

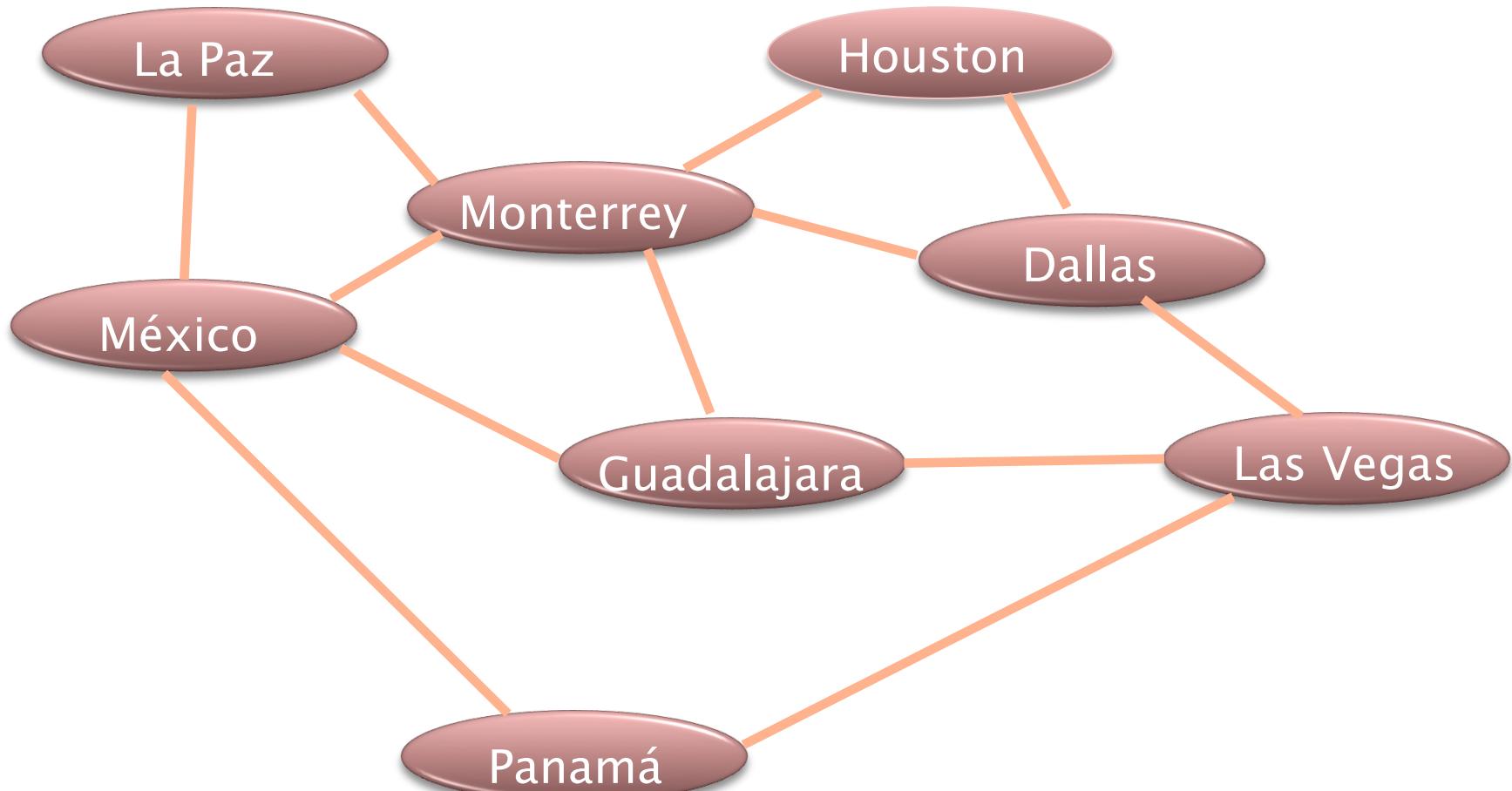
Grafos

Ing. Armandina Leal Flores

Grafos

- ▶ Es una estructura de tipo RED.
- ▶ Un grafo mantiene una relación de “muchos a muchos” (N:M) entre sus elementos.
- ▶ Qué se puede representar en un grafo:
 - La red de destinos de una aerolínea.
 - La red de carreteras entre las diferentes ciudades de un país.

Ejemplo:



Terminología de Grafos...

▶ Grafo

- Conjunto de Nodos y Arcos.

▶ Nodo

- Elemento básico de información en un grafo.

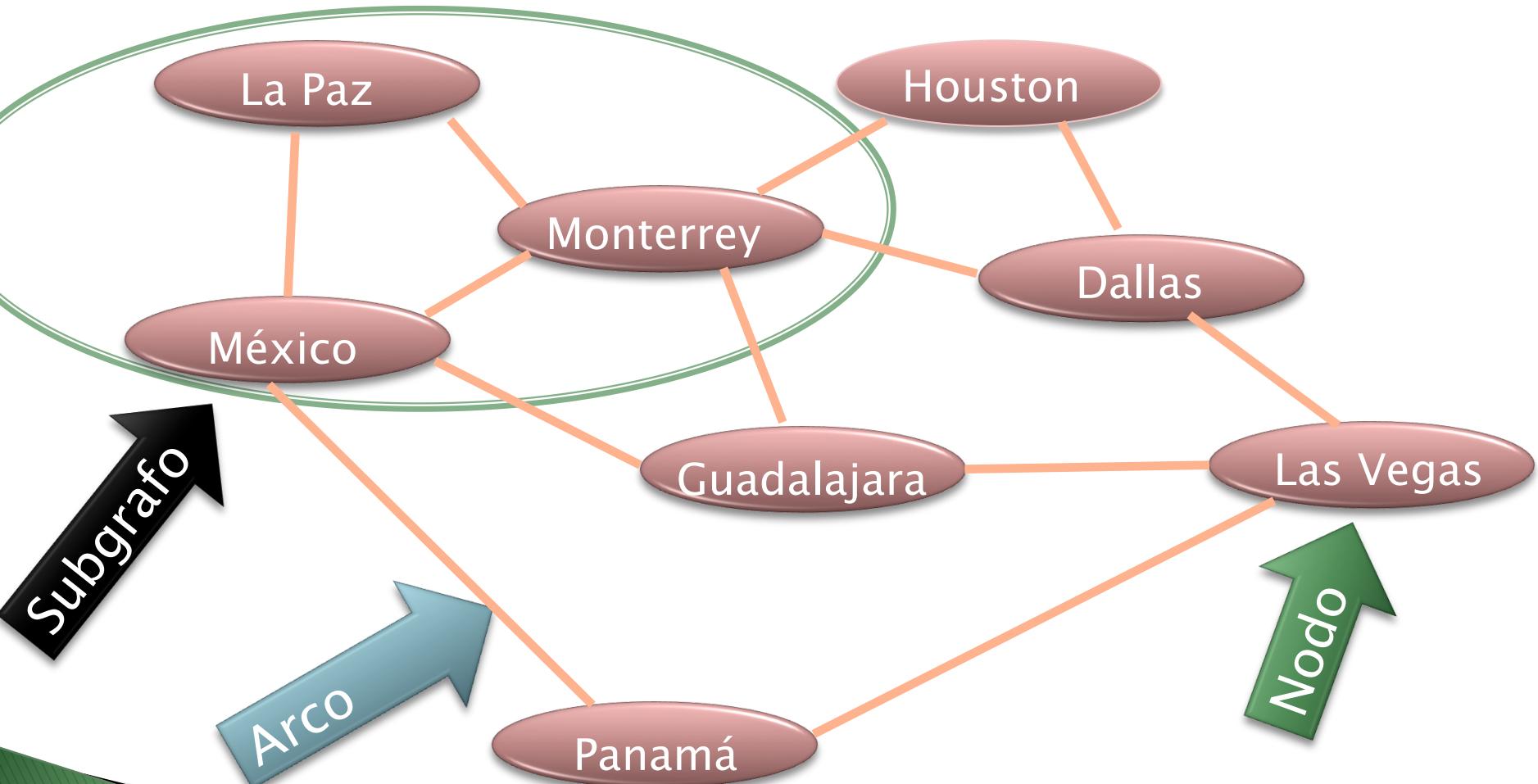
▶ Arco

- Liga que une dos nodos de un grafo.

▶ Subgrafo

- Es un grafo que contiene a un subconjunto de Nodos y Arcos.

Terminología...



Terminología de Grafos...

▶ Nodos Adyacentes

- Nodos que tienen un arco que los conecta.

▶ Vecinos de un Nodo

- Todos los nodos que son adyacentes al Nodo.

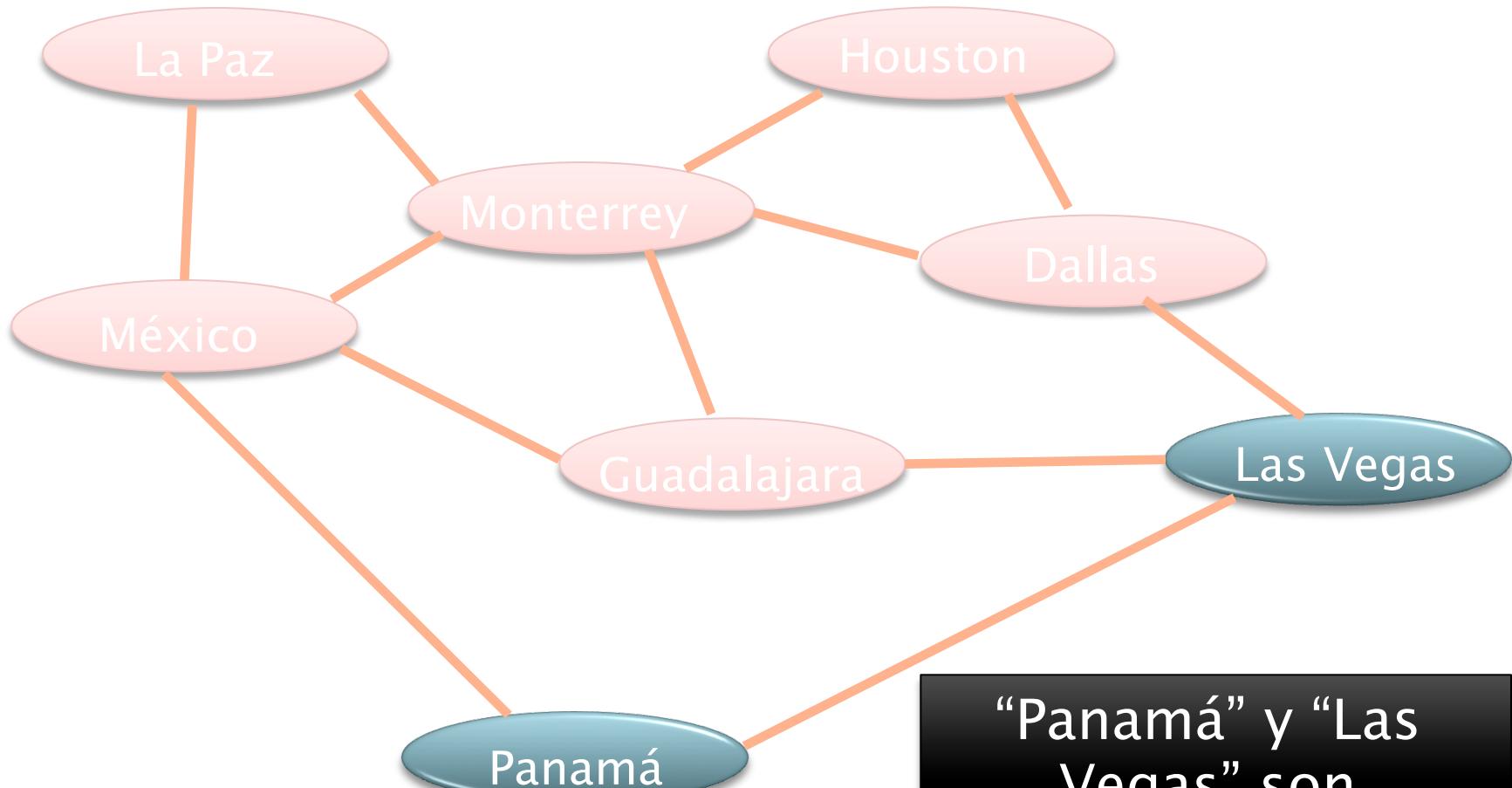
▶ Camino o Trayectoria (PATH)

- Secuencia de Nodos, de tal forma que cada par de nodos son adyacentes.

▶ Trayectoria Simple

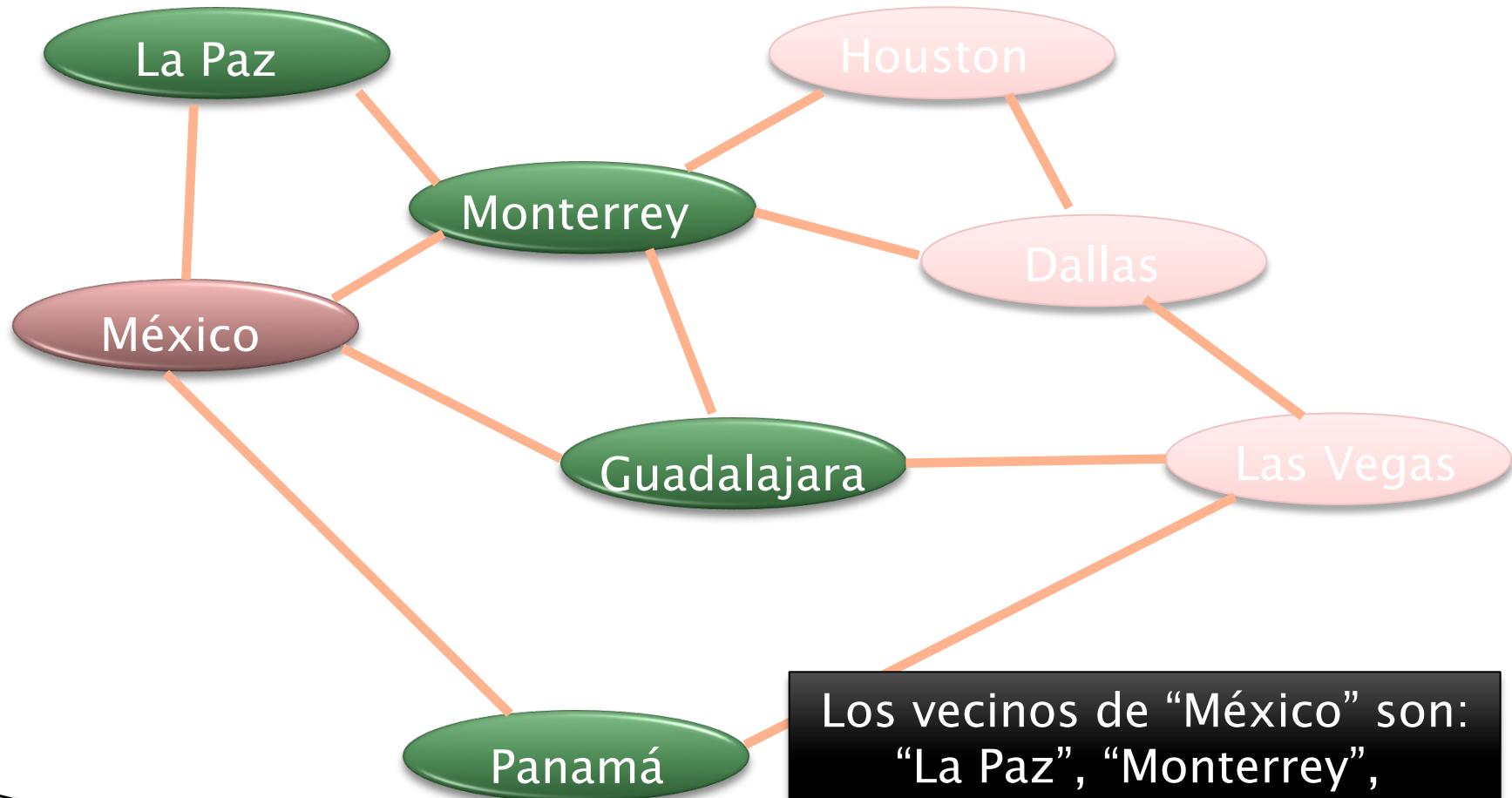
- Camino donde todos los nodos contenidos son distintos.

Terminología...

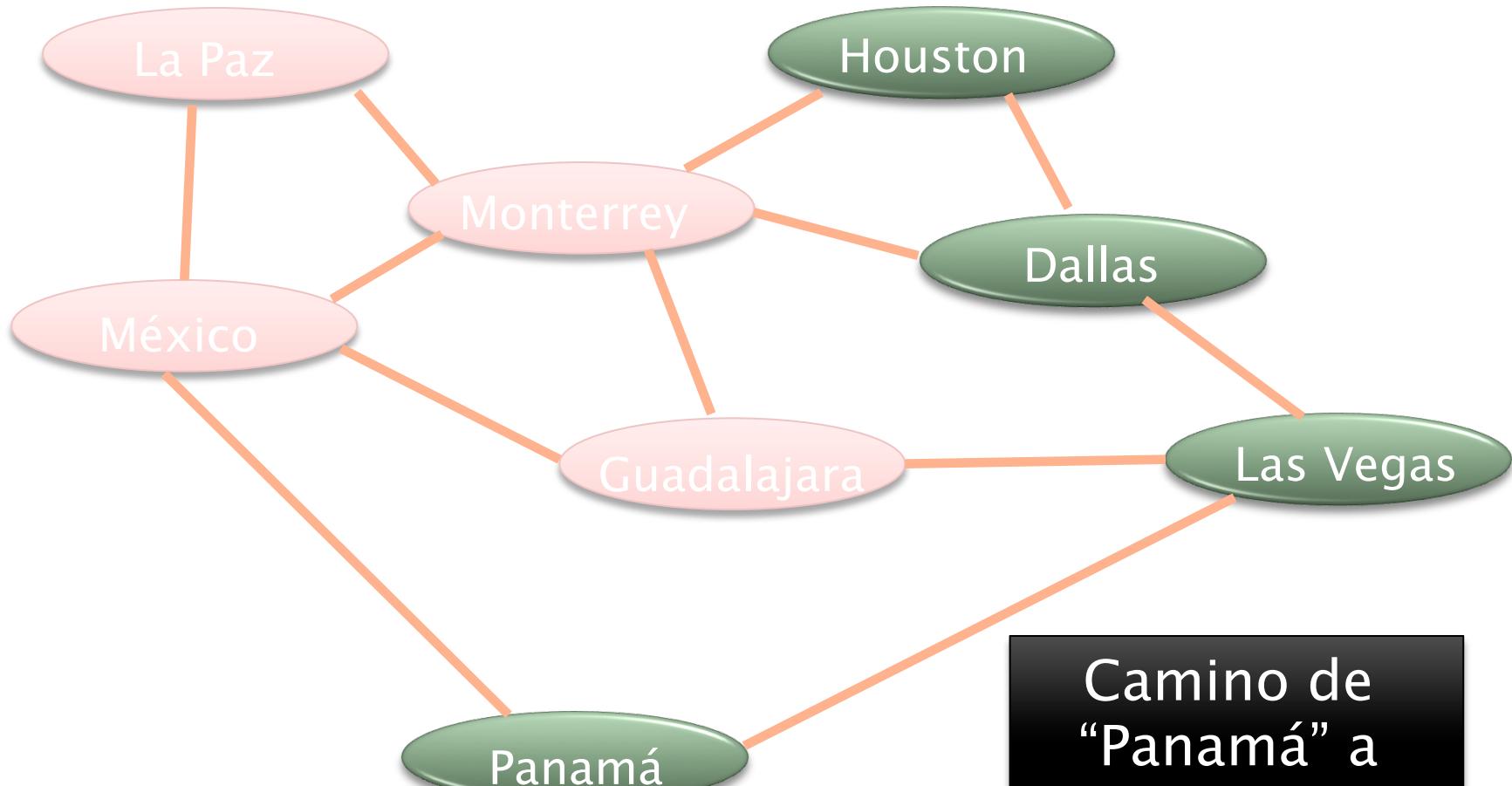


“Panamá” y “Las Vegas” son Nodos Adyacentes

Terminología...



