

Trabalho Final de CPD

Turma A - Semestre 2020/II

1 Visão Geral

O objetivo deste trabalho consiste em fornecer aos alunos a oportunidade de se aprofundar nos métodos, estruturas e algoritmos estudados na disciplina, em especial com o armazenamento e a manipulação de dados em arquivos com dados salvos no formato binário (i.e., estruturado), além de praticar a indexação, a pesquisa e a ordenação de dados. O trabalho permitirá ainda com que os alunos demonstrem que aprenderam as características e princípios relacionados com os diferentes conteúdos estudados ao longo do semestre, confirmando a capacidade de aplicá-los e analisá-los em problemas reais.

O trabalho deve ser realizado de maneira individual ou em duplas, e envolve escolher ou especificar um problema a ser resolvido por uma aplicação (um programa) que irá desenvolver/implementar. Qualquer problema pode ser escolhido, desde que envolva (i) a coleta (ou extração) de dados brutos, (ii) a sua preparação (conversão, formatação, padronização), (iii) o seu armazenamento em estruturas de arquivo (indexadas e em formato binário), (iv) a sua pesquisa e a sua (v) ordenação.

2 Especificação do Problema

Cada pessoa ou dupla deve especificar um problema a ser resolvido, sendo que a manipulação de dados é fundamental à resolução desse problema, cabendo aos alunos demonstrar como os métodos, estruturas e algoritmos de classificação, pesquisa e organização de arquivos podem ser aplicados na solução do problema, destacando as características que auxiliam (ou não) na sua resolução. Todos são livres para escolher a aplicação que acharem mais conveniente de desenvolver, desde que ela atenda aos critérios mínimos (ver seção 3). Caso não tenha alguma ideia, selecione um problema da lista de sugestões, disponível no Moodle.

3 Critérios Mínimos

Independente do problema escolhido, o trabalho deve:

1. ser uma aplicação do tipo *desktop* que executa localmente (i.e., não pode ser uma aplicação Web) ou uma App para celular *Smartphone*;

2. **extrair ou coletar, limpar e padronizar dados brutos provenientes da Web** (via API ou direto de uma página) **ou a partir de arquivos locais** (do tipo TXT, CSV, XML ou similares), os quais serão a fonte de dados inicial;
3. após coletar ou extrair os dados, o programa deverá **usar arquivos binários (estruturados) próprios, independentes dos arquivos originais**, para armazenamento e persistência dos mesmos. Tais arquivos devem ser **de um dos seguintes tipos: sequencial ou serial**. Devem ser implementados **com índices de acesso** que auxiliem na consulta a esses dados, tais como árvore B/B+ ou árvore TRIE;
4. **realizar processamento e armazenamento incremental**, i.e., os dados armazenados devem ser persistidos nos arquivos de maneira a poderem ser reutilizados sem que precisem ser extraídos ou coletados novamente toda a vez que o programa for carregado. O programa também deve permitir ao usuário acrescentar, atualizar e remover dados carregados anteriormente;
5. **permitir ao usuário classificar (ordenar) os dados (dos arquivos locais) em ordem normal e inversa**;
6. **permitir com que o usuário realize a busca ou localização (rápida) das informações extraídas e armazenadas nos arquivos**.

Em termos de programação, **você é livre para utilizar a linguagem que lhe for mais conveniente**. Também **pode usar alguma biblioteca auxiliar**, desde que referenciada/citada e justificada em seu relatório final.

Em seu relatório final, você deve explicar e detalhar o papel de toda estrutura de dados utilizada para resolução de seu problema.

O QUE NÃO PODE SER FEITO: (i) **não é permitido utilizar softwares de banco de dados ou similares** (você deve implementar suas próprias estruturas de arquivo e índice); (ii) **os dados não podem ficar em *arrays*, listas, coleções e estruturas semelhantes em memória principal, mas sim usando os recursos de pesquisa associados a seus arquivos de dados**.

