

# Requerimiento – RQ1

## 1. DEFINIR UNA MATRIZ DE 3X4

3	Chita, Boyacá	Lat1: 6,211 Lon1: -72,482	2
		Lat2: 6,212 Lon2: -72,470	25
		Lat3: 6,105 Lon3: -72,342	25
		Lat4: 6,210 Lon4: -72,442	50

## Requerimiento – RQ2

1. VALIDAR QUE LA MATRIZ ESTE VACIA (LEN) -> True

DEBE MOSTRAR “Error sin registro de coordenadas” -> Break

2. Recorrer la matriz (Con las coordenadas y mostrarlas en el formato solicitado)

```
coordenada [latitud,longitud] 1 : ['6.533', '-75.123']  
coordenada [latitud,longitud] 2 : ['6.602', '-75.110']  
coordenada [latitud,longitud] 3 : ['6.689', '-75.090']  
Por favor elija su ubicación actual (1,2 ó 3) para calcular la distancia a los puntos de conexión
```

3. Si se coloca algo diferente a 1-2-3 debe salir “Error ubicación”

## Requerimiento – RQ2

1. Calcular las distancias (coordenada seleccionada matriz definida en el RQ1)
2. Invocar una función (que tiene como parámetros (lat. Long de la coordenada seleccionada)
3. Calcular\_distancias(matriz[0][1], matriz[1][1])
4. Crear una función para calcular las distancias de mi ubicación | las zonas WIFI de la matriz 4x3.
4. Para hallar el calculo de las distancias se puede crear un for que se repita 4 veces con la

$$Distancia = 2 * R * asin \sqrt{\sin^2 \left( \frac{\Delta lat}{2} \right) + \cos(lat1) * \cos(lat2) * \sin^2 \left( \frac{\Delta lon}{2} \right)}$$

Lat → latitud

Lon → Longitud

(lat1, lon1) → Latitud y longitud en el punto 1

(lat2, lon2) → Latitud y longitud en el punto 2

Δlat = lat2 – lat1

Δlon = lon2 – lon1

R = 6372.795477598 Km (Radio de la Tierra)

5. Retornar las 4 distancias (se recomienda retornar una lista)

Distancias = calcular\_distancia(lat1, long1)

## • Requerimiento – RQ2

- Mostrar en pantalla la siguiente información.

	distancias	80	450			
		zona1	zona2	zona3	zona4	
	usuarios	2	25	25	50	
	3	Chita, Boyacá	Lat1: 6,211 Lon1: -72,482	2	index	
			Lat2: 6,212 Lon2: -72,470	25	pop	
			Lat3: 6,105 Lon3: -72,342	25	insert	
			Lat4: 6,210 Lon4: -72,442	50	sort	

### Zonas wifi cercanas con menos usuarios

La zona wifi 1: ubicada en ['-4.453', '-72.543'] a 1510 metros , tiene en promedio 10 usuarios

La zona wifi 2: ubicada en ['-4.553', '-72.543'] a 1500 metros , tiene en promedio 55 usuarios

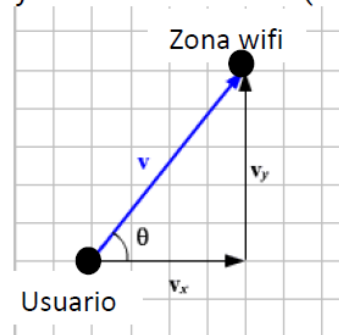
Elija 1 o 2 para recibir indicaciones de llegada

## • Requerimiento – RQ3

- Mostrar en pantalla las indicaciones de guía para encontrar esa zona WIFI

2	-Tiempo en bus -Tiempo en auto
---	-----------------------------------

- Para definir la dirección que se le debe sugerir al usuario, piense en un triángulo de vectores. Ejemplo: Si el punto está más arriba (norte) y más a la derecha (oriente) debe indicar:



“Para llegar a la zona wifi dirigirse primero al oriente y luego hacia el norte”

- Para calcular el tiempo promedio de viaje entre un punto y otro utilice como referencia la siguiente fórmula y los valores sugeridos:

$$Tiempo = \frac{Distancia\ a\ zona\ wifi}{Velocidad\ promedio.}$$

Velocidad prom. bus: 16,67 m/s – Velocidad prom. a pie: 0,483m/s

Velocidad prom. bici: 3,33m/s – Velocidad prom. moto: 19,44 m/s

Velocidad prom. auto: 20,83 m/s

## • FUNCION DISTANCIA

IMPORT MATH

```
def calcular_distacia(lat1, long1):  
    w_cerca=[[6.211,-72.482,2],[6.212,-72.470,25],[6.105,-72.342,25],[6.210,-72.442,50]]  
    dis=[]  
    for i in range(4):  
        rad = math.pi/180  
        dlat = w_cerca[i][0]-lat1  
        dlon= w_cerca[i][1]-long1  
        R = 6372.795477598  
        a = (math.sin(rad*dlat/2))**2 + math.cos(rad*lat1)*math.cos(rad*w_cerca[i][0])*(math.sin  
            (rad*dlon/2))**2  
        distancia = 2*R*math.asin(math.sqrt(a))  
        dis.append(distancia)  
    return dis
```