Terminología...



Camino de
“Panamá” a
“Houston”

Terminología de Grafos...

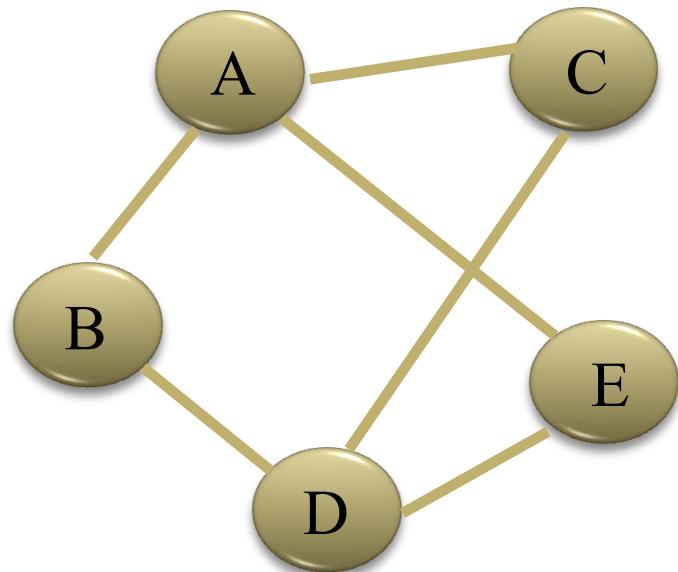
► Grafo No-Dirigido

- Los arcos en el grafo son bidireccionales. Un arco de A a B es igual que uno de B a A.

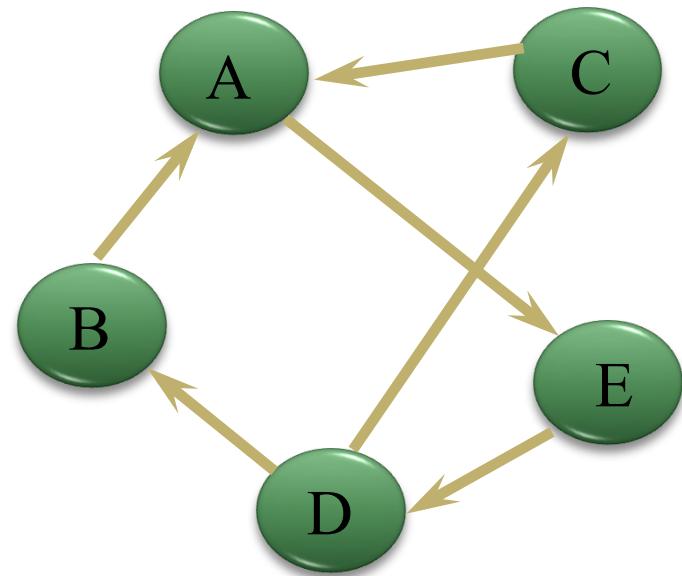
► Grafo Dirigido

- Los arcos tienen dirección. El primer elemento del arco es denominado el ORIGEN y el segundo el DESTINO. Una arco de A a B es diferente de un arco de B a A.

Terminología...



Grafo No-Dirigido



Grafo Dirigido (Digrafo)

Terminología de Grafos...

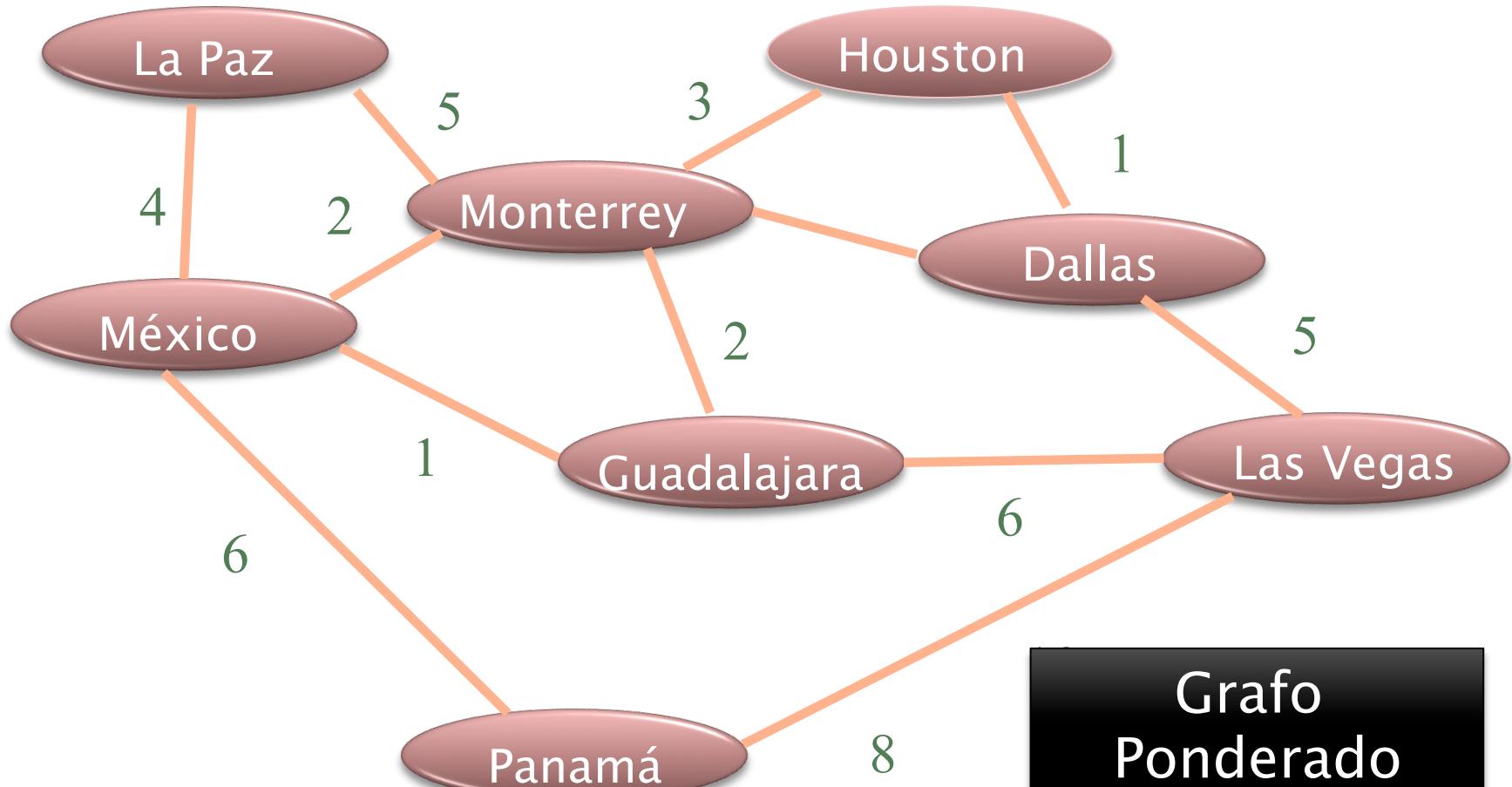
► Grafo Ponderado

- Los arcos en el grafo tienen un valor relacionado con costo, distancia,...

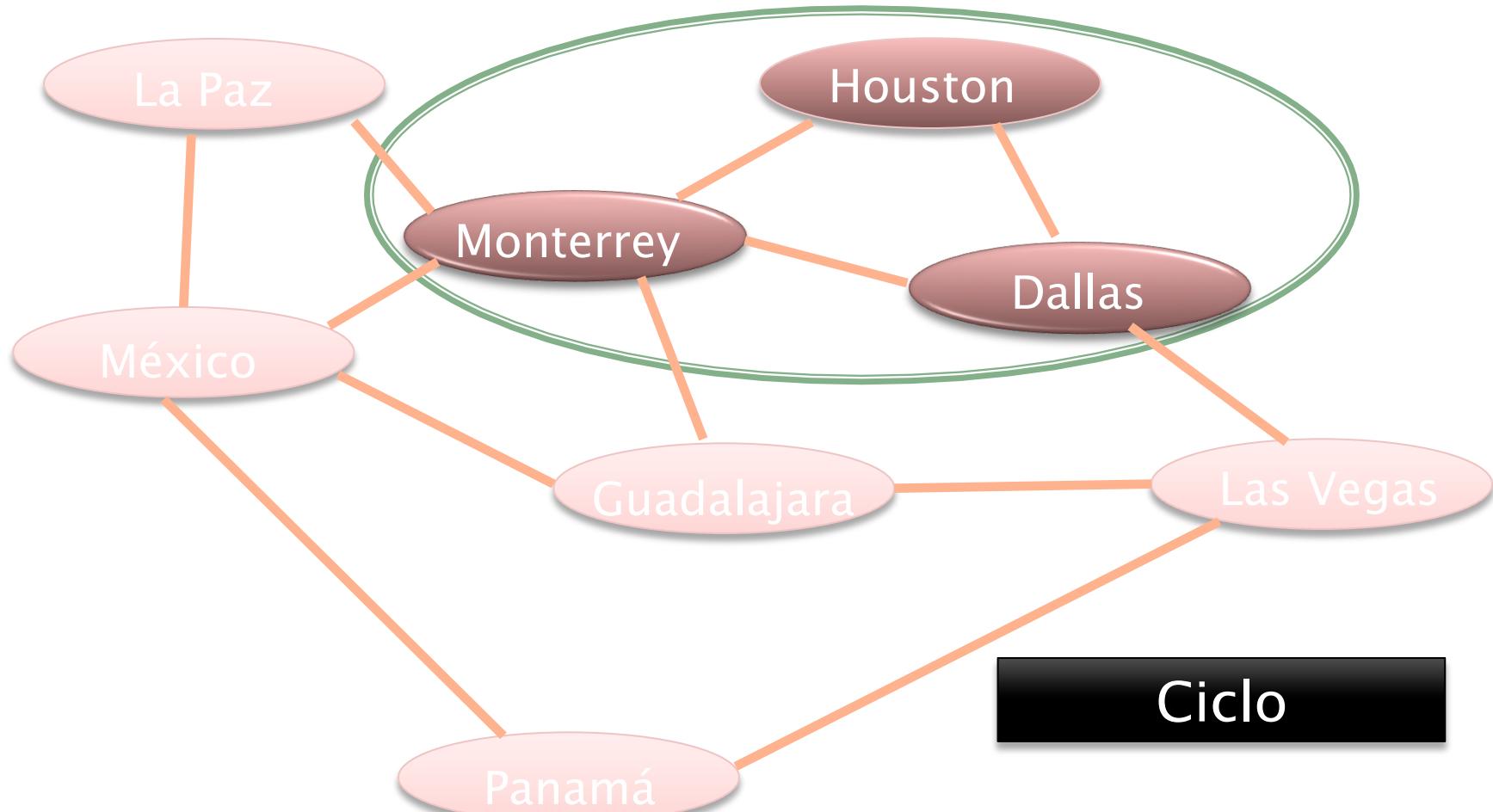
► Ciclo

- Es una trayectoria en la que el Nodo de Inicio y el de Terminación son iguales.

Terminología...



Terminología...



Ciclo

Operaciones sobre un Grafo

► Operación básicas

- Insertar un Nodo
- Insertar un Arco
- Borrar un Nodo
- Borrar un Arco
- Buscar un Nodo

► Otras Operaciones

- Recorrer el Grafo

Aplicaciones de un Grafo

► Conectividad, Redes de Transporte

- ¿Existe un camino entre dos Nodos?
- ¿Cuál es el costo mínimo de conexión para todos los Nodos?
- ¿Cuál es la ruta óptima para ir de un Nodo a otro?

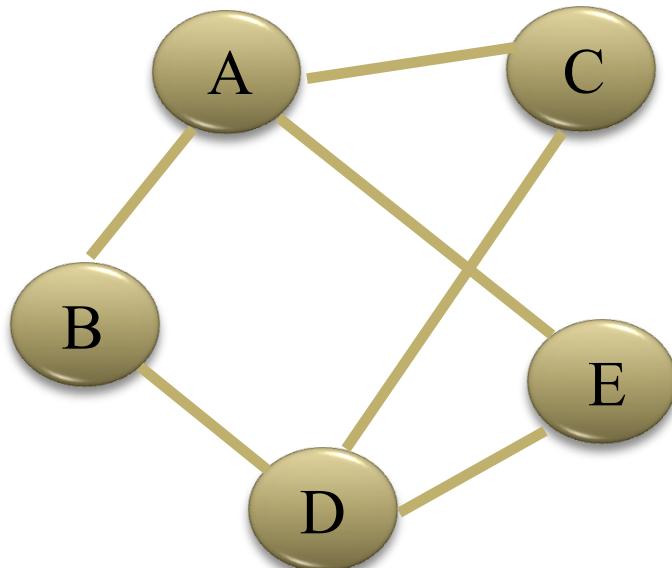
► Autómatas o Diagramas de Estado

Implementación de Grafos

► Formas de Implementar un grafo:

- Matriz de Adyacencias
- Lista de Adyacencias
- Lista de Arcos

Matriz de Adyacencias

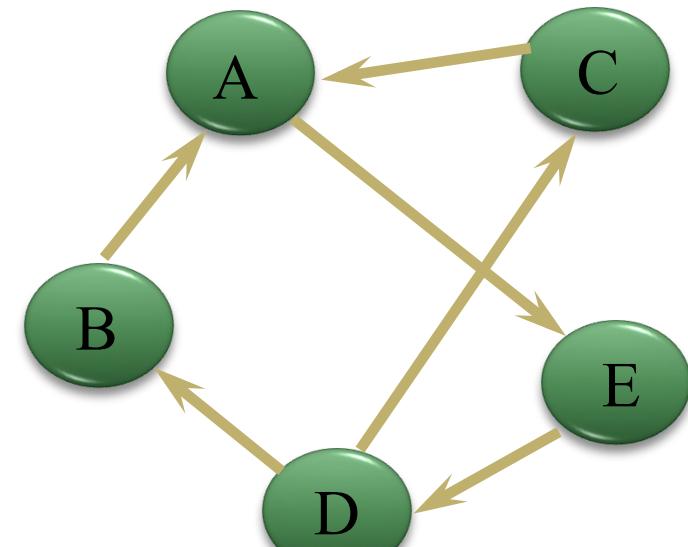


	A	B	C	D	E
A	F	T	T	F	T
B	T	F	F	T	F
C	T	F	F	T	F
D	F	T	T	F	T
E	T	F	F	T	F

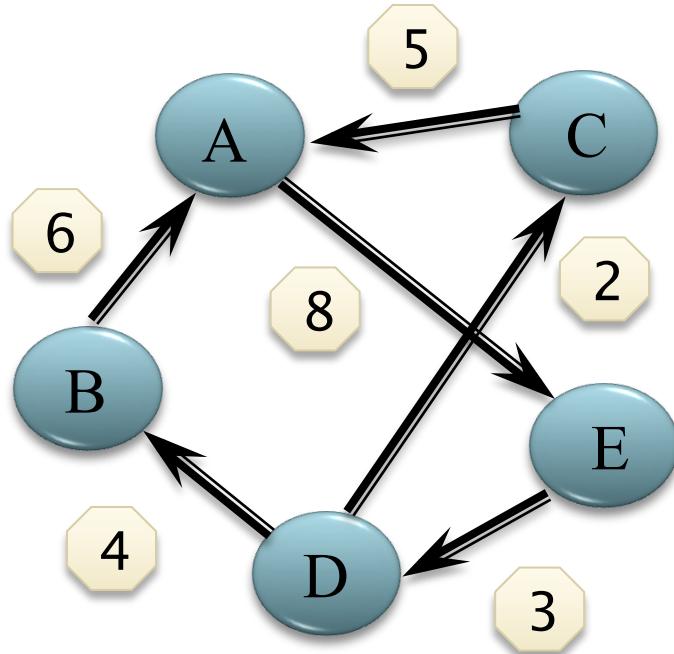
Cuando el grafo es NO DIRIGIDO, la matriz es SIMÉTRICA

Cuando el grafo es DIRIGIDO, se requiere toda la matriz

	A	B	C	D	E
A	F	F	F	F	T
B	T	F	F	F	F
C	T	F	F	F	F
D	F	T	T	F	F
E	F	F	F	T	F



Matriz de Adyacencias...



	A	B	C	D	E
A	0	0	0	0	8
B	6	0	0	0	0
C	5	0	0	0	0
D	0	4	2	0	0
E	0	0	0	3	0

Matriz de Adyacencias...

