No __ s19070104 ___

INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

Libramiento Fresnillo-Tepetongo, Fracc. Los Cardos, 99863 Jerez de García Salinas, Zac.

GARCIA FLORES ANAHI

anahi.gf@tecjerez.edu.mx

Nallely Morales

INGENIERÍA EN SISTEMAS COMPUTACIONALES

DevSecOps - Desarrollo, Seguridad y Operaciones de Software

Prof. Salvador Acevedo

9° SEMESTRE

EVALUACIÓN PARCIAL 3 [T3]

Fecha:12/11/25

REPORTE DE PRÁCTICA: Implementación de PIPELINES para CI/CD con Jenkins

1. OBJETIVO GENERAL

Implementar un proceso automatizado de Integración Continua (CI) y Entrega Continua (CD) utilizando Jenkins, con conexión a un repositorio de GitHub y webhook (Ngrok), para desplegar y analizar un proyecto en Laravel.

2. OBJETIVOS ESPECÍFICOS

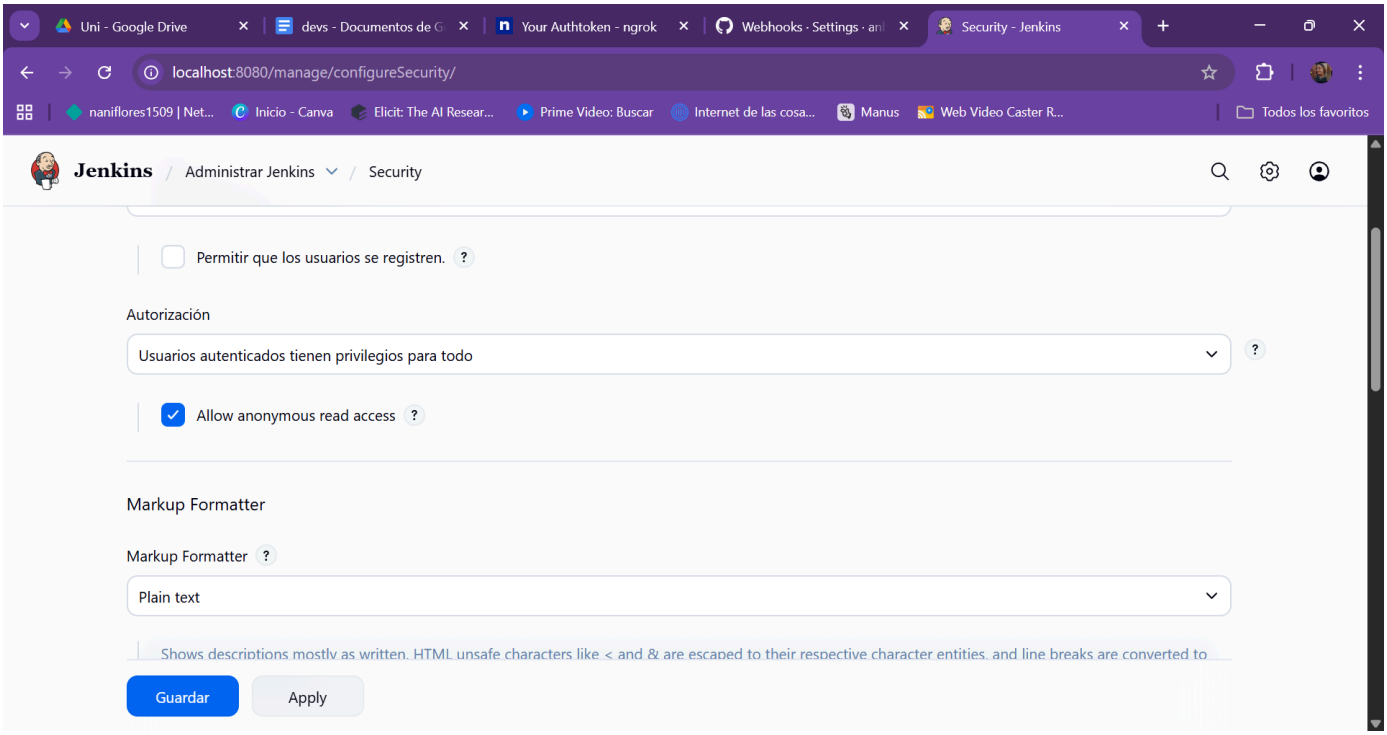
- Configurar un servidor Jenkins en entorno local.
- Establecer una conexión segura entre Jenkins y GitHub mediante Ngrok.
- Crear un Pipeline declarativo que automatice las etapas de build, test y despliegue.
- Ejecutar automáticamente el pipeline al detectar un evento `push` en GitHub.
- Analizar el código fuente con herramientas de calidad y generar un reporte.

3. DESARROLLO DE LA PRÁCTICA

3.1 Configuración inicial del entorno

- Se instaló Jenkins en un servidor local con Java 17.
- Se creó un usuario administrador y se configuraron los plugins:
    - GitHub Integration Plugin
    - Pipeline
    - NodeJS Plugin (para React)
    - PHP Plugin (para Laravel)

El proyecto a integrar fue el sistema web de tutorías institucional, alojado en el repositorio: https://github.com/anhiGF/Proyecto.git



