

# Actualizando aplicaciones empresariales en Java desde Java 8 on premise hasta Java 11 en la nube

---

Víctor Orozco - @tuxtor

21 de septiembre de 2020

Nabenik

Historias del mundo real

De Java 8 a Java 11

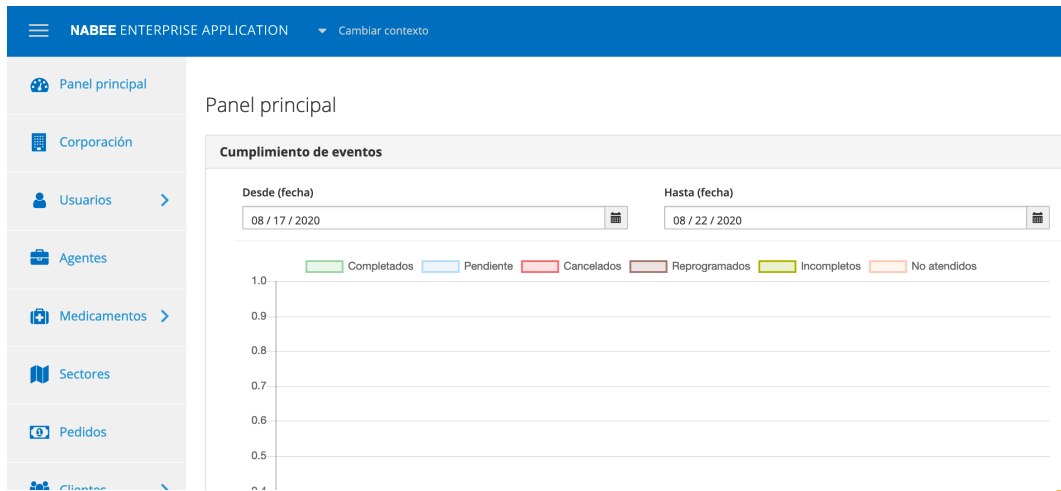
De mi data center a la nube - moderna -

Desde Java 14 hasta el infinito

## Historias del mundo real

A decorative horizontal line in a light cream color spans the width of the slide. It features a small blue segment on the left side and a white circle at the right end.

# Un sistema de geocerca



# Un sistema de geocerca

---

- 2017
- 5 modulos  
(War/Microservicios)
- 348 clases
- 17160 lineas de código  
+ dependencias
- Original: Glassfish 4, Java EE 7, Java 8
- Actual: Payara 5, Jakarta EE 8, MicroProfile 3.2
- Cliente Android y web  
(Angular)

# Un sistema de geocerca

---

```
commit 7b5b62ef84a0363b44efa6ed69eec5b4a859da1f
```

```
Author: Victor Orozco <vorozco@nabenik.com>
```

```
Date: Tue Jul 4 16:24:52 2017 -0600
```

Eclipse project definition

```
commit 78f5767a0971b9c279197721f0de9e76d2bf331d
```

```
Author: Victor Orozco <vorozco@nabenik.com>
```

```
Date: Tue Jul 4 17:01:35 2017 -0600
```

Adding glassfish generic data source definition for development

# Un sistema de geocerca

---

```
commit baae2327fa9185566abffb0ff87d52cf2cc3286a (HEAD -> develop, origin/develop)
Author: Esvin González <egonzalez@nabenik.com>
Date: Tue Jul 28 18:54:45 2020 -0600
```

Applying correct format (spaces instead of tabs).

# Un - sistemita - contable y empresarial



## Nabenik Enterprise

is a enterprise Software provider, mixing world class software with innovation

v3.24.0

Sign in

User

Password

Sign in





# Un - sistemita - contable y empresarial

---

- 10 modulos (War, EJB, EAR)
- 671 clases
- 39480 lineas de código + dependencias
- 2014
- Original: Wildfly 8, Java EE 7, Java 7
- Actual: Wildfly 17, Jakarta EE 8, MicroProfile 3.0
- Cliente web (AngularJS)

```
commit 7e7cb04ebd892d8065a43512cf200598e311bdd3
```

```
Author: Víctor Orozco <tuxtor@shekalug.org>
```

```
Date:   Wed Nov 12 12:40:31 2014 -0600
```

```
    Creating rest crud example
```

```
commit 74f184b0353bc94030858378bf3b030d23a5be07
```

```
Author: Luis Pedro <lestrada@nabenik.com>
```

```
Date:   Wed Nov 12 13:43:15 2014 -0600
```

```
Master
```

