

Título

ALGORITMOS E ESTRUTURA DE DADOS 3 - Um trabalho prático para aprofundar em conceitos e evolução da computação

Resumo

O trabalho desenvolvido neste semestre proporcionou uma perspectiva aprofundada e multifacetada sobre o complexo cenário da computação. Mais do que uma mera visão superficial, garantiu-se uma compreensão robusta que abrange não apenas o estado atual e as tendências mais recentes, mas também a intrincada evolução temporal e os desafios históricos que moldaram a área. Essa imersão permitiu desvendar as raízes das inovações, compreendendo como cada avanço se construiu sobre as bases do passado.

Um pilar central dessa jornada foi o desenvolvimento prático de algoritmos e estruturas de dados. A experiência foi intencionalmente pautada por limitações que remetem a épocas passadas da computação, uma abordagem crucial para internalizar que essas ferramentas essenciais não surgem de um vácuo conceitual. Pelo contrário, elas nascem de necessidades prementes e desafios concretos que surgiram em contextos históricos específicos. Essa vivência prática solidificou a compreensão de que, com o tempo e a evolução tecnológica, essas ferramentas são continuamente aperfeiçoadas, refinadas e otimizadas.

Consequentemente, esses algoritmos e estruturas de dados, uma vez aprimorados, servem de base sólida e indispensável para novas criações, pavimentando o caminho para o desenvolvimento de sistemas mais complexos, eficientes e inovadores. É notável como, apesar do avanço vertiginoso da tecnologia, muitas dessas estruturas e algoritmos fundamentais permanecem relevantes e aplicáveis até hoje. Sua longevidade atesta sua robustez conceitual e sua capacidade de adaptação a diferentes paradigmas computacionais, reiterando a importância de um entendimento profundo de seus princípios subjacentes.

Introdução

Na disciplina de ALGORITMOS E ESTRUTURA DE DADOS 3, foi-nos proposta uma série de trabalhos práticos concebidos para promover uma compreensão aprofundada de conceitos fundamentais em ciência da computação. Estes trabalhos abordaram uma gama diversificada de tópicos, incluindo, mas não se limitando a, algoritmos de ordenação, técnicas de criptografia, funções de hash e estruturas de dados complexas como árvores.

O objetivo principal desses exercícios práticos era não apenas assimilar a teoria por trás dessas estruturas de dados e algoritmos, mas também compreender a sua necessidade

intrínseca e as diversas formas como podem ser aplicadas em cenários do mundo real. Para ilustrar a importância da persistência de dados e a eficiência no gerenciamento de recursos, especialmente em contextos onde a memória primária era um recurso limitado, optou-se por manipular e armazenar dados diretamente em arquivos, utilizando a representação em bytes. Essa abordagem permitiu demonstrar como os dados podem ser salvos e recuperados de forma eficaz, evitando a dependência excessiva da memória volátil e espelhando as práticas que eram comuns em épocas em que as restrições de memória eram mais pronunciadas. Através desta metodologia, os alunos puderam experimentar em primeira mão os desafios e as soluções envolvidas na gestão de dados fora do ambiente da memória principal, consolidando a compreensão sobre a importância de algoritmos e estruturas de dados otimizados para diversas aplicações.

Ademais, a divisão de trabalhos foi meticulosamente delineada em quatro etapas cruciais, cada uma visando aprofundar o conhecimento em aspectos fundamentais da organização e manipulação de dados.

A primeira etapa concentrou-se na criação de um sistema CRUD (Create, Read, Update, Delete) operando sobre um arquivo de bytes. Este trabalho inicial abordou a ordenação e a manipulação de arquivos sequenciais, estabelecendo uma base sólida para a compreensão das operações básicas de armazenamento de dados. O objetivo era familiarizar os alunos com a persistência de dados em sua forma mais fundamental, permitindo a criação, leitura, atualização e exclusão de registros de forma eficiente em um ambiente de arquivo simples.

No segundo trabalho, o foco migrou para a atualização da manipulação de arquivos, com a introdução de técnicas para trabalhar com arquivos indexados. Esta fase aprofundou-se no uso de estruturas de dados avançadas, como árvores B, B+ ou B*, deixando a escolha da implementação a critério do aluno, o que incentivou a pesquisa e a compreensão das nuances de cada estrutura. Adicionalmente, foi explorada a utilização de Hash Extensível para demonstrar uma forma mais eficiente de busca de itens, otimizando o tempo de acesso aos dados. A lista invertida também foi abordada, um mecanismo de busca extremamente eficiente e ainda amplamente empregado em sistemas de busca modernos, ressaltando sua relevância prática.

