

MÉTODO DE LA INGENIERÍA

Sara Ortiz Drada (A00302324), Jose A. Galvis Nieto (A00302328), Nicolas Biojo Bermeo (A00137580)

FASE 1: IDENTIFICACIÓN DEL PROBLEMA

- **Descripción del contexto problemático (causas y síntomas)**

Caracterizan brevemente los aspectos relevantes.

Los hechos deben ser debidamente explicados e ilustrados en términos de datos, cifras, lugar, tiempo y fuente.

Identificar los síntomas señalando las variables que configuran el problema (lo que es, lo que ocurre). Se pueden hacer referencias a problemas similares en diferentes ámbitos.

Describir brevemente antecedentes realizadas en el área temática y enfoque respectivo.

- **Identificación del problema**

Síntesis del contexto situacional del problema. Se debe expresar de forma concreta en términos de una oración. (Máximo 3 líneas)

- **Requerimientos funcionales y no funcionales**

El programa debe estar en la capacidad de:

RF1- Identificar...

RF2- Generar...

RNF1-

Características de funcionamiento (rendimiento, disponibilidad, seguridad, etc.)

FASE 2: RECOPIACIÓN DE LA INFORMACIÓN NECESARIA

- **Marco teórico**

FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS

- Descripción técnica de generación de ideas

FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES

- Descarte de ideas no factibles

Se descartaron las siguientes ideas:

Idea no factible Justificación

<i>Idea 1</i>	Justificación 1
<i>Idea 2</i>	Justificación 2

- Diseños preliminares

Pseudocódigos de los algoritmos.

Análisis temporal y espacial de algoritmos.

Bosquejo de la interfaz.

Diseño (borrador) del diagrama de clases.

FASE 5: EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN

- Criterios de evaluación

	5	4	3	2	1
Complejidad temporal de la implementación de cada método.	El 100% de sus métodos presenta complejidad constante de $O(1)$	El 80% o menos de sus métodos presenta complejidad de $O(\log n)$ y el resto igual a $O(1)$	El 50% o menos de sus métodos presenta complejidad de $O(n)$ y el resto $\leq O(\log n)$	El 50% o menos de sus métodos presentan complejidad de $O(n \log n)$ y el resto $\leq O(n)$	Alguno de sus métodos presenta complejidad $\geq O(n^2)$

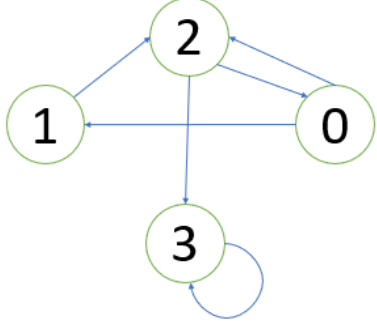
- **Evaluación según criterios**

	Complejidad temporal de la implementación de cada método.	Complejidad espacial de la implementación de cada método.	Restricciones de capacidad de la estructura de datos utilizada.	Facilidad de la implementación.	Nivel de aprendizaje en la implementación	TOTAL
	15%	15%	25%	15%	30%	100%
Arreglos	5	4	1	4	3	3,1

FASE 6: PREPARACIÓN DE INFORME Y ESPECIFICACIONES

- **Diseño del diagrama de clases de la solución**
- **Diseño del diagrama de objetos**
- **Diseño de casos de las pruebas unitarias**

Objetivo: Probar que el método bfs() funcione correctamente para diferentes casos de prueba y para tipos de grafos dirigidos y no dirigidos.				
Clase: Recorrido		Método: bfs(IGrafo<N> grafo, N s): ArrayList<N>		
Caso #	Descripción de la prueba	Escenario	Valores de entrada	Resultado
1	Dado un grafo dirigido, verifica que el algoritmo realiza el recorrido por anchura de forma correcta y retorna la lista de los vértices en el orden esperado. (caso base)	Escenario 9	grafo = new GrafoLista<Integer>(true); s = 2	El algoritmo devuelve una lista con los vértices en el orden esperado: 2,0,3,1

Escenario	Representación
9	 <pre> graph TD 2((2)) --> 1((1)) 2((2)) --> 0((0)) 1((1)) --> 0((0)) 0((0)) --> 3((3)) 2((2)) --> 3((3)) 3((3)) --> 3((3)) </pre>

FASE 7: IMPLEMENTACIÓN

La implementación de la solución se encuentra en el siguiente repositorio de github:

- **Bibliografía**

[1] Cormen, T., Leiserson, C., Rivest, R., & Stein, C. Introduction to algorithms.