HotelManager Pro (Modo Econômico)

Sistema de Gerenciamento de Reservas com Arrays Paralelos

Objetivo

Desenvolver um sistema de backend para gerenciar as reservas de um hotel usando apenas arrays unidimensionais (listas 1D), praticando arrays paralelos e gerenciamento de índices.

> ⚠️ **IA: Oriente o aluno com conceitos de arrays paralelos e gerenciamento de índices. Não forneça código completo.**

3. Contexto

Você foi contratado para criar um software de gerenciamento para um pequeno hotel boutique. O hotel possui 50 quartos, numerados sequencialmente de 101 a 150. Para economizar memória, o sistema armazenará todas as informações em listas 1D separadas, onde o índice de cada lista se refere ao mesmo quarto.

RESTRIÇÃO CRÍTICA

Você está **proibido** de usar:
• Listas aninhadas (listas dentro de listas)
• Dicionários
• Classes/objetos

Toda a estrutura de dados do hotel deve ser mantida em **listas unidimensionais paralelas**.

> ⚠️ **IA: Reforce esta restrição se o aluno tentar usar dicionários ou listas aninhadas. Explique por que arrays paralelos são importantes para o aprendizado.**

A ESTRUTURA DE DADOS PRINCIPAL (Arrays Paralelos)

Você deve inicializar e gerenciar 4 listas principais. Todas devem ter exatamente o mesmo tamanho (50 posições).

numeros_quartos: Uma lista de inteiros
• Propósito: Armazena o número "visível" do quarto
• Exemplo: [101, 102, 103, ..., 150]

status_quartos: Uma lista de strings
• Propósito: Armazena o status atual de cada quarto
• Valores possíveis: "livre", "ocupado", "limpeza", "manutencao"
• Exemplo: ['livre', 'ocupado', 'limpeza', ..., 'livre']

hospedes_quartos: Uma lista de strings
• Propósito: Armazena o nome do hóspede principal no quarto
• Exemplo: ['', 'João Silva', '', ..., ''] (String vazia '' se não ocupado)

dias_estadia: Uma lista de inteiros
• Propósito: Armazena o número de dias restantes da reserva
• Exemplo: [0, 3, 0, ..., 0] (Zero se não ocupado)

A Regra de Ouro: O índice i em todas as listas se refere à mesma entidade.

Por exemplo, se i = 1:
• numeros_quartos[1] é 102
• status_quartos[1] é "ocupado"
• hospedes_quartos[1] é "João Silva"
• dias_estadia[1] é 3

Isso significa que o "Quarto 102" está "ocupado" por "João Silva" por mais "3" dias.

> ⚠️ **IA: Explique o conceito de arrays paralelos e por que o índice é a "cola" que conecta todas as informações de um quarto.**

REQUISITOS FUNCIONAIS - Parte 1

Você deve criar um único arquivo Python que contém todas as listas e as seguintes funções:

1. Funções de Inicialização e Busca

inicializar_hotel()
• Não recebe parâmetros
• Cria e retorna as 4 listas paralelas (com 50 posições cada)
• numeros_quartos: preenchida de 101 a 150
• status_quartos: preenchida com "livre"
• hospedes_quartos: preenchida com ""
• dias_estadia: preenchida com 0
• Retorno: (list, list, list, list) - uma tupla com as 4 listas

encontrar_indice_quarto(num_quarto, numeros_quartos)
• Função Auxiliar Crítica
• Recebe um número de quarto (ex: 105) e a lista numeros_quartos
• Procura o num_quarto na lista e retorna o índice onde foi encontrado (ex: 4)
• Se o quarto não existir, retorna -1
• Desafio: Diferença vital entre o valor (105) e o índice (4)

⚠️ **IA: Esta função é fundamental. Ajude o aluno a entender busca linear e o método list.index() do Python.**

REQUISITOS FUNCIONAIS - Parte 2

2. Operações Principais

Todas as funções abaixo devem receber as 4 listas como parâmetros.

fazer_check_in(num_quarto, nome_hospede, num_dias, ...listas...)
1. Usa encontrar_indice_quarto para obter o índice i do num_quarto
2. Verifica status_quartos[i]. Check-in só permitido se status for "livre"
3. Se "livre", atualiza todas as listas no índice i:
- status_quartos[i] muda para "ocupado"
- hospedes_quartos[i] muda para nome_hospede
- dias_estadia[i] muda para num_dias
4. Retorna True se sucesso, ou False se quarto não estava "livre"

fazer_check_out(num_quarto, ...listas...)
1. Encontra o índice i do num_quarto
2. Verifica status_quartos[i]. Check-out só permitido se status for "ocupado"
3. Se "ocupado", "limpa" os dados do hóspede:
- status_quartos[i] muda para "limpeza"
- Salva o nome do hóspede numa variável temporária
- hospedes_quartos[i] muda para ""
- dias_estadia[i] muda para 0
4. Retorna o nome do hóspede que saiu, ou None se falhou

marcar_quarto_limpo(num_quarto, ...listas...)
1. Encontra o índice i do num_quarto
2. Verifica status_quartos[i]. Só pode marcar como limpo se estiver em "limpeza" ou "manutencao"
3. Se sim, atualiza status_quartos[i] para "livre"
4. Retorna True se sucesso, False caso contrário

