



# Tipo Abstracto de Datos Lista

## T.A.D. LISTA

SISTEMAS DE DATOS  
LSI

# LISTAS

Las **listas** son estructuras de datos flexibles porque pueden crecer y contraerse, y los elementos accedidos, insertados y eliminados en cualquier posición de la lista.

# LISTA

En matemáticas, un **orden total**, **orden lineal**, **orden simple**, o simplemente **orden** en un conjunto  $X$  es una relación binaria sobre  $X$  que es antisimétrica, transitiva, y total; esto es, si se denota una tal relación por  $\leq$ , lo siguiente vale para cualesquiera  $a, b$ , y  $c$  en  $X$ :

- Si  $a \leq b$  y  $b \leq a$ , entonces  $a = b$  (antisimetría).
- Si  $a \leq b$  y  $b \leq c$ , entonces  $a \leq c$  (transitividad).
- $a \leq b$  o  $b \leq a$  (totalidad o completitud : *todo los pares de elementos son comparables bajo la relación* ).

## Ejemplos

- Las letras del alfabeto con el orden alfabético usual: " $A$ " < " $B$ " < " $C$ " < " $X$ "
- Los *naturales*, *enteros*, *racionales* y los *reales*, con el orden usual de las relaciones < o >, son conjuntos bien ordenados.

# T.A.D. LISTA – Especificación (1)

**Lista:** Secuencia de 0 o mas elementos de un tipo determinado, que puede crecer y contraerse sin restricción.

$$L = (a_1, a_2, \dots, a_n), n \geq 0$$

$n$  : longitud de la lista

Si  $n=0$ , lista vacía :  $L=( )$

Si  $n>0$ , entonces :

$a_i$  es el *i-esimo elemento*

$a_1$  es el *primer elemento*

$a_n$  es el *último elemento*

$a_i$  *precede a*  $a_{i+1}$ , para  $1 \leq i < n$

$a_i$  *sucede a*  $a_{i-1}$ , para  $1 < i \leq n$

Existe una  
**relación de orden**  
dada por la  
**posición** del  
elemento en la  
lista

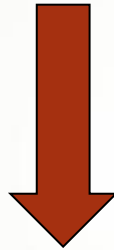
# T.A.D. LISTA – Especificación(2)

$$L = (a_1, a_2, \dots, a_i, \dots, a_n)$$

1 2 i n



¿ p ?



Ingresa el elemento X, en la lista L, en la posición p

**$1 \leq p \leq n+1$**

$$L = (a_1, a_2, \dots, X, a_i, \dots, a_n) \quad \text{si } p = i$$

1 2 i i+1 n+1

$$L = (X, a_1, a_2, \dots, a_i, \dots, a_n)$$

1 2 i+1 n+1

si p = 1

$$L = (a_1, a_2, \dots, a_i, \dots, a_n, X)$$

1 2 i n+1

si p = n+1

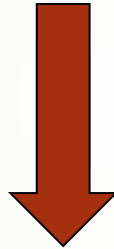
# T.A.D. LISTA – Especificación(3)

$$L = (a_1, a_2, \dots, a_i, \dots, a_n)$$

1 2 i n



¿ p ?



Eliminar de la lista L, el elemento que se encuentra en la posición p

**$1 \leq p \leq n$**

$$L = (a_1, a_2, \dots, a_{i+1}, \dots, a_n) \quad X = a_i \quad \text{si } p = i$$

