



Tchê Linux Porto Alegre 2019

# Don't Repeat Yourself



Abstraíndo funcionalidades com  
Jenkins Shared Libraries

# O que é Jenkins?

The screenshot shows the Jenkins web interface in a browser. The top navigation bar includes the Jenkins logo, a search bar, and user information (Administrator | log out). Below the navigation bar, there's a sidebar with various links: New Item (highlighted with a green box), People, Build History, Manage Jenkins, My Views, Credentials, and New View. The main content area displays a table of builds. The table has columns for Status (S), Weather (W), Name, Last Success, Last Failure, and Last Duration. Two builds are listed: 'JavaTpoint Maven' and 'JavaTpoint Pipeline'. Below the table, there are links for 'Icon: S M L', 'Legend', and three RSS feeds: 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'. On the left side, there are two panels: 'Build Queue' showing 'No builds in the queue.' and 'Build Executor Status' showing '1 Idle' and '2 Idle'.

Dashboard [Jenkins] x +

localhost:8080/jenkins/

Jenkins

Administrator | log out

ENABLE AUTO REFRESH

[add description](#)

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">JavaTpoint Maven</a>	N/A	N/A	N/A
		<a href="#">JavaTpoint Pipeline</a>	23 hr - <a href="#">#5</a>	N/A	1.7 sec

