

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*      Repository name \*

yavv98

/ segundo\_entregable\_DEVOPS-WEBAPPS

✓ segundo\_entregable\_DEVOPS-WEBAPPS is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-disco](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

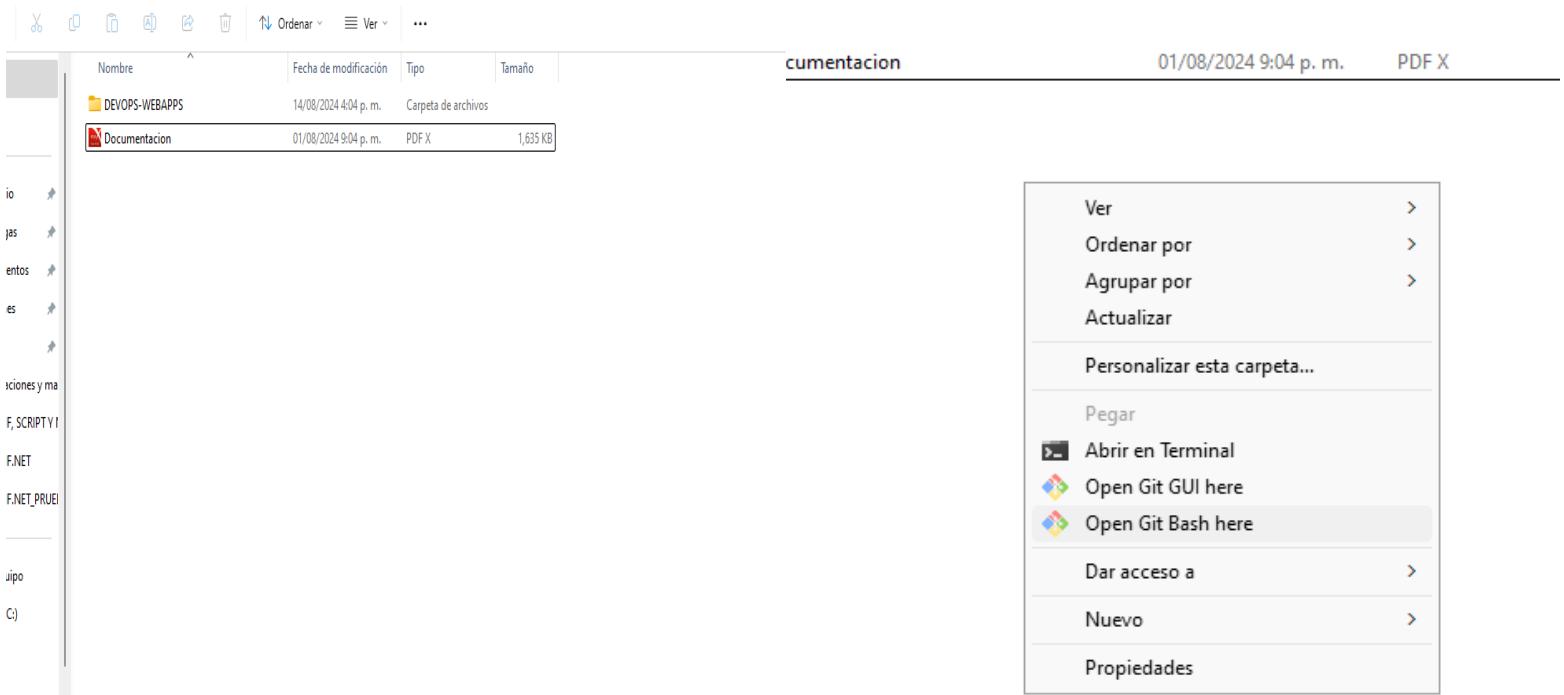
Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Empezamos con la creación de repositorio en GitHub este con el nombre de segundo\_entregable\_DEVOPS-WEBAPPS.

