

Vida más allá de Java 8

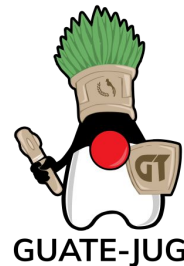


Jorge Cajas



@cajasmota

- JUG co-leader en GuateJUG
- Desarrollador Java certificado con +9 años de experiencia.
- Ganador del Duke Choice Award en 2016 con GuateJUG
- Parte del comité organizador del JConf Guatemala
- Conferencista desde 2012 en eventos nacionales e internacionales.
- Manager en MangoChango S.A.



Contenido

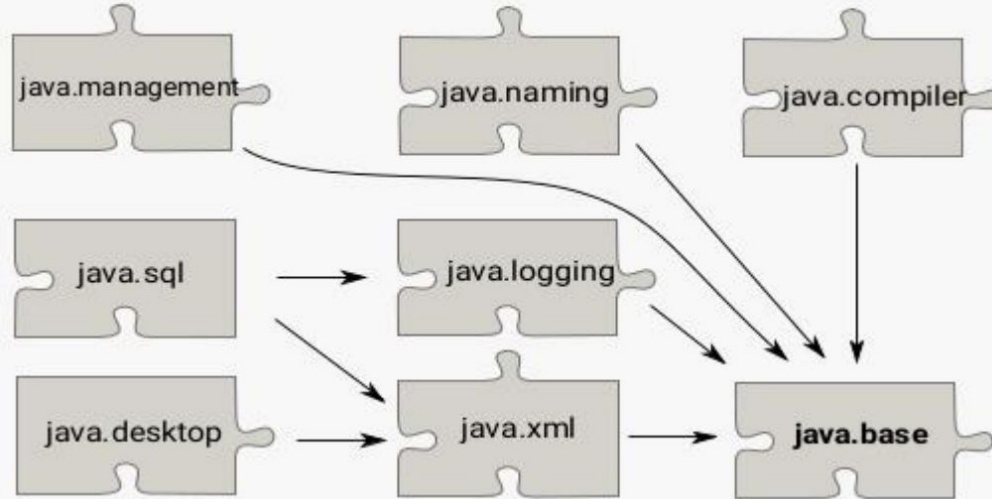
- **Java 9**
- **Java 10**
- **Java 11**
- **Java 12**
- **Java 13**
- **Java 14**
- **Java 15**
- **Java 16**
- **Java 17**
- **Java 18**
- **Java 19**
- **Java 20**

Java 9

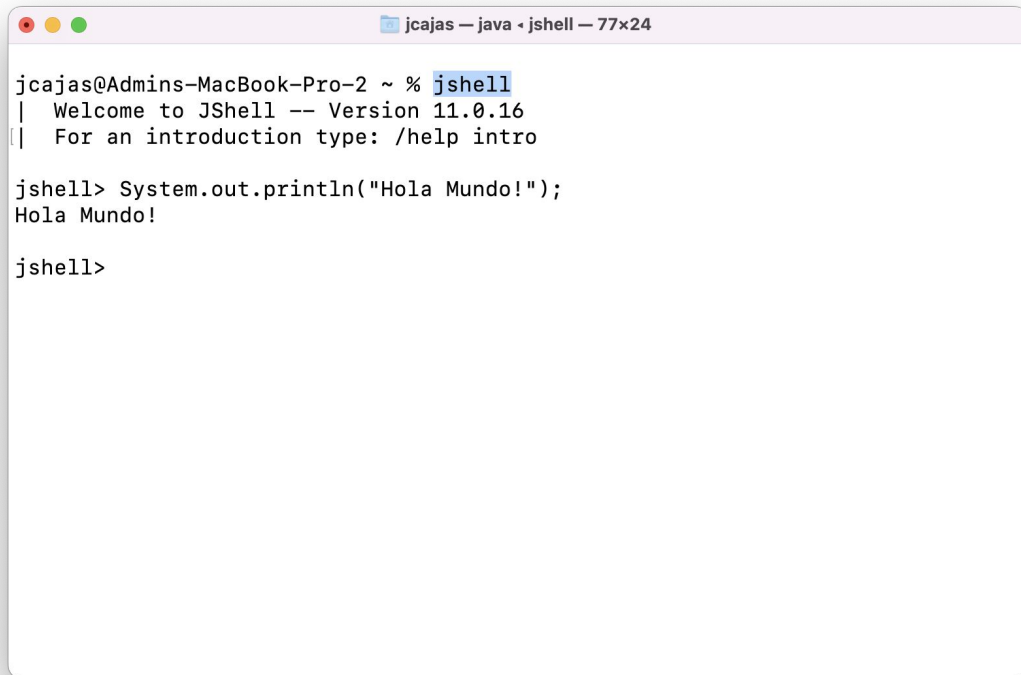
- **91 JEP aceptados**
- **JSR 376 Jigsaw:** Sistema de módulos
- **JEP 222 JShell:** Java REPL
- **JEP 213:** Métodos privados en interfaces
- **JEP 269:** Factory Methods para crear colecciones inmutables
- **JEP 282 JLink:** JVM Mínimas

