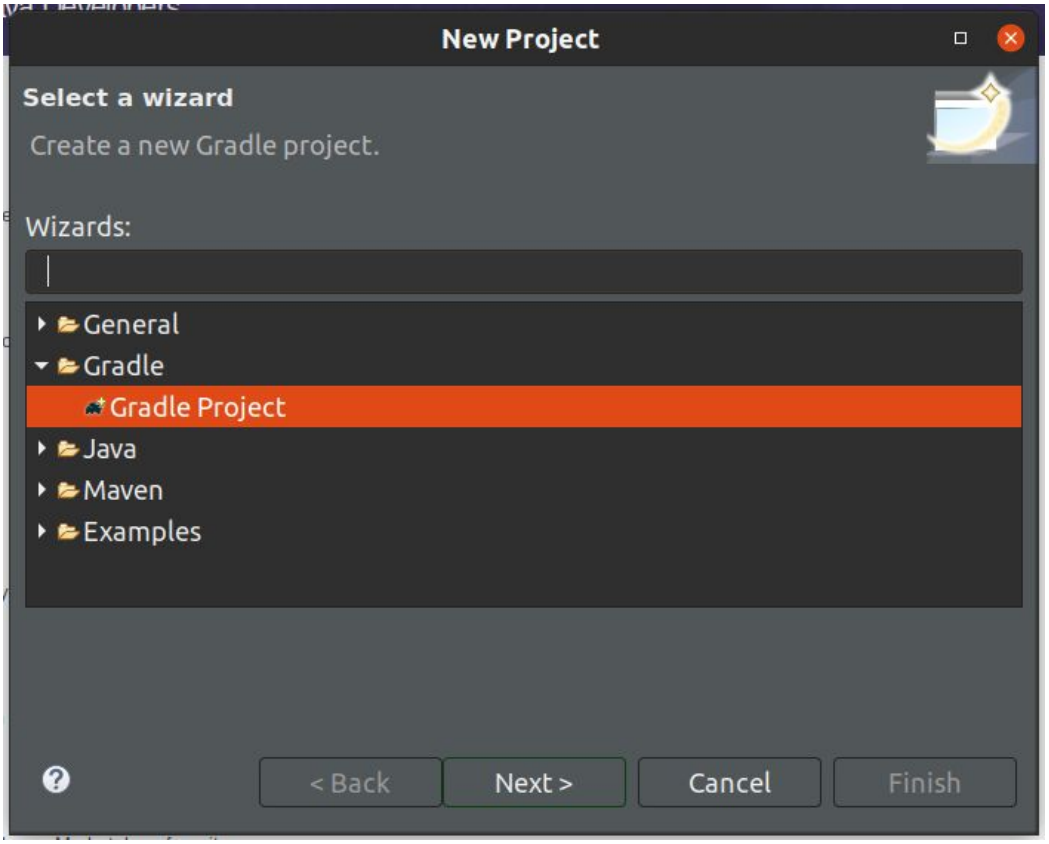
Crear un proyecto Java con Gradle e integrarlo con Github, Travis y CodeClimate.

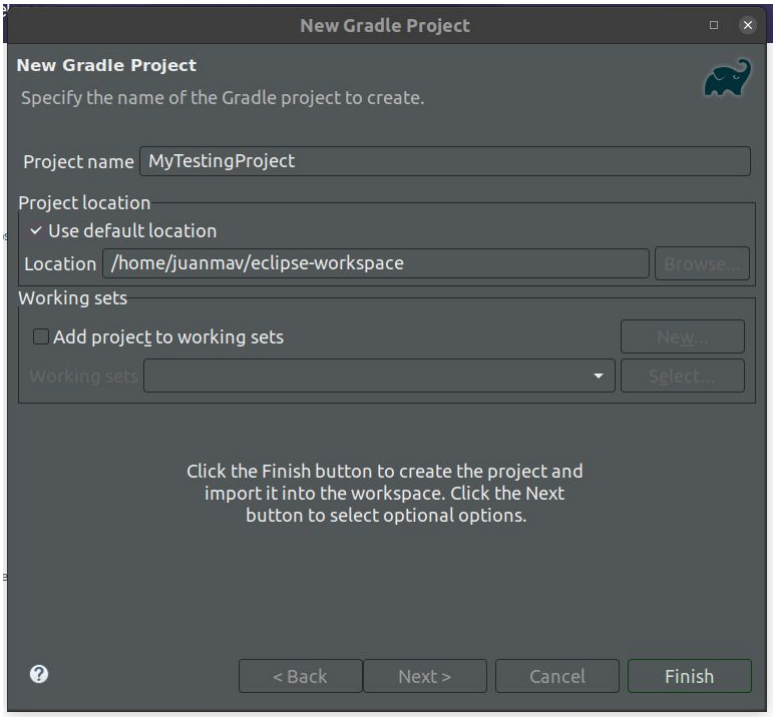
En este tutorial vamos a ver como generar un proyecto **Java** utilizando **Eclipse** más la herramientas de builing **Gradle** (https://gradle.org/) subiendo nuestro código a **Github** y a partir de ahí correr nuestra herramienta de Integración Continua en la nube. En este caso utilizaremos **Travis** (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) llamada **CodeClimate** (https://codeclimate.com/).

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**”.