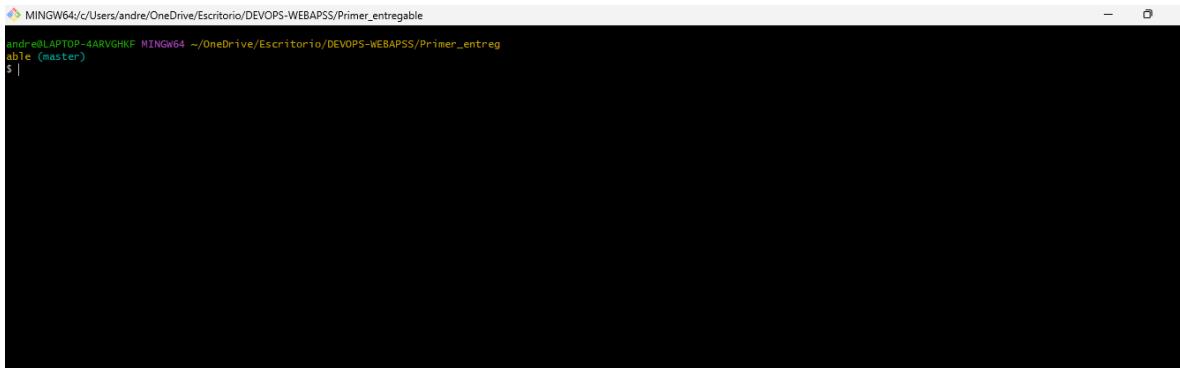
The screenshot shows the GitHub repository page for 'segundo\_entregable\_DEVOPS-WEBAPPS'. The repository is public. It features sections for GitHub Copilot setup, adding collaborators, and quick setup instructions. The URL in the address bar is [https://github.com/yavv98/segundo\\_entregable\\_DEVOPS-WEBAPPS](https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAPPS).

Aquí está creado el repositorio, ya faltaría cargar el documento que está en nuestro computador.



Nos ubicamos en la carpeta que necesitamos, pero antes de todo esto debemos de tener el Git Bash Instalado.

Damos click derecho y abrimos con el Open Git Bash here.



Se nos abre una consola para poder ingresar los comandos.

```
$ git init
Reinitialized existing Git repository in C:/Users/andre/OneDrive/Escritorio/DEVOPS-WEAPSS/Primer_entregable/.git/
andre@LAPTOP-4ARVGHKF MINGW64 ~/OneDrive/Escritorio/DEVOPS-WEAPSS/Primer_entregable (master)$
```

Ingresamos el comando `git init`

```
$ git add .
warning: in the working copy of 'DEVOPS-WEAPPS/node_modules/.package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'DEVOPS-WEAPPS/package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'DEVOPS-WEAPPS/package.json', LF will be replaced by CRLF the next time Git touches it
```

Después ingresamos el comando `git add .`

```
$ git commit -m "segundaentrega"
[master e59ff10] segundaentrega
 3 files changed, 4 insertions(+)
 create mode 100644 DEVOPS-WEAPPS/index.css
```

