Software Engineering II

# Continuous Integration Workshop

GROUP 1:
Cabezas Garrido, Josué Alberto
Sánchez Jarrín, Daniel Roberto
Villacís Rivadeneira, Ricardo Alejandro

# Tabla de contenido

# Introduction

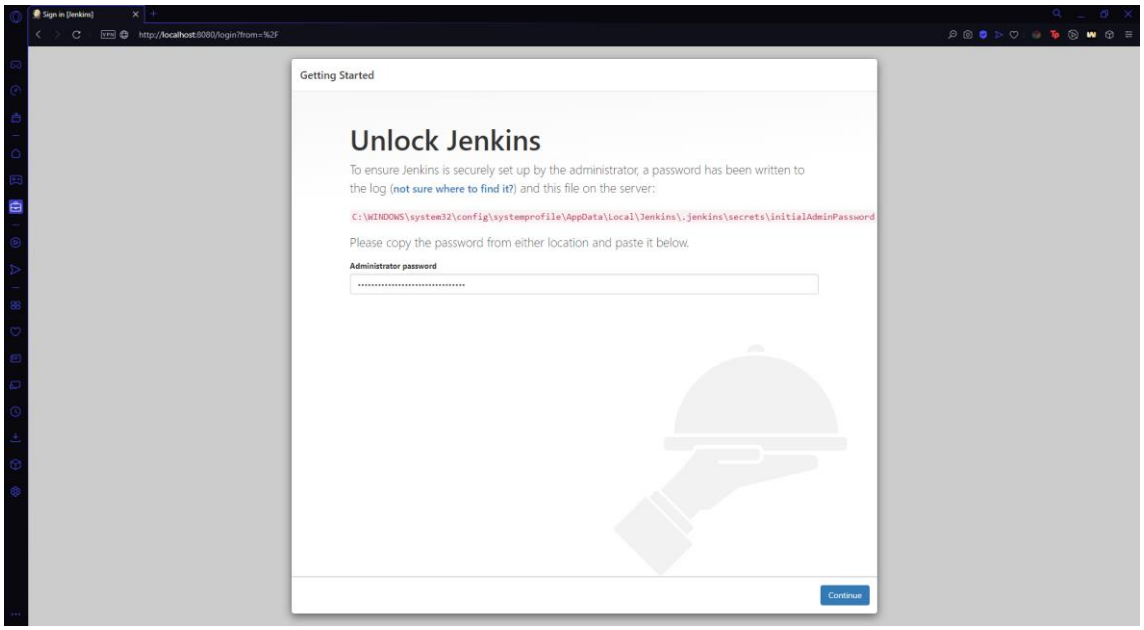
Software development is full of best practices that we regularly talk about, but rarely do. One of these cases is to have an automated system to assemble and test executable versions of our software, so that the development team can build and test the software they are working on several times a day. This concept is known as “continuous integration”.

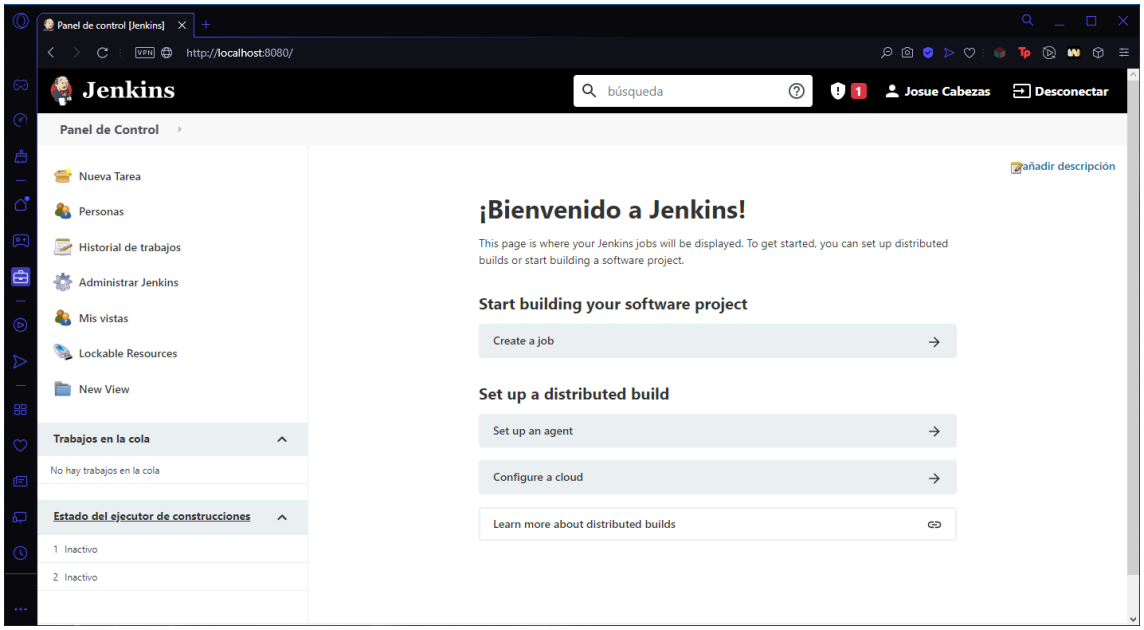
The benefits of continuous integration are "solve problems quickly." Since software is often embedded, when a bug is found it is typically not necessary to go back far to discover where the bug was introduced. In comparison, when a team does not follow a continuous integration strategy, the periods between integration are long and the code base is very different between each integration, so when errors are found, much more code has to be reviewed, which requires more time. and effort.