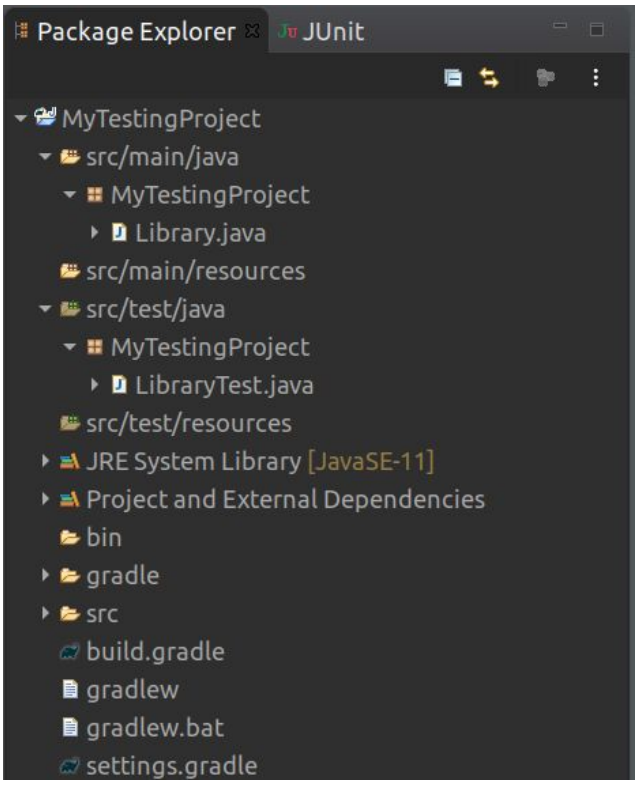


Picar “**next**” en la primer pantalla y deberias ver lo siguiente:



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

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



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

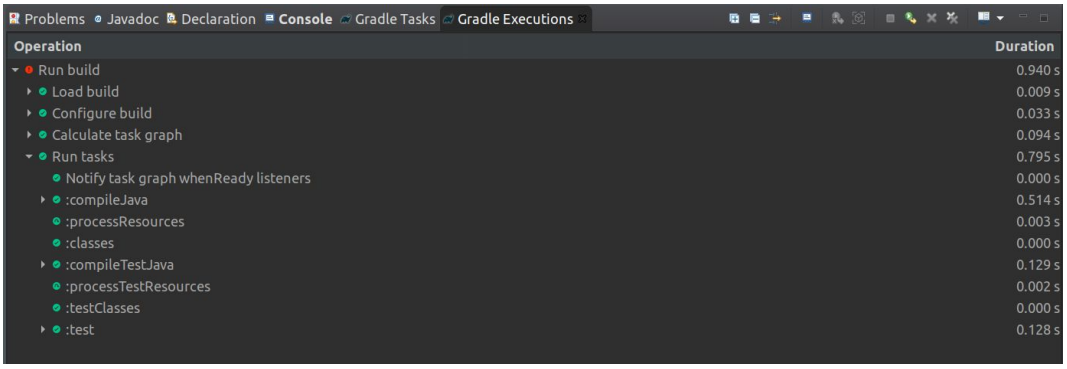
- src/main/java
- src/test/java

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.

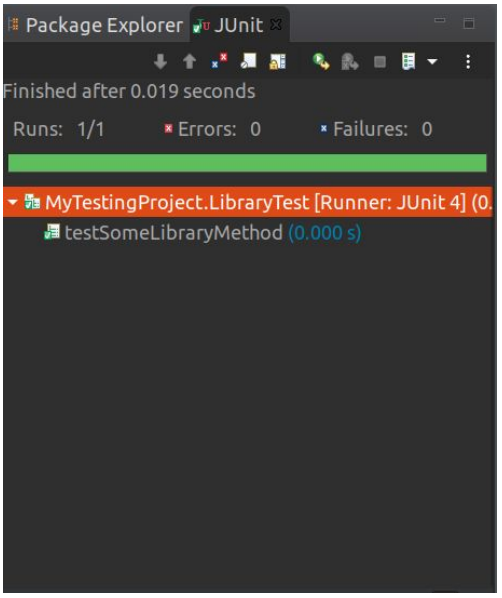
- Library.java
- LibraryTest.java

Probar nuestro proyecto:

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



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:

