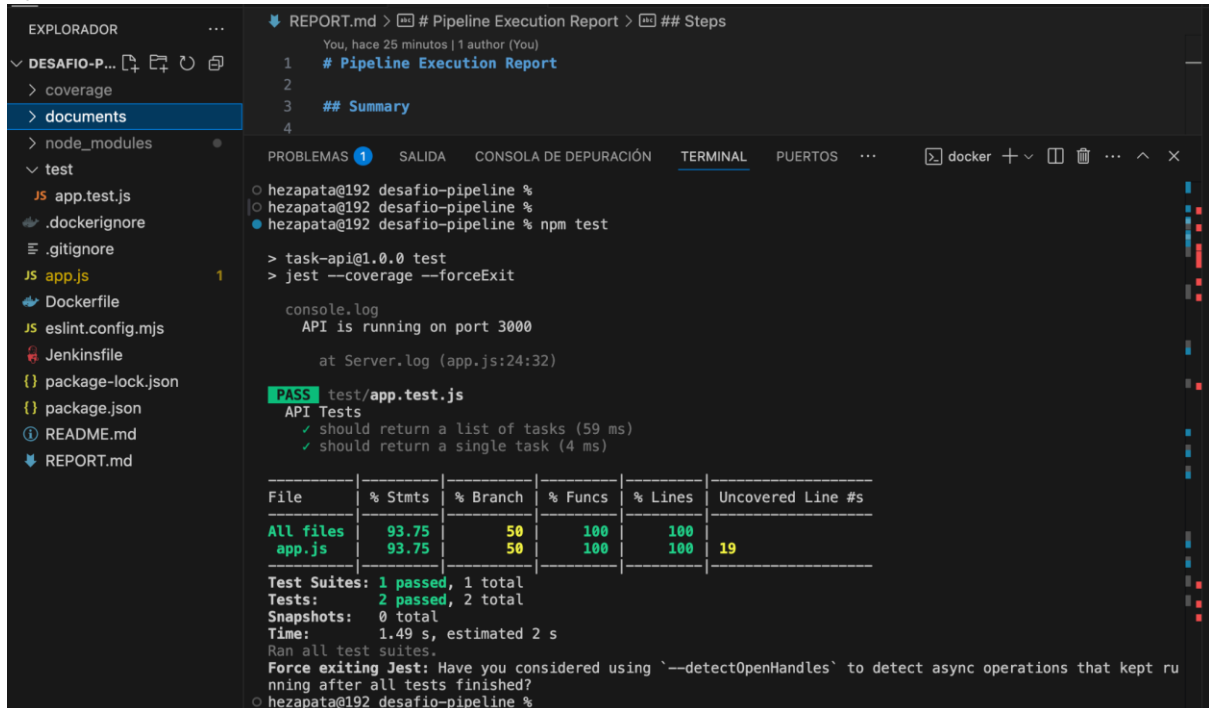


## Ejecución de Pruebas



The screenshot shows the VS Code interface with a file explorer on the left and a terminal on the right. The file explorer shows a project structure with files like `app.test.js`, `.dockerignore`, `.gitignore`, `app.js`, `Dockerfile`, `eslint.config.mjs`, `Jenkinsfile`, `package-lock.json`, `package.json`, `README.md`, and `REPORT.md`. The terminal shows the output of a pipeline execution report, including a summary, a table of test results, and a list of test suites.

```
REPORT.md > ## Pipeline Execution Report > ## Steps
You, hace 25 minutos | 1 author (You)
# Pipeline Execution Report
## Summary
1
2
3
4
```