# Development

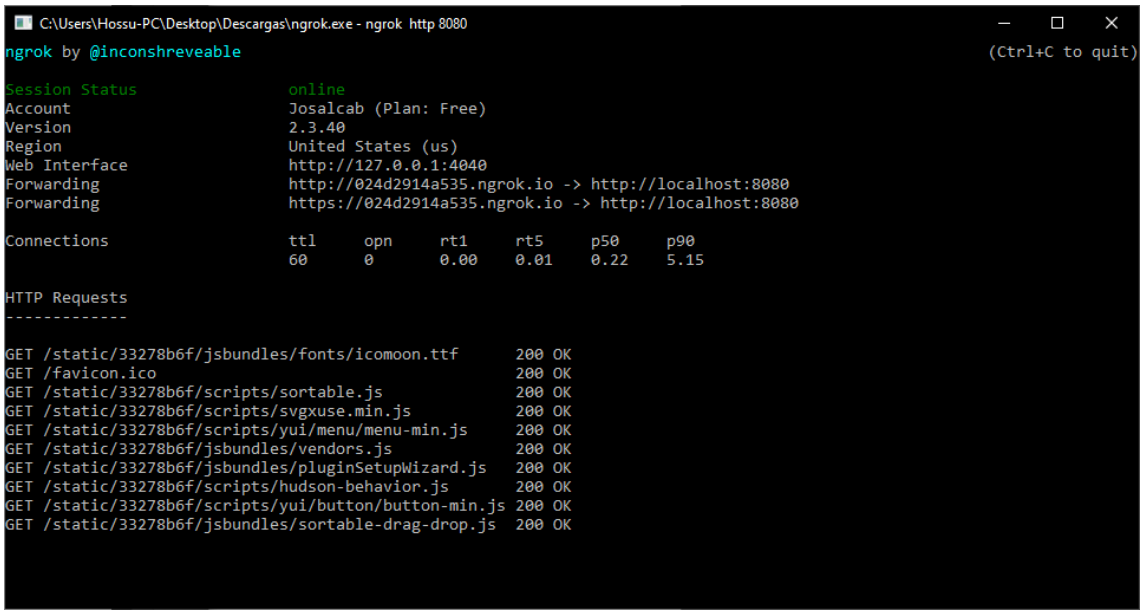
## Step 1: Install Requirements

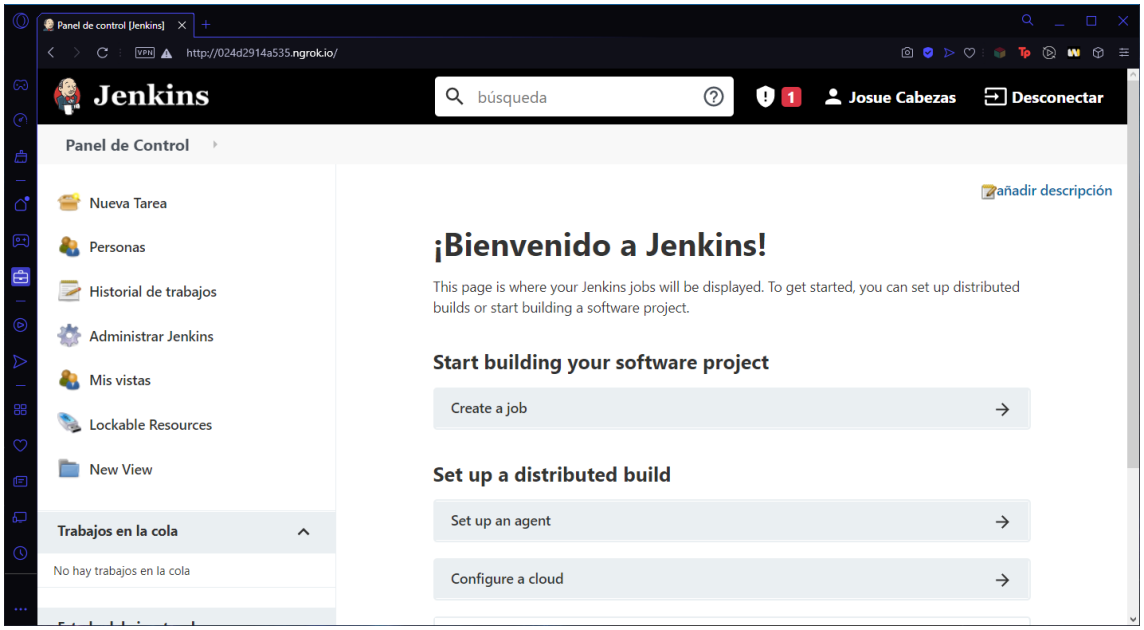
