

Universidade Federal de Uberlândia

# Projeto J – Sistema de Chamados de Suporte de TI

Adalberto Caldeira

Sistemas da Informação / 3º Período ! / 2º semestre 2025  
Onboarding Lipai 2025/2025

# Descrição do Projeto

- Problema: Dificuldade em registrar, atribuir e acompanhar solicitações de suporte técnico (hardware, rede, sistemas) de forma organizada.
- Solução: Um sistema CLI (Linha de Comando) para gestão completa do ciclo de vida de tickets de TI.
- Público-Alvo: Departamentos de TI de empresas, escolas ou organizações que precisam centralizar pedidos de ajuda.
- Diferencial: Controle de acesso (Admin vs. Usuário Comum) e relatórios de atendimento

# Demonstração Funcionalidades

## TELA INICIAL

```
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ python3 main.py

--- Login ---
Usuário: admin
Senha: 123321
```

# Demonstração Funcionalidades

- Menu administrativo

```
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ python3 main.py
```

```
--- Login ---  
Usuário: admin  
Senha: 123321  
Login bem-sucedido! Bem-vindo, admin (admin).
```

```
--- Sistema de Chamados de Suporte de TI ---
```

1. Gerenciar Usuários
2. Gerenciar Técnicos
3. Gerenciar Setores
4. Gerenciar Chamados (Admin)
5. Relatórios (Admin)
6. Gerenciar Usuários de Acesso
0. Sair

```
Escolha uma opção: □
```

# Demonstração Funcionalidades

- Cadastro de novo usuário no sistema é feito pelo admin

```
--- Menu Usuários ---
1. Cadastrar Usuário
2. Listar Usuários
0. Voltar
Escolha uma opção: 1
CPF: 111.111.111-11
Nome: Alfredo Boruto
Email: alfboruto@gmail.com
1. TI (ID: 1)
2. RH (ID: 2)
3. Financeiro (ID: 3)
Selecione o Setor do Usuário: 3
```

# Demonstração Funcionalidades

- Salva o registro do usuário cadastrado

```
Escolha uma opção: 1
CPF: 111.111.111-11
Nome: Alfredo Boruto
Email: alfboruto@gmail.com
1. TI (ID: 1)
2. RH (ID: 2)
3. Financeiro (ID: 3)
Selecione o Setor do Usuário: 3
Erro ao ler linha do usuario: Erro ao realizar a conversão -> Quantidade campos errada 7, o correto são 4
Usuário cadastrado com sucesso!

--- Menu Usuários ---
1. Cadastrar Usuário
2. Listar Usuários
0. Voltar
Escolha uma opção: 
```

# Demonstração Funcionalidades

- Listagem de usuários

```
--- Menu Usuários ---
```

- 1. Cadastrar Usuário
- 2. Listar Usuários
- 0. Voltar

Escolha uma opção: 2

CPF: 11122233344, Nome: Alice Silva, Email: alice@example.com, Setor: 1

CPF: 55566677788, Nome: Bruno Souza, Email: bruno@example.com, Setor: 2

CPF: 11111111111, Nome: Alfredo Boruto, Email: alfboruto@gmail.com, Setor: 3

```
--- Menu Usuários ---
```

- 1. Cadastrar Usuário
- 2. Listar Usuários
- 0. Voltar

Escolha uma opção: □

# Demonstração de Funcionalidades

- Gerenciar Chamados

```
--- Menu Usuários ---
1. Cadastrar Usuário
2. Listar Usuários
0. Voltar
Escolha uma opção: 0
```

```
--- Sistema de Chamados de Suporte de TI ---
1. Gerenciar Usuários
2. Gerenciar Técnicos
3. Gerenciar Setores
4. Gerenciar Chamados (Admin)
5. Relatórios (Admin)
6. Gerenciar Usuários de Acesso
0. Sair
Escolha uma opção: 4
```

# Demonstração Funcionalidades

- Menu Chamados (admin)

```
--- Menu Chamados ---  
1. Abrir Chamado  
2. Atribuir Técnico  
3. Atualizar Status  
4. Listar Todos os Chamados  
0. Voltar  
Escolha uma opção: █
```

