

# Programación I

## Python 2.7

Miguel Solis

Introducción

Facultad de Ingeniería / Escuela de Informática  
Universidad Andrés Bello, Viña del Mar.

- Clases (diurno):
  - Martes y Jueves, 15.50 → 17.30, INF306
- Correo: [miguel.solis@unab.cl](mailto:miguel.solis@unab.cl)
- Apuntes y ejercicios: UNAB Virtual

- 3 Solemnes ( $S_1$ ,  $S_2$  y  $S_3$ )
- Proyecto:
  - Hito:  $P_h$
  - Entrega final:  $P_f$
- Laboratorio ( $L$ )

Nota presentación:

$$NP = 0,2 \cdot S_1 + 0,2 \cdot S_2 + 0,2 \cdot S_3 + 0,1 \cdot P_h + 0,1 \cdot P_f + 0,2 \cdot L$$

- Si  $NP \geq 5,0$  y Notas parciales  $\geq 4,0$ :  
Alumno se exime (Nota final = NP)
- Sino:
  - Nota final =  $0,7 \cdot NP + 0,3 \cdot E$

- Solemne 1: 03/04
- Solemne 2: 03/05
- Solemne 3: 14/06
- Presentación Proyecto: 19/06

- Obligatoria:  
Joyanes, L. (2003). Fundamentos de programación: Algoritmos, estructura de datos y objetos. Madrid: McGraw-Hill.
- Recursos en línea:
  - Documentación oficial: <http://docs.python.org>
  - Python no muerde: <http://nomuerde.netmanagers.com.ar>

# ¿Qué es la programación?

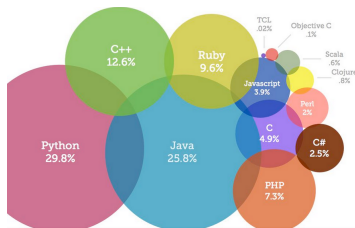
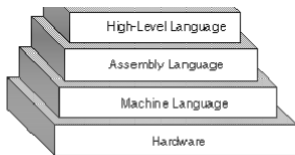
- **Proceso** de planificar una **secuencia de instrucciones** que debe seguir un computador **para realizar cierta tarea**.
- Etapas para escribir dicha secuencia:
  - **Resolver** el problema
  - **Implementar** dicha resolución

- Análisis: Comprender el problema
- Algoritmo: Desarrollar una secuencia lógica de pasos para resolver el problema
- Test: Seguir estos pasos para comprobar que realmente resuelve el problema



- Traducir el algoritmo propuesto a un lenguaje de programación
- Ejecutar el programa, comprobando los casos de prueba propuestos y corrigiendo la solución en caso de que sea necesario.

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.



# Modelos (paradigmas) de Programación

- Imperativo: Es el más común. Describe la programación en términos del estado del programa y sentencias que cambian este estado.
- Dinámica: Consiste en romper el problema en partes pequeñas para realizar el análisis.
- Orientado a objetos: Su núcleo consiste en la unión de datos y procesamiento en una entidad llamada *objeto*.

## Ventajas:

- Hace hincapié en sintáxis limpia para favorecer un código legible.
- Lenguaje de programación multiparadigma.
- Tipado dinámico: una misma variable puede tomar valores de distinto tipo en distintos momentos.

```
n = int(raw_input('Ingrese n: '))
es_primo = True
for d in range(2, n):
    if n % d == 0:
        es_primo = False
if es_primo:
    print n, 'es primo'
else:
    print n, 'es compuesto'
```

- Problema  
Entrada → Salida
- Algoritmo  
Secuencia de pasos para resolver un problema
- Programa  
Secuencia de instrucciones descritas en un lenguaje que puede ser entendido por el computador

Determine la secuencia de pasos ordenadas de forma lógica, que a partir del ingreso de un número  $n$ , determine si dicho número es primo o compuesto.

Buscar algún valor  $d$  que este entre  $2$  y  $n - 1$  que sea divisor de  $n$ .

Si existe por lo menos uno de estos valores, entonces  $n$  es compuesto; o si no, es primo.

# Solución en pseudocódigo

leer  $n$

es primo = verdadero

**para**  $d$  de 2 a  $n - 1$ :

**si**  $n$  es divisible por  $d$ :

    es primo = falso

**si** es primo es verdadero:

  escribir  *$n$  es primo*

**sino**:

  escribir  *$n$  es compuesto*



Escriba el pseudocódigo de un programa que calcule el factorial de un número natural  $n$ .