

## 50 Exercícios de Python: Listas, Dicionários e POO (Médio a Avançado)

| # | Título do Exercício           | O que fazer (Detalhes)  | Ferramentas Sugeridas                                   |
|---|-------------------------------|---|---|
| 1 | Interseção de Listas          | Crie uma <b>função</b> que receba duas listas de números inteiros geradas com <b>random</b> . Retorne uma <b>nova lista</b> contendo apenas os elementos que estão presentes em ambas as listas, sem repetição. | função, random, set (para eficiência), print            |
| 2 | Rotação de Lista              | Crie uma <b>função</b> que receba uma lista e um inteiro k. A função deve rotacionar a lista k posições para a direita. Use <i>slicing</i> de lista para resolver.  | função, list slicing, print                             |
| 3 | Remoção Condicional com While | Use um <i>loop while</i> para iterar sobre uma lista de strings. Remova todas as strings que contêm a letra 'a' (minúscula ou maiúscula).<br><b>Atenção:</b> Mudar o tamanho da lista                           | while, list.pop()/list.remove(), str.lower(), if, print |

|   |                          |   |   |
|---|--------------------------|---|---|
|   |                          | durante a iteração requer cuidado (ex: iterar de trás para frente ou usar um índice while).   |   |
| 4 | Compressão de Lista      | Dada uma lista, use <b>list comprehension</b> para criar uma nova lista que retém apenas os números, descartando strings, booleanos ou outros tipos de dados.     | list comprehension, isinstance(), if, print           |
| 5 | Encontrar Duplicatas     | Crie uma <b>função</b> que receba uma lista. Retorne uma <b>lista de tuplas</b> , onde cada tupla contém o elemento duplicado e sua frequência na lista original. | função, for, dict (para contagem), list, print        |
| 6 | Transposição de Matriz   | Dada uma lista de listas (matriz \$N \times M\$), crie uma <b>função</b> para transpor a matriz (linhas viram colunas). Use <i>list comprehension</i> aninhada.   | função, list comprehension (aninhada), range(), print |
| 7 | Maior Sequência Contígua | Crie uma <b>função</b> que receba uma lista de números e  | função, for, if/else, max(), print                    |

|    |                         |  |   |
|----|-------------------------|--|---|
|    |                         | retorne a <b>soma máxima</b> de qualquer sublista contígua (algoritmo de Kadane - Avançado).   |   |
| 8  | Unir e Somar Listas     | Crie uma <b>função</b> que receba duas listas de números. Retorne uma nova lista que é a soma dos elementos correspondentes (índice por índice). Se uma lista for mais longa, adicione os elementos restantes da lista mais longa. Use <b>try/except</b> para lidar com <code>IndexError</code> na lista mais curta. | função, for, try/except, len(), print       |
| 9  | Fatiamento Circular     | Crie uma <b>função</b> que receba uma lista e dois índices: início e fim. A função deve retornar o fatiamento (slice) da lista, mesmo que o índice final seja menor que o inicial, simulando um acesso circular.   | função, list slicing, len(), if/else, print |
| 10 | Filtro Random de Listas | Crie uma <b>função</b> que receba uma  | função, random.sample(),                    |

|  |  |  |                   |
|--|--|--|-------------------|
|  |  | <p>lista e um número n. Use o módulo <b>random</b> para retornar uma nova lista contendo n elementos aleatórios únicos da lista original. Use <b>try/except</b> para lidar com o caso de \$n\$ ser maior que o tamanho da lista.</p> | try/except, print |
|--|--|--|-------------------|

