

# Motivación

Juan Carlos Figueroa Duran

Otoño-2014

Universidad del Bío-Bío  
Departamento de Ciencias de la Computación y Tecnología de la  
Información

# Temario

- 1 La necesidad
- 2 Concepto de programa y lenguajes
- 3 Tipos de datos
- 4 Variables y constantes
- 5 TDA
- 6 Programación estructurada
- 7 Programación Orientada a Objetos
- 8 Conceptos básicos



# Continuación computar ...

- Viene del latín *computare*, que significa calcular (nos persigue el calculo). Para esto se creó con mucho esfuerzo, una máquina electrónica que recibe y procesa datos para convertirlos en información útil. La computadora.
- Adicionalmente, hoy podemos realizar “calculo”(operaciones) lógicas.

Totalmente portátil...



# ¿Que es un sistema?

## Sistema:

Es un conjunto de elementos o partes que se integran o relacionan para producir un resultado.

## Ejemplos:

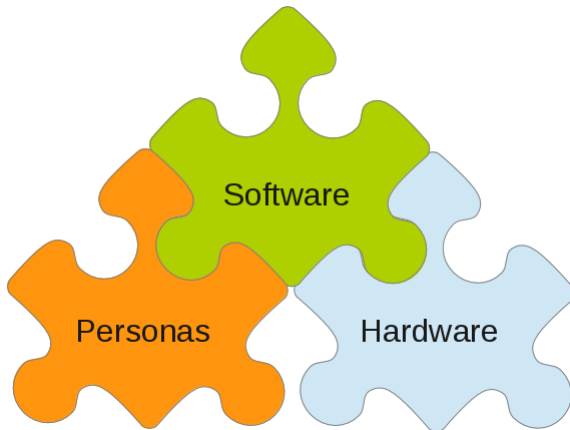
- Una planta.
- El sistema solar.
- El cuerpo humano.
- Computador.

¿Haber pero donde entramos nosotros?

Alguien le tiene que decir al computador que calcular y como calcular.

Así, un sistema basado en computadores o sistema software, es un conjunto de ciertas entidades caracterizadas por ciertos atributos (ej: clientes, productos y vendedores de una empresa y sus respectivos datos de interés) que se relacionan entre sí en un contexto determinado (ej: ventas de una empresa) y tienen como objetivo producir información (ej: control de las ventas y ganancias de una empresa).

Los sistemas basados en computadores contemplan 3 elementos esenciales:



¿Quienes son los usuarios?



Usuario cliente



Usuario final



## Concepto

“Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico”

## ¿Qué haremos nosotros sobre usabilidad?

- El sistema a desarrollar debe cumplir el objetivo para el cual fue creado.
- La interfaz que el sistema ofrece a sus usuarios debe ser comprensible y fácil de usar.

- Siempre que se presente un dato al usuario debe indicarse de qué dato se trata
- Las tareas que puede realizar con el sistema deben mostrarse siempre de la misma forma.
- Si el usuario debe ingresar datos debe indicársele claramente qué datos debe ingresar y cómo.
- Si el usuario ingresa datos erróneos el sistema debe informarle que ha cometido un error.
- Cuando una tarea se ejecuta, debe indicarse al usuario cuándo esta tarea ha terminado.

# Concepto de programa y lenguajes

## ¿Qué es un programa?

Conjunto de instrucciones que controlan el funcionamiento de un computador, dichas instrucciones deben ser almacenadas dentro del computador (en la memoria), el cual las ejecuta automáticamente.

## Lenguajes de programación

Dan la capacidad a los programadores para escribir los programas con las instrucciones que se desea que un computador ejecute.

- Lenguajes de programación
  - Lenguajes de máquina, que tienen una representación críptica, de tipo binaria
  - Lenguaje ensamblador, lenguajes de máquina simbólicos.
  - Lenguajes de alto nivel, lenguajes para escribir instrucciones más cercanos al hombre. Ejemplos:FORTRAN, COBOL, PASCAL, C, Java, C++, PHP, Python, Ruby, entre otros.

