

PRUEBA UNITARIA PIPELINE DECLARATIVO

Elaborado por: Ubeimar Lizardo Yepes Portilla – Juan Pablo Lucero Morales

Cuentas GitHub: UbeimarY – Juanp-lucero

- **Jenkinsfile Utilizado**

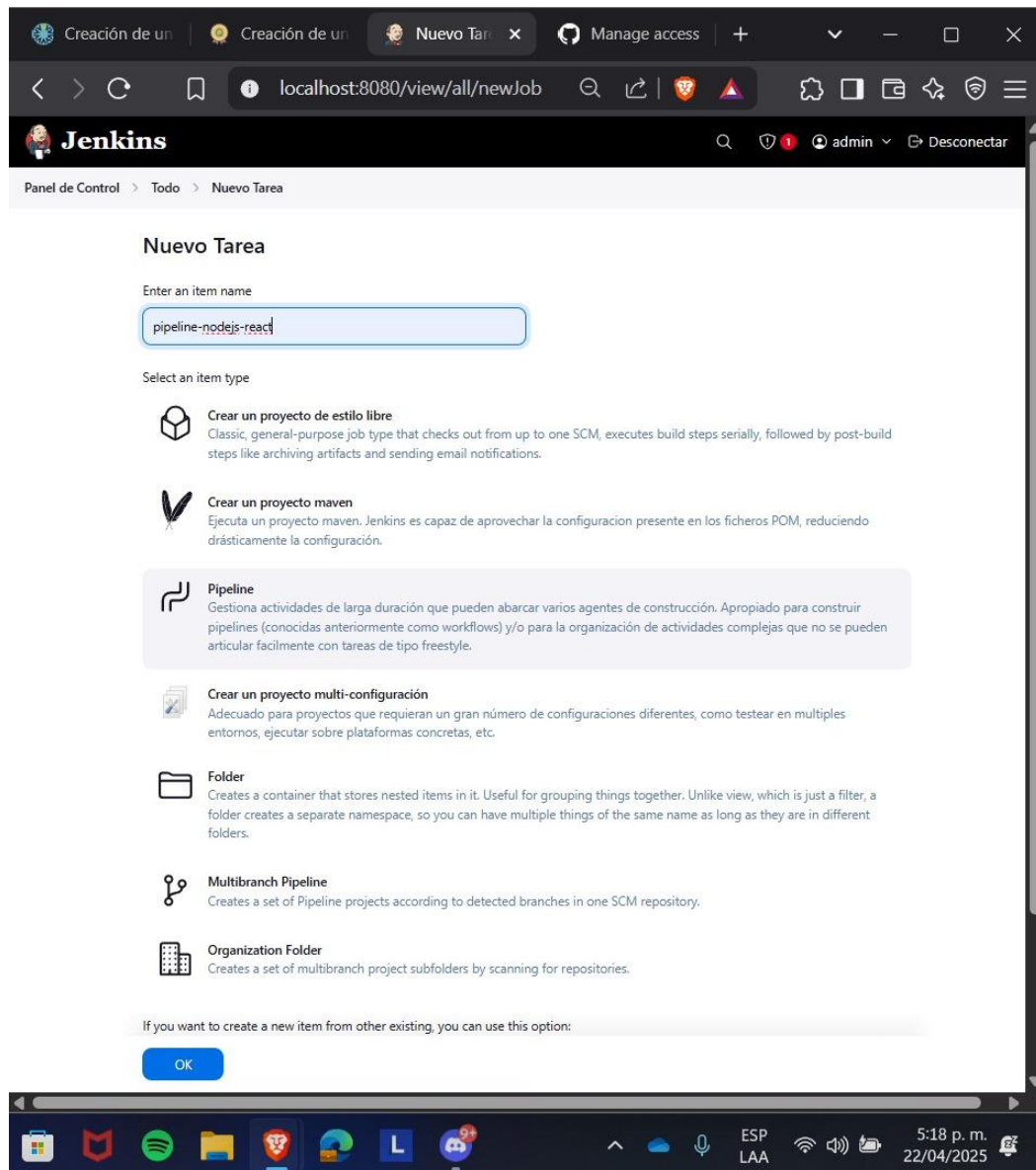
```
pipeline {
  agent any

  environment {
    CI = "false" // Desactiva que React trate los warnings como errores
  }

  stages {
    stage('Declarative: Checkout SCM') {
      steps {
        checkout scm
      }
    }
    stage('Tool Install') {
      steps {
        tool name: 'Node 20', type: 'nodejs'
      }
    }
    stage('Clean workspace') {
      steps {
        deleteDir()
      }
    }
    stage('Checkout') {
      steps {
        git url: 'https://github.com/guswill24/node-project.git', branch: 'main'
      }
    }
    stage('Install dependencies') {
      steps {
        bat 'npm install --legacy-peer-deps'
      }
    }
    stage('Run tests') {
      steps {
        bat 'npm test -- --watchAll=false'
```