## Categoria 2: Manipulação e Análise de Dicionários (Médio a Avançado)

| #  | Título do Exercício              | O que fazer (Detalhes)   | Ferramentas Sugeridas                                |
|----|----------------------------------|--|--|
| 11 | Contagem de Palavras com Exceção | <p>Refaça o exercício de frequência de palavras (Contagem) no dicionário. Use <b>try/except</b> dentro do <i>loop</i> para verificar se a palavra já existe como chave no dicionário, incrementando ou inicializando a contagem.</p> | for, str.split(), try/except (KeyError), dict, print |
| 12 | Inversão e Filtro de Dicionário  | <p>Crie uma <b>função</b> que inverte um dicionário (chave</p>   | função, dictionary comprehension, len(), if, print   |

|    |                             |  |   |
|----|-----------------------------|--|---|
|    |                             | <p><code>\$\leftarrow\$</code> valor). Use <b>dictionary comprehension</b> para filtrar e incluir no novo dicionário apenas os pares onde o valor original era uma string com mais de 5 caracteres.</p>  |   |
| 13 | Dicionário de Metadados     | <p>Dado um dicionário de números. Crie uma <b>função</b> que gere um novo dicionário contendo metadados. As chaves devem ser as mesmas, mas os valores devem ser <b>dicionários</b> com as chaves: <code>tipo</code>, <code>tamanho_string</code> (da chave) e <code>é_par</code>.</p> | <p>função, dict (aninhado), <code>type()</code>, <code>len()</code>, <code>if</code>, <code>print</code></p>  |
| 14 | Soma Condicional de Valores | <p>Crie um dicionário com chaves (strings) e valores (números). Use um <b>for</b> loop para somar apenas os valores cujas chaves comecem com uma letra específica. Use um <b>try/except</b> para garantir que apenas valores numéricos sejam</p>                                       | <p><code>for</code>, <code>dict.items()</code>, <code>str.startswith()</code>, <code>try/except</code> (<code>TypeError</code>), <code>print</code></p> |

|    |                               |  |  |
|----|-------------------------------|--|--|
|    |                               | somados (ignorando TypeError).   |  |
| 15 | Remoção Aleatória             | Use o módulo <b>random</b> para selecionar e remover <b>3 pares chave-valor aleatórios</b> de um dicionário. Se o dicionário tiver menos de 3 itens, use <b>try/except</b> para lidar com o erro e limpar o restante.                        | random.sample(), dict.pop(), try/except, print |
| 16 | Flatten Dictionary (Avançado) | Crie uma <b>função</b> que recebe um dicionário aninhado (chaves e valores também são dicionários, apenas um nível de aninhamento). A função deve "achatar" o dicionário, concatenando as chaves aninhadas com um separador (ex: pai_filho). | função, dict.items(), for (aninhado), str      |
| 17 | Ordenação por Valor           | Crie um dicionário. Use <b>sorted()</b> para retornar uma <b>lista de tuplas</b> (chave, valor) ordenadas pelo <b>valor</b> do dicionário em   | sorted(), dict.items(), lambda, print          |

|    |                                     |  |  |
|----|-------------------------------------|--|--|
|    |                                     | ordem decrescente.   |  |
| 18 | Atualização com <i>Loop e Input</i> | Use um <i>loop</i> <b>while</b> que peça ao usuário uma chave e um novo valor. Use <b>try/except</b> para adicionar ou atualizar a chave no dicionário. O loop para quando o usuário digitar uma chave vazia.  | while, input(), try/except, dict, print              |
| 19 | Dicionário de Listas                | Crie um dicionário onde a chave é o nome de um usuário e o valor é uma <b>lista</b> de suas permissões. Crie uma <b>função</b> que receba o dicionário, um usuário e uma permissão. Se o usuário existir, adicione a permissão (se ainda não existir). | função, dict, list.append(), if/else, print          |
| 20 | Validação de Chaves                 | Crie uma <b>função</b> que receba um dicionário. Use <b>dict comprehension</b> para criar um novo dicionário onde todas as chaves são strings em <b>maiúsculas</b> , e os valores permanecem os  | função, dictionary comprehension, str.upper(), print |

|  |  |         |  |
|--|--|---------|--|
|  |  | mesmos. |  |
|--|--|---------|--|