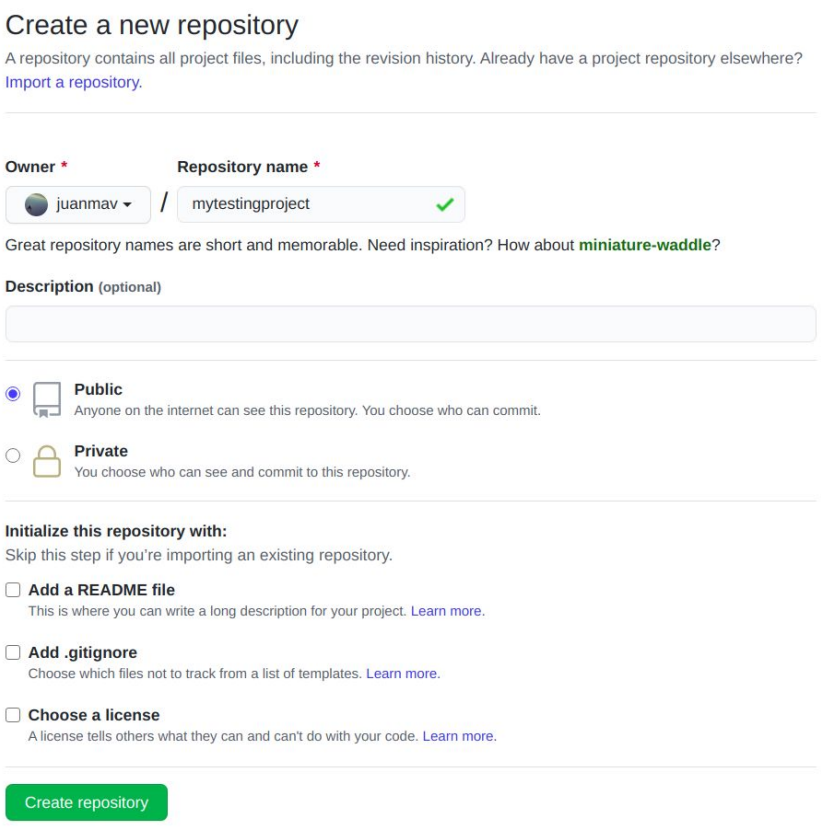
```
bin
.gradle
**/build/
!src/**/build/

# Ignore Gradle GUI config
gradle-app.setting

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

# Cache of project
.gradletasknamecache

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

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

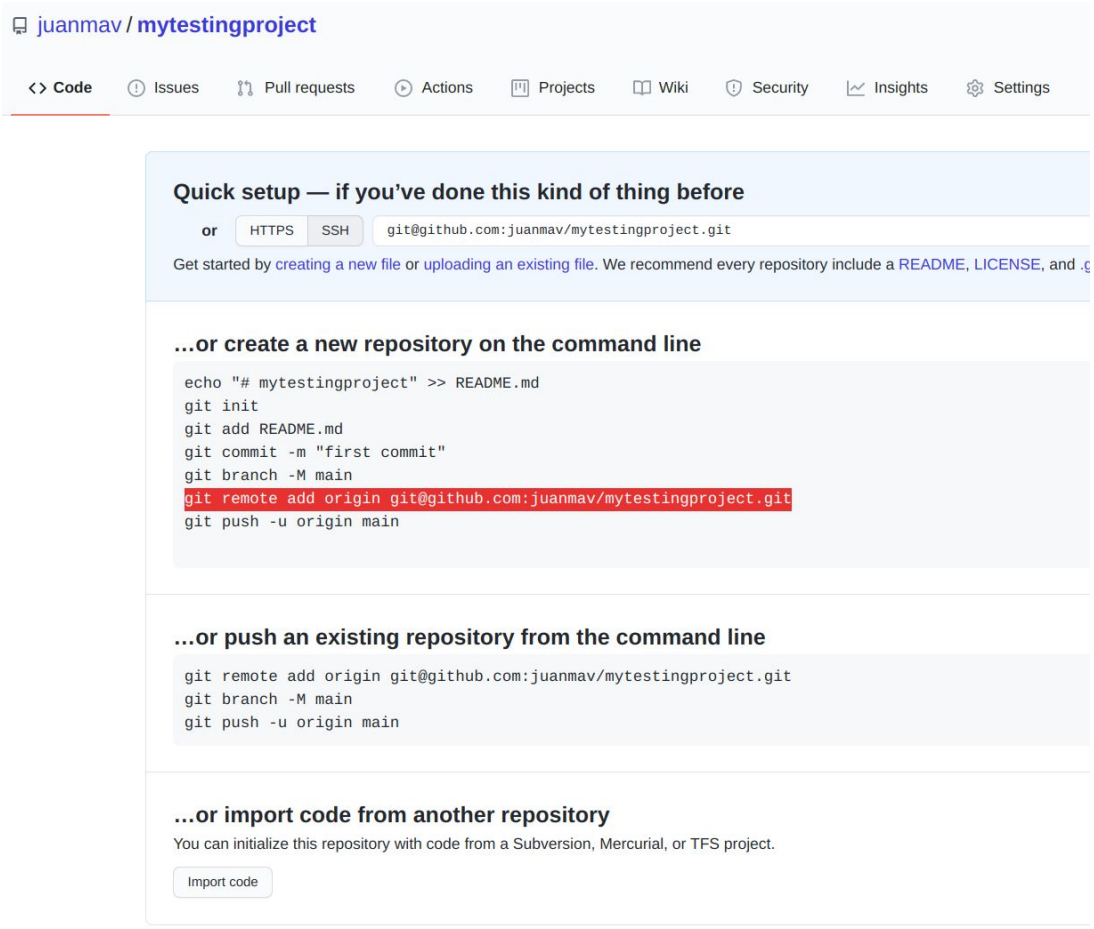
```
language: java
jdk:
  - oraclejdk11
```

Configuración de Git y Github

1 - Registrarse en github
2- Ir a crear un nuevo repositorio, crearlo vacío.



3 - El repositorio debería quedar de la siguiente manera

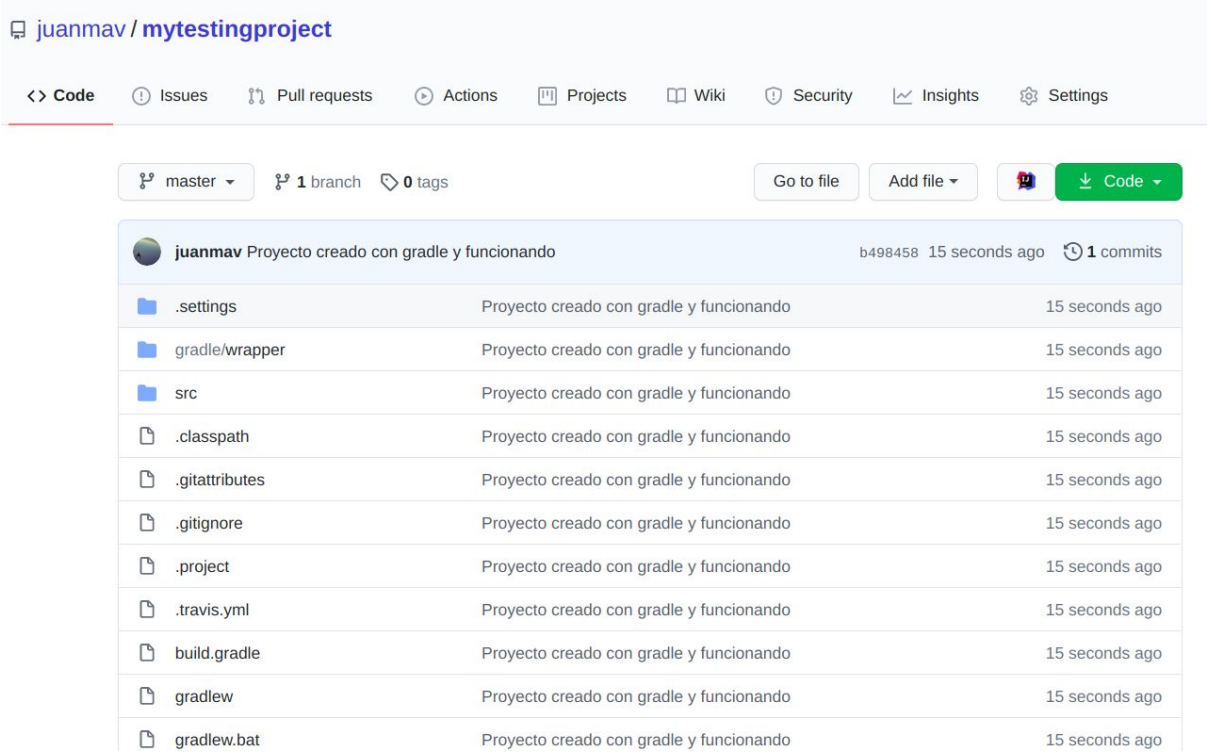


4 - Volver a nuestro proyecto en nuestro computador y ejecutar los siguientes comandos:

