# 

Lenguajes de Programación 1

Estudiante:

Alan Argotte

David Díaz

Mauricio Fragachán

Profesor:   
Ricardo Monascal

**Planificación**

|  |  |  |
| --- | --- | --- |
| Evaluación | Fecha | Ponderación |
| Quiz | Semana 5 | 20% |
| Parcial 30% | Semana 8 | 30% |
| Parcial 30% | Semana 12 | 30% |
| Tareas y asistencias | Todo el trimestre | 20% |

Para contacto con el profesor:

Oficina 202B

AUL 207

[rmonascal@gmail.com](mailto:rmonascal@gmail.com)

[rmonascal@usb.ve](mailto:rmonascal@usb.ve)

**Índice**

1. [Clase 1. Título](#Clase_1)

**Clase 01. Administrativas e Introducción**

<https://sites.google.com/view/ci3641-sd2024/clases/clase-1?authuser=0>

Los lenguajes los podemos clasificar en distintas categorías en función de sus características. Entre estas características podemos distinguir cosas como, si un programa escrito en tal lenguaje es almacenado en memoria, si es modificado durante tiempo de ejecución, si es escrito como una serie de instrucciones o un conjunto de reglas… Adicionalmente, indicamos que cuando un lenguaje puede hacer “todo lo que queramos”, decimos que son Turing completos (tienen tanto poder semántico como una máquina de Turing). Es decir, pueden computar todo aquello que es computable.

Lenguajes Imperativos

Von Neumann

Scripting

Lenguajes Declarativos

De Flujo

Funcionales

Lógicos

Relacionales

**Lenguajes declarativos**

**Lenguajes imperativos**

**Lenguajes de específicos de dominio**

**¿Cómo entiende la computadora un lenguaje?**

**Compilación**

*Los compiladores producen programas objetos a una abstracción menor*

**Interpretación**

*Las máquinas ejecutan línea a línea el código del programa*

**Máquina virtual**

*Un punto medio entre un compilador y un intérprete, pues los programas se reducen a una abstracción menor entendible por un intérprete especializado.*

**Pre-procesador**

*Proceso que cambia el Árbol Sintáctico Abstracto de un programa antes del proceso de compilación para incluir o eliminar código fuente de un programa condicionalmente. Es aplicable en compilación o interpretación*

**Traducción de alto nivel (*transpilación*)**

*Traducción de alto nivel de un lenguaje a otro.*

**Just-in Time (JIT)**

*Técnica de interpretación de las máquinas virtuales que compilan el código conforme lo necesitan. Toman secciones del código que son más utilizadas y se mantienen en ejecución, compilando solo aquello que se hace falta llevándolo a memoria.*

**¿Quién compila al compilador?**

Hacemos una breve reseña histórica sobre la historia del primer compilador. En resumen, los primeros compiladores debieron ser escritos en lenguajes ensamblables. Sin embargo, surge el siguiente problema: para cada arquitectura de máquina hay que escribir un compilador para el lenguaje. ¿Cómo resolvemos este problema, donde con nuestra solución podamos crear un compilador para un lenguaje que funcione en todas las arquitecturas?

El profesor Ricardo Monascal ingenia los diagramas T para poder explicar gráficamente este problema, originalmente resuelto por Niklaus Wirth, creador del lenguaje Pascal. Presentamos la solución ahora con diagramas T del sistema Pascal/P-code.

P-code

Pascal

Intérprete de P-code escrito en Pascal

Pascal

P-code

P-code

Traductor de Pascal a P-code escrito en P-code

Pascal

P-code

Pascal

Traductor de Pascal a P-code escrito en Pascal

asd

**Bootstrapping**

*Alimentar el traductor traducido consigo mismo*

**Clase 02. Nombres y asociaciones**