**FUNDAÇÃO EDUCACIONAL DE FERNANDÓPOLIS**

**FACULDADES INTEGRADAS DE FERNANDÓPOLIS**

Gabriel Mariano

Luiza Andrade

Gabriel Augusto

Thuliane Assis da Silva

João Vincente

**CONTROLE DE ESTOQUE**

**SISTEMA DE CONTROLE DE ESTOQUE**

**Fernandópolis**

**2024**

# Sumário

[0 Sumário 2](#_Toc152512805)

[1 INTRODUÇÃO 3](#_Toc152512806)

[1.1 Levantamento de Informações 3](#_Toc152512807)

[1.1.2 Descrição dos Objetivos Principais 3](#_Toc152512808)

[1.1 Descrição do Sistema de Informação Atual 4](#_Toc152512809)

[1.1.4 Ineditismo do Projeto 4](#_Toc152512810)

[2 ANÁLISE ORIENTADA À OBJETOS 5](#_Toc152512811)

[2.1 Lista de Atores 5](#_Toc152512812)

[2.2 Lista de Casos de Uso 6](#_Toc152512813)

[2.3 Diagrama de Contexto Geral 12](#_Toc152512814)

[2.4 Diagramas de Caso de Uso Individual 14](#_Toc152512815)

[3 PROJETO ORIENTADO À OBJETOS 21](#_Toc152512816)

[3.1 Diagrama de Classes 21](#_Toc152512817)

[3.2 Dicionário de Atributos das Classes 22](#_Toc152512818)

[3.3 Diagramas de Classes Camada Controller 25](#_Toc152512819)

[3.4 Diagramas de Classes Camada DAO 31](#_Toc152512820)

[3.5 Projeto Físico do Banco de Dados 34](#_Toc152512821)

[3.6 Especificação do Layout de Telas e Relatórios 39](#_Toc152512822)

[3.7 Diagrama de Classes – Interface Homem-Máquina 44](#_Toc152512823)

[3.8 Diagrama de Sequência – Modelo MVC 46](#_Toc152512824)

[3.9 Diagrama de Máquina de Estados 55](#_Toc152512825)

[3.10 Controle de Acesso e Segurança 56](#_Toc152512826)

# INTRODUÇÃO

A falta de um controle de estoque na escola pode gerar desperdício de recursos, interrupções nas atividades educacionais e dificuldade na prestação de contas.

O uso de um *software* web que possibilite o controle de estoque em uma escola é fundamental para garantir a eficiência e a transparência na gestão dos recursos. Com o *software* adequado, é possível monitorar o fluxo de entrada e saída de materiais (cadernos, canetas, pincel e entre outros materiais de escritório) evitando desperdícios e garantindo o uso otimizado dos recursos disponíveis. Além disso, um controle de estoque permite identificar rapidamente a necessidade de reposição dos itens promovendo uma gestão eficiente, redução de custos e melhor aproveitamento dos recursos disponíveis.

Dessa forma concluísse que a implementação de um *software*web de controle de estoque para as escolas em específica a Escola Estadual Líbero de Almeida Silvares (EELAS) se faz necessário porque resultará em uma redução significativa de desperdícios, otimização do uso dos recursos e maior transparência na gestão, proporcionando um ambiente mais eficiente e sustentável para as atividades educacionais.

## Levantamento de Informações

Foi realizado um estudo através da documentação fornecida pelo cliente, de acordo com o software desejado para que assim fosse possibilitada a implementação e desenvolvimento do sistema.

Unindo as descrições feitas na documentação e o conteúdo acadêmico disponibilizado pelo corpo docente, junto as instruções deles, a arquitetura do software foi montada, produzindo um sistema que esteja de acordo com o solicitado.

No conteúdo disposto foi possível inferir a necessidade do contratante em possuir um produto aprimorado que apresentasse o caminho percorrido pelos itens incluídos e dispostos no estoque, desde o cadastramento até a movimentação do inventário como entrada e saída, e também o funcionário que solicitou a adesão de produtos ou a saída dos mesmos.

## Descrição dos Objetivos Principais

O objetivo principal de um controle de estoque para uma escola é fazer um sistema web e mobile que garanta a disponibilidade adequada de materiais e suprimentos para atender às necessidades dos alunos, professores e funcionários.

Ao acompanhar o estoque atual e as quantidades mínimas desejadas, é possível identificar os momentos ideais para solicitar novos itens, evitando a falta de materiais essenciais ou a compra excessiva de itens que já estão disponíveis em quantidade suficiente.

Com registros detalhados sobre os itens, quantidades, custos e datas, é possível realizar análises de consumo, identificar tendências, estimar custos futuros e tomar decisões embasadas em dados concretos.

## Descrição do Sistema de Informação Atual

Efetuamos as compras de 2 maneiras:

1 - Através de site próprio da secretaria da educação com verba pré-estabelecida. Recebemos esta mercadoria através de transportadora com nota fiscal, realizamos a conferência da mercadoria e armazenamos nos armários para uso, os servidores vem pedem o material e a gente entrega.

2 - Os servidores pedem os materiais que precisam ao COE, que efetua a compra em papelarias da cidade, ele vai fazendo os pedidos, as vezes picados e com um certo prazo a papelaria gera nota fiscal \*(as vezes vem divergente dos produtos recebidos), entrega pra pessoa que pediu e armazena a nota que é feita no CNPJ da APM. Algumas compras são com verba, outras com recurso próprio.

**1.1.3 Descrição dos Principais Problemas Existentes**

Um problema comum é a falta de visibilidade do estoque disponível na escola. Isso pode resultar em dificuldades para identificar quais itens estão disponíveis, em quais quantidades e sua localização física.

Sem um controle rigoroso das entradas e saídas de itens do estoque, é difícil manter um registro preciso da quantidade disponível. A ausência desse controle pode resultar em erros de contagem, perdas, itens desaparecidos, desvios ou até furtos.

Muitas escolas ainda dependem de processos manuais e planilhas eletrônicas básicas para gerenciar seu estoque. Isso pode levar a erros de inserção de dados, dificuldades em procurar informações e dificuldade na hora de gerar um relatório.

Uma má comunicação com os fornecedores pode levar a problemas de reposição de estoque. Por exemplo, falta de comunicação sobre as quantidades mínimas desejadas, falta de coordenação na realização de pedidos ou falta de respostas rápidas aos pedidos de reposição podem resultar em atrasos ou falta de materiais essenciais.

## 

## 1.1.4 Ineditismo do Projeto

O projeto apresenta como ineditismo a segreção de acessos no sistema e estoque centralizando para somente o administrador dar baixa no estoque. O uso dessas tecnologias permitirá ao profissional agilizar todo o processo de controle de estoque, e maior transparência na gestão dos materiais escolares, permitindo assim transparência na gestão dos materiais escolares.

# ANÁLISE ORIENTADA À OBJETOS

Orientação a Objetos (OO) é um paradigma de programação que revolucionou a forma como os programas são desenvolvidos, proporcionando uma abordagem mais modular, flexível e reutilizável. Este conceito baseia-se na idéia de modelar sistemas como interações entre “objetos”, entidades que encapsulam dados e comportamentos relacionados. Vamos explorar essa abordagem inovadora, utilizando citações de renomados especialistas em engenharia de software.

Para Alan Kay, um dos idealistas por trás do conceito de OO, “Objetos se comunicam trocando mensagens”. Essa definição ressalta a importância da comunicação entre objetos, que interagem através de mensagens, permitindo a construção de sistemas complexos a partir de componentes independentes.

A orientação a objetos também é referenciada na citação de Bertrand Meyer, onde diz que: “A orientação a objetos é uma maneira de encapsular variáveis e os métodos que operam nessas variáveis, e, subsequentemente, o pacote de um objeto pode ser passado em torno do sistema.” destacando assim o encapsulamento, onde o princípio-chave da OO, promove a segurança e a modularidade do código.

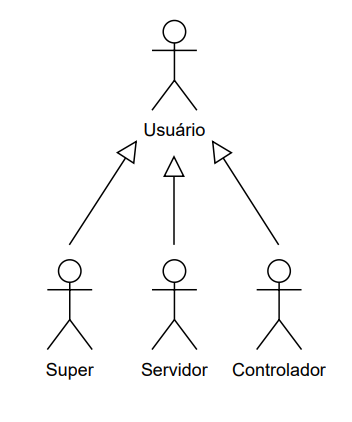
.

## Lista de Atores

De acordo com Guedes (2011) a lista de atores é um dos dois principais itens do diagrama de casos de uso, sendo o outro os próprios casos de uso. Os atores representam os papéis desempenhados pelos diversos usuários que poderão utilizar, de alguma maneira, os serviços e funções do sistema. Um ator pode ser qualquer elemento externo que interaja com um software.

Ator é um usuário que interage com o sistema, é um objeto ou um conjunto de objetos que se comunica com o sistema, e assim são chamados porque cada um desempenha um papel dentro do sistema (GÓES, 2014).

A figura abaixo demonstra todos os atores que interagem no sistema.



* **Ator Usuário**: interage com as funcionalidades do sistema e realiza todas funções permitidas aos Atores Usuário Super, Usuário Servidor e Usuário Controlador que herdam todas as funcionalidades existentes no Ator Usuário. De acordo com Sbrocco (2014) o conceito de herança representa um mecanismo específico do paradigma orientado a objetos que permite criar classes a partir de outras já existentes, aproveitando as características da classe que será estendida.
* **Ator Usuário Super:** interage com todo o sistema através das seguintes funcionalidades: Realizalogin, Consulta produto, altera produto, Consulta Estoque, altera estoque, Consulta Fornecedor, Consulta Funcionário, Altera Funcionário, Cadastro de Materiais, Realiza Acerto de Estoque, Geração Relatórios, Aprovação de Solicitações.
* **Ator Usuário Servidor:**interage com o sistema através das seguintes funcionalidades: Realiza Login, Consultar/localizar/buscar, Solicitação de materiais, Consulta de disponibilidade, acompanha solicitações, recebe notificações de status.
* **Ator Usuário Controlador:**interage com o sistema através das seguintes funcionalidades: Realiza Login, Lista Requisições, Aprova Requisições, Reprova Requisições, realiza retirada de estoque, Cadastra Relatório, Pesquisa Relatório, Altera Relatório, Exclui Relatório.

## Lista de Casos de Uso

Os casos de uso é uma técnica para captar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre o usuário de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o sistema é utilizado (FOWLER, 2005).

De acordo com Guedes (2011) os casos de uso podem ser classificados em casos de uso primários (quando se refere a um processo importante que enfoca um dos requisitos funcionais de um software) e secundários (processos periféricos do sistema).

A documentação dos casos de uso servirá de base para outros diagramas, como a diagrama de sequência, por exemplo. O objetivo principal da documentação de um caso de uso é fornecer ao cliente um relatório que explique qual o comportamento pretendido por esse caso de uso e quais funções ele deverá executar.

Nas tabelas de nº 01 a 03 que estão logo abaixo, serão demonstradas as listas de caso de uso de forma individualizada por funcionalidades dos atores Usuário Super, Usuário Servidor e Usuário Controlador com o sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Usuário Super/Gestor da Escola** | | | | |
| **Nº** | **Descrição** | **Caso de Uso** | **Entrada** | **Saída** |
| 01 | Usuário realiza login no Software | Autenticação | Credenciais de login | Msg: 01  Acesso ao sistema |
| 02 | Consultar/  Localizar/  Buscar Produtos | Gestão de Compras | Parâmetros de busca (fornecedor, data, valor) | Msg: 02 |
| 03 | Altera Produto | Alterar Produtos | Dados Produtos | Msg: 11 |
| 04 | Consultar/  Localizar/Buscar Estoque | Buscar Estoque, Gestão de Estoque |  | Msg:03  Lista de materiais disponíveis |
| 05 | Altera Estoque | Gestão de Estoque | Dados Estoque | Msg:11 |
| 06 | Consultar/Localizar/Buscar Fornecedor | Gestão de Fornecedores | Parâmetros de busca (nome, contato) | Msg: 04  Informações detalhadas do fornecedor |
| 07 | Consultar/Localizar/Buscar Funcionário | Gestão de Funcionários | Parâmetros de busca (nome, cargo, contato) | Msg: 05  Informações detalhadas do funcionário |
| 08 | Altera Funcionário | Gestão de Funcionários |  | Msg: 06 |
| 09 | Cadastro de Materiais | Gestão de Estoque | Informações do material (nome, quantidade) | Msg: 07  Confirmação do cadastro do material |
| 10 | Geração de Relatórios | Relatórios | Parâmetros do relatório (vendas, estoque) | Msg: 08 |
| 11 | Realiza acerto de estoque | Painel de Controle |  | Msg: 09 |
| 12 | Aprovação de Solicitações | Gestão de Solicitações | Solicitação (pedido de compra, folga) | Msg: 10 |

A tabela 1 exibe a lista de casos de usos entre o Ator Pessoa Funcionário e o Sistema.

**Tabela 1 - Lista de Casos de Uso - Usuário Super**

A tabela 2 exibe a lista de casos de usos entre o Ator Pessoa Profissional e o Sistema.

**Tabela 2 - Lista de Casos de Uso - Usuário Servidor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Usuário Servidor** | | | | |
| **Nº** | **Descrição** | **Caso de Uso** | **Entrada** | **Saída** |
| 01 | Usuário realiza login no Software | Autenticação | Credenciais de login | Msg: 01  Acesso ao sistema |
| 02 | Consultar/  Localizar/  Buscar Produtos | Gestão de Compras | Parâmetros de busca (fornecedor, data, valor) | Msg: 02  Informações detalhadas da compra |
| 03 | Solicitação de materiais | Gestão de Estoque |  | Msg: 03  Lista de materiais indisponíveis |
| 04 | Consulta de disponibilidade | Gestão de Estoque | Parâmetros de busca (nome, contato) | Msg: 04  Informações detalhadas do fornecedor |
| 05 | Acompanhamento de Solicitação | Gestão de Solicitações | Solicitação (pedido de compra, folga) | Msg: 05 |
| 06 | Status Compra | Gestão de Compras | Parâmetros de busca (compra, pedido) | Msg: 06  Informações detalhadas da compra |

A tabela 3 exibe a lista de casos de usos entre o Ator Pessoa Paciente e o Sistema.

**Tabela 3 - Lista de Casos de Uso - Usuário Controlador*.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Usuário Controlador** | | | | |
| **Nº** | **Descrição** | **Caso de Uso** | **Entrada** | **Saída** |
| 01 | Usuário realiza login no Software | Autenticação | Credenciais de login | Msg: 01  Acesso ao sistema |
| 02 | Lista Requisições | Gestão de Compras | Parâmetros de busca (pedidos,  requisições) | Msg: 02  Visualiza Requisições pendentes |
| 03 | Aprova Requisições | Gestão de Compras | Parâmetros de busca (pedidos,  Requisições) | Msg:  Aprova Requisições  Pendentes |
| 04 | Reprova Requisições | Gestão de Compras | Parâmetros de busca (pedidos,  Requisições) | Msg:  Reprova Requisições  Pendentes |
| 05 | Realiza retirada | Gestão de Estoque | Parâmetros de Retirada (pedidos,  Requisições) | Msg: 05  Atualização de estoque |
| 06 | Cadastra Relatório | Gestão de Relatórios | Parâmetros de cadastro (pedidos,  Requisições) | Msg: 06  Registra relatório no sistema |
| 07 | Pesquisa Relatório | Gestão de Relatórios | Parâmetros de Relatórios  (Pedidos,  Requisições) | Msg: 07  Pesquisa de Relatórios |
| 08 | Altera Relatório | Gestão de Relatórios | Parâmetros de alteração (Relatórios) | Msg: 08  Alteração de relatório |
| 09 | Exclui Relatório | Gestão de Relatórios | Parâmetros de alteração (Relatórios) | Msg:09  Remove relatório |

#### 2.2.1 Lista de Mensagens

A tabela 4 demonstra as mensagens que o sistema retornará em resposta às entradas requisitadas pelos usuários nas tabelas de Caso de Uso acima demonstradas.

Tabela - 4 Lista de Mensagens - Usuário Super.

|  |  |
| --- | --- |
| **Nº Mensagem** | **Mensagem** |
| Msg 01 | “Login realizado com sucesso. Bem-vindo(a)” / “Login ou senha inválidos!” |
| Msg 02 | "Consulta realizada com sucesso" / "Nenhuma compra encontrada” |
| Msg 03 | "O estoque atualizado está disponível para consulta. “ / "Não há informações disponíveis sobre o estoque no momento." |
| Msg 04 | "Detalhes completos sobre seus fornecedores estão disponíveis aqui." / "Nenhum fornecedor encontrado com base na sua pesquisa." |
| Msg 05 | "Aqui estão as informações detalhadas sobre seus funcionários." / "Nenhum funcionário encontrado com base nos critérios de pesquisa." |
| Msg 06 | "As informações do funcionário foram atualizadas com sucesso." / "Erro ao tentar atualizar as informações do funcionário. Tente novamente." |
| Msg 07 | "Novo material cadastrado com sucesso no sistema." / "Erro ao tentar cadastrar o novo material. Verifique os dados e tente novamente." |
| Msg 08 | "Relatório gerado com sucesso. Você pode visualizá-lo na seção de relatórios." / "Erro ao tentar gerar o relatório. Por favor, revise as configurações e tente novamente." |
| Msg 09 | "Acerto estoque disponivel." / "Sem dados disponíveis para exibir no momento." |
| Msg 10 | "A solicitação foi aprovada com sucesso. Os procedimentos serão iniciados." / "A solicitação foi reprovada. Entre em contato para mais informações." |
| Msg 11 | “Dados alterados com sucesso” / “Os dados não pode ser alterados” |

Tabela - 5 Lista de Mensagens - Usuário Servidor.

|  |  |
| --- | --- |
| **Nº Mensagem** | **Mensagem** |
| Msg 01 | “Login realizado com sucesso. Bem-vindo(a)” / “Login ou senha inválidos!” |
| Msg 02 | "Solicitação de materiais registrada com sucesso." / "Erro ao processar a solicitação. ” |
| Msg 03 | "Sua solicitação de materiais foi registrada. Acompanhe o status." / "Houve um problema ao processar sua solicitação. Tente novamente mais tarde." |
| Msg 04 | "A disponibilidade dos materiais desejados foi verificada com sucesso." / "Não há disponibilidade para os materiais solicitados no momento." |
| Msg 05 | "Aqui estão as últimas atualizações sobre suas solicitações em andamento." / "Não há solicitações em andamento para mostrar." |
| Msg 06 | "Você recebeu uma notificação sobre o status da sua solicitação. Confira agora." / "Nenhuma notificação nova no momento." |

Tabela – 6 Lista de Mensagens - Usuário Controlador.

|  |  |
| --- | --- |
| **Nº Mensagem** | **Mensagem** |
| Msg 01 | “Login realizado com sucesso. Bem-vindo(a)” / “Login ou senha inválidos!” |
| Msg 02 | "Você tem [X] requisições pendentes para análise." / "A requisição [ID] não foi encontrada.” |
| Msg 03 | "Requisição [ID] aprovada com sucesso” |
| Msg 04 | "Requisição [ID] reprovada” |
| Msg 05 | "Retirada de materiais da requisição [ID] realizada com sucesso." / "Relatório [Número do Relatório] atualizado.” |
| Msg 06 | "Relatório cadastrado com sucesso.” |
| Msg 07 | “Relatório [ID] encontrado “ / "O relatório [ID] não foi encontrado. ” |
| Msg 08 | “Relatório [Número do Relatório] alterado com sucesso." / “ Relatório não pode ser gerado” |
| Msg 09 | "Relatório [Número do Relatório] excluído com sucesso." |

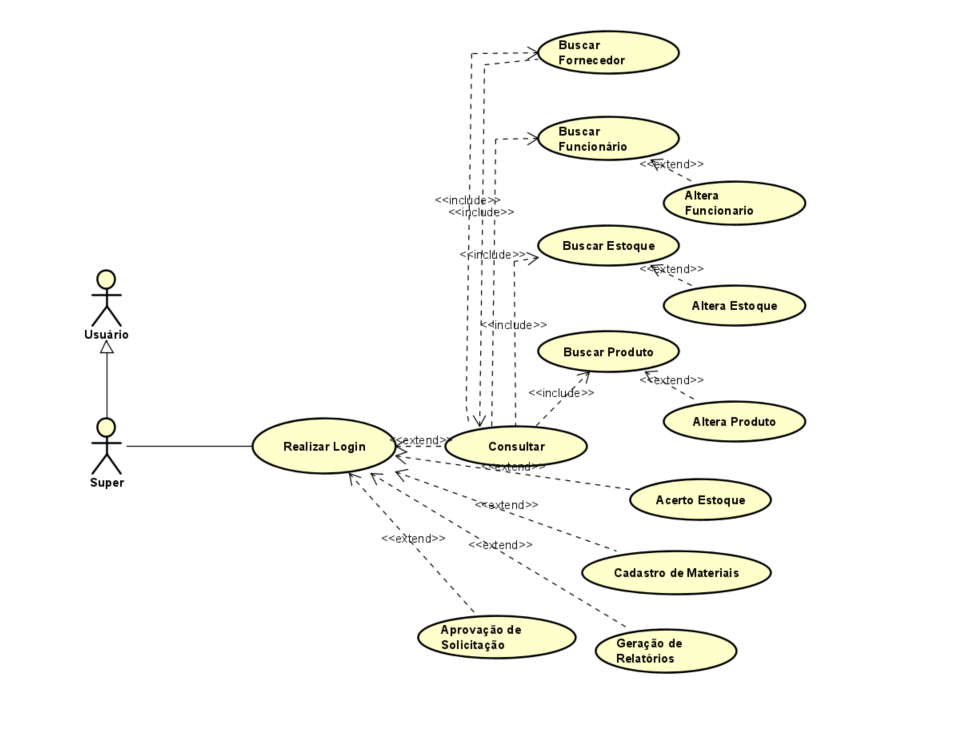
## Diagrama de Contexto Geral

O Diagrama de contexto geral é um diagrama de casos de uso que representa de forma geral todos os atores e casos de uso apresentando o sistema como um todo (SBROCCO, 2014).

