

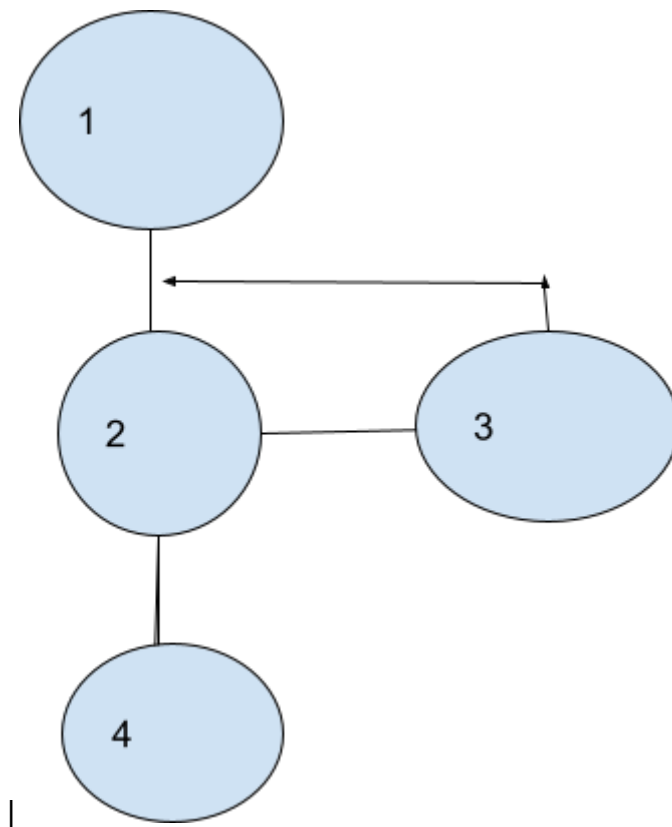
La funció "buscarAliment" té una complexitat ciclomàtica de 2

El graf té quatre nodes i quatre arestes, per la qual cosa segons la fórmula, la complexitat ciclomàtica seria:

$$M = 4 - 4 + 2(1) = 2$$

Per tant, la complexitat ciclomàtica d'aquest mètode és 2, la qual cosa significa que hi ha dos camins d'execució independents a través del codi.

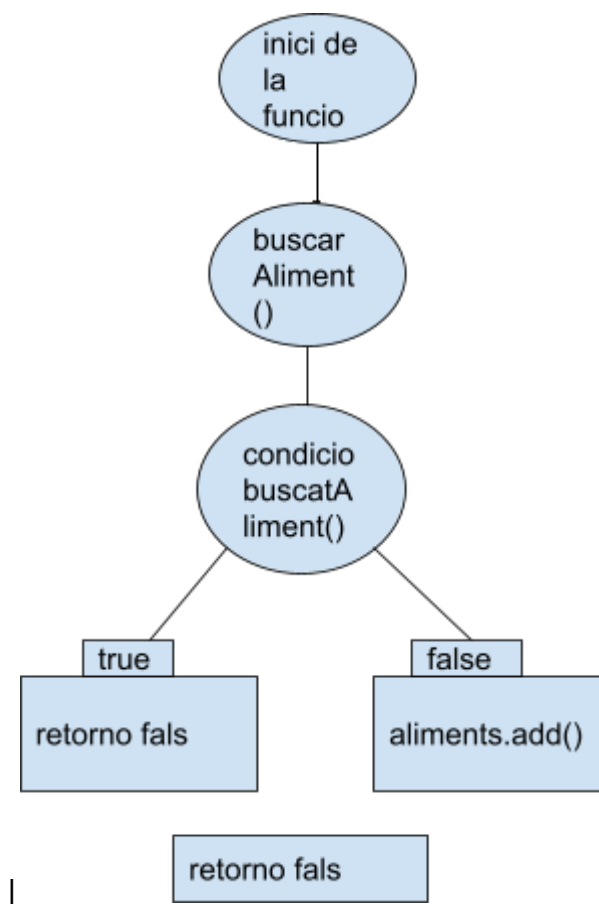
```
public int buscarAliment(String nomAliment){  
    for (int i = 0; i < aliments.size(); i++) {  
        if(aliments.get(i).getNom().equals(nomAliment)){  
            return i;  
        }  
    }  
  
    return -1;  
}
```



