



**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS DE QUIXADÁ

CURSO DE ENGENHARIA DE SOFTWARE

**RELATÓRIO FINAL
MANUTENÇÃO DE SOFTWARE**

Consultório

Gabriel Uchoa de Vasconcelos

Professora: Carla Ilane Moreira Bezerra

Monitores: Darwin Pinheiro; Humberto Damasceno

QUIXADÁ

Janeiro, 2022

SUMÁRIO

1	DESCRIÇÃO DO PROJETO.....	<pág>
2	Avaliação do Projeto.....	<pág>
2.1	Medição 1 – Antes das práticas de manutenção.....	<pág>
2.2	Revisão 1 – Antes das práticas de manutenção.....	<pág>
2.3	Detecção dos Code Smells.....	<pág>
2.4	Implementação dos padrões de Projeto	<pág>
2.5	Refatoração dos Code Smells	<pág>
2.6	Revisão 2 – Após as práticas de manutenção.....	<pág>
2.7	Medição 2 – Após as práticas de manutenção.....	<pág>
3	COMPARAÇÃO DOS RESULTADOS.....	<pág>
	REFERÊNCIAS.	4
	APÊNDICE A..	5

1 DESCRIÇÃO DO PROJETO

Nessa Seção deve ser descrito o projeto. Importante incluir uma Tabela com as seguintes informações: Número de Linhas de Código (LOC), Número de Classes e Número de Releases.

Incluir o link do github do projeto. Falar sobre a natureza do projeto, se é JAVA OO, se é de código aberto ou fechado, quais tecnologias utilizadas no projeto.

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases	Link do Projeto
Consultório	2406		1	https://github.com/GuchoaVasconcelos/manutencao

2 AVALIAÇÃO DO PROJETO

2.1 Medição 1 – Antes das práticas de manutenção













Nessa Seção deve ser incluída a Tabela com a medição das métricas de coesão, acoplamento, complexidade, herança e tamanho, antes do projeto ter as práticas de manutenção implementadas. Para isso será utilizada a ferramenta **Understand**. A Tabela 2 apresenta a descrição das métricas, faça uma tab

Atributos	Métricas	Valores
Coesão	Lack of Cohesion of Methods(LCOM2)	1050
Acoplamento	Coupling Between Object (CBO)	731
Complexidade	Average Cyclomatic Complexity (ACC)	153
	Sum Cyclomatic Complexity	1547

