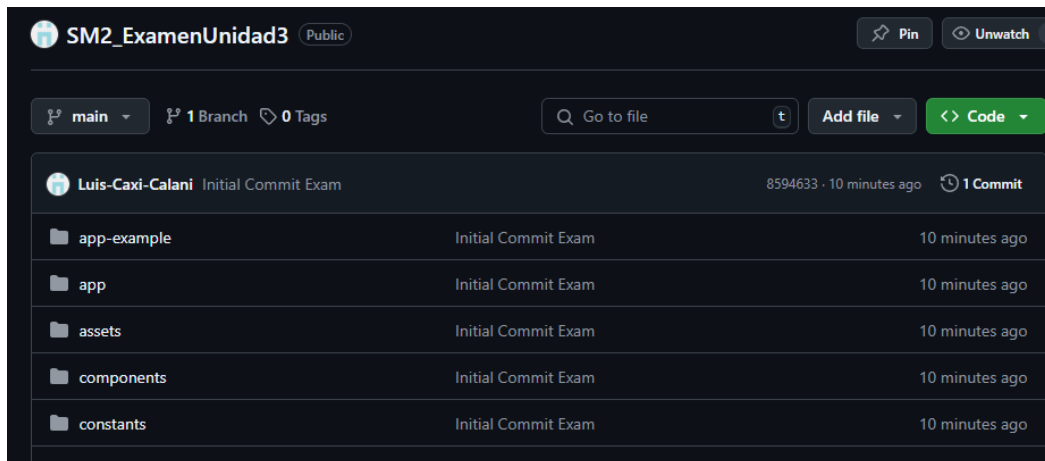


Examen Unidad III – Práctica

Nombre: Luis Eduardo Caxi Calani

URL-REPOSITORIO: https://github.com/Luis-Caxi-Calani/SM2_ExamenUnidad3

1. Crear un repositorio público en github llamado "SM2_ExamenUnidad3".



2. Clona el repositorio de tu proyecto movil del curso, al repositorio "SM2_ExamenUnidad3".

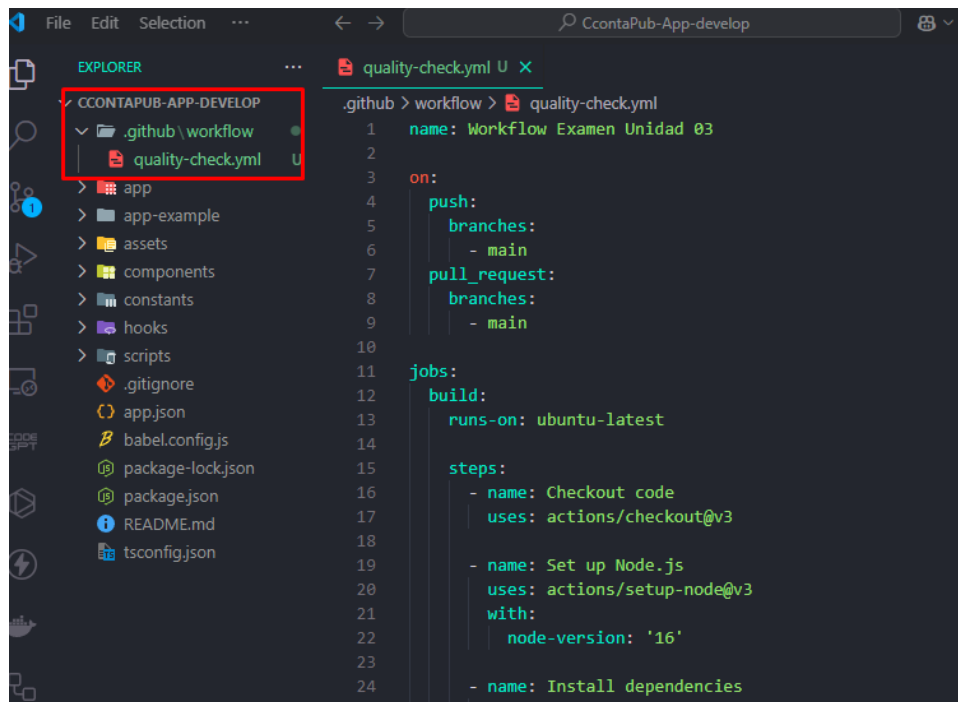
```
MINGW64/c/Users/Stryker/Desktop/Final-2024-II/EXAMMOviles/ContaPub-App-develop/ContaPub-A
create mode 100644 components/ExternalLink.tsx
create mode 100644 components/HelloWave.tsx
create mode 100644 components/ParallaxScrollView.tsx
create mode 100644 components/ThemedText.tsx
create mode 100644 components/ThemedView.tsx
create mode 100644 components/TopBar.tsx
create mode 100644 components/__tests__/ThemedText-test.tsx
create mode 100644 components/__tests__/__snapshots__/ThemedText-test.tsx.snap
create mode 100644 constants/Colors.ts
create mode 100644 hooks/useColorScheme.ts
create mode 100644 hooks/useColorScheme.web.ts
create mode 100644 hooks/useThemeColor.ts
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 scripts/reset-project.js
create mode 100644 tsconfig.json

Stryker@DESKTOP-N7GVH97 MINGW64 ~/Desktop/Final-2024-II/EXAMMOviles/ContaPub-Ap
p-develop/ContaPub-App-develop (master)
$ git branch -M main

Stryker@DESKTOP-N7GVH97 MINGW64 ~/Desktop/Final-2024-II/EXAMMOviles/ContaPub-Ap
p-develop/ContaPub-App-develop (main)
$ git remote add origin https://github.com/Luis-Caxi-Calani/SM2_ExamenUnidad3.git

Stryker@DESKTOP-N7GVH97 MINGW64 ~/Desktop/Final-2024-II/EXAMMOviles/ContaPub-Ap
p-develop/ContaPub-App-develop (main)
$ git push -u origin main
Enumerating objects: 63, done.
Counting objects: 100% (63/63), done.
Delta compression using up to 6 threads
Compressing objects: 100% (57/57), done.
Writing objects: 100% (63/63), 321.40 KiB | 9.45 MiB/s, done.
Total 63 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Luis-Caxi-Calani/SM2_ExamenUnidad3.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