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

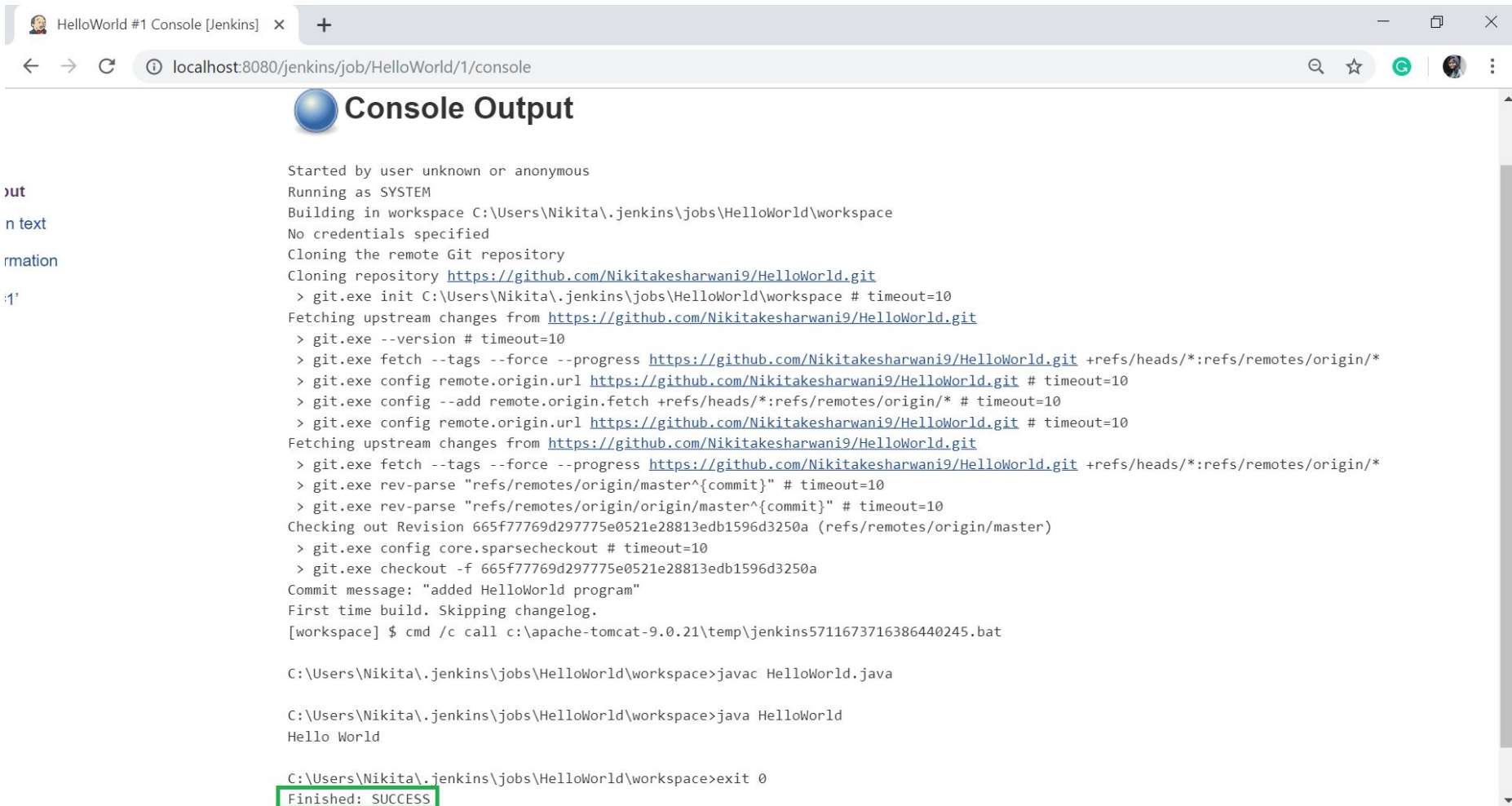
**Build Queue** —

No builds in the queue.

**Build Executor Status** —

1 Idle  
2 Idle

# O que é Jenkins?



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/jenkins/job/HelloWorld/1/console'. The page title is 'HelloWorld #1 Console [Jenkins]'. The main content area is titled 'Console Output' and displays the following text:

```
Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\Users\Nikita\.jenkins\jobs\HelloWorld\workspace
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Nikitakesharwani9/HelloWorld.git
> git.exe init C:\Users\Nikita\.jenkins\jobs\HelloWorld\workspace # timeout=10
Fetching upstream changes from https://github.com/Nikitakesharwani9/HelloWorld.git
> git.exe --version # timeout=10
> git.exe fetch --tags --force --progress https://github.com/Nikitakesharwani9/HelloWorld.git +refs/heads/*:refs/remotes/origin/*
> git.exe config remote.origin.url https://github.com/Nikitakesharwani9/HelloWorld.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/Nikitakesharwani9/HelloWorld.git # timeout=10
Fetching upstream changes from https://github.com/Nikitakesharwani9/HelloWorld.git
> git.exe fetch --tags --force --progress https://github.com/Nikitakesharwani9/HelloWorld.git +refs/heads/*:refs/remotes/origin/*
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
> git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision 665f77769d297775e0521e28813edb1596d3250a (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 665f77769d297775e0521e28813edb1596d3250a
Commit message: "added HelloWorld program"
First time build. Skipping changelog.
[workspace] $ cmd /c call c:\apache-tomcat-9.0.21\temp\jenkins5711673716386440245.bat


C:\Users\Nikita\.jenkins\jobs\HelloWorld\workspace>javac HelloWorld.java

C:\Users\Nikita\.jenkins\jobs\HelloWorld\workspace>java HelloWorld
Hello World


C:\Users\Nikita\.jenkins\jobs\HelloWorld\workspace>exit 0
Finished: SUCCESS
```

# Automação


- Continuous Integration
- Testes
- Análise de Código
- Continuous Delivery

 **Build #526 (Mar 26, 2012 4:02:04 PM)** [Keep this build forever](#) [Delete this build](#)

[add description](#) Started 18 hr ago  
Took 6 min 21 sec


 Changes

1. [MEDTZIN-1105](#) bug-fix Se corrige ordenamiento de las notas del Expediente Completo.  
([commit: 16d8075fba05ee5d071d90824c7b8b2b73bed486](#)) ([detail](#))