### *1. Install Jenkins*





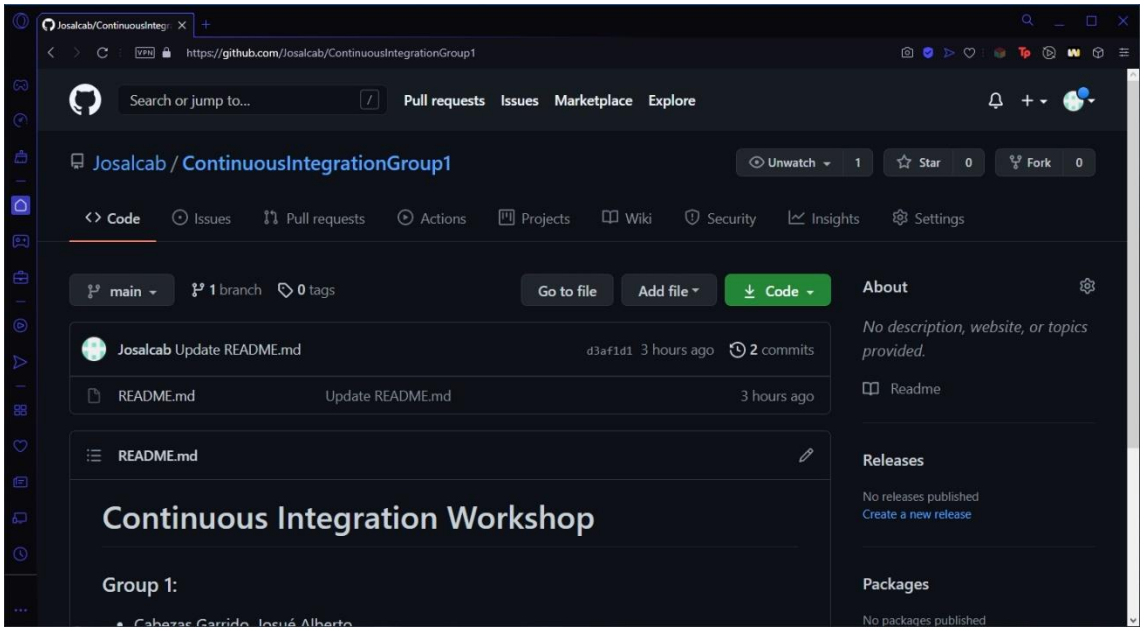

## 2. *Install ngrok*



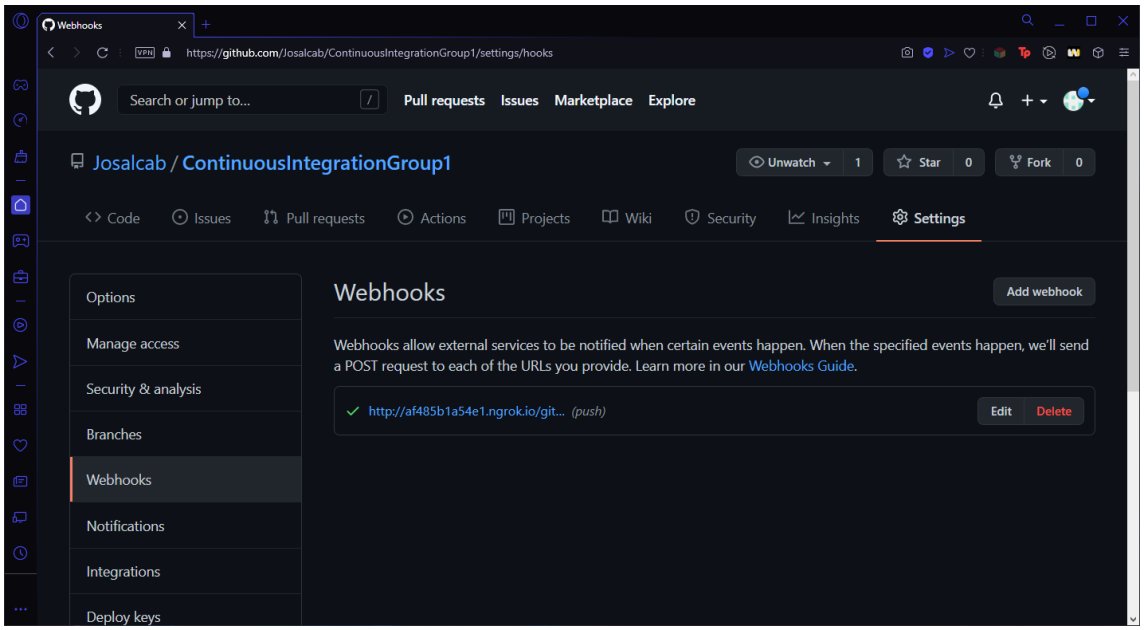



## Step 2: Create a Repository

