

Especificação do Trabalho Final de CPD

2016/1

April 27, 2016

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 (binários)**, a **indexação**, a **pesquisa** e a **classificação (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, demonstrando capacidade de aplicá-los e analisá-los em problemas reais.

1 Introdução

Cada grupo, formado de **três pessoas**, deve escolher ou especificar um problema a ser resolvido por uma aplicação (um programa) que será desenvolvida pelo grupo. Qualquer problema pode ser escolhido, desde que envolva a **coleta (ou extração)** de dados brutos, o **armazenamento em estruturas de arquivo (indexadas)**, a **pesquisa** e a **classificação (ordenação)** de dados.

2 Especificação do Problema

Cada grupo deve **especificar um problema a ser resolvido**, sendo que a manipulação de dados é central à resolução deste problema. O objetivo é 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 resolução do problema. **Os grupos são livres para escolher a aplicação que acharem mais conveniente de desenvolver, desde que ela atenda aos critérios mínimos** (ver seção 3). Lembre-se de que programa a ser desenvolvido para solucionar o problema deverá obrigatoriamente:

- (i) **criar e manipular arquivos**, i.e., índices e estruturas de de arquivo, os quais o programa deverá utilizar para fazer a persistência das informações. Esses arquivos devem ser **binários** (i.e. formato não textual);
- (ii) **oferecer a busca (localização)** de dados por algum critério;
- (iii) **envolver a classificação ou ordenação (em ordem normal e inversa)** de dados na resolução do problema.

Caso não tenha alguma ideia, selecione um problema da lista de sugestões, apresentada a seguir.

2.1 Sugestão de Problemas

Sugerem-se os seguintes problemas:

2.1.1 Catalogador de Currículos Lattes de pesquisadores

O objetivo deste trabalho é desenvolver um programa que permita indexar, classificar e consultar informações sobre os pesquisadores catalogados. O CV Lattes é um site do CNPq que armazena informações sobre os pesquisadores do Brasil (<http://lattes.cnpq.br>), incluindo suas publicações. A ideia geral é desenvolver um programa capaz de indexar os pesquisadores, incluindo seus dados pessoais e publicações, catalogando-os em arquivos, permitindo a classificação, ordenação e consulta por nome, título de publicação, tipo de publicação e ano (entre outras).

2.1.2 indexador e catalogador de mensagens ou posts em redes sociais

Implementar um programa capaz de indexar as mensagens postadas em uma rede social (Twitter, Facebook, Steam ou similares), incluindo conteúdo, autor e outras informações, catalogando-as em arquivo e permitindo sua classificação, ordenação e consulta (por conteúdo, hashtag, autor, data de postagem, entre outras).

2.1.3 indexador e catalogador de dados abertos em algum tema específico

Considerando dados abertos existentes (e.g. sobre políticas governamentais do município, estado ou união; atuação de políticos nas diferentes esferas do legislativo; trânsito e meios de transporte; dados de segurança pública como assaltos ou roubos, etc), implementar um programa capaz de indexar e catalogar as entidades sendo descritas por algum critério (e.g. nome), permitindo sua classificação, ordenação e consulta por suas diferentes propriedades.

3 Critérios Mínimos

Independente da opção escolhida, **o trabalho deve:**

1. executar localmente (não pode ser uma aplicação Web) ou em um Smartphone;
2. processar (i.e., extrair, coletar) dados brutos provenientes da Web ou a partir de arquivos locais, do tipo TXT, HTML ou XML, os quais serão a fonte de dados inicial;
3. após coletar ou extrair os dados, o programa deverá usar arquivos binários próprios, independentes dos arquivos originais. Tais arquivos devem ser de um dos seguintes tipos: sequencial ou serial. Devem ser implementados índices de acesso que auxiliem na consulta a esses dados, tais como árvore B/B+ ou árvore TRIE;

4. realizar a coleta e a persistência de dados de maneira incremental, i.e., ao terminar, o programa deve manter os dados no(s) arquivo(s) e, ao carregar, o programa deve ler os dados processados anteriormente. Se novos dados forem processados, eles devem ser adicionados aos já existentes;
5. permitir ao usuário classificar (ordenar) os dados (dos arquivos locais) em ordem normal e inversa;
6. permitir a busca de informações (dos arquivos locais) por algum critério.

Em termos de linguagem 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.

O QUE NÃO PODE SER FEITO: não é permitido utilizar softwares de banco de dados ou similares (você deve implementar suas próprias estruturas de arquivo e índice). O acesso aos dados não deve ser feito majoritariamente a partir de arrays, listas, coleções e estruturas semelhantes em principal, mas sim usando os recursos de pesquisa associados a seus arquivos de dados. Em seu relatório final, você deve explicar e detalhar o papel de toda estrutura de dados utilizada para resolução de seu problema.

