**Documento de Visão de Projeto**



**Livraria**

**Histórico de Versões**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** |
| 01/05/23 | 1.0 | Modelagem e desenvolvimento | Kleberson | - |

|  |  |
| --- | --- |
| **Cliente** | FATEC - Interno |
| **Documento** | Documento de Visão de Projeto: *Livraria* |
| **Data** | 01 de maio de 2024 |
| **Autor** | **Kleberson dos santos silva** |

**Página de Assinaturas**

|  |  |  |
| --- | --- | --- |
| Revisado e Aprovado por: |  |  |
|  |  |  |

**Índice**

Camada de Apresentação 7

Camada de Persistência 11

Diagrama de caso de uso geral do sistema 5

Escopo 4

Objetivo 4

Pacote Model 10

Qualidade 13

Realização dos Casos de Uso Significativos 11

Representação Arquitetural 4

Tamanho e Performance 13

Visão de Dados 13

Visão de Implementação 13

1. **Objetivo**

Este projeto visa estabelecer uma livraria que harmonize os clássicos com as mais recentes obras literárias, abrangendo desde as primeiras edições até as mais recentes de renomadas obras.

* 1. **Escopo**

A Livraria é um refúgio para amantes da literatura, onde clássicos intemporais se encontram com as últimas obras dos grandes autores contemporâneos. Nossa missão é proporcionar uma experiência envolvente e inspiradora para os leitores, oferecendo uma cuidadosa seleção de obras que abrangem diversas épocas, gêneros e estilos literários. Buscamos ser reconhecidos como um destino de referência para os aficionados por livros, onde podem explorar novas perspectivas e redescobrir os tesouros da literatura em um ambiente acolhedor e culturalmente enriquecedor.

* 1. **Referências**

Para a construção deste documento foram utilizadas as seguintes referências:

* Reuniões informais entre o professor e o desenvolvedor.
* Documentos elaborados pela Engenharia De Software.

Este documento influencia os seguintes documentos:

* Documento de Requisitos
* WBS

1. **Necessidades de Negócio**

Um sistema informatizado para controle de uma livraria é necessário para que a instituição que utilizar esta solução consiga ter total controle de todos os livros que a mesma possui. Com o sistema, também é possível controlar trocas e devoluções de livros, além de cadastrar clientes. O sistema contará com administrador que conseguirá controlar o estoque e as vendas de forma invisual, além de controlar o envio e a aprovação dos livros.

1. **Objetivo do Projeto**

Exemplo 1: Desenvolver uma plataforma para soluções web capaz de:

* armazenar informações em uma base de dados
* utilizar o protocolo HTTP
* ser executado em qualquer navegador

Exemplo 2: Desenvolver uma solução para uma livraria, que possibilite:

* Controlar os livros presentes no estoque da livraria e entrada de novos livros.
* Gerenciamento das operações realizadas pela livraria: venda, troca de livros, controle de clientes, gestão de carrinho de compras.

Minimizando o tempo de atendimento da livraria aumentou a taxa de vendas através da internet e garantido assim o controle sobre o estoque, vendas e entregas de livros.

1. **Declaração Preliminar de Escopo**
   1. **Descrição**
   2. **Produtos a serem entregues**

Os seguintes itens são considerados produtos do projeto, na sua etapa 1.

* Sistema do módulo da livraria, etapa 1, implementado de acordo com a especificação feita na fase de análise. (Código objeto e código fonte).
* Documentos de especificação do sistema, concebido na fase de elaboração.
* Hospedagem do sistema em ambiente 24 x 7.
  1. **Requisitos**

É importante salientar que o documento de visão estará sendo feito no momento de iniciação do projeto. No qual, ainda não se tem todo o detalhamento. Portanto, os requisitos devem ser descritos em alto nível e depois detalhados em outro documento, caso o projeto seja aprovado.

* + 1. **Requisitos Funcionais**

O sistema deve ser capaz de efetuar o cadastro, exclusão, alteração e consulta de livros do estoque da livraria.

O sistema deve ser capaz de efetuar o controle do estoque.

O sistema deve ser capaz de fazer o controle da ativação e inativação dos livros

O sistema deve ser capaz de produzir uma listagem com os livros que serão vendidos.

O sistema deve ser capaz de cadastrar, excluir, alterar e consultar clientes.

O sistema deve ser capaz de controlar o estoque

* + 1. **Requisitos Não Funcionais**

Utilizar linguagem Java

Utilizar o banco de dados Mysql

A arquitetura da solução obedece ao design pattern MVC.

O sistema deve rodar nos seguintes browsers:

* IE
* Google Chrome
* Firefox
* Cronograma de Marcos Sumariado
  + 1. **Regras de Negócio**

1. **Premissas**

* O Front End do projeto pode ser a parte o mais difícil do projeto pois a falta de experiência pode pesar, além disso o Json pode influenciar as entregas por que a o envio de dados utilizando cadeias de objetos e caso não haja projetos anteriores isso influenciará as futuras entregas.
* Outro ponto de observação deve ser as entregas pois esse é o primeiro grande projeto que une Front End e Back end em um único projeto.
* O projeto será orientado pelo professor Rodrigo Rocha.