 Started by an SCM change

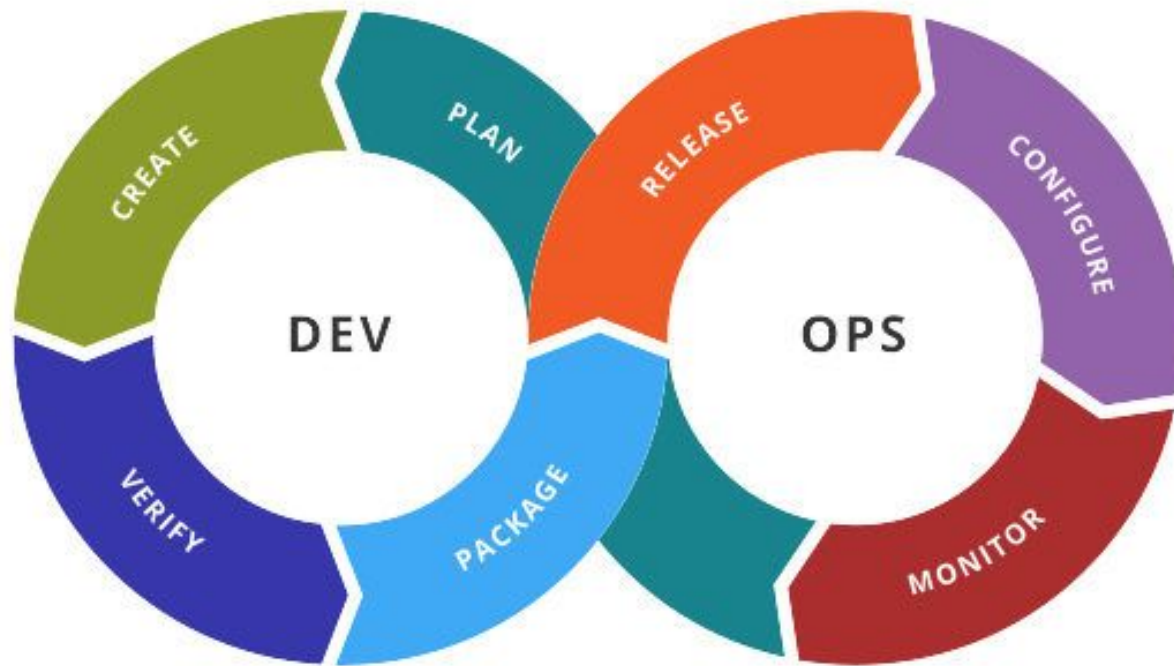
 Revision: 16d8075fba05ee5d071d90824c7b8b2b73bed486

- origin/HEAD
- origin/master

 There is no Esc key on Chuck Norris' keyboard, because no one escapes Chuck Norris.

 The build was worth [1 points](#)





Over  
1000 Jenkins  
Plugins

Integration  
with over 100  
DevOps Tools

Orchestration  
of the DevOps  
Toolchain

End-to-End  
CD Pipeline  
Management



Code & Commit

Build & Config


Scan & Test

Release

Deploy



# Início do Jenkins

☐ Prevent Cross Site Request Forgery exploits 

☒ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project. 

## Global properties

☐ Environment variables


## SCM Sync configuration


SCM

☐ None 

☐ Subversion

☒ Git

Repository URL  

Never bother me with commit messages ☐ 

Display SCM Sync Status

☒ 

Commit message pattern



Manual synchronization includes



Reload config from SCM

[Reload](#) 