### Categoria 3: Listas de Dicionários (POO e Integração)

| #  | Título do Exercício            | O que fazer (Detalhes)  | Ferramentas Sugeridas            |
|----|--------------------------------|---|----------------------------------|
| 21 | Classe Base ItemInventario     | Crie uma <b>classe</b> ItemInventario com atributos: nome (string), quantidade (int) e valor_unitario (float). Implemente um método valor_total().        | class, __init__, função (método) |
| 22 | Classe Perecível (Herança)     | Crie uma <b>classe</b> Perecível que <b>herda</b> de ItemInventario. Adicione o atributo data_validade. Sobrescreva o método __str__ para incluir a data. | class, herança, super(), __str__ |
| 23 | Classe Inventario (Composição) | Crie uma <b>classe</b> Inventario. O atributo principal deve ser uma <b>lista</b> (estoque) de objetos ItemInventario e Perecível.                        | class, list, objeto              |
| 24 | Adicionar e                    | Na classe   | função (método),                 |



|    |                            |   |   |
|----|----------------------------|---|---|
|    | Remover                    | Inventario, implemente métodos para: 1) <b>Adicionar</b> um item à lista de estoque. 2) <b>Remover</b> um item pelo nome. Use <b>try/except</b> para lidar com o erro se o item a ser removido não for encontrado.  | list.append(), for, if, try/except              |
| 25 | Relatório de Baixo Estoque | Na classe Inventario, crie um método que utilize <b>list comprehension</b> para retornar uma nova lista contendo os nomes de todos os itens cuja quantidade seja inferior a 5.  | função (método), list comprehension, if, objeto |
| 26 | Venda com While            | Na classe Inventario, crie um método vender_item(nome, quantidade_vendida). Use um <i>loop while</i> para permitir a venda. O <i>loop</i> deve garantir que a quantidade não se torne negativa. Use <b>try/except</b> para lidar com o erro de item não encontrado. | while, if, try/except, função (método)          |

|    |                                   |  |  |
|----|-----------------------------------|--|--|
| 27 | Lista de Dicionários para Objetos | Crie uma <b>função</b> que receba uma <b>lista de dicionários</b> (simulando um JSON). Cada dicionário representa um item. A função deve converter e retornar uma <b>lista de objetos</b> ItemInventario.            | função, for, dict, class, objeto                       |
| 28 | Busca e Edição                    | Na classe Inventario, crie um método que receba o nome de um item e um novo preço. Use um <b>for</b> loop para buscar o objeto e atualizar seu valor_unitario.   | função (método), for, objeto, print                    |
| 29 | Ordenação por Validade            | Na classe Inventario, crie um método que retorne a lista de estoque <b>ordenada</b> pelos itens Perecível com a data de validade mais próxima. Use a função sorted() com uma lambda e lide com itens não-perecíveis. | função (método), sorted(), lambda, isinstance(), print |
| 30 | Herança Múltipla ItemComImposto   | Crie uma classe CalculoImpostoMixin com um método  | class, herança múltipla, função (método)               |

|  |  |  |  |
|--|--|--|--|
|  |  | calcular_imposto(taxa). Crie uma classe ItemComImposto que <b>herda</b> de ItemInventario e CalculoImpostoMixin. O novo método valor_total deve incluir o imposto. |  |
|--|--|--|--|

#### Categoria 4: Desafios de Strings e Dicionários com Try/Except e Random

| #  | Título do Exercício            | O que fazer (Detalhes)  | Ferramentas Sugeridas                      |
|----|--------------------------------|---|--|
| 31 | Geração de Matrícula Aleatória | Crie uma <b>função</b> que gere um ID de matrícula de 8 caracteres, sendo os 4 primeiros letras maiúsculas aleatórias e os 4 últimos dígitos aleatórios. Use <b>random</b> e métodos de string (como str.join()). | função, random.choice(), str.join(), print |
| 32 | Criptografia Simples           | Crie uma <b>função</b> que receba uma string. Use um <b>dicionário</b> para mapear cada letra (chave) para uma letra cifrada  | função, dict, for, str                     |

