

Introducción a la programación

CI-0112 Programación 1

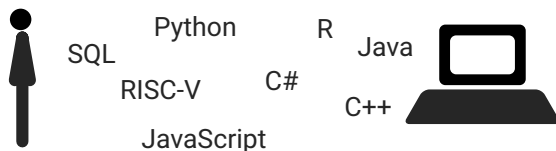
Sivana Hamer - sivana.hamer@ucr.ac.cr
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
Licencia: CC BY-NC-SA 4.0



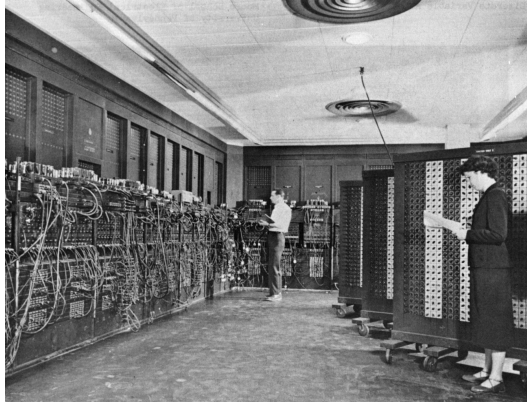
UNIVERSIDAD DE COSTA RICA

¿Qué es un programa?

Los programas son conjuntos de instrucciones escritos en un lenguaje formal que indica a una computadora cómo realizar una tarea

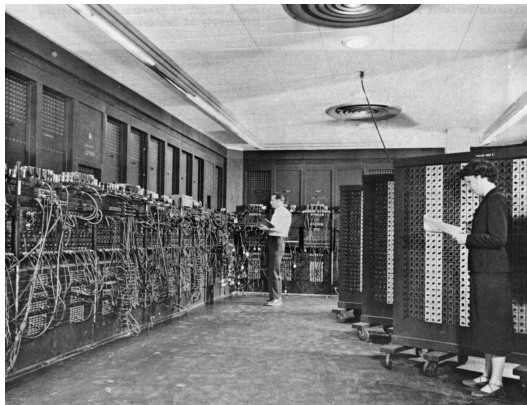


Dado que los programas se escriben en lenguajes, vamos a ver sobre los lenguajes de programación

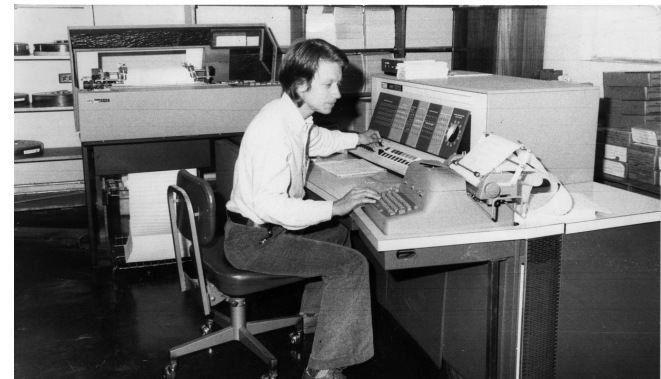


¿Qué era eso?

ENIAC, una de las primeras computadoras de los 40s



Matilde, la primera computadora en Costa Rica



¿Cómo se escribía código en esas máquinas? Lenguaje máquina

"Hello world" en lenguaje máquina [1]

```
00000000 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 |.ELF
.....|
00000010 02 00 03 00 01 00 00 00 54 80 04 08 34 00 00 00 |.....T
...4...|
00000020 00 00 00 00 00 00 00 00 34 00 20 00 01 00 00 00 |.....4.
.....|
00000030 00 00 00 00 01 00 00 00 00 00 00 00 00 80 04 08
|.....|
00000040 00 80 04 08 74 00 00 00 74 00 00 00 05 00 00 00 |...t...t
.....|
00000050 00 10 00 00 b0 04 31 db 43 b9 69 80 04 08 31 d2 |.....1.C.i
...1..|
00000060 b2 0b cd 80 31 c0 40 cd 80 48 65 6c 6c 6f 20 77 |...1.@@.Hello
w|
00000070 6f 72 6c 64                                     |orld|
```

