

Escuela Superior Politécnica del Litoral

Software Engineering 2

Continuous Integration

Members:

Luis Lama

Piero Ulloa

July 28th, 2020

Guayaquil, Ecuador

Repository: [luislama/IntegracionContinuaTaller](https://github.com/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

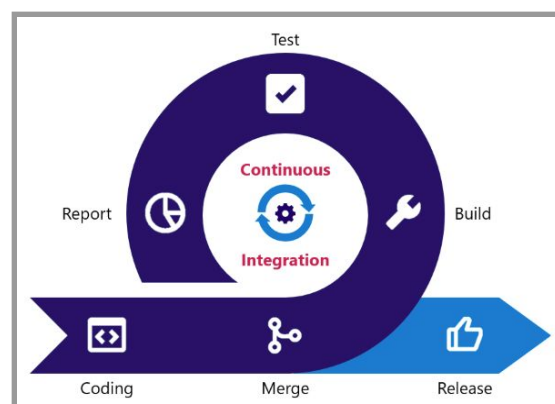


Figure 1 How continuous integration works

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.

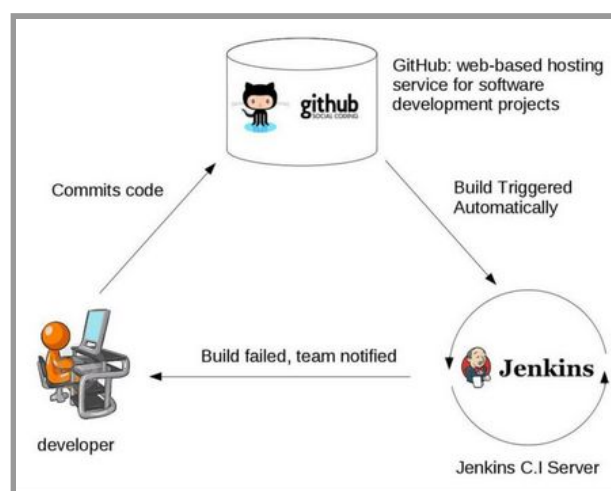


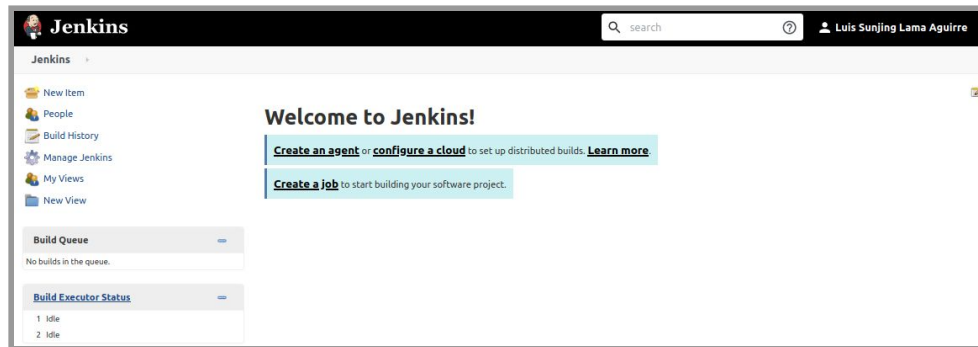
Figure 2 Continuous integration with Jenkins and Github

