dos puntos en el plano 2D? Responda planteando una lista de pasos:

---

---

---

---

---

---

---

---

---

---

## 2.2 Diseño de los casos de prueba.

- Escriba al menos 4 doctests con base a la formulación de ejemplos y el diseño del algoritmo
- Documente cada caso de prueba.
- Ejemplo de la función `suma()` :

```
>>> suma(0, 0) # caso solo ceros
0
```

## 2. Construcción

### 2.1 Construcción de la signatura de su función.

- Ejemplo de la función `suma()` :

```
def suma(primer_operando: int, segundo_operando: int) -> int:
```

- Implemente aquí la signatura de su función en Python:

```
def
```

### 2.2 Construcción del cuerpo de su función.

- Ejemplo de la función `suma()` :

```
return num1 + num2
```

- Implemente aquí el cuerpo (sin documentación ni doctests) de su función en Python:

## ¿Y las pruebas?

- Probar una función es un **paso fundamental** de la metodología.
  - Usted probará su función en **N1-L2**.