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS ...

```
hezapata@192 desafio-pipeline %
hezapata@192 desafio-pipeline %
hezapata@192 desafio-pipeline % npm test

> task-api@1.0.0 test
> jest --coverage --forceExit

console.log
  API is running on port 3000

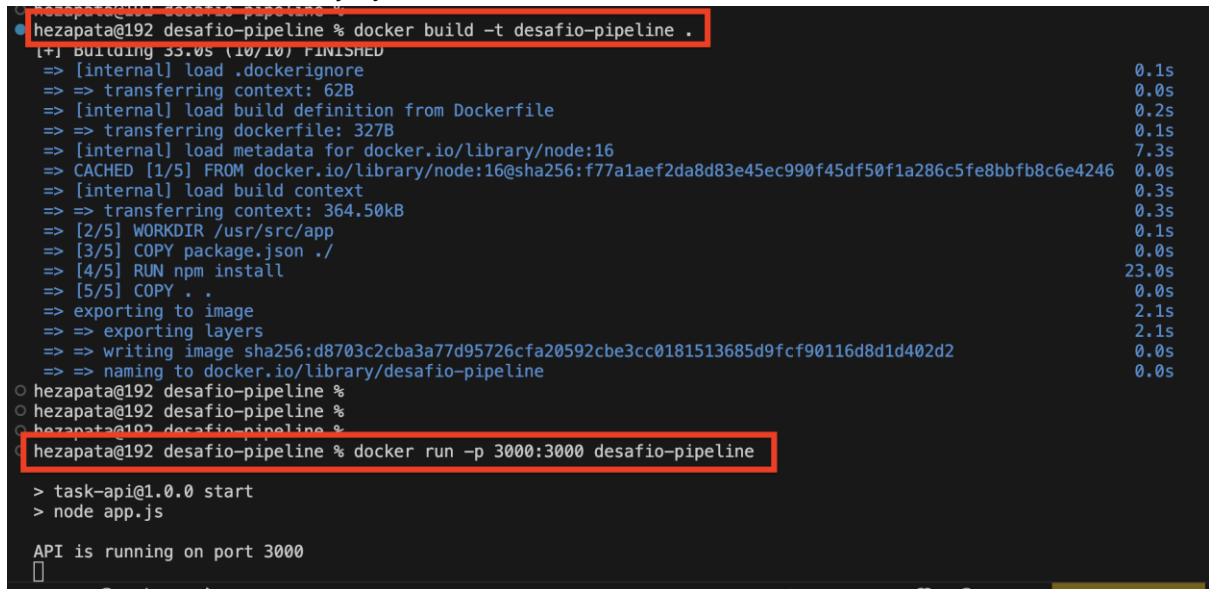
at Server.log (app.js:24:32)

PASS test/app.test.js
  API Tests
    ✓ should return a list of tasks (59 ms)
    ✓ should return a single task (4 ms)

File | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files | 93.75 | 50 | 100 | 100 | 
app.js | 93.75 | 50 | 100 | 100 | 19

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 1.49 s, estimated 2 s
Ran all test suites.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
hezapata@192 desafio-pipeline %
```

## Creación del contenedor y ejecución



The screenshot shows a terminal window with the following commands and output:

