


ESTRUCTURAS DE DATOS

TIPOS ABSTRACTOS DE DATOS LINEALES

El TAD Lista

Manuel Montenegro Montes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática – Universidad Complutense de Madrid

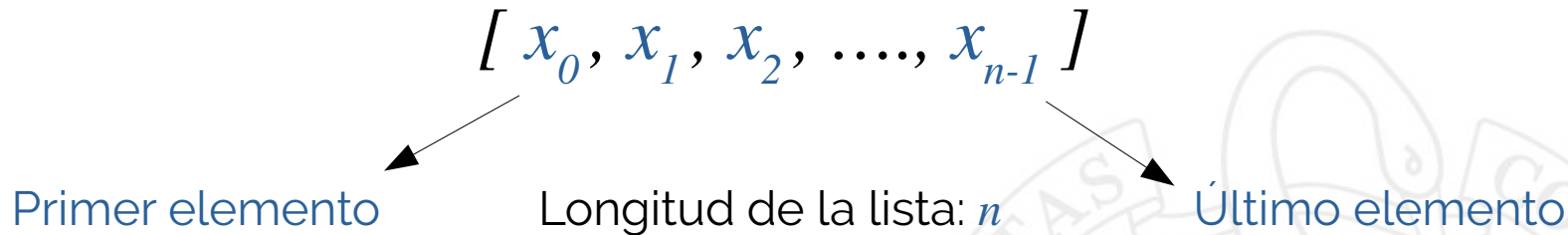


El TAD Lista

- Una **lista** es una colección de elementos, con las siguientes características:
 - Tiene longitud finita.
 - Los elementos siguen un orden secuencial:
 - Existe un primer elemento y un último elemento.
 - Todos los elementos en la lista tienen un predecesor (excepto el primero) y un sucesor (excepto el último).
 - Se permiten elementos duplicados en la lista.

Modelo del TAD Lista

- Las listas representan **secuencias** de elementos de la siguiente forma:



- Ejemplos:

$[3, 1, 6, 14, 5]$

$[1, 3, 5, 6, 14]$

$[25, 25, 7, 5, 7, 7]$

$[Marta, David, Gerardo]$ $[true]$

$[[1, 0, 0], [0, 1, 0], [0]]$

$[]$

Lista vacía (longitud 0)

Operaciones sobre el TAD Lista

- **Constructoras:**
 - Crear una lista vacía (***create_empty***).
- **Mutadoras:**
 - Añadir un elemento al principio de la lista (***push_front***).
 - Añadir un elemento al final de la lista (***push_back***).
 - Eliminar el elemento del principio de la lista (***pop_front***).
 - Eliminar el elemento del final de la lista (***pop_back***).
- **Observadoras:**
 - Obtener el tamaño de la lista (***size***).
 - Comprobar si la lista es vacía (***empty***).
 - Acceder al primer elemento de la lista (***front***).
 - Acceder al último elemento de la lista (***back***).
 - Acceder a un elemento que ocupa una posición determinada (***at***).

Operaciones constructoras

{ true }

create_empty() $\rightarrow (L: List)$

{ L = [] }



Operaciones mutadoras

$\{ L = [x_0, \dots, x_{n-1}] \}$

push_front(x : elem, L : List)

$\{ L = [x, x_0, \dots, x_{n-1}] \}$

$\{ L = [x_0, x_1, \dots, x_{n-1}], n \geq 1 \}$

pop_front(L : List)

$\{ L = [x_1, \dots, x_{n-1}] \}$

$\{ L = [x_0, \dots, x_{n-1}] \}$

push_back(x : elem, L : List)

$\{ L = [x_0, \dots, x_{n-1}, x] \}$

$\{ L = [x_0, \dots, x_{n-2}, x_{n-1}], n \geq 1 \}$

pop_back(L : List)

$\{ L = [x_0, \dots, x_{n-2}] \}$

Operaciones observadoras

$\{ L = [x_0, \dots, x_{n-1}] \}$

size($L: \text{List}$) \rightarrow (tamaño: int)

$\{ \text{tamaño} = n \}$

$\{ L = [x_0, \dots, x_{n-1}] \wedge n \geq 1 \}$

front($L: \text{List}$) $\rightarrow e: \text{elem}$

$\{ e = x_0 \}$

$\{ L = [x_0, \dots, x_{n-1}] \wedge 0 \leq i < n \}$

at($i: \text{int}, L: \text{List}$) $\rightarrow (e: \text{elem})$

$\{ e = x_i \}$

$\{ L = [x_0, \dots, x_{n-1}] \}$

empty($L: \text{List}$) $\rightarrow b: \text{bool}$

$\{ b \Leftrightarrow n = 0 \}$

$\{ L = [x_0, \dots, x_{n-1}] \wedge n \geq 1 \}$

back($L: \text{List}$) $\rightarrow e: \text{elem}$

$\{ e = x_{n-1} \}$

Operaciones adicionales

- Algunas implementaciones permiten las siguientes operaciones:
 - Mostrar una lista por pantalla.
 - Insertar/eliminar en una posición determinada.
 - Concatenar dos listas.
 - Invertir el orden de los elementos de una lista.
 - Recorrer una lista.

