

Ingeniería de Software I

Integración Continua

Jocelyn Simmonds

Departamento de Ciencias de la Computación

Son las 4am ...

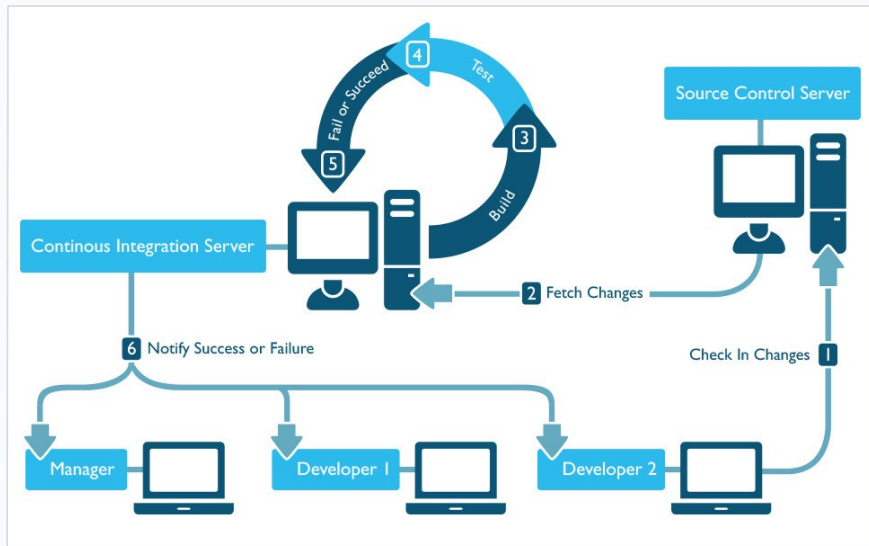
¿Cuáles son las palabras más temidas durante un paso a producción?

Son las 4am ...

¿Cuáles son las palabras más temidas durante un paso a producción?

... ¡pero si funciona en mi máquina!

Continuous Integration and Delivery (CI/CD)



Objetivo de CI/CD

El objetivo principal de CI/CD es **siempre** tener un producto que funcione y este en ambiente de producción.

Es posible que algunas de las funcionalidades no funcionen por completo aun, pero no deben romper el producto.

Grady Booch definió el termino CI en 1991, pero no anticipo la frecuencia/escala actual.

Principios de CI/CD

- Usar versionamiento de código
- Automatizar el proceso de compilación (el **build**)
- El build debe ser self-testing
- Todos deben hacer commits sobre el baseline cada día
- Cada commit debe ser compilado
- Testear el build en un ambiente de testing (clon de producción)
- Dar visibilidad al resultado del ultimo build (success/fail)
- Automatizar el proceso de deployment

¿Por qué aplicar CI/CD?

- Para reducir el trabajo manual y repetitivo
- Para recibir información temprana acerca de problemas
 - ¿Olvidaste versionar un archivo?
 - ¿Diferencias en la plataforma local vs. producción?
 - ¿Dejo de funcionar algo que ya funcionaba?
- Para ayudar evitar el caos antes de una fecha de entrega
- Siempre se tiene información actualizada acerca del estado del producto

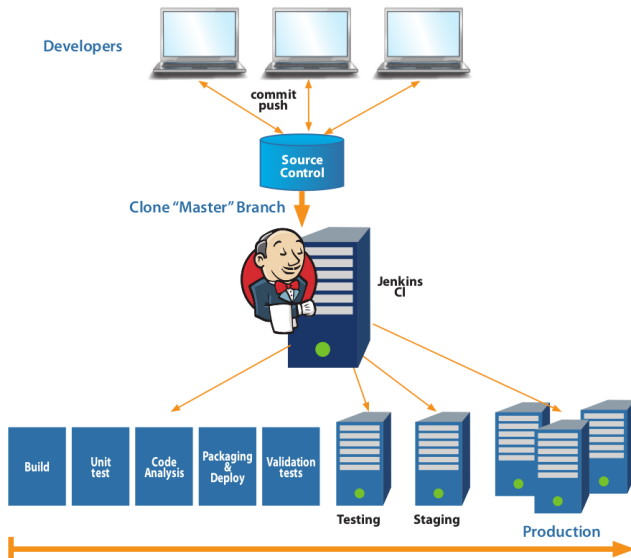
Hoy hay muchas herramientas/suites

Algunas ofrecen cuentas gratuitas limitadas:

- Hudson/Jenkins (open source)
- Teamcity
- Travis CI
- Circle CI
- Bamboo
- Codeship

Ojo que CI/CD implica también un cambio de “cultura”, es un hábito que debe desarrollar el equipo.

Construcción y paso a producción con Jenkins



Demo de Jenkins

Pasos previos

Para ejecutar esta demo, deben tener instalado

- Java:
`https://java.com/en/download/help/download_options.xml`
- Jenkins: `https://jenkins.io/download/`
 - yo elegí la opción de set inicial de plugins estándares
 - después veremos como activar más plugins
- Maven: `https://maven.apache.org/install.html`
- Git: `https://atlassian.com/git/tutorials/install-git`

También necesitamos un repositorio

The screenshot shows a web browser window with the address bar displaying 'https://github.com/jsimmond/tda_test'. The page title is 'jsimmond / tda_test'. The repository has 1 star, 0 forks, and 0 pull requests. The 'Code' tab is selected, showing a table of commits. The table has three columns: file name, commit message, and time ago. The files listed are '.gitignore', 'DLinkedList.java', and 'Node.java', all committed '10 hours ago'. The commit messages are 'Initial commit', 'First version', and 'First version' respectively. There is a button 'Add a README' at the bottom right of the commit table.

jsimmond / tda_test

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. [Add topics](#) [Edit](#)

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

jsimmond committed on GitHub First version Latest commit 279eb4c 10 hours ago

.gitignore	Initial commit	10 hours ago
DLinkedList.java	First version	10 hours ago
Node.java	First version	10 hours ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

© 2017 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

El Dashboard: <http://localhost:8080/>

The screenshot shows the Jenkins Dashboard in a Mozilla Firefox browser. The address bar shows `localhost:8080`. The dashboard has a left sidebar with navigation links: New Item, People, Build History, Manage Jenkins, My Views, and Credentials. The main content area displays a table of builds, which is highlighted with a blue box. A blue arrow points from the text 'Proyectos' to this table. Another blue arrow points from the text 'Nodes' to the 'Build Executor Status' section, which shows a 'master' node with 2 idle executors and a 'slave_fly' node that is offline. The footer of the dashboard indicates the page was generated on Sep 28, 2017, at 8:46:17 AM CLST, using REST API, and is Jenkins ver. 2.73.1.

