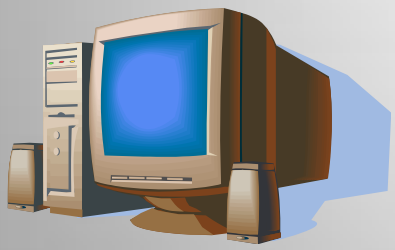
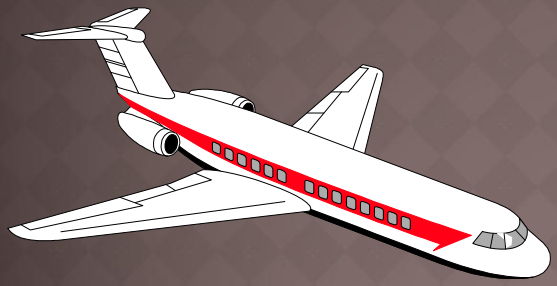
PARADIGMA ORIENTADO A OBJETOS

Yalù Galicia Hernàndez





CONTENIDO

- Introducción
  - ¿Qué es la Programación Orientada a Objetos?
- Conceptos básicos
  - Abstracción
  - Jerarquía
  - Encapsulación
  - Objeto
  - Clase
  - Herencia
  - Polimorfismo

PARADIGMA ORIENTADO A OBJETOS

- Se organiza el software como una colección de objetos discretos que encapsulan estructuras de datos y comportamiento.
- Un sistema OO funciona mediante la colaboración entre los objetos que se comunican entre sí.
- El concepto se extiende a los métodos de análisis y diseño
  - Se utilizan los objetos del mundo real como base para construir modelos
  - Los elementos que forman los sistemas del mundo real se corresponden con objetos del software

PARADIGMA ORIENTADO A OBJETOS

- Grady Booch define a la programación orientada a objetos como:
  - Un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de alguna clase, y cuyas clases son todas miembros de una jerarquía de clases unidas mediante relaciones.
- Wikipedia dice:
  - La **programación orientada a objetos** es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento

¿Y QUÉ CON LA POO?

- La Programación Orientada a Objetos funciona de la misma forma:
- Al programar utilizando técnicas de programación orientada a objetos, nuestro programa estarán formados por muchos componentes independientes y diferentes; cada uno con una funcionalidad específica en el programa y que puede comunicarse o dar respuesta a solicitudes de los demás componentes de manera predefinida a través de mensajes.

CONTENIDO

- Introducción
  - ¿Qué es la Programación Orientada a Objetos?
- Conceptos básicos
  - Abstracción
  - Jerarquía
  - Encapsulación
  - Objeto
  - Clase
  - Herencia
  - Polimorfismo

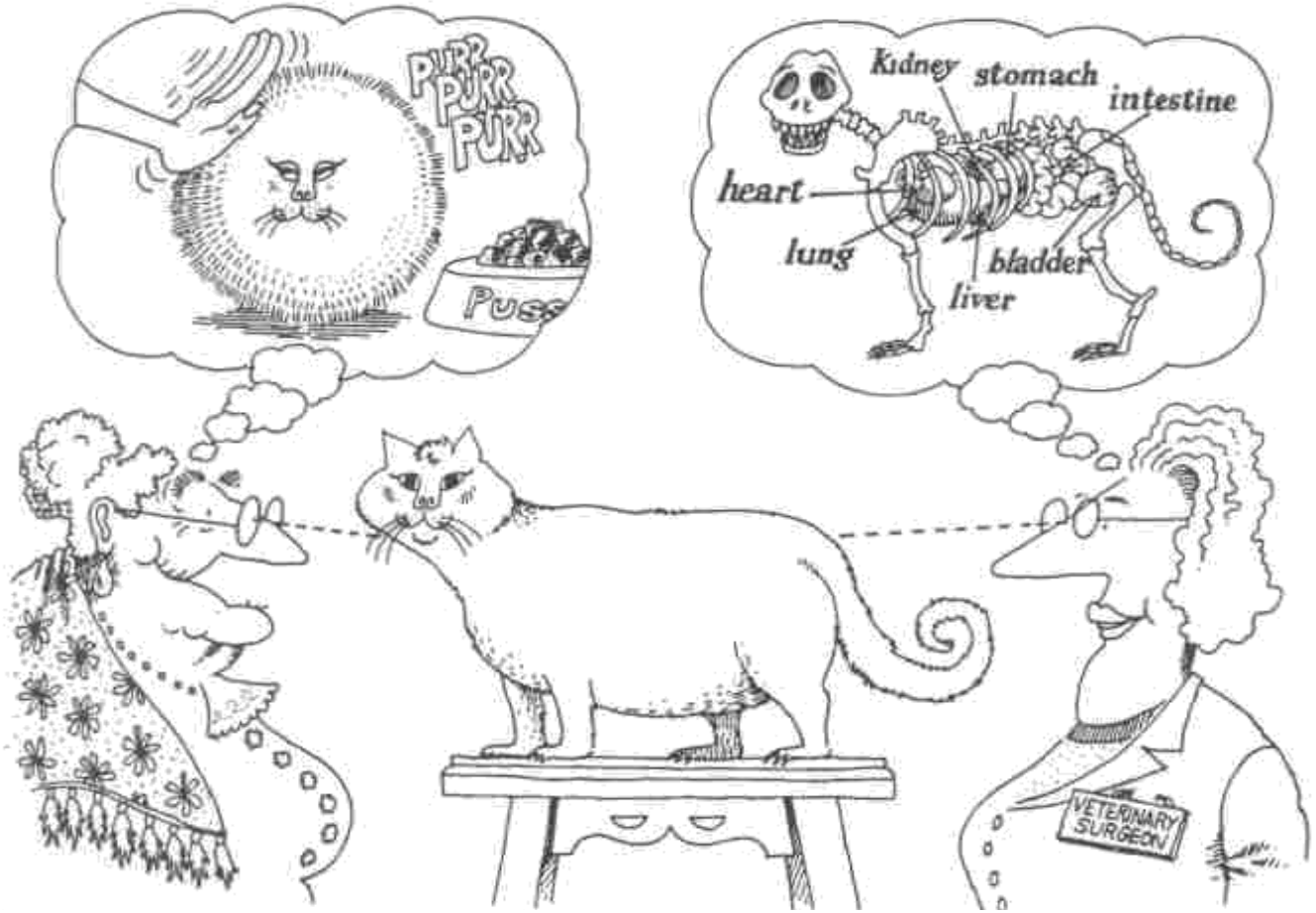
PROGRAMACIÓN OO

- La programación orientada a objetos está basada en los siguientes conceptos:
  - Abstracción
  - Encapsulación
  - Jerarquía
  - Clase
  - Objeto
  - Modularidad
  - Herencia
  - Polimorfismo
- Se dice que si alguno de estos elementos no existe, entonces el modelo no es orientado a objetos.

ABSTRACCIÓN

- Aún cuando existe una multitud de aves diferentes, podemos reconocer **un AVE** en cuanto la vemos, incluso aún cuando esa ave en particular, no lo hayamos visto nunca.

