

Java en la Educación - Mejoras Recientes en Java ☕

Por qué deberías aprender
Java?

Por qué deberías enseñar
Java?

Prepared by Ken Fogel, the JCP Executive Committee (EC)
and, the Java in Education Working Group



DONDE DESCARGAR LA
PRESENTACIÓN

<https://bit.ly/3FVcUK0>



SCAN ME



BRAYAN MUÑOZ V.

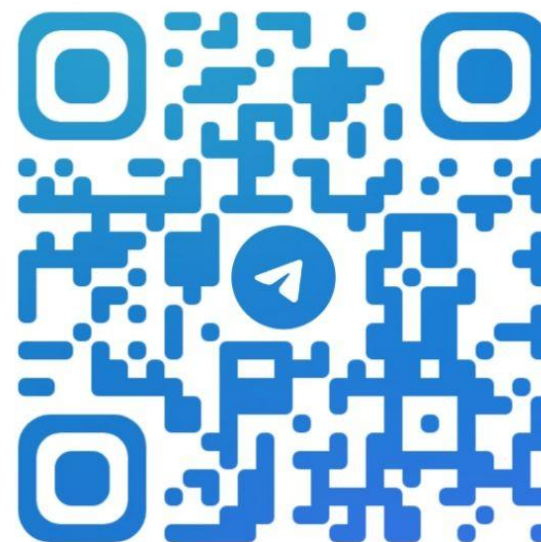
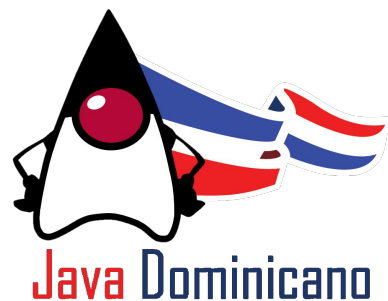
- **INGENIERO TELEMÁTICO**  →  **PUCMM**
Pontificia Universidad Católica
Madre y Maestra
- **MsEng CIENCIA DE DATOS**  (en curso) →  **Universidad
Camilo José Cela**
- **Miembro Java Dominicano** → 
- **Miembro organizador JConfDominicana** → 
- **Miembro asociado JCP** (Java Community Process) → 

**Aproximadamente 4 años de experiencia profesional
trabajando alrededor del ecosistema Java.**

Contribuyo ocasionalmente a proyectos FOSS.

**GRUPO DE USUARIOS
JAVA, REPÚBLICA
DOMINICANA**

***JAVA DOMINICANO
(DOMINICAN JUG)***



@JAVADOMINICANO



Mitos y Beneficios de Aprender Java ☕

Ejemplos en el directorio
example-1

- Java puede tener una **pronunciada** curva de aprendizaje para un principiante.
 - Sólo si el instructor en su etapa de aprendizaje también la tuvo.
- Java no es adecuado para tareas rápidas y ligeras.
 - Es más adecuado para aplicaciones **grandes y complejas**.
- *¿Has visto el Single-File Source-Code y en Linux has probado la ejecución de shebang?*
- **Recién incluido** → *Unnamed Classes and Instance Main Methods ([JEP 445](#))*

Mitos y Beneficios de Aprender Java ☕

- Java es un lenguaje **viejo** (Java 1996 & Python 1991)
 - *Significa que está establecido, ampliamente utilizado y bien documentado.*
- Java deriva su sintaxis de **C**
 - *Aprende Java, te hará aprender lenguajes como Javascript, C#, C++ e incluso Python más fácil.*

Java está
en todas
partes



#1
Programming
Language



12+
Million
Developers
Run Java



51 Billion
Active
Virtual Machines



30 Billion
Cloud Connected
Virtual Machines

JUGs around the World - Driving Adoption



Mitos y Beneficios de Aprender Java ☕

Ranking programming languages by energy efficiency

- Java es *lento*.