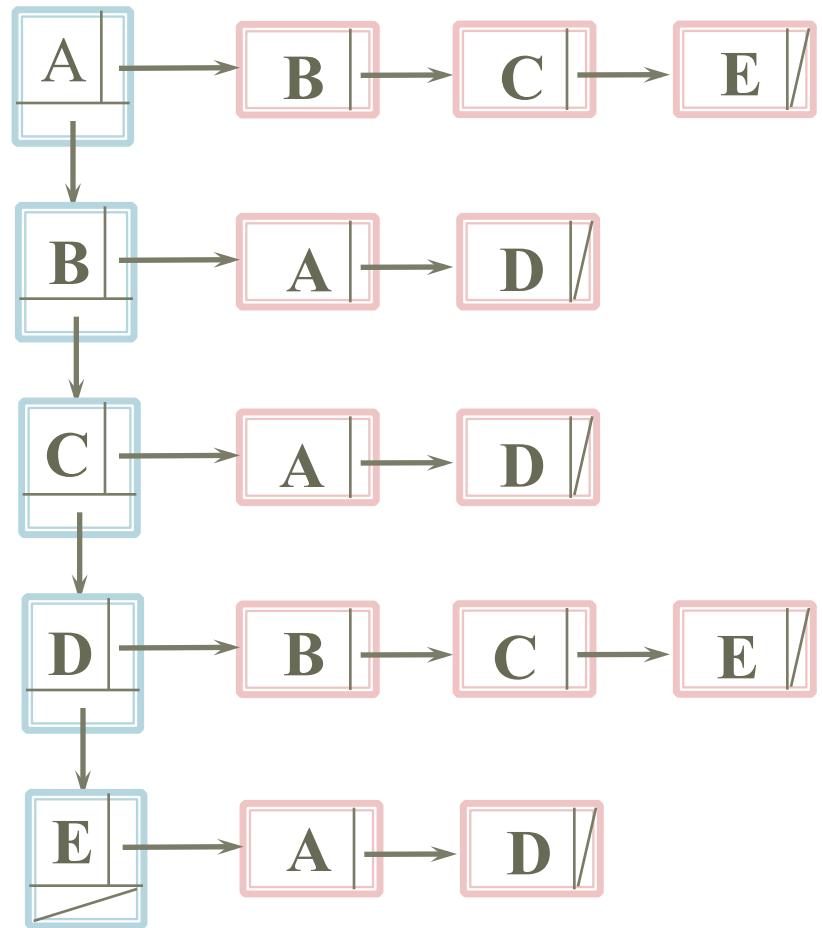
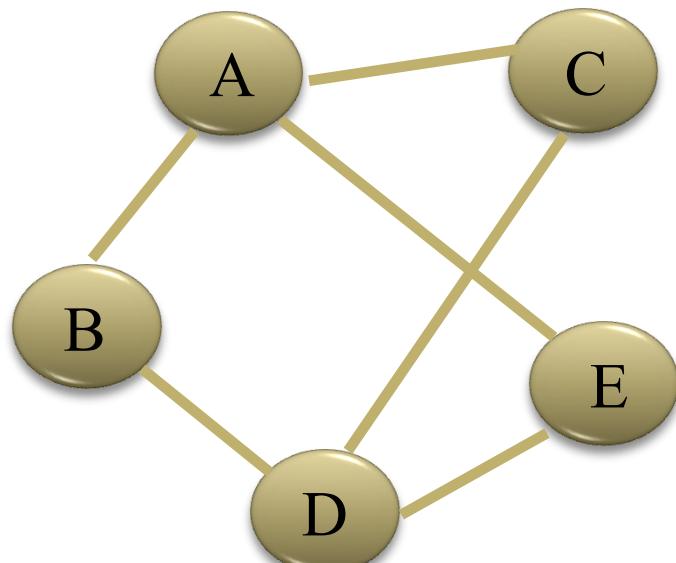
► VENTAJA

- Las operaciones sobre grafos son muy sencillas de implementar.

► DESVENTAJA

- Se requiere conocer con anticipación la cantidad de elementos que conformarán al grafo.

Lista de Adyacencias...



Lista de Adyacencias...

► VENTAJA

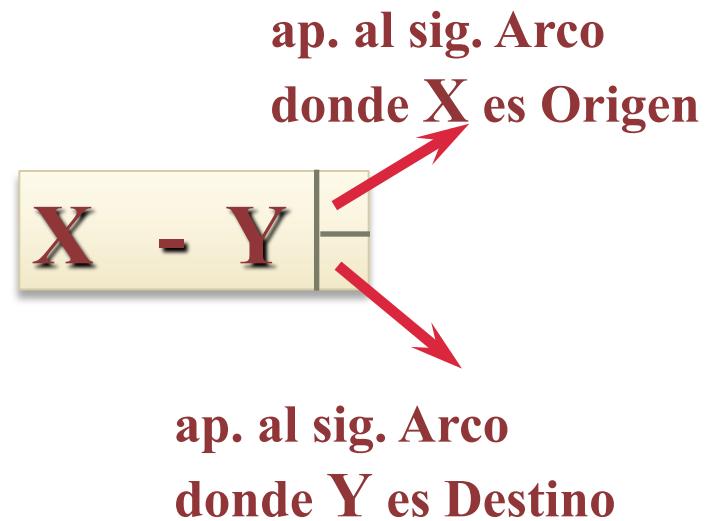
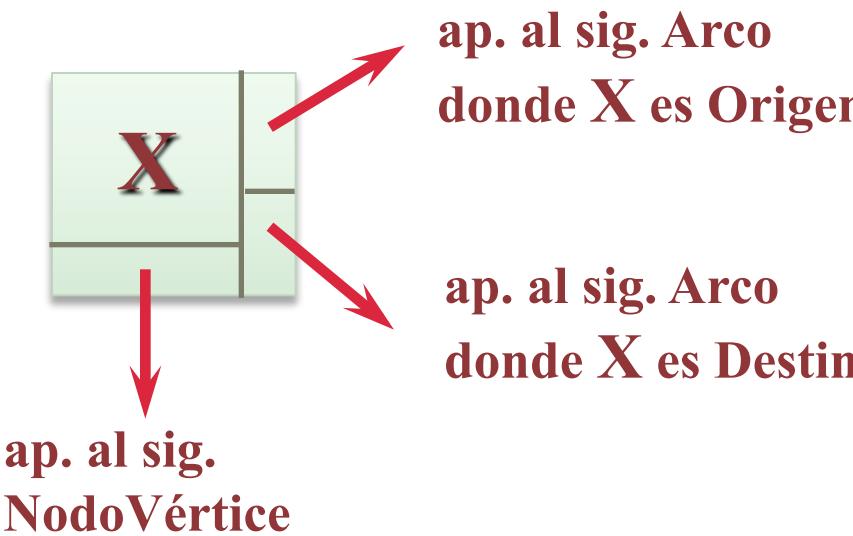
- No es necesaria conocer con anterioridad la cantidad de Nodos y de Arcos que conformarán al grafo.

► DESVENTAJA

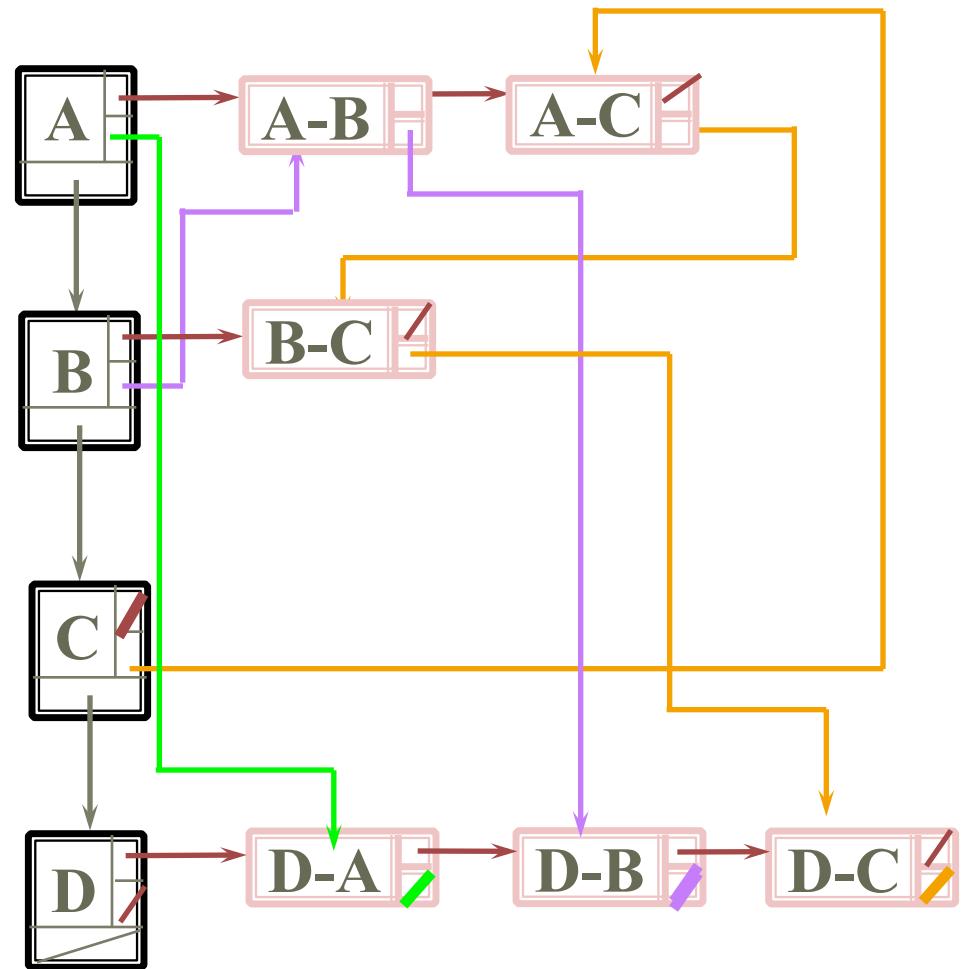
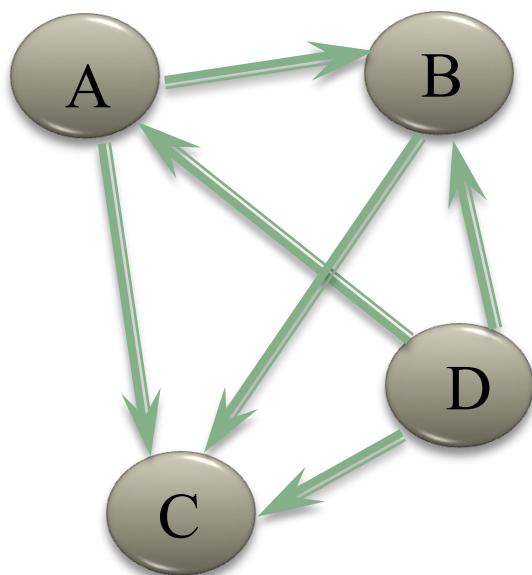
- Duplica información.
- Requiere más espacio de memoria debido a manejo de apuntadores.

Lista de Arcos...

- ▶ Implementación compleja pero más eficiente.
- ▶ Gráficamente los Nodos Vértice y los elementos de la lista de Arcos se verían como:



Ejemplo de Lista de Arcos..



Lista de Arcos...

▶ Ventajas

- Es una representación muy eficiente.

▶ Desventajas

- Es bastante compleja de implementar.
- Requiere más espacio de memoria debido al manejo de apuntadores.

Recorridos sobre un Grafo

- ▶ Los recorridos sobre un grafo más utilizados son:

Breadth First (Primero en Anchura)

Depth First (Primero en Profundidad)

Breadth First

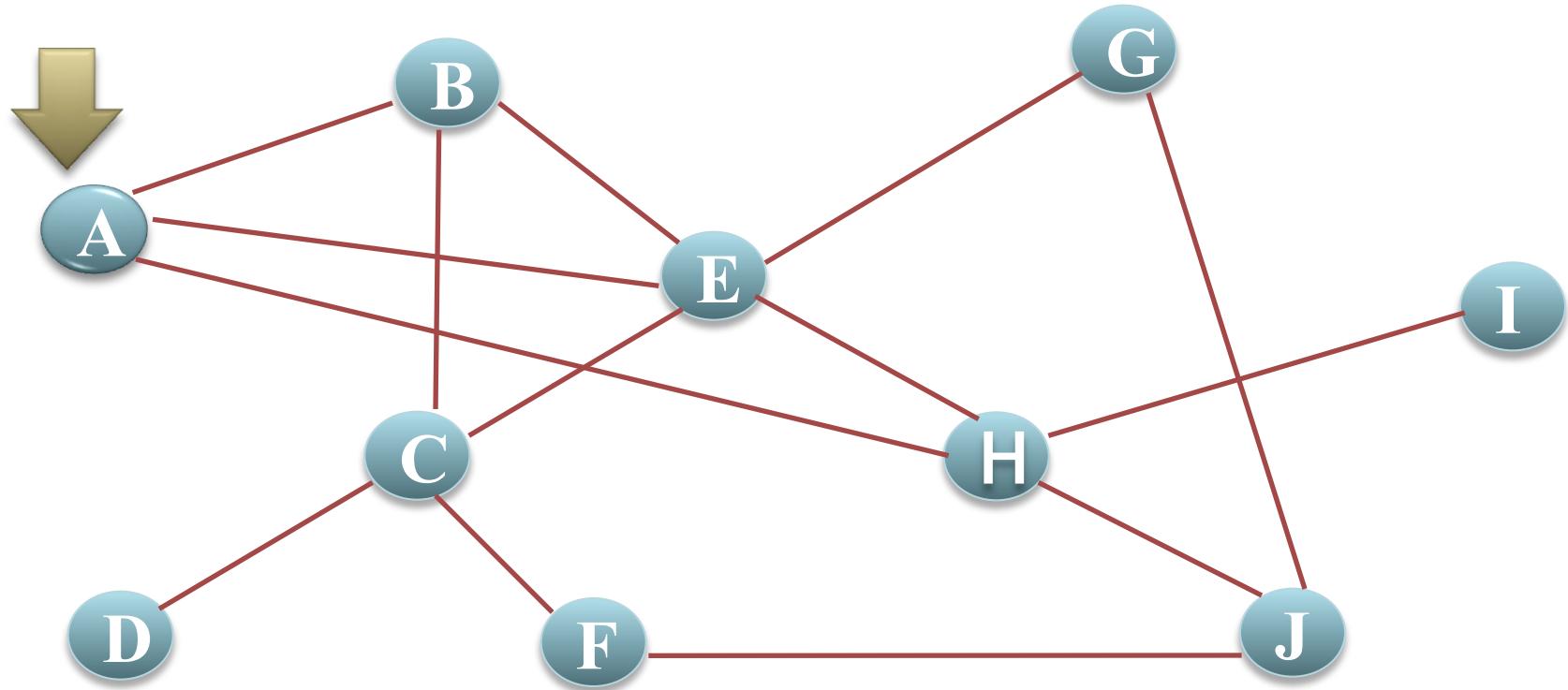
- ▶ Empezando con el Nodo de Inicio, “visita” un Nodo y luego a todos sus Vecinos.
- ▶ El orden en el que visita a los Vecinos depende del “orden” de almacenamiento.
- ▶ Utiliza una FILA auxiliar.
- ▶ Para evitar ciclos infinitos, todos los Nodos guardan un “status” (En Espera, Listo, Procesado)

Breadth First...

Iniciar el status de todos los Nodos a “En Espera”.

- Meter el Nodo de Inicio a la Fila y cambiarle el estatus a “Listo”.
 - Mientras la FILA no esté vacía:
 - Sacar Nodo de la FILA y procesarlo.
 - Cambiar el estatus de dicho nodo a “Procesado”.
 - Meter a la FILA todos los Vecinos del Nodo que tengan status “En Espera”, cambiándolo a “Listo”.

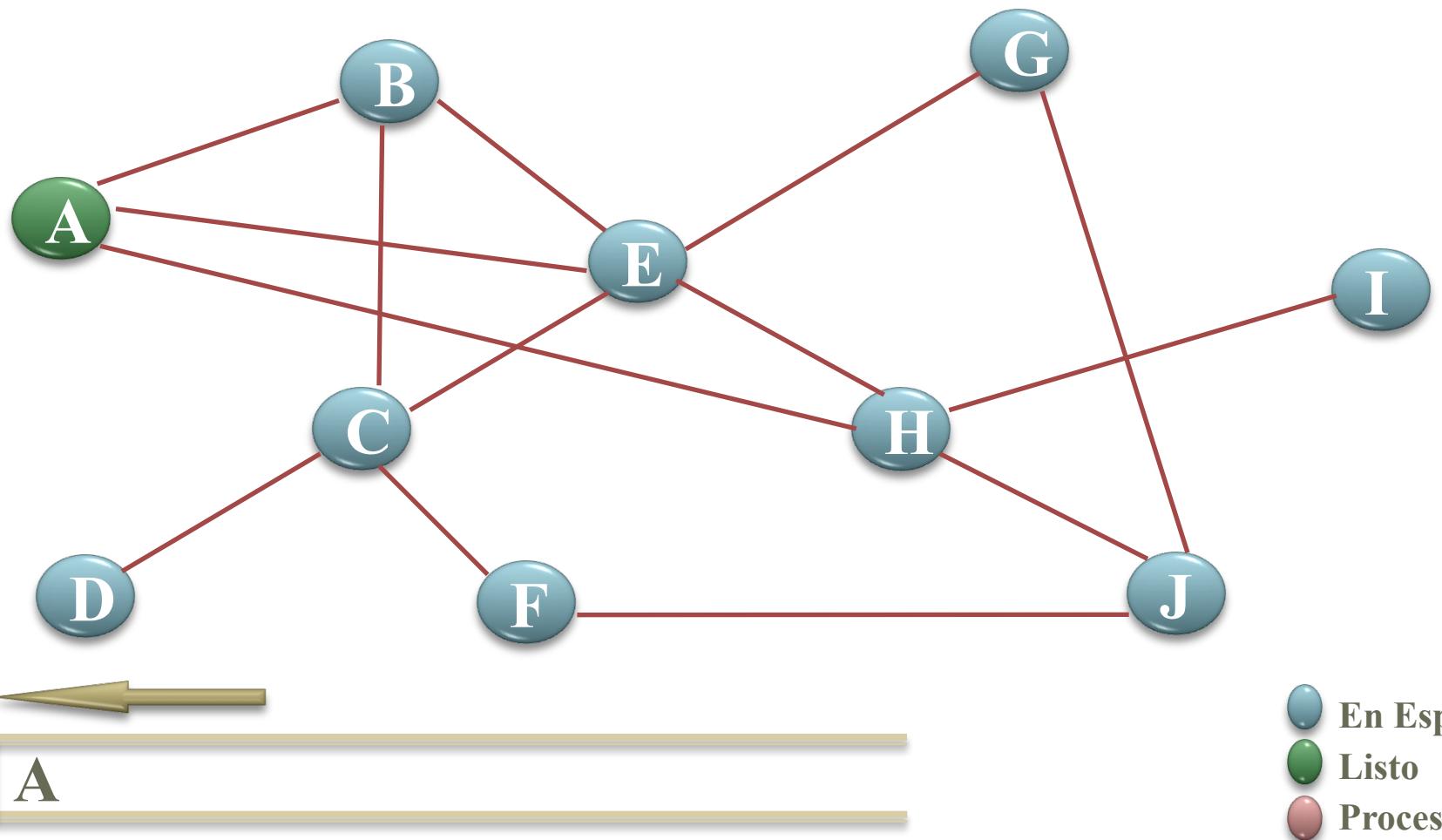
Ejemplo... Breadth First



Suponer que están almacenados
en orden alfabético.