# Demonstração Funcionalidades

- Lista Chamados Abertos

```
--- Menu Chamados ---
```

1. Abrir Chamado
2. Atribuir Técnico
3. Atualizar Status
4. Listar Todos os Chamados
0. Voltar

Escolha uma opção: 4

```
ID: 1, Usuário: 11122233344, Título: Problema de Rede, Status: aberto, Prioridade: baixa, Técnico: N/A
ID: 2, Usuário: 55566677788, Título: Erro no Sistema X, Status: em atendimento, Prioridade: media, Técnico: 101
ID: 3, Usuário: 11122233344, Título: Computador lento, Status: resolvido, Prioridade: alta, Técnico: 102
ID: 101, Usuário: 11122233344, Título: Problema com o mouse, Status: aberto, Prioridade: alta, Técnico: N/A
```

```
--- Menu Chamados ---
```

1. Abrir Chamado
2. Atribuir Técnico
3. Atualizar Status
4. Listar Todos os Chamados
0. Voltar

Escolha uma opção: □

# Demonstração Funcionalidades

- Área do Usuário

```
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ ls ./data/senhas.csv  
./data/senhas.csv  
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ cat ./data/senhas.csv  
admin,123321,admin,  
alice_user,alicepass,user,11122233344  
bruno_user,brunopass,user,55566677788  
jose_user,josepass,user,103334889629  
novousuario,senhanova,user,22222222222  
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ python3 main.py  
  
--- Login ---  
Usuário: 
```

# Demonstração Funcionalidades

- Login

```
--- Login ---  
Usuário: bruno_user  
Senha: brunopass  
Login bem-sucedido! Bem-vindo, bruno_user (user).
```

```
--- Sistema de Chamados de Suporte de TI ---
```

1. Gerenciar Meus Chamados
2. Meus Relatórios
0. Sair

```
Escolha uma opção: █
```

# Demonstração Funcionalidades

- Erro na listagem dos Relatórios

```
--- Sistema de Chamados de Suporte de TI ---
1. Gerenciar Meus Chamados
2. Meus Relatórios
0. Sair
Escolha uma opção: 2
Traceback (most recent call last):
  File "/home/adalberto/github/lipai/atividades/semana4/projeto-j/main.py", line 419, in <module>
    main()
  File "/home/adalberto/github/lipai/atividades/semana4/projeto-j/main.py", line 410, in main
    menu_relatorios(repo_chamados, user_role, user_identity)
TypeError: menu_relatorios() missing 2 required positional arguments: 'user_role' and 'user_identity'
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ 
```

# Demonstração Funcionalidades

- Correção do problema, faltou passar mais argumentos



```
> __pycache__          402     print("1. Gerenciar Meus Chamados")
  + __init__.py         403     print("2. Meus Relatórios")
  + repositorio_autenticacao.py 404     print("0. Sair")
  + repositorio_chamados.py    405     opcao = input("Escolha uma opção: ")
  + repositorio_setores.py    406
  + repositorio_tecnicos.py   407
  + repositorio_usuarios.py   408         if opcao == "1":
  + adalbertocbfilho.md       409             menu_chamados(repo_chamados, user_role, user_identity)
  - admin_roteiro.txt          410         elif opcao == "2":
  - erros.txt                  411             menu_relatorios(repo_chamados, user_role, user_identity)
  - main.py                    412         elif opcao == "0":
  - README.md                  413             print("Saindo do sistema...")
  - user_roteiro.txt           414             break
  - else:                      415             print("Opção inválida.")
  - 416
  - 417
  - 418     if __name__ == "__main__":
  - 419         main()
  - 420
  - 421
```

# Demonstração Funcionalidades

- Listagem Relatórios

```
--- Login ---  
Usuário: bruno_user  
Senha: brunopass  
Login bem-sucedido! Bem-vindo, bruno_user (user).
```

```
--- Sistema de Chamados de Suporte de TI ---  
1. Gerenciar Meus Chamados  
2. Meus Relatórios  
0. Sair  
Escolha uma opção: 2
```

```
--- Menu Relatórios ---  
1. Listar Meus Chamados Abertos  
2. Listar Todos os Meus Chamados  
0. Voltar  
Escolha uma opção: 1  
Você não possui chamados abertos.
```

```
--- Menu Relatórios ---  
1. Listar Meus Chamados Abertos  
2. Listar Todos os Meus Chamados  
0. Voltar
```