La funció "agregarAliment" té una complexitat ciclomàtica de 2 a causa de la presència d'una estructura de control condicional (if-else)

```
public boolean agregarAliment(Aliment aliment){  
  
    //verificar si l'aliment ja existeix  
    if(buscarAliment(aliment.getNom()) != -1)  
        return false;  
  
    alimentos.add(aliment);  
    return true;  
}
```

Aquí hi ha el graf de flux de control per a la funció:



El graf de flux de control té un total de 5 nodes i 6 arcs. En aquest cas, $E = 6$, $N = 5$ i $P = 1$. Per tant, la complexitat ciclomàtica de la funció és:

$$M = E - N + 2P = 6 - 5 + 2(1) = 2$$

Per tant, la complexitat ciclomàtica de la funció "agregarAliment" és 2.

La funció "eliminarAliment" té una complexitat ciclomàtica de 2 a causa de la presència d'una estructura de control condicional (if-else)

```
public boolean eliminarAliment(Aliment aliment)
{
    int idx = buscarAliment(aliment.getNom());

    //verificar si aliment existeix
    if(idx == -1)
        return false;

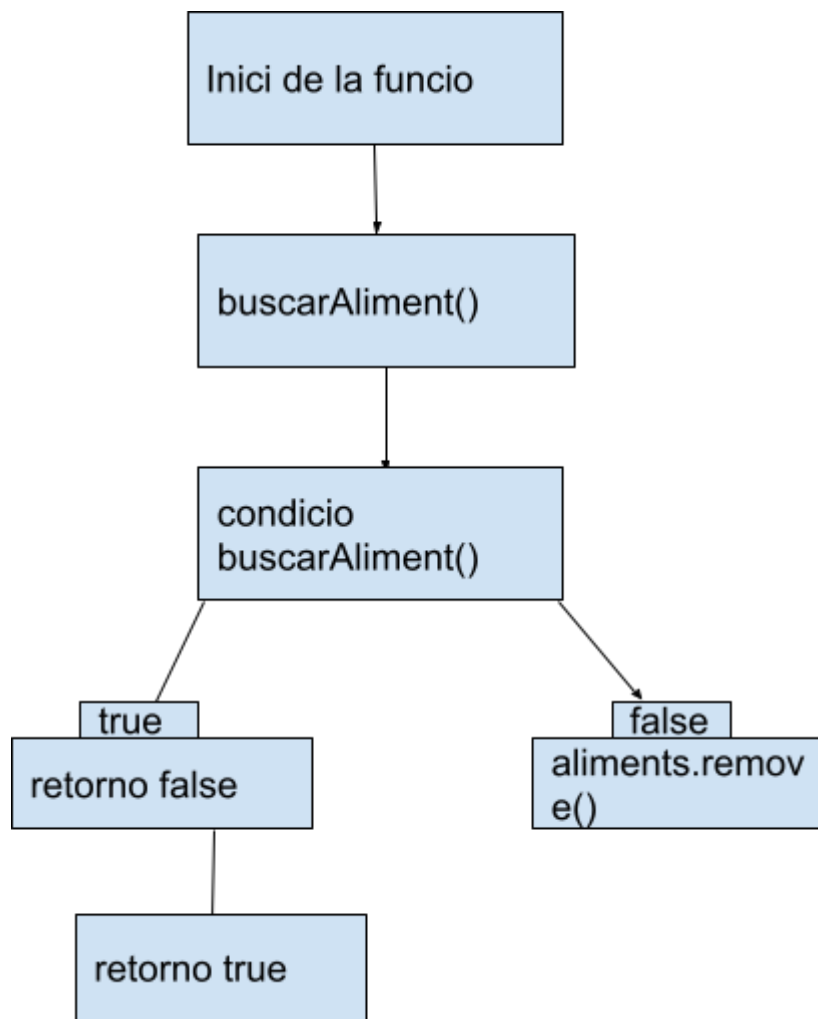
    aliments.remove(idx);
    return true;
}
```

En aquest cas, $E = 6$, $N = 5$ i $P = 1$. Per tant, la complexitat ciclomàtica de la funció és:

$$M = E - N + 2P = 6 - 5 + 2(1) = 2$$

Per tant, la complexitat ciclomàtica de la funció "eliminarAliment" és 2.