3.2 Conexión NGROK con GITHUB (Creación del Webhook)

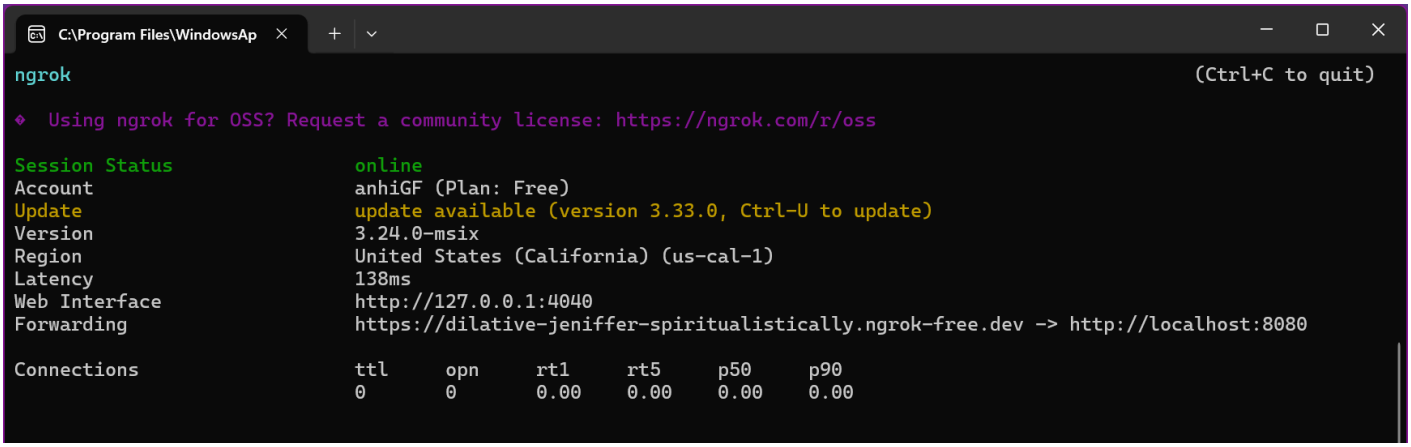
Para exponer Jenkins en un entorno local y recibir eventos de GitHub, se utilizó Ngrok, que genera una URL pública temporal.

Pasos realizados:

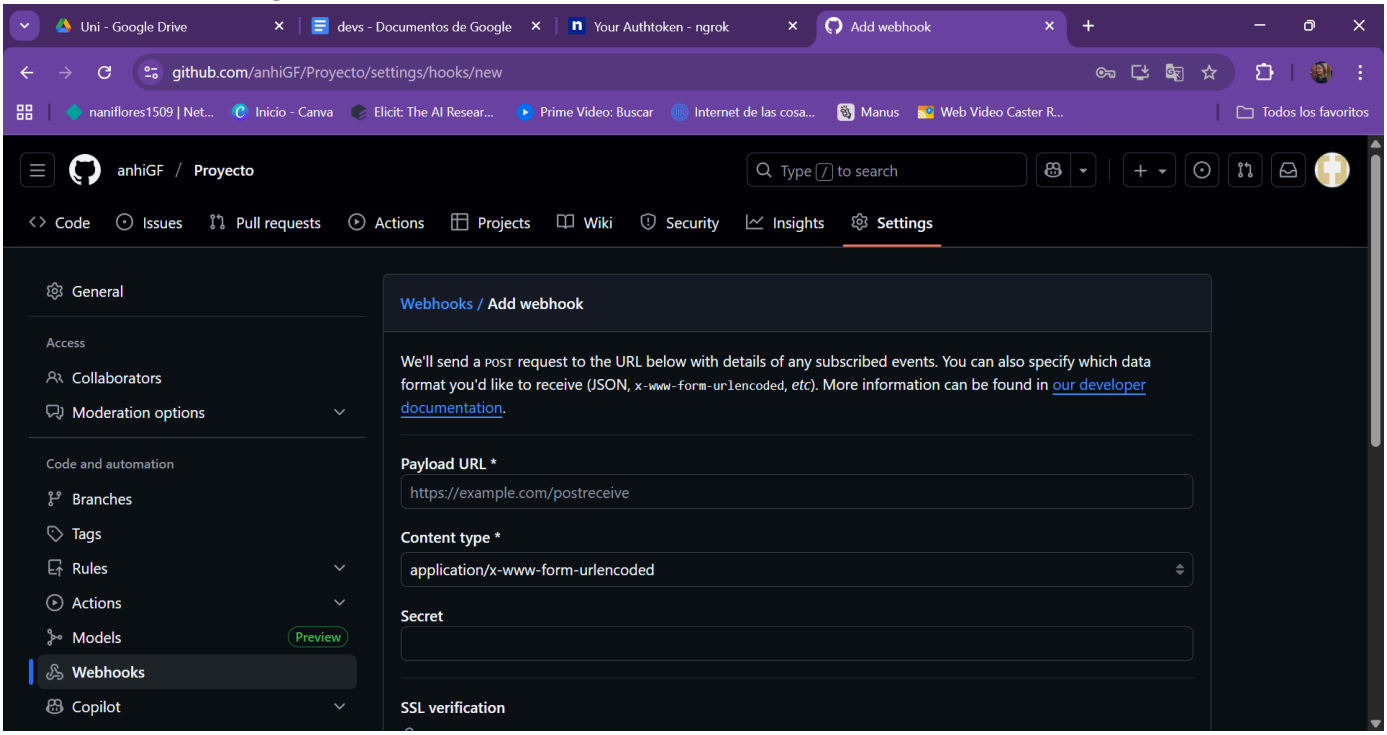
Ejecutar en consola:

```
ngrok http 8080
```