A seguir os diagramas serão apresentados em módulos representando a interação de cada ator com o sistema.

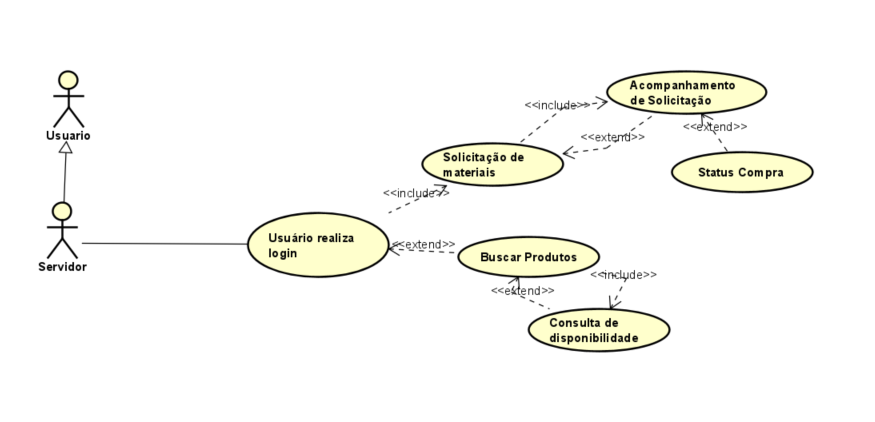
A figura abaixo representa o Diagrama de Contexto Geral no módulo Pessoa Super.

Diagrama de Contexto Geral - Módulo Super

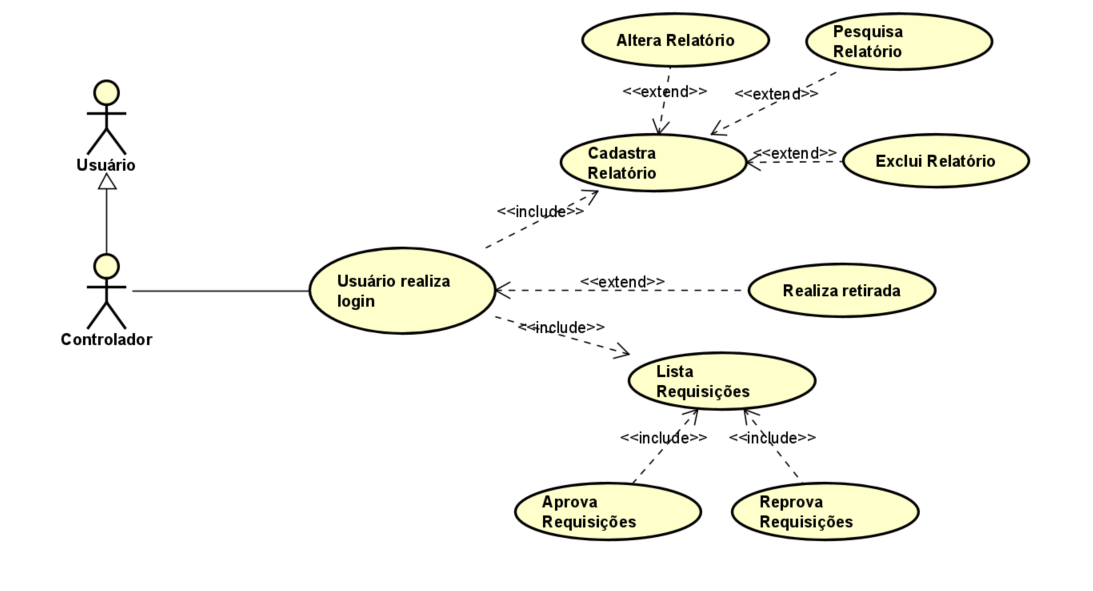


A figura abaixo representa o Diagrama de Contexto Geral no módulo Pessoa Servidor.

Diagrama de Contexto Geral - Módulo Servidor



A figura abaixo representa o Diagrama de Contexto Geral no módulo Pessoa Controlador.

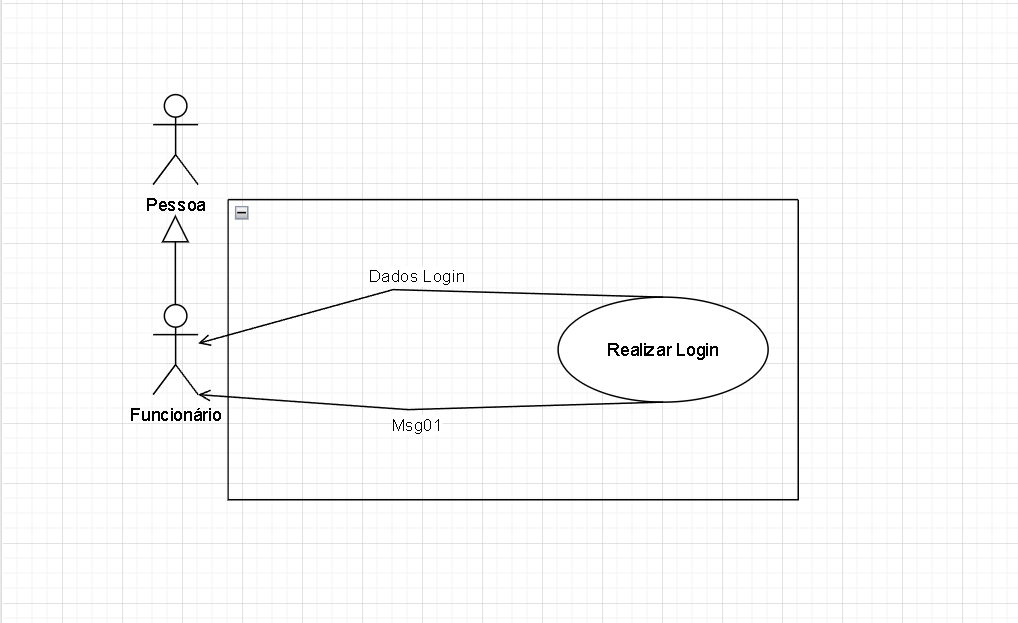
Diagrama de Contexto Geral - Módulo Controlador

## Diagramas de Caso de Uso Individual

O Diagrama de caso de uso individual mostra de forma individualizada as interações entre um dos atores e o sistema, representa cada funcionalidade do sistema permitindo uma visualização destacada de como cada ator desempenhará suas funções destro do sistema (SBROCCO, 2014).

#### 2.4.1 DCU Individual Ator Funcionário - realizar login

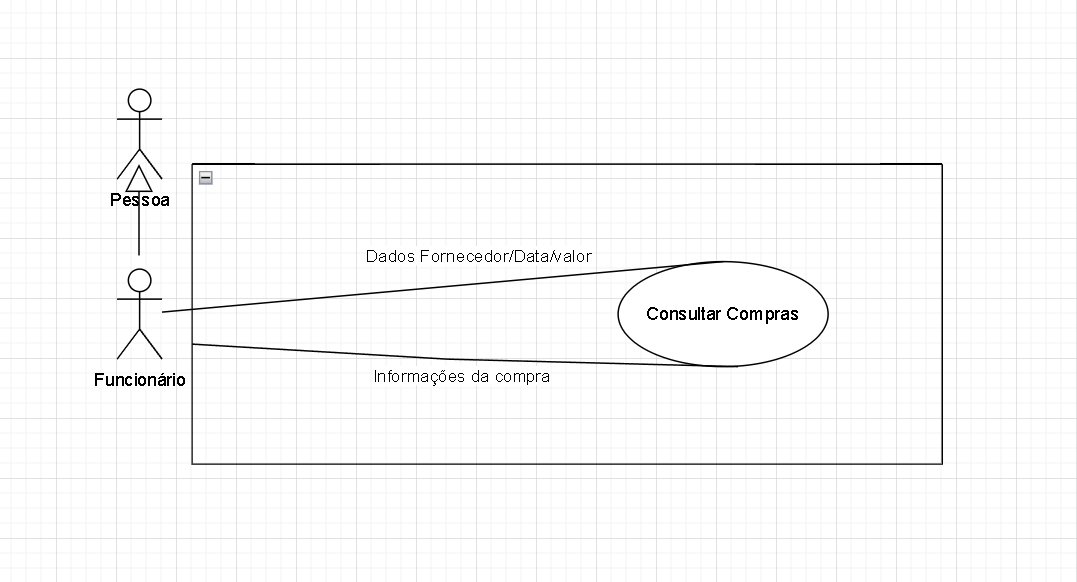
A figura abaixo representa o Caso de Uso Individual Realizar Login no módulo Ator Funcionário.



|  |
| --- |
| Ator Principal: Funcionário |
| Descrição: Funcionário informa login e senha. O sistema valida e permite o acesso |
| Pré-requisito: Funcionário não estar logado. |
| Fluxo Normal:   1. Funcionário Loga no sistema. 2. Insere as informações de login na página de login. 3. Sistema valida as informações de login. 4. Sistema exibe página principal. 5. Fim |
| Fluxo Alternativo:  3.1 -Informações de login erradas o sistema retorna uma mensagem.  3.1.1 - Retorna para o item 2. |
| Dados: Usuário e senha. |

#### 2.4.2 DCU Individual Ator Funcionário - Consulta Compras

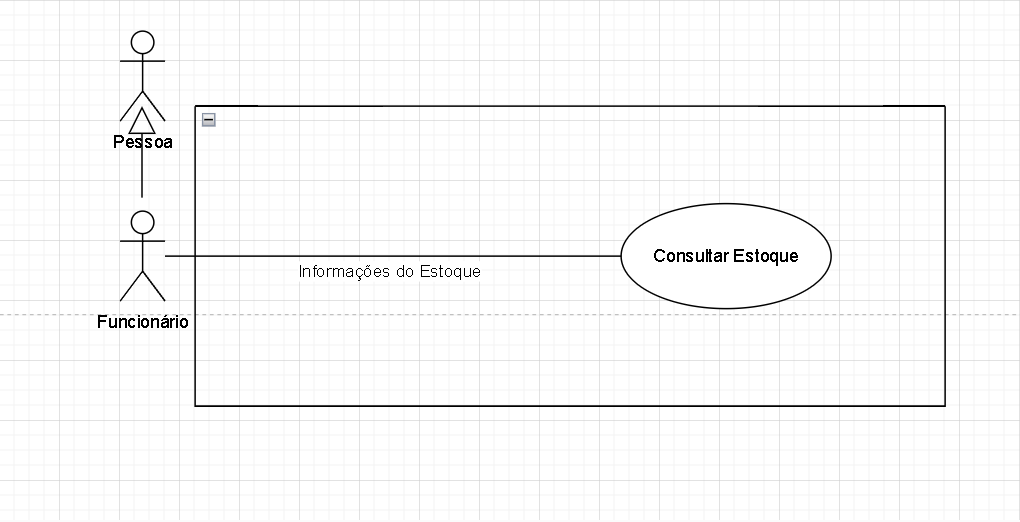
A figura abaixo representa o Caso de Uso Individual Consultar Compras no modulo Compras



|  |
| --- |
| Ator Principal: Funcionário |
| Descrição: Funcionário informa parâmetros de busca para compras e sistema retorna a pesquisa. |
| Pré-requisito: Ter uma compra cadastrada no sistema. |
| Fluxo Normal:   1. Funcionário Informa parâmetros 2. Sistema usa os parâmetros para fazer pesquisa 3. Sistema retorna o resultado da pesquisa. 4. Fim |
| Fluxo Alternativo:  2.1 - Parâmetros insuficientes ou errados  2.1.1 - Sistema retorna mensagem |
| Dados: Fornecedor, valor, data. |

#### 2.4.3 DCU Individual Ator Funcionário - Consulta Estoque

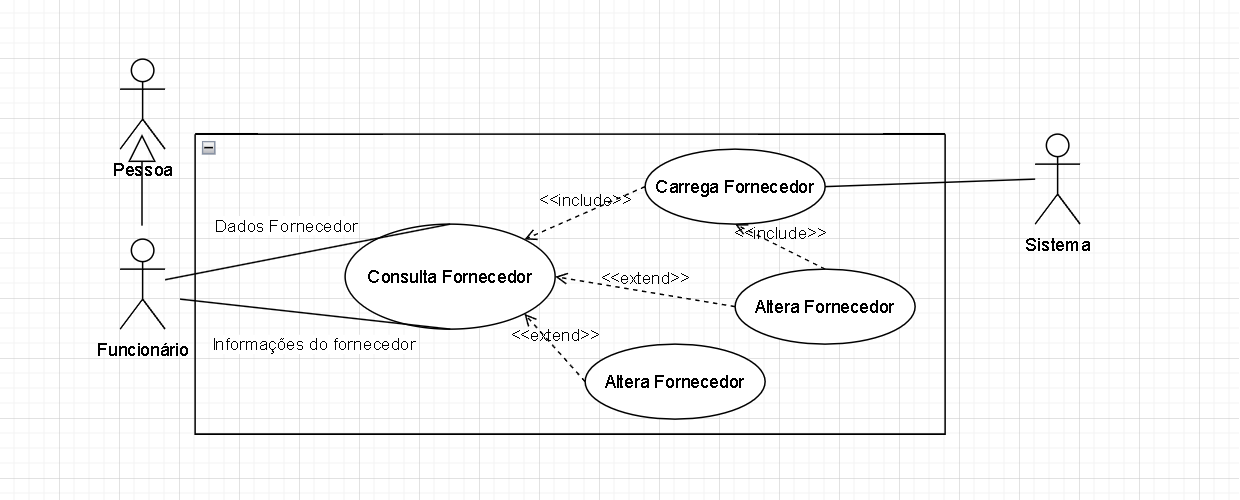
A figura abaixo representa o Caso de Uso Individual Consultar Estoque no modulo Estoque



|  |
| --- |
| Ator Principal: Funcionário |
| Descrição: Funcionário abre tela do estoque |
| Pré-requisito: Ter algum material no estoque. |
| Fluxo normal:  1 - Após login funcionário clica no botão “Consultar Estoque”  2 – Sistema direciona funcionário para tela de Consulta do Estoque.  3 – Fim. |
| Fluxo Alternativo:  2.1 - Caso estoque estiver vazio o sistema retorna uma mensagem. |
| Dados: nenhum. |

#### 2.4.4 DCU Individual Ator Funcionário - Consulta Fornecedor

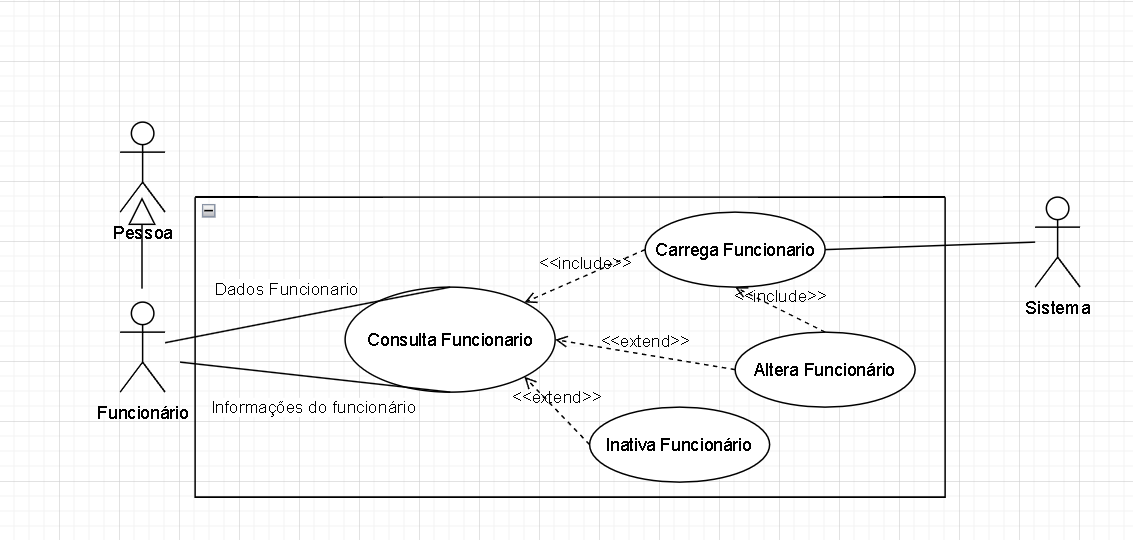
A figura abaixo representa o Caso de Uso Individual Consultar Fornecedor no modulo fornecedor.



|  |
| --- |
| Ator Principal: Funcionário |
| Descrição: Funcionário pesquisa fornecedor. |
| Pré-requisito: Ter um fornecedor Cadastrado |
| Fluxo Normal:   1. Funcionário informa dados do fornecedor 2. Sistema retorna o/os fornecedores 3. Fim. |
| Fluxo Alternativo:  2.1 - Sistema retorna mensagem caso não tiver fornecedor. |
| Dados: Nome, Cnpj, Cpf, Contato |

#### 2.4.5 DCU Individual Ator Funcionário - Consulta Funcionário

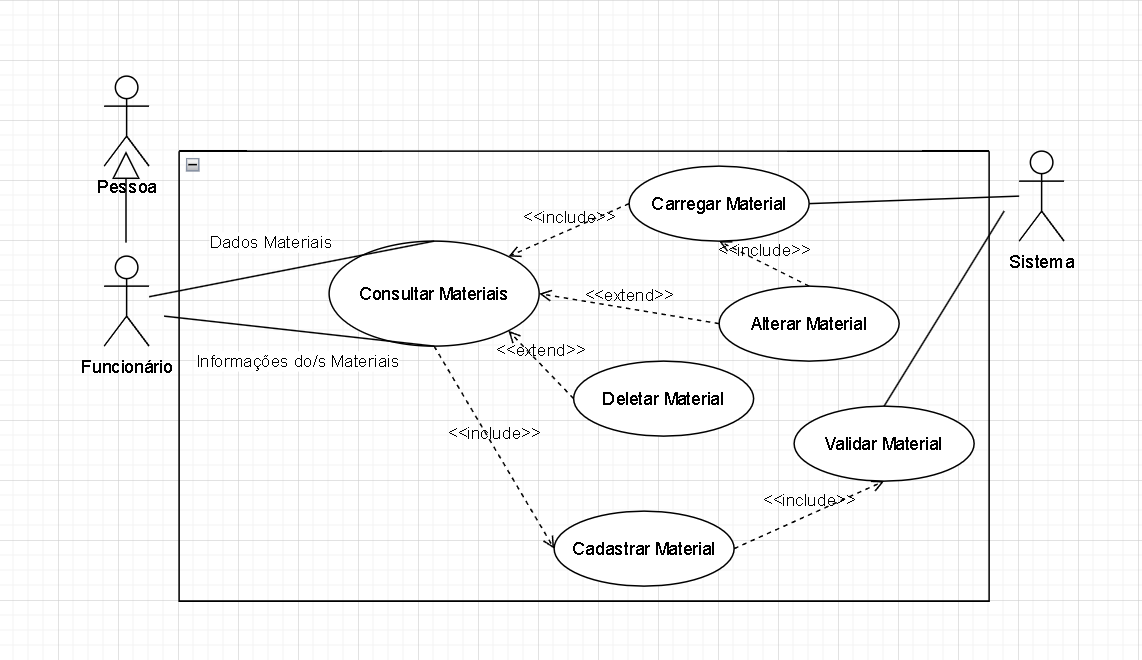
A figura abaixo representa o Caso de Uso Individual Consultar Funcionário no modulo Funcionário.



|  |
| --- |
| Ator Principal: Funcionário |
| Descrição: Funcionário pesquisa lista de funcionários. |
| Pré-requisito: Ter funcionário cadastrado. |
| Fluxo Normal:   1. Funcionário informa dados do funcionário 2. Sistema retorna lista dos funcionários. 3. Fim |
| Fluxo Alternativo:  2.1 Parâmetros errados ou sistema sem funcionários retorna mensagem. |
| Dados: Nome, Cpf, Cargo, Contato. |

