Introducción

I

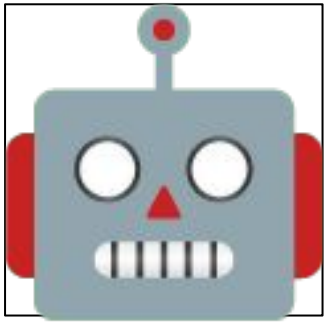
2024

UNRN

Universidad Nacional
de Río Negro

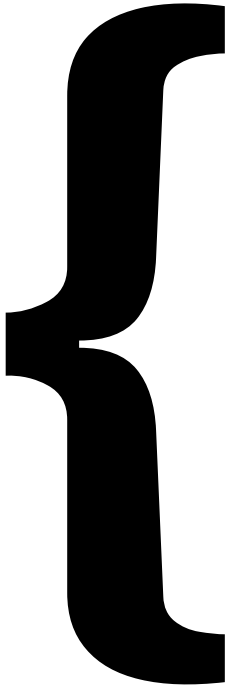
r14

Programación II es

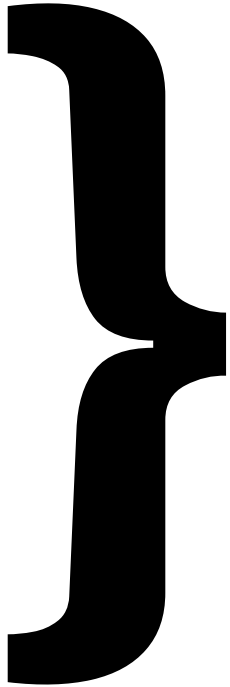


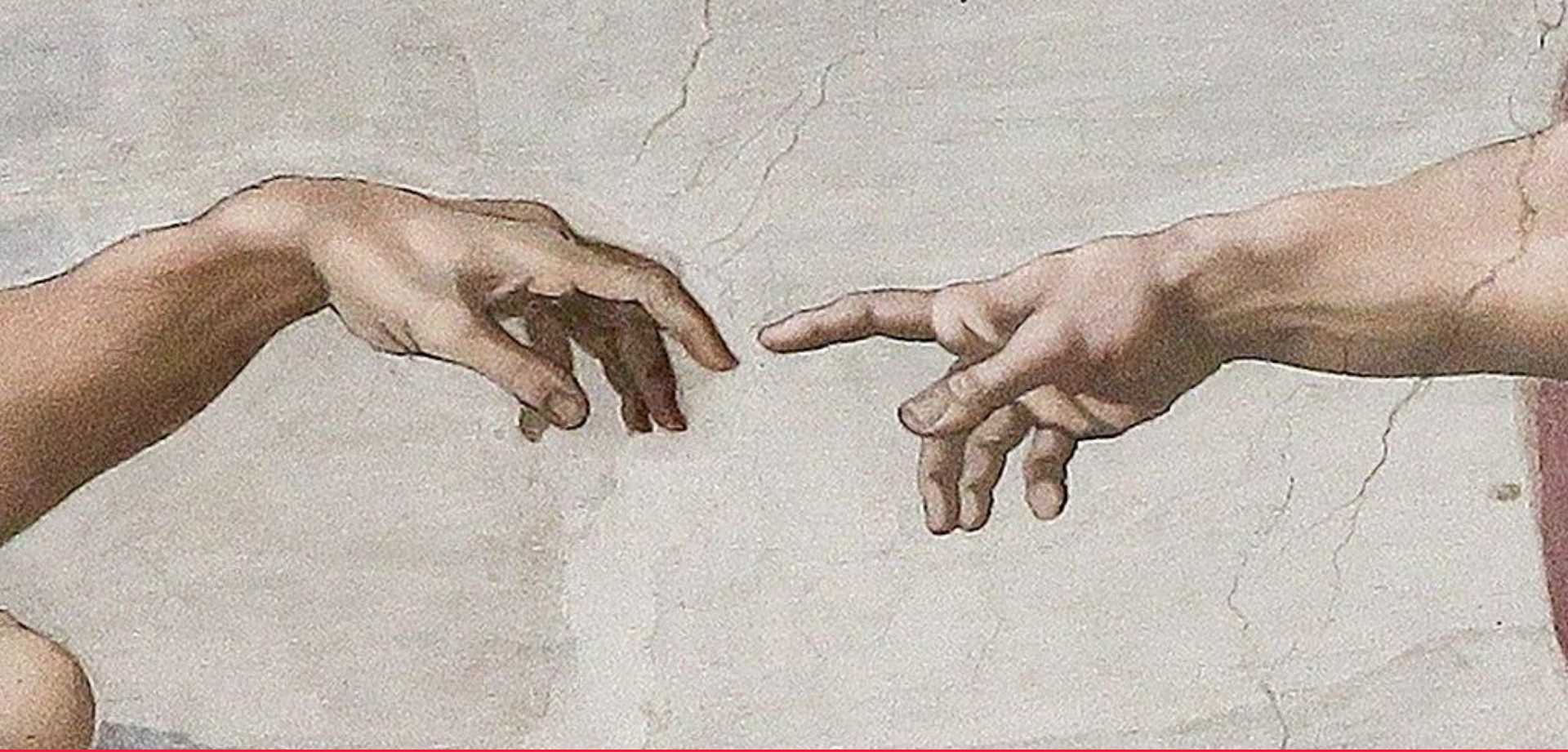
Martín René Vilugrón

Miguel Mariguin



Antes de empezar





Contacto

Vías

exclusivamente

- Directamente en clases
- Github Discussions
- Pull Request de corrección
- email - mrvilugron@unrn.edu.ar
  mmariguin@unrn.edu.ar

