

OpenCourseWare  
Grado Ingeniería Informática  
**Estructura de Datos y Algoritmos**

**Tema 1. Tipos Abstractos de Datos**

# Objetivos

---

- Comprender el concepto de abstracción.
- Comprender los conceptos de Tipo Abstracto de Datos (TAD) y estructura de datos. Explicar la diferencia entre ambos conceptos.
- Desarrollar implementaciones correctas para TAD simples.

# Índice

---

- Introducción
- Abstracción
- Tipos Abstractos de Datos
- Estructuras de Datos

# Introducción

---

- Los lenguajes de programación proporcionan **tipos primitivos de datos**:
  - **tipos simples**: no pueden ser descompuestos en partes más pequeñas. Por ejemplo: Integer, Float, String, Boolean.
  - **tipos complejos**: están formados por múltiples componentes. Por ejemplo: List, Dictionary.

# Introducción

---

- Los tipos primitivos no son suficientes para resolver problemas complejos.
- Los programadores podemos definir nuestros propios tipos de datos (**user-defined types**).

# Introducción

---

```
class CreditCard:
    """A credit card"""
    def __init__(self, customer, idCard, limit):
        """Creates a new credit card object"""
        self._customer=customer #the name of the customer
        self._idCard=idCard #the id of the credit card
        self._limit=limit #credit limit
        self._balance=0 #the initial balance is 0
```

# Índice

---

- Introducción
- **Abstracción**
- Tipos Abstractos de Datos
- Estructuras de Datos

# Definición de Abstracción (RAE)

---

## abstracción

Del lat. tardío *abstractio*, *-ōnis*.

1. f. Acción y efecto de abstraer o abstraerse.



# Definición de Abstracción (RAE)

---

**abstraer**

Conjugar

Del lat. *abstrahĕre* 'arrastrar lejos', 'apartar, separar'.

Conjug. c. *traer*.

1. **tr.** Separar por medio de una operación intelectual un rasgo o una cualidad de algo para analizarlos aisladamente o considerarlos en su pura esencia o noción. **U. t. c. intr.** *Pensar es olvidar diferencias, es generalizar, abstraer.*
2. **intr.** Hacer caso omiso de algo, o dejarlo a un lado. *Centremos la atención en lo esencial abstrayendo DE consideraciones marginales. U. t. c. prnl.*
3. **prnl.** Concentrarse en los propios pensamientos apartando los sentidos o la mente de la realidad inmediata. *Aspirando suavemente su cigarro, se abstrajo DEL ruido de la disputa.*
4. **prnl.** Retirarse o recogerse, apartándose del trato social. *Abstraerse DE los negocios mundanos.*

# Abstracción

---

- Poner el foco de atención en los detalles importantes para un problema, descartando los irrelevantes.

# Abstracción

---



## **Student**

NIA  
email  
marks  
...

## **Patient**

Birthdate  
Height  
Weight  
ListOfAllergies  
...

# Índice

---

- Introducción
- Abstracción
- **Tipos Abstractos de Datos**
- Estructuras de Datos

# ¿Qué es un Tipo Abstracto de Datos (TAD)?

---

- Es un tipo definido por el usuario que especifica un conjunto de valores y un conjunto de operaciones para manipular dichos valores.
- Un TAD debe ser independiente de su implementación.
- Un TAD se centra en **qué debe hacer** en lugar de cómo hacerlo.

# Ejemplo: TAD Número Complejo

---

Un número complejo tiene dos partes:

- la parte real (denotamos con  $a$ )
- la parte imaginaria (denotamos con  $b$ )

El valor de un número complejo es:  $a+ib$ , donde  $i=\sqrt{-1}$

# Ejemplo: TAD Número Complejo

---

Operaciones de los números complejos:

- *La negación de un número complejo  $a+ib$  es:*  
 $-a + i(-b)$
- **Suma y resta:**
  - $(a+ib) + (c+id) = (a+c) + (b+d)i$
  - $(a+ib) - (c+id) = (a-c) + (b-d)i$

# Ejemplo: TAD Número Complejo

---

Operaciones de los números complejos:

- ***multiplicación:***

$$(a+ib) * (c+id) = (ac-bd) + (ad+bc)i$$

- ***valor absoluto o módulo:***

$$|a+ib| = \sqrt{a^2+b^2}$$



# Ejemplo: TAD Fecha

---

- Una fecha representa un único día en nuestro calendario (por ejemplo, 4 de Febrero de 2021).
- Podemos definir las siguientes operaciones para operar con fechas:
  - **Date**(*day,month,year*): crea una nueva instancia de tipo fecha.
  - **day**(): devuelve el día (valor numérico) de la fecha.

## Ejemplo: TAD Fecha

---

- **month()**: devuelve el mes (valor numérico) de la fecha.
- **monthName()**: devuelve el nombre del mes de la fecha.
- **year()**: devuelve el año de la fecha.

# Ejemplo: TAD Fecha

---

- Más operaciones:
  - **numDays(otherDate)**: devuelve el número de días que hay de diferencia entre la fecha y la fecha pasada como parámetro otherDate.
  - **isLeapYear()**: devuelve True si el año de la fecha es bisiesto y False en otro caso.

# Ejemplo: TAD Fecha

---

- Más operaciones
  - **compareTo(otherDate)**: compara la fecha con la fecha otherDate para determinar su orden lógico:
    - Si la fecha es anterior a otherDate, devuelve -1.
    - Si las dos fechas son iguales, devuelve 0.
    - Si la fecha es posterior a otherDate, returns 1.
  - **str()**: devuelve un string representando la fecha en el formato *'dd/mm/yyyy'*.

# Índice

---

- 1) Introducción
- 2) Abstracción
- 3) Tipos Abstractos de Datos
- 4) **Estructuras de Datos**

# Índice

---

- Introducción
- Abstracción
- Tipos Abstractos de Datos
- **Estructuras de Datos**

# ¿Qué es una estructura de datos?

---

- Una estructura de datos es una representación (implementación) de un Tipo Abstracto de Datos (TAD) en un determinado lenguaje de programación (por ejemplo, Python).
- La estructura de datos se centra en **cómo** almacenar y organizar los datos y en implementar las operaciones para manipular los datos.

# ¿Qué es una estructura de datos?

---

- En Python, los TAD son representados por **clases**.
- Un TAD puede tener varias implementaciones (estructura de datos).



# Ejemplos

---

- Complex

# Resumen

---

- Un TAD define **qué valores y operaciones**, pero no cómo implementarlos.
- Una **estructura de datos** es la **implementación** de un TAD (**cómo**).
- Un TAD puede tener distintas implementaciones.