#### 2.4.6 DCU Individual Ator Funcionário - Consulta Materiais

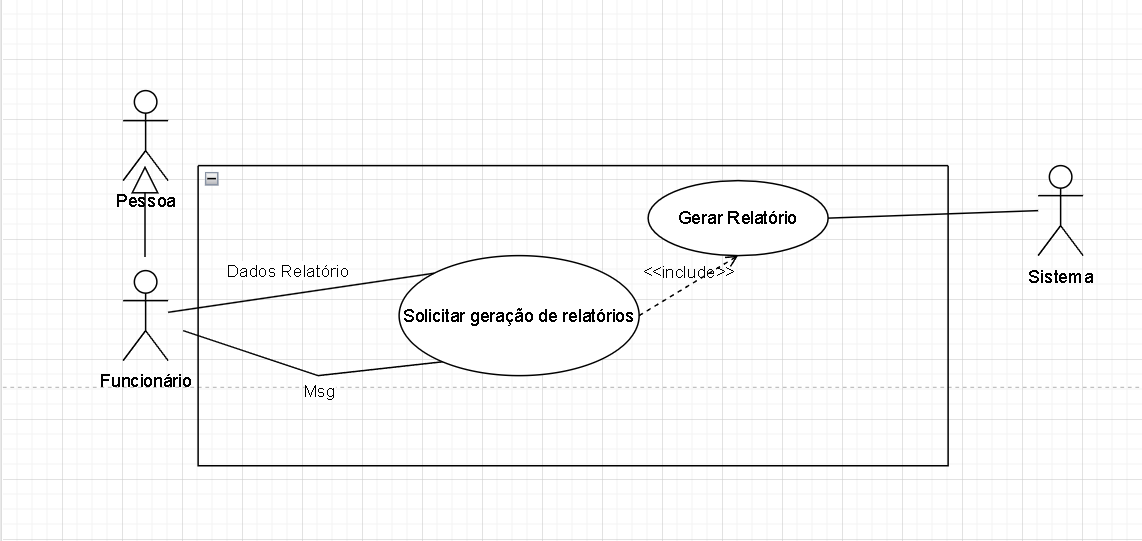
A figura abaixo representa o Caso de Uso Individual Consultar Materiais no modulo materiais.



|  |
| --- |
| Ator Principal: Funcionário. |
| Descrição: Consulta materiais. |
| Pré-requisito: Ter materiais Cadastrado |
| Fluxo Normal:   1. Funcionário informa dados materiais 2. Sistema retorna lista com os materiais. 3. Fim |
| Fluxo Alternativo:  2.1 - Caso não tenha material com os parâmetros informados sistema retorna mensagem com escolha de cadastrar um novo material. |
| Dados: Nome, Fornecedor, Quantidade |

#### 2.4.7 DCU Individual Ator Funcionário - Gerar Relatório

A figura abaixo representa o Caso de Uso Individual Gerar Relatório no modulo Relatórios.



|  |
| --- |
| Ator Principal: Funcionário |
| Descrição: Gerar Relatórios. |
| Pré-requisito: Ter as informações necessárias para gerar o relatório. |
| Fluxo Normal:   1. Funcionário informa qual relatório necessita. 2. Sistema gera o relatório. 3. Fim |
| Fluxo Alternativo:  2.1 - Dados fornecidos não possuem informações para gerar relatório.  2.1.1 - Sistema retorna mensagem |
| Dados: Funcionário, data, fornecedor, material, valor, estoque |

# PROJETO ORIENTADO À OBJETOS

Um objeto é uma entidade que possui atributos (características), comportamentos, responsabilidades e se relaciona com ele mesmo ou com outros objetos por meio de mensagens. Podem ser concretos ou conceituais (GÓES, 2014).

Guedes (2014) explica que através de classificação e abstração conseguimos definir classes, ou seja, grupo de objetos, sendo que cada objeto é um exemplo de um determinado grupo, possuindo as mesmas características e os mesmos comportamentos de qualquer objeto do grupo em questão.

De acordo com Sbrocco (2014) existem diversas vantagens na utilização do paradigma orientado a objetos, tais como proporcionar maior facilidade de manutenção dos sistemas, maior índice de reaproveitamento do código, e menor tempo para desenvolvimento dos programas em virtude dos códigos serem menores pela utilização do mecanismo de herança existente na orientação a objetos.

No paradigma orientado a objetos é preciso escolher o que é importante e o que pode ser omitido, seguindo-se um ponto de vista, decompondo o sistema através da visão funcional (quem vai utilizar, interagir com o sistema), visão dos objetos (foco e busca da estrutura dos problemas e não apenas dos dados) e visão dos dados (os dados podem ser utilizados para mais de uma função) (SBROCCO, 2014).

Esse capítulo irá tratar da fase de projeto orientado a objetos demonstrando através de diagramas a formação das classes, seus atributos e métodos, e a interação e as atividades que serão executadas dentro do sistema.

## Diagrama de Classes

Classe é a descrição de um conjunto de objetos que compartilham os mesmos atributos, operações e relacionamentos, podem ser utilizadas para representar elementos de software, hardware e qualquer coisa que possa ser conceituada (SBROCCO, 2014).

O diagrama de classe deve ser detalhado para poder descrever com precisão o sistema a ser construído. Os sistemas orientados a objetos organizam-se ao redor de classes de objetos, que representam tanto os elementos do domínio do problema, incorporados ao modelo, como os elementos propostos para a implementação da solução (SBROCCO, 2014).

De acordo com Guedes (2014) é um dos mais importantes e mais utilizado da UML, e seu principal enfoque é permitir a visualização das classes que irão compor o sistema com seus respectivos atributos e métodos.

O diagrama de classes pode ser utilizado na fase de análise para produzir o modelo conceitual por meio do qual representamos as classes e seus atributos, sem levar em conta a implementação, com ênfase no domínio do problema, na fase de projeto é utilizada para enfoque na solução do problema, evidenciando a implementação, como os tipos de dados dos atributos, e suas operações e métodos (GÓES, 2014).

A rigor em todas as classes deve haver um método get17 e um set18 para cada atributo da classe, porém não é muito prático fazer essa representação no diagrama de classes, por deixa-lo muito extenso quando alguma classe possuir muitos atributos (GUEDES, 2014). Com base nesta argumentação será representada no diagrama de classes apenas a classe Pessoa com os métodos get e set de seus atributos. A figura 16 demonstra o diagrama de classes a ser utilizado durante a fase de projeto com enfoque na solução do problema identificado durante a fase de análise.

## Dicionário de Atributos das Classes

Dicionário de Atributos das Classes Os atributos representam características de uma classe que costumam variar de objeto para objeto, e que permitem diferenciar um objeto de outro da mesma classe (GUEDES, 2014).

Somente os atributos que são de interesse do sistema devem ser descritos em uma classe, e cada um deles ainda pode ser de um tipo diferente. O valor correspondente a todos os atributos definidos é o que chamamos de estado do objeto (SBROCCO, 2014).

De acordo com Guedes (2014) os atributos são apresentados na segunda divisão da classe e contêm, normalmente, duas informações: o nome que o identifica e eventualmente o tipo de dado que o atributo armazena, como, por exemplo, integer, float ou character.

Góes (2014) nos explica que os atributos também possuem sinais indicadores que indicam sua visibilidade, a qual define por quem uma propriedade desse atributo pode ser utilizada. Os sinais indicadores são “+ (público)”, “- (privado)”, “# (protegido) e “~(pacote)”.

O dicionário de atributos das classes irá representar essa estrutura dos atributos, e além de conter, o nome do atributo irá descrever de forma sucinta a descrição do atributo.

As tabelas abaixo irão apresentar os atributos das classes identificadas no desenvolvimento do sistema.

**3.2.1. Classe Pessoa**

|  |  |
| --- | --- |
| Classe Pessoa | |
| Atributo | Descrição |
| IdPessoa | Código da pessoa |
| nome | Nome da Pessoa |
| cpfCnpj | Dados regionais |
| DataNascimento | Data do Nascimento |
| Login | Usuario para Login |
| senha | Senha para login |

**3.2.2. Classe Super**

|  |  |
| --- | --- |
| Classe Super | |
| Atributo | Descrição |
| IdSuper | Código pessoa super |
| idPessoa | Código pessoa |
| nivelAcesso | Nivel de acesso no sistema |
| situação | Situação do super |
| permiteLogin | Permissão para logar |

**3.2.3 Classe Controlador**

|  |  |
| --- | --- |
| Classe Controlador | |
| Atributo | Descrição |
| idControlador | Código do Controlador |
| idPessoa | Código da pessoa |
| observação | observação |
| situação | Situação do controlado |
| permiteLogin | Permissão para logar |

**3.2.4 Classe Funcionário**

|  |  |
| --- | --- |
| Classe Funcionário | |
| Atributo | Descrição |
| idFuncionário | Código Funcionario |
| idPessoa | Código Pessoa |
| observação | Observação |
| situação | Situação do Funcionário |
| permiteLogin | Permissão para logar |

**3.2.5 Classe Tipo Produto**

|  |  |
| --- | --- |
| Classe TipoProduto | |
| Atributo | Descrição |
| idTipoProduto | Código do tipo do Produto |
| descrição | Descrição do produto |

**3.2.6 Classe Unidade Medida**

|  |  |
| --- | --- |
| Classe UnidadeMedida | |
| Atributo | Descrição |
| idUnidadeMedida | Código da unicada de medida |
| descrição | Descrição da unidade de medida |
| sigla | Sigla da unidade de medida |

**3.2.7 Classe Produto**

|  |  |
| --- | --- |
| Classe Produto | |
| Atributo | Descrição |
| idProduto | Código do produto |
| nomeProduto | Nome do produto |
| ultimoPrecoPagp | Valor do último preço pago |
| saldoAtual | Quantidade |
| IdTipoProduto | Código do tipo do Produto |
| idunidadeMedida | Código da Unidade de Medida |

**3.2.8 Classe Requisição Mercadoria**

|  |  |
| --- | --- |
| Classe RequisiçãoMercadoria | |
| Atributo | Descriçaõ |
| idRequisicaoMercadoria | Código da Requisição |
| descricaoProduto | Descrição do produto adquirido |
| Data | Data |
| quantidade | Quantidade adquirida |
| idproduto | Código do produto |
| idFuncionário | Código do Funcionário |
| aprovadoReprovado | Status Requisição |

**3.2.9 Classe Nota Fiscal**

|  |  |
| --- | --- |
| Classe NotaFiscal | |
| Atributo | Descrição |
| idNotaFiscal | Código da nota fiscal |
| nroNotaFiscal | Número da Nota Fiscal |
| data | Data |
| valorNota | Valor da Nota |
| idProduto | Código do produto |
| idFuncionário | Código Funcionario |
| aprovadoReprovado | Status Nota |

**3.2.10 Classe RecebimentoMercadoria**

|  |  |
| --- | --- |
| Classe RecebiimentoMercadoria | |
| Atributo | Descrição |
| idRecebimentoMercadoria | Código do recebimento da mercadoria |
| origem | Origem da mercadoria |
| data | data |
| valorMercadoria | Valor da Mercadoria |
| idProduto | Código do Produto |
| idFuncionario | Código Funcionario |
| idNotaFiscal | Código da nota fiscal |
| aprovadoReprovado | Status |

**3.2.11 Classe AcertoEstoque**

|  |  |
| --- | --- |
| Classe AceretoEstoque | |
| Atributo | Descrição |
| idAcertoEstoque | Código acerto Estoque |
| data | data |
| idProduto | Código do produto |
| IdFuncionário | Código do Funcionário |

## Diagramas de Classes Camada Controller

O Controlador (Controller) é responsável por receber as requisições do usuário montar os objetos correspondentes e passá-los ao Model para que faça o que tem que ser feito. Depois de receber o retorno da operação executada, o controlador apresenta o resultado para o usuário, geralmente o redirecionando para a View (CORDEIRO, 2012).

Os diagramas das figuras abaixo representam as classes na camada Controller.

Diagrama de Classe Controller Login.

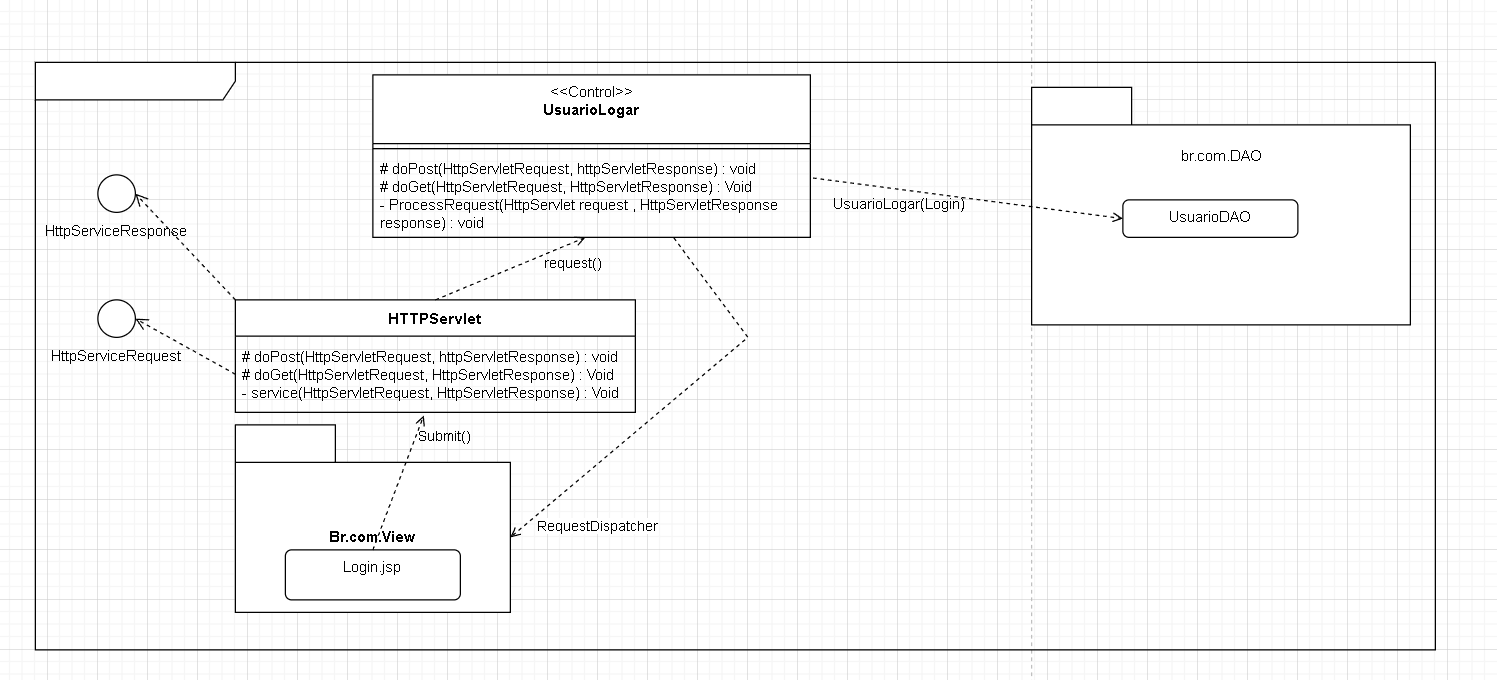


Diagrama de Classe Controller Listar Compras.

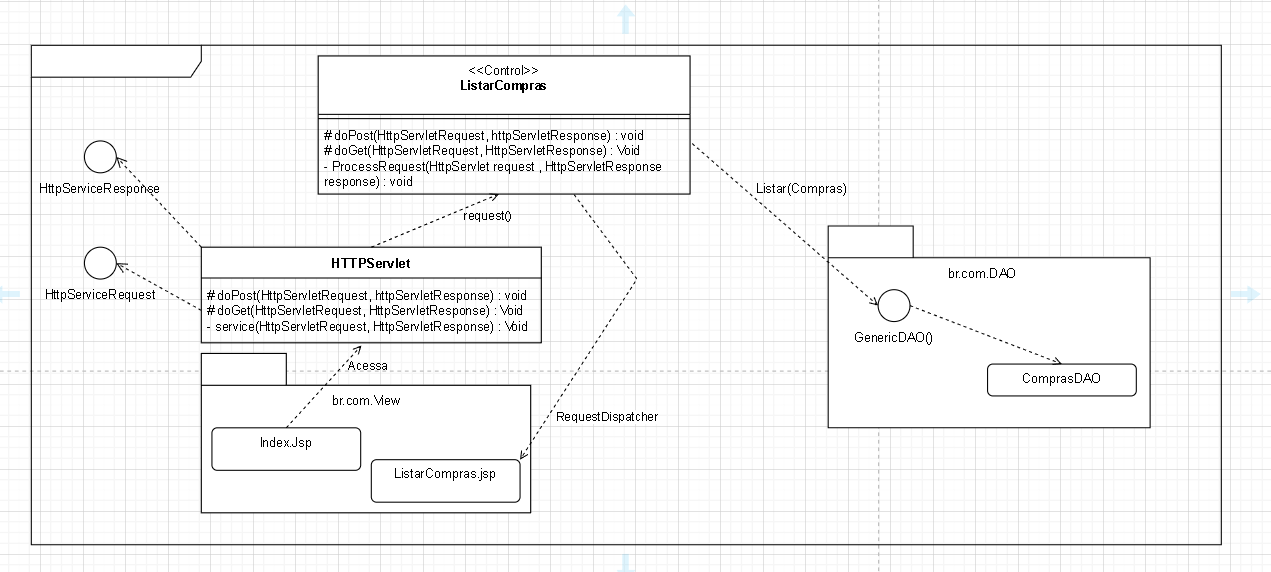


Diagrama de Classe Controller Listar Estoque.

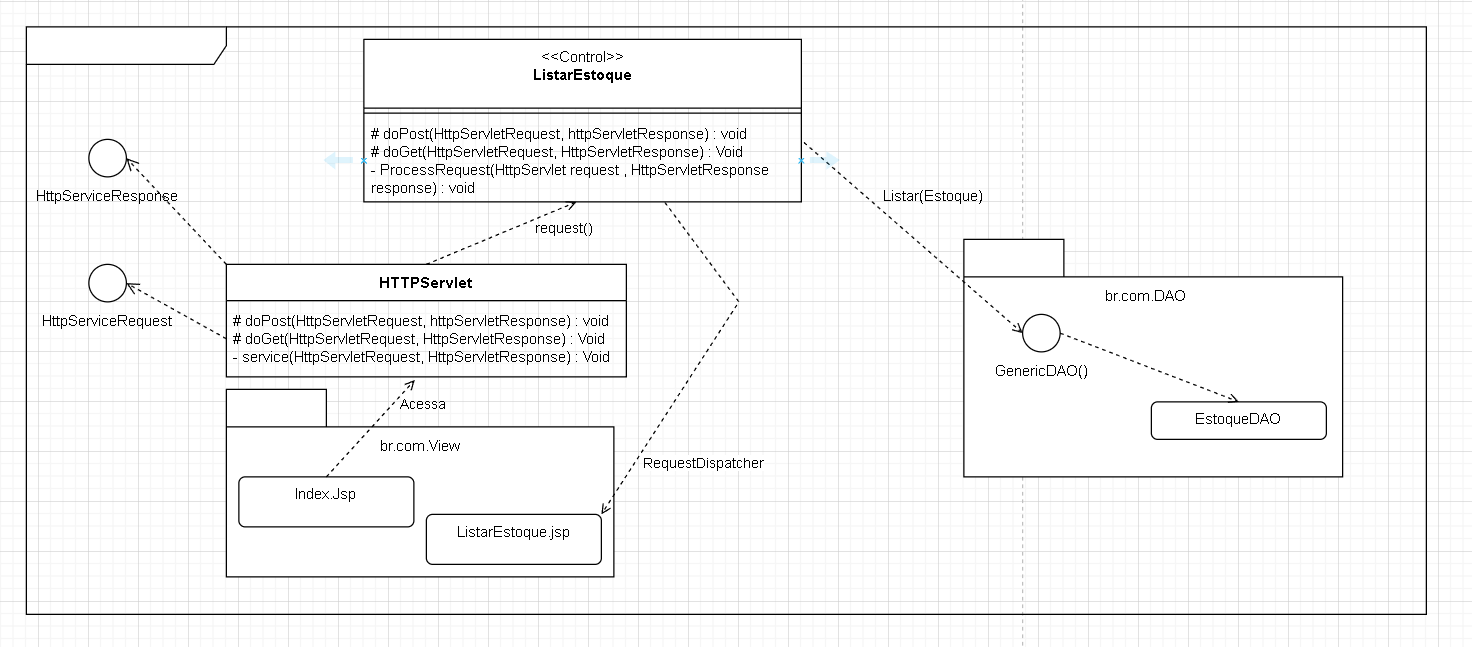


Diagrama de Classe Controller Listar Fornecedor.

