



Tema 11

Contenido

- **Abstracción de los Datos.**
 - **Introducción**
 - **Historia**
 - **Definición y Conceptos Básicos.**
 - **Ejemplo**
- **Tarea**

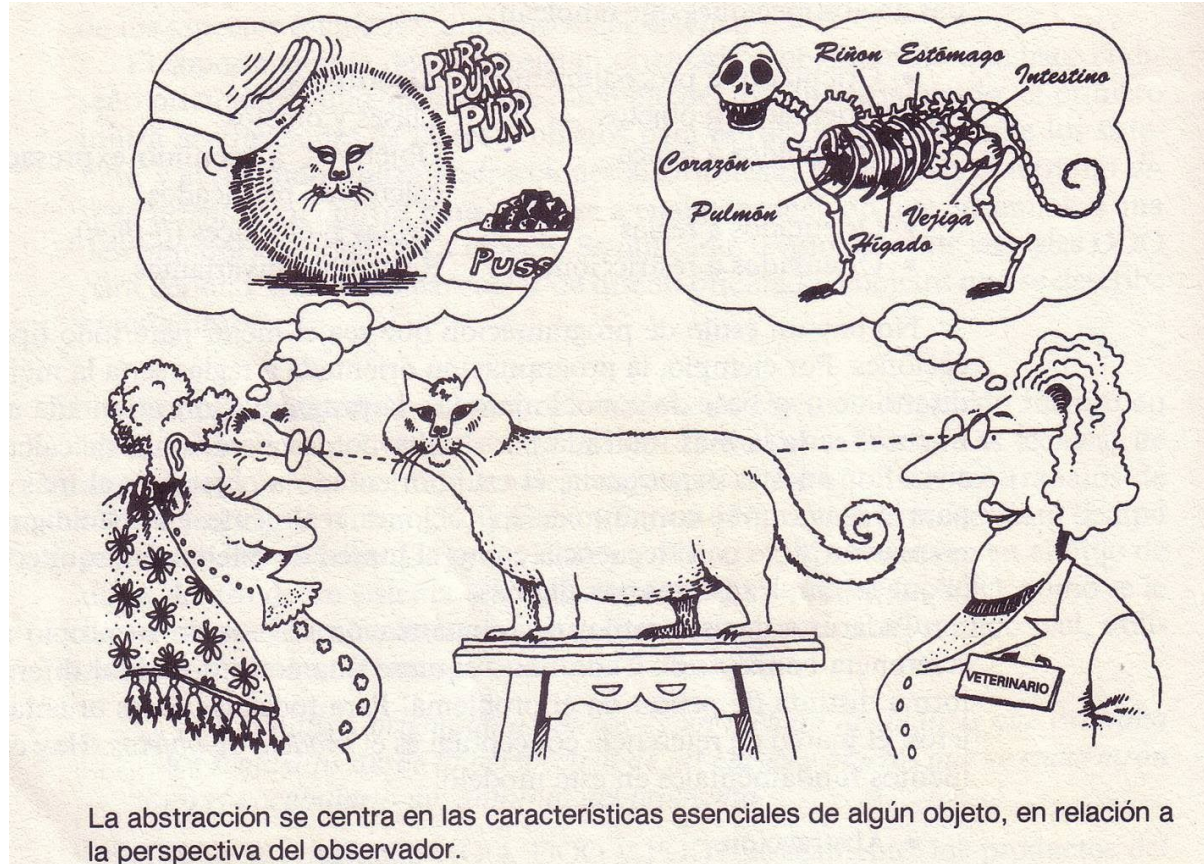
Contenido de Practica

- Diagrama de Redes
- Algoritmo (Diagrama de Flujo)
- Diseño de la Base de Datos.

Introducción.



Abstracción del dato



La abstracción se centra en las características esenciales de algún objeto, en relación a la perspectiva del observador.

