# Reporte Semanal

Ramirez Moreno
Mauricio Damian

## 1. INTRODUCCIÓN Y OBJETIVOS

### Objetivos Logrados

- Desarrollo de BFS en C# (búsqueda por niveles).
- Implementación de DFS tanto en versión recursiva como iterativa.
- Evaluación de la conectividad del grafo urbano utilizado.
- Cálculo de rutas más cortas entre ciudades.
- Identificación de componentes conectadas.
- Validación completa mediante 8 pruebas unitarias (8/8 correctas).

### Red Urbana Estudiada

Ciudades incluidas: Centro, Reforma, Polanco, Roma, Condesa, Juárez, Lomas, Satélite, Doctores, Coyoacán, Narvarte, Iztapalapa, Iztacalco, Xochimilco, Interlomas, Tlalnepantla, Tláhuac, Huixquilucan y Naucalpan.

Total: 19 nodos (ciudades) y 40 aristas (conexiones bidireccionales).

## 2. ARQUITECTURA E IMPLEMENTACIÓN

Clase principal: GraphTraversal.cs — Métodos implementados

| Método | Complejidad | Descripción |
|---|---|---|
| BFS(inicio) | O(V + E) | Devuelve orden de recorrido por niveles. |
| BFSDistancias(inicio) | O(V + E) | Calcula las distancias mínimas en saltos. |
| BFSCaminoMasCorto(inicio, fin) | O(V + E) | Reconstruye la ruta mínima entre dos nodos. |
| DFSRecursivo(inicio) | O(V + E) | Recorrido profundo con recursión. |
| DFSIterativo(inicio) | O(V + E) | DFS con pila manual, ideal para grafos profundos. |
| EncontrarComponentesConectadas() | O(V + E) | Detecta subgrafos desconectados. |
| DistanciaMaxima() | $O(V^2 + V \cdot E)$ | Obtiene el diámetro del grafo. |
| EncontrarNodoCentral() | $O(V^2 + V \cdot E)$ | Determina el nodo con menor distancia promedio. |

Características destacadas

- Validación de nodos inválidos mediante excepciones.
- Reutilización del tipo Grafo<string> de trabajos previos.
- Implementación genérica que admite diferentes tipos de datos.
- Integración directa con GraphValidator (Semana 4).

## 3. RESULTADOS DE LAS PRUEBAS UNITARIAS

| Test | Descripción | Entrada | Resultado |
|---|---|---|---|
| BFS_GrafoLineal | Recorrido BFS en estructura lineal | A→B→C→D | ✓ |
| BFS_Distancias | Verificación de distancias | Grafo: A-B, A-C, B-D | ✓ |
| BFS_CaminoMasCorto | Encontrar la ruta mínima | Origen A → Destino E | ✓ |
| DFS_Recursivo | DFS recursivo visita todos | Grafo lineal | ✓ |
| DFS_Iterativo | DFS con pila | Grafo lineal | ✓ |
| ComponentesConectadas | Detección de 3 componentes | (A,B), (C,D), (E,F) | ✓ |
| BFS_NodoInexistente | Manejo de nodos inválidos | Nodo “Z” | ✓ |
| DFS_NodoInexistente | Manejo de nodos inválidos | Nodo “Z” | ✓ |

Resultado final: 8/8 pruebas completadas correctamente.

## 4. ANÁLISIS DEL GRAFO URBANO

Resumen estructural

- Vértices: 19
- Aristas: 40 (20 conexiones únicas)
- Tipo: No dirigido, ponderado
- Conectividad: 1 componente (la red es completamente conexa)

BFS desde “Centro”

Orden de visita:
Centro → Reforma, Polanco, Roma, Condesa → Juárez, Lomas, Satélite, Coyoacán, Narvarte → ...