- **Evidencias del Proceso**
 1. **Configuración de Node.js en Jenkins**





3. Configuración del SCM desde GitHub

Creación de un Pipeline Declarativo | Creación de un Pipeline Declarativo | pipeline-nodejs-react Config [X] | UbeimarY/node-project-task | +

< > ↻ | 📖 | ⓘ localhost:8080/job/pipeline-nodejs-react/configure

Panel de Control > pipeline-nodejs-react > Configuration

Configure

- General
- Triggers**
- Pipeline
- Advanced

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- ☐ Construir tras otros proyectos ⓘ
- ☐ Consultar repositorio (SCM) ⓘ
- ☐ Ejecutar periódicamente ⓘ
- ☒ GitHub hook trigger for GITScm polling ⓘ
- ☐ Lanzar ejecuciones remotas (ejem: desde 'scripts') ⓘ

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ⓘ

Git ⓘ

Repositories ⓘ

Repository URL ⓘ

https://github.com/UbeimarY/node-project-task.git

Credentials ⓘ

- none -

+ Add

Avanzado ▾

Add Repository

Branches to build ⓘ

Branch Specifier (blank for 'any') ⓘ

/

Add Branch

Navegador del repositorio ⓘ

(Auto)

Additional Behaviours

Save Apply

4. Ejecución del Pipeline

Creación de un

Creación de un

pipeline-n

UbeimarY/nod

localhost:8080/job/pipeline-no...

Jenkins

admin

Desconectar

Panel de Control > pipeline-nodejs-react > Tendencia del tiempo de ejecución

Status

</> Changes

▶ Construir ahora

⚙ Configurar

🗑 Borrar Pipeline

📁 Stages

✎ Rename

🔍 Pipeline Syntax

📄 GitHub Hook Log

Builds

Filter

Today

#1 17:24

Tendencia del tiempo de ejecución

S	Ejecución ↑	Time Since	Duración
...	#1	4.3 Seg	4.3 Seg y contando

Icono: S M L

REST API | Jenkins 2.492.3

Creación de un | Creación de un | pipeline-n | UbeimarY/nod | +

localhost:8080/job/pipeline-no...

Jenkins

Panel de Control > pipeline-nodejs-react > #1

Status #1 (22 abr 2025, 17:24:39) Progreso: [Progress bar] añadir descripción Conservar esta ejecución para siempre

</> Changes

Console Output

Edit Build Information

Timings

Git Build Data

Git Build Data

Pipeline Overview

Pipeline Console

Thread Dump

Pause/resume

Replay

Pipeline Steps

Workspaces

Iniciado por el usuario admin

This run spent 5 Ms waiting in the queue.

Revision: 6ec16bb8ba194c50b9143586ea273a542c2da5
Repository: <https://github.com/UbeimarY/node-project-task.git>
• origin/main

Revision: c3bdc7a8b7a05ee06a0e2b97644af2c961a46fb
Repository: <https://github.com/guswill24/node-project.git>
• refs/remotes/origin/main

</> Sin cambios

Comenzó hace 42 Seg
Compilación siendo ejecutada desde

REST API Jenkins 2.492.3

5. Resultados de Pruebas Unitarias

Creación de un

Creación de un

pipeline-no

UbeimarY/nod

+

<

>

↺

🔖

localhost:8080/job/pipeline-no...

