

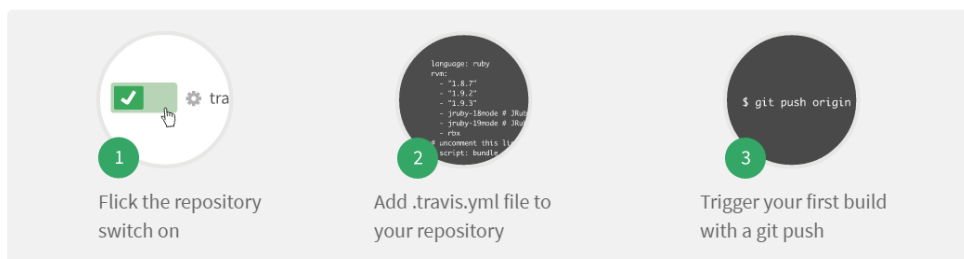
CASO PRÁCTICO: Integración de Travis con GitHub.

Se pretende conectar Travis a nuestra cuenta de GitHub para que construya el proyecto y pase las pruebas de este.

1. Conectar Travis con GitHub

Lo primero que se debe hacer es acceder a la página de Travis <https://travis-ci.org/>, una vez en la página hay que pulsar en el botón de la esquina superior izquierda donde dice registrarse con GitHub, donde se nos pedirá nuestras credenciales de GitHub. A continuación, GitHub nos pedirá autorizar a Travis para acceder a nuestra cuenta, una vez autorizado debemos esperar un tiempo para que Travis busque nuestros repositorios de GitHub.

2. Configurar Travis



Una vez accedido deben seguirse las instrucciones que ofrece Travis que podemos ver en la imagen, en nuestro caso añadimos el repositorio del proyecto y añadimos en la raíz del proyecto el fichero de configuración de Travis .travis.yml con el siguiente código:

```
language:
  - java
services:
  - mysql
jdk:
  - oraclejdk8
  - openjdk7
before_install:
  - mysql -u root < bbdd_Travis.sql
script:
  - mvn clean test
cache:
  directories:
    - $HOME/.m2
dist:
  - trusty
sudo:
  - false
notifications:
  - email: false
```

Con este fichero estamos indicando que:

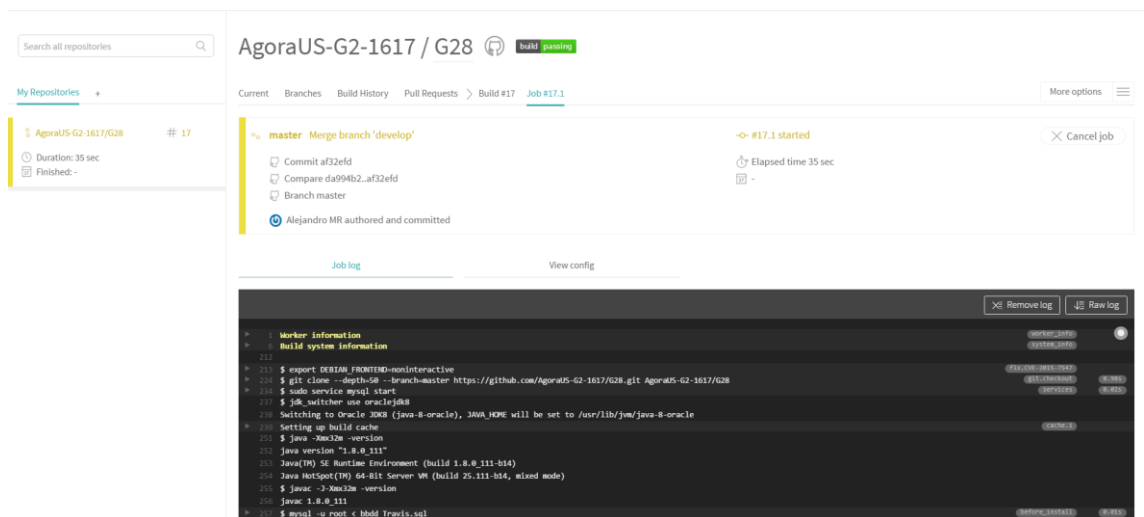
- Se trata de un proyecto java
- Se usa una base de datos de MySQL
- Debe funcionar tanto en java 7 como en java 8
- Antes de comenzar el script cargar el fichero de la base de datos bbdd_Tavis.sql que crea la base de datos y usuarios.
- Los comandos de maven a ejecutar son clean y test

Fichero bbdd.Travis.sql:

```
start transaction;
DROP DATABASE IF EXISTS `egc-frontend`;
CREATE DATABASE `egc-frontend`;
USE `egc-frontend`;
GRANT ALL ON `egc-frontend`.* TO 'user-frontend'@'localhost'
IDENTIFIED BY 'us4r-front4nd';
COMMIT;
```

3. Observar funcionamiento de Travis

A partir de este punto Travis ejecutará las pruebas cada vez que se haga un commit del proyecto en GitHub mostrando en el commit de GitHub un indicador de si las prueba se han pasado correctamente o ha surgido algún problema. Si accedemos de nuevo a la página en el instante en el que se haya ejecutado un push en GitHub, podremos observar que Travis está realizando lo indicado e el archivo de configuración (téngase en cuenta que Travis está trabajando en una maquina externa y el tiempo de respuesta puede ser alto).



The screenshot displays the Travis CI interface for the repository 'AgoraUS-G2-1617 / G28'. The build status is 'passing'. The job log shows the following steps:

```
1 Worker information
2 Build system information
3 $ export DEBIAN_FRONTEND=noninteractive
4 $ git clone --depth=1 --branch=master https://github.com/AgoraUS-G2-1617/G28.git AgoraUS-G2-1617/G28
5 $ sudo service mysql start
6 $ java -version
7 java version "1.8.0_111"
8 Java(TM) SE Runtime Environment (build 1.8.0_111-b04)
9 Java HotSpot(TM) 64-Bit Server VM (build 25.111-b04, mixed mode)
10 $ javac -j -Xmx32m -version
11 javac 1.8.0_111
12 $ mysql -u root < bdd.Travis.sql
```

Una vez acabado el proceso aparecerá todo en verde si no ha surgido algún problema durante la ejecución y la consola mostrará el siguiente mensaje

```
1679 The command "mvn clean test" exited with 0.  
▶ 1680 store build cache  
1692  
1693 Done. Your build exited with 0.
```