

Trabalho prático python I

Nome: Ana Laura dos Santos Franco-30/01/2025

Documento descritivo

Sobre a empresa

(empresa não existe de verdade)

Empresa: EcoClean

Descrição: Oferece soluções de limpeza sustentável para residências e escritórios, utilizando produtos biodegradáveis e técnicas de baixo impacto ambiental. O objetivo é promover ambientes mais limpos e saudáveis, enquanto contribuírem para a preservação do planeta.

Tipo de usuários:

- **Dono:** Tem acesso completo ao sistema, podendo gerenciar usuários, serviços e agendamentos.
- **Cliente:** Pode visualizar os serviços disponíveis, agendar serviços e ver seus próprios agendamentos.

Serviços:

- Limpeza com produtos biodegradáveis
- Consultoria para práticas de limpeza ambientalmente responsáveis
- Serviços de dedetização para eliminar pragas de forma ecológica

Implementação

- Estrutura de dados
 1. Usuários:
 - Cada usuário é representado por um dicionário com as seguintes chaves:
 - "nome" Nome do usuário.
 - "nivel": Nível de acesso ("dono" ou "cliente").
 - "senha": Senha do usuário.
- Já existem dois usuários criados para testar:
 - O dono:
Nome de usuário: Ecoclean
Senha: 1234
 - O cliente:
Nome de usuário: Juliana
Senha: 5678

Arquivo de Registro:

- O arquivo dados-usuario.json armazena uma lista de dicionários, onde cada dicionário representa um usuário.

Funcionalidades (CRUD):

- **Create (Criar):** criar_usuario - Adiciona um novo usuário ao sistema.
- **Read (Ler):** listar_usuarios - Lista todos os usuários cadastrados.
- **Update (Atualizar):** atualizar_usuario - Atualiza os dados de um usuário existente.
- **Delete (Deletar):** deletar_usuario - Remove um usuário do sistema.

2. Serviços

Estrutura de Dados:

- Cada serviço é representado por um dicionário com as seguintes chaves:
 - "nome": Nome do serviço.
 - "preço": Preço do serviço.

Arquivo de Registro:

- O arquivo dados-serviço.json armazena uma lista de dicionários, onde cada dicionário representa um serviço.

Funcionalidades (CRUD):

- **Create (Criar):** criar_serviços - Adiciona um novo serviço ao sistema.
- **Read (Ler):** listar_servicos - Lista todos os serviços cadastrados, com opção de ordenação por nome ou preço.
- **Update (Atualizar):** atualizar_servico - Atualiza os dados de um serviço existente.
- **Delete (Deletar):** deletar_servicos - Remove um serviço do sistema.

3. Agendamentos

Estrutura de Dados:

- Cada agendamento é representado por um dicionário com as seguintes chaves:
 - "cliente": Nome do cliente que agendou o serviço.
 - "serviço": Nome do serviço agendado.
 - "data": Data do agendamento.
 - "hora": Hora do agendamento.

Arquivo de Registro:

- O arquivo dados-agendamentos.json armazena uma lista de dicionários, onde cada dicionário representa um agendamento.

Funcionalidades (CRUD):

- **Create (Criar):** agendar_servico - Cria um novo agendamento.
- **Read (Ler):** listar_agendamentos - Lista os agendamentos, com visões diferentes para donos e clientes.
- **Update (Atualizar):** atualizar_agendamento - Atualiza os detalhes de um agendamento existente.
- **Delete (Deletar):** deletar_agendamento - Remove um agendamento do sistema.

Conclusão:

Dificuldades Encontradas:

- **Lógica de programação.**
- **Validar os dados:** Implementar validações robustas para garantir que os dados inseridos sejam corretos.
- **Persistência de Dados:** Garantir que os dados fossem corretamente carregados e salvos nos arquivos JSON.
- **Tratamento de Erros:** Identificar a causa desses erros e solucioná-los.

Escolhas Bem-Sucedidas:

- **Estrutura de Dados.**
- **Divisão do sistema em módulos (usuários, serviços, agendamentos).**

O que Faltou Fazer:

- **Validação de Entrada:** Adicionar mais validações para garantir que os dados inseridos sejam corretos e consistentes.
- **Interface Gráfica:** Implementar uma interface gráfica para melhorar a usabilidade e a experiência do usuário.
- **Relatórios:** Adicionar funcionalidades para gerar relatórios, como listar todos os agendamentos de um determinado dia ou calcular o faturamento mensal.
- **Implementar validações robustas:** Garantir que os dados inseridos sejam corretos (por exemplo, formatos de data e hora)

O que Faria Diferente:

- **Segurança:** Implementar criptografia para senhas e outras informações sensíveis.

- **Implementar validações robustas:** Garantir que os dados inseridos sejam corretos (por exemplo, formatos de data e hora)