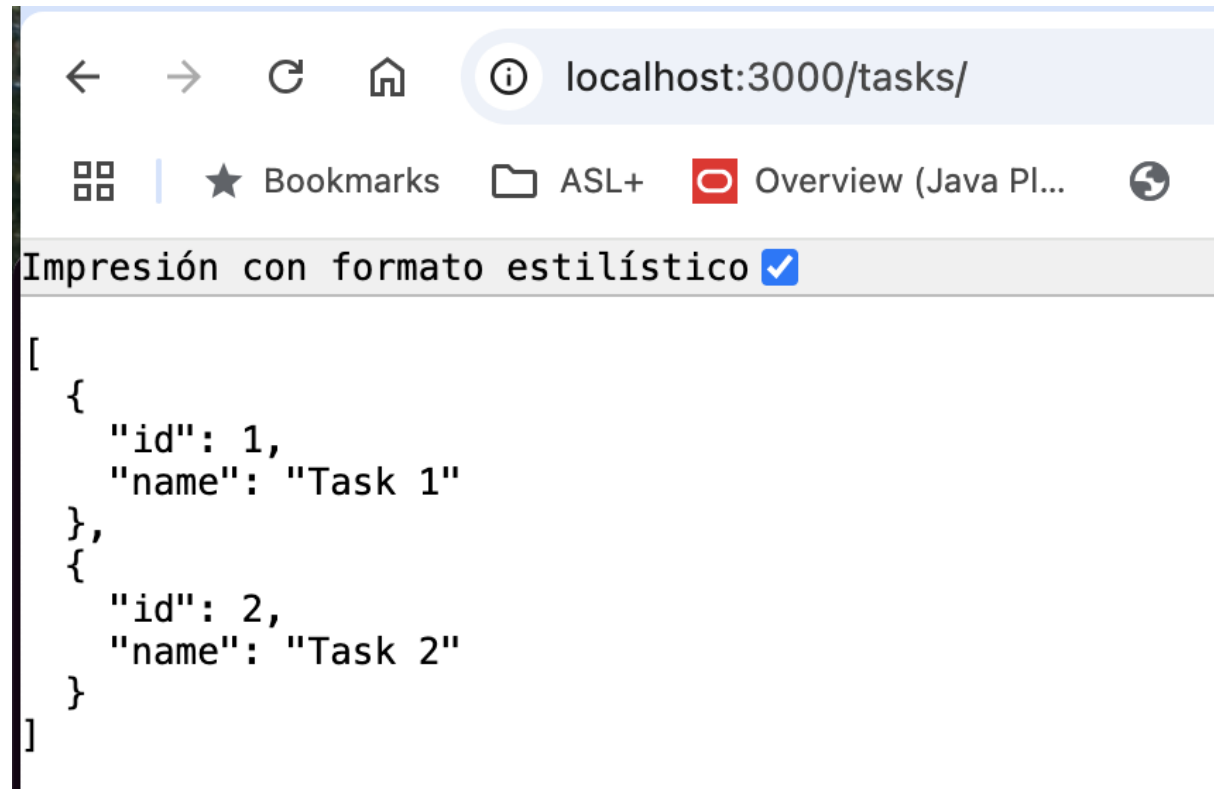
```
hezapata@192 desafio-pipeline % docker build -t desafio-pipeline .
[+] Building 33.0s (10/10) FINISHED
=> [internal] load .dockerignore 0.1s
=> => transferring context: 62B 0.0s
=> [internal] load build definition from Dockerfile 0.2s
=> => transferring dockerfile: 327B 0.1s
=> [internal] load metadata for docker.io/library/node:16 7.3s
=> CACHED [1/5] FROM docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246 0.0s
=> [internal] load build context 0.3s
=> => transferring context: 364.50kB 0.3s
=> [2/5] WORKDIR /usr/src/app 0.1s
=> [3/5] COPY package.json ./ 0.0s
=> [4/5] RUN npm install 23.0s
=> [5/5] COPY . . 0.0s
=> exporting to image 2.1s
=> => exporting layers 2.1s
=> => writing image sha256:d8703c2cba3a77d95726cfa20592cbe3cc0181513685d9fcf90116d8d1d402d2 0.0s
=> => naming to docker.io/library/desafio-pipeline 0.0s

hezapata@192 desafio-pipeline %
hezapata@192 desafio-pipeline %
hezapata@192 desafio-pipeline %
hezapata@192 desafio-pipeline % docker run -p 3000:3000 desafio-pipeline

> task-api@1.0.0 start
> node app.js

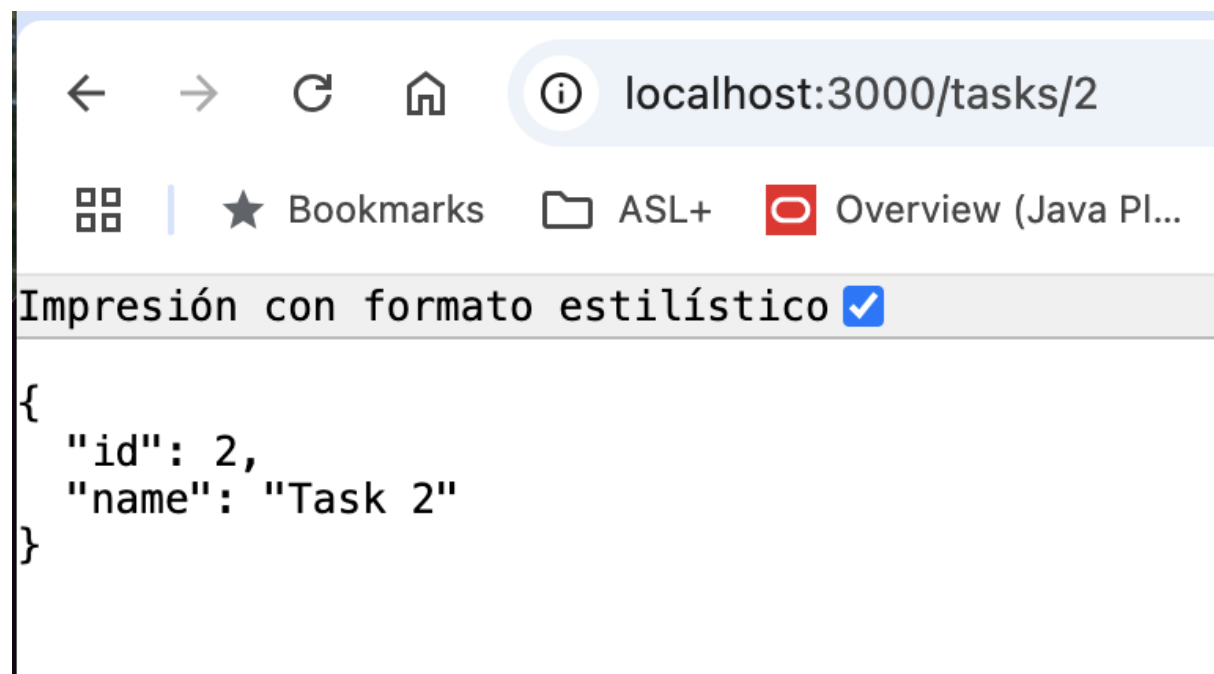
API is running on port 3000
```

Pruebas de las rutas creadas



A screenshot of a web browser window. The address bar shows 'localhost:3000/tasks/'. Below the address bar, there are navigation icons and a search bar. The main content area displays the response of a GET request to the '/tasks/' endpoint. The response is a JSON array containing two task objects. A checkbox labeled 'Impresión con formato estilístico' is checked.

```
[  
  {  
    "id": 1,  
    "name": "Task 1"  
  },  
  {  
    "id": 2,  
    "name": "Task 2"  
  }  
]
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/tasks/2'. Below the address bar, there are navigation icons and a search bar. The main content area displays the response of a GET request to the '/tasks/2' endpoint. The response is a JSON object representing a single task. A checkbox labeled 'Impresión con formato estilístico' is checked.