```
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



Se recomienda leer la siguiente guia de comandos git:

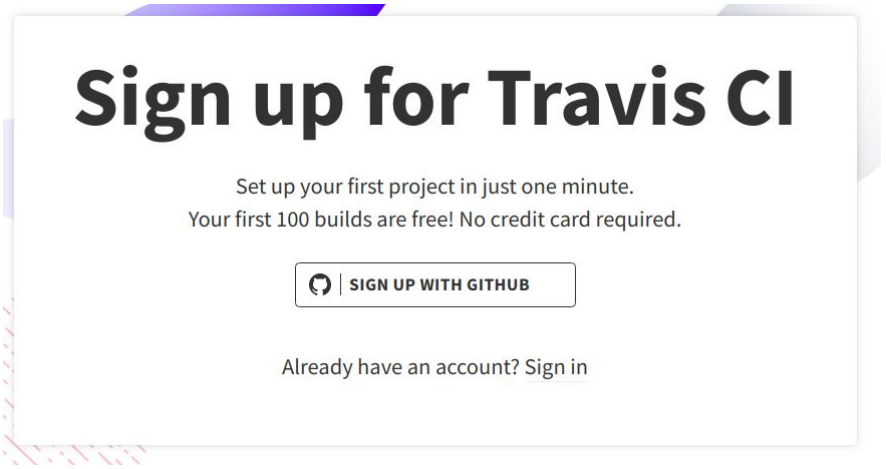
https://www.hostinger.es/tutoriales/comandos-de-git

Igualmente en este punto solo necesitamos saber:

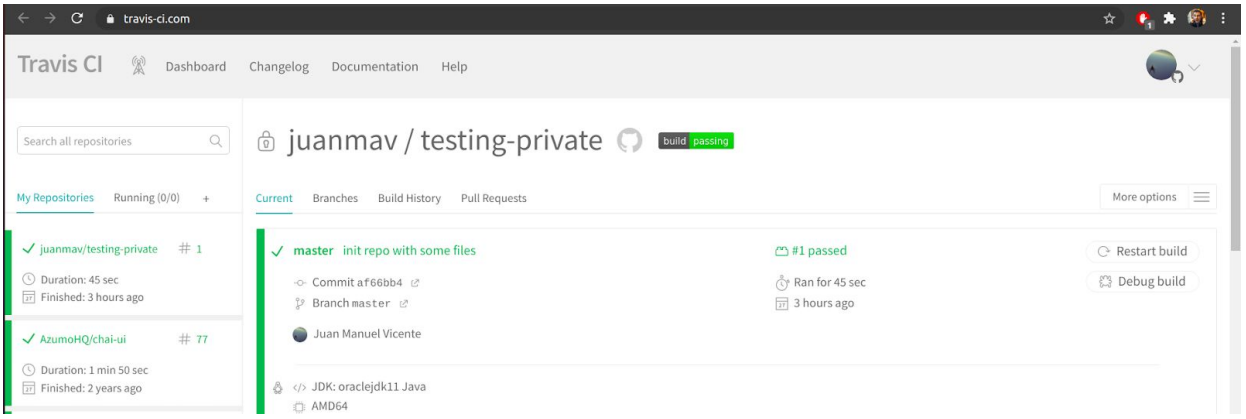
- git add <file>
- git commit -m “crear un commit con files cambiados y/o agregados”
- git push origin master (subir nuestros commits a github)

