

Universidad Nacional Autónoma de  
México

FACULTAD DE CIENCIAS

ANÁLISIS DE ALGORITMOS

*Práctica 04*  
*Componentes conexas*

*María de Luz Gasca Soto*  
*Teresa Becerril Torres*  
*Rodrigo Fernando Velázquez Cruz*

Autor:

*Hermes Alberto Delgado Díaz*  
*319258613*



01 de diciembre del 2024

## Lenguaje de programación utilizado

Para esta práctica se utilizó Python en su versión **Python 3.12.6**

## Comandos

Para ejecutar el programa, dentro de la carpeta src:

```
?- Python componentesConexas.py
```

## Ejemplo de ejecución

```
?- Python componentesConexas.py
Gráfica:
1: 7, 3
2: 10, 9
3: 5, 1
4: 9
5: 3
6: 11
7: 1
8: 13, 12
9: 2, 4
10: 2
11: 6
12: 13, 8
13: 12, 8
En la gráfica anterior se encuentran las siguientes
componentes conexas:
[1, 7, 3, 5]
[2, 10, 9, 4]
[6, 11]
[8, 13, 12]
```

El programa indica que en la gráfica descrita se encuentran 4 componentes conexas.

## Solución

Se utilizo el algoritmo DFS con una modificación

Función **DFS**(*Grafica, vértice, conjunto\_visitados*):

1. **Añadir** vértice a visitados
2. **Crear** lista de componente con vértice
3. Para cada vecino del vértice:
  - Si vecino no está en visitados:
    - **Llamar** recursivamente DFS
    - **Añadir** resultado a la componente
4. **Devolver** componente

El algoritmo busca una componente conexa iniciando en un vértice, al terminar devuelve la componente.

Después se utilizo un algoritmo para encontrar componentes conexas dentro de la gráfica.

Algoritmo EncontrarComponentes(Gráfica):

1. **Crear** un conjunto de vértices visitados
2. **Crear** una lista de componentes conexas
3. Para cada vértice no visitado:
  - a. **Iniciar** DFS desde ese vértice
  - b. **Explorar** todos los vértices conectados
  - c. **Formar** una componente conexa
  - d. **Marcar** vértices como visitados
  - e. **Agregar** componente conexa a la lista de componente conexas.
4. **Devolver** lista de componentes