

Controle de estoque

Documento de Arquitetura de Software

Versão <2.2>

Histórico da Revisão

Data	Versão	Descrição	Autor
29/05/2022	1.0	Elaboração inicial do documento	Gabriel Arcanjo
04/06/2022	1.1	Inclusão das decisões arquiteturais e justificativas	Gabriel Arcanjo
12/06/2022	1.2	Adição dos Requisitos Não Funcionais	Marcos Victor Ennes
15/06/2022	1.3	Adição dos casos de uso, Diagrama de casos de uso e Diagrama de pacotes	Gabriel Arcanjo
16/06/2022	1.4	Adição do Diagrama de classes	Marcos Victor Ennes
28/06/2022	1.5	Inclusão do Padrão de Projeto Observer	Marcos Victor Ennes
02/07/2022	2.0	Adição dos VCPs	Gabriel Arcanjo
04/07/2022	2.1	Adição do Sumário e organização do documento	Marcos Victor Ennes
07/07/2022	2.2	Adição de diagramas sequenciais	Gabriel Arcanjo

Índice Analítico

1. Introdução	3
2. Decisões e justificativas	4
3. Requisitos Arquiteturalmente Significantes	5
4. Diagramas	26
5. Camadas da Arquitetura	35

Documento de Arquitetura de Software

1. Introdução

Neste documento será feito o detalhamento das etapas de desenvolvimento do projeto de software E-Stock. Nosso produto tem como finalidade ser um software de gerenciamento de estoque, permitindo a entrada e saída de produtos, assim como o controle dos mesmos, armazéns e clientes.

1.1 Escopo

Este documento tem como objetivo documentar todas as decisões feitas ao longo do projeto E-Stock, juntamente com a justificativa do porque elas foram tomadas, a fim de deixar o processo de desenvolvimento claro e organizado não apenas para os funcionários que estejam atuando em seu desenvolvimento no momento atual, mas também para futuros novos integrantes da equipe ou para revisões futuras. Desta forma, procura-se assegurar que o produto final tenha o máximo de compatibilidade possível com os requisitos desejados e uma boa qualidade.

1.2 Decisões de projeto

Antes de iniciarmos o desenvolvimento, chegamos a um consenso de como nosso software seria desenvolvido, incluindo linguagem de programação, princípios, banco de dados, responsividade, repositório para monitoramento e controle, banco de dados, padrão arquitetural. Segue o resultado:

- Programação orientada a objetos
- Linguagem JavaScript
- Uso de banco de dados PostgreSQL
- Sistema responsivo e seguro
- GitHub como repositório principal
- Padrões arquiteturais de MVC e Microserviços como os principais

2. Decisões e justificativas

2.1 Linguagem

Escolhemos como a linguagem JavaScript como nossa principal por atender aos requisitos de orientação a objetos, ser de amplo conhecimento de grande parte dos membros do grupo, possibilidade de inserir comando efeitos variados, deixando o produto mais dinâmico, grande interatividade com o usuário, portabilidade e expansividade, política de testes eficiente, além de ser bastante popular no mercado e bem aceita.

2.2 Repositório

A escolha do Git como repositório principal, além da óbvia popularidade e acessibilidade, se deu pelos benefícios que o sistema apresenta, como controle de versionamento, com fácil acesso às versões anteriores e seus status de modificações, feature de *Branches*, política eficiente de merge, fácil de identificação de bugs, através de busca binária, além de vários facilitadores como o *git bisect*, *git ignore*, entre outros.

2.3 Banco de dados

Os principais motivos que nos levaram a escolher o PostgreSQL como nosso banco de dados foram a sua facilidade de uso, compatibilidade, segurança, suporta grandes volumes de dados, além da economia e alto desempenho oferecido pelo SGBD.

2.4 Padrões Arquiteturais

Foram utilizados os seguintes Padrões Arquiteturais:

- MVC (Model-view-controller);
- Layers (Camadas);
- Microservices (microserviços);
- Client-server (cliente-servidor).

