

Algoritmos y Estructuras de Datos I

Tema = "Abstracción -
Modularización - Encapsulamiento"

Dr. Carlos A. Catania
Ing. Lucia Cortés
Lic. Javier Rosenstein



Cómo se llega de una situación del mundo real a un modelo computacional?



proceso de abstracción

Entonces, qué es «abstracción»?

- La abstracción consiste en captar las **características esenciales** de un objeto, así como su **comportamiento**

Veamos un ejemplo



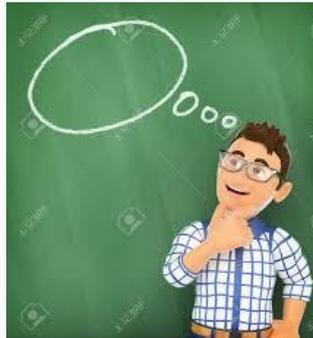
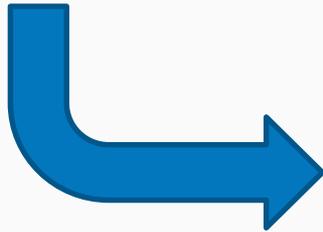
A primera vista, diríamos que se trata del mismo objeto?

- **Características esenciales:** marca, modelo, número de chasis, peso, tipo de combustible que utiliza, etc...
- **Comportamiento:** se pone en marcha, avanza, retrocede, acelera, frena, etc...

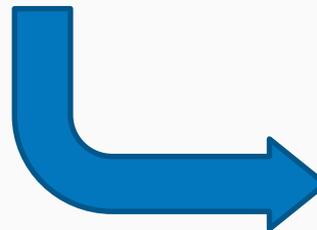
Otro ejemplo



Observación del mundo real



Abstracción



	Columns		
Rows	(0, 0)	(0, 1)	(0, 2)
	(1, 0)	(1, 1)	(1, 2)
	(2, 0)	(2, 1)	(2, 2)

Element position: (row, column)

Arreglo

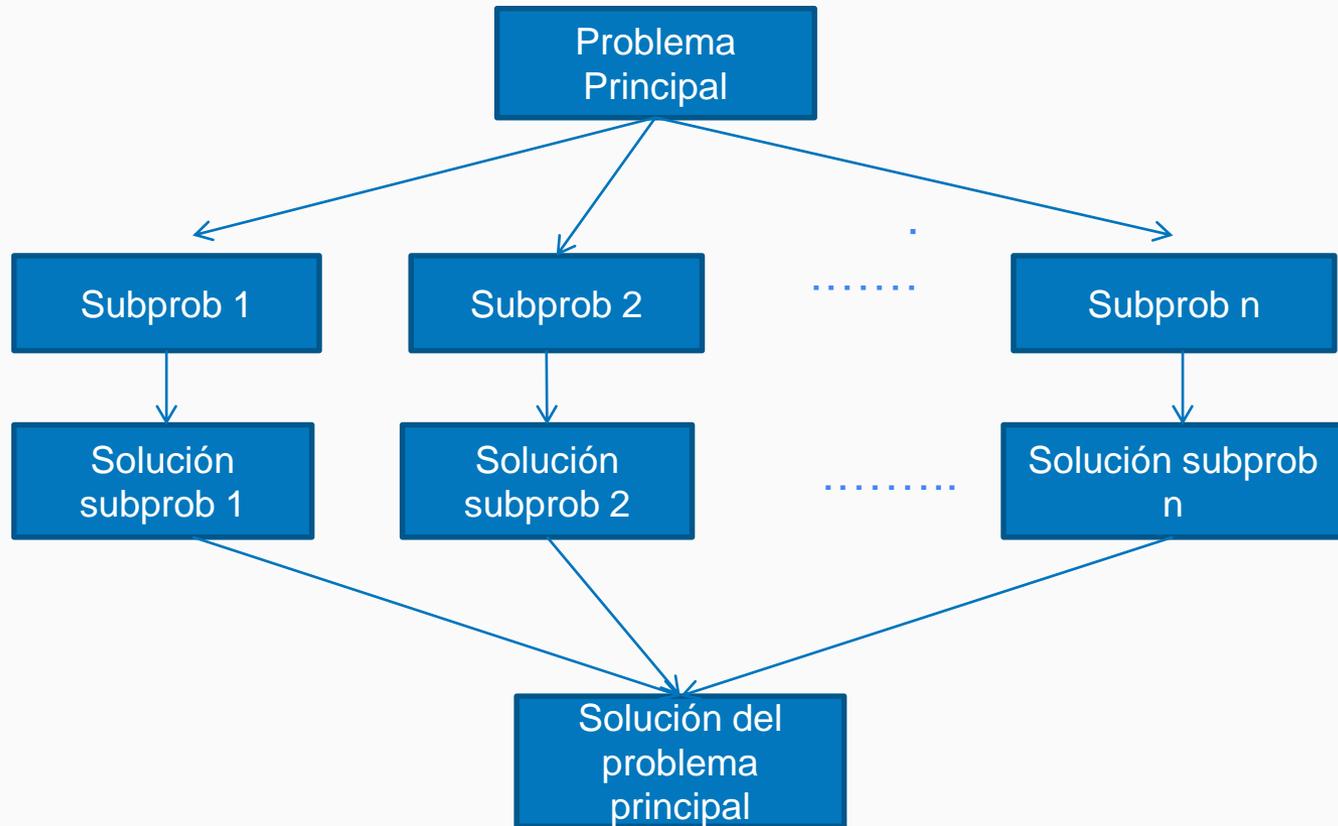
- Abstracción, consiste en buscar generalidades que ayuden a **resolver un problema** a través de un **programa**

