

Atividade 3

Manipulação de Dados no Redis 🚀

1. Implemente em Python a retirada de 3 itens do Projeto Mercado Livre armazenado no MongoDB para o Redis

a. Retirar do MongoDB e colocar no Redis

```
def retirar_do_mongo_para_redis(email):
    try:
        cliente = clientes_collection.find_one({"email": email})

        if cliente:
            favoritos = cliente.get("favoritos", [])

            itens = []
            ids_adicionados = set()

            for produto_id in favoritos[:3]:
                try:
                    produto = produtos_collection.find_one({"_id": ObjectId(produto_id)})

                    if produto and produto_id not in ids_adicionados:
                        print(f"Produto encontrado: {produto}")
                        itens.append({
                            "id_produto": str(produto.get("_id")),
                            "nome_produto": produto.get("nome", "Produto sem nome"),
                            "preco": produto.get("preco", 0)
                        })
                        ids_adicionados.add(produto_id)
                    else:
                        print(f"Produto não encontrado ou já adicionado com ID: {produto_id}")
                except Exception as e:
                    print(f"Erro ao buscar produto com ID {produto_id}: {e}")

            salvar_redis(email, itens, "itens")
        else:
            print("Cliente não encontrado!")
    except Exception as e:
        print(f"Erro ao retirar do MongoDB para Redis: {e}")
```

b. Manipular os itens no Redis

```
def manipular_itens_no_redis(email):
    chave = f'{email}:itens'

    if conR.exists(chave):
        itens_json = conR.get(chave)
        itens = json.loads(itens_json)

        if itens:
            print("Itens carregados:")
            for idx, item in enumerate(itens):
                print(f"{idx + 1}. ID: {item['id_produto']], Nome: {item['nome_produto']], Preço: {item['preco']}")

            escolha = int(input("Digite o número do item que deseja alterar o preço (0 para cancelar): "))
            if 1 <= escolha <= len(itens):
                novo_preco = float(input("Novo preço: "))
                itens[escolha - 1]["preco"] = novo_preco
                conR.set(chave, json.dumps(itens, default=json_serializer))
                print(f"Item atualizado: {itens[escolha - 1]}")
            else:
                print("Operação cancelada.")
        else:
            print("Nenhum item encontrado no Redis.")
    else:
        print("Não há itens armazenados no Redis!")
```

c. Devolver os itens para o MongoDB

```
def devolver_para_mongo(email):
    chave = f'{email}:itens'

    if conR.exists(chave):
        itens_json = conR.get(chave)
        itens = json.loads(itens_json)

        cliente = clientes_collection.find_one({"email": email})
        if cliente:
            favoritos = cliente.get("favoritos", [])

            novos_favoritos = favoritos[:3] + [item["id_produto"] for item in itens if item["id_produto"] not in favoritos]

            clientes_collection.update_one({"email": email}, {"$set": {"favoritos": novos_favoritos}})

            for item in itens:
                produtos_collection.update_one(
                    {"_id": ObjectId(item["id_produto"])},
                    {"$set": {"preco": item["preco"]}},
                    upsert=False
                )

            print("Itens devolvidos ao MongoDB!")
        else:
            print("Cliente não encontrado no MongoDB!")
    else:
        print("Não há itens armazenados no Redis para devolver!")
```