**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'

}

}

stage('Build app') {

steps {

bat 'npm run build'

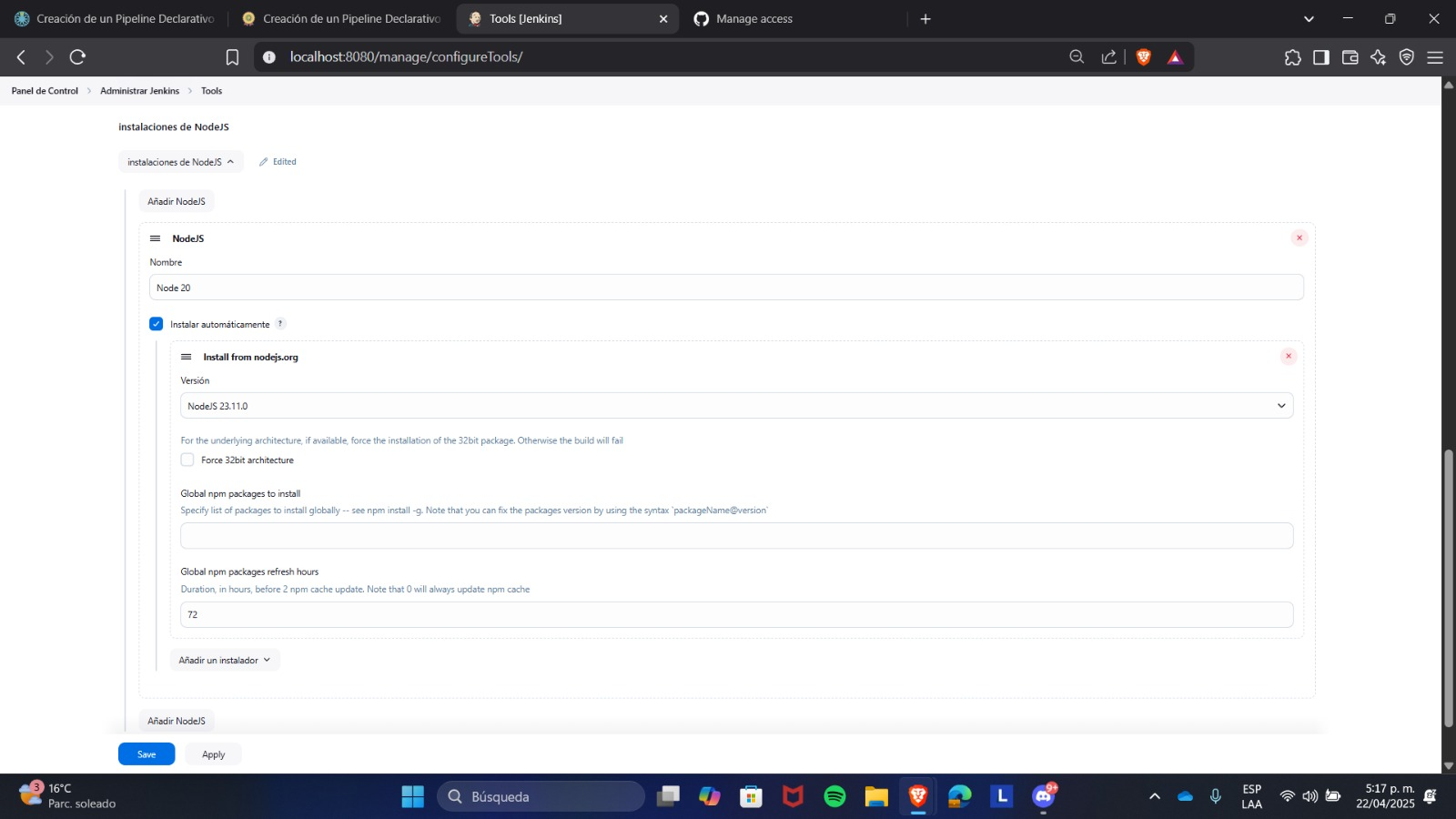
}

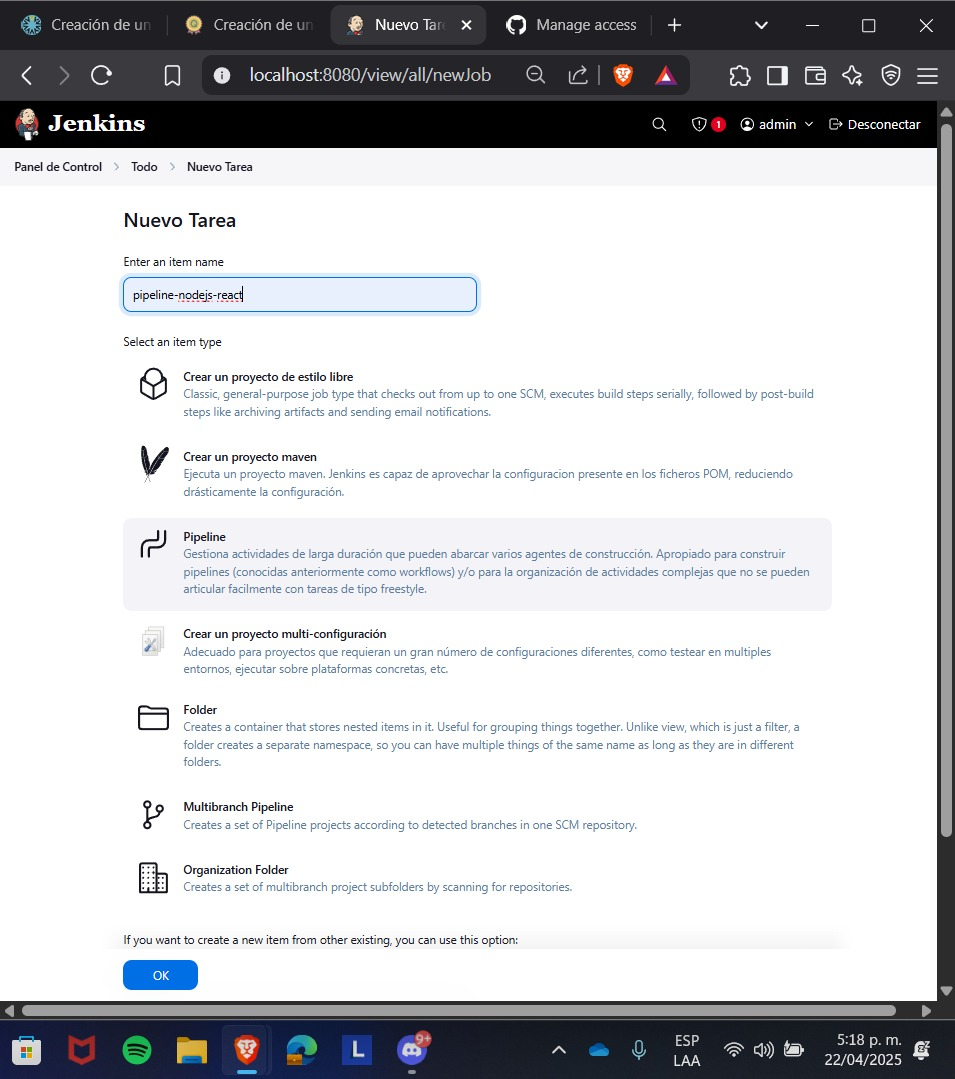
}