Distancias por niveles

| Nivel | Ciudades |
|---|---|
| 0 | Centro |
| 1 | Reforma, Polanco, Roma, Condesa |
| 2 | Juárez, Lomas, Satélite, Coyoacán, Narvarte |
| 3 | Doctores, Iztapalapa, Interlomas, Tlalnepantla, Xochimilco, Iztacalco |
| 4 | Huixquilucan, Naucalpan, Tláhuac, Chalco |

Centralidad

| Ciudad | Distancia Promedio | Tipo |
|---|---|---|
| Centro | 1.89 | MÁS CENTRAL |
| Reforma | 2.05 | Muy accesible |
| Polanco | 2.42 | Accesible |
| Roma | 2.21 | Accesible |
| Juárez | 2.47 | Accesible |
| Iztapalapa | 3.16 | Periférico |
| Xochimilco | 3.58 | MÁS ALEJADO |
| Chalco | 3.79 | MÁS ALEJADO |

Conclusión: Centro es el nodo más estratégico; Xochimilco y Chalco son los más distantes.

Diámetro del grafo

- Diámetro: 4 saltos
- Pares más lejanos: Centro ↔ Chalco, Centro ↔ Tláhuac
- Toda ciudad se alcanza en máximo 4 transbordos.

5. COMPARACIÓN ENTRE BFS Y DFS

| Criterio | BFS | DFS |
|---|---|---|
| Mejor uso | Rutas mínimas | Exploración profunda |
| Garantía de camino mínimo | Sí | No |
| Orden de descubrimiento | Por niveles | Por ramas |
| Memoria | Cola de hasta 5 elementos | Pila máx. 4 elementos |
| Problema típico | "¿Qué ciudades están a 2 saltos?" | "¿Existe camino a X ciudad?" |

Orden de exploración

- BFS desde Centro:
  Centro → Reforma, Polanco, Roma, Condesa → ...
- DFS desde Centro (orden alfabético):
  Centro → Condesa → Narvarte → Iztapalapa → …

Conclusión:
BFS es preferible porque garantiza recorridos mínimos y organiza resultados por niveles.

6. APLICACIONES PRÁCTICAS

Caso 1: Planificación de rutas (BFS)

- Ruta mínima Centro → Chalco
- Ejemplo calculado:
  Centro → Reforma → Juárez → Doctores → Iztapalapa → Iztacalco → Chalco
- Complejidad: O(59) operaciones

Caso 2: Zonas de influencia

- Ciudades a ≤ 2 saltos desde Centro:
  {Centro, Reforma, Polanco, Roma, Condesa, Juárez, Lomas, Satélite, Coyoacán, Narvarte}

Caso 3: Análisis de resiliencia (DFS)

- Detección de ciclos o puntos críticos
- Permite evaluar qué cierres desconectan la red

## 7. COMPLEJIDAD COMPUTACIONAL

BFS

- Iteraciones: 19 nodos
- Vecinos promedio: ~2
- Complejidad: O(V + E) = O(59)
- Memoria: hasta 5 elementos en cola
- Tiempo: < 1 ms

DFS

- Profundidad: hasta 4
- Memoria: 4 llamadas recursivas
- Complejidad: O(V + E)
- Tiempo: < 1 ms

Resumen

- Ambos: O(V + E)
- BFS → más seguro
- DFS → menor uso de memoria

## 8. CONCLUSIONES

- Validaciones
- La red es completamente conexa.
- Centro es el nodo óptimo por su centralidad.
- Toda ciudad está a ≤ 4 saltos.
- La estructura es coherente con trabajos anteriores.
- Ambas búsquedas son eficientes.

Recomendaciones

- Para rutas mínimas → BFS
- Para análisis estructural y resiliencia → DFS
- Para instalaciones estratégicas → ubicar en Centro
- Para mejorar conectividad → reforzar enlaces en Xochimilco y Chalco

Estadísticas finales

- 8/8 pruebas unitarias aprobadas
- Integración completa de Semanas 3, 4 y 5
- Datos reales modelados con 19 ciudades
- Complejidad óptima para el tamaño del problema