

Test Driven Development & Continuous Integration

1.- Install nodejs <https://nodejs.org/es/download/releases/>

```
$ node --version
```

```
$ npm --version
```

2.- Install react

```
$ npm install -g create-react-app
```

3.- create app

```
$ npx create-react-app react-demo
```

```
$ cd react-demo
```

4.- Test app

```
$ npm start
```

```
$ ng run build
```

```
$ ng test
```

4.- Create a repository <https://github.com> (react-demo), configure repository

```
$ git init
```

```
$ git remote add origin <url>
```

```
....
```

```
....
```

4.- Edit README.md, add # v0.1 tag

```
$ vi README.md
```

React Demo

```
# v0.1 tag
```

```
...
```

```
...
```

```
:wq
```

5.- Commit, push changes

```
$ git add .
```

```
$ git commit -m "update README maps with v0.1 tag"
```

```
$ git push -u origin master
```

6.- Tag base code with v0.1

```
$ git tag -a v0.1 -m "my maps version 0.1"
```

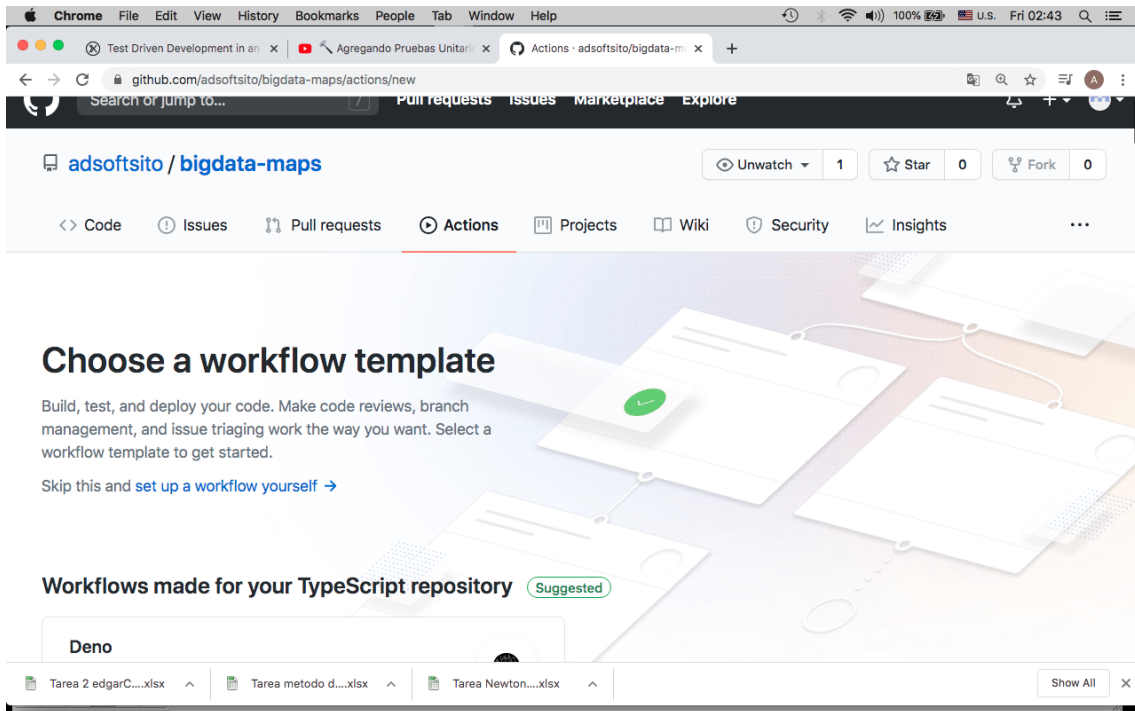
```
$ git tag
```

```
$ git show v0.1
```

