

Análise e Desenvolvimento de Sistemas
Programação Orientada a Objetos II
Projeto da disciplina

Regras:

1. Deverá ser feito individualmente;
2. Entrega até o final da data de postagem do moodle;
3. O prazo para entrega não será alterado, a não ser que o servidor do Moodle apresente problemas técnicos inviabilizando o envio.
4. Faça a tarefa o quanto antes e envie para evitar problemas. Você poderá reenviar quantas vezes desejar durante o prazo. Somente a versão final que ficará disponível para correção.
5. Deverá ser apresentado para o professor em aula (a versão enviada no Moodle).

Descrição:

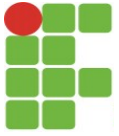
Você criar um programa que vai gerir cadastro de conteúdo para banco de dados. Para tanto você vai realizar esse programa em etapas.

Primeira etapa:

Nessa etapa será necessário definir o seu tema e entregar um programa que persista em arquivos os dados de uma entidade do seu problema.

Entregáveis:

- 1) *A ideia do que será implementado* → Aqui você deverá entregar a descrição da ideia do que será implementado (pode ser uma descrição textual) e o diagrama ER das entidades que serão mapeadas. Esse sistema deverá prever um BD com pelo menos duas entidades e ao menos um relacionamento 1:n ou n:n.
- 2) *Uma classe de modelo* → a implementação de uma classe de modelo do seu problema: correta definição dos atributos, construtores, getters e setters etc.
- 3) *Acesso ao BD:*
 1. O *schema* (comandos DDL que criam as tabelas e eventuais INSERT's) do seu banco de dados com a tabela que será utilizada. Considere a criação da tabela em um BD de nome **“projeto”** para que facilite a correção.
 2. Deve ser criado uma classe que será a Fábrica de conexões (isolando a criação de uma conexão com o BD): espera-se, por exemplo, uma conexão em localhost:5432, no bd:projeto, com user:postgres e senha: postgres.
 3. Interface DAO. A interface deve possuir os seguintes métodos (onde ObjetoModelo é um tipo genérico ou da sua classe de modelo):
 - boolean insert(ObjetoModelo)
 - boolean delete(int id)
 - boolean update(ObjetoModelo)
 - List<ObjetoModelo> list(int limit, int offset)
 - ObjetoModelo get(id)



4. Uma classe (DAO) que a implementa a sua interface DAO e que possibilita a persistência em uma das suas tabelas e mapeiam para os objetos da classe modelo implementada.
- 4) *Classe de teste (modelo) do item 2* → uma classe de testes que testa (no formato de assertivas) a classe de modelo e seus métodos.
- 5) *Classe de teste (persistência) do item 3.4* → uma classe que realize testes (no formato de assertivas) que validem todos os métodos do seu DAO.

Ideias (*brainstorm*): para ajudar a refletir, segue alguns exemplos de ideia que podem ser propostas (mas o tema é livre). Você pode também estender umas dessas ideias...

- Cadastro de *Livros* e *Autores* de livros (duas entidades com relacionamento n:n)
- Cadastro de *Fotos* e *Álbuns* (pode ser modelado como relacionamento n:n ou 1:n)
- Cadastro de *Filmes* e *Categorias* (relacionamento 1:n)
- Para um jogo, *Personagem* e *Poderes* (relacionamento n:n)
- Algum aspecto de uma rede social, por exemplo *Usuários* e *Postagens* (1:n) ou *Postagens* e *Comentários* (1:n)