1 2 i n-1

$$L = (a_2, \dots, a_i, \dots, a_n) \quad X = a_1$$

1 i-1 n-1

si p = 1

$$L = (a_1, a_2, \dots, a_{n-1}) \quad X = a_n$$

1 2 n-1

si p = n

# T.A.D. LISTA – Especificación (4)

## Operaciones Abstractas

Sean **L**: Lista; **X**: elemento y **p,p1**: posiciones

<i>NOMBRE</i>	<i>ENCABEZADO</i>	<i>FUNCION</i>	<i>ENTRADA</i>	<i>SALIDA</i>
Insertar	Insertar(X,L,p)	Ingresa el elemento X, en la lista L, en la posición p	L , X y p	$L=(a_1, \dots, a_{p-1}, X, a_{p+1}, \dots, a_n)$ o $L=(a_1, \dots, a_n, X)$ o $L=(X)$ , si $1 \leq p \leq n+1$ ; Error en caso contrario
Suprimir	Suprimir(L,p,X)	Elimina de la lista L, el elemento que se encuentra en la posición p	L y p	$L=(a_1, \dots, a_{p-1}, a_{p+1}, \dots)$ y $X=a_p$ , si $1 \leq p \leq n$ ; Error en caso contrario
Recuperar	Recuperar(L,p,X)	Recupera de la lista L, el elemento que se encuentra en la posición p	L y p	$X=a_p$ , si $L=(a_1, \dots, a_p, \dots, a_n)$ y $1 \leq p \leq n$ ; Error en caso contrario
Buscar	Buscar(X,L,p)	Localiza en la lista L, el elemento X	L y X	$p=i$ , si $L=(a_1, \dots, a_i=X, \dots, a_n)$ ; Error en caso contrario

# T.A.D. LISTA – Especificación (5)

## Operaciones Abstractas

Sean **L**: Lista; **X**: elemento y **p**: posición

<i>NOMBRE</i>	<i>ENCABEZADO</i>	<i>FUNCION</i>	<i>ENTRADA</i>	<i>SALIDA</i>
Primer_elemento	Primer_elemento (L,X)	Reporta el primer elemento de la lista L	L	$X=a_1$ , si $n>0$ ; Error en caso contrario
Ultimo_elemento	Ultimo_elemento (L,X)	Reporta el último elemento de la lista L	L	$X=a_n$ , si $n>0$ ; Error en caso contrario
Siguiente	Siguiente(L,p, <b>p<sub>1</sub></b> )	Recupera de la lista L la posición ( <b>dirección</b> ) siguiente a p	L y p	<b>p<sub>1</sub>=dirección(p+1)</b> , si $1 \leq p \leq n$ ; Error en caso contrario
Anterior	Anterior(L,p, <b>p<sub>1</sub></b> )	Recupera de la lista L la posición ( <b>dirección</b> ) anterior a p	L y p	<b>p<sub>1</sub>=dirección(p-1)</b> , si $1 < p \leq n$ ; Error en caso contrario
Recorrer	Recorrer(L)	Procesa todos los elementos de la lista L	L	Está sujeta al proceso que se realice sobre los elementos de L
Crear	Crear(L)	Inicializa L	L	$L=( )$
Vacía	Vacía(L)	Evalúa si L tiene elementos	L	Verdadero si L No tiene elementos, Falso en caso contrario.



# T.A.D. LISTA – Representación

R  
E  
P  
R  
E  
S  
E  
N  
T  
A  
C  
I  
Ó  
N

*Representación Secuencial*

*Representación Encadenada*

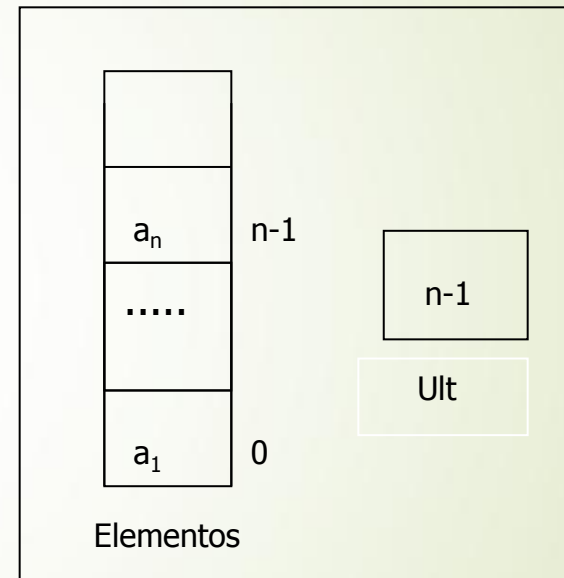
Variables dinámicas

***Cursores:*** enteros que indican posiciones en un arreglo o en otros tipos de datos

# T.A.D. LISTA – Representación

Representación secuencial

$L = (a_1, a_2, \dots, a_n)$



relación    posición “lógica”    ubicación “física”

