

Tipos de Datos Abstractos

ALGORITMOS Y ESTRUCTURAS DE DATOS II

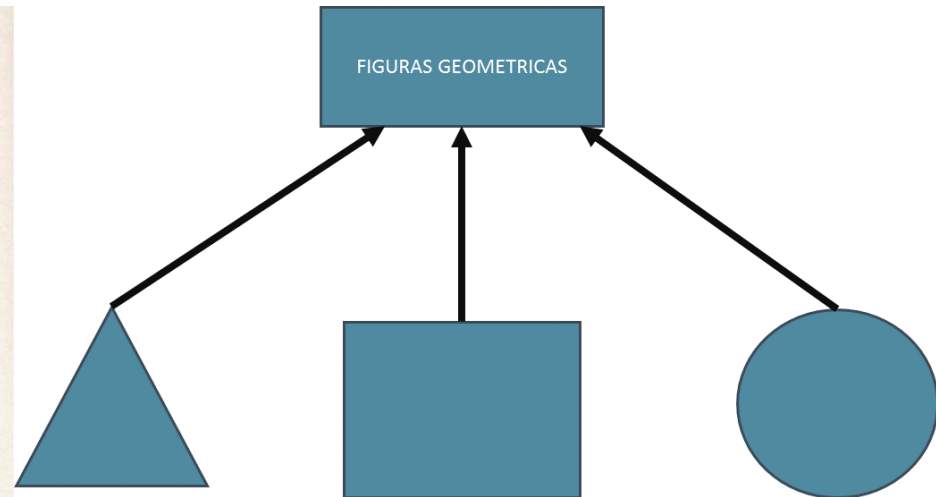
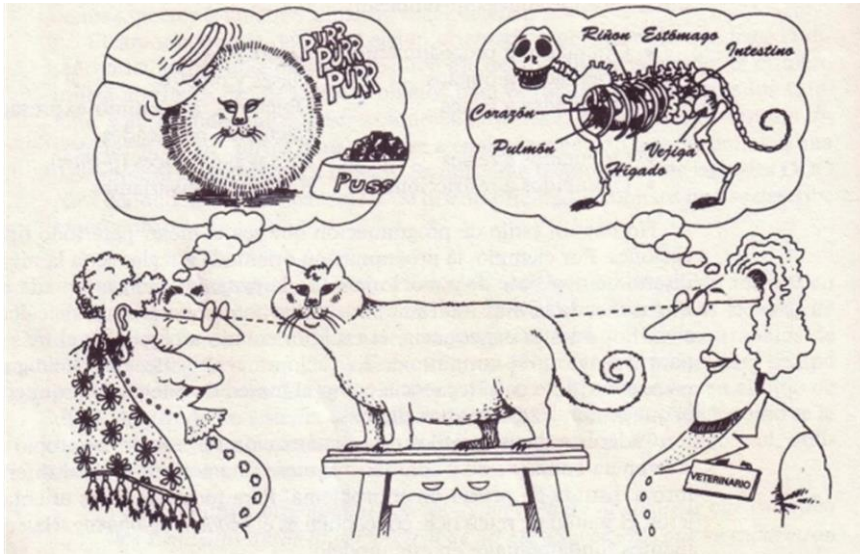
Abstracción

Abstracción → Centrar la atención en las **características relevantes al problema a resolver**, y disponerlas del modo más conveniente para su mejor comprensión.

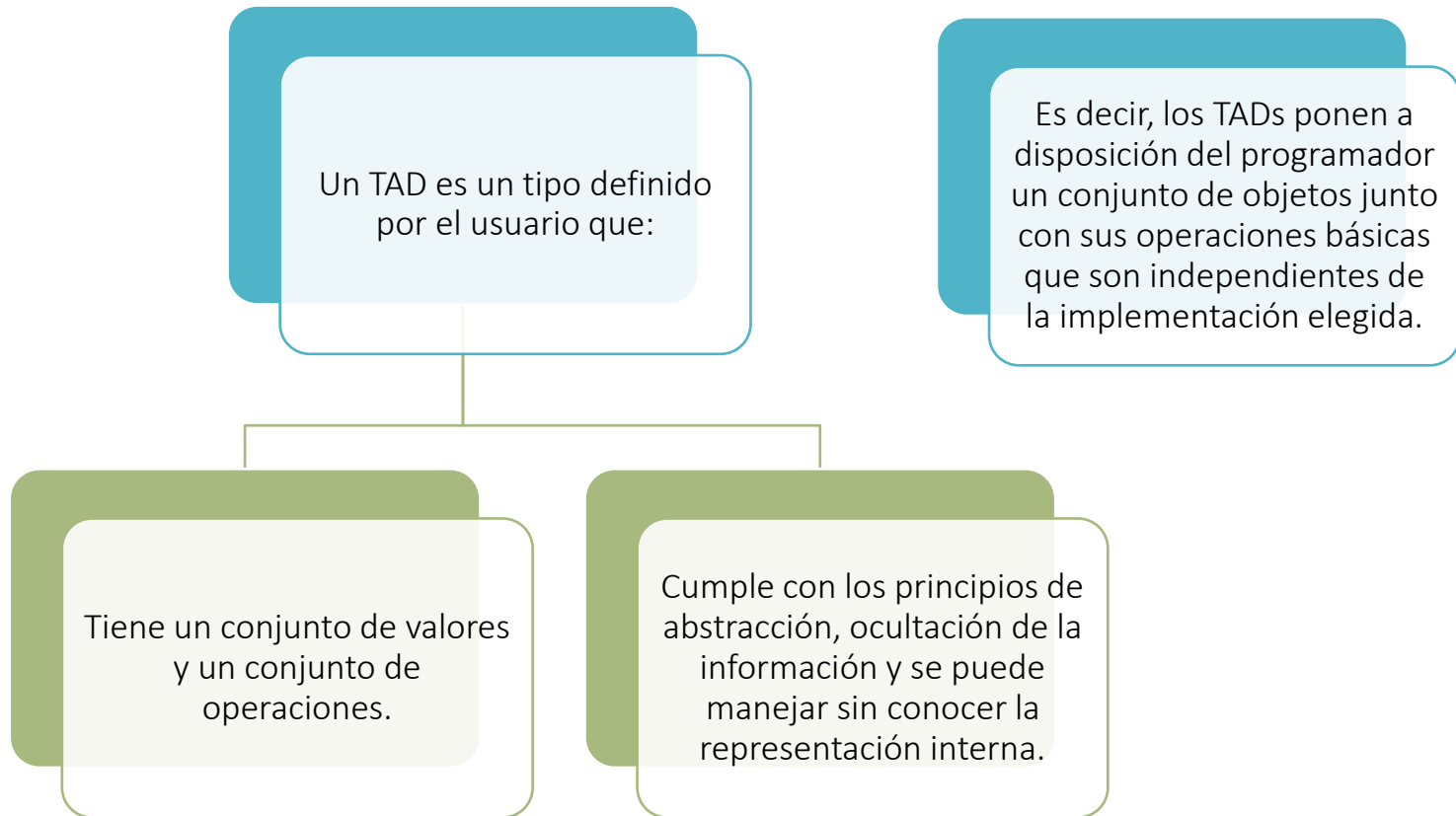
La **Abstracción de Datos** → Reconocer objetos del mundo real y abstraer sus aspectos fundamentales y su comportamiento de modo de poder representarlos en un ordenador.

Abstracción

Es la simplificación de un objeto de la realidad, sólo se consideran los aspectos más relevantes.



TAD



Tipos de Datos Abstractos

Lo que se persigue con los tipos de datos abstractos es que el programador deje de preocuparse por los detalles menores de implementación, **concentrándose en las operaciones globales del tipo de datos.**

Son una extensión del concepto de **tipo de dato definido por el usuario.**

Pueden considerarse como **cajas negras**: el programador-usuario solo ve su comportamiento y no sabe (ni le interesa saber) que contienen.

Ventajas del TDA

Facilita la reusabilidad de código.

Las modificaciones internas de los TDA no afectan a quienes los utilizan.

Permite mejor conceptualización y modelización del mundo real.

Mejora la robustez del sistema.

Mejora el rendimiento.