Dashboard [Jenkins] - Mozilla Firefox

jsimmond/tda_test x Dashboard [Jenkins] x +

localhost:8080

Search

Jenkins

admin | log out

ENABLE AUTO REFRESH

add description

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Addressbook-checkWarnings	7 hr 35 min - #3	N/A	5.9 sec
		Addressbook-compile	8 hr 6 min - #2	N/A	3.5 sec
		Addressbook-unit_tests	7 hr 35 min - #6	8 hr 28 min - #3	4.7 sec
		TDAs	7 hr 34 min - #5	9 hr 37 min - #3	19 sec

Legend RSS for all RSS for failures RSS for just latest builds

Build Queue

No builds in the queue.

Build Executor Status

master

- 1 Idle
- 2 Idle

slave_fly (offline)

Nodes

Proyectos

Page generated: Sep 28, 2017 8:46:17 AM CLST [REST API](#) Jenkins ver. 2.73.1

Valores por defecto

Los valores por defecto se encuentran en un archivo

- Linux: `/etc/default/jenkins`
- Datos importantes:
 - ubicación de Java
 - usuario, ubicación, etc. de Jenkins
 - puerto de conexión
 - ubicación del log de Jenkins
 - ...
- Si no crean un usuario admin
 - usuario por defecto: `admin`
 - password: listada en el archivo
`/var/lib/jenkins/secrets/initialAdminPassword`

Los plugins de la instalación estándar

Getting Started

Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ build timeout plugin	✓ Credentials Binding Plugin	Credentials Binding Plugin
✓ Timestampar	✓ Workspace Cleanup Plugin	✓ Ant Plugin	✓ Gradle Plugin	Timestampar
⚙ Pipeline	⚙ GitHub Branch Source Plugin	⚙ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View Plugin	** Durable Task Plugin
⚙ Git plugin	⚙ Subversion Plug-in	⚙ SSH Slaves plugin	✓ Matrix Authorization Strategy Plugin	** Pipeline: Nodes and Processes
✓ PAM Authentication plugin	✓ LDAP Plugin	⚙ Email Extension Plugin	✓ Mailer Plugin	** Resource Disposer Plugin
				Jenkins Workspace Cleanup Plugin
				Ant Plugin
				Gradle Plugin
				** Pipeline: Milestone Step
				** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) plugin
				** Jackson 2 API Plugin
				** JavaScript GUI Lib: ACE Editor bundle plugin
				** Pipeline: SCM Step
				** Pipeline: Groovy
				** Pipeline: Input Step
				** Pipeline: Stage Step
				** Pipeline Graph Analysis Plugin
				** Pipeline: REST API Plugin
				** JavaScript GUI Lib: Handlebars bundle plugin
				** JavaScript GUI Lib: Moment.js bundle plugin
				Pipeline: Stage View Plugin

Manage Jenkins → Manage Plugins

The screenshot shows the Jenkins Update Center interface in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8080/pluginManager/installed`. The Jenkins logo and navigation links (Jenkins, Plugin Manager) are at the top. On the left sidebar, there are links for 'Back to Dashboard', 'Manage Jenkins', and 'Update Center'. The main content area has tabs for 'Updates', 'Available', 'Installed', and 'Advanced', with 'Installed' being the active tab. Below the tabs is a table of installed plugins. The table has columns for 'Enabled', 'Name', 'Version', 'Previously installed version', and 'Uninstall'. The following table represents the data shown in the screenshot:

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Ant Plugin Adds Apache Ant support to Jenkins	1.7		Uninstall
<input checked="" type="checkbox"/>	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.	4.5.3-2.0		Uninstall
<input checked="" type="checkbox"/>	Authentication Tokens API Plugin This plugin provides an API for converting credentials into authentication tokens in Jenkins.	1.3		Uninstall
<input checked="" type="checkbox"/>	bouncycastle API Plugin This plugin provides a stable API to Bouncy Castle related tasks.	2.16.2		Uninstall
<input checked="" type="checkbox"/>	Branch API Plugin This plugin provides an API for multiple branch based projects.	2.0.11		Uninstall
	Build Pipeline Plugin This plugin renders upstream and downstream connected jobs that			

Algunos plugins útiles

Plugins para integrar con:

- Maven: simplifica procesos de compilación, define fases típicas como `validate`, `compile`, `test`
- Build Pipeline: permite juntar varios proyectos en un pipeline
- PMD: analiza en forma estática el código, arrojando warnings
- Cobertura: genera reportes de cobertura de código de tests

Hay miles de plugins, para una variedad de tecnologías y tareas de compilación. Ver <https://plugins.jenkins.io/> para más info.