Table 4: Normalized global results for Energy, Time, and Memory

Total					
	Energy (J)		Time (ms)		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. P., & Saraiva, J. (2021). Ranking programming languages by energy efficiency.

Science of Computer Programming, 205, 102609. <https://doi.org/10.1016/j.scico.2021.102609>

Java Language Enhancements

Esta presentación analiza varias mejoras al lenguaje Java.

Estas mejoras ayudan a disipar algunos de los mitos que rodean a Java.

Se trata de por qué *Java* debería ser el lenguaje a enseñar en *todos los niveles educativos* al día de hoy.

Compararemos programas en Java, con Python.
(para fines didácticos)

JShell - Read-Evaluate- Print Loop (REPL)

JDK 9

Una herramienta para simplificar las instrucciones.

Ejecución a medida que ingresa el código y presiona ***Intro***.

Respuesta inmediata, línea a línea.

También puedes escribir tus métodos y luego ejecutarlos.

Ideal para enseñar o aprender Java una línea a la vez.

JEP 330 - Launch Single-File Source-Code Programs

JDK 11

- Aborda la sobrecarga de ejecutar código.
 - Estilo tradicional:
 - Dos pasos para la ejecución:
 - `javac` //compilar el código
 - `java -jar` //ejecutar el código
 - Single-File Source-Code Style
 - Un solo paso para la ejecución
 - `java` //se encarga de compilar y luego ejecutar
 - Si el archivo tiene una clase pública con un método ***main***, se compila y ejecuta.
 - Funciona también con preview-features.
 - Un solo archivo puede contener múltiples clases.
- Esta es una de las nuevas capacidades del lenguaje más importantes para aquellos que desean aprender el lenguaje.

var – Mejorar la experiencia del desarrollador reduciendo toda la ceremonia asociada con la escritura de código Java.

JEP 286 - Local-Variable Type Inference

JDK 10

No más:

- `MyClass m = new MyClass();`

Ahora se convierte
en:

- `var m = new MyClass();`

Solo para objetos
con inicializadores

- Reducirá la aparición de nuestros amigos `NullPointerException`

Son una nueva forma de String Literals que es particularmente adecuada para texto de varias líneas.

JEP 378 - JDK 15 Text Blocks

Finalmente, lo que ingresas en tu código fuente es lo que obtienes.

Especialmente útil para cadenas que contienen HTML, XML y JSON.

¿A quién no le gusta escribir tres comillas seguidas?

''''''

''''''

Concatenación de la vieja forma

```
String htmlStr = "<html><head><link rel='stylesheet' "  
    + "href='styles/main.css' "  
    + "type='text/css'/><title>The Learning Servlet</title></head>"  
    + "<body><h1>GET method</h1>"  
    + "<form id='form:index' action = 'index.html'>"  
    + "<br/><input type= 'submit' value='Return to Home page' /></form>"  
    + "</body></html>";
```


Bloque de Texto - Nueva forma

```
String htmlStr = ""
<html>
  <head>
    <link rel='stylesheet'href='styles/main.css' type='text/css' />
    <title>The Learning Servlet</title>
  </head>
  <body>
    <h1>GET method</h1>
    <form id='form:index' action = 'index.html'>
      <br/>
      <input type= 'submit' value='Return to Home page' />
    </form>
  </body>
</html>""
```

switch – una expresión y sin *break*

Un *switch* que se puede explicar con sensatez

Reducción de la duplicación de código cuando se utiliza para establecer un valor

El *switch* ya no necesita ser el original *switch* de C.

El *break* llegó a su fin, todos los casos terminan!

Cual prefieres aprender o enseñar

```
double value = 0;
switch (point) {
    case NORTH:
        value = 12.12;
        break;
    case SOUTH:
        value = 14.14;
        break;
    case EAST:
        value = 16.16;
        break;
    case WEST:
        value = 18.18;
        break;
}
```

```
double value = switch (point) {
    case NORTH -> 12.12;
    case SOUTH -> 14.14;
    case EAST -> 16.16;
    case WEST -> 18.18;
    default -> 0.0;
};
```

records – Reducción del boilerplate requerido con inmutabilidad agregada al manejo de la data.

