

Java 9, 10, 11, 12 & 13

Vida más allá de Java 8

¿ Quien ?



Jorge Cajas
@cajasmota



<https://github.com/Motojo/slides>

Contenido



Temas

- Java 9
 - Cambios
 - Módulos
 - JShell
 - Collections Factory Methods
 - Metodos privados en interfaces
- Java 10
 - Cambios
 - Inferencia de Tipos
- Java 11
 - Cambios
 - String Methods
 - File Methods
 - Ejecucion directa de archivos .java
- Java 12
 - Cambios
 - Switch Expression
- Java 13
 - Cambios
 - Switch Exression / Yield
 - Text Blocks
- Preguntas



Java 9

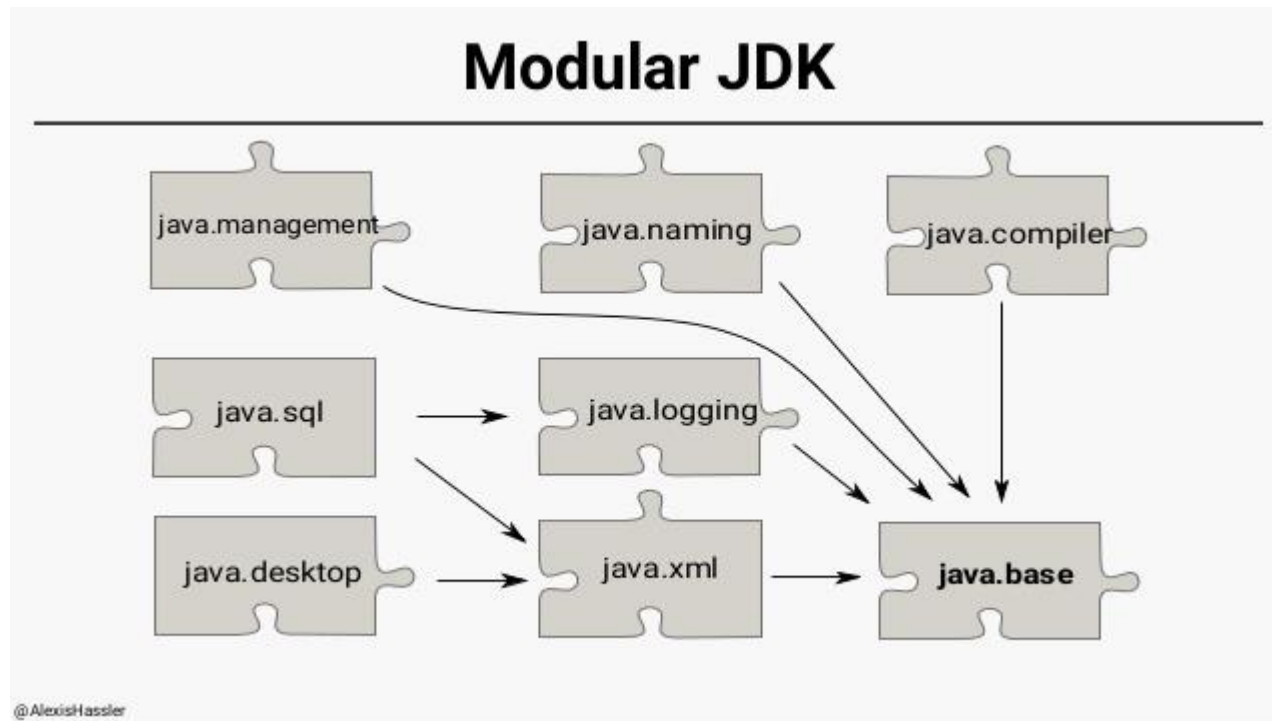
Cambios

- **91 JEP aceptados**
- **JSR 376 Jigsaw:** Sistema de módulos
- **JEP 213:** Métodos privados en interfaces, mejoras a try-with-resources
- **JEP 222 JShell:** Java REPL
- **JEP 223:** Sistema de versiones `$MAJOR.$MINOR.$SECURITY.$PATCH`
- **JEP 269:** Factory Methods para crear colecciones inmutables
- **JEP 282 JLink:** JVM Mínimas