## JDK

JDK installations

List of JDK installations on this system

## Ant

Ant installations

List of Ant installations on this system

## Maven

Maven installations

 Maven



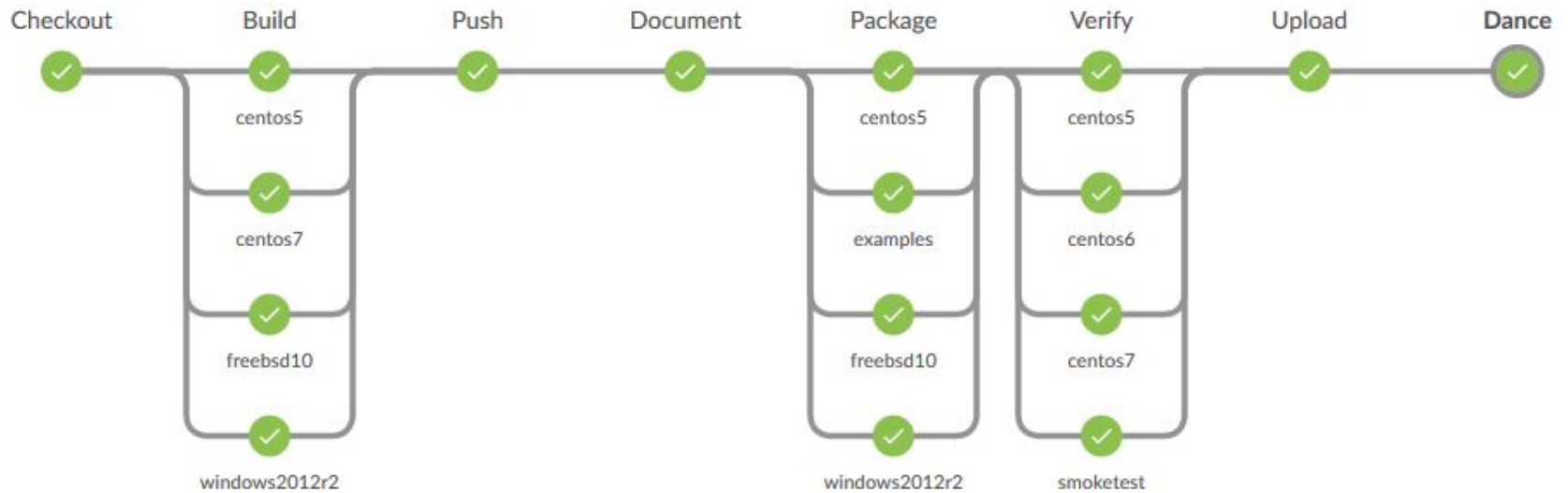
# Jenkins 2.0

[\(just show failures\)](#) [enlarge](#)

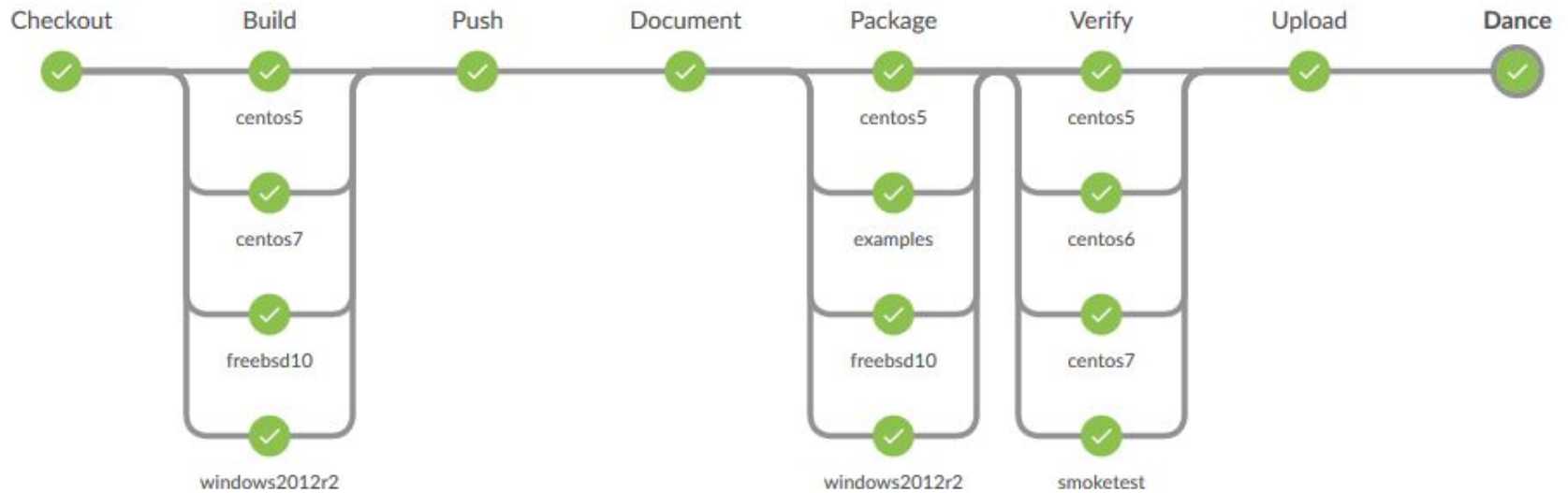
Average stage times: (Average <u>full</u> run time: ~1min 12s)		Clone the project	Compilation and Analysis	Checkstyle	Tests and Deployment	Runing unit tests	Runing integration tests	Staging
		4s	169ms	23s	69ms	20s	41s	42s
#37	Nov 30 06:47 No Changes	5s	216ms	23s	69ms	21s	39s	40s
#36	Nov 30 06:20 No Changes	4s	122ms	23s	70ms	20s	44s	44s