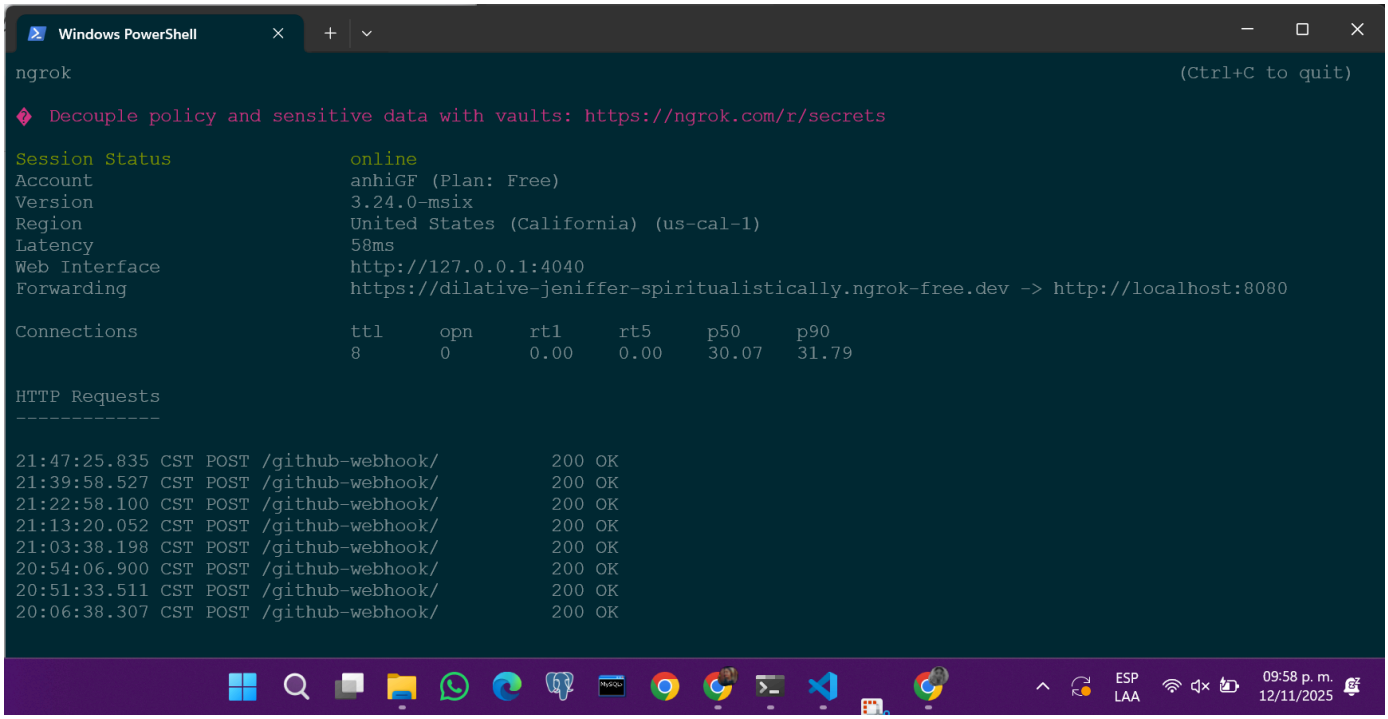
1. Copiar la URL generada



En **GitHub → Settings → Webhooks**:



2. Confirmar que el webhook devuelva `Status: 200 OK`.

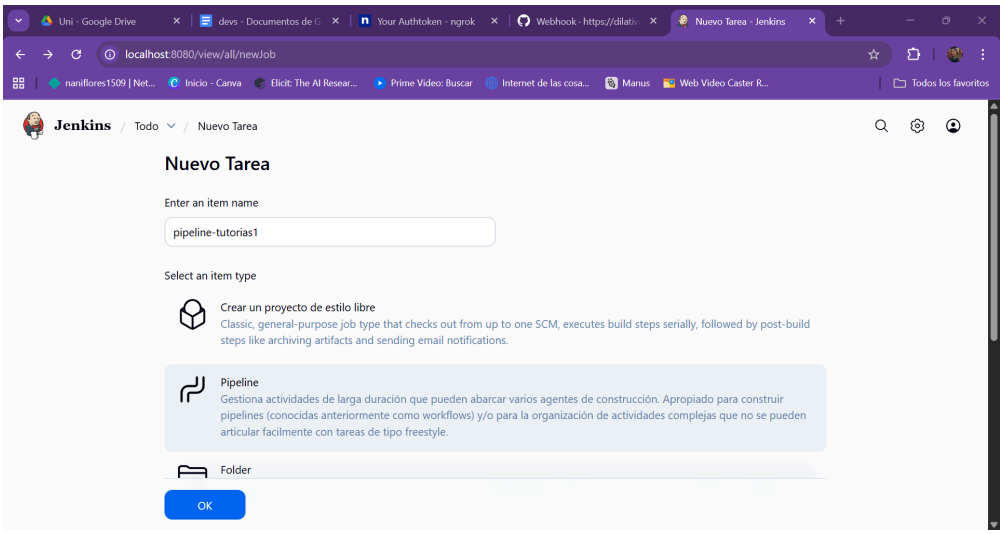


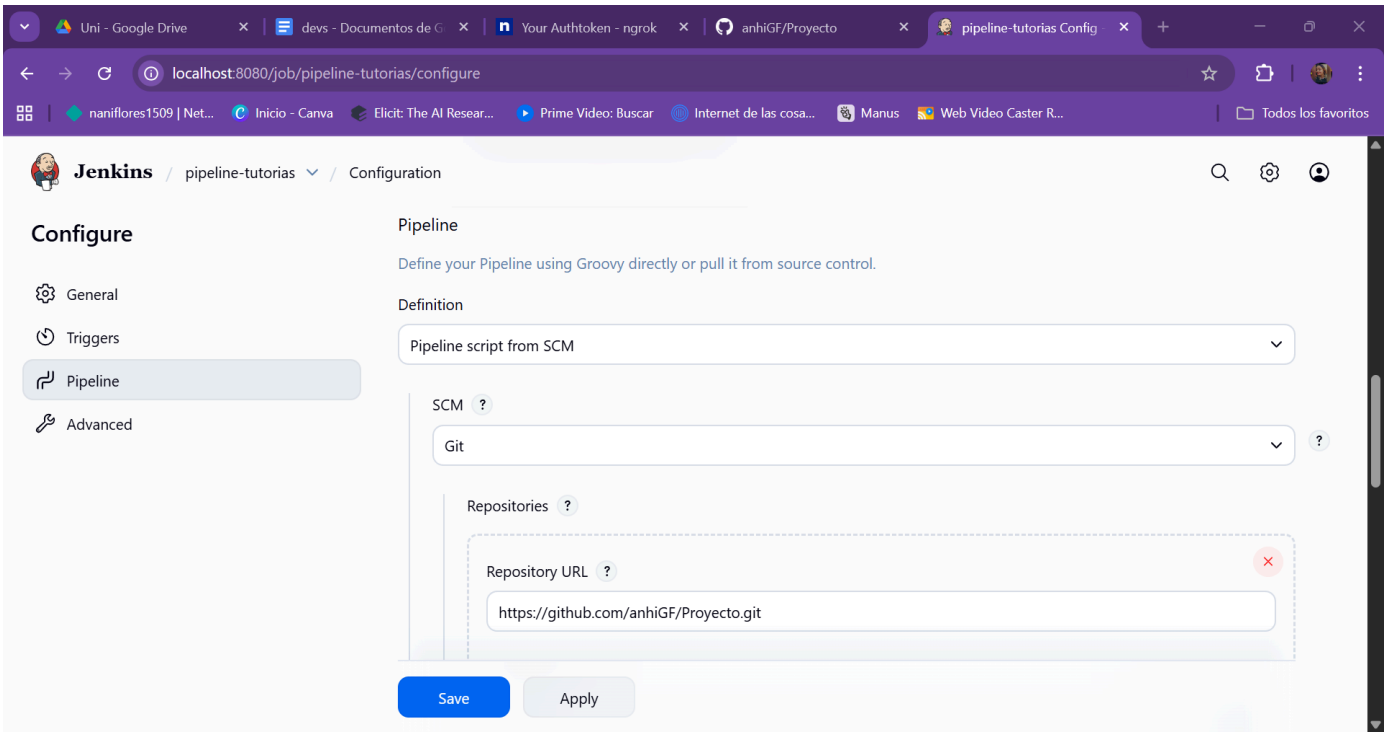
Resultado: GitHub puede enviar eventos a Jenkins cada vez que se hace `git push`.

3.3 Creación del Pipeline en Jenkins

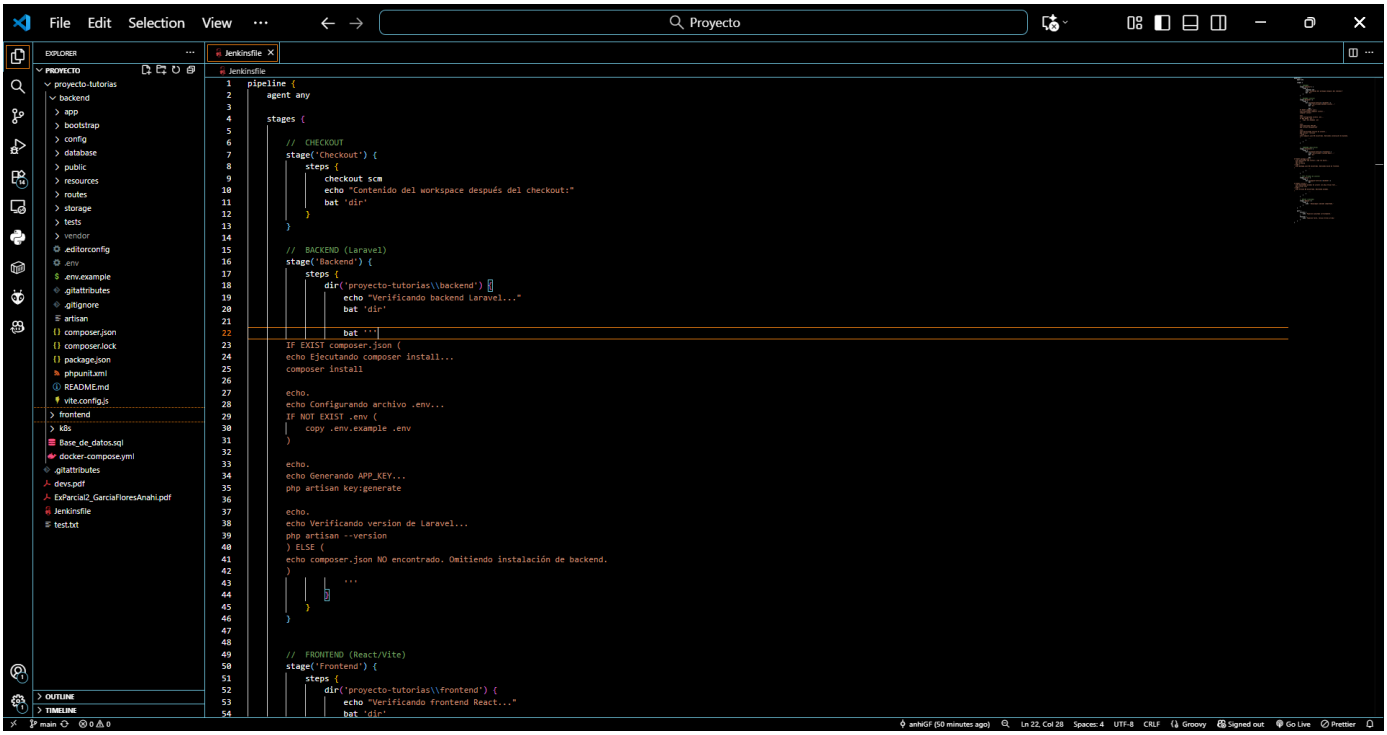
Se creó un nuevo trabajo tipo Pipeline en Jenkins

