

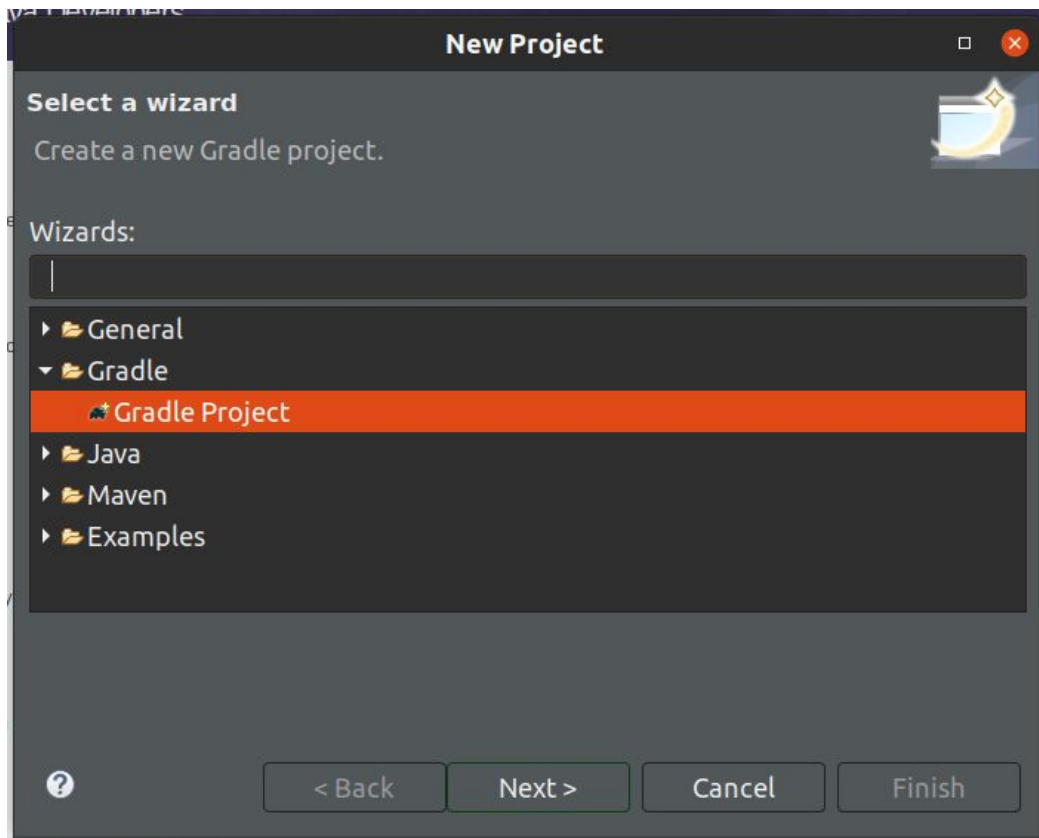
# Crear un proyecto Java con Gradle e integrarlo con Github, Travis y CodeClimate.

En este tutorial vamos a ver como generar un proyecto **Java** utilizando **Eclipse** más la herramientas de builing **Gradle** (<https://gradle.org/>) subiendo nuestro código a **Github** y a partir de ahí correr nuestra herramienta de Integración Continua en la nube. En este caso utilizaremos **Travis** (<https://travis-ci.org/>) a tal fin. Como herramienta adicional utilizaremos una herramienta de análisis estático de código([https://es.wikipedia.org/wiki/An%C3%A1lisis\\_est%C3%A1tico\\_de\\_software](https://es.wikipedia.org/wiki/An%C3%A1lisis_est%C3%A1tico_de_software)) llamada **CodeClimate** (<https://codeclimate.com/>).