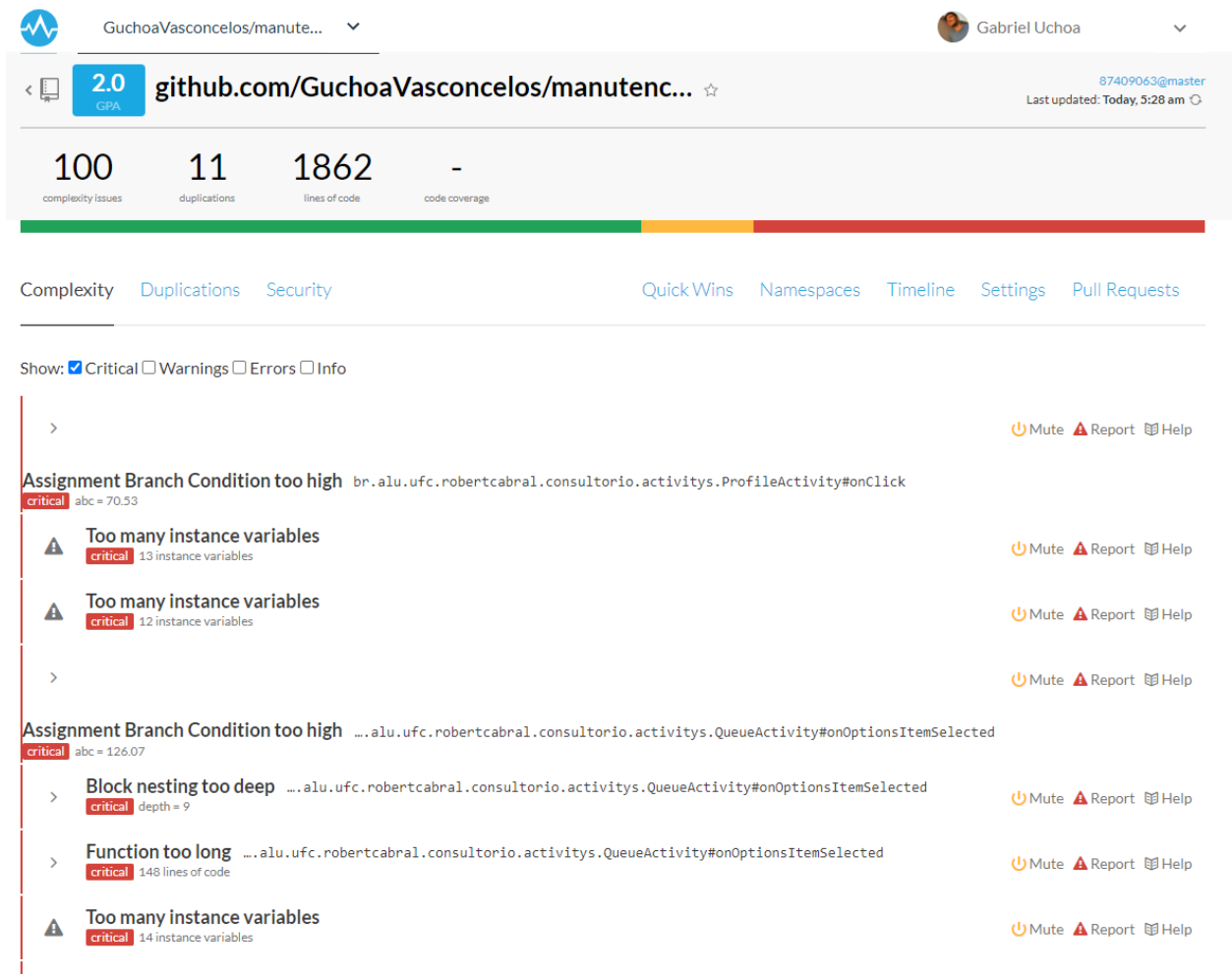
	(SCC)	
	Nesting (MaxNest)	211
	Essential Complexity (EVG)	224
Herança	Number of Children (NOC)	1
	Depth of Inheritance Tree (DIT)	126
	Bases Classes (IFANIN)	78
Tamanho	Lines of Code (LOC)	2406
	Lines with Comments (CLOC)	109
	Classes (CDL)	134
	Instance Methods (NIM)	549

2.2 Revisão 1 – Antes das práticas de manutenção

MEASURES

New Code	Overall Code			
2	 Bugs			Reliability 
2	 Vulnerabilities			Security 
8	 Security Hotspots 	 0.0% Reviewed		Security Review 
4d 7h	Debt	263	 Code Smells	Maintainability 
 0.0%	Coverage on 1.3k Lines to cover	-	Unit Tests	 7.1%
			Duplications on 4.1k Lines	15
			Duplicated Blocks	

ACTIVITY



2.3 Detecção dos Code Smells

Nessa Seção deve ser indicado quais e quantos code smells foram detectados no projeto. Faça uma Tabela indicando os code smells detectados pela ferramenta JSPirit e quantos code smells para cada tipo foram detectados.

Tabela 3 – Code smells do projeto.

Nome do Code Smell	Quantidade
Dead Code	19
Duplicate code	17

Speculative Generality	39
------------------------	----

2.4 Implementação dos padrões de Projeto

Nessa Seção devem ser apresentados os padrões de projeto que foram implementados, juntamente com a localização de tais padrões no código fonte.

2.5 Refatoração dos Code Smells

Apresente o resultado das refatorações feitas, indicando o código, o nome do code smells, e a técnica adotada. Para cada code smell, siga o exemplo da tabela abaixo.