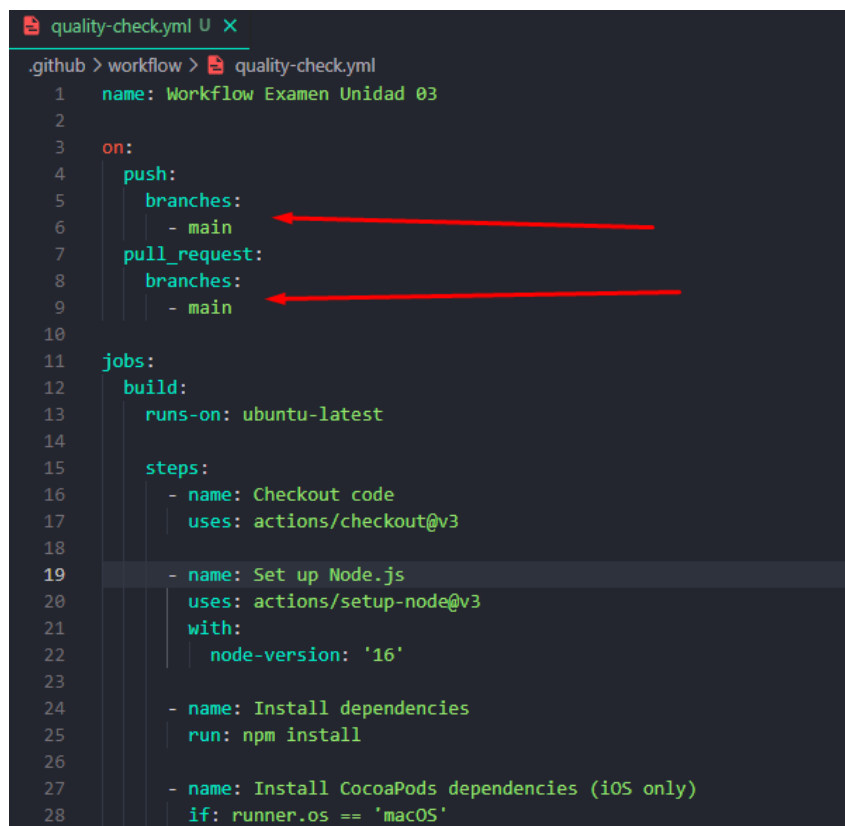
3. Luego en tu repositorio "SM2_ExamenUnidad3", genera un "actions" creando el Workflow en ".github/workflows/quality-check.yml"



The screenshot shows the Visual Studio Code interface. In the Explorer view on the left, the file explorer is expanded to show the contents of the 'CCONTAPUB-APP-DEVELOP' workspace. The '.github/workflows' folder is selected, and the 'quality-check.yml' file is highlighted. The main editor window displays the content of 'quality-check.yml'.

```
.github > workflow > quality-check.yml
1  name: Workflow Examen Unidad 03
2
3  on:
4    push:
5      branches:
6        - main
7    pull_request:
8      branches:
9        - main
10
11 jobs:
12   build:
13     runs-on: ubuntu-latest
14
15     steps:
16       - name: Checkout code
17         uses: actions/checkout@v3
18
19       - name: Set up Node.js
20         uses: actions/setup-node@v3
21         with:
22           node-version: '16'
23
24       - name: Install dependencies
```

4. El workflow debe ejecutarse automáticamente al hacer un commit o pull request.



This is a close-up view of the 'quality-check.yml' file. Two red arrows point to the 'push' and 'pull_request' sections under the 'on:' key, indicating that the workflow should trigger on both events. The file content is as follows:

```
quality-check.yml
.github > workflow > quality-check.yml
1  name: Workflow Examen Unidad 03
2
3  on:
4    push:
5      branches:
6        - main
7    pull_request:
8      branches:
9        - main
10
11 jobs:
12   build:
13     runs-on: ubuntu-latest
14
15     steps:
16       - name: Checkout code
17         uses: actions/checkout@v3
18
19       - name: Set up Node.js
20         uses: actions/setup-node@v3
21         with:
22           node-version: '16'
23
24       - name: Install dependencies
25         run: npm install
26
27       - name: Install CocoaPods dependencies (iOS only)
28         if: runner.os == 'macOS'
```