# Tipos de datos

## Tipos de datos

Son conjuntos de elementos que cumplen con unas características bien definidas, de manera que es posible determinar si un elemento dado pertenece a él o no.

## Tipos de datos primitivos

Comunes a casi todos los lenguajes de programación. Su característica principal es que sus instancias son valores. Un tipo de dato primitivo es la definición de un grupo de valores y de las operaciones que existen para trabajar con ellos, por ejemplo: boolean, double.

# Tipos de datos

Tipo de datos	Operaciones	Nombre	Tamaño	Descripción
Boolean	Operaciones lógicas	boolean	1 bit	Comprende aquellos valores empleados para definir el valor de verdad de algo: falso y verdadero. Específicamente, en Java, se manejan los valores false y true
Caracter	Operaciones de relación y de asignación Aritméticas, de relación y de asignación	char	16 bits	Contiene la posición en la cual se encuentra cada caracter en la tabla Unicode1, es decir comprende valores entre 0 y 65536.
Numéricos		byte	8 bits	Comprende el rango de valores enteros entre 128 y -127. Lo anterior, se debe a que sólo usa 8 bits de memoria para guardar un valor, de ahí su nombre, pues 8 bits forman un byte
		short	16 bits	Definido para valores enteros, y el rango que puede albergar está entre $-2^{15} \dots 2^{15} - 1$
		int	32 bits	Definido para valores enteros, y el rango que puede albergar está entre $-2^{31} \dots 2^{31} - 1$
		long	64 bits	Definido para valores enteros, y el rango que puede albergar está entre $-2^{63} \dots 2^{63} - 1$
		float	32 bits	Definido para valores de tipo real, y el rango que puede albergar está entre $-10^{32} \dots 10^{32}$ , aprox
		double	64 bits	Definido para valores de tipo real, y el rango que puede albergar está entre $-10^{300} \dots 10^{300}$ , aprox

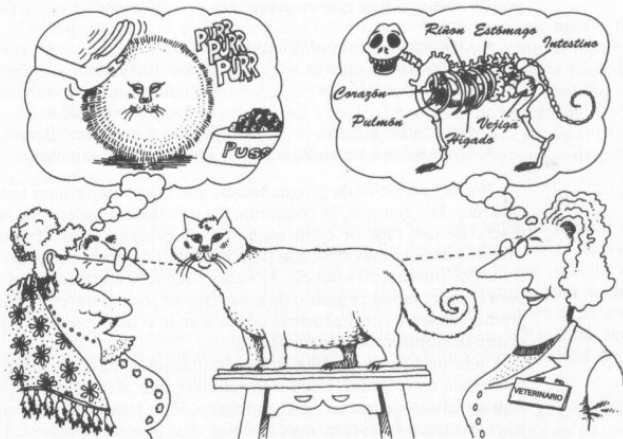
## ¡Me ayudan a definir!

- Variable
- Constante
- instrucción y sus tipos



- En un lenguaje de programación (por ejemplo Java), no se puede definir un nuevo tipo de dato primitivo.
- Sin embargo, en la mayoría de ellos sí es posible definir lo que se denominan tipos de datos abstractos (TDA).

## Abstracción



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

- Son conjunto de valores y un grupo de operaciones sobre dichos valores.

## Definition

Tipo de dato, que se agrega al lenguaje de programación con el fin de representar un tipo de elemento involucrado en el problema que se quiere resolver. De esta forma se hace que el lenguaje se acerque al mundo del problema, manejando los elementos que allí se encuentran.

- Estructura compuesta por un conjunto de objetos abstractos, que modelan elementos del mundo, y un conjunto de operaciones para su manipulación.
- En Java los TDA se conocen como tipos referenciados y a sus instancias se les denomina objetos.

