

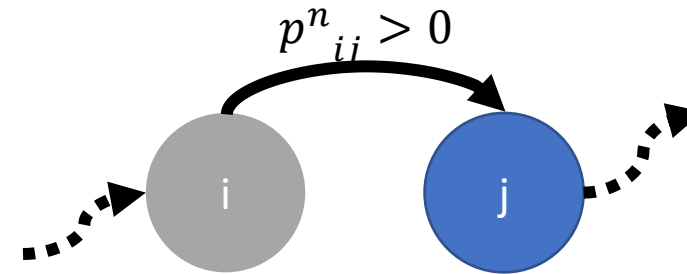
# Tipos de Cadenas de Markov y estado estacionario

Rodrigo Maranzana

# Clasificación de estados

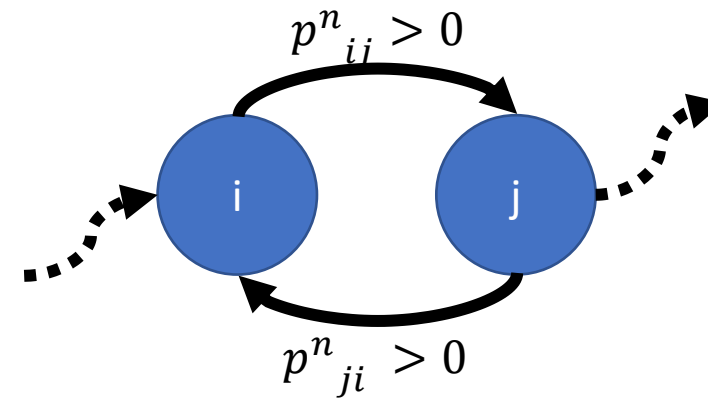
## Estado accesible:

Se puede acceder a estado "j" desde "i" en n pasos.



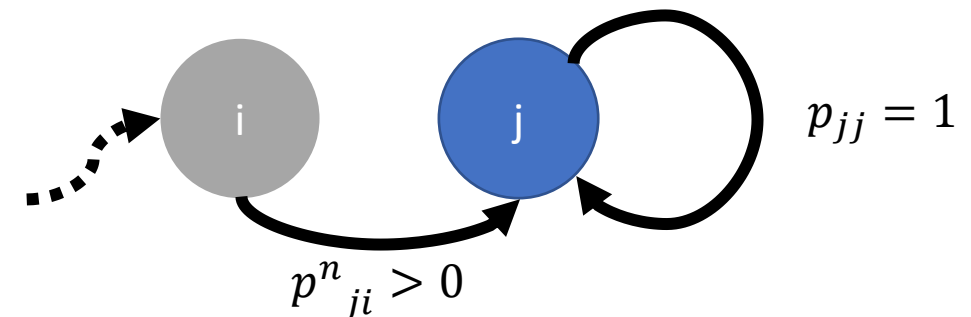
## Estados comunicantes:

Ambos estados "i", "j" son accesibles entre sí.

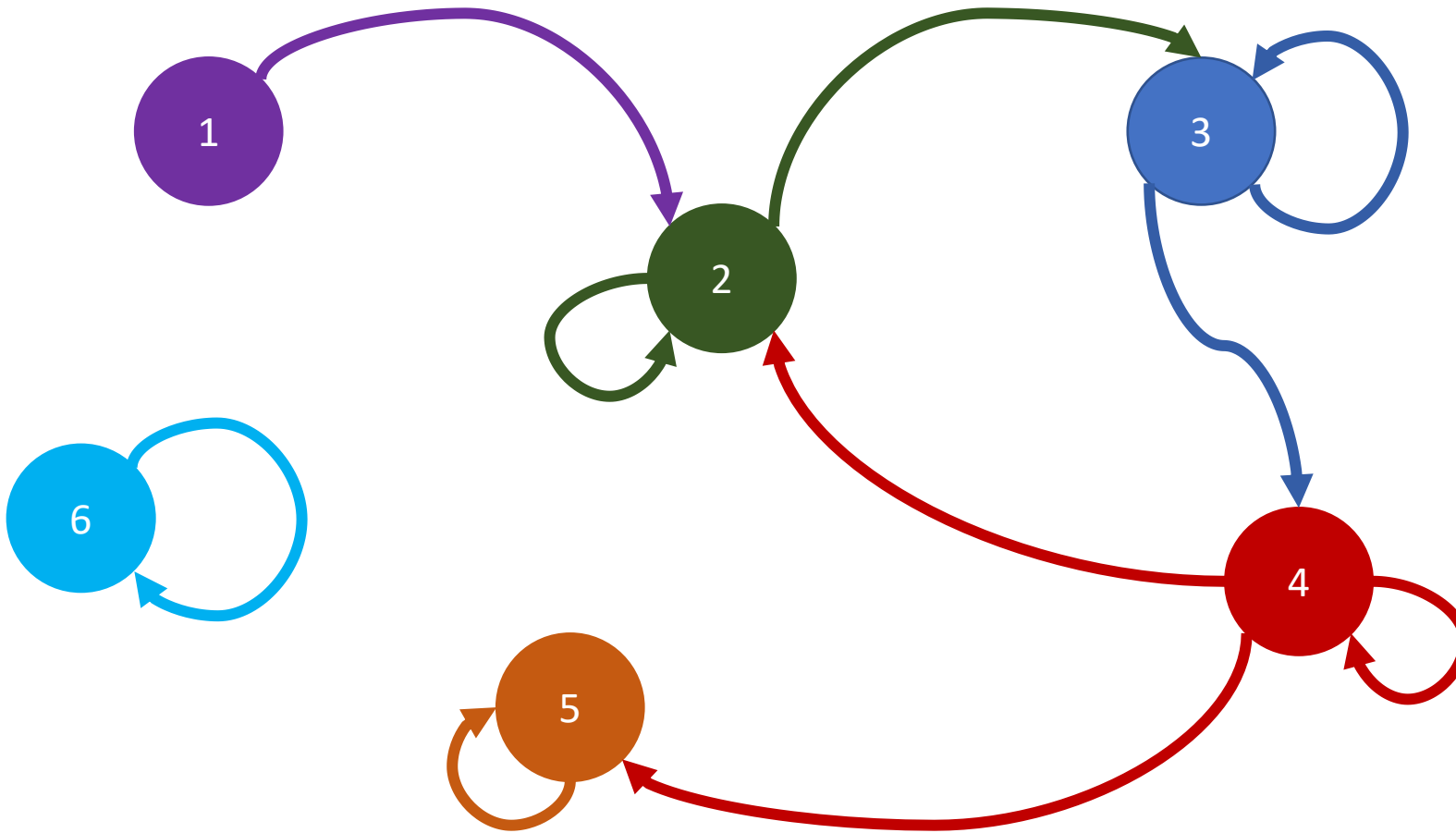


## Estado absorbente:

Una vez alcanzado no se puede escapar de él.



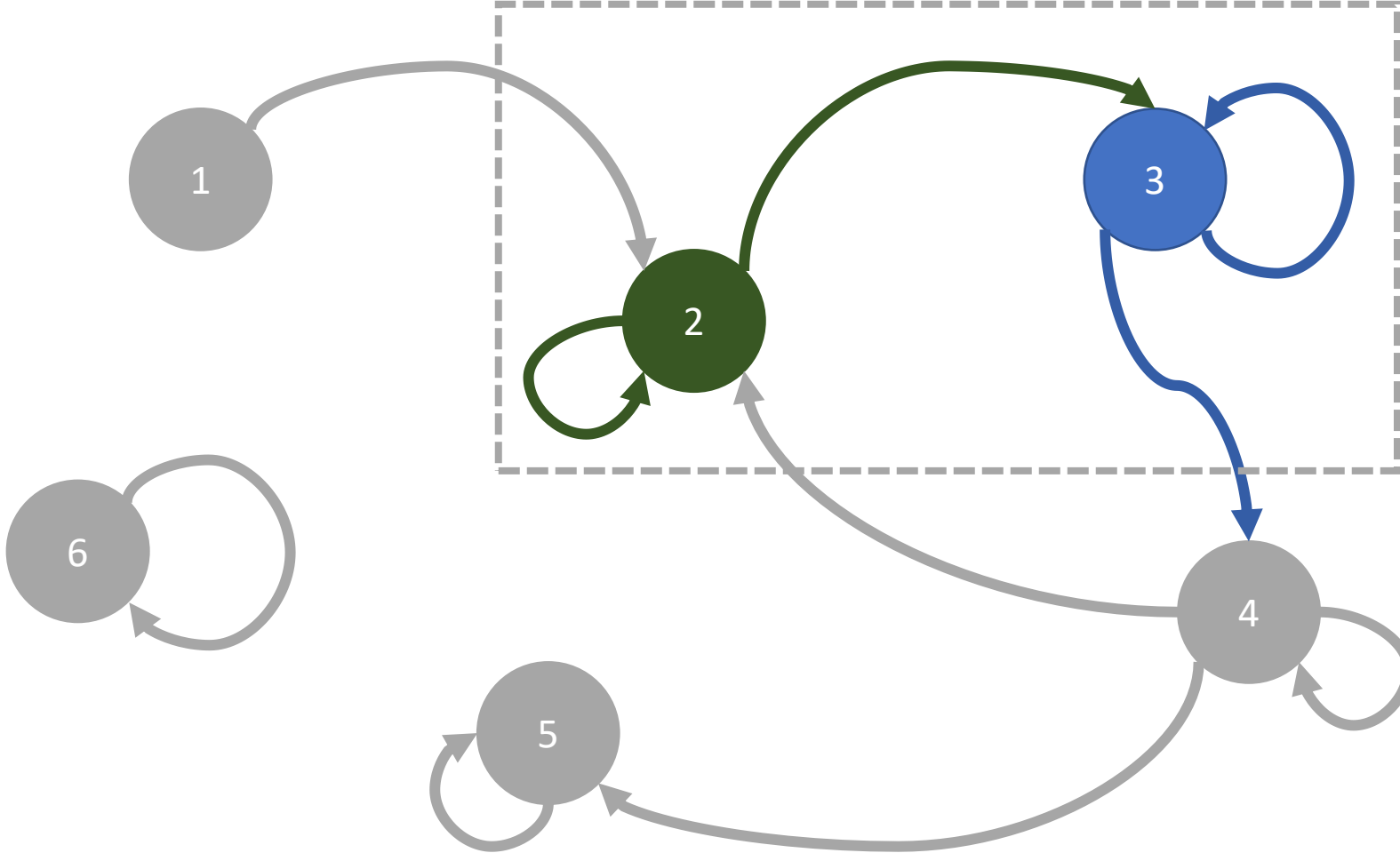
# Clasificación de estados



- Estados 1 y 6 *no son accesibles* desde ningún otro estado
- Estado 2 *es accesible* desde:
  - Estado 3
  - Estado 4
  - Estado 1
- Estado 2 y 4 *son comunicantes*.
- Estado 5 *es absorbente*.

# Clasificación de clases

Ejemplo de clase

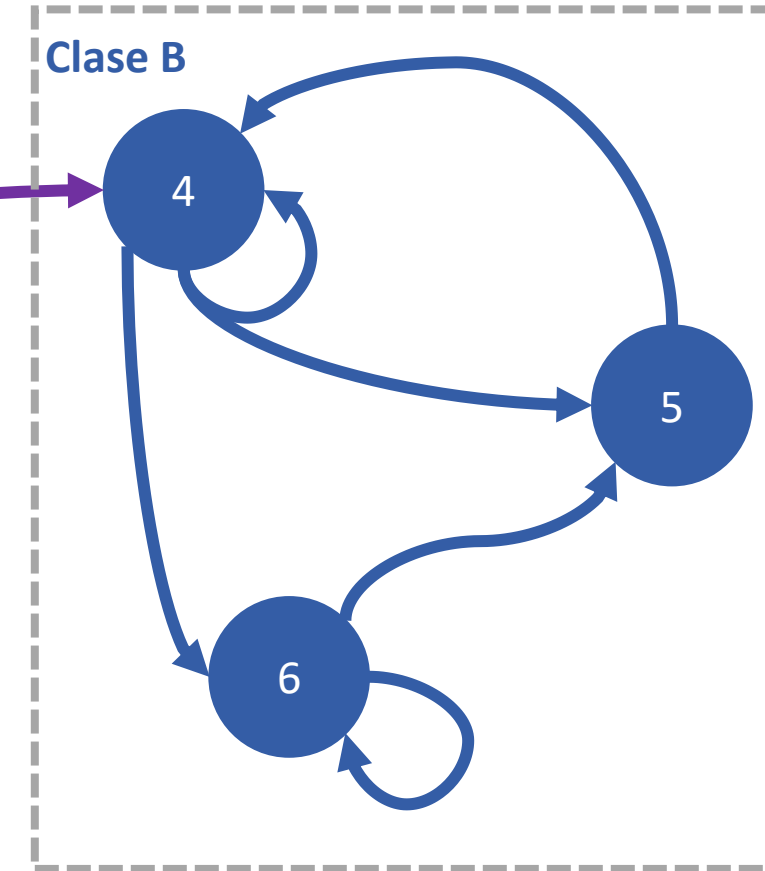
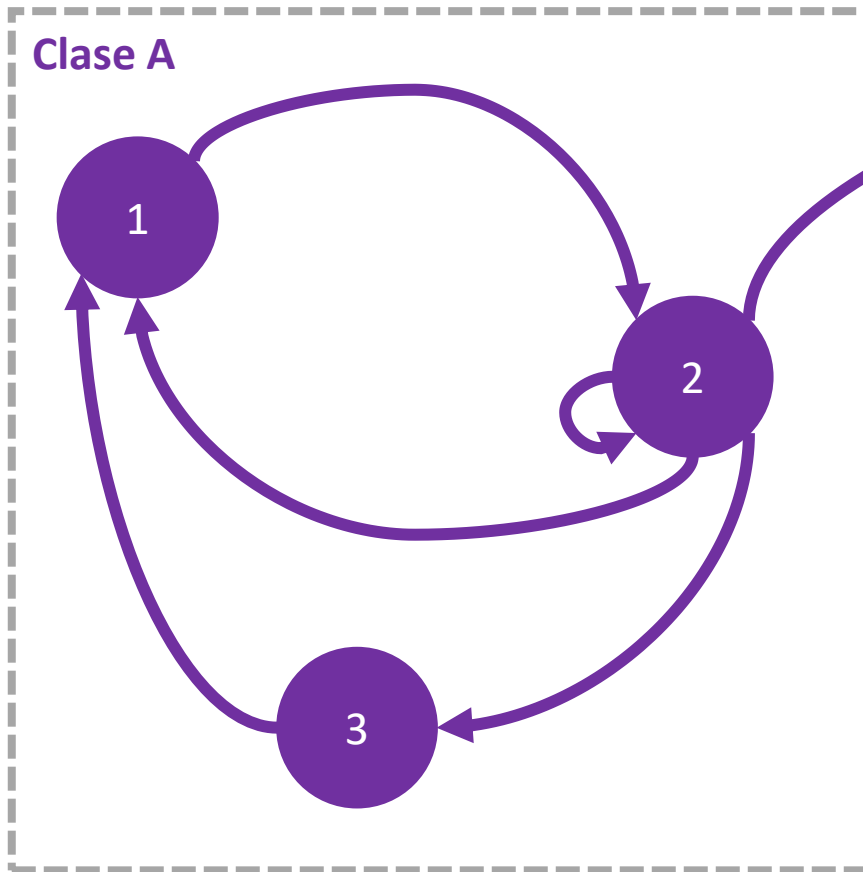


- Clase: conjunto de estados.