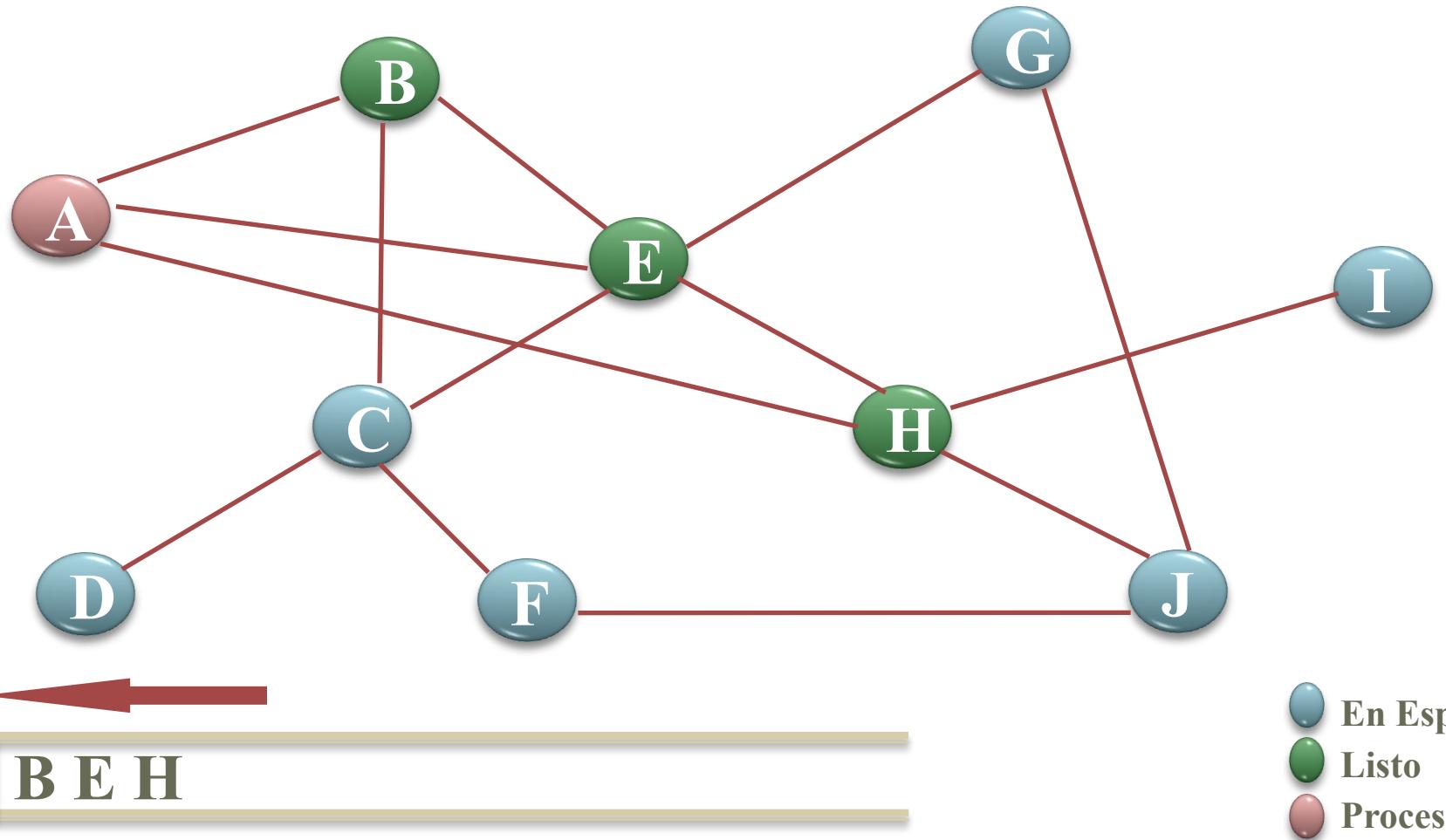
- En Espera
- Listo
- Procesado

Ejemplo... Breadth First



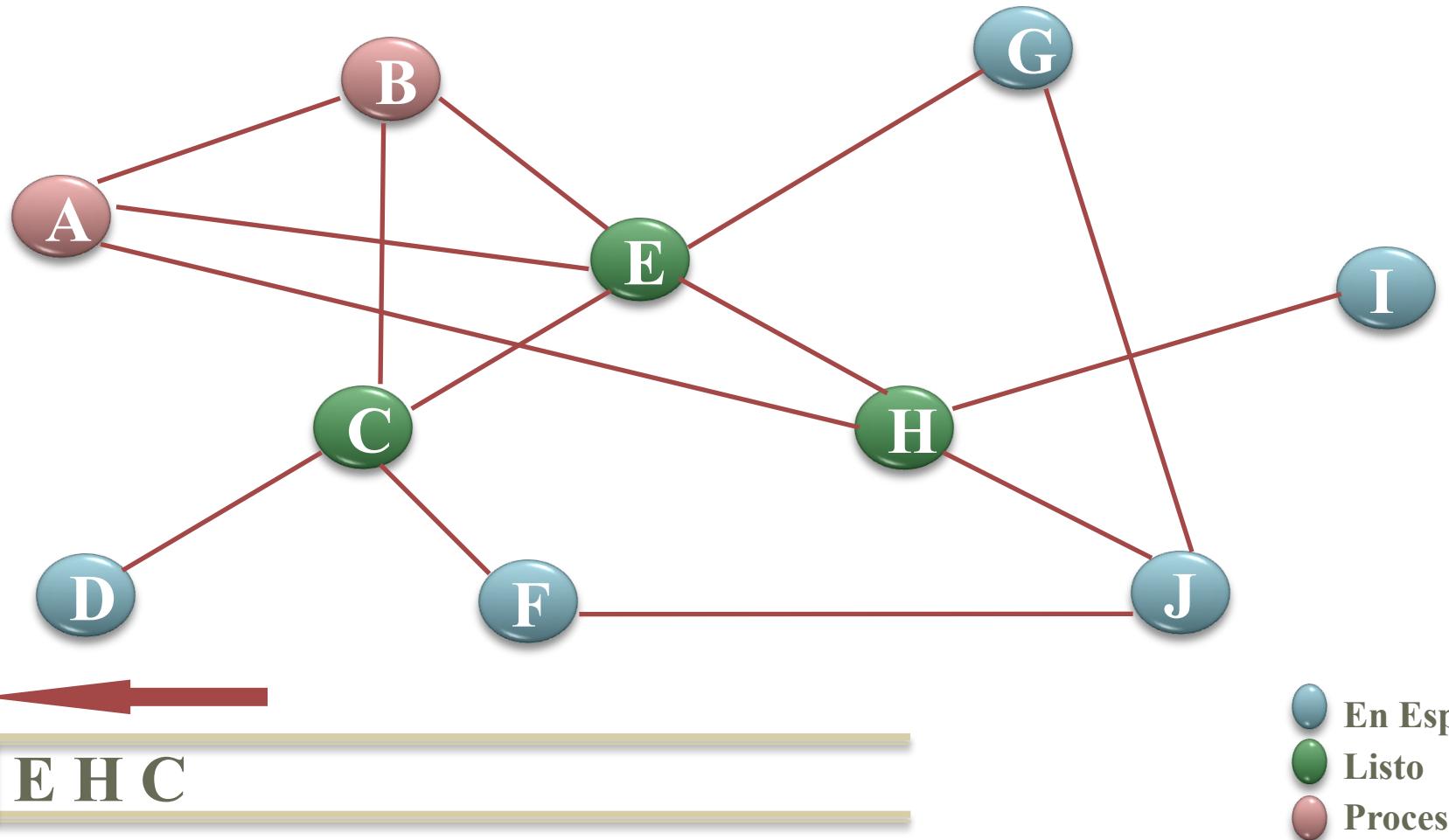
Desplegado:

Ejemplo... Breadth First



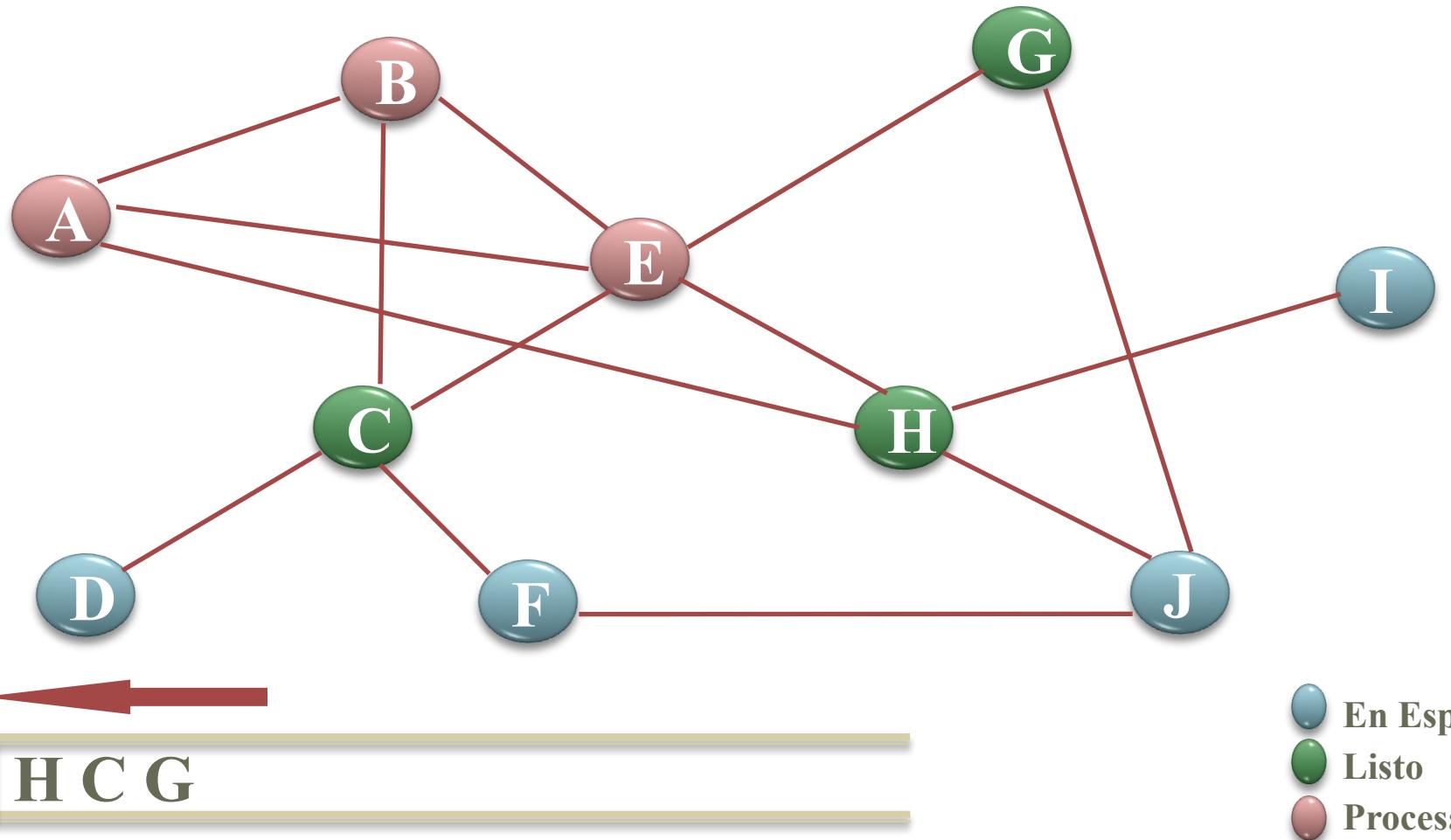
Desplegado: A

Ejemplo... Breadth First



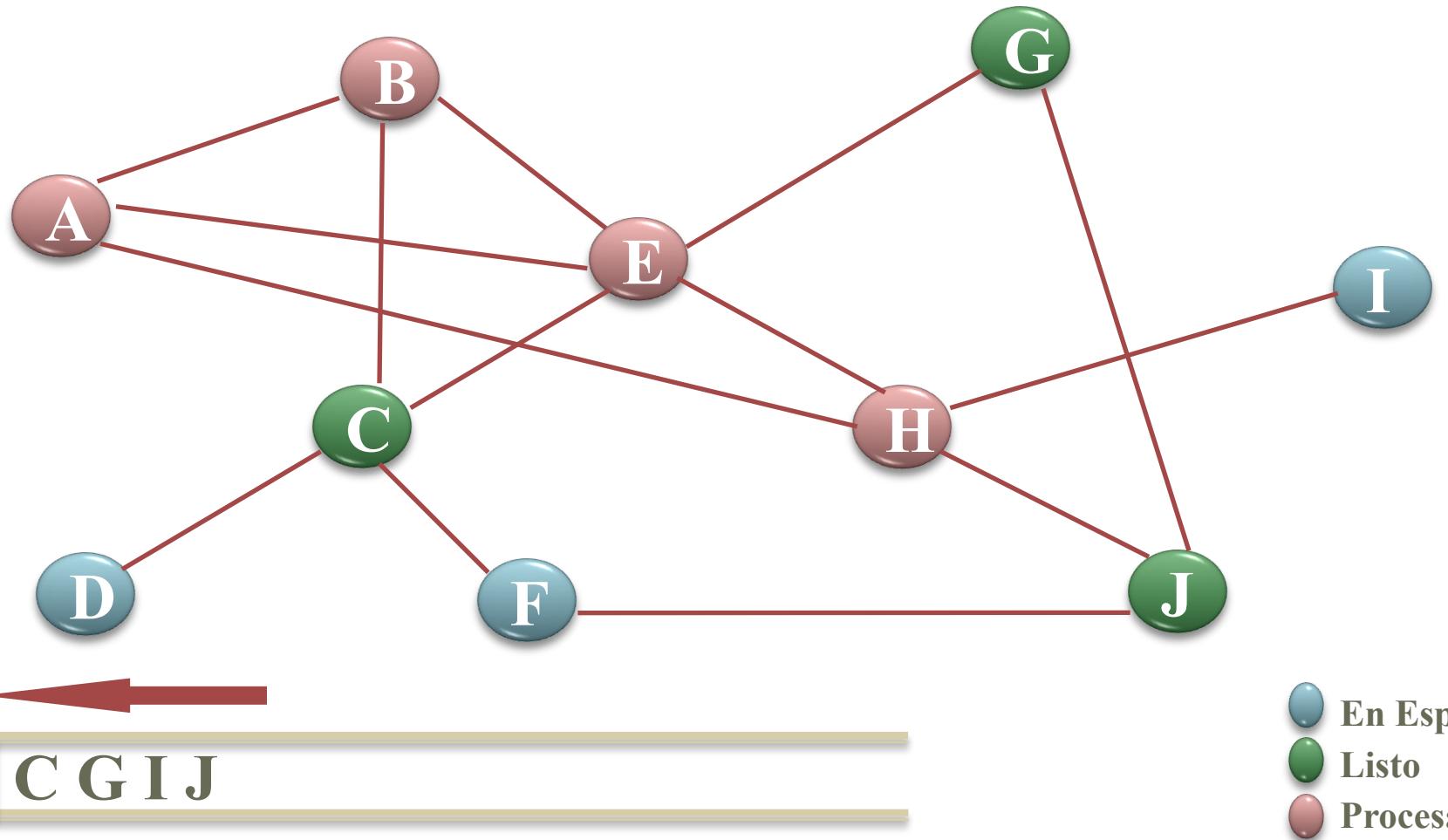
Desplegado: **A B**

Ejemplo... Breadth First



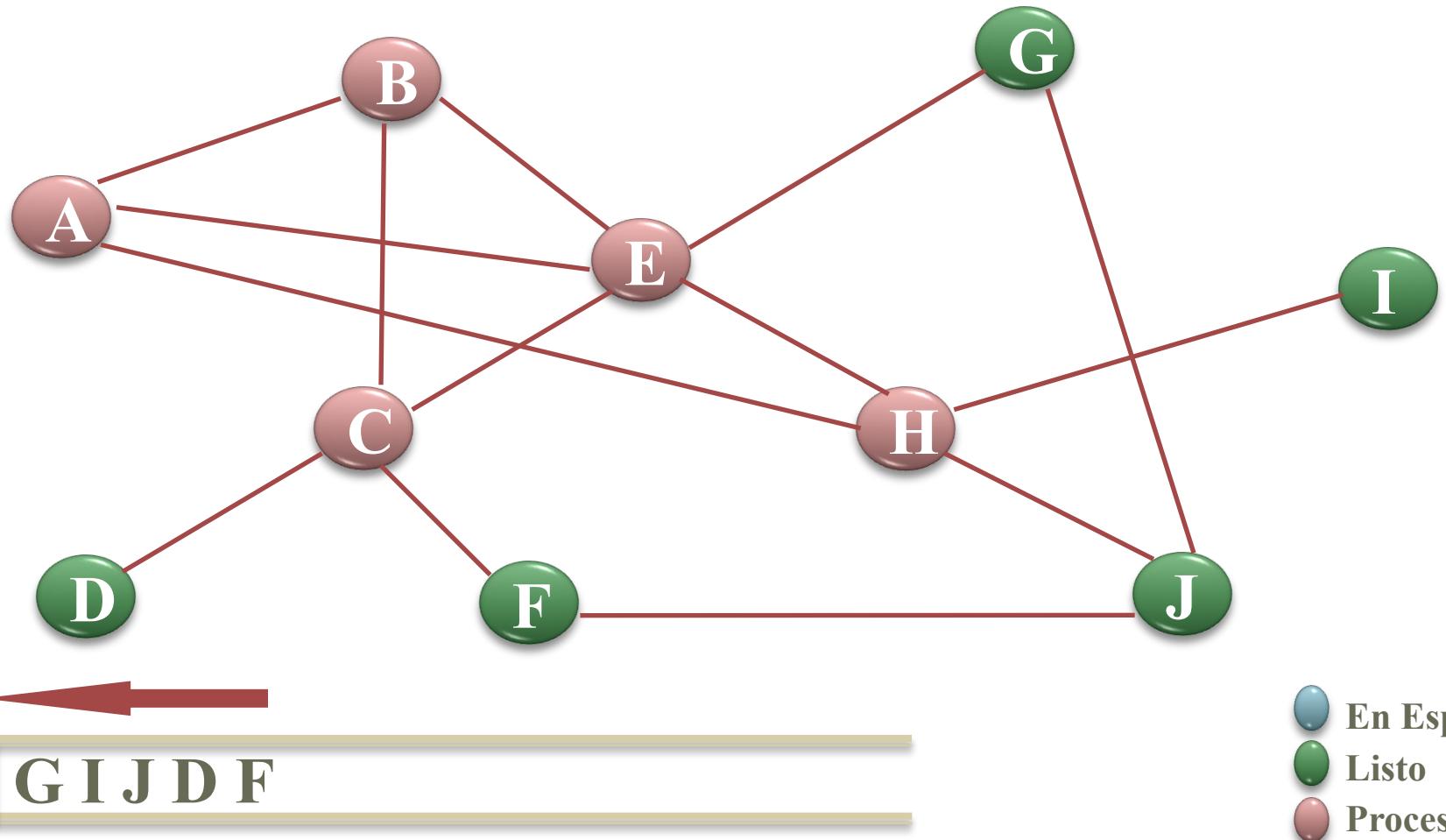
Desplegado: **A B E**

Ejemplo... Breadth First



Desplegado: **A B E H**

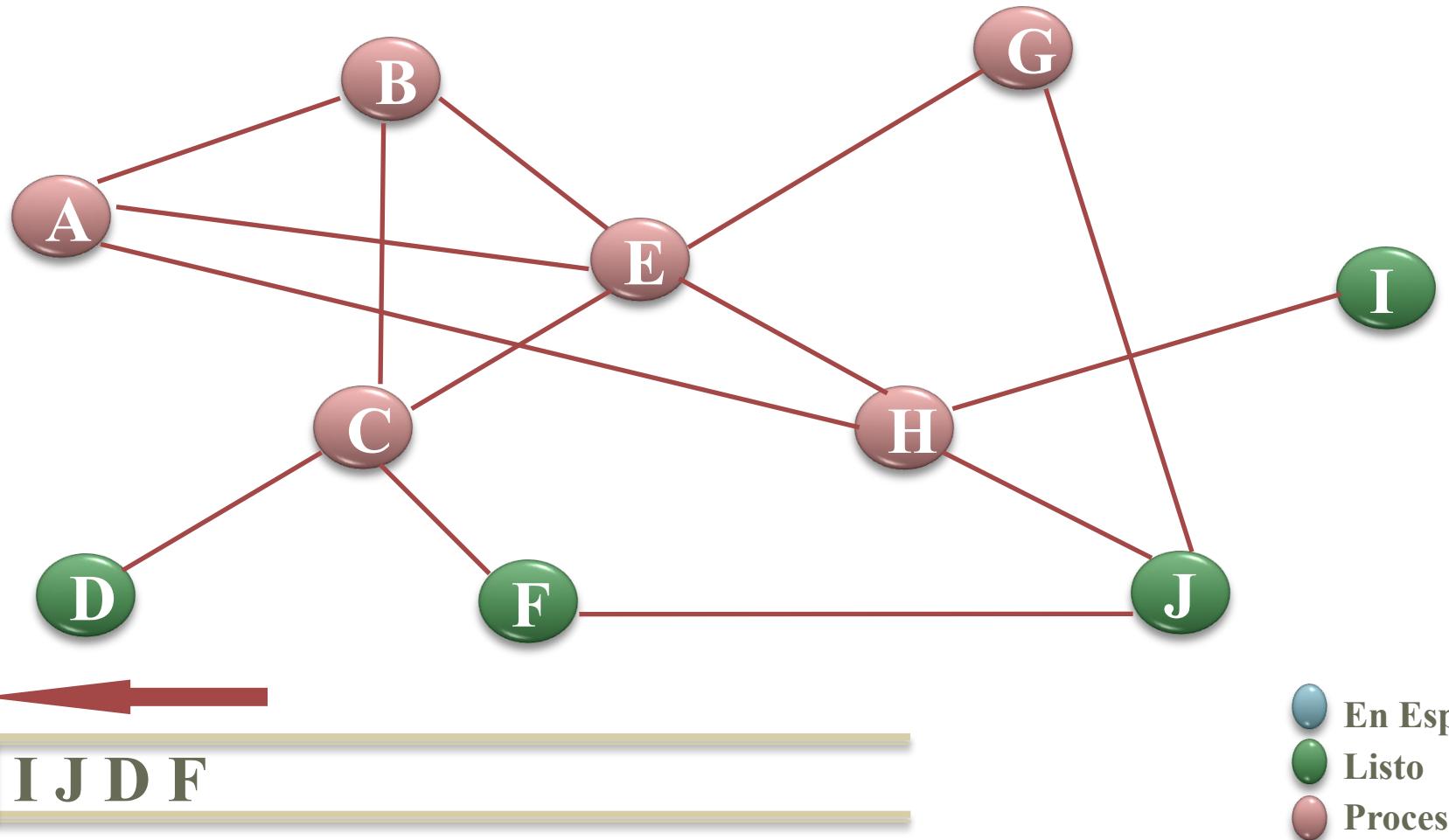
Ejemplo... Breadth First



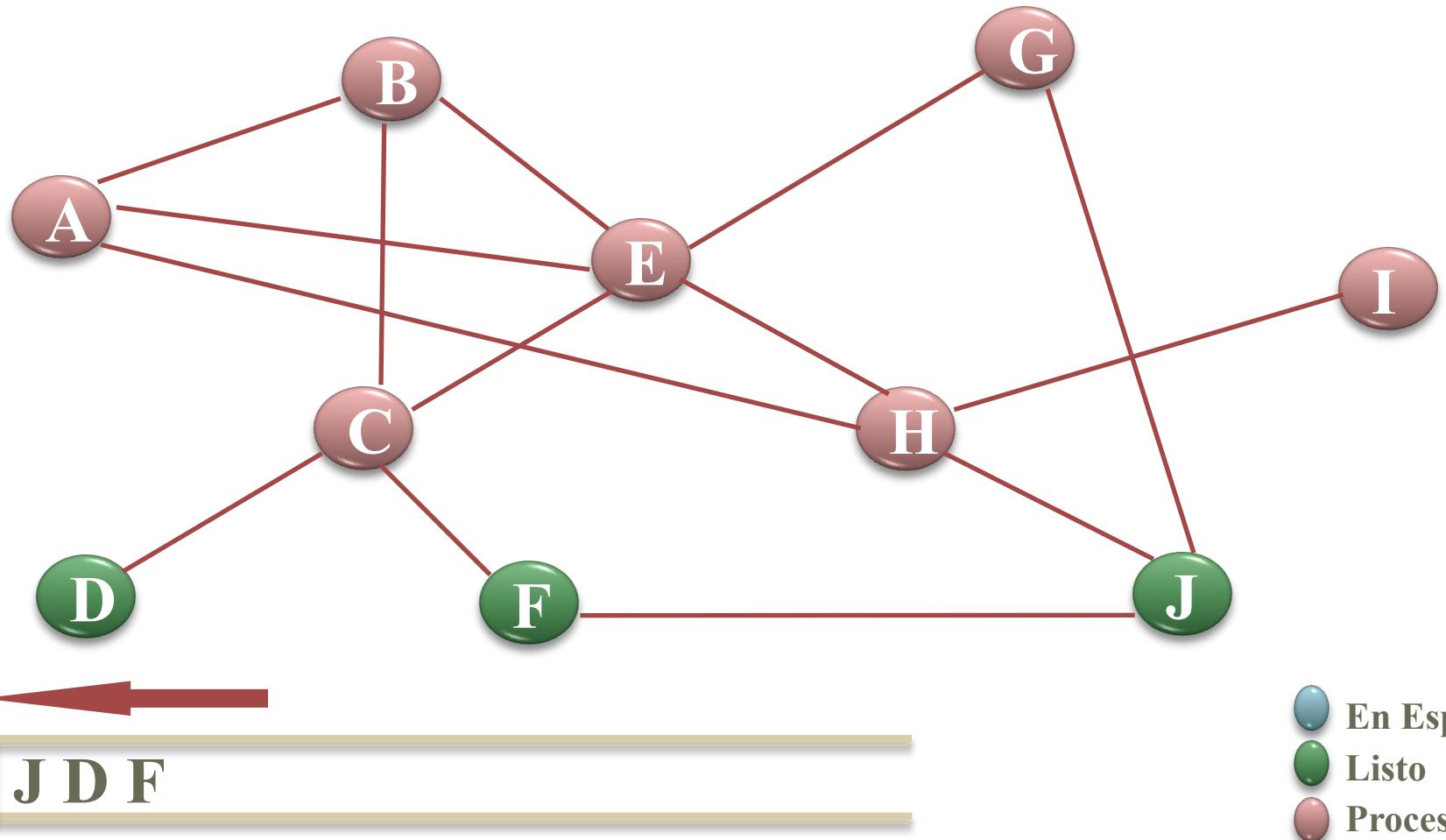
Fila **G I J D F**

Desplegado: **A B E H C**

Ejemplo... Breadth First

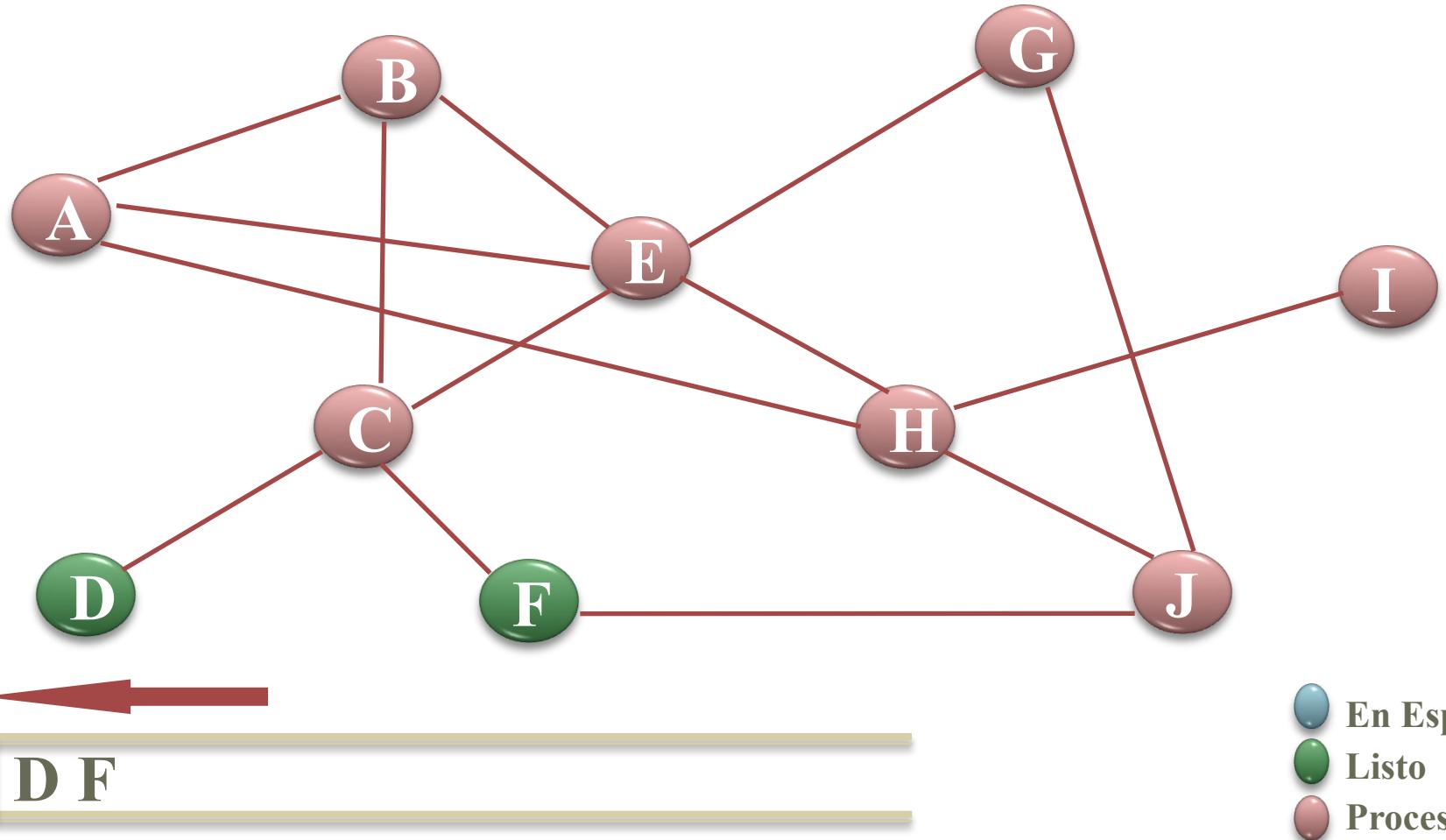


Ejemplo... Breadth First



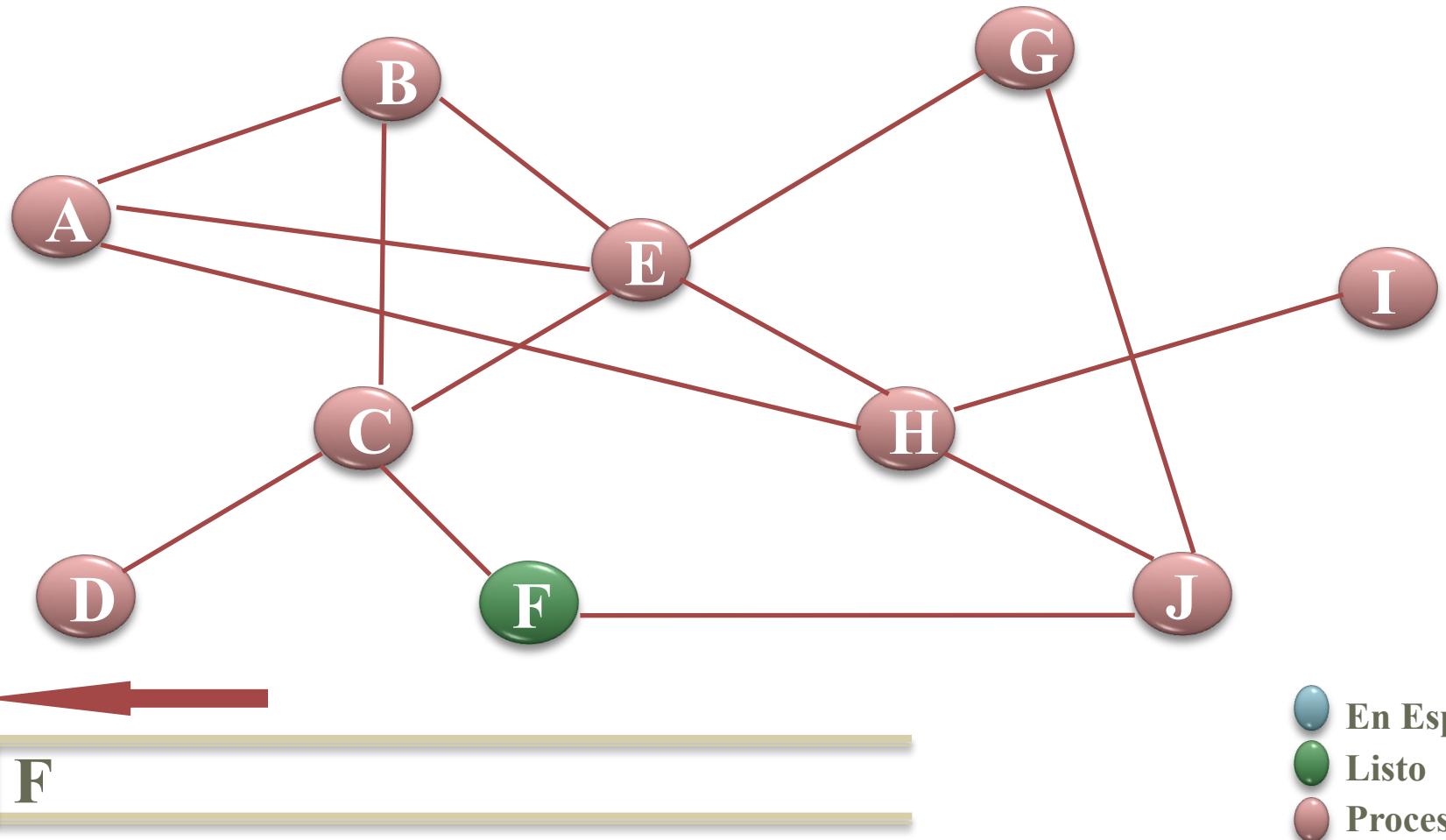
Desplegado: **A B E H C G I**

Ejemplo... Breadth First



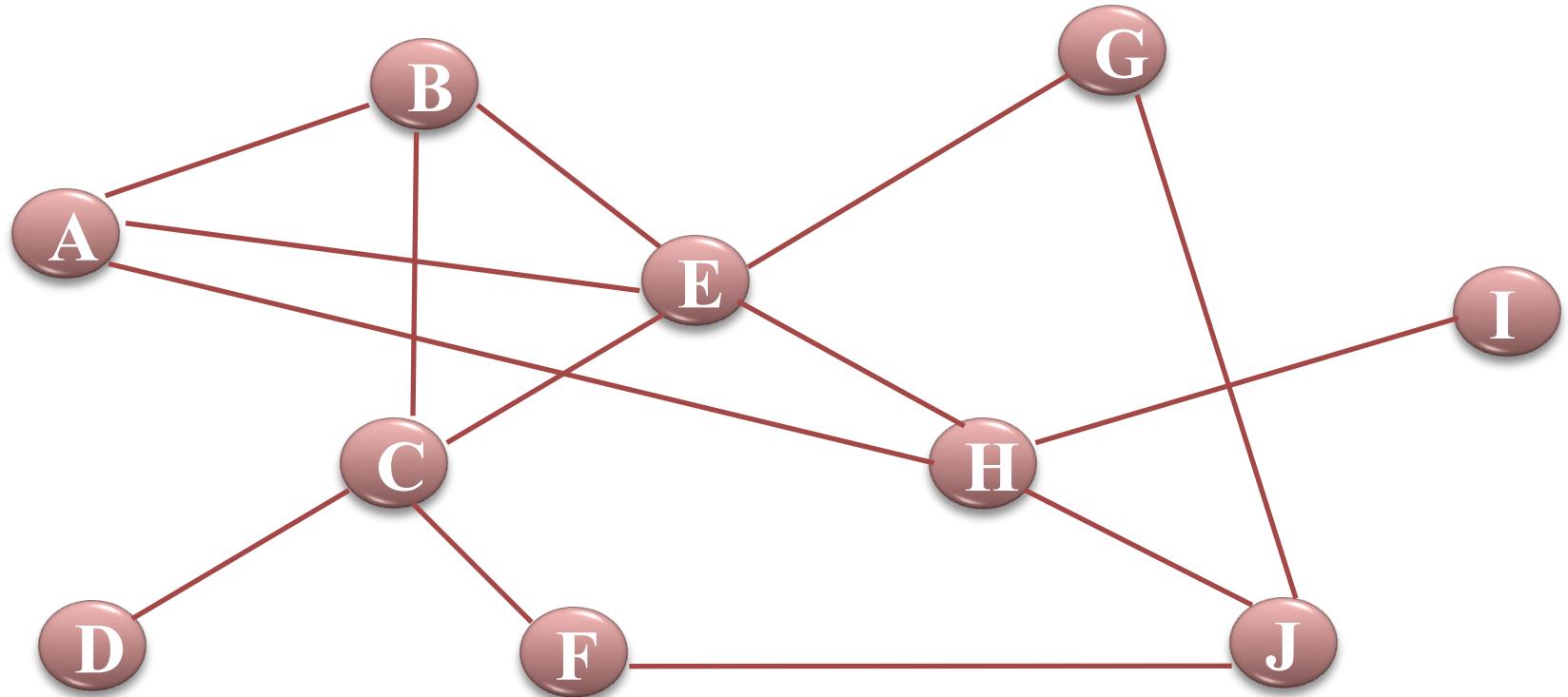
Desplegado: **A B E H C G I J**

Ejemplo... Breadth First



Desplegado: **A B E H C G I J D**

Ejemplo... Breadth First

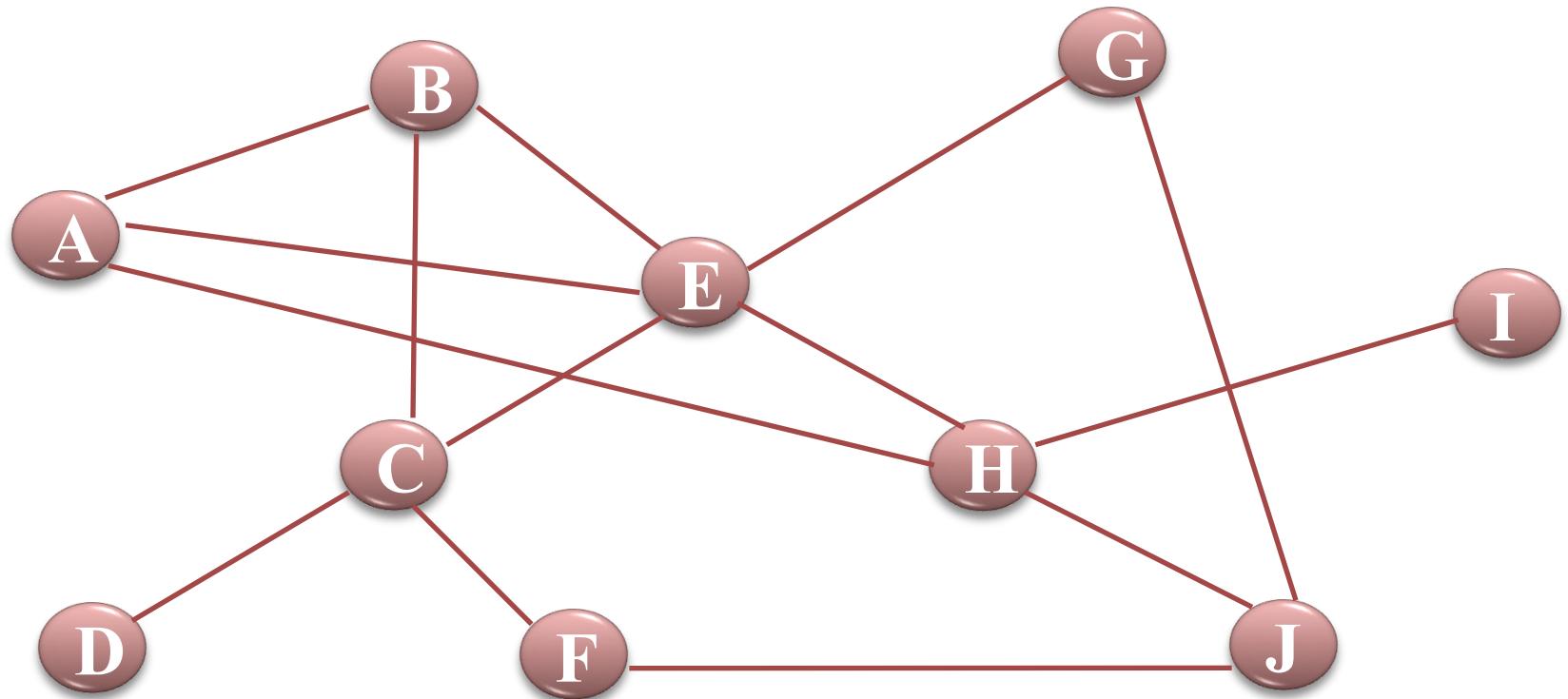


- En Espera
- Listo
- Procesado

Fila

Desplegado: **A B E H C G I J D F**

Ejemplo... Breadth First



