



# UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

## **ESCUELA PROFESIONAL DE CIENCIAS DE LA COMPUTACIÓN**

### **INGENIERÍA DE SOFTWARE II**

Edgar Sarmiento Calisaya

2022-B

Reporte SonarQube y SonarLint

Alejandro Villa Herrera  
Samuel Chambi Ytusaca  
Julio Yauri Ituccayasi  
Moises Casaverde Aleman

30 de diciembre del 2022

## 1. Análisis Estático (SonarQube):

### a. ISSUES:

- Lista de “Issues” detectadas por la herramienta SonarQube, con la ubicación y code smell correspondiente.

 lists/views.py

☐

**Define a constant instead of duplicating this literal "lists:todoist" 3 times.**

 Code Smell  Critical  Open  Not assigned 6min effort [Comment](#)

2 years ago  L34   design

☐

**Define a constant instead of duplicating this literal "lists:index" 3 times.**

 Code Smell  Critical  Open  Not assigned 6min effort [Comment](#)

2 years ago  L38   design

 lists/tests.py

☐

**Define a constant instead of duplicating this literal "lists:overview" 3 times.**

 Code Smell  Critical  Open  Not assigned 6min effort [Comment](#)

2 years ago  L55   design

☐

**Define a constant instead of duplicating this literal "This field is required." 4 times.**

 Code Smell  Critical  Open  Not assigned 8min effort [Comment](#)

2 years ago  L86   design

 api/tests.py

☐

**Define a constant instead of duplicating this literal "test@example.com" 3 times.**

 Code Smell  Critical  Open  Not assigned 6min effort [Comment](#)

2 years ago  L11   design

 lists/templates/lists/overview.html

☐

**Add a description to this table.**

 Bug  Minor  Open  Not assigned 5min effort [Comment](#)

7 years ago  L45   accessibility, wcag2-a

 accounts/tests.py

☐

**Define a constant instead of duplicating this literal "test@example.com" 3 times.**

 Code Smell  Critical  Open  Not assigned 6min effort [Comment](#)

2 years ago  L17   design

☐

**Define a constant instead of duplicating this literal "auth:register" 3 times.**

 Code Smell  Critical  Open  Not assigned 6min effort [Comment](#)

2 years ago  L23   design

☐

**Define a constant instead of duplicating this literal "auth:login" 5 times.**


 Code Smell  Critical  Open  Not assigned 10min effort [Comment](#)

2 years ago  L28   design

☐

**Define a constant instead of duplicating this literal "This field is required." 4 times.**

 Code Smell  Critical  Open  Not assigned 8min effort [Comment](#)

2 years ago  L117   design

accounts/views.py

Merge this if statement with the enclosing one.

Code Smell

Major

Open

Not assigned

5min effort

Comment

7 years ago

L17

clumsy

Define a constant instead of duplicating this literal "lists:index" 3 times.

Code Smell

Critical

Open

Not assigned

6min effort

Comment

2 years ago

L19

design

b. SECURITY HOTSPOTS:

31 Security Hotspots to review

Review priority: HIGH

Authentication

19

Cross-Site Request Forgery (CSRF)

9

Review priority: LOW

Insecure Configuration

1

Others

2

31 of 31 shown

"password" detected here, review this potentially hard-coded credential.

Hard-coded credentials are security-sensitive python:S2068

Status: TO REVIEW

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Change status

Assignee: Not assigned

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

accounts/tests.py

Open in IDE

Get Permalink

```
7
8 MAIL_SAMPLE = "test@example.com"
9 CONST_FIELD_REQUIRED = "This field is required."
10 CONST_AUTH_REGISTER = "auth:register"
11 CONST_AUTH_LOGIN = "auth:login"
12 class AccountsTests(TestCase):
13     def setUp(self):
14         self.register_data = {
15             "email": "new@user.com",
16             "username": "new_user",
17             "password": "test",
18
19             "password_confirmation": "test",
20         }
```

"password" detected here, review this potentially hard-coded credential.


Comment

## 2. Refactoring (SonarLint):



- Múltiples campos repiten los mismos valores, SonarLint lo muestra como Code Smell.
- Para solucionarlo se extraen los valores repetidos a variables globales.

```
def test_get_todolist_overview(self):
    response = self.client.get(reverse("lists:overview"))
    self.assertTemplateUsed(response, "lists/overview.html")
    self.assertIsInstance(response.context["form"], TodoListForm)

def test_get_todolist_overview_redirect_when_not_logged_in(self):
    self.client.logout()
    response = self.client.get(reverse("lists:overview"))
    self.assertRedirects(response, "/auth/login/?next=/todolists/")
```

 SonarLint Rule Description ✕ ▶ □ ⋮

### String literals should not be duplicated (python:S1192)

 Code Smell  Critical

Duplicated string literals make the process of refactoring error-prone, since you must be sure to update all occurrences.

On the other hand, constants can be referenced from many places, but only need to be updated in a single place.

#### Noncompliant Code Example

With the default threshold of 3:

```
def run():
    prepare("this is a duplicate") # Noncompliant - "this i
    execute("this is a duplicate")
    release("this is a duplicate")
```

---

#### Compliant Solution

```
ACTION_1 = "action1"

def run():
    prepare(ACTION_1)
    execute(ACTION_1)
    release(ACTION_1)
```


```

red_overview = "lists:overview"
alert = "This field is required."


class ListTests(TestCase):
    def setUp(self):
        self.user = User.objects.create_user("test", "test@example.com", "test")
        self.todolist = TodoList(title="test title", creator=self.user)
        self.todolist.save()
        self.todo = Todo(
            description="save todo", todolist_id=self.todolist.id, creator=self.user

```

- Algunas declaraciones de condicionales se encontraban anidadas, SonarLint lo representa como Code Smell.
- Para solucionarlo se mezclan los condicionales en uno solo.


SonarLint Rule Description

### Collapsible "if" statements should be merged (python:S1066)

Code Smell  Major

Merging collapsible `if` statements increases the code's readability.

#### Noncompliant Code Example

```

if condition1:
    if condition2:
        # ...

```

#### Compliant Solution

```

if condition1 and condition2:
    # ...

```

```

if user is not None:
    if user.is_active:
        login(request, user)
        return redirect("lists:index")
else:
    return render(request, "accounts/login.html", {"form": form})

```

```

if user is not None and user.is_active:
    login(request, user)
    return redirect(index)

```