4 Etapas, pesos e prazos

O trabalho será dividido em várias etapas, a saber:

1. **Definição dos problemas e conjuntos de dados (peso: 5%)**. Uma vez que o indivíduo ou dupla tenha escolhido um problema, deverá informar no Moodle (via ferramenta disponibilizada pelo professor) um nome de equipe (mesmo que individual), o(s) integrante(s) da equipe, o conjunto de dados que irá trabalhar e uma breve descrição do problema e do aplicativo que pretende desenvolver.
Prazo: 11/04/2021 (via tarefa específica - wiki - no Moodle).
2. **Definição dos arquivos da aplicação e detalhamento do problema (peso: 20%)**. Após a etapa anterior (e assim que tivermos estudado os principais tipos de arquivo e como modelá-los), as equipes terão a oportunidade de redefinir e refinar a descrição de seus problemas e aplicações. Além disso, devem entregar um modelo

ER (o qual será explicado pelo professor ao longo do semestre) descrevendo todas as entidades e atributos que suas aplicações irão manipular.

Prazo: 18/04/2021 (via tarefa específica no Moodle).

3. **Desenvolvimento da aplicação.** O desenvolvimento iniciará o quanto antes, e todos os membros das equipes devem participar. Sugere-se fortemente que os alunos utilizem as plataformas GitHub ou GitBucket para gerenciar a implementação, seja individual ou em duplas, gerenciar versões e registrar a contribuição de todos. As equipes devem adicionar o professor como membro da equipe, no GitHub, para que esse possa acompanhar o desenvolvimento.

Prazo: junto com a etapa seguinte (mas usando continuamente a ferramenta GitHub).

4. **Entrega final (peso: 40%).** A entrega final do trabalho será realizada através de um arquivo compactado (em formato .gz, .7z ou .zip) contendo o(s) Código(s)-fonte(s) e Makefiles (se necessário), com um arquivo texto explicando como compilar/lançar o código-fonte e uma breve explicação da aplicação que foi desenvolvida.

Prazo: 11/05/2019 (via tarefa específica no Moodle).

5. **Apresentação ao professor (peso: 35%).** No final do semestre, em datas definidas no cronograma da disciplina disponibilizado na plataforma Moodle, os resultados dos trabalhos serão apresentados ao professor. A apresentação será por equipe (e somente os membros da equipe e o professor participarão da atividade). O objetivo é demonstrar e explicar a aplicação implementada ao professor. No dia da apresentação, o qual pode ser combinado com o professor dentro do prazo estabelecido no cronograma, a equipe mostrará o software em execução e demonstrará suas funcionalidades. Também mostrará e explicará o código-fonte. O professor observará o funcionamento do aplicativo, analisará o código e fará perguntas de sua escolha, relacionadas com as funcionalidades básicas, para verificar o funcionamento e implementação do programa. O não funcionamento do programa ou de uma de suas funcionalidades implicará em nota nula naquele quesito. Todos os membros da equipe devem participar da apresentação (o aluno que não participar, não terá sua nota de trabalho integralizada na média final).

Os prazos de cada etapa serão rigorosamente observados. Qualquer atraso implicará em perda de nota (5% a cada dia).

5 Boas Práticas

Sugere-se uma lista de boas práticas para a execução deste trabalho. Elas são opcionais. Caso o grupo tenha interesse em utilizá-las, o professor pode auxiliar no seu uso durante o semestre.

Trello - para manter o registro de cada atividade e seus responsáveis.

Editor online de documentos - para que todos possam criar e editar de maneira colaborativa o relatório final (sugere-se Overleaf, mas você pode usar qualquer outro como o Google Docs ou Office online).

Github ou Bitbucket - para gerenciar o desenvolvimento em grupo e manter um repositório único de código, permitindo não só gerenciar versões, mas também

controlar a contribuição de cada participante.

Máquina Virtual ou Docker - para que você possa configurar todas as bibliotecas, plug-ins e componentes necessários para o desenvolvimento e a execução de seu software.

6 Avaliação

A avaliação seguirá os seguintes critérios: desenvolvimento das funcionalidades obrigatórias, aplicação dos conceitos e estruturas de programação estudados, atendimento aos critérios mínimos do enunciado, completude e correção da implementação. Outros aspectos de avaliação poderão ser incluídos a critério do professor. O peso do trabalho prático na composição da nota final corresponde ao valor especificado no plano da disciplina disponível na plataforma de apoio pedagógico.

7 Dúvidas

Em caso de dúvidas, não hesite em consultar o professor.