Relatório do Projeto: Sistema de Gerenciamento de Combate a Queimadas em Vegetação

Wilson Fernandes - Ciência da Computação

2025

1 Introdução

Este relatório apresenta o desenvolvimento do projeto "Sistema de Gerenciamento de Combate a Queimadas em Vegetação", realizado para a disciplina de Algoritmos em Grafos. O objetivo do sistema é permitir o gerenciamento eficiente de regiões, queimadas e conexões entre regiões, possibilitando o registro, consulta e análise de dados relevantes para o combate a queimadas, utilizando estruturas de grafos e algoritmos clássicos da área. O projeto foi desenvolvido em linguagem C, com foco em modularização, clareza e aderência aos requisitos propostos.

2 Contextualização e Estrutura do Projeto

O sistema foi dividido em módulos, cada um responsável por uma parte fundamental da aplicação:

- **graph.c / graph.h**: Responsável pela estrutura do grafo, representando regiões como vértices e conexões (distâncias) como arestas. Utiliza listas de adjacência para garantir eficiência nas operações de inserção e consulta, justificando a escolha pela facilidade de manipulação e baixo custo para grafos esparsos.
- **region.c / region.h**: Gerencia os dados das regiões, como nome, tipo de vegetação e área. As regiões são armazenadas em um vetor de ponteiros para estruturas `Region`, facilitando o acesso e a manipulação dinâmica.
- **wildfire_management.c / wildfire_management.h**: Implementa a lista encadeada de queimadas (`WildfireList`), permitindo o registro de queimadas associadas a regiões, datas e intensidades.
- **report.c / report.h**: Gera relatórios detalhados sobre regiões e queimadas, atendendo aos requisitos de consulta do projeto.

- **main.c**: Implementa a interface textual (menu), integrando as funcionalidades dos módulos e controlando o fluxo do sistema.

A modularização foi uma decisão importante para garantir a organização do código, facilitar a manutenção e permitir a reutilização de funções.

3 Funcionalidades Implementadas

O sistema implementa as seguintes funcionalidades, conforme os requisitos do projeto:

- Cadastro de regiões, com nome, tipo de vegetação e área.
- Adição de arestas (distâncias) entre regiões.
- Registro de queimadas, associando-as a uma região, data e intensidade.
- Geração de relatórios:
  - Listagem de todas as regiões cadastradas.
  - Listagem de todas as queimadas registradas.
  - Relatório de queimadas por região.
- Persistência de dados: salvamento e carregamento de regiões, arestas e queimadas em arquivos binários.
- Interface de menu interativo, com validação de entradas e mensagens de erro.

Exemplo de uso:

```
=== Sistema de Gerenciamento de Combate a Queimadas ===
1. Cadastrar região
2. Adicionar aresta (distância entre regiões)
3. Registrar queimada
4. Relatório de regiões
5. Relatório de queimadas
6. Relatório de queimadas por região
7. Salvar dados
8. Carregar dados
0. Sair
Escolha uma opção: 1
Nome da região: Cerrado Norte
Tipo de vegetação: Cerrado
Área (hectares): 150.0
Região cadastrada!
```

4 Decisões de Projeto

- **Listas de adjacência**: Escolhidas para representar o grafo devido à eficiência em operações de inserção e consulta, especialmente em grafos esparsos.
- **Modularização**: O código foi dividido em módulos para separar responsabilidades, facilitar testes e manutenção.
- **Uso de ponteiros**: Permitiram a manipulação dinâmica das estruturas, como regiões e queimadas, otimizando o uso de memória.
- **Persistência binária**: Arquivos binários foram utilizados para garantir a integridade e eficiência no salvamento e carregamento dos dados.

5 Principais Funções e Algoritmos

- `bfs` (*graph.c*): Implementa a busca em largura, permitindo percorrer o grafo e identificar componentes conexas.
- `dfs` (*graph.c*): Realiza a busca em profundidade, útil para análise de conectividade.
- `dijkstra` (*graph.c*): Calcula o menor caminho entre duas regiões, permitindo simular deslocamentos eficientes das equipes de combate.
- `save_regions`, `save_edges`, `save_wildfires` (*region.c*, *graph.c*, *wildfire_management.c*): Funções responsáveis por salvar os dados em arquivos binários.
- `load_regions`, `load_edges`, `load_wildfires`: Realizam a leitura dos dados salvos, restaurando o estado do sistema.
- `report_all_regions`, `report_all_wildfires`, `report_wildfires_by_region` (*report.c*): Geram relatórios detalhados para consulta.
- `validate_date` (*utils.c*): Garante a integridade das datas inseridas pelo usuário.

6 Testes Realizados

Foram implementados testes automatizados (arquivo *tests.c*) para validar funções essenciais, como `validate_date`, criação de regiões, inserção de arestas e registro de queimadas. Além disso, testes manuais foram realizados durante o uso do menu interativo, garantindo o correto funcionamento das funcionalidades e a robustez da persistência de dados.

7 Desafios e Melhorias

Desafios: O principal desafio foi garantir a integridade dos dados ao salvar e carregar arquivos binários, especialmente ao lidar com ponteiros e estruturas dinâmicas. Esse desafio foi superado com a implementação cuidadosa das funções de persistência, garantindo que os dados fossem serializados e desserializados corretamente.

Melhorias Futuras:

- Utilizar uma fila de prioridade (heap) para otimizar o algoritmo de Dijkstra em grafos maiores, reduzindo a complexidade e melhorando o desempenho.
- Implementar interface gráfica para facilitar o uso por operadores não técnicos.
- Adicionar testes automatizados mais abrangentes, cobrindo casos de borda e validação de entradas.
- Permitir remoção e edição de regiões e queimadas, aumentando a flexibilidade do sistema.
- Melhorar a validação de datas e outros campos, tornando o sistema mais robusto contra entradas inválidas.

Essas melhorias visam tornar o sistema mais eficiente, amigável e confiável, alinhando-se com as necessidades reais de um sistema de gerenciamento de queimadas.

8 Atendimento aos Requisitos do Projeto

Todas as funcionalidades essenciais foram implementadas conforme os requisitos do projeto: cadastro e consulta de regiões e queimadas, geração de relatórios, persistência de dados e uso de algoritmos clássicos de grafos. O sistema foi testado e demonstrou atender aos objetivos propostos, estando preparado para futuras expansões.

9 Conclusão

O desenvolvimento deste projeto proporcionou uma experiência prática valiosa com algoritmos de grafos, manipulação de estruturas dinâmicas e persistência de dados em C. As decisões de projeto adotadas garantiram a clareza, eficiência e robustez do sistema. O sistema está pronto para ser expandido e melhorado, servindo como base sólida para aplicações reais de gerenciamento de queimadas e análise de redes.

10 Referências

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. *Algoritmos: Teoria e Prática* (3ª edição).
- Szwarcfiter, J. L. *Teoria Computacional de Grafos.*