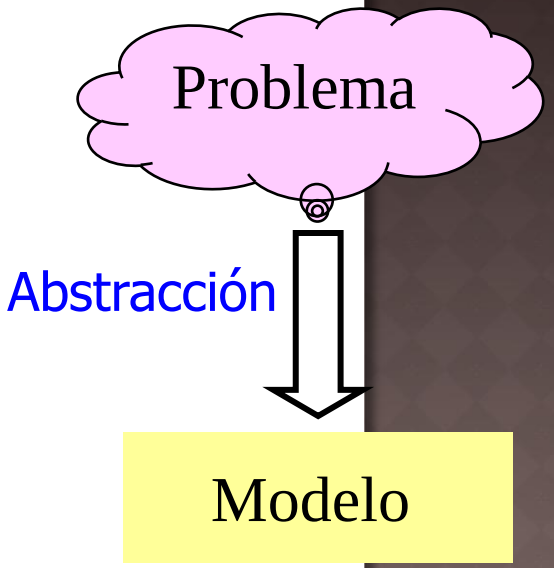
ABSTRACCIÓN



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

ABSTRACCIÓN

- **Abstracción** consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan, identificando sus características **esenciales,** las cuales lo **distinguen** de los demás.
- En otras palabras, es la capacidad de conceptualizar entidades genéricas de información a partir de cosas concretas

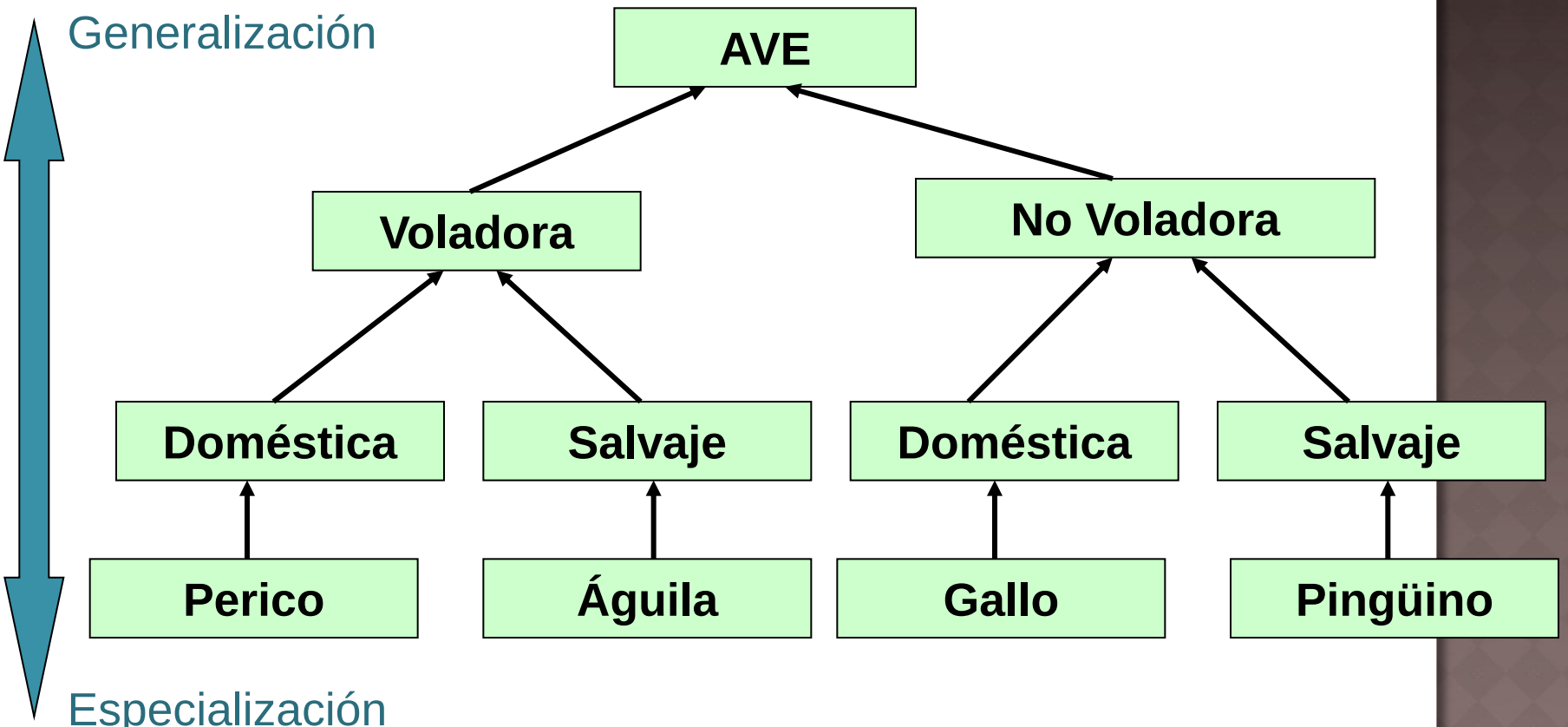


La Abstracción Minimiza la Complejidad

JERARQUÍA

- La abstracción es algo bueno, pero excepto en las aplicaciones más triviales, puede haber muchas más abstracciones diferentes de las que se pueden comprender simultáneamente.
- Frecuentemente un conjunto de abstracciones forma una jerarquía, y la identificación de esas jerarquías en el diseño simplifica en gran medida la comprensión del problema.
- La jerarquía es una clasificación u ordenación de abstracciones

