PROJETO DESENVOLVE

BOM DESPACHO/MG

PROGRAMAÇÃO BÁSICA COM PYTHON

**TRABALHO PRATICO:**

**SISTEMA DE AGENDAMENTOS DE VISITAS TECNICAS**

ALINE BERVIAN

BOM DESPACHO/MG

MARÇO/2025

Sumário

[INTRODUÇÃO 1](#_Toc193452354)

[IMPLEMENTAÇÃO 1](#_Toc193452355)

[SUBTITULO 1](#_Toc193452356)

[CONCLUSÃO 1](#_Toc193452357)

# INTRODUÇÃO

Este trabalho foi projetado para atender a uma dificuldade de gerenciamento de visitas técnicas a produtores rurais, sendo apenas o escopo inicial de um programa maior a ser desenvolvido futuramente.

Este escopo prevê a realização e atualização (*Create*, *Update*) de cadastros de técnicos e clientes, agendamentos e ajustes de agendamentos por parte dos técnicos e gestores (*Create*, *Update*), além da gestão de todas as informações, ajustes e exclusões de clientes e profissionais (*Create*, *Update, Delete*) por parte dos gestores administradores. Também permite a visualização (*Read*) de diferentes informações de acordo com o nível de acesso de cada usuário. Para tal, foram definidos 3 diferentes níveis de acesso para clientes, profissionais e administradores.

O programa está sendo desenvolvido para suprir a dificuldade de gestão e agendas de visitas técnicas em propriedades rurais por parte de uma empresa que presta consultoria e assessoria rural, através de diferentes profissionais cadastrados e a gestão destes por parte da diretoria da empresa.

A empresa oferece diferentes tipos de assessoria, desde treinamento e gestão de equipe, gestão financeira, estruturação de processos e implementação de boas práticas agropecuárias para aprovação em auditorias de laticínios, aprimoramento e gestão da qualidade do produto (leite), capacitação de técnicos de laticínios etc. Ocorre dificuldade de gestão diante da amplitude geográfica de atuação e, por isso, distância física para com os profissionais cadastrados.

Além disso, é necessário que gestores gerenciem a rota realizada pelo profissional, profissionais justifiquem cancelamentos e reagendamentos, profissionais e clientes reportem dificuldades na evolução do serviço, permissão de reagendamento por parte do cliente porém com antecedência de um determinado ‘x’ de dias ou imputar em multas de acordo com a politica da empresa prestadora do serviço. Funções estas que estão no radar e espera-se implementar futuramente, implementando juntamente o front-end que deverá ter fácil usabilidade devido a aptidão dos usuários em questão.

 IMPLEMENTAÇÃO

Para fazer o programa foi utilizado como base inicial o programa exemplo disponibilizado, redefinidas variáveis, alteradas e criadas novas estruturas. Foram utilizadas por completo as funções de estilização do programa disposto como exemplo, uma vez que não houve interesse de tamanha busca externa por parte da programadora que vos fala (acredito que o tempo dispendido seja melhor utilizado se aplicado em aprendizagem de programação front-end futura).

Texto

O conteúdo gerado por IA pode estar incorreto.Dados, em formato de lista, foram utilizados inicialmente apenas para startar os arquivos de armazenamento dos dados. Os arquivos foram estruturados no formato CSV separado por vírgulas.

Qualquer usuário do programa terá acesso às seguintes possibilidades de interação, através do menu inicial:

1. Fazer login – no caso de usuários já cadastrados, através de login (nome\_usuario e senha)
2. Cadastrar Usuário – realizar o cadastro de usuários com permissão ‘cliente’ para utilizar as funções relacionadas à esse nível.
   1. Apenas usuários logados e com permissão “administrador” podem cadastrar também usuários com permissão ‘profissional’ ou ‘administrador’.
3. Visualizar Serviços disponíveis – apresenta os serviços cadastrados no momento da solicitação.
4. Sair do Sistema – fecha o programa.

Ao realizar o login (opção 1), o usuário é levado para tela de Menu interno, onde serão apresentadas opções de ação de acordo com a permissão do usuário logado. Quando o usuário deseja sair da tela, basta digitar ‘0’, tanto no menu inicial quanto no menu interno.