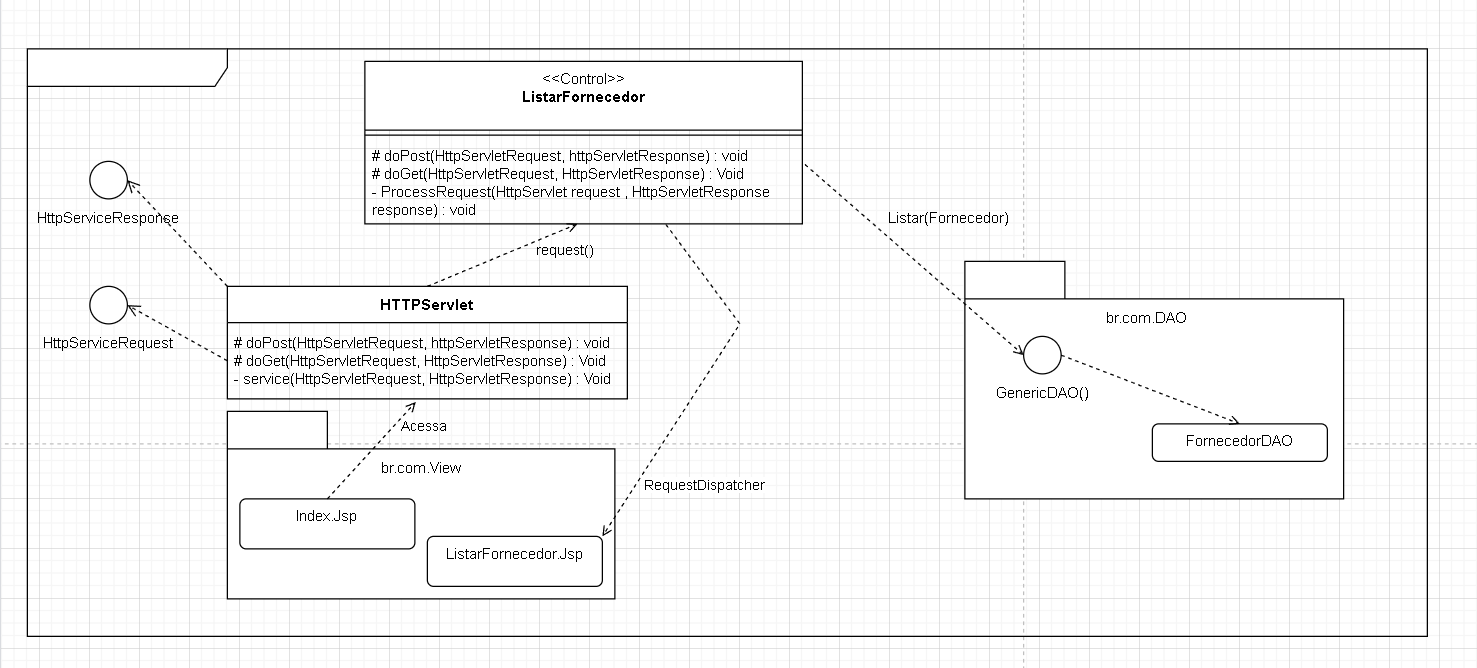


Diagrama de Classe Controller Alterar Fornecedor.

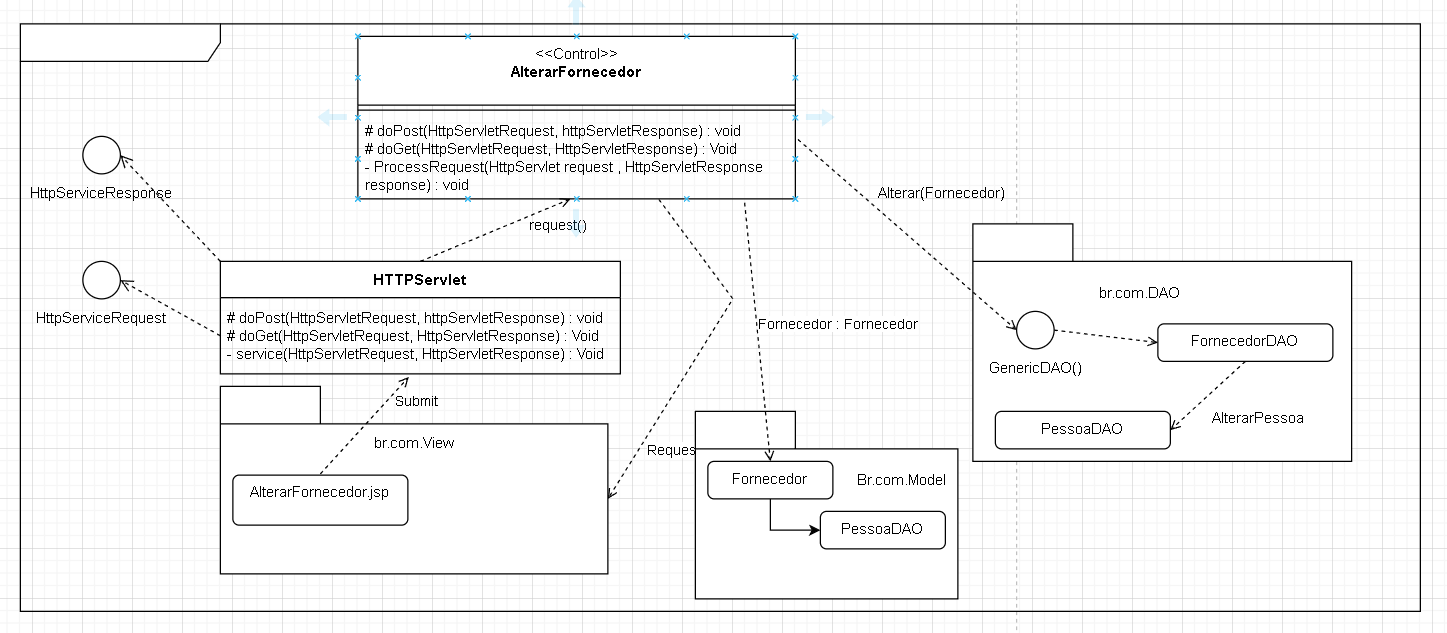


Diagrama de Classe Controller Inativar Fornecedor.

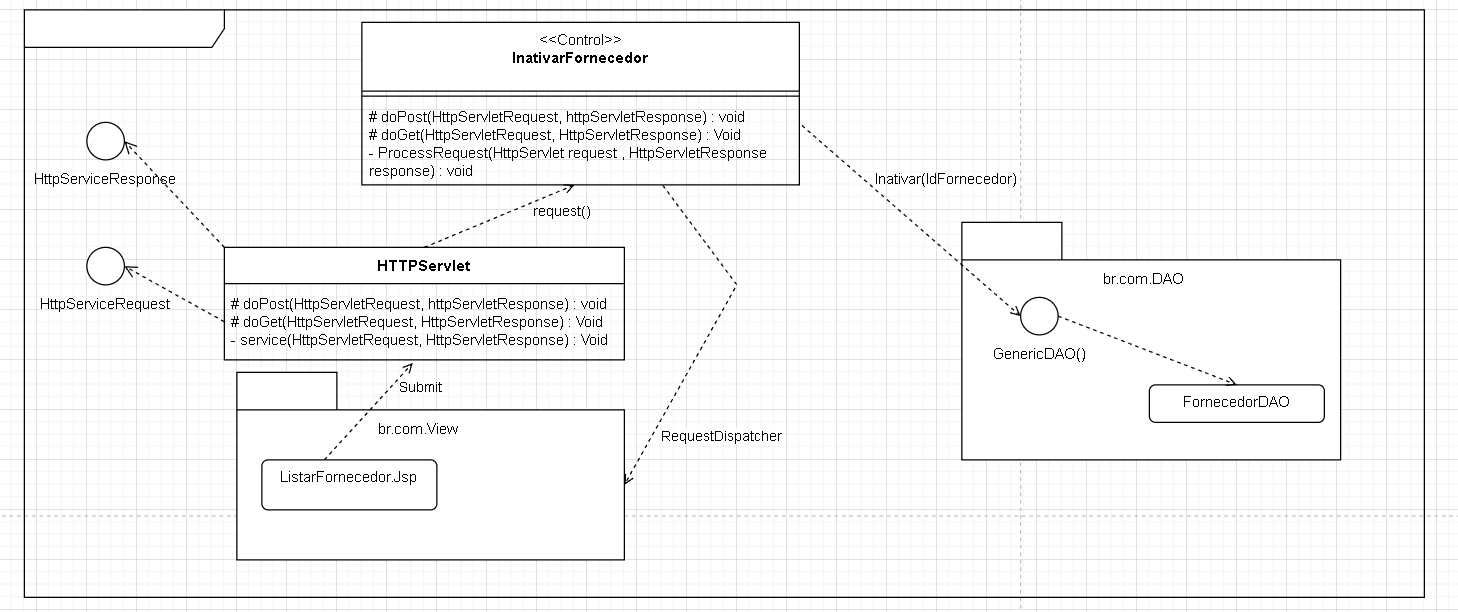


Diagrama de Classe Controller Cadastrar Fornecedor.

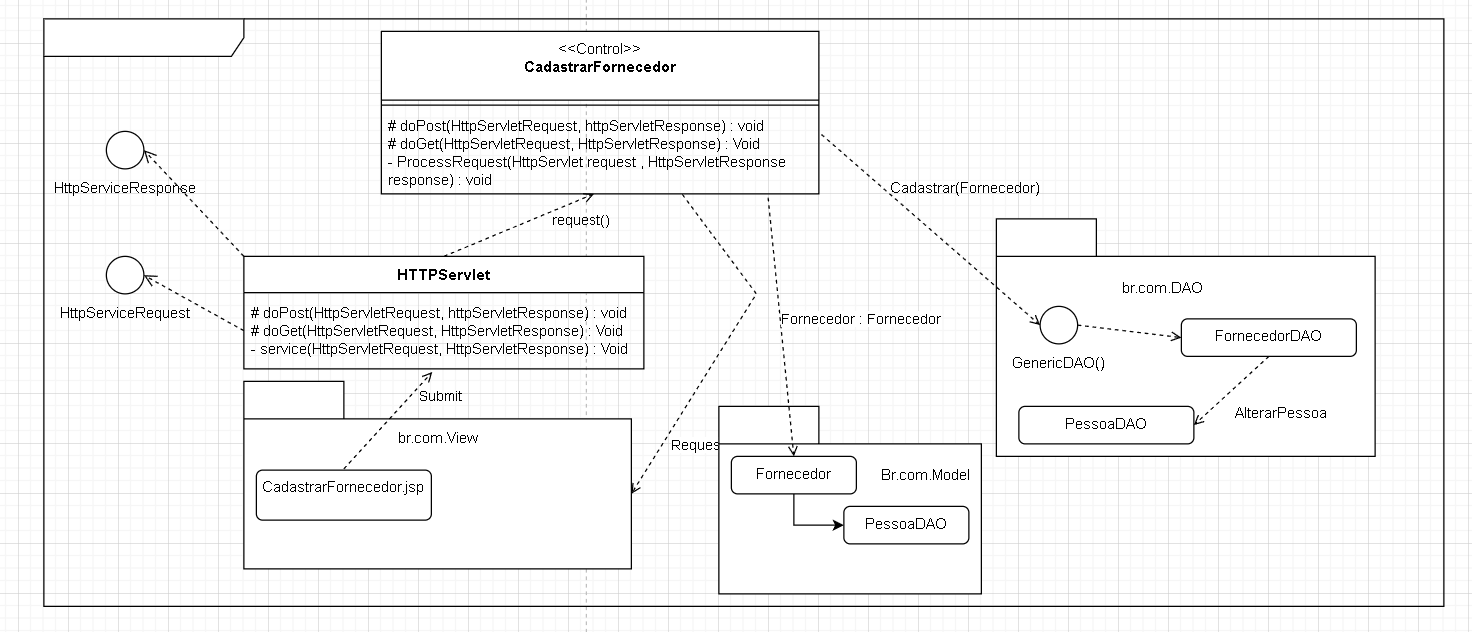


Diagrama de Classe Controller Listar Funcionário.

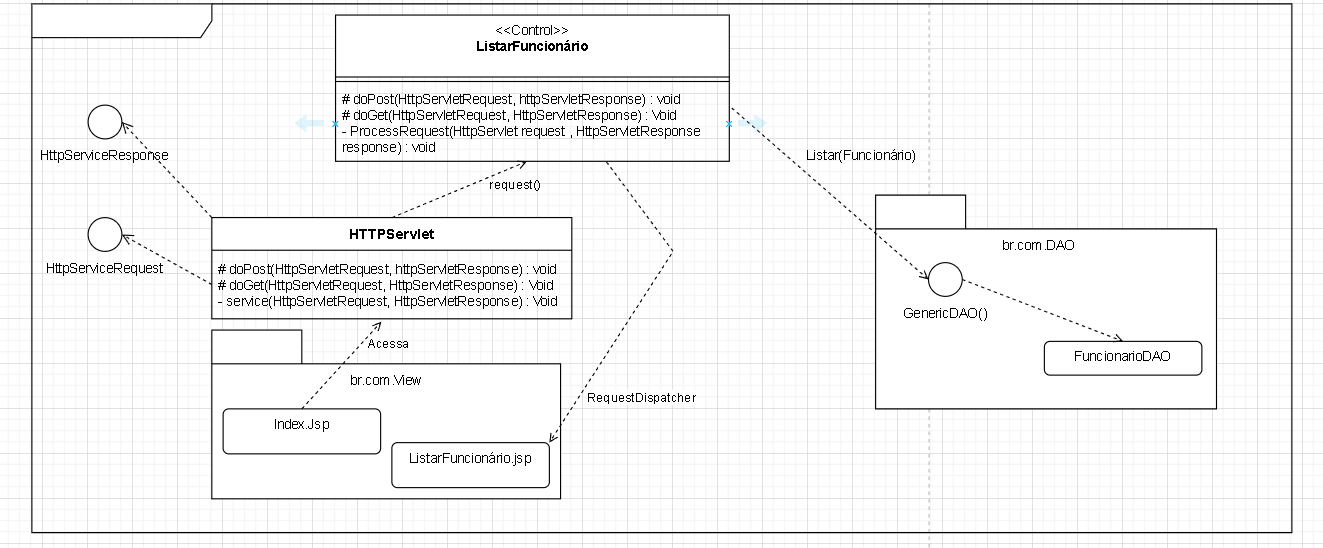


Diagrama de Classe Controller Alterar Funcionário.

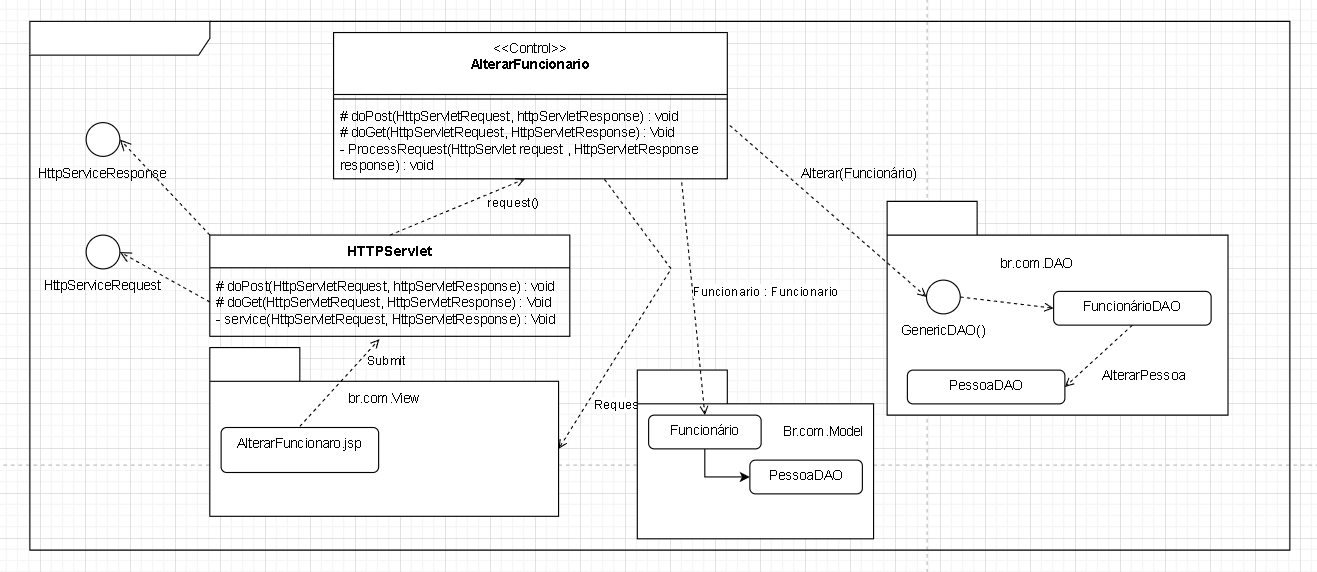


Diagrama de Classe Controller Inativar Funcionário.

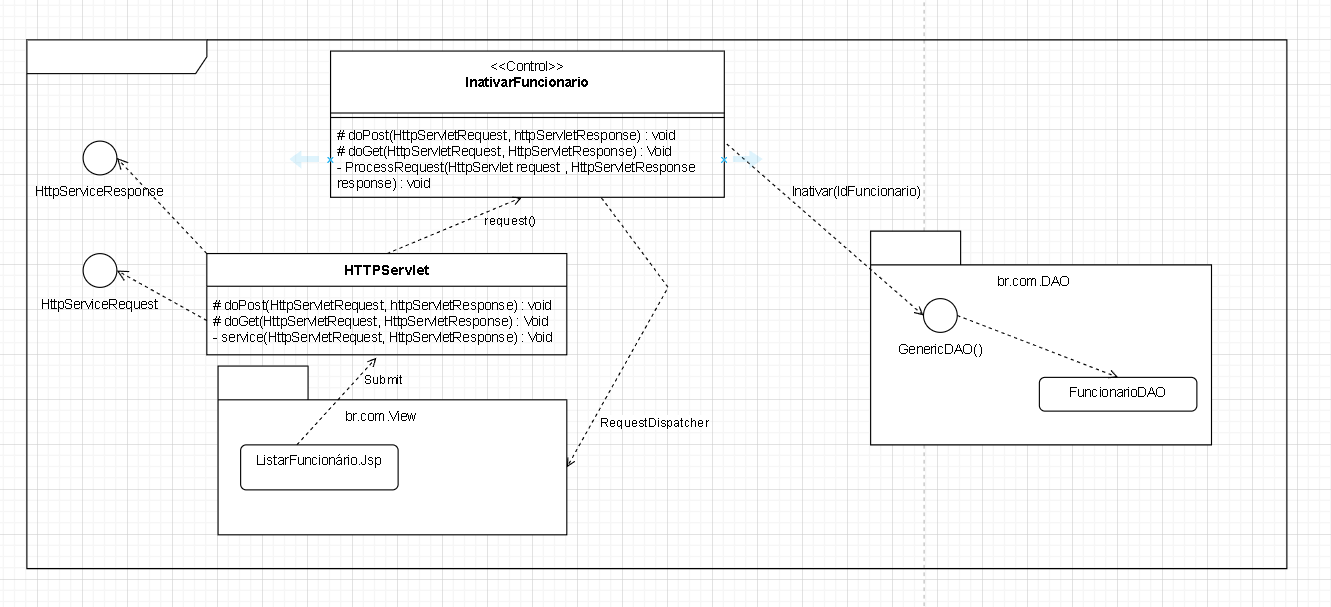


Diagrama de Classe Controller Cadastrar Funcionário.

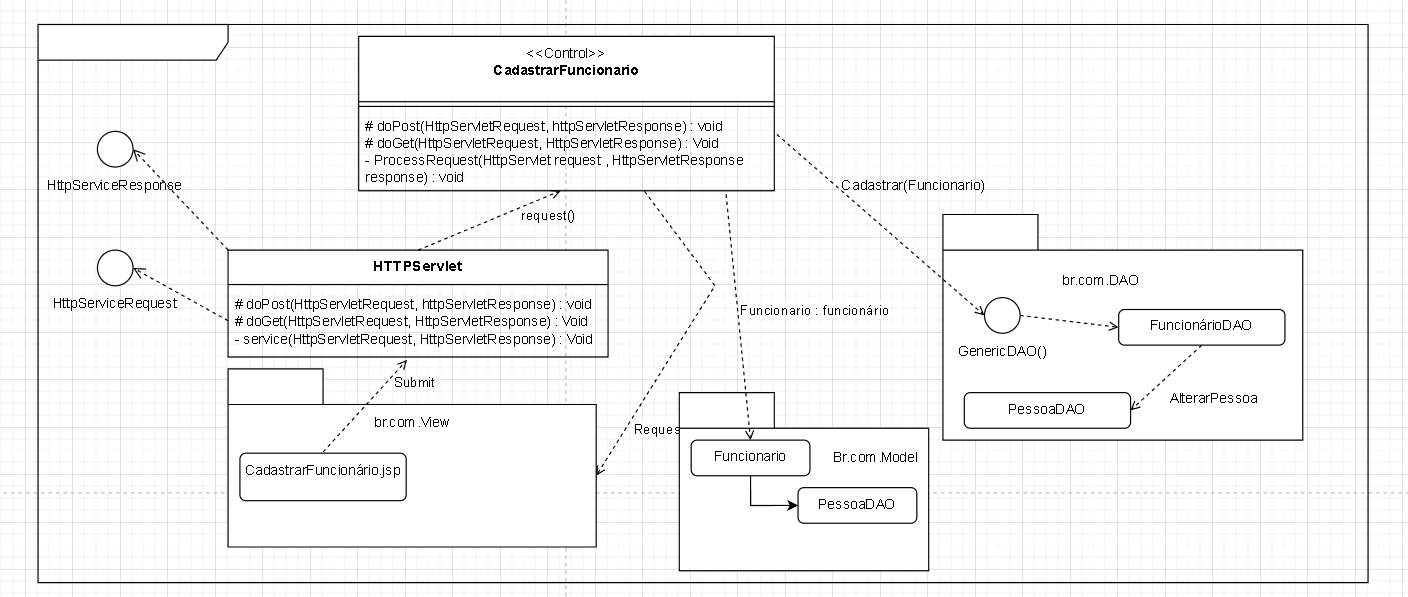


Diagrama de Classe Controller Listar Materiais.

