

Tema 1: Hospital

Uma aplicação para um Hospital precisa armazenar informações sobre os médicos, os pacientes e sobre cada atendimento que um médico realiza com um paciente (ou seja, a consulta). Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Médico = (CRM, Nome, Data de Nascimento, Sexo, Especialidade, Universidade em que se formou, E-mails, Telefones)

Paciente = (CPF, Nome, Data de Nascimento, Sexo, Plano de Saúde, E-mails, Telefones)

Consulta = (CRM, CPF, Data, Hora, Diagnostico, Medicamentos)

Atenção: os atributos grifados são **chaves** e você **NÃO** deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Médicos
2. Submenu de Pacientes
3. Submenu de Consultas
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que os atributos que estão no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Telefones indica que uma pessoa pode ter vários telefones (a quantidade é indefinida). Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos os telefones que a entidade possuir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir.

Relatórios:

- a) Mostrar todos os dados de todos os médicos a partir de uma especialidade X fornecida pelo usuário;
- b) Mostrar todos os dados de todos os pacientes menores de X anos de idade, onde X é fornecido pelo usuário;
- c) Mostrar o CRM, o nome do médico, o CPF do paciente, o nome do paciente, data, hora, diagnostico e medicamentos para todas as consultas realizadas nos últimos X dias, onde o valor de X é fornecido pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de médico, paciente e de cada consulta que um médico realiza com um paciente deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 2: Cinema

Uma aplicação para um Cinema precisa armazenar informações sobre os filmes, as salas de exibição de filmes e sobre cada sessão (ou seja, a exibição de um filme em determinada sala). Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Sala = (Código, Nome, capacidade, Tipo de exibição, Acessível)

Filme = (Código, Nome, Ano de lançamento, Diretor, Atores)

Sessão = (Código do Filme, Código da Sala, Data, Horário, Preço do Ingresso)

Atenção: os atributos grifados são **chaves** e você **NÃO** deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Salas
2. Submenu de Filmes
3. Submenu de Sessões
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que atributos no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Atores indica que um filme pode ter um número indefinido de atores. Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos eles. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir.

Relatórios:

- a) Mostrar todos os dados de todas as salas cujo tipo de exibição seja X e capacidade para mais de Y pessoas, onde X e Y são fornecidos pelo usuário;
- b) Mostrar todos os dados de todos os filmes que foram lançados a partir do ano X, onde X é fornecido pelo usuário;
- c) Mostrar o código do filme, nome do filme, atores, código da sala, nome da sala e os demais atributos de todas as sessões exibidas a partir de uma data inicial X até uma data final Y, onde ambas as datas são fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de sala, filme e sessão deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 3: Biblioteca:

Uma aplicação para uma Biblioteca precisa armazenar informações sobre os seus usuários (alunos e professores), livros e sobre o empréstimo de livros. Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Usuário = (CPF, Nome, Rua, Nro, CEP, E-mails, Telefones, Data de Nascimento, Profissão)

Livro = (ISBN, Título, Gênero, Autores, Número de Páginas)

Empréstimo = (CPF Pessoa, ISBN Livro, Data de Retirada, Data de Devolução, Valor Diário da Multa por atraso)

Atenção: os atributos grifados são **chaves** e você **NÃO** deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Usuários
2. Submenu de Livros
3. Submenu de Empréstimos
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que os atributos que estão no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Telefones indica que uma pessoa pode ter vários telefones (a quantidade é indefinida). Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos os telefones que a entidade possuir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir.

Relatórios:

- a) Mostrar todos os dados de todos os usuários com mais de X anos de idade, onde X é fornecido pelo usuário;
- b) Mostrar todos os dados de todos os livros que tenham mais do que X autores, sendo X fornecido pelo usuário;
- c) Mostrar o CPF da pessoa, o nome da pessoa, o ISBN do livro, o título do livro e todos os demais atributos dos empréstimos que possuem data de devolução entre as datas X e Y (inclusive), ambas fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de usuário (aluno ou professor), livro e empréstimo deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 4: Universidade

Uma aplicação para uma Universidade precisa armazenar informações sobre os seus professores, disciplinas e sobre as disciplinas ministradas por cada professor. Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Professor = (Registro Funcional, Nome, Data de Nascimento, Sexo, Área de Pesquisa, titulação, graduação, E-mails, Telefones)

Disciplina = (Sigla, Nome, Ementa, Bibliografia, Numero de Créditos, Carga Horária)

Prof_Disc = (Registro Funcional Professor, Sigla Disciplina, Ano, Semestre, Dias da semana, Horários de início, curso)

Atenção: os atributos grifados são **chaves** e você NÃO deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Professores
2. Submenu de Disciplinas
3. Submenu de Professores-Disciplinas
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que os atributos que estão no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Telefones indica que uma pessoa pode ter vários telefones (a quantidade é indefinida). Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos os telefones que a entidade possuir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados.