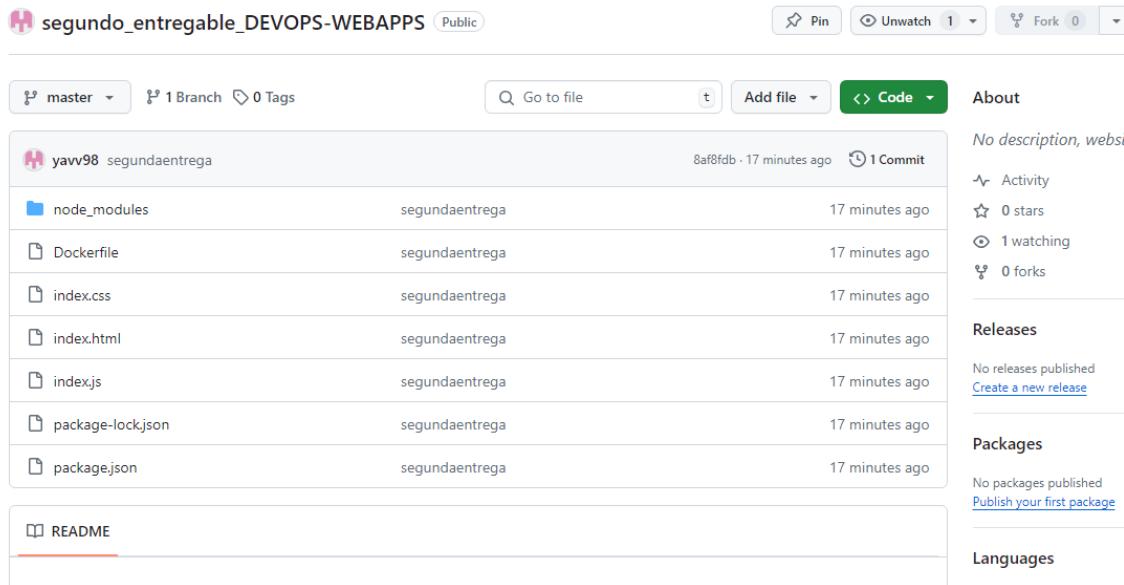
El siguiente paso seria utilizar el comando `git commit -m "nombre del proyecto"`

```
andre@LAPTOP-4ARVGHKF MINGW64 ~/OneDrive/Escritorio/DEVOPS-WEAPSS/Primer_entregable/DEVOPS-WEAPPS (master)
$ git remote add origin https://github.com/yavv98/segundo_entregable_DEVOPS-WEAPPS.git
```