# Demonstração Funcionalidades

- Lista todos os chamados

```
--- Menu Relatórios ---
1. Listar Meus Chamados Abertos
2. Listar Todos os Meus Chamados
0. Voltar
Escolha uma opção: 2
ID: 2, Título: Erro no Sistema X, Status: em atendimento, Prioridade: media
```

```
--- Menu Relatórios ---
1. Listar Meus Chamados Abertos
2. Listar Todos os Meus Chamados
0. Voltar
Escolha uma opção: ■
```

# Demonstração funcionalidades

- Gerenciar Chamados, Status de Chamados, Listar Chamados

```
1. Gerenciar Meus Chamados
2. Meus Relatórios
0. Sair
Escolha uma opção: 1

--- Menu Chamados ---
1. Abrir Chamado
2. Listar Meus Chamados
3. Ver Status de um Chamado
0. Voltar
Escolha uma opção: 3
1. Erro no Sistema X (ID: 2)
Selecione o Chamado para verificar status (ID): 1
Chamado ID: 2, Título: Erro no Sistema X, Status: em atendimento, Prioridade: media

--- Menu Chamados ---
1. Abrir Chamado
2. Listar Meus Chamados
3. Ver Status de um Chamado
0. Voltar
Escolha uma opção: 2
ID: 2, Título: Erro no Sistema X, Status: em atendimento, Prioridade: media

--- Menu Chamados ---
1. Abrir Chamado
2. Listar Meus Chamados
3. Ver Status de um Chamado
0. Voltar
Escolha uma opção: 
```

# Estrutura do projeto

- Diretórios e arquivos principais

```
adalberto@adalberto-X751LJ:~/github/lipai/atividades/semana4/projeto-j$ tree --gitignore
```

```
.  
|__ adalbertocbfilho.md  
|__ admin_roteiro.txt  
|__ data  
|   |__ chamados.csv  
|   |__ senhas.csv  
|   |__ setores.csv  
|   |__ tecnicos.csv  
|   |__ usuarios.csv  
|__ erros.txt  
|__ main.py
```