# Clasificación de estados

Clase comunicante:

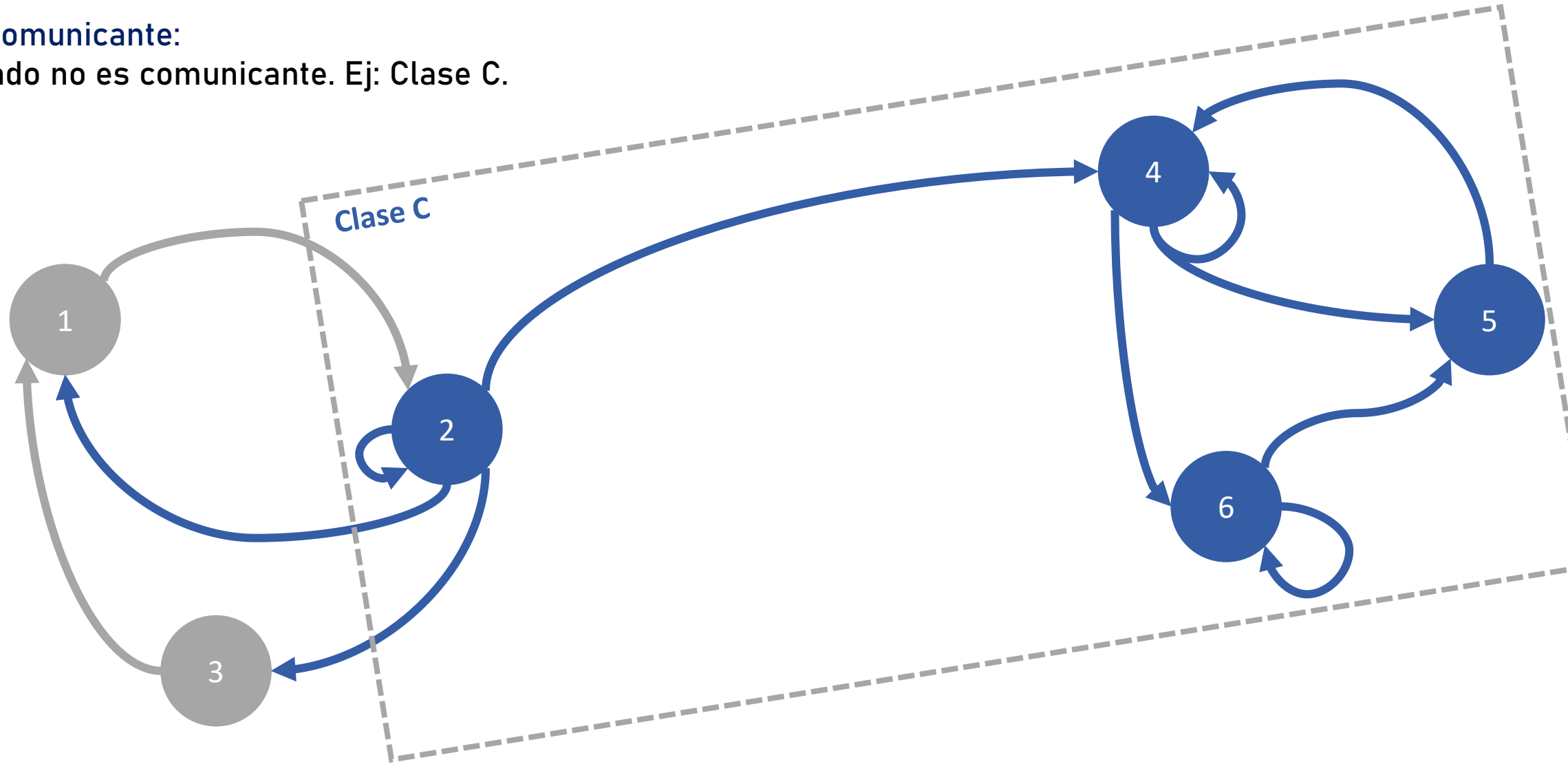
Todos sus estados son comunicantes. Ej: Clase A y B.



# Clasificación de estados

Clase no comunicante:

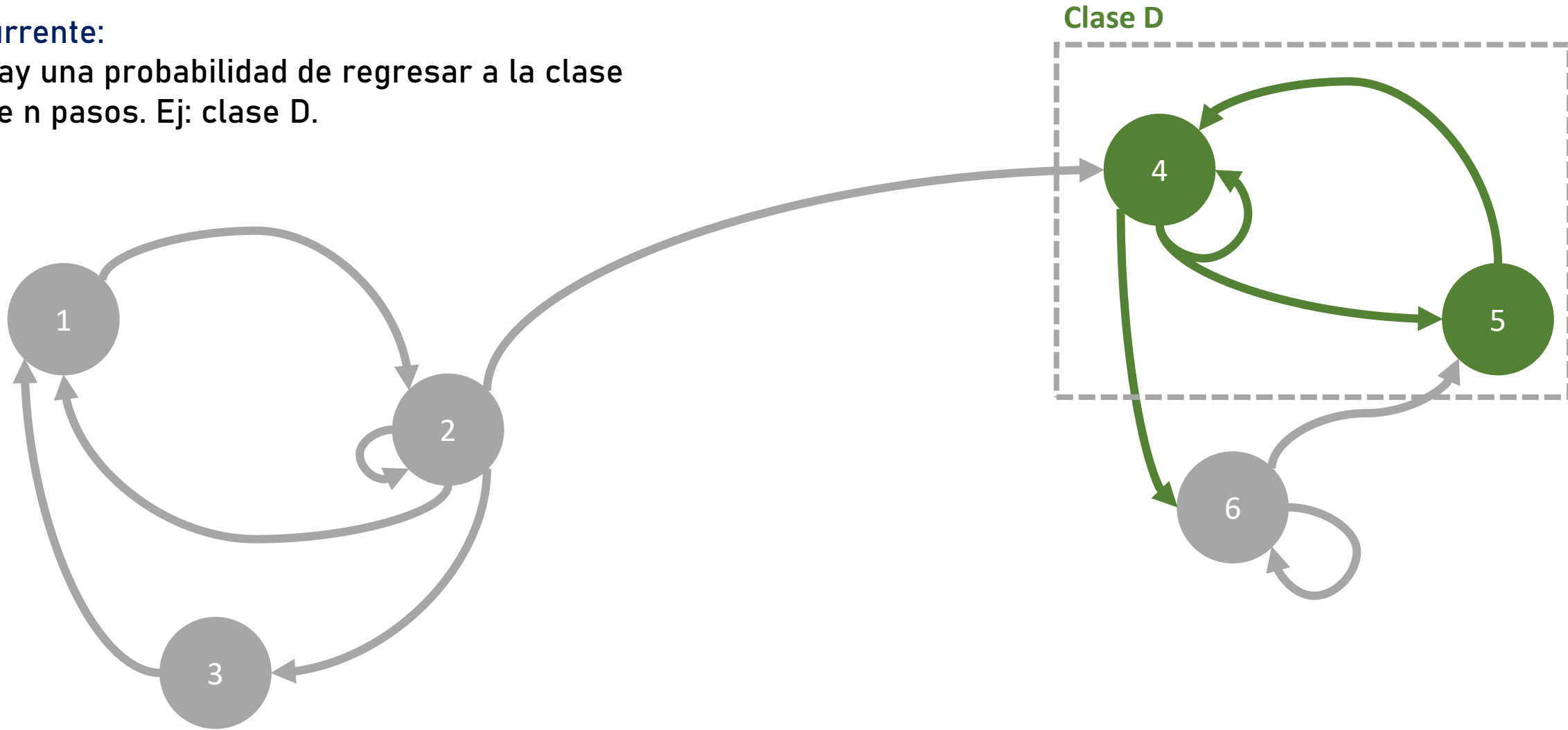
Algún estado no es comunicante. Ej: Clase C.



# Clasificación de estados

## Clase recurrente:

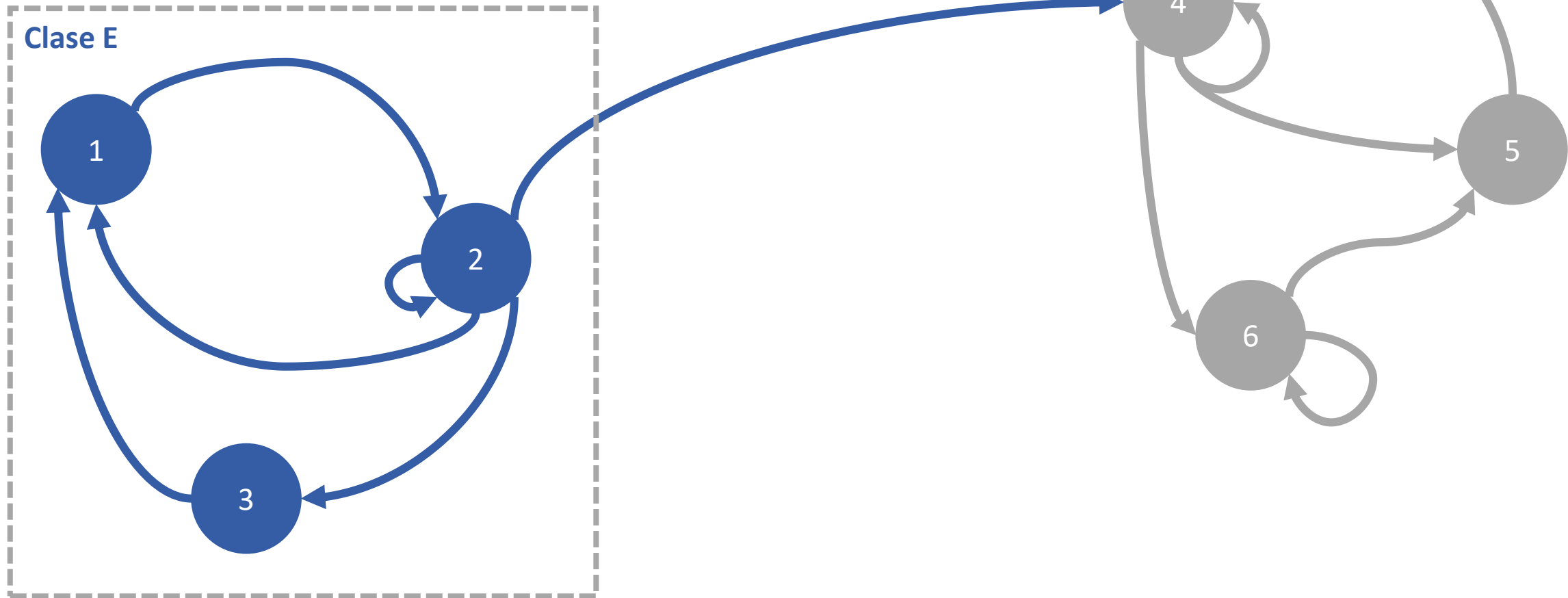
Siempre hay una probabilidad de regresar a la clase después de  $n$  pasos. Ej: clase D.



# Clasificación de estados

Clase transitoria:

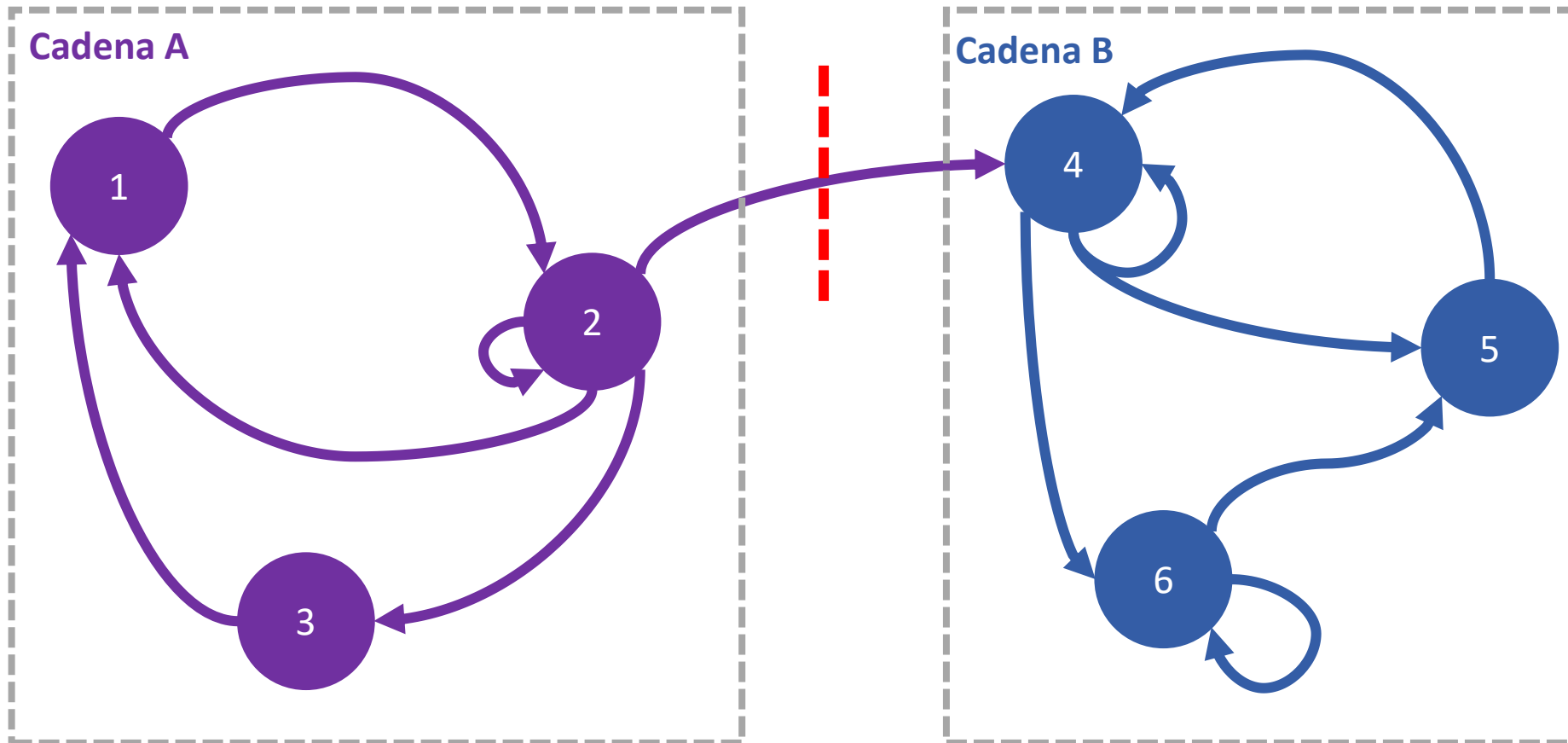
Existe la probabilidad de nunca regresar a la clase después de  $n$  pasos. Ej: clase E.