Members:
Luis Lama
Piero Ulloa
July 28th, 2020

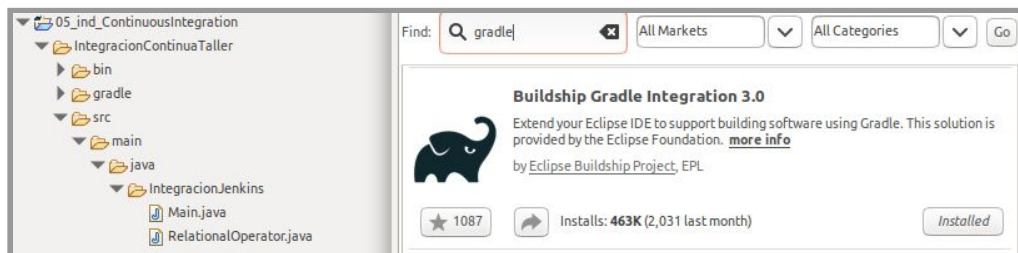
Development

Installing

Jenkins



Gradle



Ngrok

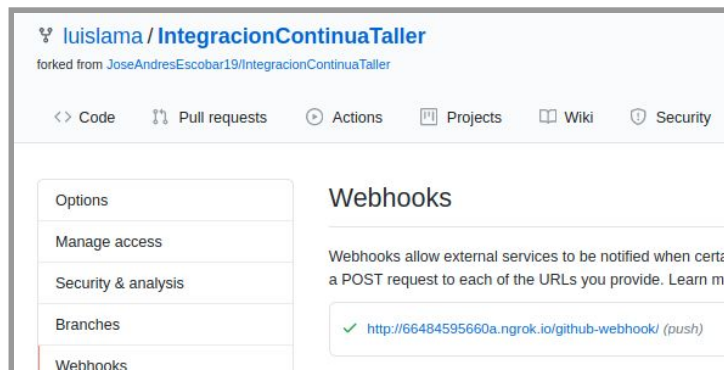
```
ngrok by @inconshreveable (Ctrl+C to quit)

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
                    https://66484595660a.ngrok.io -> http://localhost:8080

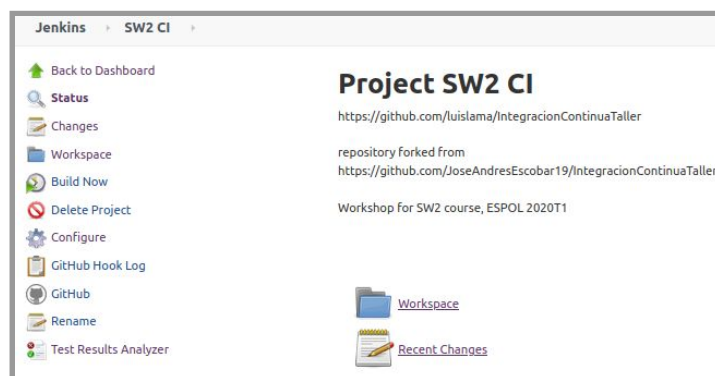
Connections
  ttl    opn    rt1    rt5    p50    p90
   17     0    0.00   0.01   0.66   8.70

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

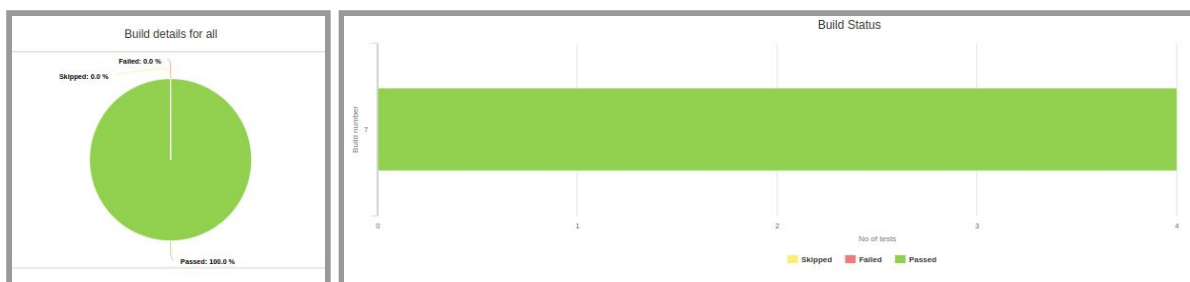
Ngrok - Github Webhook



Jenkins - Project Github



Build - Test Results Analyzer



Workshop

Correct the error in isLess method

Upload changes

Fix RelationalOperator.java: correct isLess method [Browse files](#)

master

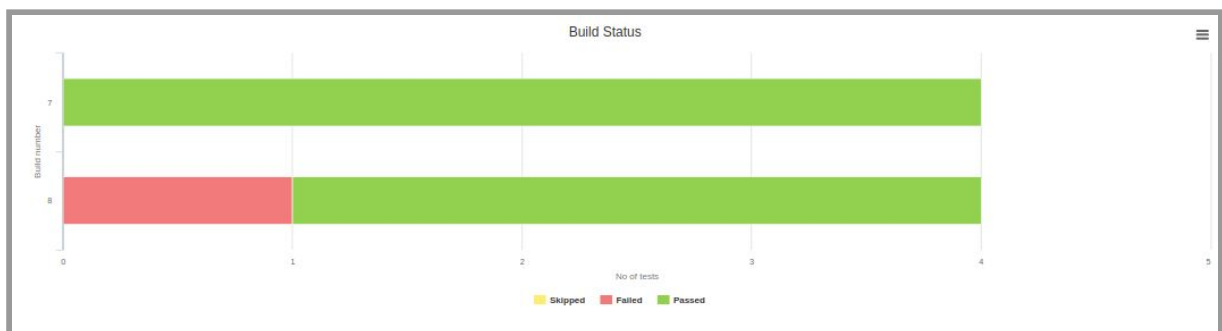
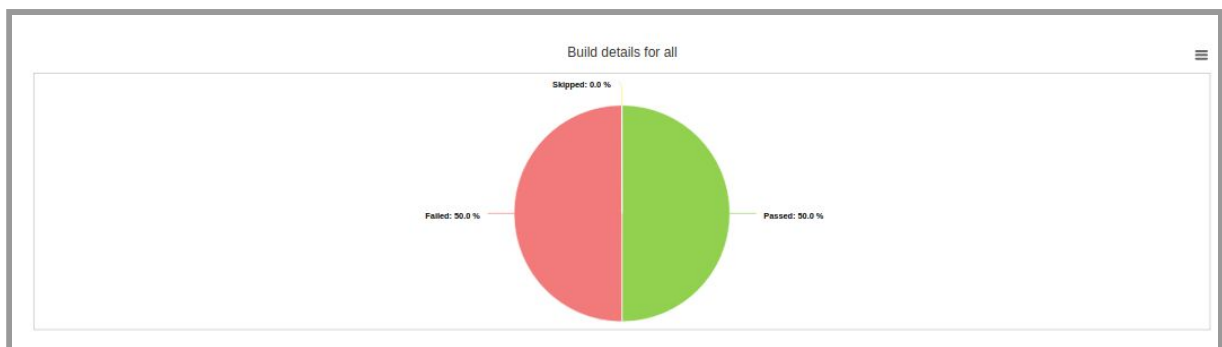
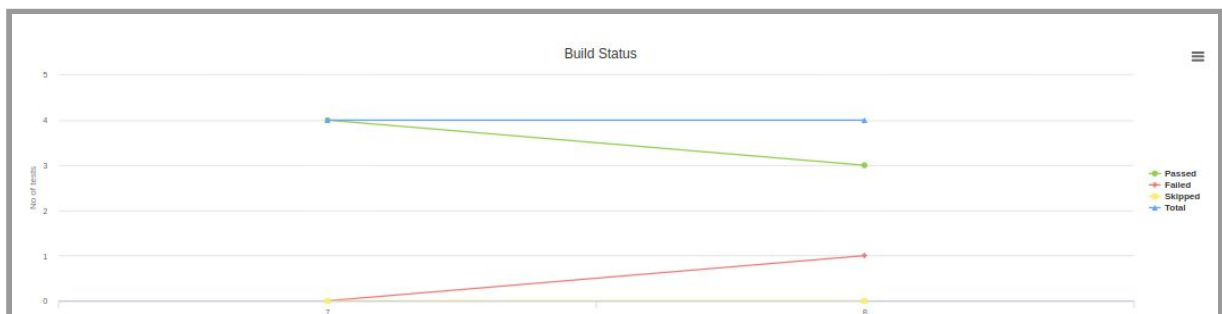
luislama committed 11 minutes ago 1 parent 98e9888 commit 432b8b44ab846d4c49dacdb34f409aab9c99e8c

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

