



TIPOS ABSTRACTOS DE DATOS T.A.D.

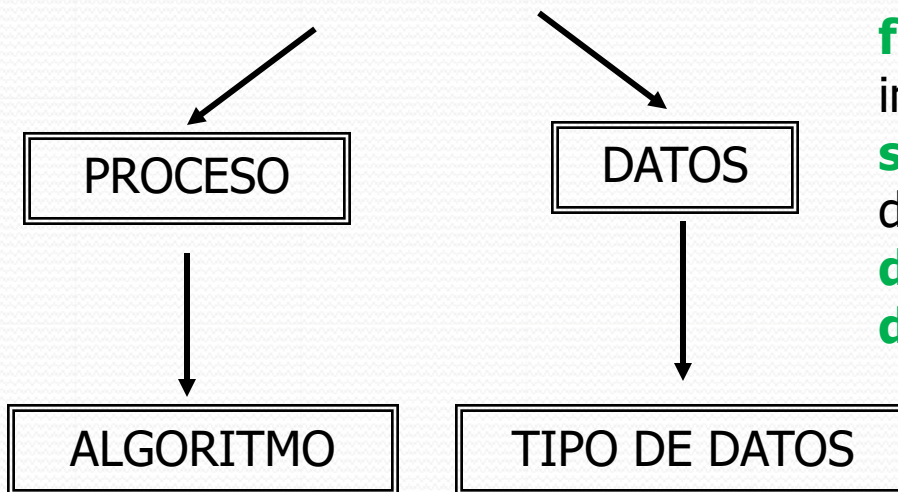
ESTRUCTURAS DE DATOS
y ALGORITMOS
LCC - LSI - TUPW

Objetivos

- Entender la abstracción de datos.
- Especificar un Tipo Abstracto de Datos-T.A.D.
- Conocer distintas alternativas de representación.

TAD Introducción

ABSTRACCIÓN



En programación, la **abstracción** es el **proceso** de ocultar detalles complejos y mostrar solo la información esencial para usar un objeto o función, permitiendo a los desarrolladores **concentrarse en la funcionalidad** sin preocuparse por la implementación interna. Se trata de **simplificar la complejidad** y reducir detalles innecesarios para **facilitar el diseño y la implementación de sistemas de software**.

Un **tipo de datos** es una clase de objetos de datos ligados a un conjunto de operaciones para crearlos y manipularlos

TAD Introducción



TIPO DE DATOS

➤ Especificación

- Atributos
- Valores
- Operaciones

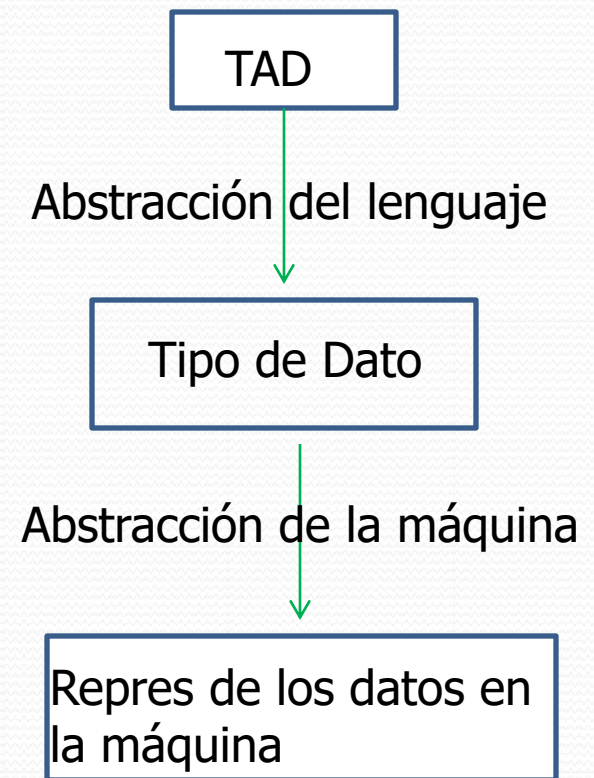
➤ Implementación

- Representación de Almacenamiento
- Algoritmos o procedimientos

TIPO ABSTRACTO DE DATOS

Tipo Abstracto de Datos – T.A.D.

Un **Tipo Abstracto de Datos (T.A.D.)** se define como un modelo matemático de objetos de datos que constituyen un tipo de datos, así como de las funciones que operan sobre ellos.



Tipo Abstracto de Datos – T.A.D.

Concepto matemático de *Entero*

Conjunto de números

Unión de $\{-1, -2, \dots, -\infty\}$ y $\{0, 1, 2, \dots, \infty\}$,

Operaciones: suma, resta, multiplicación y división entera.