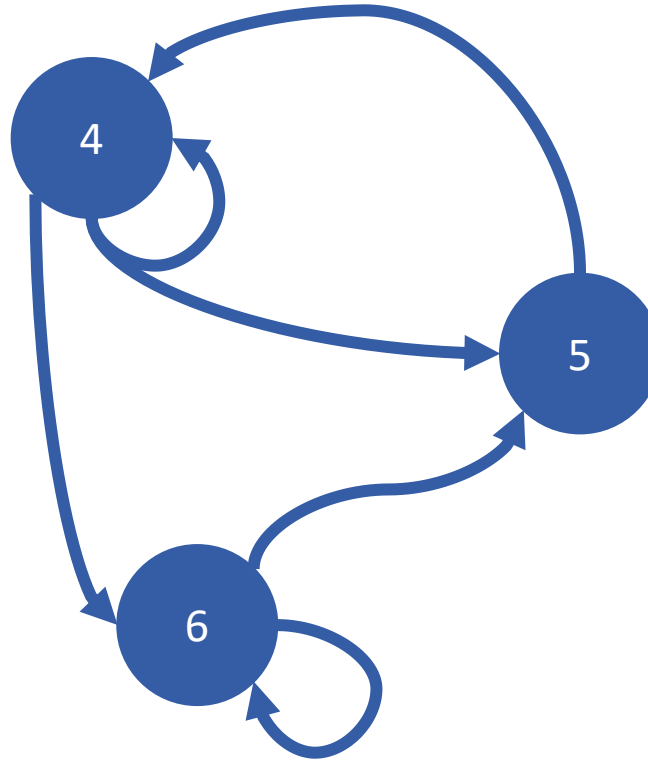
# Reducibilidad

Una cadena es **reducible**, si se puede descomponer en dos o mas subgrafos, entre los cuales, es imposible transicionar en “n” pasos.



# Reducibilidad

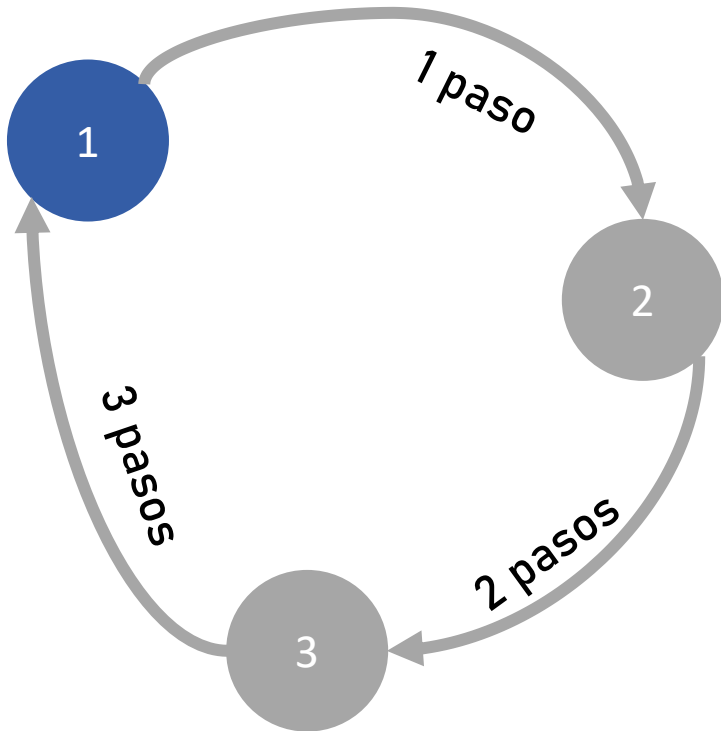
Una cadena es **irreducible**, si está formada por estados comunicantes, es decir, son recurrentes.



# Periodicidad

El período  $d(i)$  de un estado “i”, es el número de pasos que le toma a una cadena regresar al mismo estado “i”.  $d(i) \in \mathbb{N}$

Es el **Máximo Común Divisor** de todas las iteraciones “k” hasta lograr volver al estado.

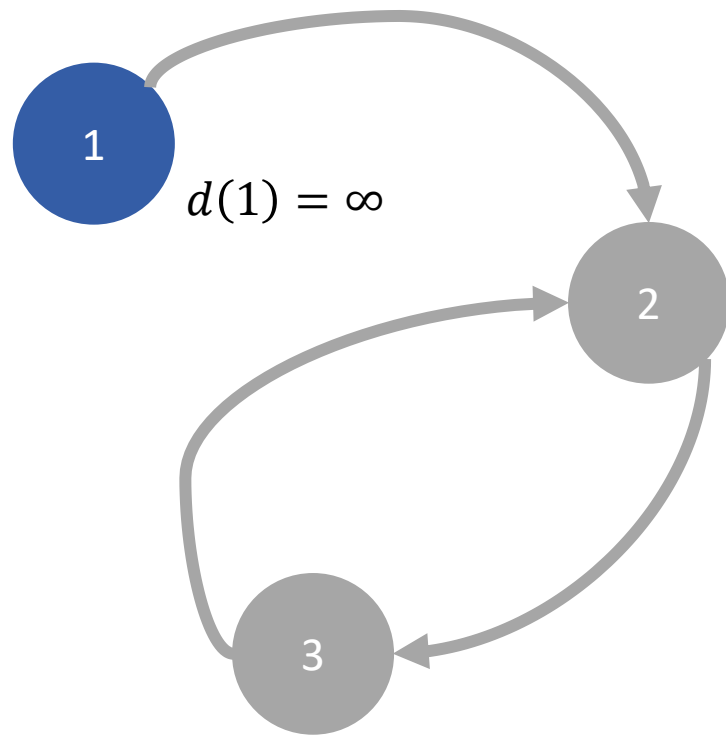


En este caso,  $p^n_{11} > 0$ , y retorna en  $k = \{3, 6, 9, \dots, 3 * (vuelta)\}$

Por lo tanto:  
 $d(1) = 3$

# Periodicidad

Si un estado no es accesible a sí mismo en “n” pasos (transitorio). Es decir,  $p^n_{ii} = 0$  el período resulta:

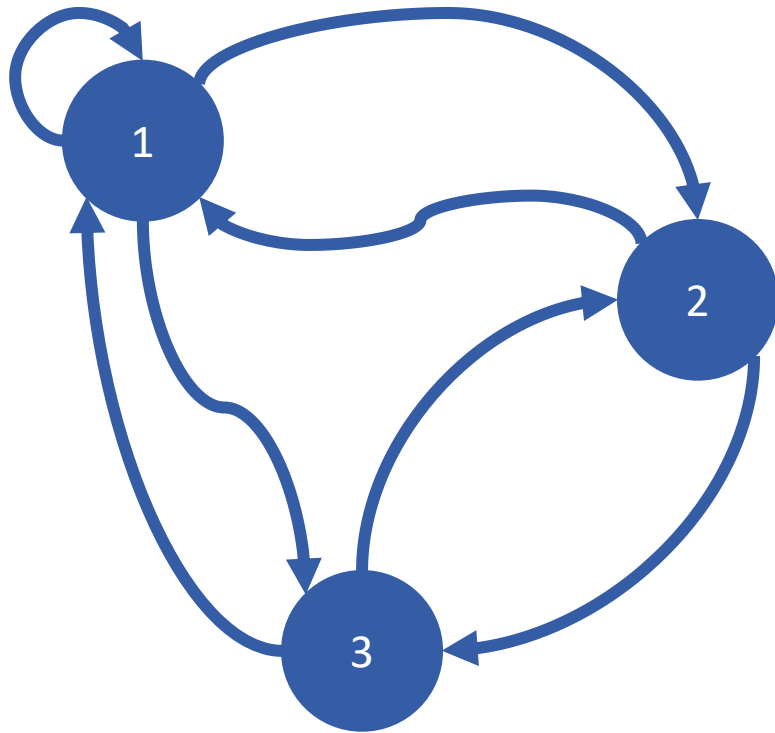


$$d(i) = \infty$$

# Periodicidad

Si el período  $d(i) = 1$ , se dice que el estado es aperiódico

Ej:  $d(1) = 1$



Si una cadena tiene todos sus estados aperiódicos, se dice que la **cadena de Markov es aperiódica**.

En una **cadena irreducible** basta con que un estado sea aperiódico para que toda la cadena sea aperiódica.

En una **cadena irreducible** periódica todos los estados tienen el mismo período.

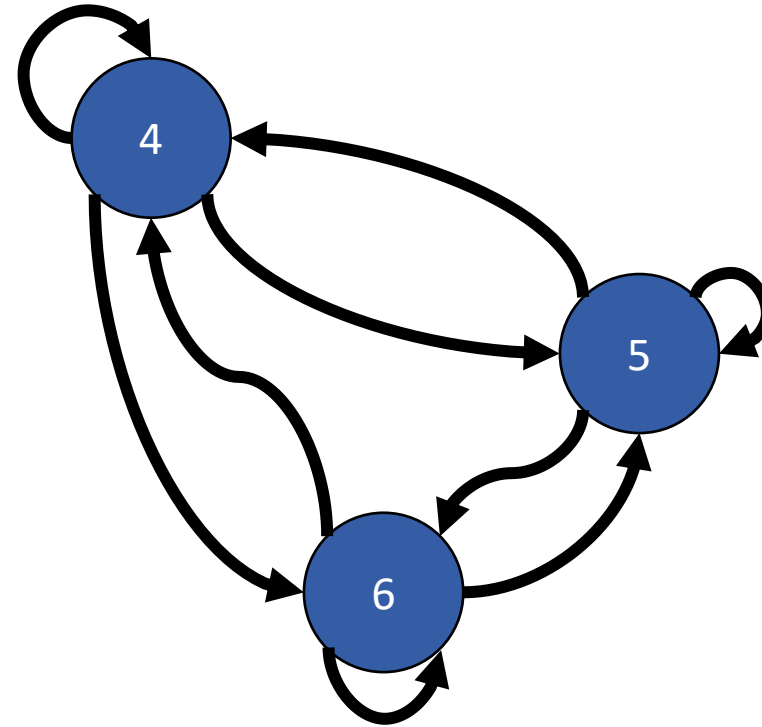
# Cadenas de Markov Ergódicas

Una cadena es **ergódica** si:

- Es irreducible: es una clase comunicante.
- Es aperiódica: no existe periodicidad.

¿Por qué es útil esta categoría?

**Garantiza la convergencia al estado estacionario, único, desde cualquier distribución inicial.**



# Estado estacionario (steady-state)

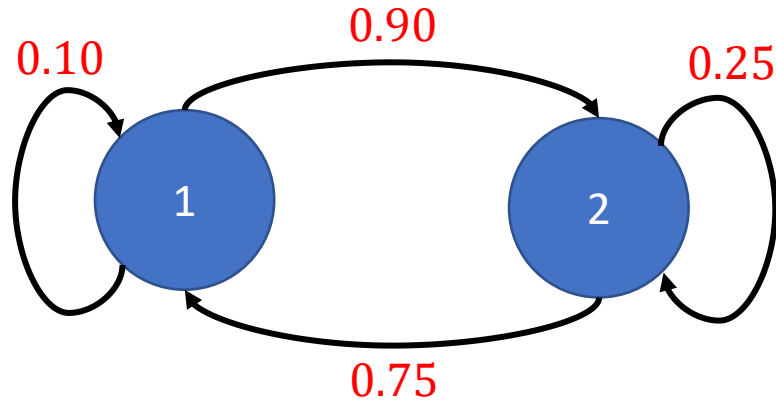
- Es un **estado de equilibrio**, en el que la probabilidad de estado se mantiene constante a través del tiempo, sin importar el estado inicial.

Cadenas **sin estado estacionario único**:

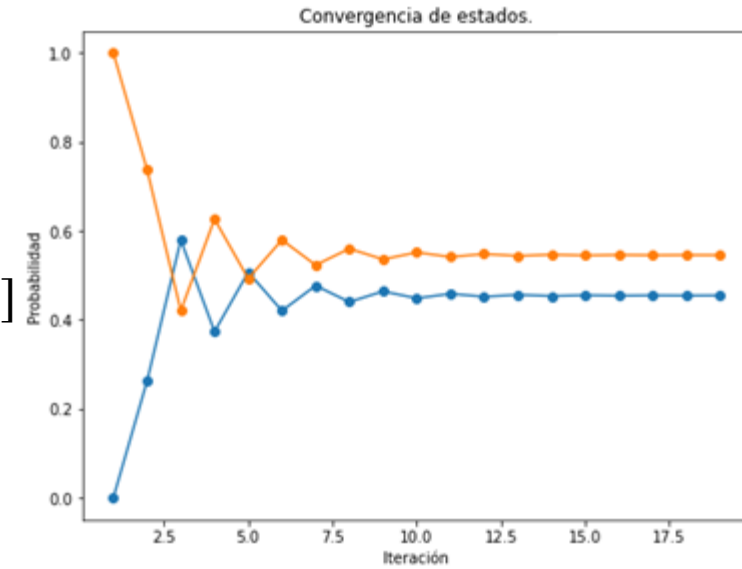
- Carecen de estado estacionario.
- Tienen distintos estados de convergencia dependiendo de la posición de inicio.

# Estado estacionario (steady-state)

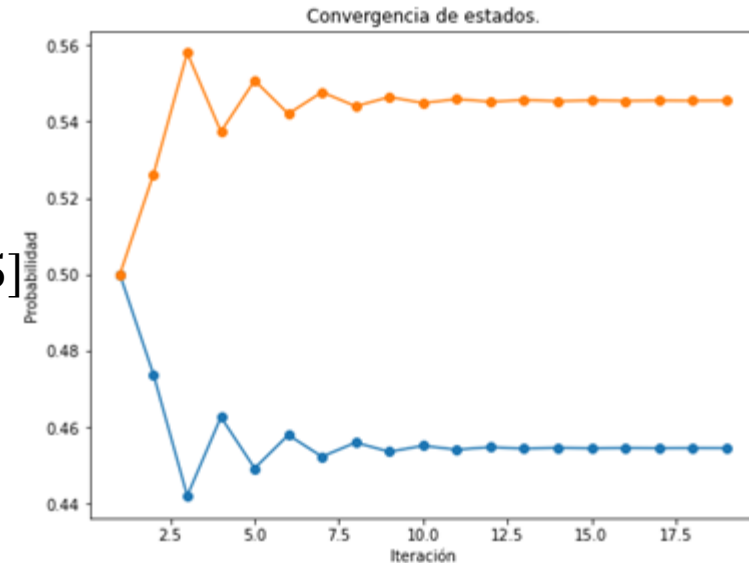
## ■ Grafo con estado estacionario:



$$p_0 = [1.0, 0.0]$$



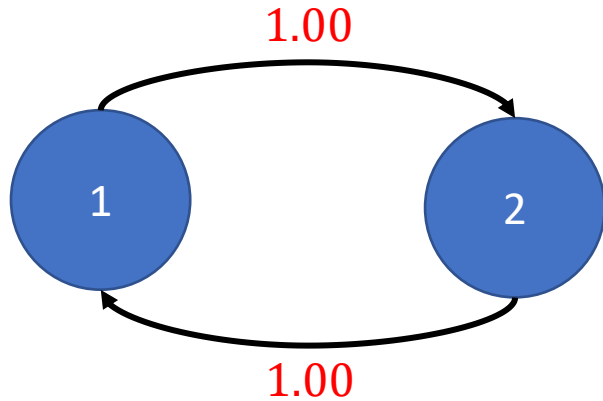
$$p_0 = [0.5, 0.5]$$



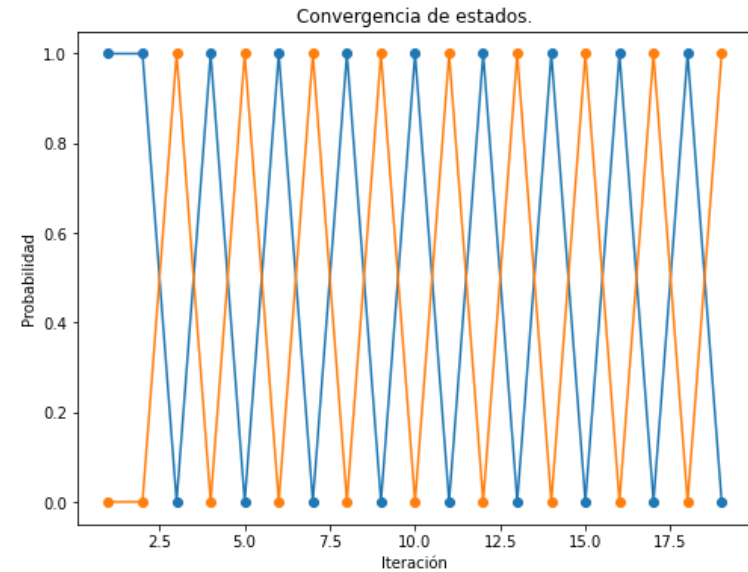


# Estado estacionario (steady-state)

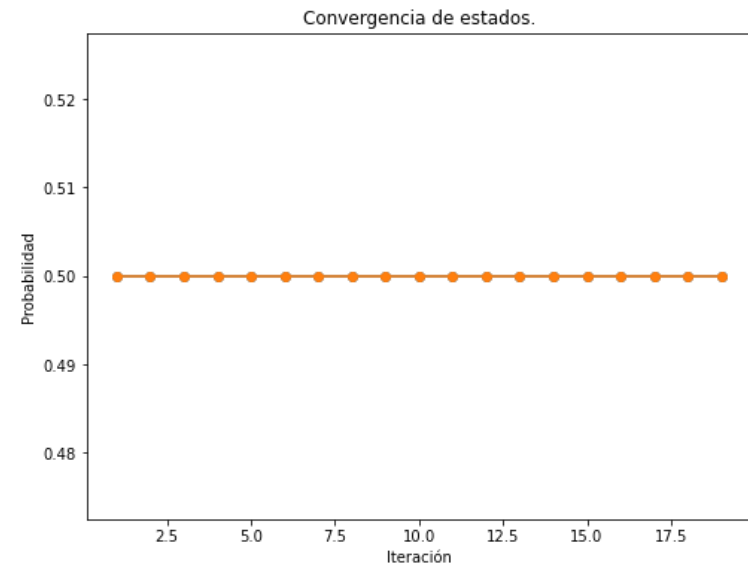
## ■ Grafo sin estado estacionario:



$$p_0 = [1.0, 0.0]$$

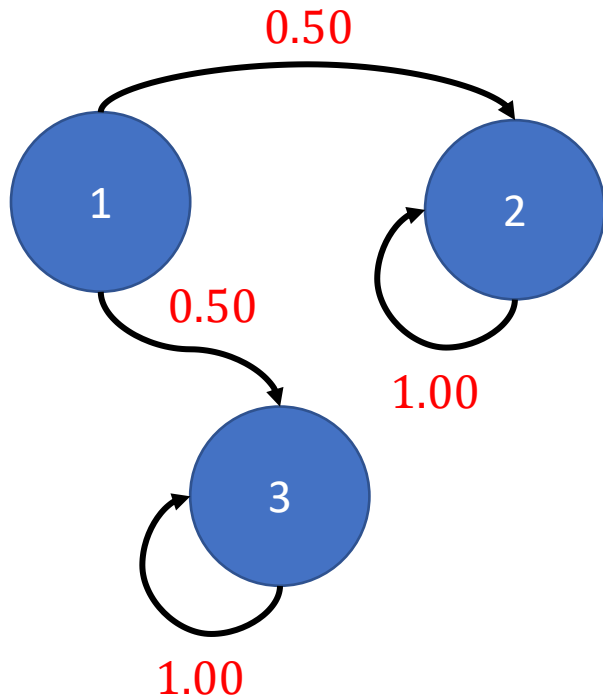


$$p_0 = [0.5, 0.5]$$



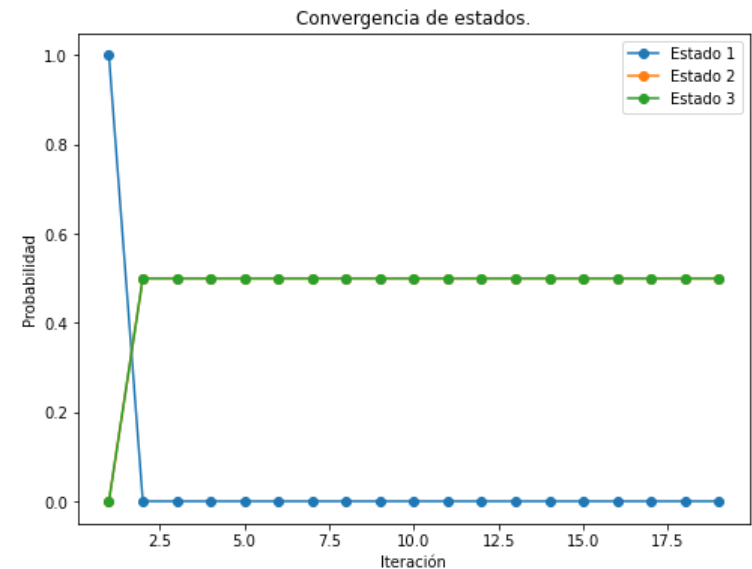
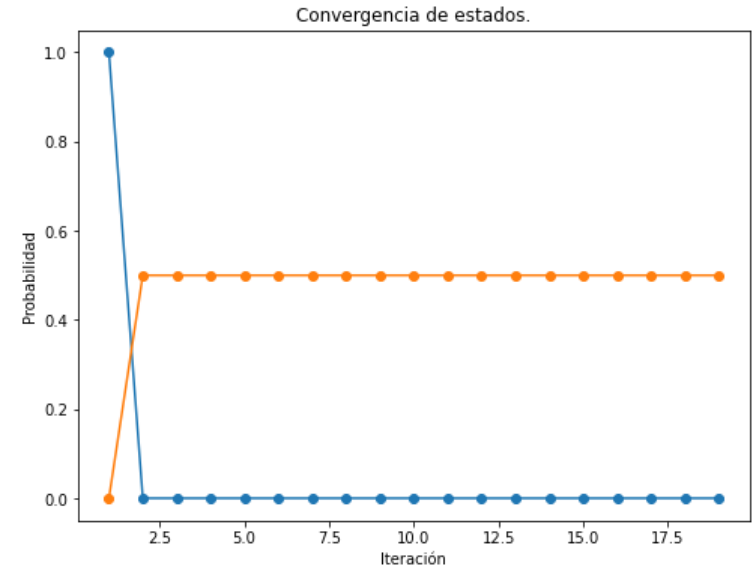
# Estado estacionario (steady-state)

## ■ Grafo sin estado estacionario:



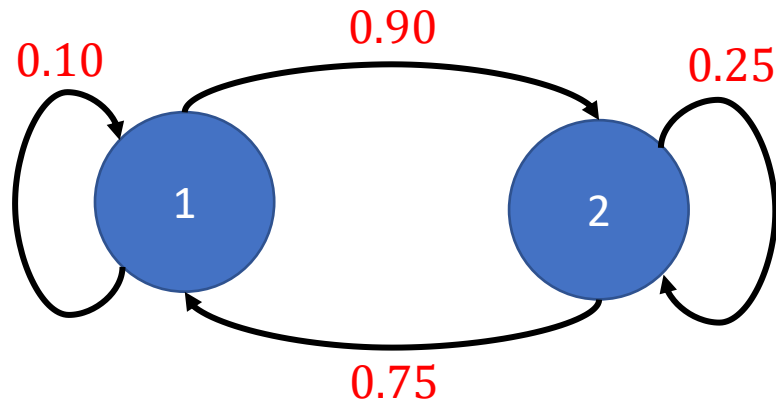
$$p_0 = [1.0, 0.0, 0.0]$$

$$p_0 = [0.0, 1.0, 0.0]$$

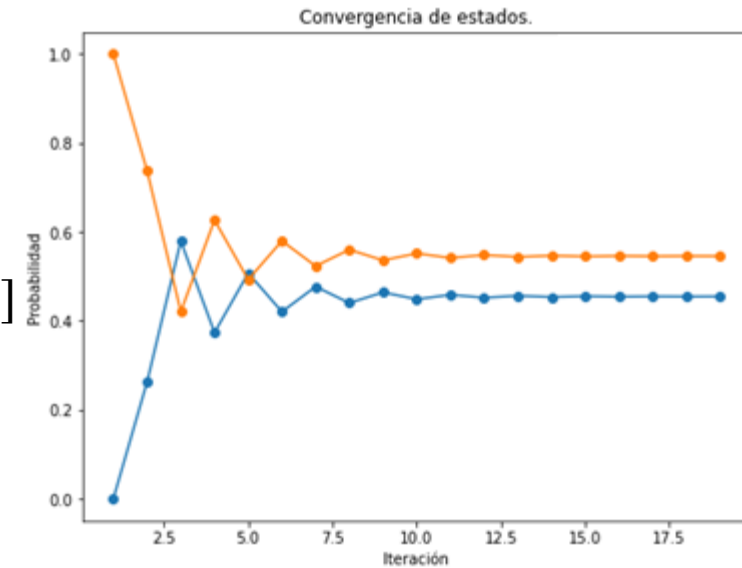


# Estado estacionario (steady-state)

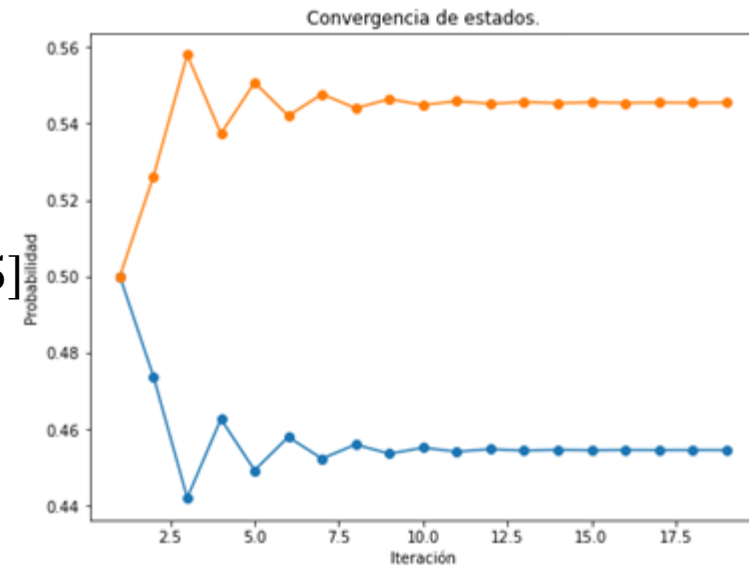
- Una cadena **Ergódica** garantiza estado estacionario.



$$p_0 = [1.0, 0.0]$$



$$p_0 = [0.5, 0.5]$$



# Matriz de Transición y Estado Estacionario

- El estado estacionario no depende del vector de estado inicial.
- Podemos identificarlo directamente sobre la matriz de transición.

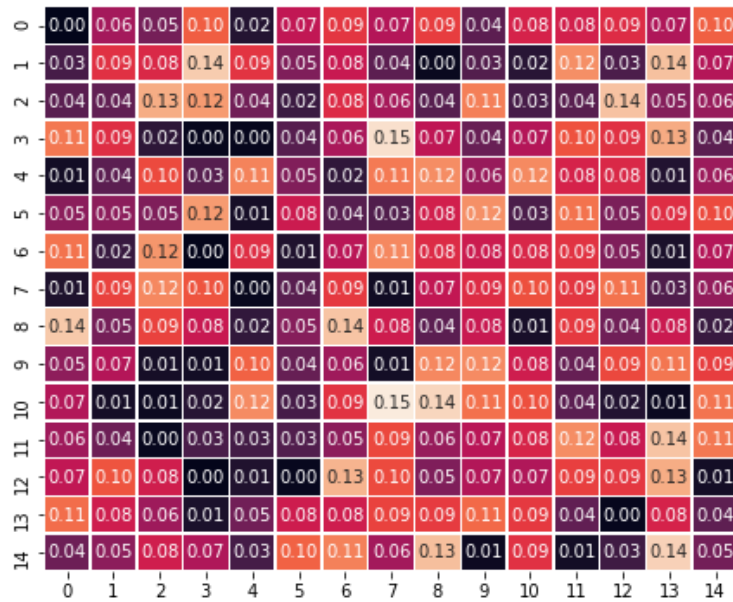
Ejemplo: matriz de transición de 15 estados:

0	0.00	0.06	0.05	0.10	0.02	0.07	0.09	0.07	0.09	0.04	0.08	0.08	0.09	0.07	0.10
1	0.03	0.09	0.08	0.14	0.09	0.05	0.08	0.04	0.00	0.03	0.02	0.12	0.03	0.14	0.07
2	0.04	0.04	0.13	0.12	0.04	0.02	0.08	0.06	0.04	0.11	0.03	0.04	0.14	0.05	0.06
3	0.11	0.09	0.02	0.00	0.00	0.04	0.06	0.15	0.07	0.04	0.07	0.10	0.09	0.13	0.04
4	0.01	0.04	0.10	0.03	0.11	0.05	0.02	0.11	0.12	0.06	0.12	0.08	0.08	0.01	0.06
5	0.05	0.05	0.05	0.12	0.01	0.08	0.04	0.03	0.08	0.12	0.03	0.11	0.05	0.09	0.10
6	0.11	0.02	0.12	0.00	0.09	0.01	0.07	0.11	0.08	0.08	0.08	0.09	0.05	0.01	0.07
7	0.01	0.09	0.12	0.10	0.00	0.04	0.09	0.01	0.07	0.09	0.10	0.09	0.11	0.03	0.06
8	0.14	0.05	0.09	0.08	0.02	0.05	0.14	0.08	0.04	0.08	0.01	0.09	0.04	0.08	0.02
9	0.05	0.07	0.01	0.01	0.10	0.04	0.06	0.01	0.12	0.12	0.08	0.04	0.09	0.11	0.09
10	0.07	0.01	0.01	0.02	0.12	0.03	0.09	0.15	0.14	0.11	0.10	0.04	0.02	0.01	0.11
11	0.06	0.04	0.00	0.03	0.03	0.03	0.05	0.09	0.06	0.07	0.08	0.12	0.08	0.14	0.11
12	0.07	0.10	0.08	0.00	0.01	0.00	0.13	0.10	0.05	0.07	0.07	0.09	0.09	0.13	0.01
13	0.11	0.08	0.06	0.01	0.05	0.08	0.08	0.09	0.09	0.11	0.09	0.04	0.00	0.08	0.04
14	0.04	0.05	0.08	0.07	0.03	0.10	0.11	0.06	0.13	0.01	0.09	0.01	0.03	0.14	0.05

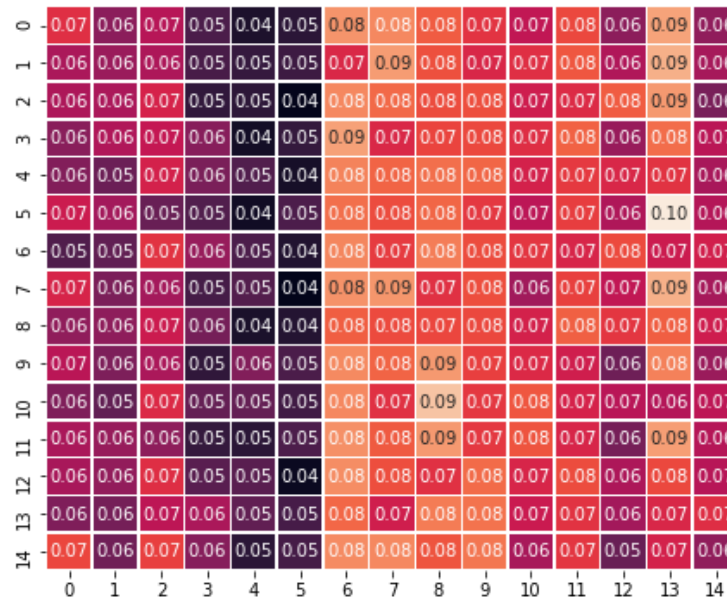
1 Paso

# Matriz de Transición y Estado Estacionario

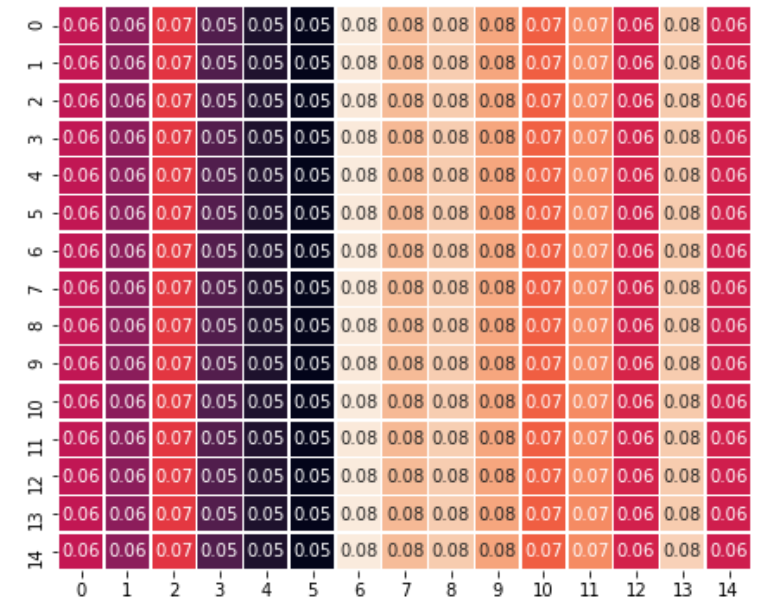
Al aumentar los pasos y acercarnos al estacionario, la probabilidad de ir de “i” a “j” se estabiliza.



1 Paso



2 Pasos



4 Pasos

# Matriz de Transición y Estado Estacionario

Por lo tanto, se cumple:

Dado el estado estacionario:

$$\pi = [\pi_0 \quad \pi_1 \quad \pi_2 \quad \dots \quad \pi_n]$$

La matriz de transición resulta:

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \pi_0 & \pi_1 & \pi_2 & \dots & \pi_n \\ \pi_0 & \pi_1 & \pi_2 & \dots & \pi_n \\ \pi_0 & \pi_1 & \pi_2 & \dots & \pi_n \\ \dots & \dots & \dots & \dots & \dots \\ \pi_0 & \pi_1 & \pi_2 & \dots & \pi_n \end{bmatrix}$$

Las filas de la matriz de transición tendiendo a infinitos pasos, muestran el estado estacionario.

# Cálculo de estado estacionario (steady-state)

Formas de calcular el estado estacionario en Cadenas Ergódicas:

- Método de fuerza bruta.
- Sistema de ecuaciones Chapman-Kolmógorov.
- Flujos probabilísticos.
- Teorema de Perron-Frobenius

# Cálculo de estado estacionario con fuerza bruta

- Evolucionamos la cadena hasta lograr el estado estable.

Repetimos mientras  $\Delta > tol$ :

$$p_{i+1} = p_i T^1$$

$$\Delta = \|p_{i+1} - p_i\|_2$$

Siguiente  $i$

Siendo:

- $p_i$ : vector de estado.
- $T_{st}$ : matriz de transición de 1 paso.
- $tol$ : tolerancia de corte.

```
# inicializamos:
tol = 10e-4
norma = 999
contador = 1

# mientras el error sea mayor a la tolerancia:
while norma > tol:

    # calculamos el estado siguiente:
    p_imas1 = np.dot(p_i, T_st)

    # calculamos la diferencia entre estados:
    dif = p_imas1 - p_i

    # calculamos la norma del error:
    norma = np.linalg.norm(dif)

    # actualizamos p_i y el contador de estados
    p_i = p_imas1

    # Aumentamos contador.
    contador += 1
```



# Cálculo de estado estacionario con fuerza bruta

## ■ Ejemplo:

```
# Matriz de transición.  
T_st = np.array(  
    [[0.10, 0.90],  
     [0.75, 0.25]]  
)  
  
# Estado inicial.  
p_i = [0.5, 0.5]
```

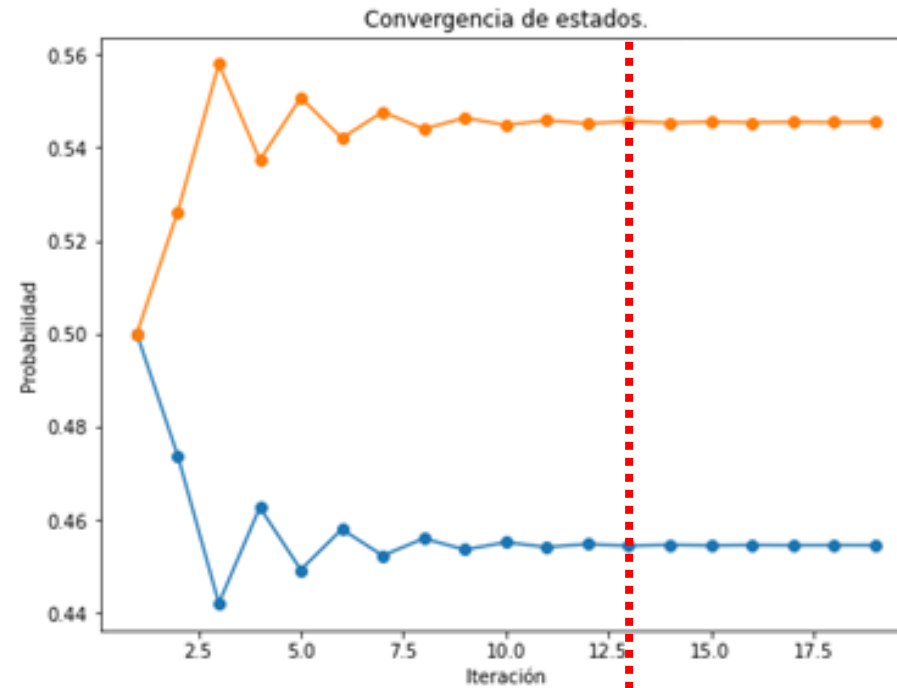
Siendo:

- $p_i$ : vector de estado.
- $T_{st}$ : matriz de transición.
- $tol$ : tolerancia de corte.

```
# inicializamos:  
tol = 10e-4  
norma = 999  
contador = 1  
  
# mientras el error sea mayor a la tolerancia:  
while norma > tol:  
  
    # calculamos el estado siguiente:  
    p_imas1 = np.dot(p_i, T_st)  
  
    # calculamos la diferencia entre estados:  
    dif = p_imas1 - p_i  
  
    # calculamos la norma del error:  
    norma = np.linalg.norm(dif)  
  
    # actualizamos p_i y el contador de estados  
    p_i = p_imas1  
  
    # Aumentamos contador.  
    contador += 1
```

# Cálculo de estado estacionario con fuerza bruta

## ■ Ejemplo:



```
>>Estado estacionario: [0.454804 0.545196]  
alcanzado en 13 iteraciones
```

# Cálculo con sistema de ecuaciones Chapman-Kolmógorov

Sabemos que tendiendo al infinito, si hay distribución estacionaria, el vector de estado se estabiliza:

$$\lim_{n \rightarrow \infty} p_{n+1} = \lim_{n \rightarrow \infty} p_n T$$

Esto implica que  $p_{n+1}$  y  $p_n$  se igualan.

$$\pi T = \pi$$

# Cálculo con sistema de ecuaciones Chapman-Kolmógorov

Condición de estado estable:

$$\pi T = \pi$$

$$(\pi_1 \quad \pi_2 \quad \pi_3 \quad \dots \quad \pi_n) \begin{bmatrix} p_{11} & p_{12} & p_{13} & \dots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \dots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \dots & p_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & p_{n3} & \dots & p_{nn} \end{bmatrix} = (\pi_1 \quad \pi_2 \quad \pi_3 \quad \dots \quad \pi_n)$$

Armamos sistema de ecuaciones:

$$\begin{aligned} p_{11} \pi_1 + p_{21} \pi_2 + p_{31} \pi_3 + \dots + p_{n1} \pi_n &= \pi_1 \\ p_{12} \pi_1 + p_{22} \pi_2 + p_{32} \pi_3 + \dots + p_{n2} \pi_n &= \pi_2 \\ p_{13} \pi_1 + p_{23} \pi_2 + p_{33} \pi_3 + \dots + p_{n3} \pi_n &= \pi_3 \\ &\dots \\ p_{1n} \pi_1 + p_{2n} \pi_2 + p_{3n} \pi_3 + \dots + p_{nn} \pi_n &= \pi_n \end{aligned}$$

# Cálculo con sistema de ecuaciones Chapman-Kolmógorov

$$\begin{aligned} p_{11} \pi_1 + p_{21} \pi_2 + p_{31} \pi_3 + \cdots + p_{n1} \pi_n &= \pi_1 \\ p_{12} \pi_1 + p_{22} \pi_2 + p_{32} \pi_3 + \cdots + p_{n2} \pi_n &= \pi_2 \\ p_{13} \pi_1 + p_{23} \pi_2 + p_{33} \pi_3 + \cdots + p_{n3} \pi_n &= \pi_3 \\ &\vdots \\ p_{1n} \pi_1 + p_{2n} \pi_2 + p_{3n} \pi_3 + \cdots + p_{nn} \pi_n &= \pi_n \end{aligned}$$

$$\begin{aligned} (p_{11}-1) \pi_1 + p_{21} \pi_2 + p_{31} \pi_3 + \cdots + p_{n1} \pi_n &= 0 \\ p_{12} \pi_1 + (p_{22}-1) \pi_2 + p_{32} \pi_3 + \cdots + p_{n2} \pi_n &= 0 \\ p_{13} \pi_1 + p_{23} \pi_2 + (p_{33}-1) \pi_3 + \cdots + p_{n3} \pi_n &= 0 \\ &\vdots \\ p_{1n} \pi_1 + p_{2n} \pi_2 + p_{3n} \pi_3 + \cdots + (p_{nn}-1) \pi_n &= 0 \end{aligned}$$

# Cálculo con sistema de ecuaciones Chapman-Kolmógorov

$$\begin{aligned}(p_{11}-1)\pi_1 + p_{21}\pi_2 + p_{31}\pi_3 + \cdots + p_{n1}\pi_n &= 0 \\ p_{12}\pi_1 + (p_{22}-1)\pi_2 + p_{32}\pi_3 + \cdots + p_{n2}\pi_n &= 0 \\ p_{13}\pi_1 + p_{23}\pi_2 + (p_{33}-1)\pi_3 + \cdots + p_{n3}\pi_n &= 0 \\ &\vdots \\ p_{1n}\pi_1 + p_{2n}\pi_2 + p_{3n}\pi_3 + \cdots + (p_{nn}-1)\pi_n &= 0\end{aligned}$$

Forma matricial:

$$\begin{bmatrix} (p_{11}-1) & p_{21} & p_{31} & \cdots & p_{1n} \\ p_{12} & (p_{22}-1) & p_{32} & \cdots & p_{2n} \\ p_{13} & p_{23} & (p_{33}-1) & \cdots & p_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{1n} & p_{n2} & p_{n3} & \cdots & (p_{nn}-1) \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \cdots \\ \pi_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$$

Sistema Homogéneo  
Det(Matriz) = 0  
-> Compatible indeterminado



# Cálculo con sistema de ecuaciones Chapman-Kolmógorov

Condición adicional:

$$\sum_i \pi_i = 1 \rightarrow \pi_1 + \pi_2 + \pi_3 + \cdots + \pi_n = 1$$

$$\begin{aligned}(p_{11}-1) \pi_1 + p_{21} \pi_2 + p_{31} \pi_3 + \cdots + p_{n1} \pi_n &= 0 \\ p_{12} \pi_1 + (p_{22}-1) \pi_2 + p_{32} \pi_3 + \cdots + p_{n2} \pi_n &= 0 \\ p_{13} \pi_1 + p_{23} \pi_2 + (p_{33}-1) \pi_3 + \cdots + p_{n3} \pi_n &= 0 \\ &\vdots \\ p_{1n} \pi_1 + p_{2n} \pi_2 + p_{3n} \pi_3 + \cdots + (p_{nn}-1) \pi_n &= 0\end{aligned}$$

# Cálculo con sistema de ecuaciones Chapman-Kolmógorov

Forma matricial con condición adicional:

$$\begin{bmatrix} (p_{11}-1) & p_{21} & p_{31} & \dots & p_{1n} \\ p_{12} & (p_{22}-1) & p_{32} & \dots & p_{2n} \\ p_{13} & p_{23} & (p_{33}-1) & \dots & p_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ p_{1n} & p_{n2} & p_{n3} & \dots & (p_{nn}-1) \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \dots \\ \pi_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix}$$

Al resolver este sistema de ecuaciones, obtenemos el vector de estado estacionario  $\pi$



# Cálculo con flujos probabilísticos

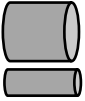
Entendemos la dinámica de la cadena como un grafo de flujo.

Podemos imaginarlo como un grafo en donde:

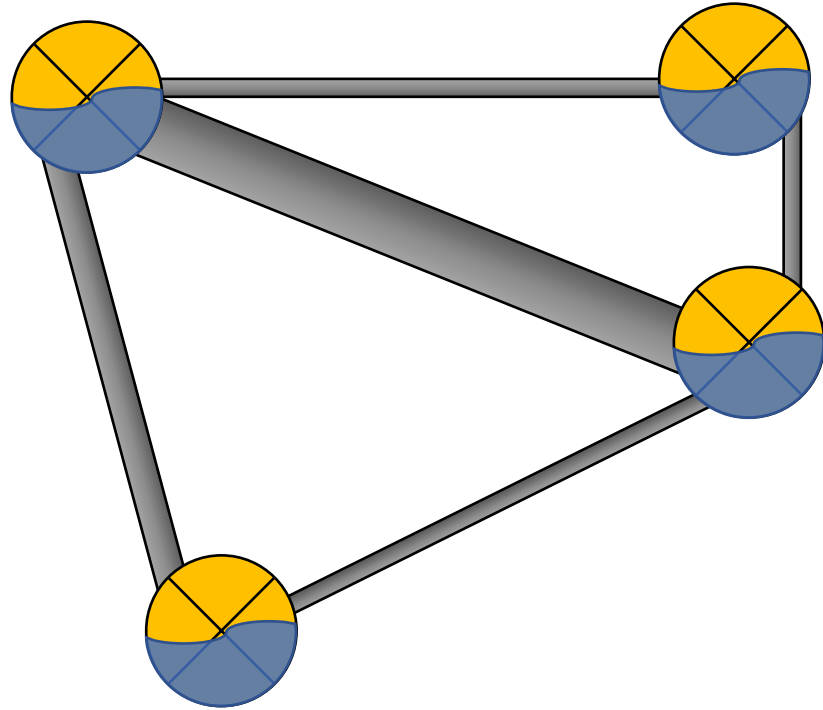
 ▪ Los nodos son medidores de fluido en un punto.

 ▪ Los arcos son tubos de transferencia de fluido.

 ▪ La probabilidad de estado es la medición de fluido que pasa por el medidor.

 ▪ La probabilidad de transición es el caudal de fluido que puede transportar un tubo de un tanque a otro, expresado en % del medidor del que parte.

# Cálculo con flujos probabilísticos

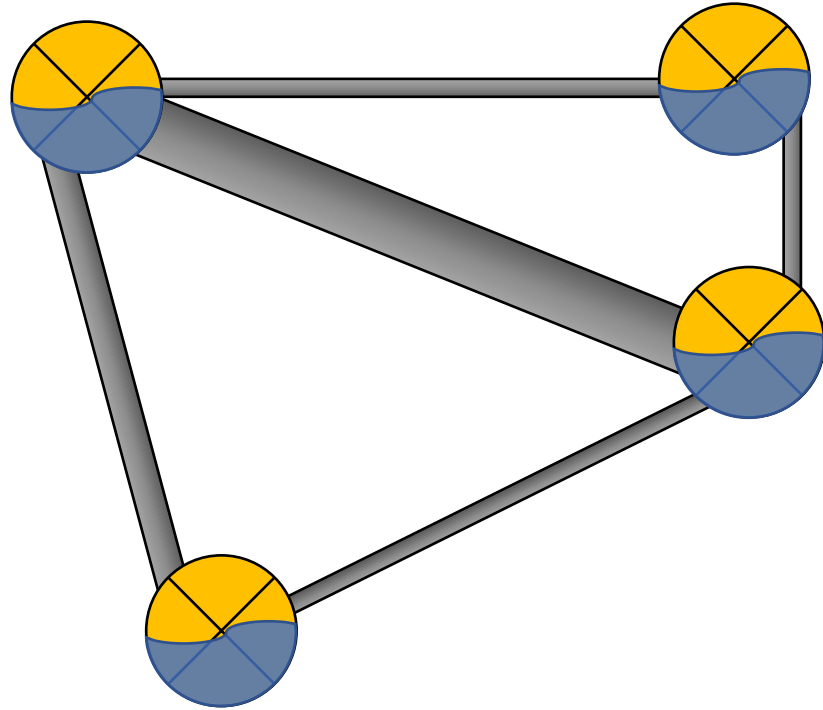


- Dado un estado inicial, el fluido se estabiliza dentro de la red.
- Esto representa el estado estacionario, las probabilidades de estado se estabilizan.

*Nota: hay que tener en cuenta que el dibujo muestra un grafo no direccionado sin ciclos. Estamos tratando de resolver grafos direccionados con ciclos.*

*Solo sirve para ganar intuición sobre el problema.*

# Cálculo con flujos probabilísticos



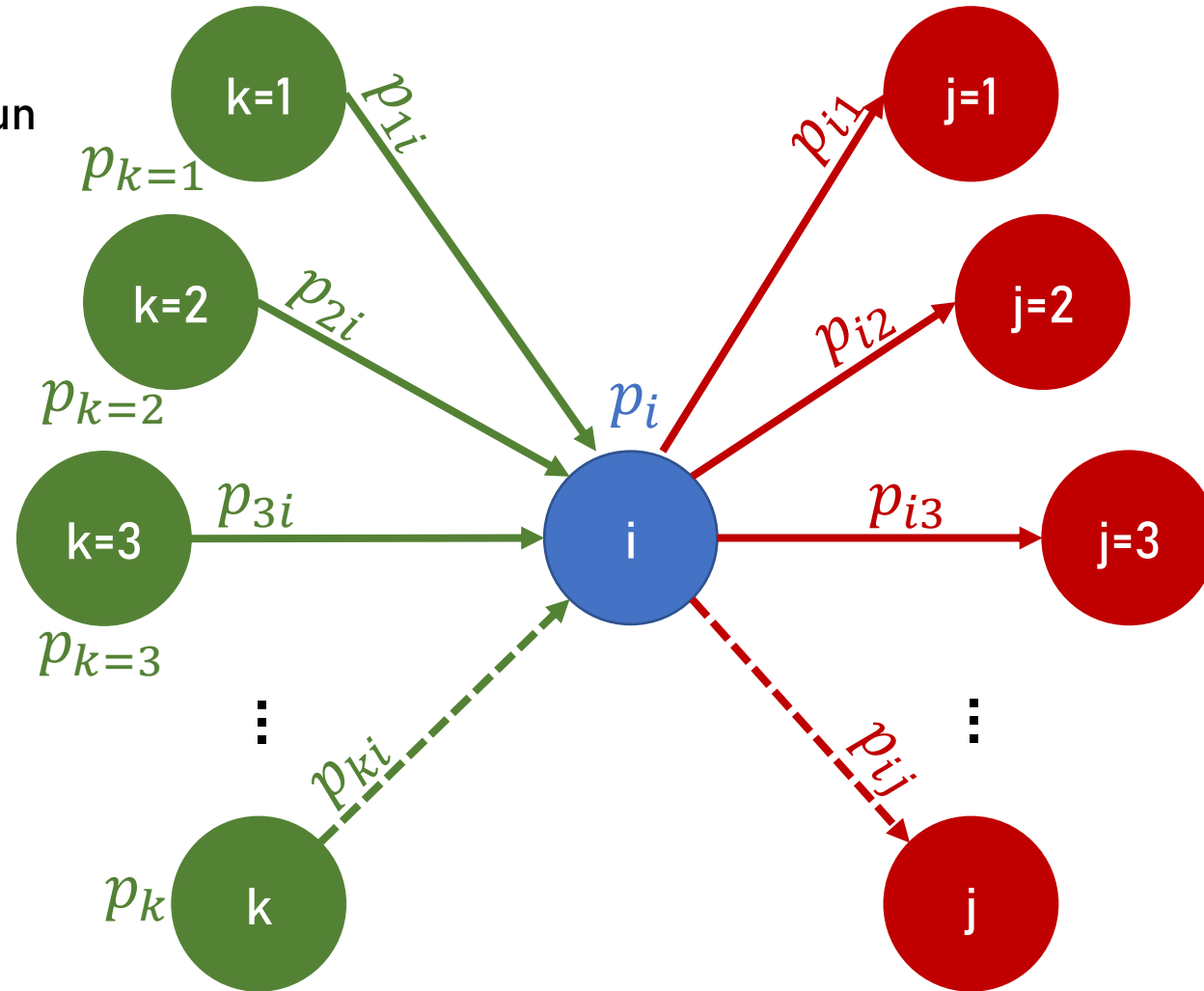
- Importante: el fluido nunca queda en los medidores, **siempre los abandona en el siguiente instante de tiempo.**

“Todo lo que entra de un medidor sale”