Dashboard → New Item

New Item [Jenkins] - Mozilla Firefox

jsimmond/tda_test x New Item [Jenkins] x +

localhost:8080/view/all/newJob 110% Search


Jenkins search admin | log out

Jenkins > All >


Enter an item name

ejemplo


** Required field*

**Freestyle project**


This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**


Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**

Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**External Job**

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK Folder

TDAs: un proyecto “freestyle”

Jenkins | search | admin | log out

Jenkins > TDAs > [ENABLE AUTO REFRESH](#)

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

Project TDAs

[add description](#)

[Disable Project](#)

[Workspace](#)

[Recent Changes](#)

Permalinks

- [Last build \(#5\), 11 hr ago](#)
- [Last stable build \(#5\), 11 hr ago](#)
- [Last successful build \(#5\), 11 hr ago](#)
- [Last failed build \(#3\), 13 hr ago](#)
- [Last unsuccessful build \(#3\), 13 hr ago](#)
- [Last completed build \(#5\), 11 hr ago](#)

Build History

[trend](#)

#	Time
#5	Sep 28, 2017 1:11 AM
#4	Sep 27, 2017 11:13 PM
#3	Sep 27, 2017 11:08 PM
#2	Sep 27, 2017 11:06 PM
#1	Sep 27, 2017 11:05 PM

TDA: Configure

The screenshot shows the Jenkins web interface in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8080/job/TDAs/configure`. The page title is "TDAs Config [Jenkins]". The Jenkins logo and navigation links are visible at the top. The configuration page is for a job named "TDAs". The "General" tab is selected, showing fields for "Project name" (TDAs) and "Description". Below these are checkboxes for various options: "Discard old builds", "GitHub project", "This project is parameterized", "Throttle builds", "Disable this project", "Execute concurrent builds if necessary", and "Restrict where this project can be run" (which is checked). A "Label Expression" field contains the text "slave_fly". At the bottom, there are "Save" and "Apply" buttons, and a status message indicating the label is serviced by 1 node. The browser's developer tools are open at the bottom, showing the "Label slave_fly is serviced by 1 node" message.

TDAs Config [Jenkins] - Mozilla Firefox

jsimmond/tda_test x TDAs Config [Jenkins] x

localhost:8080/job/TDAs/configure 110% Search

Jenkins search admin | log out

Jenkins > TDAs >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Project name TDAs

Description

[Plain text] [Preview](#)

☐ Discard old builds

☐ GitHub project

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

☒ Restrict where this project can be run

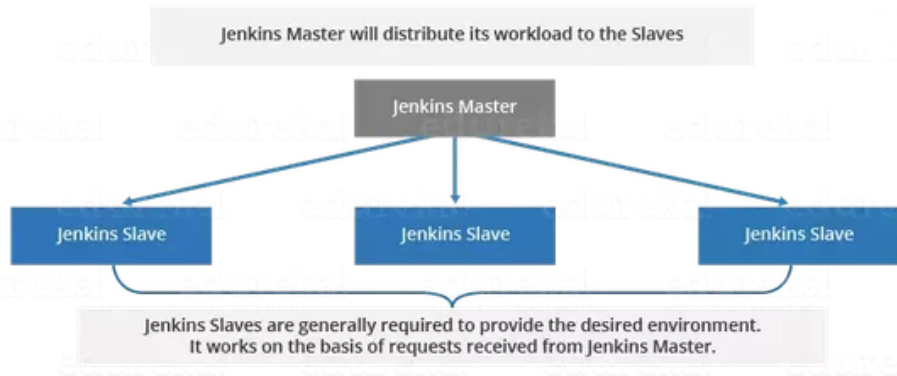
Label Expression slave_fly

Save Apply

Label slave_fly is serviced by 1 node

Advanced...

Jenkins tiene una arquitectura distribuida



Los nodos esclavos pueden tener una configuración distinta al maestro.
P.ej: master = Ubuntu, slave₁ = Mac, slave₂ = Windows, etc.

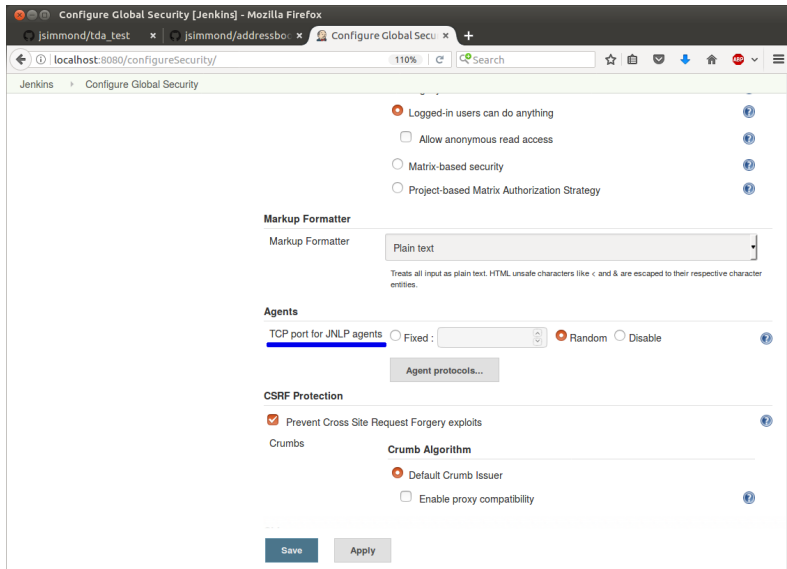
Formas de crear un nodo esclavo

Manage Jenkins → Manage Nodes → New Node → Permanent Agent

- Usando Java Web Start:
 - la manera más simple de lanzar un nodo
 - el esclavo solo debe tener instalado Java
 - el maestro debe tener habilitado el Java Network Launch Protocol (JNLP)
- Ejecutando algún comando en el maestro
- Por SSH
- Creando un servicio Windows (solo esclavos Windows)

También deben indicar el Remote root directory

Habilitar JNLP en el maestro



Esclavo configurado con Java Web Start

slave_fly Configuration [Jenkins] - Mozilla Firefox

localhost:8080/computer/slave_fly/configure

Jenkins

admin | log out

Jenkins > Nodes > slave_fly

- Back to List
- Status
- Delete Agent
- Configure**
- Build History
- Load Statistics
- Log

[Build Executor Status](#)

Parameters to configure

Name: slave_fly

Description:

of executors: 1

Remote root directory: /Users/fly

Labels:

Usage: Use this node as much as possible

Launch method: Launch agent via Java Web Start

Disable WorkDir: ☐

Custom WorkDir path:

Internal data directory: remoting

Fail if workspace is missing: ☐

[Advanced...](#)

[Save](#)

Abrir Jenkins en el esclavo y lanzar agente

The screenshot shows the Jenkins web interface in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8080/computer/slave_fly/`. The page title is "slave_fly [Jenkins]". The Jenkins logo and navigation links (admin, log out) are at the top. The breadcrumb trail shows "Jenkins > Nodes > slave_fly".

