



# Análise e Teste de Software

Bruno Martins A80410  
Eduardo Barbosa A83344  
Filipe Monteiro A80229  
Márcio Sousa A82400  
Miguel Brandão A82349

# 2

## Índice

- ▶ **Code Smells**
- ▶ **Refactors Aplicados**
- ▶ **Teste Unitários**
- ▶ **Gerador de Logs**
- ▶ **Geração Automática de Testes**
- ▶ **Análise Energética**
- ▶ **Conclusão**

# 3

A close-up photograph of a hand holding a blue pen, poised to write on a piece of paper. The hand is wearing a grey, textured sweater. The background is blurred, showing more of the paper and the pen.

## Code Smells

- ▶ **Deteção Humana;**
- ▶ **SonarQube;**

# 4

## Humano

- Classe Longa;
- Métodos Longos;
- Switch Statements;
- Código Morto;
- Código Duplicado;

# 5

## SonarQube

- Demasiados Parametros;
- Métodos Longos;
- Código Morto;
- Substituir a implementação do clone;

# 6

# Refactors Aplicados

```
private void pickChildMenus() {  
    this.options.clear();  
    switch (this.menu) {  
        case Initial:  
            this.options.add(MenuInd.Login);  
            this.options.add(MenuInd.Register);  
            break;  
        case Register:  
            this.options.add(MenuInd.RegisterClient);  
            this.options.add(MenuInd.RegisterOwner);  
            break;  
        case Client:  
            this.options.add(MenuInd.HistoryClient);  
            this.options.add(MenuInd.Pending);  
            this.options.add(MenuInd.Rentals);  
            this.options.add(MenuInd.Top10Clients);  
            break;  
        case Top10Clients:  
            this.options.add(MenuInd.NUses);  
            this.options.add(MenuInd.Distance);  
            break;  
        case Rentals:  
            this.options.add(MenuInd.Closest);  
            this.options.add(MenuInd.Cheapest);  
            this.options.add(MenuInd.CheapestNear);  
            this.options.add(MenuInd.Specific);  
            this.options.add(MenuInd.Autonomy);  
            break;  
        case Owner:  
            this.options.add(MenuInd.HistoryOwner);  
            this.options.add(MenuInd.CarOverview);  
            this.options.add(MenuInd.ReviewRental);  
            this.options.add(MenuInd.AddCar);  
            break;  
    }  
}
```

```
private void pickChildMenus() {  
    this.options.clear();  
    switch (this.menu) {  
        case Initial:  
            printMenuOptions(MenuInd.Login, MenuInd.Register);  
            break;  
        case Register:  
            printMenuOptions(MenuInd.RegisterClient, MenuInd.RegisterOwner);  
            break;  
        case Client:  
            printMenuOptions(MenuInd.HistoryClient, MenuInd.Pending, MenuInd.Rentals, MenuInd.Top10Clients);  
            break;  
        case Top10Clients:  
            printMenuOptions(MenuInd.NUses, MenuInd.Distance);  
            break;  
        case Rentals:  
            printMenuOptions(MenuInd.Closest, MenuInd.Cheapest, MenuInd.CheapestNear, MenuInd.Specific, MenuInd.Autonomy);  
            break;  
        case Owner:  
            printMenuOptions(MenuInd.HistoryOwner, MenuInd.CarOverview, MenuInd.ReviewRental, MenuInd.AddCar);  
            break;  
    }  
}
```

# 7

## Testes Unitários

```
@Test
public void distanceTest1() {
    Point p1 = new Point( x: 0.0, y: 1.0);
    assertEquals( expected: 1.0,
                 p1.distanceBetweenPoints(new Point( x: 0.0, y: 0.0)));
}
```

Exemplo de teste unitário na Demo 1

# 8

## Testes Unitários

```
class CoordinateTest {  
  
    @Test  
    void getDistancia1() {  
        Coordinate c1 = new Coordinate( latitude: 20, longitude: 20);  
        Coordinate c2 = new Coordinate( latitude: 5, longitude: 5);  
  
        assertEquals( expected: 21.2132, c1.getDistancia(c2));  
    }  
}
```

Exemplo de teste unitário na Demo 2



# 9

## Cobertura de Testes

Demo 1: 70% das classes, 50% do código

Demo 2: 83% das classes, 53% do código

# 10

# Gerador de Logs

- Haskell
- QuickCheck
- Gerador condicional genérico
- Monads e Monads Transformers

11

**Monads**

**Monads**

**Monads**

**Monads**

**Monads**

**Monads**

# 12

# Interface

```
herulume@tardis generator :) > stack run -- -h
generator - gerador de valores aleatorios para UmCarroJa

Usage: generator-exe [-c|--clientes INT] [-p|--proprietarios INT]
                  (-f|--ficheiro FICHEIRO) --carros INT --alugueres INT
      Gera um ficheiro de logs para UmCarroJa

Available options:
  -c,--clientes INT      Numero de clientes (default: 20)
  -p,--proprietarios INT  Numero de proprietarios (default: 20)
  -f,--ficheiro FICHEIRO  Ficheiro de dump
  --carros INT            Numero de carros
  --alugueres INT         Numero de alugueres
  -h,--help              Show this help text
herulume@tardis generator :) >
```

# 13

# Interface

```
herulume@tardis generator :) > stack run -- -c 2000 -p 2000 --carros 150 --alugueres 1000 -f logs; cat logs | head -20
NovoCliente:Alberto da Silva,149325289,149325289@gmail.com,Viana do Castelo,78.955246,-44.313335
NovoCliente:Bernardo de Corte-Real Figueiredo Martins Freitas,565941508,565941508@gmail.com,Coimbra,26.055504,19.908554
NovoCliente:Paula Gama Freitas,468545971,468545971@gmail.com,Evora,21.041931,-33.683357
NovoCliente:Alex Cardoso da Silva Pereira de Amorim,788759695,788759695@gmail.com,Portalegre,-53.274437,8.101486
NovoCliente:Andre Barbosa Cortez,144360411,144360411@gmail.com,Viana do Castelo,31.098442,53.626114
NovoCliente:Abel de Beir Simoes,895207181,895207181@gmail.com,Bragança,-4.268486,84.64853
NovoCliente:Bernardo Silva Oliveira Matos,287171659,287171659@gmail.com,Vila Real,-33.078796,57.32469
NovoCliente:Sara Fernandes Silva,563407120,563407120@gmail.com,Madeira,58.620377,-44.80773
NovoCliente:Jose Barbosa Figueiredo Gama Machado,907987939,907987939@gmail.com,Aveiro,74.72746,58.357834
NovoCliente:Alberto Correia Castro,668380049,668380049@gmail.com,Guarda,26.860458,54.55789
NovoCliente:Francisca Silva,038098120,038098120@gmail.com,Guarda,-72.61451,31.980911
NovoCliente:Margarida Barbosa Silva,082282876,082282876@gmail.com,Castelo Branco,-55.801464,-63.15989
NovoCliente:Andre de Beir Castro Pinheiro Gama,591241570,591241570@gmail.com,Viana do Castelo,45.285843,-34.368214
NovoCliente:Mariana de Lima de Corte-Real,136057872,136057872@gmail.com,Coimbra,23.11937,58.16594
NovoCliente:Barbara de Beir,614873721,614873721@gmail.com,Faro,-57.60427,5.347275
NovoCliente:Xana Castro Pereira Fernandes Machado,758455275,758455275@gmail.com,Bragança,-17.025345,0.8855362
NovoCliente:Isabel Machado Oliveira de Beir,244040564,244040564@gmail.com,Guarda,-55.38125,-82.39434
NovoCliente:Carolina Silva Rodrigues,846648671,846648671@gmail.com,Guarda,72.28862,-7.2100296
NovoCliente:Fabio Rodrigues Figueiredo Cortez,869455462,869455462@gmail.com,Aveiro,-42.426903,-50.872723
NovoCliente:Armando Oliveira Veloso Antunes Castro,842601806,842601806@gmail.com,Viana do Castelo,-37.674744,-58.31765
herulume@tardis generator :) > █
```

# 14

# CondGen

```
type CondGen st a = StateT st Gen a
```

```
getGen :: CondGen st a -> st -> Gen a
```

```
getGen = evalStateT
```

```
genMultiCond :: (Arbitrary a, Eq b) => Int -> (a -> (a, b)) -> Gen [a]
```

```
genMultiCond n f = getGen (genMultiCond' n f) []
```

```
genMultiCondFrom :: (Arbitrary a, Eq c) => [b] -> (b -> a -> a) -> (a -> c) -> Gen [a]
```

```
genMultiCondFrom bs f g = getGen (genMultiCondFrom' f g) (bs, [])
```

```
genMultiFrom :: (Arbitrary a) => [b] -> (b -> a -> a) -> Gen [a]
```

```
genMultiFrom bs f = getGen (genMultiFrom' f) bs
```

