

**INSTITUTO FEDERAL**  
**CATARINENSE**  
Câmpus Videira

**Turma: Ciência da Computação**

**Aluno: Bruno Pergher**

**Disciplina: Estrutura de dados**

**Professor: Manasses Ribeiro**

---

## **Relatório do desenvolvimento do projeto final**

### **1. Introdução**

O problema a ser resolvido com o projeto final é utilizarmos os conhecimentos que obtivemos ao longo do semestre e aplicarmos em um único projeto que consistia em a partir de listas encadeadas duplas criarmos um “banco” de nomes inserindo os nomes através de um arquivo de texto que o professor disponibilizou ou inserindo nomes manualmente, e após termos vários nomes em diversas listas que foram separadas utilizando a técnica de hashing o objetivo era ordená los em ordem alfabética.

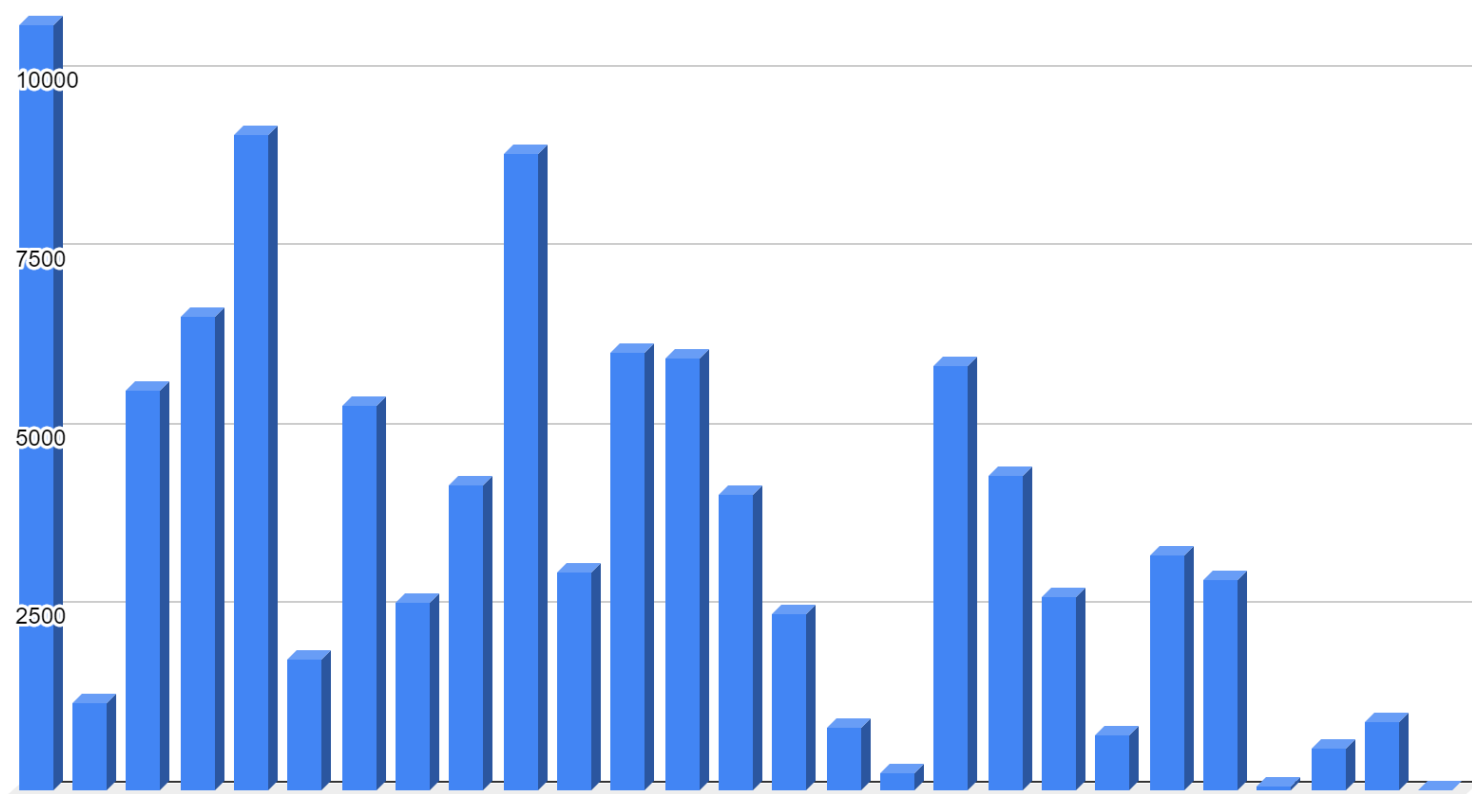
### **2. Desenvolvimento**

Ao iniciar o processo de desenvolvimento comecei por algo que já havia feito, inserindo ao projeto o funcionamento de uma lista dupla encadeada, após isso estar pronto chegamos em um novo desafio a implementação de hashing na hora de salvar nomes nas listas.

Para implementar o hashing necessitamos nos preocupar para evitar as colisões e para esse caso a resposta já estava pronta, através da nossa lista encadeada dupla evitamos as colisões devido a algumas características da lista como uma dela é não existir um limite de tamanho, e também todas as validações que implementamos nela garantindo que não deixe elementos nulos nem ocupando uma mesma posição.

O hash implementando se baseia no padrão ASCII que convertendo uma string para um valor inteiro nos possibilita usarmos o resto de divisão baseado em um número de chaves que irá ser criados separa os dados de maneira a agrupar palavras semelhantes juntas, no meu caso eu utilizei um valor de chaves que seria suficiente para se necessário criar uma chave por letra do alfabeto sendo elas maiuscula ou minuscula.