On the left sidebar, there are links: "Back to List", "Status", "Delete Agent", "Configure", "Build History", "Load Statistics", and "Log". A button "Build Executor Status" is also present.

The main content area is titled "Agent slave_fly" with a red error icon. A message states: "Ping response time is too long or timed out." Below this, it says "Connect agent to Jenkins one of these ways:" and lists two options:

- Launch: Launch agent from browser
- Run from agent command line:

The command line instructions are:

```
java -jar slave.jar -jnlpUrl http://localhost:8080/computer/slave_fly/slave-agent.jnlp
-secret ae2d6a1ea7f720c9724ff612990bfdde59f50d322344556e5f5227ec497f487c -workDir
"/Users/fly"
```

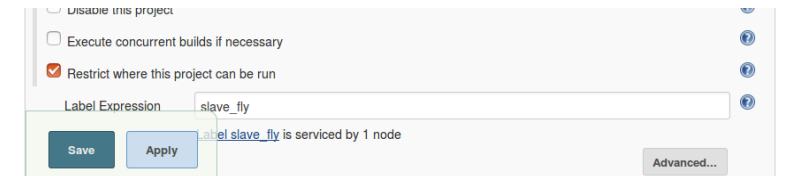
Below the command line, it says "Projects tied to slave_fly" and "None".

At the bottom right, there is a button "Mark this node temporarily offline".

Page generated: Sep 28, 2017 4:11:55 PM CLST [REST API](#) [Jenkins ver. 2.73.1](#)

TDAs: Configure

Ahora pueden indicar el nodo donde debe ejecutarse un proyecto:



The screenshot shows the Jenkins configuration interface for a project. It includes several checkboxes for project settings:

- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☒ Restrict where this project can be run

Below the checkboxes, there is a text input field labeled "Label Expression" containing the value "slave_fly".

At the bottom left, there are two buttons: "Save" (dark blue) and "Apply" (light blue). A tooltip is visible over the "Apply" button, displaying the text: "Label 'slave_fly' is serviced by 1 node".

At the bottom right, there is a button labeled "Advanced...".

TDAs: repositorio

The screenshot shows the Jenkins web interface in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8080/job/TDAs/configure`. The page title is "TDAs Config [Jenkins]". The "Source Code Management" tab is selected, showing the following configuration:

- Source Code Management:** The "Git" radio button is selected.
- Repositories:** The "Repository URL" is `https://github.com/jsimmond/tda_test.git`. The "Credentials" dropdown is set to "- none -" with an "Add" button next to it. There are "Advanced..." and "Add Repository" buttons.
- Branches to build:** The "Branch Specifier (blank for 'any')" is `*/master`. There is an "Add Branch" button.
- Repository browser:** The dropdown is set to "(Auto)".
- Additional Behaviours:** There is an "Add" button.
- Buttons:** "Save" and "Apply" buttons are at the bottom left.

Clonando el repositorio

```
jsimmond@pita:~/Dropbox/postitulos/testing/demo$ ls
total 8.0K
drwxr-xr-x 3 jsimmond users 4.0K Dec  8 2016 addressbook
jsimmond@pita:~/Dropbox/postitulos/testing/demo$ git clone https://github.com/jsimmond/tda_test.git
Cloning into 'tda_test'...
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
Checking connectivity... done.
jsimmond@pita:~/Dropbox/postitulos/testing/demo$ ls
total 16K
drwxr-xr-x 3 jsimmond users 4.0K Dec  8 2016 addressbook
drwxr-xr-x 3 jsimmond users 4.0K Sep 28 12:40 tda_test
jsimmond@pita:~/Dropbox/postitulos/testing/demo$
```

TDAs: Build y Post-build Actions

The screenshot shows the Jenkins 'TDAs Config' page in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8080/job/TDAs/configure`. The page has a breadcrumb trail: `Jenkins > TDAs >`. Below the breadcrumb, there are tabs for configuration sections: `General`, `Source Code Management`, `Build Triggers`, `Build Environment`, `Build` (which is selected), and `Post-build Actions`.

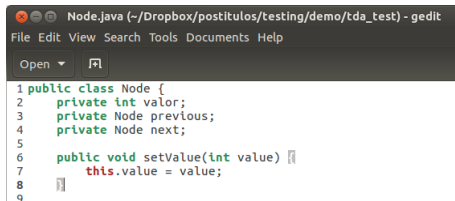
Build Section:

- There is a section titled **Execute shell** with a red 'X' icon and a help icon.
- The **Command** field contains the text `javac DLinkedList.java`.
- Below the command field is a link: [See the list of available environment variables](#).
- An **Advanced...** button is located at the bottom right of the Execute shell section.
- At the bottom left of the Build section is an **Add build step** button.

Post-build Actions Section:

- There is a section titled **E-mail Notification** with a red 'X' icon and a help icon.
- The **Recipients** field contains the text `jsimmond@gmail.com`.
- Below the recipients field is a description: "Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable."
- There are two checkboxes:
 - ☒ **Send e-mail for every unstable build**
 - ☐ **Send separate e-mails to individuals who broke the build**
- At the bottom left of the Post-build Actions section are two buttons: **Save** and **Apply**.

