Escuela Superior Politécnica del Litoral

Software Engineering 2

Continuous Integration

Members:
Luis Lama
Piero Ulloa

July 28th, 2020
Guayaquil, Ecuador

Repository: luislama/IntegracionContinuaTaller

Introduction

Continuous integration is a software development practice for teams that submits their changes in code into a central repository that allows the automatic builds and tests to run immediately after commit so they know if it is ready to be released[1].

This practice is important due to it saves developers and team managers time and frustration by automating the review of new changes, find and fix bugs really fast, it enhances the productivity and quality of the system, and accelerates the updates of releases to customers

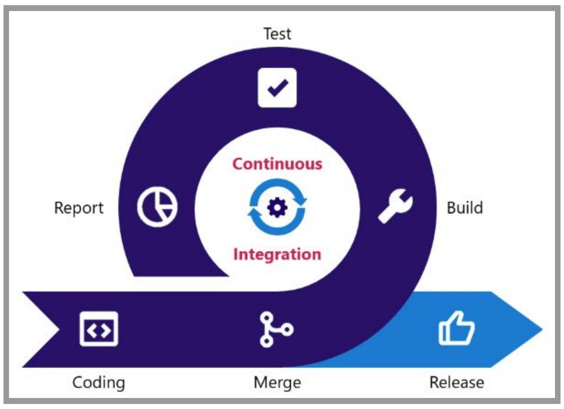


Figure 1 How continuous integration works

For this workshop we will use Jenkins locally to build and test the changes in code pushed, to a repository in github, using ngrok to expose a webhook so jenkins can receive the response from the server.

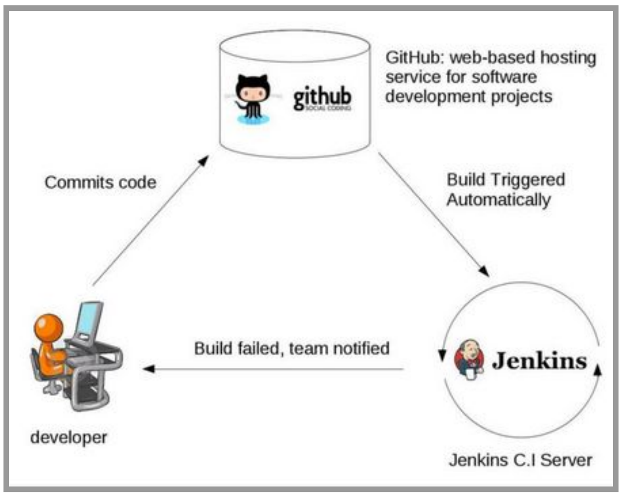

github
GitHub: web-based hosting service for software development projects
Commits code
Build Triggered Automatically
Build failed, team notified
Jenkins
developer
Jenkins C.I Server

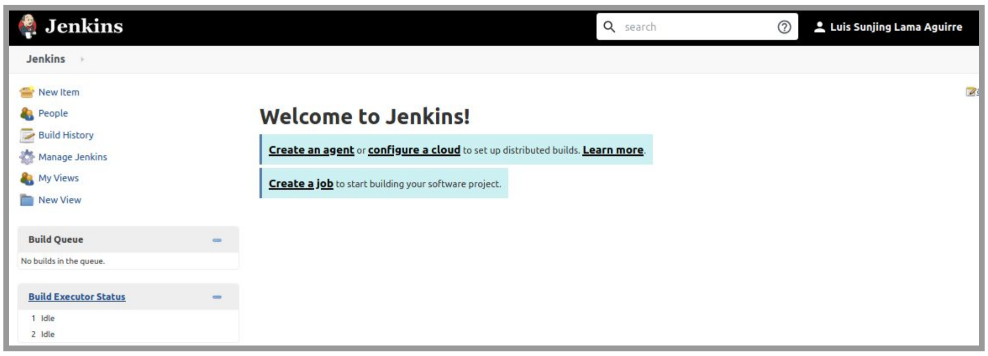
Figure 2 Continuous integration with Jenkins and Github

Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Development

Installing

Jenkins

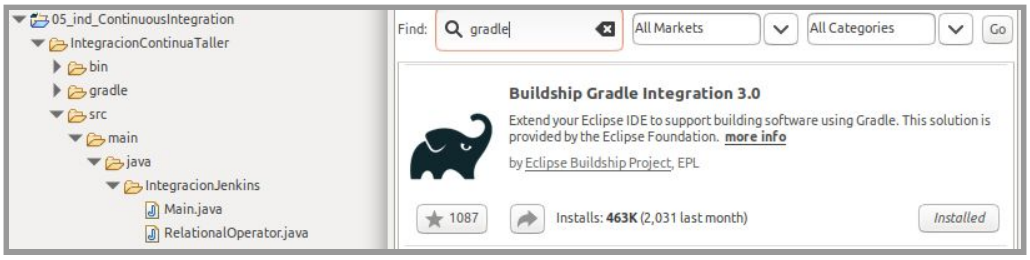


Welcome to Jenkins!

Create an agent or configure a cloud to set up distributed builds. Learn more.

Create a job to start building your software project.

Gradle



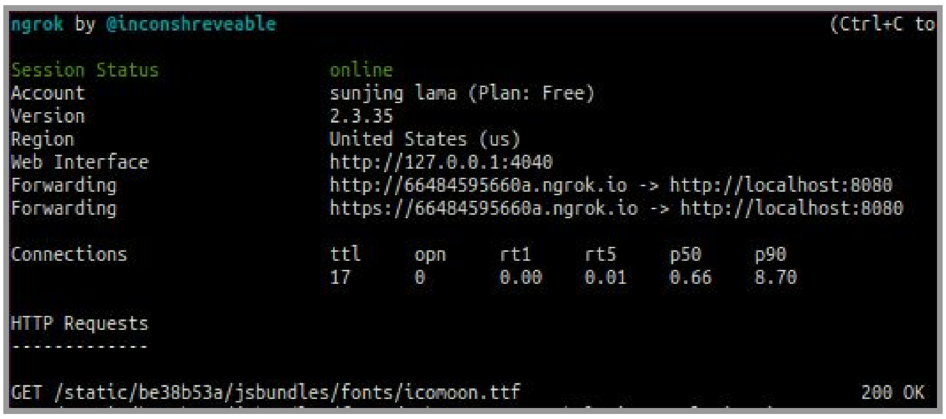
Buildship Gradle Integration 3.0

Extend your Eclipse IDE to support building software using Gradle. This solution is provided by the Eclipse Foundation. more info

by Eclipse Buildship Project, EPL

Installs: 463K (2,031 last month)

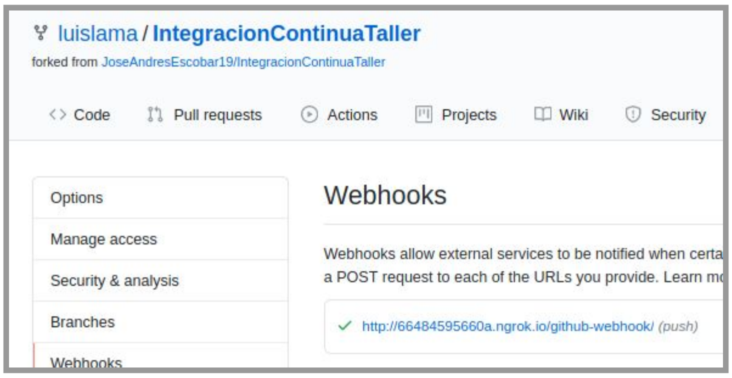
Ngrok



```
ngrok by @inconshreveable                                          (Ctrl+C to

Session Status                online
Account                       sunjing lama (Plan: Free)
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://66484595660a.ngrok.io -> http://localhost:8080
Forwarding                    https://66484595660a.ngrok.io -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              17      0       0.00    0.01    0.66    8.70

HTTP Requests
-------------

GET /static/be38b53a/jsbundles/fonts/icomoon.ttf                    200 OK
```

Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Ngrok - Github Webhook


luislama / IntegracionContinuaTaller

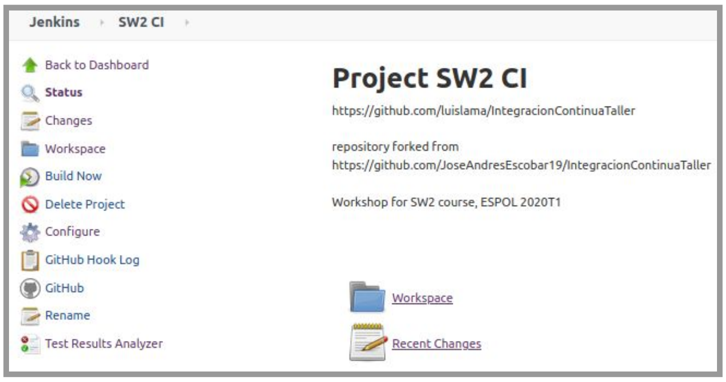
forked from JoseAndresEscobar19/IntegracionContinuaTaller

Webhooks

Webhooks allow external services to be notified when certa
a POST request to each of the URLs you provide. Learn mo

✓ http://66484595660a.ngrok.io/github-webhook/ (push)

Jenkins - Project Github



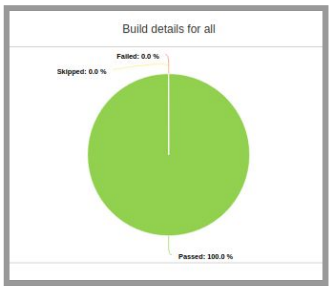
Project SW2 CI

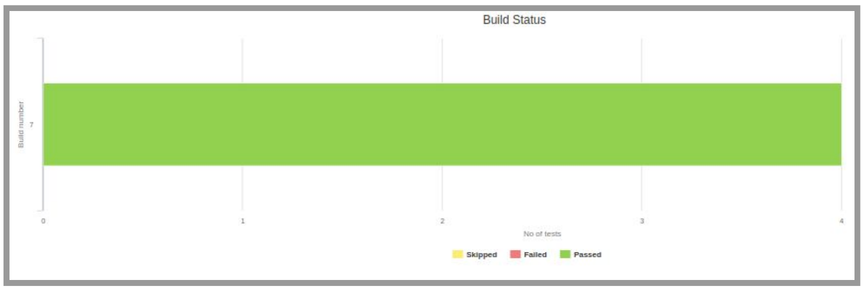
https://github.com/luislama/IntegracionContinuaTaller

repository forked from
https://github.com/JoseAndresEscobar19/IntegracionContinuaTaller