JERARQUÍA

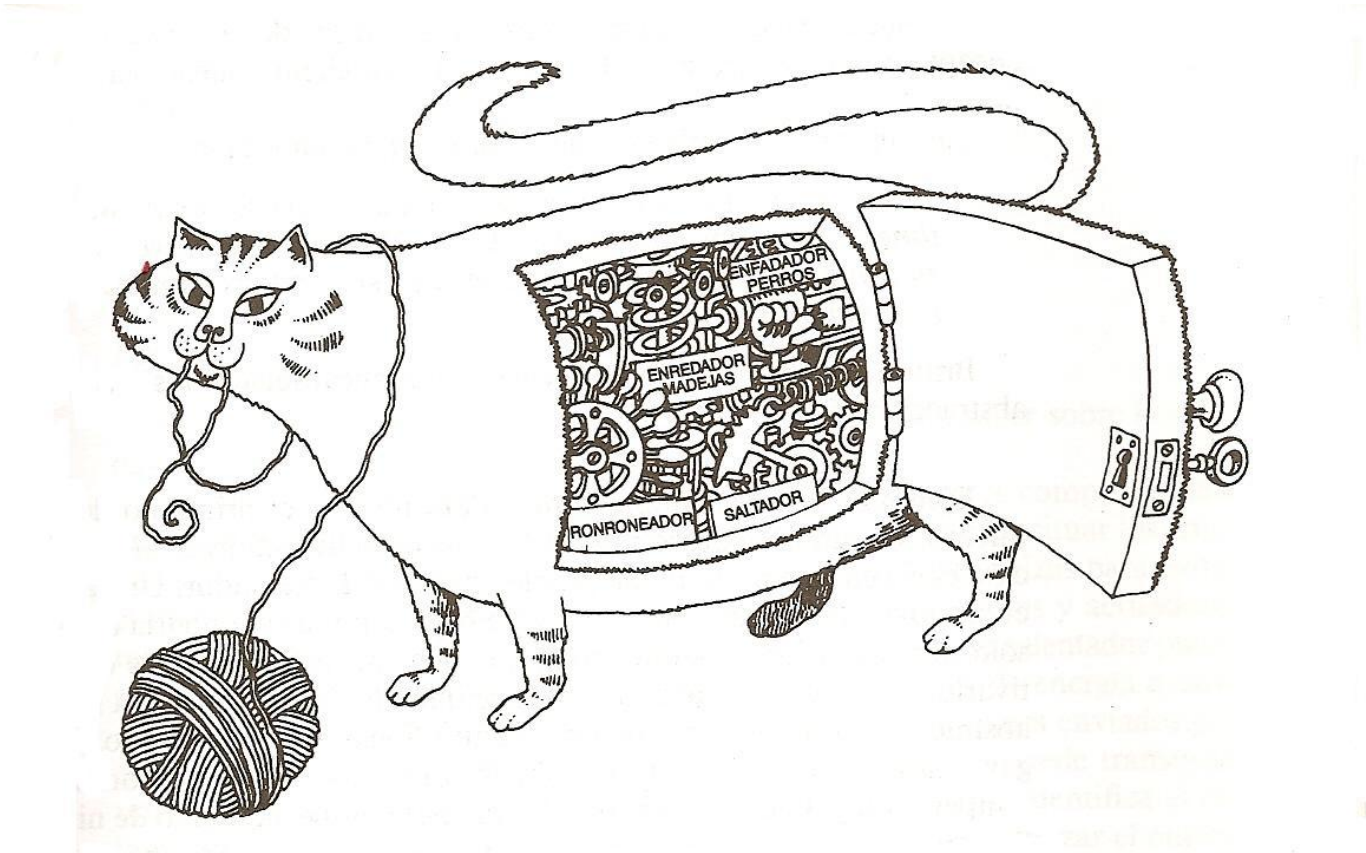


La Jeraquia Organiza la Complejidad

ENCAPSULACIÓN

- Un principio fundamental de la POO es la ocultación de la información, que es el proceso de ocultar dentro de sí mismo, todos los “secretos” de un objeto, que no contribuyen a sus características esenciales.
- Al encapsular los datos del objeto, se maximiza la reutilizabilidad, se reduce la dependencia de los datos y se minimiza el tiempo de depuración

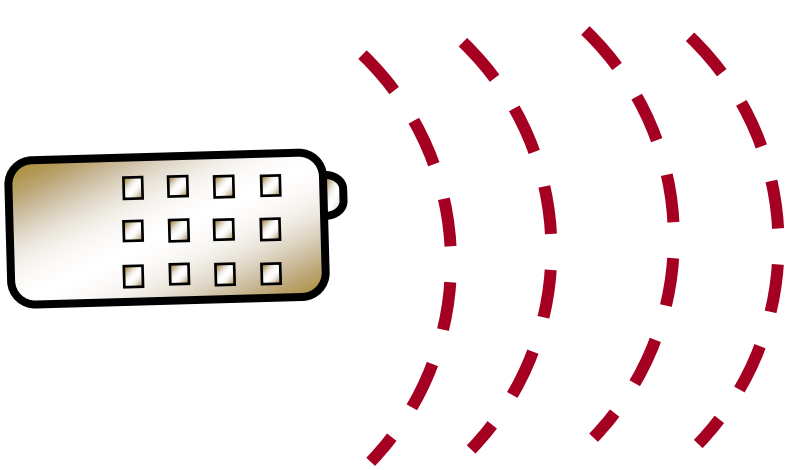
ENCAPSULACIÓN

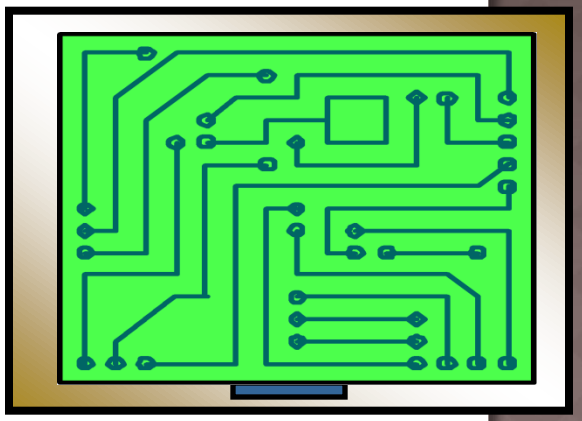


La encapsulación esconde la complejidad

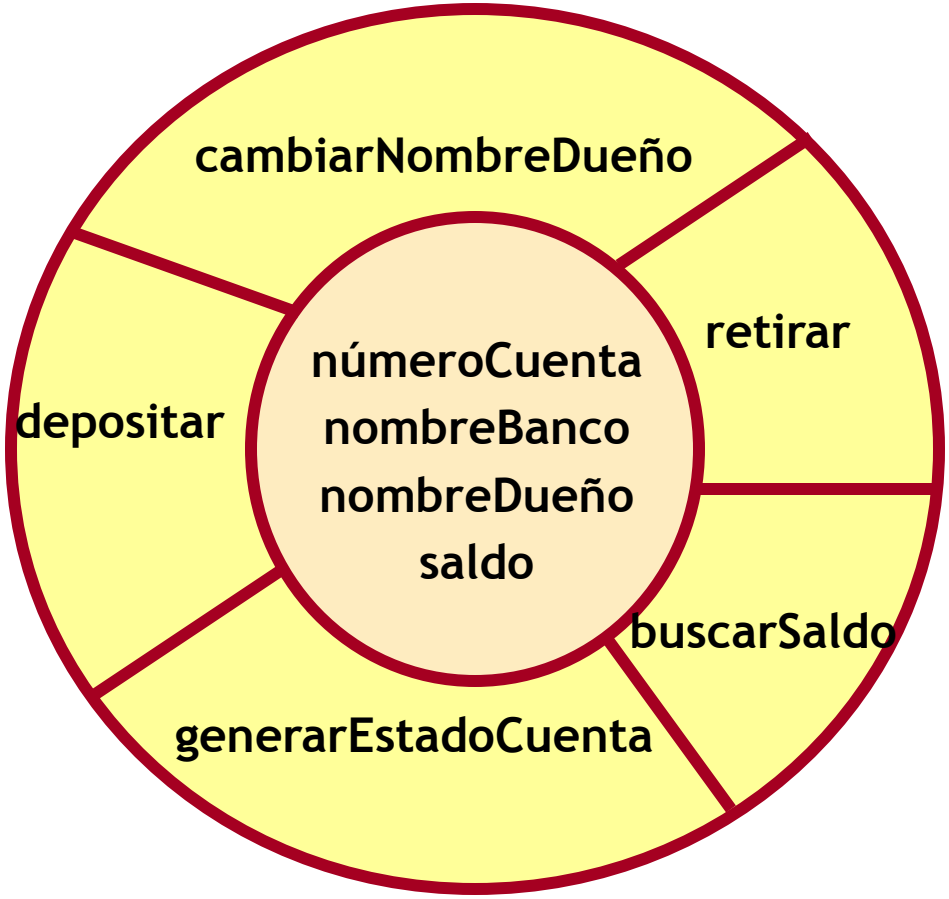
ENCAPSULACIÓN

- En otras palabras, es la capacidad de esconder los detalles de como funciona algo, detrás de una interfaz.
  - Solo se necesita conocer la interfaz para poder usar alguna cosa
  - El usuario no se ve afectado si se cambia o mejora el funcionamiento interno de algo, mientras se mantenga la interface



