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 do programa a ser desenvolvido.

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 praticas agropecuárias para aprovação em auditorias de laticínios, aprimoramento e gestão da qualidade do produto final (leite), capacitação de técnicos de laticínios etc.

# IMPLEMENTAÇÃO

## BANCOS DE DADOS

## USUÁRIOS

Todos os usuários podem acessar suas informações básicas de cadastro como:

* Login
* Senha
* Nome cadastrado
* Telefone cadastrado
* Cargo ocupado
* Nível de permissão

Esse banco de dados aceita criação, leitura e exclusão de informações (CR~~U~~D). Os usuários foram estruturados em três níveis de permissão, que prevê diferentes níveis de acessos às informações de cadastro e agendamentos:

1. Administrador: tem acesso as informações de todos os clientes, profissionais e todos os agendamentos.
2. Profissional: tem acesso de cadastro (*Create*) e leitura (*Read*) das informações de todos os clientes e atualização (*Update*) e leitura (*Read*) das informações referente aos agendamentos vinculados ao seu usuário/cadastro.
3. Cliente: tem acesso de leitura (*Read*) as informações de cadastro e agendamento somente respectivo ao seu usuário.

# controle de acesso: criar todos os niveis somente adminisrtadores, profissional e clientes podem add apenas clientes

## SERVIÇOS PRESTADOS

Todos os serviços prestados são cadastrados de acordo com o profissional que realiza o serviço e sua respectiva região e preço cobrado:

* Código
* Serviço
* Região
* Técnico
* Preço

Esse banco de dados aceita criação, leitura, atualização e exclusão de informações (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 acesso as informações de todos os clientes, profissionais e todos os agendamentos.
2. Profissional: tem acesso de cadastro (*Create*) e leitura (*Read*) das informações de todos os clientes e atualização (*Update*) e leitura (*Read*) das informações referente aos agendamentos vinculados ao seu usuário/cadastro.
3. Cliente: tem acesso de leitura (*Read*) as informações de cadastro e agendamento somente respectivo ao seu usuário.

* Implementação: Para cada elemento da sua solução (usuário e produto/serviço) você deve: (1) descrever a estrutura de dados escolhida para carregar as informações, (2) como o arquivo de registro está estruturado e (3) a lista de funcionalidades daquele elemento (associe cada funcionalidade a uma letra da sigla CRUD).

## AGENDAMENTO

Todos os agendamentos

Pedro Souza,rp01,reproducao,Beatriz Almeida,2025-03-29

Maria Oliveira,le01,limpeza de equipamento,Ana Costa,2025-04-01

Joao Silva,aq02,assistencia qualidade,Lucas Pereira,2025-03-28

# CONCLUSÃO

TEXTO

Conclusão: Aproveite esse espaço para descrever coisas como dificuldades encontradas, escolhas bem sucedidas, o que faria diferente, etc.

# cliente apenas atualizam (lógica interna para atualizar somente seu próprio cadastro)

if USUARIO\_LOGADO.permissao == 'administrador':

console.print("[bold cyan]1.[/bold cyan] [bold white]Atualizar senha[/bold white]")

console.print("[bold cyan]2.[/bold cyan] [bold white]Visualizar cadastros de usuários[/bold white]") mostrar informacoes

console.print("[bold cyan]3.[/bold cyan] [bold white]Excluir cadastro usuario[/bold white]")

        console.print("[bold cyan]4.[/bold cyan] [bold white]Visualizar cadastros de serviços[/bold white]")

        console.print("[bold cyan]5.[/bold cyan] [bold white]Atualizar valores de serviços[/bold white]")

        console.print("[bold cyan]6.[/bold cyan] [bold white]Adicionar serviço[/bold white]")

        console.print("[bold cyan]7.[/bold cyan] [bold white]Excluir serviço[/bold white]")

        console.print("[bold cyan]8.[/bold cyan] [bold white]Visualizar agendamentos[/bold white]")