As explicações dos padrões estão presentes na sessão 5.1.

2.5 Padrão de projeto

Para nosso projeto do E-Stock, utilizaremos o padrão *Factory*. Escolhemos esse padrão, porque basicamente em nosso projeto, teremos a superclasse de produto, que será o representante das mercadorias em estoque, e seus derivados, com as subcaracterísticas de produto e suas variações. Além disso, utilizamos o padrão *Observer* (Explicação no capítulo 5.2).

3.0 Requisitos Arquiteturalmente Significantes

3.1 Requisitos Funcionais

Requisitos Funcionais	Descrição
(RF1) - Cadastro de produto	O software deve permitir a inserção e armazenamento dos atributos do produto (Id, nome, custo, peso).
(RF2) - Atualizar Produto	O software deve permitir que as informações de um produto cadastrado sejam atualizadas/alteradas.
(RF3) - Deletar produto	O software deve permitir que o cadastro de um produto seja deletado.
(RF4) - Pesquisar produtos	O software deve permitir o acesso e leitura de dados cadastrados.
(RF5) - Transação (Entrada e saída de estoque)	O software deve oferecer um mecanismo de estoque, adicionando ou reduzindo a quantidade em estoque de determinado produto em um local de armazenagem.

Grupo 02

(RF6) - Relatório	O software deve ter uma feature que permita o controle das informações e transações dos produtos, tais como listagem de quantidade, produtos com maior e menor taxa de venda, saída (venda) e saldo atual do estoque por local de armazenagem.
(RF7) - Cadastrar usuário	Capacidade de armazenar os dados dos usuários.
(RF8) - Efetuar login	Funcionalidade para permitir que os usuários tenham acesso ao sistema.
(RF9) - Alterar informações do usuário	Funcionalidade que permite atualizar as informações cadastradas do próprio usuário.
(RF10) - Recuperar senha	Funcionalidade que permite, aos usuários que esqueceram sua senha, recuperá-la ou criar uma nova através do e-mail cadastrado.
(RF11) - Cadastrar local de armazenagem	O software deve permitir o cadastro de locais de armazenagem de produtos.
(RF12) - Atualizar dados de local de armazenagem	O software deve permitir a alteração de informações dos locais de armazenagem e disponibilidade de produtos nos locais de armazenagem.
(RF13) - Deletar local de armazenagem	O software deve permitir a remoção de locais de armazenagem.
(RF14) - Transferir Produtos	O software deve permitir a transferência de produtos entre locais de armazenagem.

(RF15) - Relatório de transferências de produtos	O software deve ser capaz de gerar um relatório com as informações de transferência entre locais de armazenagem - produto transferido, de/para de local de armazenagem, quantidade transferida e data
--	---

3.2 Requisitos Não Funcionais

Não Funcionais	Descrição
(RNF1) - Usabilidade	<p>Um usuário treinado deve ser capaz de cadastrar um produto em menos de 10 minutos. Já um usuário não treinado deve ser capaz de efetuar esta tarefa em até 15 minutos.</p> <p>Um usuário não treinado deve ser capaz de realizar a remoção de um produto em menos de 5 minutos.</p>
(RNF2) - Manutenibilidade	O sistema deve ser mantido com sua documentação atualizada, incluindo seus diagramas UML.
(RNF3) - Confiabilidade	A cada 1000 execuções, não mais que 2 podem apresentar falhas de software.

Grupo 02

(RNF4) - Portabilidade	<p>O sistema deve poder ser acessado através dos principais navegadores do mercado: Google Chrome, Mozilla Firefox, Safari e Microsoft Edge.</p> <p>O sistema deve poder ser acessado por dispositivos móveis Android e iOS.</p>
(RNF5) - Desempenho	O sistema deve processar transações de forma rápida e apresentar um curto tempo de resposta do usuário/evento
(RNF6) - Segurança	O sistema deve armazenar as senhas dos usuários no banco de dados de forma segura e garantir o sigilo das informações.
(RNF7) - Disponibilidade	O sistema deve ficar disponível por 99,95% em dias úteis das 6h às 22h e 99,5% no restante do tempo.
(RNF8) – Eficiência	A requisição que efetua a pesquisa de produtos deve ser processada em menos de 3 segundos.