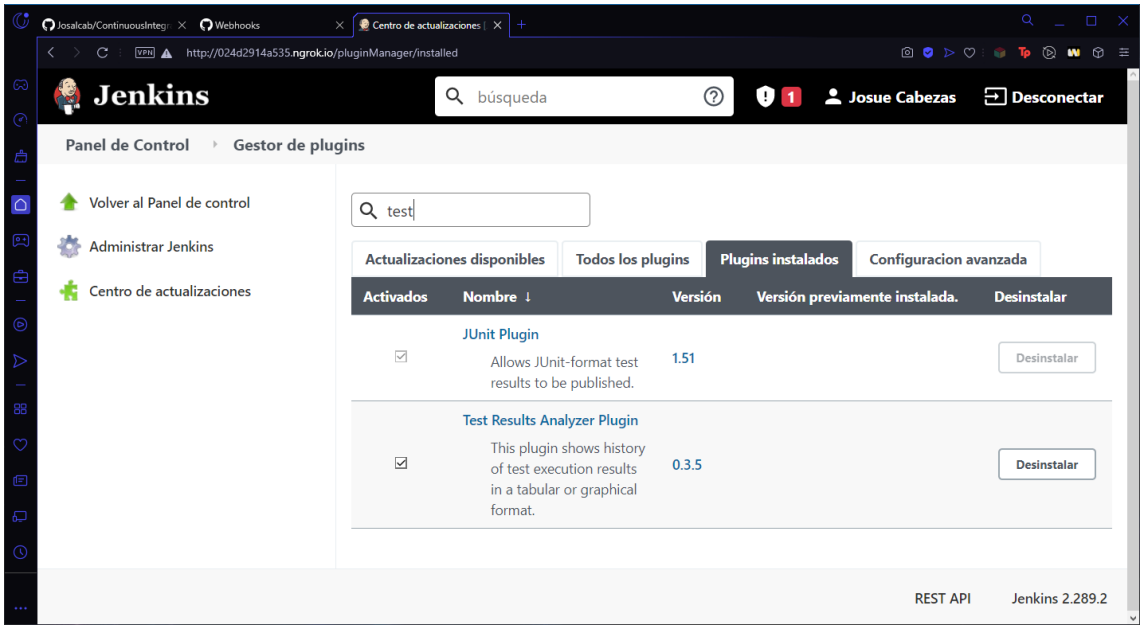
### 1. *Create a Github repository*



## 2. *Set up your Jenkins connection*




## Step 3: Prepare