

# Introducción

Tipos de Datos Abstractos y Estructuras de Datos

Verónica E. Arriola-Rios

Facultad de Ciencias, UNAM

1 de marzo de 2021



# Tipos

1 Tipos

2 Bibliografía

# Temas

## 1 Tipos

- Tipos de datos
- Estructuras de datos
- Arreglos y referencias

# Tipo de datos

- Un *tipo de datos* especifica la interpretación semántica de la información introducida en la computadora, es decir, su significado y, por ende, cómo operar con ella.

## Código 1: Tipo de dato

```
1  int x = 5, y = 8;
2  int res = x + y;
3
4  String a = "Tipo_";
5  String g = "de_dato";
6  String concat = a + g;
```

# Tipos abstracto de datos (TAD)

- Un *tipo abstracto de datos* es una especificación formal algebraica de:
  - Un conjunto de datos
  - Las operaciones que pueden realizarse con ellos con:
    - 1 Las *precondiciones* que se deben cumplir para poder realizar la operación.
    - 2 Las *postcondiciones* o relaciones que se satisfarán tras haber ejecutado la operación.
- No indica cómo serán representados estos datos en ningún sistema en particular, ni los detalles de cómo se llevarán a cabo las operaciones. Por ello son **abstractos**.

## Código 2: API (Interfaz de programación de aplicaciones)

```
1 public interface Complejo {
2     /// DATOS  $z = x + yi$ 
3     // Lectura
4     Real parteEntera();           // x
5     Real parteImaginaria();      // y
6
7     // Escritura
8     Real parteEntera(Real r);
9     Real parteImaginaria(Real r);
10
11     /// OPERACIONES
12     /** Norma = raíz cuadrada de  $(x*x + y*y)$  */
13     Real norma();
14     /** conjugado( $z$ ) =  $x - yi$  */
15     Complejo conjugado();
16 }
```

# Temas

## 1 Tipos

- Tipos de datos
- Estructuras de datos
- Arreglos y referencias

# Estructuras de datos

*“Las estructuras de datos son las formas de representación interna de datos de la computadora, mediante las que se representa cualquier situación en la computadora, es decir, son los tipos de datos que maneja la máquina.*

*Por ejemplo, podemos representar a un trabajador mediante los datos nombre del empleado, número de horas trabajadas, cuota por hora, etcétera.” López Román 2011*



### Código 3: Estructura de datos

```
1 public class Real {
2     private double valor;
3     public Real(double valor) { this.valor = valor; }
4     // ...
5 }
6
7 public class Complejo {
8     /// DATOS  $z = x + yi$ 
9     private Real x;
10    private Real y;
11
12    /// OPERACIONES
13    public Complejo(Real x, Real y) { this.x = x; this.y = y; }
14    /** Norma = raíz cuadrada de  $(x*x + y*y)$  */
15    public Real norma() {
16        return new Real(Math.sqrt(x * x + y * y));
17    }
18    /** conjugado( $z$ ) =  $x - yi$  */
19    public Complejo conjugado() {
20        return new Complejo(x, -y);
21    }
22 }
```

# Estructuras de Datos como contenedores

- Las *estructuras de datos* son particularmente importantes también para el **almacenamiento** eficiente de otros datos.
- Estructuras como *listas*, *árboles* y *diccionarios* permiten almacenar objetos de otros tipos, recuperándolos de forma eficiente.
- Se definen TDAs con las políticas de almacenamiento, manejo y recuperación dentro de estas estructuras.
- Cada TDA puede ser implementado en más de una forma. Algunas variantes serán más o menos eficientes dependiendo del contexto donde se utilizará a la estructura.

# Temas

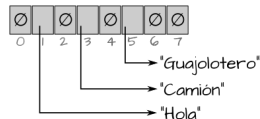
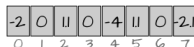
## 1 Tipos

- Tipos de datos
- Estructuras de datos
- Arreglos y referencias

# Bloques constructores

Podemos identificar dos técnicas básicas para construir estructuras que funcionen como almacén:

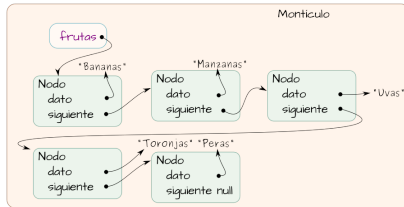
- *Arreglos*. El espacio apartado para los objetos almacenados son direcciones contiguas en la memoria.



# Bloques constructores

Podemos identificar dos técnicas básicas para construir estructuras que funcionen como almacén:

- **Arreglos**. El espacio apartado para los objetos almacenados son direcciones contiguas en la memoria.
- **Referencias**. Un objeto o **nodo** contiene la dirección de otros objetos o nodos, que contienen a los datos de interés, generando una topología que permite accederlos según es requerido.



# Bibliografía

1 Tipos

2 Bibliografía

# Bibliografía I



López Román, Leobardo (2011). *Programación estructurada y orientada a objetos. Un enfoque algorítmico*. 3.<sup>a</sup> ed. Alfaomega.

# Licencia

Creative Commons  
Atribución-No Comercial-Compartir Igual

