

# Elementos, estructura y funcionamiento de un sistema de cómputo + Lenguajes de Programación

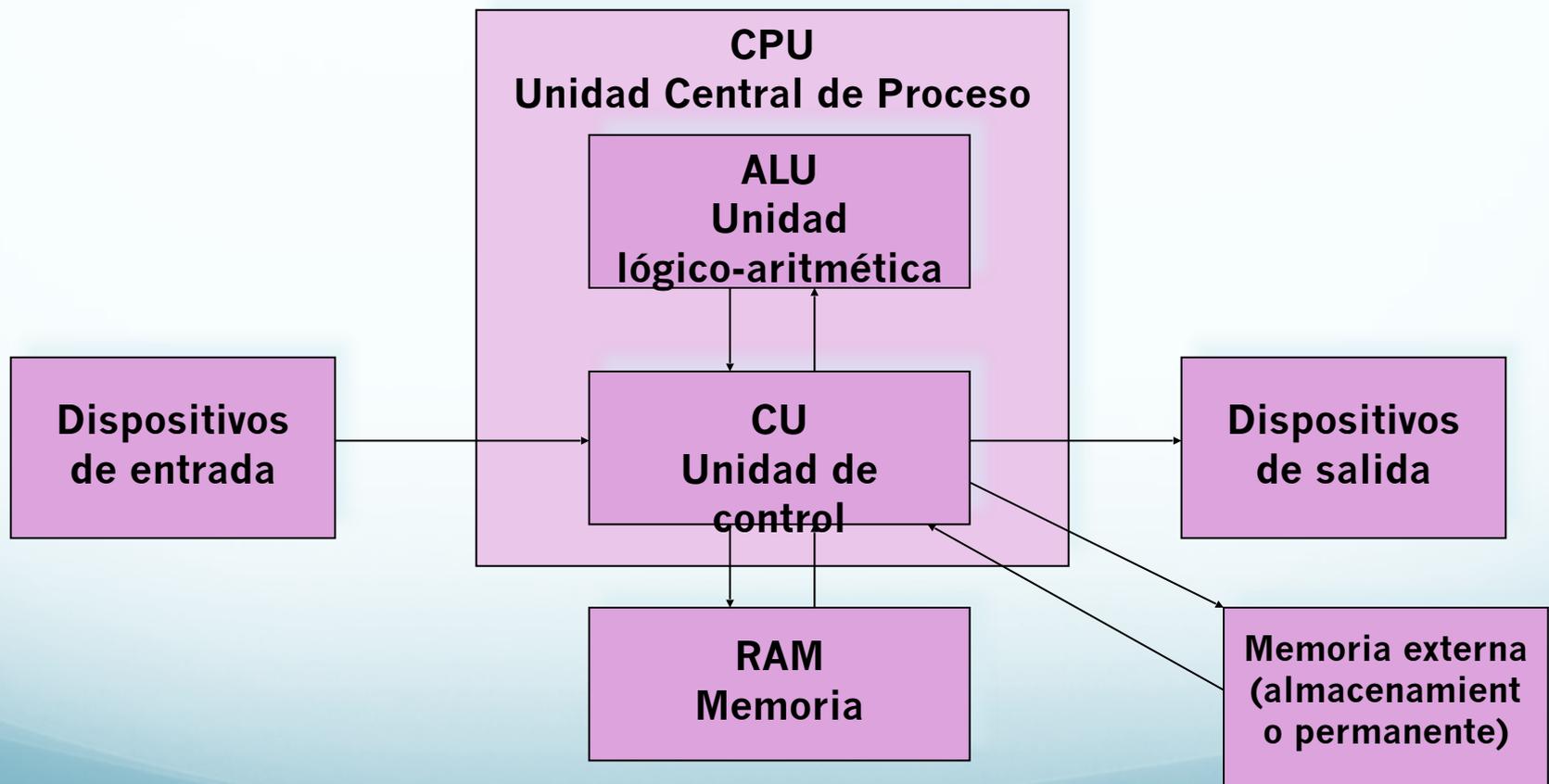
Ana Lilia LAUREANO CRUCES

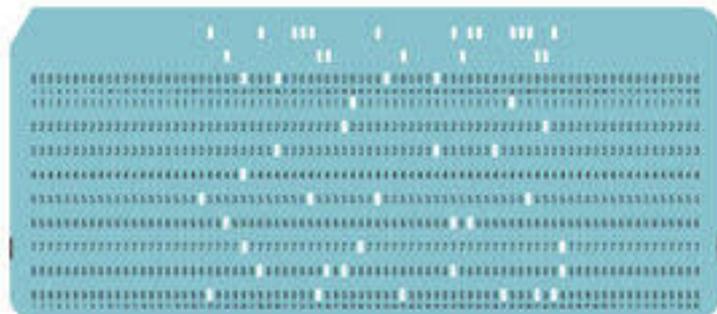
# Diferentes tipos de Código

- Código de máquina es ininteligible:
  - 00000010101111001010
  - 00000010111111001000
  - 00000011001110101000