...entonces,

Un **programa** puede definirse como un **modelo abstracto** de un procedimiento o fenómeno que sucede en el mundo real

Esto implica un proceso de **descomposición del problema en partes** para su mejor comprensión

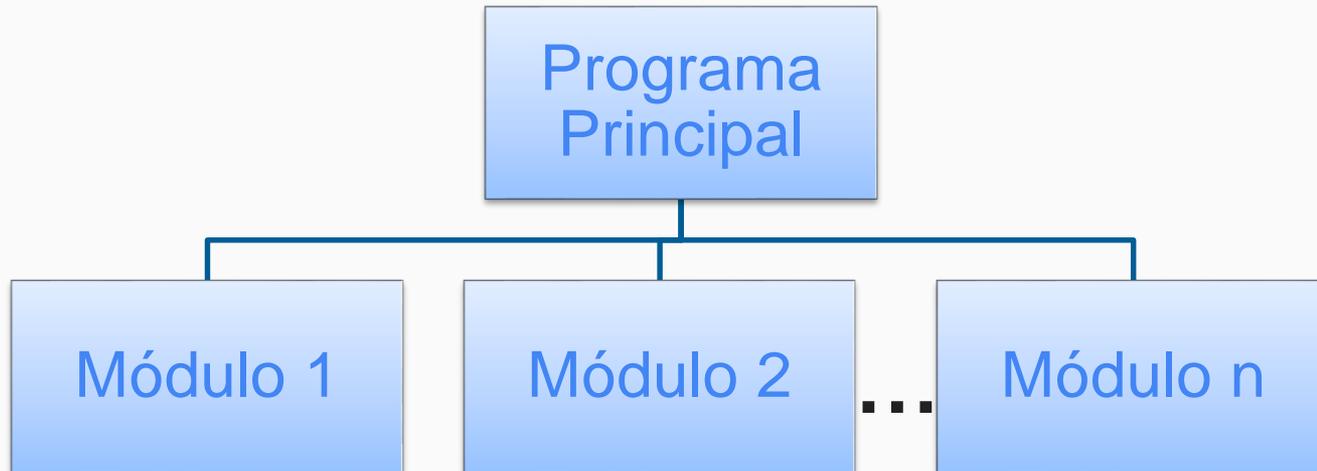
Descomposición de un problema



Cada parte involucra elementos de similares características y comportamiento

Análogamente, en términos informáticos:

nos referimos a la **división de un programa en módulos** para su mejor tratamiento



