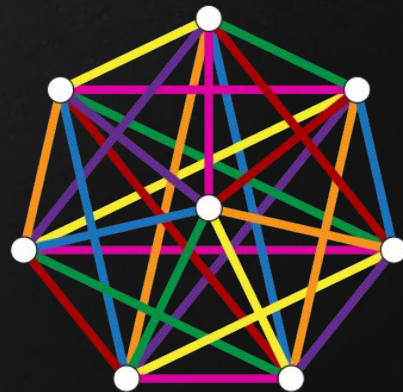




# Grafos, Dijkstra, R y otras

## historias donde aplicar caminos

### mínimos!





# HOLA!

## Soy Inés Huertas



Data Dreamer  
CEO en Datatons  
Amante de los unicornios  
Coorganizadora Rladies-Madrid  
Datanauts Open NASA  
Woman Startup Community Forum  
@quierodata



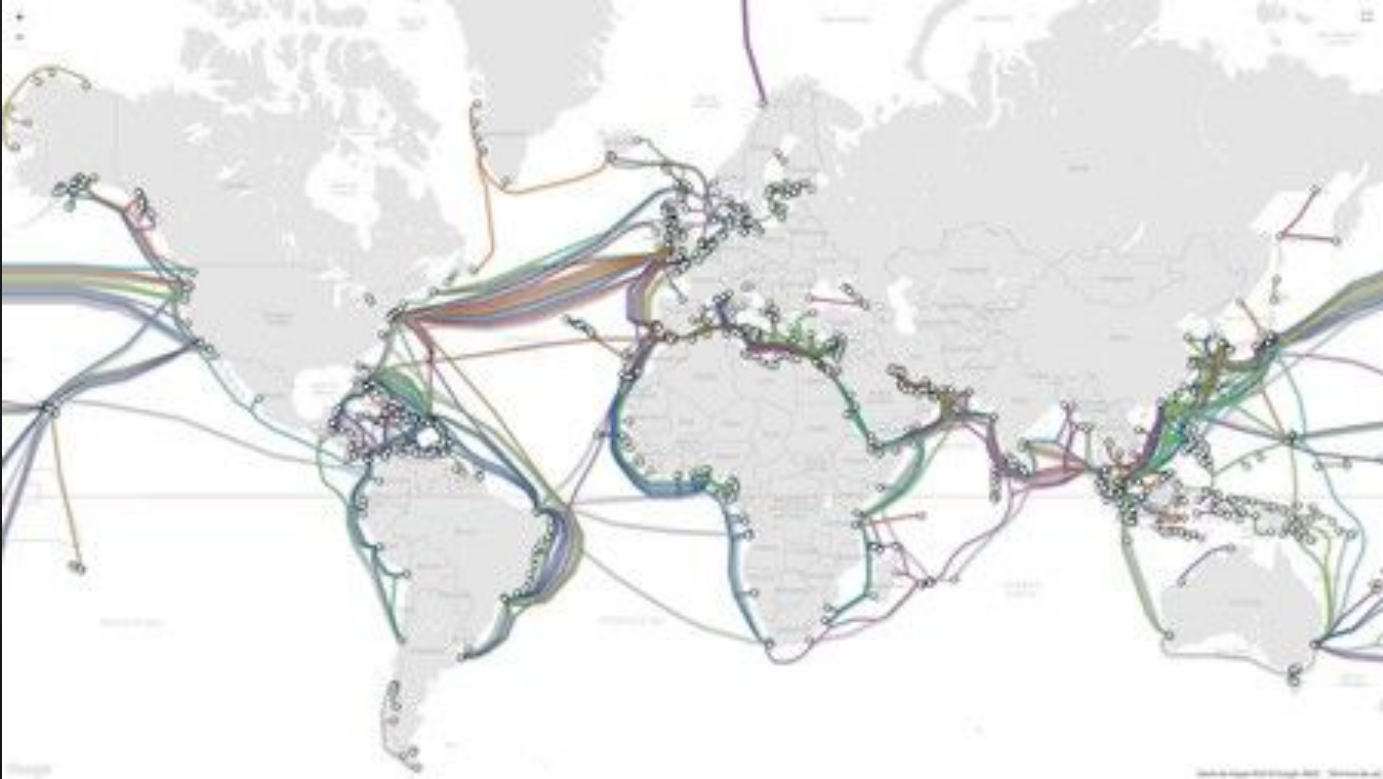


# Cómo funciona internet?





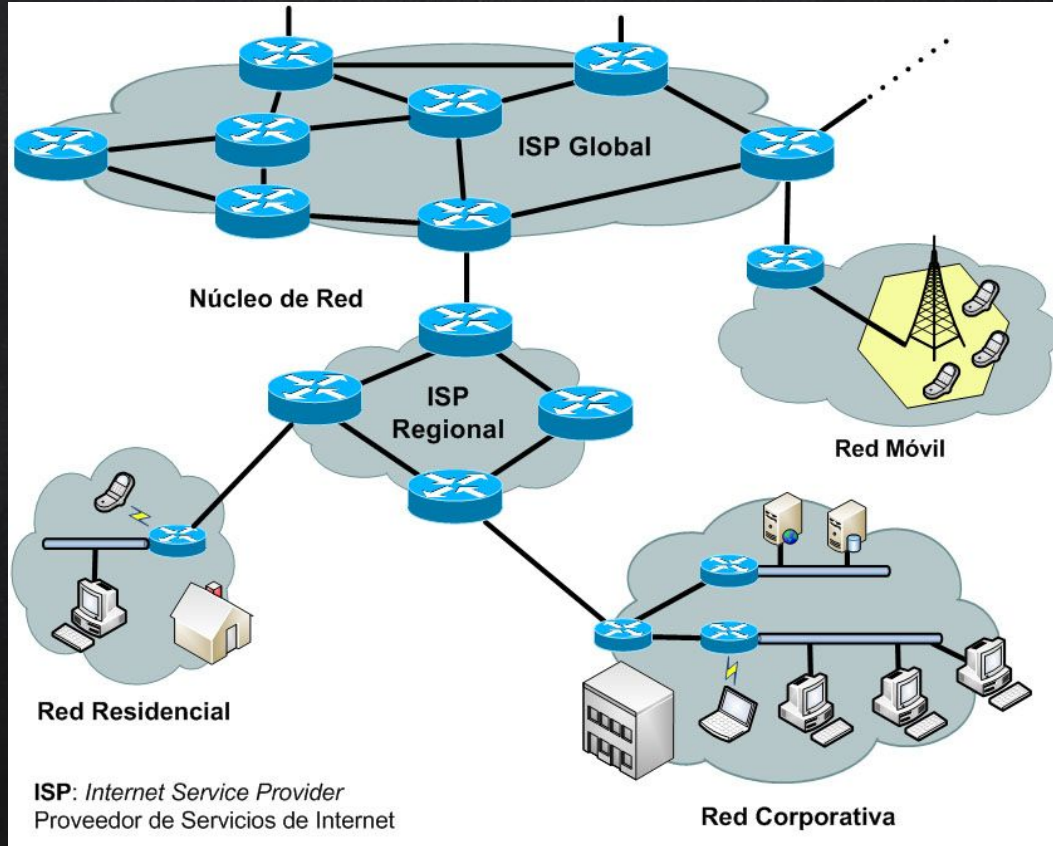
# Cómo funciona internet II





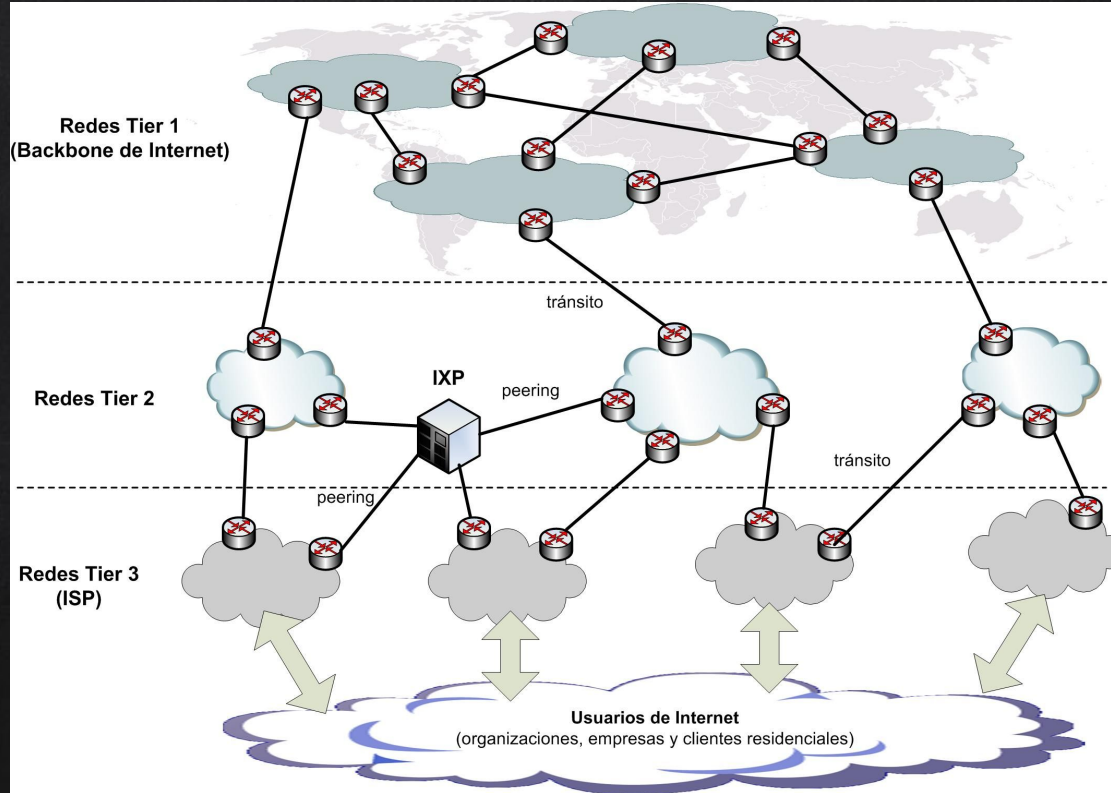