Abran hilo

Vean también en apunte sobre este tema

El material de la cátedra

Campus Bimodal +
github.com/INGCOM-UNRN-PII/cursada-2024

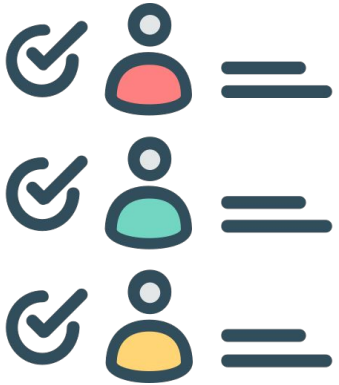
Incluyendo

- prácticas
- presentaciones
- bibliografía
- material extra

~~Torneos y Competencias~~
Términos y Condiciones

Como quedará todo grabado

No se toma asistencia



Por lo que es su responsabilidad estar al día

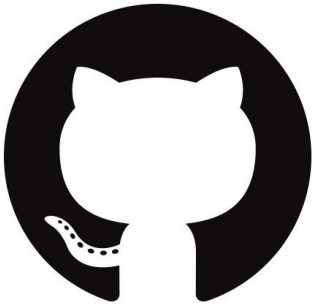
Los trabajos prácticos son la asistencia

1 x
semana

La programación se basa en la práctica

Les recomendamos hacer las prácticas *a conciencia*

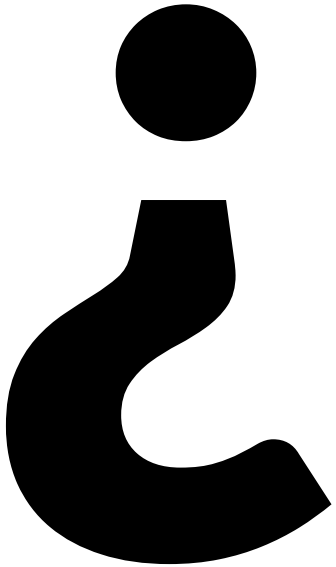
Serán sobre GitHub*



*Si no lo han hecho antes, les pediremos que nos compartan su nombre de usuario

La entrega es vía formulario

Que les enviará un acuse de recibo al entregar.



Cómo serán evaluados

con los ayudantes

Igual, el ‘resultado’ es manual

Es la ley

Dredd

Que juzga el estilo

Mirá como te mira Conan

Conan

Que ejecuta tests

{ están a prueba }

Abran hilo

Sí creen que la auto-corrección no es correcta

Estados posibles de un TP

- ✅ TP OK
- ⚠️ Correcciones
- ⏰ Entrega tarde
- ❌ Rechazado
- 🔆 Sin Corregir
- ⭕ Sin entregar
- 🚫 Sin comenzar

De la planilla de control y que recibirán por correo

TP OK

El práctico no tiene nada para resolver*.

***puede tener observaciones**

Correcciones

El práctico requiere correcciones y reentrega

Tienen tiempo hasta la fecha límite

Entrega tarde

Después de la fecha de entrega y antes del límite

(se combina con los otros)

y

Rechazado

revisar completo o entregado fuera de plazo

Sin corregir

Entregado pero aún no revisado por la cátedra

Sin entregar

Con actividad en el repositorio pero sin entregar

Sin comenzar

El repositorio no fue creado

De esto

Promoción y regularización

Con las prácticas

Para regularizar

Todas las prácticas entregadas

Máximos para promocionar

una (1) entrega con correcciones (⚠️)

Máximos para promocionar

tres (3) entregas tarde (⏰)

Van a ser un *montón* de prácticas*

*aproximadamente 12

Fechas

Fecha de entrega

Generalmente, una semana luego de su emisión

Fecha límite

Un día antes del parcial siguiente.

Luego de esta fecha no se aceptarán entregas y correcciones


¿Preguntas?

Parciales

hasta el momento

Los parciales

uno en papel, otro en la computadora*

*por ahora

Se promociona con:

80%

Se aprueba con:

60%

Y junto a los Trabajos Prácticos

Y [probablemente] un trabajo integrador al final con nota

Regularización

- Todos los TP entregados
- Parciales con nota ≥60%

Promoción

- 1 TP máximo con correcciones ⚠️
- 3 TP máximo con entrega tarde ⏰
- Parciales con nota ≥80%


¿Preguntas?