Configurar Travis

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

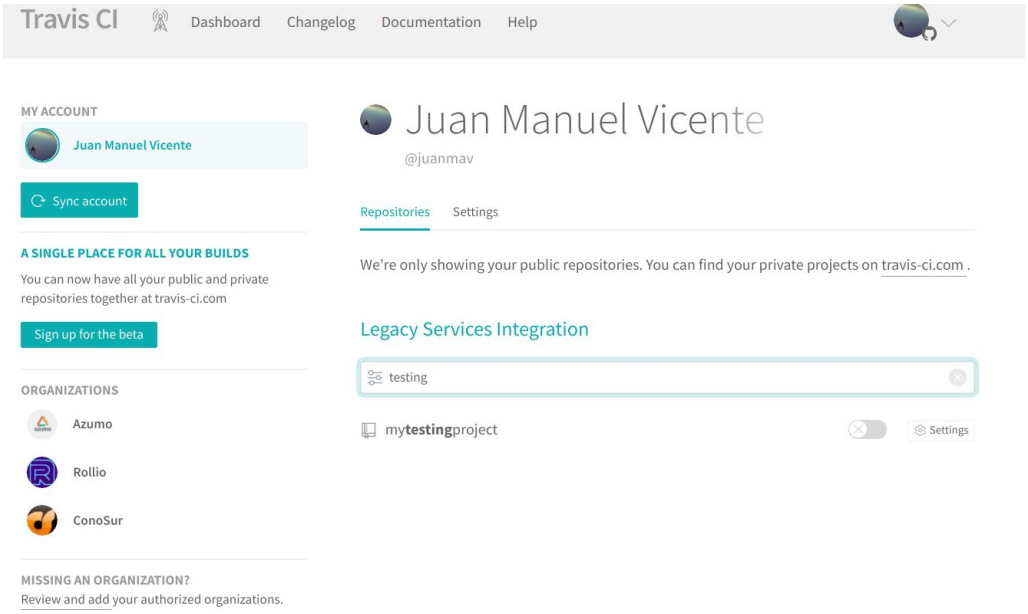


3- Una vez registrados Travis irá al Dashboard

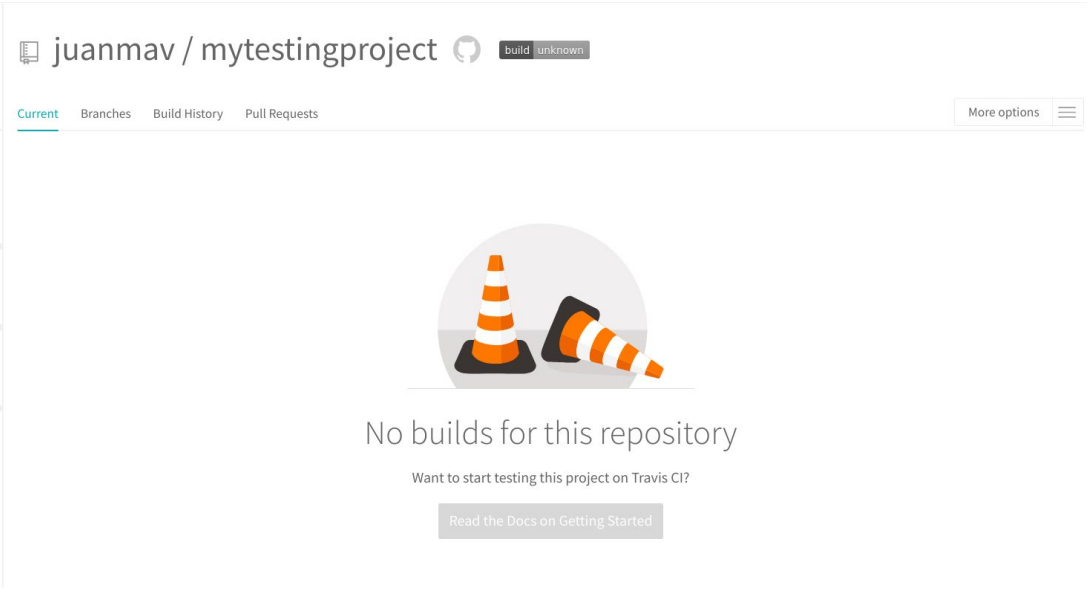


4 - Ir a agregar un nuevo repositorio

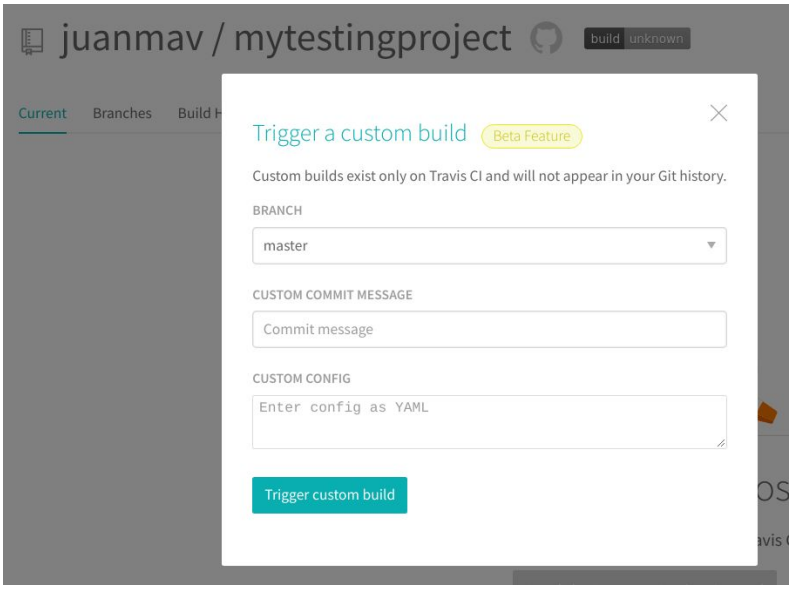
5- Buscar nuestro repositorio y activalor



