

Documentación del Pipeline



Automatización CI/CD para React + Vite

Proyecto II

Mora Ariel

Nagua Fernando

Propósito

Este [Jenkinsfile](#) orquesta el **CI/CD** de tu proyecto **React + Vite** en la rama [main](#), automatizando:

- Instalación de dependencias
 - Validación de estilo y ejecución de tests
 - Compilación de la aplicación
 - Construcción y push de la imagen Docker
 - Archivo de artefactos generados
-

Partes principales

1. Declaración del pipeline

2. Agente y herramientas

3. Variables de entorno

4. Stages (etapas)

- Install dependencies
- Lint
- Test
- Build
- Build Docker image
- Push Docker image

5. Post-actions

Detalle del Jenkinsfile

[1] Declaración del pipeline y configuración inicial

```
pipeline {
    agent any
    tools { nodejs 'NodeJS-22' }
    environment {
        IMAGE_NAME = "aidmora/dragonforge:${env.BUILD_NUMBER}"
    }
    stages {
        /* ...etapas aquí... */
    }
    post {
        /* ...acciones post-ejecución... */
    }
}
```

- **pipeline:** Bloque raíz.
 - **agent any:** Usa cualquier nodo disponible.
 - **tools:** Activa la instalación “NodeJS-22” configurada en Jenkins.
 - **environment:** Define `IMAGE_NAME` con el tag `${BUILD_NUMBER}` .
-

② Stage – Install dependencies

```
stage('Install dependencies') {
    steps {
        sh 'npm ci'
    }
}
```

Ejecuta `npm ci` para instalar exactamente las versiones de `package-lock.json`, garantizando reproducibilidad.

③ Stage – Lint

```
stage('Lint') {
    steps {
        sh 'npm run lint'
    }
}
```

Invoca ESLint (`npm run lint`) y detiene el pipeline si hay errores críticos de estilo.

④ Stage – Test

```
stage('Test') {  
    steps {  
        sh 'npm test'  
    }  
}
```

Ejecuta Jest (`npm test`), validando que todas las pruebas unitarias e integraciones pasen.

5 Stage – Build

```
stage('Build') {  
    steps {  
        sh 'npm run build'  
    }  
}
```

Compila y optimiza la aplicación para producción.

Archivos generados:

Al finalizar esta etapa, los archivos de producción estarán disponibles en la carpeta `dist/`.

6 Stage – Build Docker image

```
stage('Build Docker image') {  
    steps {  
        sh "docker build -t ${env.IMAGE_NAME} ."  
    }  
}
```

Construye la imagen Docker usando tu Dockerfile en la raíz, etiquetándola como `aidmora/dragonforge:<BUILD_NUMBER>`.

7 Stage – Push Docker image

```
stage('Push Docker image') {  
    steps {  
        withCredentials([usernamePassword(  
            credentialsId: 'dockerhub-creds',  
            usernameVariable: 'DOCKER_USER',  
            passwordVariable: 'DOCKER_PASS'  
        )]) {  
            sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'  
            sh "docker push ${env.IMAGE_NAME}"  
        }  
    }  
}
```

```
    }
}
}
```

Autentica en Docker Hub de forma segura y sube la imagen recién construida a tu repositorio [aidmora/dragonforge](#).

8 Post-actions

```
post {
  always { archiveArtifacts artifacts: 'dist/**', fingerprint: true }
  success { echo "Build y push completados correctamente." }
  failure { echo "Falló el pipeline. Revisa la consola para detalles." }
}
```

- **always:** Archiva `dist/**` tras cada ejecución.
- **success/failure:** Mensajes finales según el estado del build.

Resultados esperados

Etapa	Resultado típico
Install dependencies	<code>npm ci</code> instala dependencias sin errores.
Lint	ESLint reporta solo warnings o pasa limpio.
Test	Jest muestra todos los tests passed.
Build	Vite genera la carpeta <code>dist/</code> con los assets optimizados.
Build Docker image	Docker construye la imagen <code>aidmora/dragonforge:<BUILD_NUMBER></code> .
Push Docker image	Imagen subida exitosamente a Docker Hub.
Post-actions	Artefactos de <code>dist/</code> disponibles en la sección “Artifacts” de Jenkins.