Workshop for SW2 course, ESPOL 2020T1

Build - Test Results Analyzer



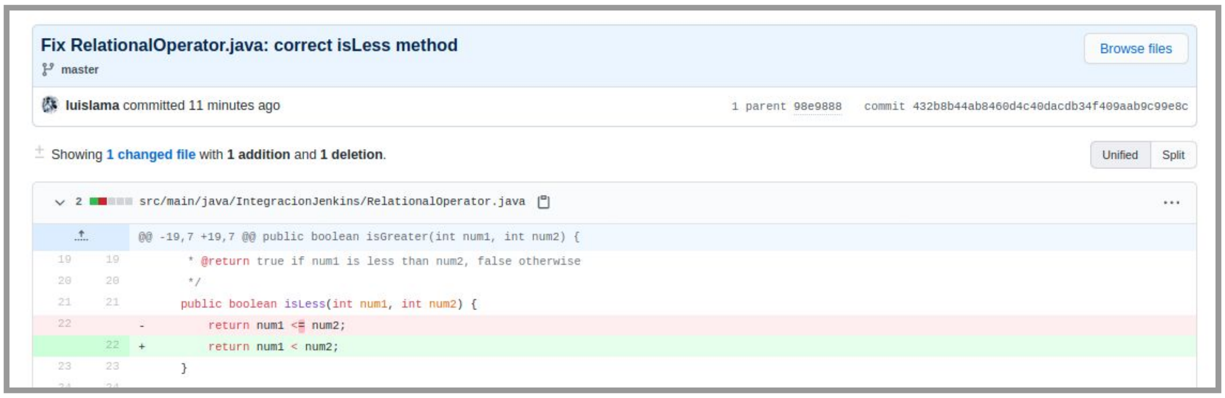


Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Workshop

Correct the error in isLess method

Upload changes



Fix RelationalOperator.java: correct isLess method

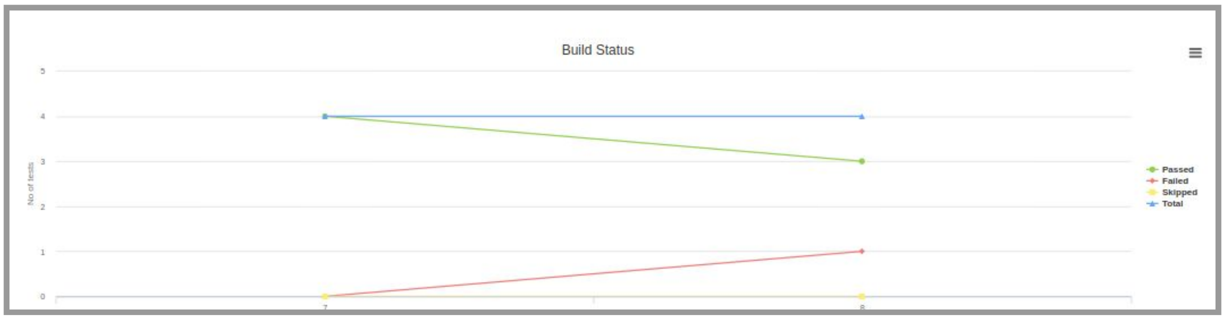
master

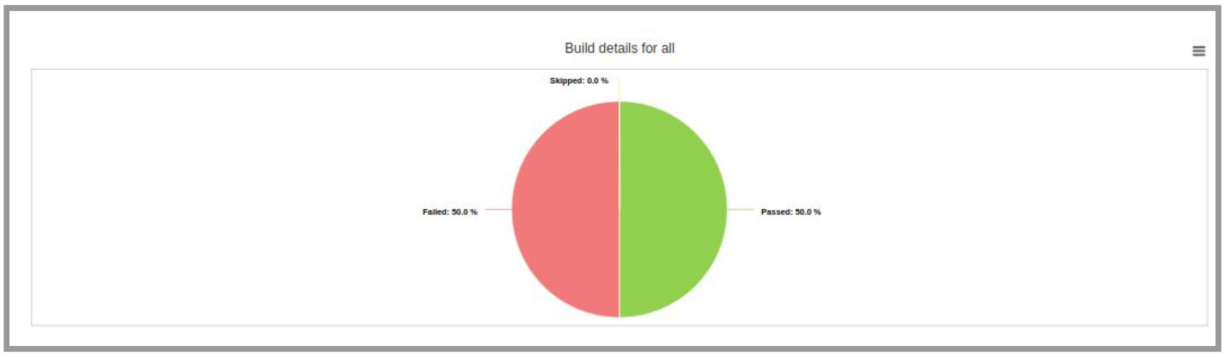
luislama committed 11 minutes ago   1 parent 98e9888   commit 432b8b44ab8460d4c40dacdb34f409aab9c99e8c

