

# 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. 11.

## Método

Objetivo desse método é retirar a consulta dentro função que está em uma dependência indesejada e trazer uma função que traga sempre o mesmo resultado que se chama transparência referencial.

## 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, sendo que esse mal cheiro é para método 'Replace Parameter with Query', mas esse método e o inverso do 'Replace Query with Paramater', sendo assim o seu mal cheiro será o inverso. O mal cheiro será de **consultas longas** que trazem **confusão** para realizar a **compreensão do código**.

## Procedimento

### Código sem refatoração

```
def tratar ausentes(): Função
    for lista in mamografia.columns: Consulta
        if lista != 'Idade':
            moda_benigno = mamografia[(mamografia['Severidade'] == 0)][lista].mode()[0]
            moda_maligno = mamografia[(mamografia['Severidade'] == 1)][lista].mode()[0]
            mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 0), lista] = moda_benigno
            mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 1), lista] = moda_maligno
        else:
            media_benigno = mamografia[(mamografia['Severidade'] == 0)][lista].mean()
            media_maligno = mamografia[(mamografia['Severidade'] == 1)][lista].mean()
            mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 0), lista] = media_benigno
            mamografia.loc[(mamografia[lista].isnull()) & (mamografia['Severidade'] == 1), lista] = media_maligno
```

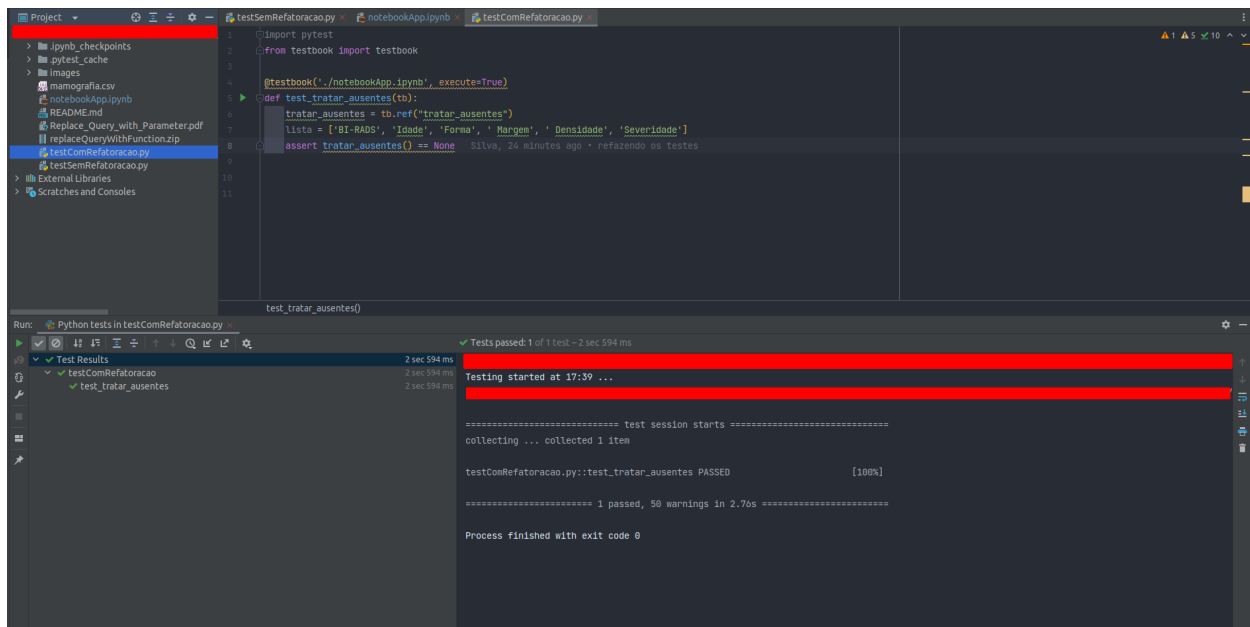
```
mamografia.columns Lista de colunas
Index(['BI-RADS', 'Idade', 'Forma', 'Margem', 'Densidade', 'Severidade'], dtype='object')
```

```
tratar_ausentes() Chamada da função
```

```
mamografia.isnull().sum()
```

```
BI-RADS      0
Idade         0
Forma         0
Margem        0
Densidade     0
Severidade    0
dtype: int64
```

## Teste sem refatoração



## Código novo aplicando a refatoração

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

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]  
            moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0)] = moda_benigno  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1)] = 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)] = media_benigno  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1)] = media_maligno
```

tratar\_ausentes()

refatoraMamografia.isnull().sum()

