

# RIO MAROCASO

## Taller 6 – Modelo Heurístico

---

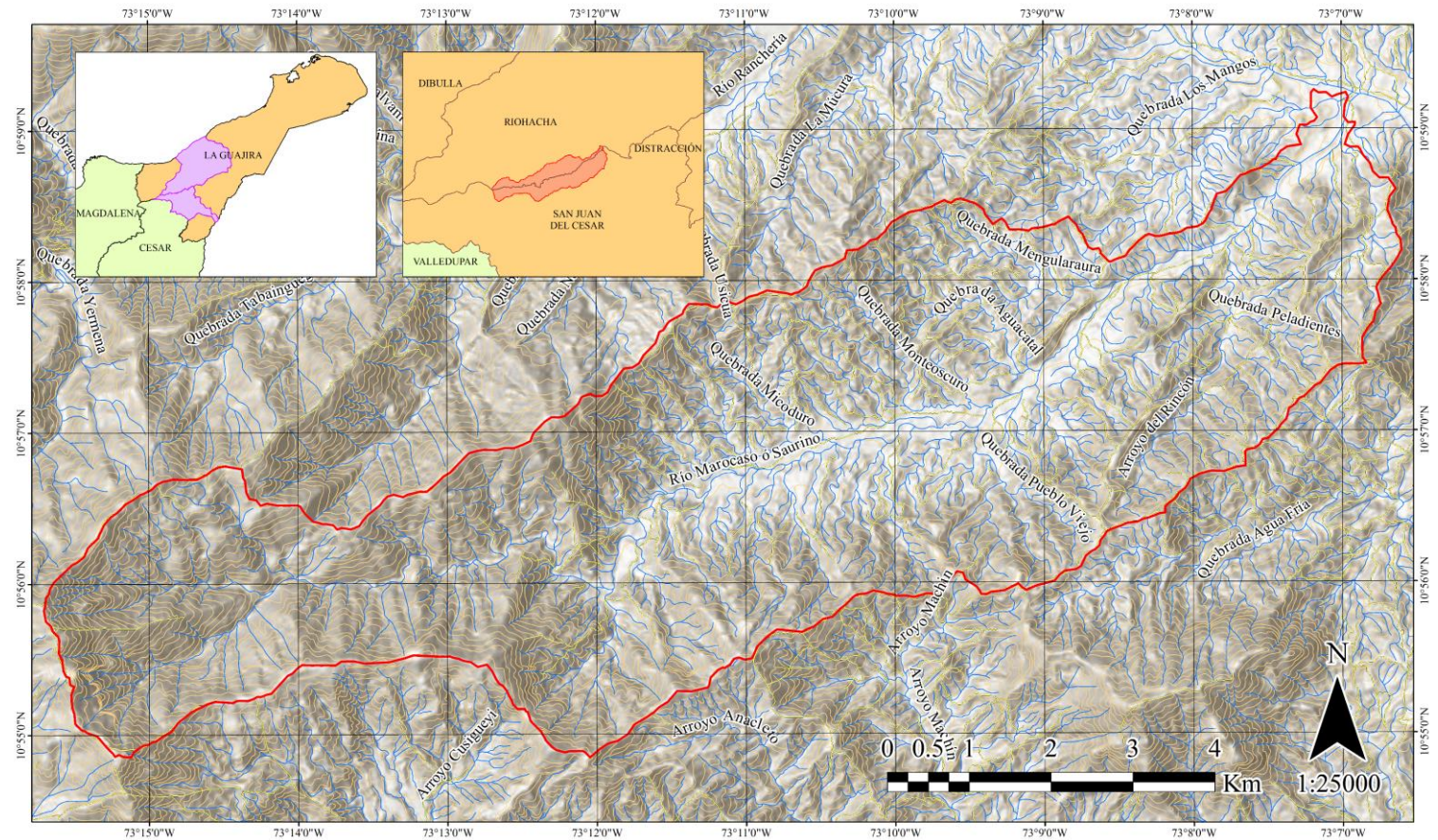
Moisés Carvajal Angarita

2023-S2

# Generalidades

La cuenca del río Marocaso se encuentra en el Departamento de la Guajira a una altitud de aproximadamente 658 m.s.n.m., entre los municipios de Rihoacha y San Juan del Cesar.

Nace directamente desde las montañas de la Sierra Nevada de Santa Marta y es un afluente directo del Rio Ranchería.