1. **Influência das Partes Interessadas**

* Rodrigo Rocha patrono do projeto e também o Gerente do projeto, todas as ações do projeto devem ser direcionadas para ele e os prazos cumpridos de acordo com as suas especificações além disso os requisitos que devem ser atendidos serão enviados via teams e também estarão disponíveis via documento de requisitos.
* Atenção: atenda as tarefas enviadas via teams em detrimento ao documento de requisitos.

1. **Representação Arquitetural**

Toda a arquitetura será baseada nos padrões de projetos tradicionais do GoF a arquitetura da aplicação seguirá um padrão de design modular, com separação clara das responsabilidades entre as diferentes camadas sendo executado em um servidor apache.

* Controller: Recebe os dados através do DTO (Data Transfer Object) para criar o objeto correspondente e, em seguida, invoca um serviço.
* Entidade: Utiliza padrões de Clean Code e SOLID, realizando a validação dos dados obrigatórios e seus respectivos tamanhos. Segue o padrão de Entidade Rica.
* Service: Responsável por validar as regras de negócio, incluindo a validação e persistência dos dados.
* Validator: Garante a conformidade dos dados com as regras de negócio estabelecidas, seguindo os princípios do SOLID.
* DAO (Data Access Object): Interage com o banco de dados para realizar operações de persistência de dados.

As Classes de Domínio são as classes que representam os Value Object, contendo os atributos e os métodos getters/setters além de alguns métodos.

Os casos de uso representam as classes responsáveis por aplicar as regras de negócio do sistema como, por exemplo, Cadastrar Cliente. Constarão também nas classes de negócio o relacionamento com os Daos responsáveis por recuperar os objetos no banco de dados.

A aplicação irá utilizar o Spring boot e o servidor será um Tomcat que irá subir a aplicação, para a interação entre o Front e o Back end o escolhido será o Javascript para o envio e recuperação dos dados

* 1. **Restrições Arquiteturais**

Foram identificadas algumas orientações / restrições pertinentes ao desenvolvimento deste subsistema:

* Utilização do Java 17;
* Utilização do TomCat;
* Framework Spring Boot
* Utilização do Html e Css para desenvolvimento da camada de apresentação além do Javascript para recuperar e enviar os dados;
* Utilização do framework Spring para desenvolvimento da camada de negócio e controle de transações;
* Utilização do framework de mapeamento objeto-relacional Hibernate;
* Utilização do SGBD Mysql.
  1. **Objetivos e Restrições Arquiteturais**

Alguns requisitos registrados que impactam diretamente a arquitetura do sistema são:

Considerando premissas definidas para os Sistemas pode-se citar as seguintes restrições:

* Utilização da Linguagem Java versão 17
* Considerar a utilização de software Livre, quando possível
* O Sistema de Gerenciamento de Banco de Dados a ser considerado em implementações de âmbito corporativo será o SGBD Mysql.

1. **Visão de Use Case**

Esta seção apresenta os Casos de Uso arquiteturalmente significativos, que foram selecionados considerando-se o pacote do Modelo de Casos de Uso que representa o sistema.

A classificação dos casos de uso, em termos de significância, foi realizada com base na observação de pelo menos um dos seguintes critérios:

* Casos de uso que estendem outros Casos de Uso
* Casos de Uso que são incluídos em outros Casos de Uso e
* Casos de uso que acessam sistemas externos
  1. **Diagrama de Caso de uso geral**

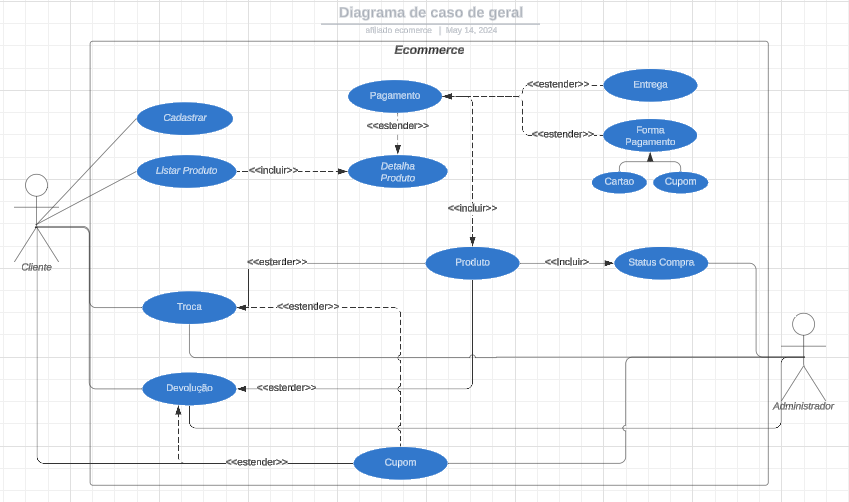


Figura 4.1 Diagrama de Caso de Uso de XXXXXXX

1. **Visão de Lógica**

Esta visão apresenta elementos de design significativos do ponto de vista da arquitetura, descrevendo a organização do Sistema em pacotes, bem como a organização desses pacotes em camadas.