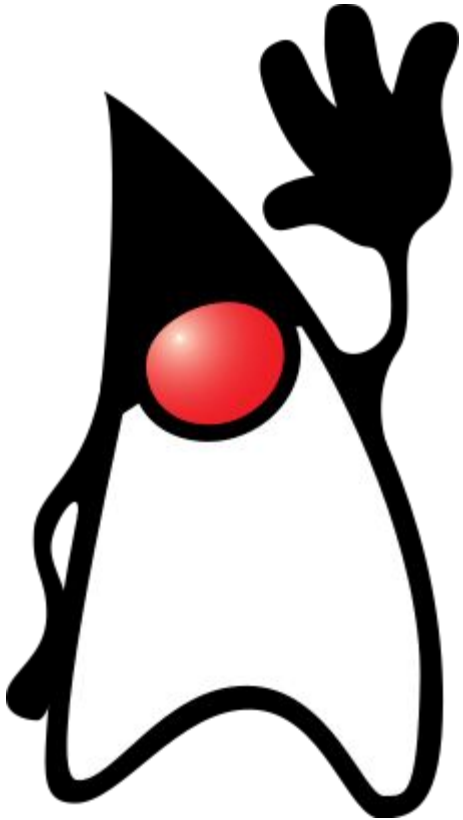
Abran hilo

{ De existir cambios, serán agregados al final. }

¿Qué vamos a ver?

+cronograma tentativo

Java



Parcial

3 de abril

Orientación a objetos

Estructuras de datos

Parcial II
29 de mayo

Trabajo Final
12 de junio

Recuperatorios

12 de junio

De todas formas, consulten el cronograma

Excelente

¿Qué es Java?

en.wikipedia.org/wiki/James_Gosling

Creado en ~1994
Por James Gosling

Java es:

1

Es un lenguaje de programación

[Más adelante vamos a ver que significa esto]

2 Es una plataforma

En constante actualización

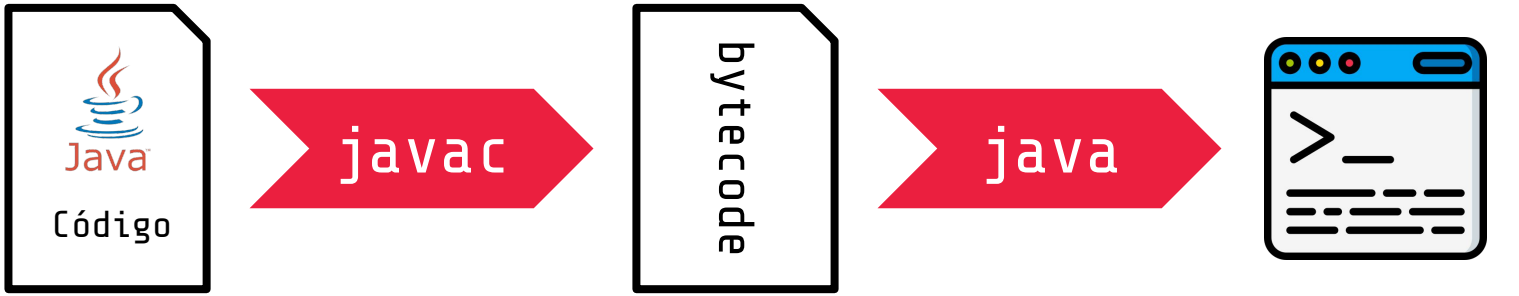
(¡cada 6 meses!)

Vamos a usar la versión 17*

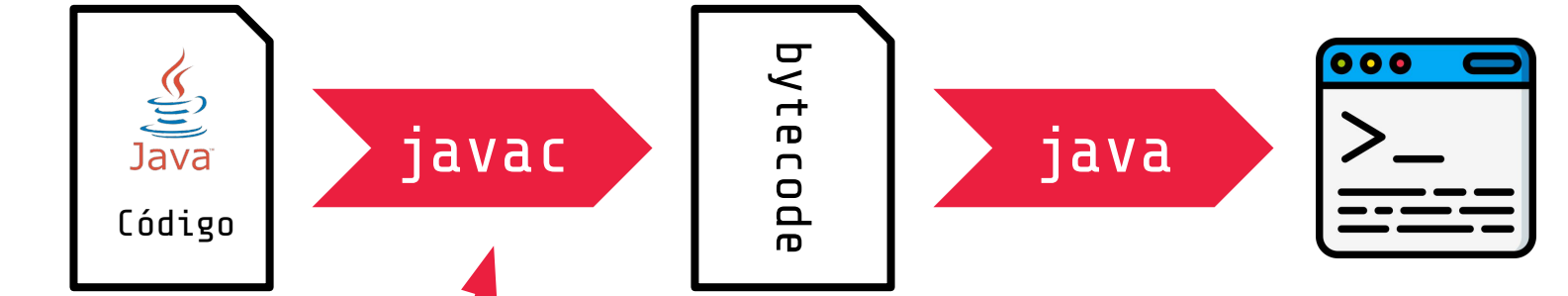
¿Como se ejecuta un programa?

