

PROGRAMACIÓN LÓGICA Y FUNCIONAL

SESIÓN 01

CICLO: AGOSTO 2020



Universidad
Tecnológica
del Perú

Contenido

- Bienvenida
- Pautas de trabajo
- Introducción
- Paradigmas de programación
- Características de los paradigmas de programación
- Laboratorio

Pautas de trabajo

- Los días que tengamos clases debemos conectarnos a través de Zoom.
- La participación de los estudiantes se dará través del **chat de Zoom.**
- En Canvas encontrarán la clase de hoy, el ppt de la sesión 01, Laboratorio 01

Conceptos previos

¿ Qué se entiende por programación?

¿ Qué se entiende por lógica?

¿ Qué se entiende por función?

Levantemos la mano para participar



Logro de la sesión

Al finalizar la presente sesión el estudiante:

1. Conoce las principales características de los paradigmas de programación.
2. Conoce la definición del paradigma declarativo.



Utilidad

¿“Porque es importante conocer los paradigmas de programación”

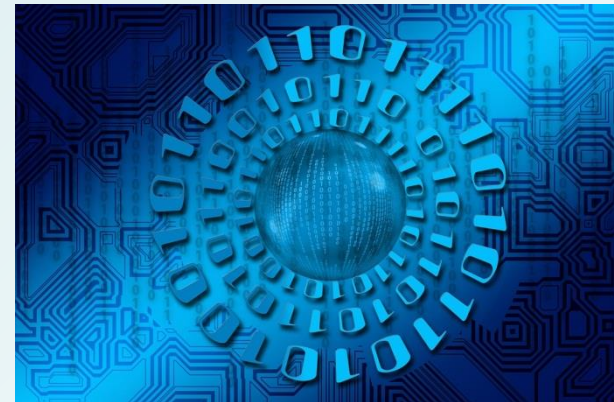


Utiliza el chat para participar



Contenido

1. Introducción
2. Paradigmas de programación
3. Características de los paradigmas de programación
4. Bibliografía



1. Introducción

¿Qué paradigmas /
tipos de
programación
conoces?

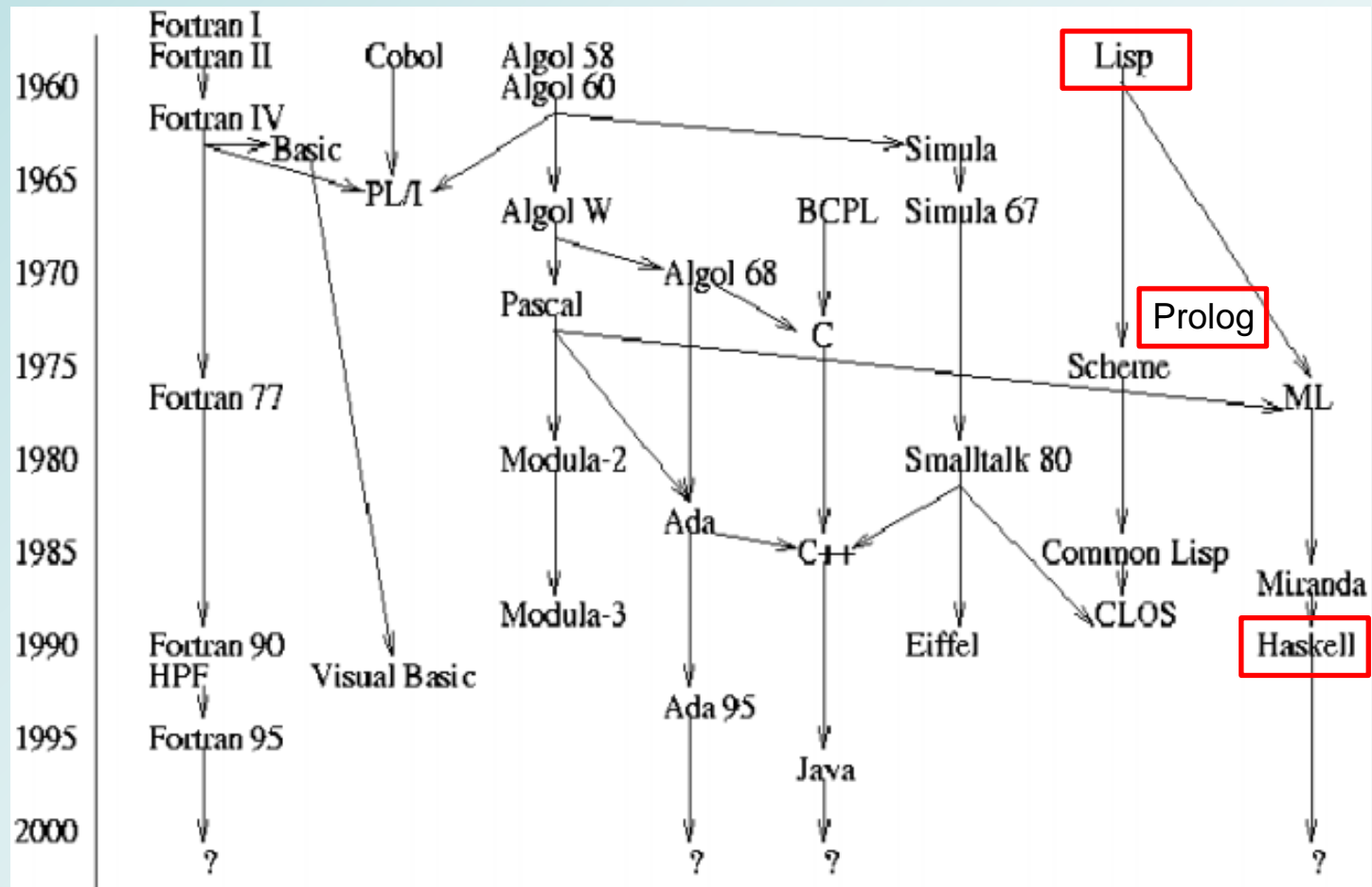


1. Introducción

- Imperativo (procedural)
- O. O.
- Declarativo
 - Funcional
 - Lógico
- Híbridos



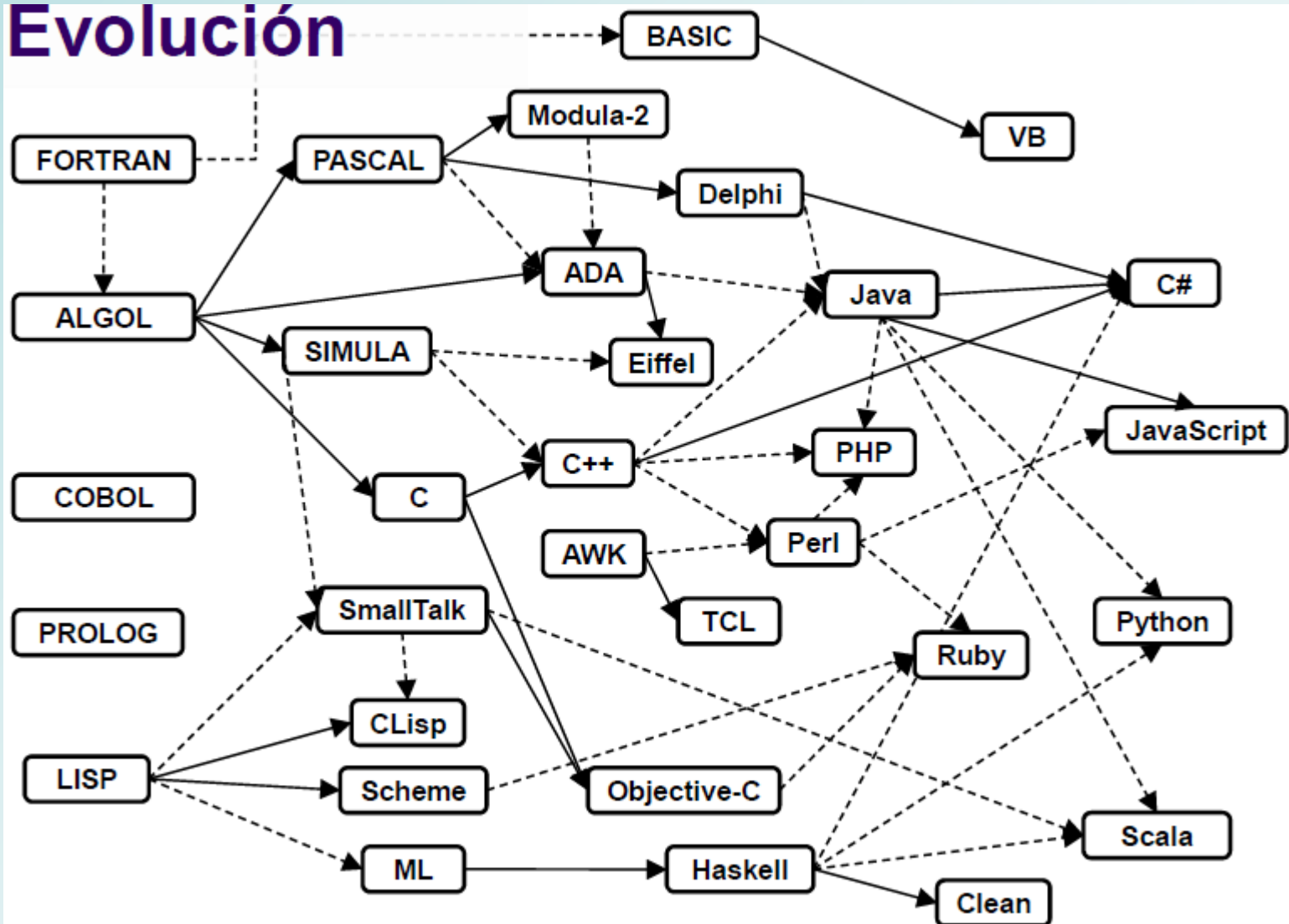
1. Introducción



Fuente: http://www.cs.fsu.edu/~engelen/courses/COP402001/notes1_4.pdf

1. Introducción

Evolución



1. Introducción



2. Paradigmas de programación

Concepto de paradigma

paradigma.

(Del lat. *paradigma*, y este del gr. παράδειγμα).

1. m. Ejemplo o ejemplar.

2. m. Teoría cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento; p. ej., en la ciencia, las leyes del movimiento y la gravitación de Newton y la teoría de la evolución de Darwin.

Real Academia Española © Todos los derechos reservados



2. Paradigmas de programación

Concepto de paradigma

- Un paradigma de programación indica un **método** de realizar cálculos y la **manera** en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa.
- Los paradigmas están asociados a determinados **modelos de cómputo** y **estilos de programación**.
- → híbridos



2. Paradigmas de programación



2. Paradigmas de programación

PARADIGMA IMPERATIVO (Procedurales)

- Es el estándar “de facto”.
- Se enfoca en **cómo** la computadora debe **realizar las cosas**.
- Basado en la arquitectura Von Neumann, ya que tienen una estrecha relación con la arquitectura de la computadora e implican uso y modificación de variables.



2. Paradigmas de programación

PARADIGMA IMPERATIVO (Procedurales)

- Las variables son celdas de memoria que contienen datos (o referencias), pueden ser modificadas y representan el estado del programa.
- Un cómputo consiste en una serie de sentencias, ejecutadas según un control de flujo explícito, que modifican el estado del programa.



2. Paradigmas de programación

PARADIGMA ORIENTADO A OBJETOS (POO)

La unidad de la POO es la **clase**, donde se definen **atributos** (propiedades) y **métodos** (acciones) en su declaración.

Los objetos son INSTANCIACIONES de una clase.

Nombre de la clase
Atributos
Métodos

2. Paradigmas de programación

PILARES POO

1. Abstracción

2. Encapsulamiento

3. Relaciones

4. Herencia

2. Paradigmas de programación

PARADIGMA DECLARATIVO

Está basado en el desarrollo de programas en los que se especifica un conjunto de condiciones, proposiciones o transformaciones, etc. que **describen el problema y detallan su solución**, pero **no indican cómo encontrarla**.

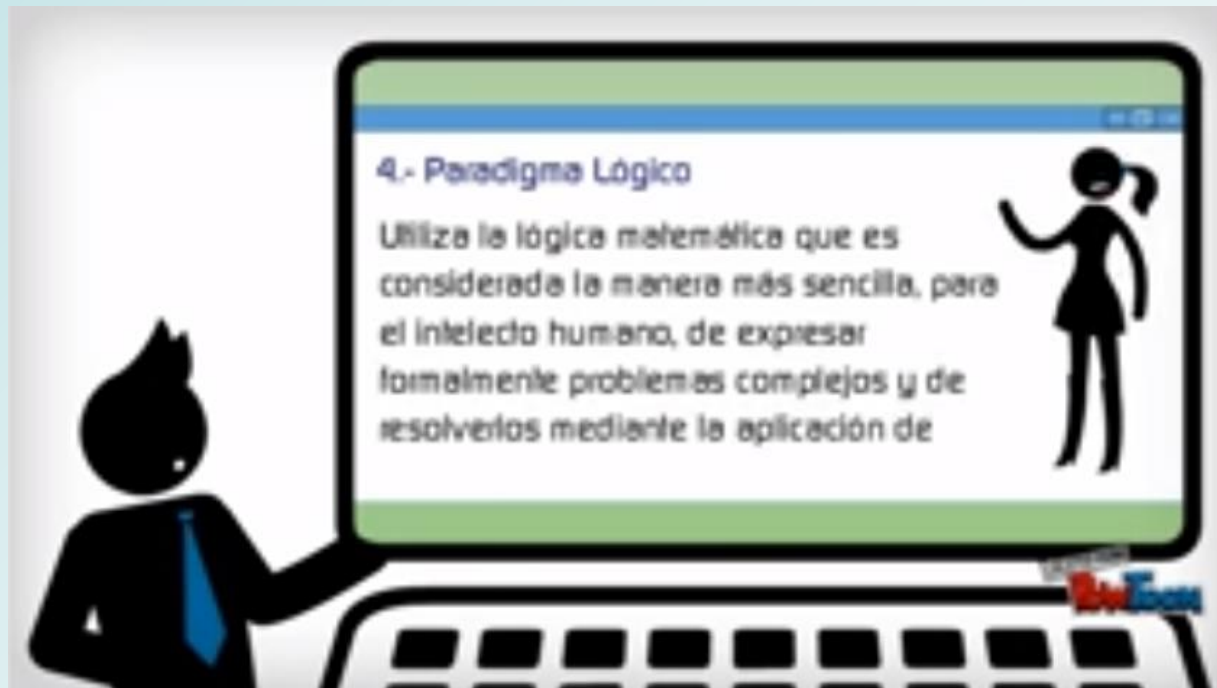
```
(define divide-list  
  (lambda (l)  
    (if (null? l)  
        '()  
        (cons (car l) (divide-list (cdr l))))))  
  
(define merge  
  (lambda (s1 s2 . pred?)  
    (let ((p1 (if (null? s1) '() s1))  
          (p2 (if (null? s2) '() s2))  
          (pred? (if (null? pred?) #f pred?)))  
      (let merge ((s1 s1) (s2 s2))  
        (if (and (pred? (car s1) (car s2))  
                (pred? (car s2) (car s1)))  
            (cons (car s1) (merge (cdr s1) (cdr s2)))  
            (cons (car s1) (merge (cdr s1) s2)))  
          (cons (car s2) (merge s1 (cdr s2)))))))  
  
(define merge-sort  
  (lambda (l . pred?)  
    (if (null? l)  
        '()  
        (let ((mid (divide-list l)))  
          (merge (merge-sort (car mid) pred?)  
                  (merge-sort (cadr mid) pred?)  
                  pred?))))))  
  
(merge-sort (cons 1 2 3 4 5 6 7 8 9 10))
```



3. Características de los paradigmas de programación

PARADIGMA IMPERATIVO	PARADIGMA ORIENTADO A OBJETOS (POO)	PARADIGMA DECLARATIVO
Basado en variables.	Basado en atributos y métodos.	Basado en funciones y relaciones.
Un elemento clave son los procedimientos.	Un elemento clave son las clases y objetos.	Un elemento clave son las funciones de orden superior, concepto que permite que las funciones sean parámetros de entradas y salida de otras funciones.
La sentencia principal es la asignación.	Una característica muy importante es la herencia entre clases.	La unidad son las funciones, una función que llama a otra, pudiendo ser la misma (recursividad).

Video



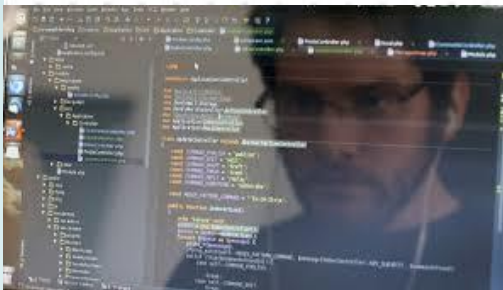
Prueba de control

Relacione las columnas:

1. PARADIGMA IMPERATIVO	<input checked="" type="checkbox"/> (2) Atributos
2. PARADIGMA O. O.	<input type="checkbox"/> () Constructores
3. PARADIGMA DECLARATIVO	<input type="checkbox"/> () Funciones
	<input type="checkbox"/> () Objetos
	<input type="checkbox"/> () Variables
	<input type="checkbox"/> () Relaciones
	<input type="checkbox"/> () Herencia
	<input type="checkbox"/> () Recursividad
	<input type="checkbox"/> () Procedimientos

Conclusiones

1. Aprendimos que existen paradigmas de programación, que tienen sus propias particularidades.
2. Todos los paradigmas son importantes, comparten ciertos elementos y cada uno tiene un contexto específico.



Qué hemos aprendido el día de hoy?



Utiliza el chat para participar



ACTIVIDAD GRUPAL

Investigue en internet sobre los paradigmas de programación y realice lo siguiente:

1. Adicione otras características importantes a cada uno de los paradigmas.
2. Explique un ejemplo básico (en código) de c/u.

PRESENTAR EL TRABAJO EN DIAPOSITIVAS





**Universidad
Tecnológica
del Perú**