1. Texto

   O conteúdo gerado por IA pode estar incorreto.Usuario permissao == ‘cliente:
2. Texto

   O conteúdo gerado por IA pode estar incorreto.Usuario permissao == ‘profissional:

* ATENÇÃO: função 11. Editar agendamento (data) é a única função que o administrador não possui permissão! Desta forma pensando no bom fluxo da empresa a qual o programa poderá futuramente ser implementado.

1. Texto

   O conteúdo gerado por IA pode estar incorreto.Usuario permissao == ‘administrador’:

## BANCOS DE DADOS

Foram criados manualmente dentro do programa a base de dados inicial, através de listas para os 3 diferentes bancos de dados. Todas essas bases e funções para criação foram inutilizadas (#) após a criação destas.

## USUÁRIOS

Arquivo e csv, utilizado no programa em diversas funções, sendo atribuído à sua chamada o nome de variável de arq\_user\_csv. Possui as seguintes informações:

* Nome-usuario: gerado automaticamente através da variável nome, removendo espaços e transformando todos os caracteres em minúsculo
* Senha: senha de acesso (string)
* Nome: nome e sobrenome do usuario
* Telefone
* Permissão: será a variável responsável pelo condicionamento as possibilidades de função. Gerada automaticamente como ‘cliente’ quando o usuário for diferente de administrador.

Esse banco de dados aceita criação, leitura, edição e exclusão de informações (todas funções CRUD). Os usuários foram estruturados em três níveis de permissão, que prevê diferentes níveis de acessos às informações de cadastro e agendamentos:

1. Administrador: tem permissão de cadastro (*Create*), leitura (*Read*), atualização (*Update*) e exclusão (*Delete*) das informações de todos os clientes, profissionais e administradores. Atenção: há apenas uma função, relacionada a base de dados agenda, que o usuário administrador não pode aplicar.
2. Profissional: tem permissão de cadastro (*Create*) e leitura (*Read*) das informações de todos os clientes e as suas próprias; e permissão de atualização (*Update*) de informação do seu usuário e de usuários com permissão ‘cliente’.
3. Cliente: tem acesso de leitura (*Read*) as informações de cadastro e agendamento restrito às entradas que mencionem seu usuário.
   1. Estrutura de dados

arq\_user\_csv 🡪 arquivo csv de armazenamento dos dados cadastrais de clientes, profissionais e administradores.

USUARIO\_LOGADO 🡪 variável que atualiza de acordo com o login para garantir acesso ao programa.

usuarios 🡪 dicionários com logins como chaves e tuplas nomeadas 'usuario' com os valores contendo todas as informações de cada chave.

usuario 🡪 tuplas que com campos nomeados. Representa os dados estruturados de forma imutável e permitindo acesso por nome (login:nome\_usuario)

* 1. Estrutura do arquivo de registros - arq\_user\_csv

Arquivo CSV – valores separados por vírgulas, onde cada linha representa um registro e as colunas os campos destes registros, da seguinte forma:

* Cabeçalho: é a primeira linha do arquivo, contém 5 campos de registro:

nome\_usuario, senha, nome, telefone, permissao

* Linhas de dados: onde cada linha representa um registro, com todas as informações (valores) dos usuários, sendo cada valor (“célula”) correspondente a um campo (‘collumn’).
  1. Lista de Funcionalidades (CRUD)

1 – Função ‘ler\_usuarios’: serve de base para demais funções, abre o arquivo de usuarios e lê;

* CRUD: *read*

2 – Função ‘fazer\_login’: realiza o login do usuário. Altera a variável USUARIO\_LOGADO;

* CRUD: *read*

3 – Função ‘mostrar\_informacoes’: também serve para ler as informações, mas retorna a impressão das informações solicitadas, de acordo com a permissão;

* CRUD: *read*

4 – Função ‘cadastrar\_usuario’: cria um novo usuário (cria uma nova linha com todas as informações);

* CRUD: *create*

5 – Função ‘atualizar\_senha’: altera a senha do usuário (altera o campo da coluna 4);

