

Replace Query with Parameter

🎓 College: [Faculdade Metodista Granbery](#).

👤 Teacher: [Marco Antônio - Github](#) | [Linkedin](#)

📖 Book: [Refatoração - Aperfeiçoando o design de códigos existentes - Martin Fowler](#)

FOWLER, Martin. “Replace Query with Parameter” no código. *In*: REFATORAÇÃO: Aperfeiçoando o design de códigos existentes. 2. ed. [S. l.: s. n.], 2019. cap. 8.

Método

Objetivo desse método é trazer uma função que traga sempre o mesmo resultado que se chama transparência referencial. A função que foi criada, não acessar um referencial transparente, a função não possui transparência.

Código

O código que escolhi foi uma atividade dado em aula pelo Professor Ricardo. Essa função recebe um parâmetro de nomes de coluna do dataset, os dados são verificados pela sua severidade e depois ocorre sua substituição dos valores ausentes de acordo com o seu tipo.

O código pode ser encontrado na plataforma [Kaggle](#)

Ferramentas

```
pip install ipytest
```

```
pip install testbook
```

```
pip install pandas
```

```
pip install jupyter
```

Objetivo da atividade

1. Elabore ou escolha um código que a refatoração possa ser aplicada.
2. Identifique o mau cheiro associado.
3. Escreva os casos de teste e execute
4. Aplique a refatoração
5. Execute os casos de teste novamente
6. Documente cada etapa para a apresentação

Mau Cheiro

Lista longa de parâmetros

O mau cheiro que encontrei dentro do código abaixo foi o lista longa de parâmetros, por possuir consultas **longas e não claras**. Esse mal cheiro pode trazer **dificuldades de identificação ou erros no código**.

Procedimento

Código antigo

```

1 def tratar_ausentes():
2     for lista in mamografia.columns:
3         if lista != 'Idade':
4             moda_benigno = mamografia[(mamografia['Severidade'] == 0)][lista].mode()[0]
5             moda_maligno = mamografia[(mamografia['Severidade'] == 1)][lista].mode()[0]
6             mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 0), lista] = moda_benigno
7             mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 1), lista] = moda_maligno
8         else:
9             media_benigno = mamografia[(mamografia['Severidade'] == 0)][lista].mean()
10            media_maligno = mamografia[(mamografia['Severidade'] == 1)][lista].mean()
11            mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 0), lista] = media_benigno
12            mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 1), lista] = media_maligno

1 mamografia.columns

Index(['BI-RADS', 'Idade', 'Forma', 'Margem', 'Densidade', 'Severidade'], dtype='object')

1 tratar_ausentes()

1 mamografia.isnull().sum()

```

Código novo aplicando a refatoração

Use Extrair variável (Extract Variable) no código da consulta para separá-lo do restante do corpo da função.

```

def tratar_ausentes():
    colunas = refatoraMamografia.columns
    for lista in colunas:
        if lista != 'Idade':
            moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mode()[0]
            moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mode()[0]
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = moda_benigno
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = moda_maligno
        else:
            media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mean()
            media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mean()
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = media_benigno
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = media_maligno

```

Aplique Extrair função (Extract Function) no código do corpo que não seja a chamada da consulta.

Dê à nova função um nome fácil de pesquisar, para poder renomeá-la depois.

```
def XXXNEWsubstitui_valores_ausentes_com_moda(lista):
    moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mode()[0]
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = moda_benigno
    moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mode()[0]
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = moda_maligno

def XXXNEWsubstitui_valores_ausentes_com_media(lista):
    media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mean()
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = media_benigno
    media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mean()
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = media_maligno

def tratar_ausentes_refatora(): # tratar_ausentes_refatora() está escrito refatora, para não chamar a outra função tratar_ausentes()
    colunas = refatoraMamografia.columns
    for listaCol in colunas:
        if listaCol != 'Idade':
            XXXNEWsubstitui_valores_ausentes_com_moda(listaCol)
        else:
            XXXNEWsubstitui_valores_ausentes_com_media(listaCol)

    tratar_ausentes_refatora()
```

Use Internalizar variável (Inline Variable) para se livrar da variável que você acabou de criar.

```
def XXXNEWsubstitui_valores_ausentes_com_moda(lista):
    moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mode()[0]
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = moda_benigno
    moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mode()[0]
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = moda_maligno

def XXXNEWsubstitui_valores_ausentes_com_media(lista):
    media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mean()
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = media_benigno
    media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mean()
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = media_maligno

def tratar_ausentes_refatora(): # tratar_ausentes_refatora() está escrito refatora, para não chamar a outra função tratar_ausentes()
    for listaCol in refatoraMamografia.columns:
        if listaCol != 'Idade':
            XXXNEWsubstitui_valores_ausentes_com_moda(listaCol)
        else:
            XXXNEWsubstitui_valores_ausentes_com_media(listaCol)

    tratar_ausentes_refatora()
```

Aplice Internalizar função (Inline Function) na função original.

Renomeie a nova função dando-lhe o nome da função original.

```
def substitui_valores_ausentes_com_moda(lista):
    Silva, Today * projeto
    moda_benigno = refatoraMamografia[refatoraMamografia['Severidade'] == 0][lista].mode()[0]
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = moda_benigno
    moda_maligno = refatoraMamografia[refatoraMamografia['Severidade'] == 1][lista].mode()[0]
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = moda_maligno

def substitui_valores_ausentes_com_media(lista):
    media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista].mean()
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0), lista] = media_benigno
    media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista].mean()
    refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1), lista] = media_maligno

def tratar_ausentes_refatora(colunas): # tratar_ausentes_refatora() está escrito refatora, para não chamar a outra função tratar_ausentes()
    for listaCol in colunas:
        if listaCol != 'Idade':
            substitui_valores_ausentes_com_moda(listaCol)
        else:
            substitui_valores_ausentes_com_media(listaCol)

tratar_ausentes_refatora(refatoraMamografia.columns)
```

Realização dos Testes

The screenshot shows a JupyterLab environment with a Python test file named `test_tratamento_de_valores_ausentes.py`. The code defines a test function `test_retorna_substituicao_pela_moda` that checks if a list of values is not None. The test is executed, and the output shows that all tests passed successfully.

```

1 assert tratar_ausentes(lista) == None
2
3 @testbook('./notebookApp.ipynb', execute=True)
4 def test_retorna_substituicao_pela_moda(tb):
5     Silva, Today ~ projeto
6     moda = tb.ref("substitui_valores_ausentes_com_moda")
7     lista = ['BI-RADS', 'Forma', 'Margem', 'Densidade', 'Severidade']
8     for lis in lista:
9         assert moda(lis) == None
10
11     test_retorna_substituicao_pela_...
  
```

Run: Python tests in test_tratamento_de_valores_ausentes.py

Tests passed: 3 of 3 tests - 7 sec 249 ms

Test Results

- test_tratamento_de_valores_ausentes 7 sec 249 ms
- test_retorna_substituicao_valores_ausentes 2 sec 859 ms
- test_retorna_substituicao_pela_moda 2 sec 230 ms
- test_retorna_substituicao_pela_media 2 sec 160 ms

Testing started at 17:12 ...

===== test session starts =====

collecting ... collected 3 items

test_tratamento_de_valores_ausentes.py::test_retorna_substituicao_valores_ausentes PASSED [33%]

test_tratamento_de_valores_ausentes.py::test_retorna_substituicao_pela_moda PASSED [66%]

test_tratamento_de_valores_ausentes.py::test_retorna_substituicao_pela_media PASSED [100%]

===== 3 passed, 139 warnings in 7.81s =====