JSR 376 - Jigsaw

Modular JDK



JEP 222 - JShell



```
jcajas@Admins-MacBook-Pro-2 ~ % jshell
| Welcome to JShell -- Version 11.0.16
[| For an introduction type: /help intro

jshell> System.out.println("Hola Mundo!");
Hola Mundo!

jshell>
```

JEP 213 - Interface private methods

```
public interface MyInterface {  
  
    void action();  
  
    default void defaultMethod(){  
        magic();  
    }  
  
    1 usage  
    private void magic(){  
        //Magic goes here  
    }  
}
```

JEP 269 - Collections factory methods

```
public class CollectionsFactoryMethods {
```

Estas colecciones son inmutables!

```
    void example(){
```

```
        List<String> strings = List.of("a", "b", "c");
```

```
        Map<String, String> emptyMap = Map.of();
```

```
        Map<String, String> map = Map.of( k1: "key", v1: "value");
```

```
        Stream<String> stream = Stream.of( ...values: "a", "b", "c");
```

```
    }
```

```
}
```


Java 10

- **12 JEP aceptados**
- **JEP 286:** Inferencia de tipos
- **JEP 317:** Java Based JIT Compiler - Graal

JEP 286 - Type Inference

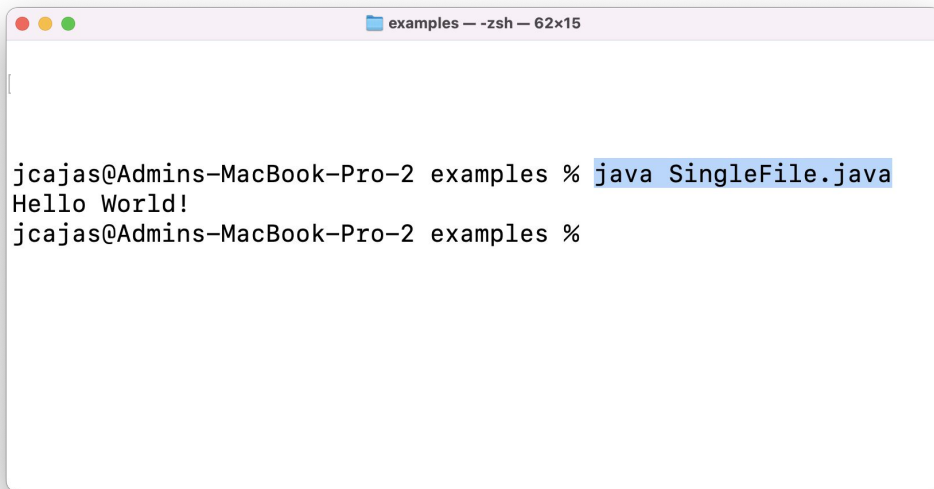
```
public class TypeInference {  
  
    var name = "Jorge Cajas"; // <----- Only valid on methods  
  
    Map<Integer, String> example(){  
  
        var key = 1;  
        var value = "name";  
        var map : Map<Integer, String> = Map.of(key, value);  
  
        return map;  
    }  
  
}
```

Java 11

- **Versión LTS**
- **17 JEP aceptados**
- **JEP 320:** JavaEE Y CORBA removidos del JDK
- **JEP 330:** Ejecución directa de archivos .java
- **JDK Changes:** Strings, Files methods

JEP 330 - *.java file execution

```
public class SingleFile {  
  
    public static void main(String ... args){  
        System.out.println("Hello World!");  
    }  
}
```



