

**A+**

**INTEGRACIÓN CONTINUA CON  
TRAVIS CI**

<b>Introducción</b>	<b>3</b>
<b>Requisitos</b>	<b>3</b>
<b>Conectar repositorio con Travis CI</b>	<b>3</b>
<b>Configuración pom.xml en el proyecto</b>	<b>4</b>
<b>Configurar repositorio mediante .travis.yml</b>	<b>4</b>
Secciones usadas en nuestro proyecto	5
<b>Comprobar resultados de la integración continúa</b>	<b>5</b>

## Introducción

Para la implementación de la técnica de integración continua hemos usado Travis CI. Travis CI es un servicio web gratuito con modalidades de pago que nos permite conectar un repositorio Github de manera sencilla y realizar sobre éste las compilaciones y la ejecución de las pruebas unitarias en un entorno controlado

De esta forma podremos saber con exactitud en qué momento del proyecto nuestros tests unitarios han dejado de funcionar o directamente cuando el proyecto ha dejado de compilar.

Para la compilación y ejecución de las pruebas nos apoyaremos en Maven, una herramienta para la gestión de dependencias y construcción de proyectos, lo que nos facilitará muchísimo el trabajo.

## Requisitos


- Disponer de repositorio en Github (en el caso de ser público, Travis CI será gratuito).
- Configurar la conexión de nuestro repositorio con Travis CI.
- Configurar nuestro repositorio con el archivo de configuración `.travis.yml`


## Conectar repositorio con Travis CI



Lo primero a realizar es activar la conexión de nuestro repositorio con Travis. Para ello accedemos al [servicio web de Travis](#) y conectamos con nuestra cuenta de Github.

Una vez registrados observamos un listado con nuestros repositorios, lo activamos en aquellos que nos interese. Ten en cuenta que posteriormente tendrás que realizar una configuración específica para cada proyecto/repositorio.

Travis CI

 [Blog](#) [Status](#) [Help](#)


Javier Santos 

 **Javier Santos**  
Repositories 41  
Token: 

Beta Features

See what's new!

Organizations

 **bjjr**  
Repositories 13

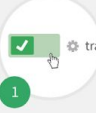
Is an organization missing?  
[Review and add your authorized organizations.](#)

Javier Santos

[Sync account](#)


We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

1




Click the repository switch on

2





Add .travis.yml file to your repository

3





Trigger your first build with a git push







javiersantos/AboutLibraries





javiersantos/AdBlockerAdvise





javiersantos/Android-DirectoryChooser

## Configuración pom.xml en el proyecto

En este archivo, tendremos que configurar ciertas configuraciones que nos ayuden a construir el proyecto, que como dijimos antes realizaremos a través de Maven.

En nuestro caso aquí hemos definido la realización del PopulateDatabase, para que en los tests se puedan hacer uso de estos datos.

## Configurar repositorio mediante .travis.yml

Para configurar nuestro repositorio necesitaremos crear un archivo YAML. Un archivo YAML, no es más que un archivo en uso para serialización de archivos, pero que en este caso Travis CI lo usa para indicar en secciones la configuración y cada paso a seguir para la compilación y ejecución de pruebas sobre nuestro proyecto. El archivo en cuestión es el .travis.yml.

```
1 language:
2   - java
3
4 services:
5   - mysql
6
7 jdk:
8   - oraclejdk7
9
10 before_install:
11   - mysql -u root < init.sql
12
13 script:
14   - cd Item\ 3/Acme-CinemaDB/
15   - mvn clean install -DskipTests=true -Dmaven.javadoc.skip=true -B -V
16   - mvn clean test -B
17   - ls -lah
18
19 cache:
20   directories:
21     - $HOME/.m2 # Cache para las dependencias de maven
22
23 dist:
24   - trusty
25
26 sudo:
27   - false
28
29 notifications:
30   - email: false
```

## Secciones usadas en nuestro proyecto

- language: indica el lenguaje de programación en el que está basado nuestro proyecto.
- services: indica los servicios que se necesitan ejecutar, como base de datos.
- jdk: indica que JDK específico de Java usar para la compilación.
- before\_install: configuraremos las acciones que tendrá que realizar antes de realizar la instalación (como por ejemplo instalar alguna dependencia adicional o inicializar la base de datos).
- script: especifica las acciones que tendrá que ejecutar a la hora de la compilación.
- cache: especifica acciones relacionadas con la caché, como por ejemplo el directorio donde se almacenará.

