

# CI-2693: Laboratorio de Algoritmos y Estructuras 3

29 Mayo 2024

## Objetivos

El objetivo de este laboratorio es que los estudiantes apliquen los algoritmos de Roy-Warshall, componentes de grafos, y puntos de articulación.

## Instrucciones Generales

- La solución debe ser subida a GitHub a un repositorio privado y debe contener un `README.md` con una breve descripción.
- Deben escribir al menos un caso de prueba por cada algoritmo (tests unitarios). Referencia
- El código debe estar comentado con docstrings y especificar el tipo de salida y entrada en los argumentos de cada función.

## Ejercicio 1: Cálculo de la Matriz de Alcanzabilidad

Implemente el algoritmo de Roy-Warshall para calcular la matriz de alcanzabilidad de un grafo dirigido. Utilice la versión 1 y la versión 2 del algoritmo y compare los resultados.

- **Entrada:** Una matriz de adyacencia  $M$  de un grafo dirigido.
- **Salida:** La matriz de alcanzabilidad  $M^*$ .
- **Requerimientos:**

- Implemente ambas versiones del algoritmo de Roy-Warshall.
- Compare y verifique que ambas versiones producen el mismo resultado.

## Ejercicio 2: Componentes Conexas

Utilizando la matriz de alcanzabilidad obtenida en el ejercicio 1, implemente una función que determine las componentes conexas de un grafo no dirigido.

- **Entrada:** Una matriz de adyacencia  $M$  de un grafo no dirigido.
- **Salida:** Un conjunto de componentes conexas.
- **Requerimientos:**
  - Utilice el algoritmo de Roy-Warshall para obtener la matriz de alcanzabilidad.
  - Implemente la función `componentes_conexas(M)`.
  - Pruebe su implementación con grafos de prueba.

## Ejercicio 3: Puntos de Articulación

Implemente una función que encuentre los puntos de articulación en un grafo no dirigido. Un punto de articulación es un vértice que, al ser removido, aumenta el número de componentes conexas del grafo.

- **Entrada:** Una matriz de adyacencia  $M$  de un grafo no dirigido.
- **Salida:** Una lista de puntos de articulación.
- **Requerimientos:**
  - Implemente la función `find_articulation_points_naive(graph)`.
  - Pruebe su implementación con diferentes grafos.

## Ejercicio 4: Integración y Análisis de Redes

Con los algoritmos implementados, realice un análisis completo de una red social simulada.

- **Entrada:** Una matriz de adyacencia  $M$  que representa una red social.
- **Salida:** Un informe que incluya:
  - La matriz de alcanzabilidad.
  - Las componentes conexas y fuertemente conexas.
  - Los puntos de articulación.
  - Pueden referir <https://docs.python.org/3/tutorial/inputoutput.html> para la generación del informe.