A terminal window titled "examples -- zsh -- 62x15" showing the execution of the Java file. The prompt is "jcajas@Admins-MacBook-Pro-2 examples %". The command "java SingleFile.java" is entered and highlighted in blue. The output "Hello World!" is displayed on the next line. The prompt "jcajas@Admins-MacBook-Pro-2 examples %" is shown again on the following line.

```
jcajas@Admins-MacBook-Pro-2 examples % java SingleFile.java  
Hello World!  
jcajas@Admins-MacBook-Pro-2 examples %
```

String::IsBlank

```
void example1(){  
  
    String s1 = "    ";  
    System.out.println( s1.isBlank() ); // True  
  
    String s2 = "Hello";  
    System.out.println( s2.isBlank() ); // False;  
  
    String s3 = "";  
    System.out.println( s3.isBlank() ); // True  
  
}
```

String::lines

```
void example2(){  
    String s1 = "Hello \n World!";  
  
    System.out.println( s1.lines().collect(Collectors.toList()));  
    // [Hello, World]  
}
```

String::strip*

```
void example3(){  
  
    String s1 = "    Hello    ";  
  
    System.out.println("Start" + s1.strip() + "End");  
    //StartHelloEnd  
  
    System.out.println("Start" + s1.stripLeading() + "End");  
    //StartHello    End  
  
    System.out.println("Start" + s1.stripTrailing() + "End");  
    //Start    HelloEnd  
}
```

String::repeat

```
void example4(){  
  
    String s1 = "Hello";  
    System.out.println( s1.repeat(count: 3));  
    // HelloHelloHello  
  
}
```


Files::writeString

```
void example1() throws IOException {  
    Path path = Files.createTempFile(prefix: "example", suffix: "txt");  
    String content = "Hello World!";  
  
    Files.writeString(path, content);  
}
```

Files::readString

```
void example2() throws IOException {  
  
    Path path = Path.of( first: "/path/to/file.csv");  
    String content = Files.readString(path);  
  
    content.lines().forEach(record ->{  
        //Do Magic  
    });  
  
}
```

Java 12

- **8 JEP aceptados**
- **JEP 189:** Shenadona Garbage Collector
- **JEP 325:** Switch Expressions (Preview)

Java 13

- **5 JEP aceptados**
- **JEP 354:** Switch Expressions (Preview)
- **JEP 355:** Text Blocks (Preview)

Java 14

- **16 JEP aceptados**
- **JEP 305:** Pattern Matching for instanceof (Preview)
- **JEP 358:** Helpful NullPointerExceptions
- **JEP 361:** Switch Expressions

JEP 358 - Helpful NullPointerExceptions

```
a.i = 99;
```

```
//Exception in thread "main" java.lang.NullPointerException at Prog.main(Prog.java:2)
```

```
a.b.c.d = 99;
```

```
//Exception in thread "main" java.lang.NullPointerException at Prog.main(Prog.java:7)
```

JEP 358 - Helpful NullPointerExceptions

```
a.b.c.d = 99;
```

```
/*
```

```
Exception in thread "main" java.lang.NullPointerException:  
    Cannot assign field "c" because "b" is null  
    at Prog.main(Prog.java:5)
```

```
*/
```

JEP 361 - Switch Expressions

```
String java8(int day) {  
  
    String isWeekend = "";  
  
    switch (day) {  
        case MONDAY:  
        case TUESDAY:  
        case WEDNESDAY:  
        case THURSDAY:  
            isWeekend = "Nope";  
            break;  
        case FRIDAY:  
            isWeekend = "Almost";  
            break;  
        case SATURDAY:  
        case SUNDAY:  
            isWeekend = "Yes!";  
    }  
  
    return isWeekend;  
}
```

```
String java14(int day) {  
  
    String isWeekend = switch (day) {  
        case MONDAY, TUESDAY, WEDNESDAY, THURSDAY -> "Nope";  
        case FRIDAY -> "Almost";  
        case SATURDAY, SUNDAY -> "Yes!";  
  
        default -> {  
            System.out.println("Invalid day number");  
            yield "error";  
        }  
    };  
  
    return isWeekend;  
}
```


Java 15

- **14 JEP aceptados**
- **JEP 360:** Sealed Classes (Preview)
- **JEP 378:** Text Blocks (Preview)

