

# Lenguajes de Programación Científica

**Giovanni Ramírez García, PhD**

Escuela de Ciencias Físicas y Matemáticas  
Universidad de San Carlos de Guatemala

Guatemala, 26 de enero de 2021



# Lenguajes de Programación Científica

## Lenguajes compilados y lenguajes interpretados

### Práctica con Fortran

# Lenguajes de Programación Científica

Lenguajes compilados y lenguajes interpretados

Práctica con Fortran

*"The determined Real Programmer can write FORTRAN programs in any language."* – Ed Post, Real Programmers Don't Use Pascal, 1982.

un programador resuelve  
un problema y luego  
lo escribe en cualquier  
lenguaje

# Lenguajes de programación

- ▶ Un *lenguaje científico* es un lenguaje de programación que está optimizado para fórmulas matemáticas y matrices.
- ▶ Entre los lenguajes más famosos están: Matworks MATLAB, Python, FORTRAN, Julia, C/C++, R.
- ▶ ¿Por qué usar software libre?

# Lenguajes de programación

- ▶ Un *lenguaje científico* es un lenguaje de programación que está optimizado para fórmulas matemáticas y matrices.
- ▶ Entre los lenguajes más famosos están: Matworks MATLAB, Python, FORTRAN, Julia, C/C++, R.
- ▶ ¿Por qué usar software libre?

## Software libre en ciencias

- ▶ Libertad para correr el software.
- ▶ Libertad para copiar el software.
- ▶ Libertad para distribuir el software.
- ▶ Libertad para modificar el software.

# Lenguajes en investigación científica

- ▶ Cuando las computadoras se vuelven herramientas indispensables, los lenguajes de programación se vuelven más importantes.
- ▶ ¿Cómo elegimos un lenguaje de programación u otro?
- ▶ Lo importante es aprender a resolver un problema, una vez resuelto, sólo hay que traducirlo al lenguaje que sea necesario usar.

- ▶ Cuando las computadoras se vuelven herramientas indispensables, los lenguajes de programación se vuelven más importantes.
- ▶ ¿Cómo elegimos un lenguaje de programación u otro?
- ▶ Lo importante es aprender a resolver un problema, una vez resuelto, sólo hay que traducirlo al lenguaje que sea necesario usar.

Factores importantes [Jeff Templon] para elegir un lenguaje son

- ▶ el incremento de la complejidad de las tareas que se programan;
- ▶ el incremento de la complejidad y diversidad de las máquinas que se programan;
- ▶ el incremento de la visualización gráfica e interfaz de los programas;
- ▶ el incremento en el número de lenguajes de programación disponibles.



# Fortran

- ▶ Fortran es el lenguaje de la computación científica pasada y uno de los principales lenguajes del presente [Jeff Templon].
- ▶ Sus principales fortalezas son:
  - ▶ una cantidad de código probado y que funciona
  - ▶ una larga experiencia
  - ▶ fácil de usar (desde la versión 90).
- ▶ Es un lenguaje fácil de aprender, portable, fácil de implementar y de buscar errores.

# Fortran

- ▶ *Fortran es el lenguaje de la computación científica pasada y uno de los principales lenguajes del presente* [Jeff Templon].
- ▶ Sus principales fortalezas son:
  - ▶ una cantidad de código probado y que funciona
  - ▶ una larga experiencia
  - ▶ fácil de usar (desde la versión 90).
- ▶ Es un lenguaje fácil de aprender, portable, fácil de implementar y de buscar errores.
- ▶ Fortran significa “Formula Translation” y fue desarrollado por el equipo de John Backus en 1957 para la IBM 704.
- ▶ Unas de sus desventajas son: tiene pocas estructuras de datos, tiene una forma de programar a la antigua.
- ▶ Es fácil de convertir a código paralelo.