# Cómo funciona internet II





# Cómo funciona internet II



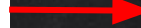


# Cómo funciona internet II





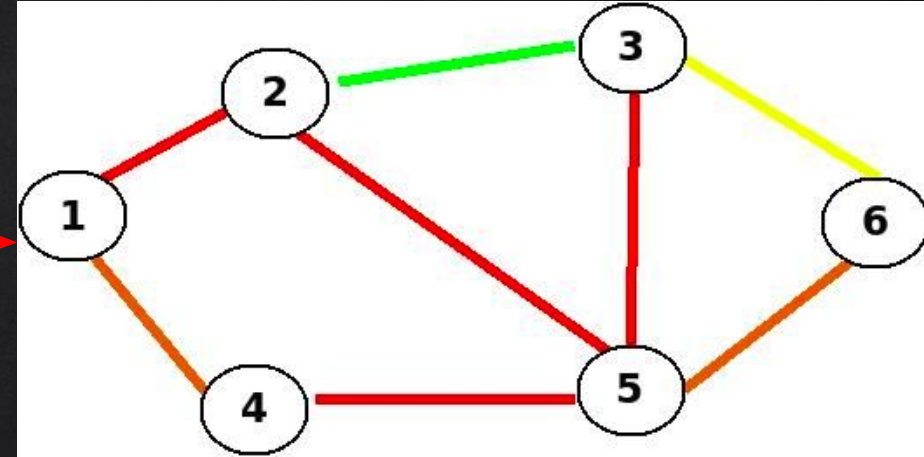
# Grafos







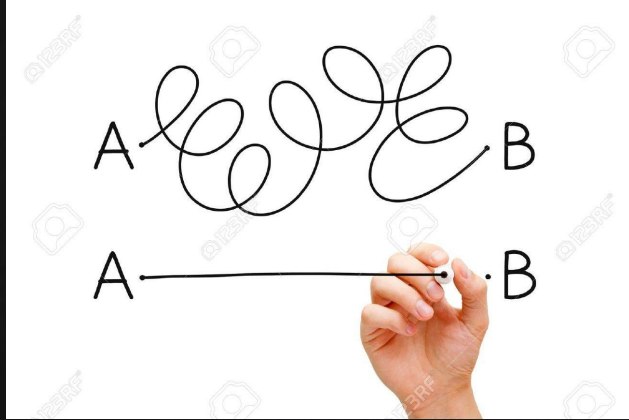
# Grafos con pesos



¿Camino de 1 a 6?