Evidencias del Funcionamiento

Install dependencies

This screenshot shows the Jenkins pipeline interface for the 'Install dependencies' step. The step has three stages: 'Use a tool from a predefined Tool Installation NodeJS-22', 'Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.', and 'npm ci'. The 'npm ci' stage is expanded, showing its command and output. The output includes npm warnings about deprecated modules like 'inflight@1.0.6' and 'glob@7.2.3', statistics about packages audited, and a list of vulnerabilities. It also provides instructions for addressing issues and running audit commands.

```
0 + npm ci
1 npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
2 npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
3
4 added 754 packages, and audited 755 packages in 13s
5
6 115 packages are looking for funding
7 run `npm fund` for details
8
9 4 vulnerabilities (3 low, 1 critical)
10
11 To address all issues, run:
12   npm audit fix
13
14 Run `npm audit` for details.
```

Lint

This screenshot shows the Jenkins pipeline interface for the 'Lint' step. Similar to the previous step, it has three stages: 'Use a tool from a predefined Tool Installation NodeJS-22', 'Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.', and 'npm run lint'. The 'npm run lint' stage is expanded, showing its command and output. The output lists ESLint warnings across several files, including 'MenuBar.jsx', 'AuthContext.jsx', and 'MisEjerciciosPage.jsx', with specific line numbers and descriptions of the errors.

```
0 + npm run lint
1
2 > dragon-forge-client@0.0.0 lint
3 > eslint .
4
5
6 /var/jenkins_home/workspace/Pipeline-Proyecto-Construccion/src/components/MenuBar.jsx
7 22:6 warning React Hook useEffect has a missing dependency: 'setOpen'. Either include it or remove the dependency array react-hooks/exhaustive-deps
8
9 /var/jenkins_home/workspace/Pipeline-Proyecto-Construccion/src/contexts/AuthContext.jsx
10 6:14 warning Fast refresh only works when a file only exports components. Move your React context(s) to a separate file react-refresh/only-export-components
11
12 /var/jenkins_home/workspace/Pipeline-Proyecto-Construccion/src/pages/MisEjerciciosPage.jsx
13 36:6 warning React Hook useEffect has a missing dependency: 'setError'. Either include it or remove the dependency array react-hooks/exhaustive-deps
14
15 ✘ 3 problems (0 errors, 3 warnings)
```

Test

Test

3.9s Comenzó hace 1d 7h Jenkins

- Use a tool from a predefined Tool Installation NodeJS-22 > 33ms
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. > 43ms
- npm test > 3.7s

```

0 + npm test
1
2 > dragon-forge-client@0.0.0 test
3 > node --experimental-vm-modules node_modules/jest/bin/jest.js
4
5 (node:2337) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
6 (Use `node --trace-warnings ...` to show where the warning was created)
7 (node:2342) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
8 (Use `node --trace-warnings ...` to show where the warning was created)
9 PASS src/services/_test/_ejercicios.unit.test.js
10 PASS src/services/_test/_usuarios.unit.test.js
11 (node:2320) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
12 (Use `node --trace-warnings ...` to show where the warning was created)
13 (node:2327) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
14 (Use `node --trace-warnings ...` to show where the warning was created)
15 PASS src/services/_test/_Formulario.unit.test.jsx
16 PASS src/services/_test/_ejercicioslist.integracion.test.jsx
17 (node:2319) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
18 (Use `node --trace-warnings ...` to show where the warning was created)
19 PASS src/services/_test/_EjercicioForm.integracion.test.jsx
20 ● Console
```

```

## Build

Build

3.4s Comenzó hace 1d 7h Jenkins

- Use a tool from a predefined Tool Installation NodeJS-22 > 45ms
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. > 61ms
- npm run build > 3,2s

```

0 + npm run build
1
2 > dragon-forge-client@0.0.0 build
3 > vite build
4
5 vite v6.3.5 building for production...
6 transforming...
7 ✓ 427 modules transformed.
8 [esbuild css minify]
9 ▲ [WARNING] Unexpected "/" [css-syntax-error]
10
11 <stdin>:1829:1:
12 1829 | /* .contenidoEjercicios {
13 | ^
14
15
16 rendering chunks...
17 computing gzip size...
18 dist/index.html 0.79 kB | gzip: 0.50 kB
19 dist/assets/espalda-BLY8MQ5k.jpg 35.27 kB
20 dist/assets/hipT-Dr43-o0G.jpeg 84.74 kB
21 dist/assets/hombro-BcYl1g0.jpeg 170.66 kB
22 dist/assets/pierna-ACKQswDk.jpg 247.03 kB
23 dist/assets/triceps-C3SLseXG.jpg 317.67 kB
24 dist/assets/pecho-oNcwTYtJ.png 556.75 kB
25 dist/assets/DragonForge-su556shV.png 1,223.43 kB
26 dist/assets/DragonForgePerfill-Ce5LzkhN.png 1,710.45 kB
27 dist/assets/DragonFogeExercises-Df6hvac0.png 1,872.52 kB
28 dist/assets/index-CCQJb1mj.css 265.98 kB | gzip: 37.53 kB
29 dist/assets/index-BDm-y5NM.js 299.96 kB | gzip: 97.58 kB
30 ✓ built in 2.76s
```