O Diagrama com as camadas do sistema é ilustrado na figura 5.1.

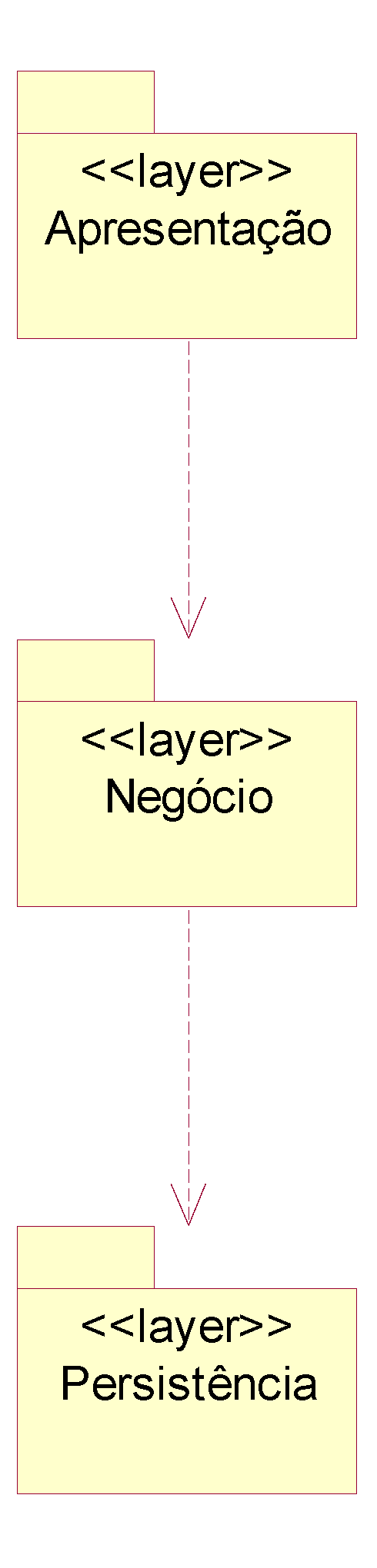


Figura 5.1 – Diagrama de camadas do XXXX

**Apresentação**: Contém classes para as interfaces gráficas com os usuários (GUI). Através destas interfaces os usuários conseguem interagir com o sistema, com o intuito de incluir, alterar e excluir produtos.

**Negócio**: Contém classes que controlam a execução das funcionalidades.

**Persistência**: Contém classes responsáveis por salvar as entidades de modelo. Por exemplo,

contém as classes que permitem ler e gravar os objetos no banco de dados relacional.

A Figura 5.2 ilustra o diagrama de camadas com as tecnologias utilizadas no desenvolvimento, já descritas na figura 5.1.

Neste momento, é importante ressaltar que a camada de apresentação envolve componentes que são executados na porção servidora e na porção cliente. Na porção servidora, são executados os componentes que montam as páginas html de resposta e controlam o fluxo de interação com o usuário. Assim, envolve interpretação de páginas, utilização de javascript para montagem das páginas de resposta, delegação de solicitação para a camada de negócio, obtenção de respostas da camada de negócio, dentre outros. Já na porção cliente, o javascript será executado nos navegadores dos usuários com o intuito de facilitar a interação. Por exemplo, podem fazer uma pré-verificação dos dados digitados.

* 1. **Camada de Apresentação**

Nesta camada, temos o pacote que contém todos os arquivos relacionados à exibição de informações para usuário, o que engloba páginas HTML, imagens, javascript, dentre outros. Já o pacote de controle desta camada, contém as Controllers responsáveis pela comunicação com as classes da camada de negócio. Estes pacotes podem ser vistos na Figura 5.3.