Relatórios:

- a) Mostrar todos os dados de todos os professores que têm determinada titulação fornecida pelo usuário (mestrado ou doutorado);
- b) Mostrar todos os dados de todas as disciplinas que possuem mais do que X créditos;
- c) Mostrar o Registro Funcional do Professor, o nome do professor, a Sigla da disciplina, o nome da disciplina e todos os demais atributos de Prof-Disc para aquelas disciplinas que serão ministradas às terças e às quintas feiras de cada semana.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de professor, disciplina e disciplina ministrada por um professor deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 5: Imobiliária

Uma aplicação para uma Imobiliária precisa armazenar informações sobre os seus clientes, imóveis e sobre os aluguéis (ou seja, quando um cliente aluga um imóvel). Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Cliente = (CPE, Nome, Data de Nascimento, Sexo, Salário, E-mails, Telefones)

Imóvel = (Código, Descrição, Endereço, Cidade, Estado, CEP, Tipo (comercial, residencial), Valor aluguel)

Aluguel = (CPF Cliente, Código Imóvel, Data de Entrada, CPF Fiador, Nome Fiador, Data de Saída, Valor Mensal)

Atenção: os atributos grifados são **chaves** e você **NÃO** deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Clientes
2. Submenu de Imóveis
3. Submenu de Aluguéis
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que os atributos que estão no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Telefones indica que uma pessoa pode ter vários telefones (a quantidade é indefinida). Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos os telefones que a entidade possuir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir.

O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados.

Relatórios:

- a) Mostrar todos os dados de todos os clientes que possuem mais do que X telefones, onde X deve ser fornecido pelo usuário;
- b) Mostrar todos os dados de todos os imóveis que possuem de um tipo específico fornecido pelo usuário;
- c) Mostrar o CPF do cliente, nome do cliente, código do imóvel, descrição e os demais dados dos imóveis com data de entrada entre X e Y, onde ambas as datas devem ser fornecidas pelo usuário.
- d) de cada semana.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de cliente, imóvel e aluguel deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 6: Oficina mecânica

Uma aplicação para uma oficina mecânica precisa armazenar informações sobre os seus mecânicos, veículos e sobre cada conserto de veículo. Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Mecânico = (CPF, Nome, Data de Nascimento, Sexo, Salário, E-mails, Telefones)

Veículo = (Placa, Tipo, Marca, Modelo, Ano, Portas, Combustível, Cor)

Conserto = (CPF, Placa, Data de Entrada, Data de Saída, Descrição dos Problemas, Valor Conserto)

Atenção: os atributos grifados são **chaves** e você **NÃO** deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Mecânicos
2. Submenu de Veículos
3. Submenu de Consertos
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que os atributos que estão no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Telefones indica que uma pessoa pode ter vários telefones (a quantidade é indefinida). Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos os telefones que a entidade possuir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir.

O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados.

Relatórios:

- a) Mostrar todos os dados de todos os mecânicos que possuem mais do que X anos de idade, onde X deve ser fornecido pelo usuário;
- b) Mostrar todos os dados de todos os veículos de determinada marca fornecida pelo usuário;
- c) Mostrar o CPF do mecânico, nome do mecânico, a placa completa, a marca, o modelo e o ano do veículo e os demais dados dos consertos com data de entrada entre X e Y, onde ambas as datas devem ser fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de mecânico, veículo e conserto deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 7: Companhia Aérea

Uma aplicação para uma companhia aérea precisa armazenar informações sobre os seus pilotos, voos e sobre cada viagem realizada por um piloto. Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Piloto = (Registro do Piloto, Nome, Data de Nascimento, Sexo, Curso de Especialização, E-mails, Telefones)

Voo = (Número, Cidade Origem, Cidade Destino, Distância, Tempo Médio, Aeronave, Escalas)

Viagem = (Número do Voo, Piloto, Data da Saída, Hora Saída, Data da Chegada, Hora Chegada, Ocorrências)

Atenção: os atributos grifados são **chaves** e você NÃO deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu de Opções:

1. Submenu de Pilotos
2. Submenu de Voos
3. Submenu de Viagens
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um elemento específico do conjunto, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que os atributos que estão no **plural** indicam que deverá ser possível incluir vários itens daquele mesmo atributo. Por exemplo, o atributo Telefones indica que uma pessoa pode ter vários telefones (a quantidade é indefinida). Portanto, deve-se utilizar uma estrutura que seja adequada para armazenar todos os telefones que a entidade possuir. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados a seguir.