Jenkins

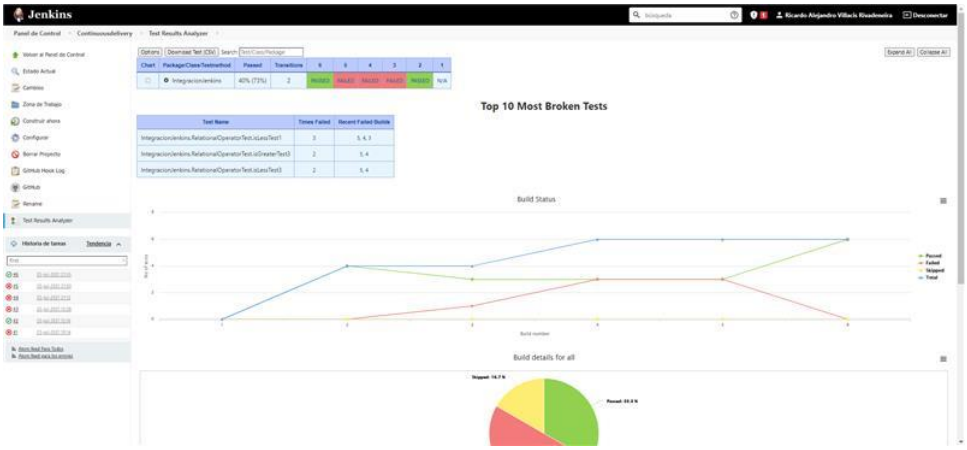


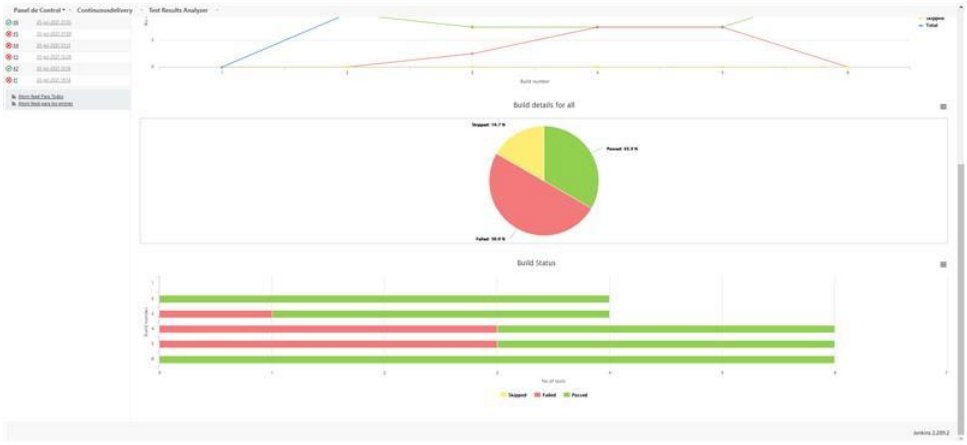
## Step 4: Check communication between Github and Jenkins