}

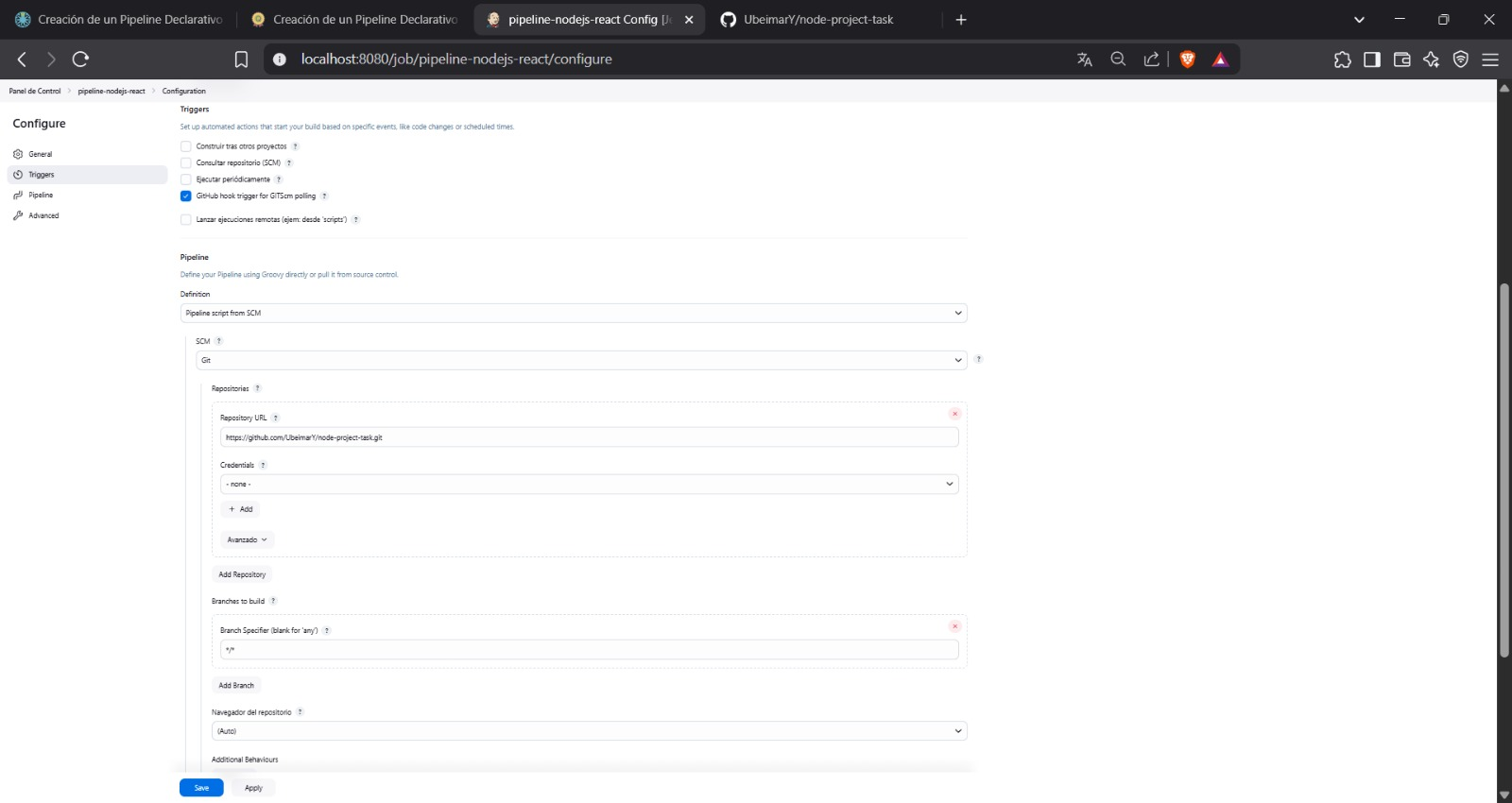
}

* **Evidencias del Proceso**

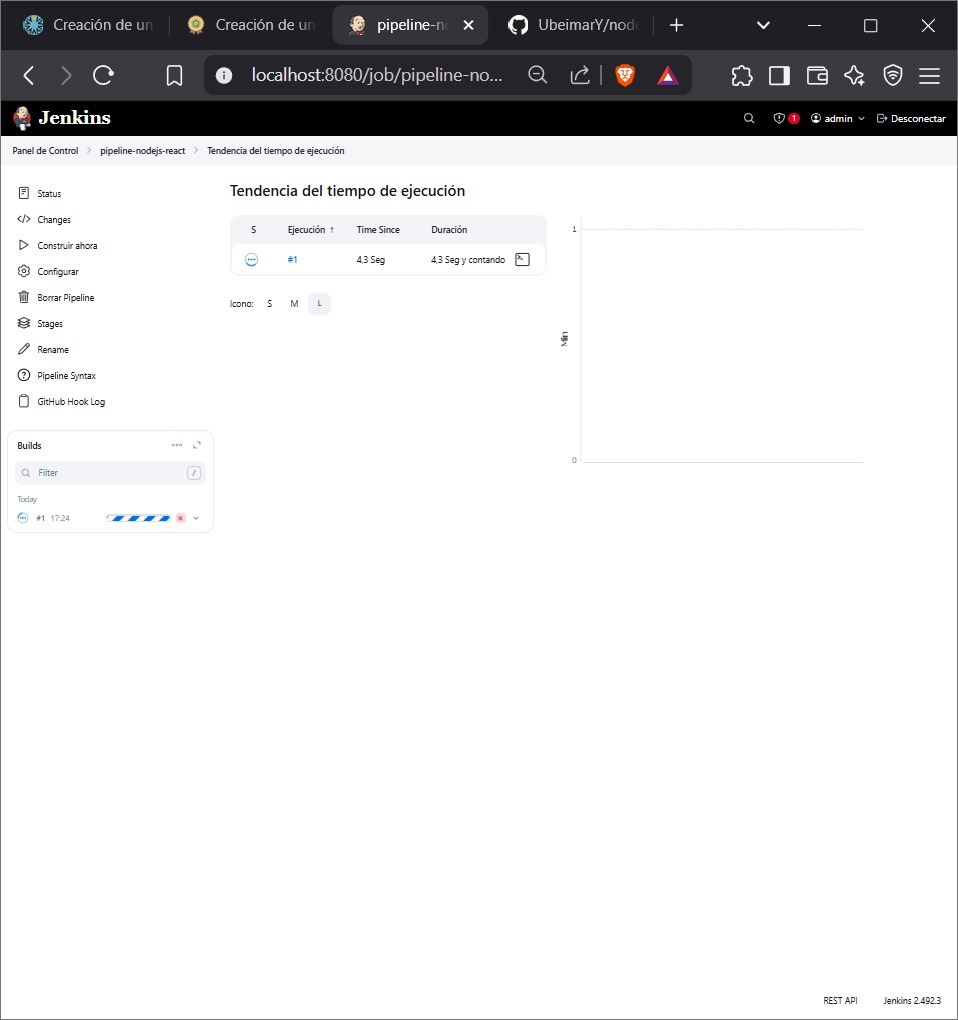
1. **Configuración de Node.js en Jenkins**
2. **Creación del Proyecto Tipo PipeLine**