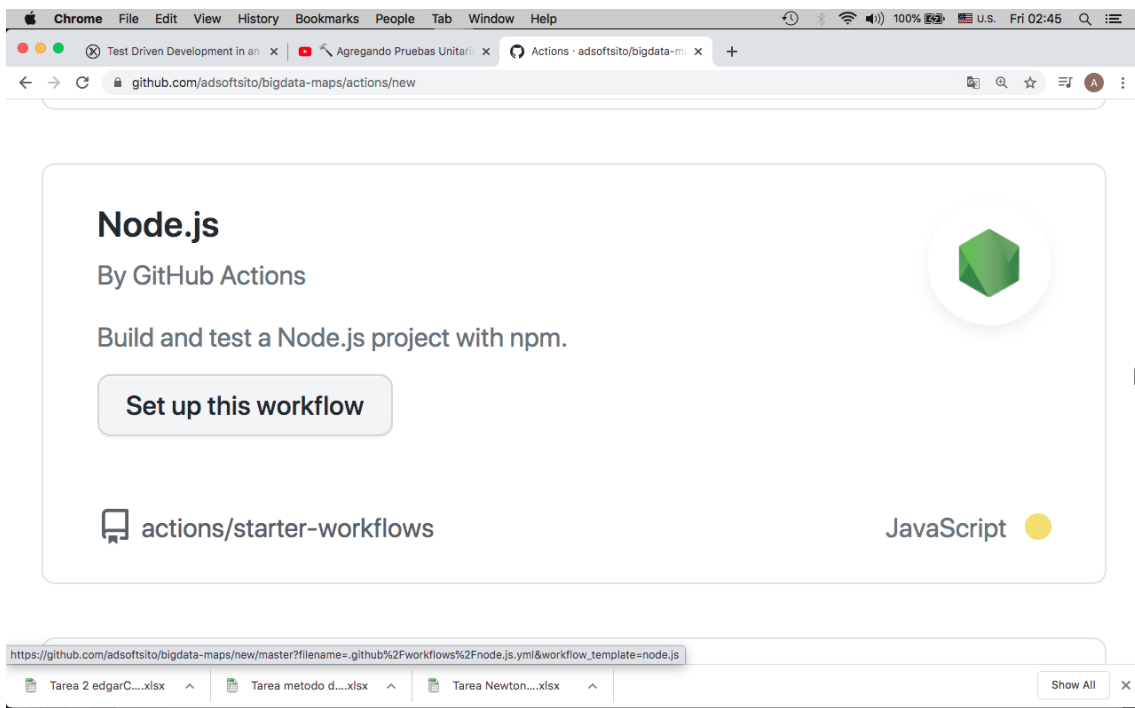
7.- Publish tag

\$ *git push origin v0.1*

9.- Configure CI in Github Actions



Select nodejs template



Edit nodejs template :

```
name:
Node.js CI

on:
  push:
    branches: [ master, hotfixes, release, develop ]
  pull_request:
    branches: [ master, hotfixes, release, develop ]

jobs:
  build:

    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [10.x]

    steps:
      - uses: actions/checkout@v2
      - name: Use Node.js ${ matrix.node-version }
        uses: actions/setup-node@v1
        with:
          node-version: ${ matrix.node-version }
      - run: npm ci
      - run: npm run test
      - run: npm run build --if-present
```

8.- create branch hotfixes, release and develop

```
$ git checkout master
$ git branch hotfixes
$ git checkout hotfixes
```

```
$ vi HOTFIXES.md
# hotfixes maps here
:wq
```

```
$ git add .
$ git commit -m "init hotfixes branch"
$ git push -u origin hotfixes
```

#####

```
$ git checkout master
$ git branch release
$ git checkout release
```

```
$ vi RELEASE.md
# release maps here
:wq
```

```
$ git add .
$ git commit -a -m "init release branch"
$ git push -u origin release
```

#####

```
$ git checkout master
$ git branch develop
$ git checkout develop
$ vi DEVELOP.md
# develop maps here
:wq
```

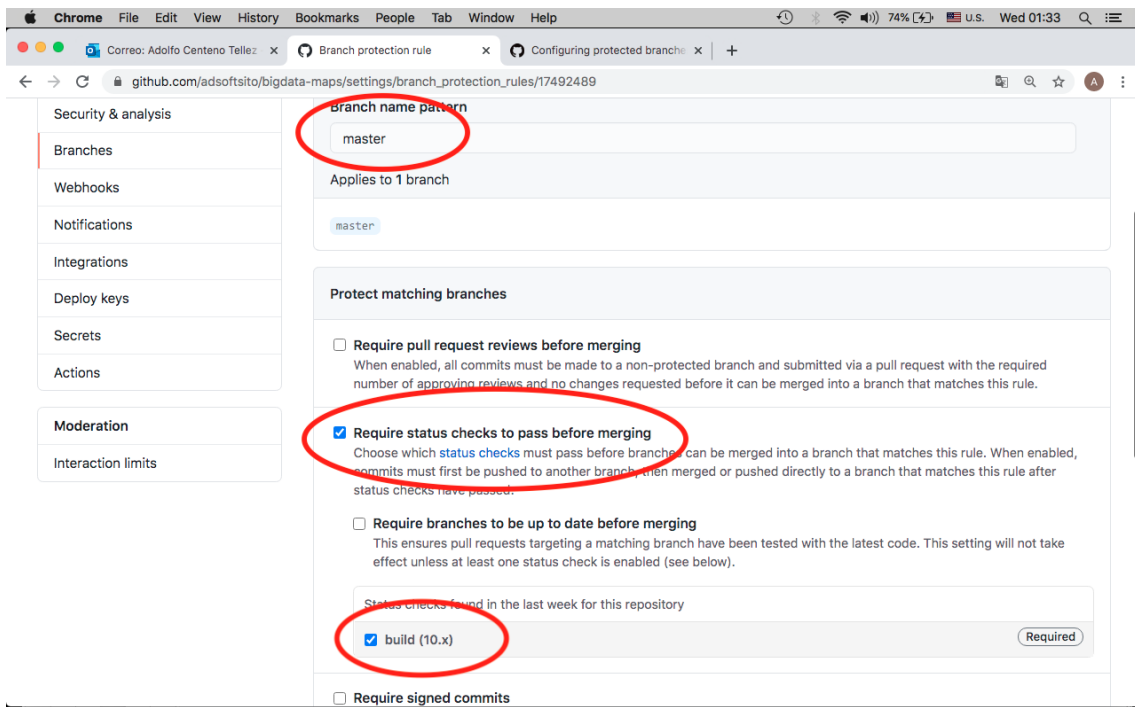
```
$ git add .
```

```
$ git commit -a -m "init develop branch"
$ git push -u origin develop
```

10.- Configure branches protection

The screenshot shows the GitHub repository settings page for 'adsoftsito/bigdata-maps'. The page is viewed in a Chrome browser. The top navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Settings' tab is selected. On the left sidebar, the 'Branches' section is highlighted. The main content area is titled 'Branch protection rules'. It explains that the default branch is 'master' and provides an 'Add rule' button, which is circled in red. Below this, there is a section for defining branch protection rules, with a table showing a rule for 'master' that currently applies to 1 branch. The table has columns for the branch name and the number of branches it applies to. The 'master' branch is listed with '1' in the second column. There are 'Edit' and 'Delete' buttons for this rule. At the bottom, there are 'Previous' and 'Next' navigation buttons. The URL in the address bar is 'https://github.com/adsoftsito/bigdata-maps/settings/branch_protection_rules/new'.

- Create master rule



- Repeat for hotfixes, release and develop branch