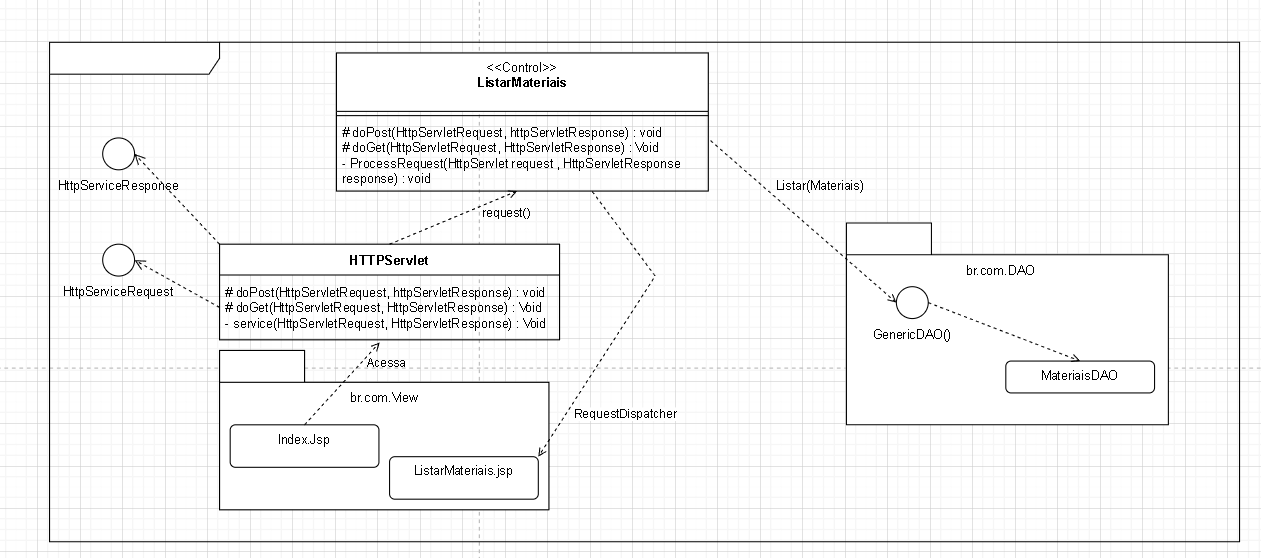


Diagrama de Classe Controller Cadastrar Materiais.

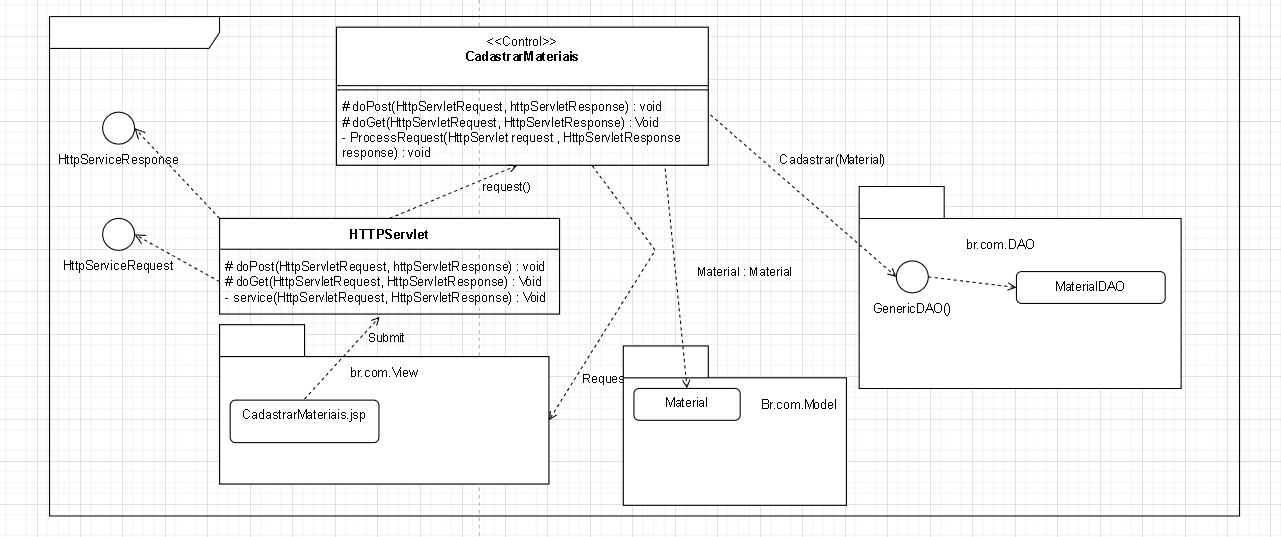


Diagrama de Classe Controller Alterar Materiais.

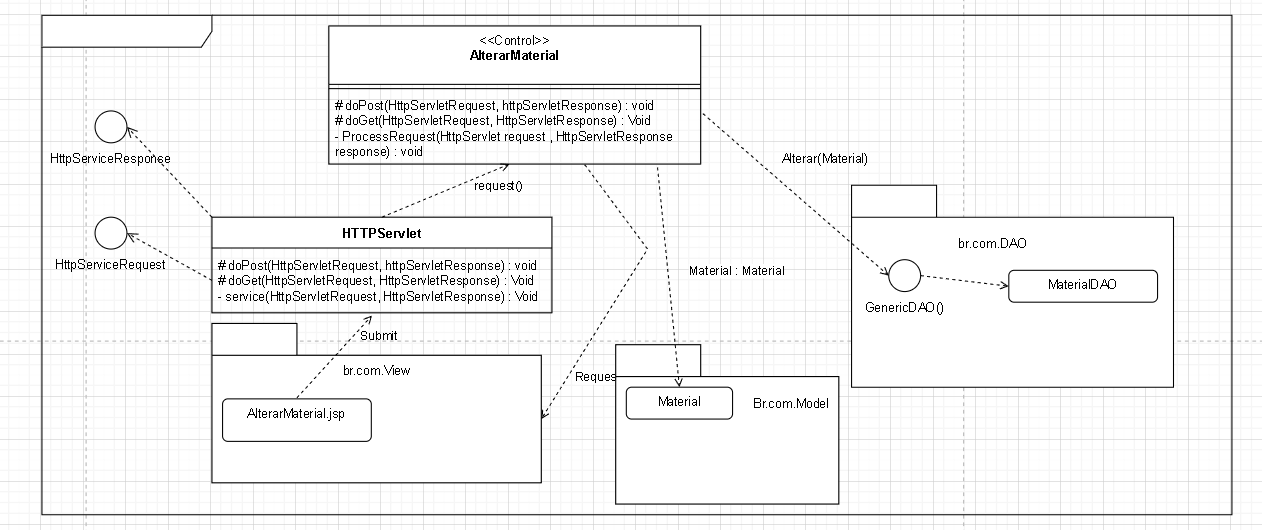


Diagrama de Classe Controller Deletar Materiais.

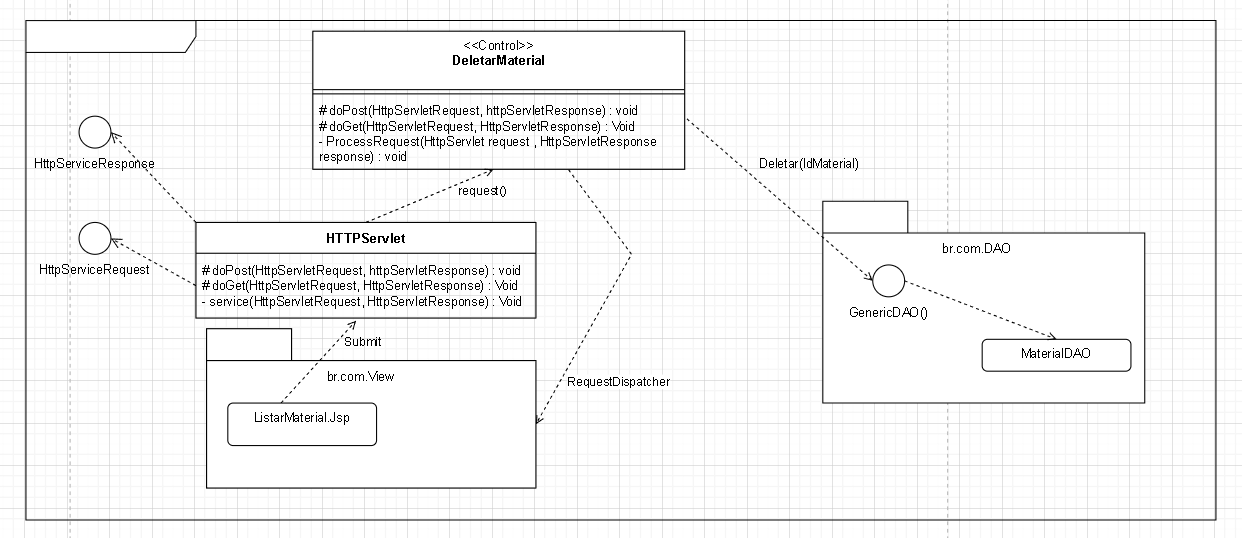
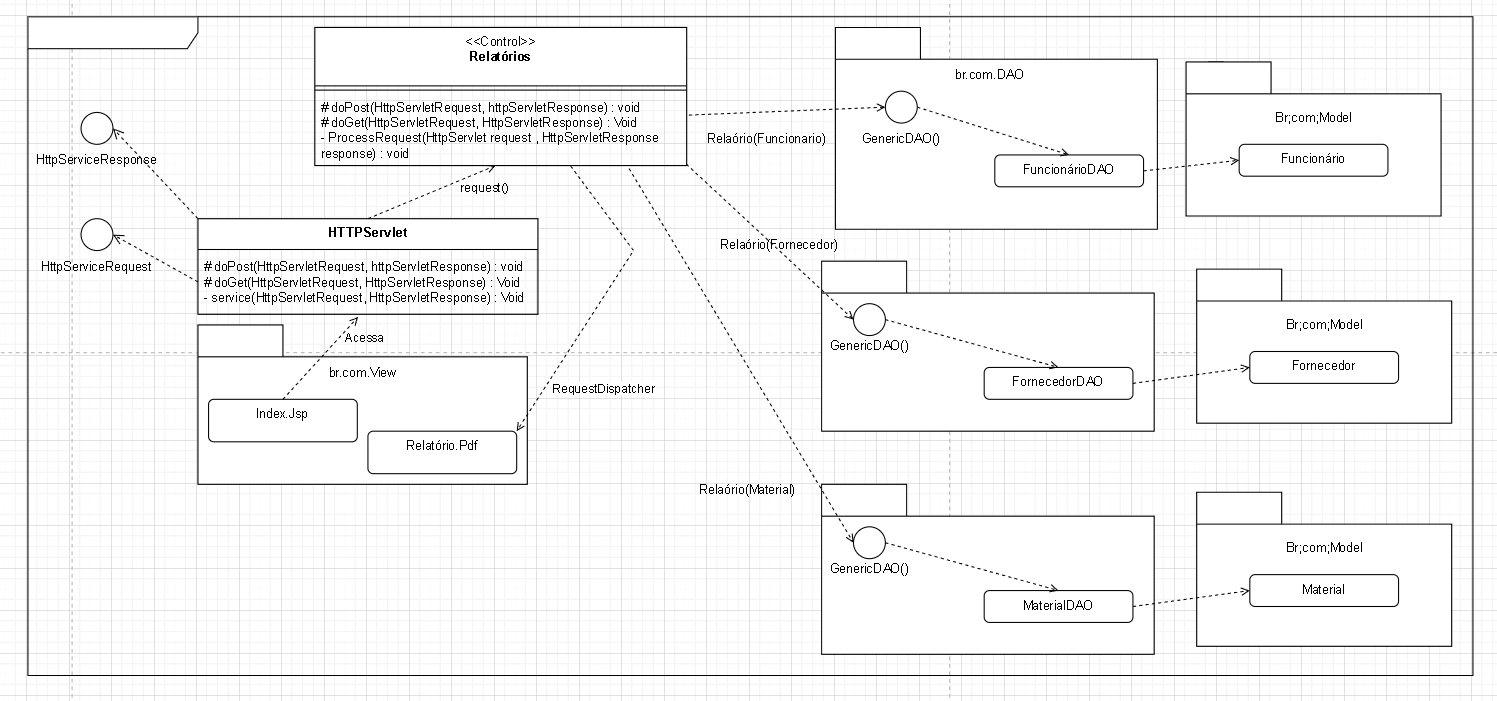


Diagrama de Classe Controller Gerar Relatórios.



## Diagramas de Classes Camada DAO

De acordo com Gonçalves (2007) o padrão DAO (Data Access Object) é o padrão mais utilizado para acesso a dados contidos em um banco de dados. Sempre que você precisa acessar um banco de dados que está mantendo seu modelo de objetos, é melhor empregar o padrão DAO.

Tipicamente um DAO inclui métodos para inserir, selecionar, atualizar e excluir objetos de um banco de dados. Dependendo de como você implementa o padrão DAO, você poderá ter um DAO para cada classe de objetos em sua aplicação ou poderá ter um único DAO que é responsável por todos os seus objetos (GONÇALVES, 2007).

Os Diagramas das figuras abaixo representam as classes na camada DAO.

Diagrama de Classe Camada DAO Classe Pessoa.

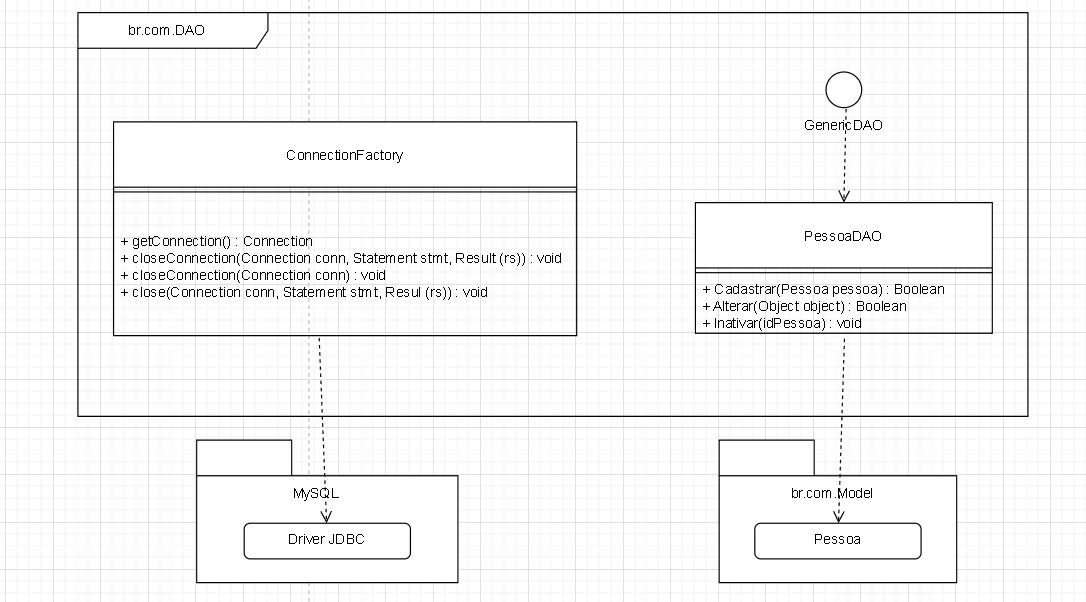


Diagrama de Classe Camada DAO Classe Fornecedor.

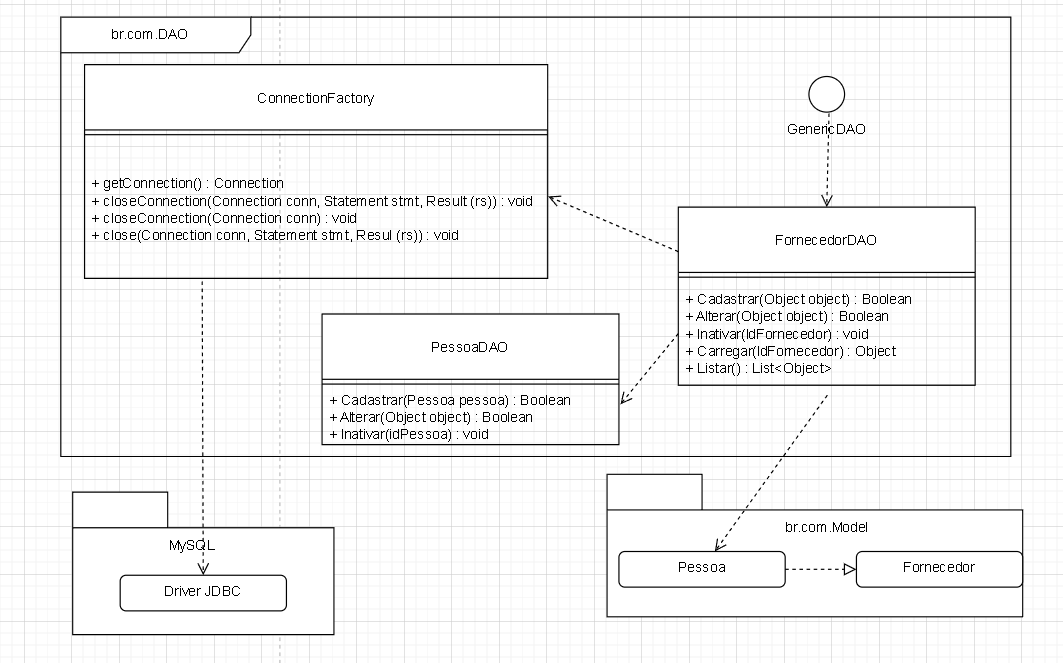


Diagrama de Classe Camada DAO Classe Funcionário.

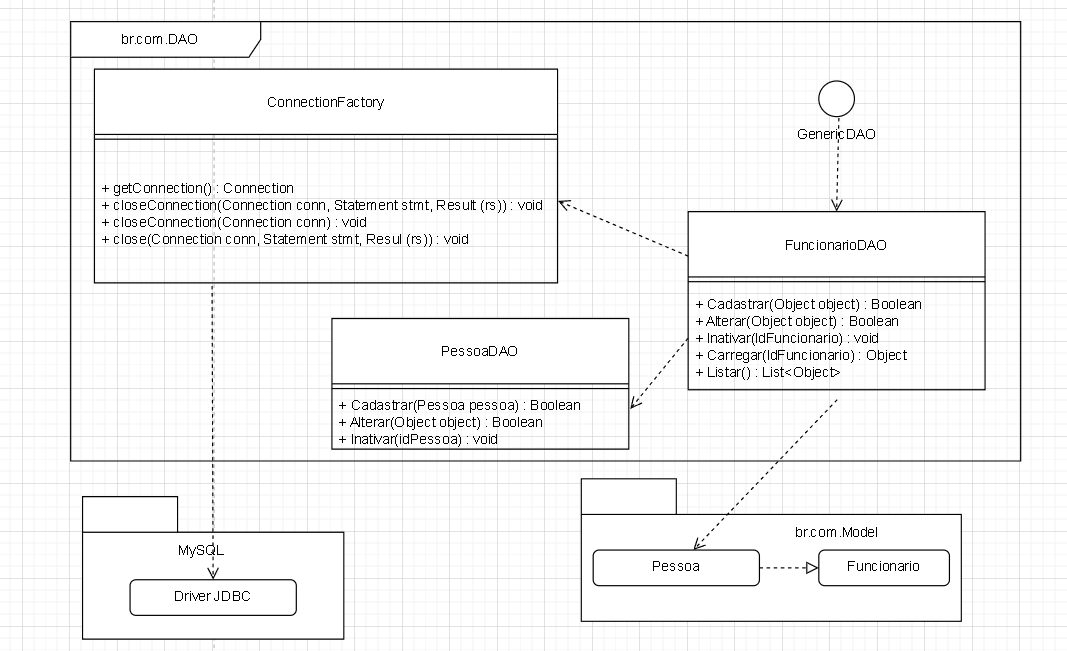


Diagrama de Classe Camada DAO Classe Compras.