Añadimos donde vamos guardar el repositorio remoto poniendo `git remote add origin` y añadimos la ruta que nos brinda GitHub

```
andre@LAPTOP-4ARVGHKF MINGW64 ~/OneDrive/Escritorio/DEVOPS-WEAPSS/Primer_entregable/DEVOPS-WEAPPS (master)
$ git push -u origin master
Enumerating objects: 617, done.
Counting objects: 100% (617/617), done.
Delta compression using up to 4 threads
Compressing objects: 100% (582/582), done.
Writing objects: 100% (617/617), 683.48 KiB | 4.92 MiB/s, done.
Total 617 (delta 114), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (114/114), done.
To https://github.com/yavv98/segundo_entregable_DEVOPS-WEAPPS.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Por último, ingresamos el comando `git push -u origin master`

A screenshot of a GitHub repository page. The repository name is "segundo\_entregable\_DEVOPS-WEBAPPS". It shows 1 branch and 0 tags. The master branch has 1 commit by user "yavv98" from 17 minutes ago. The commit message is "segundaentrega". The repository contains files: node\_modules, Dockerfile, index.css, index.html, index.js, package-lock.json, and package.json, all last modified 17 minutes ago by "segundaentrega". A README file is also present. On the right side, there are sections for About (no description), Activity (0 stars, 1 watching, 0 forks), Releases (no releases published, Create a new release), Packages (no packages published, Publish your first package), and Languages (a progress bar).