Sinónimos o Palabras Claves

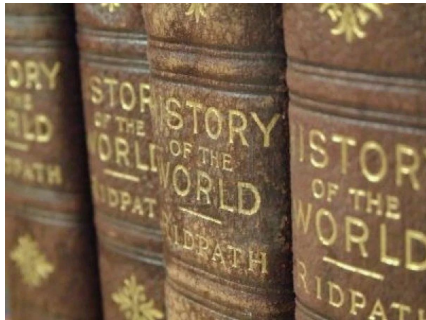
- Abstracción
- Aislar
- Datos
- Características esenciales
- Objeto
- Código
- Reutilizar
- Datos relevantes
- Objetivo
- Descomponer
- Partes
- Subpartes

Historia.

El origen del concepto del tipos de dato abstracto viene cuando se crea el tipo “*class*” con el lenguaje SIMULA 67 (Birtwistle et al. 1973)

Desde entonces se han ido desarrollando varios lenguajes que manejan estos tipos de datos abstractos.

- En 1977 Liskov lo introduce para el Lenguaje CLU-llina.
- En 1982 Stroustrup lo incorpora para el lenguaje C (C con clases)



```
! todo programa empieza con un begin y termina con un end ;
Begin

Class Saludos;
Begin
    OutText(";Hola Mundo!");
    OutImage;
End of class saludos;

REF(Saludos) objeto;
objeto :- New Saludos;

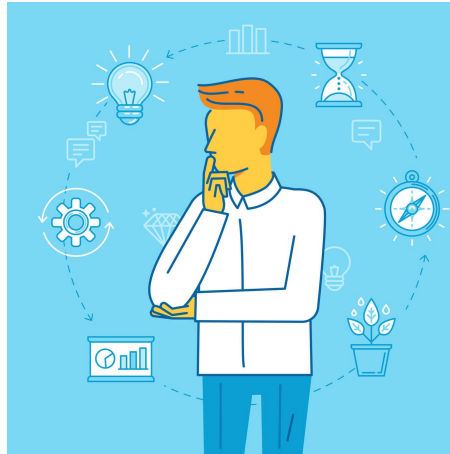
End of module program;
```


Definición TDA

1988 señalan que un TDA se entiende como “un modelo matemático con una serie de operaciones definidas en ese momento.”

1977 se expresa que un TDA es: “Una clase de objetos definida por una especificación independiente de las representación”

1974 se establece que “Un TDA define una clase de objetos abstractos la cual está completamente caracterizada por las operaciones definidas para estos objetos.”



Características del TDA

Los TDA son generalizaciones de los tipos de datos primitivos (enteros, reales, etc) al igual que los procedimientos y funciones son de las generalizaciones de operaciones primitivas (Suma, Resta, etc).

Se caracteriza por un conjunto de operaciones las cuales representan el comportamiento del TDA.



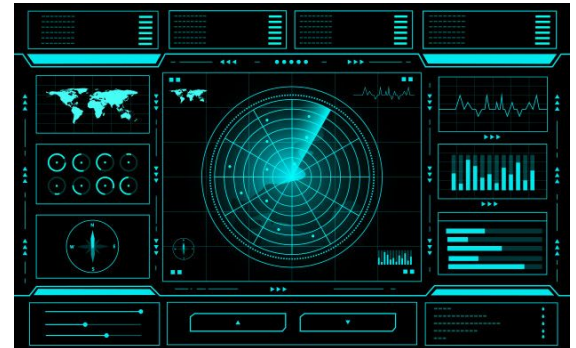
Función.

Consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan.

En programación, se da énfasis en el ¿que hace? más que en ¿como lo hace?
(Característica de la caja negra)

En algunos lenguajes de programación se puede dividir en 2 tipos de categorías.