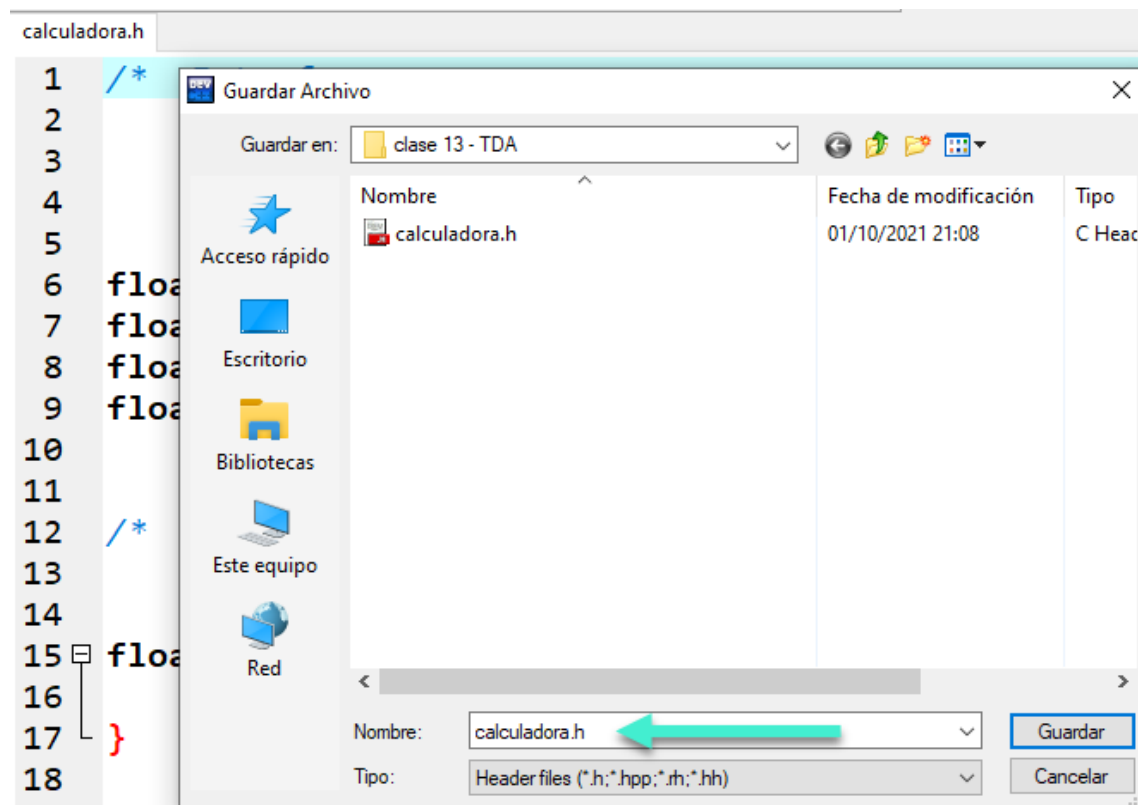
Permite la extensibilidad del sistema. Los componentes de software son mas fáciles de crear y mantener.

BIBLIOTECAS

- ✓ Las bibliotecas nos permite el uso de funciones en un programa sin la necesidad de escribir su código en nuestro programa, basta con llamar a la biblioteca.
- ✓ Existen bibliotecas estándares en C que ya vienen incluida en la mayoría de los compiladores (stdio.h, math.h...).
- ✓ Para utilizar nuestra biblioteca, basta con situar en la cabecera del programa el nombre de la biblioteca para poder utilizar las funciones contenidas en las mismas.

Implementación TDA en C

- ✓ Hay que crear Header file o archivo cabecera (.h)
- ✓ El archivo .h debe contener los prototipos y las implementaciones de las funciones.
- ✓ Para usarlo se debe colocar en la misma carpeta del archivo C que lo utilice.



Implementación TAD en C

✓ Desde el archivo fuente incluirlos como:

`#include "calculadora.h"`

PruebaLibCalculadora.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <string.h>
4  #include "calculadora.h" ←
5
6  /* Función principal */
7  int main() {
8      float nro1 = 10.5;
9      float nro2 = 2.25;
10
11     printf("%.2f + %.2f: %.2f \n", nro1, nro2, sumaDosNros(nro1, nro2) );
12
13     printf("%.2f - %.2f: %.2f \n", nro1, nro2, restaDosNros(nro1, nro2) );
14
15     printf("%.2f * %.2f: %.2f \n", nro1, nro2, productoDosNros(nro1, nro2) );
16
17     printf("%.2f / %.2f: %.2f \n", nro1, nro2, cocienteDosNros(nro1, nro2) );
18
19     return 0;
20 }
```

Bibliografía

Apuntes teóricos de la Cátedra Algoritmos y Estructuras de Datos II.

Algoritmos a Fondo, con implementaciones en C y JAVA; Pable A. Sznajdleder; 2012; Alfaomega; ISBN: 978-987-1609-37-6