



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

Análise Estática do código fonte do grupo: PromoKING

Ana Paula

Andressa Santana

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/spicollys/PromoKING>

Ferramenta Utilizada: Sonarqube

Resultado da análise estática.

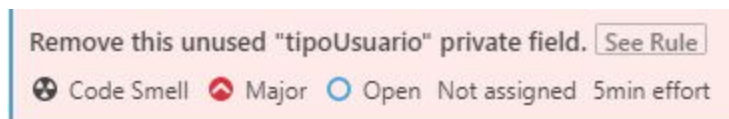
Versões	v1(31/10)	v2(04/11)
Bugs	1	6
Vulnerabilidades	5	16
Code Smells	83	123
Complexidade Ciclomática	261	323
Linhas duplicadas	4.8%	10%



Foram encontrados 123 Code Smells na segunda versão impactando diretamente de forma negativa na manutenibilidade e evolução do código. Dentre os 123 Code Smells, 74 eram de baixa prioridade, 30 de prioridade alta e 19 de prioridade crítica.

Alguns serão listados abaixo:

Variável não utilizada:



Imports não utilizados:

Local: src/.../dominio/EstabelecimentoComercial.java:



Local: src/.../persistencia/EstabelecimentoComercialDAO.java

Remove this unused import 'com.mpoo.promoking.infra.exception.IdTipoUsuarioInvalidoException'.

🚫 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort

Remove this unused import 'com.mpoo.promoking.usuario.dominio.TipoUsuario'. [See Rule](#)

🚫 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort

Variáveis com formatos inadequado.

src/.../mpoo/promoking/infra/exception/MarcaJaCadastradaException.java

Rename this local variable to match the regular expression `^[a-z][a-zA-Z0-9]*$`. [See Rule](#)

🚫 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort

src/.../mpoo/promoking/infra/exception/ProdutoJaCadastradoException.java

Rename this local variable to match the regular expression `^[a-z][a-zA-Z0-9]*$`. [See Rule](#)

🚫 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort

Obs: Em todas as classes de exceções ocorreu este erro.

System.out.Println.

```
protected SQLiteDatabase getReadableDatabase() {  
    BancoDadosHelper bancoDadosHelper = new BancoDadosHelper();  
    System.out.println(bancoDadosHelper);  
}
```

Repetições que poderiam ser evitadas.

src/.../promoking/infra/persistencia/BancoDadosHelper.java

Define a constant instead of duplicating this literal " CREATE TABLE " 5 times. [See Rule](#)

🔧 Code Smell 🚨 Critical 🔵 Open Not assigned 12min effort

Define a constant instead of duplicating this literal " INTEGER PRIMARY KEY AUTOINCREMENT, " 5 times. [See Rule](#)

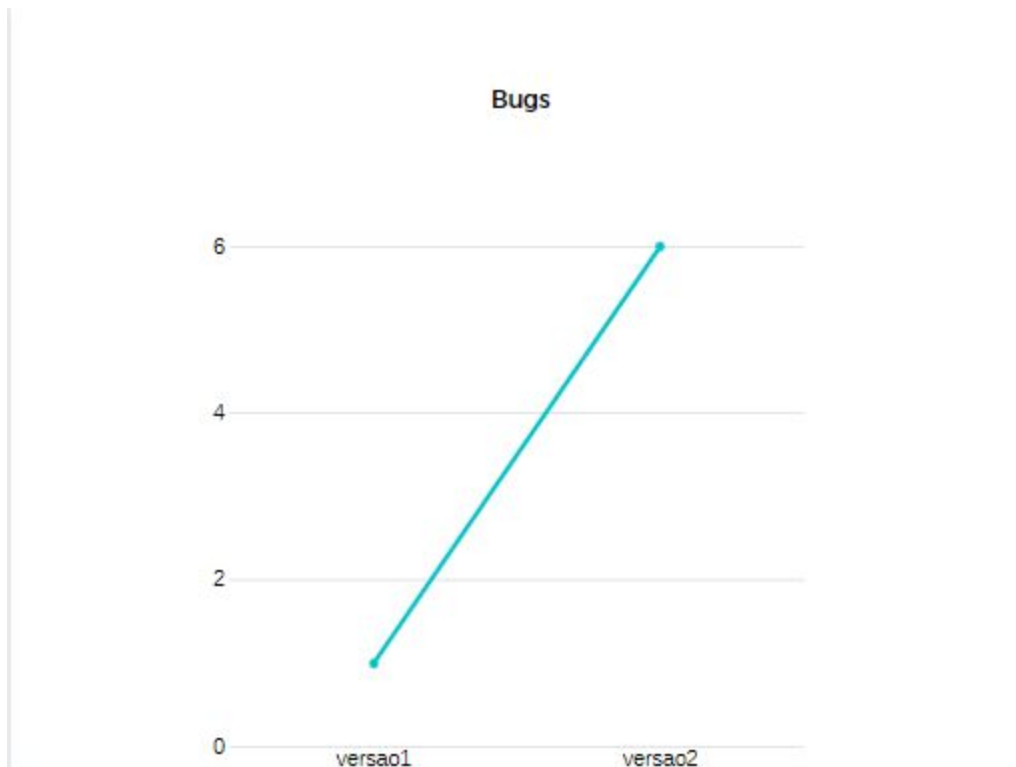
🔧 Code Smell 🚨 Critical 🔵 Open Not assigned 12min effort

Define a constant instead of duplicating this literal " TEXT, " 10 times. [See Rule](#)

🔧 Code Smell 🚨 Critical 🔵 Open Not assigned 22min effort

Define a constant instead of duplicating this literal " REAL, " 3 times. [See Rule](#)

🔧 Code Smell 🚨 Critical 🔵 Open Not assigned 8min effort



Aqui temos um ArrayList de Strings recebendo um resultado Long.

```
public ArrayList<String> getArrayListIdPublicacoes() { return idPublicacoes; }
```

src/.../promoking/publicacao/negocio/PublicacaoServices.java

A "ArrayList<String>" cannot contain a "long" [See Rule](#)

🐛 Bug 🚨 Major 🔓 Open Not assigned 15min effort

A "ArrayList<String>" cannot contain a "long" [See Rule](#)

🐛 Bug 🚨 Major 🔓 Open Not assigned 15min effort

As Vulnerabilidades do App é apenas para utilizar o Log ao invés de `e.printStackTrace();` pois dessa maneira evitará de expor alguma informação confidencial.

Complexidade Ciclomática