- Código ensamblador
  - LOAD I
  - ADD J
  - STORE K
- Código en FORTRAN
  - $k = i + j$

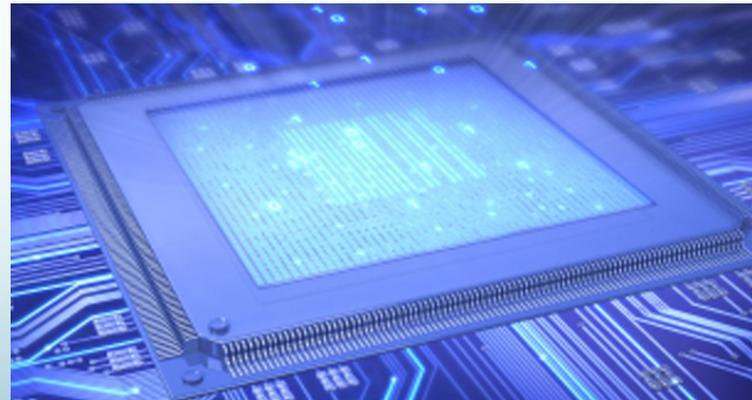
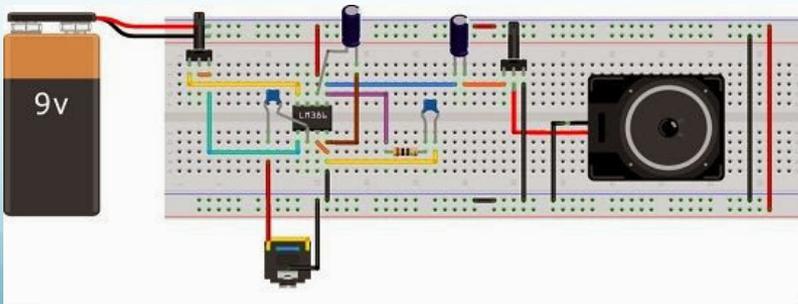
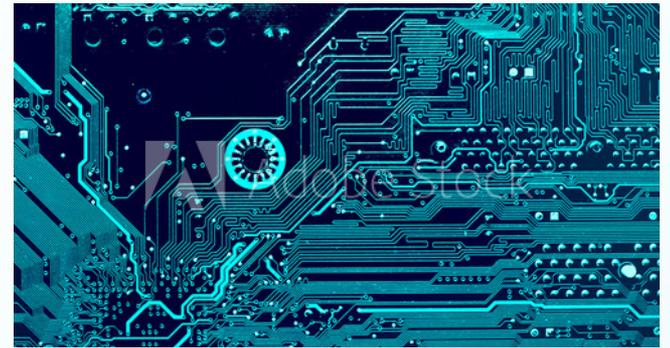
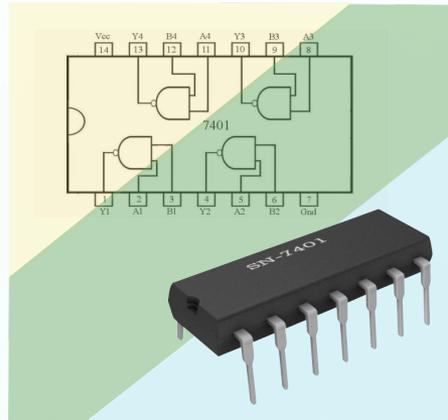
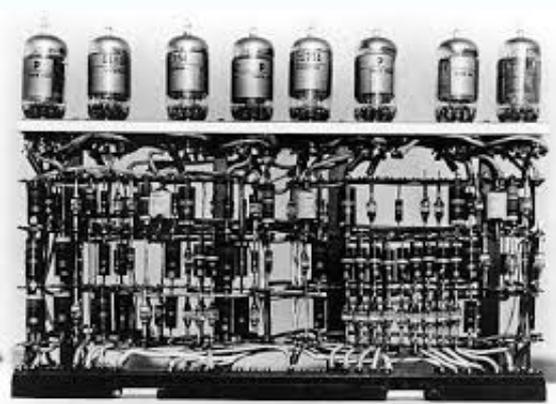
# Elementos, estructura y funcionamiento de un sistema de cómputo

