

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

📝 Descrição da 30ª Atividade: 🌟

⌚ INDIVIDUAL Leiam o case abaixo e resolvam. Uma escola está organizando sua primeira olimpíada do conhecimento e deseja separar os 100 alunos em dois grupos de 50. Além de testar os conhecimentos dos alunos, querem estimular a formação de novos laços sociais e, por isso, a divisão dos grupos de alunos será feita seguindo um critério:

Alunos com número de matrícula par, ficarão no grupo azul. Alunos com número de matrícula ímpar, ficarão no grupo amarelo.

Os alunos ainda não sabem dessa regra de separação dos grupos e, no dia do evento, quando digitarem o número da matrícula na catraca, deve aparecer no painel a cor do grupo que ele deve integrar.

DESAFIO: Desenvolvam uma função para retornar se o número passado pelo usuário no console é par ou ímpar.

Caso o número de matrícula do(a) aluno(a) seja par imprima: VOCÊ ESTÁ NO TIME AZUL

Caso o número de matrícula do(a) aluno(a) seja ímpar imprima: VOCÊ ESTÁ NO TIME AMARELO.

```
def verificar_grupo(numero_matricula):
    if numero_matricula % 2 == 0:
        print("VOCÊ ESTÁ NO TIME AZUL")
    else:
        print("VOCÊ ESTÁ NO TIME AMARELO")

numero_matricula = int(input("Digite o número da matrícula: "))

verificar_grupo(numero_matricula)
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade30.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

💡 Descrição da 31ª Atividade: 🌟

💡 EM SQUADS Agora, criem um script para:

- Ter um input de usuário para inserir os números de matrícula em uma lista.
- Ter um validador nessa lista que permita a inserção de dados até ocupar 5 espaços index.
- Fazer um laço de repetição para passar todos os números da lista em uma função para verificar se o número é par ou ímpar.

```
numeros_de_matricula = []

while len(numeros_de_matricula) < 5:
    numero = input("Digite um número de matrícula ou 'q' para sair: ")

    if numero.lower() == 'q':
        break

    if numero.isdigit():
        numeros_de_matricula.append(int(numero))
    else:
        print("Por favor, insira um número válido.")

for numero in numeros_de_matricula:
    if numero % 2 == 0:
        print(f"{numero} é um número de matrícula par.")
    else:
        print(f"{numero} é um número de matrícula ímpar.)
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade31.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

💡 Descrição da 32ª Atividade: 🌟

🔍 EM SQUAD Leiam o case e resolvam a situação. A Loja do Joga Junto conta mais uma vez com a colaboração do seu squad! Desta vez, surge a necessidade de desenvolver um programa que analisa o CEP inserido pelo usuário e determina se ele é elegível para frete grátis. Para realizar essa tarefa, foi definida uma política de frete grátis abrangendo todos os estados das regiões Norte e Nordeste do país.

- Faça um brainstorming com sua equipe sobre o fluxo e requisitos necessários para construção desse programa
- Desenvolva o programa
- Faça casos de teste para este cenário, documente os testes realizados e insira no Bitrix
- Caso seja encontrado algum bug no seu código, documente-o.

Brainstorming:

1. Requisitos Principais:

- O programa deve receber o CEP do usuário.
- Deve verificar se o CEP pertence a um estado das regiões Norte ou Nordeste.
- Se o CEP estiver nas regiões elegíveis, o frete deve ser marcado como grátis.
- Caso contrário, o frete não deve ser grátis.

2. Interface de Usuário:

- Criar uma interface simples para que o usuário possa inserir o CEP.
- Exibir uma mensagem clara sobre a elegibilidade do frete grátis após a verificação.

3. Lógica de Verificação:

- Criar uma lista de estados que fazem parte das regiões Norte e Nordeste.
- Verificar se o estado associado ao CEP está na lista de estados elegíveis.

4. Testes:

- Realizar testes com CEPs de diferentes estados e regiões para garantir que o programa esteja funcionando corretamente.
- Documentar os casos de teste e os resultados esperados.

Desenvolvimento do programa:

```
import requests

def verificar_frete_gratis(cep):
    cep_formatado = ''.join(filter(str.isdigit, cep))

    url = f"https://viacep.com.br/ws/{cep_formatado}/json/"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        estado = data.get("uf", "").upper()

        estados_elegiveis = ["AC", "AL", "AM", "AP", "BA", "CE", "MA", "PA", "PB",
"PE", "PI", "RN", "RO", "RR", "TO"]

        if estado in estados_elegiveis:
            return True
        else:
            return False
    else:
        print("Erro ao consultar o CEP. Verifique se o CEP é válido.")
        return False

cep_usuario = input("Digite o CEP: ")

if verificar_frete_gratis(cep_usuario):
    print("Frete grátis disponível para o seu CEP!")
else:
    print("Não há frete grátis para o seu CEP.")
```

Casos de teste:

De acordo com a atividade, abaixo constam o que podem ser possíveis casos de testes para esse cenário.

Caso de Teste 1:

- CEP: 69000-000 (Amazonas)
- Resultado Esperado: Frete grátis disponível para o seu CEP!

Caso de Teste 2:

- CEP: 87000-000 (Paraná)
- Resultado Esperado: Não há frete grátis para o seu CEP.

Caso de Teste 3:

- CEP: 60000-000 (Ceará)
- Resultado Esperado: Frete grátis disponível para o seu CEP!

Caso de Teste 4:

- CEP: 15000-000 (São Paulo)
- Resultado Esperado: Não há frete grátis para o seu CEP.

Caso de Teste 5:

- CEP: 58000-000 (Paraíba)
 - Resultado Esperado: Frete grátis disponível para o seu CEP!
-

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade32.py.

Integrantes da Squad:

| Beatriz Souza | [Bruno Soares](#) | [Leanderson Lima](#) | [Rebeca Borges](#) | Sara Cruz |