



**UNIVERSIDADE  
FEDERAL RURAL  
DE PERNAMBUCO**

## **Análise Estática do código fonte do grupo: WanderLust**

**Ana Paula**

**Guilherme Marinho**

**Jaime Victor**

**Jéssica Alves**

**Vinícius de França**

**Recife, 2019**

A análise estática tem como objetivo detectar possíveis bugs ao longo do código fonte e ajudar em sua manutenibilidade para aumentar o seu ciclo de vida.

Código analisado através do link:

<https://github.com/app-wanderlust/wanderlust-app/tree/master>

Ferramenta Utilizada: Sonarqube

### Resultado da análise estática.

Versões	v1(11/11)	v2(23/11)
Bugs	0	1
Vulnerabilidades	0	1
Code Smells	58	247
Complexidade Ciclomática	304	645
Linhas duplicadas	0.9%	10.3%

Foram encontrados apenas 247 code smells na ultima versão do código avaliada, onde em sua grande maioria se trata de repetição por falta de constantes, principalmente na parte do DBHelper, Refactor em métodos para diminuir complexidade cognitiva e também ainda há blocos de códigos duplicados não necessários.

O único bug detectado:

```
private List<Avaliacao> getRecomendacao(){
    List<Avaliacao> notasUsuario = new ArrayList<>();
    2 Usuario usuario = 1 findUsuario();
    HashMap<PratoTipico, Float> avaliacoes = predicao.get(usuario);
    for(Map.Entry prato: avaliacoes.entrySet()){
        PratoTipico pratoTipico = (PratoTipico) prato.getKey();
        float nota = (float)prato.getValue();
        if (pessoaPratoDAO.getPessoaPrato( 3 usuario.getPessoa().getId(), pratoTipico.getId()) == null){
```

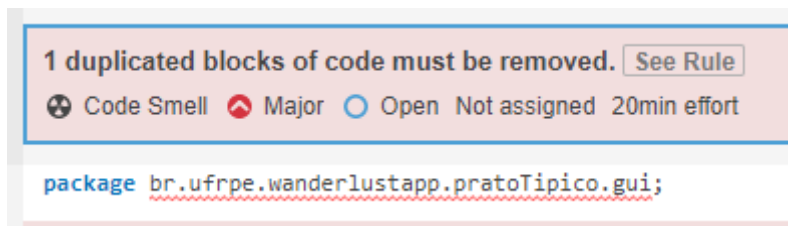
A "NullPointerException" could be thrown; "usuario" is nullable here. [See Rule](#) yesterday ▾ L68 🔗

🐛 Bug ▾ 🚨 Major ▾ 🔓 Open ▾ Not assigned ▾ 10min effort [Comment](#) cert, cwe ▾

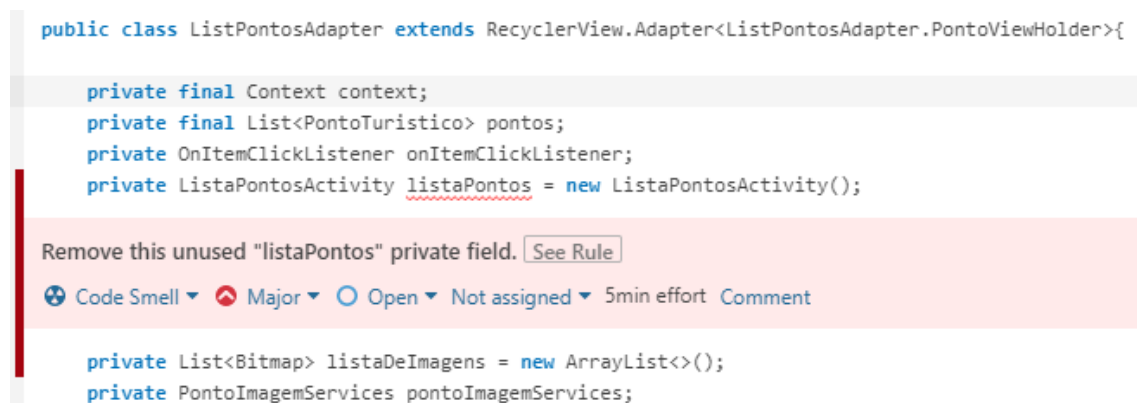
```
        notasUsuario.add(new Avaliacao(pratoTipico, nota));
    }
```

Onde na chamada do método, pode retornar Null e com isso não ter o funcionamento desejado.

Há blocos duplicados:



Atributos não utilizados:



Em relação as complexidade do código:

▼ Complexity ⓘ	
Cyclomatic Complexity	645
Cognitive Complexity	280

Onde nota-se maior complexidade ciclomática no pacote pratoTipico com valor de complexidade de 179.