## Organización física de una computadora





# Evolución de los Circuitos



# Elementos, estructura y funcionamiento de un sistema de cómputo

## Funcionamiento de una computadora

### **La unidad central de proceso (del inglés, CPU)**

La unidad central de proceso (del inglés, CPU) es el “cerebro” de la computadora. La CPU dirige y controla el proceso de información efectuado por la computadora, es la unidad donde se ejecutan las instrucciones de los programas y donde se controla el funcionamiento de los distintos componentes de la computadora. La CPU procesa información almacenada en la memoria, puede recuperar información desde la memoria (datos o programas) y almacenar nueva información o información procesada en la memoria. La CPU está compuesta por dos componentes: la unidad de control (CU) y la unidad lógico-aritmética (ALU).

# Elementos, estructura y funcionamiento de un sistema de cómputo

## Funcionamiento de una computadora

### **La unidad de control (CU)**

La unidad de control (CU) dirige y coordina las actividades de la computadora, determina la secuencia de operaciones a realizar, interpreta instrucciones codificadas, controla las funciones de decodificación, ejecución y almacenamiento. Esta unidad controla y sincroniza todo el proceso que tiene lugar en la computadora.

### **La unidad lógico-aritmética (ALU)**

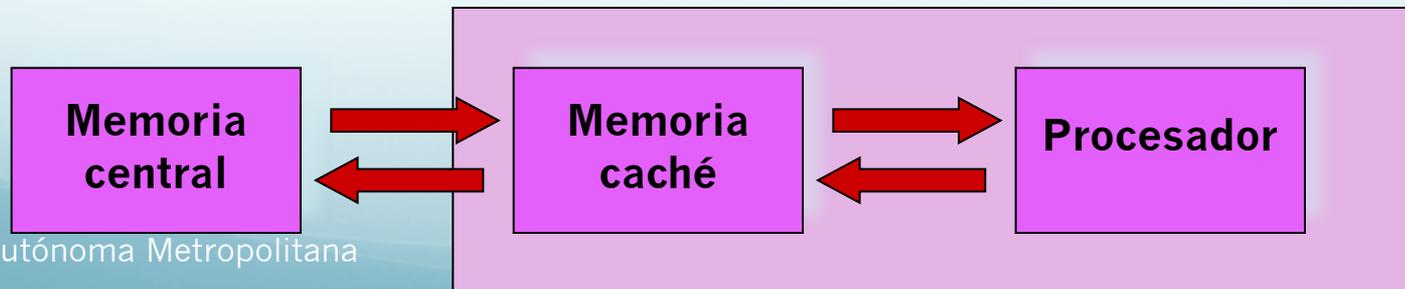
Como su nombre lo indica, la unidad lógico-aritmética es la encargada de realizar operaciones aritméticas y lógicas (suma, resta, multiplicación, división y comparaciones) sobre datos que provienen de la memoria principal (RAM).