Definición: Pipeline script from SCM
SCM: Git
Repository URL: https://github.com/anhiGF/Proyecto

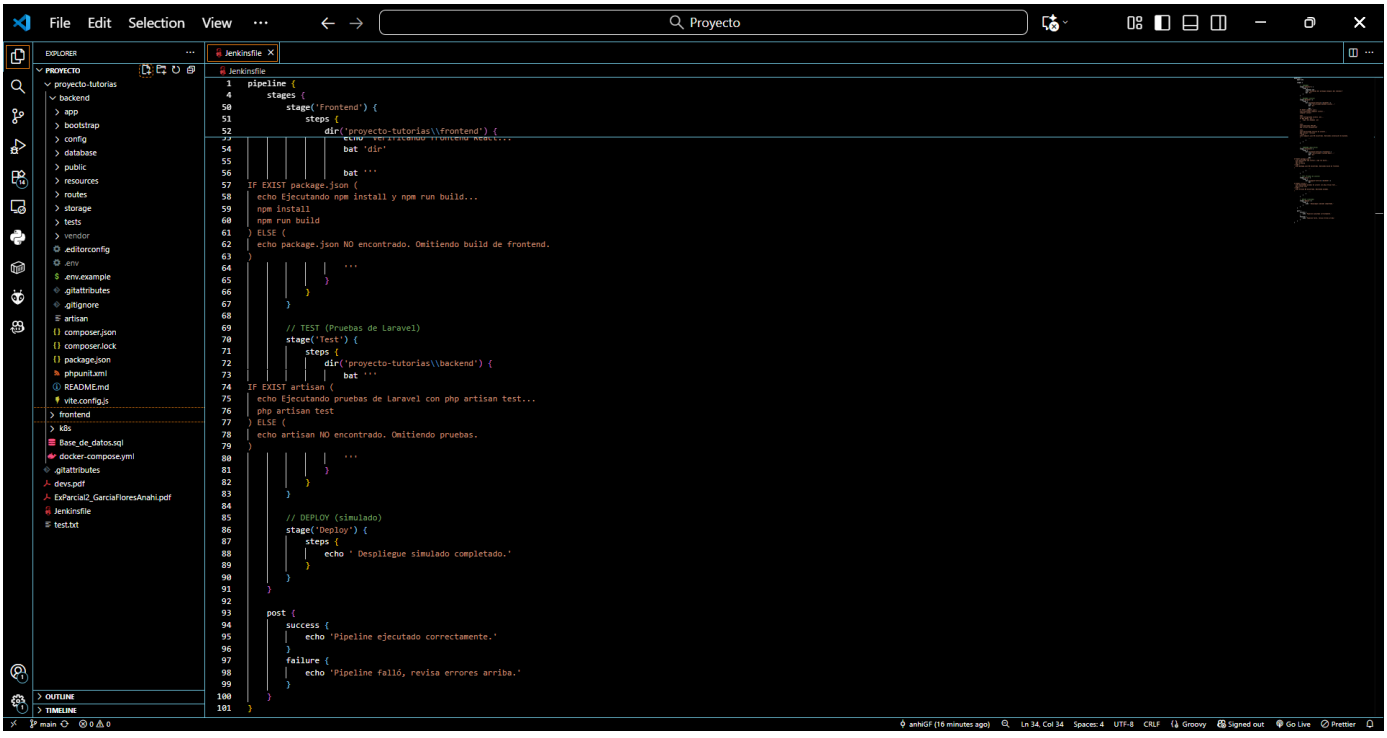




Contenido del archivo `Jenkinsfile`:



```
pipeline {
    agent any

    stages {

        // CHECKOUT
        stage('Checkout') {
            steps {
                checkout scm
                echo "Contenido del workspace después del checkout:"
                bat 'dir'
            }
        }

        // BACKEND (Laravel)
        stage('Backend') {
            steps {
                dir('proyecto-tutorias\\backend') {
                    echo "Verificando backend Laravel..."
                    bat 'dir'

                    bat '''
        IF EXIST composer.json (
        echo Ejecutando composer install...
        composer install

        echo.
        echo Configurando archivo .env...
        IF NOT EXIST .env (
            copy .env.example .env
        )

        echo.
        echo Generando APP_KEY...
        php artisan key:generate

        echo.
        echo Verificando version de Laravel...
        php artisan --version
        ) ELSE (
        echo composer.json NO encontrado. Omitiendo instalación de backend.
        )
                    '''
                }
            }
        }


        // FRONTEND (React/Vite)
        stage('Frontend') {
            steps {
                dir('proyecto-tutorias\\frontend') {
                    echo "Verificando frontend React..."
                    bat 'dir'
```



```
pipeline {
    stages {
        stage('Frontend') {
            steps {
                dir('proyecto-tutorias\\frontend') {
                    bat 'dir'

                    bat '''
IF EXIST package.json (
 echo Ejecutando npm install y npm run build...
 npm install
 npm run build
) ELSE (
 echo package.json NO encontrado. Omitiendo build de frontend.
)
                    '''
                }
            }
        }

        // TEST (Pruebas de Laravel)
        stage('Test') {
            steps {
                dir('proyecto-tutorias\\backend') {
                    bat '''
IF EXIST artisan (
 echo Ejecutando pruebas de Laravel con php artisan test...
 php artisan test
) ELSE (
 echo artisan NO encontrado. Omitiendo pruebas.
)
                    '''
                }
            }
        }

        // DEPLOY (simulado)
        stage('Deploy') {
            steps {
                echo ' Despliegue simulado completado.'
            }
        }
    }

    post {
        success {
            echo 'Pipeline ejecutado correctamente.'
        }
        failure {
            echo 'Pipeline falló, revisa errores arriba.'
        }
    }
}
```

