

Atividades- Estruturas de Repetição

1) Adesão de passos: quem bateu a meta?

Uma clínica sênior acompanha passos diários (contagens de pedômetro). Há **leituras ausentes** (0) e **leituras duplicadas**. Calcule quem bateu a meta diária (≥ 6000 passos) e a taxa de adesão semanal.

Dicas:

- Filtrar dados com `continue`, contar adesões com `for`, encerrar cedo se a semana tiver mais de 7 registros válidos usando `break`.
- Mantenha contadores: `validos`, `dias_meta`, `soma_passos`.
- Cuidado com divisão por zero se não houver válidos.

Dados

```
# cada sublista = semana de um participante (7 a 10 leituras)
semanas = [
    [6500, 0, 7200, 5900, 8000, 0, 10000, 7500, 7200],
    [3000, 2000, 0, 4000, 5000, 6100, 6300],
    [0, 0, 0, 7000, 7001, 0, 8000],
]
META = 6000
```

Tarefas.

1. Para cada semana:
 - Percorra as leituras com `for`.
 - Ignore (com `continue`) leituras **0**.
 - Considere **no máximo 7 valores válidos**; ao atingir 7, use `break`.
 - Conte quantos dias atingiram META e calcule a **taxa de adesão** (`dias/7`).
2. Imprima um relatório por semana:
 - Semana X: dias na meta = d, taxa = t% (`validos=k`)
3. Desafio: compute a **média de passos** considerando apenas válidos.

Restrições.

- Não use `numpy` ou `pandas`.
 - Use `continue` para pular zeros e `break` para limitar a 7 válidos.
-

2) Coletor de pressão em campo

Em uma ação de extensão, vocês coletam pressões **sistólica/diastólica** de idosos. É preciso registrar **várias medidas** até o usuário encerrar, **pular entradas inválidas** e **parar imediatamente** em caso de emergência.

Dicas

- Usar `while + input()` para ler interativamente;
- `continue` para entradas inválidas;
- `break` para emergência (sistólica ≥ 180 **ou** diastólica ≥ 120).
- Normalize entradas com `.strip().lower()`.

Tarefas.

1. Em um `while True`:
 - Leia `sistolica` e `diastolica` (inteiros).
 - Se valores fora de faixa (ex.: < 60 ou > 260), **avise** e `continue`.
 - Classifique: **Alta** (≥ 140 ou ≥ 90) ou **Ok**.
 - Conte: total, altas, ok.
 - Pergunte: "Deseja registrar outra medida? (s/n)".
 - Se n, **saia** com `break`.
 - **Emergência**: se sistólica ≥ 180 **ou** diastólica ≥ 120 , imprima alerta e **interrompa** com `break`.
2. Ao final, apresente um resumo com contagens e porcentagens.

Restrições.

- Somente `while`, `break`, `continue`, `input`, `print`.

3) Validação de idades com `for...else`

Uma planilha com **idades de participantes** chegou com erros de digitação (negativos ou acima de 120). Você deve validar a lista e **parar no primeiro erro** encontrado.

Dica

- Usar `for` para percorrer, `break` ao detectar **primeiro inválido** e `else` para confirmar que **todos** eram válidos quando não houver `break`.
- Para mediana, copie e ordene com `sorted(lista)` antes.

Tarefas.

1. Para cada lista:
 - Percorra com `for`.

- Se idade < 0 or idade > 120: imprima Valor inválido encontrado: X e break.
 - Use for...else para imprimir Todos os dados são válidos! quando não houver break.
2. Extensão: calcule média e mediana **apenas** se todos os dados forem válidos (sem numpy).

Restrições.

- Não usar bibliotecas externas.

4) Simulador de atendimento: quando o estoque de vacinas acaba?

Um posto possui **estoque inicial** de vacinas. Idosos chegam em fila; **nem todos podem ser vacinados** (motivos: febre, alergia recente, etc.). Simule o processo até:

- **Acabar o estoque** (encerra com break), ou
- **Atender N pessoas** (limite operacional), o que ocorrer primeiro.

Dica

- Usar for (N tentativas) e continue para **pular** quem **não pode** ser vacinado; usar break quando **estoque==0**.
- Mantenha contadores atendidos, inaptos, tentativas, nao_atendidos_por_falta.

Parâmetros

```
import random
random.seed(42)
ESTOQUE = 25
N_TENTATIVAS = 60
PROB_INAPTO = 0.25 # chance de não vacinar
```

Tarefas.

1. Em um for de range(N_TENTATIVAS):
 - Sorteie inapto = random.random() < PROB_INAPTO.
 - Se inapto, registre e continue.
 - Caso contrário, **aplique vacina** (estoque -= 1).
 - Se estoque == 0, imprima Estoque esgotado no atendimento X e break.
2. Ao final, reporte:

- Atendidos, inaptos, não atendidos por falta de estoque, estoque remanescente.

Restrições.

- Não use pandas. Apenas random, laços e condicionais.
-

5) Histograma manual de minutos de caminhada

Você recebeu uma lista de **minutos de caminhada/dia**. Quer visualizar a distribuição em **faixas**: $[0, 10)$, $[10, 20)$, ..., $[90, 100]$. Há valores inválidos (negativos, > 300) a serem **ignorados**.

Dicas

- Construir um **histograma sem** bibliotecas (apenas laços), usando continue para pular inválidos e for para percorrer as amostras.
- Cuidado com o **intervalo final** $[90, 100]$ (fechado à direita).
- Para imprimir barras, use '#' * contagem.

Tarefas.

1. Crie uma estrutura para contagem por faixa (ex.: lista contagens de tamanho 10 ou dicionário).
2. Para cada m em minutos:
 - Se $m < 0$ ou $m > 300$, **ignore** com continue.
 - Descubra o **índice do bin**:
 - Para $m == 100$, considere o último bin.
 - Caso geral: $idx = m // 10$.
 - Incremente a contagem do bin.
3. Imprima o histograma no formato:

```
[0,10): #### (4)
[10,20): ## (2)
...
[90,100]: # (1)
```

Restrições.

- Sem numpy/pandas/matplotlib.
-