

Estructuras de Datos Lineales: Listas Enlazadas

Prof. Jaime Dávila
Ingeniería de Sistemas
Universidad de Nariño

Fundamentación Teórica

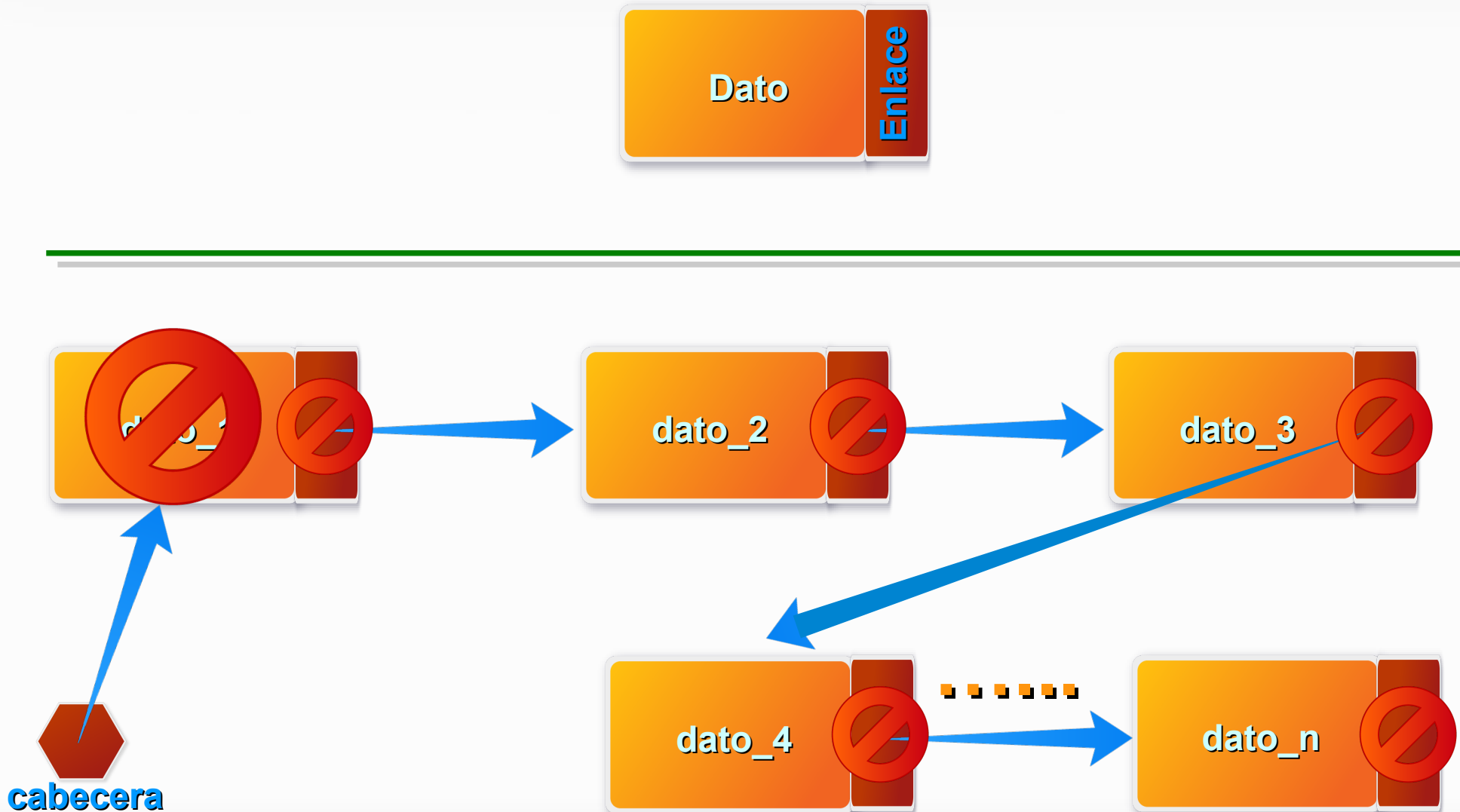
Antecedentes

- Anteriormente, la implementación de las estructuras de datos lineales de elementos **homogéneos** utilizaba **Arrays** (vector).
- Los **Arrays**, obligan a fijar por adelantado el **espacio** a ocupar en **memoria**.
- Los **Arrays** son propensos al **desbordamiento** de la **memoria** (*uso ineficiente de la memoria*).
- Las **referencias** (*apuntadores*), permiten la asignación dinámica de variables...haciendo un uso más **eficiente** de la **memoria**.