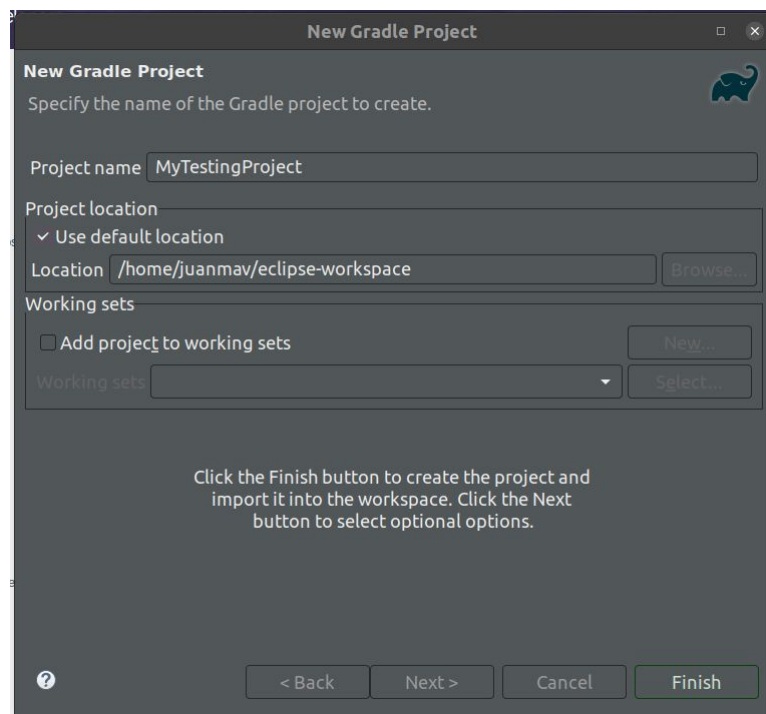
<b>Creando un Proyecto Java con Gradle Plugin usado Eclipse</b>	<b>2</b>
<b>Configuración de Git y Github</b>	<b>6</b>
<b>Configurar Travis</b>	<b>9</b>
<b>Configurar code climate</b>	<b>13</b>

## Creando un Proyecto Java con Gradle Plugin usado Eclipse

Lanzar eclipse y seleccionar tu workspace de preferencia, luego **File -> New -> Project...** .  
Luego seleccionar el proyecto de tipo **"Gradle Project"**.

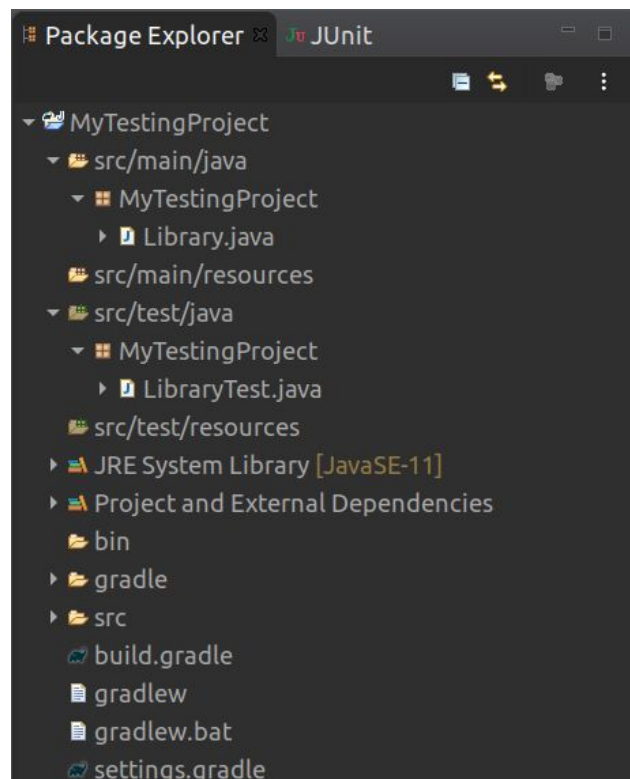


Picar “**next**” en la primer pantalla y deberias ver lo siguiente:



Completar el nombre del proyecto y picar “**Finish**”

En el package manager podran ver una estructura similar a la imagen:



A tener en cuenta Gradle crea el proyecto con una estructura puntual que debemos seguir que separa el código del código de testeo.