console.print("[bold cyan]0.[/bold cyan] [bold white]Para fazer logout digite '0'[/bold white]")

opcao = Prompt.ask("[bold yellow]Digite o número da opção desejada[/bold yellow]", choices=["0","1", "3", "4", "5", "6", "7"])

elif USUARIO\_LOGADO.permissao == 'profissional':

console.print("[bold cyan]1.[/bold cyan] [bold white]Atualizar senha[/bold white]")

console.print("[bold cyan]2.[/bold cyan] [bold white]Visualizar informações de cadastro [/bold white]")

console.print("[bold cyan]3.[/bold cyan] [bold white]Excluir cadastro[/bold white]")

        console.print("[bold cyan]4.[/bold cyan] [bold white]Visualizar cadastros de serviços[/bold white]")

        console.print("[bold cyan]8.[/bold cyan] [bold white]Visualizar agendamentos[/bold white]")

        console.print("[bold cyan]9.[/bold cyan] [bold white]Criar agendamento[/bold white]")

console.print("[bold cyan]0.[/bold cyan] [bold white]Para fazer logout digite '0'[/bold white]")

opcao = Prompt.ask("[bold yellow]Digite o número da opção desejada[/bold yellow]", choices=["0","1", "2", "3","7", "8"])

else:

console.print("[bold cyan]1.[/bold cyan] [bold white]Atualizar senha[/bold white]")

console.print("[bold cyan]2.[/bold cyan] [bold white]Visualizar informações de cadastro [/bold white]")

console.print("[bold cyan]0.[/bold cyan] [bold white]Para fazer logout digite '0'[/bold white]") ##visualizar agendamento

opcao = Prompt.ask("[bold yellow]Digite o número da opção desejada[/bold yellow]", choices=["0","1", "2"])

Cliente, Codigo Serviço,Serviço, Técnico,Data

with open(arq\_agen\_csv, mode='r', newline='', encoding='utf-8') as file:

reader = csv.reader(file)

rows = list(reader) # Armazenar todas as linhas do arquivo

# Encontrar a linha correspondente ao código de agendamento

for row in rows:

if row[1] == cod\_agenda: # Supondo que a segunda coluna (índice 1) seja o código de agendamento

row[4] = nova\_data # Atualizar a data (supondo que a data seja a 5ª coluna, índice 4)

updated = True

break

def atualizar\_agenda(agendas, arq\_agen\_csv):

    if USUARIO\_LOGADO is not None and USUARIO\_LOGADO.permissao == 'profissional':

        console.print(Panel('''[bold yellow]Atualização de Data do serviço a ser prestado[/bold yellow]\nPor favor, insira o nome do cliente e codigo do serviço para reagendar.''',

                            title="Atualizar agenda", expand=False))

        cod\_agenda = Prompt.ask("[bold cyan]Codigo de agendamento: [/bold cyan]")

        nova\_data = Prompt.ask("[bold cyan]Nova data do serviço (aaaa-mm-dd): [/bold cyan]")

        if agendas.get(cod\_agenda, None) is not None:

            with open(arq\_agen\_csv, mode='w', newline='') as file:

                writer = csv.writer(file)

                for \_, agenda in agendas.items():

                    if agenda.cod\_agenda == cod\_agenda:

                        writer.writerow([agenda.cliente, agenda.cod\_serv, agenda.servico, agenda.tecnico, nova\_data, agenda.cod\_agenda])

                    else:

                       console.print("[bold yellow]Código de agendamento incorreto[/bold yellow]", style="yellow")

                       return False

            console.print(f"[bold orange]Reagendamento confirmado[/bold green]\nAgenda de código '{cod\_agenda}'\nReagendado para '{nova\_data}'")

            return True

    else:

        console.print("[bold yellow]Você não possui permissão para realizar esta ação. Contate a administração![/bold yellow]", style="yellow")

        return False