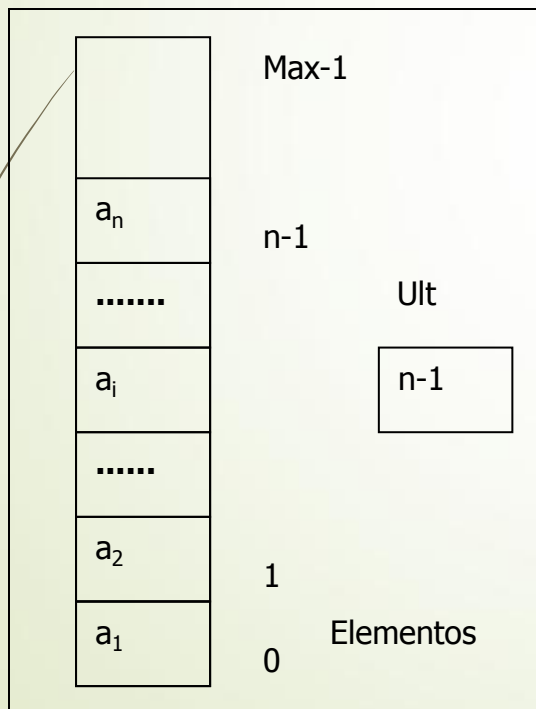
operación insertar, debe prever el desplazamiento (shifteo) de los elementos almacenados, *overflow*.

# T.A.D. LISTA –

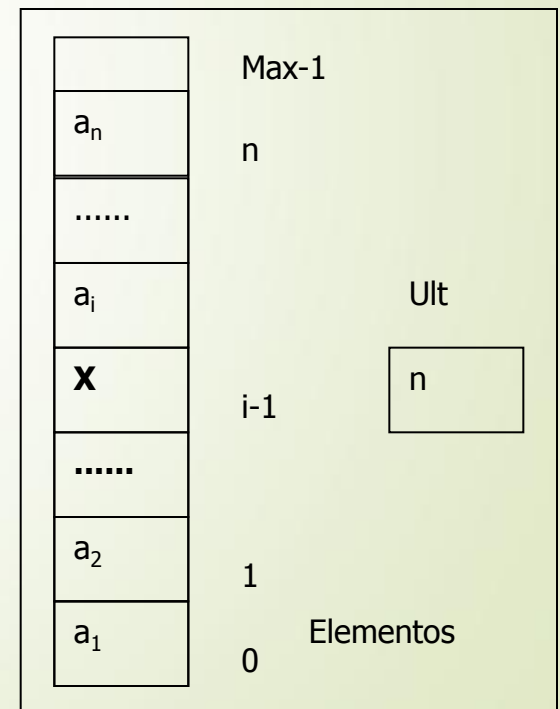
## Construcción de operaciones abstractas