# Elementos, estructura y funcionamiento de un sistema de cómputo

## Funcionamiento de una computadora

### La memoria central (RAM)

El papel de la memoria central (memoria RAM, *Random Access Memory*) es el almacenamiento de información: datos y programas. Para que un programa se pueda ejecutar, debe ser almacenado en la memoria central (carga o *load* del programa). También los datos que serán usados por el programa durante su ejecución deben ser situados en la memoria. Es decir, la memoria RAM se usa para mantener los programas mientras se están ejecutando y los datos mientras se están procesando.



# Elementos, estructura y funcionamiento de un sistema de cómputo

## Funcionamiento de una computadora

### Dispositivos de entrada/salida

Los dispositivos de entrada/salida establecen la comunicación entre la computadora y el usuario. Los dispositivos de entrada (teclado, mouse, unidad CD-ROM, unidad DVD, escaner, cámara, etc.) permiten la introducción de datos y programas en la computadora para su procesamiento. Una vez que los datos son leídos de los dispositivos de entrada, éstos son almacenados en la memoria central para su procesamiento. Los dispositivos de salida (pantalla, impresora, etc.) permiten representar/visualizar el resultado de la ejecución de los programas.

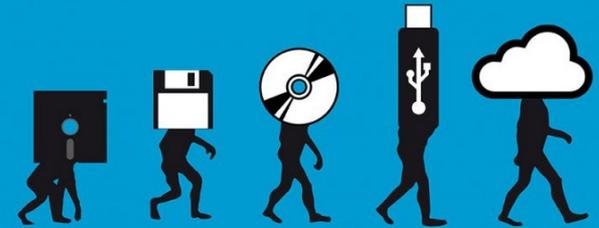
# Elementos, estructura y funcionamiento de un sistema de cómputo

## Funcionamiento de una computadora

### La memoria externa

La memoria externa es un medio de almacenamiento permanente tanto para programas como para datos. Entre los dispositivos de almacenamiento (o memorias auxiliares) más comúnmente utilizados se encuentran:

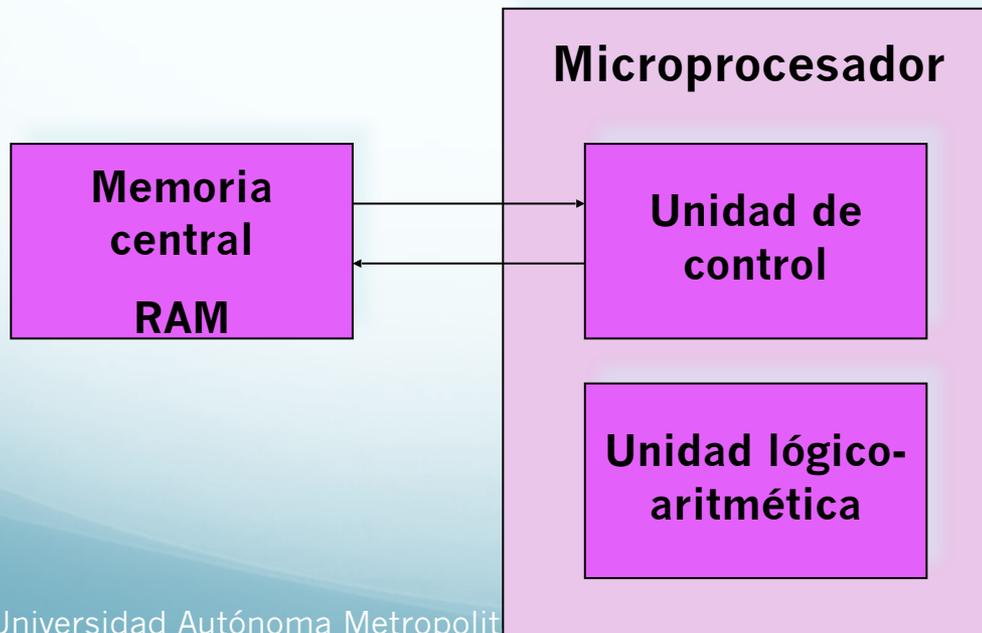
- El disco duro
- Disquetes magnéticos
- Unidad CD-ROM
- Unidad DVD
- Memoria USB



# Elementos, estructura y funcionamiento de un sistema de cómputo

## El microprocesador

El microprocesador es un chip, un circuito integrado electrónico que es utilizado como unidad central de proceso en una computadora. Las funciones del microprocesador son el cálculo y control computacional.



# Lenguajes de programación: niveles, sintaxis y semántica

## Niveles de lenguajes de programación



# Lenguajes de programación: niveles, sintaxis y semántica

## Niveles de lenguajes de programación

### Lenguajes de máquina

Fue el primer lenguaje utilizado en la programación de computadoras. Constituye la capa más interna en los niveles de lenguaje en la computación. Se podría decir que el lenguaje de máquina es el único que entiende directamente la computadora. Este lenguaje se expresa a través del alfabeto binario, el cual consta solo de los símbolos 0 y 1. El lenguaje de máquina es dependiente de la computadora.

### Lenguajes ensambladores

El lenguaje ensamblador es una representación simbólica del lenguaje de máquina asociado. Es decir, cada computadora tiene un lenguaje ensamblador propio. El lenguaje ensamblador usa palabras nemotécnicas para representar cada instrucción (cadenas de bits) del lenguaje de máquina asociado.

# Lenguajes de programación: niveles, sintaxis y semántica

## Niveles de lenguajes de programación

### Lenguajes de alto nivel

Constituyen los lenguajes de programación más utilizados. Son lenguajes independientes de la computadora. Un programa escrito en lenguaje de alto nivel puede ser ejecutado en cualquier computadora, siempre que ésta disponga del traductor o compilador del lenguaje en cuestión. Los programas pueden ser escritos y comprendidos de una forma más sencilla, dada la aproximación de estos lenguajes con el lenguaje natural. Se caracterizan por proporcionar librerías de uso frecuente (entrada/salida, funciones matemáticas, manejo de cadenas, etc.). Ejemplos de estos lenguajes son: Pascal, Fortran, C, C++, Java, Ada, Lisp y Prolog.

### Lenguajes declarativos

Son aquellos lenguajes en los que se indica a la computadora qué es lo que debe hacer, qué es lo que se desea obtener o qué es lo que se está buscando. Son comúnmente lenguajes de órdenes, donde se expresa lo que hay que hacer en lugar de cómo hacerlo. Ejemplos de lenguajes declarativos son SQL y Prolog.

# Lenguajes de programación: niveles, sintaxis y semántica

Lenguaje de programación =  
Léxico + Sintaxis + Semántica

# Lenguajes de programación: niveles, sintaxis y semántica

Aspectos a considerar en el diseño e implementación de los lenguajes de programación

- Léxico
- Sintaxis
- Semántica
- Gestión de memoria
- Manejo de excepciones
- Implementación
- Pragmática

# Lenguajes de programación: niveles, sintaxis y semántica

Aspectos a considerar en el diseño e implementación de los lenguajes de programación

## Léxico

Es el conjunto de categorías gramaticales que definen el vocabulario del lenguaje de programación. Comúnmente, el léxico de un lenguaje de programación posee las siguientes categorías gramaticales:

- *Identifier* (nombre de variables, nombre de constantes, nombre de funciones, etc.)
- *Literal* (números enteros y decimales)
- *Operator* (+, -, \*, /, etc.)
- *Separator* (; , etc.)
- *Keyword* (while, read, write, if, for, case, int, real, etc.)

# Lenguajes de programación: niveles, sintaxis y semántica

Aspectos a considerar en el diseño e implementación de los lenguajes de programación

## Sintaxis

La sintaxis de un lenguaje de programación es la definición gramatical de lo que constituye un programa en dicho lenguaje. La sintaxis se especifica en términos de un conjunto de reglas que rigen la construcción de los programas. La definición de la sintaxis de un lenguaje de programación debe ser clara, concisa y formal. La sintaxis del lenguaje de programación se describe a partir de otros lenguajes, llamados metalenguajes. Uno de los metalenguajes más difundidos es el formalismo BNF (Backus-Naur Form), una gramática libre de contexto.

# Lenguajes de programación: niveles, sintaxis y semántica

Aspectos a considerar en el diseño e implementación de los lenguajes de programación

## **Semántica**

Se refiere al significado de los programas. Es el conjunto de reglas que permiten determinar el significado de cualquier construcción del lenguaje. La semántica asocia un significado a cada posible construcción gramatical del lenguaje.

## **Gestión de memoria**

La gestión de memoria se refiere al proceso de asignar valores a la memoria, considerando tanto las características estáticas como dinámicas de dichos valores. Los valores pueden ser asignados a una de tres categorías de memoria: memoria estática, la pila en tiempo de ejecución y el montículo.

# Lenguajes de programación: niveles, sintaxis y semántica

Aspectos a considerar en el diseño e implementación de los lenguajes de programación

## **Manejo de excepciones**

Una excepción es una condición detectada por una operación la cual no se puede resolver en el contexto de la operación. Dos procesos básicos se asocian al manejo de excepciones: (1) lanzamiento de la excepción y (2) captura de la excepción. El lanzamiento de una excepción es señalar que se ha producido una excepción, mientras que la captura de la excepción consiste en proporcionar una forma de manipular dicha excepción, transfiriéndose el control del programa al manipulador de la excepción.

# Lenguajes de programación: niveles, sintaxis y semántica

Aspectos a considerar en el diseño e implementación de los lenguajes de programación

## **Implementación**

Cómo se ejecutan los programas.

## **Pragmática**

Se refiere a los aspectos prácticos del uso de los programas.

# Paradigmas de programación

- Programación imperativa
- Programación funcional
- Programación lógica
- Programación orientada a objetos
- Programación concurrente
- Programación guiada por eventos

# Paradigmas de programación

## Programación imperativa

El programa se define como una serie de acciones o pasos, cada uno de los cuales recibe una entrada, ejecuta un cálculo, o produce como resultado una salida. La programación imperativa se basa en la asignación de valores, en la utilización de variables para almacenar valores y en la realización de operaciones con estos valores almacenados. Ejemplos de lenguajes de programación imperativa son los lenguajes de alto nivel (Fortran, C, Pascal, etc.) y los lenguajes ensambladores.

## Programación funcional

Los programas están formados por una colección de definiciones de funciones. Las funciones interactúan entre sí utilizando condicionales, recursividad y composición funcional. Ejemplos de lenguajes de programación funcional son Lisp y Scheme.

# Paradigmas de programación

## **Programación lógica**

Un lenguaje de programación lógica es un tipo de lenguaje declarativo, en el cual el programa está formado por un conjunto de declaraciones lógicas (predicados, hechos). La ejecución de un programa consiste en aplicar estas declaraciones para obtener todas las soluciones posibles a un problema. El ejemplo clásico de lenguaje de programación lógica es Prolog.

## **Programación orientada a objetos**

El programa es una colección de objetos que interactúan entre sí a través del paso de mensajes, los cuales comúnmente transforman el estado de los objetos. Ejemplos de lenguajes orientados a objetos son Smalltalk, C++, Eiffel y Java.