4 Definição dos grupos

Uma vez que um grupo tenha sido formado e os participantes tenham escolhido um problema e definido a linguagem que o grupo irá utilizar, um dos participantes deverá informar no Moodle (via ferramenta disponibilizada pelo professor) um nome de grupo, a lista de pessoas que fazem parte do grupo, a linguagem escolhida e o problema a ser abordado. **A realização dos trabalhos que não sejam em trios devem ser discutidas e autorizadas pelo professor.**

5 Relatório

O grupo deve apresentar um relatório técnico com os itens descritos abaixo. Sugere-se que este relatório seja escrito utilizando recursos do L^AT_EX, mas se preferir utilizar outro software (Word, OpenOffice), fique a vontade. Estimula-se o uso de modelos e formatos próprios para a elaboração de relatórios acadêmicos. Segue a lista dos itens obrigatórios.

1. **Capa.** Com identificação do grupo, dos alunos, título do problema (ou nome do programa desenvolvido), e data.
2. **Problema.** Apresentação do problema solucionado, descrevendo quais são os requisitos e as funcionalidades implementadas.
3. **Implementação.** Detalhamento das implementações, focando na descrição de estruturas de dados e algoritmos desenvolvidos para atender aos critérios mínimos. Descrever como tais elementos foram usados para resolver o problema escolhido. Descreva, ainda, as vantagens ou desvantagens, dificuldades ou facilidades que o seu uso trouxe para a solução do problema. Caso utilize algum algoritmo, framework, módulo, classe, componente, biblioteca, etc., de terceiros, sua respectiva fonte/origem deve ser citada;

4. **Guia de Uso.** Guia ou manual do usuário da aplicação/solução desenvolvida, descrevendo características, exemplos de uso e resultados obtidos.
5. **Contribuição.** Descrição da contribuição do trabalho, ou seja, diferencial em relação ao que já existe, tanto em termos de aplicação (outros softwares que façam algo similar) quanto em termos de estruturas de dados e de código (no caso de reuso de código/componentes, descrever em detalhes como foi feito o reuso e onde está o diferencial – sua contribuição).
6. **Considerações Finais.** Descrever se atingiu ou não o objetivo (se resolveu ou não o problema). Descrever limitações do programa desenvolvido. Descrever quais técnicas ou tecnologias estudadas na disciplina foram empregadas e facilitaram ou dificultaram o desenvolvimento das soluções. Relatar as facilidades e dificuldades encontradas durante o desenvolvimento.
7. **Referências.** (livros, artigos, páginas na Internet, ...). As referências devem seguir algum formato normatizado.

6 Boas Práticas de Implementação

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

Git para manter o histórico de modificações, por exemplo, utilizando o serviço do GitHub ou Bitbucket.

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 ShareLatex ou Overleaf).

7 Forma da Entrega

A entrega do trabalho é realizada através de um arquivo compactado (em formato .tar.gz ou .zip) contendo:

- O relatório técnico do trabalho em formato PDF e contendo os itens descritos na seção 5;
- Código(s)-fonte(s) e Makefiles (se necessário), com um arquivo texto explicando como compilar/lançar o código-fonte.

Tal arquivo compactado deverá ser encaminhado até a data estipulada pelo professor e indicada em atividade específica de entrega via Moodle.

8 Apresentação

Os resultados dos trabalhos serão apresentados ao professor, demonstrando e explicando a aplicação implementada, em datas definidas no cronograma da disciplina disponibilizado na plataforma Moodle. No dia da apresentação 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 implicará em nota nula.

9 Prazos

Os seguintes prazos serão rigorosamente observados e qualquer atraso implicará em perda de nota (20% no 1o dia e 10% a cada dia subsequente):

- **Definição do trabalho:** até o dia 13 de maio de 2016 (até as 23:55 horas do servidor da UFRGS)
- **Relatório Final:** definido em cada turma, de acordo com o calendário próprio de aulas. Os alunos devem confirmar através das tarefas abertas para este fim no Moodle.

10 Avaliação

A avaliação do trabalho será dividida em definição do trabalho (10%), e trabalho final, dividido em relatório (40%) e apresentação da implementação (50%). A avaliação seguirá os seguintes critérios: desenvolvimento e detalhamento dos itens do relatório, aplicação dos conceitos de programação estudados, atendimento aos critérios mínimos do enunciado, formatação e estilo do texto, completude e correção da implementação. Outros aspectos de avaliação poderão ser incluídos a critério de cada professor.

O peso do trabalho trabalho prático na composição da nota final corresponde ao valor especificado no plano da disciplina disponível na plataforma de apoio pedagógico.

Atenção: conforme instruções presentes no plano de ensino da disciplina, todas as etapas do trabalho devem ser cumpridas para que a sua nota de trabalho seja contabilizada!

11 Dúvidas

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