```

Build Docker image

The screenshot shows a Jenkins build log for the job 'Build Docker image'. The log details the execution of a 'docker build -t aidmora/dragonforge:4.' command. The log output is as follows:

```
0 + docker build -t aidmora/dragonforge:4 .
1 #1 [internal] load build definition from Dockerfile
2 #2 sha256:d306b1fcf7b18f385bb8910baa1b19b2f57679d1f2546c43c5145cd9950e326
3 #1 transferring dockerfile: 316B done
4 #1 DONE 0.0s
5
6 #2 [internal] load .dockerignore
7 #2 sha256:5c89971541b97faa1b003a0496d183cb484dc971dda2dc0eefb6941e03604922
8 #2 transferring context: 2B done
9 #2 DONE 0.1s
10
11 #4 [internal] load metadata for docker.io/library/node:18-alpine
12 #4 sha256:c7ddcdb449347fb14fd88eab6b6351683fc669399d296a38e5a23e64c4ea7112
13 #4 DONE 1.4s
14
15 #3 [internal] load metadata for docker.io/library/nginx:stable-alpine
16 #3 sha256:b20e08fa637e5be7e9e0f8e5fa86fcf181c0b743f9fd966a3577efddcf5491cc
17 #3 DONE 1.5s
18
19 #10 [internal] load build context
20 #10 sha256:fbeb34530ee511cadd6081387f56d4a83b66c9069f334952f274020157e12bb1
21 #10 DONE 0.0s
22
23 #12 [builder 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
24 #12 sha256:51b980bfc9d7b82b82d0a5373e73b34efafde66ce7752d1f70f7f9482898b66f
```

Push Docker image

Push Docker image

⌚ 11s ⌚ Comenzó hace 1d 7h 🛡 Jenkins ⚙

- Use a tool from a predefined Tool Installation NodeJS-22 > 73ms ⚙
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. > 77ms ⚙
- echo \$DOCKER_PASS | docker login -u \$DOCKER_USER --password-stdin > 1,2s ⚙
- docker push aidmora/dragonforge:4 > 9,2s ⚙

```

0 + docker push aidmora/dragonforge:4
1 The push refers to repository [docker.io/aidmora/dragonforge]
2 462dcdb3654c: Preparing
3 0c68f9363bb2: Preparing
4 365f05739a0b: Preparing
5 09a78e1c8457: Preparing
6 aabf53a58c5a: Preparing
7 39b5112a4779: Preparing
8 243fb2529a82: Preparing
9 3c0990fff3868: Preparing
10 daf8a96d7e43: Preparing
11 7003d23cc217: Preparing
12 243fb2529a82: Waiting
13 3c0990fff3868: Waiting
14 7003d23cc217: Waiting
15 39b5112a4779: Waiting
16 daf8a96d7e43: Waiting
17 aabf53a58c5a: Mounted from library/nginx
18 365f05739a0b: Mounted from library/nginx
...

```

[Explore / aidmora / dragonforge](#)

aidmora/dragonforge By aidmora · Updated 1 day ago [Manage Repository](#)

0 3

[Overview](#) [Tags](#)

Sort by [Newest](#) [Filter tags](#)

TAG	Digest	OS/ARCH	Last pull	Compressed size
4	fa54e10c1bad	linux/amd64	1 day	25.98 MB

Post Actions

Post Actions

⌚ 0,23s ⌚ Comenzó hace 1d 7h 🛡 Jenkins ⚙

- Archive the artifacts > 0,15s ⚙
- Archiving artifacts
- Recording fingerprints
- Build y push completados correctamente. > 30ms ⚙
- Build y push completados correctamente.

Pipeline-Proyecto-Construccion - Stage View

Pipeline-Proyecto-Construcción - Stage View



💻 Desarrolladores e Información

- **Ariel** (ariel.mora@epn.edu.ec)
- **Fernando** (fernando.nagua@epn.edu.ec)

Repositorio:

github.com/Aidmora/DragonForge