# Otros lenguajes

## C/C++

- ▶ C fue desarrollado por Dennis Ritchie y C++ fue desarrollado por Bjarne Stroustrup en Bell Labs.
- ▶ Extremadamente flexible.
- ▶ Produce código más lento que su equivalente en Fortran (esto es menos relevante en máquinas más rápidas).
- ▶ Permite interactuar con el hardware fácilmente.

## Python

- ▶ Lenguaje interpretado, diseñado por Guido von Rossum.
- ▶ Bastante útil para hacer pruebas y prototipos.
- ▶ Ideal para la enseñanza en clases introductorias.
- ▶ Considerablemente lento

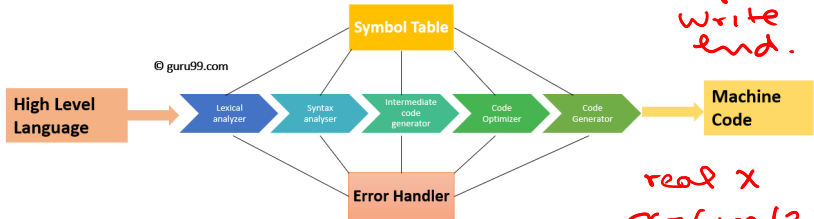
# Lenguajes de Programación Científica

## Lenguajes compilados y lenguajes interpretados

### Práctica con Fortran

gfortran

Print  
read  
Program  
write  
end.



real x  
 $x = 5m(3.14)$

- ▶ Transforma código escrito en lenguaje de alto nivel a código máquina.
- ▶ Análisis léxico: constantes, variables, palabras reservadas.
- ▶ Análisis de sintaxis: formato.

— (+) —

- ▶ Análisis semántico: consistencia del código.
- ▶ Código intermedio: máquina abstracta.
- ▶ Optimización de código.
- ▶ Generación de código.

{ 0 ✓  
1  
2  
⋮

# Intérprete

- ▶ Un intérprete convierte cada línea de un lenguaje de alto nivel en código máquina.
- ▶ Un intérprete convierte cada línea en una forma secuencial, aunque se puede cambiar la secuencia.
- ▶ El código interpretado es más lento que el código compilado.
- ▶ Si el código interpretado tiene errores, éstos aparecen hasta que se interpreta esa línea.

```
import random ✓  
dado = [1, 2, 3] ✓  
random.shuffle(dado)  
random.shuffle(dado) ✓
```

- ▶ Los intérpretes suelen ser más fáciles de usar para principiantes.
- ▶ Los intérpretes permiten tener códigos más portables.
- ▶ El código interpretado reside en la memoria durante la interpretación.
- ▶ No tiene opciones robustas de optimización.

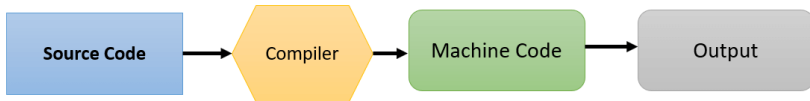
# Comparación

Real X  
Print(S.O) ✓  
Read(X)



5.0

## How Compiler Works



© guru99.com

## How Interpreter Works



5.0

# Lenguajes de Programación Científica

Lenguajes compilados y lenguajes interpretados

Práctica con Fortran



# holaMundo.f90

1. PROGRAM holaMundo
2. WRITE (\*,\*) "Hola Mundo"
3. WRITE (99,\*) "Hola Mundo"
4. END PROGRAM holaMundo

1. PROGRAM holaMundo
2. IMPLICIT NONE ✓
3. INTEGER(4)::a , b, c,
4. REAL(4)::x 0., 223
5. REAL(8)::y
6. READ (\*,\*) a → formato
7. READ (\*,\*) x,y → formato
8. WRITE (\*,\*) a, x
9. WRITE (\*,\*) y
10. END PROGRAM holaMundo

¡Muchas gracias!

Contacto:

Giovanni Ramírez García, PhD

[ramirez@ecfm.usac.edu.gt](mailto:ramirez@ecfm.usac.edu.gt)

<http://ecfm.usac.edu.gt/ramirez>