🔍

🔗

🛡️

⚠️

🔧

📄

🌟

🛡️

☰

Jenkins

🔍 🔒 🧑 admin Desconectar

Panel de Control > pipeline-nodejs-react >

Status

Changes

Construir ahora

Configurar

Borrar Pipeline

Stages

Rename

Pipeline Syntax

GitHub Hook Log

✅ pipeline-nodejs-react

añadir descripción

Enlaces permanentes

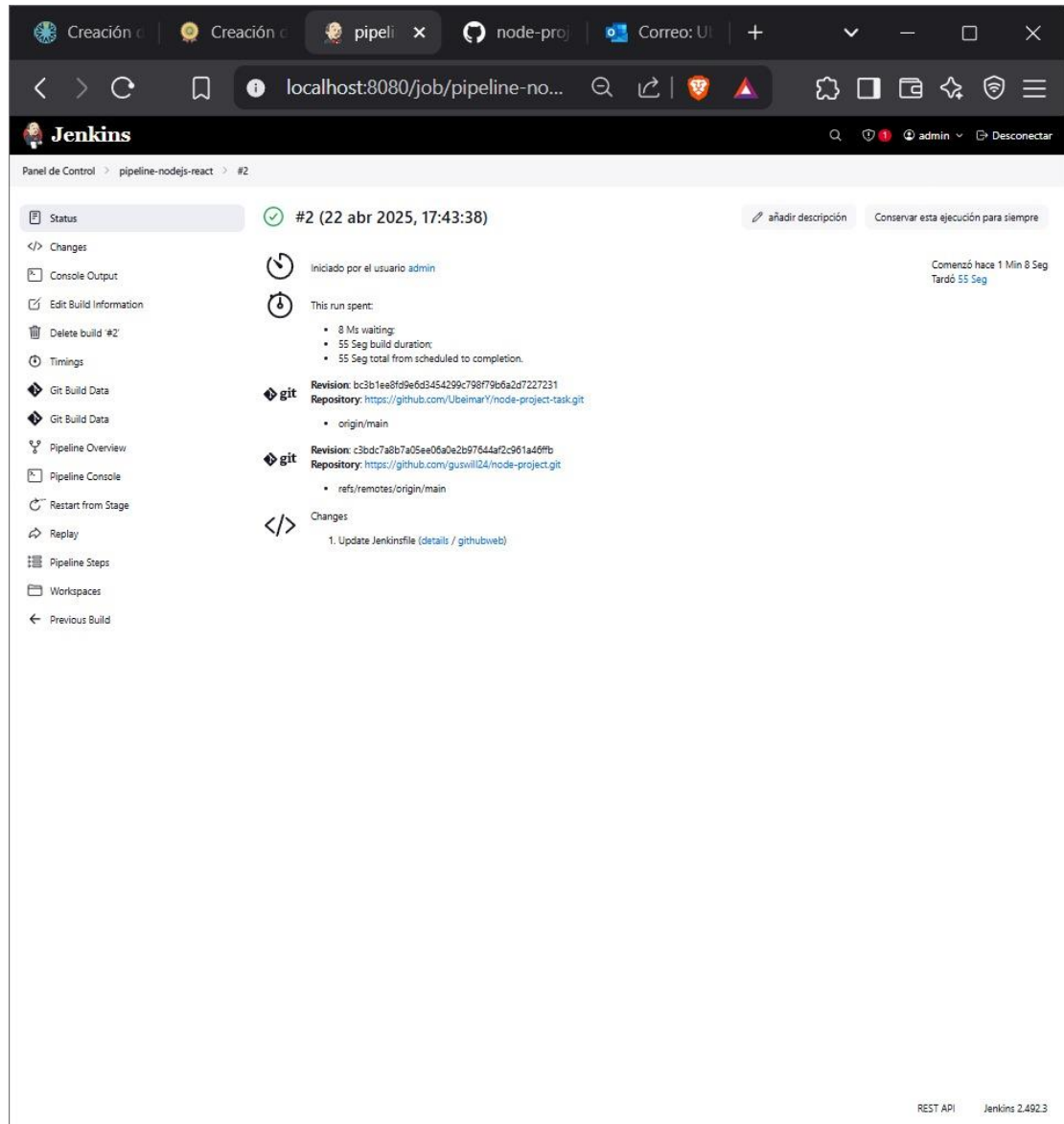
- "Última ejecución (#1) hace 2 Min 50 Seg"
- "Última ejecución estable (#1) hace 2 Min 50 Seg"
- "Última ejecución correcta (#1) hace 2 Min 50 Seg"
- "Last completed build (#1) hace 2 Min 50 Seg"

Builds

Filter

Today

#1 17:24



• Conclusiones

1. En esta actividad se logró implementar un pipeline declarativo completo en Jenkins para un proyecto Node.js con React, permitiendo automatizar la instalación de dependencias, ejecución de pruebas unitarias y compilación del proyecto. Esta práctica refuerza los principios de Integración Continua y DevOps en entornos reales.
2. Automatización Eficiente:
La configuración del pipeline en Jenkins permite automatizar tareas repetitivas como la instalación de dependencias, ejecución de pruebas y build del proyecto, optimizando el tiempo de desarrollo y reduciendo errores humanos.

3. Integración con GitHub:

Al vincular Jenkins directamente con un repositorio GitHub, se asegura que cualquier cambio en el código fuente pueda ser validado de forma inmediata mediante pruebas automatizadas, mejorando la calidad del software.

4. Escalabilidad y Adaptabilidad:

El uso del enfoque declarativo en Jenkins facilita la comprensión y mantenimiento del pipeline, lo que lo hace escalable para proyectos más grandes o para incluir pasos adicionales como despliegues en producción o integración con otras herramientas.