**Ver Video Insertar-representacion secuencial TAD Lista.mp4**

REPRESENTACIÓN  
SECUENCIAL



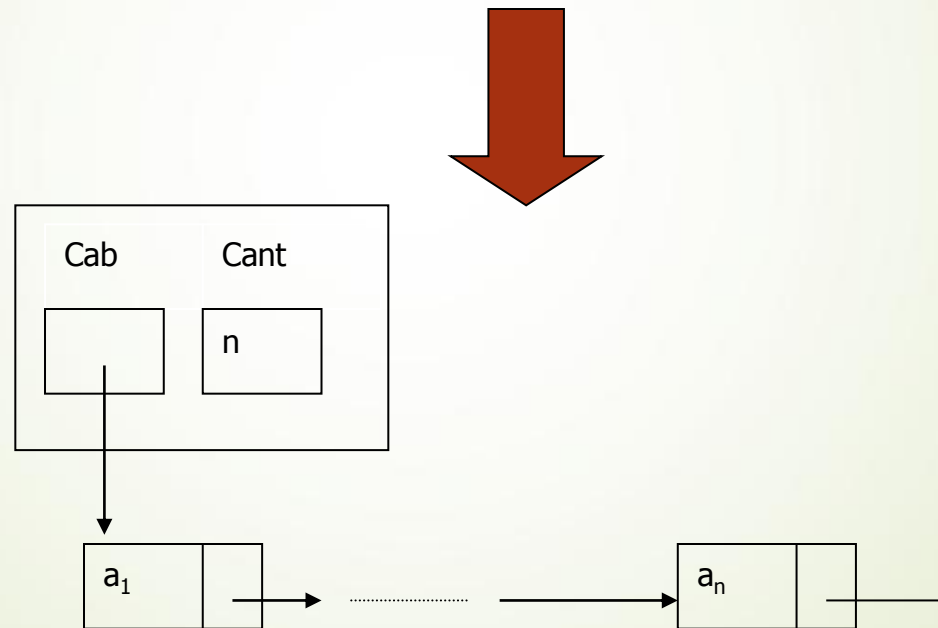
Insertar( $X, L, p=i$ )



# T.A.D. LISTA – Representación

Representación encadenada:

$L = (a_1, a_2, \dots, a_n)$

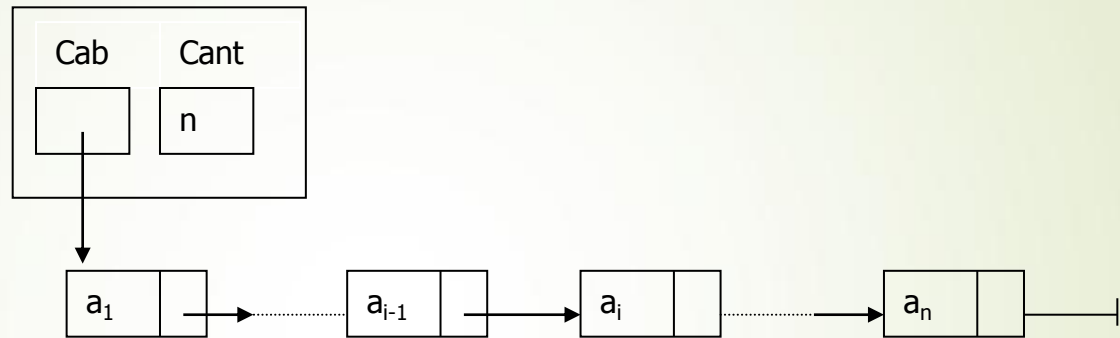


relación posición “lógica” no coincide con ubicación “física”.

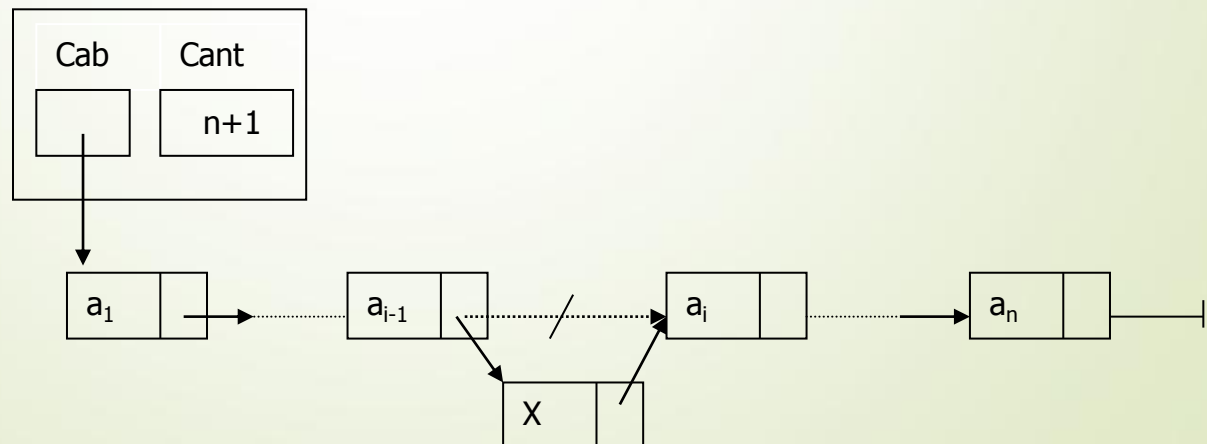
# T.A.D. LISTA

## Construcción de operaciones abstractas

REPRESENTACIÓN  
ENCADENADA



Insertar( $X, L, p=i$ )