3.3 Casos de uso (Tiago Espanha e Bruno Xavier)

Nome	UC01 - Cadastrar loja
Ator(es)	Usuário empresa, sistema de estoque
Descrição	Cadastramento de uma loja que o sistema gerenciará estoque

Grupo 02

Referências	Casos de uso relacionados: 1- UC02 - Cadastrar produto 2- UC03 - Atualizar produto 3- UC04 - Deletar produto 4- UC05 - Listar produtos 5- UC06 - Adicionar produto ao estoque 6- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	-
Pós-condições	Loja cadastrada
Fluxo principal	1- O usuário entra no site 2- O usuário cadastra dados pessoais e da loja 3- O sistema guarda os dados e redireciona para a página de administração do estoque da loja
Fluxo alternativo	-

Nome	UC02 - Cadastrar produto
Ator(es)	Usuário empresa, sistema de estoque
Descrição	Usuário cadastrando um produto que poderá ser adicionado ao estoque futuramente

Grupo 02

Referências	Casos de uso relacionados: 1- UC03 - Atualizar produto 2- UC04 - Deletar produto 3- UC05 - Pesquisar produto 4- UC06 - Adicionar produto ao estoque 5- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário cadastrado
Pós-condições	Produto cadastrado
Fluxo principal	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O usuário escolhe a opção 'adicionar novo produto' 4- O usuário entra com os dados do novo produto 5- O usuário confirma a inserção do novo produto 6- O usuário agora vê o produto na listagem de produtos
Fluxo alternativo	1- O usuário loga na plataforma 2- O usuário escolhe o menu 'produtos' 3- O usuário escolhe a opção de 'adicionar novo produto' 4- O usuário adiciona um identificador já utilizado 5- O usuário confirma a inserção do novo produto 6- O sistema apresenta uma tela de erro, informando que o identificador já existe
Nome	UC02 - Cadastrar produto

Grupo 02

Ator(es)	Usuário empresa, sistema de estoque
Descrição	Usuário cadastrando um produto que poderá ser adicionado ao estoque futuramente
Referências	Casos de uso relacionados: 1- UC03 - Atualizar produto 2- UC04 - Deletar produto 3- UC0x - Pesquisar produto 4- UC06 - Adicionar produto ao estoque 5- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário cadastrado
Pós-condições	Produto cadastrado
Fluxo principal	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O usuário escolhe a opção 'adicionar novo produto' 4- O usuário entra com os dados do novo produto 5- O usuário confirma a inserção do novo produto 6- O usuário agora vê o produto na listagem de produtos
Fluxo alternativo	1- O usuário loga na plataforma 2- O usuário escolhe o menu 'produtos' 3- O usuário escolhe a opção de 'adicionar novo produto' 4- O usuário adiciona um identificador já utilizado 5- O usuário confirma a inserção do novo produto 6- O sistema apresenta uma tela de erro, informando que o identificador já existe

Grupo 02

Nome	UC03 - Atualizar produto
Ator(es)	Usuário empresa, sistema de estoque
Descrição	Usuário atualizando os dados de um produto do estoque
Referências	Casos de uso relacionados: 1- UC02 - Cadastrar produto 2- UC04 - Deletar produto 3- UC05 - Listar produto 4- UC06 - Adicionar produto ao estoque 5- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário empresa cadastrado, Produto cadastrado
Pós-condições	Produto atualizado, produto atualizado nos estoques
Fluxo principal	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O usuário escolhe um produto 4- O usuário escolhe a opção 'atualizar produto' 5- O usuário entra com os dados novos do produto 6- O usuário confirma a atualização do produto 7- O usuário agora vê o produto atualizado na listagem de produtos

