Licenciatura en Ciencias de la Computación.
Facultad de Ingeniería - Universidad Nacional de Cuyo

Algoritmos y Estructuras de Datos I

```
Tema = "Arreglos"
```

Dr. Carlos A. Catania
Ing. Lucia Cortes
Lic. Javier Rosenstein

TAD Secuencia de números enteros de longitud variable

Un ejemplo: Se especifica informalmente un TAD que permita representar secuencias de números reales desordenados

TAD SECUENCIA

VALORES: números reales

OPERACIONES:

```
CREATE () <- retorna una Secuencia vacía
ACCESS (Secuencia, posición)<- retorna elemento
INSERT (Secuencia, elemento, posición)
DELETE (Secuencia, elemento)
SEARCH (Secuencia, elemento)
```

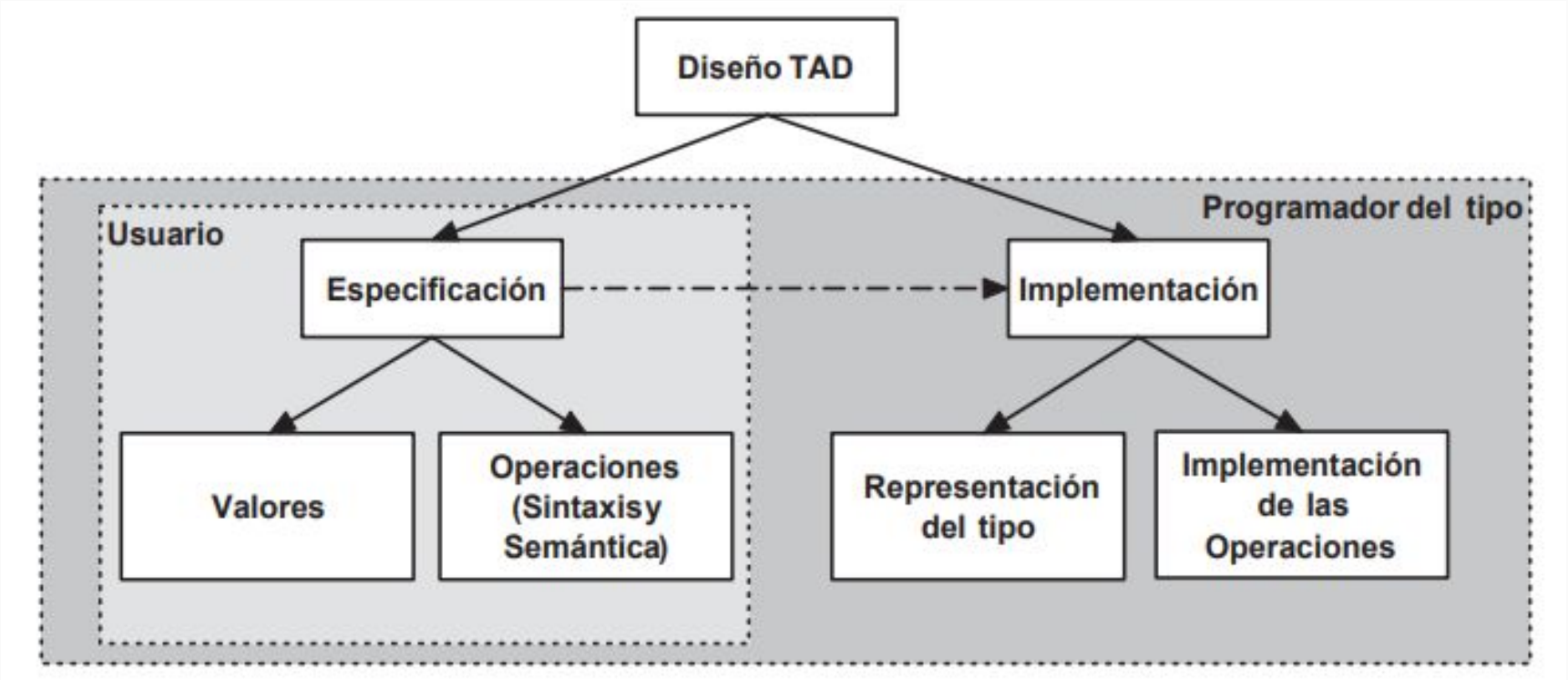
Implementación del TAD secuencia

EL **TAD** especificado anteriormente puede implementarse de diferentes maneras.

Para ello hay que:

- Elegir una **representación del tipo**.
- Implementar las **operaciones** especificadas en el TAD

Diseño TAD


Diseño TAD
Usuario
Programador del tipo
Especificación
Implementación
Valores
Operaciones (Sintaxis y Semántica)
Representación del tipo
Implementación de las Operaciones

¿QUÉ TIPO (ESTRUCTURA) DE DATO CONOCEMOS QUE PUEDE AYUDARNOS A **REPRESENTAR** EL **TAD SECUENCIA?**

Array

noun

Es una estructura de datos que permite almacenar de manera continua un conjunto **de longitud fija** de elementos del **mismo tipo de datos.**

WARNING!!!

El arreglo **NO ES LA ÚNICA ESTRUCTURA** que podemos utilizar para representar el TAD secuencia

TAD **Secuencia** representado con Arreglos (array)

El TAD Secuencia se puede representar con variable de tipo arreglo (**`array`**) unidimensional de reales (**`float`**) que tiene como máximo 6 elementos.