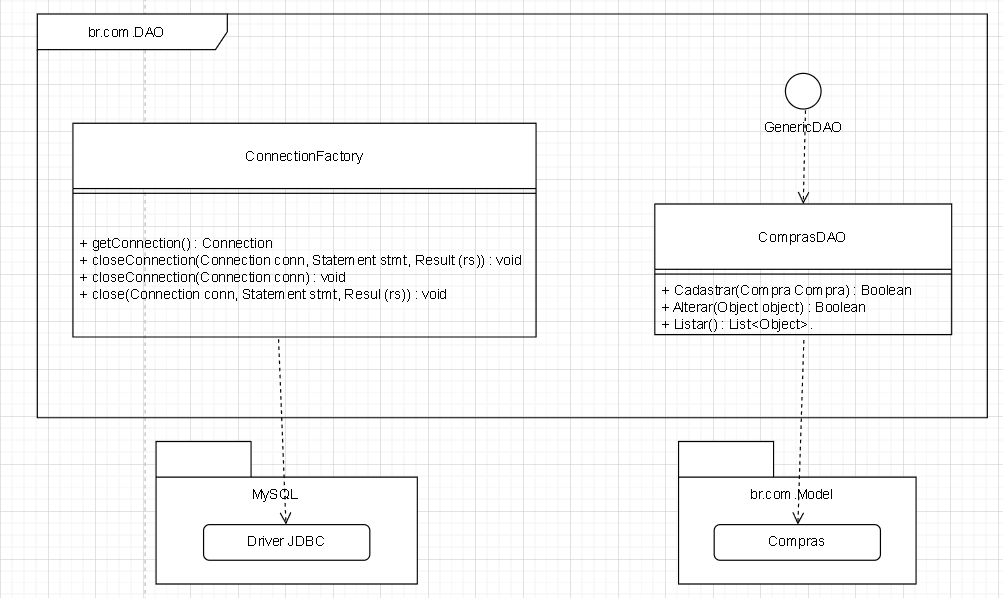


Diagrama de Classe Camada DAO Classe Estoque.

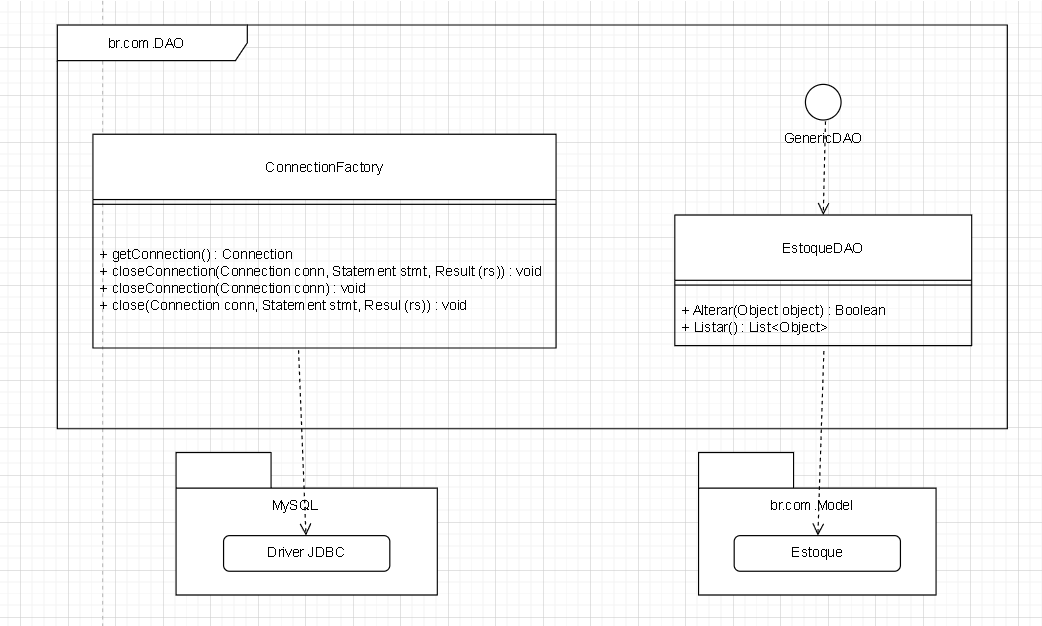
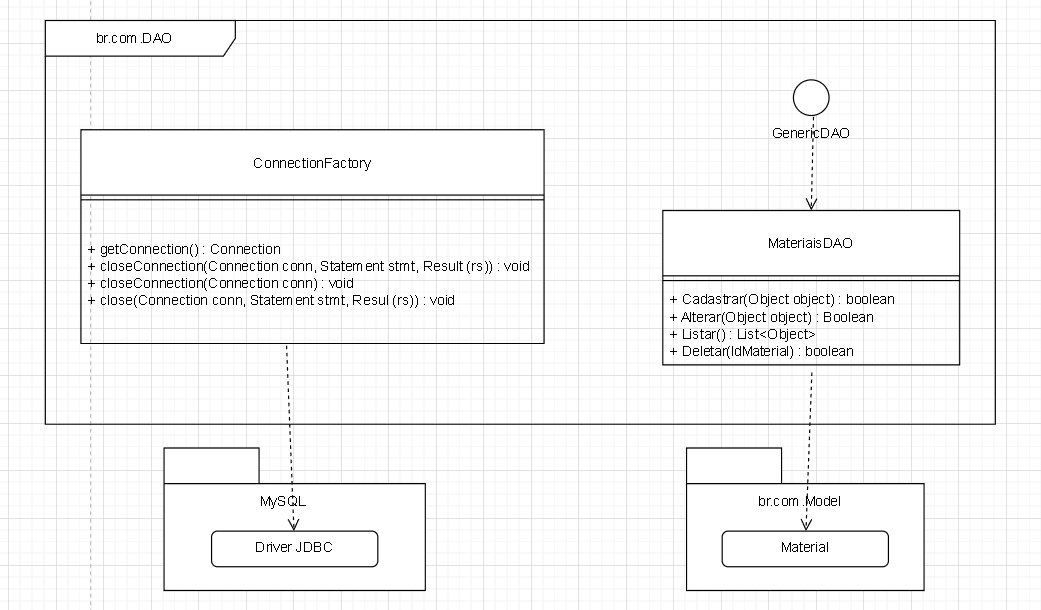


Diagrama de Classe Camada DAO Classe Materiais.



## Projeto Físico do Banco de Dados

No projeto físico do banco de dados estão contidos os scripts para criação das

tabelas que serão utilizadas para o desenvolvimento do sistema.

Para desenvolvimento do projeto foi utilizado o PostgreSQL (Windows).

O PostgreSQL é um sistema gerenciador de banco de dados (SGBD) relacional, é um software livre para banco de dados bastante usado em aplicações por sua versatilidade e por ser suportado em diversas plataformas e diferentes linguagens.

#### 3.5.1. Tabela Pessoa

O código abaixo representa a estrutura SQL para criação da tabela Pessoa.

create table Pessoa(

idPessoa serial primarykey,

nomevarchar(100) notnull,

cpfcnpjvarchar(14) notnullunique,

datanascimento date,

login varchar(20),

senha varchar(20)

);

#### 3.5.2. Tabela Super

O código abaixo representa a estrutura SQL para criação da tabela Super.

createtableSuper (

idSuper serial primarykey,

idUsuarionumericunique,

nivelAcessovarchar(1),

situacaovarchar(1),

permiteloginvarchar(1),

constraintfk\_Super\_Usuarioforeignkey (idPessoa) references Pessoa

);

**3.5.3. Tabela Controlador**

O código abaixo representa a estrutura SQL para criação da tabela Controlador.

create table Controlador (

idControlador serial primarykey,

idUsuariointunique,

observacaovarchar(100),

situacaovarchar(1),

permiteloginvarchar(1),

constraintfk\_Controlador\_Usuarioforeignkey (idPessoa) references Pessoa

);

**3.5.4. Tabela Servidor**

O código abaixo representa a estrutura SQL para criação da tabela Servidor..

create table Servidor (

idServidor serial primarykey,

idUsuariointunique,

enderecowebvarchar(100),

situacaovarchar(1),

permiteloginvarchar(1),

constraintfk\_Super\_Usuarioforeignkey (idPessoa) references Pessoa

);

**3.5.5. Tabela TipoProduto**

O código abaixo representa a estrutura SQL para criação da tabelatipoProduto..

createtabletipoProduto (

idTipoProduto serial primarykey,

descricaovarchar(100)

);

**3.5.6. Tabela UnidadeMedida**

O código abaixo representa a estrutura SQL para criação da tabela UnidadeMedida .

createtableunidadeMedida (

idunidademedida serial primarykey,

descricaovarchar(100),

sigla varchar(2)

);

**3.5.7. Tabela Produto**

O código abaixo representa a estrutura SQL para criação da tabela Produto.

create table produto (

idproduto serial primary key,

nomeprodutovarchar(100),

ultimoprecopagonumeric,

saldoatualnumeric,

idtipoprodutonumeric,

idunidademedidanumeric,

foreignkey (idtipoproduto) referencestipoproduto (idtipoproduto),

foreignkey (idunidademedida) referencesunidademedida (idunidademedida)

);

**3.5.8. Tabela RequisiçãoMercadoria**

O código abaixo representa a estrutura SQL para criação da tabela RequisiçãoMercadoria.

createtableRequisiçãoMercadoria (

idRequisiçãoMercadoria serial primarykey,

descricaoProdutovarchar(100),

data varchar(10),

quantidadenumeric,

valormercadorianumeric,

idprodutonumeric,

idUsuarionumeric,

aprovadoReprovaVarchar(1),

foreignkey (idproduto) references produto (idproduto),

foreignkey (idUsuario) referencesUsuario (idUsuario)

);

**3.5.9. Tabela NotaFiscal**

O código abaixo representa a estrutura SQL para criação da tabela NotaFiscal.

createtableNotaFiscal (

idNotaFiscal serial primarykey,

nroNotanumeric,

data varchar(10),

valorNotanumeric,

idprodutonumeric,

idUsuarionumeric,

aprovadoReprovaVarchar(1),

foreignkey (idproduto) references produto (idproduto),

foreignkey (idUsuario) referencesUsuario (idUsuario)

);

**3.5.10. Tabela RecebimentoMercadoria**

O código abaixo representa a estrutura SQL para criação da tabela RecebimentoMercadoria.

createtableRecebimentoMercadoria (

idRecebimentoMercadoria serial primarykey,

origem varchar(100),

data varchar(10),

valormercadorianumeric,

idprodutonumeric,

idUsuarionumeric,

idNotaFiscalnumeric,

aprovadoReprovaVarchar(1),

foreignkey (idproduto) references produto (idproduto),

foreignkey (idNotaFiscal) referencesNotaFiscal (idNotaFiscal),

foreignkey (idUsuario) referencesUsuario (idUsuario)

);

**3.5.11. Tabela AcertoEstoque**

O código abaixo representa a estrutura SQL para criação da tabela AcertoEstoque.

create table AcertoEstoque (

IdAcertoEstoque serial primary key,

data varchar(10),

idprodutonumeric,

idUsuarionumeric,

foreignkey (idproduto) references produto (idproduto),

foreignkey (idUsuario) referencesUsuario (idUsuario)

);

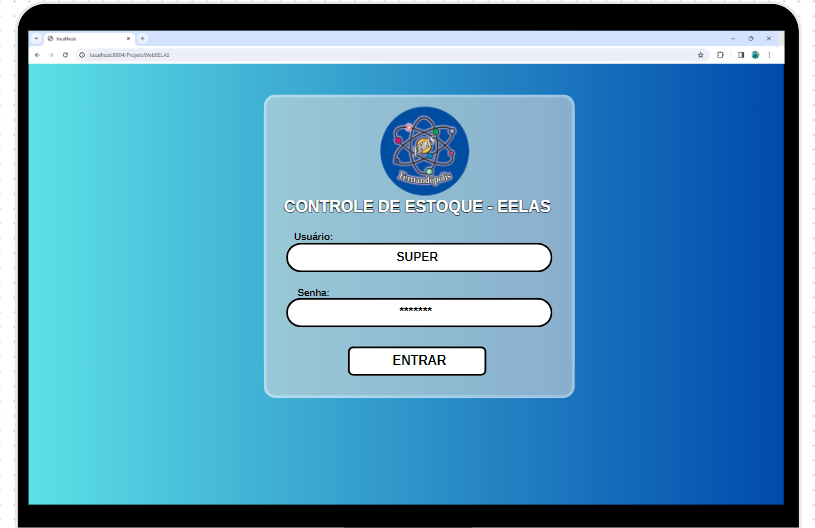
## Especificação do Layout de Telas e Relatórios

Nesse capitulo se encontra a representação das telas do sistema, bem como suas determinadas descrições.

Para o desenvolvimento da interface gráfica foi utilizado o Programa CANVA.

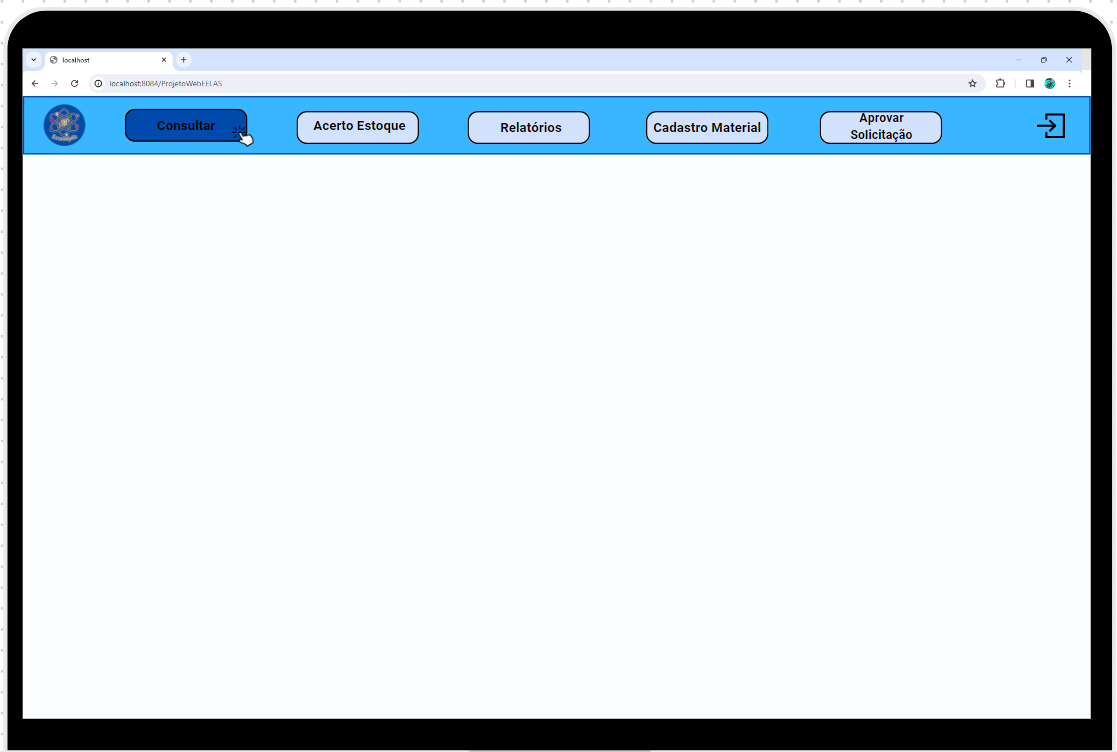
#### 3.6.1 – Tela de Login– Controle de estoque EELAS

Esta figura apresenta a tela principal do sistema onde o usuário poderá efetuar o login para ter acesso exclusivo às páginas do sistema vinculadas ao tipo de usuário ao qual ele pertença.

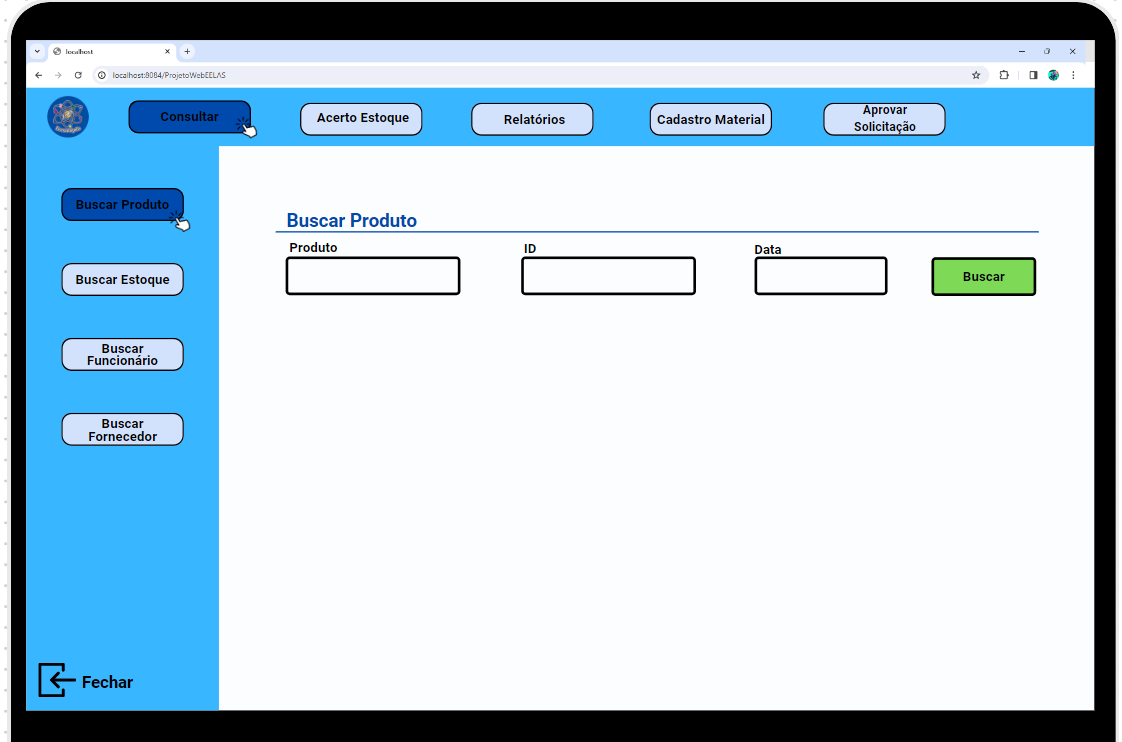


#### 3.6.2 Tela de acessos – Pessoa Super.

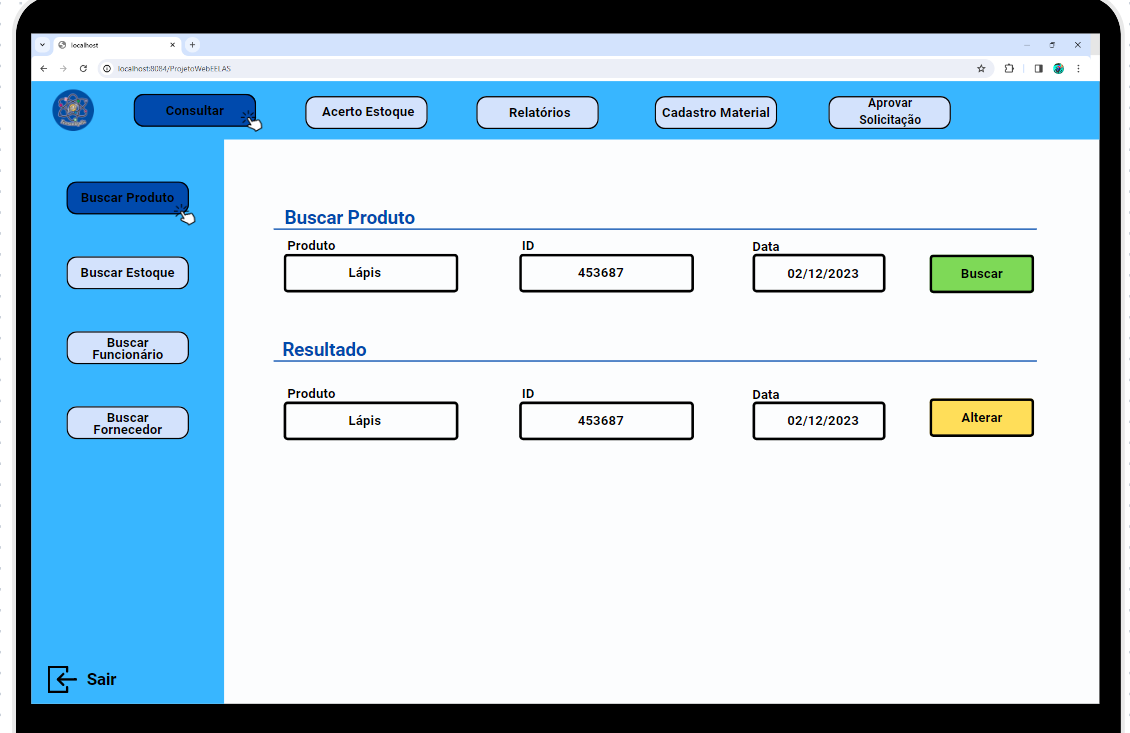
Nessa figura apresenta a tela de acessos que é utilizada pelo usuário Super do sistema. E como demonstração cliquei no módulo consultar.