Ejemplo: TDA Racionales  $\frac{x}{y}$

- **Nombre, valor, invariante**

TDA Racional

numerador, denominador enteros

condición: *denominador*  $\neq 0$

- Operadores

- CrearRac(int x, int y) /\* retorna un racional \*/  
 {precondición :  $y \neq 0$ }  
 {postcondición :  $numerador = x, denominador = y$ }
- Numerador(Racional a) /\* retorna un entero \*/  
 {postcondición :  $numerador$ }
- Denominador(Racional a) /\* retorna un entero \*/  
 {postcondición :  $denominador$ }
- SumarRac(Racional a, Racional b) /\* retorna un racional \*/  
 {postcondición :  $denominador =$   
 $Denominador(a) * Denominador(b), numerador = Numerador(a) * Denominador(b) + Numerador(b) * Denominador(a)$ }

- Estilo de codificación (programación) libre del salto incondicional, cuyos objetivos esenciales son:
  - Mejorar la fiabilidad de los programas
  - Mejorar la legibilidad de los programas
  - Minimizar la complejidad de los programas
  - Simplificar el mantenimiento de los programas
  - Incrementar la productividad de los programadores
  - Proveer una metodología de programación disciplinada
- Hoy en día la PE es el estilo básico que todo programador debe conocer.

- Es una técnica clásica relacionada con el tratamiento de la complejidad.
- Se centra en la descomposición de un problema complejo en problemas más pequeños (módulos) que se puedan resolver por separado.
- Un módulo se puede usar sin conocer más que la función que realiza y la información que hay que suministrarle para trabajar. Con esto se logra una abstracción procedimental (para usarlo no se necesita conocer sus detalles internos).

- El enfoque orientado a objeto toma elementos del enfoque estructurado (pues dicho enfoque tiene aspectos relevantes e insustituibles), pero plantea que un problema se puede caracterizar como un conjunto de conceptos o entidades interactuando, cada una de estas entidades es parte datos y parte funciones.



- En este enfoque, para la solución de un problema lo que corresponde es descubrir cuáles son dichas entidades o conceptos, cuál es su descripción y en qué forma interactúan entre ellos.
- Tarea no necesariamente simple, pero que permite superar algunas de las deficiencias del enfoque estructurado, sobre todo tratándose de problemas grandes y complejos.

- La PE plantea que el desarrollo de Sw requiere una descripción basada en funciones y una descripción basada en datos, describir un problema basándose en los datos y las funciones constituye analizar el problema bajo dos puntos de vista. Luego, la relación entre las descripciones que se obtienen (que se hacen con técnicas diferentes) resulta difícil, y, de todas maneras, una visión primará sobre la otra.
- En la POO hay una mezcla de ambos puntos de vista.

- La POO es un paradigma de programación poderoso y natural para crear programas que sobreviven a los cambios inevitables que acompañan al crecimiento y envejecimiento de cualquier software.

- **Clase:**

- Una clase es un tipo de dato abstracto equipado con una implementación posiblemente parcial.
- Una clase es una agrupación de objetos de un mismo tipo, que tienen un conjunto de características y/o un comportamiento común.
- Una clase es una representación abstracta de algo.

- **Objeto:**

- Un objeto es una instancia de alguna clase.

## Ejemplo: Gallina



- Tiene características como: color, peso, edad, ubicación
- Estas características definen su estado, y la variación de alguna de ellas altera su estado.
- Puede realizar ciertas tareas como: cacarear, picar, poner huevos...
- Las tareas definen el comportamiento de la gallina

Con el estado y el comportamiento podemos conocer que le pasa a la gallina en un instante dado. Así, cualquier objeto físico queda definido por su estado y comportamiento.

Otro Ejemplo Gatito: Definamos sus características y operaciones genéricas.



- Características: raza..
- Operaciones: maullar..

- Objetos: Para simular una gallina/gato en un programa:
  - Necesitaremos datos (variables) para indicar su estado
  - Y métodos que alteren esos datos para representar su comportamiento.
- Si queremos programar un gallinero, la gallina sería un objeto que posee variables que almacenan su edad, posición, etc.; y métodos que la representan en una posición en el gallinero y calculan su producción.
- Las variables y métodos definidos establecen el estado y comportamiento que es relevante para el problema
- De hecho los requisitos del programa indican los datos y métodos necesarios.