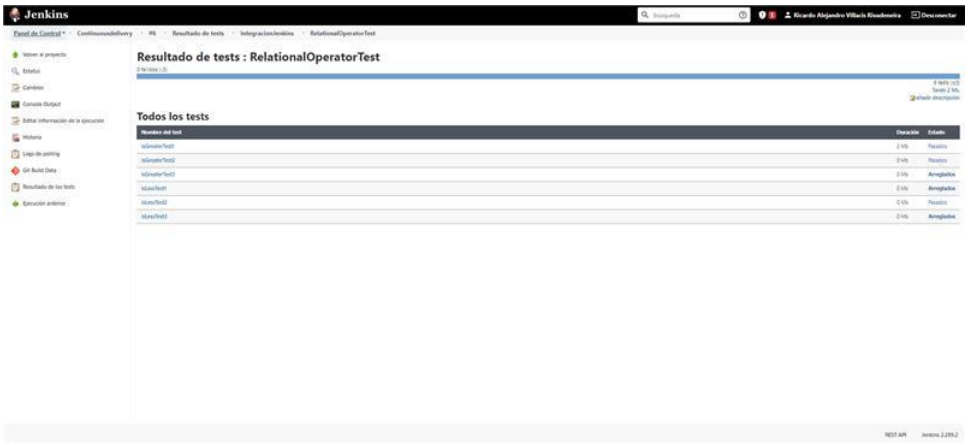

## Development

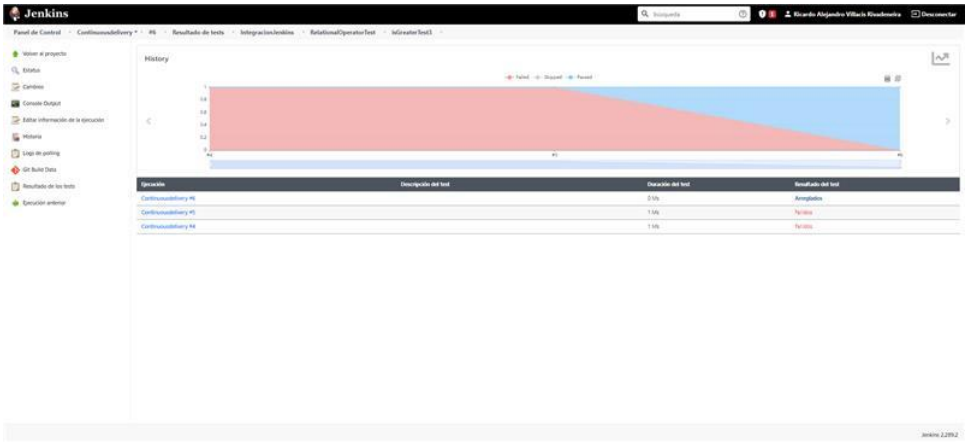
Main.java | RelationalOperator.java | RelationalOperatorTest.java

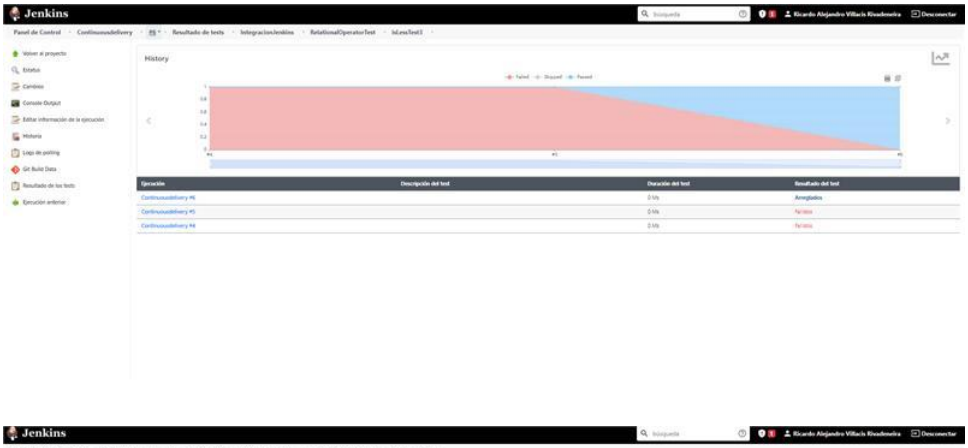
```java
package IntegracionJenkins;

import static org.junit.Assert.*;

import org.junit.Test;

public class RelationalOperatorTest {

    @Test
    public void isGreaterTest1() {
        RelationalOperator tester = new RelationalOperator();
        assertFalse(tester.isGreater(2, 3));
    }

    @Test
    public void isGreaterTest2() {
        RelationalOperator tester = new RelationalOperator();
        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();
        assertFalse(tester.isLess(4, 4));
    }

    @Test
    public void isLessTest2() {
        RelationalOperator tester = new RelationalOperator();
        assertFalse(tester.isLess(5, 1));
    }

    @Test
    public void isLessTest3() {
        RelationalOperator tester = new RelationalOperator();
        assertTrue(tester.isLess(5, 7));
    }

}