Grupo 02

Fluxo alternativo	<ul style="list-style-type: none">1- O usuário loga na plataforma2- O usuário escolhe o menu 'produtos'3- O usuário escolhe um produto4- O usuário escolhe a opção 'atualizar produto'5- O usuário coloca os mesmos dados do produto já cadastrado6- O usuário confirma a atualização do produto7- O sistema apresenta uma tela de erro, informando que os dados do produto são iguais ao já cadastrado.
-------------------	--

Nome	UC04 - Deletar produto
Ator(es)	Usuário, sistema de estoque
Descrição	Usuário remove um produto da lista de produtos
Referências	<p>Casos de uso relacionados:</p> <ul style="list-style-type: none">1- UC03 - Atualizar produto2- UC05 - Listar produtos3- UC06 - Adicionar produto ao estoque4- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário cadastrado, produto cadastrado
Pós-condições	Produto deletado, produto removido de todos os estoques

Grupo 02

Fluxo principal	<p>1- O usuário loga na plataforma</p> <p>2- O usuário escolhe a opção 'produtos' no menu</p> <p>3- O usuário escolhe a opção 'remover produto'</p> <p>4- O sistema mostra uma mensagem informado o usuário que essa ação não pode ser desfeita e pode alterar o estoque</p> <p>5- O usuário confirma a deleção do produto</p> <p>6- O sistema redireciona o usuário para a tela de listagem de produtos</p> <p>6- O usuário agora não vê o produto na listagem de produtos</p>
Fluxo alternativo	<p>1- O usuário loga na plataforma</p> <p>2- O usuário escolhe a opção 'produtos' no menu</p> <p>3- O usuário escolhe a opção 'remover produto'</p> <p>4- O sistema mostra uma mensagem informado o usuário que essa ação não pode ser desfeita e pode alterar o estoque</p> <p>5- O usuário desiste da deleção do produto</p> <p>6- O sistema redireciona o usuário para a tela de listagem de produtos</p> <p>6- O usuário ainda vê o produto na listagem de produtos</p>

Nome	UC05 - Listar produtos
Ator(es)	Usuário empresa, sistema de estoque
Descrição	Sistema lista os produtos cadastrados

Grupo 02

Referências	Casos de uso relacionados: 1- UC03 - Atualizar produto 2- UC04 - Deletar produto 3- UC02 - Cadastrar produto 4- UC06 - Adicionar produto ao estoque 5- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário empresa cadastrado, produto cadastrado
Pós-condições	Produtos listados
Fluxo principal	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O sistema mostra os produtos cadastrados
Fluxo alternativo	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O sistema informa que não há produtos cadastrados

Nome	UC06 - Adicionar produto ao estoque
Ator(es)	Usuário empresa, sistema de estoque
Descrição	Usuário adiciona um produto a um estoque

Grupo 02

Referências	Casos de uso relacionados: 1- UC02 - Cadastrar produto 2- Pesquisar produto 3- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário cadastrado, produto cadastrado
Pós-condições	Produto presente ao estoque
Fluxo principal	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O usuário escolhe a opção 'adicionar produto ao estoque' 4- O sistema abre um modal com atributos necessários para adicionar um produto ao estoque 5- O usuário entra com os dados (preço e quantidade) e confirma 6- O sistema mostra mensagem que confirma a adição do produto e fecha o modal
Fluxo alternativo	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'produtos' no menu 3- O usuário escolhe a opção 'adicionar produto ao estoque' 4- O sistema abre um modal com atributos necessários para adicionar um produto ao estoque 5- O usuário entra com os dados (preço e quantidade) e confirma 6- O usuário desiste e fecha o modal