# Doroti!!! Sigue el camino de baldosas amarillas!



Distancia entre dos puntos puede ser medida en función:

- Del número de saltos
- Del estado de los enlaces



# ¿Para qué más se usa el camino mínimo de un grafo?



## Problemas reales en los que la solución es Dijkstra:

- Llegar a desde un punto de una ciudad hasta otro por el camino más rápido
- Estudios sobre la probabilidad de, por ejemplo, frases más usadas por los españoles.
- Cómo rodear una montaña por el camino más corto
- Conocer el camino más rápido que sigue la información a través de las neuronas.

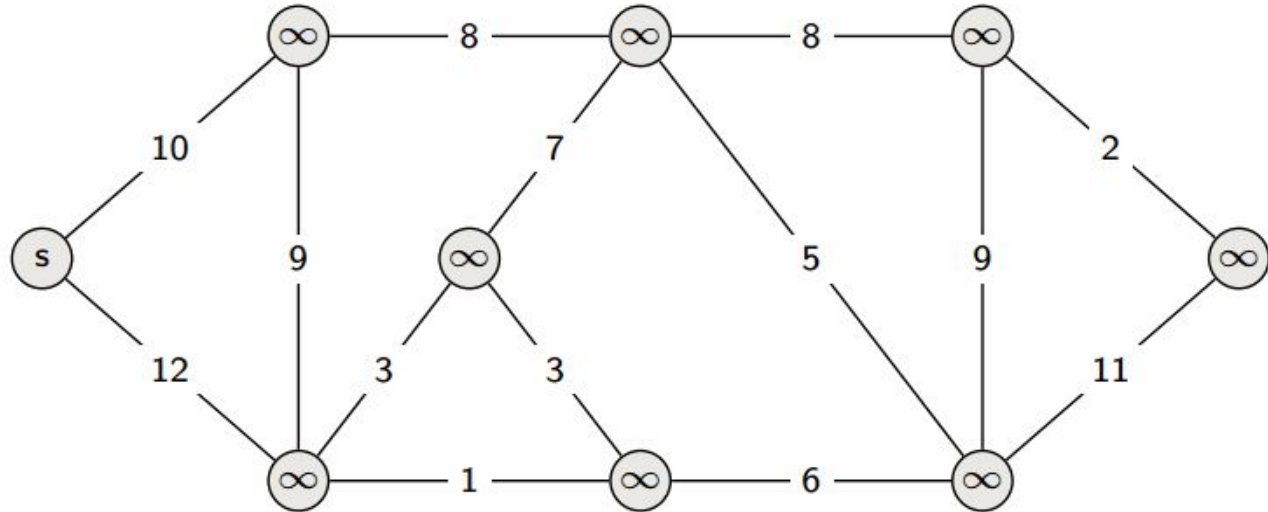


# Os presento a Dijkstra!!



Algoritmo de Dijkstra lo podemos aplicar sobre un grafo con topología conocida

Se aplica sobre cada uno de los nodos para que cada uno sepa enrutar







# Os presento a Dijkstra!!



---

## Paso 0. Inicialización:

Escoger un nodo como nodo de trabajo, examinar sus adyacentes y etiquetar cada uno con su distancia al nodo de referencia

---

## Paso 1. Encontrar el nodo más cercano:

Escoge el nodo con la etiqueta más pequeña y se hace nodo permanente, es el nuevo nodo de trabajo.

---

## Paso 2. Actualización de etiquetas:

Se suman las etiqueta del nuevo nodo y si su distancia al nodo adyacente es menor a la etiqueta del nodo tenemos nuevo camino

---

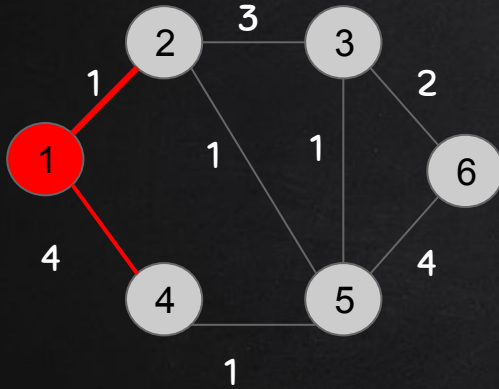
## Paso 3. Volver al paso 1:

Se elige el nodo con la etiqueta más pequeña y se hace nodo permanente, es el nuevo nodo de trabajo

---



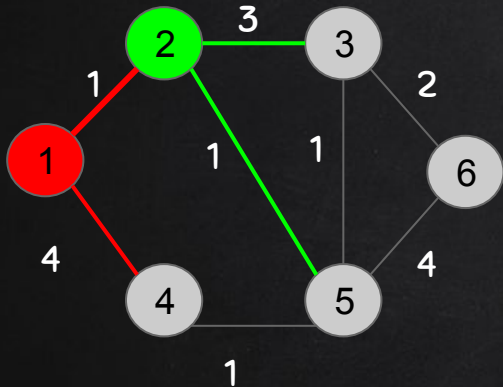
# Destripando Dijkstra



NODO 1	NODO 2	NODO 3	NODO 4	NODO 5	NODO 6
P(1) T(2,3,4,5,6)	1 N2	INF ----	4 N4	INF ---	INF ---
	1 N2				



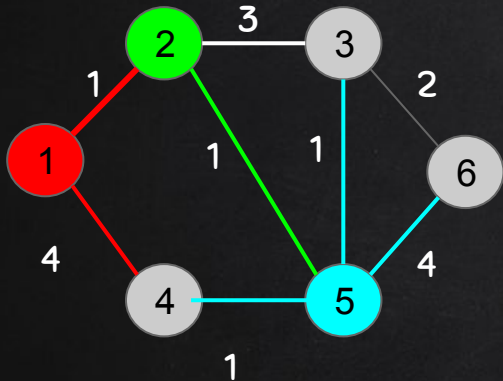
# Destripando Dijkstra



NODO 1	NODO 2	NODO 3	NODO 4	NODO 5	NODO 6
P(1) T(2,3,4,5,6)	1 N2	INF ----	4 N4	INF ---	INF ---
P(1,2) T(3,4,5,6)	1 N2	4 N2,N3	4 N4	2 N2,N5	INF ---
	1 N2			2 N2,N5	



# Destripando Dijkstra

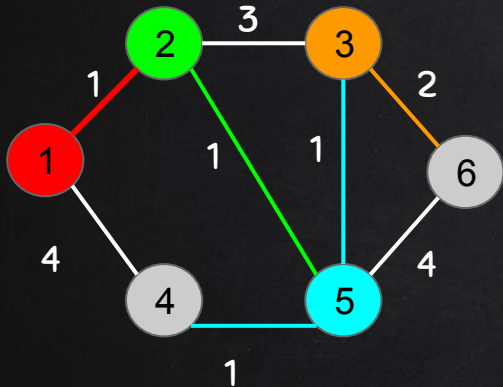


NODO 1	NODO 2	NODO 3	NODO 4	NODO 5	NODO 6
P(1) T(2,3,4,5,6)	1 N2	INF ----	4 N4	INF ---	INF ---
P(1,2) T(3,4,5,6)	1 N2	4 N2,N3	4 N4	2 N2,N5	INF ---
P(1,2,5) T(3,4,6)	1 N2	3 N2,N5,N3	3 N2,N5,N4	2 N2,N5	6 N2,N5,N6
	1 N2	3 N2,N5,N3	3 N2,N5,N4	2 N2,N5	





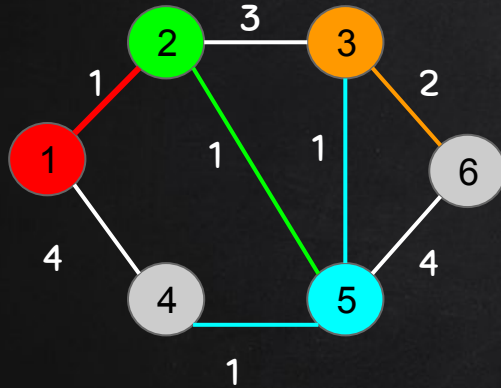
# Destripando Dijkstra



NODO 1	NODO 2	NODO 3	NODO 4	NODO 5	NODO 6
P(1) T(2,3,4,5,6)	1 N2	INF ----	4 N4	INF ---	INF ---
P(1,2) T(3,4,5,6)	1 N2	4 N2,N3	4 N4	2 N2,N5	INF ---
P(1,2,5) T(3,4,6)	1 N2	3 N2,N5,N3	3 N2,N5,N4	2 N2,N5	6 N2,N5,N6
P(1,2,5,3) T(6)	1 N2	3 N2,N5,N3	3 N2,N5,N4	2 N2,N5	5 N2,N5,N3,N6