* CRUD: *update*

6 – Função ‘excluir\_usuario’: exclui o usuário escolhido do arquivo de registros (deleta toda linha);

* CRUD: *delete*

## SERVIÇOS PRESTADOS

Todos os serviços prestados são cadastrados de acordo com o profissional que realiza o serviço e sua respectiva região, além de conter o valor cobrado. Informações armazenadas em arquivo csv arq\_serv\_csv:

* Código: o usuário que cadastrar gera um código para o serviço;
* Atividade: nome do serviço prestado. Campo para melhor compreensão do usuário;
* Região: região na qual o serviço é prestado, varia de acordo com o técnico responsável.
* Técnico: profissional que executará o serviço;
* Preço: valor a ser cobrado, em reais. Esta variável, coletada em string, é convertida em inteiro no momento da criação e atualização do valor, para possibilitar a ordenação.

Esse banco de dados aceita criação, leitura, atualização e exclusão de informações (CRUD). Aqui, a variável global *USUARIO\_LOGADO* define as permissões para cada função de acordo com o nível de acesso do usuário em questão:

1. Administrador: tem acesso as funções para criar, editar e excluir os dados do arquivo, bem como de leitura das informações.
2. Profissional e Cliente: tem acesso de leitura (*Read*) as informações de cadastro de serviços.
   1. Estrutura de dados

arq\_serv\_csv 🡪 arquivo csv de armazenamento dos dados cadastrais de cada serviço disponível.

USUARIO\_LOGADO 🡪 variável que atualiza de acordo com o login para garantir acesso ao programa.

servicos 🡪 dicionários com codigo como chaves e tuplas nomeadas ‘servico’ com os valores contendo todas as informações de cada chave.

servico 🡪 tuplas que com campos nomeados. Representa os dados estruturados de forma imutável e permitindo acesso por nome (codigo:codigo)

* 1. Estrutura do arquivo de registros - arq\_serv\_csv

Arquivo CSV – valores separados por vírgulas, onde cada linha representa um registro e as colunas os campos destes registros, da seguinte forma:

* Cabeçalho: é a primeira linha do arquivo, contém 5 campos de registro:

codigo,atividade,regiao,tecnico,preco

* Linhas de dados: onde cada linha representa um registro, com todas as informações (valores) do serviço, sendo cada valor (“célula”) correspondente a um campo (‘collumn’).
  1. Lista de Funcionalidades (CRUD)

1 – Função ‘ler\_servicos’: serve de base para demais funções, abre o arquivo de serviços/produtos e lê;

* CRUD: *read*

2 – Função ‘mostrar\_servicos’: também serve para ler as informações, mas retorna a impressão das informações solicitadas, de acordo com a forma de visualização escolhida. Informação filtrada por região (para mostrar ao cliente quais serviços estão disponíveis na região dele), ordenadas alfabeticamente de acordo com código (que tbm corresponde a atividade) ou por ordem crescente de preços;

* CRUD: *read*

4 – Função ‘cadastrar\_serviço’: cria um novo serviço (cria uma nova linha com todas as informações), restrito a administradores; a variável preço coletada em *string*, é convertida em *inteiro* no momento da criação do valor preco, para possibilitar a ordenação.

* CRUD: *create*

5 – Função ‘atualizar\_preco’: altera o preço do serviço prestado (altera o campo da coluna 5), restrito a administradores; a variável preço coletada em *string*, é convertida em *inteiro* no momento da atualização do valor preco, para possibilitar a ordenação.

* CRUD: *update*

6 – Função ‘excluir\_servico’: exclui o serviço escolhido do arquivo de registros (deleta toda linha) , restrito a administradores;

* CRUD: *delete*

## AGENDAMENTO

Os usuários e serviços cadastrados acabam por ser necessários para que se possa preencher esta base de dados, e para que o programa fala sentido. Todos os agendamentos são cadastrados pelo profissional ou administrador. Informações armazenadas em arquivo csv arq\_agen\_csv:

* Cliente: nome e sobrenome do cliente. Deve ser igual ao cadastrado na base de dados de usuário;
* Codigo do serviço: código único para cada serviço (linha) de acordo com serviço, região e profissional.
* Serviço: igual ao campo Atividade na base de dados de serviços. Campo para melhor compreensão do usuário;
* Técnico: nome e sobrenome do profissional. Deve ser igual ao cadastrado na base de dados de usuário;
* Data: data do agendamento. Este campo pode ser atualizado.
* Codigo de agendamento: código único gerado automaticamente através da concatenação de nome do cliente + serviço.

