

# Desafio 01

Assim ela gostaria de realizar uma pesquisa com seus clientes. Conforme abaixo:

1. Qual sua faixa de Idade:
  - a. Até 30 Anos
  - b. De 30 a 50 anos
  - c. De 50 a 65 anos
  - d. Acima de 65 anos
  
2. Qual seu convênio:
  - a. INSS
  - b. SIAPE
  - c. Forças Armadas
  - d. Outros
  
3. Qual sua faixa salarial:
  - a. Até 2 SM
  - b. De 2 a 4 SM
  - c. De 4 a 6 SM
  - d. Acima de 6 SM
  
4. Por que você realizou o empréstimo:
  - a. Pagar contas
  - b. Reforma da Casa
  - c. Compra de Carro
  - d. Outras

O banco foi feito utilizando o MySQL Benchmark, sendo dividido em 3 tabelas:

Tabela de perguntas → onde o pgt\_ID é referenciado em outras tabelas para fins de manipulação dos dados no python.

	pgt_ID	pergunta_texto
▶	1	Qual a sua idade:
	2	Qual o seu convênio:
	3	Qual sua faixa salarial:
	4	Por que você realizou o empréstimo:
*	NULL	NULL

Tabela de respostas → alternativas\_id representa cada uma das alternativas. Estão posicionadas de 1 a 16. No python estão restringidos os pgt\_ID conforme alternativas\_id, por exemplo, as alternativas\_id de pgt\_ID = 1 estão no intervalo de 1 até 4.

	pgt_ID	alternativas_id	alternativas_texto
	2	6	SIAPE
	2	7	Forças Armadas
	2	8	Outros
	3	9	Até 2 SM
	3	10	De 2 a 4 SM
	3	11	de 4 a 6 SM
	3	12	Acima de 6 SM
	4	13	Pagar contas
	4	14	Reforma da casa
	4	15	Compra de carro
	4	16	Outras
*	NULL	NULL	NULL

Tabela de respostas → A tabela de respostas apenas faz o link entre o pgt\_ID e a escolha.

	pgt_ID	escolha
▶	0	1
	1	5
	2	10
	3	14
*	NULL	NULL

Nas imagens representas abaixo, pode-se o código realizado no python utilizando a IDE do pycharm.

Conexão com o banco de dados.

```
import mysql.connector

banco = mysql.connector.connect(
    host='localhost',
    user='root',
    password='',
    database='teste1',
)
print("Conectado")
```

Set de cursores de das variáveis utilizadas no código. Pode-se observar que ao final desta seção de código consiste um execute para limpar o banco. Utilizado apenas para fins de teste.

```
mycursor1_2 = banco.cursor(buffered=True)
mycursor1_2.execute("SELECT pergunta_texto from perguntas")
pgt = mycursor1_2.fetchall()
mycursor2_1 = banco.cursor(buffered=True)
mycursor2_1.execute("SELECT alternativas_texto from alternativas")
alt_txt = mycursor2_1.fetchall()
mycursor2_2 = banco.cursor(buffered=True)
mycursor2_2.execute("SELECT alternativas_id from alternativas")
alt_id = mycursor2_2.fetchall()
mycursor3_1 = banco.cursor(buffered=True)
mycursor3_1.execute("SELECT * from respostas")
choice = mycursor3_1.fetchone()
res = 0
val = []
mycursor = banco.cursor(buffered=True)
mycursor.execute("SELECT * from respostas")
sql = "INSERT INTO respostas(pgt_ID,escolha) VALUES(%s,%s)"
mycursor.execute("TRUNCATE TABLE respostas")
banco.commit()
```

O primeiro percorre de 1 até a *length* de **pgt\_ID(4)**. A primeira condição verifica a posição no for (i) para então escolher a pergunta. No for (j) seguinte, são impressas na tela as alternativas. O while logo abaixo garante que o valor digite no terminal seja aquele pertencente as alternativas daquela pergunta. Logo em seguida é feito um *append* com o **pgt\_ID(i)** e a alternativa escolhida **res**. Logo em seguida é executado o comando para inserir os valores obtidos na tabela de respostas.

Depois de cada teste de condição é zerada a lista **val**, para então receber novos valores.

```
for i in range(len(pgt)):
    if i == 0:
        print(pgt[i])
        for j in range(i*4, i*4+4):
            print(f"Digite {alt_id[j]} para {alt_txt[j]}")
        while res not in (1, 2, 3, 4):
            res = int(input(f"Digite um valor válido entre {i * 4 + 1} e {i * 4 + 4}: >>> "))
        val.append(i)
        val.append(res)
        print(val)
        mycursor.execute(sql, val)
    if i == 1:
        val.clear()
        print(pgt[i])
        for j in range(i*4, i*4+4):
            print(f"Digite {alt_id[j]} para {alt_txt[j]}")
        while res not in (5,6,7,8):
            res = int(input(f"Digite um valor válido entre {i * 4 + 1} e {i * 4 + 4}: >>> "))
        val.append(i)
        val.append(res)
        print(val)
        mycursor.execute(sql, val)
```

```
    if i == 2:
        val.clear()
        print(pgt[i])
        for j in range(i*4, i*4+4):
            print(f"Digite {alt_id[j]} para {alt_txt[j]}")
        while res not in (9,10,11,12):
            res = int(input(f"Digite um valor válido entre {i * 4 + 1} e {i * 4 + 4}: >>> "))
        val.append(i)
        val.append(res)
        print(val)
        mycursor.execute(sql, val)
    if i == 3:
        val.clear()
        print(pgt[i])
        for j in range(i*4, i*4+4):
            print(f"Digite {alt_id[j]} para {alt_txt[j]}")
        while res not in (13,14,15,16):
            res = int(input(f"Digite um valor válido entre {i * 4 + 1} e {i * 4 + 4}: >>> "))
        val.append(i)
        val.append(res)
        print(val)
        mycursor.execute(sql, val)
```

A tela de execução do código fica conforme as imagens abaixo

```
('Qual a sua idade:',)
Digite (1,) para ('Até 30 Anos',)
Digite (2,) para ('De 30 a 50 anos',)
Digite (3,) para ('De 50 a 65 anos',)
Digite (4,) para ('Acima de 65 anos',)
Digite um valor válido entre 1 e 4: >>> 5
Digite um valor válido entre 1 e 4: >>> 1
[0, 1]
('Qual o seu convênio:',)
Digite (5,) para ('INSS',)
Digite (6,) para ('SIAPE',)
Digite (7,) para ('Forças Armadas',)
Digite (8,) para ('Outros',)
Digite um valor válido entre 5 e 8: >>> 5
[1, 5]
```

```
('Qual sua faixa salarial:',)
Digite (9,) para ('Até 2 SM',)
Digite (10,) para ('De 2 a 4 SM',)
Digite (11,) para ('de 4 a 6 SM',)
Digite (12,) para ('Acima de 6 SM',)
Digite um valor válido entre 9 e 12: >>> 10
[2, 10]
('Por que você realizou o empréstimo:',)
Digite (13,) para ('Pagar contas',)
Digite (14,) para ('Reforma da casa',)
Digite (15,) para ('Compra de carro',)
Digite (16,) para ('Outras',)
Digite um valor válido entre 13 e 16: >>> 14
[3, 14]
```