



UNICAMP – COLÉGIO TÉCNICO DE CAMPINAS

Departamento de Processamento de Dados

Estruturas de Dados 1 & Arquitetura Orientada a Serviços

3º Período / Mat&Ves

Segundo Projeto

Este projeto é um projeto que visa complementar a avaliação de DUAS disciplinas, a saber: Estruturas de Dados I e Arquitetura Orientada a Serviços. Ele poderá ser feito em grupos de até 3 alunos. Os alunos que fazem apenas uma das duas disciplinas devem se agrupar com alunos que fazem apenas a outra disciplina, preferencialmente, ou então com alunos que fazem ambas as disciplinas, o que seria mais indesejável.

Pede-se:

1. Crie as seguintes tabelas MANUAL e DIRETAMENTE no BD, usando exclusivamente SQL, sem nenhuma outra programação associada. Para criá-las, use os seguintes comandos:

```
CREATE TABLE Alunos (  
  RA NUMBER(5) PRIMARY KEY,  
  Nome NVARCHAR2(40))
```

```
CREATE TABLE Disciplinas (  
  Cod NUMBER(3) PRIMARY KEY,  
  Nome NVARCHAR2(40))
```

```
CREATE TABLE Matriculas (  
  RA NUMBER(5),  
  Cod NUMBER(3),  
  PRIMARY KEY (RA,Cod),  
  FOREIGN KEY (RA)  
  REFERENCES Alunos(RA),  
  FOREIGN KEY (Cod)  
  REFERENCES Disciplinas(cod))
```

```
CREATE TABLE Resultados (  
  RA NUMBER(5),  
  Cod NUMBER(3),  
  Nota NUMBER(4,1) NOT NULL,  
  Freq NUMBER(5,1) NOT NULL,  
  PRIMARY KEY (RA,Cod),  
  FOREIGN KEY (RA)  
  REFERENCES Alunos(RA),  
  FOREIGN KEY (Cod)  
  REFERENCES Disciplinas(cod))
```

2. Insira, MANUAL e DIRETAMENTE NO BD, usando exclusivamente SQL, sem nenhuma outra programação associada, ALGUNS alunos na tabela Alunos, ALGUMAS disciplinas na tabela Disciplinas e ALGUMAS matrículas na tabela Matriculas.

3. Implemente em NodeJS um WEB-Service acionável com o verbo **delete** pela rota **/matriculas/XXX/YYYYY** (sendo XXX o código de uma disciplina e YYYYYY o RA de um aluno), que, ao ser acionado, remova da tabela **Matriculas** a matrícula daquele aluno naquela disciplina.
4. Implemente em NodeJS um WEB-Service acionável com o verbo **post** pela rota **/resultados**, que, ao ser acionado, insira na tabela **Resultados** o RA do aluno, o código da disciplina, a nota e a frequência (fornecidos no corpo da requisição).
5. Implemente uma classe genérica que representa uma lista DUPLAMENTE ligada desordenada com, pelo menos métodos para inserir um elemento no final da lista, para recuperar o primeiro elemento da lista, para remover o primeiro elemento da lista e para verificar se a lista está vazia, além dos métodos obrigatórios; outros métodos que julgar úteis para a classe em questão poderão ser implementados sem problemas;
6. Implemente uma classe genérica chamada **Fila**, usando para isso a implementação de lista pedida no item 5 acima;
7. Implemente uma classe chamada **Resultado** com os atributos **ra** (para representar o RA de um aluno), **cod** (para representar o código de um aluno), **nota** (para representar a nota daquele aluno naquela disciplina) e **freq** (para representar a frequência daquele aluno naquela disciplina), bem como **getters**, **setters**, um construtor inicializa TODOS os atributos de uma só vez a partir de 4 parâmetros que lhe trarão os valores a serem utilizados na inicialização, além dos métodos obrigatórios que se fizerem necessários;
8. Implemente um programa que solicita que o usuário digite quantas vezes desejar o RA de um aluno, o código de uma disciplina, a nota (de 0 a 10) do aluno na disciplina e a frequência (de 0,0 a 1,0) do aluno na disciplina, sempre armazenando tais dados em uma instância da classe **Resultado** e sempre armazenando a instância em questão em uma **Fila<Resultado>**; quando o usuário indicar que não deseja digitar mais nada, recupere, uma a uma, todas as

instâncias de **Resultado** da fila e, com seus dados, acione o WEB-Service solicitado no item 3; em caso de fracasso, emita uma mensagem de erro, indicando o RA e o código de disciplina que não foi possível remover da tabela de matrículas; em caso de sucesso, acione o WEB-SERVICE solicitado no item 4 acima, mostrando na tela os dados de cada resultado recuperado da fila, bem como uma mensagem apropriada à resposta dada pelo último WEB-Service acionado.

Seu projeto deverá ser entregue, impreterivelmente, até as 23h59 do dia 02/junho/2021. Para ser considerado entregue, vocês deverão gravar vídeos curtos em formato MP4; nos vídeos todos os integrantes devem participar de forma equilibrada; eles (os vídeos) deverão mostrar amplamente o sistema funcionando, além de mostrar, com detalhes, o código implementado. Foram sugeridos pequenos vídeos, porque o Google Classroom não aceita vídeos muito grandes. Os vídeos entregues deverão ser numerados (video1, video2, etc) para eu saber em que ordem vocês querem que eu assista. Reservem o primeiro vídeo para apresentar o grupo, dizendo os nomes, os RAs. **NÃO PRECISA ENTREGAR O CÓDIGO e APENAS UM INTEGRANTE** deve postar os videos no Google Classroom.

Os vídeos podem ser reuniões gravadas no Meet ou similares, ou podem usar um programa ESPECÍFICO para gravar videos de telas, por exemplo, Active Presenter (da Atomi Systems); outras alternativas existem, por exemplo, o OBS.