Objeto de análisis: **medidor**

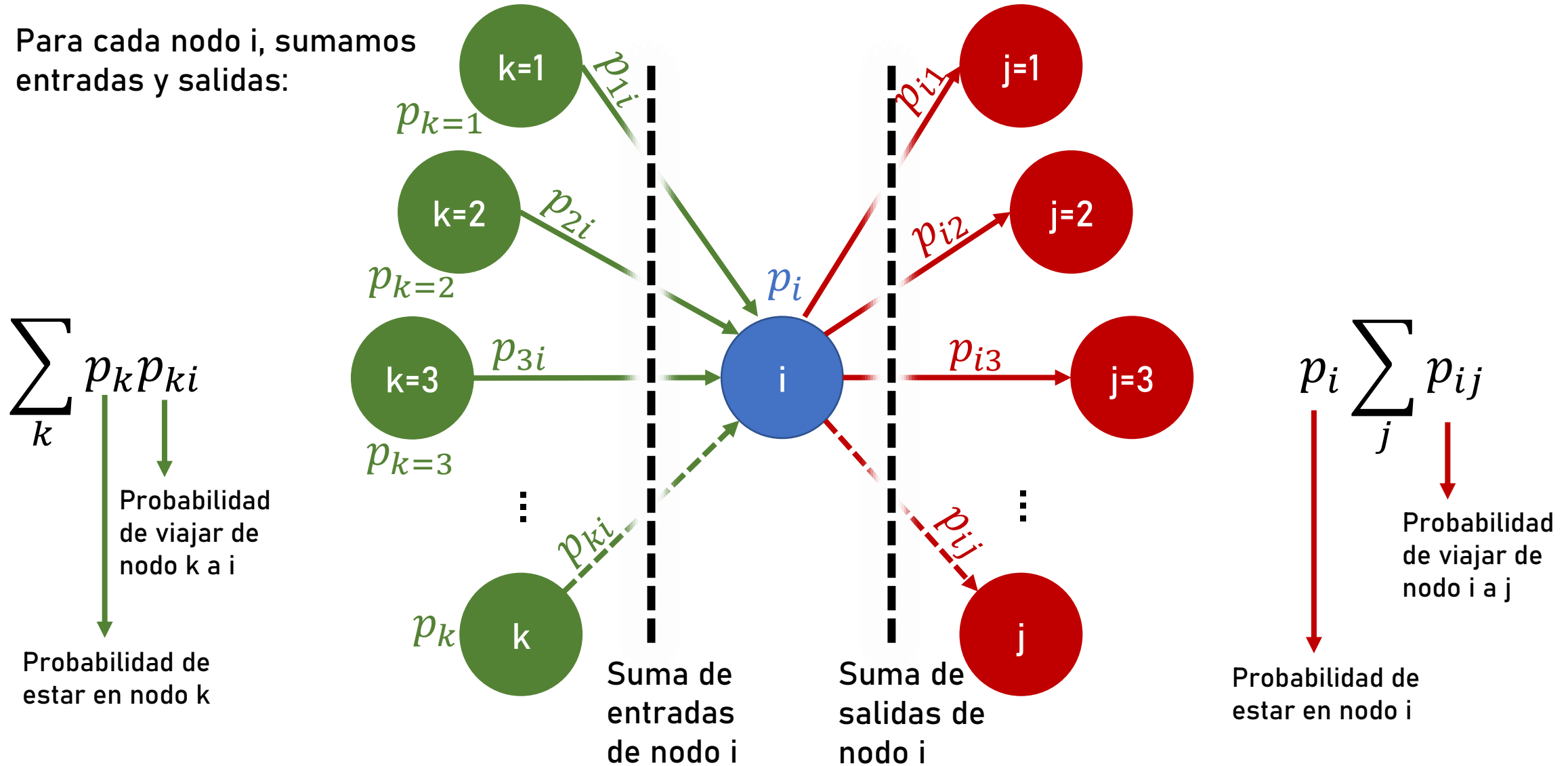
# Cálculo con flujos probabilísticos

- Centramos el análisis en cada nodo  $i$ , como un balance de flujo:



# Cálculo con flujos probabilísticos

- Para cada nodo  $i$ , sumamos entradas y salidas:



# Cálculo con flujos probabilísticos

Dado que es un balance de masa, todo lo que entra a un nodo “i” sale del mismo nodo hacia algún lado.

*suma de entradas* = *suma de salidas*

$$\sum_k p_k p_{ki} = p_i \sum_j p_{ij} \quad \forall i$$

Completamos con la condición de probabilidad de estado:

$$\sum_i p_i = 1$$

# Cálculo con flujos probabilísticos

La resolución de este sistema de ecuaciones lleva al estacionario.

*suma de entradas = suma de salidas*

$$\begin{aligned} \cancel{p_1 p_{11}} + p_2 p_{21} + p_3 p_{31} + \cdots + p_n p_{n1} &= \cancel{p_1 p_{11}} + p_1 p_{12} + p_1 p_{13} + \cdots + p_1 p_{1n} \\ p_1 p_{12} + \cancel{p_2 p_{22}} + p_3 p_{32} + \cdots + p_n p_{n2} &= p_2 p_{21} + \cancel{p_2 p_{22}} + p_2 p_{23} + \cdots + p_2 p_{2n} \\ p_1 p_{13} + p_2 p_{23} + \cancel{p_3 p_{33}} + \cdots + p_n p_{n3} &= p_3 p_{31} + p_3 p_{32} + \cancel{p_3 p_{33}} + \cdots + p_3 p_{3n} \\ &\vdots \\ p_1 p_{1n} + p_2 p_{2n} + p_3 p_{3n} + \cdots + \cancel{p_n p_{nn}} &= p_n p_{n1} + p_n p_{n2} + p_n p_{n3} + \cdots + \cancel{p_n p_{nn}} \end{aligned}$$

$$p_1 + p_2 + p_3 + \cdots + p_n = 1$$

# Cálculo con flujos probabilísticos

La resolución de este sistema de ecuaciones lleva al estacionario.

$$p_2 p_{21} + p_3 p_{31} + \cdots + p_n p_{n1} = p_1 \boxed{(p_{12} + p_{13} + \cdots + p_{1n})} = (1 - p_{11})$$

$$p_1 p_{12} + p_3 p_{32} + \cdots + p_n p_{n2} = p_2 \boxed{(p_{21} + p_{23} + \cdots + p_{2n})} = (1 - p_{22})$$

$$p_1 p_{13} + p_2 p_{23} + \cdots + p_n p_{n3} = p_3 \boxed{(p_{31} + p_{32} + \cdots + p_{3n})} = (1 - p_{33})$$

...

$$p_1 p_{1n} + p_2 p_{2n} + p_3 p_{3n} + \cdots = p_n \boxed{(p_{n1} + p_{n2} + p_{n3} + \cdots)} = (1 - p_{nn})$$

$$p_1 + p_2 + p_3 + \cdots + p_n = 1$$



# Cálculo con flujos probabilísticos

## ¡Sistema de ecuaciones con Chapman-Kolmógorov!

$$p_1(p_{11} - 1) + p_2p_{21} + p_3p_{31} + \cdots + p_np_{n1} = 0$$

$$p_1p_{12} + p_2(p_{22} - 1) + p_3p_{32} + \cdots + p_np_{n2} = 0$$

$$p_1p_{13} + p_2p_{23} + p_3(p_{33} - 1) + \cdots + p_np_{n3} = 0$$

...

$$p_1p_{1n} + p_2p_{2n} + p_3p_{3n} + p_n(p_{nn} - 1) = 0$$

$$p_1 + p_2 + p_3 + \cdots + p_n = 1$$

¿Por qué sería útil esta variante si llevan a lo mismo?

Conceptualmente el problema de flujo se puede resolver con optimización dinámica o algoritmos específicos de optimización de flujo.

# Cálculo con flujos probabilísticos

¿Por qué sería útil esta variante si llevan a lo mismo?

Conceptualmente el problema de flujo se puede resolver con optimización dinámica o algoritmos específicos de optimización de flujo.

# Cálculo con teorema de Perron-Frobenius

Partiendo de Chapman-Kolmogórov:

$$p_t T = p_{t+1}$$

$p_{n+1}$  y  $p_n$  son vectores fila. Si trasponemos  $T$  podemos convertirlos en vectores columna:

$$T^T p_t = p_{t+1}$$

$$\begin{bmatrix} p_{11} & p_{21} & p_{31} & \dots & p_{1n} \\ p_{12} & p_{22} & p_{32} & \dots & p_{2n} \\ p_{13} & p_{23} & p_{33} & \dots & p_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ p_{1n} & p_{n2} & p_{n3} & \dots & p_{nn} \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ \dots \\ p_n(t) \end{bmatrix} = \begin{bmatrix} p_1(t+1) \\ p_2(t+1) \\ p_3(t+1) \\ \dots \\ p_n(t+1) \end{bmatrix}$$

# Cálculo con teorema de Perron-Frobenius

Los **vectores de estado "p"**:

- Cambian su dirección y norma al iterar con la matriz de transición.
- Excepto en el estacionario.

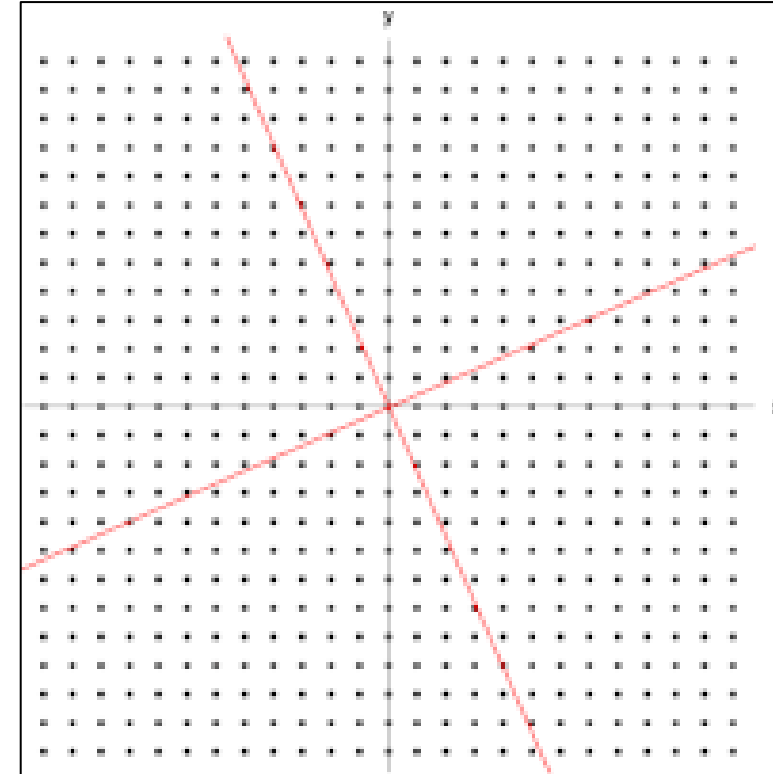
Esta propiedad se puede representar con la ecuación de **autovalores/autovectores**:

$$A\vec{v} = \lambda\vec{v}$$

$\vec{v}$ : autovector

$\lambda$ : autovalor

A: matriz nxn



<https://twitter.com/i/status/1354822527487340550>

# Cálculo con teorema de Perron-Frobenius

## Teorema de Perron-Frobenius:

- Dada una matriz no negativa e irreducible, existe el mayor de todos los autovalores, dominante y no negativo.
- El autovalor dominante corresponde a un autovector denominado: **vector de Perron-Frobenius**.
- Este vector representa la **distribución estacionaria**.
- En cadenas de **Markov Ergódicas** el autovalor dominante es **1**.

# Cálculo con teorema de Perron-Frobenius

Una matriz ergódica converge a un único estado estacionario.

Al calcular los autovalores de la matriz  $T^T$ :

$$\text{eigen}(T^T) = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \dots \\ \lambda_n \end{bmatrix}$$

**Siempre existe** un autovalor  $\lambda_i = 1$ , su **autovector**  $\vec{v}_i$  **escalado** representa el **estado estacionario**.

# Cálculo con teorema de Perron-Frobenius

Ejemplo simple anterior:

$$T = \begin{bmatrix} 0.10 & 0.90 \\ 0.75 & 0.25 \end{bmatrix}$$

1) Trasponemos la matriz de transición:

$$T^T = \begin{bmatrix} 0.10 & 0.75 \\ 0.90 & 0.25 \end{bmatrix}$$

# Cálculo con teorema de Perron-Frobenius

2) Calculamos los autovalores:

$$\det(A - \lambda I) = 0$$

Siendo  $A = T^T$

Calculamos el argumento:

$$A - \lambda I = \begin{bmatrix} 0.10 & 0.75 \\ 0.90 & 0.25 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 0.10 - \lambda & 0.75 \\ 0.90 & 0.25 - \lambda \end{bmatrix}$$

Resolvemos el determinante:

$$\det \left( \begin{bmatrix} 0.10 - \lambda & 0.75 \\ 0.90 & 0.25 - \lambda \end{bmatrix} \right) = (0.10 - \lambda)(0.25 - \lambda) - (0.75 * 0.90)$$



# Cálculo con teorema de Perron-Frobenius

2) Calculamos los autovalores:

$$\det \begin{pmatrix} 0.10 - \lambda & 0.75 \\ 0.90 & 0.25 - \lambda \end{pmatrix} = (0.10 - \lambda)(0.25 - \lambda) - (0.75 * 0.90) = 0$$

$$\lambda^2 - 0.35\lambda - 0.65 = 0$$

Resolvemos:

$$\lambda_1 = 1$$

$$\lambda_2 = -0.65$$

# Cálculo con teorema de Perron-Frobenius

2) Calculamos el autovector para  $\lambda_1 = 1$ :

$$(A - \lambda_1 I) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Siendo  $A = T^T$

$$\begin{bmatrix} 0.10 - \lambda_1 & 0.75 \\ 0.90 & 0.25 - \lambda_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -0.90 & 0.75 \\ 0.90 & -0.75 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

# Cálculo con teorema de Perron-Frobenius

2) Calculamos el autovector para  $\lambda_1 = 1$ :

$$\begin{bmatrix} -0.90 & 0.75 \\ 0.90 & -0.75 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{cases} -0.90x + 0.75y = 0 \\ 0.90x - 0.75y = 0 \end{cases}$$

Resolvemos el sistema indeterminado:

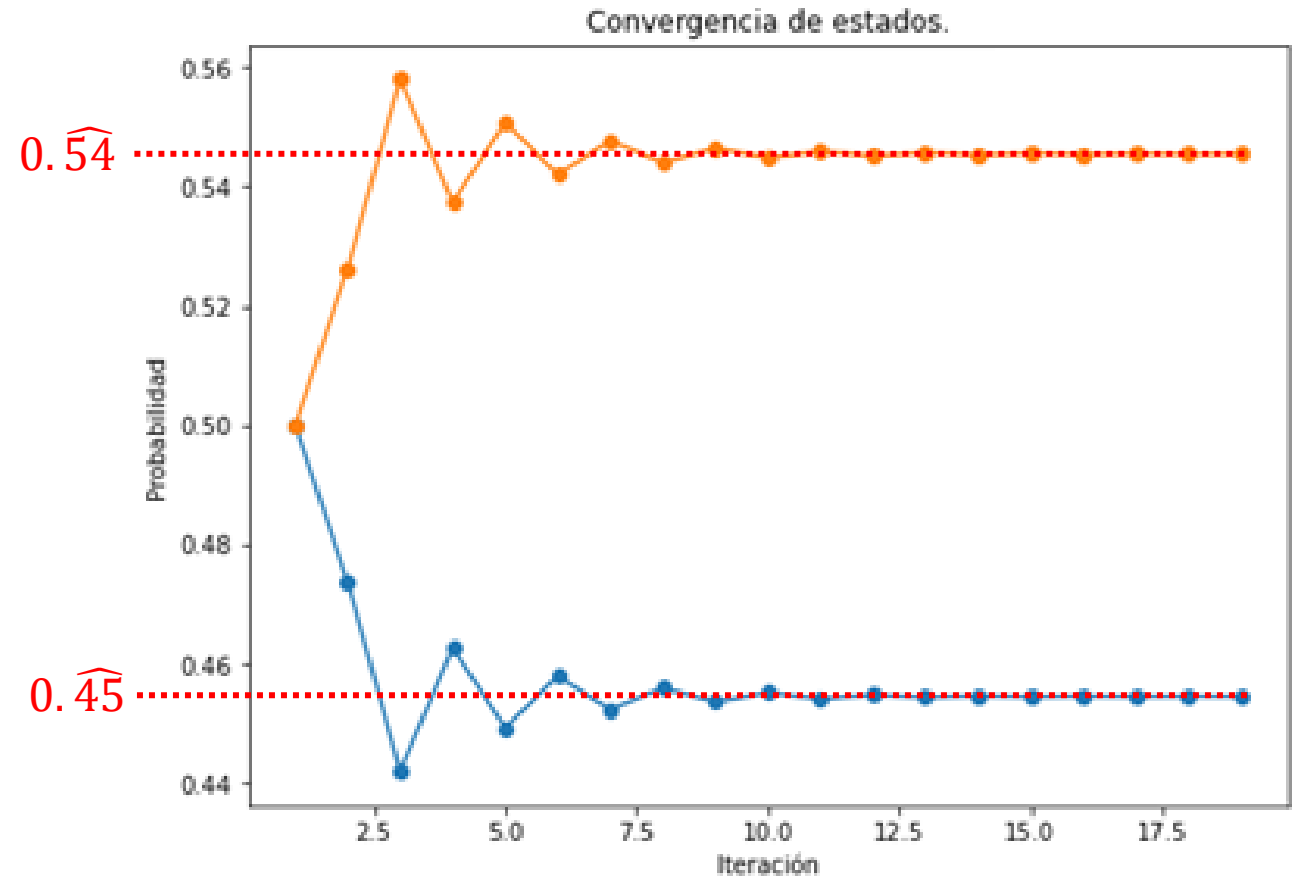
$$\vec{v}_1 = \begin{bmatrix} 0.8333 \\ 1 \end{bmatrix} u, \text{ si } u \in \mathbb{R}$$

