

Instrumento de avaliação / exercícios / Atividades

Componente curricular: FPOO	Atividade: VPF02	Nota:
Unidade: SENAI Jaguariúna	Data: / /	
Aluno:	Nº:	Turma:

Situação Problema

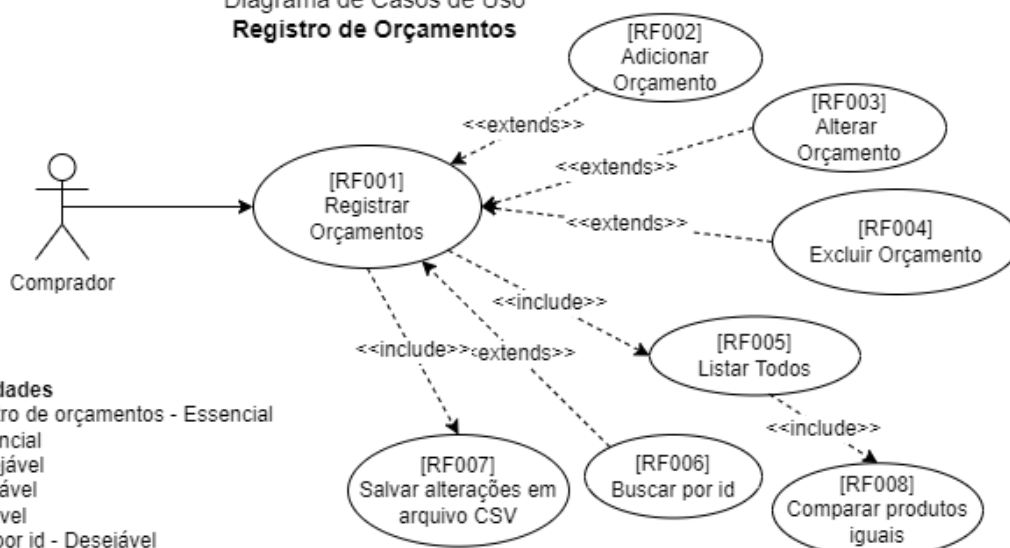
Contextualização: O setor de Compras da empresa XPTO Infraestrutura de Redes, precisa de um sistema Desktop para ser instalado no computador do comprador que registre todas os **orçamentos**. Atualmente o controle é feito através de planilhas.

Dados: O Analista de Sistemas junto ao comprador elencou os requisitos e elaborou os diagramas a seguir conforme UML.

Desafio: Desenvolver uma interface gráfica GUI, com as funcionalidades CRUD, para que o funcionário possa registrar seus orçamentos e que compare os produtos iguais e informe qual o mais barato, normalmente são realizados três orçamentos de cada produto, o sistema destaca o mais barato para que o comprador tome a decisão correta.

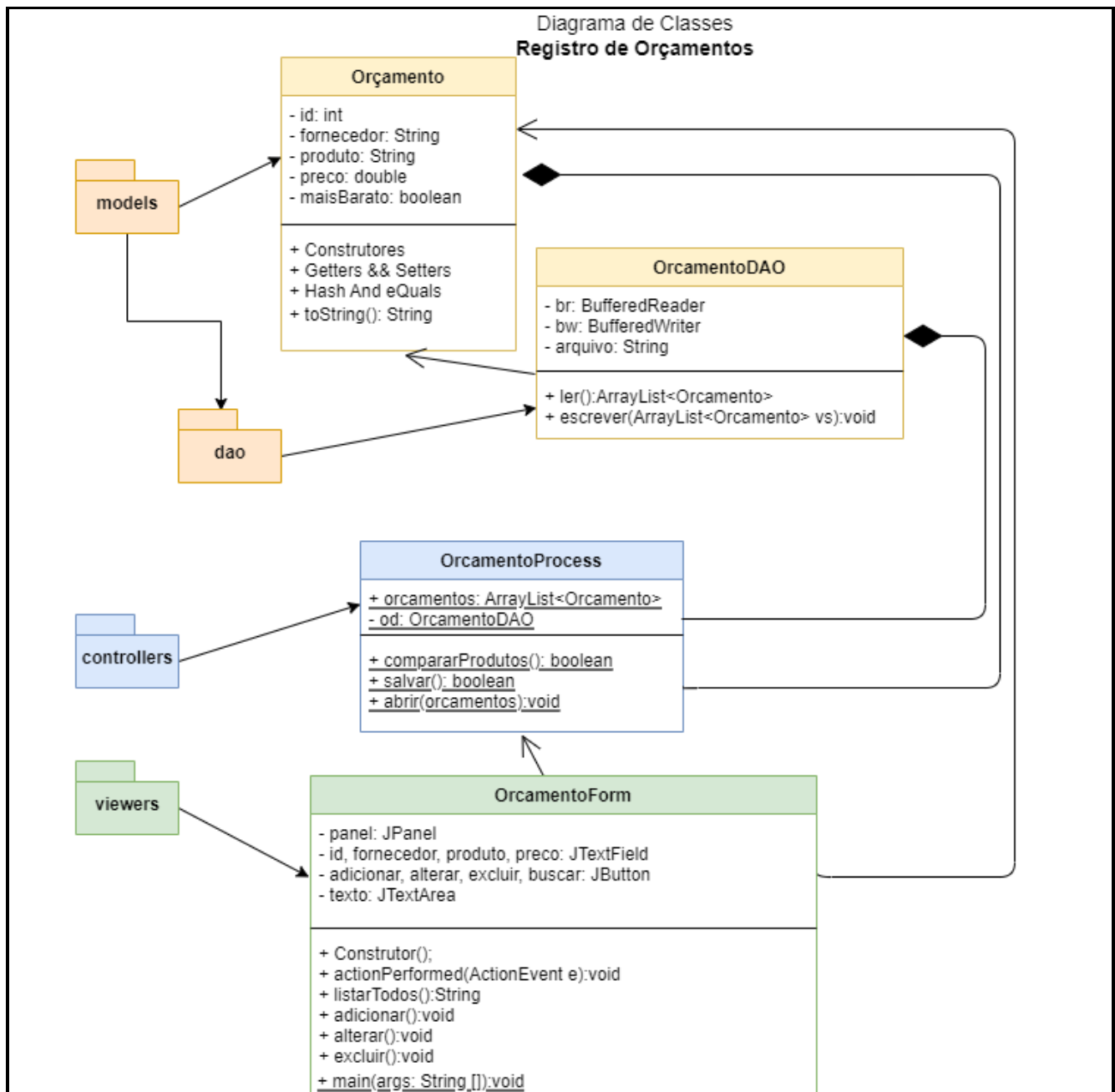
Entregas: Endereço do repositório github (xptoOrçamento) contendo apenas este projeto.

