

EJERCICIO PRÁCTICO DE JENKINS

Instalar y configurar Jenkins

Actividad 1

Tomando como base el repositorio <https://github.com/vero0311/calculator>

1. Crear un job que haga lo siguiente:

1.1. Descargue el proyecto de la rama master

1.2. Compile el proyecto

1.3. Genere el archivo .jar del proyecto

2. Crear un segundo job que haga lo siguiente:

2.1 Tomar la build creada en el anterior job (En este deben de estar ubicados en la carpeta donde se generó la build)

2.2 ejecutar el comando `java-jar NOMBRE_DEL_ARCHIVO.jar` usando la build generada.

Para cada actividad, crear un documento PDF donde van a adjuntar los pantallazos en los cuales de forma clara se aprecien los pasos realizados para crear los jobs y el log de los resultados finales (console output).

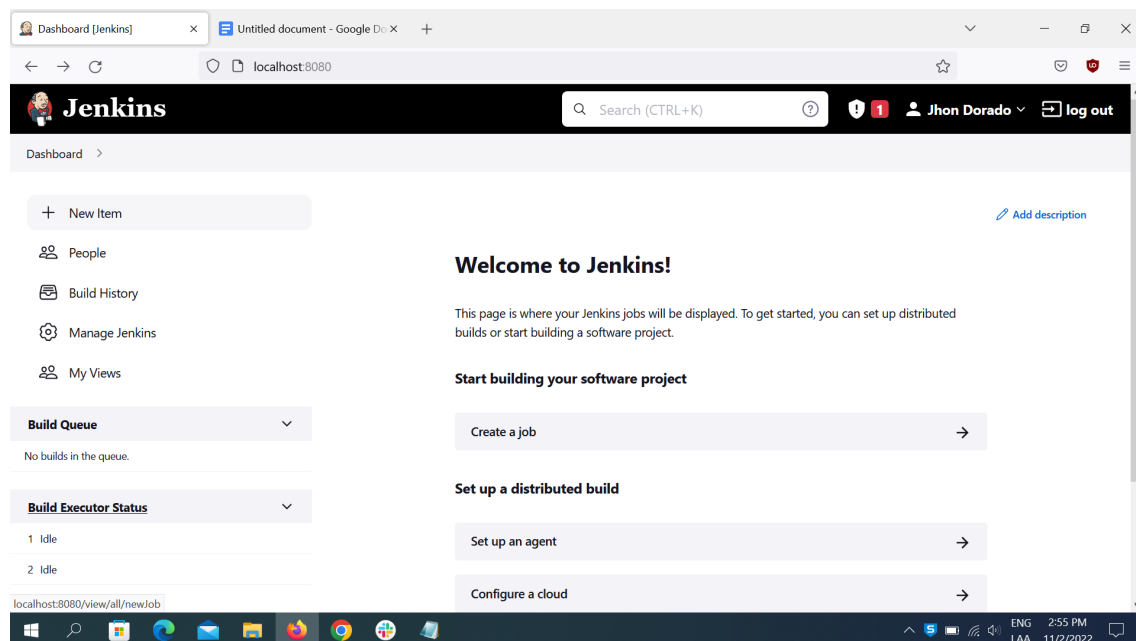
En Github crear un repositorio llamado CI-CD, donde van a subir los dos archivos **en formato PDF**

Mandar un correo electrónico donde indique el link del repositorio a veronica.restrepo@globant.com con el siguiente formato [CI][Nombre_Apellido] - Entregable Jenkins