Esse banco de dados aceita criação, leitura, atualização e exclusão de informações (CRUD). Aqui, a variável global *USUARIO\_LOGADO* define as permissões para cada função de acordo com o nível de acesso do usuário em questão:

1. Administrador: tem acesso as funções para visualizar (fazer a gestão) e criar dados do arquivo;
2. Profissional: tem acesso a todas as funções desta base de dados: criar, editar e visualizar as informações;
3. Cliente: tem acesso de leitura (*Read*) as informações de agendamento referentes a seu nome cadastrado.
   1. Estrutura de dados

arq\_agen\_csv 🡪 arquivo csv de armazenamento dos dados de agendamento de cada serviço a ser prestado.

agendas 🡪 dicionários com cod\_agenda como chaves e tuplas nomeadas ‘agenda com os valores contendo todas as informações de cada chave.

agenda 🡪 tuplas que com campos nomeados. Representa os dados estruturados de forma imutável e permitindo acesso por nome (cod\_agenda: cod\_agenda)

* 1. Estrutura do arquivo de registros - arq\_serv\_csv

Arquivo CSV – valores separados por vírgulas, onde cada linha representa um registro e as colunas os campos destes registros, da seguinte forma:

* Cabeçalho: é a primeira linha do arquivo, contém 6campos de registro:

cliente, cod\_serv, servico, tecnico, data, cod\_agenda

* Linhas de dados: onde cada linha representa um registro, com todas as informações (valores) do agendamento, sendo cada valor (“célula”) correspondente a um campo (‘collumn’).
  1. Lista de Funcionalidades (CRUD)

1 – Função ‘ler\_agendas’: serve de base para demais funções, abre o arquivo de agendamentos e lê;

* CRUD: *read*

2 – Função ‘mostrar\_agendas: também serve para ler as informações, mas retorna a impressão das informações solicitadas, de acordo com a permissão do usuário logado. Permissões de cliente e profissionais só permitem a visualização no caso do seu usuário (campo nome) ser mencionado na linha do agendamento.

* CRUD: *read*

4 – Função ‘cadastrar\_agenda: cria um novo agendamento (cria uma nova linha com todas as informações), restrito a profissionais e administradores. Cria automaticamente a variável valor *cod\_agenda*.

* CRUD: *create*

5 – Função ‘atualizar\_agenda: altera a data do agendamento prestado (altera o campo da coluna 5), restrito a profissionais.

* CRUD: *update*

# CONCLUSÃO

O trabalho, apesar de iniciar com uma prévia pronta foi bastante laboroso, foi realizado pensando num problema real de uma empresa real (ou mais de uma inclusive).

Precisa de vários ajustes, gostaria de colocar a possibilidade de seleção de serviço, cliente e tecnico no momento do agendamento para que não haja necessidade de escrita dos termos da forma correta, além disso, gerar código de serviço de forma automática e padronizada.

Acredito que foi uma ótima proposta de trabalho para testar os conhecimentos adquiridos durante o curso. Compreendi a lógica porem sigo com algumas dificuldades de digitar os códigos (que as vezes tem peculiaridades especificas para gerar o comando), um pouco diferente do pensamento lógico do excel que já possuía um pouco de conhecimento.

Acredito que a parte de layout, customização do terminal seja um pouco “sem função real”, uma vez que nenhum usuário utilizará o programa via terminal. Pelo menos não deste programa. Por isso não fiz questão de me aprofundar nesta documentação.

Houve muitos testes, muitos códigos escritos, testados e apagados para que chegasse nessa versão. Identifiquei que preciso aprimorar minha produtividade de escrita, o que melhora com a experiencia também. Estou ansiosa para os próximos desafios.

Solicito e fico aberta aos feedbacks de melhorias.