a grandes razgos

Esto es lo que uno ve

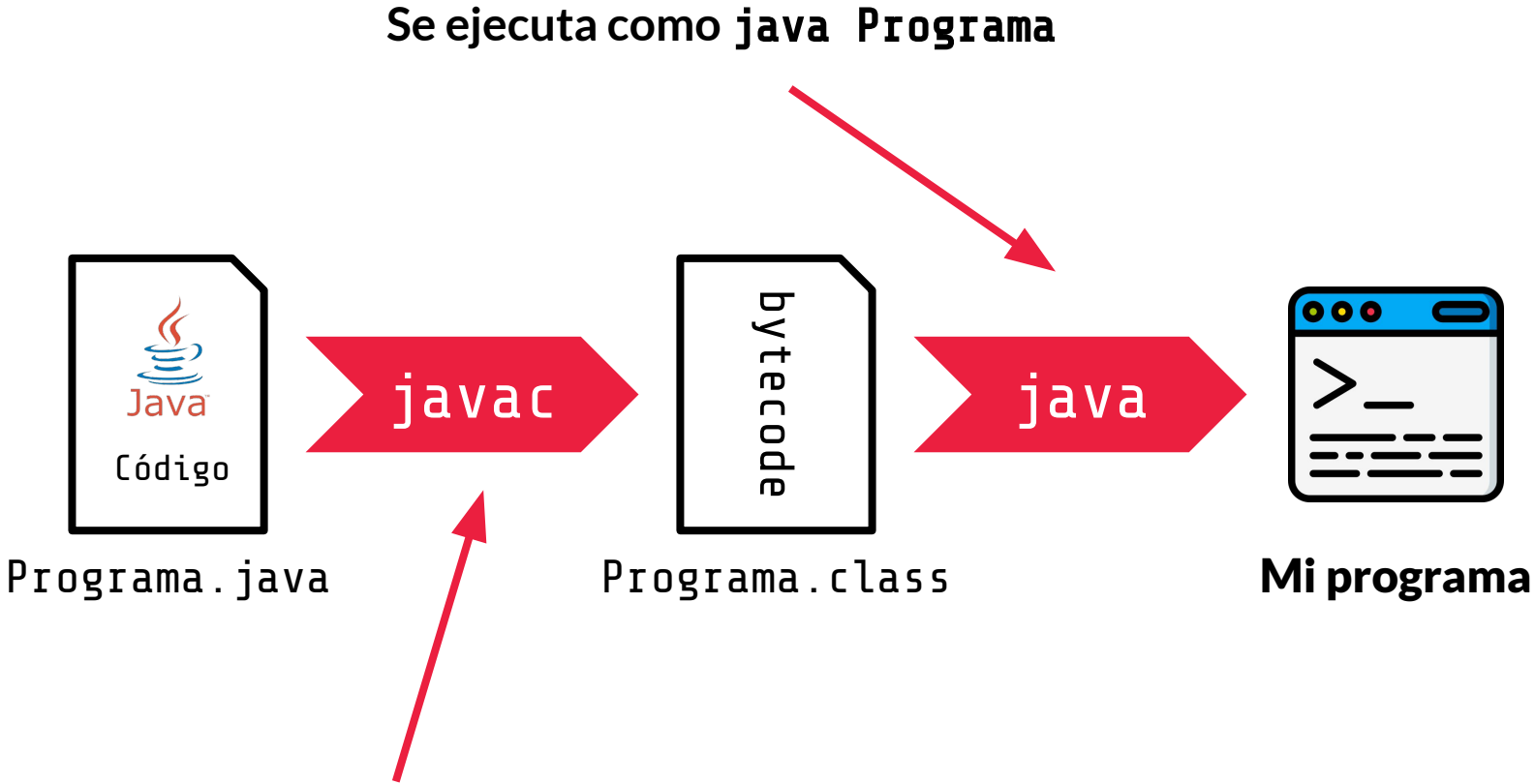


`Programa.java` `Programa.class` **Mi programa**



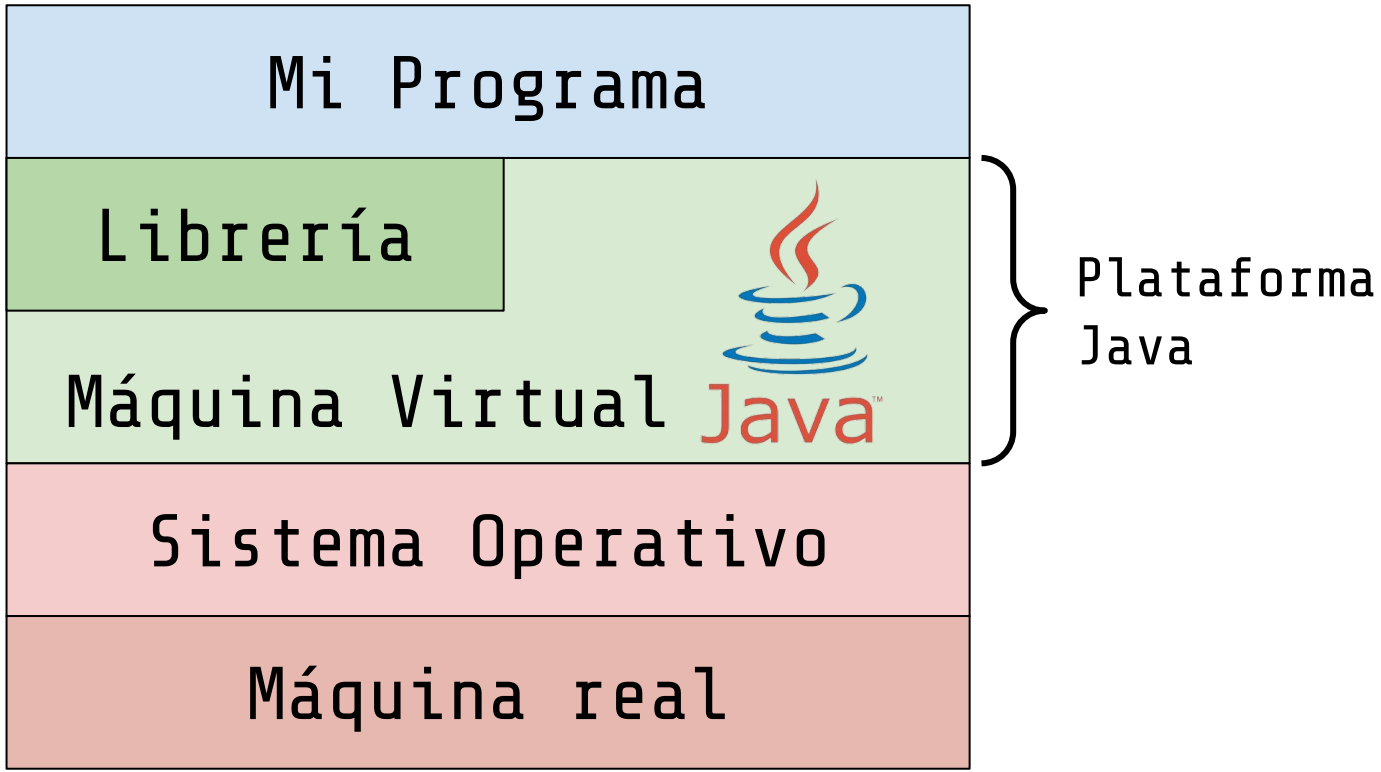
Hasta acá, lo mismo que con C

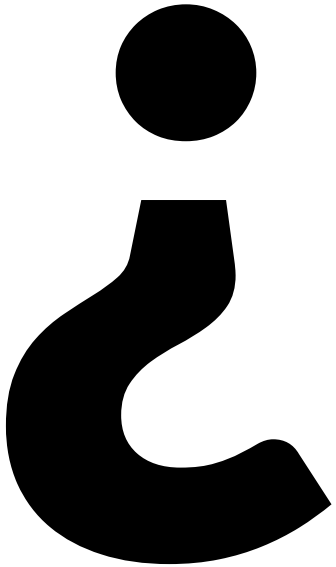