Solución

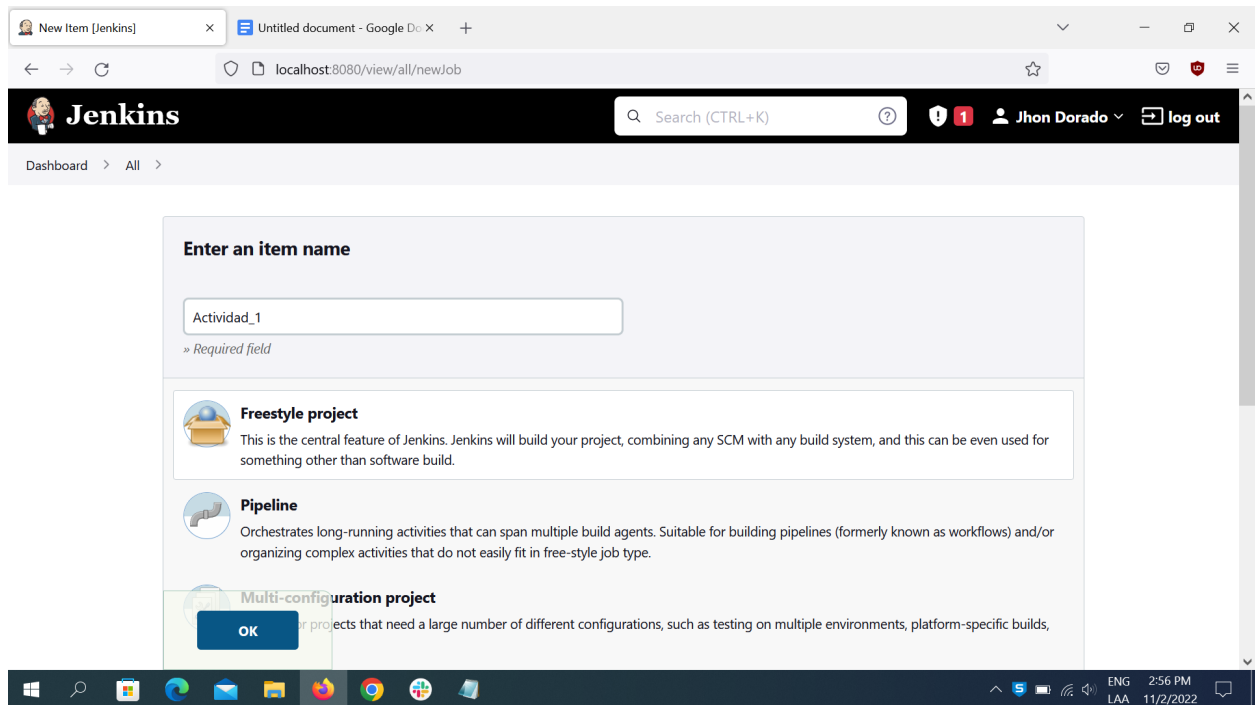
Actividad 1

Paso 1: Hacemos click en new item



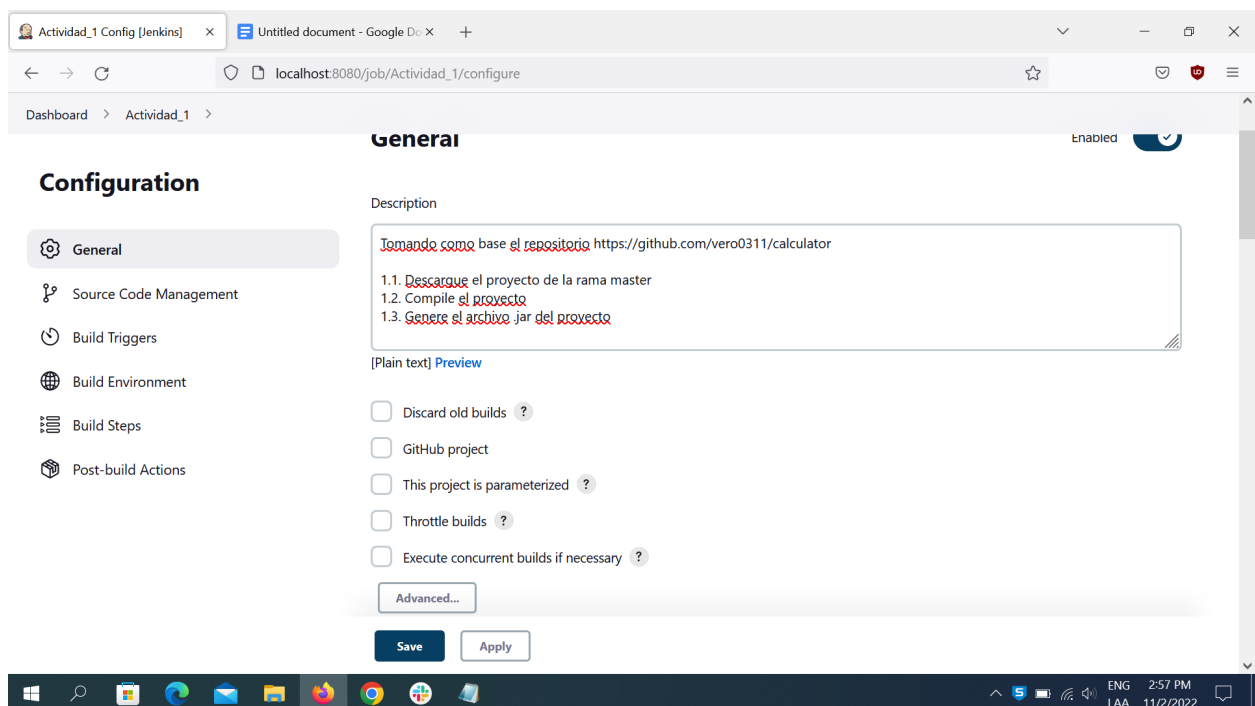
Paso 2:

Escribimos el nombre del jpb, luego elegimos Freestyle project y luego ok



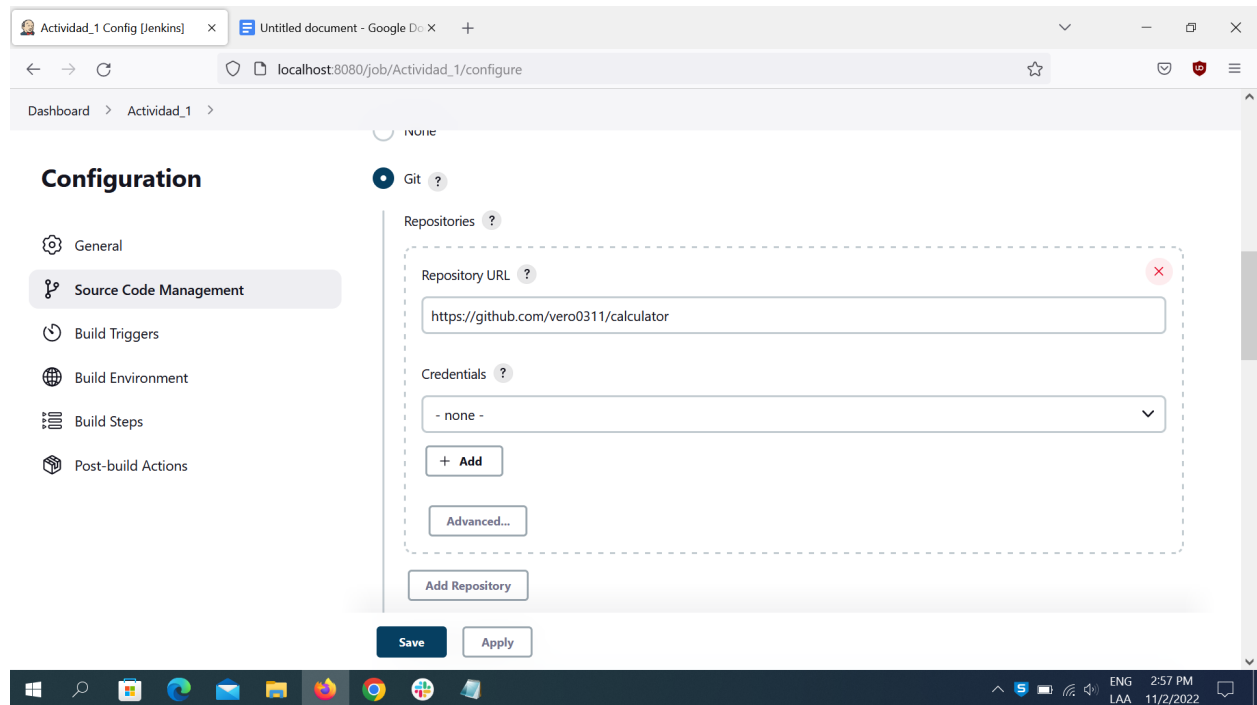
Paso 3:

Escribimos la descripción del job.