# Paradigmas de programación

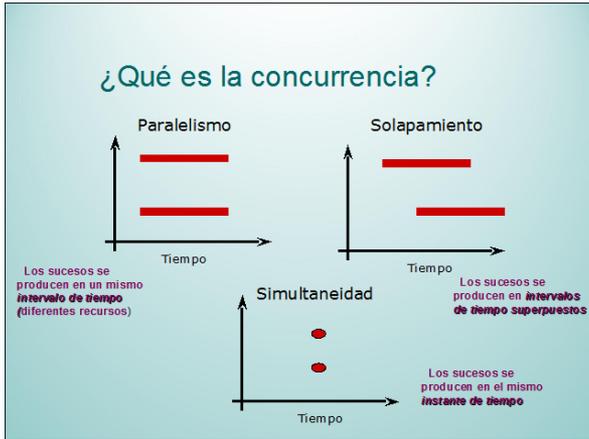
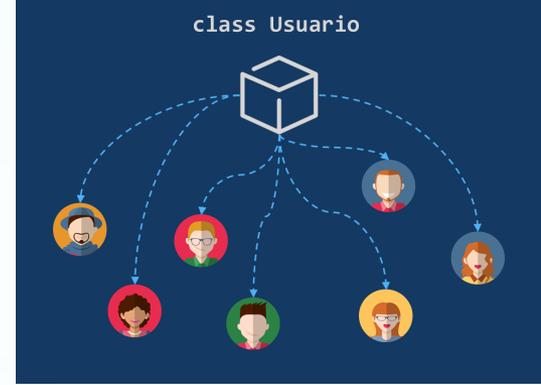
## **Programación concurrente**

La programación se basa en la ejecución simultánea de procesos, ya sea en una misma computadora con uno o varios procesadores, que en un cluster de computadoras. El programa se define como una colección de procesos cooperativos y asíncronos. Ejemplos de lenguajes concurrentes son Linda y Fortran de alto rendimiento.

## **Programación guiada por eventos**

El programa responde a eventos externos generados en un orden no predecible. Los eventos comúnmente se generan a partir de acciones del usuario en la pantalla (por ejemplo, clics del mouse o pulsaciones de teclas), pudiendo existir también otras fuentes generadoras de eventos. Ejemplos de lenguajes de programación guiada por eventos son Visual Basic, Delphi y Java.

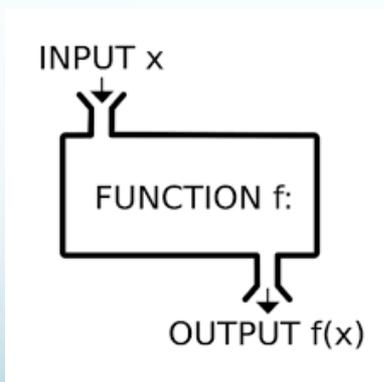
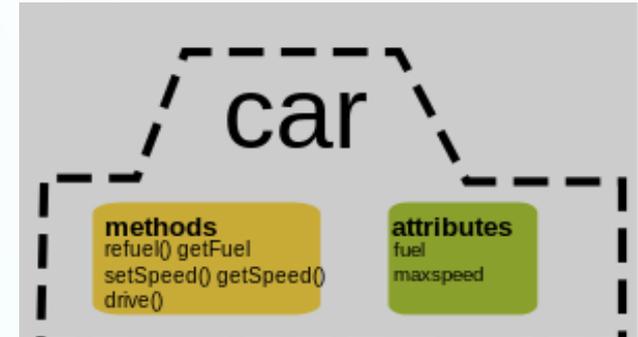
# Paradigmas



### PARADIGMA IMPERATIVO

Uno de los paradigmas más representativos de la programación y de hecho el primer paradigma formalmente aceptado es el imperativo.

Imperar significa mandar, ordenar, y eso es exactamente lo que hacemos al programar.



$$\frac{p \rightarrow q}{\therefore q}$$

# Fin de la presentación

- Gracias por su atención.
- Usted puede ampliar los conceptos vistos consultando la bibliografía, o bien algún otro medio de información donde consulte los conceptos mencionados en estas diapositivas.