Los objetos de datos son conocidos por su código repetitivo:

- Inicializar constructores, setters, getters, equals, hashCode, and toString

Al rescate, llegan los *records* inmutables

Más que una simple simplificación de un bean

Es el camino hacia objetos inmutables por defecto

Y luego está el constructor compacto.

- Validando los valores iniciales sin separar el constructor.

No setters, solo simples getters
Free equals, hashCode and toString
Y, un compacto constructor para validación

```
public record Person(String firstName,  
                    String lastName,  
                    int age,  
                    String position,  
                    LocalDate birthday) {  
    public Person{  
        if (age < 18) {  
            throw new IllegalArgumentException( "Too young to work for us!");  
        }  
    }  
}
```

Que está dejando a Java a un lado?

JavaScript

- Pequeño para descargar.
- Disponible en todos los navegadores de cada computador.
- Numerosos entornos de programación en línea.

Python

- Asociado a las dos grandes tendencias:
 - Big Data
 - AI/ML
- Jupyter notebook en línea son populares.

Por qué Python está ganando popularidad en la educación?

El patrón de diseño más temido:

- Flujo de conciencia

El programa fluye a medida que las tareas vienen a mi.

Atrae a personas que necesitan codificar pero que no necesariamente quieren aprender a codificar profesionalmente.

Comparemos Python con Java.

Para fines didácticos...

En las siguientes diapositivas está el mismo programa en Python y Java.

Estos programas solicitan tres valores decimales (float).

- Cantidad de dinero prestada llamada loan.
- La tasa de porcentaje anual (APR) de interés sobre el dinero prestado.
- La duración del préstamo expresada en meses se llama plazo (*term*).

A partir de estos valores el programa calcula el pago mensual y lo muestra.

Basic Python

```
loan = input("          loan: ")
interest = input("        interest: ")
term = input("          term: ")
tempInterest = float(interest) / 12;
result = float(loan)*(tempInterest / (1 - ((1 + tempInterest) ** -float(term))));
print("Monthly Payment: %.2f" % result)
```

