



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
CÂMPUS DE PRESIDENTE PRUDENTE

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO” -
FACULDADE DE CIÊNCIAS E TECNOLOGIA, CAMPUS PRESIDENTE
PRUDENTE

Nathan Alves da Cruz Silveira
Nicolas Fernando da Silva
Lucas Henrique Gregorio

**Trabalho de Estrutura de Dados I:
Projeto Semestral**

1. Introdução

Este relatório documenta o desenvolvimento de um sistema de atendimento urbano inteligente, com suporte a registros de bairros, cidadãos, serviços públicos e ocorrências emergenciais. O projeto utiliza estruturas de dados como listas encadeadas, tabelas de hashing, pilhas, filas com prioridade e árvores binárias (BST e AVL).

2. Estrutura do Projeto

O sistema foi dividido em três arquivos principais:

- ``Estruturas.h``: Contém as definições das estruturas de dados e os protótipos das funções utilizadas.
- ``EstruturasCodigos.c``: Implementa todas as funções do sistema, incluindo cadastro, busca, exibição, simulação e organização em árvores.
- ``main.c``: Arquivo principal com o menu interativo para execução e teste das funcionalidades.

3. Etapas Implementadas

3.1 Parte 1 – Estrutura Básica

- Cadastro de bairros com tabela hash baseada em ID.
- Cadastro de cidadãos e serviços.
- Registro de ocorrências em uma fila com prioridade baseada na gravidade.

3.2 Parte 2 – Pilha e Hash

- Inclusão da estrutura de pilha para armazenar o histórico de atendimentos.
- Implementação de funções de hashing para busca eficiente por CPF e ID de bairros.
- Atualização das funções ``registrarOcorrencia``, ``simulacao`` e ``limparTudo`` para integrar a pilha.

3.3 Parte 3 – Árvores BST e AVL

- Inserção e busca em Árvore Binária de Busca (BST) para ordenação por ID de ocorrência.
- Inserção e busca em Árvore AVL para ordenação e busca por gravidade.
- Visualização das ocorrências organizadas em ordem crescente de ID ou gravidade.

4. Funcionalidades do Menu

O menu presente no arquivo ``teste.c`` permite executar operações como:

- Cadastrar bairros, cidadãos, serviços e ocorrências.
- Visualizar todas as entidades cadastradas.

- Buscar cidadãos por CPF e bairros por ID.
- Verificar a fila de ocorrências por prioridade.
- Acessar histórico de ocorrências (pilha).
- Consultar árvores BST e AVL por ID e gravidade.

5. Simulação Automatizada

A função `simulacao` executa automaticamente 10 operações aleatórias com base em arquivos `.txt` contendo nomes, CPFs, bairros, etc. Essa simulação garante que sempre haverá pelo menos uma entidade de cada tipo registrada e uma aleatoriedade baseada em dez eventos distintos.

6. Conclusão

O sistema foi projetado para simular um ambiente urbano inteligente, aplicando conceitos de estruturas de dados fundamentais para melhorar o desempenho de busca, cadastro e priorização de atendimentos.