Pero esto no se puede ejecutar directamente (`./Programa.class`)


Se ejecuta como java Programa
Java
Código
javac
bytecode
java
Programa.java
Programa.class
Mi programa
Hasta acá, lo mismo que con C

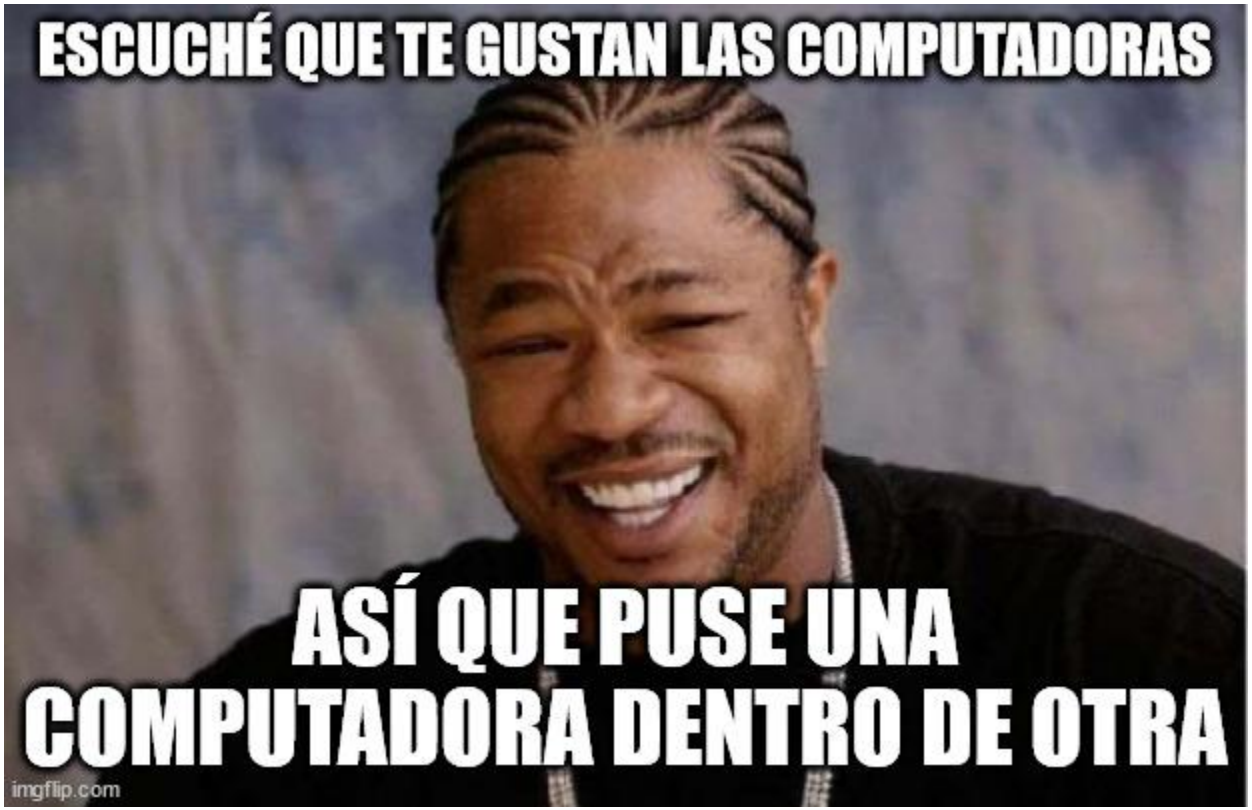
La plataforma está compuesta de:

La máquina virtual
Librerías de soporte

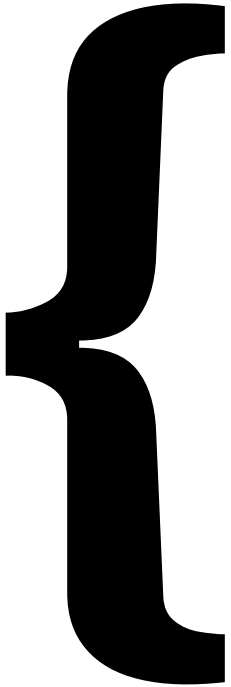

Mi Programa
Librería
Máquina Virtual
Java
Sistema Operativo
Máquina real
Plataforma
Java



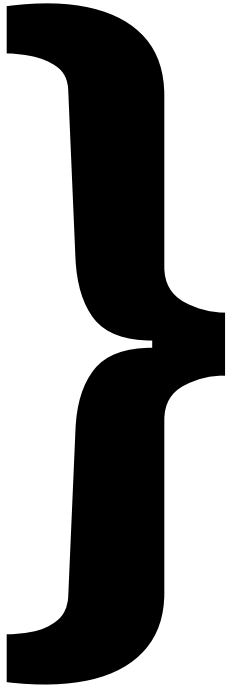
¿Máquina virtual?

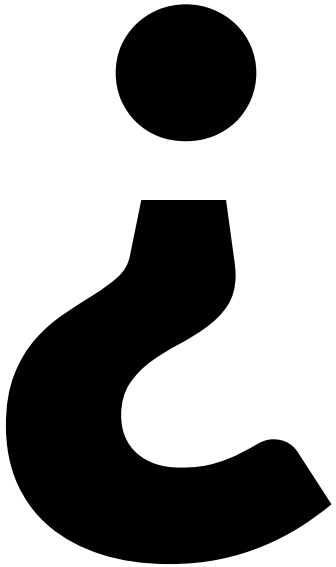

ESCUCHÉ QUE TE GUSTAN LAS COMPUTADORAS
ASÍ QUE PUSE UNA COMPUTADORA DENTRO DE OTRA
imgflip.com

o
un programa que actúa como una computadora



La llamaremos JVM
Java Virtual Machine





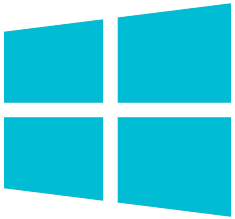
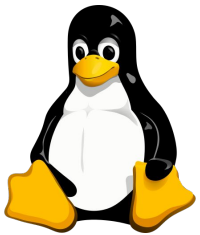
Pero...
Por qué