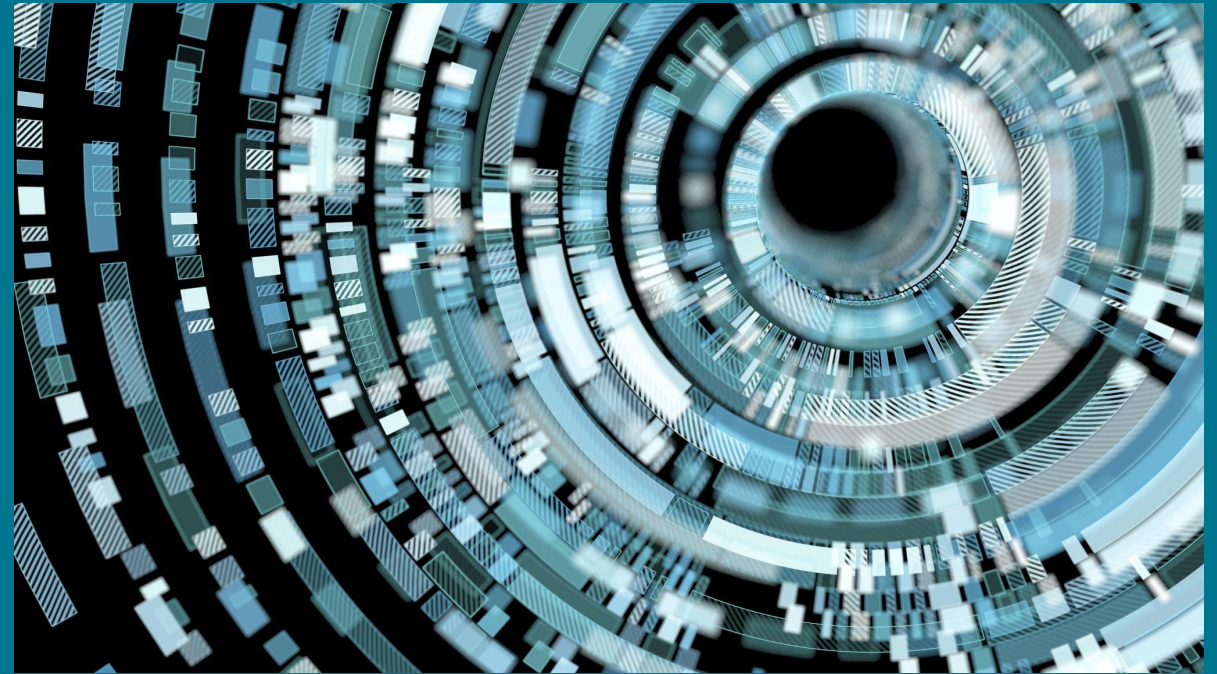
```
import java.util.Scanner;

public class JavaCalculator01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("          Loan: ");
        double loan = sc.nextDouble();
        System.out.print("          Interest: ");
        double interest = sc.nextDouble();
        System.out.print("          Term: ");
        double term = sc.nextDouble();
        double tempInterest = interest / 12.0;
        double result = loan *
            (tempInterest / (1.0 - Math.pow((1.0 + tempInterest), -term)));
        System.out.println("Monthly Payment: " + String.format("%.2f", result));
    }
}
```