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant
Código	
<p>Antes:</p> <pre>Query list = mDatabase.child("Queue");</pre> <p>Depois:</p> <pre>Query list = mDatabase.child(QUEUE);</pre>	

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant
Código	

Antes:

```
Query queryLast = mDatabase.child("lastPosition");
```

Depois:

```
Query queryLast = mDatabase.child(LAST_POSITION);
```

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant
Código	
Antes: Log.d("mylog", "onPostExecute lineoptions decoded"); Depois: Log.d(MYLOG, "onPostExecute lineoptions decoded");	

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant
Código	
Antes: place1 = new MarkerOptions().position(new LatLng(latitude, longitude)).title("Localização atual").icon(BitmapDescriptorFactory.fromBitmap(resizeMapIcons("icon_location_user", 80, 104))); place2 = new MarkerOptions().position(new LatLng(-4.965378, -39.006108)).title("Consultorio").icon(BitmapDescriptorFactory.fromBitmap(resizeMapIcons("icon_location", 80, 104))); Depois: place1 = new MarkerOptions().position(new LatLng(latitude, longitude)).title(LOCALIZAÇÃO_ATUAL).icon(BitmapDescriptorFactory.fromBitmap(resizeMapIcons(ICON_LOCATION_USER, 80, 104)));	


```
place2 = new MarkerOptions().position(new LatLng(-4.965378,
-39.006108)).title(CONSULTORIO).icon(BitmapDescriptorFactory.fromBitmap(resizeMapIcons(ICON
_LOCATION, 80, 104)));
```

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant
Código	
<p>Antes:</p> <pre>Query query = mDatabase.child("Users").child(user.getUid());</pre> <p>Depois:</p> <pre>Query query = mDatabase.child(USERS).child(user.getUid());</pre>	

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant
Código	
<p>Antes:</p> <pre>Log.d("login", "Login Google: " + account.getId());</pre> <p>Depois:</p> <pre>Log.d(LOGIN, "Login Google: " + account.getId());</pre>	

Code Smell	Duplicate Code
Técnica de Refatoração	Introduce Constant e Extract Method

Código
<p>Antes:</p> <pre>progress.start("Aguarde", "Fazendo o login");</pre> <p>Depois:</p> <pre>private void fazendoLogin() { progress.start(AGUARDE, FAZENDO_O_LOGIN); }</pre>

Code Smell	Speculative Generality
Técnica de Refatoração	Adicionar um throw
Código	
<p>Antes:</p> <pre>@Override public void onCancelled(@NonNull DatabaseError databaseError) { }</pre> <p>Depois:</p> <pre>@Override public void onCancelled(@NonNull DatabaseError databaseError) { throw new UnsupportedOperationException(); }</pre>	

Code Smell	Speculative Generality
Técnica de Refatoração	Adicionar um throw
Código	

Antes:

```
@Override
public void failure(TwitterException exception) {

}
```

Depois:

```
@Override
public void failure(TwitterException exception) {
    throw new UnsupportedOperationException();
}
```

Code Smell	Speculative Generality
Técnica de Refatoração	Adicionar um throw
Código	
<p>Antes:</p> <pre>@Override public void onClick(View v) { }</pre> <p>Depois:</p> <pre>@Override public void onClick(View v) { throw new UnsupportedOperationException(); }</pre>	

Code Smell	Speculative Generality
Técnica de Refatoração	Adicionar um throw
Código	

Antes:

```
public void onProviderDisabled(String provider) {  
  
    }  

```

Depois:

```
public void onProviderDisabled(String provider) {  
    throw new UnsupportedOperationException();  
  
    }  

```

Code Smell	Speculative Generality
Técnica de Refatoração	Adicionar um throw
Código	
<p>Antes:</p> <pre>public void onStatusChanged(String provider, int status, Bundle extras) { } </pre> <p>Depois:</p> <pre>public void onStatusChanged(String provider, int status, Bundle extras) { throw new UnsupportedOperationException(); } </pre>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	

Antes:

```
final SimpleDateFormat dt = new SimpleDateFormat("dd/MM/yyyy");
```

Depois:

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
Antes: //mDatabase.child(fila1.getId()).removeValue(); //ArrayList<Fila> values = new ArrayList<>(); Depois:	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
Antes: //ArrayList<Fila> values = new ArrayList<>(); Depois:	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>// mDatabase.addValueEventListener(new ValueEventListener() { // @Override // public void onDataChange(@NonNull DataSnapshot dataSnapshot) { // values.clear(); // for (DataSnapshot data : dataSnapshot.getChildren()){ // values.add(data.getValue(Fila.class)); // } // } // Collections.sort(values, (o1, o2) -> Integer.compare(o1.getPosition(), o2.getPosition())); // } // // @Override // public void onCancelled(@NonNull DatabaseError databaseError) { // } // });</pre> <p>Depois:</p>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>//progress = new Progress(AdminActivity.this); //progress.start("Aguarde", "Atualizando");</pre> <p>Depois:</p>	