Trae muy interesantes ventajas

Un `.class` funciona en cualquier JVM

sin recompilar

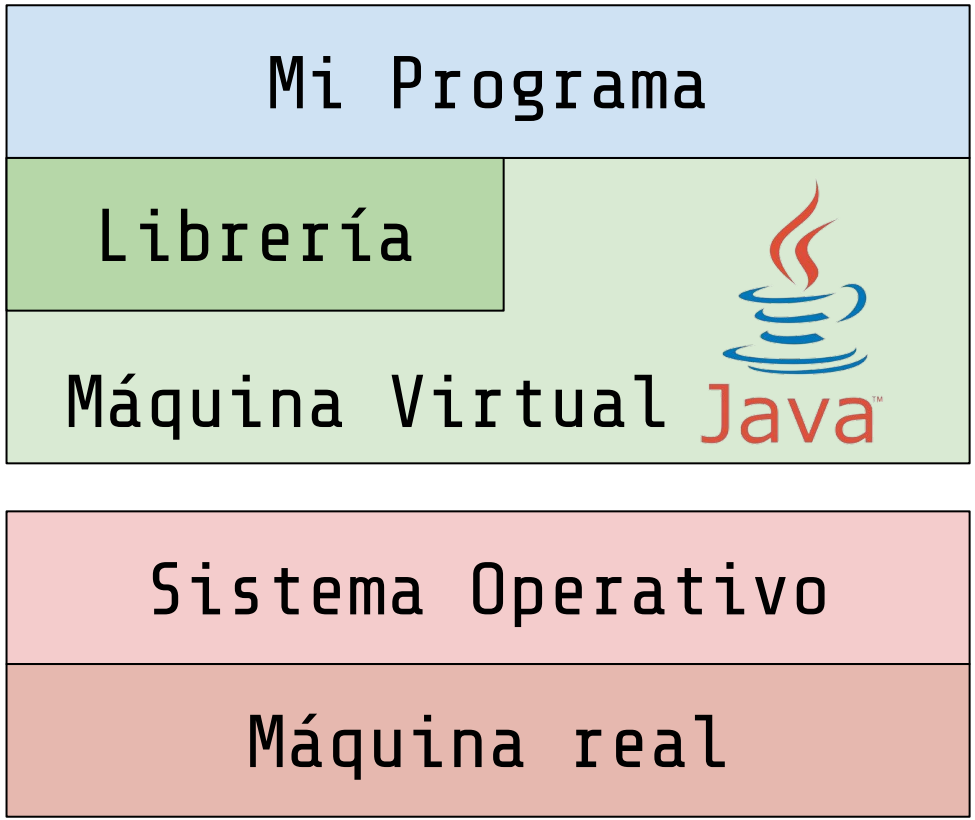


Esto es ‘portabilidad’

Independiente del procesador real

Ademas del sistema operativo



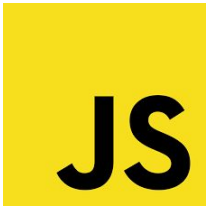
Nos independiza de lo que este abajo

Gestión de errores detallada

Seguridad

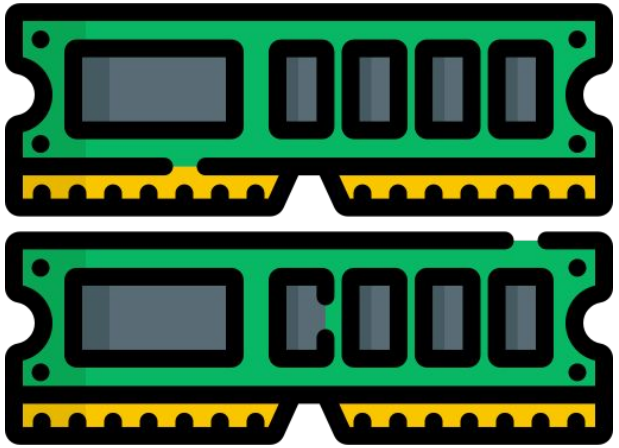
Como programa, podemos analizar la JVM en gran detalle