**Taller 1**  
**Cuenca del Rio Marocaso**  
**La Guajira**

**Moisés Carvajal Angarita**  
**Cartografía Geotécnica**  
**S2-2023**

**Leyenda**

Cuenca	Drenajes
Vias	Cotas

**Generalidades**  
Área: 55.4 Km<sup>2</sup>  
Perímetro: 46 Km  
Ancho: 4.3 Km  
Largo: 17 Km

# Modelo heurístico

Se planea principalmente realizar un Índice de Susceptibilidad, como se indica a continuación:

$$IS = W_1a_c + W_2b_c + W_3c_c$$

En donde:

- W es el peso de las variables, como por ejemplo la Geología.
- a,b y c es el peso de las clases, como, por ejemplo R. ígneas, sedimentarias y metamórficas.

Para establecer dichos pesos, se utilizarán 2 métodos en específico:

- **Método APH** para el peso de las variables.
- **Método de asignación directa** para el peso de las clases.

# Método de Análisis Jerárquico de Procesos (APH)

Se pretende hallar el Radio de Consistencia (CR), como se indica a continuación:

$$CR = \frac{CI}{ICA} \qquad \qquad \qquad CR < 0.1$$

En donde:

- CI: Índice de Consistencia.
- ICA: Índice de Consistencia aleatorio.

El Índice de Consistencia se establece así:

$$CI = \frac{ValorPropio_{max} - n}{n - 1}$$

El Índice de Consistencia aleatorio se establece

Matrix sizes (n)	1	2	3	4	5	6	7	8	9	10
RI Value	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

(iv) Data analysis: The degree of consistency is satisfactory when CR values are less than 0.1 (SAATY)

Con n = orden de la matriz-



# Método de Análisis Jerárquico de Procesos (APH)

Para hallar el valor propio máximo se crea la matriz con la escala de Saaty.

Jerarquización de variables		
Escala numérica	Escala verbal	Explicación
1	Igualmente preferida	Dos elementos contribuyen en igual medida al objetivo
2	Valor intermedio	
3	Moderadamente preferida	La experiencia y juicio favorecen levemente a un elemento sobre otro
4	Valor intermedio	
5	Fuertemente preferida	La experiencia y juicio favorecen fuertemente a un elemento sobre otro
6	Valor intermedio	
7	Preferencia muy fuerte o demostrada	Un elemento es mucho más favorecido que el otro, predominancia demostrada en la práctica
8	Valor intermedio	
9	Extremadamente preferida	La evidencia que favorece una sobre la otra es la más alta posible
<b>Valores recíprocos</b>	Cuando se asigna uno de los valores anteriores al elemento $i$ respecto de $j$ , el elemento $j$ tendrá el valor recíproco.	

$$A = \begin{pmatrix} \begin{bmatrix} 1 & 3 & 1/7 & 1/5 & 1 & 1/3 & 1 \\ 1/3 & 1 & 1/7 & 1/7 & 1/3 & 1/3 & 1 \\ 7 & 7 & 1 & 1 & 1 & 3 & 7 \\ 5 & 7 & 1 & 1 & 3 & 3 & 5 \\ 1 & 3 & 1 & 1/3 & 1 & 1/3 & 1 \\ 3 & 3 & 1/3 & 1/3 & 3 & 1 & 7 \\ 1 & 1 & 1/7 & 1/5 & 1 & 1/7 & 1 \end{bmatrix} \end{pmatrix}$$

Estos valores fueron asignados teniendo en cuenta los valores estadísticos arrojados en el taller 5.

Estos son los valores propios:

```
[ 7.63718313+0.j          -0.19089618+1.92305777j -0.19089618-1.92305777j
 -0.02397119+0.87613592j -0.02397119-0.87613592j -0.1037242  +0.49527008j
 -0.1037242  -0.49527008j]
```

# Método de Análisis Jerárquico de Procesos (APH)

Posteriormente, se halla CR como se muestra a continuación:

```
CI = (valores[0] - len(valores)) / (len(valores) - 1)

CR = CI / 1.32

if CR < 0.1:
    print('La designación de datos fue aceptable:\n', CR)
else:
    print('La designación de datos NO fue aceptable:\n', CR)

La designación de datos fue aceptable:
(0.08045241593263713+0j)
```

En donde valores[0] es el máximo valor propio, como se mostró en la diapositiva anterior.

