Universidade Federal do Amazonas

Instituto de Computação

Bancos de Dados I – 2022/02 (realizada em 2023)

Professor: Altigran Soares da Silva <[alti@icomp.ufam.edu.br](mailto:alti@icomp.ufam.edu.br)>

Monitor: Tarsis Azevedo <[tarsis.azevedo@icomp.ufam.edu.br](mailto:tarsis.azevedo@icomp.ufam.edu.br)>

# Trabalho Prático 2 – 17 de Abril de 2023

# Data da Entrega: 15 de Maio de 2023

1. Apresentação

Este trabalho consiste na implementação de programas para armazenamento e consulta em dados armazenados em memória secundária utilizando as estruturas de arquivo de dados e índice estudadas nas aulas. Os programas devem fornecer suporte para a inserção, assim como diferentes formas de busca, seguindo as técnicas apresentadas nas aulas de organização e indexação de arquivos.

O trabalho deve ser implementado na linguagem C++ utilizando as bibliotecas padrão de chamadas de sistema disponíveis no Linux.

Os dados para testes e avaliação dos programas implementados estão disponíveis [neste arquivo](https://drive.google.com/file/d/1EVoP0d9Wwzj1O6eoFIkel9I3cpe43Gbv/view?usp=sharing) em formato CSV que servirá apenas como entrada de dados.

1. Descrição do trabalho

O arquivo de dados deverá armazenar registros de dados sobre artigos científicos publicados em conferências. A estrutura deste arquivo será a seguinte:

| Campo | Tipo | Descrição |
| --- | --- | --- |
| ID | inteiro | Código identificador do artigo |
| Título | alfa 300 | Título de artigo |
| Ano | inteiro | Ano de publicação do artigo |
| Autores | alfa 150 | Lista dos autores do artigo |
| Citações | inteiro | Número de vezes que o artigo foi citado |
| Atualização | data e hora | Data e hora da última atualização dos dados |
| Snippet | alfa entre 100 e 1024 | Resumo textual do artigo |

Os seguintes programas devem ser implementados:

1. *upload <file>*: Programa que fará a carga inicial dos dados de entrada que irá criar um banco de dados composto pelos seguintes arquivos:

* Arquivo de dados organizado por hashing
* Arquivo de índice primário usando uma B+Tree armazenada em memória secundária
* Arquivo de índice secundário usando uma outra B+Tree em memória secundária

1. *findrec <ID>:* Programa que busca diretamente no arquivo de dados por um registro com o ID informado e, se existir, retorna os campos do registro, a quantidade de blocos lidos para encontrá-lo e a quantidade total de blocos do arquivo de dados;
2. *seek1 <ID>*: Programa que devolve o registro com ID igual ao informado, se existir, pesquisando através do arquivo de índice primário, mostrando todos os campos, a quantidade de blocos lidos para encontrá-lo no arquivo de índice e a quantidade total de blocos do arquivo de índice primário;
3. *seek2 <Titulo>*: Programa que mostra os dados do registro que possua o Título igual ao informado, se existir, pesquisando através do arquivo de índice secundário, informando a quantidade de blocos lidos para encontrá-lo no arquivo de índice e a quantidade total de blocos do arquivo de índice secundário
4. Registro das Equipes

Para registrar a equipe, os membros deverão postar uma mensagem com os nomes e e-mails dos membros da equipe para o canal #trabalho-pratico-2 até as 23:59 do dia 26/04/2023.

Atenção: (1) os e-mails de todos os integrantes devem ser do domínio icomp.ufam.edu.br; (2) As equipes devem ter 2 ou 3 integrantes; (2) Não serão aceitas mudanças nos integrantes da equipe depois que mensagem for postada portanto mensagens editadas serão desconsideradas.

1. O que entregar
2. Os arquivos-fonte dos programas, com comentários adequados para permitir a correção
3. Os programas prontos para serem testados. Os nomes dos programas devem seguir as instruções indicadas no item 2 deste trabalho.
4. A documentação do projeto dos programas deve ser disponibilizada em um único arquivo PDF nomeado TP2\_documentação.pdf, registrando as todas as decisões de projeto tomadas, incluindo:
   1. A estrutura de cada arquivo de dados e índices;
   2. Quais fontes formam cada programa;
   3. As funções que cada fonte contém;
   4. Quem desenvolveu cada fonte/função;
   5. Qual o papel de cada função;

5. Como entregar

A entrega será feita em um container do Docker que deverá incluir, além da documentação e dos programas fonte, todo o ambiente operacional necessário para o desenvolvimento e execução do trabalho, e toda e qualquer biblioteca necessária para execução do trabalho sem erros.

O container deverá ser nomeado "tp2\_<nome1\_nome2\_nome3>" onde <nome1\_nome2\_nome3> deve ser substituido pelos nomes dos integrantes do trio que desenvolveu o trabalho. O container deverá ser copiado para [este endereço](https://drive.google.com/drive/folders/1YW3nne-UIJUGSsVpmykPvDlRS-GP8I1m?usp=sharing) no Google Drive até às 23:59 do dia 15/05/2023. No dia seguinte, passarei a lista dos containers que recebi para controle.

6. Avaliação

Execução: Execução correta: 30%; Saída legível: 15%; Estilo de programação: Código bem estruturado: 15%; Documentação: Código legível: 15%, Descrição das estruturas de dados e principais decisões: 25%

7. Comentários Gerais

Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar. Seja ético, desenvolva seu trabalho, não copie de outras equipes e nem da Internet.

O monitor e o professor poderão pedir defesa do trabalho apresentado, sendo que somente um dos alunos será chamado para defender o trabalho pela equipe. Assim, é importante que todos os alunos participem do desenvolvimento.

Pode ser uma boa oportunidade para que os alunos exercitem o desenvolvimento usando [programação por pares,](https://medium.com/@thiagoctr/pair-programming-programa%C3%A7%C3%A3o-em-par-5cd19a1e3b22) que além de ajudar na qualidade do código desenvolvido, contribui para o aprendizado de todos os membros da equipe. Alguns ambientes de desenvolvimento integrado tem bom suporte para [programação por pares remota](https://ibrahimbutt.medium.com/how-to-set-up-vscode-for-remote-pair-programming-e5f5c30a48e2).