# Un - sistemita - contable y empresarial

---

```
commit 9291d35cb7d5de01840fe35870ab4e9fd1f304f8 (HEAD -> develop, origin/feat  
Author: Bryan Gerardo Ascuc de Paz <bascuc@nabenik.com>  
Date: Mon Jun 1 17:11:36 2020 -0600  
  
ERP-71 #comment Adding migrations for Provider by organization #time 1h
```

# Retos del mundo real

---

## Mi "mundo real"

- Venta/Geocerca (5 WAR) Payara Application Server
- ERP - 10 modulos (1 EAR, 9 EJB, 1 WAR), JBoss/Wildfly
- POS - JavaFX y Windows

## El rompe cabezas

- Módulos en Java 9
- sun.misc.unsafe
- Corba y Java EE
- JavaFX
- IDE
- Licenciamiento

## De Java 8 a Java 11

---



# Algoritmo de actualización

---

## Estrategia @tuxtor

1. Verificar y probar la compatibilidad del runtime/servidor/framework
2. Múltiples JVM en desarrollo con cambio fácil
3. Actualizar el compilador de Maven
4. Actualizar las bibliotecas
5. Incluir los módulos EE en los war/jar
6. Actualizar el IDE
7. Preparar el proyecto para módulos en el caso de JavaFX
8. Determinar previamente el Java que necesito
9. Ejecutar distintas versiones de Java en producción

## Compatibilidad con Java 11

- Tomcat
- Spring
- Micronaut
- Vert.x
- Jakarta EE (JBoss/Wildfly, OpenLiberty, Payara, WebLogic)





## Generación dinámica de Bytecode

- ByteBuddy
- ASM
- glib
- Spring
- Java EE
- Hibernate
- Mockito

InfoQ Homepage > Articles > A Post-Apocalyptic Sun.misc.Unsafe World

JAVA

[Virtual Event] Discover #

### A Post-Apocalyptic sun.misc.Unsafe World

LIKE

7

PRINT

BOOKMARK

AUG 30, 2015 • 15 MIN READ

by

Christoph  
Engelbert

FOLLOW

Java as a language and the JVM as a platform just celebrated its 20th birthday. With its noble origins on set-top-boxes, mobiles and java-cards, as well as all kinds of server systems, Java is emerging as the lingua franca of the Internet of Things. Quite obviously Java is everywhere!

Less obvious is that Java is also heavily immersed in all sorts of low-latency applications such as game servers and high frequency trading applications. This was only made possible thanks to a propitious deficiency in the Java visibility rules for classes and packages, offering access to a controversial little class called *sun.misc.Unsafe*. This class was and still is a divider; some love it, others hate it with a passion - the essential part is, it helped the JVM and the Java ecosystem to evolve to where it is today. The Unsafe class basically compromised on some of Java's hallmark strict safety standards in favor of speed.

- Maven 3.5.0
- Compiler 3.8.0
- surefire 2.22.0
- failsafe 2.22.0
- release version 11.0

## JEP 320: Remove the Java EE and CORBA Modules

<i>Owner</i>	Lance Andersen
<i>Type</i>	Feature
<i>Scope</i>	SE
<i>Status</i>	Closed / Delivered
<i>Release</i>	11
<i>Component</i>	other-libs
<i>Discussion</i>	jdk dash dev at openjdk dot java dot net
<i>Effort</i>	M
<i>Reviewed by</i>	Alan Bateman, Alex Buckley, Brian Goetz, Mark Reinhold
<i>Created</i>	2017/10/11 18:36
<i>Updated</i>	2019/05/23 17:37
<i>Issue</i>	<a href="#">8189188</a>

### Summary

Remove the Java EE and CORBA modules from the Java SE Platform and the JDK. These modules were [deprecated in Java SE 9](#) with the declared intent to remove them in a future release.

## Jakarta EE 8 Full Platform Compatible Products



Apusic AAS

Kingdee Apusic cloud computing  
Co., Ltd.

Version: 10.1

[Download](#)

[Proof of compatibility](#)



Eclipse Glassfish

Eclipse Foundation

Version: 5.1.0, Full Profile

[Download](#)

[Proof of compatibility](#)



JBoss Enterprise  
Application Platform

Red Hat

Version: 7.3.0.GA

[Download](#)

[Proof of compatibility](#)



JEUS

TmaxSoft Co., Ltd

Version: 8.5

[Download](#)

[Proof of compatibility](#)



Open Liberty

IBM Corporation



Payara Server



Primeton AppServer

Primeton



WildFly

Red Hat



18

## JAF (javax.activation)

```
1 <dependency>
2   <groupId>jakarta.activation</groupId>
3   <artifactId>jakarta.activation-api</artifactId>
4   <version>1.2.2</version>
5 </dependency>
```

CORBA = RIP

## JAXB (java.xml.bind)

```
1  <!-- API -->
2  <dependency>
3      <groupId>jakarta.xml.bind</groupId>
4      <artifactId>jakarta.xml.bind-api</artifactId>
5      <version>2.3.2</version>
6  </dependency>
7
8  <!-- Runtime -->
9  <dependency>
10     <groupId>org.glassfish.jaxb</groupId>
11     <artifactId>jaxb-runtime</artifactId>
12     <version>2.3.2</version>
13 </dependency>
```

## JAX-WS (java.xml.ws)

```
1  <!-- API -->
2  <dependency>
3      <groupId>jakarta.xml.ws</groupId>
4      <artifactId>jakarta.xml.ws-api</artifactId>
5      <version>2.3.2</version>
6  </dependency>
7
8  <!-- Runtime -->
9  <dependency>
10     <groupId>com.sun.xml.ws</groupId>
11     <artifactId>jaxws-rt</artifactId>
12     <version>2.3.2</version>
13 </dependency>
```

## Common Annotations (java.xml.ws.annotation)

```
1 <dependency>
2   <groupId>jakarta.annotation</groupId>
3   <artifactId>jakarta.annotation-api</artifactId>
4   <version>1.3.5</version>
5 </dependency>
```



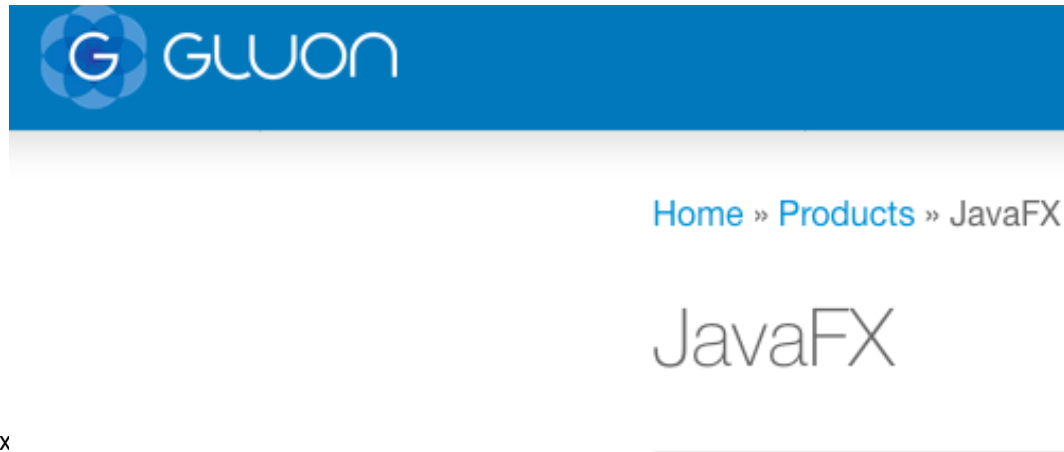
## IDEs compatibles con Java 11

- Eclipse
- NetBeans
- IntelliJ IDEA
- Visual Studio Code

## Algunos plug-ins problemáticos

1. Glassfish
2. WebLogic
3. Icefaces

JavaFX es un módulo independiente del JDK a partir de Java 11, compatible con JPMS, casi todos usan la compilación de Gluon



# ¿Cual Java necesito?

---

## Obligatorios por contrato

- Software comercial de Oracle (HotSpot)
- Software comercial de SAP (SAP VM)
- Software comercial de Red Hat (OpenJDK + RHEL)
- Software comercial de IBM (J9)

## Algunos otros "Javas"

- AdoptOpenJDK (soporte de IBM en J9)
- Correto
- Azul Zulu
- Java en Linux

## Linux

- Docker
- RHEL
- Oracle Linux
- Debian
- Gentoo

## Windows

- Docker / Containerd
- Variables de entorno en proyecto/runtime
- Lo importante es la salud