Finalmente, se estipula que la matriz A se generó correctamente, ya que CR obtuvo un valor inferior al 10%

# Método de Análisis Jerárquico de Procesos (APH)

Finalmente, con el vector propio normalizado, se obtienen los pesos para cada variable.

```
geologia: (0.05927837617095107+0j)
geomorfologia: (0.034674763051279395+0j)
pendiente: (0.27646543364695597+0j)
curvatura: (0.2737535663357603+0j)
aspecto: (0.09508619201141219+0j)
flujo acumulado: (0.16176555802728804+0j)
altitud: (0.04616332948606875+0j)
```

# Método de asignación directa

Para este método, el experto deberá decidir que valores deberán asignarse a cada clase, Cómo se muestra a continuación.

```
geologiaReclass = np.where(geologiaB1 == 1, 0.1, geologiaB1)
geologiaReclass = np.where(geologiaB1 == 2, 1, geologiaReclass)
geologiaReclass = np.where(geologiaB1 == 3, 0.5, geologiaReclass)
```

```
geomorfologiaReclass = np.where(geomorfologiaB1 == 6, 0.1, geomorfologiaB1)
geomorfologiaReclass = np.where(geomorfologiaReclass == 7, 0.1, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 8, 0.3, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 4, 0.4, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 2, 0.4, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 1, 0.5, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 10, 0.7, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 5, 0.8, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 9, 1, geomorfologiaReclass)
geomorfologiaReclass = np.where(geomorfologiaReclass == 3, 1, geomorfologiaReclass)
```

```
# Geologia Conocida
# 1: Granito granofírico
# 2: Granito
# 3: Cuarzo monzonita
```

```
# Geomorfología Conocida
# 1: Ladera contrapendiente
# 2: Lomo de falla
# 3: Espolón moderado de longitud larga
# 4: Espolón bajo de longitud media
# 5: Lomo denudado bajo de longitud larga
# 6: Planicie aluvial confinada
# 7: Montículos y ondulaciones denudadas
# 8: Ladera ondulada
# 9: Espolón faceteado bajo de longitud media
# 10: Sierra
```



# Método de asignación directa

Para este método, el experto deberá decidir que valores deberán asignarse a cada clase, Como se muestra a continuación. Para los datos continuos, se tuvo en cuenta los Natural Breaks en los histogramas de densidad.

```
pendienteReclass = np.where ( (np.logical_and (pendienteB1 >= 0, pendienteB1 < 9 )), 0.1, pendienteB1 );
pendienteReclass = np.where ( (np.logical_and (pendienteReclass >= 7, pendienteReclass < 19)), 0.3, pendienteReclass);
pendienteReclass = np.where ( (np.logical_and (pendienteReclass >= 19, pendienteReclass < 29 )), 0.5, pendienteReclass);
pendienteReclass = np.where ( (np.logical_and (pendienteReclass >= 29, pendienteReclass < 36 )), 1, pendienteReclass);
pendienteReclass = np.where ( pendienteReclass >= 36, 0.8, pendienteReclass);
```

```
curvaturaReclass = np.where ( (np.logical_and (curvaturaB1 > p40, curvaturaB1 < p60)), 0.1, curvaturaB1)
curvaturaReclass = np.where(curvaturaReclass >= p60, 0.4, curvaturaReclass)
curvaturaReclass = np.where(curvaturaReclass <= p40, 1, curvaturaReclass)
```

```
aspectoReclass = np.where ( aspectoB1 < 125, 0.1, aspectoB1);
aspectoReclass = np.where ( (np.logical_and (aspectoReclass >= 125, aspectoReclass < 230)), 1, aspectoReclass)
aspectoReclass = np.where ( aspectoReclass >= 230, 0.4, aspectoReclass);
```

# Método de asignación directa

Para este método, el experto deberá decidir que valores deberán asignarse a cada clase, Como se muestra a continuación. Para los datos continuos, se tuvo en cuenta los Natural Breaks en los histogramas de densidad.

```
flujoAcumReclass = np.where ( flujoAcumB1 < 100, 0.3, flujoAcumB1)
flujoAcumReclass = np.where ( (np.logical_and (flujoAcumReclass >= 100, flujoAcumReclass < 6000)), 0.6, flujoAcumReclass)
flujoAcumReclass = np.where ( flujoAcumReclass > 6000, 0.1, flujoAcumReclass);
```

```
altitudReclass = np.where ( altitudB1 < 700, 0.1, altitudB1)
altitudReclass = np.where ( (np.logical_and (altitudReclass >= 700, altitudReclass < 1000)), 0.4, altitudReclass)
altitudReclass = np.where ( (np.logical_and (altitudReclass >= 1000, altitudReclass < 1700)), 0.9, altitudReclass)
altitudReclass = np.where ( (np.logical_and (altitudReclass >= 1700, altitudReclass < 2100)), 0.2, altitudReclass)
altitudReclass = np.where ( (np.logical_and (altitudReclass >= 2100, altitudReclass < 2550)), 0.7, altitudReclass)
altitudReclass = np.where ( altitudReclass >= 2550, 0.1, altitudReclass)
```

# Mapa de susceptibilidad

A continuación, se muestra el mapa de susceptibilidad obtenido haciendo uso del método heurístico.