# Cálculo con teorema de Perron-Frobenius

3) Escalamos  $\vec{v}_1 = \begin{bmatrix} 0.8333 \\ 1 \end{bmatrix}$

$$\pi = \frac{\vec{v}_1}{\|\vec{v}_1\|_1} = \begin{bmatrix} 0.8\hat{3}/1.8\hat{3} \\ 1/1.8\hat{3} \end{bmatrix}$$

$$\pi = [0.\hat{4}5, 0.\hat{5}4]$$



# Cálculo con teorema de Perron-Frobenius

## Cálculo con Python:

```
T = np.array(
    [[0.10, 0.90],
     [0.75, 0.25]]
)

# Traspuesta.
TT = T.T

# Búsqueda de autovalores (w) y
autovectores (v)
w, v = np.linalg.eig(TT)
```

```
# Imprimimos autovalores.
print(w)

>> array([-0.65, 1.  ])

# Imprimimos autovectores.
print(v)

>> array([[ -0.70710678, -0.6401844 ],
          [  0.70710678, -0.76822128]])
```

Autovector  
correspondiente  
a autovalor 1

# Cálculo con teorema de Perron-Frobenius

## Cálculo con Python:

```
# El autovector de interés está en la  
columna de los autovectores  
correspondiente al autovalor 1.
```

```
v_steady = v[:, 1]
```

```
print(v_steady)
```

```
>> array([-0.6401844 , -0.76822128])
```

```
# Normalizamos
```

```
steady_state = v_steady / np.sum(v_steady)
```

```
print(f'Estado estacionario: {steady_state}')
```

```
>> Estado estacionario: [0.45454545 0.54545455]
```

# Spectral Gap

La velocidad de convergencia hacia el estacionario se puede medir con autovalores.

**Spectral gap:** mide la separación entre los dos mayores autovalores de la matriz de transición.

$$sg = 1 - |\lambda_2|$$

Siendo  $\lambda_2$ , el segundo autovalor más grande.

**sg** ↑: convergencia rápida.

**sg** ↓: convergencia lenta.

# Spectral Gap

Ejemplo:

Cadena A:

$$\lambda_1 = 1$$

$$\lambda_2 = -0.65$$

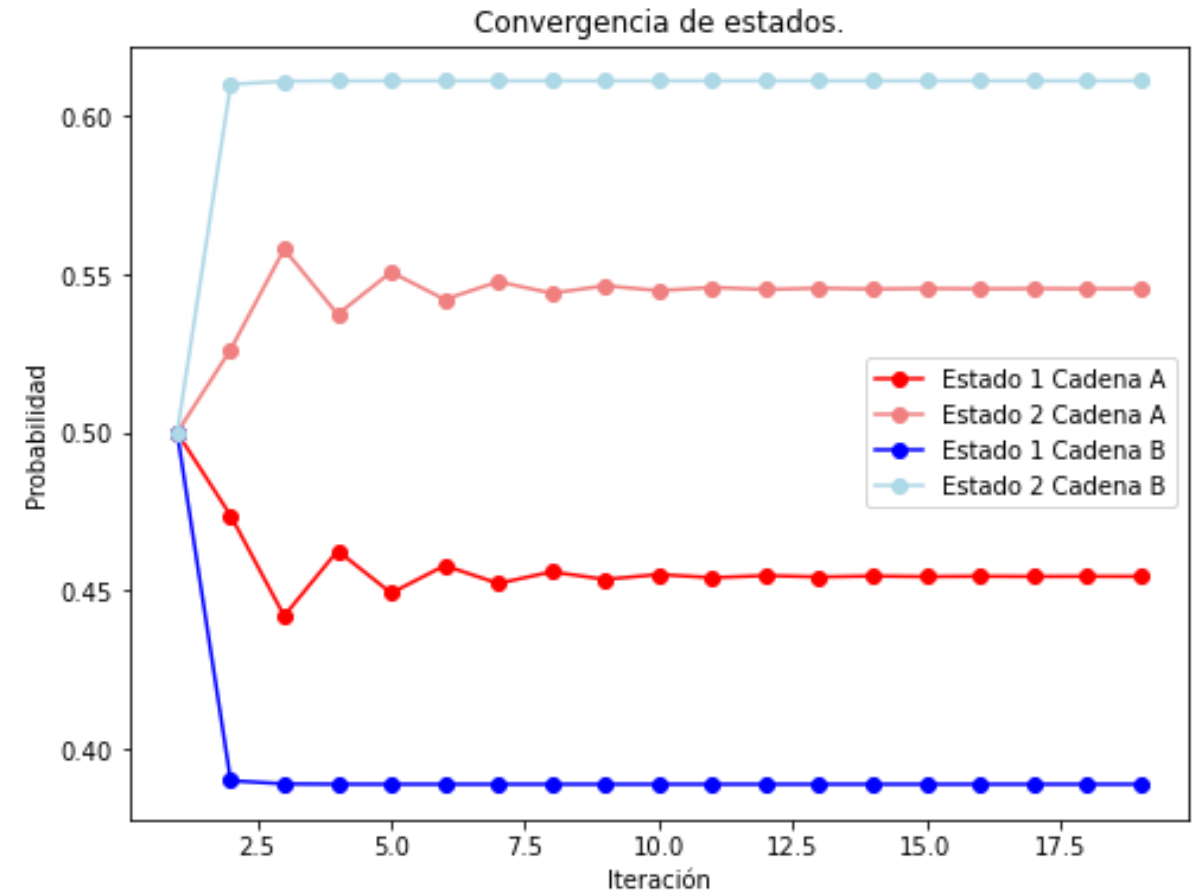
$$sg = 1 - |\lambda_2| = 1 - |-0.65| = \mathbf{0.35}$$

Cadena B:

$$\lambda_1 = 1$$

$$\lambda_2 = 0.1$$

$$sg = 1 - |\lambda_2| = 1 - |0.1| = \mathbf{0.90}$$





# Caso google PageRank

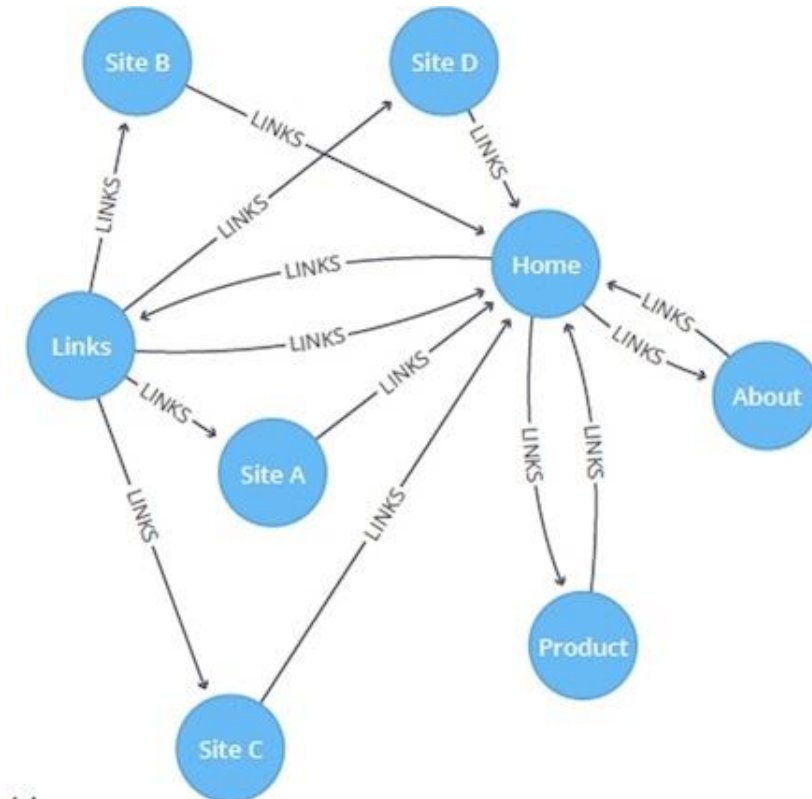
**Algoritmo de ranking** de páginas del buscador.

Existe un grafo que explica el viaje de los usuarios por páginas web.

Este grafo suele ser una **matriz no ergódica**.

El algoritmo la convierte en ergódica y calcula el estacionario para asignar el score.

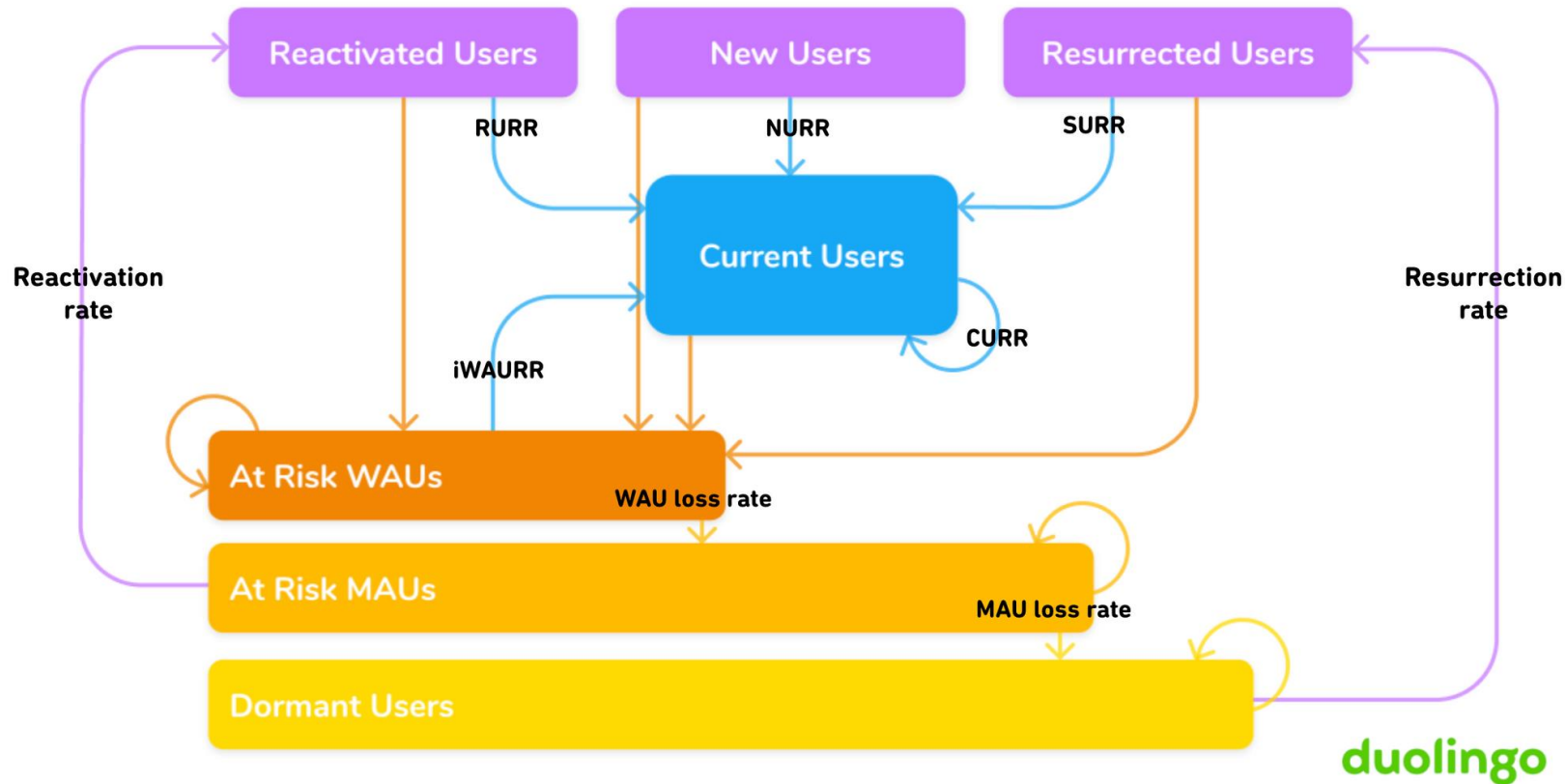
Brin and Page (1998), "The Anatomy of a Large-Scale Hypertextual Web Search Engine" <https://research.google/pubs/pub334/>



Graph Model

<https://neo4j.com/blog/graph-algorithms-neo4j-pagerank>

# Caso Duolingo: retención de usuarios con Markov



Fuente: Duolingo Blog (2023) "Meaningful metrics: How data sharpened the focus of product teams"  
" <https://blog.duolingo.com/growth-model-duolingo/>