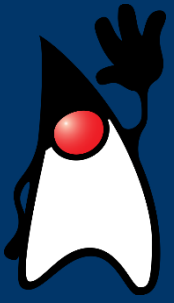
Java 9



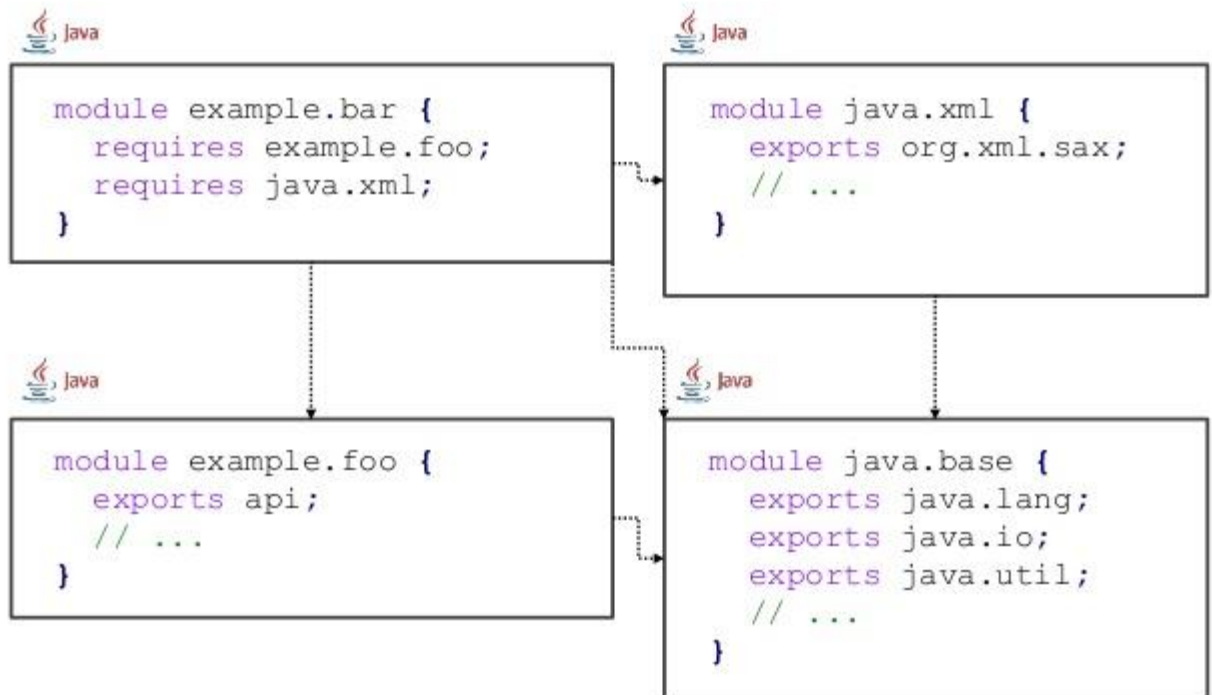
Módulos



Java 9



Módulos





Java 9

JShell

```
C:\Windows\system32\cmd.exe - jshell

c:\Program Files\Zulu\zulu-11\bin>jshell
| Welcome to JShell -- Version 11.0.2
| For an introduction type: /help intro

jshell> System.out.println("Hello world!");
Hello World!

jshell> _
```

Java 9



Collections Factory Methods

```
C:\Windows\system32\cmd.exe - jshell
jshell> List list = List.of();
list ==> []

jshell> List list2 = List.of(1,2,3,4,5)
list2 ==> [1, 2, 3, 4, 5]

jshell> Map map = Map.of();
map ==> {}

jshell> Map map = Map.of(1,"Uno",2,"Dos",3,"Tres")
map ==> {2=Dos, 1=Uno, 3=Tres}

jshell>
```




Java 9

JEP 213: Métodos privados en interfaces

```
public interface MyInterface{  
  
    void action();  
  
    default void defaultMethod(){  
        |    magic();  
    }  
  
    private void magic(){  
        |    //Magic occurs here  
    }  
}
```