- Objetos:
- Un programa escrito según el paradigma OO, consiste en una serie de objetos que interaccionan entre sí.
- Un objeto es toda entidad activa de un programa. Un objeto queda definido por las siguientes propiedades:

**Estado:** Situación de un objeto en un momento determinado. Los valores que se almacenan de un objeto se llaman atributos

**Comportamiento:** Manera en que el objeto reacciona a los mensajes recibidos. Las distintas formas de respuestas se denominan métodos.

**Identidad:** Lo que diferencia a un objeto de otros objetos.



# Conceptos básicos POO

Ejemplo

Clase: Perro

Atributos:

- Sexo
- Color
- Peso

Métodos:

- Mover la cola.
- Correr.
- Oler la entre pierna.

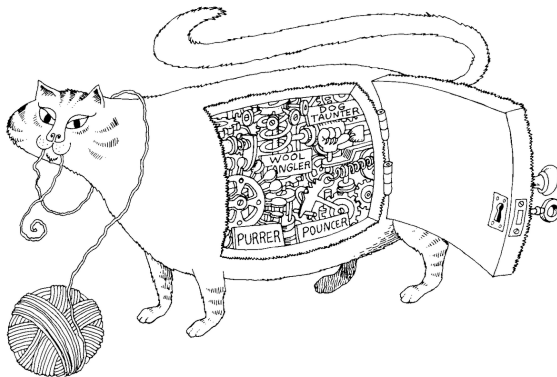
Me dan alguna instancias de los chiguaguas.



La POO consta de tres mecanismos fundamentales:

- Encapsulado
- Herencia
- Polimorfismo

## Encapsulado



El encapsulado oculta los detalles de la implementación de un objeto

## Encapsulado

- La encapsulación aplica el principio de abstracción: los objetos son accesibles mediante sus operaciones visibles (interfaz externa) , y su implementación (estructura de datos y código) queda oculta.

Clase: Perro

Atributos:

- Sexo
- Color
- Peso

Métodos:

- Mover la cola.
- Correr.
- Oler la entre pierna.

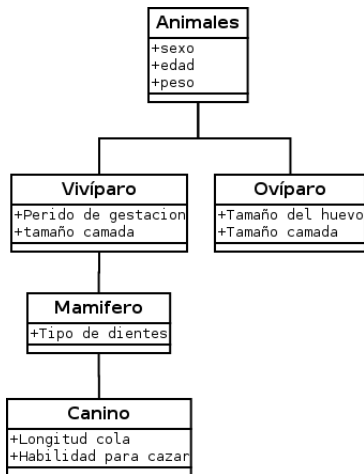
Objeto: Chiguagua

- Sexo=Macho
- Color=Café
- Peso=1 kilo

## Herencia

- Las clases se relacionan entre ellas en forma jerárquica.
- De este modo podemos tener clases y subclases.
- Una subclase hereda todos los atributos de la clase o las clases que la anteceden.
- Esto significa que si una clase dada encapsula algunos atributos, entonces cualquier subclase tendrá los mismos atributos más cualquiera que añada como parte de su especialización.

# Conceptos básicos POO



- Las operaciones heredadas pueden modificarse para ajustarlas a las necesidades particulares de la clase que hereda.
- La clase que hereda debe definir nuevos atributos y/o nuevas operaciones.

## Polimorfismo

- A los métodos que actúan sobre los objetos se les pasa información en forma de parámetros en la llamada al método.
- Estos parámetros representan valores de entrada a una función que debe realizar el método.
- El polimorfismo significa un objeto y muchas formas.
- Esto es, un método puede tener múltiples implementaciones que se seleccionan en base al tipo de objeto sobre el que se aplicará el método o bien por la forma en que se invoca al método.

## Ejemplo polimorfismo

- Suponga que tenemos la clase perro, el método oler con un parámetro de la clase gato y el método oler con un parámetro de la clase comida.
- Si en la llamada pasamos un objeto de la clase gato, el comportamiento del perro sería ladrar.
- Si en la llamada pasamos un objeto de la clase comida, el comportamiento del perro será salivar y correr hacia el lugar donde está la comida.