La JVM puede ejecutar código en otros lenguajes

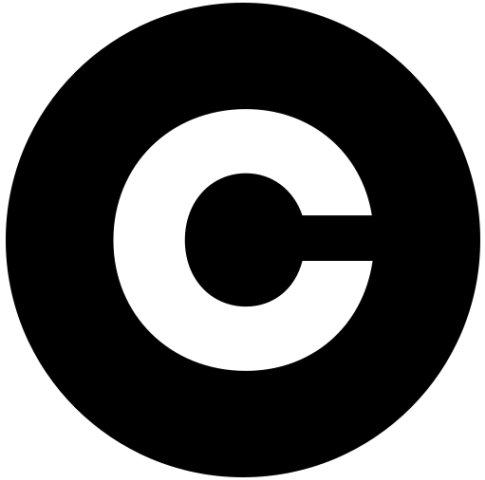


Pero

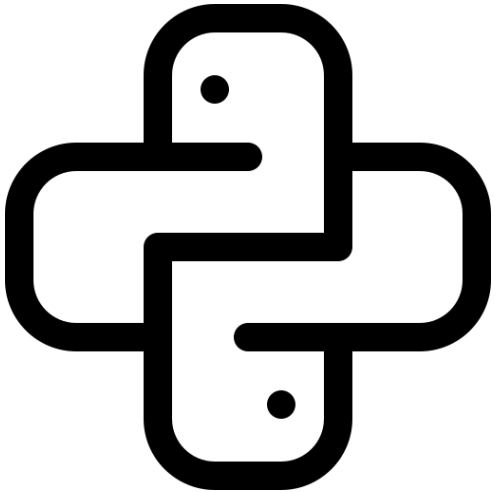
Requiere más memoria

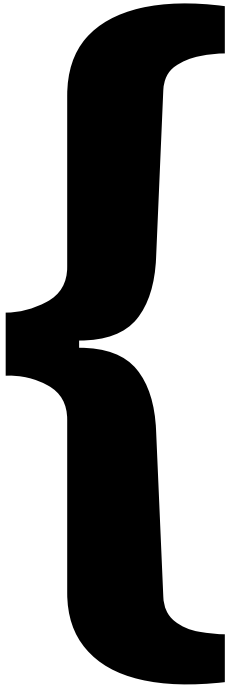


Es más lento que un programa en C

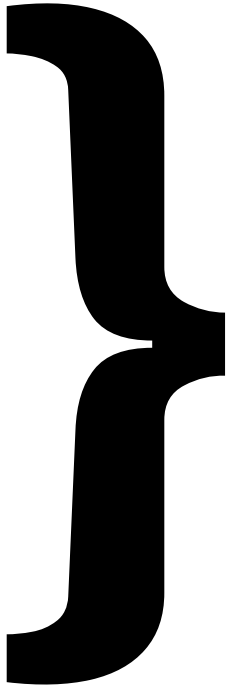


Aunque más rápido que uno en Python

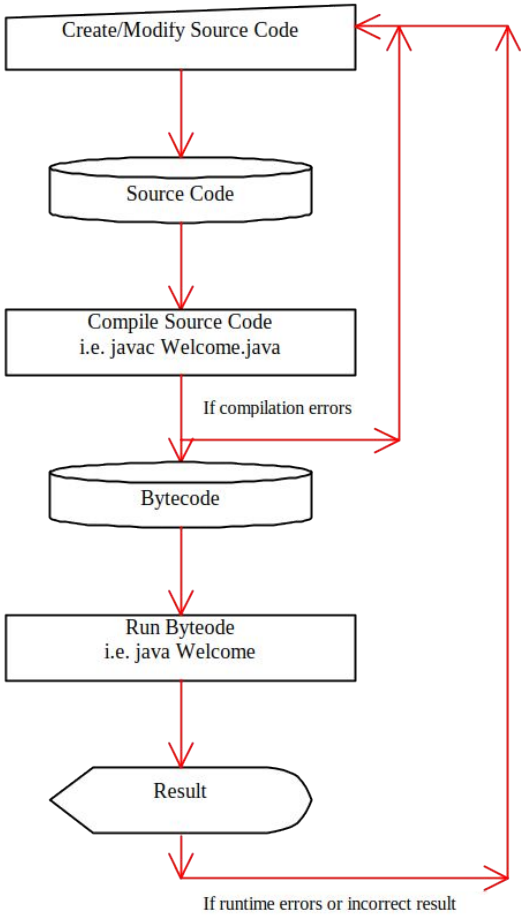




Está en el medio

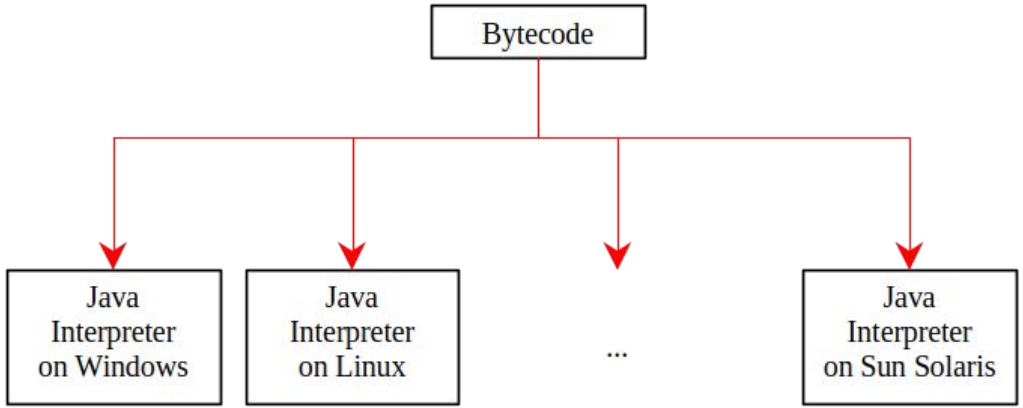



¿Preguntas?



Executing Applications

- On command line
  - **`java classname`**



Devices		
Access control systems	Airplane systems	ATMs
Automobiles	Blu-ray Disc™ players	Building controls
Cable boxes	Copiers	Credit cards
CT scanners	Desktop computers	e-Readers
Game consoles	GPS navigation systems	Home appliances
Home security systems	Internet-of-Things gateways	Light switches
Logic controllers	Lottery systems	Medical devices
Mobile phones	MRIs	Network switches
Optical sensors	Parking meters	Personal computers
Point-of-sale terminals	Printers	Robots
Routers	Servers	Smart cards
Smart meters	Smartpens	Smartphones
Tablets	Televisions	Thermostats
Transportation passes	TV set-top boxes	Vehicle diagnostic systems

Fig. 1.1 | Some devices that use Java.

Software Engineering Observation 1.1

Use a building-block approach to creating your programs. Avoid reinventing the wheel—use existing high-quality pieces wherever possible. This software reuse is a key benefit of object-oriented programming.

Scoring System "Java Fundamentals" Course

- Exam – 80%
- Labs – 10% (added to the exam results)
- Homework + evaluation – 5% + 5%
- Team work – 10%
- Bonuses – up to 10%
- Presence in class – 5% (onsite students only)

unrn.edu.ar

UNRN Universidad Nacional de Río Negro

| unrionegro