

CENTRO UNIVERSITÁRIO FEI

Vitor da Silva Rodrigues de Souza

PROJETO FEIFOOD

Relatório Final do Projeto apresentado ao
Centro Universitário FEI, como parte dos
requisitos da Disciplina “CCP110 –
Fundamentos de Algoritmos”

São Bernardo do Campo, SP

2025

SUMÁRIO

1	INTRODUÇÃO	3
2	DESENVOLVIMENTO	3
3	RESULTADOS.....	4
4	CONCLUSÕES	5
5	REFERÊNCIAS	5

1 INTRODUÇÃO

O projeto FEIFOOD tem como objetivo desenvolver uma plataforma simples de pedidos de comida, inspirada em serviços como o iFood. A proposta é permitir que usuários possam se cadastrar, buscar alimentos, montar um carrinho, finalizar pedidos e avaliá-los, utilizando uma interface de terminal em Python.

As tecnologias utilizadas foram a linguagem de programação **Python** e o uso de arquivos de texto (.txt) para a persistência de dados. O sistema foi desenvolvido com foco no perfil de **usuário**, contemplando todas as funcionalidades obrigatórias da disciplina.

2 DESENVOLVIMENTO

O sistema foi estruturado em um único arquivo Python (menu.py) e utiliza quatro arquivos de texto para armazenar os dados:

- **cadastros.txt** → armazena nome, usuário e senha
- **alimentos.txt** → armazena os alimentos disponíveis
- **pedidos.txt** → armazena os pedidos realizados
- **avaliacoes.txt** → armazena as avaliações dos pedidos

A seguir, são descritas as principais funções implementadas no sistema:

Cadastro e Login

A função `novo_cadastro()` permite que o usuário insira seu nome, escolha um nome de usuário e defina uma senha. Esses dados são armazenados em `cadastros.txt`, com cada linha representando um usuário.

A função `login()` verifica se o nome de usuário e a senha informados correspondem a uma linha existente no arquivo. Se a autenticação for bem-sucedida, o usuário é direcionado ao menu logado, onde pode acessar as funcionalidades do sistema.

Menu Logado

A função `menu_usuario_logado(usuario)` apresenta as opções disponíveis para o usuário autenticado: buscar alimentos, acessar o carrinho, avaliar pedidos ou fazer logout. O carrinho é mantido como uma lista temporária durante a sessão.

Busca de Alimentos

A função `buscar_alimentos(usuario)` permite que o usuário visualize todos os alimentos cadastrados ou filtre por nome. Os dados são lidos do arquivo `alimentos.txt`, e o usuário pode escolher um alimento pelo ID para adicioná-lo ao carrinho.

Carrinho

A função `carrinho(carrinho_do_usuario, usuario)` exibe os itens adicionados pelo usuário, permitindo que ele remova produtos ou finalize o pedido. O carrinho é uma lista de IDs de alimentos que o usuário selecionou durante a navegação.

Finalização de Pedido

A função `finalizar_pedido(usuario, lista_ids)` grava o pedido no arquivo `pedidos.txt`. Cada usuário possui sua própria sequência de IDs de pedidos, garantindo organização e evitando conflitos. Após finalizar, o sistema pergunta se o usuário deseja avaliar o pedido imediatamente.

Avaliação de Pedido

A função `avaliar_pedido(usuario, id_pedido=None)` permite que o usuário atribua uma nota de 1 a 5 ao pedido e escreva um comentário. O sistema valida se o pedido pertence ao usuário antes de permitir a avaliação. As avaliações são registradas em `avaliacoes.txt` com o separador “;”, evitando problemas com vírgulas nos comentários.

Estrutura dos Arquivos de Dados

Todos os arquivos `.txt` são armazenados na mesma pasta do programa principal. Cada linha representa uma entidade (usuário, alimento, pedido ou avaliação), e os dados são separados por vírgulas ou ponto e vírgula, conforme o tipo de informação. Essa abordagem garante simplicidade na leitura e escrita dos dados, sem necessidade de banco de dados externo.

4 RESULTADOS

O sistema foi testado com sucesso e todas as funcionalidades foram implementadas corretamente.

Apresentação das Funcionalidades O programa exibe menus claros e interativos. O fluxo de uso segue a seguinte estrutura:

Menu principal

1 → Cadastro 2 → Login 0 → Sair

Menu logado

1 → Buscar alimentos 2 → Carrinho 3 → Avaliar pedido 0 → Logout

Carrinho

1 → Remover item 2 → Finalizar pedido 0 → Voltar

Demonstração da Persistência de Dados Os arquivos .txt são atualizados corretamente após cada ação. Por exemplo, ao cadastrar um novo usuário, uma nova linha é adicionada em cadastros.txt.

Validação das Regras O sistema impede login com senha incorreta, não permite finalizar pedidos com carrinho vazio e valida se o pedido pertence ao usuário antes de permitir a avaliação.

4 CONCLUSÕES

O projeto FEIFOOD atingiu todos os objetivos propostos. A estrutura em Python com arquivos .txt permitiu uma implementação simples e funcional. O sistema oferece uma experiência completa para o usuário, desde o cadastro até a avaliação de pedidos. O uso de IDs independentes por usuário e a separação clara dos dados garantem organização e escalabilidade. O projeto pode ser expandido futuramente com funcionalidades administrativas e interface gráfica.

5 REFERÊNCIAS

FEI – Centro Universitário. Manual de Normas da ABNT. Documentação oficial do Python: <https://docs.python.org/3/>
Material da disciplina CCP110 – Fundamentos de Algoritmos.