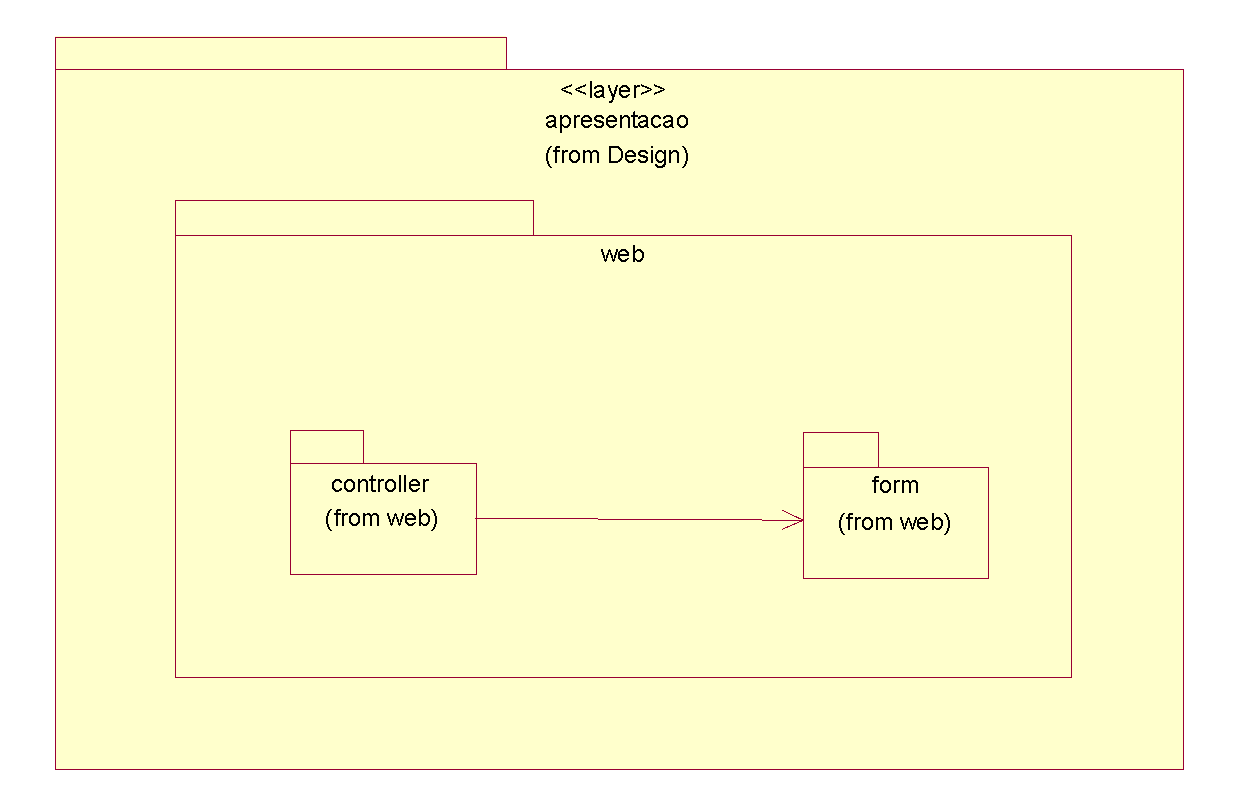


Figura 5.3: Camada de Apresentação

## Camada de Caso de Uso

Nesta camada, temos o pacote controle que contém as classes responsáveis por controlar as regras de negócio da aplicação. O pacote entity, contém as classes que representam o modelo, ou seja, aquelas que contém as informações sobre o Sistema. O pacote controller instância as entidades e invoca uma servisse que irá verificar a existência desta entidade e assim chamará uma lista de validações.

A imagem 4.4 ilustra os pacotes descritos.

