

# TUP - PROGRAMACION II

Tipos Abstractos de Datos

TAD

# TIPOS ABSTRACTOS DE DATOS

**Abstracción:** consiste en ocultar las características de un objeto y obviarlas, de manera que solamente utilizamos el nombre del objeto en nuestro programa.

Abstracción funcional:

- Crear procedimientos y funciones e invocarlos mediante un nombre donde se destaca que hace la función y se ignora cómo lo hace.

Abstracción de datos:

- Tipo de datos: proporcionado por los lenguajes de alto nivel. La representación usada es invisible al programador, al cual solo se le permite ver las operaciones predefinidas para cada tipo.
- Tipos definidos: por el programador que posibilitan la definición de valores de datos más cercanos al problema que se pretende resolver.

# TIPOS ABSTRACTOS DE DATOS vs ORIENTACIÓN A OBJETOS

## POO: Como contenedores

- En general, no se sabe cuántos objetos se van a necesitar para resolver un problema en particular, cuánto durarán o cómo almacenar esos objetos. ¿Cómo puede saber cuánto espacio crear si esa información no se conoce hasta el tiempo de ejecución?
- Las Arreglos y Matrices poseen este problema.
- La solución a la mayoría de estos problemas se resuelve creando otro objeto contenedor que tiene referencia a objetos internos.
- El nuevo tipo de objeto se encargará de “acomodar” todos los objetos que se coloquen dentro de él, sin saber cuántos serán. Simplemente crea un objeto contenedor y deja que se encargue de los detalles.
- Los contenedores proporcionan diferentes tipos de interfaces y comportamiento externo, además de permitir incluir operaciones que tienen diferentes eficiencias para ciertas operaciones.

# TIPOS ABSTRACTOS DE DATOS vs ORIENTACIÓN A OBJETOS

## Ejemplo con Arreglo

- Se puede crear una variable simple de tipo numérico y asignarle un valor.

```
int numero = 5;
```

- Cuando necesito trabajar con una gran cantidad de valores de un mismo tipo, se puede recurrir a un nuevo tipo denominado Arreglo.
- El Arreglo permite “agrupar” los valores de tipo numérico en una sola variable.

```
int[] arreglo = new int[10];
```

```
arreglo[0] = 5;
```

- Además, permite manipular los valores con nuevas operaciones, como la asignación o lecturas a través de un índice.
- También agrega nuevas operaciones, como la capacidad de obtener la longitud del arreglo.

```
System.out.println(arreglo.length);
```

# TIPOS ABSTRACTOS DE DATOS vs ORIENTACIÓN A OBJETOS

## TAD

- Al igual que los Arreglos, se puede crear nuevos “Tipos de Datos” con capacidades diferentes, que posean operaciones especiales y que oculten la implementación de las mismas a los Usuarios que lo quieren utilizar.
- Un Tipo de dato abstracto es un conjunto de datos u objetos al cual se le asocian operaciones.
- El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en cómo están implementadas dichas operaciones.
- Los objetos permiten encapsular valores, métodos y funciones que pueden ser usados por otras partes de un sistema mediante una clara definición o interfaz que define a los mismos.

# TIPOS ABSTRACTOS DE DATOS vs ORIENTACIÓN A OBJETOS

## Ejemplo de TAD

- Se define un nuevo “Tipo de Dato” que posee las mismas capacidades que un Arreglo, pero que posee la capacidad de crecer dinámicamente a medida que nuevos valores sean asignados.
- El nuevo tipo contendrá operaciones para agregar, actualizar y eliminar valores, así también como operaciones para recorrerlo y retorna la longitud del mismo.

```
MyTad miTad = new MyTad();  
  
miTad.agregar(5);  
  
miTad.length();  
  
miTad.recorrer();
```