

## Descrição do Projeto Final

Prof.: João Luis Garcia Rosa  
email: joaoluis@icmc.usp.br

PAE: Fabiano Berardo de Sousa  
email: fabber@icmc.usp.br

Monitor: Tales Prates Correia  
email: tales.correia@usp.br

### 1. Descrição

Desenvolver um programa em qualquer linguagem desejada que implementa um dos temas aqui propostos (ver Seção 4). O projeto deve ser desenvolvido em grupos de 6 pessoas. Obrigatoriamente, deve-se usar: algoritmos de ordenação, algoritmos de busca (sequencial, binária ou em árvore) e *hashing*. A ordenação, em alguns casos, deverá ser *multinível* (ou seja, ordenação considerando mais de um item). Os dados a serem manipulados pelo programa devem ser armazenados em e recuperados de arquivos, simulando uma base de dados real. Neste sentido, devem ser implementadas operações como *adicionar*, *remover* e *atualizar* dados nos arquivos.

Cada grupo deve eleger um **Teamleader**, o qual será a pessoa diretamente responsável pelo desenvolvimento do projeto. Caberá ao *Teamleader*: (i) coordenar o andamento do projeto (isto é: dividir tarefas entre os membros do grupo, promover reuniões periódicas, estipular prazos, verificar a qualidade dos códigos implementados, etc), (ii) apresentar (ou escolher alguém para apresentar) o trabalho oralmente em sala de aula (recomenda-se fortemente que o *Teamleader* faça a apresentação), (iii) avaliar os demais membros do grupo (no relatório a ser entregue).

### 2. Produtos

Cada grupo deve: (i) submeter o código fonte (zipado) para o Escaninho do Tidia; (ii) submeter um relatório (arquivo PDF) para o Escaninho do Tidia. **Ambos – código fonte e relatório, devem ser submetidos até o dia 01/12/2014;** (iii) elaborar uma apresentação oral de 15 minutos sobre o projeto. As apresentações estão marcadas para dias **02/12/2014** e **04/12/2014**. Seguem maiores informações sobre cada um destes itens.

#### 2.1. Código fonte

O programa pode ser desenvolvido em qualquer linguagem de programação que o grupo achar conveniente (incluindo linguagens para desenvolvimento web). Se for escolhida a linguagem C, o programa deve ser feito em mais de um arquivo, com divisão em arquivos de implementação de funções (arquivos .c) e arquivos de cabeçalho (arquivos .h).

## 2.2. Relatório

O relatório (arquivo PDF) deve conter as seguintes informações:

- nomes dos membros do grupo;
- *Teamleader* responsável pelo grupo;
- tema escolhido;
- descrição do funcionamento do sistema (incluindo *screenshots* das telas de saída);
- descrição dos algoritmos de ordenação e busca utilizados (incluindo *hashing*);
- discussão sobre a razão de tais algoritmos terem sido escolhidos \* (item de maior peso);
- dificuldades de implementação;
- funcionalidades a serem ainda implementadas (a fazer);
- como foi dividida a equipe (alocação de tarefas);
- avaliação dos membros do grupo feita pelo *Teamleader* (nota de 0 a 10 com justificativa).

Além disso, cada membro do grupo, se não for o *Teamleader*, deve, individualmente, postar no Escaninho do Tidia, uma avaliação sobre o *Teamleader* de seu grupo (arquivo .txt com o nome do *Teamleader*, uma nota de 0 a 10 e a justificativa para tal nota).

## 2.3. Apresentação

A apresentação oral deve ser feita preferencialmente pelo *Teamleader* do grupo com duração de, no máximo, 15 minutos, incluindo, principalmente:

- nome dos membros do grupo;
- demonstração da execução do sistema (mostrar sistema funcionando);
- listagem e discussão sobre o uso dos algoritmos de ordenação e busca utilizados;
- divisão da equipe (quem fez o quê).