Una vez configurado el archivo, lo subimos a la raíz de nuestro repositorio para comenzar las pruebas de integración continua.

## Comprobar resultados de la integración continua

Accedemos nuevamente a Travis y observaremos como nuestro repositorio aparecerá ahora en verde si el resultado ha sido el correcto o en rojo en caso contrario. Ten en cuenta que cada vez que publiquemos nuevos commits Travis CI realizará nuevas compilaciones, ejecutará los tests y nos indicará si ha habido problemas o no.

Travis CI

[Blog](#) [Status](#) [Help](#)

Javier Santos

Search all repositories

My Repositories +

✗ bjjr/d12 # 16

Duration: 2 min 2 sec  
Finished: about 4 hours ago

✓ bjjr/d10 # 63

Duration: 1 min 42 sec  
Finished: about 13 hours ago

✓ bjjr/d11 # 24

Duration: 1 min 6 sec  
Finished: about 13 hours ago

✓ javiersantos/AppUpdater # 218

Duration: 8 min 29 sec  
Finished: 4 days ago

✓ javiersantos/BottomDialogs # 48

bjjr / d10

build passing

Current Branches Build History Pull Requests

More options

✓ master Add travis link -o- #63 passed

Restart build

Commit 269ea7b  
Compare de18e22..269ea7b  
Branch master  
Roberto García Calero authored and committed

Job log View config

Remove log Raw log

1 Worker information  
6 Build system information

212  
213 \$ jdk\_switcher use oraclejdk7  
214 Switching to Oracle JDK7 (java-7-oracle), JAVA\_HOME will be set to /usr/lib/jvm/java-7-oracle  
215 \$ git clone --depth=50 --branch=master https://github.com/bjjr/d10.git bjjr/d10  
216 \$ sudo service mysql start  
217 Setting up build cache  
218 \$ java -Xmx32m -version  
219 java version "1.7.0\_80"

worker\_info  
system\_info  
git\_checkout: 2.41s  
services: 0.02s  
cache: 1

Travis CI

[Blog](#) [Status](#) [Help](#)

Javier Santos

Search all repositories

My Repositories +

✗ bjjr/d12 # 16

Duration: 2 min 2 sec  
Finished: about 4 hours ago

✓ bjjr/d10 # 63

Duration: 1 min 42 sec  
Finished: about 13 hours ago

✓ bjjr/d11 # 24

Duration: 1 min 6 sec  
Finished: about 13 hours ago

✓ javiersantos/AppUpdater # 218

Duration: 8 min 29 sec  
Finished: 4 days ago

✓ javiersantos/BottomDialogs # 48

bjjr / d12

build failing

Current Branches Build History Pull Requests

More options

✗ hall Add the functional tests -o- #16 failed

Restart build

Commit 8ddcb4a  
Compare feeb508..8ddcb4a  
Branch hall  
Alejandro Hall authored and committed

Job log View config

Remove log Raw log

1 Worker information  
6 Build system information

212  
213 \$ jdk\_switcher use oraclejdk7  
214 Switching to Oracle JDK7 (java-7-oracle), JAVA\_HOME will be set to /usr/lib/jvm/java-7-oracle  
215 \$ git clone --depth=50 --branch=master https://github.com/bjjr/d10.git bjjr/d10  
216 \$ sudo service mysql start  
217 Setting up build cache  
218 \$ java -Xmx32m -version  
219 java version "1.7.0\_80"

worker\_info  
system\_info  
git\_checkout: 2.41s  
services: 0.02s  
cache: 1