# Estrutura do projeto

- Diretórios e arquivos principais

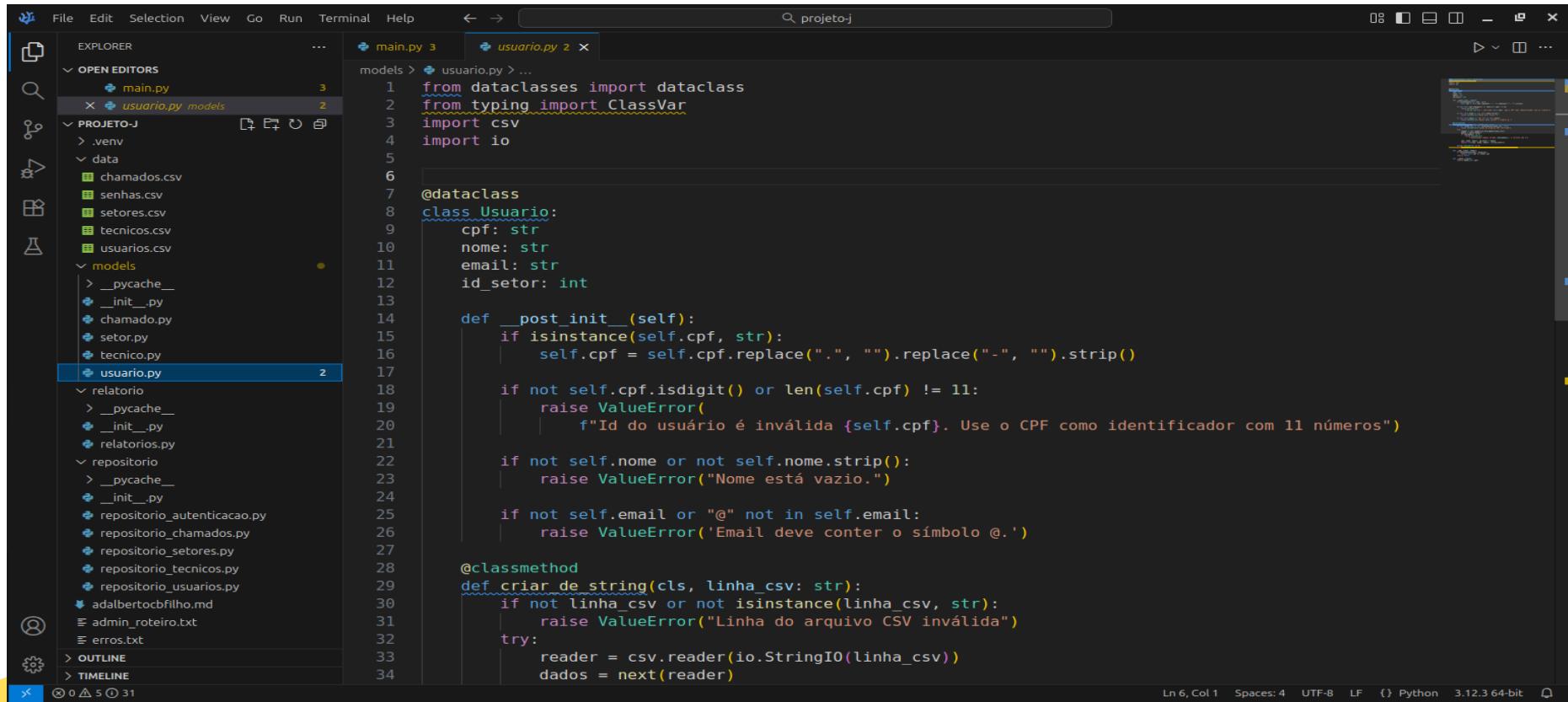
```
└── main.py
└── models
    ├── chamado.py
    ├── __init__.py
    └── __pycache__
    ├── setor.py
    ├── tecnico.py
    └── usuario.py
└── README.md
└── relatorio
    ├── __init__.py
    └── __pycache__
        └── relatorios.py
```

```
└── .
    └── repositorio
        ├── __init__.py
        ├── __pycache__
        ├── repositorio_autenticacao.py
        ├── repositorio_chamados.py
        ├── repositorio_setores.py
        ├── repositorio_tecnicos.py
        └── repositorio_usuarios.py
    └── user_roteiro.txt
```

8 directories, 24 files

adalberto@adalberto-X751LJ:~/github/lipai/ativida

# Classes e Objetos Usuario.py



The screenshot shows a dark-themed Python code editor interface. On the left is the Explorer sidebar, which lists the project structure. The main area displays the content of the `usuario.py` file.

```
models > usuario.py > ...
1  from dataclasses import dataclass
2  from typing import ClassVar
3  import csv
4  import io
5
6
7  @dataclass
8  class Usuario:
9      cpf: str
10     nome: str
11     email: str
12     id_setor: int
13
14     def __post_init__(self):
15         if isinstance(self.cpf, str):
16             self.cpf = self.cpf.replace(".", "").replace("-", "").strip()
17
18         if not self.cpf.isdigit() or len(self.cpf) != 11:
19             raise ValueError(
20                 f"Id do usuário é inválida {self.cpf}. Use o CPF como identificador com 11 números")
21
22         if not self.nome or not self.nome.strip():
23             raise ValueError("Nome está vazio.")
24
25         if not self.email or "@" not in self.email:
26             raise ValueError('Email deve conter o símbolo @.')
27
28     @classmethod
29     def criar_de_string(cls, linha_csv: str):
30         if not linha_csv or not isinstance(linha_csv, str):
31             raise ValueError("Linha do arquivo CSV inválida")
32         try:
33             reader = csv.reader(io.StringIO(linha_csv))
34             dados = next(reader)
```

```
28     @classmethod
29     def criar_de_string(cls, linha_csv: str):
30         if not linha_csv or not isinstance(linha_csv, str):
31             raise ValueError("Linha do arquivo CSV inválida")
32         try:
33             reader = csv.reader(io.StringIO(linha_csv))
34             dados = next(reader)
35             if len(dados) != 4:
36                 raise ValueError(
37                     f"Quantidade campos errada {len(dados)}, o correto são 4")
38
39             cpf, nome, email, id_setor = dados
40             return cls(cpf, nome, email, int(id_setor))
41
42         except ValueError as e:
43             raise ValueError(f"Erro ao realizar a conversão -> {e}")
44
45     def __eq__(self, other):
46         if isinstance(other, Usuario):
47             return self.cpf == other.cpf
48         return False
49
50     def __hash__(self):
51         return hash(self.cpf)
52
```