# T.A.D. LISTA

Complete la siguiente tabla con el tiempo de ejecución de cada operación abstracta – parámetro para evaluar la eficiencia-, en cada una de las representaciones trabajadas

Representaciones Operaciones	SECUENCIAL	ENCADENADA
Insertar(X,L,p)		
Suprimir(X,L,p)		
Recuperar(L,p,X)		
Buscar(X,L,p)		
Primer_elemento (L,X)		
Ultimo_elemento (L,X)		
Siguiente(L,p,p <sub>1</sub> )		
Anterior(L,p,p <sub>1</sub> )		
Recorrer(L)		

# T.A.D. LISTA – Representación

R  
E  
P  
R  
E  
S  
E  
N  
T  
A  
C  
I  
Ó  
N

*Representación Secuencial*

*Representación Encadenada*

Variables dinámicas

***Cursores:*** enteros que indican posiciones en un arreglo o en otros tipos de datos

# T.A.D. LISTA – Representación

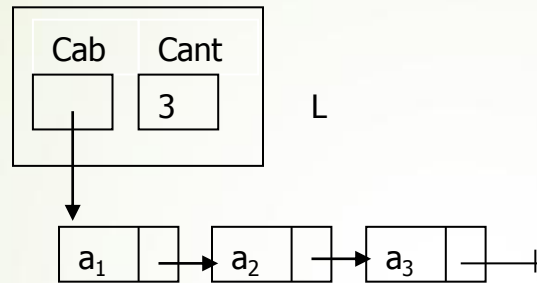
- **Cursores:** valores enteros que indican posiciones en un arreglo o en un archivo. Los cursores pueden ser usados, al igual que las variables dinámicas, para construir objetos de datos con representación vinculada.
- Para una lista con representación vinculada, cada celda es un registro con dos campos: **Elemento** y **Enlace** al siguiente elemento. Cuando trabajamos con **cursores**, **Enlace** es un valor entero.



# T.A.D. LISTA – Representación

REPRESENTACIÓN ENCADENADA

BASADA EN CURSOR



Lista L con representación enlazada

+

item	sig

0

Max-1

espacio

Espacio para cursores,  
definido sobre arreglo.