Cambios locales



```
Node.java (~/.Dropbox/postitulos/testing/demo/tda_test) - gedit
File Edit View Search Tools Documents Help
Open [file icon]
1 public class Node {
2     private int valor;
3     private Node previous;
4     private Node next;
5
6     public void setValue(int value) {
7         this.value = value;
8     }
9
```

```
jsimmond@pita:~/Dropbox/postitulos/testing/demo/tda_test$ gedit Node.java
jsimmond@pita:~/Dropbox/postitulos/testing/demo/tda_test$ git add Node.java
jsimmond@pita:~/Dropbox/postitulos/testing/demo/tda_test$ git commit Node.java
[master 67f65f7] Oops, changed a variable name.
 1 file changed, 1 insertion(+), 1 deletion(-)
jsimmond@pita:~/Dropbox/postitulos/testing/demo/tda_test$
jsimmond@pita:~/Dropbox/postitulos/testing/demo/tda_test$ git push
Username for 'https://github.com': jsimmond
Password for 'https://jsimmond@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 294 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/jsimmond/tda_test.git
 279eb4c..67f65f7 master -> master
jsimmond@pita:~/Dropbox/postitulos/testing/demo/tda_test$
```

TDAs: “quebramos” el build

The screenshot shows the Jenkins web interface in a Mozilla Firefox browser. The address bar shows the URL `localhost:8080/job/TDAs/10/`. The Jenkins logo and navigation links are at the top. The main content area displays the details for Build #10, which occurred on Sep 28, 2017 at 1:00:17 PM. The build status is indicated by a red ball icon. The build was started 24 seconds ago and took 1.4 seconds on the master. The build was started by the user 'admin'. The revision is 67f65f72786c4accb020280378f74b8c313d768, which is the master branch of the repository. The build log shows a single change: '1. Oops, changed a variable name.' with links to the details and the GitHub web page. The left sidebar contains links to various build-related actions like 'Back to Project', 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete Build', 'Git Build Data', 'No Tags', and 'Previous Build'. The bottom of the page shows the page generation timestamp and links to the REST API and Jenkins version information.

TDA #10 [Jenkins] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jsimmond/tda_test x TDA #10 [Jenkins] x +

localhost:8080/job/TDAs/10/ 110% Search

Jenkins search admin log out

Jenkins > TDAs > #10 [ENABLE AUTO REFRESH](#)

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Delete Build](#)

[Git Build Data](#)

[No Tags](#)

[Previous Build](#)

Build #10 (Sep 28, 2017 1:00:17 PM)

Started 24 sec ago
Took [1.4 sec](#) on master

[add description](#)

[Changes](#)

1. Oops, changed a variable name. ([detail](#) / [githubweb](#))

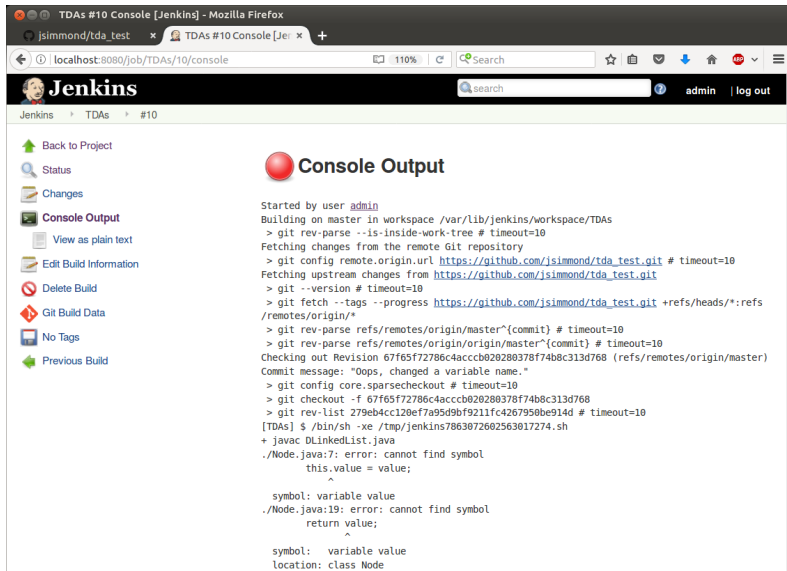
Started by user [admin](#)

Revision: 67f65f72786c4accb020280378f74b8c313d768

- refs/remotes/origin/master

Page generated: Sep 28, 2017 1:00:41 PM CLST [REST API](#) [Jenkins ver. 2.73.1](#)

TDAs: pueden ver el detalle en Console Output



The screenshot shows the Jenkins web interface in a Mozilla Firefox browser. The address bar shows the URL `localhost:8080/job/TDAs/10/console`. The Jenkins header includes the logo, the job name `TDAs`, and the build number `#10`. On the left sidebar, the `Console Output` option is selected. The main content area is titled `Console Output` and displays the following text:

```
Started by user admin
Building on master in workspace /var/lib/jenkins/workspace/TDAs
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/jsimmond/tda_test.git # timeout=10
Fetching upstream changes from https://github.com/jsimmond/tda_test.git
> git --version # timeout=10
> git fetch --tags --progress https://github.com/jsimmond/tda_test.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 67f65f72786c4acccb020280378f74b8c313d768 (refs/remotes/origin/master)
Commit message: "Oops, changed a variable name."
> git config core.sparsecheckout # timeout=10
> git checkout -f 67f65f72786c4acccb020280378f74b8c313d768
> git rev-list 279eb4cc120ef7a95d9bf9211fc4267950be914d # timeout=10
[TDAs] $ /bin/sh -xe /tmp/jenkins7863072602563017274.sh
+ javac DLinkedList.java
./Node.java:7: error: cannot find symbol
    this.value = value;
              ^
    symbol:   variable value
    location: class Node
./Node.java:19: error: cannot find symbol
    return value;
           ^
    symbol:   variable value
    location: class Node
```


TDAs: configurar los Build Triggers

The screenshot shows the Jenkins configuration interface for a job named 'TDAs'. The 'Build Triggers' tab is active, displaying several options for triggering builds. The 'Poll SCM' option is selected with a red checkmark. Below this, a text field for the 'Schedule' contains the wildcard pattern '*****'. A yellow warning icon and message are present, advising that the current pattern might mean 'every minute' and suggesting 'H * * * *' for polling once per hour. Below the warning, it states the last run was on Thursday, September 28, 2017 at 1:58:28 PM CLST, and the next run is scheduled for the same time on the same day. The 'Ignore post-commit hooks' option is unchecked. At the bottom, the 'Build Environment' section is partially visible, showing 'Save' and 'Apply' buttons.