|    |                             |   |  |
|----|-----------------------------|---|--|
|    |                             | aleatória (valor). Retorne a string cifrada.  |  |
| 33 | Verificação de Palíndromo   | Crie uma <b>função</b> que verifique se uma frase (string) é um palíndromo (ignorando espaços, pontuação e caixa alta/baixa). Use métodos de string e <i>list comprehension</i> .   | função, str.lower(), str.replace(), list comprehension, if/else, print |
| 34 | Jogo da Forca Simples       | Use a biblioteca <b>random</b> para escolher uma palavra secreta em uma lista. Use um <i>loop while</i> para permitir 5 tentativas. Armazene as letras já chutadas em uma <b>lista</b> e use um if/else para imprimir a palavra com as letras escondidas (ex: P _ T H _ N). | random.choice(), while, list, if/else, print                           |
| 35 | Validação de CPF (Simulada) | Crie uma <b>função</b> que receba um CPF (string). Use <b>try/except</b> para garantir que a string tem 11 caracteres e que todos são dígitos (lançando uma   | função, try/except (ValueError), len(), str.isdigit(), print           |

|    |                                       |   |   |
|----|---------------------------------------|---|---|
|    |                                       | exceção se falhar).<br>Retorne o CPF<br>formatado (ex:<br>000.000.000-00).  |   |
| 36 | Formatação de<br>Dicionário           | Crie uma <b>função</b><br>que receba um<br>dicionário. Retorne<br>uma única <b>string</b><br>formatada onde<br>cada par<br>chave-valor está<br>em uma nova linha,<br>com a chave<br>justificada à<br>esquerda e o valor<br>justificado à direita<br>(usando métodos<br>de string como<br>str.ljust() ou<br><i>f-strings</i><br>formatadas). | função, for,<br>dict.items(),<br>f-string, str.ljust(),<br>print            |
| 37 | Lista de<br>Argumentos<br>Arbitrários | Crie uma <b>função</b><br>que aceite um<br>número arbitrário<br>de argumentos<br>nomeados<br>(**kwargs). A<br>função deve usar<br>um <b>for</b> loop para<br>imprimir as chaves<br>e valores, mas<br>apenas para as<br>chaves que<br>começam com 'r'.   | função (**kwargs),<br>for, str.startswith(),<br>print                       |
| 38 | Busca em<br>Dicionário<br>Aninhado    | Crie uma <b>função</b><br>que receba um<br>dicionário aninhado<br>(múltiplos níveis). A   | função (recursiva<br>ou com while),<br>dict.items(),<br>isinstance(), print |

|    |                                  |   |   |
|----|----------------------------------|---|---|
|    |                                  | função deve procurar uma chave específica e retornar o valor correspondente, não importa o quão fundo ela esteja aninhada.  |   |
| 39 | Remoção com <i>Slice</i>         | Dada uma lista de 20 números aleatórios. Use <i>list slicing</i> (fatiamento) para remover os 5 elementos centrais da lista em uma única linha.   | random, list slicing, len(), print          |
| 40 | Validação de Tipo com Try/Except | Crie uma <b>função</b> que tente adicionar um valor a uma chave em um dicionário. Use <b>try/except</b> para verificar se o valor já existente é um número. Se for, some o novo valor. Se não for, substitua o valor existente pelo novo, e imprima um aviso. | função, try/except (TypeError), dict, print |

## Categoria 5: Listas e Dicionários para Problemas Reais (Avançado)

| # | Título do Exercício | O que fazer (Detalhes) | Ferramentas Sugeridas |
|---|---------------------|------------------------|-----------------------|
|---|---------------------|------------------------|-----------------------|