A terceira entrega teve como objetivo principal aprofundar o entendimento de casamento de padrões e compactação de arquivos. Na parte de compactação, foram definidos o uso dos algoritmos LZW (Lempel-Ziv-Welch) e Huffman, ambos algoritmos de compressão de dados sem perdas, essenciais para otimizar o armazenamento e a transmissão de informações. O estudo desses algoritmos permitiu aos alunos compreender os princípios subjacentes à redução do tamanho de arquivos, um tópico de grande importância na computação.

Por último, a quarta e derradeira etapa foi dedicada ao aprofundamento em criptografia. Foi definido o estudo de algoritmos de criptografia que servem como bases para sistemas de segurança modernos ou que ainda são amplamente utilizados hoje, como o RSA (Rivest–Shamir–Adleman). Esta fase final visou proporcionar uma compreensão robusta dos princípios e aplicações da criptografia, capacitando os alunos a entender como a segurança da informação é garantida em diversos contextos, desde comunicações digitais até transações financeiras.

Desenvolvimento

O processo de desenvolvimento, embora árduo, revelou-se imensamente gratificante. Esta fase crucial não se limitou à mera implementação de funcionalidades, mas englobou a aplicação rigorosa de técnicas de versionamento de código, como o Git, que se mostrou indispensável para a gestão colaborativa do projeto. A constante busca por conhecimento externo foi uma prática comum, com a internet servindo como um vasto repositório de informações e exemplos práticos. Um recurso notável foi o repositório GitHub do professor Kutova, que ofereceu insights valiosos e soluções para desafios específicos.

A fluidez do desenvolvimento foi garantida por uma divisão de tarefas clara e eficaz entre os membros da equipe, Pedro Félix e João Paulo Vaz. Essa colaboração permitiu que cada um se concentrasse em áreas de expertise, otimizando o tempo e a qualidade do trabalho. Contudo, o caminho não foi isento de obstáculos. Um dos maiores desafios foi a transição da teoria para a prática, especialmente no que tange à manipulação de bytes e arquivos. Compreender os conceitos abstratos e aplicá-los de forma concreta, adaptando-se às nuances do trabalho com dados em baixo nível, exigiu persistência e um profundo mergulho na documentação e em exemplos práticos. A superação desses desafios não só solidificou o aprendizado, mas também reforçou a capacidade da equipe de transformar o conhecimento em soluções funcionais e robustas.

Aprofundando nos detalhes do desenvolvimento, nossa equipe empregou uma gama diversificada de ferramentas e recursos, indo além das metodologias tradicionais. A integração de Inteligências Artificiais (IAs) desempenhou um papel crucial, não apenas como um auxílio, mas como um catalisador para otimizar processos complexos e gerar soluções inovadoras. Estas IAs foram aplicadas em várias fases do projeto, desde a análise de dados e otimização de algoritmos até a automatização de tarefas repetitivas, permitindo que a equipe se concentrasse em desafios mais estratégicos e criativos.

Para o desenvolvimento e codificação, optamos por ambientes de desenvolvimento integrado (IDEs) robustos e versáteis. O IntelliJ, com sua interface intuitiva e recursos avançados de refatoração, depuração e análise de código, foi fundamental para garantir a qualidade e a

eficiência do desenvolvimento. Sua capacidade de oferecer sugestões inteligentes e realizar verificações em tempo real contribuiu significativamente para a produtividade da equipe e a redução de erros.

Paralelamente, o Visual Studio Code foi empregado devido à sua leveza, flexibilidade e, notavelmente, pela vasta gama de extensões disponíveis. Uma funcionalidade particularmente valiosa que exploramos foi a capacidade de, por meio de extensões específicas, ler e interpretar arquivos em seu formato binário (bytes). Isso se mostrou indispensável para tarefas que exigiam uma compreensão aprofundada da estrutura de dados em um nível de baixo nível, permitindo-nos manipular e analisar informações de forma precisa e eficiente, especialmente em cenários de depuração complexos ou otimização de performance.