```

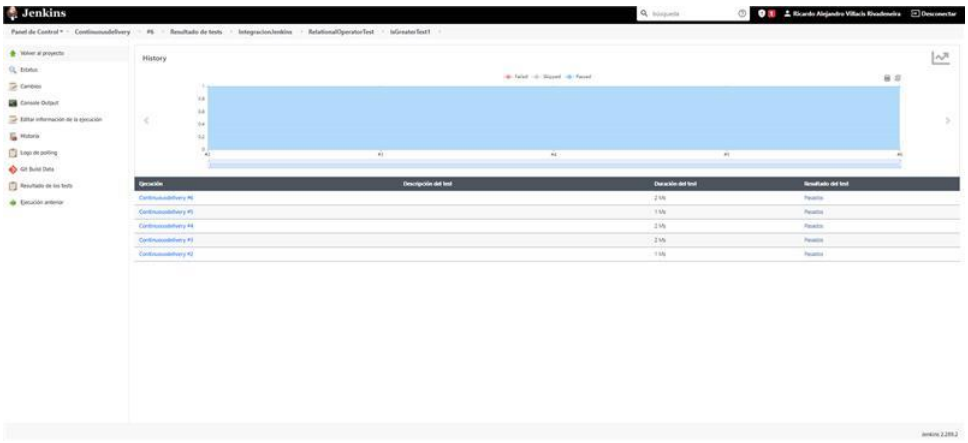

Jenkins
Top 10 Most Broken Tests
Build Status
Build details for all

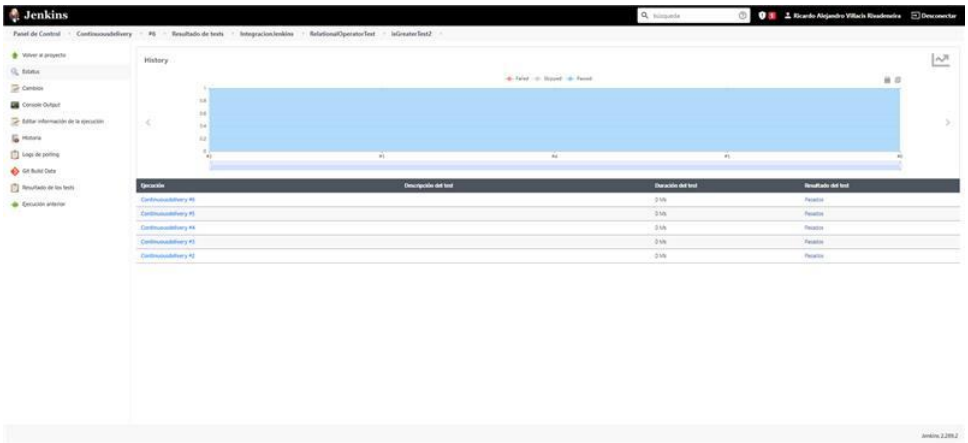

Build details for all
Build Status

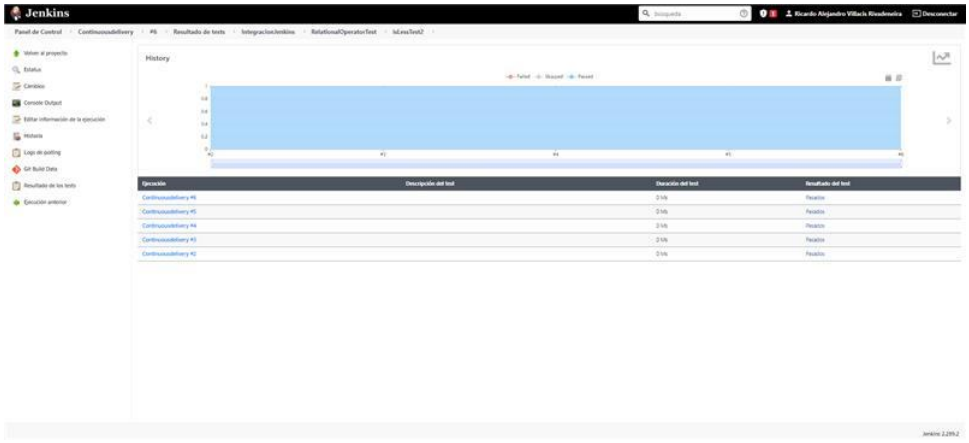

Jenkins
Resultado de tests : RelationalOperatorTest
Todos los tests


Jenkins
Panel de Control · ContinuousDelivery · #6 · Resultado de tests · Integracion.Jenkins · RelationalOperatorTest · isGreaterTest1
Volver al proyecto
Estatus
Cambios
Console Output
Editar información de la ejecución
Historia
Logs de polling
Git Build Data
Resultado de los tests
Ejecución anterior
History
Failed
Skipped
Passed
Ejecución
Descripción del test
Duración del test
Resultado del test
ContinuousDelivery #6
0 Ms
Arreglados
ContinuousDelivery #5
1 Ms
Fallidos
ContinuousDelivery #4
1 Ms
Fallidos


Jenkins
Panel de Control · ContinuousDelivery · #5 · Resultado de tests · Integracion.Jenkins · RelationalOperatorTest · isLessTest1
Volver al proyecto
Estatus
Cambios
Console Output
Editar información de la ejecución
Historia
Logs de polling
Git Build Data
Resultado de los tests
Ejecución anterior
History
Failed
Skipped
Passed
Ejecución
Descripción del test
Duración del test
Resultado del test
ContinuousDelivery #6
0 Ms
Arreglados
ContinuousDelivery #5
0 Ms
Fallidos
ContinuousDelivery #4
0 Ms
Fallidos


Jenkins
Panel de Control · ContinuousDelivery · #6 · Resultado de tests · Integracion.Jenkins · RelationalOperatorTest · isGreaterTest1
Volver al proyecto
Estatus
Cambios
Console Output
Editar información de la ejecución
Historia
Logs de polling
Git Build Data
Resultado de los tests
Ejecución anterior
History
Failed
Skipped
Passed
Ejecución
Descripción del test
Duración del test
Resultado del test
ContinuousDelivery #6
2 Ms
Pasados
ContinuousDelivery #5
1 Ms
Pasados
ContinuousDelivery #4
2 Ms
Pasados
ContinuousDelivery #3
2 Ms
Pasados
ContinuousDelivery #2
1 Ms
Pasados


Jenkins
Panel de Control · ContinuousDelivery · #6 · Resultado de tests · Integracion.Jenkins · RelationalOperatorTest · isGreaterTest2