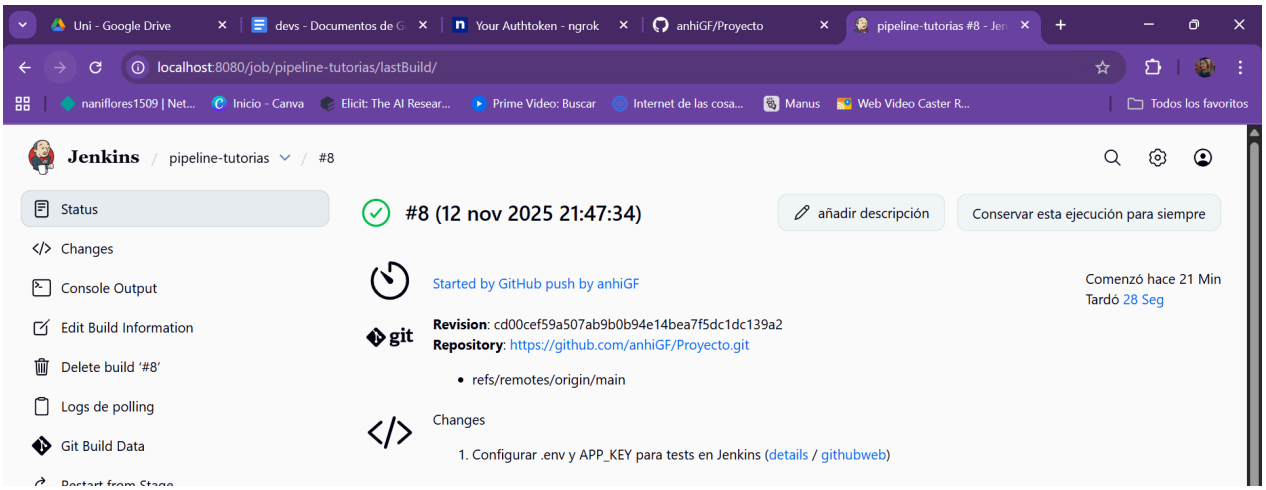
3.4 Ejecución automática del pipeline por evento push

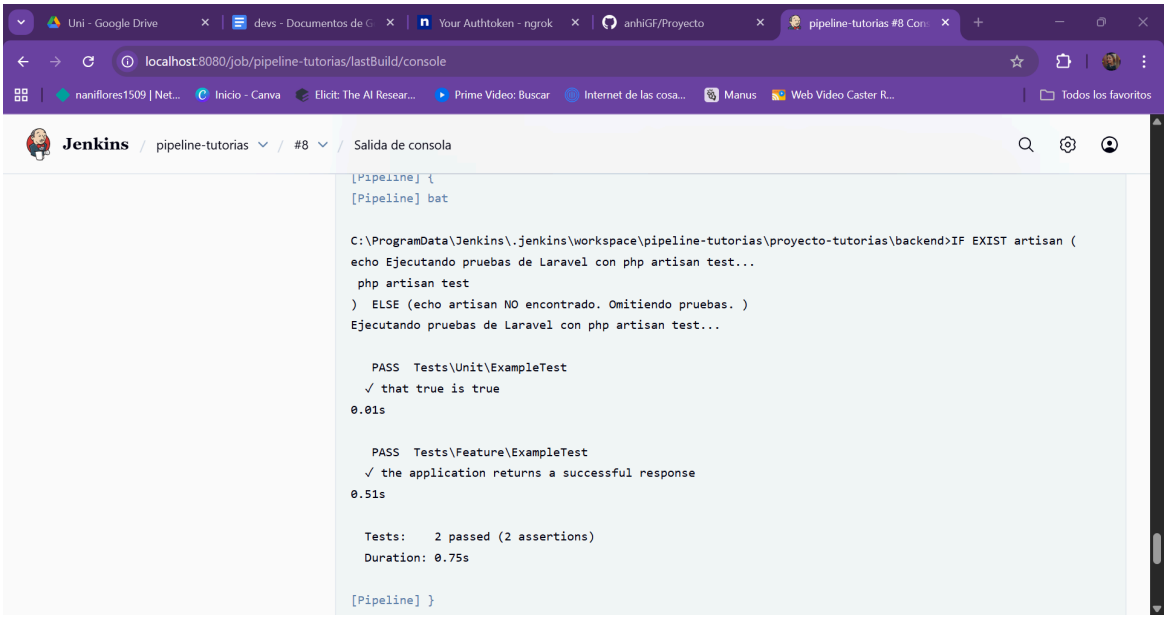
Cada vez que se realiza un `git push` al repositorio, el webhook de GitHub activa automáticamente Jenkins.