Machine Learning and VisRec

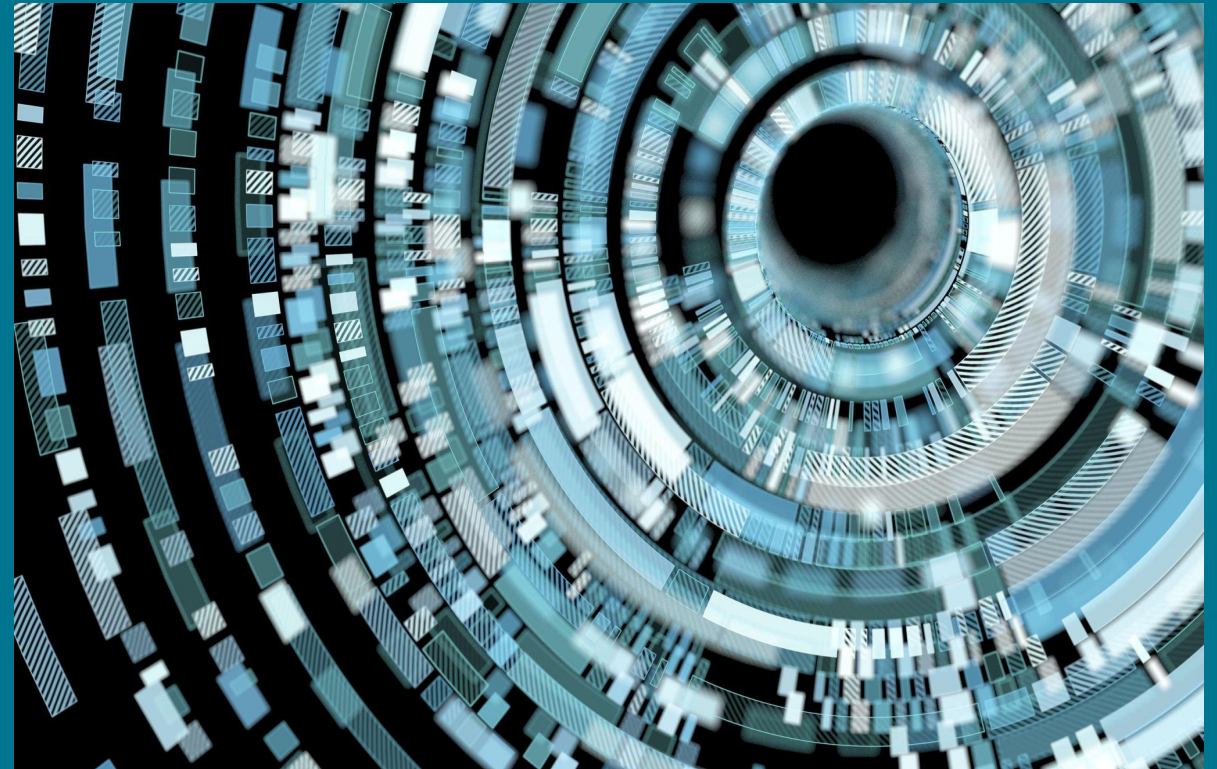
JSR 381

- Java está haciendo aprendizaje automático ahora!
- La biblioteca Deep Java (DJL) de Amazon es una de varias implementaciones de este nuevo JSR
- La profundidad y amplitud de las herramientas Java la convierten en la mejor plataforma para ML



Why is Python widely used for AI/ML?

- El lenguaje Python está escrito en C.
- La mayoría de librerías AI/ML están escritas en C.
- Esto simplifica el uso de estas librerías en Python.
- Desde Java 16 tenemos:
Foreign Linker API & Foreign Memory Access API que simplifican el acceso a librerías escritas en C.



La máquina
virtual Java
JVM:

hogar de más
que Java

- Kotlin, Scala, Groovy, Clojure y más.
- Incluso existe un Python llamado **Jython** que se ejecuta en la JVM y admite interoperabilidad entre Java y Python.

Cuáles son tus expectativas laborales si aprendes Java?



Muchas instituciones financieras dependen de Java para ejecutar su core bancario.



Twitter, LinkedIn, Amazon y otros usan Java.



Tus expectativas son una función de qué tan bien puedes escribir código.

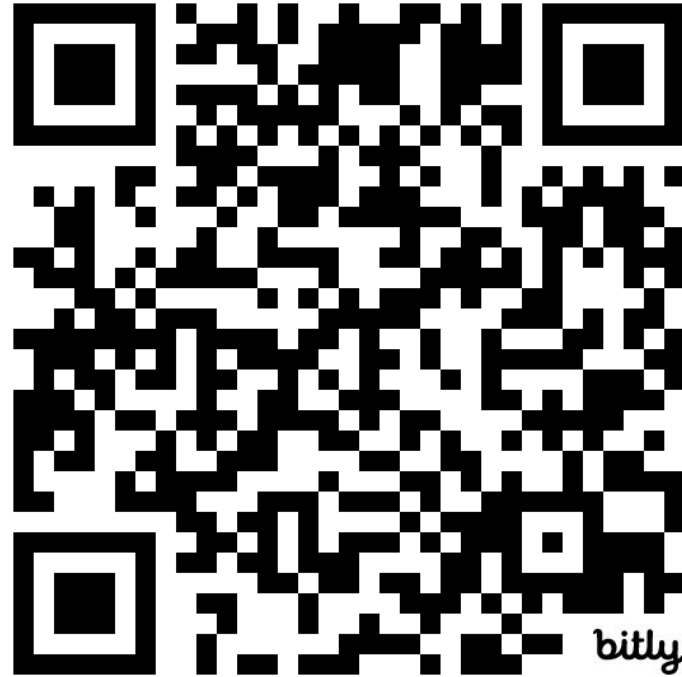


Aprender Java es el mejor idioma para aprender y prepararte para trabajar con cualquier idioma durante tu carrera.



Es el mejor lenguaje para enseñar y brindarles a los estudiantes una comprensión clara de lo que significa programar.

El código fuente lo pueden
encontrar en:



<https://bit.ly/475Yym7>