# Acesso aos dados (CSV)

- No início do programa os dados já estão em memória

```
> relatorio
└ repository
    > __pycache__
    ⚡ __init__.py
    ⚡ repository_autenticacao.py 2
    ⚡ repository_chamados.py
    ⚡ repository_setores.py
    ⚡ repository_tecnicos.py
    ⚡ repository_usuarios.py
    adalbertocbfilho.md
    admin_roteiro.txt
    erros.txt
    main.py 3
    README.md
    user_roteiro.txt

③
> OUTLINE
> TIMELINE

> 0 ▲ 5 ⓘ 36
389
390 def main():
391     repo_usuarios = RepositorioUsuarios()
392     repo_tecnicos = RepositorioTecnicos()
393     repo_chamados = RepositorioChamados()
394     repo_setores = RepositorioSetores()
395     repo_autenticacao = RepositorioAutenticacao()
396
397     user_role, user_identity = login(repo_autenticacao)
398
399     if not user_role:
400         print("Falha na autenticação. Encerrando o sistema.")
401         return
402
403     while True:
404         print("\n--- Sistema de Chamados de Suporte de TI ---")
405         if user_role == "admin":
406             print("1. Gerenciar Usuários")
407             print("2. Gerenciar Técnicos")
408             print("3. Gerenciar Setores")
409             print("4. Gerenciar Chamados (Admin)")
410             print("5. Relatórios (Admin)")
411             print("6. Gerenciar Usuários de Acesso")
412             print("0. Sair")
```

Ln 390, Col 12 Spaces: 4 UTF-8 LF {} Python 3.12.3 64-bit

# Acesso aos Dados

- Ao Cadastrar (Adicionar):
  - Linha 80: `repo_usuarios.adicionar(usuario)` -> Salva no CSV de usuários.
  - Linha 104: `repo_tecnicos.adicionar(...)` -> Salva no CSV de técnicos.
  - Linha 131: `repo_setores.adicionar(...)` -> Salva no CSV de setores.
  - Linha 165: `repo_chamados.adicionar(...)` -> Salva no CSV de chamados.
  - Linha 294: `repo_autenticacao.adicionar_usuario_acesso(...)` -> Salva no CSV de autenticação.
- Ao Modificar (Atualizar):
  - Linha 190: `repo_chamados.atualizar(chamado)` -> Atualiza o status e reescreve o CSV.
  - Linha 219: `repo_chamados.atualizar(chamado)` -> Atualiza o status e reescreve o CSV.
  -

# Problemas

- O usuário tem acesso a todos os chamados do sistema
- O sistema de senhas não é criptografado
- O sistema pode ficar muito lento se a quantidade de chamados for alta, pois sua estrutura de dados é uma lista, manejo de memória inadequado
- O programador não testou adequadamente o sistema
- Necessidade de automatizar testes e diminuir o tamanho da main , criando objetos do tipo menu()



# Melhorias

- Criar um sistema gráfico
- Possibilidade de adicionar , som , vídeo e imagem para que o relato do problema fique mais claro para o técnico.
- Adicionar o número serial do produto que deu problema
- Colocar o sistema em nuvem com redundância e banco de dados sql
- Ir acessando os dados e empilhando e desempilhando da memória
- Usar testes unitários automatizados
- Adicionar um tutorial em vídeo para os usuários e administradores do sistema

# Obrigado