Volver al proyecto
Estatus
Cambios
Console Output
Editar información de la ejecución
Historia
Logs de polling
Git Build Data
Resultado de los tests
Ejecución anterior
History
Failed
Skipped
Passed
Ejecución
Descripción del test
Duración del test
Resultado del test
ContinuousDelivery #6
0 Ms
Pasados
ContinuousDelivery #5
0 Ms
Pasados
ContinuousDelivery #4
0 Ms
Pasados
ContinuousDelivery #3
0 Ms
Pasados
ContinuousDelivery #2
0 Ms
Pasados





## Repository

https://github.com/Ricavill/-ContinuousIntegrationGroup1

## Conclusions

In this workshop Analize the process of continuous integration in collaboration with jenkins and github. With continuous integration we can analyze code and run test faster than other methods.

Making builds readily available to stakeholders and testers can reduce the amount of rework necessary when rebuilding a feature that doesn't meet requirements. Additionally, early testing reduces the chances that defects survive until deployment. Finding errors earlier can reduce the amount of work necessary to resolve them.
All programmers should start the day by updating the project from the repository. That way, they will all stay up to date.

## Recommendations

Make sure to be aware that jenkins still uses master in the linking process of github and jenkins.

Continuous integration should occur frequently enough that no intervening window remains between commit and build, and such that no errors can arise without developers noticing them and correcting them immediately.

## Referencias


[1] P. Galván, «Integración Continua,» SG.com, [En línea]. Available: https://sg.com.mx/revista/54/integraci-n-continua. [Último acceso: 22 Julio 2021].

[2] P. Nguyen, «How to Install Jenkins on Windows,» DZone, 14 Julio 2018. [En línea]. Available: https://dzone.com/articles/how-to-install-jenkins-on-windows. [Último acceso: 22 Julio 2021].

[3] ngrok, «Setup & Installation,» ngrok, [En línea]. Available: https://dashboard.ngrok.com/get-started/setup. [Último acceso: 22 Julio 2021].