Diagrama de Casos de Uso
Registro de Orçamentos



Requisitos Descrição Prioridades

- [RF001] FormPrincipal Cadastro de orçamentos - Essencial
- [RF002] CRUD Create - Essencial
- [RF003] CRUD Update - Desejável
- [RF004] CRUD Delete - Desejável
- [RF005] CRUD Read - Desejável
- [RF006] CRUD Read Buscar por id - Desejável
- [RF007] DAO - Salvar em arquivo CSV (Persistência de dados) Essencial
- [RF008] Comparar produtos iguais e destacar o mais barato



FUNDAMENTOS TÉCNICOS E CIENTÍFICOS + CRITÉRIOS (v) Críticos (v) Desejáveis

- 2. Utilizar lógica de programação para a resolução de problemas (7)
(Resolver problemas Essenciais)
- 10. Manipular os diferentes tipos de dados na elaboração de programas
(Tratar conversões de String para Ponto flutuante, datas e outros)
- 11. Utilizar o ambiente integrado de desenvolvimento (IDE)
(Utilizar adequadamente as funcionalidades do Eclipse para a solução do problema)
- 12. Utilizar técnicas de versionamento através de softwares específicos
(Criar repositório no GitHub, postar o código fonte da solução)
- 13. Utilizar o paradigma da programação orientada a objetos
(Encapsulamento, Construtores, Getters && Setters, toString e/ou toVetor, toCSV: funcionais)
- 14. Elaborar diagramas de classe
(Interpretar os diagramas UML e programar a solução seguindo as especificações)

CAPACIDADES SOCIAIS, ORGANIZATIVAS E METODOLÓGICAS.

- 4. Demonstrar raciocínio lógico na organização das informações (24)
(Resolver problemas Essenciais - CRUD)
- 5. Demonstrar visão sistêmica (28)
(Resolver o problema apresentado “marcar como true automaticamente o produto mais barato”)
- 6. Seguir método de trabalho (21)
(Padrão de projetos MVC, DAO, GUI, funcionalidades CRUD)

CONHECIMENTOS

- | | |
|---|---|
| <ul style="list-style-type: none">3. Programação Orientada a Objetos3.1. Definição3.2. Pacotes3.3. Classes3.3.1. Abstrata3.3.2. Interna3.3.3. Anônima3.3.4. Atributos3.3.5. Métodos3.3.6. Modificadores de acesso
(encapsulamento)3.4. Objetos3.5. Interface3.6. Polimorfismo3.7. Enumerações3.8. Relacionamentos de objetos3.8.1. Herança | <ul style="list-style-type: none">3.8.2. Agregação3.8.3. Composição4. Ambiente de desenvolvimento4.1. Instalação e configuração4.2. Gerenciamento de dependências4.3. Recursos e interfaces5. Manipulação de arquivos5.1. Escrita5.2. Leitura6. Controle de versões6.1. Definição6.2. Repositório6.2.1. Inicialização6.2.2. Remoto6.2.3. Ramificações (branch)6.2.4. Versionamento |
|---|---|