Implementación del concepto matemático

Lenguajes

C *tipo de datos* **int**

Pascal *tipo de datos* **integer**.

tipos de datos provistos por un lenguaje de programación,

Tipos Abstractos de Datos para construirlos usaremos los tipos de datos disponibles en un lenguaje de programación de alto nivel.

Tipo Abstracto de Datos – T.A.D.

Componentes de un T.A.D.

Un **tipo abstracto de datos** definido por el usuario consta de:

- Un conjunto de Objetos de Datos.
- Un conjunto de Operaciones Abstractas.
- Encapsulamiento.

Tipo Abstracto de Datos – T.A.D.

Metodología para construir T.A.D.

Los tres pasos a seguir son:

- 1) Realizar la *especificación* del nuevo tipo abstracto de datos.
- 2) Seleccionar la *representación* del objeto de datos.
- 3) Construir las *operaciones abstractas* por medio de subprogramas.

Metodología para construir T.A.D.

Especificación

En esta etapa se *describe el **Modelo Matemático***, y se establecen sus atributos, valores y operaciones.

Atributos:

- Nombre del Tipo Abstracto de Datos.
- Número de componentes (fijo o variable)
- Tipo de cada componente.
- Número máximo de componentes, etc.

Valores

Operaciones:

- Nombre de la operación.
- Encabezado de la operación.
- Función
- Entrada
- Salida

Metodología para construir T.A.D.

Representación

En esta etapa se selecciona la **representación de almacenamiento** a usar para los objetos de datos.

R E P R E S E N T A C I Ó N

Representación secuencial: la estructura de datos se guarda en un solo bloque contiguo de memoria

Representación vinculada, encadenada o enlazada: la estructura de datos es mantenida en varios bloques no contiguos de almacenamiento, vinculados entre sí por medio de enlaces.

- Variables dinámicas
- Cursores

Metodología para construir T.A.D.

Construcción de operaciones abstractas

En esta etapa se precisan, en términos de **algoritmos** o procedimientos concretos, las operaciones definidas para el tipo abstracto de datos; son estos algoritmos los que manipulan la representación de almacenamiento elegida para los objetos de datos.

T.A.D. Conjunto

1) ESPECIFICACIÓN

Atributos:

- Nombre del Tipo Abstracto de Datos.
- Número de componentes
- Tipo de cada componente.
- Número máximo de componentes, etc.

Valores

Operaciones:

- Nombre de la operación.
- Encabezado de la operación
- Función
- Entrada
- Salida

T.A.D. Conjunto

Especificación

Atributos:

- Nombre del Tipo Abstracto de Datos
- Número de componentes
- Tipo de cada componente.
- Número máximo de componentes

Conjunto

Variable

**Depende de los
elementos que integran
el conjunto**

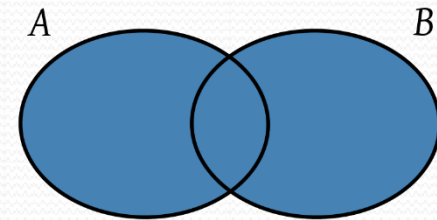
Cardinalidad(U)

T.A.D. Conjunto

Especificación

Operaciones:

- Nombre de la operación.
- Encabezado de la operación
- Función
- Entrada
- Salida



Union

Union(A,B,C)

$C = \{x / x \in A \vee x \in B\}$

A , B : Conjuntos

C: Conjunto

<i>NOMBRE</i>	<i>ENCABEZADO</i>	<i>FUNCION</i>	<i>ENTRADA</i>	<i>SALIDA</i>
Union	Union(A,B,C)	$C = \{x / x \in A \vee x \in B\}$	A , B : Conjuntos	C: Conjunto

T.A.D. Conjunto

Representación

¿Qué aspectos influyen en la elección de una representación particular?

¿Cómo pueden representarse los conjuntos?

Construcción de las operaciones abstractas

¿Qué debe considerarse al construir una operación abstracta?

Ventajas de la abstracción de datos

- Separa Especificación de Implementación.
- Programas de aplicación más simples, comprensibles y fáciles de entender
- **Reutilización:** Un mismo TAD puede ser utilizado en diferentes contextos de aplicación.
- **Ocultamiento de información:** El poder escoger entre varias alternativas de implementación es la razón principal para usar abstracción de datos. Los detalles de implementación quedan totalmente ocultos dentro de las barreras del TAD, y es posible hacer modificaciones locales, generalmente para mejorar la eficiencia, que no tengan una repercusión global.
- **Integridad:** Si se acepta que la integridad es la exactitud y totalidad de la información, los TADs, a través del encapsulamiento, permiten contar con información fidedigna tanto en datos como en resultados de procesos.

Arquitectura para desarrollo de software