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ GitHub hook trigger for GITScm polling
- ☒ Poll SCM

Schedule: *****

Do you really mean "every minute" when you say "*****"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Thursday, September 28, 2017 1:58:28 PM CLST; would next run at Thursday, September 28, 2017 1:58:28 PM CLST.

Ignore post-commit hooks ☐

Build Environment

Save Apply build starts

TDAs: ahora todo esta ok

TDA [Jenkins] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jsimmond/tda_test x TDA [Jenkins] x +

localhost:8080/job/TDAs/ 110% Search

Jenkins search admin | log out

Jenkins > TDAs > [ENABLE AUTO REFRESH](#)

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Git Polling Log](#)

Project TDAs

[add description](#)

[Disable Project](#)

[Workspace](#)

[Recent Changes](#)

Permalinks

- [Last build \(#11\), 54 sec ago](#)
- [Last stable build \(#11\), 54 sec ago](#)
- [Last successful build \(#11\), 54 sec ago](#)
- [Last failed build \(#10\), 59 min ago](#)
- [Last unsuccessful build \(#10\), 59 min ago](#)
- [Last completed build \(#11\), 54 sec ago](#)

Build History [trend](#)

find x

- #11** Sep 28, 2017 1:58 PM
- #10** Sep 28, 2017 1:00 PM
- #6** Sep 28, 2017 12:45 PM
- #5** Sep 28, 2017 1:11 AM
- #4** Sep 27, 2017 11:13 PM

Ahora cambiemos de repositorio

Repo: `https://github.com/jsimmond/addressbook`

- implementación simple de una libreta de direcciones
- tiene un pequeño test suite
- ya tiene un archivo de configuración de Maven, con dos targets:
compile y test

Nuevo proyecto - compilación

The screenshot displays the Jenkins configuration interface for a project named 'Addressbook-compile'. The browser window shows the title 'Addressbook-compile Config [Jenkins] - Mozilla Firefox' and the URL 'localhost:8080/job/Addressbook-compile/configure'. The configuration page has several tabs: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment' (which is active), 'Build', and 'Post-build Actions'.

In the 'Build Environment' tab, there are four unchecked checkboxes:

- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Use secret text(s) or file(s)
- ☐ With Ant

Below these is the 'Build' section, which contains a configuration for 'Invoke top-level Maven targets'. The 'Goals' field is set to 'compile'. There is an 'Advanced...' button and an 'Add build step' dropdown.

The 'Post-build Actions' section has an 'Add post-build action' dropdown.

At the bottom of the configuration page are two buttons: 'Save' and 'Apply'.

Nuevo proyecto - análisis estático

The screenshot shows the Jenkins configuration page for a job named 'Addressbook-checkWarnings'. The browser is Mozilla Firefox. The page has several tabs: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'Build' tab is active, showing 'Invoke top-level Maven targets' with a text field for 'Goals' containing '-P metrics pmd:pmd'. Below this is an 'Add build step' button. The 'Post-build Actions' tab is also visible, showing 'Publish PMD analysis results' with a text field for 'PMD results'. A blue arrow points to the 'Publish PMD analysis results' option with the text 'Si no tienen esta opción, falta instalar el plugin de PMD'. At the bottom of the page are 'Save' and 'Apply' buttons.

Addressbook-checkWarnings Config [Jenkins] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jsimmond/tda_test x jsimmond/addressbo: x Addressbook-checkW: x +

localhost:8080/job/Addressbook-checkWarnings/configure 110% Search

Jenkins > Addressbook-checkWarnings >

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Build

Invoke top-level Maven targets

Goals -P metrics pmd:pmd

Advanced...

Add build step

Post-build Actions

Publish PMD analysis results

PMD results

Fileset includes setting that specifies the generated raw PMD XML report files, such as **/pmd.xml. Basedir of the fileset is the workspace root. If no value is set, then the default **/pmd.xml is used. Be sure not to include any non-report files into this pattern.

Advanced...

Save Apply

Si no tienen esta opción, falta instalar el plugin de PMD

Nuevo proyecto - tests unitarios

The screenshot shows the Jenkins configuration interface for a project named 'Addressbook-unit_tests'. The browser address bar indicates the URL is 'localhost:8080/job/Addressbook-unit_tests/configure'. The 'Build' tab is selected, showing the following configuration:

- General**: Includes a checkbox for 'With Ant' which is currently unchecked.
- Build**: Contains a section 'Invoke top-level Maven targets' with a red 'X' icon. Below this, the 'Goals' field is set to 'test'. An 'Advanced...' button is visible to the right of the goals field.
- Post-build Actions**: Includes a button to 'Add post-build action'.

At the bottom of the configuration area, there are 'Save' and 'Apply' buttons. The footer of the page states: 'Page generated: Sep 28, 2017 2:23:47 PM CLST [REST API](#) [Jenkins ver. 2.73.1](#)'.

Ahora podemos definir un Pipeline

Los tres proyectos que definimos recién están relacionados:

- primero queremos compilar el código
- después revisar los warnings que se generan en forma estática
- y por ultimo queremos ejecutar los casos de prueba

Debemos construir un pipeline para enlazar los proyectos.