Nome	UC07 - Remover produto do estoque
------	--

Grupo 02

Ator(es)	Usuário empresa, sistema
Descrição	Usuário remove um produto do estoque
Referências	Casos de uso relacionados: 1- UC02 - Cadastrar produto 2- Pesquisar produto 3- UC06 - Adicionar produto ao estoque
Gatilho	-
Pré-condições	Usuário cadastrado, produto cadastrado, produto adicionado ao estoque
Pós-condições	Produto removido do estoque
Fluxo principal	1- O usuário loga na plataforma 2- O usuário escolhe a opção 'estoque' no menu 3- O sistema redireciona para uma página que lista os produtos presentes no estoque 3- O usuário escolhe a opção de remover 3- O usuário escolhe a opção 'adicionar produto ao estoque' 4- O sistema abre um modal com atributos necessários para adicionar um produto ao estoque 5- O usuário entra com os dados (preço e quantidade) e confirma 6- O sistema mostra mensagem que confirma a adição do produto e fecha o modal

Grupo 02

Fluxo alternativo	<ul style="list-style-type: none">1- O usuário loga na plataforma2- O usuário escolhe a opção 'produtos' no menu3- O usuário escolhe a opção 'adicionar produto ao estoque'4- O sistema abre um modal com atributos necessários para adicionar um produto ao estoque5- O usuário entra com os dados (preço e quantidade) e confirma6- O usuário desiste de remover produto e fecha o modal
-------------------	---

Nome	UC08 - Relatório do estoque
Ator(es)	Usuário empresa, sistema
Descrição	O sistema lista o relatório do estoque ao usuário, assim como também informa produtos com maior e menor taxa de venda e saldo atual do estoque.
Referências	Casos de uso relacionado: <ul style="list-style-type: none">1- UC02 - Cadastrar produto2- Pesquisar produto3- UC06 - Adicionar produto ao estoque4- UC07 - Remover produto do estoque
Gatilho	-
Pré-condições	Usuário cadastrado, produto cadastrado, produto adicionado ao estoque
Pós-condições	Relatório do estoque listado pelo usuário

Grupo 02

Fluxo principal	<p>1 - O usuário loga na plataforma</p> <p>2 - O usuário escolhe a opção 'estoque' no menu</p> <p>3 - O sistema redireciona para uma página que lista os produtos presentes no estoque</p> <p>4 - O usuário escolhe a opção 'relatório'</p> <p>5 - O sistema redireciona para uma página que lista o relatório de entrada e saída dos produtos estoque</p>
Fluxo alternativo	<p>1 - O usuário loga na plataforma</p> <p>2 - O usuário escolhe a opção 'estoque' no menu</p> <p>3 - O sistema redireciona para uma página que lista os produtos presentes no estoque</p> <p>4 - O usuário escolhe a opção 'relatório'</p> <p>5 - O sistema informa que é necessário adicionar um produto pela primeira vez.</p>

Nome	UC09 - Efetuar login
Ator(es)	Usuário, sistema de estoque
Descrição	Usuário faz o login para entrar na plataforma
Referências	<p>Casos de uso relacionados:</p> <p>1 - Cadastro de usuário</p> <p>2 - UC10 - Atualização do usuário</p> <p>3 - UC11 - Recuperar senha</p>
Gatilho	-
Pré-condições	-

Grupo 02

Pós-condições	Usuário faz o login
Fluxo principal	1- O usuário loga na plataforma
Fluxo alternativo	1- O usuário loga na plataforma 2- O sistema informa que os dados não estão corretos

Nome	UC10 - Atualizar usuário
Ator(es)	Usuário, sistema de estoque
Descrição	Usuário faz o login para entrar na plataforma
Referências	Casos de uso relacionados: 1 - Cadastro de usuário 2 - UC09 - Efetuar login 3 - UC11 - Recuperar senha
Gatilho	-
Pré-condições	Usuário cadastrado
Pós-condições	Usuário faz o login
Fluxo principal	1 - O usuário loga na plataforma 2 - O usuário escolhe a opção 'Perfil' no menu 3 - O sistema redireciona para uma página de perfil do usuário 4 - O usuário atualiza os dados e confirma