Desplegado: **A B E H C G I J D F**

Depth First

- ▶ Empezando en el Nodo de Inicio, “visita” el Nodo y luego visita recursivamente a todos sus Vecinos NO procesados.
- ▶ El orden en el que visita a los Vecinos depende del “orden” de almacenamiento.
- ▶ Este algoritmo utiliza una PILA auxiliar (para simular la recursividad).
- ▶ Para evitar ciclos infinitos, todos los Nodos guardan un “status” (En Espera, Procesado)

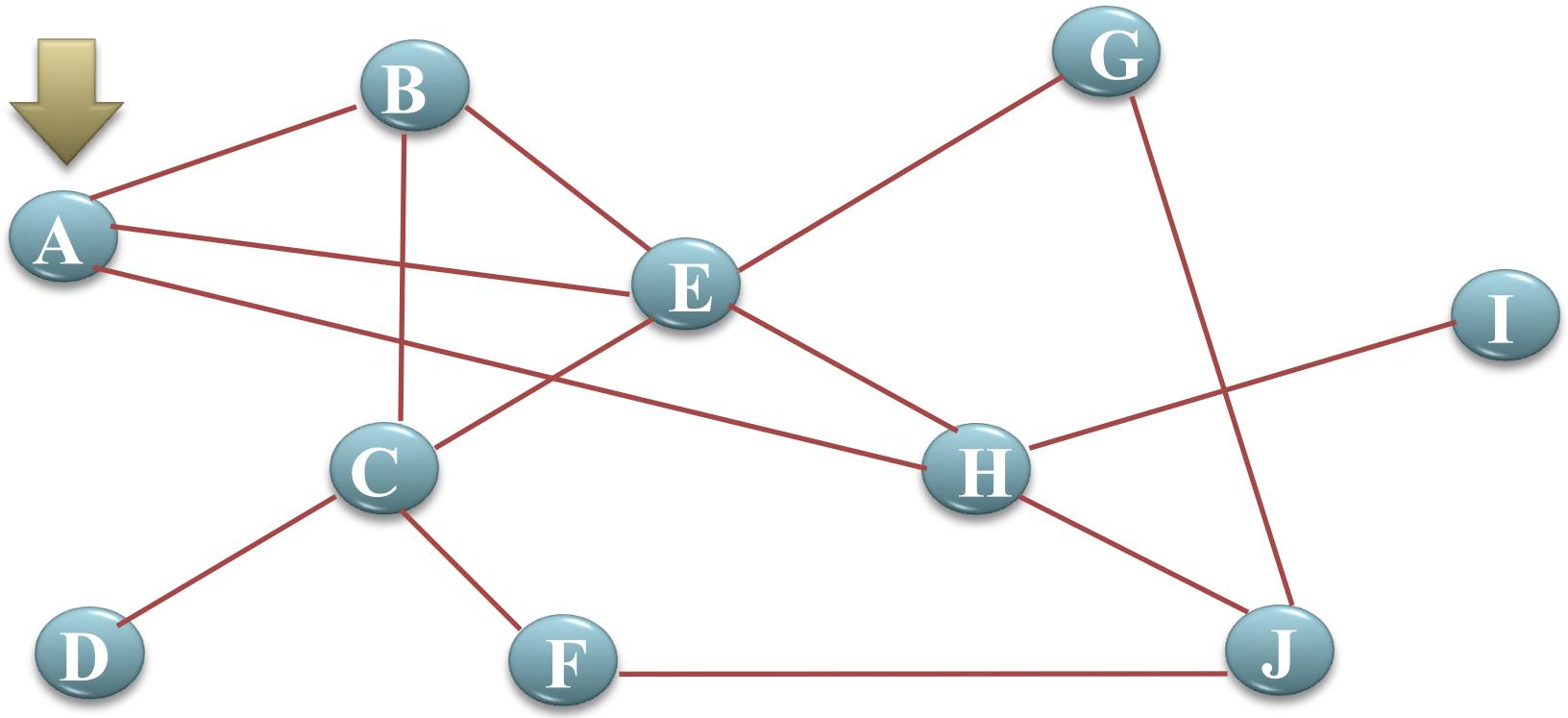
Depth First

Iniciar el status de todos los Nodos a “En Espera”.
Agrega el Nodo de inicio a la Pila

Mientras la PILA no esté vacía:

- * Sacar Nodo de la PILA y procesarlo.
- * Cambiar su status a “Procesado”.
- * Meter a la PILA todos los Vecinos del Nodo que NO tengan status “Procesado”.
- * Si un Nodo ya existía, se deja el último que llegó.

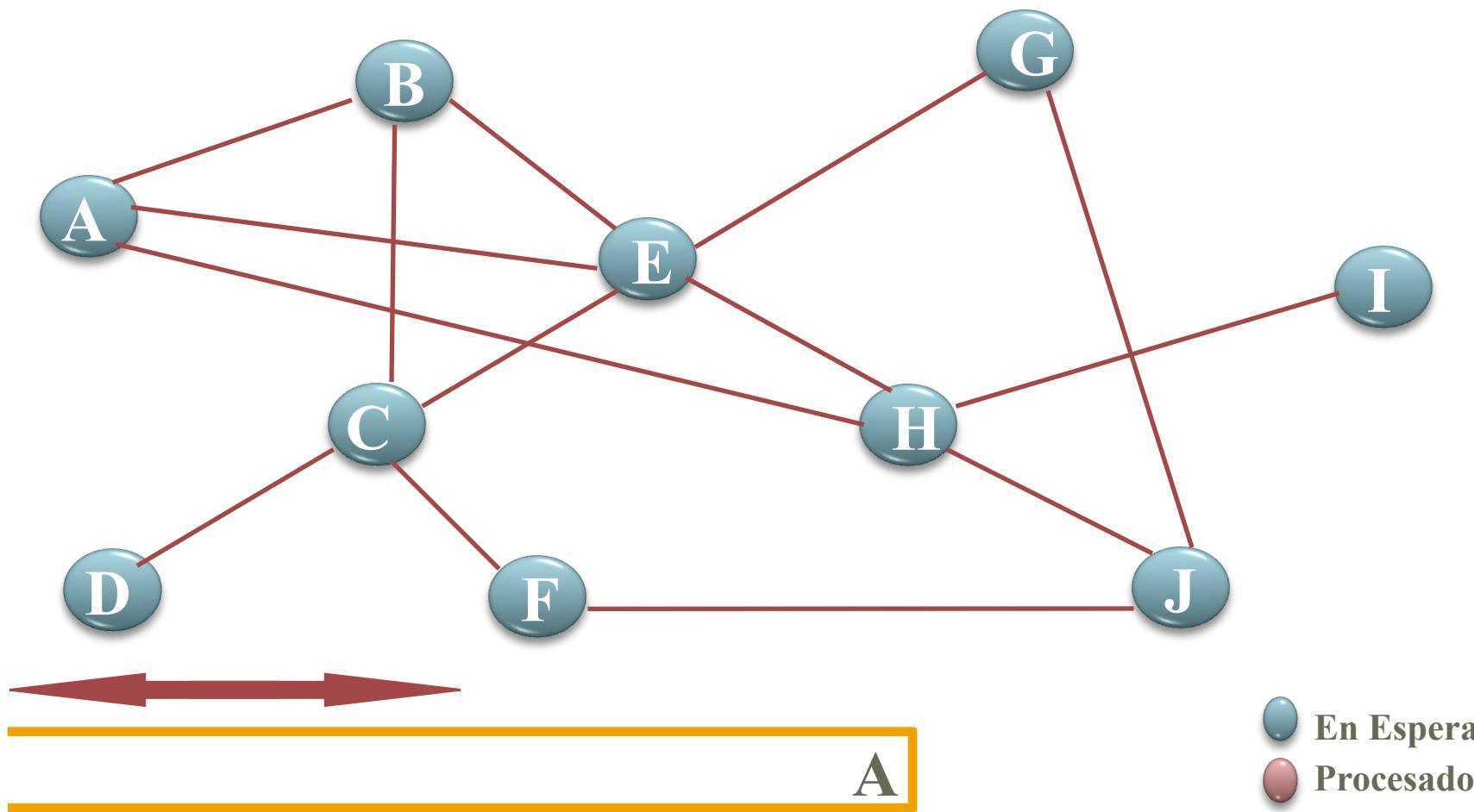
Ejemplo... Depth First



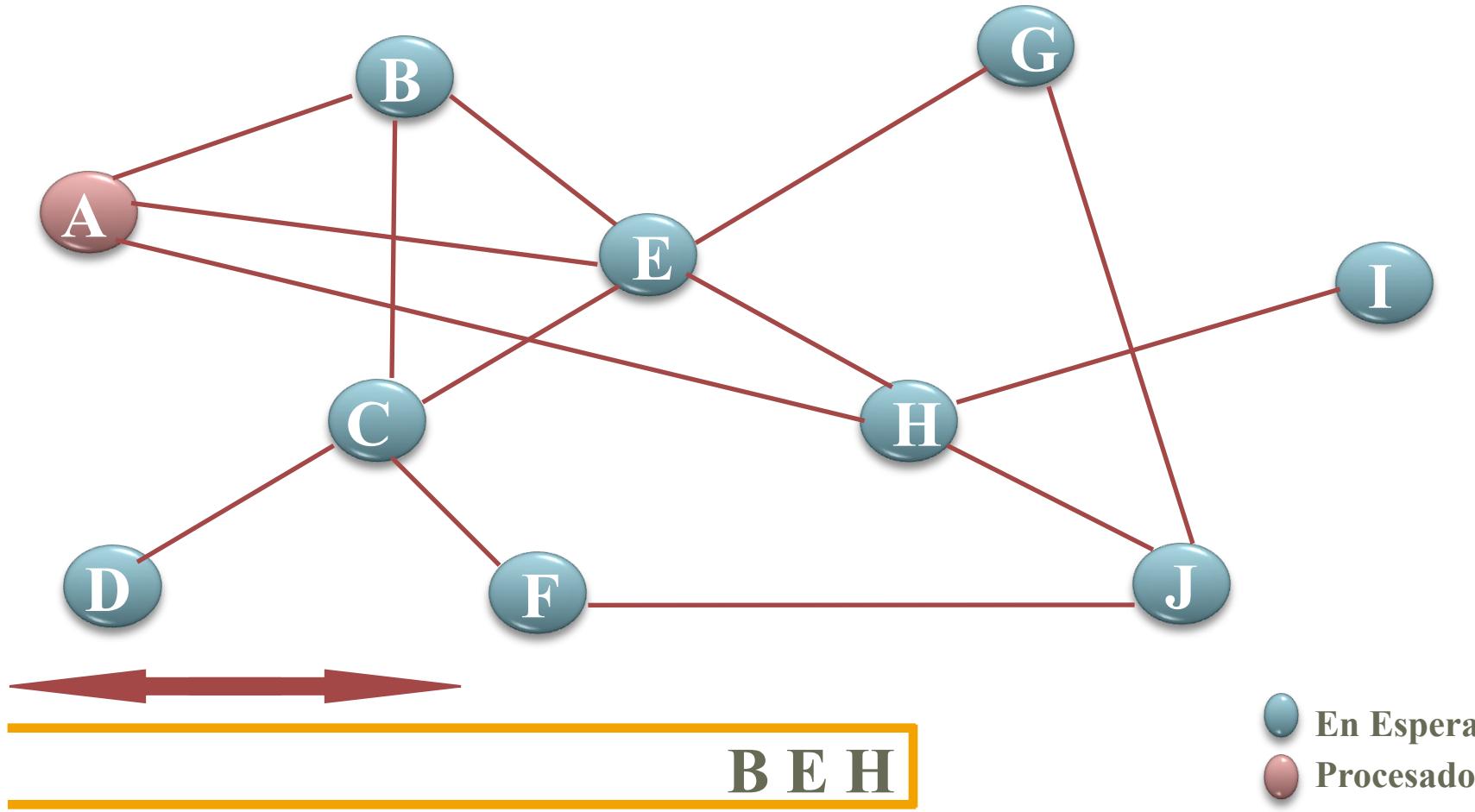
- En Espera
- Procesado

Suponer que están almacenados
en orden alfabético.

