Tarea: Integración de SonarCloud con GitHub para el Análisis Estático de Código

Objetivo:

Integrar SonarCloud con un repositorio de GitHub y realizar un análisis estático de código en un script de Python de ejemplo. Esto ayudará a los estudiantes a comprender cómo utilizar SonarCloud para identificar y corregir problemas de codificación y vulnerabilidades de seguridad.

Requisitos:

1. Cuenta de GitHub
2. Cuenta de SonarCloud
3. Conocimientos básicos de Python y Git
4. Un script de Python de ejemplo (proporcionado a continuación)

Pasos:

Paso 1: Crear un Repositorio en GitHub

1. Vaya a GitHub e inicie sesión en su cuenta.
2. Haga clic en el icono + en la esquina superior derecha y seleccione New repository.
3. Nombre el repositorio sonarcloud-analysis y hágalo público.
4. Inicialice el repositorio con un archivo README.md y cree el repositorio.

Paso 2: Agregar el Script de Python de Ejemplo

1. Clone el repositorio en su máquina local:

   git clone https://github.com/su-usuario/sonarcloud-analysis.git

2. Navegue hasta el directorio del repositorio:

```
cd sonarcloud-analysis
```

3. Cree un nuevo archivo llamado main.py y agregue el siguiente script de Python de ejemplo:

```
import os

def read_file(file_path):
    try:
        with open(file_path, 'r') as file:
            data = file.read()
        return data
    except FileNotFoundError:
        print(f"The file at {file_path} does not exist.")
        return None

def write_file(file_path, data):
    with open(file_path, 'w') as file:
        file.write(data)

def get_user_input():
    user_input = input("Enter some text: ")
    return user_input

def process_data(data):
    processed_data = data.lower()
    return processed_data
```

```
def main():
    file_path = "example.txt"

    # Reading from a file
    data = read_file(file_path)
    if data is None:
        return

    # Processing data
    processed_data = process_data(data)
    print(f"Processed Data: {processed_data}")

    # Getting user input and writing to a file
    user_input = get_user_input()
    write_file(file_path, user_input)

if __name__ == "__main__":
    main()
```

4. Haga commit y push de los cambios a GitHub:

```
git add main.py
git commit -m "Add sample Python script"
git push origin main
```

Paso 3: Configurar SonarCloud

1. Vaya a SonarCloud e inicie sesión utilizando su cuenta de GitHub.

2. Haga clic en el icono + en la esquina superior derecha y seleccione Analyze new project.

3. Seleccione su repositorio sonarcloud-analysis de la lista y haga clic en Set Up.

4. Siga las instrucciones para otorgar acceso a SonarCloud.

Paso 4: Configurar SonarCloud en GitHub

1. Genere un token de SonarCloud:

   - Vaya a la configuración de su cuenta de SonarCloud.

   - Haga clic en Security y genere un nuevo token.

   - Copie el token y guárdelo de forma segura.

2. Vaya a la configuración de su repositorio en GitHub.

   - Navegue a Settings > Secrets and variables > Actions.

   - Haga clic en New repository secret.

   - Nombre el secreto SONAR_TOKEN y pegue el token de SonarCloud como valor.

   - Haga clic en Add secret.

Paso 5: Crear un Workflow de GitHub Actions para SonarCloud

1. En su repositorio local, cree un nuevo directorio llamado .github/workflows.

2. Dentro del directorio .github/workflows, cree un archivo llamado sonarcloud.yml con el siguiente contenido:

```
name: SonarCloud

on:
  push:
    branches:
      - main
```

```
  pull_request:
    types: [opened, synchronize, reopened]

jobs:
  sonarcloud:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Set up JDK 11
        uses: actions/setup-java@v1
        with:
          java-version: '11'

      - name: Cache SonarCloud packages
        uses: actions/cache@v1
        with:
          path: ~/.sonar/cache
          key: ${{ runner.os }}-sonar
          restore-keys: ${{ runner.os }}-sonar

      - name: Install SonarCloud Scanner
        run: |
                                curl  -sL  https://sonarcloud.io/static/cpp/build-wrapper-linux-x86.zip  -o
```

```
build-wrapper-linux-x86.zip
          unzip build-wrapper-linux-x86.zip -d ~/sonar-scanner
          export PATH="$PATH:~/sonar-scanner/build-wrapper-linux-x86"

        - name: SonarCloud Scan
          env:
            SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
          run: |
              sonar-scanner -Dsonar.organization=your-organization-key
-Dsonar.projectKey=your-project-key -Dsonar.sources=.
-Dsonar.host.url=https://sonarcloud.io
```

Reemplace your-organization-key y your-project-key con los valores reales de su proyecto en SonarCloud.

3. Haga commit y push de los cambios a GitHub:

```
git add .github/workflows/sonarcloud.yml
git commit -m "Add SonarCloud GitHub Action workflow"
git push origin main
```

Paso 6: Analizar el Código

1. Una vez que empuje el archivo del workflow, GitHub Actions se ejecutará automáticamente y comenzará a analizar el código utilizando SonarCloud.
2. Vaya a la pestaña Actions en su repositorio de GitHub para verificar el estado del workflow.
3. Una vez que el workflow se complete, vaya a su panel de proyecto en SonarCloud para ver los resultados del análisis.

Entregables:

1. Un repositorio de GitHub llamado sonarcloud-analysis con el script de Python de ejemplo y la integración de SonarCloud.

2. Una captura de pantalla del panel de análisis de SonarCloud mostrando los resultados.

Criterios de Calificación:

1. Configuración del Repositorio (20%): Configuración adecuada de un repositorio público de GitHub con el script de Python de ejemplo.

2. Integración de SonarCloud (40%): Integración exitosa de SonarCloud con el repositorio de GitHub y configuración del workflow de GitHub Actions.

3. Resultados del Análisis (30%): Proporcionar una captura de pantalla de los resultados del análisis de SonarCloud.

4. Informe (10%): Escribir un informe breve (1-2 páginas) explicando los problemas identificados por SonarCloud y cómo se pueden corregir.

Envío:

Envíe el enlace a su repositorio de GitHub y el informe de análisis a través del sistema de envío de la universidad.