## 3. Critérios de Avaliação

Serão avaliados os seguintes itens:

- execução e aparência do programa; organização (modularização) e legibilidade do código fonte (33,3%);
- relatório: relevância das informações e criatividade na escolha dos algoritmos (33,3%);
- apresentação oral: relevância das informações apresentadas e tempo de apresentação (33,3%).

## 4. Temas de Projetos

Cada grupo pode escolher um dos temas abaixo, sendo que não pode haver grupos com o mesmo tema. O grupo também pode decidir por um tema não listado aqui, desde que, obrigatoriamente, o tema permita ordenação (multinível), busca por um item específico e *hashing*.

### 4.1. Sistema para postagem de mensagens (Twitter.com)

- Ideia básica: implementar um sistema que posta mensagens de diferentes autores.
- Tipos de dados a serem manipulados: cada autor pode conter informações como, por exemplo: *nickname*, nome completo, profissão, texto informativo, conjunto de mensagens postadas, etc; cada mensagem pode conter informações como, por exemplo: data da postagem, autor que a postou, texto propriamente dito, tipo de mensagem (*hashtags*), etc.
- Dicas para ordenação: ordenar autores alfabeticamente, ordenar autores por profissão, ordenar todas mensagens por data, ordenar mensagens por *hashtag* (tipo da mensagem); ordenar mensagens de um autor específico, etc.
- Dicas para busca: exibir todas as informações de um autor específico, exibir todas as informações de uma mensagem específica, exibir apenas o texto de uma mensagem específica, etc.

### 4.2. Sistema para compra de passagens aéreas (Decolar.com)

- Ideia básica: implementar um sistema que oferece passagens aéreas de diferentes companhias aéreas.
- Tipos de dados a serem manipulados: cada companhia aérea pode conter informações do tipo: nome, tempo de serviço, número de aviões, conjunto de voos disponíveis, etc; cada voo pode conter, por exemplo: preço da passagem, código da passagem, código da aeronave, data de embarque, horário de embarque, duração do voo, aeroporto de origem, aeroporto de destino, etc.
- Dicas para ordenação: ordenar companhias aéreas alfabeticamente, ordenar companhias aéreas por número de aviões, ordenar voos por data e horário, ordenar voos por preço da passagem, etc.
- Dicas para busca: exibir detalhes de uma companhia aérea, exibir detalhes de um voo / passagem, etc.

#### **4.3. Sistema para reserva de quartos de hotel (Booking.com)**

- Ideia básica: implementar um sistema que oferece quartos de hotéis em diferentes locais / cidades.
- Tipos de dados a serem manipulados: cada hotel cadastrado pode conter informações do tipo: nome, endereço, quantidade de estrelas, itens oferecidos (wifi, TV, lavanderia, café da manhã, etc), conjunto de quartos, avaliação feita pelos usuários do sistema, etc; cada reserva pode conter informações do tipo: data da reserva, preço da diária, quantidade de pessoas, tipo do quarto (cama de solteiro, cama de casal, etc), quantidade de camas disponíveis no quarto, etc.
- Dicas para ordenação: ordenar hotéis por estrelas, ordenar hotéis por avaliação, ordenar hotéis por preço de um determinado tipo de quarto, ordenar hotéis por data desejada da reserva, etc.
- Dicas para busca: exibir detalhes de um hotel, exibir detalhes de quarto, exibir informações de uma reserva já realizada, etc.

#### **4.4. Sistema para venda de produtos (Amazon.com)**

- Ideia básica: implementar um sistema que oferece diferentes tipos de produtos para clientes cadastrados.
- Tipos de dados a serem manipulados: cada produto pode conter informações do tipo: id, nome, preço, tipo, quantidade em estoque, etc; cada cliente pode conter informações do tipo: *nickname*, nome completo, e-mail, endereço, conjunto de produtos já comprados, etc.
- Dicas para ordenação: ordenar clientes por ordem alfabeticamente, ordenar produtos por tipo, ordenar produtos por preço, ordenar produtos já comprados por um cliente, pela data da compra, etc.
- Dicas para busca: exibir detalhes de um cliente, exibir detalhes de um produto, etc.

#### **4.5. Sistema para classificados de serviços (Lecto.com.br)**

- Ideia básica: implementar um sistema que lista diferentes estabelecimentos pelo tipo de serviço e localidade (cidade).
- Tipos de dados a serem manipulados: cada estabelecimento pode conter informações do tipo: nome, tipo (categoria), endereço, telefone, etc; cada categoria pode conter informações do tipo: id, nome, etc.

- Dicas para ordenação: ordenar categorias alfabeticamente, ordenar estabelecimentos por tipo e ordem alfabética, ordenar estabelecimentos por endereço, etc
- Dicas para busca: exibir detalhes de um estabelecimento, etc.

#### **4.6. Sistema para disponibilização de softwares (BaixaKi.com)**

- Ideia básica: implementar um sistema que oferece softwares de diferentes tipos.
- Tipos de dados a serem manipulados: cada software pode conter informações do tipo: nome, versão, tamanho, sistema operacional em que pode ser executado, licença, categoria, avaliação dos clientes do sistema, etc; cada categoria pode conter informações do tipo: id, nome, etc.
- Dicas para ordenação: ordenar categorias por ordem alfabética, ordenar softwares de certa categoria por tamanho, ordenar softwares pela nota de avaliação dos clientes do sistema, etc.
- Dicas para busca: exibir detalhes de um software, etc.

#### **4.7. Sistema para catálogo de filmes (IMDb.com)**

- Ideia básica: implementar um sistema que exibe informações sobre filmes e artistas.
- Tipos de dados a serem manipulados: cada filme pode conter informações do tipo: nome, gênero, duração, diretor, roteiristas, conjunto de atores principais, ano de lançamento, língua falada, nota dada pelos clientes do sistema, etc; cada artista pode conter informações do tipo: nome, nacionalidade, tipo (ator, atriz, diretor, roteirista, etc), conjunto de filmes em que participou, etc.
- Dicas para ordenação: ordenar filmes por gênero e nota dada pelos clientes do sistema, ordenar filmes de um artista específico por data de lançamento, etc.
- Dicas para busca: exibir detalhes de um artista, exibir detalhes de um filme, etc.

#### **4.8. Sistema para organização de campeonatos de futebol**

- Ideia básica: implementar um sistema que oferece informações sobre um campeonato de futebol.
- Tipos de dados a serem manipulados: cada clube pode conter informações do tipo: nome, ano de fundação, presidente, técnico, conjunto de jogadores, etc; cada jogador pode conter informações do tipo: nome, idade, posição, número da camisa, etc; cada partida pode conter informações: data, times, placar, etc.

- Dicas para ordenação: ordenar jogos por data, ordenar times por pontos ganhos, ordenar times por saldo de gols, ordenar alfabeticamente os jogadores de um time específico, etc.
- Dicas para busca: exibir detalhes de um clube, exibir detalhes de uma partida, etc.

#### **4.9. Sistema de revisão de artigos científicos**

- Ideia básica: implementar um sistema que coordena a distribuição de artigos científicos para revisores de uma determinada revista científica.
- Tipos de dados a serem manipulados: cada revisor pode conter informações do tipo: nome, instituto para o qual trabalha, área de pesquisa, lista de artigos já revisados, etc; cada artigo pode conter informações do tipo: id, nome, lista de autores, data de submissão, palavras chave, conjunto de revisões feitas pelos revisores (cada artigo é revisado por 2 revisores), etc.
- Dicas para ordenação: ordenar revisores por ordem alfabética, ordenar artigos por área (palavras chave), ordenar por ano artigos já revisados por um revisor específico, etc.
- Dicas para busca: exibir detalhes de um revisor, exibir detalhes de um artigo, etc.

#### **4.10. Outras ideias**

- Sistema para anúncio de produtos (MercadoLivre.com.br);
- Sistema para anúncio de promoções (GroupOn.com);
- Sistema para reclamação sobre produtos e empresas (ReclameAqui.com);
- Lista telefônica;
- ...

Use a imaginação!

Bom projeto a todos!

:-)