item	sig
a <sub>1</sub>	1
a <sub>2</sub>	2
a <sub>3</sub>	-1

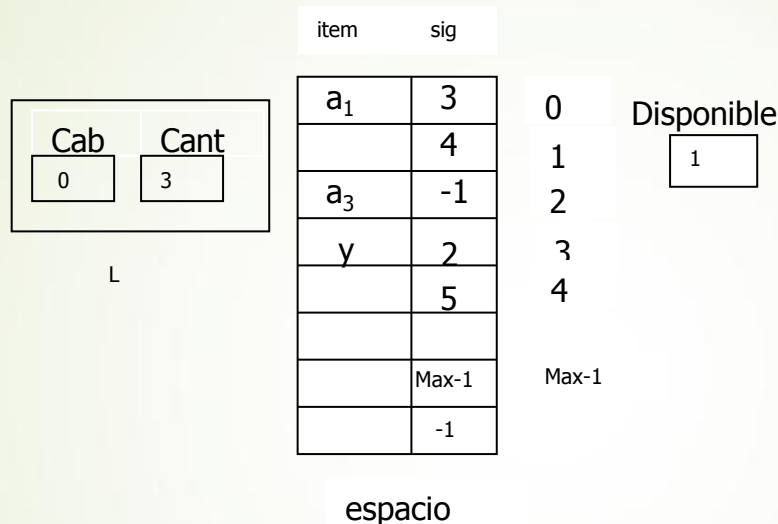
0

Max-1

espacio

Lista L representada con  
cursores, sobre arreglo

# T.A.D. LISTA – Representación

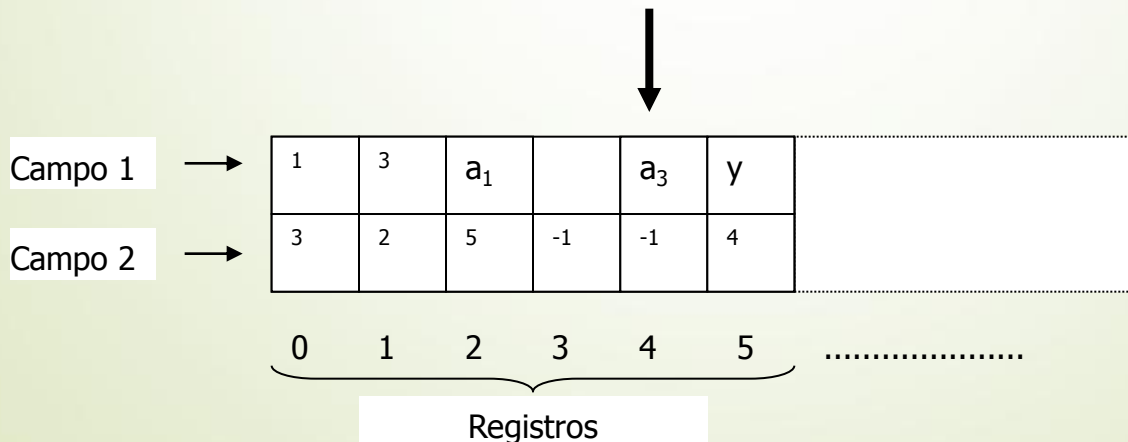


**Registro 0:** Cumple la función del campo **Disponibile**, contenido este en el *Campo 2*, pero en esta oportunidad se adiciona la cantidad de registros libres intercalados, contenida en *Campo 1*.

**Registro 1:** Este registro contiene **cab** en *Campo 2*, y **cant** en *Campo 1*, ambos correspondientes a la lista L.

**Registros 2, 4 y 5:** representan a las tres celdas de la lista **L = (a<sub>1</sub>, y, a<sub>3</sub>)**. En ellas *Campo 1*, es el recíproco de **item**, mientras que *Campo 2* lo es de **sig**.

**Registro 3:** Este es el único registro que pertenece, en esta instancia, a la pila de espacios libres.



Archivo F conteniendo la lista L y la pila de espacios libres.

# T.A.D. LISTA

## ORDENADA POR CONTENIDO

**Lista ordenada por contenido:** Lista en la que debe conservarse un orden lineal entre los valores de alguno de los atributos que conforman sus elementos.

### Insertar(X,L)

Entrada :  $L=(a_1, \dots, a_n)$   $n \geq 0$  y  $X$

Función : Insertar el elemento  $X$  manteniendo el orden lineal entre los valores de los elementos de  $L$ , esto es:

$a_i \leq a_{i+1}$ , para  $1 \leq i < n$ .

Salida:  $L=(a_1, \dots, a_i, X, a_{i+1}, \dots, a_n)$  si  $a_i \leq X < a_{i+1}$ ,  $1 \leq i < n$   
 $L=(X, a_1, \dots, a_n)$  si  $X < a_1$   
 $L=(a_1, \dots, a_n, X)$  si  $a_n < X$

Que modificación  
requiere la  
**Especificación**  
de Lista?



**¿ Suprimir ?**

## T.A.D. LISTA – Aplicación

Diseñe el algoritmo que, **apoyado en el TAD Lista**, elimine elementos con valores repetidos de una lista de números naturales.

Entrada		Salida
$L = (10, 5, 7, 5, 2, 10)$	$\rightarrow$	$L = (10, 5, 7, 2)$

**Nota:** No usar estructuras de datos adicionales, solo la lista de entrada