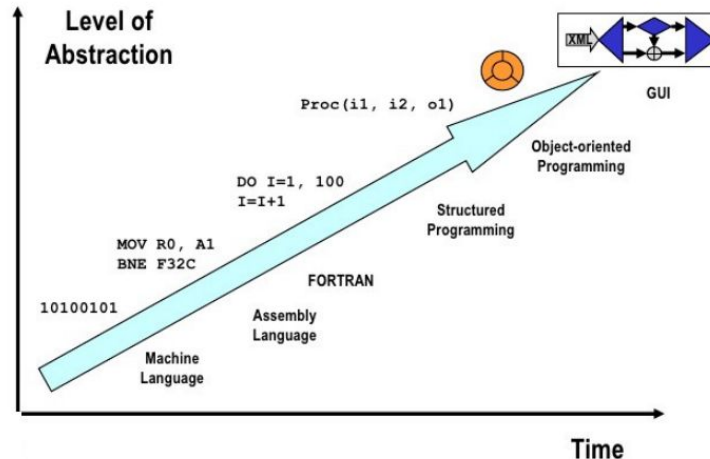
- Abstracción del Dato (Pertenece a los datos)
- Abstracción del Control (Pertenece a las estructuras de control)



Evolución.

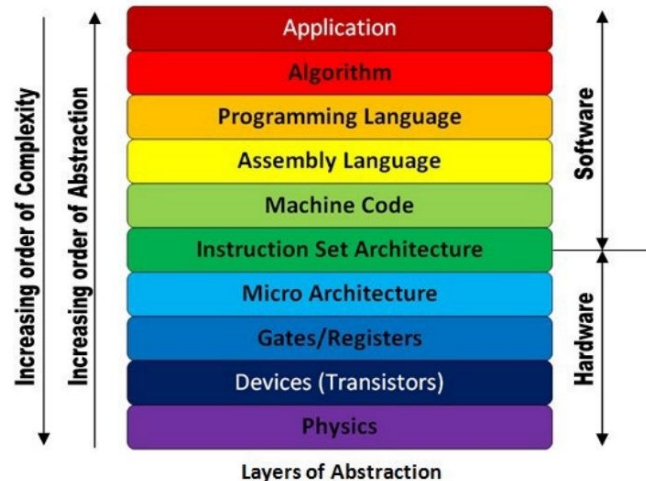
- Los paradigmas de programación han aumentado su nivel de abstracción.
 - Lenguajes de máquina (más cerca del computador y menos de entendimiento humano.
 - Pasando por los lenguajes de comando, imperativos, orientados a objetos, etc

Computer Science Is About Abstraction



Importancia.

- Si se ve desde el punto de vista de programación.
 - La abstracción expresa características esenciales de un objeto. Distinguiéndose de los demás.
 - Entonces la encapsulación separa las características esenciales de las no esenciales de un objeto.
 - La misma genera una ilusión de simplicidad dado a que minimiza la cantidad de características que definen a un objeto.



Importancia.

- Debido a que un problema no solo lo tiene un programador, la mejor solución en el momento fue la programación orientada a objetos.
 - Con esto garantiza la interoperabilidad entre usuarios de manera que el código se pueda reutilizar.



- Se puede decir que la abstracción permite que se disponga de las características de un objeto que necesitemos.

Persona



En sistema administrativo -> Podríamos poner Nombre, Edad, Dirección, etc.

En sistema biológico -> se necesita ADN, RND, Genx1, Genx2, etc

Ambos requieren diferentes atributos, pero si asignamos todos los atributos necesarios para ambos sistemas, con el sistema de abstracción podríamos excluir todos aquellos que no son relevantes para el respectivo sistema.

Objetivo Principal.

Visualizar el objeto en mente y descomponerlo en varias partes, y a su vez descomponer estas partes en otras subpartes y así sucesivamente hasta tener las partes elementales y atómicos que no se pueden descomponer.



Ejemplo Simple.

Programemos una Silla



Ejemplo Simple.

Programemos una Silla



Que es la silla?

Como esta compuesta?

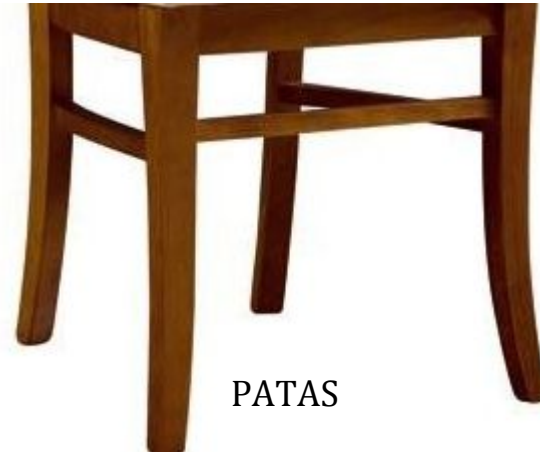
Ejemplo Simple.