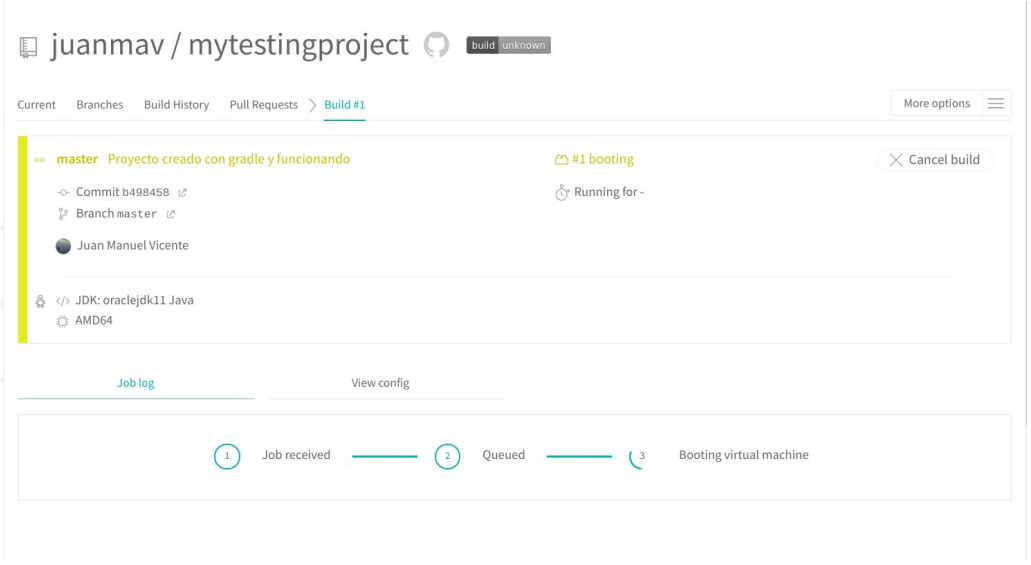
6 - Ir a ver el detalle del mismo y veríamos por primera vez lo siguiente



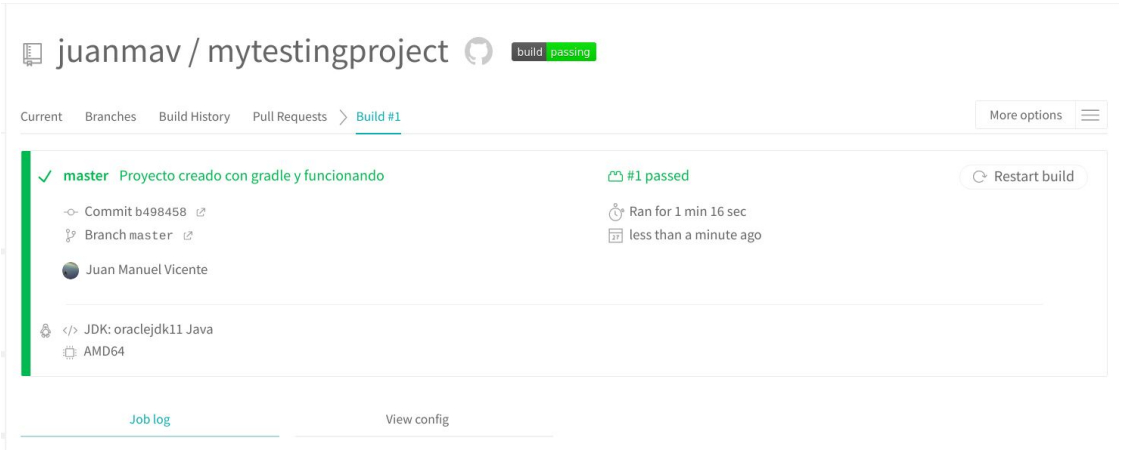
7- Ir a “More Options” -> Trigger builds



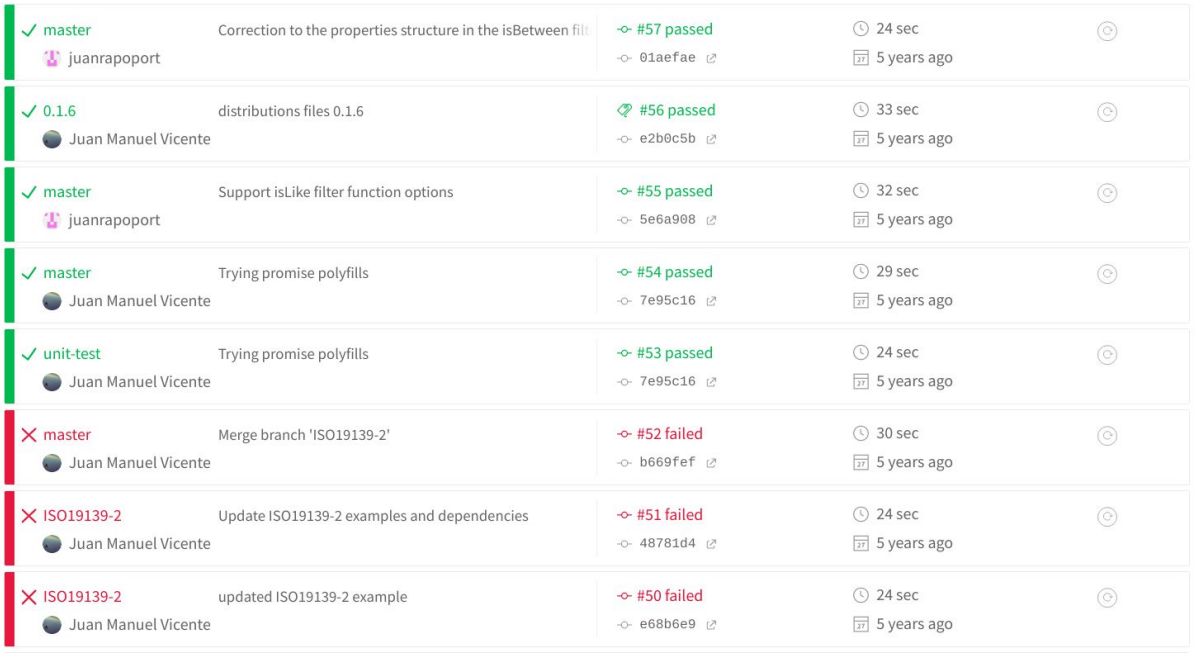
8 - Seguir el status de Build y la corrida de tests



