

FIAP - Faculdade de Informática e Administração Paulista

Análise e Desenvolvimento de Sistemas

Moisés Waidemann - 1TDSPH

Gabriel Sbrana - 1TDSPH

Thiago Mota - 1TDSPH



PetSOS

Python

São Paulo - SP

2025

PetSOS – Plataforma de Apoio a Tutores de Animais

PetSOS é uma aplicação Python simples, com interface via terminal, que simula a lógica de uma plataforma onde um tutor pode cadastrar e gerenciar informações de animais perdidos e encontrados. Focada em um único usuário, a solução serve como protótipo funcional para a futura interface de um sistema real de apoio à causa animal em situações de desaparecimento.

Visão geral do PetSOS.py **O arquivo reúne quatro blocos principais:**

“Banco de dados” em memória – listas globais que guardam todas as informações durante a sessão.

Funções de entrada e validação – cada campo do tutor é solicitado e validado separadamente (pedir_nome, pedir_cpf, etc.).

Funções de domínio – cadastro / atualização de animais, criação de usuário, autenticação, exibição de perfil e reset.

Função principal() – orquestra todo o fluxo oferecendo um menu de opções em loop e limpa a memória após a execução.

1. Estrutura de dados

Logo no topo há 14 listas vazias (por exemplo nomes_cadastrados, meus_animais_perdidos) que simulam tabelas de um banco de dados relacional. Esse design simplifica testes, pois nada é escrito em disco, mas implica que todos os dados são perdidos quando o script termina.

2. Interface de menu

A função `exibir_menu()` imprime o cabeçalho e devolve a escolha do usuário como string. Todas as interações posteriores dependem desse retorno, que é tratado com o match-case.

3. Funções de domínio **Cadastro de animais**

`cadastrear_animal_perdido()` cria uma chave simples concatenando nome + data + descrição e a põe em `meus_animais_perdidos`, evitando duplicatas.

`cadastrear_animal_encontrado()` tenta localizar essa mesma chave em “perdidos”; se achar, move-a para `meus_animais_encontrados`.

Usuário e autenticação

`criando_usuario()` compõe uma string multilinhas com todos os dados pessoais e armazena em `dados_tutores`.

`login_tutor(cpf, senha)` concatena CPF+senha para formar um token e grava em `tutores_cadastrados`, que depois é usado por `exibir_perfil()` para validar até 5 tentativas.

Limpeza total

`resetar_dados()` esvazia todas as listas globais.

4. Orquestração em principal()

A função inicia variáveis de controle (`contador`, `dados_tutor`) e entra num laço que só termina quando o usuário responde “não” à pergunta “Deseja voltar ao menu?”.

O match `opcao`: executa sete ramos:

Cadastro de animal perdido

... encontrado

Cadastro ou alteração do tutor – o primeiro registro é detectado por `contador == 0`; chamadas subsequentes sobrescrevem dados existentes.

Exibição de perfil (exige login)

Listar animais perdidos

Listar animais encontrados

Reset geral

Após cada operação o fluxo volta ao menu principal, mantendo o estado em memória.