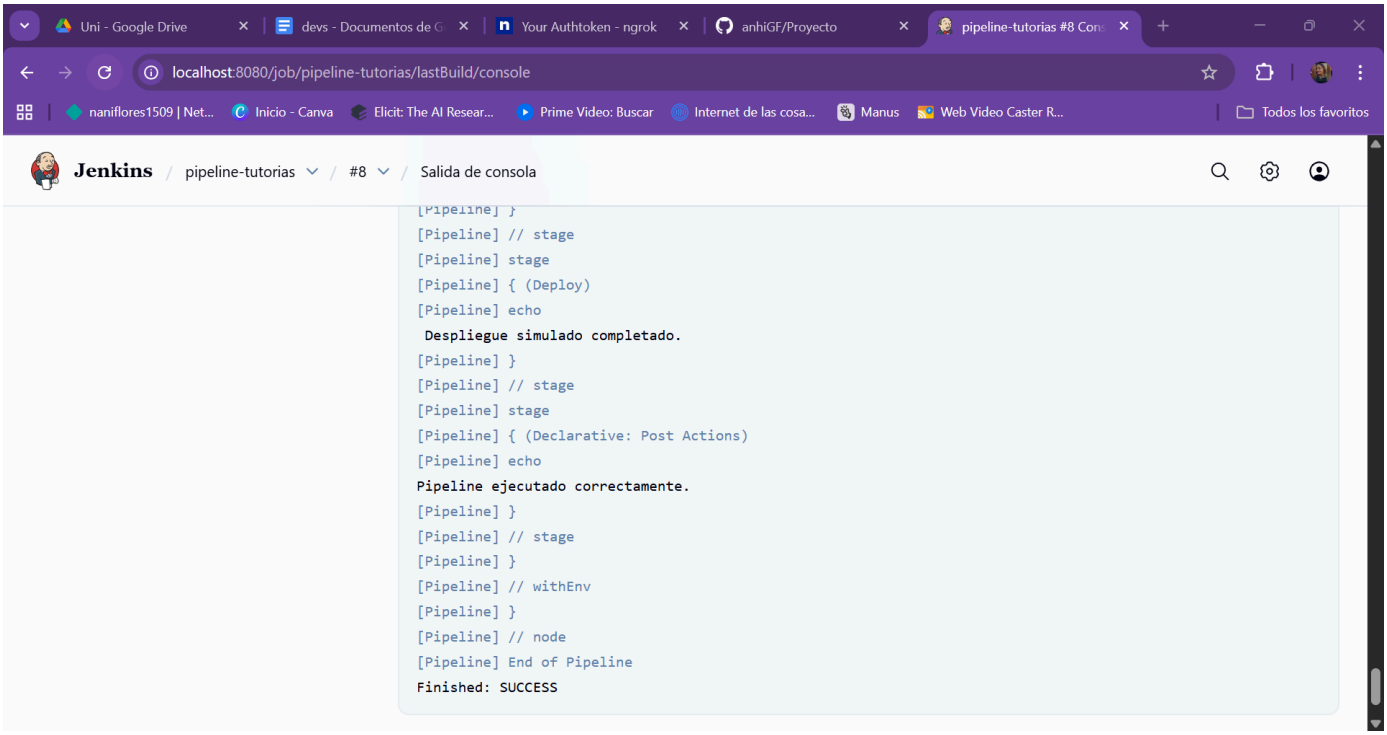
Evidencias:

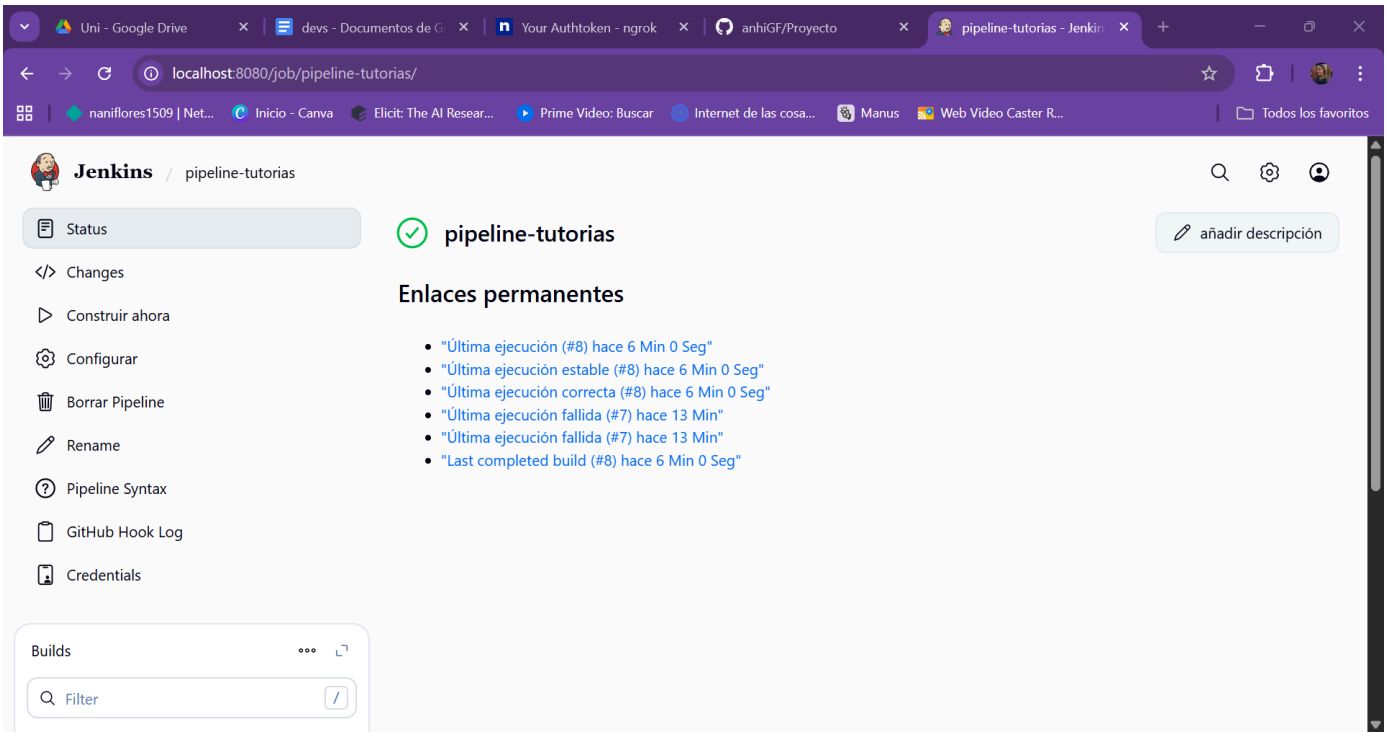
- El pipeline se ejecuta con todas las etapas (`Checkout` → `Backend` → `Frontend` → `Test` → `Deploy`).
- En el historial de Jenkins se registran builds automáticos con #1, #2, etc.