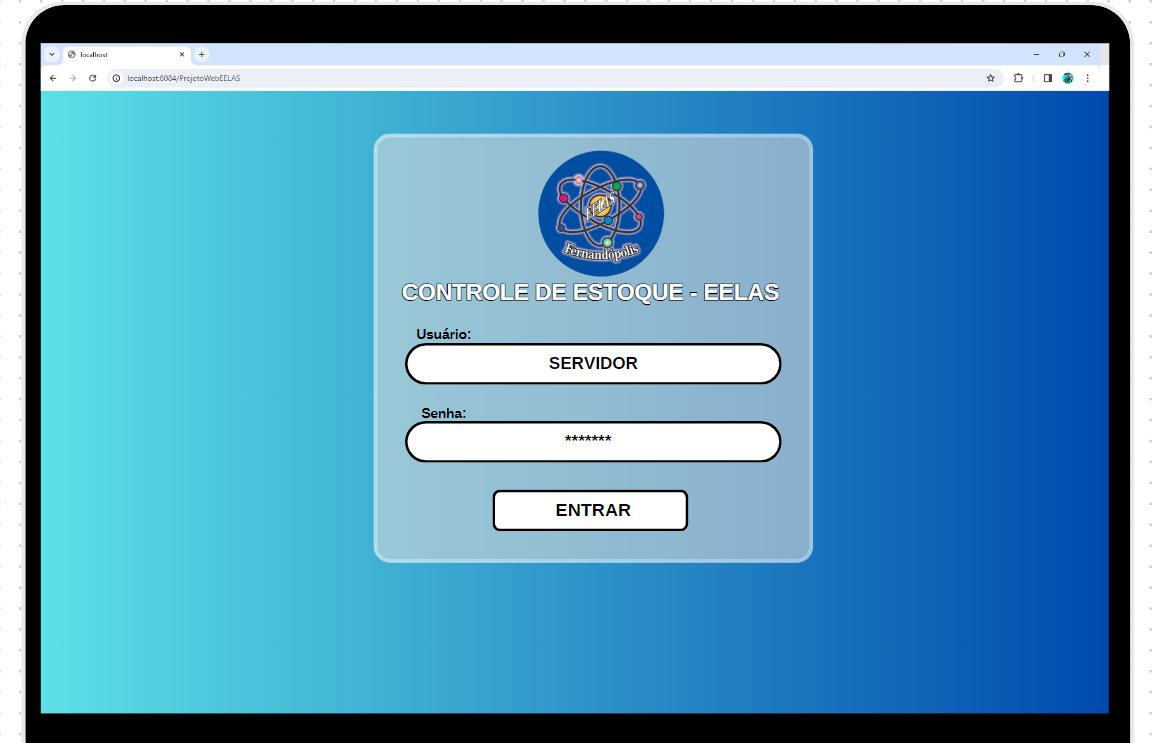
#### 3.6.3 Tela busca – Pessoa Super.



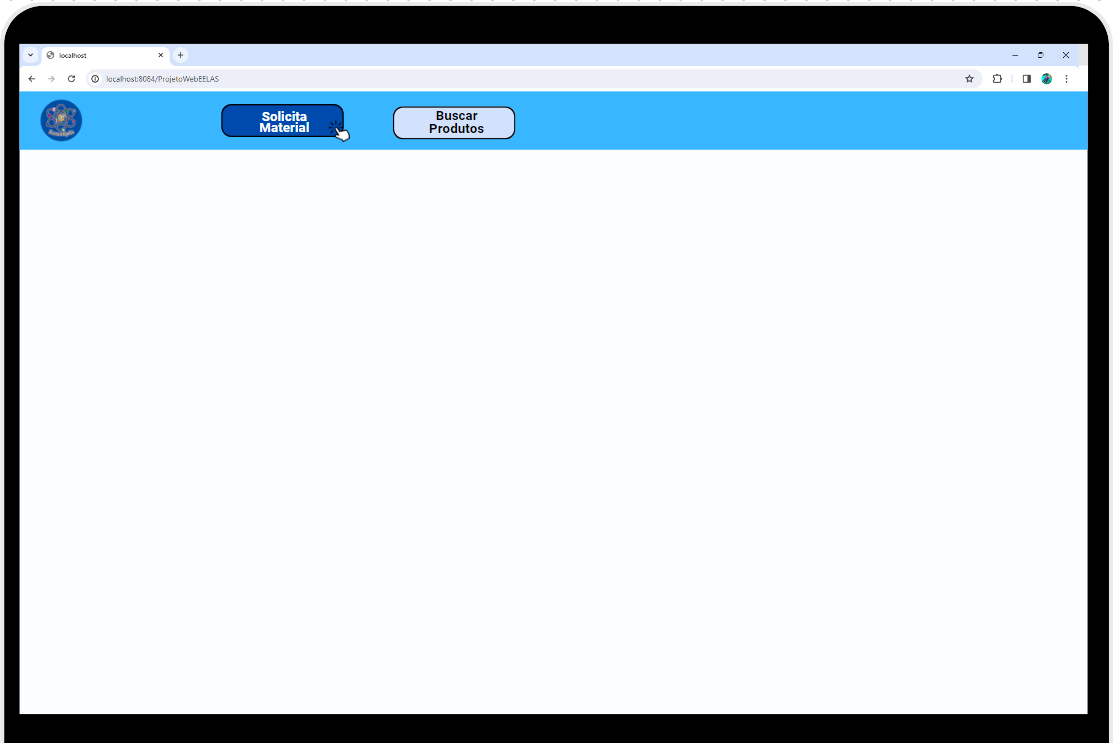
#### 3.6.4 Tela de Resultado da Busca – Pessoa Super.

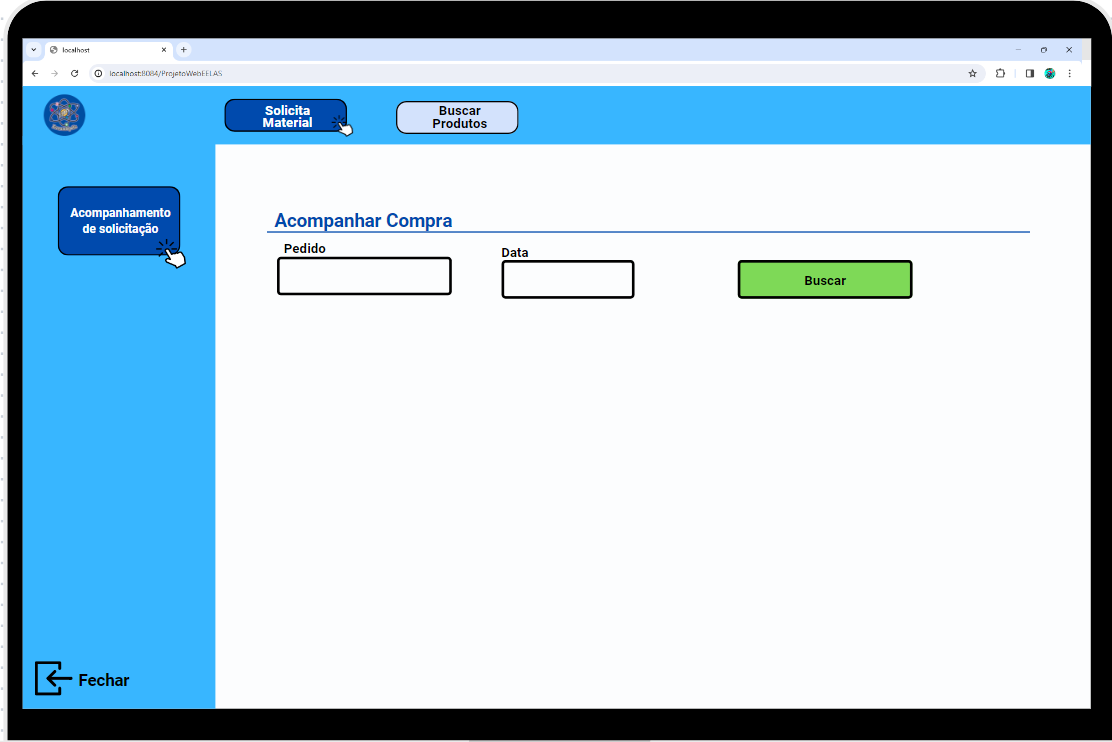


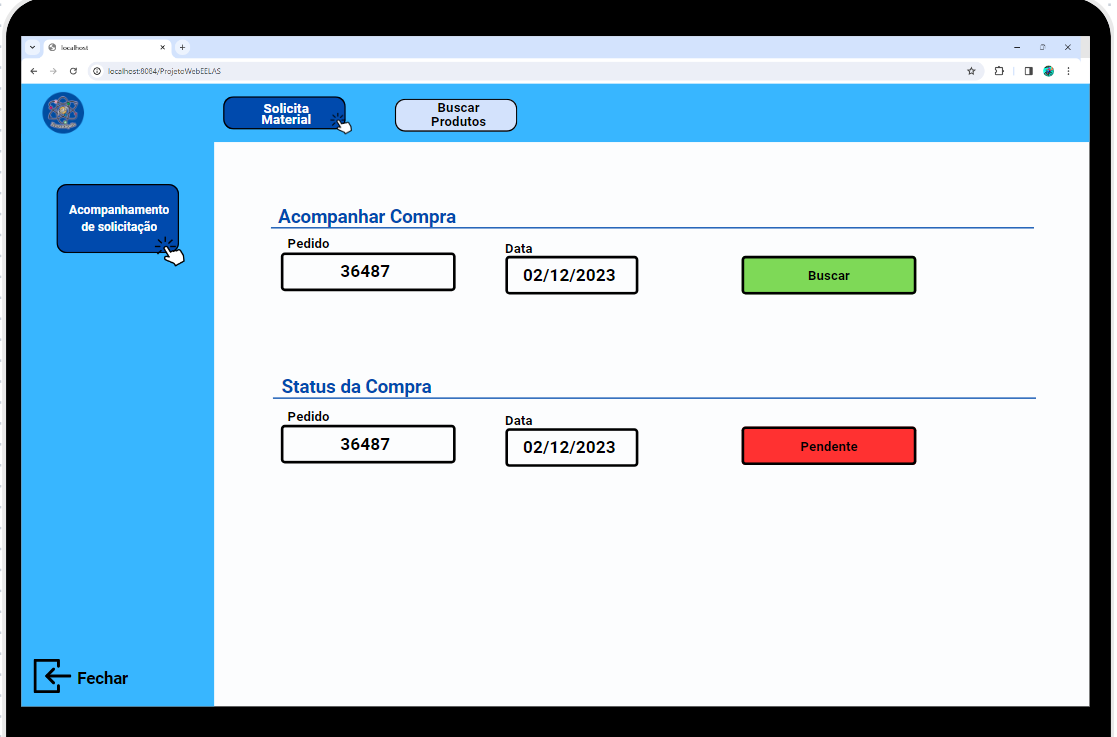
#### 3.6.5 Tela de Login – Pessoa Servidor

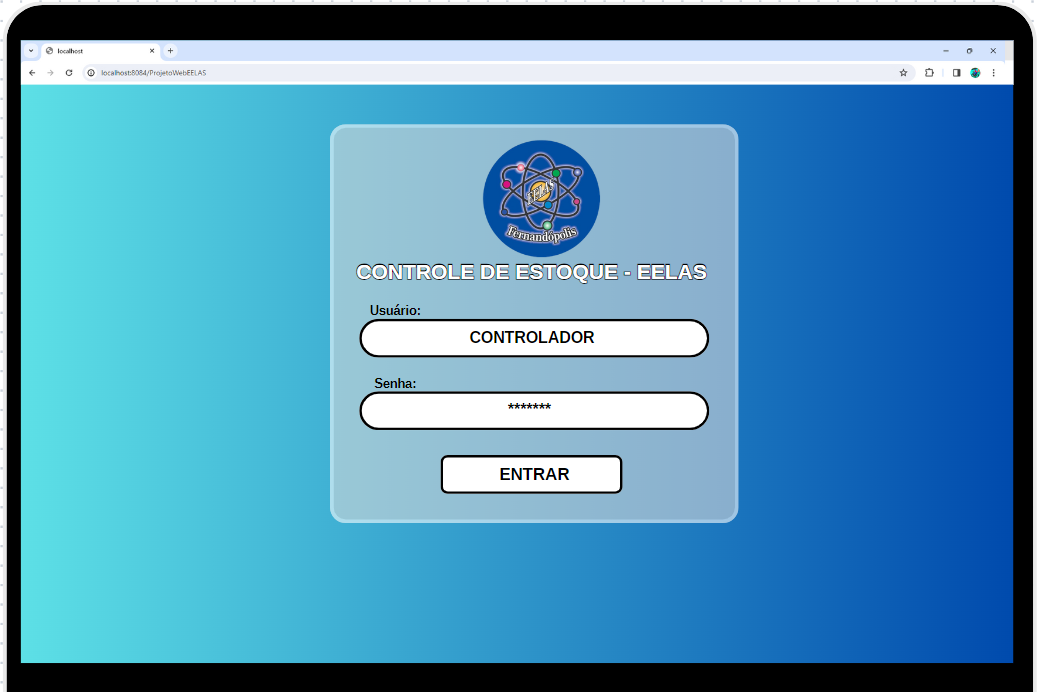


#### 3.6.6 Tela de acessos – Pessoa Servidor.

Nessa figura apresenta a tela de acessos que é utilizada pelo usuário Servidor do sistema. E como demonstração cliquei no módulo Acompanhamento de solicitação.

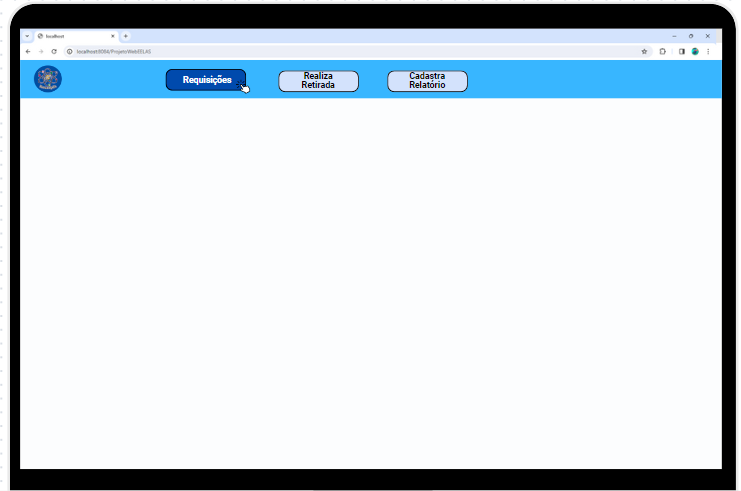
3.6.7 Tela Solicita Material – Pessoa Servido**r**

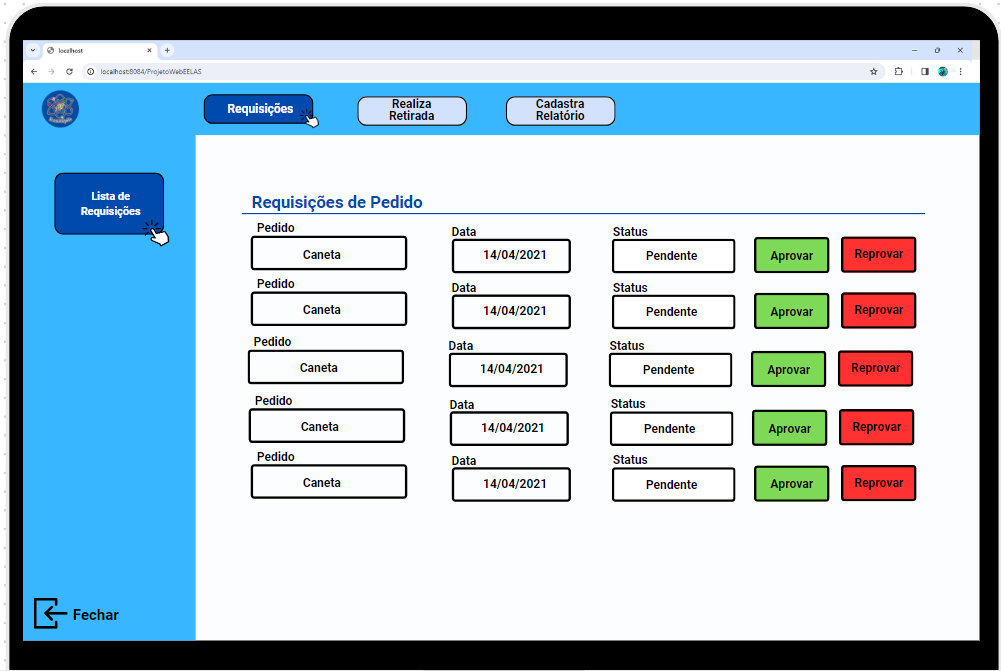
3.6.8 Tela Acompanha Solicitação – Pessoa Servido**r**

**3.6.9 Tela de Login – Pessoa Controlador**

**3.6.10 Tela de acessos – Pessoa Controlador.**

Nessa figura apresenta a tela de acessos que é utilizada pelo usuário Servidor do sistema. E como demonstração cliquei no módulo Requisições



**3.6.11 Tela de Requisições – Pessoa Controlador**

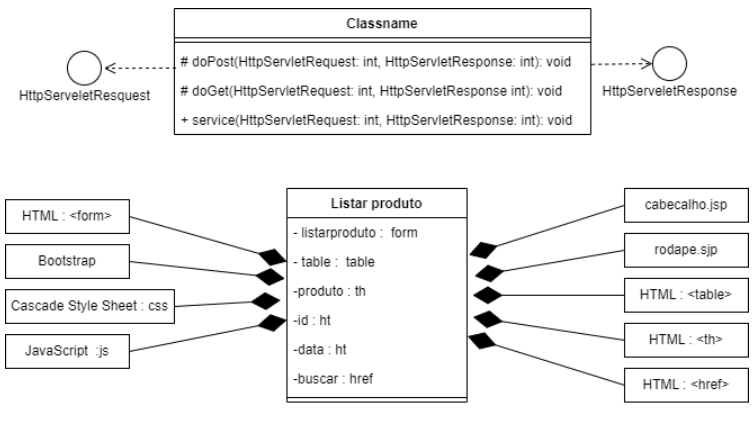
## Diagrama de Classes – Interface Homem-Máquina

De acordo com Góes (2014), as interfaces permitem que objetos externos ao sistema possam colaborar com uma ou mais classes do sistema. Também são exemplos de interface, nesse caso denominadas interfaces gráficas, as interações que os usuários fazem com a tela do sistema, por exemplo, selecionar uma opção.

Interfaces são coleções de operações que especificam serviços de um componente. É por meio delas que os componentes se comunicam com o mundo externo, seja para oferecer ou receber serviços. As interfaces fazem as associações entre os componentes do software (GÓES, 2014).

Os diagramas nas abaixo representam as classes na camada VIEW.

A figura abaixo demonstra o Diagrama de Classe de Interface Homem-Máquina na tela de Listar Produto.

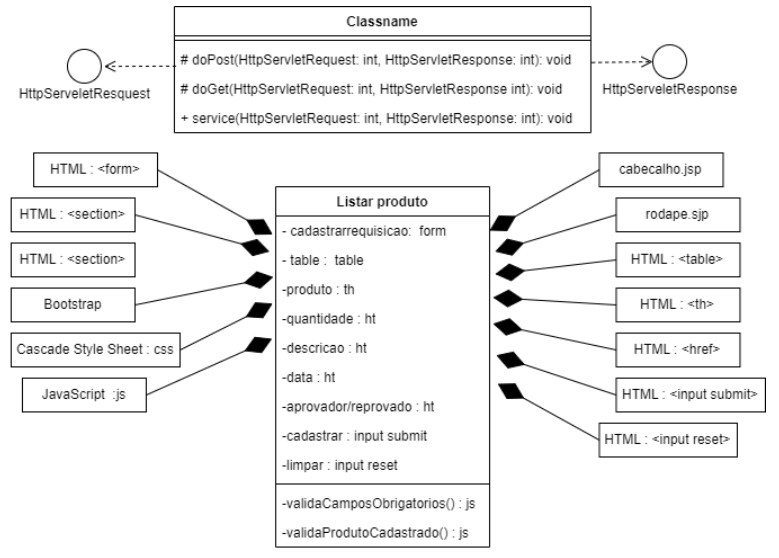
- DCIHM - Listar Produto

Fonte: Os Autores

A figura abaixo demonstra o Diagrama de Classe de Interface Homem-Máquina na tela de Cadastro de Requisição de Merdadoria.