```
src/main/java/IntegracionJenkins/RelationalOperator.java
@@ -19,7 +19,7 @@ public boolean isGreater(int num1, int num2) {
19 19     * @return true if num1 is less than num2, false otherwise
20 20     */
21 21     public boolean isLess(int num1, int num2) {
22 22 -     return num1 <= num2;
22 22 +     return num1 < num2;
23 23 }
24 24
```

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

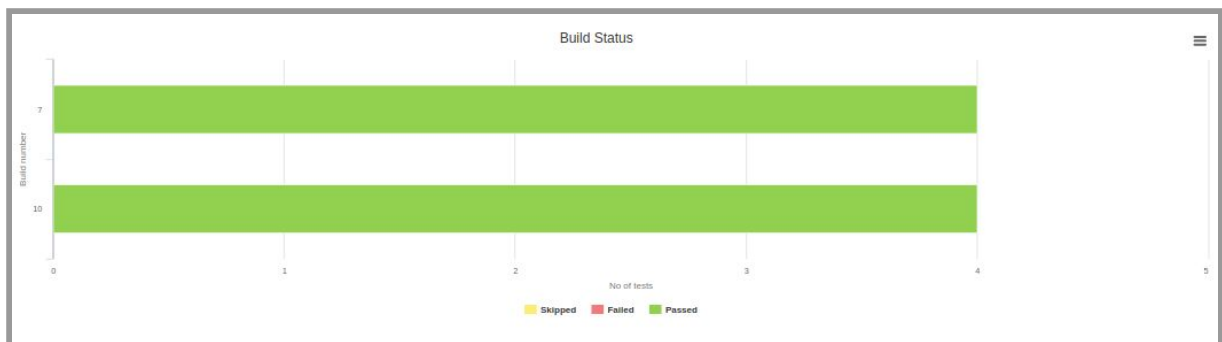
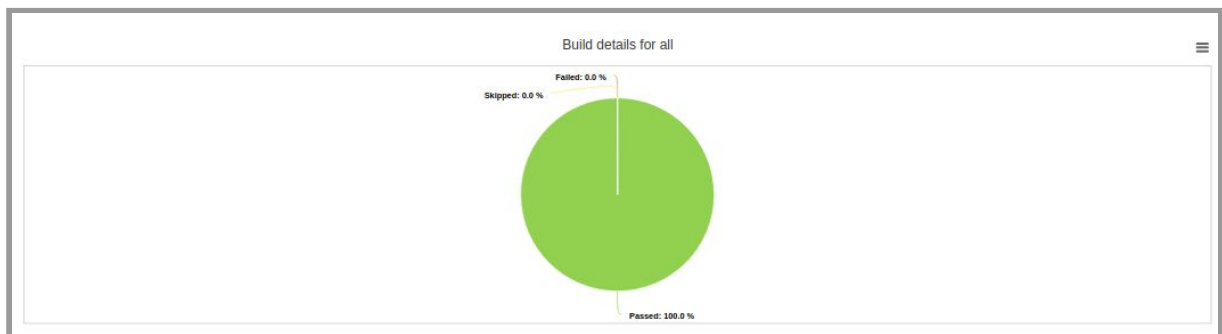
1 parent 432b8b4 commit 5f816d8e1f8e98fb52da88929061f2dae79f55d9

Showing 1 changed file with 1 addition and 1 deletion.

src/test/java/IntegracionJenkins/RelationalOperatorTest.java