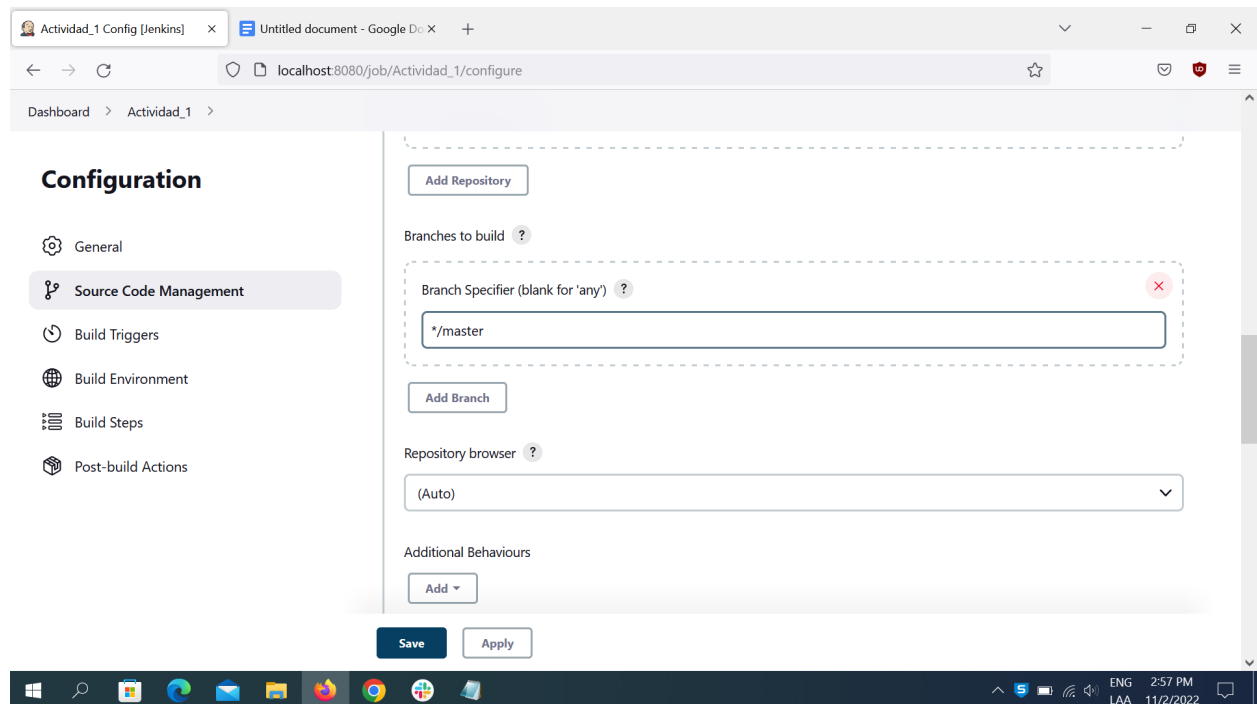
Paso 5:

En source code management seleccionamos git y escribimos la dirección del repositorio.



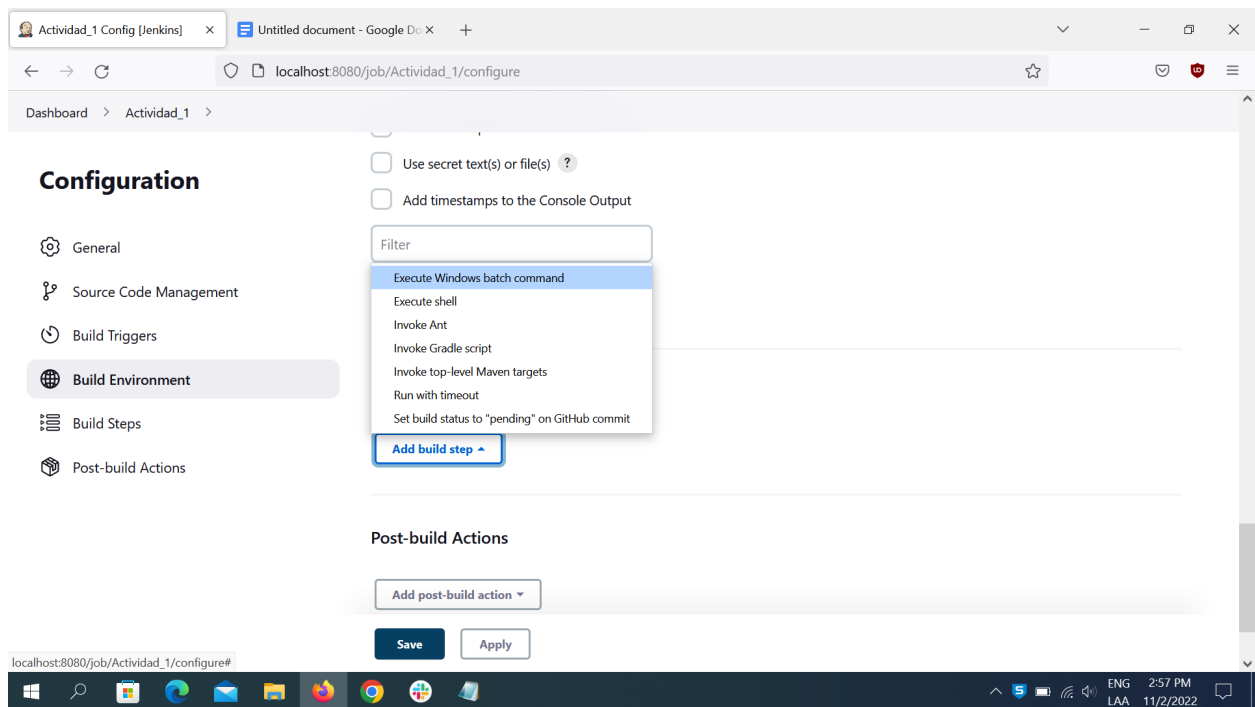
Paso 6:

Elegimos la rama del proyecto en la que trabajaremos



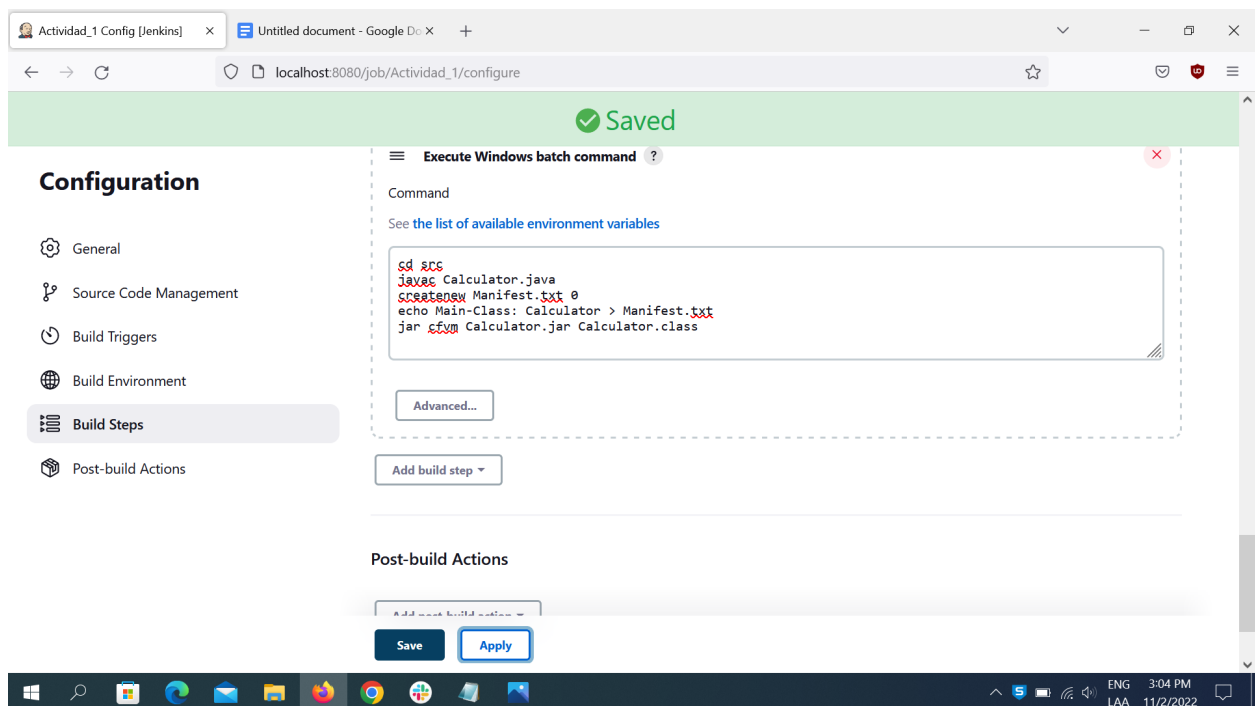
Paso 6:

En Build Steps build actions elegimos la consola de windows



Paso 7:

Escribimos el código que queremos ejecutar



Paso 8:

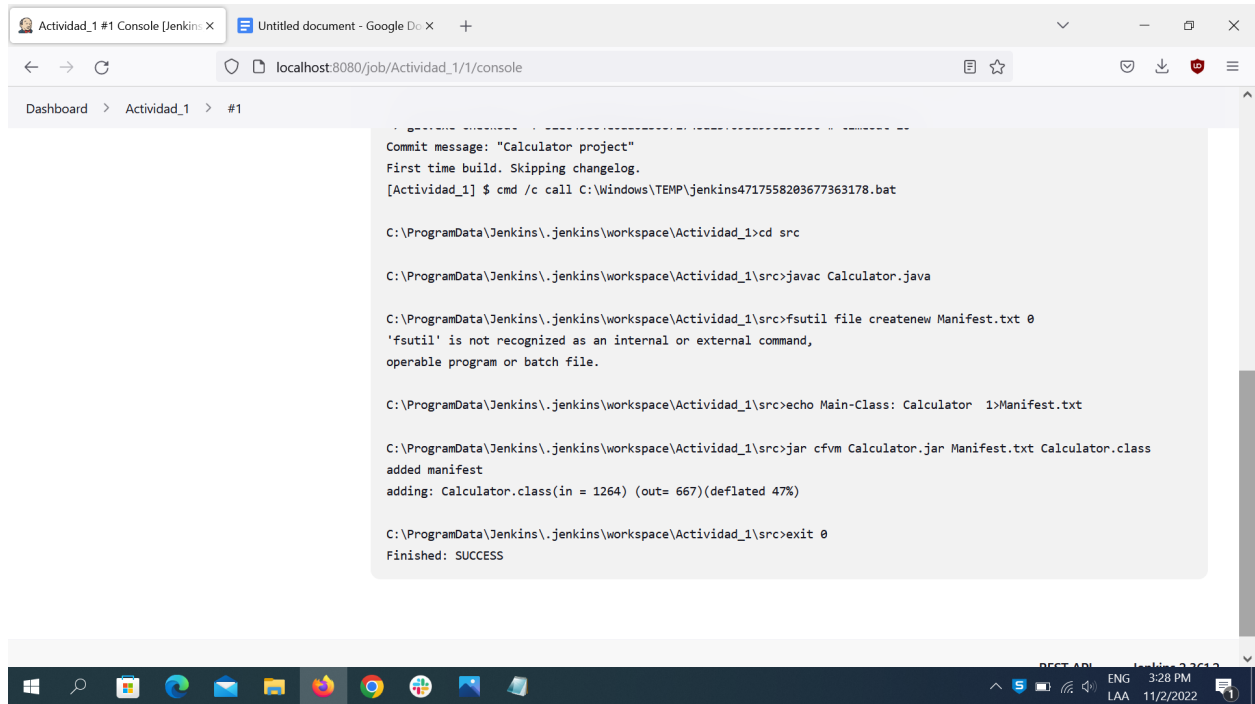
Hacemos click en la build que se generó en la parte inferior izquierda, este tiene un check verde para indicar que fue exitosa

The screenshot shows the Jenkins web interface for a job named 'Actividad_1'. The left sidebar contains navigation links: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main area displays the job's description, which includes instructions for cloning a repository, downloading the project, compiling it, and generating a JAR file. Below the description is a 'Permalinks' section. The 'Build History' section shows a list of builds, with build #2 highlighted. Build #2 has a green checkmark icon, indicating it was successful. The bottom of the page shows the Jenkins version (2.361.2) and the REST API endpoint.