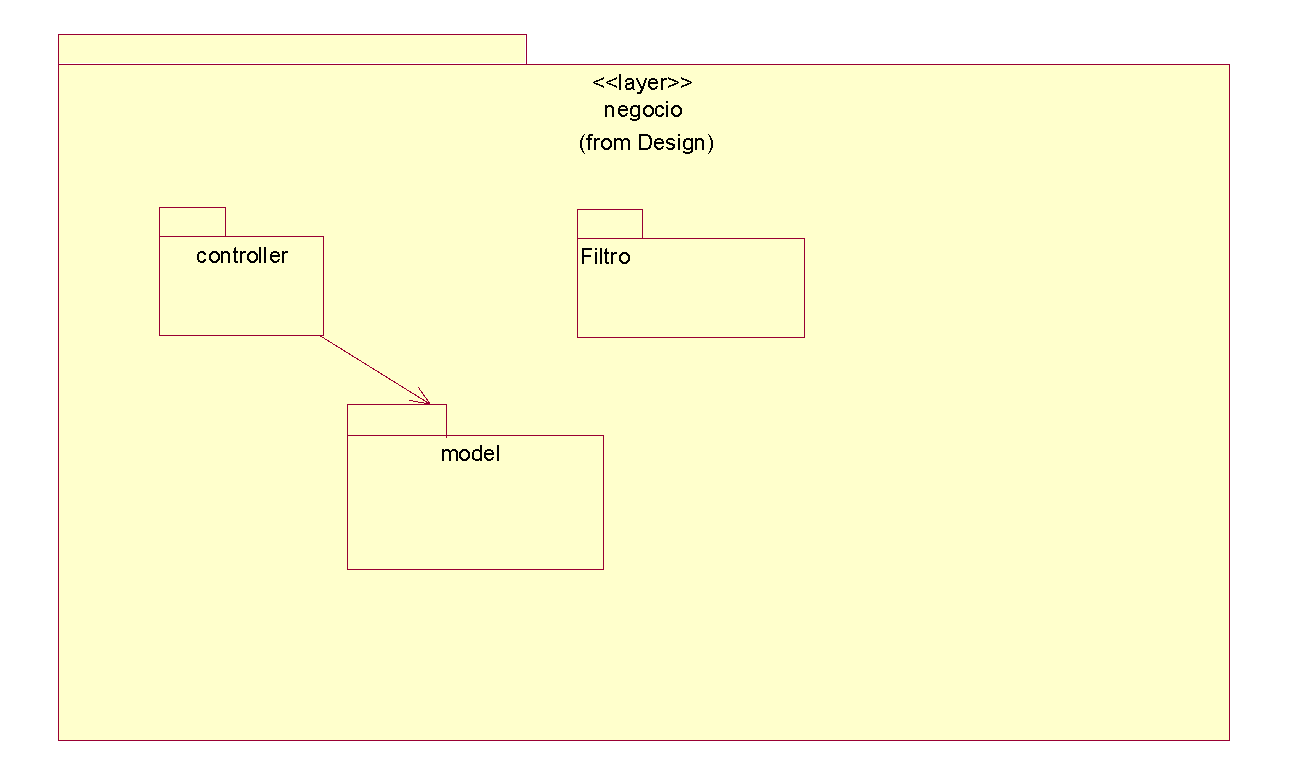


Figura 5.4: Camada de Negócios

* + 1. **Pacote Controller**

A figura 5.5 ilustra as principais classes de controle.

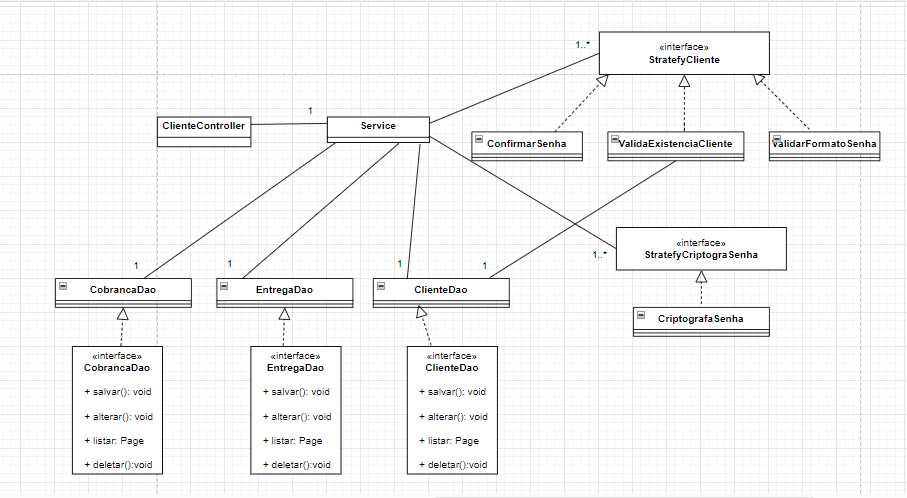


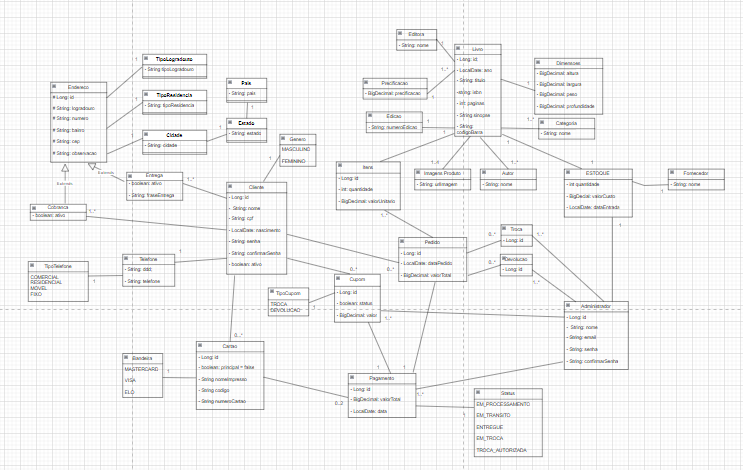
Figura 5.5: Classes de controle

* + 1. **Pacote Model**

Ilustração do diagrama de classe completo.

Acesse o link abaixo ou veja abaixo.

[**https://drive.google.com/file/d/1oBSRG7uMFdrEgYk--LqpYbDHMF1n0pRN/view?usp=sharing**](https://drive.google.com/file/d/1oBSRG7uMFdrEgYk--LqpYbDHMF1n0pRN/view?usp=sharing)

****

## Camada de Persistência

Nesta camada temos o pacote dao que contém as classes e interfaces responsáveis por persistir as informações no BD relacional. O pacote hibernate contido em Dao, possui as classes que dependem diretamente do Hibernate, que é o framework utilizado para realizar o mapeamento objeto relacional.