Grupo 02

Fluxo alternativo	<p>1 - O usuário loga na plataforma</p> <p>2 - O usuário escolhe a opção 'Perfil' no menu</p> <p>3 - O sistema redireciona para uma página de perfil do usuário</p> <p>4 - O usuário atualiza os dados e confirma</p> <p>5 - O sistema apresenta uma tela de erro, informando que os dados do usuário são iguais ao já cadastrado.</p>
-------------------	--

Nome	UC11 - Recuperar Senha
Ator(es)	Usuário, sistema de estoque
Descrição	Usuário recupera senha de login para entrar na plataforma
Referências	<p>Casos de uso relacionados:</p> <p>1 - Cadastro de usuário</p> <p>2 - UC09 - Efetuar login</p> <p>3 - UC10 - Atualizar usuário</p>
Gatilho	-
Pré-condições	Usuário cadastrado
Pós-condições	Usuário recupera senha para login
Fluxo principal	<p>1 - O usuário escolhe no login a opção de recuperar senha</p> <p>2 - O sistema redireciona para uma página de recuperar senha</p> <p>3 - O usuário entra com os dados e confirma</p> <p>4 - O sistema atualiza a senha e redireciona para página de login</p>

Grupo 02

Fluxo alternativo	<p>1 - O usuário escolhe no login a opção de recuperar senha</p> <p>2 - O sistema redireciona para uma página de recuperar senha</p> <p>3 - O usuário entra com os dados e confirma</p> <p>4 - O sistema apresenta uma tela de erro, informando que a senha digitada é a mesma do usuário cadastrado.</p>
-------------------	---

Nome	UC12 - Cadastrar local de armazenagem
Ator(es)	Usuário, sistema de estoque
Descrição	O software deve permitir o cadastro de locais de armazenagem de produtos.
Referências	TODO
Gatilho	-
Pré-condições	Usuário cadastrado e logado
Pós-condições	Local de armazenagem está cadastrado
Fluxo principal	<p>1 - O usuário seleciona a opção cadastrar local de armazenagem</p> <p>2 - O usuário informa os dados de cadastro de local de armazenagem (Nome, endereço, capacidade total)</p> <p>3 - O sistema realiza o cadastro</p>
Fluxo alternativo	<p>2.1 - O usuário não preenche todos os campos.</p> <p>2.1.1 - Gera um erro</p> <p>2.1.2 - Retorna a tela de cadastro</p>

Grupo 02

Nome	UC13 - Atualizar dados de local de armazenagem
Ator(es)	Usuário, sistema de estoque
Descrição	O software deve permitir a alteração de informações dos locais de armazenagem e disponibilidade de produtos nos locais de armazenagem.
Referências	
Gatilho	-
Pré-condições	Usuário cadastrado e logado
Pós-condições	Local de armazenagem está atualizado
Fluxo principal	1 - O usuário seleciona a opção editar local de armazenagem 2 - O usuário informa os dados de cadastro de local de armazenagem (Nome, endereço, capacidade total) 3 - O sistema realiza a atualização
Fluxo alternativo	2.1 - O usuário não preenche todos os campos. 2.1.1 - Gera um erro 2.1.2 - Retorna a tela de cadastro

Nome	UC14 - Deletar local de armazenagem
Ator(es)	Usuário, sistema de estoque
Descrição	O software deve permitir a remoção de locais de armazenagem.