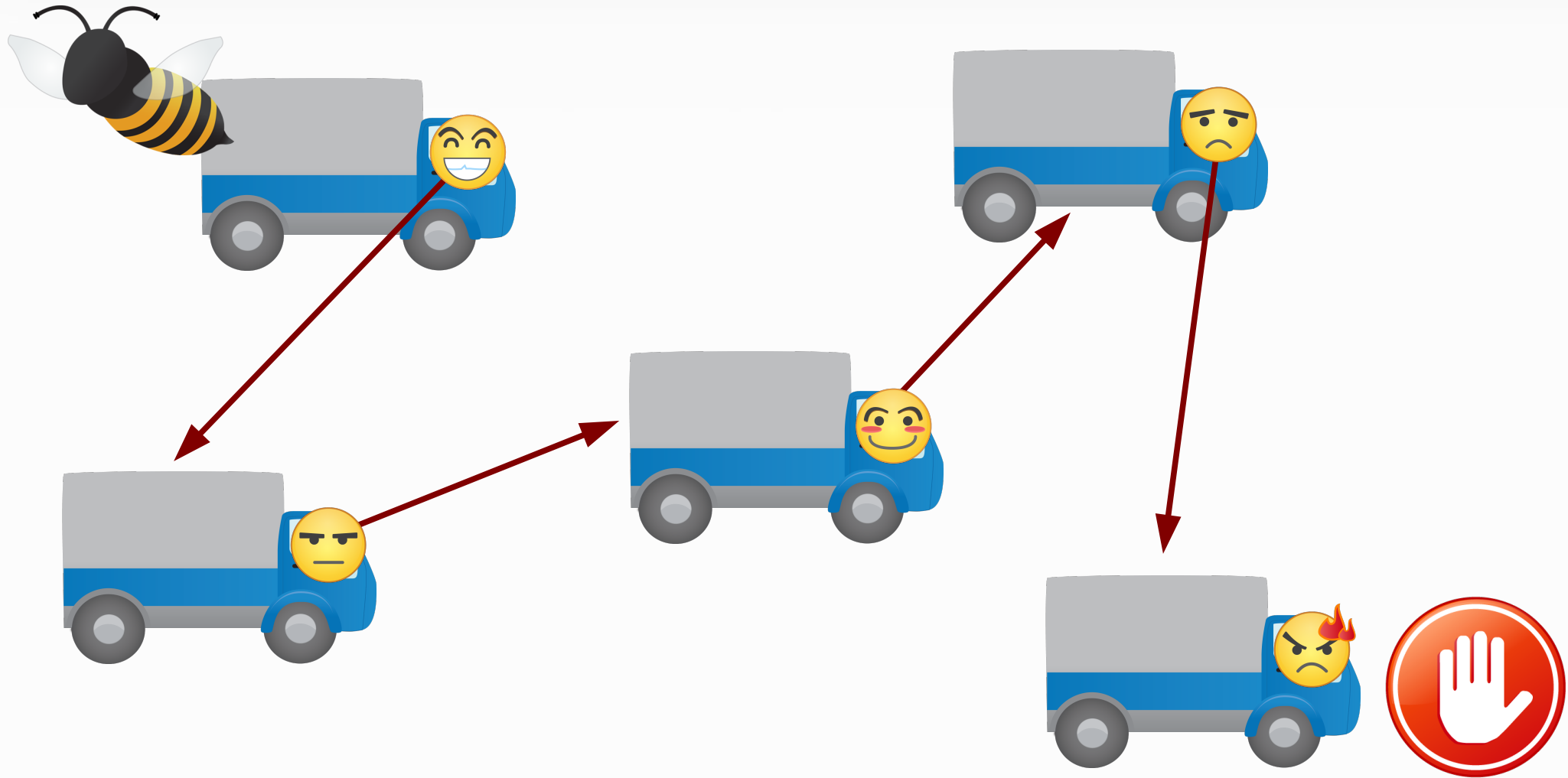
Definición

- Una **Lista Enlazada** es una **colección** o secuencia de **elementos** dispuestos uno detrás de otro, en la que cada elemento se **conecta** al siguiente elemento por un **enlace** o **referencia**.
- Los elementos de una lista se llaman **nodos**, y se componen de 2 **partes/campos**:
 - **Dato**: contiene la información y es, por consiguiente, un **valor** de un tipo genérico (denominado *dato*, *tipo_elemento*, *info*, etc.)
