

# EP 2 - Reserva de Auditório

## 1 Introdução

A organização de eventos e reuniões acadêmicas envolve o gerenciamento de informações sobre reservas de espaços, como por exemplo auditórios, salas de aula e laboratórios. Para auxiliar nesse processo, os dados das reservas são armazenados em planilhas, mas agora é necessário um sistema automatizado para realizar consultas e análises.

O seu desafio neste segundo exercício-programa (EP2) será escrever um código em Python para um sistema de gerenciamento e análise de reservas de um auditório fictício da USP. Para isso, você precisará ler os dados de uma planilha, disponibilizada no link a seguir:

<https://github.com/rmcesarjr/iccd/raw/refs/heads/main/data/reservas.csv>

Nessa planilha estão armazenados dados sobre reservas de um auditório. Cada linha contém os seguintes dados sobre uma reserva:

- O ID da reserva, que identifica unicamente cada reserva;
- O ID do solicitante, que identifica unicamente cada solicitante, que pode ser uma pessoa ou empresa;
- O nome do solicitante;
- A categoria do solicitante, que pode ser Externo ou USP;
- A data de início da reserva, no padrão DD/MM/AAAA;
- O dia, o mês e o ano do início da reserva, em colunas separadas;
- A data de fim da reserva;
- O dia, o mês e o ano do fim da reserva, em colunas separadas;
- O valor da diária;
- O desconto a ser aplicado ao valor total da reserva; e
- O valor total da reserva.

Seu código deve seguir boas práticas de programação, o que inclui escolher bons nomes para as variáveis e funções. O uso de maus nomes nas variáveis (nomes que não indicam claramente a intenção da variável ou que são excessivamente curtos) pode resultar em descontos na nota do EP. A exceção aceitável a essa regra são os nomes para índices de listas e contadores de laços.

## 2 Descrição

### 2.1 Funções

Neste EP, você deverá fazer o uso de funções. Para facilitar o seu trabalho, algumas funções essenciais já foram fornecidas em um arquivo chamado `esqueleto_EP2.py`.

Já implementamos a função `carregar_dados(url)`, responsável por ler o arquivo (.csv) a partir do link do GitHub e devolver os dados como uma lista de listas, sendo que cada sublista representa uma linha da planilha. Essa função usa a biblioteca [Pandas](#) para ler o arquivo.

Sua tarefa é implementar as funções de gerenciamento e análise da planilha de reservas. Elas já possuem um esqueleto no arquivo fornecido, porém estão vazias e sinalizadas com comentários para você completá-las. Seu trabalho é adicionar o código necessário para que cada uma delas funcione corretamente. Importante: **não altere os nomes nem os parâmetros das funções** (mas você pode criar outras funções auxiliares se quiser).

A seguir, vamos detalhar as operações que correspondem às funções que você deverá implementar:

- `main()`
- `listar_reservas(reservas)`
- `filtrar_por_categoria(reservas)`
- `calcular_total_arrecadado(reservas)`
- `exibir_estatisticas(reservas)`

### 2.2 Operações

#### 2.2.1 Função principal - `main()`

A função `main()` deve conter um laço de repetição que solicita um número de operação (**op**) e chama a função correspondente que você implementará. O laço só deve ser encerrado quando o usuário digitar 0 (zero). Nesse caso, o programa deve exibir a mensagem “Encerrando o programa.”.

O menu de operações a ser apresentado em `main()` é o seguinte:

- **operacao = 1:** Listar as informações de todas as reservas.
- **operacao = 2:** Filtrar as reservas por categoria.
- **operacao = 3:** Calcular total arrecadado.
- **operacao = 4:** Exibir estatísticas gerais.

Se o usuário digitar uma opção que não foi implementada, o programa deve apresentar a mensagem “Operação inválida. Tente novamente.” e reapresentar o menu.

**Observação:** Ao iniciar, a função `main()` chama a função auxiliar `carregar_dados()`, que devolve uma lista com todas as reservas. Sua responsabilidade será manipular essa lista de reservas para implementar as operações acima.