Resultado: Integración continua confirmada.






Uni - Google Drive
devs - Documentos de G
Your Authtoken - ngrok
anhiGF/Proyecto
pipeline-tutorias #8 Cons
localhost:8080/job/pipeline-tutorias/lastBuild/console
naniflores1509 | Net...
Inicio - Canva
Elicit: The AI Resear...
Prime Video: Buscar
Internet de las cosa...
Manus
Web Video Caster R...
Todos los favoritos
Jenkins / pipeline-tutorias / #8 / Salida de consola
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
Despliegue simulado completado.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline ejecutado correctamente.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS


Uni - Google Drive
devs - Documentos de G
Your Authtoken - ngrok
anhiGF/Proyecto
pipeline-tutorias - Jenkin
localhost:8080/job/pipeline-tutorias/
naniflores1509 | Net...
Inicio - Canva
Elicit: The AI Resear...
Prime Video: Buscar
Internet de las cosa...
Manus
Web Video Caster R...
Todos los favoritos
Jenkins / pipeline-tutorias
Status
Changes
Construir ahora
Configurar
Borrar Pipeline
Rename
Pipeline Syntax
GitHub Hook Log
Credentials
Builds
Filter
pipeline-tutorias
añadir descripción
Enlaces permanentes
"Última ejecución (#8) hace 6 Min 0 Seg"
"Última ejecución estable (#8) hace 6 Min 0 Seg"
"Última ejecución correcta (#8) hace 6 Min 0 Seg"
"Última ejecución fallida (#7) hace 13 Min"
"Última ejecución fallida (#7) hace 13 Min"
"Last completed build (#8) hace 6 Min 0 Seg"

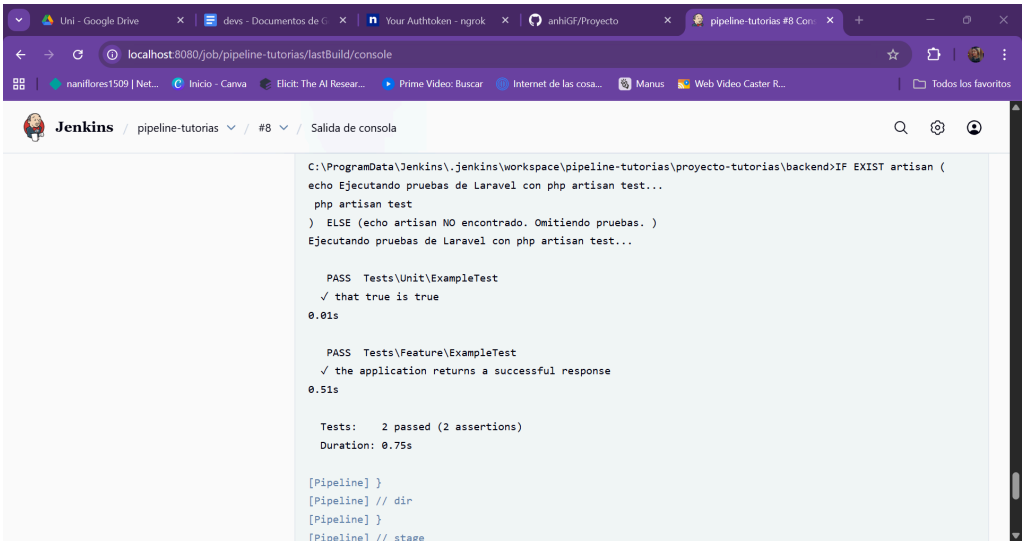
3.5 Reporte de análisis de código

Se agregaron pruebas utilizando php artisan test.

Gracias a la etapa Backend que genera automáticamente:

- .env
- APP_KEY

Las pruebas pudieron ejecutarse correctamente:



Resultado:

- Esto demuestra que el backend está correctamente configurado dentro del entorno CI.

3.6 Enlace del repositorio en GitHub

Repositorio oficial del proyecto:
https://github.com/anhiGF/Proyecto.git

El repositorio contiene:

- Código fuente completo (frontend y backend)
- Script SQL `Base_de_datos.sql`
- Jenkinsfile

CONCLUSIÓN

La práctica permitió configurar un flujo CI/CD completo y funcional usando Jenkins, GitHub y Ngrok. El pipeline implementado logró automatizar exitosamente:

- Descarga del código (Checkout)
- Instalación de dependencias de Laravel y React
- Construcción del frontend
- Generación del archivo .env y APP_KEY
- Ejecución de pruebas unitarias y funcionales
- Un proceso de despliegue simulado

Esto garantiza una entrega continua confiable y con control de calidad, reduciendo tiempo y errores humanos durante el desarrollo del proyecto.

Referencias:

Download and deploy. (n.d.). Download and Deploy. https://www.jenkins.io/download/

ANHIGF/Proyecto. (n.d.). GitHub. https://github.com/anhiGF/Proyecto/tree/main

Ngrok, & Ngrok. (n.d.). *Download ngrok*. https://ngrok.com/download/windows

Laravel LLC. (2025). *Testing: Getting Started*. https://laravel.com/docs/testing

Jenkins Project. (2023). *Building a JavaScript and PHP application with Jenkins Pipeline*. https://www.jenkins.io/doc/tutorials/

GitHub Docs. (2025). *Triggering CI/CD With Webhooks*. https://docs.github.com/en/webhooks-and-events/webhooks/creating-webhooks

Laravel Docs. (2025). *Configuring Environment Variables & APP_KEY*. https://laravel.com/docs/configuration