Relatórios:

- a) Mostrar todos os dados de todos os pilotos que possuem mais do que X anos de idade, onde X deve ser fornecido pelo usuário;
- b) Mostrar todos os dados de todos os voos que possuem escalas nas cidades X, onde X devem ser fornecidos pelo usuário;
- c) Mostrar o registro do piloto, nome do piloto, número do voo, cidade origem, cidade destino e os demais dados da viagem com data de saída entre X e Y, onde ambas as datas devem ser fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de piloto, voo e viagem deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 8: Controle de compras e vendas

Uma aplicação para uma loja de cosméticos precisa armazenar informações sobre os seus clientes, bem como sobre a compra/venda de produtos. Os dados a serem armazenados sobre cada cliente, produto e sobre a compra/venda de produtos por cliente são apresentados a seguir:

Cliente = (CPF, Nome, Data de Nascimento, Sexo, Salário, E-mails, Telefones)

Produto = (Código, Descrição, Peso, Preço, Desconto, Data de Validade)

Compra/Venda = (CPF Cliente, Código Produto, Data, Hora, Valor)

Atenção: os atributos grifados são **chaves** e você **NÃO** deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu Principal:

1. Submenu de Clientes
2. Submenu de Produtos
3. Submenu de Compra/Venda
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. Observe que as informações que estão no **plural** significam que deve ser possível incluir vários itens do mesmo atributo, por exemplo, os atributos E-mails e Telefones, devem permitir a inclusão de todos os e-mails e telefones daquele cliente. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados abaixo.

Relatórios:

- a) Mostrar todos os dados de todos os clientes que possuem mais do que X telefones, onde X deve ser fornecido pelo usuário;
- b) Mostrar todos os dados de todos os produtos que já tiveram sua data de validade vencida, considerando a data do sistema no momento da execução;
- c) Mostrar o CPF e nome do cliente, código do produto, descrição e os demais dados das vendas que foram realizadas entre as datas X e Y, onde ambas as datas devem ser fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de cliente, produto e compra/venda deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 9: Hotel

Uma aplicação para um hotel precisa gerenciar informações sobre as reservas de apartamentos realizadas por seus clientes, bem como armazenar os dados de cada cliente. Os dados a serem armazenados sobre os apartamentos, clientes e reservas são apresentados a seguir:

Clientes = (CPF, Nome, Endereço, Telefone fixo, Telefone celular, Data de nascimento)

Reservas = (Código, CPF do cliente)

Apartamentos = (Código, descrição, número de adultos, número de crianças, valor)

Reserva Apartamento = (Código da reserva, Código do apartamento, data de entrada, data de saída)

Atenção: os atributos grifados são **chaves** e você NÃO deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu Principal:

1. Submenu de Clientes
2. Submenu de Reservas
3. Submenu de Apartamentos
4. Submenu de Reserva Apartamentos
5. Submenu Relatórios
6. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados abaixo.

Relatórios:

- a) Mostrar todas as reservas de determinado apartamento cujo código deverá ser fornecido pelo usuário;
- b) Mostrar todas as reservas de determinado cliente cujo CPF deverá ser fornecido pelo usuário;
- c) Mostrar o CPF e nome de cada cliente que fizeram reservas para o período entre as datas X e Y, onde ambas as datas devem ser fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de cliente, reserva, apartamento e reserva de apartamento deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 10: Locadora de Automóveis

Uma aplicação para um locadora de automóveis precisa gerenciar informações sobre os aluguéis de veículos realizados por seus clientes, bem como armazenar os dados de cada cliente e de cada carro disponível para locação. Os dados a serem armazenados sobre os clientes, carros e aluguéis são apresentados a seguir:

Cientes = (CPF, Nome, Endereço, Telefone fixo, Telefone celular, Data de nascimento)

Veículos = (Código, descrição, categoria, capacidade, combustível, ano, modelo)

Alugueis = (CPF do cliente, Código do veículo, data de entrada, data de saída)

Atenção: os atributos grifados são **chaves** e você NÃO deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu Principal:

1. Submenu de Clientes
2. Submenu de Veículos
3. Submenu de Alugueis
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados abaixo.

Relatórios:

- a) Mostrar todas as reservas de determinado cliente cujo CPF deverá ser fornecido pelo usuário;
- b) Mostrar todas as reservas de determinado veículo cujo código deverá ser fornecido pelo usuário;
- c) Mostrar os dados de cada veículo e o CPF e nome de cada cliente que fizeram reservas para o período entre as datas X e Y, onde ambas as datas devem ser fornecidas pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de cliente, veículo e aluguel deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 11: Gerenciamento de cursos

Uma escola de cursos *online* de rápida duração deseja uma sistema para gerenciar os dados de cada aluno e de cada curso realizado por um aluno. Os dados a serem armazenados sobre cada uma dessas entidades são apresentados a seguir:

Alunos = (CPF, Nome, Data de Nascimento, Sexo, E-mails, Telefones)

Cursos = (Código, Descrição, Carga horária, Preço)

Matrícula = (CPF Aluno, Código curso, Data de início, Data de término, Desconto)

Atenção: os atributos grifados são **chaves** e você NÃO deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu Principal:

1. Submenu de Alunos
2. Submenu de Cursos
3. Submenu de Matrículas
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados abaixo.

Relatórios:

- a) Mostrar o nome do curso, o CPF, o nome e os e-mails de todos os alunos que cursaram determinado curso cujo o código será fornecido pelo usuário;
- b) Mostrar os dados de todos os cursos oferecidos entre as datas X e Y, onde as datas devem ser fornecidas pelo usuário;
- c) Mostrar os dados de todos os cursos realizados por determinado aluno, cujo CPF será fornecido pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de aluno, curso e matrícula deverá ser armazenado em um **arquivo texto** específico, que conterà apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.

Tema 12: Contratação de serviço de limpeza

Um sistema para uma prestadora de serviço de limpeza (faxinas) precisa gerenciar os dados sobre cada faxineiro(a) e os clientes que contratam os seus serviços. Os dados a serem armazenados sobre cada entidade são apresentados a seguir:

Faxineiro = (CPF, RG, nome, sexo, data de nascimento, telefones)

Cliente = (CPF, nome, data de nascimento, endereço, CEP, cidade, e-mails, telefones)

Serviço = (CPF faxineiro, CPF cliente, data, valor)

Atenção: os atributos grifados são **chaves** e você NÃO deve permitir a inclusão de mais de um cadastro com os mesmos valores para os atributos chaves.

Utilizando os conhecimentos aprendidos sobre Dicionários, Listas e Funções, desenvolva um programa em Python que apresente o seguinte menu de opções para o usuário e implemente cada operação usando **função**. Escolha a estrutura de dados mais apropriada para armazenar os dados de cada entidade descrita anteriormente.

Menu Principal:

1. Submenu de Faxineiros
2. Submenu de Clientes
3. Submenu de Serviços
4. Submenu Relatórios
5. Sair

Cada Submenu deverá oferecer as opções: Listar todos, Listar um, Incluir (sem repetição), Alterar e Excluir (após confirmação dos dados) um elemento do conjunto. O Submenu Relatórios deverá ter uma opção para cada um dos relatórios solicitados abaixo.

Relatórios:

- a) Mostrar o nome, os e-mails e telefones de todos(as) os(as) clientes que contrataram o serviço de determinado(a) faxineiro(a) entre as datas X e Y. O CPF do(a) faxineiro(a) e as datas deverão ser fornecidos pelo usuário;
- b) Mostrar os dados de todos os serviços contratados para uma data específica a ser fornecida pelo usuário. Apresente também o nome do(a) faxineiro(a) e do(a) cliente; Mostrar os dados de todos os serviços realizados por determinado(a) faxineiro(a), cujo CPF será fornecido pelo usuário.

OBS: Utilize **Arquivos** para a persistência dos dados manipulados pela aplicação. Em outras palavras, cada registro de faxineiro, cliente e serviço deverá ser armazenado em um **arquivo texto** específico, que conterá apenas registros daquele mesmo tipo de entidade. Os relatórios também devem ser guardados em arquivos.