Java 9

JEP 213: try-with-resources

```
void java7() throws IOException
{
    BufferedReader reader1 = new BufferedReader(new FileReader( fileName: "file.txt"));
    try (BufferedReader reader2 = reader1) {
        System.out.println(reader2.readLine());
    }
}
```

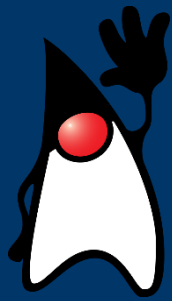
```
void java9() throws IOException
{
    BufferedReader reader1 = new BufferedReader(new FileReader( fileName: "file.txt"));
    try (reader1) {
        System.out.println(reader1.readLine());
    }
}
```

Java 10



Cambios

- **12 JEP aceptados**
- **JEP 286:** Inferencia de tipos
- **JEP 317:** Java Based JIT Compiler - Graal
- **JEP 322:** Sistema de versiones `$FEATURE.$INTERIM.$UPDATE.$PATCH`



Java 10

Inferencia de Tipos

```
private String doMagic() {  
    var magic = "Abra kadabra";  
    //more magic  
    //more complex magic  
    //even more complex magic  
    return magic;  
}
```

Java 11



Cambios

- **Version LTS**
- **17 JEP aceptados**
- **JEP 320:** JavaEE Y CORBA removidos del JDK
- **JEP 323:** Inferencia de tipos en lambdas
- **JEP 330:** Ejecución directa de archivos .java
- **JDK Changes:** Strings, Files methods



Java 11

Inferencia de Tipos en Lambdas

```
private void java8 () {  
  
    BiConsumer<BigDecimal, BigDecimal> example1;  
    BiConsumer<BigDecimal, BigDecimal> example2;  
  
    example1 = (BigDecimal a, BigDecimal b) -> a.add(b);  
    example2 = (a, b) -> a.add(b);  
}
```

```
private void java11 () {  
  
    BiConsumer<BigDecimal, BigDecimal> example1;  
  
    example1 = (var a, var b) -> a.add(b);  
}
```

Java 11



Ejecución directa de archivos .java

```
1
2 public class SingleFile
3 {
4     public static void main(String ... args){
5         System.out.println("Hello "+ args[0]);
6     }
7 }
8
9
```

Java 11



Ejecución directa de archivos .java

A screenshot of a Windows command prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The command prompt is at the directory "c:\Program Files\Zulu\zulu-11\bin". The user has entered the command "java C:/ejemplo/SingleFile.java Jorge". The output of the command is "Hello Jorge". The prompt is now waiting for the next command.

```
C:\Windows\system32\cmd.exe
c:\Program Files\Zulu\zulu-11\bin>java C:/ejemplo/SingleFile.java Jorge
Hello Jorge
c:\Program Files\Zulu\zulu-11\bin>
```




Java 11

String Methods

- `String::isBlank`

```
String s1 = " ";  
System.out.println(" ".isBlank()); //true
```

```
String s2 = "Hello";  
System.out.println(s.isBlank()); //false
```

```
String s3 = "";  
System.out.println(s1.isBlank()); //true
```



Java 11

String Methods

- `String::lines`

```
String s4 = "Hola \n Mundo";  
System.out.println(s4);  
//Hola  
//Mundo
```

```
System.out.println(s4.lines().collect(Collectors.toList()));  
//[Hola, Mundo]
```



Java 11

String Methods

- `String::strip`
- `String::stripLeading`
- `String::stripTrailing`

```
String s5 = " Hola ";  
System.out.print("Start");  
System.out.print(s5.strip());  
System.out.println("End");  
// StartHolaEnd
```

```
System.out.print("Start");  
System.out.print(s5.stripLeading());  
System.out.println("End");  
// StartHola End
```

```
System.out.print("Start");  
System.out.print(s5.stripTrailing());  
System.out.println("End");  
// Start HolaEnd
```



Java 11

String Methods

- String::repeat

```
String str = "=".repeat(3);  
System.out.println(str);  
// ==
```