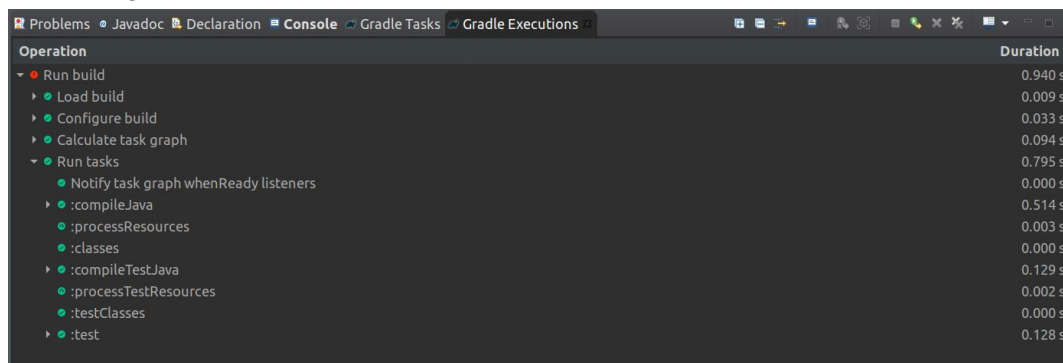
- src/main/java
- src/test/java

Además de esto ya tenemos dos clases generadas, que luego podremos quitar, para probar si el proyecto fue correctamente creado. Tener en cuenta que JUnit ya estará instalado y configurado.

- Library.java
- LibraryTest.java

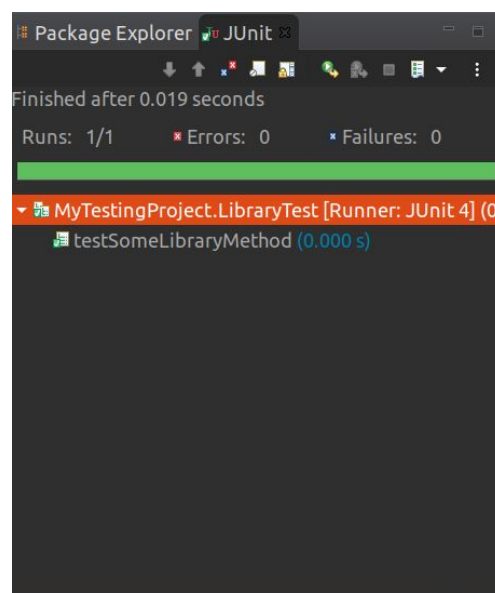
Probar nuestro proyecto:

Sobre la clase **LibraryTest.java** -> Run as / Debug as -> Gradle Test y debería aparece en la consola lo siguiente:



Operation	Duration
Run build	0.940 s
Load build	0.009 s
Configure build	0.033 s
Calculate task graph	0.094 s
Run tasks	0.795 s
Notify task graph whenReady listeners	0.000 s
:compileJava	0.514 s
:processResources	0.003 s
:classes	0.000 s
:compileTestJava	0.129 s
:processTestResources	0.002 s
:testClasses	0.000 s
:test	0.128 s

También está la posibilidad de correr solamente los Junit test. Sobre la clase **LibraryTest.java** -> Run as / Debug as -> JUnit Test



Agregar archivo **.gitignore** en la raíz del proyecto:

```
bin
.gradle
**/build/
!src/**/build/

# Ignore Gradle GUI config
.gradle-app.setting

# Avoid ignoring Gradle wrapper jar file (.jar files are usually ignored)
!gradle-wrapper.jar

# Cache of project
.gradle/taskname/cache

# # Work around https://youtrack.jetbrains.com/issue/IDEA-116898
# gradle/wrapper/gradle-wrapper.properties
```

Agregar archivo **.travis.yml** en la raíz del proyecto:

```
language: java
jdk:
  - oraclejdk11
```

# Configuración de Git y Github

1 - Registrarse en github

2- Ir a crear un nuevo repositorio, crearlo vacío.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

 juanmav ▾

Repository name \*

/ mytestingproject ✓

Great repository names are short and memorable. Need inspiration? How about **miniature-waddle**?

Description (optional)



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

### Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

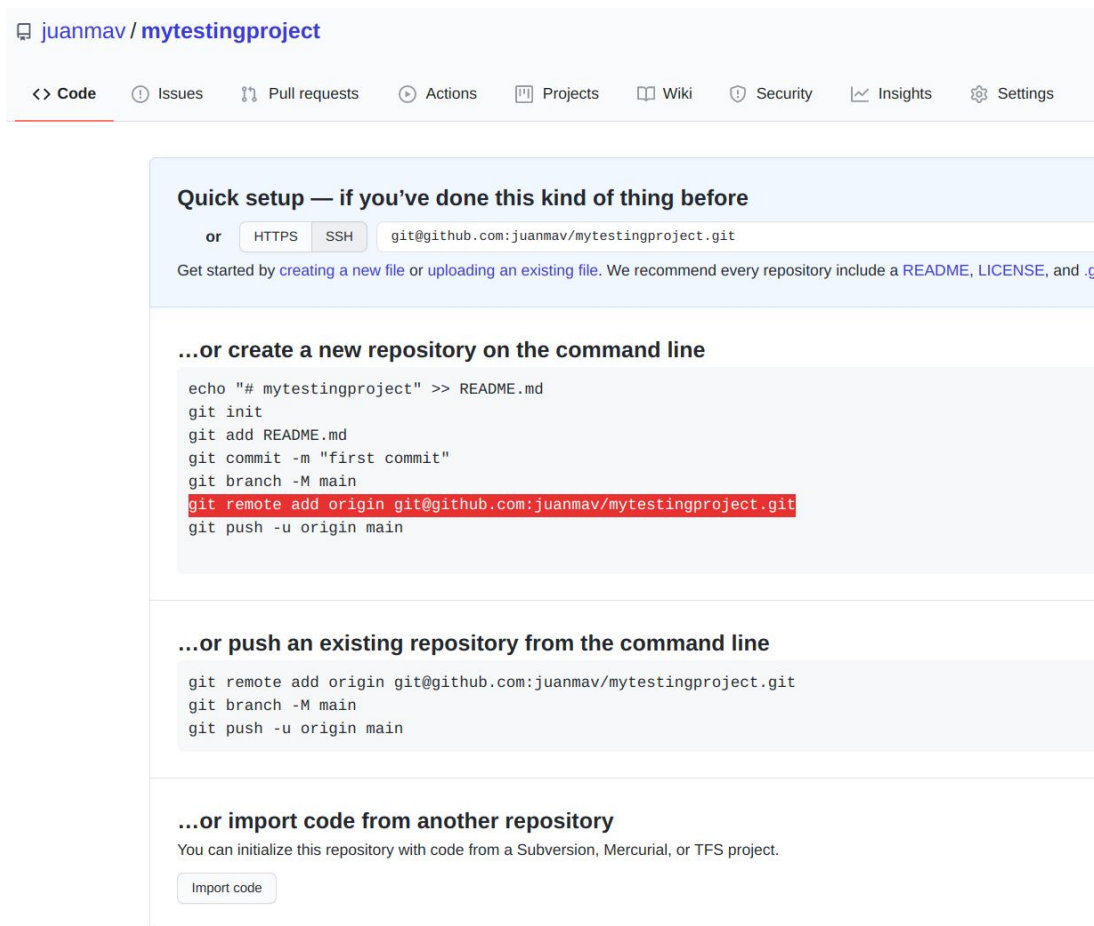
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

### 3 - El repositorio debería quedar de la siguiente manera



The screenshot shows the GitHub interface for a repository named 'mytestingproject' by user 'juanmav'. The repository page includes a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there is a 'Quick setup' section with instructions for cloning the repository using HTTPS or SSH. The SSH URL is 'git@github.com:juanmav/mytestingproject.git'. Below this, there are instructions for creating a new repository on the command line, including commands for initializing the repository, adding the README file, committing, creating a branch, adding the remote, and pushing. Finally, there are instructions for pushing an existing repository from the command line and for importing code from another repository.

juanmav / mytestingproject

<> Code ⓘ Issues 🔗 Pull requests ⚙️ Actions 📁 Projects 📖 Wiki ⓘ Security 📈 Insights ⚙️ Settings

**Quick setup — if you've done this kind of thing before**

or   `git@github.com:juanmav/mytestingproject.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# mytestingproject" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:juanmav/mytestingproject.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin git@github.com:juanmav/mytestingproject.git
git branch -M main
git push -u origin main
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

### 4 - Volver a nuestro proyecto en nuestro computador y ejecutar los siguientes comandos:

```
git init
git remote add origin git@github.com:juanmav/mytestingproject.git
git add -A
git commit -m "Proyecto creado con gradle y funcionando"
git push origin master
```

En el último paso deberán completar su **user** y **password** de github.com

Tras refrescar la web de nuestro repositorio se veria algo similar a lo siguiente

The screenshot shows the GitHub interface for the repository 'juanmav / mytestingproject'. At the top, there's a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the repository name is displayed. A secondary bar shows the current branch 'master', the number of branches '1 branch', and the number of tags '0 tags'. To the right of this bar are buttons for 'Go to file', 'Add file', and a green 'Code' button with a download icon. The main content area shows a commit by 'juanmav' with the message 'Proyecto creado con gradle y funcionando', commit hash 'b498458', and a timestamp of '15 seconds ago'. Below the commit message is a table listing the files included in the commit.

File	Commit Message	Time
.settings	Proyecto creado con gradle y funcionando	15 seconds ago
gradle/wrapper	Proyecto creado con gradle y funcionando	15 seconds ago
src	Proyecto creado con gradle y funcionando	15 seconds ago
.classpath	Proyecto creado con gradle y funcionando	15 seconds ago
.gitattributes	Proyecto creado con gradle y funcionando	15 seconds ago
.gitignore	Proyecto creado con gradle y funcionando	15 seconds ago
.project	Proyecto creado con gradle y funcionando	15 seconds ago
.travis.yml	Proyecto creado con gradle y funcionando	15 seconds ago
build.gradle	Proyecto creado con gradle y funcionando	15 seconds ago
gradlew	Proyecto creado con gradle y funcionando	15 seconds ago
gradlew.bat	Proyecto creado con gradle y funcionando	15 seconds ago

Se recomienda leer la siguiente guia de comandos git:

<https://www.hostinger.es/tutoriales/comandos-de-git>

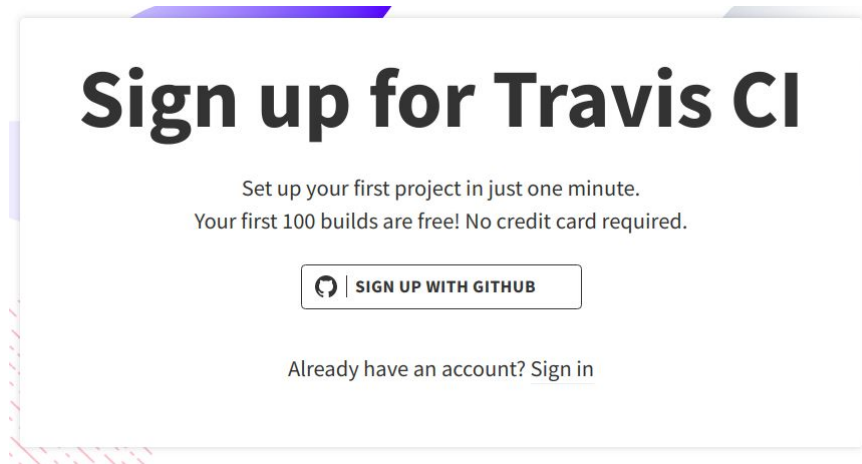
Igualmente en este punto solo necesitamos saber:

- `git add <file>`
- `git commit -m "crear un commit con files cambiados y/o agregados"`
- `git push origin master` (subir nuestros commits a github)

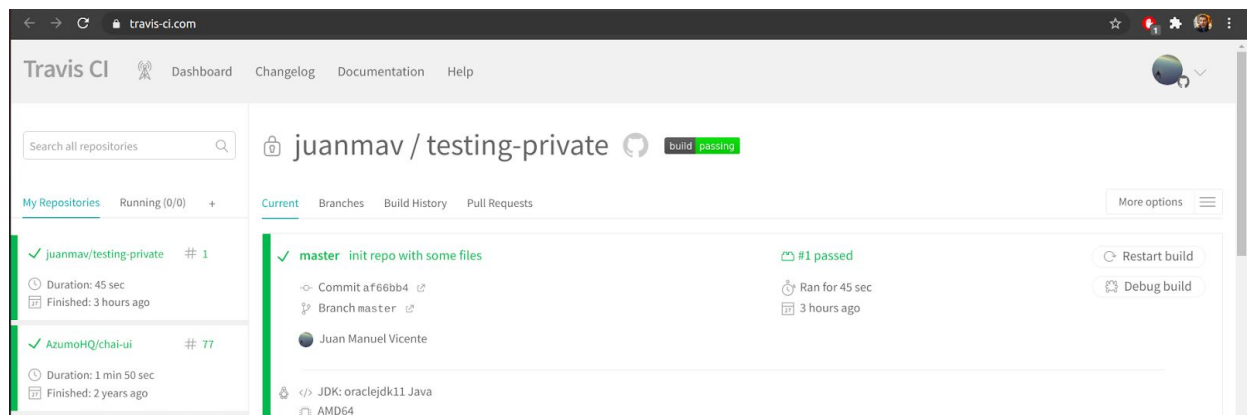


# Configurar Travis

- 1 - Ir a <https://travis-ci.com/>
- 2 - Registrarse utilizando nuestra cuenta de github



- 3- Una vez registrados Travis irá al Dashboard



- 4 - Ir a agregar un nuevo repositorio

## 5- Buscar nuestro repositorio y activarlo

Travis CI

Dashboard Changelog Documentation Help

Juan Manuel Vicente

@juanmav

Repositories Settings

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

Legacy Services Integration

testing

mytestingproject

Settings

MY ACCOUNT

Juan Manuel Vicente

Sync account

A SINGLE PLACE FOR ALL YOUR BUILDS

You can now have all your public and private repositories together at [travis-ci.com](#)

Sign up for the beta

ORGANIZATIONS

Azumo

Rollio

ConoSur

MISSING AN ORGANIZATION?

[Review and add your authorized organizations.](#)


## 6 - Ir a ver el detalle del mismo y veríamos por primera vez lo siguiente

juanmav / mytestingproject

build unknown

Current Branches Build History Pull Requests

More options

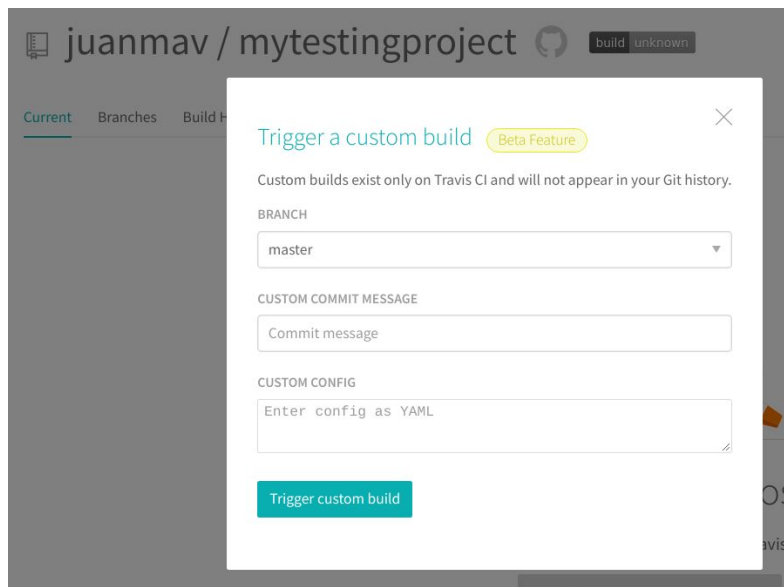


No builds for this repository

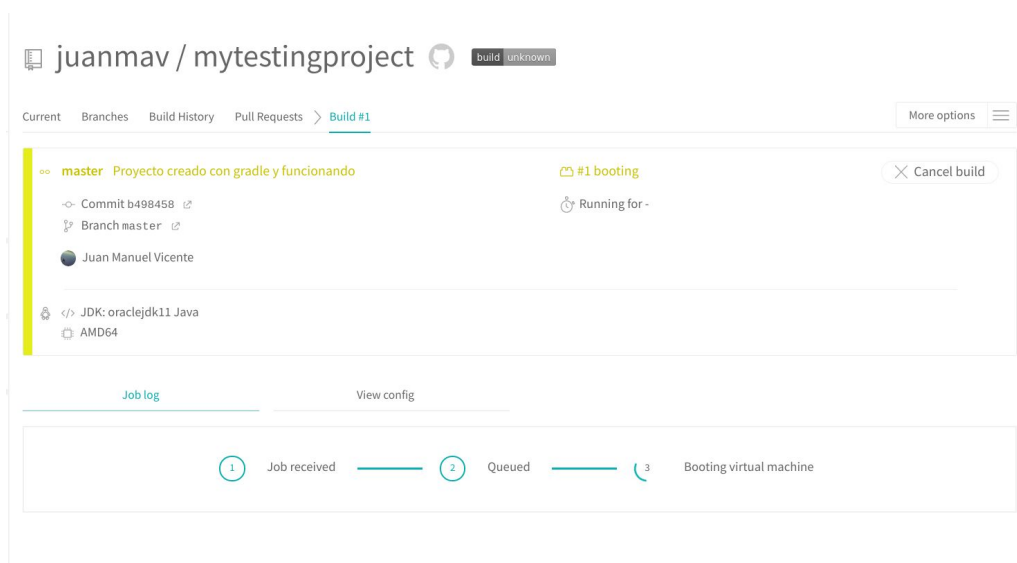
Want to start testing this project on Travis CI?

Read the Docs on Getting Started

## 7- Ir a “More Options” -> Trigger builds



## 8 - Seguir el status de Build y la corrida de tests



Al final de correr los tests y el build de nuestro proyecto veríamos el estado final

juanmav / mytestingproject build passing

Current Branches Build History Pull Requests > Build #1 More options

✓ **master** Proyecto creado con gradle y funcionando #1 passed Restart build

Commit b498458 [↗](#)

Branch master [↗](#)

Juan Manuel Vicente

JDK: oraclejdk11 Java

AMD64

[Job log](#) [View config](#)

A partir de este punto cada vez que subamos cambios a Github Travis detectará los cambios automáticamente y correrá el procesos de build and test. Al transcurrir el proceso de desarrollo veremos algo similar a la siguiente image.

✓ <b>master</b>	Correction to the properties structure in the isBetween fil	➔ #57 passed	🕒 24 sec	🔄
juanrapoport		➔ 01aefae <a href="#">↗</a>	🕒 5 years ago	
✓ <b>0.1.6</b>	distributions files 0.1.6	➔ #56 passed	🕒 33 sec	🔄
Juan Manuel Vicente		➔ e2b0c5b <a href="#">↗</a>	🕒 5 years ago	
✓ <b>master</b>	Support isLike filter function options	➔ #55 passed	🕒 32 sec	🔄
juanrapoport		➔ 5e6a908 <a href="#">↗</a>	🕒 5 years ago	
✓ <b>master</b>	Trying promise polyfills	➔ #54 passed	🕒 29 sec	🔄
Juan Manuel Vicente		➔ 7e95c16 <a href="#">↗</a>	🕒 5 years ago	
✓ <b>unit-test</b>	Trying promise polyfills	➔ #53 passed	🕒 24 sec	🔄
Juan Manuel Vicente		➔ 7e95c16 <a href="#">↗</a>	🕒 5 years ago	
✗ <b>master</b>	Merge branch 'ISO19139-2'	➔ #52 failed	🕒 30 sec	🔄
Juan Manuel Vicente		➔ b669fef <a href="#">↗</a>	🕒 5 years ago	
✗ <b>ISO19139-2</b>	Update ISO19139-2 examples and dependencies	➔ #51 failed	🕒 24 sec	🔄
Juan Manuel Vicente		➔ 48781d4 <a href="#">↗</a>	🕒 5 years ago	
✗ <b>ISO19139-2</b>	updated ISO19139-2 example	➔ #50 failed	🕒 24 sec	🔄
Juan Manuel Vicente		➔ e68b6e9 <a href="#">↗</a>	🕒 5 years ago	

Donde se puede observar como los tests fallan y luego al ser desarrollado el codigo empieza a “pasar”.


## Configurar code climate

- 1 - Ir a <https://codeclimate.com/login>
- 2 - Login with Github Account.
- 3 - Seleccionar organización "Open Source"

### Pick an organization


RollioForce

1 REPOSITORY




Open source

2 REPOSITORIES



[⊕ Create a new organization](#)


### 4- Agregar un nuevo repositorio




Open source

▼

Repositories





▼

Repositories

[⊕ Add a repository](#)

NAME

MAINTAINABILITY

TEST COVERAGE

LINES OF CODE

LAST COMMIT

### 5 - Seleccionar el repositorio de nuestra lista

**Welcome to Quality by Code Climate**  
Automated code review and quality analytics.

Add public repositories

Last synchronized with GitHub 5 minutes ago. [Sync now](#)

ConoSur/Carrito

Add Repo

ConoSur/FrameWork

Add Repo

ConoSur/camada1667

Add Repo

ConoSur/conosurmanagement

Add Repo

## 6 - Code climate comenzará el proceso de análisis y nos llevará a los resultados

The screenshot shows the Code Climate interface for the repository `juanmav/mytestingproject`. A large white notification box in the center displays a green checkmark and the text "Congratulations Your first build completed successfully!". Below this, a green button labeled "See the Results" is visible. In the background, a build progress bar shows the status "COMPLETE" with a duration of "00:00:11". Below the notification, a list of build steps is shown, each with a number, a description, a green checkmark, and a duration:

Step	Description	Status	Duration	Action
1	git clone	✓	00:00:00	
2	codeclimate validate-config	✓	00:00:01	View output
3	codeclimate prepare	✓	00:00:01	View output
4	builder pull-engines	✓	00:00:00	View output
5	structure	✓	00:00:02	View output

The screenshot shows the Code Climate interface for the repository `juanmav/mytestingproject`. The "Overview" tab is selected. The "Breakdown" section shows "13 FILES" and a green progress bar for "MAINTAINABILITY". The "Codebase summary" section displays the following metrics:

Metric	Value
MAINTAINABILITY	A 0 mins
TEST COVERAGE	0
Repository stats	
CODE SMELLS	0
DUPPLICATION	0
OTHER ISSUES	0

7 - A medida que vayamos desarrollando code climate irá generando issues para los distintos “code smells”

juanmav/tictactoetdd

Starred

OverviewProgressIssuesCodeTrendsRepo Settings

Last master build 23 days ago Refresh

Showing 3 of 3 total issues

Clear all filtersApply filters

Similar blocks of code found in 2 locations. Consider refactoring.

OPEN

52

if( (this.board[0] !== 0) && (this.board[0] === this.board[4]) && (this.board[0] === this.

53

return true;

54

) else if ( (this.board[2] !== 0) && (this.board[2] === this.board[4]) && (this.board[2] =

55

return true;

56

)

Found in TicTacToe.js and 1 other location - About 1 hr to fix

Similar blocks of code found in 2 locations. Consider refactoring.

OPEN

54

) else if ( (this.board[2] !== 0) && (this.board[2] === this.board[4]) && (this.board[2] =

55

return true;

56

)

Found in TicTacToe.js and 1 other location - About 1 hr to fix

SEVERITY

☐ Major

☐ Minor

CATEGORY

☐ Complexity

☐ Duplication

STATUS

☒ Open

☒ Confirmed

☐ Invalid

☐ Wontfix

SOURCE

☐ Code Climate

Explore 3rd-party plugins