Paso 9:

Hacemos click en la opción console output para ver el resultado de los comandos que ejecutamos

The screenshot shows the Jenkins console output for build #2. The left sidebar contains navigation links: Back to Project, Status, Changes, Console Output, View as plain text, Edit Build Information, Delete build '#2', and Git Build Data. The main area displays the console output, which shows the results of the commands executed during the build. The output starts with 'Started by user Jhon Dorado' and 'Running as SYSTEM'. It then shows the workspace path, the recommended git tool, and the cloning of the repository. The output continues with the execution of git commands to initialize the repository, fetch upstream changes, and checkout the specific revision. The final commit message is 'Calculator project'.



The screenshot shows a web browser window with the address bar at `localhost:8080/job/Actividad_1/1/console`. The page displays the console output of a Jenkins build. The output text is as follows:

```
Commit message: "Calculator project"
First time build. Skipping changelog.
[Actividad_1] $ cmd /c call C:\Windows\TEMP\jenkins4717558203677363178.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Actividad_1>cd src

C:\ProgramData\Jenkins\.jenkins\workspace\Actividad_1\src>javac Calculator.java

C:\ProgramData\Jenkins\.jenkins\workspace\Actividad_1\src>fsutil file createnew Manifest.txt 0
'fsutil' is not recognized as an internal or external command,
operable program or batch file.

C:\ProgramData\Jenkins\.jenkins\workspace\Actividad_1\src>echo Main-Class: Calculator 1>Manifest.txt

C:\ProgramData\Jenkins\.jenkins\workspace\Actividad_1\src>jar cfvm Calculator.jar Manifest.txt Calculator.class
added manifest
adding: Calculator.class(in = 1264) (out= 667)(deflated 47%)

C:\ProgramData\Jenkins\.jenkins\workspace\Actividad_1\src>exit 0
Finished: SUCCESS
```