Ej: Una variable de tipo array que contenga **distancias**

	Distancias
0	212.5
1	10.1
2	8000.12
3	388999.021
4	3.141566
5	None

Indice Valores

TAD Secuencia representado con Arreglos (ARRAY): Operaciones básicas

Se implementan las siguientes operaciones básicas:

- `Access()`
- `Search()`
- `Insert()`
- `Delete()`

	Distancias
0	212.5
1	10.1
2	8000.12
3	388999.021
4	3.141566
5	Vacio

Índice Valores

Implementación de la operación Access()

`Access(Array,index)`

Accede al elemento de **`Array`** ubicado en el índice especificado por **`index`**

`Access(Distancias,2)-> 8000.12`

	Distancias
0	212.5
1	10.1
2	8000.12
3	388999.021
4	3.141566
5	Vacio

Índice Valores

Implementación de la operación Search()

`Search(Array,element)`

Busca el elemento **`element`** dentro de **`Array`**

`Search(Distancias,8000.12)-> 2`

	Distancias
0	212.5
1	10.1
2	8000.12
3	388999.021
4	3.141566
5	Vacio

Indice Valores

Implementación de la operación Insert()

`Insert(Array,element,index)`

Inserta el elemento **`element`** dentro del **`Array`** en la posición determinada por **`index.`** El resto de los elementos son desplazados

```
Insert(Distancias,1000.12,2)
```

	Distancias
0	212.5
1	10.1
2	1000.12
3	8000.12
4	388999.021
5	3.141566

Indice Valores

Delete(Array,element)

Elimina el elemento **element** dentro del **Array** El resto de los elementos son desplazados

```
delete(Distancias,8000.12)
```

	Distancias
0	212.5
1	10.1
2	1000.12
3	388999.021
4	3.141566
5	None (vacio)

Indice Valores

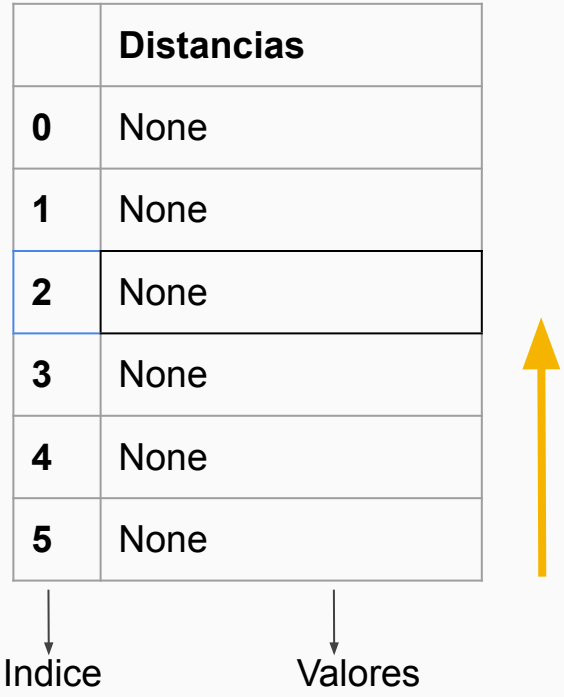
Implementación de la operación create()

`Array(size,data_type)`

Crea un arreglo de tamaño **size** del tipo **data_type**

```
Distancias<-Array(6,0.0)
```

Creamos una arreglo de longitud 6 de tipo float



Distancias2D es una variable de tipo arreglo (**`array`**) bidimensional de reales (**`float`**) que tiene como máximo 18 elementos

`Distancias2D`

	0	**1**	**2**
0	212.5	14.1	Vacio
1	10.1	20.22	212
2	8000.12	2121.	212.2
3	388999.021	2125.	2451.22
4	3.141566	888.	775.326
5	None	412.21	None

Índic

Índice

Valores

Array(size,data_type)

Crea un arreglo de tamaño **size** del tipo **data_type**

```
Distancias2d<-Array(6,Array(6,0.0))
```

Creamos una arreglo de longitud 6 de tipo de dato Array (también de longitud 6) y tipo **float**

Array: Pseudo Código en python

```
from algo import *

#Declaramos un arreglo de longitud 10 de tipo entero
a=Array(10,0) # OPERACIÓN CREATE

#Declaramos un arreglo de longitud 10 de tipo float
b=Array(10,0.0) # OPERACIÓN CREATE

#Declaramos un arreglo bidimensional de longitud 10x10 de tipo carácter
c=Array(10,Array(10,'c')) # OPERACIÓN CREATE
```

Pseudo código en python

```
1.  #qué hay en la celda 1?
2.  print (a[1]) # implementacion de access()
3.  #qué hay en la celda 1 1?
4.  print (a[1][1]) # implementacion de access()
5.  #cuál es la longitud del arreglo?
6.  #len(a)
7.  #cuál es el contenido de todo el arreglo?"
8.  print (a)
9.  #sumamos 5 a la celda 1"
    a[1]=a[1]+5
```

Titular: Dr. C.A. Catania **<harpomaxx@gmail.com>** **@harpolabs**
Adjunto: Ing. L. Cortés **<luciacortes5519@gmail.com>**
JTP: Lic. J. Rosenstein **<rosensteinjavier@gmail.com>**

HAPPY HACKING!