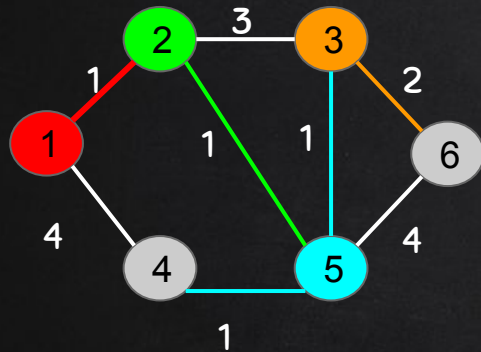
# Destripando Dijkstra



CAMINOS MINIMOS NODO 1	SIGUIENTE	MÉTRICA
NODO 2	NODO 2	1
NODO 3	NODO 2	3
NODO 4	NODO 2	3
NODO 5	NODO 2	2
NODO 6	NODO 2	5



# Dijkstra con R

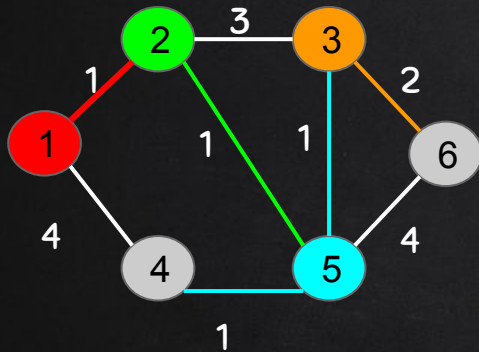


```
1 library(igraph)
2
3 # Nuestros datos
4 mat <- as.matrix(read.table(text="node X1 X2 X3 X4 X5 X6
5   1  0  1 NA  4 NA NA
6   2  2  0  3 NA  1 NA
7   3 NA  3  0 NA  1  2
8   4  4 NA NA  0  1 NA
9   5 NA  1  1  1  0  4
10  6 NA NA  2 NA  4  0", header=T))
11
```

```
> mat
      node X1 X2 X3 X4 X5 X6
[1,]    1  0  1 NA  4 NA NA
[2,]    2  2  0  3 NA  1 NA
[3,]    3 NA  3  0 NA  1  2
[4,]    4  4 NA NA  0  1 NA
[5,]    5 NA  1  1  1  0  4
[6,]    6 NA NA  2 NA  4  0
>
```



# Dijkstra con R



```
11
12 # prepare data for graph functions - set NA to zero to indicate no direct edge
13 nms <- mat[,1]
14 mat <- mat[, -1]
15 colnames(mat) <- rownames(mat) <- nms
16 mat[is.na(mat)] <- 0
17
18
```

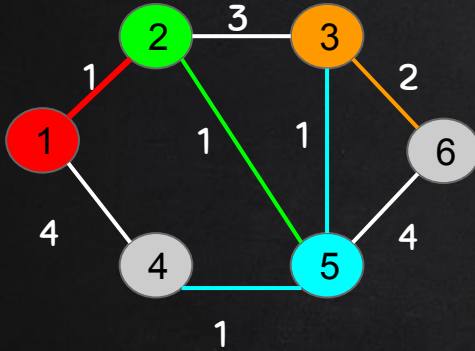
```
> mat
  1 2 3 4 5 6
1 0 1 0 4 0 0
2 2 0 3 0 1 0
3 0 3 0 0 1 2
4 4 0 0 0 1 0
5 0 1 1 1 0 4
6 0 0 2 0 4 0
> |
```

```
> mat
      node X1 X2 X3 X4 X5 X6
[1,]    1  0  1 NA  4 NA NA
[2,]    2  2  0  3 NA  1 NA
[3,]    3 NA  3  0 NA  1  2
[4,]    4  4 NA NA  0  1 NA
[5,]    5 NA  1  1  1  0  4
[6,]    6 NA NA  2 NA  4  0
> |
```





# Dijkstra con R



```
20
21 # creamos grafo desde la matriz de adyacencia
22 g <- graph.adjacency(mat, weighted=TRUE)
23 g
24
```

```
> g
IGRAPH 81705be DNW- 6 16 --
+ attr: name (v/c), weight (e/n)
+ edges from 81705be (vertex names):
[1] 1->2 1->4 2->1 2->3 2->5 3->2 3->5 3->6 4->1 4->5 5->2 5->3 5->4 5->6 6->3 6->5
> |
```



# Preguntas?!!



INÉS HUERTAS FREIRE

INES@RLADIES.ORG

@QUIERODATA