Relatório Técnico: Implementação de Estruturas de Dados Lineares

1. Estruturas de Dados Implementadas

1.1 Lista Circular Encadeada (`CircularLinkedList.cpp`)

- **Descrição**: Lista encadeada onde o último nó aponta de volta para o primeiro.
- **Operações Principais**:
  - `insertAtStart/End`: O(1)
  - `removeFromStart`: O(1)
  - `removeFromEnd`: O(n)
- **Complexidade Espaço**: O(n)
- **Problemas**: Gerenciamento de ponteiros na circularidade exigiu tratamento especial no destrutor.

1.2 Lista Duplamente Encadeada (`DoublyLinkedList.cpp`)

- **Descrição**: Nós com ponteiros para anterior e próximo.
- **Destaque**: Inserção/remoção em posições arbitrárias com O(1) após localização.
- **Complexidade Busca**: O(n)

1.3 Fila de Prioridade (`PriorityQueue.cpp`)

- **Base**: Utilizou `SortedDoublyLinkedList`.
- **Operações**:
  - `enqueue`: O(n) (mantém ordenação)
  - `dequeue`: O(1)

2. Divisão de Módulos

- **Core**:
  - `LinearDataStructure.hpp`: Interface base.

- **Implementações**:
  - Arquivos separados por estrutura (e.g., `Stack.cpp`, `Queue.cpp`).
- **Aplicação**:
  - `UsuarioBandeijao.cpp`: Demonstração prática.

3. Observações de Desenvolvimento

- **Desafios**:
  - Gerenciamento de memória com `unique_ptr` em estruturas encadeadas.
  - Validação de limites em operações de posição arbitrária.
- **Soluções**:
  - Padronização de tratamento de erros com exceções.
  - Iteradores para evitar vazamentos.

4. Conclusão

As implementações atingiram os objetivos com: ✔ Complexidades conforme esperado

✔ Segurança no gerenciamento de memória

✔ Modularidade para extensão

Próximos Passos:

- Adicionar testes unitários abrangentes
- Implementar versões thread-safe