**Nombre del archivo: evalucionSonarCloud.yml**

on:

  # Trigger analysis when pushing in master or pull requests, and when creating

  # a pull request.

  push:

    branches:

      - develop

  pull\_request:

      types: [opened, synchronize, reopened]

name: Main Workflow

jobs:

  build-and-deploy:

    runs-on: ubuntu-latest

    steps:

    # checkout the repo

    - name: 'Checkout GitHub Action'

      uses: actions/checkout@main

    - name: 'Login via Azure CLI'

      uses: azure/login@v1

      with:

        creds: ${{ secrets.AZURE\_CREDENTIALS }}

        enable-AzPSSession: true

    - name: 'Create env file'

      run: |

        echo "${{ secrets.ENV\_FILE }}" > .env

    - name: 'Build and push image'

      uses: azure/docker-login@v1

      with:

        login-server: ${{ secrets.REGISTRY\_LOGIN\_SERVER }}

        username: ${{ secrets.REGISTRY\_USERNAME }}

        password: ${{ secrets.REGISTRY\_PASSWORD }}

    - run: |

        docker build . -t ${{ secrets.REGISTRY\_LOGIN\_SERVER }}/pruebadeployconstancias:${{ github.sha }}

        docker push ${{ secrets.REGISTRY\_LOGIN\_SERVER }}/pruebadeployconstancias:${{ github.sha }}

    - name: 'Deploy in App Service'

      uses: azure/webapps-deploy@v2

      with:

        app-name: 'prueba-deploy-constancias'

        images: '${{ secrets.REGISTRY\_LOGIN\_SERVER }}/pruebadeployconstancias:${{ github.sha }}'

    - name: Azure logout

      run: |

        az logout

  sonarcloud:

    runs-on: ubuntu-latest

    steps:

    - uses: actions/checkout@v3

      with:

        # Disabling shallow clone is recommended for improving relevancy of reporting

        fetch-depth: 0

    - name: Setup Python

      uses: actions/setup-python@v2

      with:

        python-version: ${{ matrix.python }}

    - name: Install tox and any other packages

      run: pip install tox==3.28.0

    - name: 'Create env file'

      run: |

        echo "${{ secrets.ENV\_FILE }}" > .env

    - name: Run tox

      run: tox -e py

    - name: SonarCloud Scan

      uses: sonarsource/sonarcloud-github-action@master

      env:

        SONAR\_TOKEN: ${{ secrets.SONAR\_TOKEN }}

Crear una carpeta: **test**

1. dentro de la carpeta colocar el siguiente archivo con el nombre: **\_\_init\_\_.py** , el cual debe estar **vacio**.

2. colocar el archivo a evaluar dependiendo del componente.

3. dentro de la carpeta colocar el siguiente archivo con el nombre: **test\_nombre\_del\_servicio.py**

from run import app

import os

import io

from base64 import b64encode

userPass = os.environ.get('USER\_BASIC\_AUTH')+":"+os.environ.get('PASSWORD\_BASIC\_AUTH')

userPassEncoded = b64encode(userPass.encode('ascii'))

headers = {

    'Authorization': 'Basic %s' % userPassEncoded.decode("ascii"),

    'mail': 'luis.fortanel@tec.mx'

}

def test\_status\_code\_401():

    tester = app.test\_client()

    response = tester.post("/constancias-situacion-fiscal") # modificar aquí por el servicio

    assert response.status\_code == 401

def test\_no\_pdf():

    tester = app.test\_client()

    response = tester.post("/constancias-situacion-fiscal", headers=headers) #modificar aqui por el servicio

    print(response.data)

    print(type(response.data))

    assert response.status\_code == 400

    assert response.data == b'{"error":"No se encontro el pdf","result":null,"status\_code":400}\n'

def test\_status\_code\_401():

    data = {}

    data['pdf'] = io.open("tests/constancia.pdf", "rb", buffering = 0) #modificiar aqui por como se llama el archivo

    headers\_local = {

    'Authorization': 'Basic %s' % userPassEncoded.decode("ascii"),

    'mail': 'luis.fortane@tec.mx'

    }

    tester = app.test\_client()

    response = tester.post(

        "/constancias-situacion-fiscal", data=data, follow\_redirects=True,

        content\_type='multipart/form-data', headers=headers\_local

    ) # modificar aqui por el servicio y como se envia desde postman

    assert response.status\_code == 401

def test\_status\_code\_200():

    data = {}

    data['pdf'] = io.open("tests/constancia.pdf", "rb", buffering = 0) #modificar aqui

    headers\_local = {

    'Authorization': 'Basic %s' % userPassEncoded.decode("ascii"),

    'mail': 'luis.fortanel@tec.mx'

    }

    tester = app.test\_client()

    response = tester.post(

        "/constancias-situacion-fiscal", data=data, follow\_redirects=True,

        content\_type='multipart/form-data', headers=headers\_local

    ) # modificar aqui por el servicio que se envia desde postman

    assert response.status\_code == 200

Agregar archivo con el nombre: **sonar-project.properties**

sonar.organization=ti-tecnologico-de-monterrey-oficial # se queda igual

sonar.projectKey=tec-cognitive-extraccion-de-datos-documentos-oficiales # se camnbia por el proyecto que se va a escanear

# relative paths to source directories. More details and properties are described

# in https://sonarcloud.io/documentation/project-administration/narrowing-the-focus/

sonar.sources=.

sonar.tests=tests

sonar.python.coverage.reportPaths=/github/workspace/coverage.xml

sonar.exclusions=tests/\*\*, tox.ini, tests/\_\_init\_\_.py, Dockerfile, uwsgi.ini

Agregar archivo con el nombre: **tox.ini**

Donde se van a incluir las librerías que se utilizaron para el proyecto

[tox]

envlist = py39

skipsdist = True

[testenv]

deps =

    azure-ai-formrecognizer==3.3.2

    azure-common==1.1.28

    azure-core==1.29.5

    blinker==1.7.0

    certifi==2023.11.17

    charset-normalizer==3.3.2

    click==8.1.7

    colorama==0.4.6

    Flask==3.0.0

    Flask-BasicAuth==0.2.0

    Flask-Cors==4.0.0

    Flask-SQLAlchemy==3.1.1

    Flask-WTF==1.2.1

    greenlet==3.0.2

    idna==3.6

    isodate==0.6.1

    itsdangerous==2.1.2

    Jinja2==3.1.2

    MarkupSafe==2.1.3

    msrest==0.7.1

    oauthlib==3.2.2

    passlib==1.7.4

    pyodbc==5.0.1

    python-dotenv==1.0.0

    requests==2.31.0

    requests-oauthlib==1.3.1

    six==1.16.0

    SQLAlchemy==2.0.23

    typing\_extensions==4.9.0

    ua-parser==0.18.0

    urllib3==2.1.0

    user-agents==2.2.0

    Werkzeug==3.0.1

    WTForms==3.1.1

    coverage

    pytest

    pytest-cov

commands = pytest --cov=./ --cov-report=xml --cov-config=tox.ini --cov-branch

[coverage:run]

relative\_files = True