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



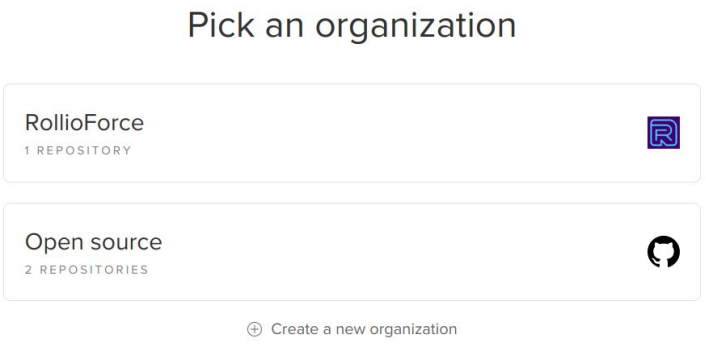
A partir de este punto cada vez que subamos cambios a Github Travis detectará los cambios automáticamente y correrá el procesos de build and test. Al transcurrir el proceso de desarrollo veremos algo similar a la siguiente image.



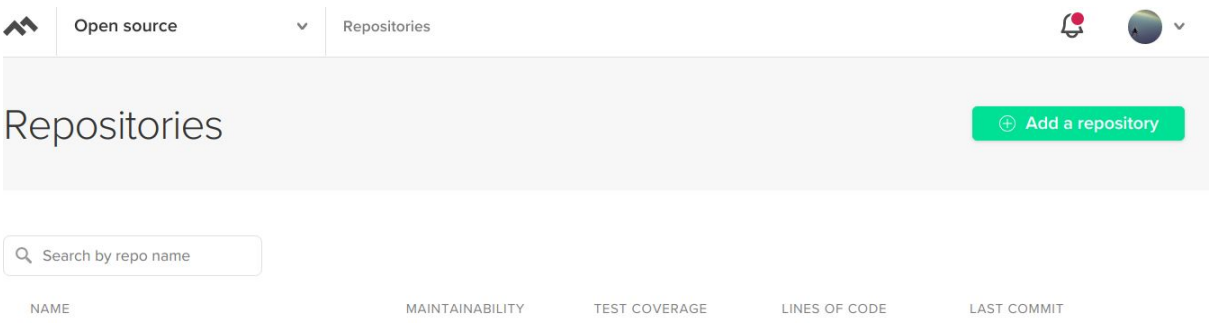
Donde se puede observar como los tests fallan y luego al ser desarrollado el codigo empieza a “pasar”.

Configurar code climate

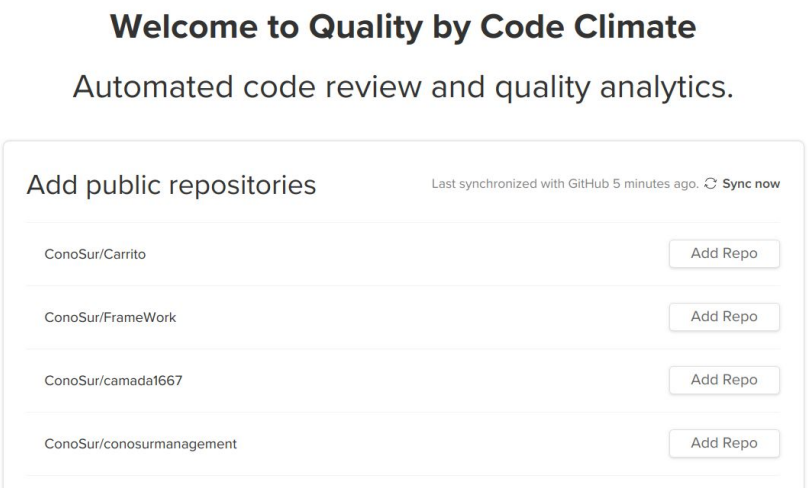
1 - Ir a https://codeclimate.com/login
2 - Login with Github Account.
3 - Seleccionar organización “Open Source”