 - **Enlace**: es una referencia que apunta (enlaza) al siguiente elemento de la lista (denominado *sig*).

Representación Gráfica

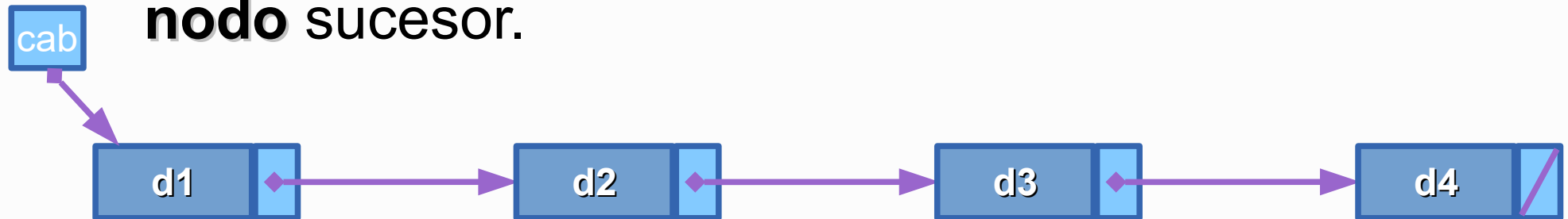


Representación Gráfica

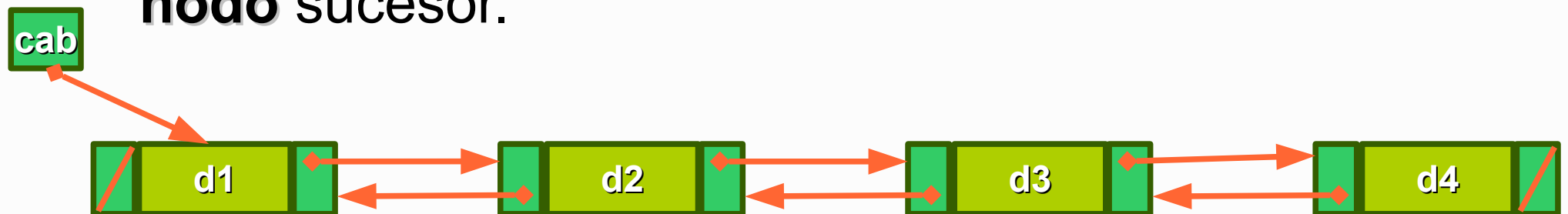


Clasificación

- Listas Simplemente Enlazadas: Cada **nodo** contiene un único enlace que lo conecta al **nodo** siguiente o **nodo** sucesor.