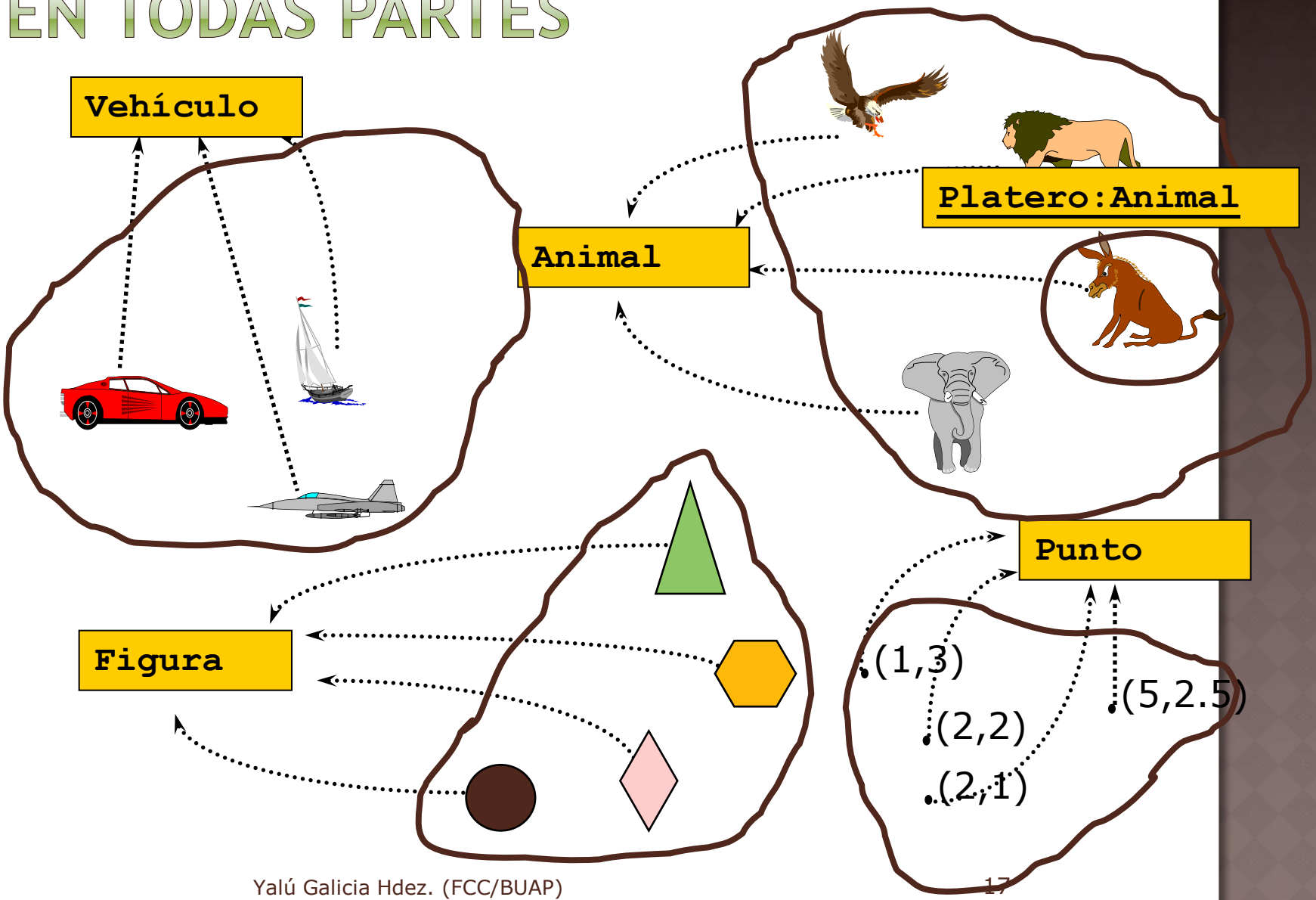


ENCAPSULACIÓN



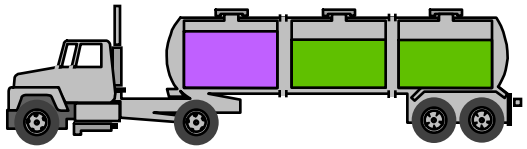
- El estado de un objeto no puede ser modificado por los objetos clientes directamente.
- Los valores de los atributos solo pueden mostrarse o cambiarse por las operaciones proporcionadas en el interface

LAS CLASES Y LOS OBJETOS ESTÁN EN TODAS PARTES



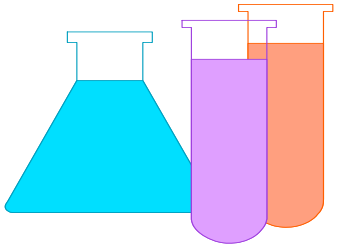
¿QUÉ ES UN OBJETO?

- Informalmente, un objeto representa a una entidad, ya sea física, conceptual o software
  - **Entidad física**