Code Smell	Dead Code
-------------------	-----------

Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>//Progress progress;</pre> <p>Depois:</p>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>//Toast.makeText(getApplicationContext(), "Position: " + dataSnapshot.getValue(Integer.class), Toast.LENGTH_SHORT).show();</pre> <p>Depois:</p>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	

Antes:

```
//      int tes = 1;
//
//      FirebaseDatabase.getInstance().getReference("lastPosition")
//          .setValue(tes);
```

Depois:

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
Antes: //Query query = databaseRef; //query.addListenerForSingleValueEvent(listener); Depois:	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
Antes: //Snackbar.make(getWindow().getCurrentFocus(), "Usuário cancelou", Snackbar.LENGTH_LONG).show(); Depois:	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>//progressDialog.dismiss();</pre> <p>Depois:</p>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>//LatLng hcmus = new LatLng((place1.getPosition().latitude + place2.getPosition().latitude) / 2, (place1.getPosition().longitude + place2.getPosition().longitude) / 2); //mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(hcmus, 14));</pre> <p>Depois:</p>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	

Antes:

```
//  
//      LatLngBounds.Builder builder = new LatLngBounds.Builder();  
//      builder.include(place1.getPosition());  
//      builder.include(place2.getPosition());  
//      LatLngBounds bounds = builder.build();  
//  
//      int padding = 200;  
//      CameraUpdate cu = CameraUpdateFactory.newLatLngBounds(bounds, padding);  
//  
//      mMap.animateCamera(cu);
```

Depois:

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	
<p>Antes:</p> <pre>//Toast.makeText(getApplicationContext(), "Lat: " + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();</pre> <p>Depois:</p>	

Code Smell	Dead Code
Técnica de Refatoração	Remoção
Código	

Antes:

case 2:

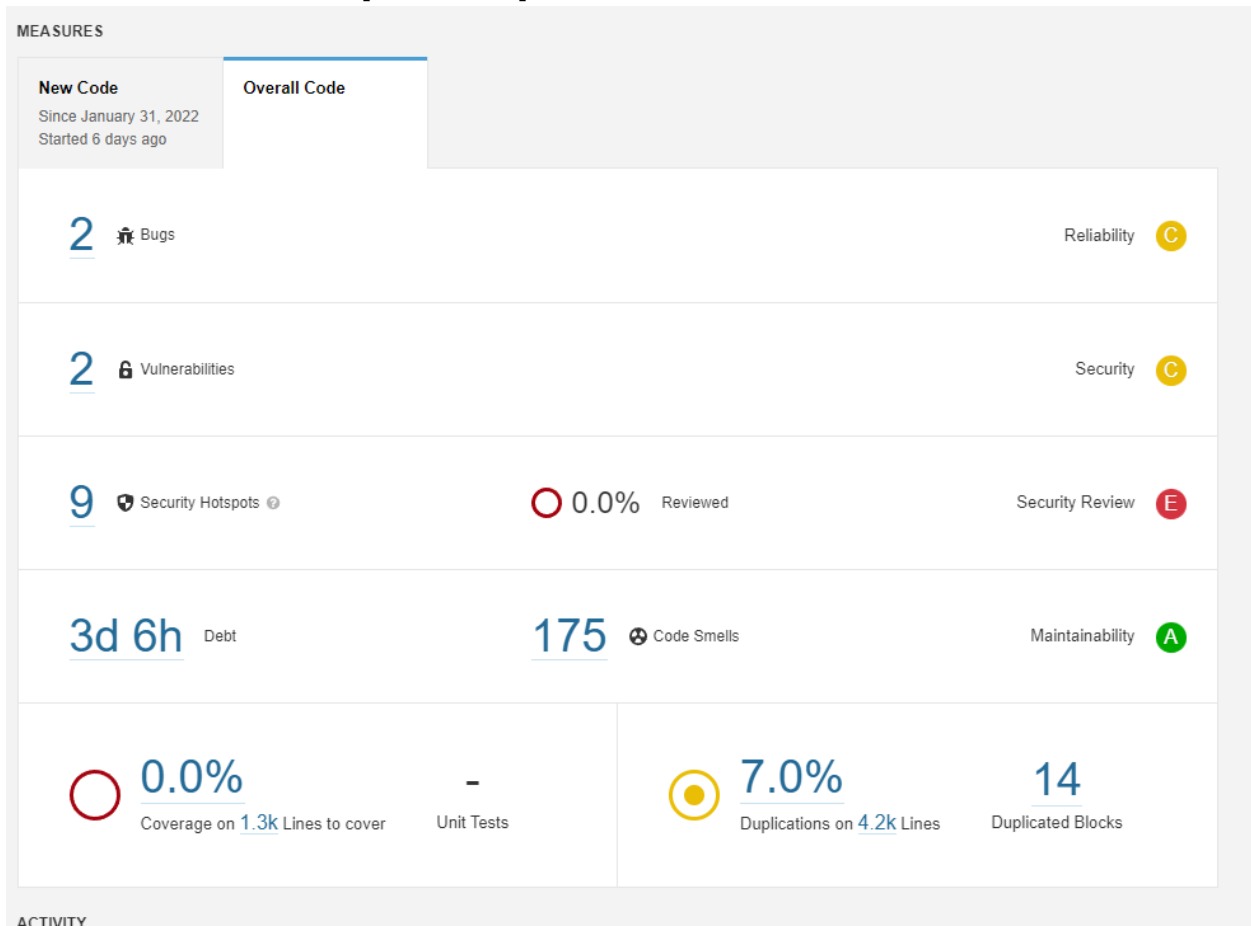
```
Progress progress = new Progress(SettingsActivity.this);  
//progress.start();  
FirebaseAuth.getInstance().signOut();  
stopService(new Intent(SettingsActivity.this, NotificationService.class));  
Intent i = new Intent(SettingsActivity.this, MainActivity.class);  
i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
startActivity(i);  
//progress.stop();  
finishAffinity();
```

Depois:

case 2:

```
Progress progress = new Progress(SettingsActivity.this);  
FirebaseAuth.getInstance().signOut();  
stopService(new Intent(SettingsActivity.this, NotificationService.class));  
Intent i = new Intent(SettingsActivity.this, MainActivity.class);  
i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
startActivity(i);  
finishAffinity();
```

2.6 Revisão 2 - Após as práticas de manutenção



Atributos	Métricas	Valores
Coesão	Lack of Cohesion of Methods(LCOM2)	1053
Acoplamento	Coupling Between Object (CBO)	772
Complexidade	Average Cyclomatic Complexity (ACC)	151
	Sum Cyclomatic Complexity (SCC)	1547
	Nesting (MaxNest)	211
	Essential Complexity (EVG)	224
Herança	Number of Children (NOC)	1
	Depth of Inheritance Tree (DIT)	125
	Bases Classes (IFANIN)	77
Tamanho	Lines of Code (LOC)	2488
	Lines with Comments (CLOC)	55
	Classes (CDL)	132
	Instance Methods (NIM)	549

3 COMPARAÇÃO DOS RESULTADOS

Atributos	Métricas	Valores(Antes)	Valores(Depois)
Coesão	Lack of Cohesion of Methods(LCOM2)	1050	1053
Acoplamento	Coupling Between Object (CBO)	731	772
Complexidade	Average Cyclomatic Complexity (ACC)	153	151
	Sum Cyclomatic Complexity (SCC)	1547	1547
	Nesting (MaxNest)	211	211
	Essential Complexity (EVG)	224	224
Herança	Number of Children (NOC)	1	1
	Depth of Inheritance Tree (DIT)	126	125
	Bases Classes (IFANIN)	78	77
Tamanho	Lines of Code (LOC)	2406	2488
	Lines with Comments (CLOC)	109	55
	Classes (CDL)	134	132
	Instance Methods (NIM)	549	549

REFERÊNCIAS

<https://refactoring.guru/pt-br/smells/speculative-generality>