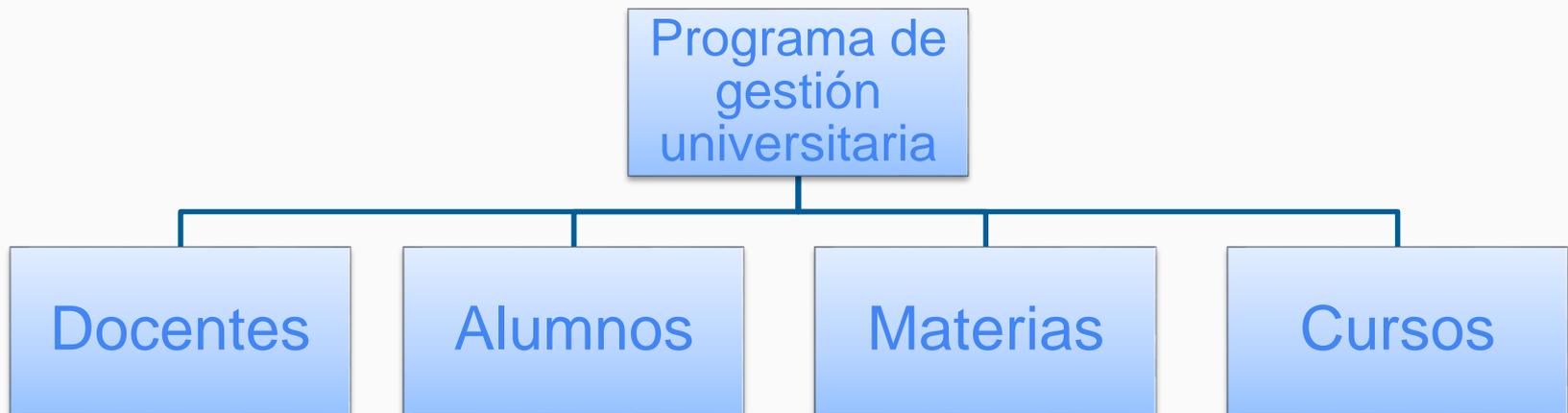
... dividir un programa en partes **funcionalmente independientes**, con el objeto de lograr un mejor tratamiento del mismo

Qué significa funcionalmente independientes?

- ✓ **Alta cohesión:** es decir lograr el mayor grado de identificación entre los elementos que conforman el módulo

- ✓ **Bajo acoplamiento:** máxima independencia entre módulos

Ejemplo: Modularización de un programa de gestión universitaria



Gestión de
Stock

La modularización contribuye a:

- ✓ Mayor claridad
- ✓ Mejorar la productividad
- ✓ Simplificar el mantenimiento
- ✓ Mayor escalabilidad
- ✓ Minimizar redundancia
- ✓ Mayor reusabilidad



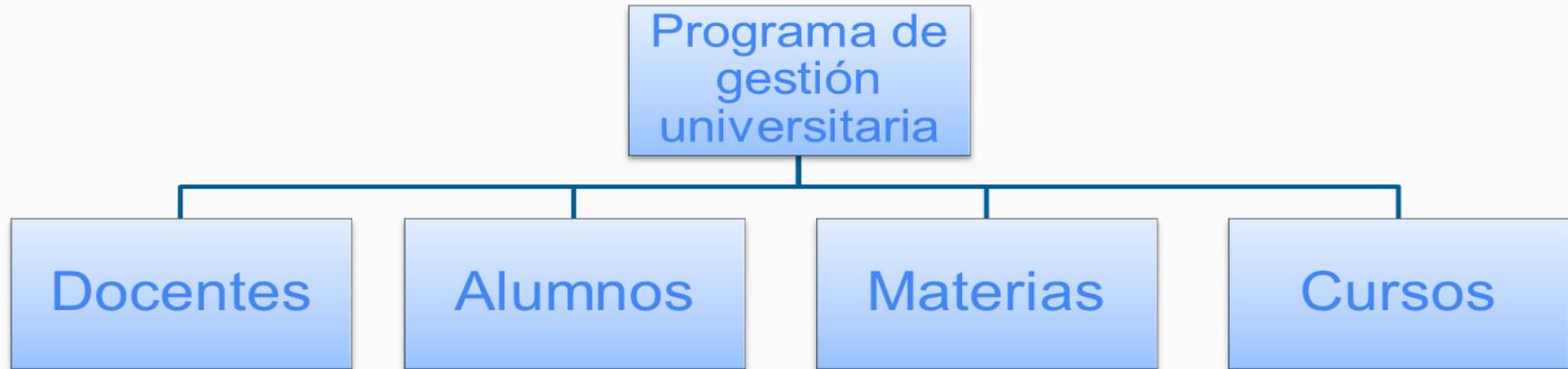
Formas de modularización

Dependiendo de cada lenguaje de programación, podemos ver distintas formas de modularización:

- ✓ Subrutinas
- ✓ Procedimientos
- ✓ Funciones
- ✓ Clases

.... retomando el ejemplo

Cada módulo **encapsula** funciones y datos que le son propios



Docentes: legajo, datos personales, carrera, materia que dicta, horario, etc.