¿Cuáles errores tiene el
lenguaje máquina?
Difícil y propenso a errores
en programas grandes

Dados las dificultades,
decidieron crear el lenguaje
ensamblador

El lenguaje ensamblador expresa las operaciones de bajo nivel en abreviaturas nemotécnicas

“Hello world” en ensamblador RISC-V para Linux [1]

```
.global _start      # Provide program starting address to linker

# Setup the parameters to print hello world and then call Linux to do it.

_start: addi a0, x0, 1      # 1 = StdOut
        la    a1, helloworld # load address of helloworld
        addi a2, x0, 13     # length of our string
        addi a7, x0, 64     # linux write system call
        ecall              # Call linux to output the string

# Setup the parameters to exit the program and then call Linux to do it.

        addi a0, x0, 0      # Use 0 return code
        addi a7, x0, 93     # Service command code 93 terminates
        ecall              # Call linux to terminate the program

.data
helloworld: .ascii "Hello World\n"
```

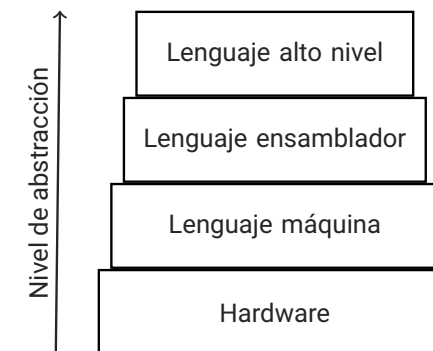
Los lenguajes de alto nivel son más legibles para humanos y portátiles entre computadoras

“Hello world” en Java

```
public class Hello {
    public static void main (String [] args){
        System.out.println("Hello world");
    }
}
```

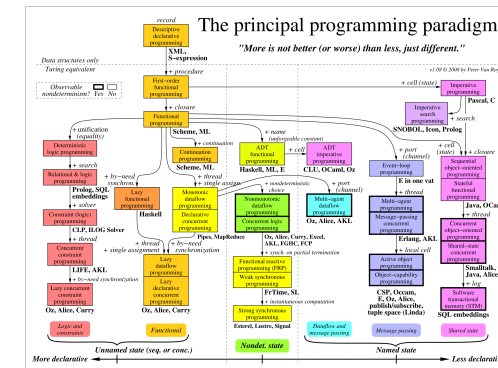
Sin embargo, es complejo escribir en ensamblador y depende de la máquina. Por lo tanto, se crearon lenguajes de alto nivel.

Siempre se traduce los lenguajes de alto nivel a los lenguajes de bajo nivel

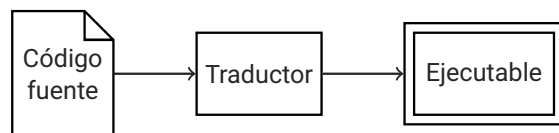


Todos los lenguajes de alto nivel tienen un paradigma.

Un paradigma de programación es una manera de programar en un lenguaje de alto nivel basado en reglas matemáticas o un conjunto de principios



Parar correr los lenguajes de alto nivel deben ser traducidos a lenguaje máquina



Nota

Traducir programas tiene un costo en tiempo, aunque tiene ventajas para la facilidad de lectura y portabilidad.

Se puede traducir el código de dos maneras...

Compilador

Traduce el código completo antes de la ejecución del programa
La traducción genera un ejecutable que puede correrse varias veces en la misma computadora.
Es como un traductor que traduce un libro de inglés a español, que produce un nuevo libro en español.

Interpretador

Traduce el código por partes durante la ejecución del programa
La traducción no genera nada, por lo que ocupa interpretarse cada vez en la misma computadora.
Es como un intérprete de LESCO que traduce al mismo tiempo que se está hablando.