## De mi data center a la nube - moderna

-

---



# Desde mi data center

SaaS



CX



HCM



ERP



SCM



EPM



Analytics



Social



Data

PaaS



Database



Java



EM



Node.js



Big Data



Big Data  
Discovery



Business  
Intelligence



Mobile



Integration



Developer



Documents



Messaging



Process



Database  
Backup

IaaS



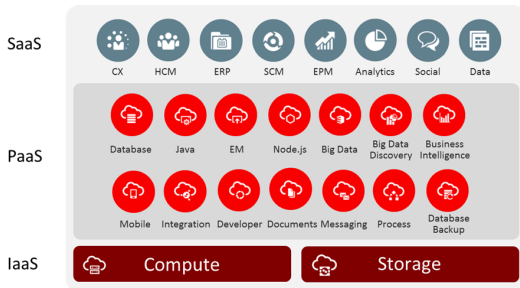
Compute




Storage

## PaaS

- Clasico: Desplegar War en servidores autonomos
- Primer abordaje: Desplegar contenedores de forma manual
- Abordaje maduro: Desplegar contenedores y orquestar con Rancher/Docker Swarm/Kubernetes/Mesos



# Desde mi data center

<a href="#">Create Instance</a>								
Name	State	Public IP	Shape	OCPU Count	Memory (GB)	Availability Domain	Fault Domain	Created
<a href="#">instance-demo-orainux</a>	 Running	150.136.235.64	VM.Standard2.1	1	15	AD-1	FD-1	Tue, Aug 18, 2020 13:53:11 UTC
Showing 1 Item < Page 1 >								



Desde Java 14 hasta el infinito

---



# ¿Que recibo con cada versión nueva de Java?

---

- Java - **Lenguaje**
- Java - Bibliotecas e APIs
- Java - Maquina Virtual de Java

# Java - Las mejoras que resaltan

---

- Java 9
  - Modulos
  - JShell
  - HTTP/2
  - Factory methods
- Java 10
  - Type Inference
  - Class Data Sharing
  - Time based release
- Java 11
  - String methods
  - File methods
  - Direct .java execution
- Java 12
  - Switch expressions
- Java 13
  - Text blocks
- Java 14
  - Pattern matching
  - Records
  - Helpfull NPE

```
1 HttpRequest request = HttpRequest.newBuilder()
2   .uri(new URI("https://swapi.co/api/starships/9"))
3   .GET()
4   .build();
5
6 HttpResponse<String> response = HttpClient.newHttpClient()
7   .send(request, BodyHandlers.ofString());
8
9 System.out.println(response.body());
```

# JEP 286: Local-Variable Type Inference

---

```
1 public static void main(String args[]){  
2     var localValue = 99;  
3     System.out.println(++localValue);  
4     //localValue = "Foo"  
5 }
```

# JEP 330: Launch Single-File Source-Code Programs

```
2. tuxtor@millenium-falcon-2: ~/Sandbox/JavaTrain/fileexecution (zsh)
→ fileexecution echo "public class HelloWorld{
    public static void main(String args[]){
        System.out.println(\"Hello world\");
    }
}" > HelloWorld.java
→ fileexecution java HelloWorld.java
Hello world
→ fileexecution ls
HelloWorld.java
→ fileexecution
```

## 325: Switch Expressions (Preview)

---

Ahora

```
1 String langType = switch (args[0]) {  
2     case "Java", "Scala", "Kotlin" -> "Static typed";  
3     case "Groovy", "JavaScript" -> "Dynamic typed";  
4     default -> {  
5         System.out.println("This meant to be a processing  
6             block");  
7         yield "Probably LISP :)";  
8     }  
9 };  
10 System.out.println(langType);
```

## 355: Text Blocks (Preview)

---

### Antes

```
1 String html = "<html>\n" +  
2 "    <body>\n" +  
3 "        <p>Hello, world</p>\n" +  
4 "    </body>\n" +  
5 "</html>\n";
```

### Ahora

```
1 String html = """  
2 <html>  
3     <body>  
4         <p>Hello, world</p>  
5     </body>  
6 </html>  
7 """;
```



## Data carrier

```
1 record Person(String name, String email, int age) {}
```

## Use

```
1 Person foo = new Person("Marco", "example@mail.com", 99);  
2 System.out.println(foo);  
3 //foo.name = "Polo";
```



**Oracle  
Groundbreakers**



**ORACLE®**  
Certified Professional  
Java SE 8 Programmer

**ORACLE®**  
Certified Associate  
Java SE 8 Programmer

- vorozco@nabenik.com
- @tuxtor
- <http://voroazco.com>
- <http://tuxtor.shekalug.org>



This work is licensed under  
Creative Commons Attribution-  
NonCommercial-ShareAlike 3.0  
Guatemala (CC BY-NC-SA 3.0 GT).