# 15

# Pretty Printer

```
module Generator.PrettyPrinter where

class PrettyPrinter a where
  pp :: a -> String
```

# 16

## CondGen

```
genMultiCondFrom' :: (Arbitrary a, Eq c) => (b -> a -> a) -> (a -> c) -> CondGen ([b], [c]) [a]
genMultiCondFrom' f g = do
  (l, c) <- get
  if null l
  then return []
  else do
    z <- lift arbitrary
    let p = g z
    if p `elem` c
    then genMultiCondFrom' f g
    else do
      let n = f (head l) z
      put (tail l, p : c)
      u <- genMultiCondFrom' f g
      return (n : u)
```



17

# Dados

```
data Aluguer = Aluguer Nif Pos Pos Tipo Preferencia deriving (Show)

data Preferencia = MaisPerto | MaisBarato deriving (Show)

instance Arbitrary Preferencia where
  arbitrary = genPref

instance Arbitrary Aluguer where
  arbitrary = genAluguer

instance PrettyPrinter Aluguer where
  pp (Aluguer nif x y t p) = ("Aluguer:" <>) . join . intersperse "," $ [nif, show x, show y, show t, show p]

genAlugueres :: [Nif] -> Gen [Aluguer]
genAlugueres nifs = genMultiFrom nifs $ \nif (Aluguer _ x y t p) -> Aluguer nif x y t p

genAluguer :: Gen Aluguer
genAluguer = Aluguer <$> genNIF <*> genPos <*> genPos <*> arbitrary <*> arbitrary

genPref :: Gen Preferencia
genPref = elements [MaisPerto, MaisBarato]
```

# 18

## Análise Energética

- ▶ Log 0: 8000 utilizadores e por volta de 2000 alugueres e viaturas
- ▶ Log 1: 12000 utilizadores e por volta de 2000 alugueres e viaturas
- ▶ Log 2: 8000 utilizadores, 4000 alugueres e 2000 viaturas
- ▶ Log 3: 8000 utilizadores, 2000 alugueres e 4000 viaturas

# 19

	log0	log1	log2	log3
demo1 (com smells)	tempo: 1160 ms energia dram: 1.30 J energia cpu: 23.64 J energia package: 38.01 J	tempo: 1212 ms energia dram: 1.37 J energia cpu: 21.80 J energia package: 27.63 J	tempo: 2123 ms energia dram: 2.39 J energia cpu: 39.20 J energia package: 49.41 J	tempo: 2435 ms energia dram: 2.77 J energia cpu: 37.76 J energia package: 67.89 J
demo2 (sem smells)	tempo: 1238 ms energia dram: 1.43 J energia cpu: 22.33 J energia package: 28.30 J	tempo: 1576 ms energia dram: 1.77 J energia cpu: 25.04 J energia package: 32.61 J	tempo: 2205 ms energia dram: 2.29 J energia cpu: 34.46 J energia package: 44.98 J	tempo: 3024 ms energia dram: 3.47 J energia cpu: 43.60 J energia package: 58.13 J

	log0	log1	log2	log3
demo2 (com smells)	tempo: 6835 ms energia dram: 8.95 J energia cpu: 84.26 J energia package: 117.92 J	tempo: 7941 ms energia dram: 11.43 J energia cpu: 120.58 J energia package: 160.11 J	tempo: 20972 ms energia dram: 27.68 J energia cpu: 267.77 J energia package: 366.13 J	tempo: 9474 ms energia dram: 12.07 J energia cpu: 108.89 J energia package: 226.41 J
demo2 (sem smells)	tempo: 6717 ms energia dram: 11.21 J energia cpu: 100.07 J energia package: 117.92 J	tempo: 6484 ms energia dram: 9.57 J energia cpu: 83.26 J energia package: 115.56 J	tempo: 21400 ms energia dram: 30.66 J energia cpu: 263.48 J energia package: 374.42 J	tempo: 6933 ms energia dram: 10.81 J energia cpu: 93.42 J energia package: 128.21 J

## ParseDados

Encontrado por	Método	Smell	Resolução	Technical Debt
Human		Duplicated code; long methods	Remove duplications by extracting methods	60

# 20

## Conclusão



# Análise e Teste de Software

Bruno Martins A80410  
Eduardo Barbosa A83344  
Filipe Monteiro A80229  
Márcio Sousa A82400  
Miguel Brandão A82349