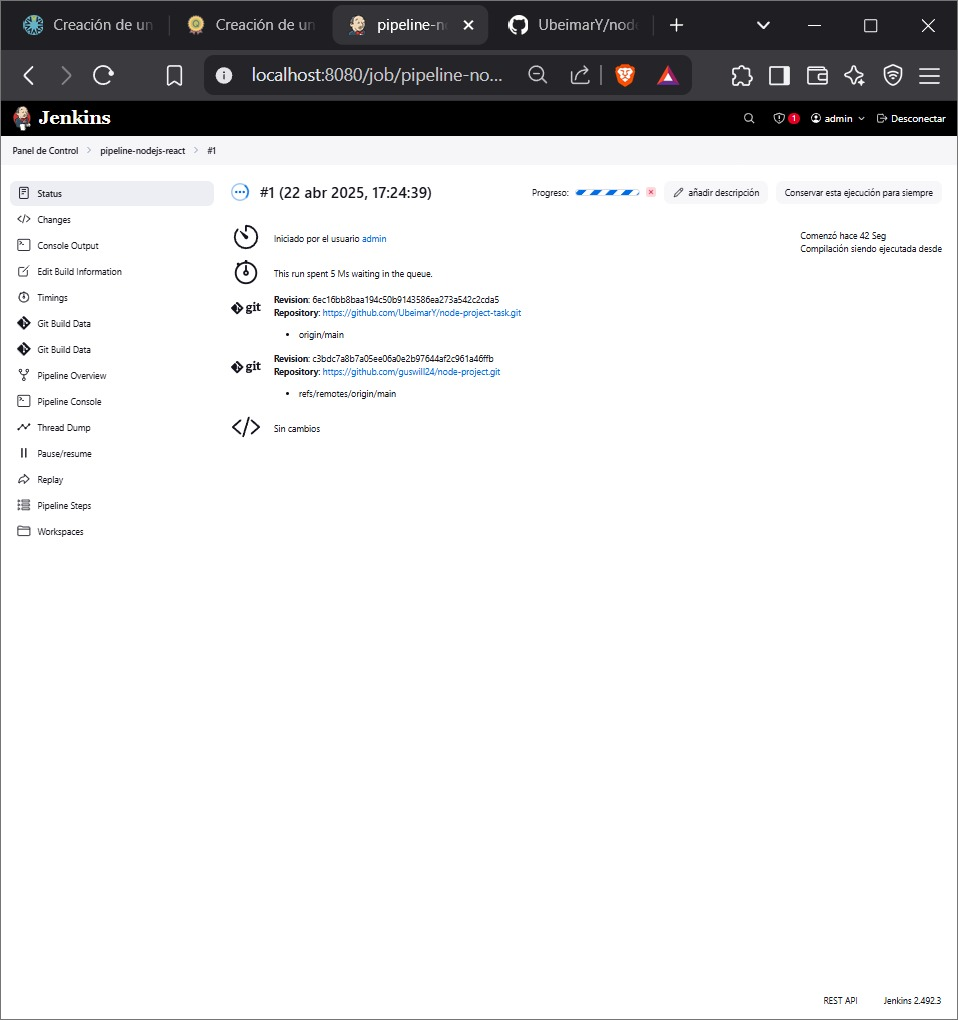


1. **Configuración del SCM desde GitHub**

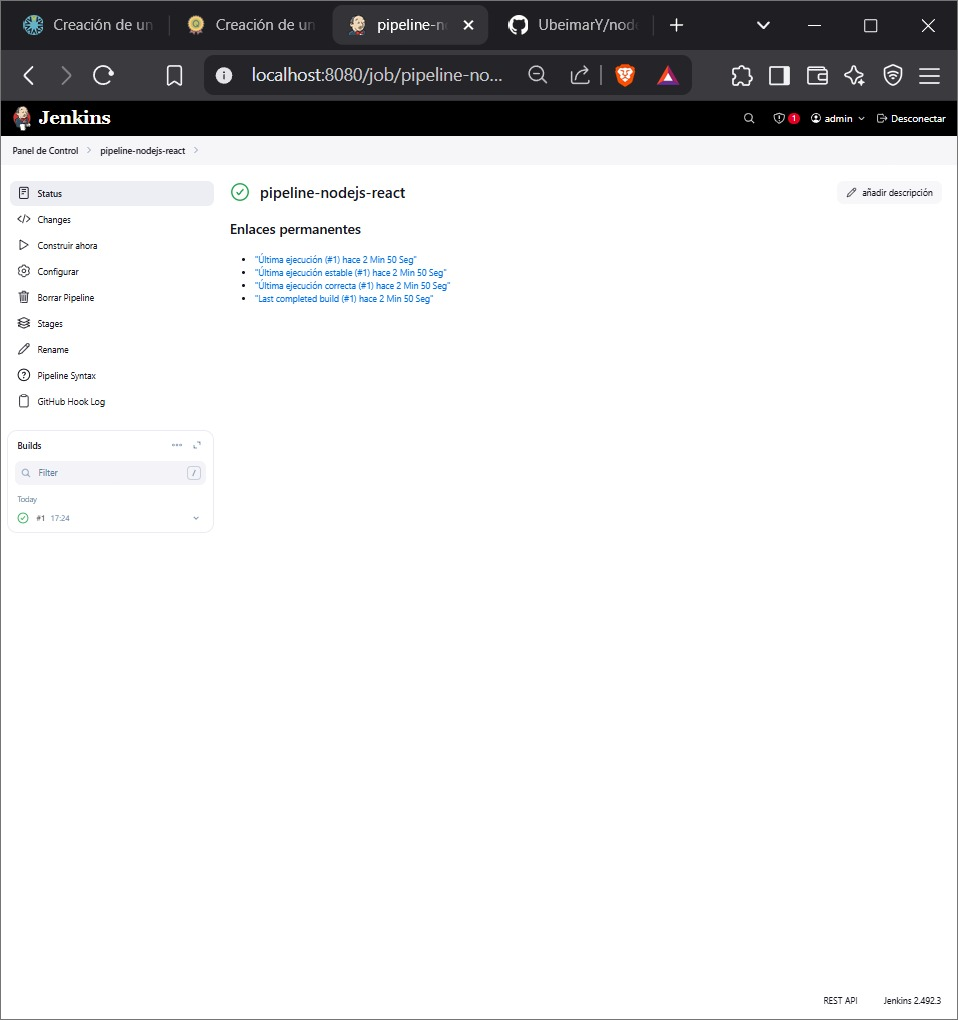


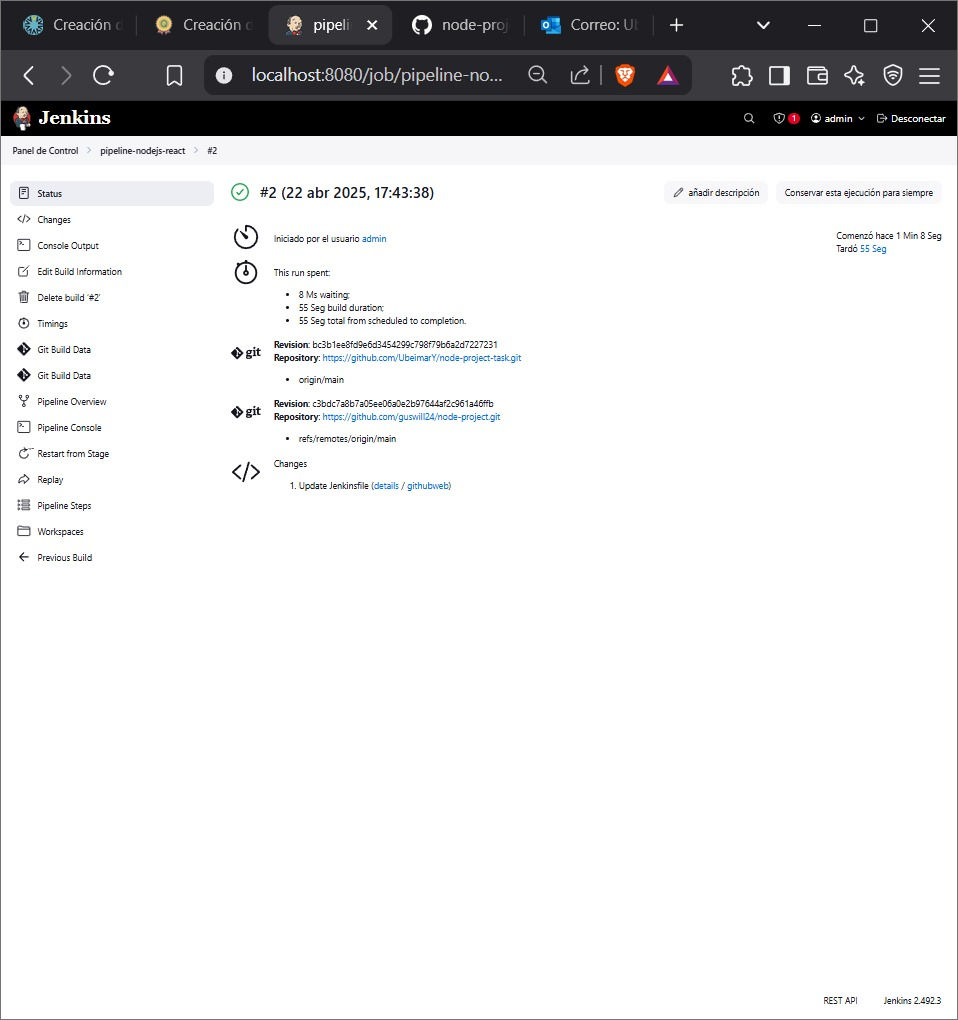
1. **Ejecución del Pipeline**

****

****

1. **Resultados de Pruebas Unitarias**

****

****

* **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.