DCIHM - Cadastro de Requisição de Merdadoria

Fonte: Os Autores



## Diagrama de Sequência – Modelo MVC

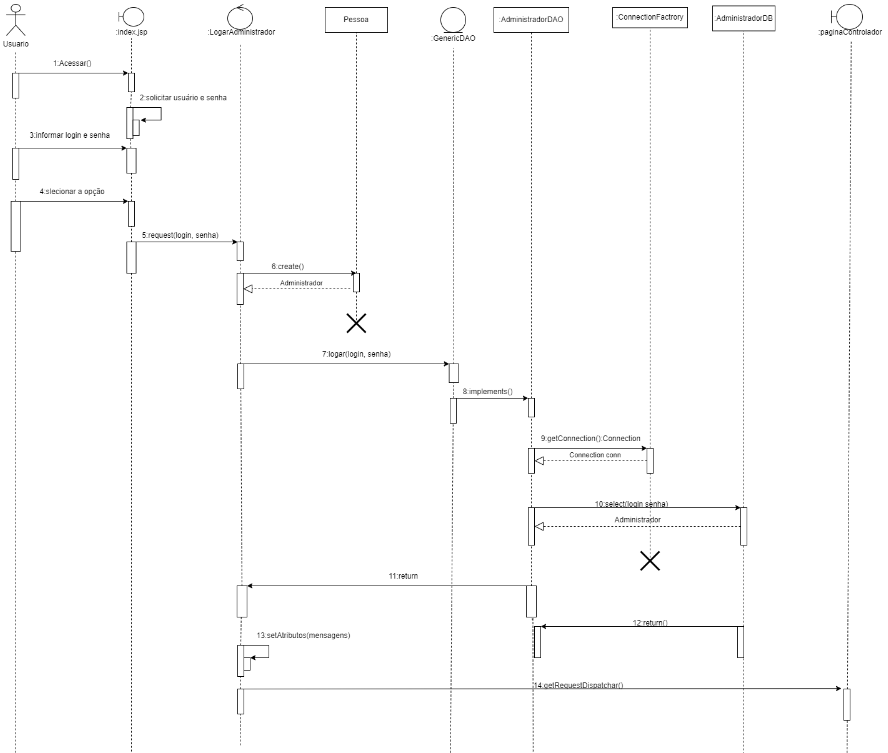
Este diagrama procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais métodos devem ser disparados entre os atores e objetos envolvidos e em que ordem (GUEDES, 2014).

De acordo com Góes (2014), de preferência, devem ser construídos após os diagramas de caso de uso e classes, pois seus principais objetivos são: documentar casos de uso e demonstrar como os objetos do sistema se comunicam por meio de mensagens em ordem temporal.

Os diagramas nas figuras abaixo demonstram os diagramas de sequência do sistema.

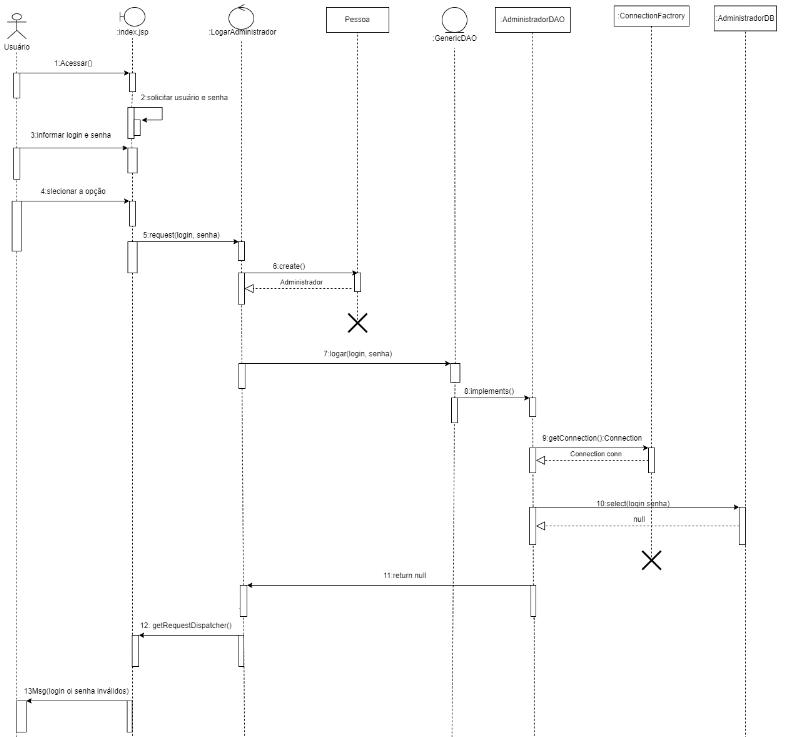
A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Administrador Realizar Login – Fluxo Normal.

Diagrama de Sequência - Administrador Realizar Login – FN

****Fonte: Os autores.

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Logar Administrador – Fluxo alternativo.

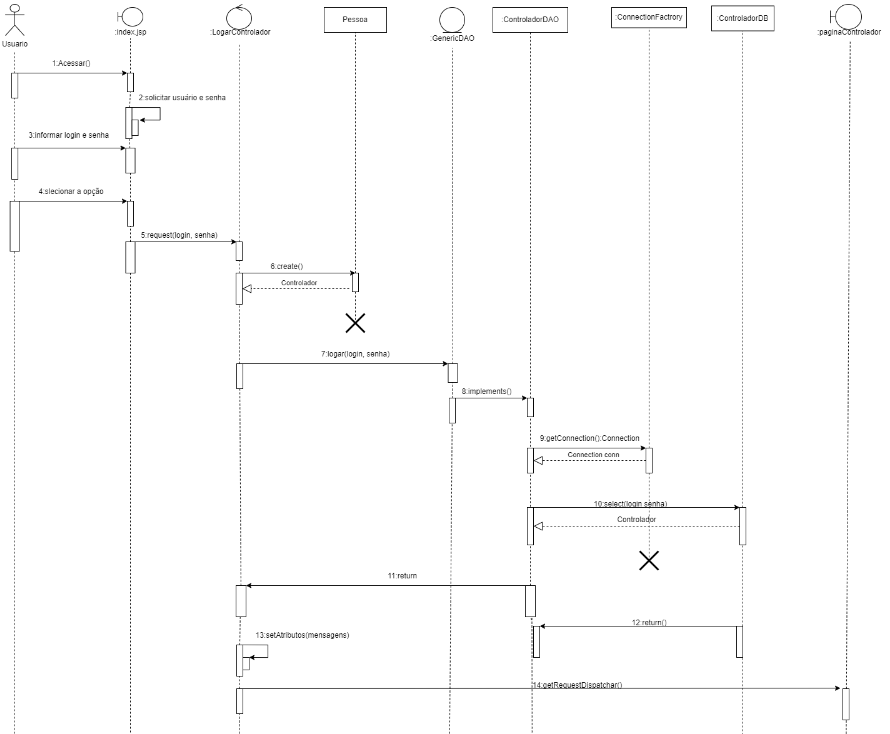
Diagrama de Sequência - Logar Administrador- FA



Fonte: Os autores.

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Cadastrar Controlador – Fluxo normal.

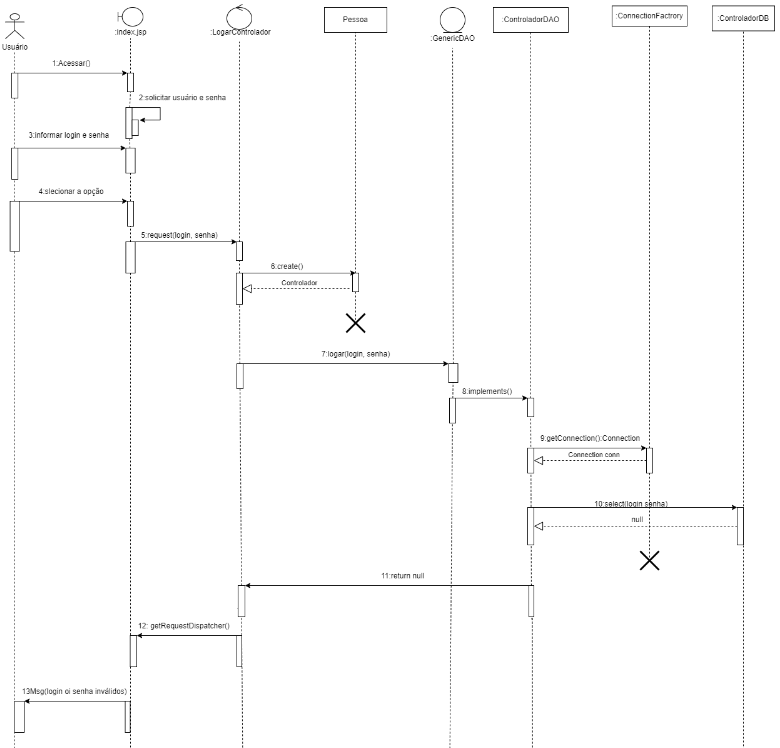
Diagrama de Sequência - Cadastrar Controlador- FN



Fonte: Os autores

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Logar Controlador - Fluxo alternativo.

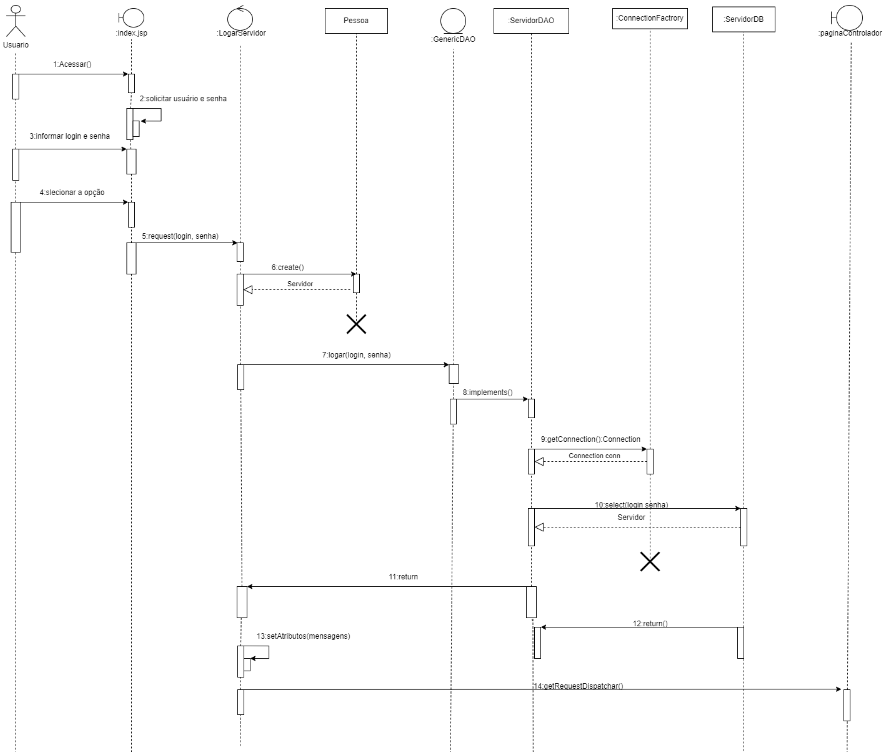
Diagrama de Sequência - Logar Controlador - FA



Fonte: Os Autores

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Cadastrar Servidor – Fluxo normal.

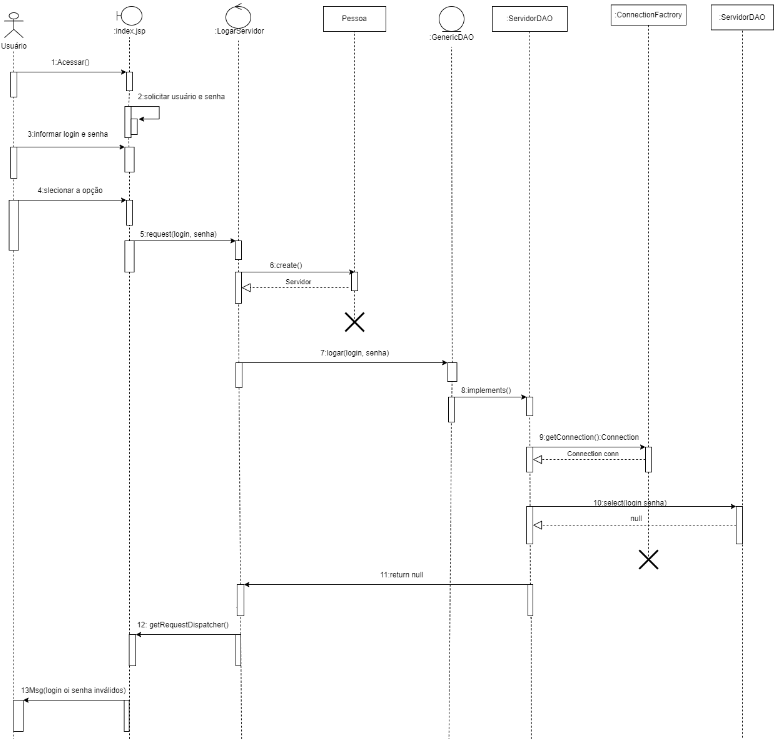
Diagrama de Sequência - Cadastrar Servidor- FN



Fonte: Os Autores

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Logar Servidor – Fluxo alternativo.

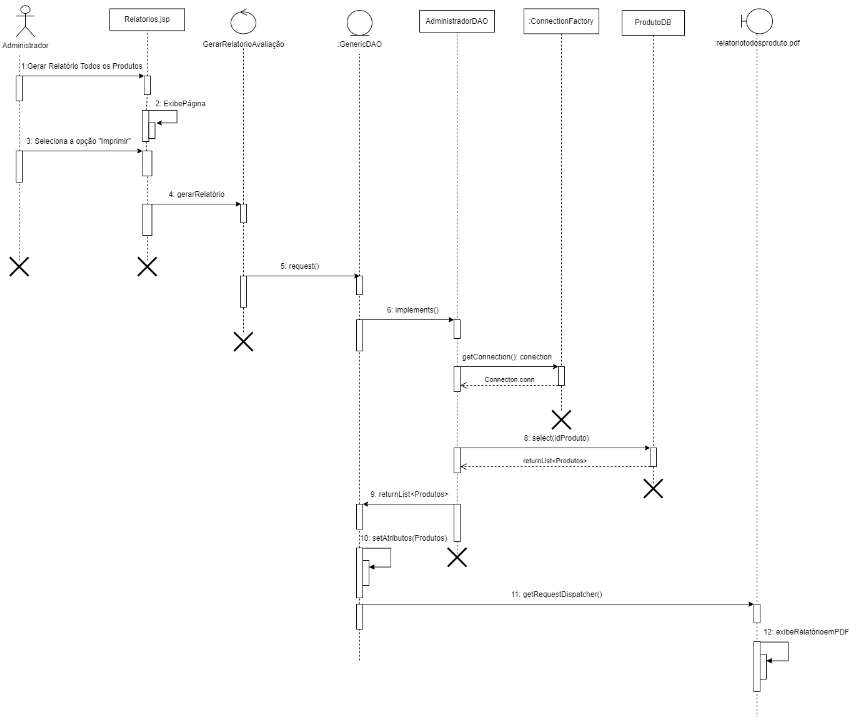
Diagrama de Sequência - Logar Servidor- FA



Fonte: Os Autores

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Gerar Relatório Todos Produtos – Fluxo Normal.

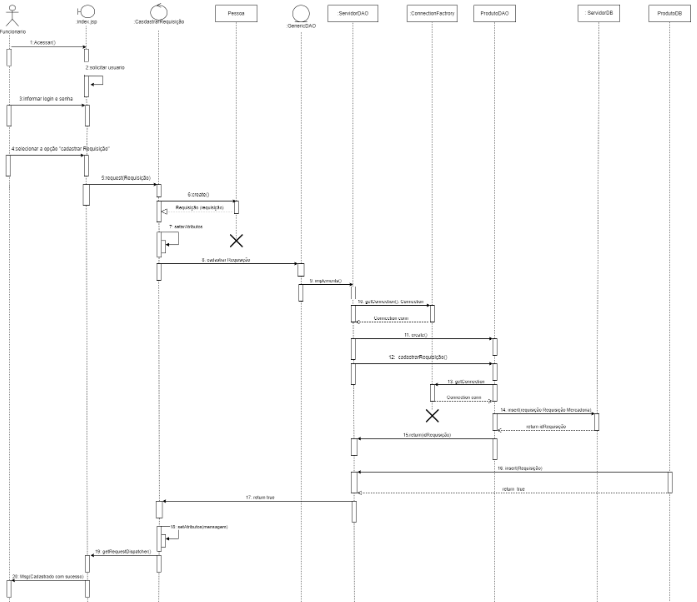
Diagrama de Sequência - Administrador - Gerar Relatório Todos Produtos- FN



Fontes: Os Autores

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Cadastrar Requisição – Fluxo Normal.

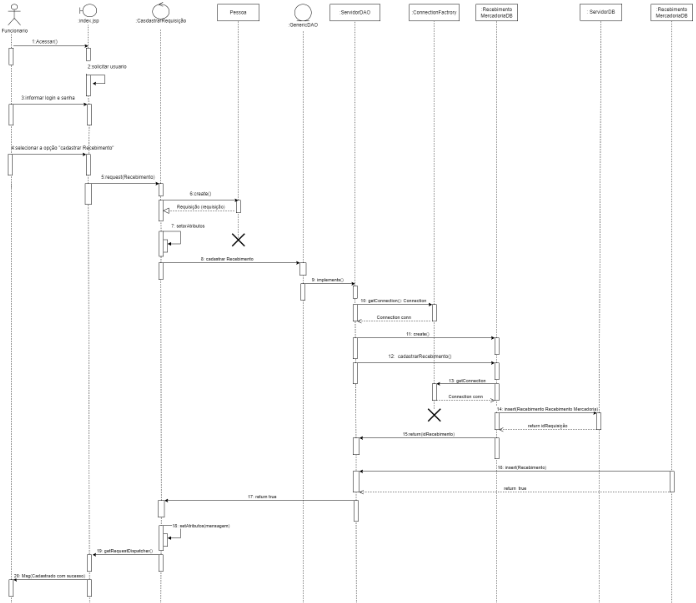
Diagrama de Sequência - Administrador - Cadastrar Requisição - FN



Fonte: Os Autores

A figura abaixo demonstra o Diagrama de Sequência do Caso de Uso Cadastrar Recebimento – Fluxo Normal.

Diagrama de Sequência - Administrador - Cadastrar Recebimento - FN



Fonte: O Autores

## Diagrama de Máquina de Estados

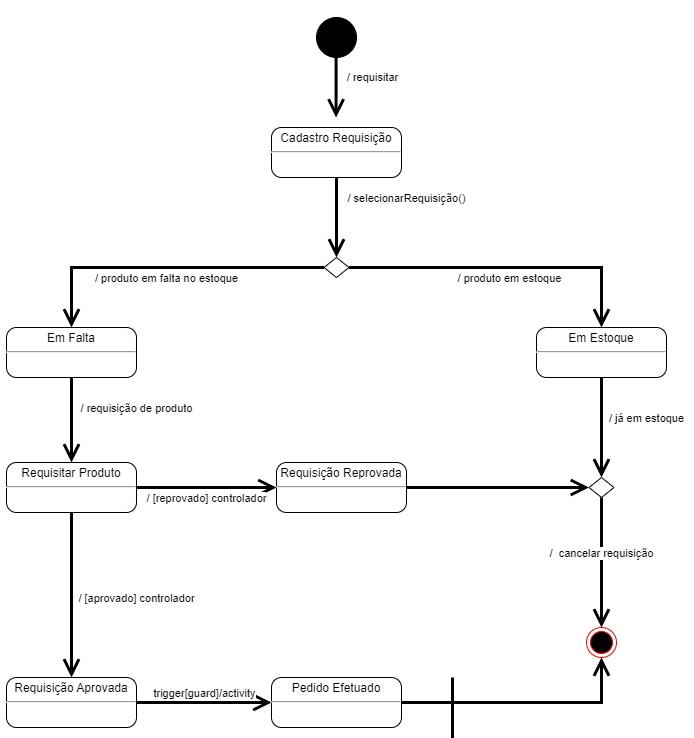
Em UML um estado representa a situação em que um objeto se encontra em um determinado momento durante o período em que participa de um processo. Um objeto pode passar por diversos estados dentro de um mesmo processo (GUEDES, 2014).

De acordo com Sbrocco (2014), o objetivo do diagrama de máquinas de estados é acompanhar as mudanças sofridas nos estados da instância de uma classe ao longo da execução de um caso de uso ou de um subsistema.

O diagrama de máquinas de estado demonstra o comportamento de um elemento por meio de um conjunto finito de transições de estado, ou seja, uma máquina de estados (GUEDES, 2014).

A figura abaixo demonstra o Diagrama de Máquina de Estados para Agendar Aula

Diagrama de Máquina de Estados – Agendar Aula



## Controle de Acesso e Segurança

A segurança da informação é a proteção da informação quanto a vários tipos de ameaças, de modo a garantir a continuidade do negócio, minimizar o risco para o negócio, maximizar o retorno sobre o investimento e as oportunidades de negócio (NBR ISO/IEC 27002:2005).

O fornecimento de senhas pelos administradores de sistemas e utilização de senhas pelos usuários é uma parte específica da política de segurança, de grande 104 importância para as organizações (GEUS E NAKAMURA, 2007).

A fim de adequar o sistema as normas da NRB ISO/IEC 27002:2005, o sistema possui controle de sessão e o controle de acesso é feito através de uma tela de acesso onde o usuário informa o seu login e sua senha, o sistema valida os dados, verifica o tipo de usuário e redireciona para página respectiva ao seu tipo.