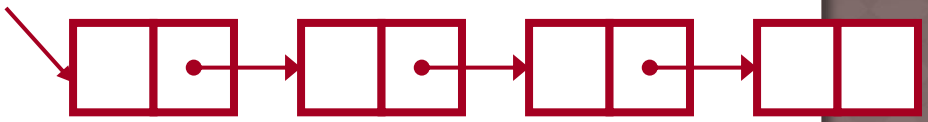
Camión

  - **Entidad conceptual**



Proceso Químico

  - **Entidad de Software**



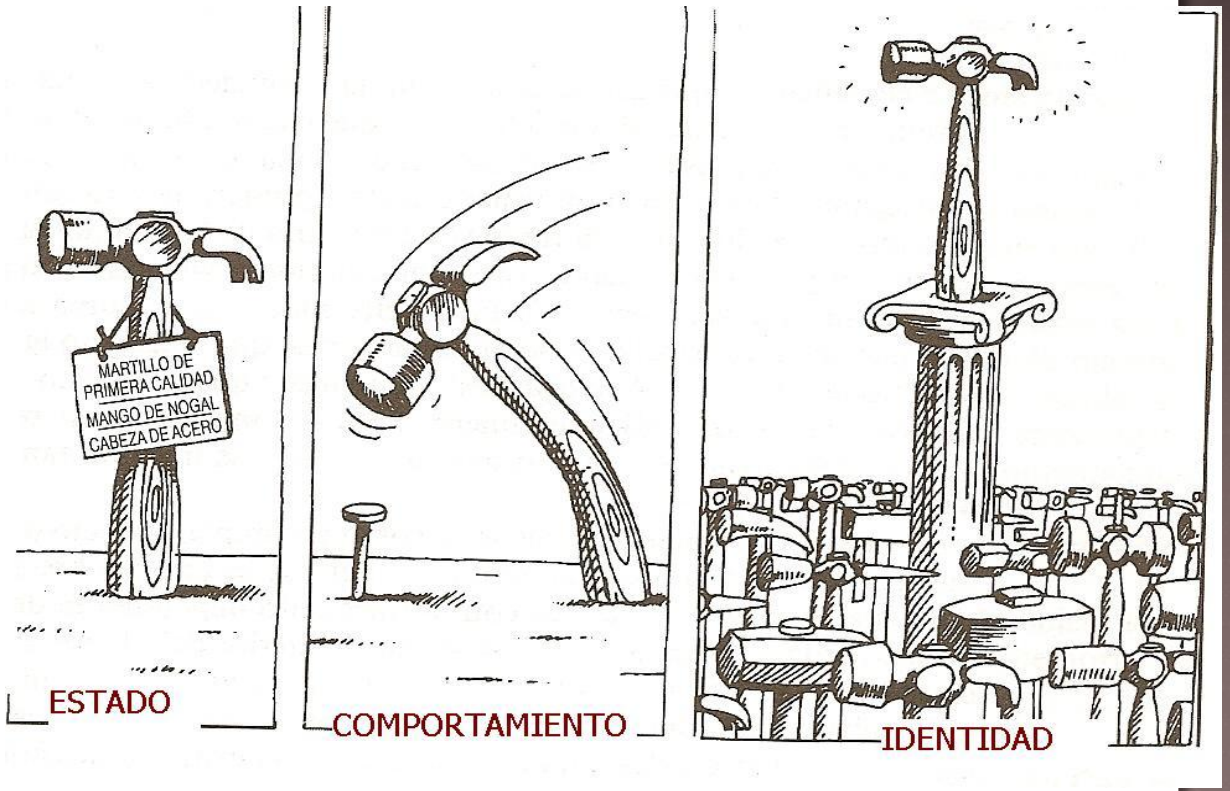
Lista Enlazada

OBJETO

- Un objeto representa un elemento, unidad o entidad individual e identificable, ya sea real o abstracta, con un papel bien definido en el dominio del problema.
- En términos generales, se define un objeto como cualquier cosa que tenga una frontera definida con nitidez.

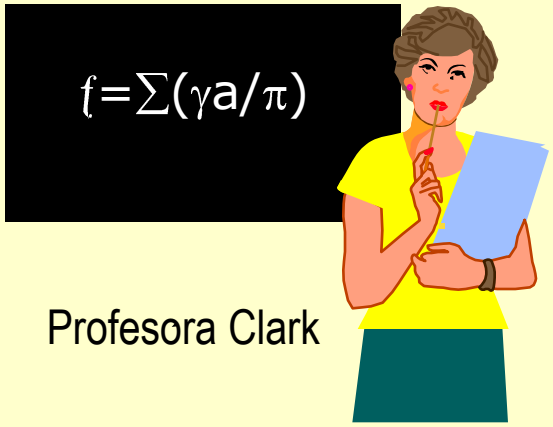
OBJETO

- Un objeto es algo que tiene:
  - Estado
  - Comportamiento
  - Identidad



UN OBJETO TIENE UN ESTADO

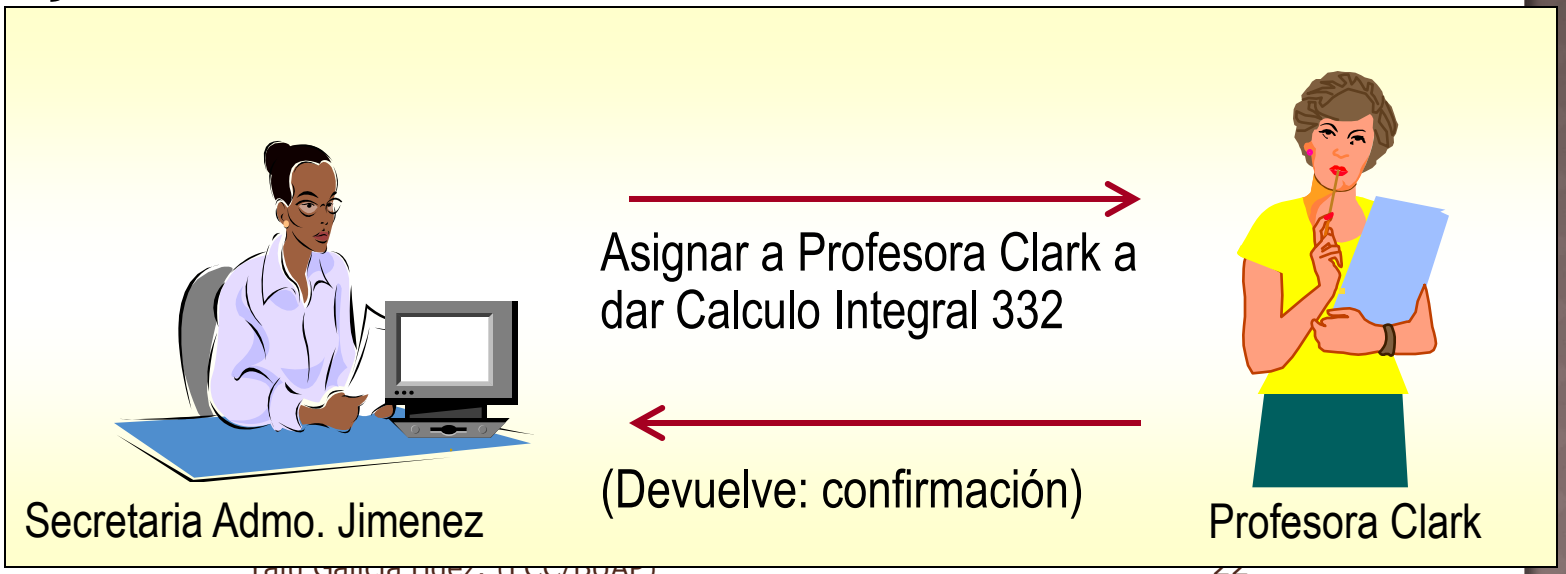
- El estado de un objeto es una de las posibles condiciones en que un objeto puede existir
- El estado de un objeto normalmente cambia con el tiempo
- El estado de un objeto es usualmente implementado por un conjunto de propiedades llamadas atributos, mas los enlaces que el objeto pueda tener con otros objetos
- El estado lo establecen los valores de los atributos y enlaces



Nombre: Joyce Clark
Id Empleado: 4322456
Contratación: 01/06/2005
Puesto: Profesora Titular

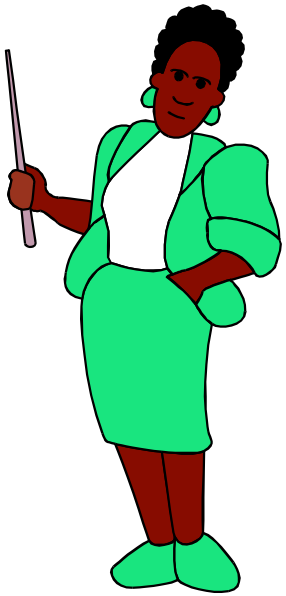
UN OBJETO TIENE COMPORTAMIENTO

- El comportamiento determina como un objeto actúa y reacciona
- El comportamiento define la manera en la que un objeto responde a las peticiones de otros objetos
- El comportamiento visible de un objeto se modela con un conjunto de mensajes a los que el puede responder
- Los mensajes se implementan como las operaciones del objeto



UN OBJETO TIENE IDENTIDAD

- Cada objeto tiene una identidad única, aun si su estado en un momento dado, es idéntico al de otros objetos



Profesora “J. Pérez”
Enseña Matemáticas

Profesor “J. Pérez”
Enseña Matemáticas

Profesora “J. Pérez”
Enseña Matemáticas

¿QUÉ SON LAS CLASES?