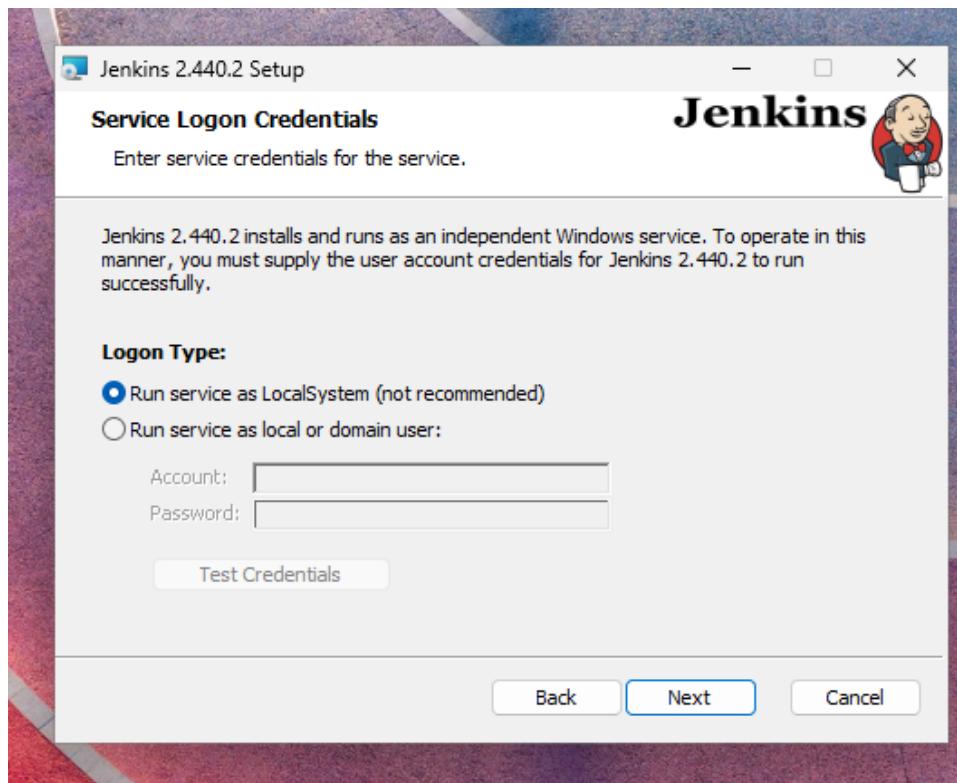
Resultado, vemos que en GitHub ya esta el repositorio listo y cargado 😊 .

## ¿Como instalar Jenkins?

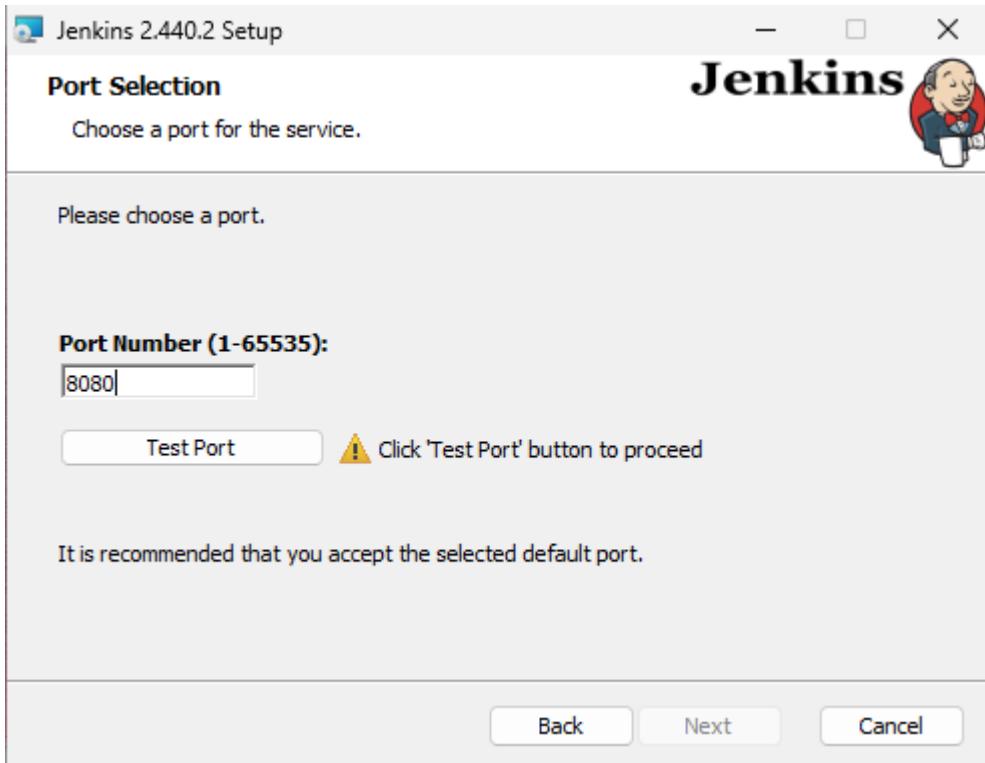
Primero debemos de instalar el jdk-21.



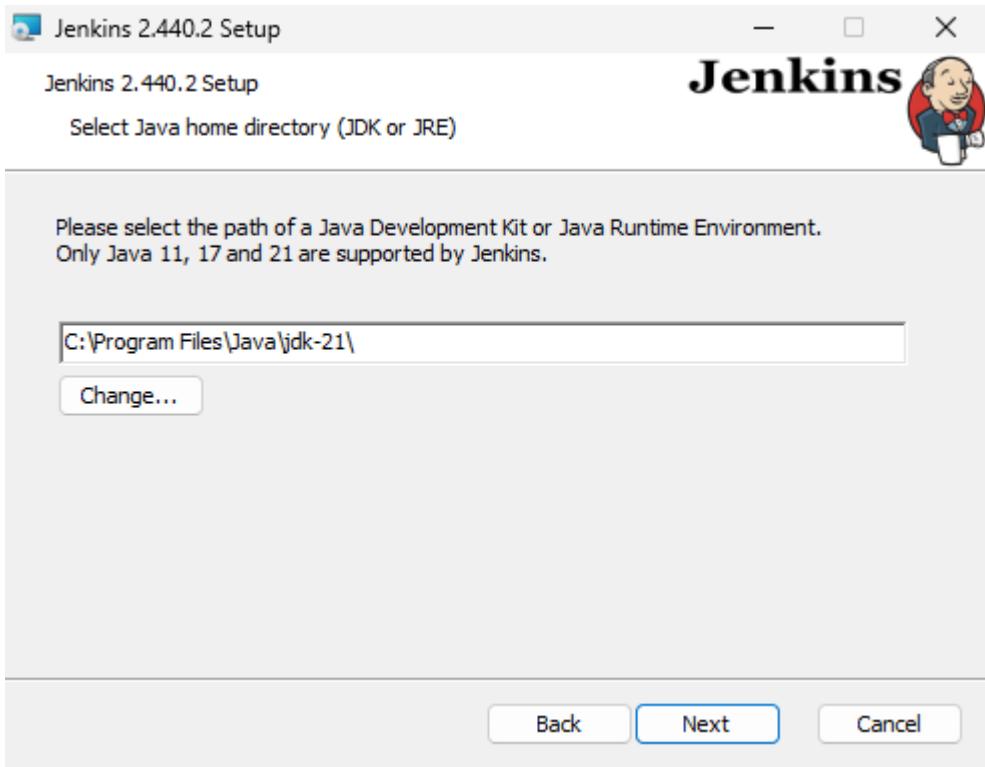
Después de haber instalado el jdk -21 procedemos a instalar JENKINS.



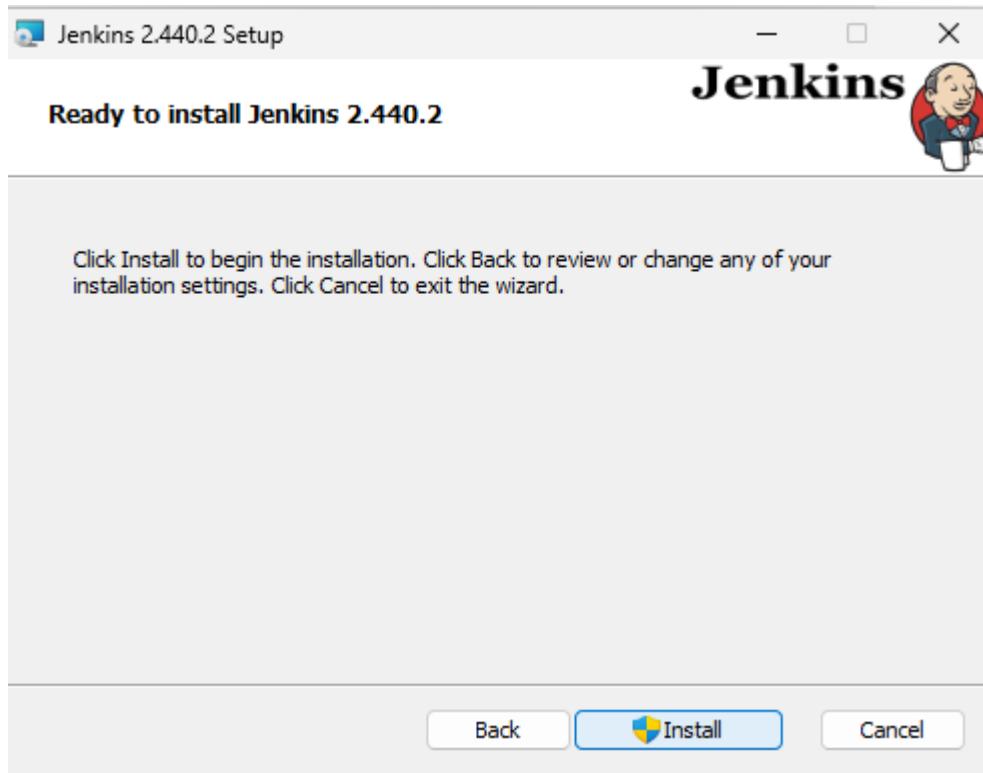
Seleccionamos la opción Run service as localSystem. Y le damos Next.



Seleccionamos el puerto que deseamos, para este ejemplo voy a dejar el puerto 8080, después le damos click en Test Port y damos Next.



Buscamos la carpeta donde se instaló el jdk-21 de Java y damos Next.



Por último, le damos **Install**.

Después en el navegador buscamos `localhost:8080`, nos logiamos y ya tenemos Jenkins listo para usar.

## Creación de PIPELINE

The screenshot shows the Jenkins Control Panel. At the top, there's a logo of a cartoon character holding a coffee cup next to the word "Jenkins". Below the logo, a navigation bar says "Panel de Control >". A large button labeled "+ Nueva Tarea" is prominent. Below it, there are several links: "Personas", "Historial de trabajos", "Administrar Jenkins", and "Mis vistas". Under "Trabajos en la cola", it says "No hay trabajos en la cola".

Le damos en la opción de nueva tarea.

