

Trabalho Prático de Engenharia de Software

Descrição Geral

O trabalho da disciplina consiste em desenvolver um sistema com 2 entidades (por exemplo: Cliente, Venda, Locação, Produto, etc.).

Uma das entidades (chamada aqui de principal) deve possuir uma associação 1..* com a outra (por exemplo, um venda possui um ou mais produtos). A classe da entidade principal deve possuir ao menos um método que faz uso dos objetos da outra.

As classes das entidades devem estender a classe abstrata Entidade, que possui um atributo chamado id (inteiro), que serve como identificador e um método abstrato que retorna uma string com o nome e valor dos atributos. Este método deve ser implementado nas classes das entidades. A classe Entidade e as classes das entidades devem possuir métodos construtores, getters e setters e devem ficar em um pacote/camada chamado modelo.

O sistema deve possuir uma camada de persistência que grava os dados em arquivos de texto. Nela haver uma classe abstrata Persistencia, com métodos abstratos para a inserção, alteração, exclusão dos dados das entidades. Também deve haver um método abstrato de busca a partir do id e outro a partir de um valor string. A classe Persistencia deve ser estendida pelas duas classes das entidades e seus métodos sobrescritos. O método de busca a partir de um valor string deve ser aplicado a um atributo a escolha em cada entidade (use conversão de tipo caso necessário).

O código pode ser implementado em qualquer linguagem orientada a objetos.

TP1

Definição do sistema, modelagem e descrição dos casos de uso.

TP2

Criação do modelo de classes inicial seguindo a descrição geral.

TP3

Implementação da interface com o usuário e arquitetura MVC. O sistema deve permitir a inserção, alteração, exclusão dos dados das entidades. Também deve ser possível fazer a busca a partir do id e de um outro atributo a escolha. Deve ser possível realizar essas operações de forma independente para cada entidade. Além disso, para a entidade principal, deve ser possível adicionar e remover objetos da outra entidade por meio da busca por id ou nome de objetos pré-cadastrados.

Todas as classes concretas de persistência devem ser Singletons.

Todos os objetos das camadas do MVC devem ser criados com o uso do padrão Abstract Factory.

O modelo de classes deve ser atualizado para refletir o estado atual do sistema.

TP4

Modelo de comunicação (ou de sequência) e modelo de atividades referentes à inserção de um objeto da entidade principal com adição de objetos da outra entidade.

Modelo de componentes do software.

TP5

Cadastro do sistema no GitHub, com criação do banch principal (main ou master) e um branch para cada membro do grupo.

TP6

Definição de casos de teste das operações de inserção, exclusão e busca da entidade principal.

Cálculo de Pontos por Função do sistema e uma outra métrica a escolha.

Organização do Trabalho

Deve ser criado um quadro Kanban no Trello ou sistema similar que permite o acompanhamento do andamento do desenvolvimento do software. Devem existir as seguintes colunas no quadro Kanban:

1. **Backlog:** cada caso de uso do software deve ser registrado como um bloco de atividade no Kaban. Também devem ser criados dois blocos adicionais:
 - a. **Arquitetura:** descreve a estrutura do software. As informações solicitadas na Caixa de Texto 1 devem ficar em um arquivo próprio que deve ser armazenado em um drive compartilhado e o link de acesso deve ser anexado no bloco Arquitetura;
 - b. **Interface Geral:** corresponde à tela a partir da qual as funções do software podem ser acessadas pelo usuário.
2. **A Fazer:** deve ser criada uma cópia de todos os blocos do Backlog aqui. Os blocos desta coluna serão aqueles que se deslocarão para demais ao longo do processo de desenvolvimento do software.
3. **Design:** A descrição detalhada dos casos de uso deve ser realizada nesta etapa. Para cada caso de uso, deve ser criado um arquivo com a descrição dos casos de uso, que deve ser armazenado em um drive compartilhado e o link de acesso deve ser anexado no bloco do caso de uso correspondente. A descrição dos casos de uso deve conter as informações indicadas na Caixa de Texto 2.
4. **Codificação:** corresponde à etapa de implementação do código-fonte do software.
5. **Testes:** corresponde à etapa de testes do software.
6. **Concluído:** corresponde ao estado finalizado do bloco.

Caixa de Texto 1

Dados Gerais do Software

Equipe: nome completo dos alunos.

Nome do Projeto:

Descrição geral: até 5 linhas com ideia geral do que é o software, o que ele faz, onde ele funciona, etc.

Usuários do software: listar e descrever os perfis de usuário do software a ser desenvolvido.

Modelo de casos de uso:

Inserir aqui a figura do modelo de casos de uso.

Modelo de classes:

Inserir aqui a figura do modelo de classes.

Caixa de Texto 2

Identificação: código identificador.

Nome: Título do caso de uso (exatamente como no modelo).

Atores Primários: atores principais.

Atores secundários: outros atores interessados.

Pré-Condição: condição ou estado prévio para que o caso de uso possa ocorrer.

Pós-Condição: estado ou situação do software após o caso de uso ter ocorrido.

Fluxo de Execução: descrição passo a passo do que acontece.

Outras Ações ou Restrições: descrição passo a passo de fluxos alternativos e restrições que podem intervir.