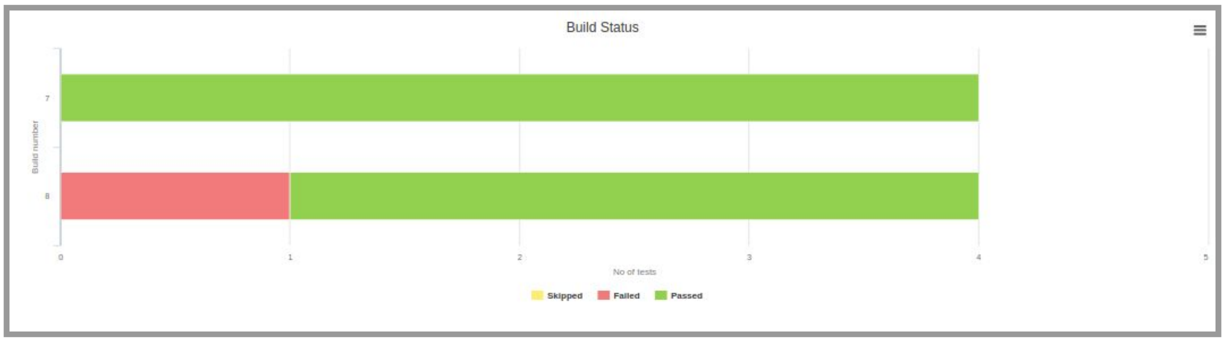
Showing **1 changed file** with **1 addition** and **1 deletion**.

```
2 src/main/java/IntegracionJenkins/RelationalOperator.java
@@ -19,7 +19,7 @@ public boolean isGreater(int num1, int num2) {
         * @return true if num1 is less than num2, false otherwise
         */
        public boolean isLess(int num1, int num2) {
-           return num1 <= num2;
+           return num1 < num2;
        }
```

Build - Test Results Analyzer



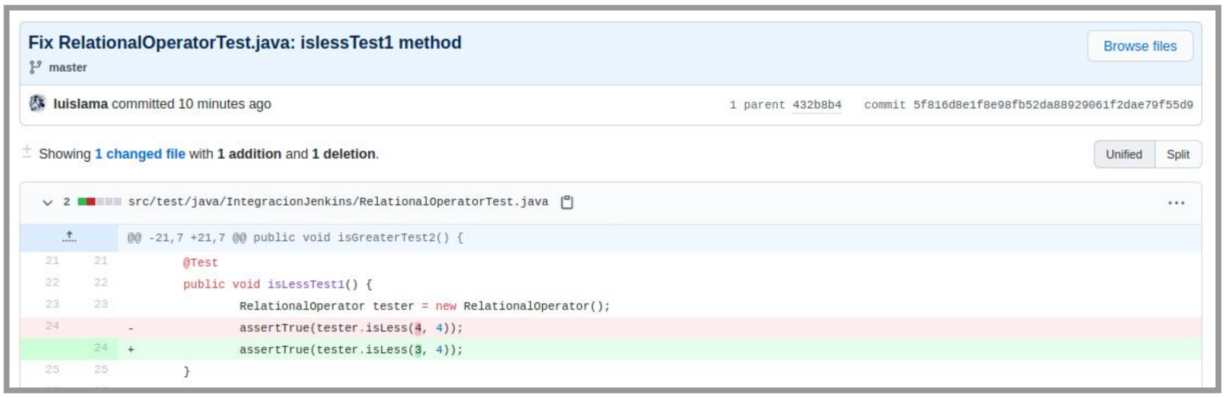




Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Fix isLessTest1 method

Upload changes



Fix RelationalOperatorTest.java: islessTest1 method

master

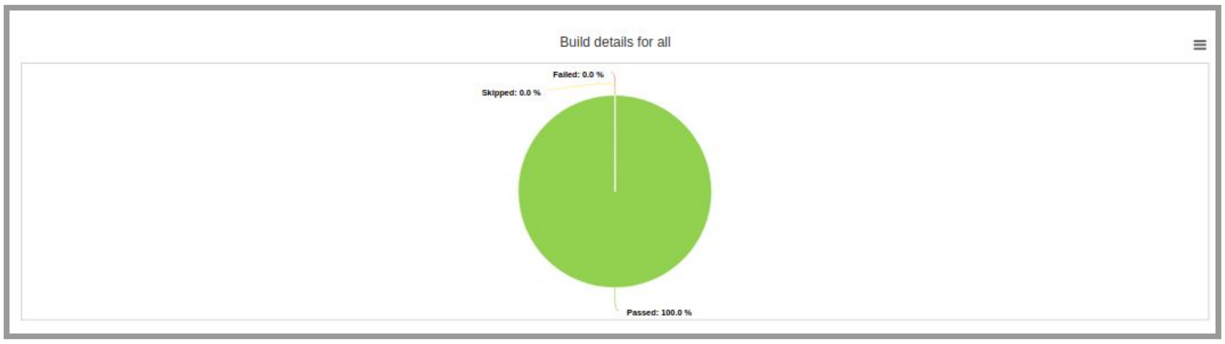
luislama committed 10 minutes ago 1 parent 432b8b4 commit 5f816d8e1f8e98fb52da88929061f2dae79f55d9