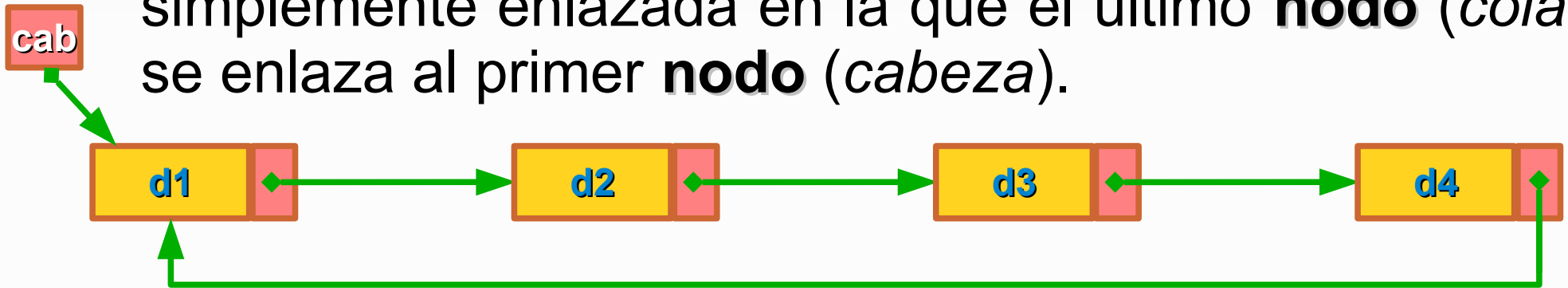


- Listas Doblemente Enlazadas: Cada **nodo** contiene dos enlaces, uno a su **nodo** predecesor y otro a su **nodo** sucesor.

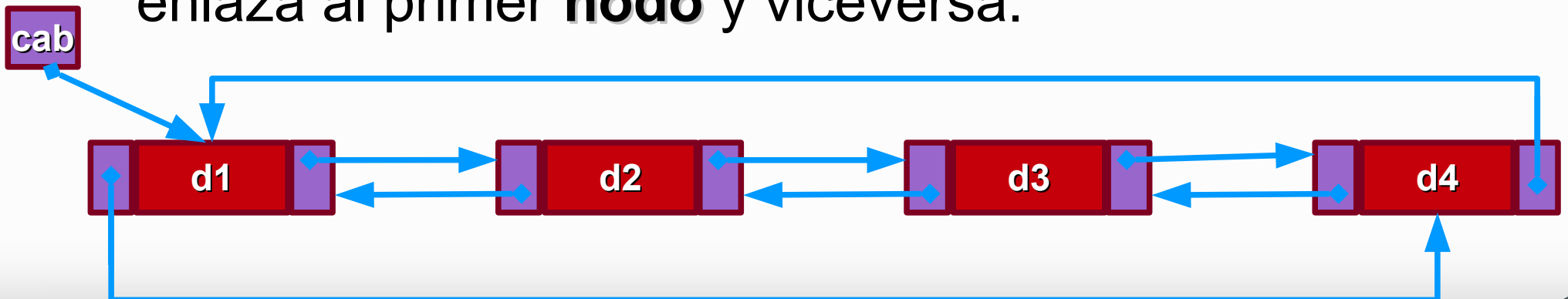


Clasificación

- Lista Circular Simplemente Enlazada: Una lista simplemente enlazada en la que el último **nodo** (cola) se enlaza al primer **nodo** (cabeza).



- Lista Circular Doblemente Enlazada: Una lista doblemente enlazada en la que el último **nodo** se enlaza al primer **nodo** y viceversa.



Operaciones

- La implementación del TAD **Lista** requiere, en primer lugar, declarar la clase **Nodo**, en la que se combinarán sus dos partes:
 - ▶ **el dato** (entero, flotante, booleano, carácter o referencias a objetos)
 - ▶ **un enlace**
- Además, la clase **Lista** con las operaciones y el *atributo* con la **cabeza** de la lista.
- Las operaciones tendrán las siguientes funciones:
 - **Inicialización** o creación (`__init__`)
 - Comprobar si la lista está **vacía** (`es_vacia`)
 - **Agregar** nodos en la lista (`final`, `posición`)
 - **Suprimir** nodos de la lista (`dato`, `posición`)
 - **Buscar** un dato en los nodos de la lista (`dato`, `posición`)
 - **Explorar** la lista enlazada (`impresión`, `__str__`, `iteraciones`)
 - **Tamaño** de la lista (`__len__`)

Implementación en Python