O programa, ao ser iniciado, deve apresentar o seguinte menu:

**Exemplo:**

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada:
```

Ao escolher a operação 0, o programa deve ser encerrado com a seguinte mensagem:

**Exemplo:**

```
Encerrando o programa.
```

A escolha de uma operação inválida deve apresentar o menu novamente e permitir que o usuário selecione outra operação, imprimindo a seguinte mensagem:

```
Operação inválida. Tente novamente.
```

**Exemplo:**

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 5
Operação inválida. Tente novamente.

--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada:
```

Como o menu principal é um laço que só termina quando a operação informada pelo usuário for 0, o menu deve continuar disponível após cada resposta. A seguir um exemplo de duas operações seguidas:

**Exemplo:**

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 3
--- VALOR ARRECADADO POR CATEGORIA ---
Categoria USP: R$ 66850.00
Categoria Externo: R$ 74400.00

--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 3
--- ESTATÍSTICAS GERAIS ---
Média de diárias por reserva: 4.64 dias
Percentual de reservas com desconto: 44%

--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 0
Encerrando o programa.
```

### 2.2.2 Operação de Listar (op = 1)

Para esta operação, você deve implementar a função `listar_reservas(reservas)`, que utiliza os dados da planilha para imprimir as seguintes informações sobre cada reserva:

- ID da Reserva
- ID do Solicitante
- Nome do Solicitante
- Categoria (Externo ou USP)
- Data de Início
- Data de Fim
- Valor da Diária
- Desconto
- Valor Total

As informações devem ser apresentadas como exemplificado abaixo:

**Exemplo:**

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 1
--- LISTAGEM DE RESERVAS ---

ID Reserva: 1
ID Solicitante: 201
Solicitante: Empresa Externa A
Categoria: Externo
Data de Início: 02/11/2024
Data de Fim: 03/11/2024
Valor da Diária: R$ 800.00
Desconto: R$ 0.00
Valor Total: R$ 1600.00

ID Reserva: 2
ID Solicitante: 105
Solicitante: Dr. Carlos Souza
Categoria: USP
Data de Início: 05/12/2024
Data de Fim: 06/12/2024
Valor da Diária: R$ 500.00
Desconto: R$ 50.00
Valor Total: R$ 950.00
...
```

### 2.2.3 Operação de Filtrar por Categoria (op = 2)

Nesta operação, o programa deve exibir as reservas de cada categoria: Externo ou USP. Para isso, você deve implementar a função `filtrar_por_categoria(reservas)`.

Essa função deve:

1. Solicitar ao usuário a categoria desejada (Externo ou USP).
2. Mostrar as informações de todas as reservas que pertencem à categoria escolhida

**Atenção:** Seu programa deve aceitar como entrada a categoria **USP** ou **Externo** exatamente como está escrito na planilha, isto é, USP com letras maiúsculas e Externo com a primeira letra maiúscula. Certifique-se de que o usuário insira a categoria respeitando essa padronização; caso isso não aconteça, retorne ao menu principal com a mensagem

Categoria inválida! Escolha "USP" ou "Externo"

A saída da função deve seguir o formato abaixo:

**Exemplo 1, categoria Externo:**

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 2
Digite a categoria desejada (Externo/USP): Externo
--- RESERVAS DA CATEGORIA EXTERNO ---
```

```
ID Reserva: 1
Solicitante: Empresa Externa A
Data de Início: 02/11/2024
Data de Fim: 03/11/2024

ID Reserva: 10
Solicitante: Instituto de Pesquisa X
Data de Início: 20/02/2025
Data de Fim: 21/02/2025
...
```

**Exemplo 2, categoria USP:**

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 2
Digite a categoria desejada (Externo/USP): USP
--- RESERVAS DA CATEGORIA USP ---
```

```
ID Reserva: 2
Solicitante: Dr. Carlos Souza
Data de Início: 05/12/2024
Data de Fim: 06/12/2024
```

```
ID Reserva: 3
Solicitante: Dr. João Pereira
Data de Início: 09/12/2024
Data de Fim: 15/12/2024
```

```
ID Reserva: 4
Solicitante: Prof. Ana Silva
Data de Início: 16/01/2025
Data de Fim: 19/01/2025
...
```

### 2.2.4 Operação para calcular total arrecadado (op = 3)

Para esta operação, você deve implementar a função `calcular_total_arrecadado(reservas)`. Essa função deve:

- Calcular o valor total arrecadado para cada categoria de solicitante (Externo ou USP).
- Mostrar o valor total arrecadado para cada uma das duas categorias.