Em suma, o desenvolvimento dos desafios foi um processo multifacetado, que demandou a utilização de uma vasta quantidade de material e recursos. A combinação estratégica de tecnologias de ponta, como IAs, com ferramentas de desenvolvimento consagradas e funcionalidades específicas de leitura de arquivos, foi a chave para superar os obstáculos e alcançar os objetivos propostos. Este esforço coletivo e a utilização inteligente de cada ferramenta resultaram em um produto final robusto e de alta qualidade.

Na escolha e aprofundamento dos algoritmos, onde o professor não havia imposto definições prévias, optamos pela implementação da Árvore B+. A decisão por este tipo de estrutura de dados foi motivada principalmente pela sua notável eficiência em operações de busca. A Árvore B+ se destaca por sua organização em páginas interligadas, o que otimiza significativamente o desempenho ao localizar registros, tornando-a ideal para grandes volumes de dados. Contudo, é importante ressaltar que essa performance superior vem acompanhada de uma complexidade de implementação consideravelmente maior, exigindo um entendimento aprofundado de seus mecanismos internos.

Em relação aos métodos de criptografia, nossa escolha recaiu sobre dois algoritmos distintos. Primeiramente, o RSA, que foi recomendado pelo professor. A relevância do RSA transcende o ambiente acadêmico, sendo amplamente reconhecido e utilizado em cenários computacionais reais devido à sua robustez e segurança comprovadas. Sua importância para a proteção de dados e comunicações é inegável, solidificando sua posição como um dos pilares da criptografia moderna. Em segundo lugar, selecionamos a Cifra de Vigenère. Embora mais antiga, a Cifra de Vigenère nos proporcionou uma flexibilidade maior para modificações e adaptações específicas às necessidades do nosso projeto. Sua natureza permite personalizações que se alinham de forma mais eficaz com os requisitos e desafios intrínsecos ao trabalho que desenvolvemos, oferecendo um contraponto interessante à complexidade do RSA.

Testes e Resultados

Os testes foram conduzidos utilizando uma base de dados com em torno 100 mil registros, o que proporcionou uma base sólida para o desenvolvimento robusto do projeto. A metodologia empregada consistiu em testes manuais, que permitiram cobrir a maioria das situações propostas pelo professor. Cada etapa do processo incluiu verificações rigorosas para garantir o funcionamento adequado dos algoritmos e identificar oportunidades de melhoria, como o aprimoramento da capacidade de armazenamento da função hash.

Os resultados alcançados foram altamente positivos, demonstrando a extrema eficácia dos algoritmos e estruturas implementadas. Mesmo em cenários de busca por dados localizados no final do arquivo, o desempenho se manteve consistente, assegurando que, mesmo diante de manipulações no arquivo de dados, todas as estruturas continuariam operando normalmente e com alta eficiência.

Conclusão

Em suma, este trabalho proporcionou uma compreensão aprofundada de diversas estruturas de dados, revelando como cada uma surgiu como uma resposta direta às necessidades e limitações tecnológicas de suas respectivas épocas. Analisar essa evolução histórica não só desmistificou a complexidade dessas estruturas, mas também ressaltou sua relevância contínua no cenário da programação.

Além disso, a colaboração na elaboração deste projeto foi crucial para o desenvolvimento de uma perspectiva prática sobre o trabalho em equipe e a manutenção de código. A experiência demonstrou a importância intrínseca de um processo contínuo de melhoria, correção e adaptação do código, antecipando e gerenciando os erros que são inerentes ao desenvolvimento de software. A capacidade de identificar falhas, implementar soluções eficazes e refatorar o código para maior clareza e eficiência tornou-se um aprendizado inestimável.

De uma forma mais ampla, a visão adquirida através deste projeto é extremamente valiosa para quem busca compreender os múltiplos setores da programação. Cada área possui suas próprias peculiaridades, limitações e demandas específicas, e este trabalho serviu como um microcosmo que ilustra a interconexão desses elementos. A análise detalhada das estruturas de dados e a prática da manutenção de código oferecem uma base sólida para navegar e inovar em qualquer campo da engenharia de software.

Por fim, expresso minha sincera gratidão ao professor Hayla. Sua orientação e os ensinamentos fundamentais que nos foram transmitidos foram a base essencial que possibilitou a realização deste trabalho. O conhecimento e a metodologia fornecidos foram cruciais para que pudéssemos explorar os temas propostos com profundidade e

rigor, culminando em um projeto que não apenas atendeu aos objetivos, mas também superou as expectativas em termos de aprendizado e aplicação prática.