```
BI-RADS      0  
Idade         0  
Forma        0  
Margem       0  
Densidade    0  
Severidade    0  
dtype: int64
```

**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.**

**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.**

```
refatoraMamografia = dataset.copy() #Apenas usei novamente para retorna o valor do dataset com  
refatoraMamografia.isnull().sum()
```

```
BI-RADS      2  
Idade        5  
Forma       31  
Margem      48  
Densidade   76  
Severidade   0  
dtype: int64
```

```
def XXXNEWtratar_ausentes_refatora(listaColuna):  
    for lista in listaColuna:  
        if lista != 'Idade':  
            moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]  
            moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 0)] = moda_benigno  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1)] = 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)] = media_benigno  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull()) & (refatoraMamografia['Severidade'] == 1)] = media_maligno
```

```
def tratar_ausentes_refatora(): # tratar_ausentes_refatora() está escrito refatora, para não  
    colunas = refatoraMamografia.columns  
    return XXXNEWtratar_ausentes_refatora(colunas)
```

```
tratar_ausentes_refatora()
```

```
refatoraMamografia.isnull().sum()
```

```
BI-RADS      0  
Idade        0  
Forma        0  
Margem       0  
Densidade    0  
Severidade    0  
dtype: int64
```

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

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

```
: refatoraMamografia = dataset.copy() #Apenas usei novamente para retorna o valor do dataset c
refatoraMamografia.isnull().sum()

: BI-RADS      2
  Idade        5
  Forma       31
  Margem       48
  Densidade    76
  Severidade   0
  dtype: int64

: def XXXNEWtratar_ausentes_refatora(listaColuna):
  for lista in listaColuna:
    if lista != 'Idade':
      moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]
      moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Se
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Se
    else:
      media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]
      media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Se
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Se

: def tratar_ausentes_refatora(): # tratar ausentes_refatora() está escrito refatora, para não
  return XXXNEWtratar_ausentes_refatora(refatoraMamografia.columns)

: tratar_ausentes_refatora()
refatoraMamografia.isnull().sum()

: BI-RADS      0
  Idade        0
  Forma        0
  Margem       0
  Densidade    0
  Severidade   0
  dtype: int64
```

**Aplique Internalizar função (Inline Function) na função original.**

**Aplice Internalizar função (Inline Function) na função original.**

```
refatoraMamografia = dataset.copy() #Apenas usei novamente para retorna o valor do dataset c  
refatoraMamografia.isnull().sum()
```

```
BI-RADS      2  
Idade        5  
Forma       31  
  Margem    48  
  Densidade 76  
Severidade   0  
dtype: int64
```

```
def XXXNEWtratar_ausentes_refatora(listaColuna):  
    for lista in listaColuna:  
        if lista != 'Idade':  
            moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]  
            moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['S  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['S  
        else:  
            media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]  
            media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['S  
            refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['S
```

```
#def tratar_ausentes_refatora():  
  # tratar_ausentes_refatora() está escrito refatora, para não chamar a outra função tratar_
```

```
XXXNEWtratar_ausentes_refatora(refatoraMamografia.columns)
```

```
refatoraMamografia.isnull().sum()
```

```
BI-RADS      0  
Idade        0  
Forma        0  
  Margem     0  
  Densidade  0  
Severidade   0  
dtype: int64
```

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

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

```
: refatoraMamografia = dataset.copy() #Apenas usei novamente para retorna o valor do dataset c
refatoraMamografia.isnull().sum()

: BI-RADS      2
  Idade        5
  Forma       31
  Margem       48
  Densidade    76
  Severidade    0
dtype: int64

: def tratar_ausentes_refatora(listaColuna):
  for lista in listaColuna:
    if lista != 'Idade':
      moda_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]
      moda_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Severidade'] == 0)] = moda_benigno
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Severidade'] == 1)] = moda_maligno
    else:
      media_benigno = refatoraMamografia[(refatoraMamografia['Severidade'] == 0)][lista]
      media_maligno = refatoraMamografia[(refatoraMamografia['Severidade'] == 1)][lista]
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Severidade'] == 0)] = media_benigno
      refatoraMamografia.loc[(refatoraMamografia[lista].isnull())&(refatoraMamografia['Severidade'] == 1)] = media_maligno

: tratar_ausentes_refatora(refatoraMamografia.columns)

: refatoraMamografia.isnull().sum()

: BI-RADS      0
  Idade        0
  Forma        0
  Margem       0
  Densidade    0
  Severidade    0
dtype: int64
```

## Realização do Teste