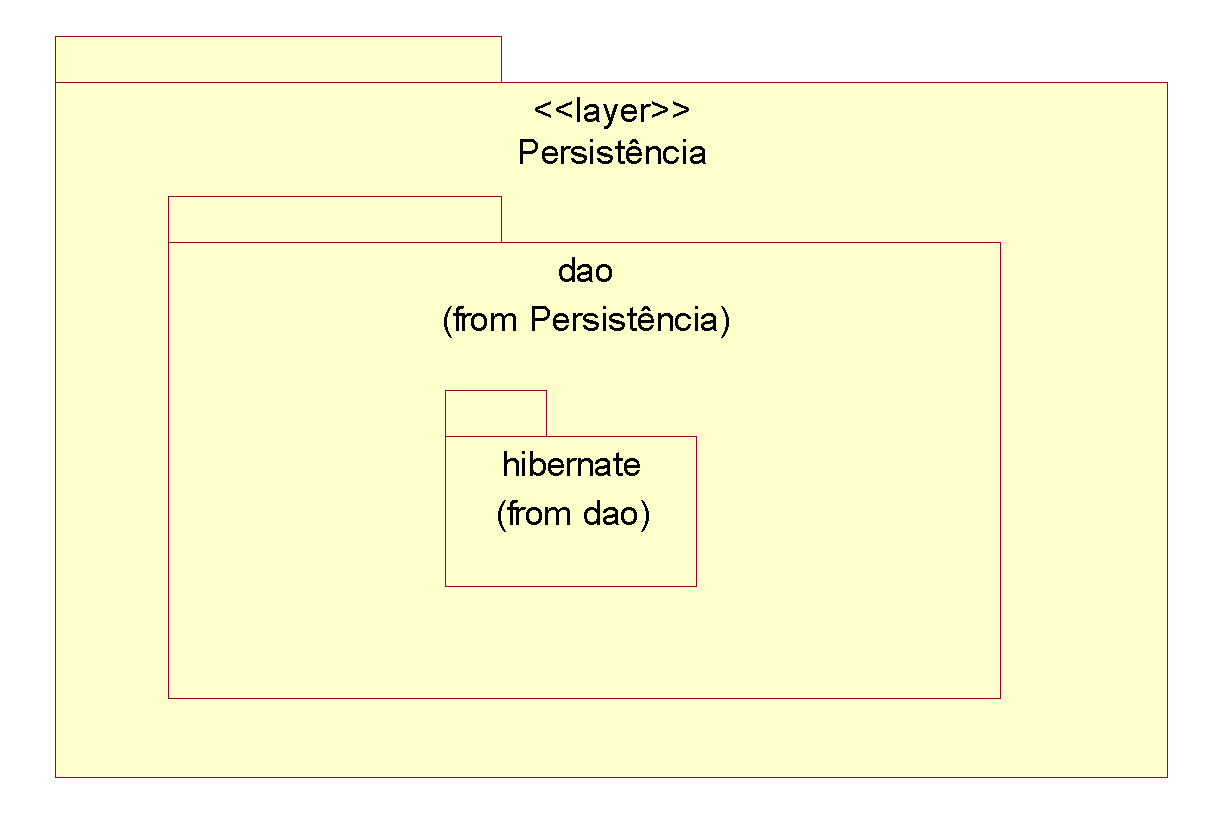
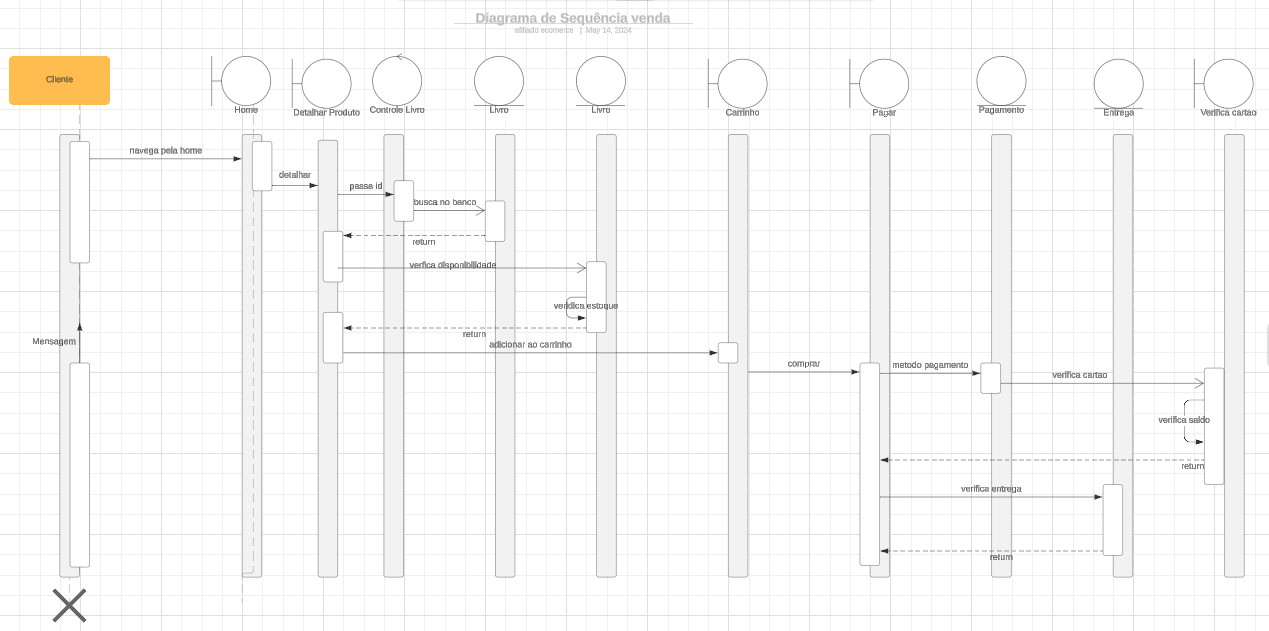