```
@@ -21,7 +21,7 @@ public void isGreaterTest2() {
21 21     @Test
22 22     public void isLessTest1() {
23 23         RelationalOperator tester = new RelationalOperator();
24 -    assertTrue(tester.isLess(4, 4));
24 +    assertTrue(tester.isLess(3, 4));
25 25     }
```

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

master

luislama committed 5 minutes ago 1 parent 5f816d8 commit 438b755c3f3cdf0af5afa884e075e8c40250243

Showing 1 changed file with 12 additions and 0 deletions.

Unified Split

12 src/test/java/IntegracionJenkins/RelationalOperatorTest.java

```
@@ -18,6 +18,12 @@ public void isGreaterTest2() {
    18      assertTrue(tester.isGreater(2, 1));
    19  }
    20
    21  + @Test
    22  + public void isGreaterTest3() {
    23  +     RelationalOperator tester = new RelationalOperator();
    24  +     assertFalse(tester.isGreater(2, 2));
    25  + }
    26  +
    27
    28  @Test
    29  public void isLessTest1() {
    30      RelationalOperator tester = new RelationalOperator();
    31
    32  @@ -29,4 +35,18 @@ public void isLessTest2() {
    33      RelationalOperator tester = new RelationalOperator();
    34      assertFalse(tester.isLess(5, 1));
    35  }
    36
    37  + @Test
    38  + public void isLessTest3() {
    39  +     RelationalOperator tester = new RelationalOperator();
    40  +     assertFalse(tester.isLess(5, 5));
    41  + }
    42  +
    43  + }
```

Build - Test Results Analyzer



Members:
Luis Lama
Piero Ulloa
July 28th, 2020

Conclusions

Using continuous integration allow us to run builds and tests immediately after we commit the changes.

We can run jenkins locally using github webhooks.

Gradle allow us to automate the testing of our apps.

Ngrok expose a local server to the internet so the server that hosts the repository can send the changes in code pushed by our team members.

Continuous integration saves us time in the merging process and let us record the build and test results.

Continuous integration is recommended to all the projects that has teams with many people working in parallel, in different functionalities, for the same product.

Not doing the continuous integration of a project would will result in developers and team managers working on the merge and integration of the changes submitted, these could be difficult and it will take some time.

Not doing the continuous integration of a project would will result in bugs accumulated for a long time, without anyone knowing.

Not doing the continuous integration of a project would will result in a harder process of software updates.

Recommendations

For this workshop we started introducing two errors, one in the isLess function and other in one of the tests for that function, this kind of error is invisible to the CI and makes it useless, but with this we wanted to elevate the importance of making good tests and make them right.

CI can show us if there are conflicts, or if the build run 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, CI check will be passed with hidden errors.

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/ch06lv11sec51/continuous-integration-with-github-and-jenkins

Members:

Luis Lama

Piero Ulloa

July 28th, 2020