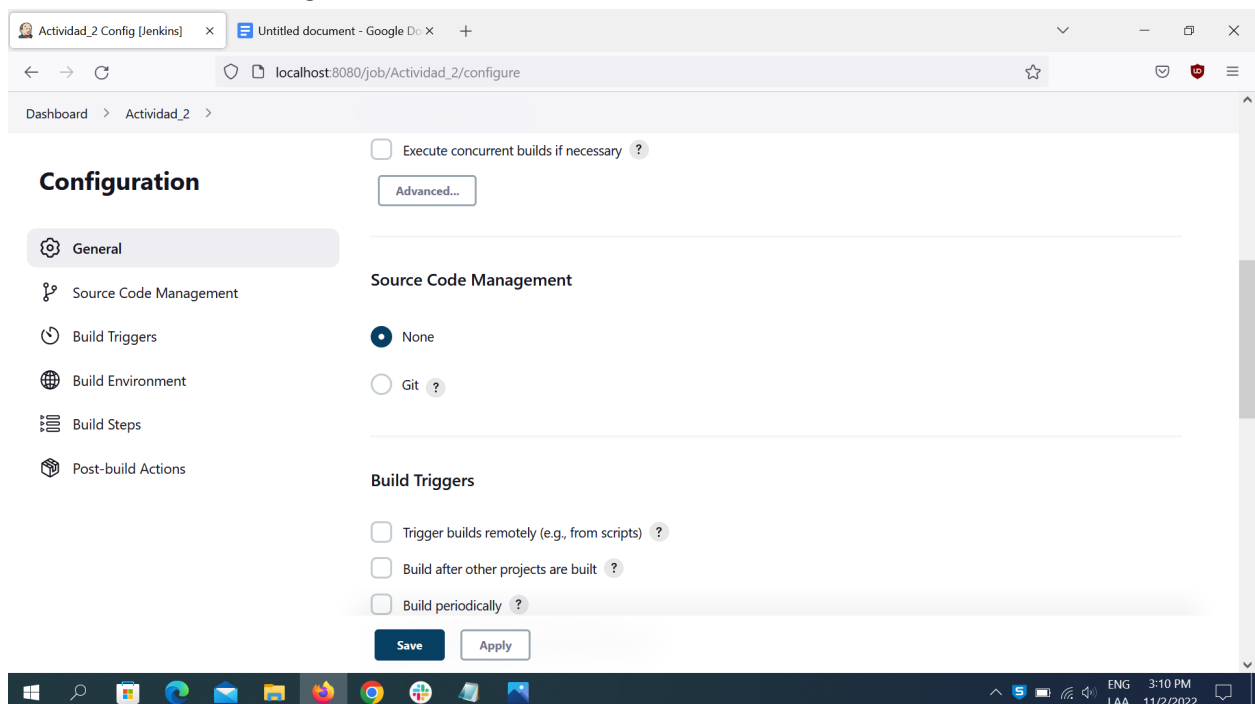
Actividad 2

Paso 1:

Repetimos los pasos 1 - 4 de la actividad 1

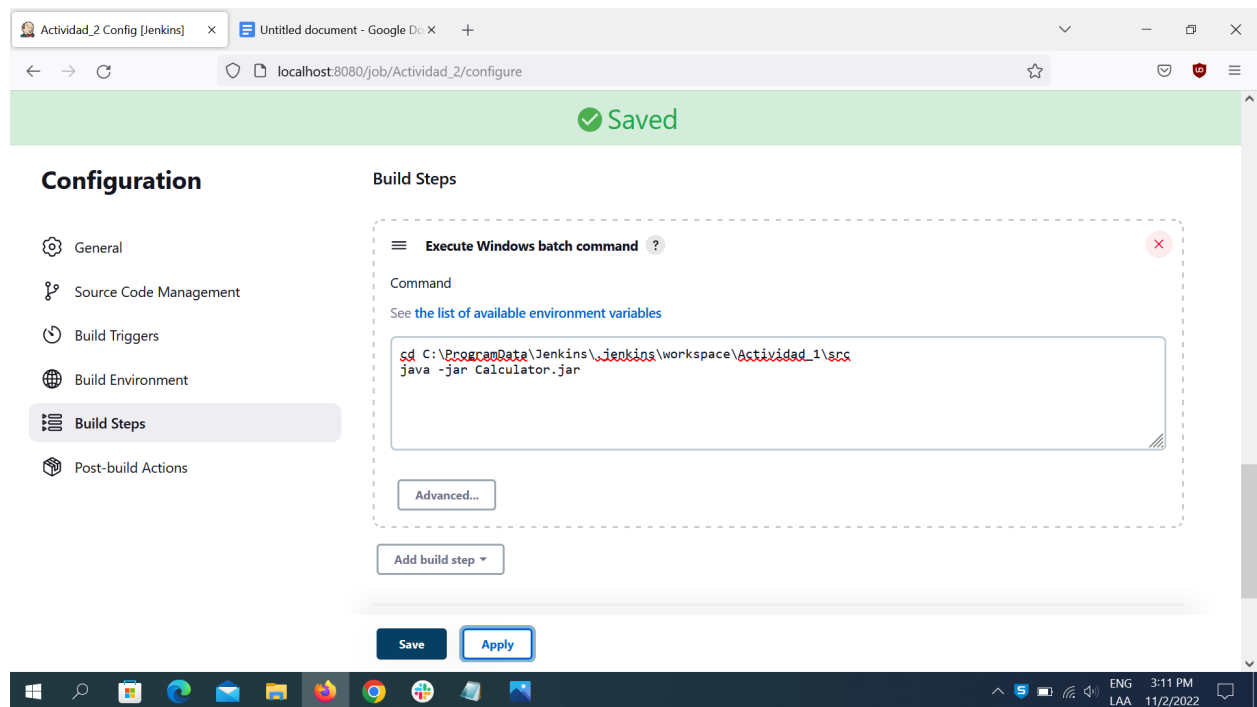
Paso 2:

En source code management seleccionamos none



Paso 3:

En build steps seleccionamos la consola de windows y escribimos nuestro código



Paso 4:

Ejecutamos la build y luego vamos a console output para ver el resultado de la ejecución de nuestro jar