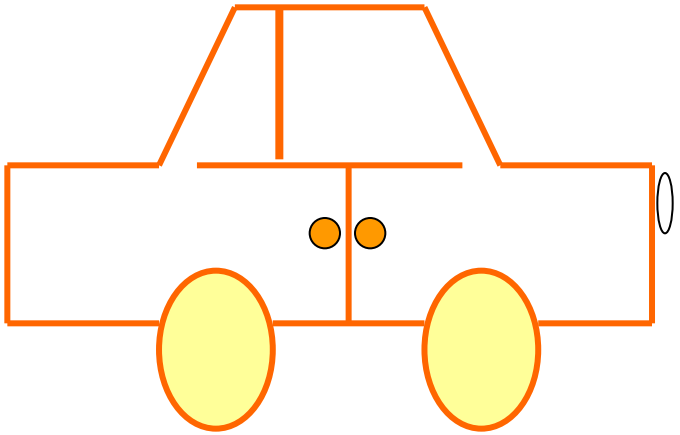
- Cuando se han identificado muchos objetos en un dominio, decimos que una clase es una abstracción que describe un grupo de objetos que tienen:
  - propiedades en común (atributos)
  - comportamiento en común (operaciones)
  - relaciones comunes con otros objetos (asociaciones)
  - semántica en común (descripción breve)
- Una clase es una abstracción porque:
  - enfatiza características relevantes al sistema
  - suprime otras características

CLASES

- En los lenguajes Orientados a Objetos, las clases están compuestas por dos características básicas:
  - Atributos
  - Comportamientos.
- Los atributos son las características individuales que diferencian un objeto de otro y determinan su apariencia (ej. color, estilo, marca,etc.), su estado (ej. encendido o pagado) y otras cualidades.
- El comportamiento de una clase determina la manera en que un objeto de esa clase opera o reacciona, esto es, su funcionabilidad (por ejemplo: vuela, rueda, navega, ladra, maulla, suma, resta, calcula, etc.).

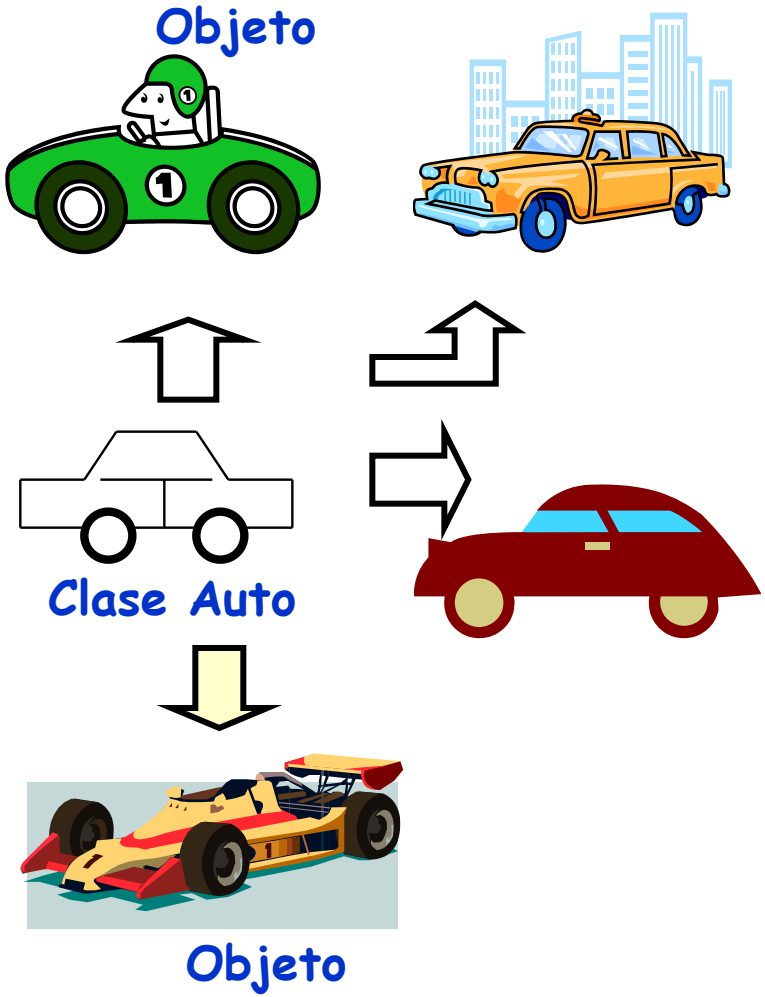
EJEMPLO: LA CLASE AUTO

- La clase AUTO es el modelo abstracto del concepto de un Auto.
- La descripción de la clase Auto podría ser:
  - Tiene Puertas
  - Tiene Llantas
  - Tiene Motor
  - Tiene Ventanas
  - Tiene modelo
  - Arranca
  - Frena
  - Se desplaza
  - Usa gasolina
  - Etc.



Una **clase** es como un **molde** o **plantilla**

LA CLASE AUTO



- A partir de la clase AUTO se pueden crear muchos objetos, o sea muchos autos, con características diferentes (color, tamaño, diseño, material, etc.), pero que pueden ser reconocidos como autos.
- Por ejemplo: un Bochito, un Audi, un BMW, un Chevy, el auto del vecino, tu auto, mi auto, etc.
- Todos estos pueden ser representadas como objetos diferentes y únicos de la clase AUTO.

CLASES V.S. OBJETOS

- Una clase en una definición abstracta de un objeto
  - Define la estructura y comportamiento de cada objeto en la clase
  - Sirve como una plantilla para crear objetos
- Un objeto es una instancia concreta de una clase (un ejemplar)
  - Los objetos pueden agruparse en clases

A. Pineda

E. Gomez

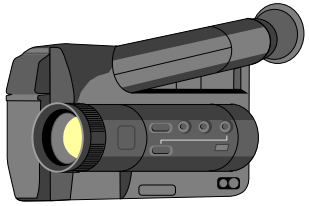
G. Rodríguez

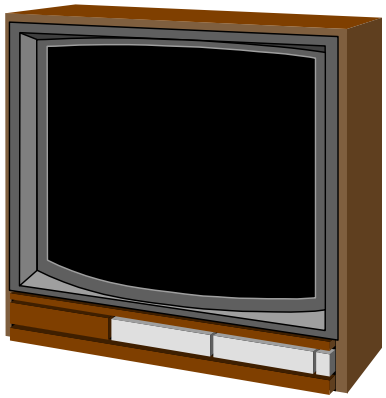
Clase

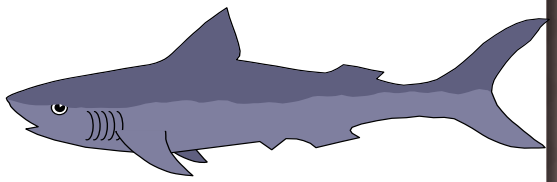
Estudiante

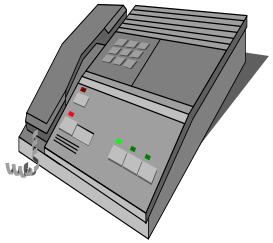
CLASES V.S. OBJETOS

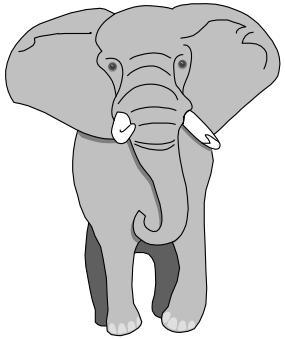
- ¿Cuántas clases ves?