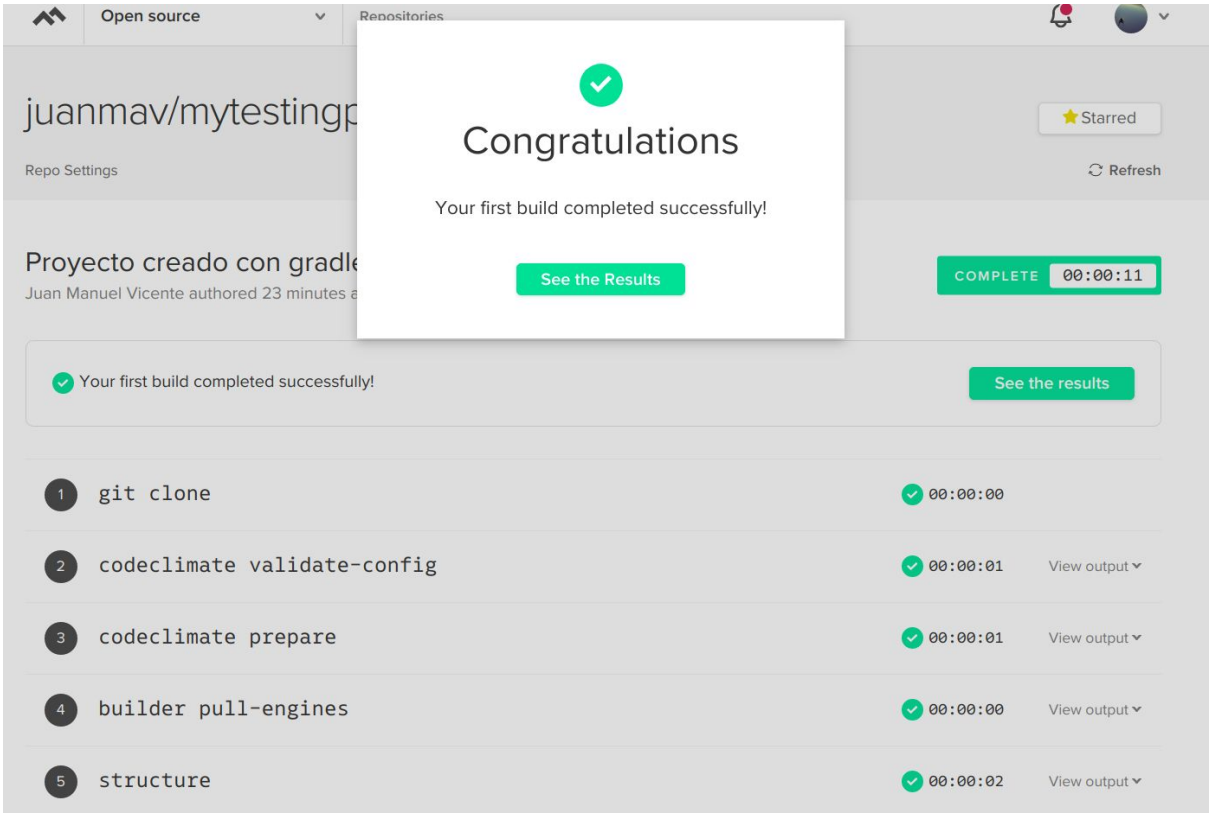
4- Agregar un nuevo repositorio

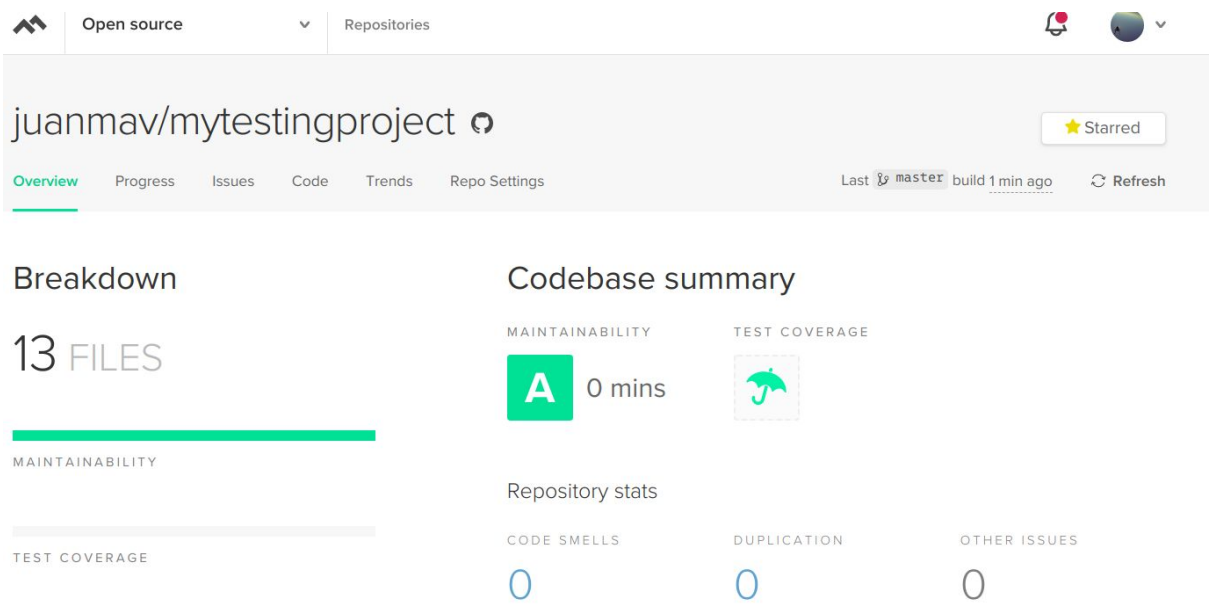


5 - Seleccionar el repositorio de nuestra lista



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





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