```
{  
  "id": 2,  
  "name": "Task 2"  
}
```

# Configuración del Pipeline en Jenkins

Panel de Control

>

pipelinedesafio

>

Configuration

Configure

General

Triggers

Pipeline

Advanced

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/hzapata9/pipeline-desafio.git

Credentials

- none -

+ Add

Avanzado

Add Repository

Branches to build

Branch Specifier (blank for 'any')

\*/main

Add Branch

Configure

General

Triggers

Pipeline

Advanced

Branches to build

Branch Specifier (blank for 'any')

\*/main

Add Branch

Navegador del repositorio

(Auto)

Additional Behaviours

Añadir

Script Path

Jenkinsfile

☒ Lightweight checkout

Pipeline Syntax

Advanced

Avanzado

Save

Apply

## Resultado de la ejecución del pipeline

Panel de Control > pipelinedesafio > #1

Status

</> Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Restart from Stage

Replay

Pipeline Steps

Workspaces

Salida de consola

Download

Copy

View as plain text

Lanzada por el usuario AdminJenkins

Obtained Jenkinsfile from git <https://github.com/hzapata9/pipeline-desafio.git>

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins in /Users/hezapata/.jenkins/workspace/pipelinedesafio

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

No credentials specified

Cloning the remote Git repository

Cloning repository <https://github.com/hzapata9/pipeline-desafio.git>

> git init /Users/hezapata/.jenkins/workspace/pipelinedesafio # timeout=10

Fetching upstream changes from <https://github.com/hzapata9/pipeline-desafio.git>

> git --version # timeout=10

> git --version # 'git version 2.39.5 (Apple Git-154)'

> git fetch --tags --force --progress -- <https://github.com/hzapata9/pipeline-desafio.git>

+refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git config remote.origin.url <https://github.com/hzapata9/pipeline-desafio.git> # timeout=10

> git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Avoid second fetch

> git rev-parse refs/remotes/origin/main^{commit} # timeout=10

Checking out Revision 7beeb37f6e5586b350f22570bc01958155dca954 (refs/remotes/origin/main)

> git config core.sparsecheckout # timeout=10

> git checkout -f 7beeb37f6e5586b350f22570bc01958155dca954 # timeout=10

Commit message: "Test pass"

First time build. Skipping changelog.

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

```
[Pipeline] echo
🚀 Clonando el repositorio...
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /Users/hezapata/.jenkins/workspace/pipelinedesafio/.git #
timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hzapata9/pipeline-desafio.git # timeout=10
Fetching upstream changes from https://github.com/hzapata9/pipeline-desafio.git
> git --version # timeout=10
> git --version # 'git version 2.39.5 (Apple Git-154)'
> git fetch --tags --force --progress -- https://github.com/hzapata9/pipeline-desafio.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 7beeb37f6e5586b350f22570bc01958155dca954 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7beeb37f6e5586b350f22570bc01958155dca954 # timeout=10
Commit message: "Test pass"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Install)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
🔧 Instalando dependencias...
[Pipeline] sh
+ whoami
hezapata
[Pipeline] sh
+ which npm
/usr/local/bin/npm
[Pipeline] sh
+ npm install
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.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
```

```
+ npm install
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.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported

added 509 packages, and audited 510 packages in 9s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[Pipeline] sh
+ npm run build

> task-api@1.0.0 build
> echo 'No hay proceso de compilación en este proyecto'

No hay proceso de compilación en este proyecto
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
✓ Ejecutando pruebas...
[Pipeline] sh
+ npm run test

> task-api@1.0.0 test
> jest --coverage --forceExit

console.log
  API is running on port 3000

    at Server.log (app.js:24:32)

PASS test/app.test.js
```

## API Tests

- ✓ should return a list of tasks (111 ms)
- ✓ should return a single task (12 ms)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	93.75	50	100	100	
app.js	93.75	50	100	100	19

Test Suites: 1 passed, 1 total

Tests: 2 passed, 2 total

Snapshots: 0 total

Time: 2.118 s

Ran all test suites.

Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy)

[Pipeline] script

[Pipeline] {

[Pipeline] echo

🚀 Desplegando aplicación...

[Pipeline] sh

+ npm start

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Declarative: Post Actions)

[Pipeline] echo

✅ Pipeline completado con éxito

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS