Disciplinas : *Algoritmos em Grafos / Engenharia de Requisitos / Banco de Dados*
Professores : *Eveline Alonso Veloso / Juliana Amaral / Rodrigo Baroni*

Regras Básicas:

- Esse projeto pode ser desenvolvido em grupos de até **cinco componentes**.
- Cada professor determinará a forma e o prazo de entrega de sua parte no trabalho.
- Cada professor determinará o valor do Trabalho Interdisciplinar em sua disciplina específica.
- Para os alunos que estão matriculados nas três disciplinas, os mesmos grupos devem ser mantidos.
- O aluno que estiver matriculado em uma ou duas disciplinas, das citadas acima, deve obedecer a entrega da parte do trabalho que lhe diz respeito.
- Cópias de trabalho, se existirem, serão encaminhadas ao Colegiado de Coordenação do Curso.

Enunciado Comum:

A malha aérea de uma região pode ser representada por um grafo, no qual os vértices são os aeroportos dessa região. Um grafo não-dirigido pode ser utilizado para modelar as diversas rotas; enquanto um grafo dirigido pode ser usado para representar os diversos voos. Uma rota possui vários voos. Por exemplo, há voos de Confins para Salvador na parte da manhã, da tarde e da noite.

I) Enunciado Específico da Disciplina Algoritmos em Grafos

Pretende-se, nesse projeto, desenvolver uma aplicação que receba como entrada um conjunto de dados acerca dessa malha aérea e responda vários tipos de perguntas detalhadas a seguir.

Tarefas:

- Criar uma estrutura de dados que seja capaz de suportar os dois grafos: o de rotas e o de voos, os quais partilham os vértices, que são os aeroportos da região modelada. O grafo que representa as rotas deve ter apenas uma aresta conectando cada par (origem, destino) de aeroportos atendidos; enquanto o grafo de voos deve ter apenas uma aresta conectando cada par ordenado (origem, destino) de aeroportos. Existem vários pesos associáveis às arestas: distância, duração do voo, horários dos voos (apenas para o grafo que representa os diversos voos).

- Dados uma origem e um destino, desenvolver um algoritmo que determine a **viagem com menor custo** em termos de: **número de conexões**, **distância total percorrida**, **tempo total de voo**, **duração total da viagem** (considerando-se que pode haver esperas nas conexões. Nesse caso, utilize o primeiro horário de voo possível).

- Desenvolver um algoritmo que determine se é possível, **para todos os aeroportos** da região, **a partir de um aeroporto atingir qualquer outro** (com ou sem escalas). Se isso não for possível, indique os conjuntos de aeroportos que,

separadamente, atendem essa condição. Se isso for possível, indique quais os aeroportos que, se ficassem fora de serviço (apenas um de cada vez), impediriam essa situação para o conjunto de aeroportos em operação restante.

- Suponhamos que seja preciso **chegar ao aeroporto B para uma reunião importante à hora H**. Desenvolver um algoritmo que determine o **último voo** em que se pode sair do aeroporto A sem chegar atrasado ao destino.

- Suponhamos que você pretenda montar uma empresa de carga aérea, com uma frota na qual cada aeronave fará **voos de ida e volta apenas entre dois aeroportos**. **Quantas aeronaves**, no mínimo, serão necessárias e **quais rotas serão efetivamente usadas**, se o objetivo é atender todos os aeroportos, com um **consumo total mínimo** (não é importante o tempo total que uma encomenda demorará para chegar ao destino, mas apenas garantir que há uma rota até esse destino).

Formato do Arquivo de Entrada:

Para a realização dos testes e avaliação do código desenvolvido, será fornecido um arquivo texto de entrada que apresentará, na primeira linha, o **número de aeroportos** da região. As linhas seguintes desse arquivo de entrada conterão, cada uma, as seguintes informações, no seguinte formato: **nome do aeroporto 1; nome do aeroporto 2; direção do voo; distância entre os aeroportos**, em quilômetros**; duração do voo; horários de partida dos voos**. Se o valor do parâmetro **direção do voo** for 1, esse voo é direcionado do aeroporto 1 para o aeroporto 2. Se o valor desse parâmetro for -1, o voo tem a direção contrária, sendo direcionado, portanto, do aeroporto 2 para o aeroporto 1.
Segue um exemplo de arquivo de entrada:

```
3
CONFINS; GUARULHOS; 1; 606; 1:16; 7:00; 9:00; 18:00
CONFINS; GUARULHOS; -1; 606; 1:16; 8:00; 12:00; 17:00; 21:00
CONFINS; SANTOS DUMONT; 1; 480; 1:10; 7:30
GUARULHOS; SANTOS DUMONT; 1; 421; 0:30; 8:00; 18:00
```

Assuma que existem voos para todas as rotas já autorizadas.

Seu grupo deve criar seus próprios arquivos de entrada para testes, mas eles devem seguir o formato especificado acima, pois será executado o código implementado com os arquivos de teste (nesse formato) durante a correção desse trabalho prático pela disciplina de algoritmos em grafos.

Código:

- Todo o código deve ser desenvolvido nas linguagens **C#**, **Java ou C++**.
- Os alunos devem **comentar todos os métodos** implementados.
- O código deverá ser desenvolvido observando-se o formato de entrada especificado.
- As estruturas de dados utilizadas devem ser alocadas dinamicamente e o código deve ser modularizado.
- Variáveis globais devem ser evitadas.
- Legibilidade e boas práticas de programação serão avaliadas.

Documentação:

Deve também ser entregue uma documentação do projeto, que não deve exceder dez páginas e deve conter pelo menos os seguintes itens:

- Uma introdução do problema em questão.
- Modelagem e solução proposta para os problemas apresentados. O algoritmo utilizado para a resolução de cada problema proposto deve ser explicado de forma clara, possivelmente através de pseudocódigo e esquemas ilustrativos.
- Uma breve conclusão do trabalho implementado.

II) Enunciado comum das Disciplinas Engenharia de Requisitos e Banco de Dados

Um aeroporto possui um código (ex: CNF), nome, endereço, capacidade em número de passageiros e data de inauguração. Em um aeroporto operam várias companhias aéreas. É necessário armazenar as informações de código da companhia, nome da companhia aérea e nome do programa de milhagem. Uma companhia aérea possui vários funcionários, porém um funcionário é de apenas uma companhia aérea. Para todos os funcionários é preciso armazenar a matrícula, nome, cargo e salário. O sistema deve controlar os cargos possíveis dos funcionários de uma companhia aérea que podem ser: aeromoça, piloto, atendente de balcão, entre outros cargos administrativos.

Várias aeronaves fazem parte de uma companhia aérea, porém uma aeronave pertence a uma única companhia. Deve-se registrar o número da aeronave, o modelo e o número de assentos de cada aeronave. Uma companhia disponibiliza vários rotas. Uma rota contém o aeroporto de origem e o aeroporto de destino. Uma mesma rota possui vários voos. Deve-se armazenar o aeroporto de origem, o aeroporto de destino, hora prevista de partida, hora prevista de chegada e o código de cada voo. Um voo pode ser realizado por uma única companhia aérea.

Um voo possui várias ocorrências e para cada ocorrência armazena-se a data, a hora real da partida e da chegada (Ex: Voo CNF-SSA de 01/08/2018, Voo CNF-SSA de 02/08/2018, etc). Cada ocorrência de um voo está associada a uma aeronave. Em cada ocorrência do voo trabalham vários funcionários (piloto, aeromoças) e um funcionário pode trabalhar em várias ocorrências de voo ao longo de um dia.

Uma empresa de alimentação presta serviços de comida de bordo para várias companhias aéreas, sendo que uma companhia aérea pode contratar várias empresas de alimentação. Para cada serviço prestado entre a companhia e a empresa de alimentação é preciso armazenar o tipo do cardápio e o preço do cardápio. O sistema registra o nome de cada empresa de alimentação.

III) Enunciado Específico da Disciplina Engenharia de Requisitos

O trabalho deve considerar que o cadastro das informações dos aeroportos é feito pela ANAC (Agência Nacional de Aviação Civil). Por sua vez, o cadastro dos voos e suas ocorrências bem como dos funcionários é feito pela companhia aérea.

O trabalho terá duas entregas com datas a serem definidas.
A 1ª. Entrega deve conter os seguintes itens:
- O grupo deve definir uma lista dos requisitos (sujeito + verbo no infinitivo).
- A tabela de Requisitos deve conter o nome do requisito, tipo (funcional, não-funcional) e prioridade (essencial, desejável, opcional). Considerando o contexto do trabalho, o grupo deve especificar 8 requisitos não-funcionais.

- O grupo deve construir o diagrama de casos de uso e redigir 3 casos de uso descritivos dos casos de uso de maior complexidade, obedecendo um modelo (*template*) a ser fornecido.
- Basta descrever um único caso de uso de cadastro.

A 2ª. Entrega deve conter os seguintes itens:
- Diagrama de Classes (UML) com pacotes.
- Artefatos da primeira entrega corrigidos conforme revisão realizada pela professora.

IV) Enunciado Específico da Disciplina Banco de Dados

O trabalho deve conter os seguintes itens:
- Diagrama de Entidades-Relacionamentos (D.E.R.) do sistema, identificando chave primária e estrangeira.
- Listagem de Arquivo .SQL contendo:
  - comandos SQL-DDL de criação das tabelas no *database* específico do grupo no laboratório. Nestes comandos deverão estar definidas as chaves primárias e estrangeiras das tabelas. Deve-se utilizar o recurso de *default* e "*check*" de domínio para alguns campos das tabelas.
  - comandos SQL-DDL de criação de índices para otimizar as consultas. Observe as junções e ORDER BY.
  - 2 comandos para criação de uma visão.
- Documento Word com 10 comandos SELECTs criativos definidos pelo grupo. Os SELECTs devem ser construídos de acordo com as prováveis necessidades de informação do usuário. Todo SELECT deve conter o enunciado do comando. O grupo deve incluir registros nas tabelas para efetuar os testes dos comandos. Os 10 comandos devem seguir as especificações abaixo:

a) junções de 3 ou mais tabelas, com ORDER BY
b) junções de 3 ou mais tabelas, com ORDER BY e filtros na cláusula WHERE
c) junção de 3 ou mais tabelas, usando os operadores LIKE e BETWEEN
d) junção de 3 ou mais tabelas, usando os operadores IN e IS NULL/IS NOT NULL
e) junção de 2 ou mais tabelas com GROUP BY, sem HAVING, usando uma função agregada qualquer (MIN, MAX, AVG, SUM, COUNT). Use ORDER BY
f) junção de 2 ou mais tabelas com GROUP BY e HAVING, usando uma função agregada qualquer (MIN, MAX, AVG, SUM, COUNT)
g) subselect sem correlação
h) subselect com correlação
i) subselect com EXISTS
j) junção de 2 ou mais tabelas com GROUP BY, HAVING e ALL