

## **Projeto de Disciplina Banco de Dados I**

**RESUMO:** A proposta de trabalho é desenvolver um sistema que permita realizar avaliações e apresente relatórios e gráficos com consultas sobre esses dados.

**OBJETIVO:** adquirir conhecimentos práticos de integração de bancos de dados em aplicações web em camadas, seguindo boas práticas de desenvolvimento.

### **DETALHAMENTO:**

Sistemas de avaliação são utilizados em vários domínios, desde avaliação de aproveitamento de estudos e cursos a avaliação de desempenho profissional. Esse tipo de sistema envolve transações e consultas relativamente simples, mas amplamente usadas em sistemas diversos, portanto o seu desenvolvimento deve estimular a aquisição prática de conhecimentos sobre o desenvolvimento de sistemas baseados em bancos de dados.

O sistema deve possibilitar cadastrar questões e avaliações, onde uma avaliação é formada por dados descritivos (e.g., descrição, data, horário, etc.) uma seleção de questões cadastradas, sendo que uma questão pode ser usada em diferentes avaliações. As questões podem ser de diferentes tipos (e.g., múltipla escolha, texto livre, etc.). A cada avaliação deve ser possível associar os usuários que deverão fazê-la e registrar as suas respostas para cada questão. Devem ser disponibilizados relatórios relevantes considerando os dados armazenados, com dados agregados, tabelas e gráficos, como média, média comparativa com outros usuários, evolução temporal do desempenho nas avaliações, etc.

A natureza do sistema de avaliação a ser desenvolvido é uma escolha do grupo, devendo ter suas particularidades implementadas. Por exemplo, em um sistema de avaliação de disciplinas as avaliações devem ter notas atribuídas, portanto, é preciso que as questões tenham um valor e uma forma de registrar a nota obtida por cada aluno em cada questão. Já em um sistema de avaliação de desempenho profissional, por exemplo, de avaliação docente, é preciso manter o sigilo das respostas. No exemplo de avaliação docente, é preciso garantir que um aluno avalie

uma única vez cada disciplina em um ano/semestre, mas suas respostas não devem ser vinculadas ao seu usuário, mantendo o anonimato.

As transações devem fazer as modificações necessárias no banco de dados utilizando, necessariamente, um ou mais controllers e os objetos das camadas de model e de acesso a dados. Os relatórios devem utilizar gráficos, tabelas e apresentar informações relevantes. Espera-se que as consultas utilizadas para produzir os relatórios explorem diferentes recursos da linguagem SQL (junções internas e externas, agregações, agrupamentos, ranking, ordenações, subconsultas, filtros diversos, etc.). Os relatórios também fazendo uso de objetos da camada de acesso a dados, possivelmente, objetos específicos para a geração dos relatórios.

O escopo do sistema deve ser cuidadosamente planejado pelo grupo para que seja factível nos prazos indicados abaixo e não seja excessivamente simples ou excessivamente complexo.

### **RESTRIÇÕES:**

O trabalho pode ser feito em grupos de até 2 alunos.

A avaliação considerará códigos e commits em um repositório git criado para o trabalho e apresentações com demonstração do sistema desenvolvido até então. Os commits precisam ser distribuídos regularmente no tempo de desenvolvimento. Commits excessivamente concentrados ou com muito conteúdo em cada commit serão interpretados como problemas no desenvolvimento. É fundamental que exista um balanceamento igualitário no grupo, em termos de número de commits, conteúdo e complexidade dos commits e tempo e conteúdo na apresentação.

#### **Opção 1 (Java):**

O back-end do sistema deve ser feito em **Java/J2EE** e o front-end em **JavaScript** (sugestão: React) ou **JSP/XHTML**. Podem ser usados frameworks e bibliotecas para facilitar o desenvolvimento, com exceção de implementações da **JPA (Java Persistence API)**, como o Hibernate e similares.

#### **Opção 2 (Python):**

O back-end do sistema deve ser feito em **Python**, utilizando frameworks como **FastAPI**, **Flask** ou **Django/DRF**, e o front-end pode ser implementado em **JavaScript** (sugestão: React) ou com **templates renderizados no servidor** (Django Templates, Jinja2 com Flask). Podem ser usados frameworks e bibliotecas auxiliares, desde que as consultas SQL relevantes sejam explicitamente implementadas pelos alunos, não sendo permitido o uso de ferramentas que abstraiam completamente a persistência ou a geração automática de relatórios.

**Banco de Dados:**

O SGBD utilizado deve ser o **PostgreSQL**.

**ENTREGAS:**

As entregas serão por meio de avaliação dos códigos e commits em um repositório git criado para o trabalho e compartilhado com o professor, bem como uma apresentação demonstrando a parte do sistema, a cada entrega.

Deve ser dado acesso de leitura ao repositório git.

**A) Data: 13/10**

Valor: 2,0 pontos na nota do 1º bimestre

- Descrição do sistema, do fluxo de suas funcionalidades e dos relatórios que se almeja criar, com figuras da visualização esperada (dados, gráficos e/ou tabelas). É fundamental vislumbrar relatórios relevantes que explorem variados operadores SQL, lembrando que um único relatório pode envolver diversas consultas.
- Diagrama ER para acomodar os conjuntos de dados sem redundâncias, esquema relacional de banco de dados e script de implementação do esquema em SQL, com todas as restrições de integridade aplicáveis aos conjuntos de dados.

**B) Data: 03/11**

Valor: 2,0 pontos na nota do 2º bimestre

- Banco de dados funcional e com algumas cargas efetuadas.
- Sistema rodando com as funcionalidades de cadastro e realização de avaliações.

**C) Data: 01/12:**

Valor: 3,0 pontos na nota do 2º bimestre

Sistema finalizado, incluindo os relatórios.