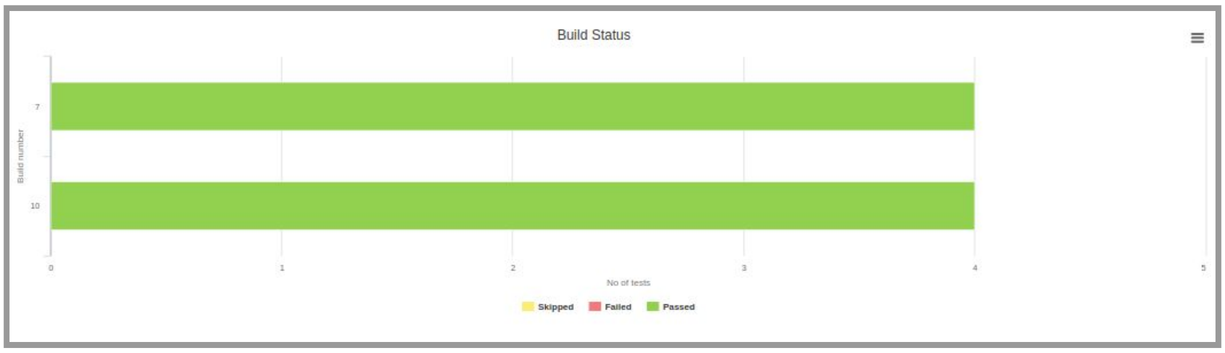
Showing 1 changed file with **1 addition** and **1 deletion**.

src/test/java/IntegracionJenkins/RelationalOperatorTest.java

```
@@ -21,7 +21,7 @@ public void isGreaterTest2() {
        @Test
        public void isLessTest1() {
                RelationalOperator tester = new RelationalOperator();
-               assertTrue(tester.isLess(4, 4));
+               assertTrue(tester.isLess(3, 4));
        }
```

Build - Test Results Analyzer

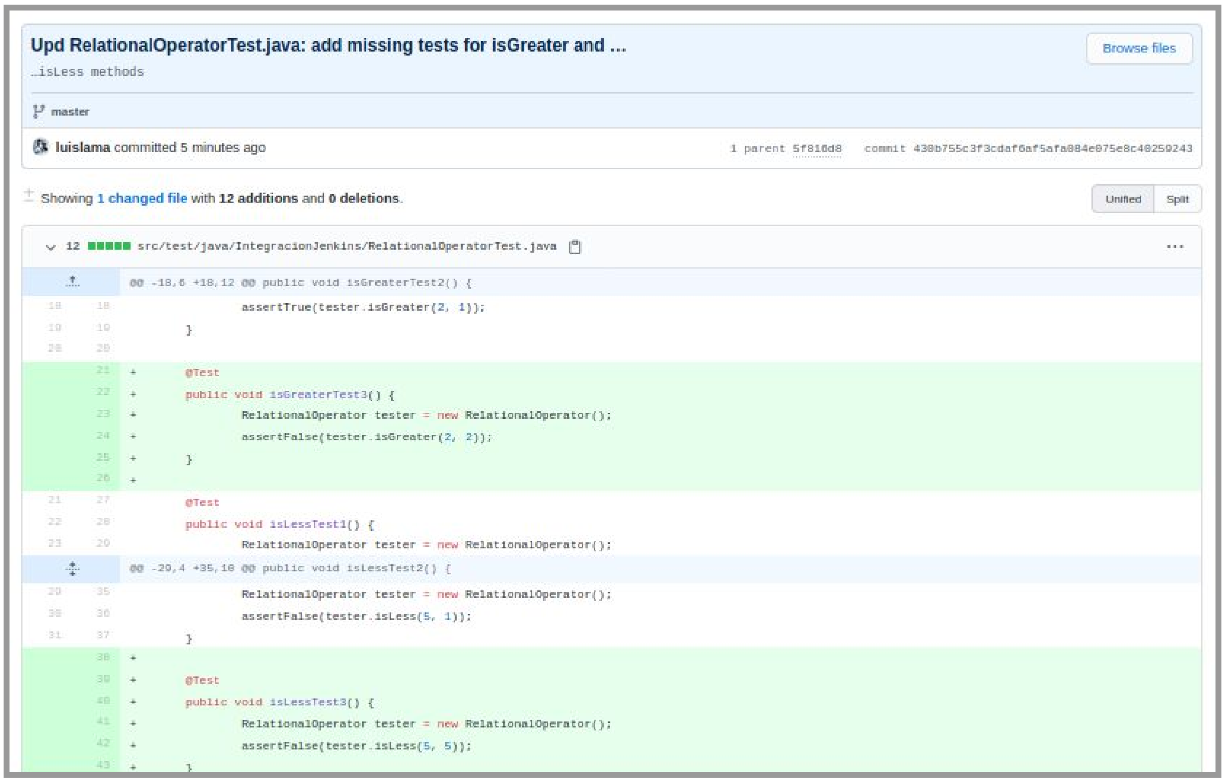




Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Add additional tests

Upload changes



Upd RelationalOperatorTest.java: add missing tests for isGreater and ...

`…isLess methods`

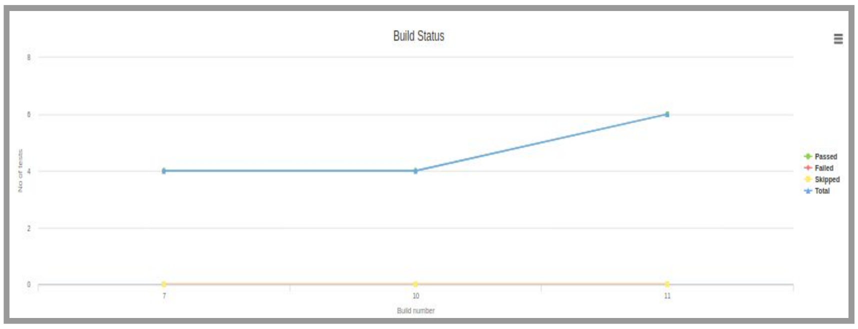
luislama committed 5 minutes ago

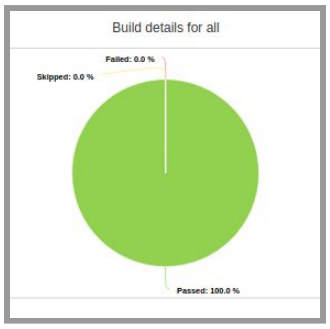
`1 parent 5f816d8    commit 430b755c3f3cdaf6af5afa884e075e8c40259243`

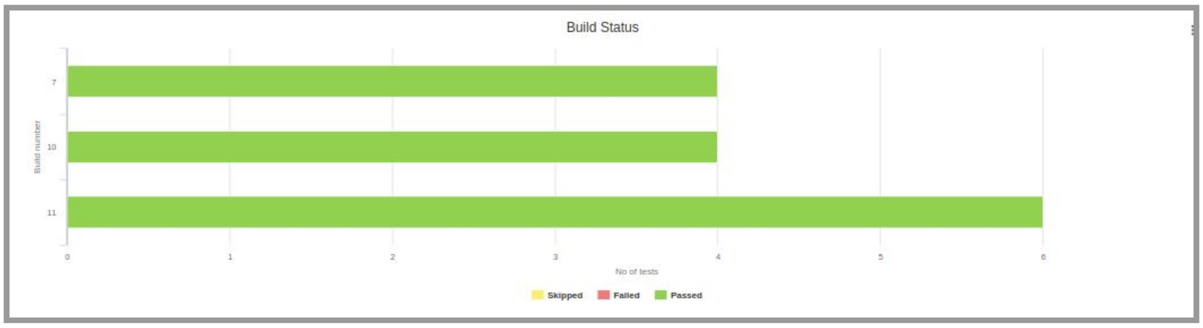
Showing **1 changed file** with **12 additions** and **0 deletions**.

```
12 src/test/java/IntegracionJenkins/RelationalOperatorTest.java
@@ -18,6 +18,12 @@ public void isGreaterTest2() {
                assertTrue(tester.isGreater(2, 1));
        }

+       @Test
+       public void isGreaterTest3() {
+               RelationalOperator tester = new RelationalOperator();
+               assertFalse(tester.isGreater(2, 2));
+       }
+
        @Test
        public void isLessTest1() {
                RelationalOperator tester = new RelationalOperator();
@@ -29,4 +35,10 @@ public void isLessTest2() {
                RelationalOperator tester = new RelationalOperator();
                assertFalse(tester.isLess(5, 1));
        }
+
+       @Test
+       public void isLessTest3() {
+               RelationalOperator tester = new RelationalOperator();
+               assertFalse(tester.isLess(5, 5));
+       }
```

Build - Test Results Analyzer







Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Conclusions

The use of continuous integration allow us to run builds and tests immediately after we commit the changes to the central repository, so we could know if the software has regressed using the tests written for it; and also if the build failed to build, we're notified immediately. In this workshop we have connected Jenkins (a FOSS CI server) to Github through a webhook that gets executed when a new commit is pushed to the main repository.
Gradle is a project management tool used to automate the building and testing of Java projects, and so, if the CI server has a integration with Gradle, most of the configuration is done automatically by the Gradle plugin and we don't have to setup a lot of stuff.
Ngrok is a free service that exposes a local server to the internet bypassing firewalls and NATs. This allows us to connect our Jenkins CI server to GitHub, the free service that hosts the repository and send a webhook notification to our build server.

Continuous integration helps us to save time when we're merging changes, since it is able to integrate this changes and run all the tests, letting us know if the changes introduced by the merge will not regress or fail to work when deployed into the production environment.

Continuous integration is recommended to all the projects that have teams with a lot of people working in parallel, in different functionalities, for the same product, and skipping this very important step in a project would will result in developers and team managers having to manually test each release and merge of the changes submitted, to make sure they're still working. These tasks can be easily automated using the aforementioned tool and would help the team find if a build will work faster, and isolate the cause of the problem quicker.

If we add regression testing to the CI reports, we will find immediately if a bug slipped before shipping and broke current functionality. Doing the inverse would result in a lot of bugs accumulated for a long time, without anyone knowing or noticing.

Finally, using CI can helps us with a way to do release versioning by helping us tag the binary release with the code it was used so if there's a bug detected on that version of the software, it is already built the way it was shipped and using tools like git bisect, isolate the build that introduced this bug and make the amendments.

Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Recommendations

For this workshop we started by introducing two errors, one in the isLess function and the other one in the tests for that function, this kind of error would not be detected by any CI, so with this we highlight one of the important parts of using CI: Making sure the tests are right.

CI can show us if there are conflicts, or if the build ran successfully after the committed changes. But if we have an error in code, and the test for that code has an error that make the test pass, this bug will pass CI without being noticed.

It would be interesting to test changes with other team members working in other branches supposed to merge at some point, to see the results in not committed but posible merge/rebase on the master branch.

References

[1] Amazon, «What is Continuous Integration?,» [En línea]. Available: https://aws.amazon.com/devops/continuous-integration/. [Último acceso: 13 Abril 2020].
Max, T. (2019). Using QF-Test in Continuous Integration Systems. [Figure 1]. Recovered from https://www.qfs.de/en/blog/article/2019/07/11/using-qf-test-in-continuous-integration-systems.html
Chebbi, C. (2018). Exploiting Git and Continuous Integration Servers. In Advanced Infrastructure Penetration Testing. [Figure 2]. Recovered from https://subscription.packtpub.com/book/networking_and_servers/9781788624480/6/ch06lvl1sec51/continuous-integration-with-github-and-jenkins

Members:
Luis Lama
Piero Ulloa
July 28th, 2020