Alumnos: legajo, datos personales, carrera, estado académico, etc.

Materias: código, carrera, nivel, semestre, horas de cursado, etc.

Curso: ubicación, ocupación, capacidad, etc.

El encapsulamiento de un módulo implica que tanto las **estructuras de datos** como **los algoritmos** que las utilizan son accedidos desde otro módulo a través de funciones desarrolladas a tal efecto llamadas **interfaces**.



- En el encapsulamiento lo importante es el **qué hace** y no **cómo lo hace**



Ejemplo:

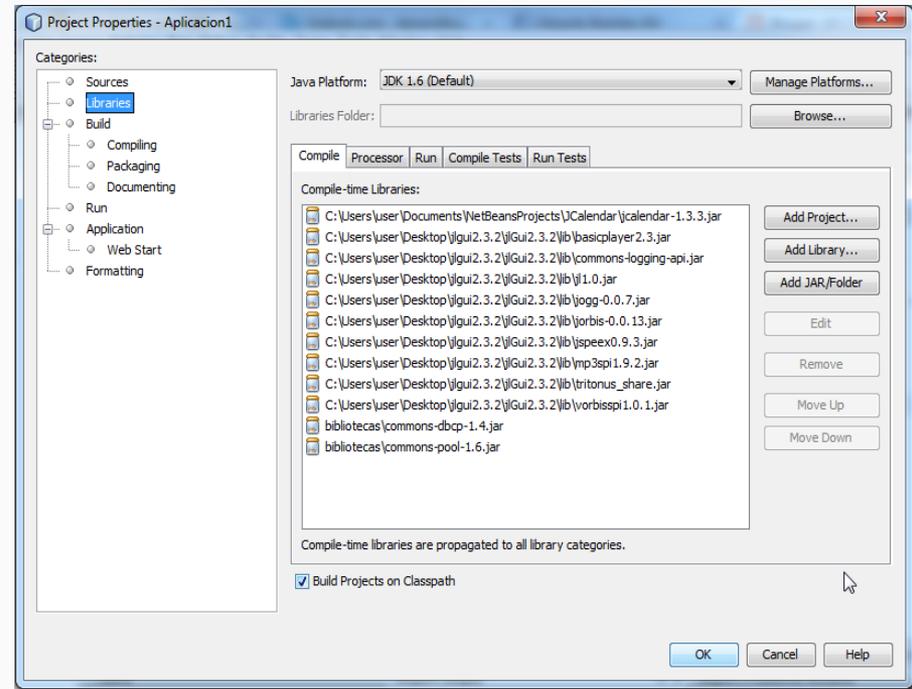
Cuando compramos un televisor, nos interesa su apariencia, calidad de imagen y sonido, conexiones... pero no nos interesa saber cómo funciona cada una de las partes internas del aparato.



Otro ejemplo:

De la misma manera, los lenguajes de programación cuentan con librerías que contienen funciones específicas que aportan soluciones para el desarrollo de programas.

En este caso, lo que le interesa al programador es la utilidad de alguna función y su interfaz; pero no su funcionamiento interno.



El encapsulamiento implica ocultamiento y las ventajas son:

- **Contribuye a la Integridad**

El usuario está forzado a utilizar funciones definidas para modificar los datos garantizando así la integridad de los mismos

- **Reduce la Complejidad**

El ocultar detalles y visualizar los módulos como «cajas negras», facilita la comprensión del programa

Conclusiones

- El **objetivo** del proceso abstracción es plantear un **fenómeno del mundo real** en términos de un **modelo computacional**.
- La adecuada **abstracción** de un **problema** permite dividirlo en partes, **disminuyendo así su complejidad**. (Técnica «Divide y Vencerás»)
- Desde el punto de vista computacional, **dichas partes conformarán módulos funcionalmente independientes que encapsularán estructuras de datos y funciones**; es decir **TADs (Tipo Abstracto de Datos)**.

*Un **tipo abstracto de datos** es un modelo matemático compuesto por una colección de operaciones definidas sobre el modelo y un nombre que lo identifica.*

Titular: Dr. C.A. Catania <harpomaxx@gmail.com> @harpolabs
Adjunto: Ing. L. Cortés <luciacortes5519@gmail.com>
JTP: Lic. J. Rosenstein <rosensteinjavier@gmail.com>

HAPPY HACKING!