Aquí hi ha el graf de flux de control per a la funció:



La funció "modificarAliment" té una complexitat ciclomàtica de 2 a causa de la presència d'una estructura de control condicional (if-else)

```
public boolean modificarAliment(String nomAnterior, String
nomNou) {
    int idx = buscarAliment(nomAnterior);

    //verificar si aliment existeix
    if(idx == -1)
        return false;

    aliments.get(idx).setNom(nomNou);
    return true;
}
```

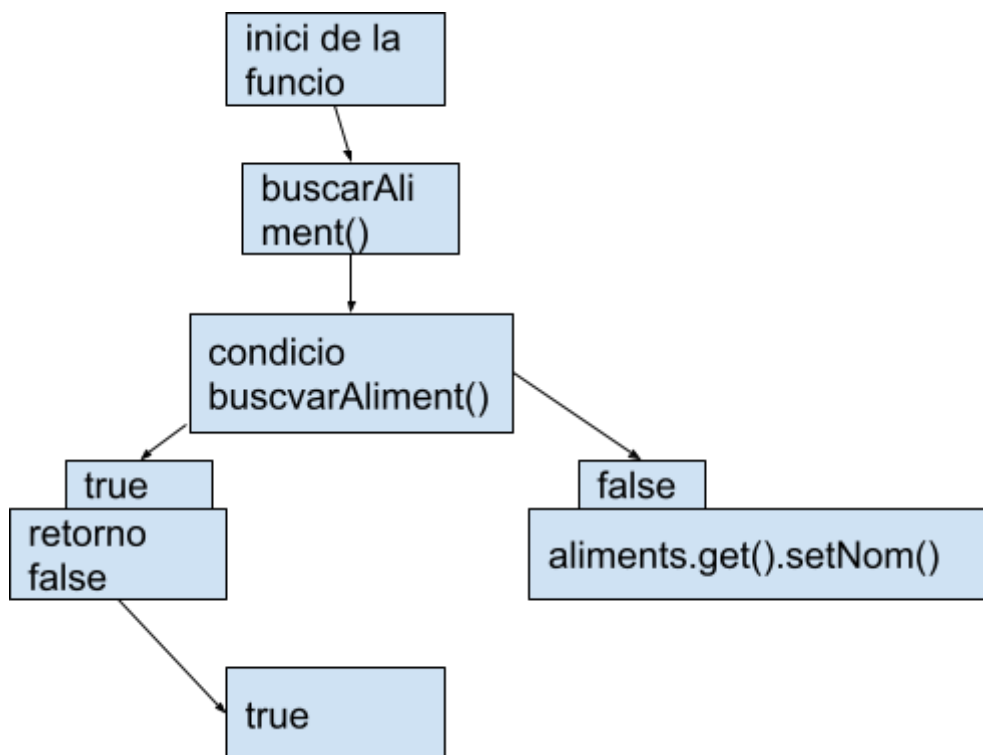
El graf de flux de control té un total de 5 nodes i 6 arcs.

En aquest cas, $E = 6$, $N = 5$ i $P = 1$. Per tant, la complexitat ciclomàtica de la funció és:

$$M = E - N + 2P = 6 - 5 + 2(1) = 2$$

Per tant, la complexitat ciclomàtica de la funció "modificarAliment" és 2.

aquí hi ha el graf de flux de control per a la funció:



La funció "mostrarAliments" té una complexitat ciclomàtica de 2 a causa de la presència d'una estructura de control condicional (if)

```
public void mostrarAliments() {  
    if(aliments.isEmpty()){  
        System.out.println("No hi ha aliments al banc  
d'aliments!");  
        return;  
    }  
    for (int i = 0; i < aliments.size(); i++) {  
        System.out.println((i+1)+" "+ aliments.get(i));  
    }  
}
```

El graf de flux de control té un total de 4 nodes i 4 arcs
En aquest cas, $E = 4$, $N = 4$ i $P = 1$. Per tant, la complexitat ciclomàtica de la funció és:

$$M = E - N + 2P = 4 - 4 + 2(1) = 2$$

Per tant, la complexitat ciclomàtica de la funció "mostrarAliments" és 2.
Aquí hi ha el graf de flux de control per a la funció:

