



**Exposición Semana 13**

Jesús Guaygua, Valeria Molina y Luis Vásquez

Universidad de Las Américas

Programación III

Luis Aguas

1 de julio de 2025

## **Tema**

Uso de Grafos en Simulación de Tráfico Espacial

## **Introducción**

La simulación del tráfico espacial es un campo emergente dentro de la ingeniería aeroespacial y la computación. Con el aumento de satélites, estaciones y vehículos espaciales, se ha vuelto necesario organizar el tráfico en el espacio exterior con el fin de evitar colisiones entre objetos en órbita, garantizar rutas seguras, mejorar la logística y establecer una comunicación eficiente entre distintas agencias.

En este contexto, los grafos se convierten en una herramienta fundamental para modelar y analizar trayectorias, puntos de encuentro y recursos compartidos en el espacio, ya que ofrecen una forma clara de representar conexiones y trayectorias espaciales, y diseñar y simular rutas orbitales, redes de comunicación satelital y mecanismos de evasión de colisiones (López, 2023).

Además, su uso permite representar nodos (como estaciones espaciales, vehículos espaciales o satélites) y aristas (rutas entre nodos), de tal forma que facilita la toma de decisiones en entornos complejos y dinámicos.

## **Objetivos**

- Comprender cómo se representan rutas espaciales mediante grafos.
- Aplicar grafos dirigidos para simular el tráfico orbital.
- Modelar redes satelitales multicapa con grafos no dirigidos.
- Visualizar las trayectorias y conexiones mediante una interfaz gráfica en Java.

## **Conceptos clave**

- **Grafo:** estructura matemática compuesta por nodos (vértices) y conexiones (aristas).

- **Grafo dirigido:** útil para representar trayectorias con sentido, como órbitas o rutas de vuelo.
- **Algoritmos de búsqueda de caminos:** como Dijkstra o A\*, que permiten encontrar rutas óptimas entre nodos (Tomás *et al.*, 2023).
- **Modelado de trayectorias espaciales:** los nodos pueden ser satélites, estaciones espaciales o puntos orbitales; las aristas representan las rutas posibles que puede tomar una nave.

## Grafo multicapa

Este grafo muestra cómo diferentes agencias están conectadas para distintas funciones espaciales.

```
import java.util.*;

class GrafoMulticapa {
    private Map<String, List<String>> red = new HashMap<>();

    public void agregarNodo(String nombre) {
        red.putIfAbsent(nombre, new ArrayList<>());
    }

    public void conectar(String a, String b) {
        red.get(a).add(b);
        red.get(b).add(a); // conexión bidireccional
    }

    public void mostrarRed() {
        for (String nodo : red.keySet()) {
            System.out.println(nodo + " ↔ " + red.get(nodo));
        }
    }

    public static void main(String[] args) {
        GrafoMulticapa grafo = new GrafoMulticapa();
        grafo.agregarNodo("NASA-Comunicación");
        grafo.agregarNodo("ESA-Ciencia");
    }
}
```

```

        grafo.agregarNodo("JAXA-Recursos");

        grafo.conectar("NASA-Comunicación", "ESA-Ciencia");
        grafo.conectar("ESA-Ciencia", "JAXA-Recursos");

        grafo.mostrarRed();
    }
}

```

### Grafo dirigido para rutas orbitales

Este grafo simula un tráfico espacial simple entre tres órbitas.

```

import java.util.*;

class TraficoEspacial {
    Map<String, List<String>> rutas = new HashMap<>();

    public void agregarRuta(String origen, String destino) {
        rutas.putIfAbsent(origen, new ArrayList<>());
        rutas.get(origen).add(destino);
    }

    public void mostrarRutas() {
        for (String nodo : rutas.keySet()) {
            System.out.println("Desde " + nodo + " hacia: " +
rutas.get(nodo));
        }
    }

    public static void main(String[] args) {
        TraficoEspacial sim = new TraficoEspacial();
        sim.agregarRuta("Orbita A", "Orbita B");
        sim.agregarRuta("Orbita B", "Orbita C");
        sim.mostrarRutas();
    }
}

```

### Aplicaciones prácticas

- **Rutas Orbitales:** Planificación de trayectorias de satélites en diferentes niveles orbitales.

- **Evitar colisiones:** Analizando rutas cruzadas para detectar y prevenir trayectorias que se intersectan.
- **Redes de comunicación satelital:** Simulación de capas de comunicación entre satélites y estaciones.
- **Colaboración internacional:** Modelar interacciones entre agencias (NASA, ESA, JAXA, etc.).

## Conclusión

El uso de grafos en la simulación de tráfico espacial permite modelar sistemas complejos con múltiples nodos y rutas. Los grafos son herramientas versátiles para modelar el tráfico espacial, permitiendo representar trayectorias, detectar posibles colisiones y organizar redes de comunicación. Su implementación en Java con estructuras simples permite simular estos sistemas de manera educativa, abriendo paso a desarrollos más complejos con interfaces visuales e inteligencia artificial para la toma de decisiones. Además, con ayuda de algoritmos como Dijkstra, se pueden encontrar trayectorias óptimas y seguras.

Estas herramientas no solo tienen aplicaciones prácticas en la industria aeroespacial, sino también en simuladores, educación y planificación de misiones. En el futuro, su importancia crecerá con la expansión de la actividad humana en el espacio.

## Referencias

- López, S. (2023). Gestión del tráfico espacial. Una solución para la futura sostenibilidad de las actividades espaciales. <https://cedaeonline.com.ar/2023/11/16/gestion-del-trafico-espacial-una-solucion-para-la-futura-sostenibilidad-de-las-actividades-espaciales/>
- Tomás, V., Hernández, J. & Núñez, F. (2023). Propuesta para Encontrar una ruta más Corta en un Entorno de Búsqueda 2D. *Ciencia Latina Revista Científica Multidisciplinar*, 7(4), 6798-6810. [https://doi.org/10.37811/cl\\_rcm.v7i4.7440](https://doi.org/10.37811/cl_rcm.v7i4.7440)

## **Bibliografía**

Aguas Bucheli, L. F. (2024b). Estructuras de Datos para Todos: Ejercicios y Teoría con Java. Our Knowledge Publishing.

Aguas Bucheli, L. F. (2023a). Dominando la Programación Orientada a Objetos con Java en NetBeans. Our Knowledge Publishing.

Aguas Bucheli, L. F. (2023b). Manos a la Obra: Prácticas de Laboratorio en Estructuras de Datos. Our Knowledge Publishing.

Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley.

Horstmann, C. S. (2018). Core Java Volume I--Fundamentals (11th ed.). Prentice Hall.

Sierra, K., & Bates, B. (2014). Head First Java. O'Reilly Media.