Creando un Pipeline

Dashboard [Jenkins] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jsimmond/tda_test x jsimmond/addressbo x Dashboard [Jenkins] x +

localhost:8080 110% Search

Jenkins search admin log out

Jenkins

ENABLE AUTO REFRESH

add description

All All-together +

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	Addressbook-neckWarnings	15 min - #4	N/A	5.1 sec
●	☁	Addressbook-cobertura	N/A	2 hr 16 min - #1	44 sec
●	☀	Addressbook-compile	15 min - #3	N/A	3.6 sec
●	☀	Addressbook-unit_tests	14 min - #7	14 hr - #3	5.1 sec
●	☀	TDA...	30 min - #11	1 hr 28 min - #10	1.6 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

master

1 Idle

2 Idle

slave_fly (offline)

Page generated: Sep 28, 2017 2:29:16 PM CLST [REST API](#) Jenkins ver. 2.73.1

Opcion para crear pipeline

Creando un Pipeline

New View [Jenkins] - Mozilla Firefox

jsimmond/tda_test x jsimmond/addressbo x New View [Jenkins] x +

localhost:8080/newView 110% Search

Jenkins search admin | log out

Jenkins

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

Build Queue

No builds in the queue.

Build Executor Status

- master
 - 1 Idle
 - 2 Idle
- slave_fly (offline)

View name

☒ **Build Pipeline View**
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

☐ **List View**
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

☐ **My View**
This view automatically displays all the jobs that the current user has an access to.

OK

Page generated: Sep 28, 2017 2:54:31 PM CLST [REST API](#) [Jenkins ver. 2.73.1](#)

Deben indicar el proyecto inicial ...

The screenshot shows the Jenkins 'Edit View [Jenkins]' configuration page for a pipeline named 'ejemplo_pipeline'. The page is divided into several sections:

- Left Sidebar:** Contains navigation links for 'New Item', 'People', 'Build History', 'Edit View' (selected), 'Delete View', 'Manage Jenkins', 'My Views', and 'Credentials'.
- Build Queue:** Shows 'No builds in the queue.'
- Build Executor Status:** Lists two executors: 'master' (Idle) and 'slave_fly' (offline).
- Main Configuration Area:**
 - Name:** 'ejemplo_pipeline'
 - Description:** A large text area for description.
 - Filter build queue:** A checkbox.
 - Filter build executors:** A checkbox.
 - Build Pipeline View Title:** A text input field.
 - Pipeline Flow:**
 - Layout:** A dropdown menu set to 'Based on upstream/downstream relationship'.
 - Upstream / downstream config:** A section with a 'Select Initial Job' dropdown menu. The dropdown is open, showing a list of jobs: 'Addressbook-checkWarnings' (highlighted), 'Addressbook-cobertura', 'Addressbook-compile', 'Addressbook-unit_tests', and 'TDAs'.
 - Trigger Options:** Includes 'OK' and 'Apply' buttons.

...y enlazar el proyecto de revisión estática ...

Addressbook-checkWarnings Config [Jenkins] - Mozilla Firefox

jsimmond/tda_test x jsimmond/addressbo... Addressbook-checkW... +

localhost:8080/job/Addressbook-checkWarnings/configure 110% Search

Jenkins > Addressbook-checkWarnings >

General Source Code Management **Build Triggers** Build Environment Build Post-build Actions

☐ Subversion

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

Projects to watch Addressbook-compile,

☐ Trigger only if build is stable

☐ Trigger even if the build is unstable

☒ Trigger even if the build fails

☐ Build periodically

☐ GitHub hook trigger for GITScm polling

☐ Poll SCM

Build Environment

☐ Delete workspace before build starts

☐ Abort the build if it's stuck

Save Apply Console Output

...y el de testing

Addressbook-unit_tests Config [Jenkins] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jsimmond/tda_test x jsimmond/addressbo: x Addressbook-unit_tes x +

localhost:8080/job/Addressbook-unit_tests/configure 110% Search

Jenkins > Addressbook-unit_tests >

General Source Code Management **Build Triggers** Build Environment Build Post-build Actions

Repository browser (Auto)

Additional Behaviours Add

☐ Subversion

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

Projects to watch Addressbook-checkWarnings

☐ Trigger only if build is stable

☐ Trigger even if the build is unstable

☒ Trigger even if the build fails

☐ Build periodically

☐ GitHub hook trigger for GITScm polling

☐ Poll SCM

Build Environment

Save Apply

build starts

El resultado es un Pipeline

The screenshot shows the Jenkins web interface in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8080/view/All-together/`. The Jenkins logo and navigation links are visible at the top. The main heading is "Build Pipeline". Below it, there are icons for Run, History, Configure, Add Step, Delete, and Manage. A visual representation of the pipeline shows three steps: "Addressbook-compile", "Addressbook-checkWarnings", and "Addressbook-unit_tests", connected by arrows. Below this, two detailed pipeline views are shown. Pipeline #3 (top) has steps: #3 Addressbook-compile (Sep 28, 2017 2:13:59 PM, 3.6 sec), #4 Addressbook-checkWarnings (Sep 28, 2017 2:14:13 PM, 5.1 sec), and #7 Addressbook-unit_tests (Sep 28, 2017 2:14:28 PM, 5.1 sec). Pipeline #2 (bottom) has steps: #2 Addressbook-compile (Sep 28, 2017 12:40:11 AM, 3.5 sec), #2 Addressbook-checkWarnings (Sep 28, 2017 12:40:24 AM, 4.9 sec), and #5 Addressbook-unit_tests (Sep 28, 2017 12:40:34 AM, 5 sec). Each step box includes a status icon, a clock icon, and a user icon.

¿Y la cobertura de estos tests?