The screenshot shows the "Create New Item" dialog in Jenkins. The title bar says "Enter an item name". Below it, there's a text input field containing "TRABAJO-YEISON-VALENCIA" with a note "» Required field". There are four options listed: "Crear un proyecto de estilo libre" (Freestyle Project), "Pipeline" (selected), "Crear un proyecto multi-configuration" (Multi-Config Project), and "Folder". The "Pipeline" option is described as managing long-running activities across multiple agents, suitable for building pipelines and complex workflows. The "OK" button at the bottom is highlighted.

Ponemos nombre, seleccionamos Pipeline y damos ok.

The screenshot shows the Jenkins configuration interface for a project named 'PROYECTO-YEISON-VALENCIA'. The 'General' tab is selected. At the top right, there is a toggle switch labeled 'Enabled' with a checked checkbox. Below the tabs, there are three sections: 'General' (selected), 'Advanced Project Options', and 'Pipeline'. The 'General' section contains a 'Descripción' field with a placeholder 'Descripción' and a 'Plain text' link. Below it are several checkboxes: 'Desechar ejecuciones antiguas', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', and 'Esta ejecución debe parametrizarse'. At the bottom are 'Guardar' and 'Apply' buttons.

Nos trae a este apartado gráfico.

The screenshot shows the Jenkins Pipeline script editor. The title bar says 'Definition' and the dropdown menu shows 'Pipeline script'. The main area is titled 'Script' with a question mark icon. It contains a code editor with the following Groovy script:

```
19    steps
20    {
21        git credentialsId: 'git_credentials', url: 'https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS'
22    }
23
24}
25
26 stage("test"){
27     steps{
28         echo 'Testeando aplicacion'
29     }
30 }
31 }
32 }
33
34 }
```

Below the code editor is a checkbox labeled 'Use Groovy Sandbox' with a checked checked checkbox.

Nos desplazamos hacia la parte de abajo y nos enfocamos en esta parte, donde dice **Script**, Aquí es donde vamos a escribir el código, en mi caso yo escribí este script.

```

1 ◀ pipeline{
2     agent any
3     stages
4 ▶   {
5         stage("build")
6 ▶         {
7             steps{
8                 echo 'Construccion de aplicacion'
9             }
10    }
11    stage("instalando dependencias")
12 ▶    {
13        steps{
14            echo 'Instalando dependencias'
15        }
16    }
17    stage("git")
18 ▶    {
19        steps
20 ▶            {
21            git credentialsId: 'git_credentials', url: 'https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git'
22        }
23    }
24 }
25
26    stage("test"){
27        steps{
28            echo 'Testeando aplicacion'
29        }
30    }
31 }
32 }
33

```

Aquí tengo 4 stages, pero el mas importante es el stage("git"), aquí es donde voy a clonar mi repositorio que esta en GitHub, pero ¿cómo hice para traerlo?, vamos a verlo a continuación.

The screenshot shows the Jenkins 'Configure' screen for a pipeline job. On the left, there are three tabs: 'General', 'Advanced Project Options', and 'Pipeline'. The 'Pipeline' tab is currently selected and highlighted in grey. To the right of the tabs is a code editor containing a Groovy pipeline script. The 'git' stage is highlighted with a light blue background. At the bottom of the code editor, there is a checkbox labeled 'Use Groovy Sandbox' with a checked status. Below the code editor, there are two buttons: 'Guardar' (Save) and 'Apply'.

```

16
17
18 ▶   {
19     steps
20 ▶         {
21         git credentialsId: 'git_credentials', url: 'https://github.com/yavv98/segundi
22     }
23
24
25
26    stage("test"){
27        steps{
28            echo 'Testeando aplicacion'
29        }
30    }
31 }
32 }
33

```

Le damos click en la opción Pipeline Syntax.

The screenshot shows the Jenkins interface. At the top, there's a navigation bar with the Jenkins logo, a search bar labeled 'búsqueda (CTRL+K)', and user information for 'Yeison Andres Valencia Vargas'. Below the navigation bar, the page title is 'Panel de Control > PROYECTO-YEISON-VALENCIA > Pipeline Syntax'. On the left, a sidebar lists various links: 'Snippet Generator' (selected), 'Declarative Directive Generator', 'Declarative Online Documentation', 'Steps Reference', 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSL'. The main content area is titled 'Overview' and contains a paragraph about the Snippet Generator. It includes a 'Steps' section with a 'Sample Step' example: 'archiveArtifacts: Guardar los archivos generados'. A dropdown menu for this step shows options like 'archiveArtifacts ?' and 'Ficheros para guardar ?'. At the bottom of the sidebar, there's a list of pipeline steps.

Al dar click en Pipeline syntax nos lleva a esta parte de Jenkins.

A screenshot of the Jenkins Pipeline Snippet Generator. The 'git: Git' option is highlighted with a blue selection bar. Below it, other options like 'deleteDir', 'dir', 'echo', etc., are listed. At the bottom of the list, there's another 'archiveArtifacts' entry.

```
deleteDir: Recursively delete the current directory from the workspace
dir: Change current directory
echo: Print Message
emailext: Extended Email
emailexrecipients: Extended Email Recipients
error: Error signal
fileExists: Verify if file exists in workspace
findBuildScans: Find published build scans
fingerprint: Almacenar firma de ficheros para poder hacer seguimiento
git: Git
githubPRStatus: GitHub PR: set 'pending' status
gitPush: Git Push
githubPRAddLabels: GitHub PR: add labels
archiveArtifacts: Guardar los archivos generados
```

Damos click archiveArtifacts: Guardar los archivos generados y seleccionamos la opción de git: Git.

A screenshot of the Jenkins Pipeline Snippet Generator. The 'git: Git' option is selected. Below it, there's a 'Repository URL' input field which is empty and has a red error message: 'Please enter Git repository.'.

Después me pide que le pongamos el URL del repositorio que tenemos en GitHub.

The screenshot shows a GitHub repository page. At the top, the URL is `github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS`. The repository name is `segundo_entregable_DEVOPS-WEBAAPPS`, which is marked as `Public`. Below the repository name, there's a file tree showing files like `node_modules`, `Dockerfile`, `index.css`, `index.html`, `index.js`, `package-lock.json`, and `package.json`, all belonging to the `segundaentrega` folder. On the right side, there's a 'Clone' section with tabs for `Local` and `Codespaces`. The `Local` tab is active, showing the `HTTPS` URL `https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git`. Other options include `SSH` and `GitHub CLI`. Below the URL, there are links to `Open with GitHub Desktop` and `Download ZIP`. The page also includes sections for `About`, `Activity`, `Releases`, and `Packages`.

Vamos a GitHub damos click en el botón verde que dice `<> code ▾`, y nos copiamos el HTTPS.

The screenshot shows the Jenkins Pipeline configuration interface. It has several input fields: 'Repository URL' with the value `https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git`; 'Branch' with the value `dev`; 'Credentials' with the value `admin/*****`; and two checkboxes: 'Include in polling?' and 'Include in changelog?', both of which are checked. At the bottom, there's a large blue button labeled 'Generate Pipeline Script'. Below this button, the generated pipeline script is shown in a text area:

