

## Introdução à Programação

### Lab 2

**Data de Entrega:** 19/04/2024

**Lab feito em duplas**

#### 1) Orientações:

Caro aluno, você está prestes a iniciar três questões de programação criadas para solidificar seu conhecimento e habilidades Pythonicas. A ideia aqui é diferente a do Lab#1, onde desenvolvemos um programa inteiro. Cada desafio foi cuidadosamente elaborado para ajudá-lo a aplicar os conceitos que aprendeu em sala de aula, como funções, estruturas de dados, loops, condicionais e manipulação de dados. Encorajo você a abordar cada desafio com curiosidade e determinação e recomendo a seguinte forma de busca ajudar:

- [Stackoverflow](#) - faça perguntas assim: “como fazer X em Python” (tente escrever inglês)
- ChatGPT - as únicas perguntas válidas são “que função uso para fazer X em Python”. Qualquer outra forma de perguntar ao Chat corre o risco de levar à uma resposta pronta

#### 2) Dicas Gerais

**Leia Cuidadosamente:** antes de começar a codar, leia cada desafio atentamente para entender exatamente o que está sendo solicitado - pense em termos de “quais os requisitos?”

**Divida o Problema em partes menores (e teste elas!!):** sempre divida o desafio em partes menores e aborde cada parte separadamente. Lembre-se partes menores = maneira de simplificar o problema!

**Esboce uma Solução:** Antes de escrever o código, pense em como você abordaria o problema e esboce uma solução. Tente escrever pseudocódigo ou desenhar um *fluxograma*!!

**Teste seu Código:** Conforme você desenvolve sua solução, teste-a com diferentes entradas para garantir que ela funcione como esperado. Verifique se todos cenários foram considerados.

**Peça Ajuda:** Se você ficar preso, peça ajuda no espaço para discussão do Teams!

### **Questão 1: gerador de senha aleatória**

Escreva uma função que gere uma senha aleatória a partir de letras maiúsculas e minúsculas, números e os seguintes caracteres especiais (!@#\$%^&\*). Ela deve ter um argumento para o comprimento da senha (quantos dígitos) e deve verificar se o comprimento está entre 8 e 16, caso contrário, deve alertar o usuário e encerrar. Sua função também deve ter um parâmetro chamado "special\_chars" que por padrão é 'True'. Se a função for chamada com o parâmetro igual à 'False', então os valores aleatórios escolhidos não devem incluir caracteres especiais.

***Dica 1: lembre-se da função 'choice' da biblioteca 'random'***

***Dica 2: strings (") possuem um método chamado '.join' capaz de concatená-las***

### **Questão 2: análise de vendas**

Escreva uma função que analise os dados de vendas de uma loja. A função deve receber uma lista de dicionários, onde cada dicionário representa a venda de um item e contém as chaves "item", "quantidade" e "preço". A função deve retornar o item mais vendido, o item que gerou mais receita e e qual foi a receita total desse item

***Dica 1: Para encontrar o item mais vendido e o que gerou mais receita, pense em como somar as quantidades e os totais de vendas por item, possivelmente usando um loop ou uma abordagem com dicionários.***

### **Questão 3: sistema de classificação de notas**

Crie uma função que classifique os alunos com base em suas notas. A função deve aceitar um dicionário onde as chaves são os nomes dos alunos e os valores são suas notas. A função deve retornar dois dicionários: um com os alunos classificados do maior para o menor (ordem decrescente de notas) e outro do menor para o maior (ordem crescente de notas). Em caso de empate, os alunos podem ser listados em qualquer ordem entre os empatados. A função também deve imprimir o nome dos alunos com a maior e a menor menor nota.

***Dica 1: Utilize a função sorted() para organizar os alunos por nota, lembrando de considerar a ordem crescente e decrescente.***

***Dica 2: Para a impressão dos resultados, percorra o dicionário classificado, imprimindo a posição com base no índice.***