Los repositorios de software son los artefactos producidos y archivos durante el desarrollo de software

Históricos	Tiempo de ejecución	Código fuente
Guardan distintas versiones de las aplicaciones. Incluyen repositorios de control de versiones, repositorios de seguimiento de problemas (<i>issue</i>) y comunicación de proyectos archivados.	Guardan la ejecución y uso de aplicaciones.	Guardan todo el código fuente de las aplicaciones.

Java es un lenguaje de alto nivel que usa el paradigma de programación orientada a objetos

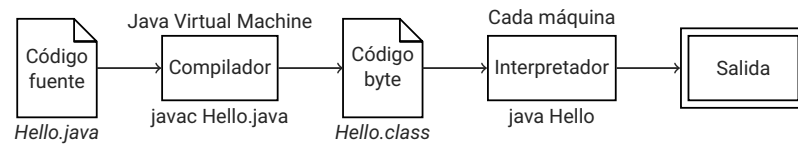


Concentremonos ahora en el lenguaje y tecnologías del curso

Java es uno de los lenguajes más populares [3, 4]

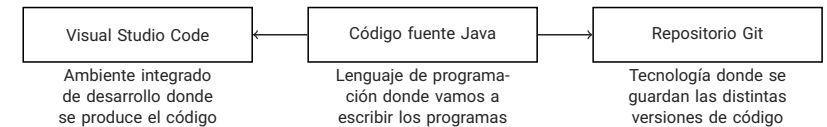


Traducir programas Java utiliza tanto un compilador como intérprete



Adaptado de Downey y Mayfield [5].

En nuestro curso vamos a usar las siguientes herramientas y tecnologías...



Referencias I

(2012) How could i write "hello world" in binary? [Online]. Available: <https://stackoverflow.com/questions/10821875/how-could-i-write-hello-world-in-binary>

P. V. Roy, "Programming Paradigms for Dummies: What Every Programmer Should Know," *New computational paradigms for computer music*, p. 39, 2009.

(2020) The state of the 2020 octo-verse. [Online]. Available: <https://octoverse.github.com/#overview>

(2021) Developer survey. [Online]. Available: <https://insights.stackoverflow.com/survey/2021>

A. B. Downey and C. Mayfield, *Think Java: How to Think Like a Computer Scientist*, second edition ed., 2020.

S. Adams, "Programming enviornment," [Image]. [Online]. Available: <https://dilbert.com/stip/2017-01-02>

Referencias II

- inconnu, “Java,” [Image]. [Online]. Available: https://www.brandsoftheworld.com/search?query_id=1&brand_id=90934
- U. Army, “Eniac,” [Image]. [Online]. Available: <https://commons.wikimedia.org/wiki/File:Eniac.jpg>
- K. O. Coto, “Matilde: una máquina con historia propia,” *Universidad de Costa Rica*, [Image]. [Online]. Available: <https://www.ucr.ac.cr/noticias/2019/01/15/matilde-una-maquina-con-historia-propia.html>
- (2019) Risc-v assembly language hello world. [Online]. Available: <https://smist08.wordpress.com/2019/09/07/risc-v-assembly-language-hello-world/>
- M. L. Scott, *Programming Language Pragmatics*, fourth edition ed. Waltham, MA: Morgan Kaufmann, an imprint of Elsevier, 2016.
- D. J. Eck, *Introduction to Programming Using Java*, eighth edition ed. Geneva (NY): Hobart and William Smith Colleges, Department of mathematics and computer science, 2020.

Referencias III

- H. Kagdi, M. L. Collard, and J. I. Maletic, “A survey and taxonomy of approaches for mining software repositories in the context of software evolution,” *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 19, no. 2, pp. 77–131, Mar. 2007.
- A. E. Hassan, “The road ahead for Mining Software Repositories,” in *2008 Frontiers of Software Maintenance*. Beijing, China: IEEE, Sep. 2008, pp. 48–57.
- S. Demeyer, A. Murgia, K. Wyckmans, and A. Lamkanfi, “Happy Birthday! A trend analysis on past MSR papers,” in *2013 10th Working Conference on Mining Software Repositories (MSR)*. San Francisco, CA, USA: IEEE, May 2013, pp. 353–362.