Ejemplo... Depth First

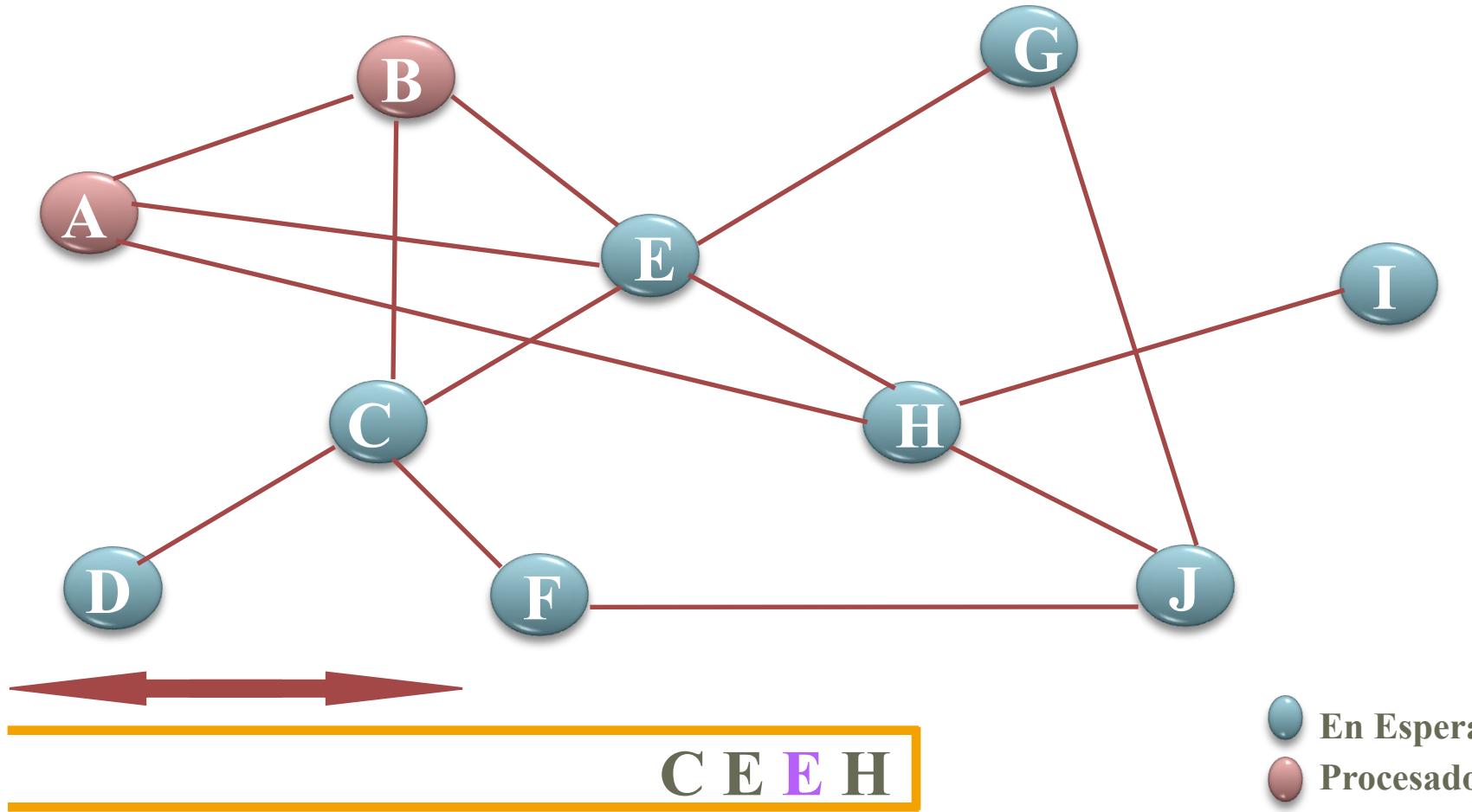


Ejemplo... Depth First



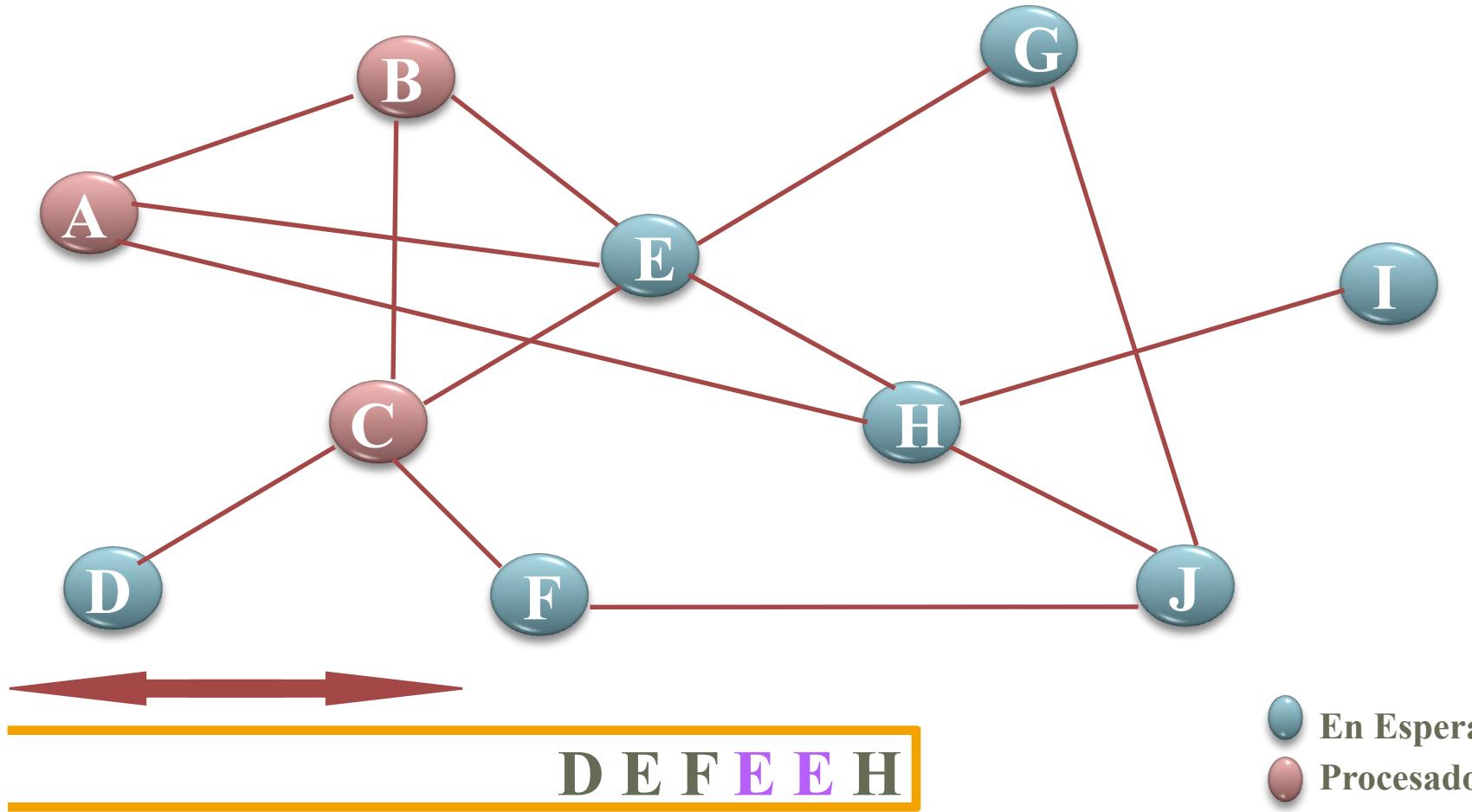
Desplegado: A

Ejemplo... Depth First



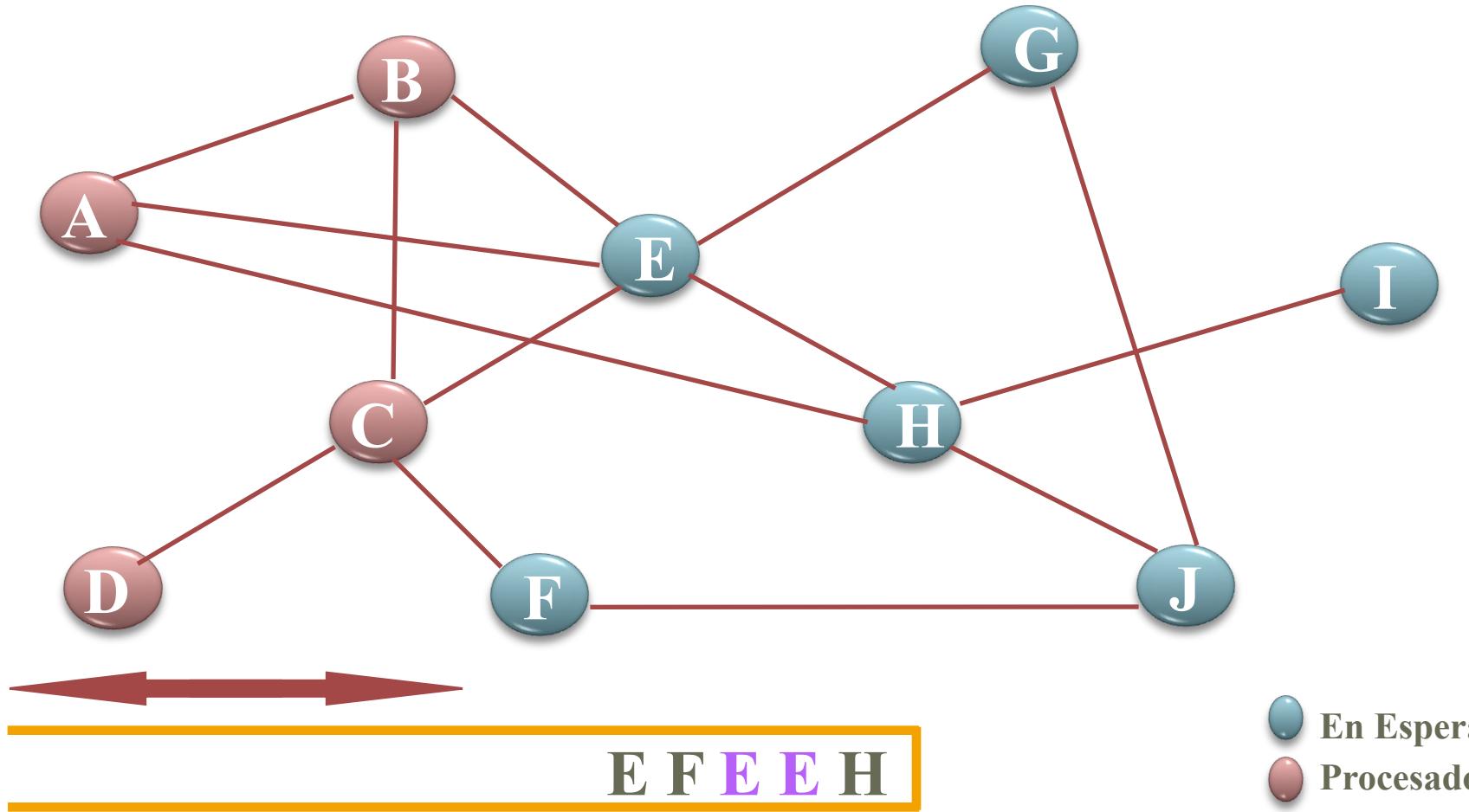
Desplegado: **A B**

Ejemplo... Depth First



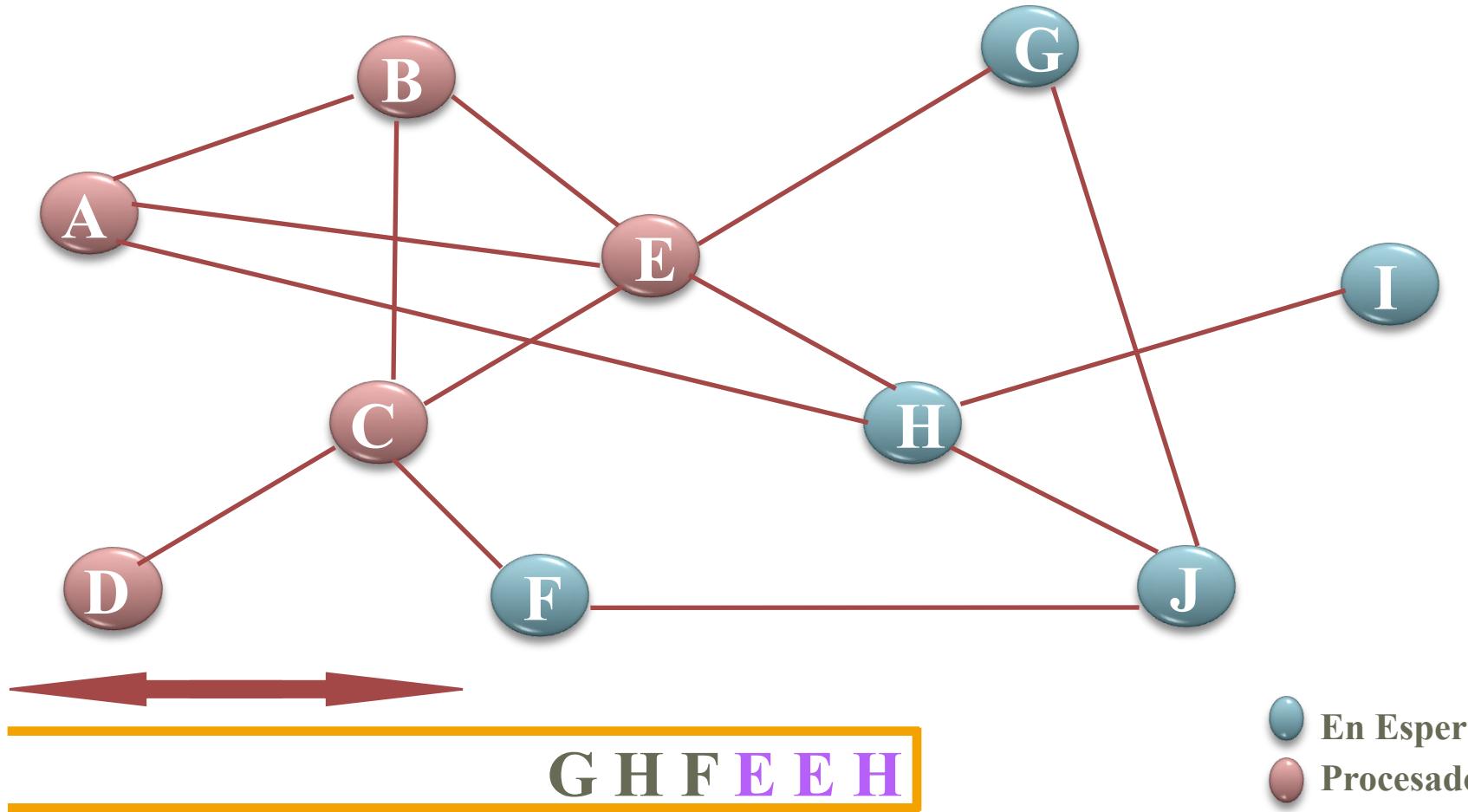
Desplegado: A B C

Ejemplo... Depth First



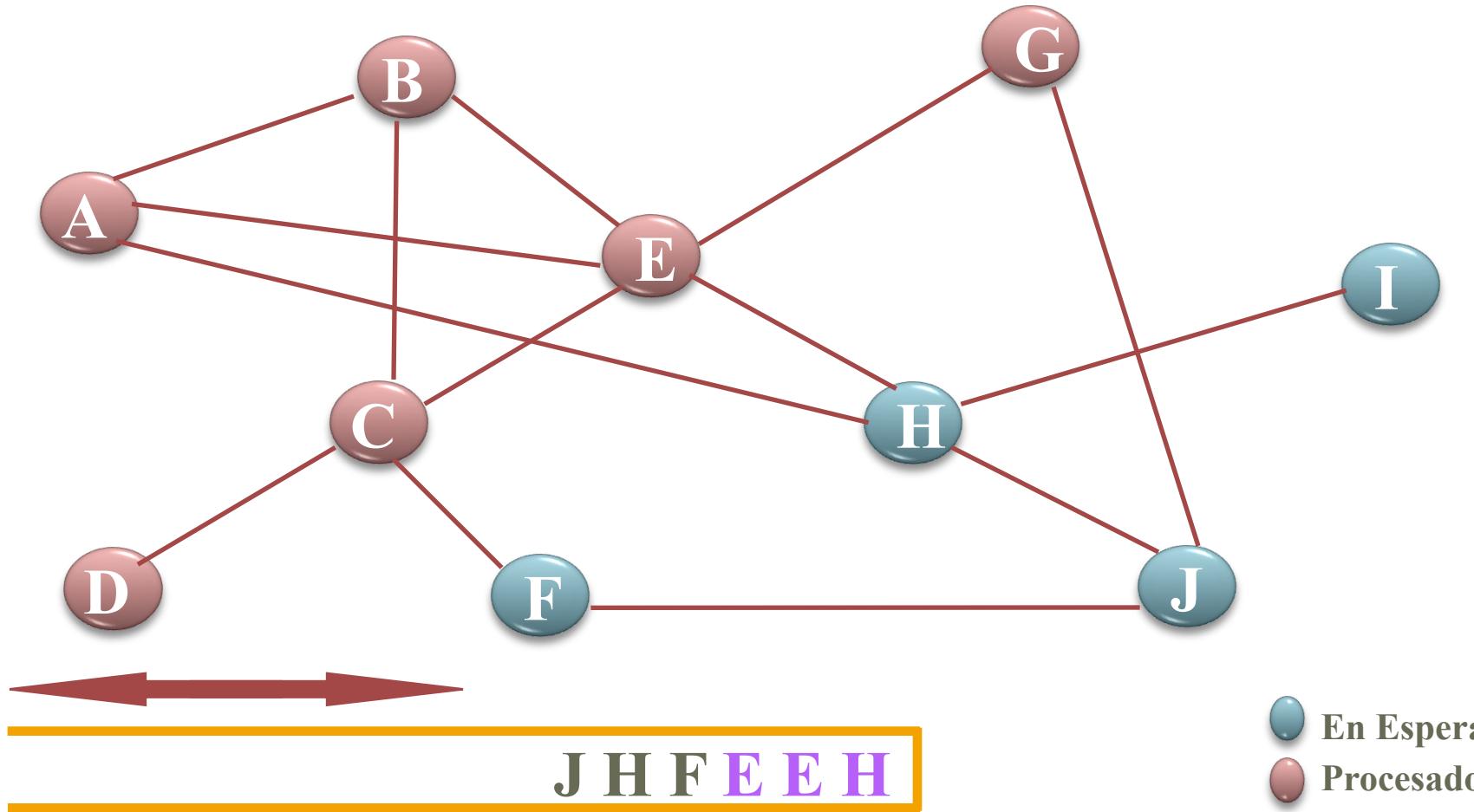
Desplegado: A B C D

Ejemplo... Depth First



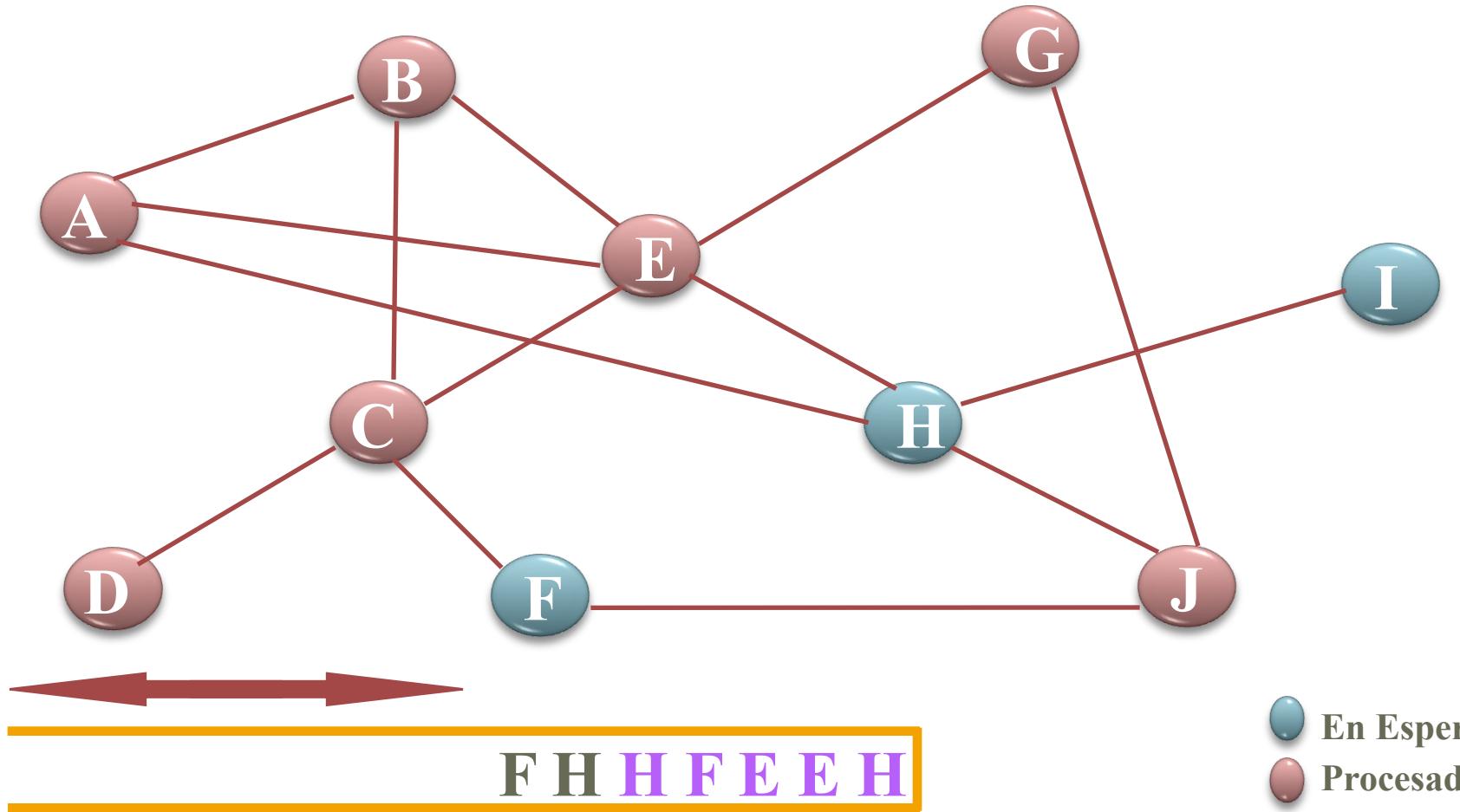
Desplegado: A B C D E

Ejemplo... Depth First



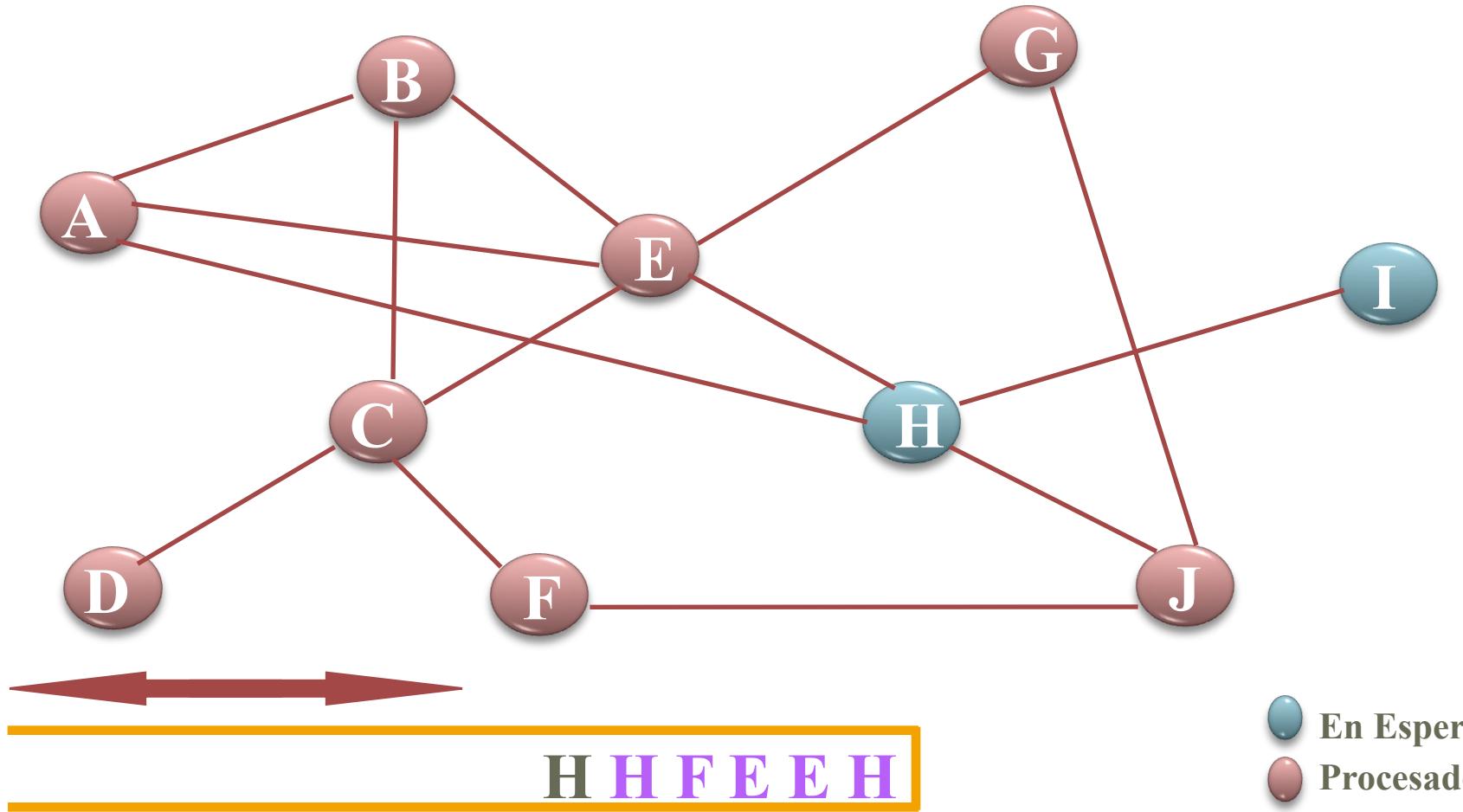
Desplegado: A B C D E G

Ejemplo... Depth First



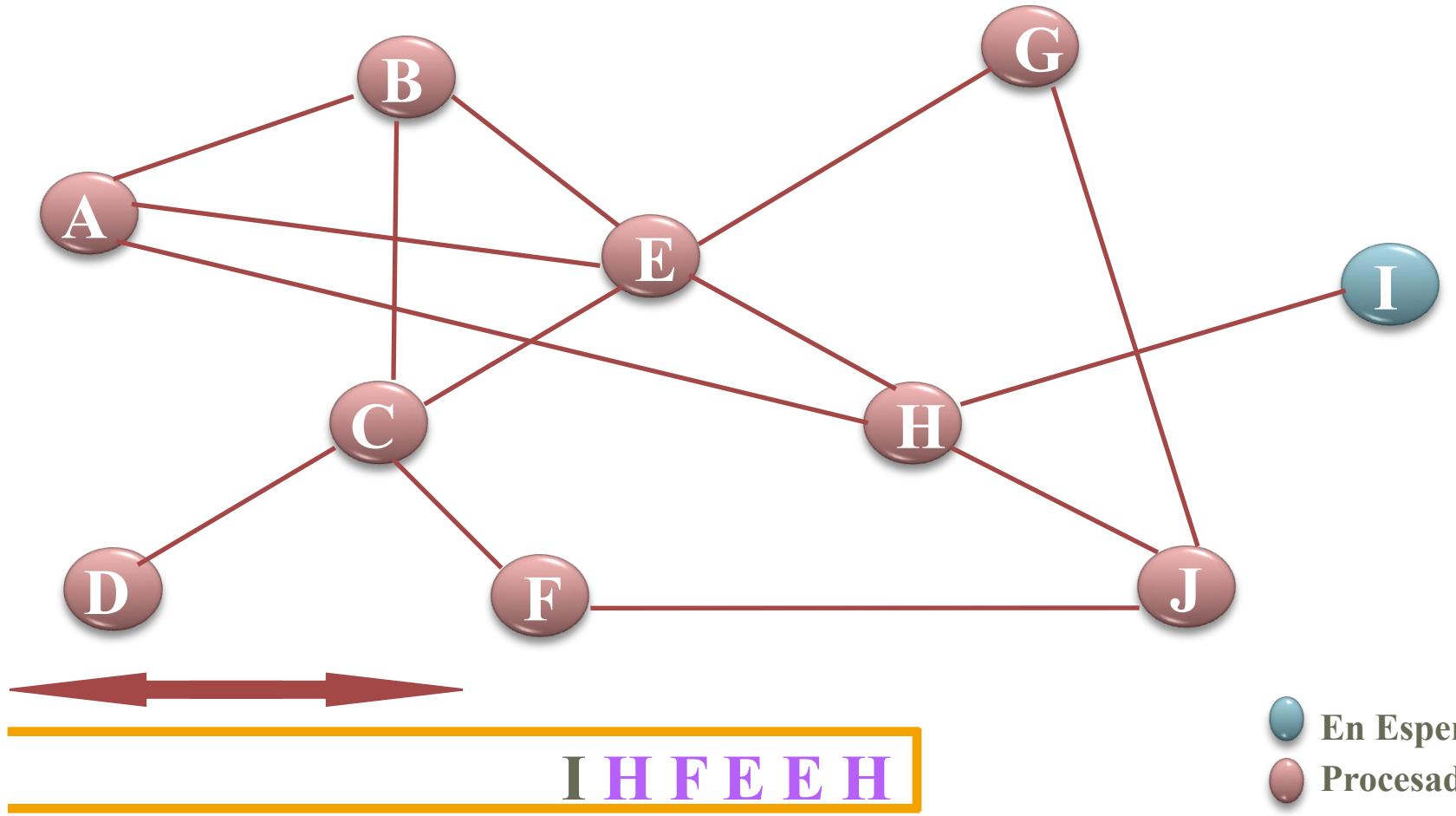
Desplegado: A B C D E G J

Ejemplo... Depth First



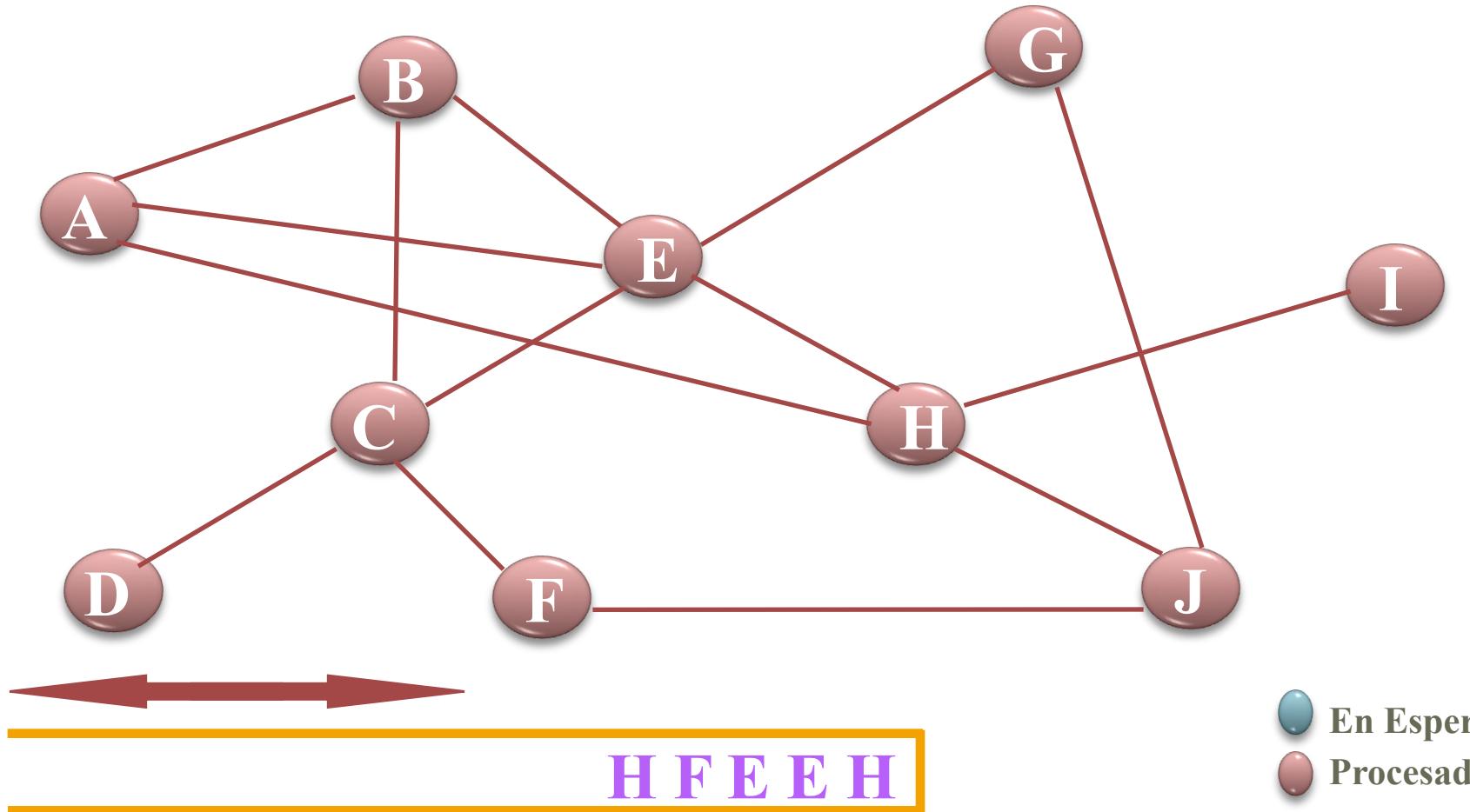
Desplegado: A B C D E G J F

Ejemplo... Depth First



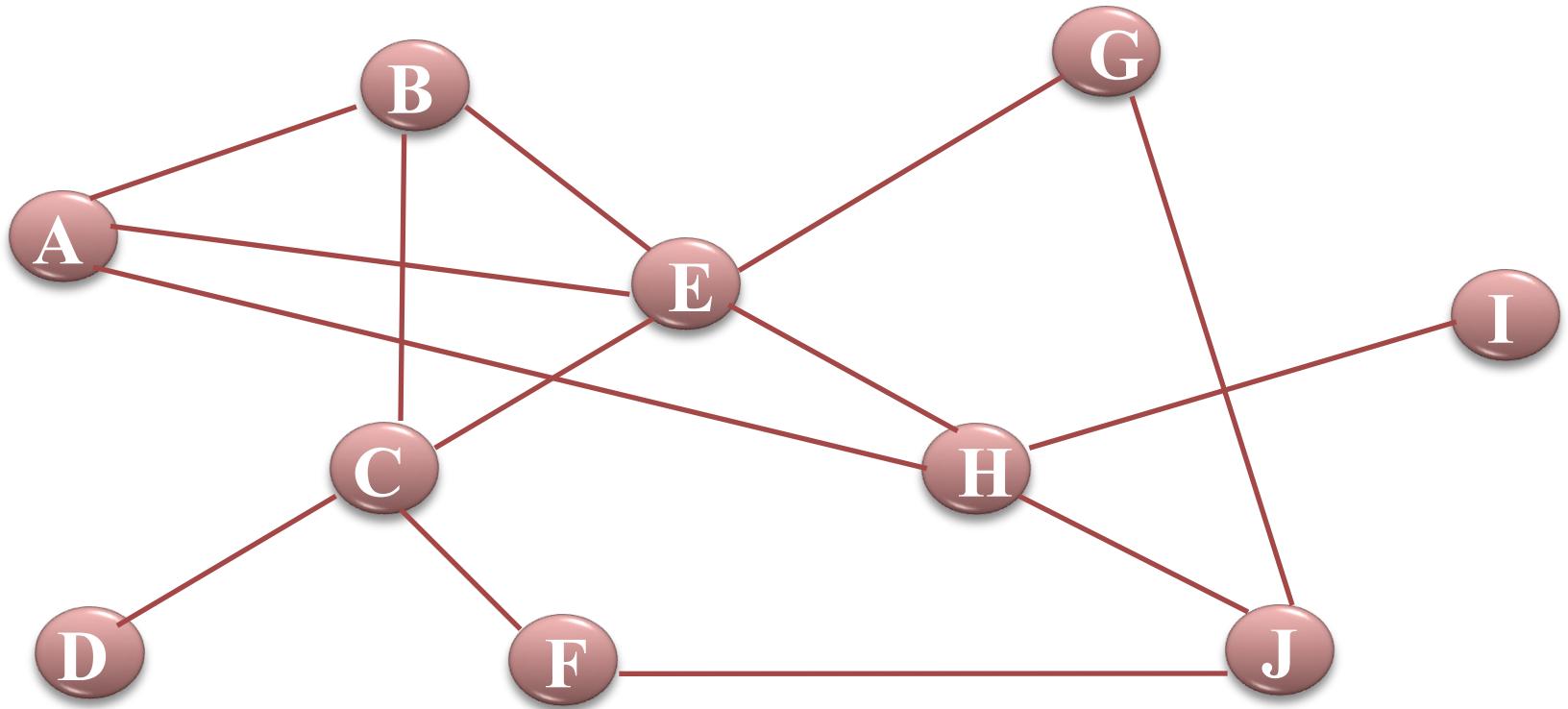
Desplegado: A B C D E G J F H

Ejemplo... Depth First



Desplegado: A B C D E G J F H I

Ejemplo... Depth First



Desplegado: A B C D E G J F H I

Aplicaciones de Grafos

- ▶ Árbol de Extensión Mínima (Minimum Spanning Tree).
Obtiene el costo mínimo para conectar a todos los Nodos del Grafo. Procurando tener la menor cantidad posible de Arcos.
- ▶ Árbol del Camino más Corto (Shortest Path).
Calcula la trayectoria más corta que puede existir de un Nodo a cualquier otro Nodo del Grafo.

Estas dos aplicaciones se utilizan para Grafos Ponderados, principalmente.

Otra Aplicación de Grafos...

- ▶ **CERRADURA TRANSITIVA** (Transitive Closure).
 - Se aplica, sobre Grafos Dirigidos (Digrafos).
 - Determina si existe o no una trayectoria entre dos Nodos cualesquiera del Grafo.
 - Es una operación importante en problemas de Conectividad y Ruteo.