Programemos una Silla



Respaldo

Asiento



PATAS

Ejemplo Simple.

Programemos una Silla



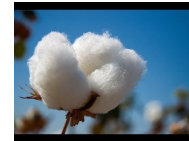
Que tienen las patas?
Que tiene el respaldo?



Que tiene el asiento?



Que tiene el
almohadón?



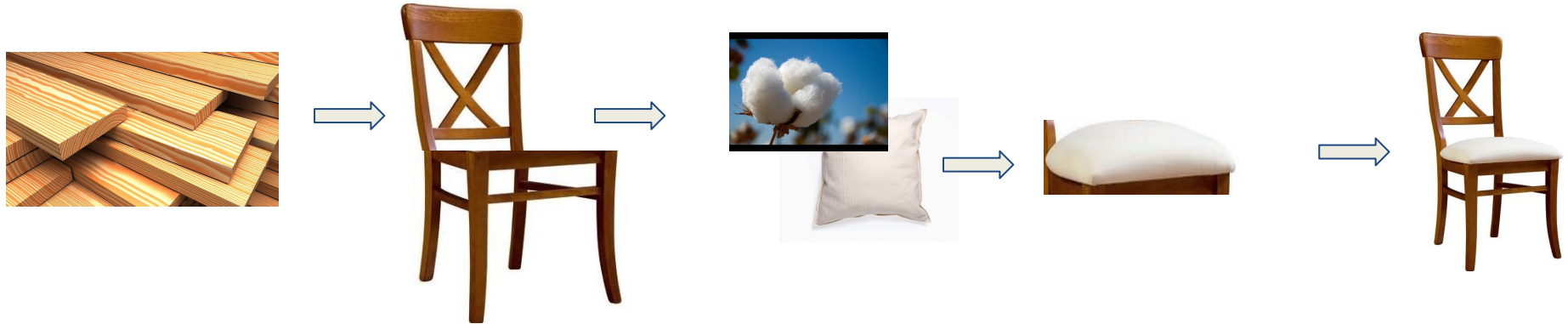
Algodón

Hemos
descompuesto
la silla en sus
partes
esenciales.

Ejemplo Simple.

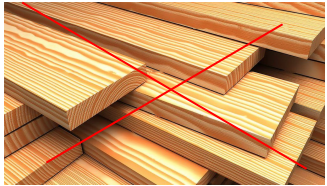
Programemos una Silla

Y la armamos.

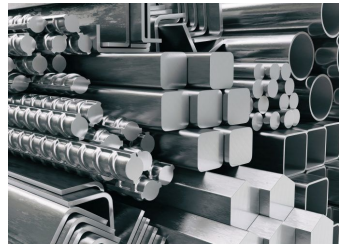


Ejemplo Simple.

Programemos una Silla



Y si quiero hacer una silla de metal?



Conclusiones.

La programación orientada a objetos es la agregación de técnicas que se han ido mejorando a través del tiempo.

En momentos se puede ver que la abstracción puede ser tediosa, sin embargo es aun mas difícil no hacerlo porque se puede durar el doble de tiempo en la fase de programación y es recomendable usar las técnicas experimentadas y recomendadas para aumentar la calidad del entregable correspondiente.

Tarea 5 de Investigación.

Computación Cuántica.

Explique con sus propias palabras cada uno de los siguientes puntos.

- Quien es Allan Turing
- Qué es y cómo funciona la Máquina Turing
- Hipercomputación.
- Test de turing
- Han superado el test? Quien y como?
- QuBit
- Superposición cuántica
- Entrelazamiento cuántico
- Teletransporte cuántico.

Explique con sus propias palabras la diferencia entre los siguientes puntos.

- Computadora cuántica con computadoras actuales.
- Bit con QuBits