

Intro a Java

Algoritmos y Estructuras de Datos

1er cuatrimestre 2025

Paradigmas de Programación

Paradigmas distintos: Haskell vs. Python y Java

Haskell → *Programación Funcional*

- ▶ Se enfoca en **datos inmutables** y **funciones puras**.
- ▶ No hay efectos secundarios, fuerte uso de recursión y funciones de orden superior.

Python y Java → *Programación Imperativa & Orientada a Objetos*

- ▶ Los programas consisten en **instrucciones que cambian el estado**.
- ▶ Uso de **variables, bucles, condicionales y objetos**.

(Algunas) diferencias entre Java y Python

	Java	Python
Tipado	Estático	Dinámico
Sintaxis	Llaves {} y ;	Bloques indentados
Compilación	Compila a <i>bytecode</i>	Interpretado (compila en runtime)
Orientación	Todo dentro de clases!	POO o procedural
Rendimiento	Más rápido (compilación JIT)	Más lento (pero más flexible)
Ejemplo	javac MiPrograma.java → java MiPrograma	python3 mi_programa.py

Aprendamos con ejemplos

La idea de esta clase es que vayan familiarizándose con la sintaxis (cómo se escribe) de Java. Vamos a ver ejemplos de problemas vistos en Introducción a la Programación/Algoritmos 1.

Problema 1: Fibonacci

Problema: hacer una función que calcule el número de Fibonacci.
Recordar que $fib_0 = 0$, $fib_1 = 1$ y $fib_n = fib_{n-1} + fib_{n-2}$ para todo $n > 1$.

Problema 1: Fibonacci

- ▶ Las funciones pueden ser recursivas.
- ▶ Cómo se escribe una función básica.
- ▶ Estructura del if,else if y else.
- ▶ Tipo int.
- ▶ Funcionamiento del return (tener cuidado!).

Problema 2: máximo

Problema: Dada una secuencia no vacía de números reales, encontrar el valor máximo.

Problema 2: máximo

Problema: Dada una secuencia no vacía de números reales, encontrar el valor máximo.

```
proc maximo (in s: seq<ℝ>) : ℝ {  
  requiere {|s| > 0}  
  asegura {res ∈ s ∧ (∀x : ℝ) (x ∈ s → x ≤ res)}  
}
```


Problema 2: máximo

- ▶ Estructura del for in.
- ▶ Tipo double, booleano, arreglos.
- ▶ Cómo buscamos errores (veremos en un rato).

Problema 3: sumarArreglos

Problema: dadas dos secuencias de enteros con el mismo tamaño, sumarlas.

Problema 3: sumarArreglos

Problema: dadas dos secuencias de enteros con el mismo tamaño, sumarlas.

```
proc sumarArreglos (in s1: seq⟨ℤ⟩, in s2: seq⟨ℤ⟩) : seq⟨ℤ⟩ {  
  requiere {|s1| = |s2|}  
  asegura {|res| = |s1| ∧L (∀ i : ℕ) (i < |res| ⟶L res[i] =  
    s1[i] + s2[i])}  
}
```

Problema 3: sumarArreglos

- ▶ Estructura del for general.
- ▶ Crear arreglos.

Problema 4: iniciales

Problema: dado un `String` (secuencia de caracteres), devolver otro `String` con las iniciales de las palabras.

Problema 4: iniciales

- ▶ Funciones de `String`.
- ▶ Operaciones con booleanos.

Algunas funciones matemáticas útiles

- ▶ Valor absoluto: `Math.abs`
- ▶ Raíz cuadrada: `Math.sqrt`
- ▶ Techo y piso: `Math.ceil` y `Math.floor`
- ▶ `Math.max` y `Math.min`