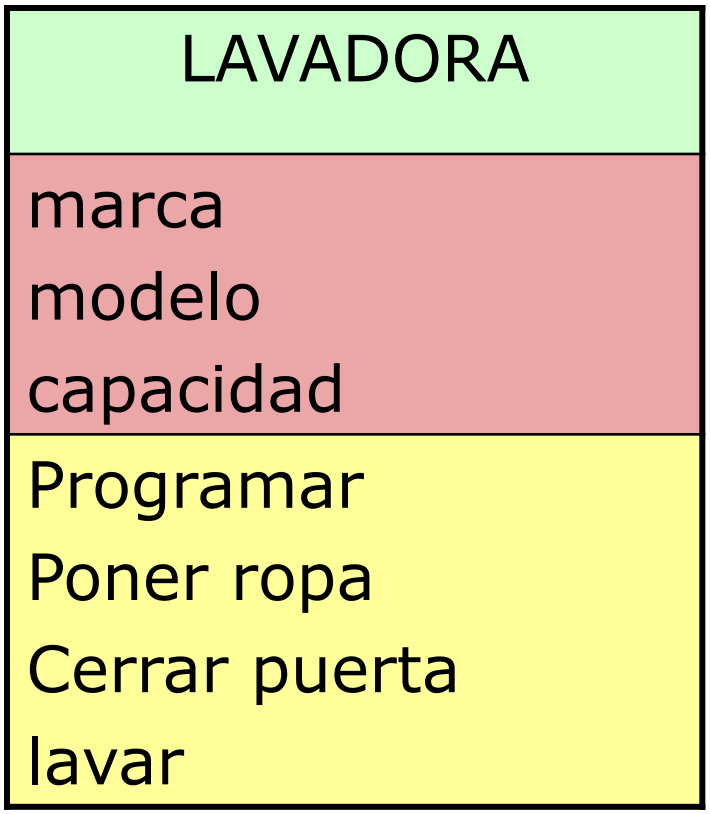






EJEMPLO DE UNA CLASE

- Modelar la clase lavadora



Atributos

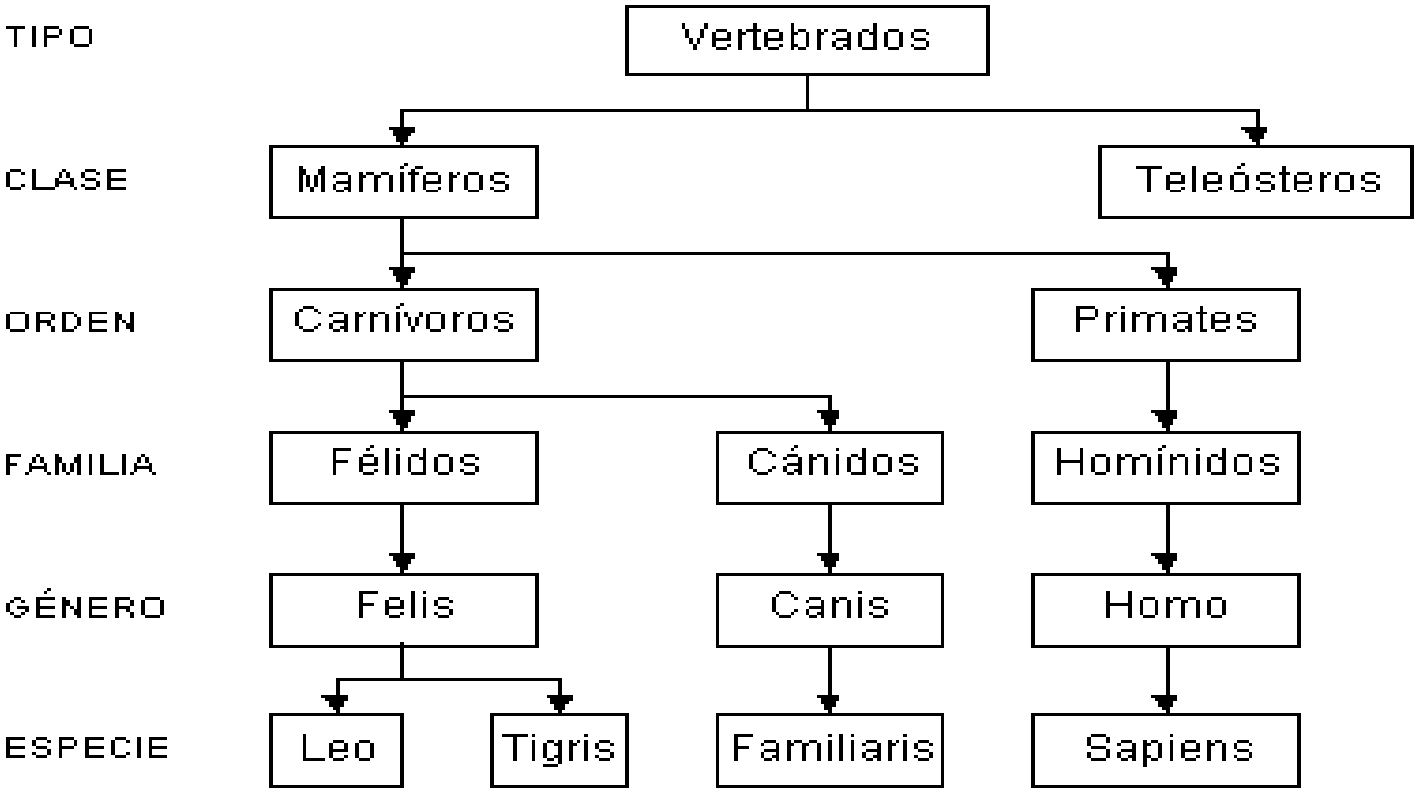
Comportamiento

HERENCIA

- La herencia representa una jerarquía de abstracciones (clases).
- Básicamente, la herencia define una relación entre clases, el la que una clase comparte la estructura de comportamiento definida en una o más clases
- De forma simple, la herencia es el proceso mediante el cual un objeto adquiere las propiedades de otro.

HERENCIA

Clasificación Taxonómica de los Vertebrados



HERENCIA

- Cada vez que se especializa una clase, esta clase hereda atributos y comportamientos de su superclase; pero además se añaden nuevos comportamientos o se modifican alguno de los ya heredados
- Veamos que hereda la clase leo de sus clases padre.