# Jenkins 2.0



# Jenkins 2.0



# Pipeline as Code

```
1 node{
2
3   git branch: "master", url: "https://github.com/robsonbittencourt/jenkins-pipeline-example"
4
5   stage('Unit Tests') {
6     withEnv(["JAVA_HOME=${ tool "java-8" }", "PATH+MAVEN=${ tool "maven" }/bin:${env.JAVA_HOME}/bin"]) {
7       sh "mvn test"
8     }
9   }
10
11   stage ('Sonarqube') {
12     withEnv(["JAVA_HOME=${ tool "java-8" }", "PATH+MAVEN=${ tool "maven" }/bin:${env.JAVA_HOME}/bin"]) {
13       sh "mvn sonar:sonar -Dsonar.host.url=http://sonar:9000"
14     }
15   }
16
17   stage ('Build Jar') {
18     withEnv(["JAVA_HOME=${ tool "java-8" }", "PATH+MAVEN=${ tool "maven" }/bin:${env.JAVA_HOME}/bin"]) {
19       sh "mvn clean package -Dmaven.test.skip=true"
20     }
21   }
22 }
23 }
```



# Pipeline as Code

```
1 node{
2
3   git branch: "master", url: "https://github.com/robsonbittencourt/jenkins-pipeline-example"
4
5   stage('Unit Tests') {
6     withEnv(["JAVA_HOME=${ tool "java-8" }", "PATH+MAVEN=${ tool "maven" }/bin:${env.JAVA_HOME}/bin"]) {
7       sh "mvn test"
8     }
9   }
10
11   stage ('Sonarqube') {
12     withEnv(["JAVA_HOME=${ tool "java-8" }", "PATH+MAVEN=${ tool "maven" }/bin:${env.JAVA_HOME}/bin"]) {
13       sh "mvn sonar:sonar -Dsonar.host.url=http://sonar:9000"
14     }
15   }
16
17   stage ('Build Jar') {
18     withEnv(["JAVA_HOME=${ tool "java-8" }", "PATH+MAVEN=${ tool "maven" }/bin:${env.JAVA_HOME}/bin"]) {
19       sh "mvn clean package -Dmaven.test.skip=true"
20     }
21   }
22 }
23
24
```

Don't  
Repeat  
Yourself



# Shared Libraries

- Bibliotecas que disponibilizam funções para serem usadas em Pipelines
- Escritas em Groovy
- Podem ser compartilhadas entre diferentes projetos



# Demo



<https://github.com/umovmecc/umovme-shared-libraries>

# Obrigado



@rluizv



robsonbittencourt



Tchê Linux Porto Alegre 2019

# Don't Repeat Yourself



Abstraíndo funcionalidades com  
Jenkins Shared Libraries