Java 11



Files Methods

- Files::writeString
- Files::readString

```
String str = "Hola Mundo";  
Path path = Files.writeString(Files.createTempFile("test", ".txt"), str);  
  
String s = Files.readString(path);  
System.out.println(s); //Hola Mundo
```



Java 12

Cambios

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

Java 12



Switch Expressions

```
public String java8(int day) {  
    String str;  
  
    switch (day) {  
        case 2: case 3: case 4: case 5: case 6: {  
            attr = "weekday";  
            break;  
        }  
        case 7: case 1: {  
            attr = "weekend";  
            break;  
        }  
        default: {  
            attr = "invalid";  
        }  
    }  
    return str;  
}
```

Java 12



Switch Expressions

```
public String java12(int day) {  
  
    String str = switch (day) {  
        case 2,3,4,5,6 -> "weekday";  
        case 7,1: -> "weekend";  
        default -> "Invalid";  
    }  
    return str;  
}
```




Java 13

Cambios

- **5 JEP aceptados**
- **JEP 353:** Reimplement the Legacy Socket API
- **JEP 354:** Switch Expressions (Preview)
- **JEP 355:** Text Blocks (Preview)

Java 13



Switch Expressions

```
private static int java12(String mode) {  
    int result;  
    switch (mode) {  
        case "a":  
        case "b":  
            result = 1;  
            break;  
        case "c":  
            result = 2;  
            break;  
        case "d":  
        case "e":  
            result = 3;  
            break;  
        default:  
            result = -1;  
    };  
    return result;  
}
```

Java 13



Switch Expressions

```
private static int java13(String mode) {  
    int result = switch (mode) {  
        case "a", "b":  
            yield 1;  
        case "c":  
            yield 2;  
        case "d", "e", "f":  
            yield 3;  
        default:  
            yield -1;  
    };  
    return result;  
}
```

Java 13



Switch Expressions

```
int value = switch (greeting) {  
    case "hi" -> {  
        System.out.println("I am not just yielding!");  
        yield 1;  
    }  
    case "hello" -> {  
        System.out.println("Me too.");  
        yield 2;  
    }  
    default -> {  
        System.out.println("OK");  
        yield -1;  
    }  
};
```



Java 13

Text Blocks

```
String multiline = "line 1\nline 2\nline 3\n";
```

```
multiline = "line 1\n" +  
            "line 2\n" +  
            "line 3\n";
```

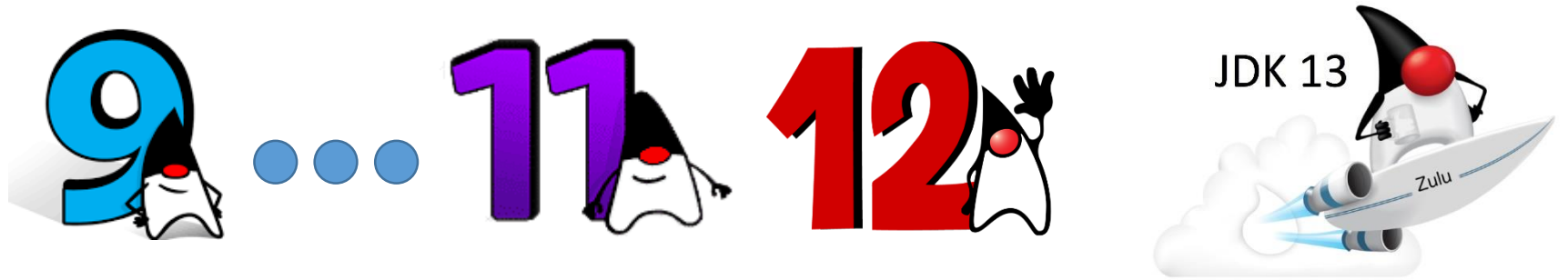


Java 13

Text Blocks

```
multiline = """  
line 1  
line 2  
line 3  
"""
```

```
String html = """  
    <html>  
        <body>  
            <p>Hello, world</p>  
        </body>.  
    </html>  
""";
```



Java 9, 10, 11, 12 & 13

¿Preguntas?

¡Muchas Gracias!



Jorge Cajas
@cajasmota



<https://github.com/Motojo/slides>