Um ponto importante sobre o uso da técnica de hashing é o **hashing uniforme** que se baseia em dividir as lista para garantir que as listas tenham tamanho parecidos distribuindo de maneira igualitária os elementos, na implementação de meu sistema eu priorizei que o hash garantisse que separasse os elementos por letra assim fugindo um pouco da ideia de hash uniforme, como podemos perceber no histograma baseado nas letras e quantidade de elementos por lista a seguir:



*Figura 1 -Histograma da quantidade de nomes por lista*

Após tudo estar funcionando, só restava a ordenação dos nomes que para tal foi utilizado o quicksort, um algoritmo que nesse caso recursivo organiza os nomes em ordem alfabética se baseando em pivôs e ordenando simultaneamente a lista da esquerda para direita e da direita pra esquerda.

### **3. Conclusão**

O trabalho realmente nos trouxe na prática um pouco de cada conteúdo que trabalhamos e colocando a prova nossos conhecimentos, em prol disso foi uma experiência muito completa desenvolver o projeto final trazendo mais desafios no desenvolvimento, mas finalizando com um programa funcional, implementando todas as funções das listas encadeadas duplas de imprimir os nomes, consulta-los, removê-los, e adicioná los tanto manualmente como através do arquivo de texto e também ordenando os utilizando o quick sort, onde todas essas funções podem ser controladas pelo usuário através de um menu.

```
===== Banco de nomes =====  
1 - Inserir nomes do arquivo  
2 - Inserir nome  
3 - Pesquisar nome  
4 - Imprimir nomes por lista  
5 - Imprimir todos os nomes  
6 - Quantidade de nomes em uma lista  
7 - Quantidade de nomes total  
8 - Ordenar uma lista  
9 - Ordenar todas as listas  
10 - Remover nome  
11 - Todas as listas e quantidade de elementos nelas.  
Opcao: █
```

*Figura 2 - Menu de ações*

Através desse menu o usuário pode realizar todas as atividades comentadas anteriormente, assim tendo liberdade para usar da maneira que desejar.