CLASE	QUE HEREDA
Vertebrados	Espina dorsal
Mamíferos	Se alimenta con leche materna
Carnívoros	Al ser adulto se alimenta de carne
Leo	Agrega: tipo y color de piel

HERENCIA

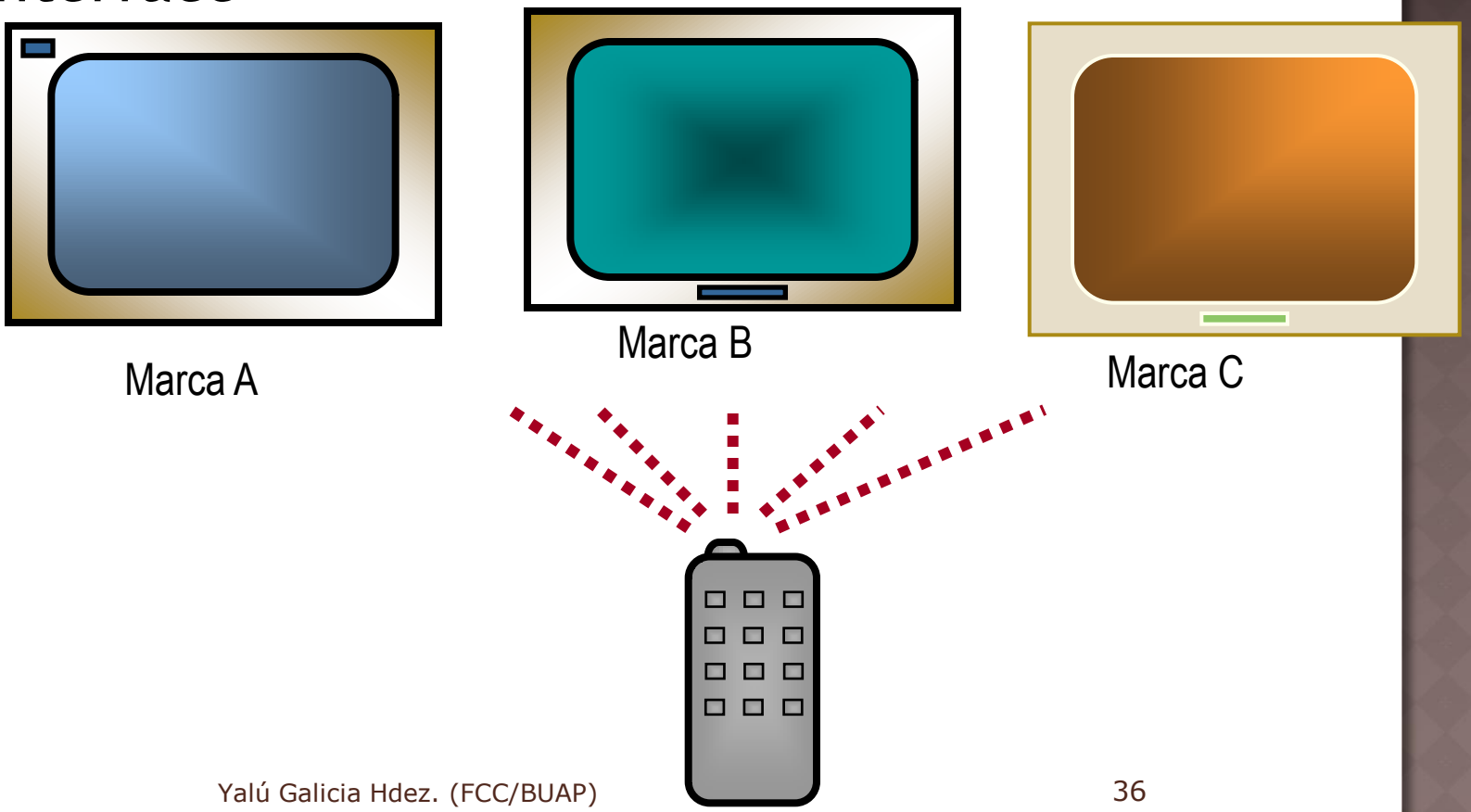
- Semánticamente, la herencia denota una relación “es un”.
- Por ejemplo, un oso *es un* tipo de mamífero, una casa *es un* tipo de bien inmueble.
- Así la herencia implica una jerarquía de generalización/especialización, en la que una subclase especializa el comportamiento o estructura más general de sus superclases.

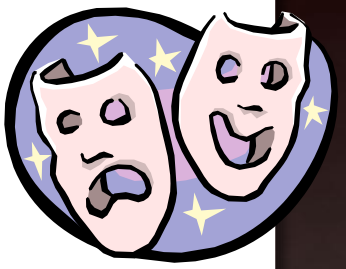
POLIMORFISMO

- Polimorfismo es la habilidad que adquieren los objetos de responder en forma diferente al mismo mensaje.
- Es decir, el mismo mensaje que se envía a muchos tipos de objetos, toma “muchas formas” y de ahí viene el término polimorfismo

POLIMORFISMO

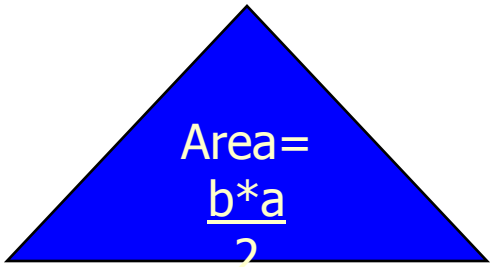
- Polimorfismo es la habilidad de esconder diferentes implementaciones tras una sola interface

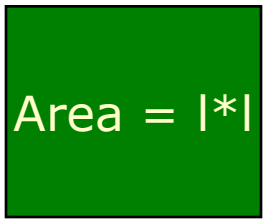


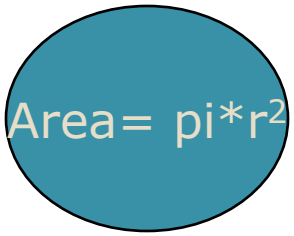


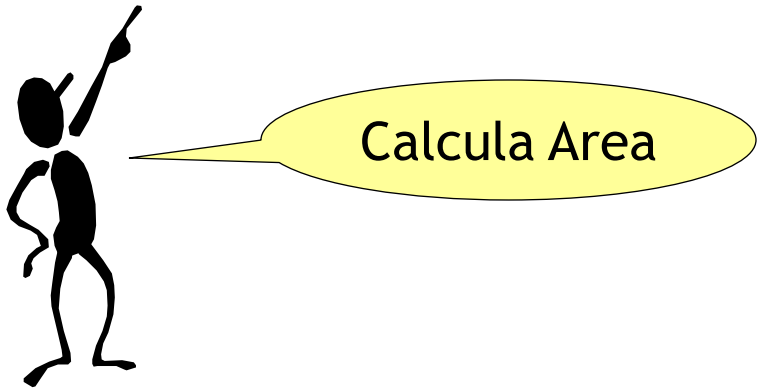
EJEMPLO DE POLIMORFISMO

- Pensemos en las FigurasGeometricas. Todas las FigurasGeometricas tienen como característica el poder calcular su área. Sin embargo, cada figura puede realizar esta operación de forma distinta.



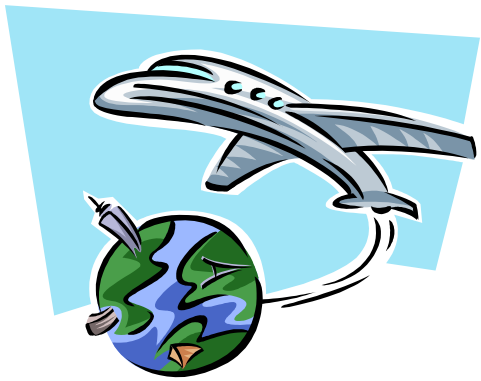






OTRO EJEMPLO,

- Tomemos Medios de Transportes: barco, avión y auto. Si les enviamos el mensaje *Desplázate*, cada uno de estos objetos los hará a su manera.



¿QUE HEMOS APRENDIDO?