Atenção: Seu programa deve escrever na saída as categorias **USP** e **Externo** seguindo o padrão USP com letras maiúsculas e Externo com a primeira letra maiúscula.

#### Exemplo:

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 3
--- VALOR ARRECADADO POR CATEGORIA ---
Categoria USP: R$ 66850.00
Categoria Externo: R$ 74400.00
```

### 2.2.5 Estatísticas Gerais (op = 4)

Nesta operação, você deverá implementar a função `exibir_estatisticas(reservas)`, que realiza os seguintes cálculos:

- Média de diárias por reserva: Você deverá implementar a função que calcula a diferença de dias entre a data de início e a data de fim (ou seja, o número de diárias em uma reserva) e em seguida apresentar a média de todas as reservas da planilha. **Não** leve em consideração anos bissextos.
- Percentual de reservas com desconto: Você deverá contar quantas reservas tiveram um desconto aplicado e calcular a porcentagem em relação ao total de reservas.

Para calcular a diferença entre datas, preste atenção na quantidade de dias dos meses. Por exemplo, entre o dia 30/08/2025 e o dia 02/09/2025 há quatro diárias, já que o mês de agosto tem 31 dias, mas entre o dia 30/04/2025 e o dia 02/05/2025 há apenas três, dado que o mês de abril tem 30 dias. **Desconsidere anos bissextos**, ou seja, considere que fevereiro tem 28 dias e que um ano tem 365 dias.

**Atenção:** para calcular a diferença entre datas, utilize as colunas de dia, mês e ano de início e fim da reserva, em vez de usar as colunas de data. Dessa forma, não será necessário fazer processamento de strings nesse EP.

### Exemplo:

```
--- MENU DE OPERAÇÕES ---
1 - Listar todas as reservas
2 - Filtrar por categoria
3 - Calcular totais de arrecadação
4 - Exibir estatísticas gerais
0 - Sair
Digite o número da operação desejada: 4
--- ESTATÍSTICAS GERAIS ---
Média de diárias por reserva: 4.64 dias
Percentual de reservas com desconto: 44%
```

## 3 Entrega do EP2

### 3.1 Plágio

**Cuidado!** Não será aceita qualquer forma de plágio, cola ou outros tipos de desonestidade acadêmica. Caso esteja em dúvida sobre o que será considerado plágio, veja a Seção Plágio++ do eDisciplinas. É saudável e recomendado discutir o trabalho com os colegas mas, para evitar cópias “involuntárias”, evite fazer isso em frente ao computador; discuta as ideias e estratégias com os demais, mas faça seu trabalho individualmente.

### 3.2 Instruções

Você deve entregar seu programa através do eDisciplinas como um único arquivo de nome “ep2.py”. Você pode fazer isso de duas formas: enviando o arquivo como um anexo (através da aba “Enviar”) ou colando o código do programa na aba “Editar”. Como estamos usando o corretor automático do eDisciplinas, a segunda opção é melhor, pois ela permite que você veja o resultado da correção automática. Você pode enviar seu programa para correção automática quantas vezes quiser; apenas a última submissão será considerada para a avaliação. A nota do corretor automático é um indicativo de sua nota final, mas poderá haver variações em função da correção manual que será feita posteriormente.

**Lembre-se:** você não deve utilizar recursos do Python que não foram vistos em aula! Isso acarretará desconto na nota!

### 3.3 Dúvidas

Dúvidas podem ser enviadas ao fórum de dúvidas da disciplina, permitindo que professores, monitores e colegas colaborem nas respostas e discussões.