JEP 378 - Text Blocks

```
void java8(){  
    String html = "<html>\n" +  
        "    <body>\n" +  
        "        <p>Hello, world</p>\n" +  
        "    </body>\n" +  
        "</html>\n";  
  
    String query = "SELECT \"EMP_ID\", \"LAST_NAME\" FROM \"EMPLOYEE_TB\"\n" +  
        "WHERE \"CITY\" = 'INDIANAPOLIS'\n" +  
        "ORDER BY \"EMP_ID\", \"LAST_NAME\";\n";  
}
```

JEP 378 - Text Blocks

```
void java15(){  
    String html = ""  
        <html>  
            <body>  
                <p>Hello, world</p>  
            </body>  
        </html>  
        "";  
  
    String query = ""  
        SELECT "EMP_ID", "LAST_NAME" FROM "EMPLOYEE_TB"  
        WHERE "CITY" = 'INDIANAPOLIS'  
        ORDER BY "EMP_ID", "LAST_NAME";  
        "";  
}
```

Java 16

- **17 JEP aceptados**
- **JEP 357:** Migrate from Mercurial to Git / **JEP 369:** Migrate to GitHub
- **JEP 388:** Windows/AArch64 Port
- **JEP 394:** Pattern Matching for instanceof
- **JEP 395:** Records

JEP 394 - Pattern Matching for instanceof

```
void java8(Object obj){  
    if (obj instanceof String) {  
        String s = (String) obj;    // Cast to String needed  
        Stream<String> lines = s.lines();  
        //Do something....  
    } else if (obj instanceof Integer) {  
        Integer integer = (Integer) obj; // Cast to Integer needed  
        integer.toString();  
        //Do something....  
    }  
}
```

JEP 394 - Pattern Matching for instanceof

```
void java16(Object obj){  
    if (obj instanceof String s) {  
        Stream<String> lines = s.lines();  
        //Do something....  
    } else if (obj instanceof Integer integer) {  
        integer.toString();  
        //Do something....  
    }  
}
```

JEP 395 - Records

```
class Point {  
    6 usages  
    private final int x;  
    6 usages  
    private final int y;  
  
    Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    int getX() { return x; }  
    int getY() { return y; }  
  
    public boolean equals(Object o) {...}  
  
    public int hashCode() { return Objects.hash(x, y); }  
  
    public String toString() { return String.format("Point[x=%d, y=%d]", x, y); }  
}
```

JEP 395 - Records

```
record Point(int x, int y) { }
```


Java 17

- **Versión LTS**
- **14 JEP aceptados**
- **JEP 391:** macOS/AArch64 Port
- **JEP 409:** Sealed Classes

JEP 409 - Sealed Classes

```
package com.example.geometry;

public abstract sealed class Shape
    permits com.example.polar.Circle,
            com.example.quad.Rectangle,
            com.example.quad.simple.Square { ... }
```

Java 18

- **9 JEP aceptados**
- **JEP 400:** UTF-8 by Default
- **JEP 408:** Simple Web Server
-

JEP 408 - jwebserver

← → ↻ ⓘ 127.0.0.1:8000

Directory listing for /

- [MyInterface.java](#)
- [Point.java](#)
- [TextBlocks.java](#)
- [StringMethods.java](#)
- [FileMethods.java](#)
- [SingleFile.java](#)
- [CollectionsFactoryMethods.java](#)
- [TypeInference.java](#)
- [PatternMatching.java](#)
- [TypeInferenceLambda.java](#)
- [SwitchExpressions.java](#)

```
examples — jwebserver — 82x15

jcajas@Admins-MacBook-Pro-2 examples % jwebserver
Binding to loopback by default. For all interfaces use "-b 0.0.0.0" or "-b ::".
Serving /Users/jcajas/Documents/Projects/Conferences/mp-fault-tolerance/src/main/j
ava/motojo/examples and subdirectories on 127.0.0.1 port 8000
URL http://127.0.0.1:8000/
127.0.0.1 -- [28/Jul/2023:02:45:23 -0500] "GET / HTTP/1.1" 200 -
127.0.0.1 -- [28/Jul/2023:02:45:23 -0500] "GET /favicon.ico HTTP/1.1" 404 -
```

Java 19

- **7 JEP aceptados**
- **JEP 425:** Virtual Threads (Preview)

Java 20

- **7 JEP aceptados**
- **JEP 436:** Virtual Threads (Second Preview)

¿Preguntas?



Vida más allá de Java 8

Gracias



Vida más allá de Java 8