```
git branch: 'dev', credentialsId: '86703045-a6b8-45c3-bb6f-c346a75b9ad2', url: 'https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git'
```

Ponemos el HTTPS del repositorio, en Branch ponemos dev, en credentials desplegamos y buscamos la opción de `admin/*****` este seria el usuario con el que ingresamos al Jenkins, Le damos click en `Generate Pipeline Script`, El nos genera un código en la parte inferior, ese código lo copiamos y no lo llevamos al script que estábamos haciendo con anterioridad.

```

stage("git")
{
    steps
    {
        git credentialsId: 'git_credentials', url: 'https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git'
    }
}

```

En el script pegamos el código que nos generó Jenkins.

```

8    echo 'Construccion de aplicacion'
9
10   }
11
12   stage("Instalando dependencias")
13   {
14       steps{
15           echo 'Instalando dependencias'
16       }
17   }
18   stage("git")
19   {
20       steps
21       {
22           git credentialsId: 'git_credentials', url: 'https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git'
23       }
24   }
25
26   stage("test"){
27       steps{
28           echo 'Testeando aplicacion'
29       }
30
31
32
33
34

```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

[Guardar](#) [Apply](#)

Por último, le damos Apply y guardar, si hay algún error en la sintaxis del script automáticamente nos mostrara un aviso mostrando error.

Panel de Control > PROYECTO-YEISON-VALENCIA >

**PROYECTO-YEISON-VALENCIA**

Status Changes Construir ahora Configurar Borrar Pipeline Full Stage View GitHub Rename Pipeline Syntax

añadir descripción Desactivar el Proyecto

No data available. This Pipeline has not yet run.

Stage View

Enlaces permanentes

Historia de tareas Tendencia

Al darle Apply y Guardar nos mandara a esta página, y le damos en la opción de construir ahora.

Rename

En

Stage View

Pipeline Syntax

Historia de tareas Tendencia ▾

Filter... /

#16 17 ago 2024, 16:28

Atom feed Para Atom feed para los



Enlaces permanentes

Al darle construir ahora el nos mostrara estas ventanas, cuando todo sale en color verde quiere decir que todo está Ok 😊.

**Stage Logs (git)**

Git (self time 5s)

```
The recommended git tool is: NONE
using credential git_credentials
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\PROYECTO-VEISON-VALENCIA\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git # timeout=10
Fetching upstream changes from https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.0.windows.1'
using GIT_ASKPASS to set credentials git_credentials
> git.exe fetch --tags --force --progress -- https://github.com/yavv98/segundo_entregable_DEVOPS-WEBAAPPS.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 8af8fdbca9370bedbee5decbda206aa8108a2dc85 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 8af8fdbca9370bedbee5decbda206aa8108a2dc85 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10
> git.exe checkout -b master 8af8fdbca9370bedbee5decbda206aa8108a2dc85 # timeout=10
Commit message: "segundaentrega"
First time build. Skipping changelog.
```

Enlaces permanentes

Si nos enfocamos en el stage git que es donde clonamos el repositorio de GitHub, muestra que esta ok y que no hubo ningún problema.