The screenshot shows the Jenkins web interface for the 'Addressbook-cobertura' job configuration. The browser window is titled 'Addressbook-cobertura Config [Jenkins] - Mozilla Firefox' and the address bar shows 'localhost:8080/job/Addressbook-cobertura/configure'. The 'Build' tab is selected, showing the 'Invoke top-level Maven targets' section with the goal 'package cobertura:cobertura -Dcobertura.report.format=xml'. Below this, the 'Post-build Actions' section is visible, featuring the 'Publish Cobertura Coverage Report' action. The 'Cobertura xml report pattern' is set to '**/coverage.xml'. A detailed note explains that this is a file name pattern used to locate cobertura.xml report files, relative to the module root or workspace root. It also states that Cobertura must be configured to generate XML reports and that concurrent builds must be enabled for this plugin to function. At the bottom, there are 'Save' and 'Apply' buttons.

Addressbook-cobertura Config [Jenkins] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jsimmond/tda_test x jsimmond/addressbo: x Addressbook-cobertura x +

localhost:8080/job/Addressbook-cobertura/configure 110% Search

Jenkins Addressbook-cobertura

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Invoke top-level Maven targets

Goals package cobertura:cobertura -Dcobertura.report.format=xml

Advanced...

Add build step

Post-build Actions

Publish Cobertura Coverage Report

Cobertura xml report pattern **/coverage.xml

This is a file name pattern that can be used to locate the cobertura.xml report files (for example with Maven2 use `**/target/site/cobertura/coverage.xml`). The path is relative to the module root unless you have configured your SCM with multiple modules, in which case it is relative to the workspace root. Note that the module root is SCM-specific, and may not be the same as the workspace root.

Cobertura must be configured to generate XML reports for this plugin to function.

NOTE: If concurrent builds are enabled for this job, and a later build finishes before an earlier build, the later build will reduce or skip trend analysis/charting.

Save Apply Advanced...

No muy buena :-)

The screenshot shows the Jenkins web interface for the 'Addressbook-cobertura' project. The browser address bar indicates the URL is `localhost:8080/job/Addressbook-cobertura/`. The Jenkins logo and 'admin | log out' links are visible at the top. The sidebar on the left contains links for 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure', and 'Coverage Report'. The main content area is titled 'Project Addressbook-cobertura' and includes links for 'Coverage Report', 'Workspace', and 'Recent Changes'. A 'Build History' section on the left shows a list of builds with their status and timestamps. The 'Code Coverage' section on the right displays horizontal bar charts for 'Packages', 'Files', 'Classes', 'Methods', 'Lines', and 'Conditionals', each with a percentage value.

Project Addressbook-cobertura

[add description](#) [Disable Project](#)

Code Coverage

Category	Percentage
Packages	25%
Files	31%
Classes	29%
Methods	22%
Lines	21%
Conditionals	14%

Build History [trend](#)

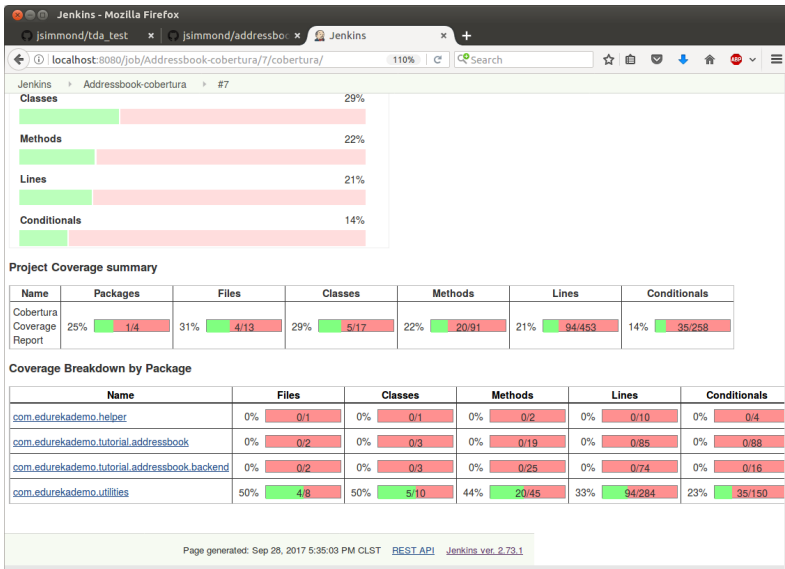
find

- #7 Sep 28, 2017 5:25 PM
- #6 Sep 28, 2017 5:20 PM
- #5 Sep 28, 2017 5:17 PM
- #4 Sep 28, 2017 5:16 PM
- #3 Sep 28, 2017 5:13 PM
- #2 Sep 28, 2017 5:09 PM
- #1

Permalinks

- [Last build \(#7\), 8 min 52 sec ago](#)
- [Last stable build \(#7\), 8 min 52 sec ago](#)
- [Last successful build \(#7\), 8 min 52 sec ago](#)
- [Last failed build \(#6\), 13 min ago](#)
- [Last unsuccessful build \(#6\), 13 min ago](#)
- [Last completed build \(#7\), 8 min 52 sec ago](#)

No muy buena :-)



¿Cómo se si mi proyecto esta lista para CI/CD?

No todos estan listos para incorporar CI/CD, y es importante entender donde estamos antes de empezar a cambiar procesos:

- ¿Logramos hacer ya un daily build?
- ¿Que aspectos de CI ya estan incorporados en el proceso de desarrollo?
- ¿Seguimos una metodologia basada en testing (TDD)?

Ejemplo

	P1	P2	P3	P4
Technology (Program Language)	J, C++	J	J	C
Platform	Lin	Lin	Lin	Hpux
Daily Build				
Compilation				
Sanity check				
Unit testing				
Installation				
Acceptance tests				
Reporting				
Policy				
Continuous Integration				
Anytime integration				
Compilation				
Sanity check				
Reporting				
Policy				
Test-Driven Development				
Metrics (Estimations)				
Integration feedback time	24h	15m	24h	3w
Build feedback time	24h	15m	24h	3w
Test coverage	?	50-80	80	?

Legend

	Not started
	Working on it
	We have it

Del paper “Measuring Continuous Integration Capability”, Vodde (2008).