



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica



# Tema 5. Tipos de Datos Lineales

Programación (PRG)

Jorge González Mollá

Departamento de Sistemas Informáticos y Computación



# Índice

## 1. Introducción

## 2. Secuencias

### 1) Recorrido y Búsqueda

### 2) Inserción y Borrado

## 3. Estructuras de Datos Lineales

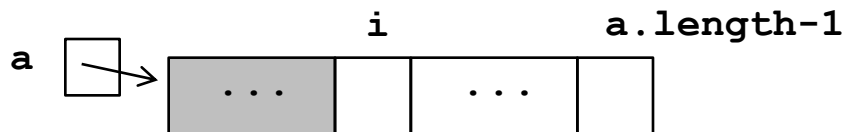
### 1) Pilas

### 2) Colas

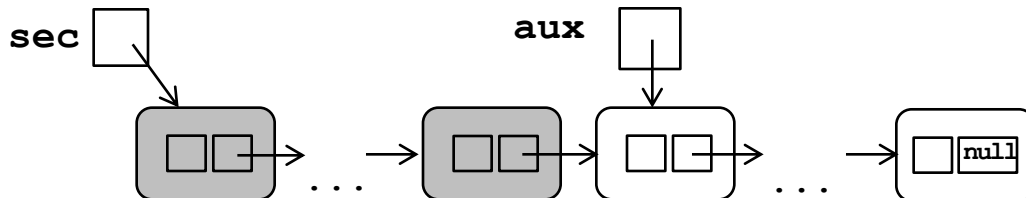
### 3) Listas con Punto de Interés

# Recorrido

- **Recorrido** de secuencias (con arrays y mediante nodos enlazados), aplicando una operación **tratar** sobre cada uno de sus elementos.



```
// recorrido de un array
int i = 0;
while (i < a.length) {
    tratar(a[i]);
    i++;
}
```



```
// recorrido de una
// secuencia enlazada
NodeInt aux = sec;
while (aux != null) {
    tratar(aux.data);
    aux = aux.next;
}
```

# Recorrido

- **Ejemplo.** Dada una secuencia implementada con nodos enlazados, se desea que sus valores saturen a un cierto valor máximo, es decir, que todos los datos mayores de máximo se sustituyan por máximo:

```
public static void saturar(NodeInt sec, int maximo) {  
    NodeInt aux = sec;  
    while (aux != null) {  
        if (aux.data > maximo) {  
            aux.data = maximo;  
        }  
        aux = aux.next;  
    }  
}
```

# Búsqueda

- **Búsqueda** en secuencias (con arrays y mediante nodos enlazados), del primero de sus elementos que satisfaga una cierta **propiedad**.

```
// búsqueda en un array
int i = 0;
while (i < a.length && !propiedad(a[i])) {
    i++;
}
if (i < a.length) { return i; }
else { return -1; }
```

```
// búsqueda en una secuencia enlazada
NodeInt aux = sec;
while (aux != null && !propiedad(aux.data)) {
    aux = aux.next;
}
return aux;
```

# Búsqueda

- **Ejemplo.** El acceso al i-ésimo elemento de una secuencia enlazada se tiene que hacer **buscando** el nodo que ocupa la i-ésima posición:

```
NodeInt aux = sec;
int k = 0;

while (aux != null && k < i) {
    aux = aux.next;
    k++;
}
```

- Inicialmente, el primer nodo de la secuencia está en la posición 0.
- La posición del nodo aux se va registrando en la variable k.
- Al acabar el bucle:
  - Si `aux != null` entonces k vale i , y aux es el i-ésimo nodo,
  - si no (`aux == null`), la búsqueda ha fracasado y, por lo tanto, dicho nodo no existe, la secuencia no tiene tantos elementos.

# Búsqueda

- **Ejemplo.** Dados una secuencia `sec` y uno de sus posibles datos `d`, se ha de devolver la posición de la primera ocurrencia de `d` en `sec`. Si `d` no formara parte de la secuencia `sec`, entonces se devuelve -1.

```
public static int buscar(NodeInt sec, int d) {  
    NodeInt aux = sec;  
    int k = 0;  
  
    while (aux != null && aux.data != d) {  
        aux = aux.next;  
        k++;  
    }  
  
    if (aux != null) { return k; }  
    else { return -1; }  
}
```