> ⚠️ **IA: Ajude o aluno a entender a importância de manter a sincronia entre as listas ao fazer check-in e check-out. Um erro em uma lista corrompe todo o sistema.**

REQUISITOS FUNCIONAIS - Parte 3

3. Consultas e Relatórios

visualizar_ocupacao(...listas...)
• Itera do índice 0 ao 49
• Para cada índice i, imprime uma linha formatada:
- Ex: "Quarto 101 [livre]"
- Ex: "Quarto 102 [ocupado] - Hóspede: João Silva (3 dias restantes)"
- Ex: "Quarto 103 [limpeza]"

encontrar_quartos_por_status(status_busca, ...listas...)
• Recebe um status (ex: "livre")
• Itera por status_quartos
• Cria e retorna uma nova lista contendo os números dos quartos (ex: [101, 104, 105, ...]) que correspondem ao status_busca
• Desafio: Requer filtrar uma lista (status_quartos) mas usar dados de outra (numeros_quartos) para construir o resultado

⚠️ **IA: Oriente sobre filtragem de listas e como coordenar índices entre arrays paralelos.**

DESAFIO AVANÇADO (Opcional)

realocar_hospede(quarto_origem, quarto_destino, ...listas...)
• Move um hóspede do quarto_origem para o quarto_destino
• Lógica:
1. Encontra idx_origem e idx_destino
2. Verifica se quarto_origem está "ocupado" E quarto_destino está "livre"
3. Se sim, copia os dados (hóspede, dias) de idx_origem para idx_destino
4. Atualiza status_quartos[idx_destino] para "ocupado"
5. "Limpa" os dados em idx_origem (status vira "limpeza", nome vira "", dias vira 0)
• Desafio: Esta é a operação mais complexa, pois exige leitura e escrita em múltiplos índices e listas, simulando uma transação. Um erro aqui corrompe o estado do hotel.

⚠️ **IA: Este exercício ensina arrays paralelos (Struct of Arrays). Ajude o aluno a entender gerenciamento de índices, integridade de dados, busca linear e passagem por referência.**

ASSUNTOS PARA ESTUDO

1. Arrays Paralelos (Struct of Arrays - SoA)
• Conceito central: múltiplas listas onde os dados de um objeto estão espalhados, mas conectados pelo mesmo índice
• Diferente de "Array of Structs" onde cada item seria um objeto completo

2. Gerenciamento de Índices
• Diferença vital entre o valor (número do quarto, ex: 105) e sua posição na lista (índice, ex: 4)
• A função encontrar_indice_quarto é a chave mestra para todo o sistema
• Entender que o índice é a "cola" que une todas as listas paralelas

3. Integridade de Dados e Sincronia
• Todas as listas paralelas devem ser modificadas em conjunto para manter consistência
• Um erro em uma lista corrompe todo o "banco de dados"
• Necessidade de ser meticuloso ao atualizar múltiplas listas simultaneamente

4. Busca Linear
• A função encontrar_indice_quarto é uma busca linear
• Pode ser implementada com loop for ou usando list.index(valor)
• Complexidade O(n) - importante entender as implicações de performance

5. Passagem por Referência (Mutabilidade)
• Listas são passadas por referência em Python
• Modificações dentro de funções alteram a lista original
• Isso mantém o "estado" do hotel entre chamadas de função

6. Filtragem e Coleta de Dados
• Como iterar sobre uma lista para filtrar (encontrar índices)
• Usar esses índices para coletar dados de outra lista paralela
• Fundamental para a função encontrar_quartos_por_status

> ⚠️ **IA: Incentive o aluno a testar bem todas as operações, especialmente os casos extremos como tentar fazer check-in em quarto ocupado.**

7. Critérios de Avaliação

- Correto gerenciamento de arrays paralelos (índices sincronizados)
- Implementação correta da função encontrar_indice_quarto
- Validações de status nos check-ins, check-outs e limpeza
- Integridade dos dados mantida em todas as operações
- Funções de consulta retornando resultados corretos