Figura 5.7: Camada de Persistência.

## Realização do diagrama de sequência

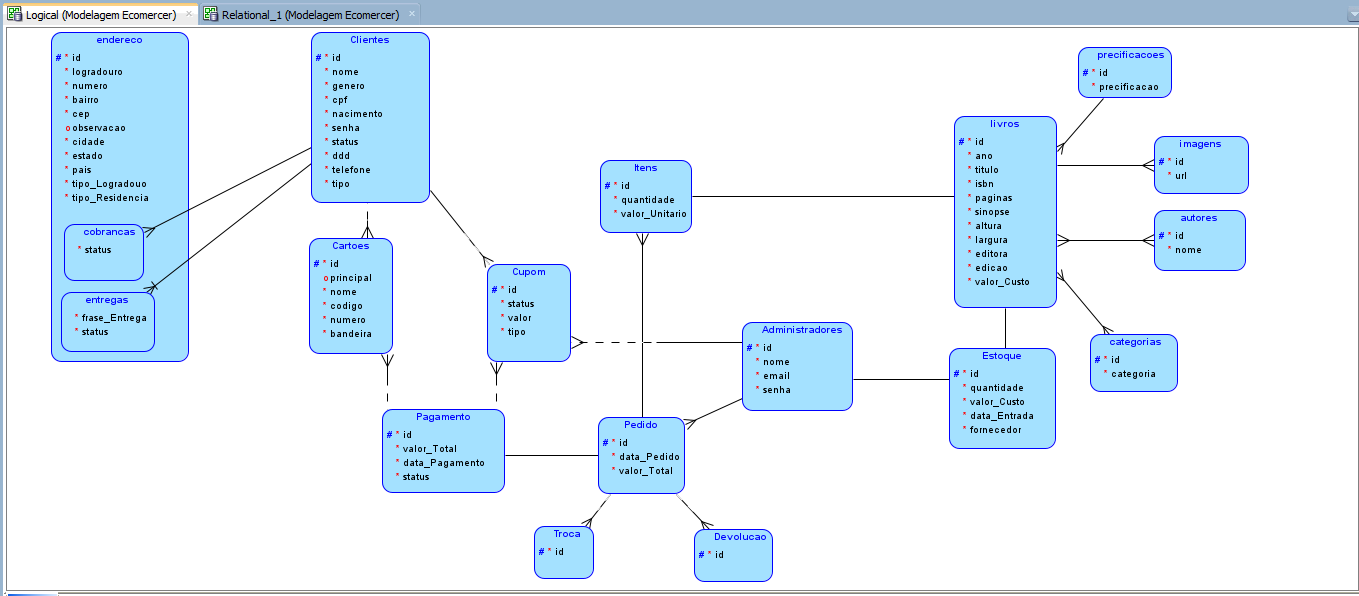


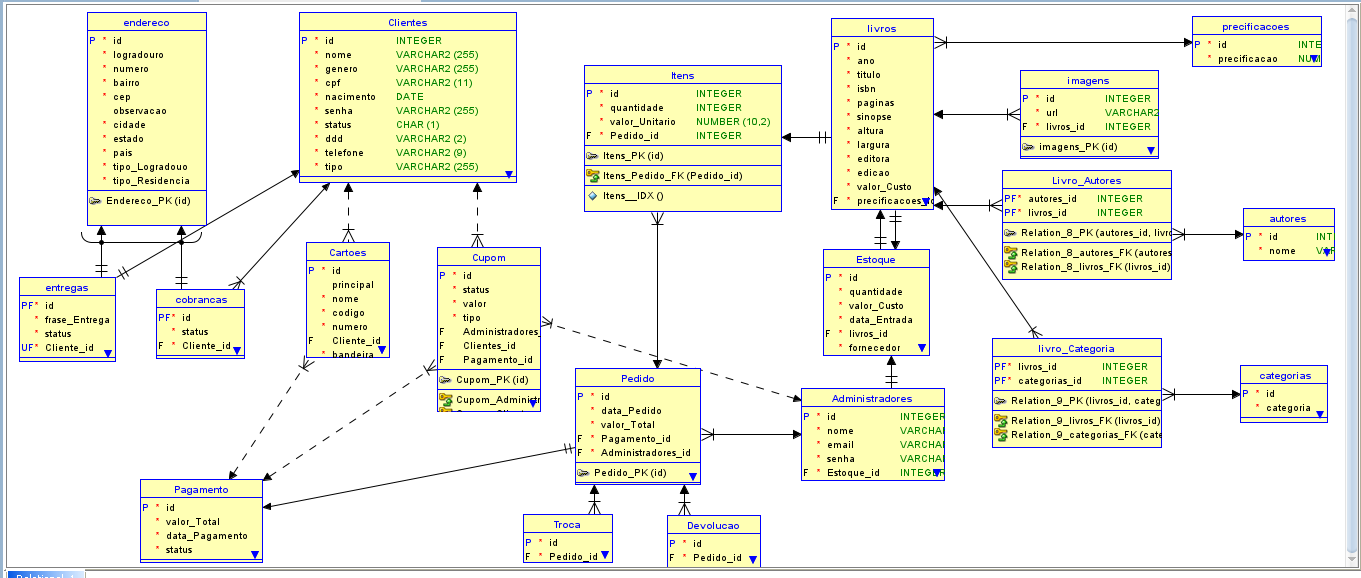
1. **Visão de Dados**

O mecanismo de persistência utilizado no sistema utiliza-se o banco de dados Relacional Mysql juntamente com o framework para mapeamento objeto-relacional, Hibernate. O controle de transações adotado envolve a utilização do Spring Framework em conjunto com o Hibernate.

As figuras XXX e YY, apresentam a visão lógica e física da base de dados da Livraria.

Figura XXX – Modelo Lógico





Figrua XXX – Modelo Físico

A Tabela 2 define o mapeamento das principais classes de modelo para entidades do modelo lógico do BD Mysql.

Tabela 2: Mapeamento Objeto-Relacional

|  |  |
| --- | --- |
| **Classe** | **Entidade** |
|  |  |
|  |  |
|  |  |
|  |  |

1. **Tamanho e Performance**

O sistema será usado para uma livraria e consequentemente terá uma grande quantidade de acesso de clientes assim como do administrador que irá cadastrar os livros.

Seus servidores provavelmente irão passar por períodos de picos de utilização (Natal, Black Friday é dia das crianças).

As estimativas do número de usuários e de carga de utilização em períodos de pico de utilização, bem como maiores informações sobre questões relacionadas ao tamanho e desempenho do sistema podem ser obtidas no documento de requisitos não funcionais.

1. **Qualidade**

