

# Relatório de Desenvolvimento do Projeto BigNumber

## Membros do Grupo

1. Guilherme Arruda Panfiglio - 11202230588
2. Guilherme Leão de Oliveira - 11202231193
3. Maurício Fagner Martins Rabelo - 11202231013

## Organização do Código

A organização do código no projeto BigNumber foi planejada para garantir modularidade, legibilidade e facilidade de manutenção. O projeto está dividido em cinco arquivos principais: `bignumber.h`, `bignumber.c`, `auxiliar.h`, `auxiliar.c` e `client.c`.

O arquivo `bignumber.h` define a interface pública do projeto. Ele contém as declarações das estruturas de dados, como a estrutura `BigNumber`, e as assinaturas das funções que implementam as operações aritméticas fundamentais, como soma, subtração, multiplicação e divisão. Sua principal função é oferecer uma descrição clara e organizada das funcionalidades disponíveis para quem utiliza a biblioteca.

O arquivo `bignumber.c` contém a implementação dessas funções, incluindo as operações mencionadas e outras funcionalidades relevantes para a manipulação de `BigNumbers`, como criação, impressão e liberação de memória. Este arquivo encapsula a lógica principal do projeto.

Já os arquivos `auxiliar.h` e `auxiliar.c` são destinados às funções de suporte que auxiliam na execução das operações principais. O arquivo `auxiliar.h` declara as assinaturas dessas funções, enquanto o `auxiliar.c` implementa os detalhes dessas operações. Entre as funcionalidades auxiliares estão o gerenciamento eficiente de nós da estrutura `BigNumber`, a remoção de zeros à esquerda e a verificação de propriedades numéricas, como paridade. Essa separação contribui para que o código principal das operações aritméticas permaneça mais limpo e focado.

O arquivo `client.c` atua como ponto de entrada do programa, centralizando a execução das operações principais. Por fim, a automação do processo de compilação foi realizada por meio de um Makefile, que garante a compilação eficiente de todos os componentes do projeto e simplifica o processo para os desenvolvedores.

## Funcionalidades Implementadas Corretamente

O projeto implementa uma série de funcionalidades para a manipulação de `BigNumbers`, cobrindo operações aritméticas fundamentais e avançadas. A inicialização de um `BigNumber` a

partir de uma string configura corretamente o sinal, o número de dígitos e os nós encadeados que representam os dígitos. A exibição de `BigNumbers` permite visualizar os valores completos, respeitando o sinal e a ordem dos dígitos, enquanto a liberação de memória garante a desalocação correta dos recursos, prevenindo vazamentos.

As operações de soma e subtração foram implementadas para lidar com números de tamanhos e sinais diferentes, incluindo o ajuste de transporte e a determinação do sinal do resultado. Além disso, a multiplicação foi desenvolvida em duas versões: a básica, utilizando algoritmos tradicionais, e a otimizada, com o uso do algoritmo de Karatsuba, conhecido por melhorar a eficiência para números grandes. A divisão e o cálculo do resto da divisão foram implementados com algoritmos tradicionais e precisos, ajustando os sinais e os dígitos conforme necessário.

Por fim, a exponenciação rápida foi implementada na sua versão recursiva, utilizando otimizações baseadas na paridade dos números para reduzir o número de multiplicações necessárias e melhorar o desempenho geral do cálculo. Essas funções oferecem uma base robusta e eficiente para a manipulação de `BigNumbers`, atendendo aos requisitos do projeto.

## Interface Pública do tipo `BigNumber`

A interface pública do tipo `BigNumber` é composta pelas assinaturas das funções disponíveis no arquivo `bignumber.h`. Essas funções permitem a criação, manipulação e execução de operações aritméticas com `BigNumbers`. Abaixo estão as assinaturas:

```
1 // -----
2
3 BigNumber create_big_number(char *str_number);
4 BigNumber sum_big_numbers(BigNumber x, BigNumber y);
5 BigNumber subtract_big_numbers(BigNumber x, BigNumber y);
6 BigNumber divide_big_numbers(BigNumber dividend, BigNumber divisor);
7 BigNumber multiply_big_numbers(BigNumber x, BigNumber y);
8 BigNumber fast_exponentiation(BigNumber base, BigNumber exponent);
9 BigNumber remainder_of_division(BigNumber dividend, BigNumber divisor);
10 BigNumber multiply_karatsuba_big_numbers(BigNumber x, BigNumber y);
11
12 void print_big_number(BigNumber x);
13 void free_big_number(BigNumber x);
14
15 // -----
```

## Algoritmos e Estruturas de Dados Avançadas Utilizadas

O projeto emprega certos algoritmos e estruturas de dados para otimizar as operações com `BigNumbers`. O algoritmo de Karatsuba foi utilizado na função de multiplicação, substituindo a abordagem tradicional e reduzindo o número de multiplicações necessárias ao dividir os números grandes em partes menores e combinar os resultados de maneira eficiente. A exponenciação rápida foi implementada de forma recursiva na função correspondente, explorando a paridade

do expoente para reduzir significativamente o número de multiplicações durante o cálculo de potências.

Além disso, o tipo `BigNumber` utiliza uma lista duplamente encadeada para armazenar os dígitos dos números, permitindo que operações como soma, subtração e remoção de zeros à esquerda sejam realizadas de forma eficiente, navegando pelos dígitos de forma flexível a partir de qualquer extremidade. Essas técnicas foram aplicadas nas operações principais do projeto, como multiplicação, divisão e manipulação de `BigNumbers`, garantindo um desempenho aprimorado para cálculos de alta complexidade.

## Divisão de Trabalho dentro do Grupo

A divisão de trabalho na equipe foi realizada de maneira equilibrada, com cada membro contribuindo para diferentes aspectos do projeto, garantindo eficiência e colaboração. Guilherme Arruda foi responsável pelas tarefas iniciais e de suporte, incluindo o controle do repositório, o commit inicial, criação do `Makefile`, implementação de funções básicas para exibir e liberar memória de um `BigNumber`, ajustes na lógica de execução do programa, implementação do algoritmo de divisão, redação do relatório e das descrições das funções. Ele também realizou a revisão final do código, focando nos aspectos visuais, como a padronização do formato, organização dos arquivos e remoção de elementos desnecessários.

Guilherme Leão concentrou-se em otimizações e implementações avançadas, como a refatoração das estruturas de dados `BigNumber` e `Node`, a introdução de paridade para otimizar a exponenciação rápida, e o desenvolvimento e integração de algoritmos sofisticados, como o método de Karatsuba para multiplicação, exponenciação rápida e cálculo de restos. Maurício, por sua vez, contribuiu significativamente nas etapas iniciais do projeto, elaborando o cabeçalho `bignumber.h`, implementando a função para criar um `BigNumber` e desenvolvendo as primeiras operações aritméticas, como soma e subtração além de contribuir com a redação das descrições das funções. Ele também lidou com casos envolvendo sinais, removeu zeros à esquerda e organizou funções auxiliares, garantindo uma base sólida para o desenvolvimento do restante do projeto.

Embora as tarefas tenham sido divididas e os commits atribuídos individualmente, é importante destacar que o desenvolvimento do projeto foi um esforço coletivo. Todos os membros do grupo contribuíram de forma ativa em todas as etapas, seja por meio de sugestões, troca de ideias, pesquisas bibliográficas ou discussões para aprimorar a lógica e a eficiência das implementações.