|    |                                       |  |   |
|----|---------------------------------------|--|---|
| 41 | Simulador de Pilha (LIFO)             | Crie uma <b>classe</b> Pilha. Use uma <b>lista</b> como estrutura de dados interna. Implemente os métodos push(), pop() (com try/except para lista vazia) e is_empty().  | class, list, função (método), try/except (IndexError) |
| 42 | Simulador de Fila (FIFO)              | Crie uma <b>classe</b> Fila. Use uma <b>lista</b> como estrutura de dados interna. Implemente os métodos enqueue() e dequeue() (com try/except para lista vazia).  | class, list, função (método), try/except (IndexError) |
| 43 | Sistema de Reserva (Dicionário)       | Crie uma <b>classe</b> Reservas. Use um <b>dicionário</b> para armazenar reservas, onde a chave é o ID da reserva (aleatório, gerado pelo random) e o valor é um <b>dicionário</b> com detalhes do cliente. Implemente fazer_reserva() e cancelar_reserva(). | class, dict, random.randint(), função (método)        |
| 44 | Gerenciador de Tarefas (POO Avançado) | Crie uma <b>classe</b> Tarefa com atributos nome, prioridade (1 a 5) e concluida (bool). Crie uma <b>classe</b>  | class, list, função (método), sorted(), lambda        |

|    |                                  |  |  |
|----|----------------------------------|--|--|
|    |                                  | <p>ListaDeTarefas que use uma <b>lista</b> de objetos Tarefa. Implemente um método para <b>ordenar</b> a lista por prioridade (usando sorted e lambda).</p>  |  |
| 45 | Herança e Polimorfismo com Notas | <p>Crie uma <b>classe</b> Avaliação com método calcular_media(). Crie classes filhas MediaAritmetica e MediaPonderada (com atributo pesos, que é uma lista), que <b>herdam</b> de Avaliação e <b>sobrescrevem</b> o método calcular_media().</p> | class, herança, função (polimorfismo), super(), list     |
| 46 | Simulação de Jogo de Dados       | <p>Crie uma <b>função</b> que simule 1000 lançamentos de um dado de 6 faces (use <b>random</b>). Armazene a frequência de cada face em um <b>dicionário</b>. Imprima a face mais e menos frequente.</p>  | função, random.randint(), dict, for, max(), min(), print |
| 47 | Matriz Esparsa (Dicionário)      | <p>Em uma matriz muito grande, a maioria dos valores é zero. Use um</p>  | função, dict, tupla, dict.get(), print                   |



|    |                                      |   |   |
|----|--------------------------------------|---|---|
|    |                                      | <p><b>dicionário</b> onde a chave é uma <b>tupla</b> (linha, coluna) e o valor é o número diferente de zero. Crie uma <b>função</b> para "ler" o valor de uma célula. Se a tupla não estiver no dicionário, retorne 0.</p>  |   |
| 48 | Sistema de Cache (Avançado)          | <p>Crie uma <b>classe</b> CacheLRU (Least Recently Used). Use um <b>dicionário</b> para armazenar o cache e uma <b>lista</b> para rastrear a ordem de uso. Sempre que um item é acessado (busca no dicionário), ele é movido para o topo da lista. Se o cache estiver cheio (tamanho máximo), remova o item do final da lista (LRU) antes de adicionar um novo.</p> | <p>class, dict, list, função (método), list.remove(), list.insert(), print</p>          |
| 49 | Validação de Estrutura de Dicionário | <p>Crie uma <b>função</b> que receba um dicionário e um <b>modelo</b> (outro dicionário com os tipos de dados esperados). A função deve usar</p>  | <p>função, dict.items(), for, isinstance(), try/except (KeyError, TypeError), print</p> |

|    |                                |   |  |
|----|--------------------------------|---|--|
|    |                                | um <b>for</b> loop e <b>try/except</b> para verificar se todas as chaves do modelo existem no dicionário e se os valores têm os tipos de dados corretos (ex: <code>isinstance(value, expected_type)</code> ).   |  |
| 50 | Dicionário de Listas Aninhadas | Crie uma <b>função</b> que receba um dicionário onde o valor é uma lista de números. Para cada chave, calcule e armazene na mesma chave a média, o valor máximo e o valor mínimo, substituindo a lista original por um novo <b>dicionário</b> de estatísticas (ex: <code>{'media': 10, 'max': 15}</code> ). | função, for, dict, <code>sum()</code> , <code>len()</code> , <code>max()</code> , <code>min()</code> , print |