O sistema será usado para a venda dos livros e terá um alto volume de transações.

Eventuais erros e/ou falhas na sua operação podem levar a prejuízos significativos tanto em termos financeiros quanto na imagem da Livraria, portanto na fase de design deve-se levar em consideração como fatores prioritários a confiabilidade e robustez do sistema.

Adicionalmente, o sistema pode ser alvo de ataques de “hackers” para roubar ou simplesmente corromper informações, possibilidade aumentada pela interface do sistema disponível na Internet, para evitar que tais ataques sejam bem sucedidos uma infra-estrutura de segurança deve ser especificada e projetada.

Maiores informações sobre questões relacionadas aos requisitos de qualidade do sistema podem ser obtidas no documento de requisitos não funcionais.

1. **Cronograma Macro.**

Os prazos podem ser dados em semanas ou meses dentro do projeto ou ate mesmo data.

|  |  |
| --- | --- |
| **Resultado** |  |
| Plano Preliminar | Semana 2 |
| Plano Fase 1 | Semana 3 |
| Especificação Fase 1 | Semana 5 |
| Piloto Fase 1 | Semana 11 |
| Solução Testada Fase 1 | Semana 12 |
| Plano Fase 2 | Semana 13 |
| Especificação Fase 2 | Semana 14 |
| Piloto Fase 2 | Semana 20 |
| Solução Testada Fase 2 | Semana 21 |

Obs: Os prazos apresentados são uma estimativa inicial considerando as informações disponíveis nesta etapa do projeto. Um cronograma detalhado será elaborado na fase de planejamento e, eventualmente, estes prazos podem ser modificados.