Grupo 02

Referências	
Gatilho	-
Pré-condições	Usuário cadastrado e logado
Pós-condições	Local de armazenagem é deletado
Fluxo principal	1 - O usuário seleciona a opção deletar local de armazenagem 2 - O usuário seleciona o local de armazenagem que deseja deletar e clica em deletar 3 - O sistema realiza a remoção
Fluxo alternativo	

Nome	UC15 - Transferir Produtos
Ator(es)	Usuário, sistema de estoque
Descrição	O software deve permitir a transferência de produtos entre locais de armazenagem.
Referências	
Gatilho	-
Pré-condições	Usuário cadastrado e logado
Pós-condições	

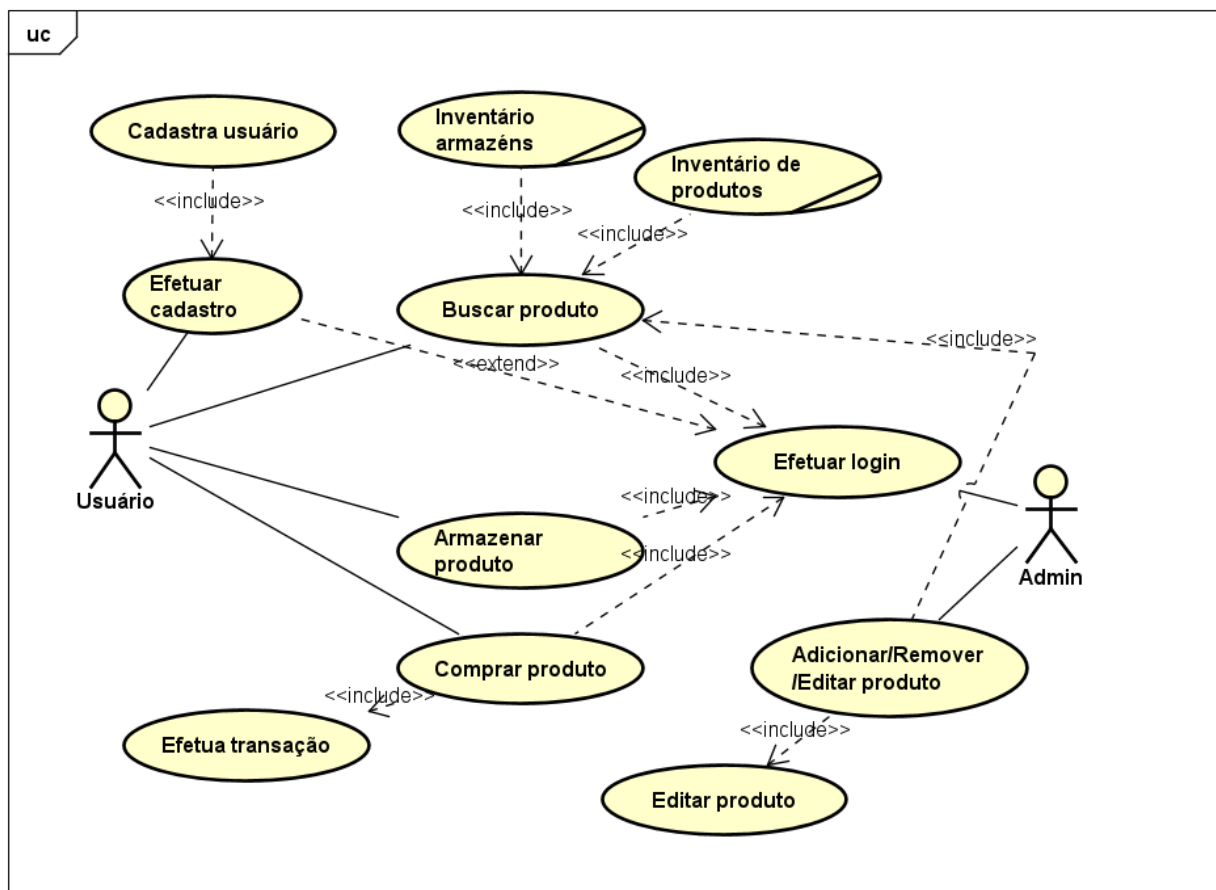
Grupo 02

Fluxo principal	1 - O usuário seleciona dois locais de armazenagem de uma listagem (de/para) 2 - O usuário informa um produto e a quantidade a ser transferida 3 - O sistema realiza a transferência
Fluxo alternativo	1.1 - Não existem ao menos 2 locais cadastrados. Abortar transferência. 3.1 - A quantidade não está disponível no local (de) 3.2 - O local destino (para) não tem capacidade para receber os produtos

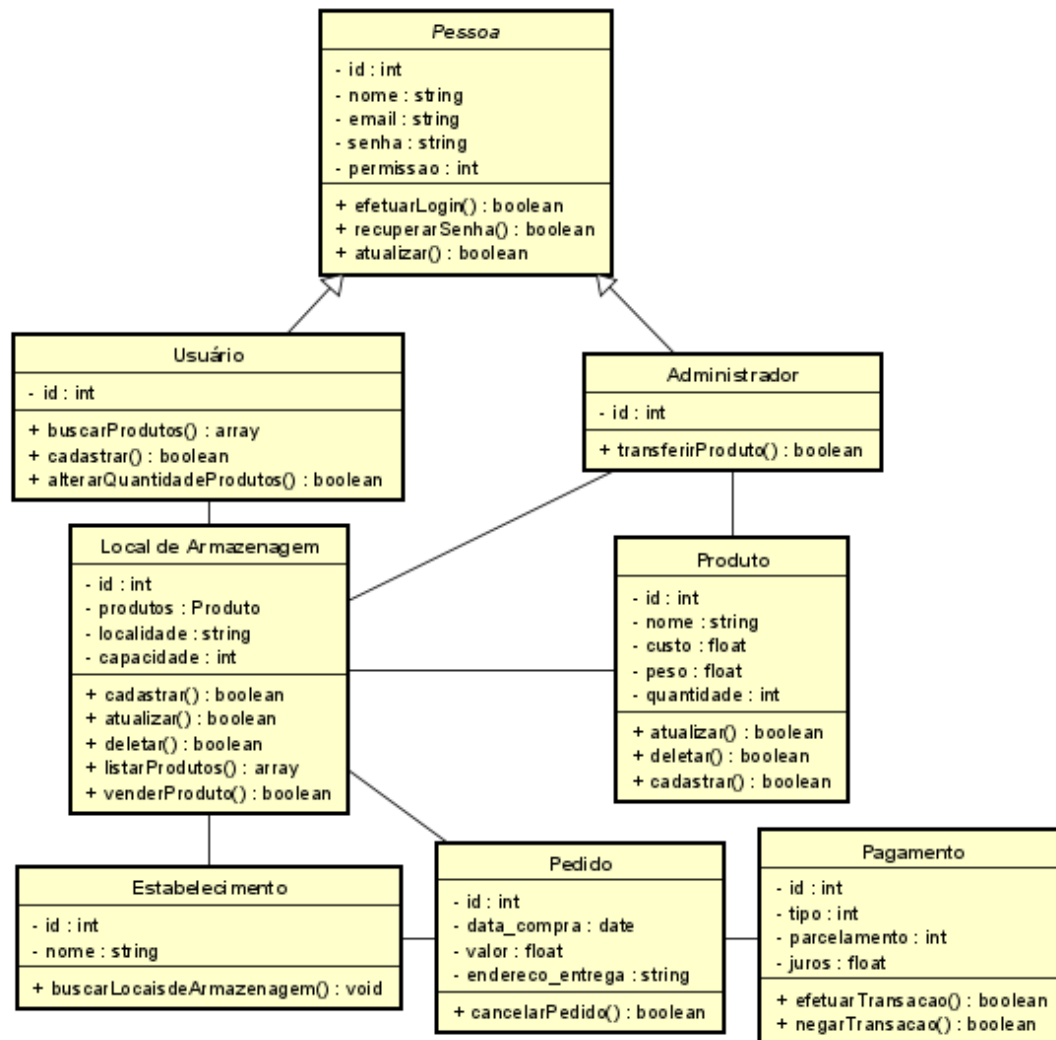
Nome	UC16 - Relatório de transferências de produtos
Ator(es)	Usuário, sistema de estoque
Descrição	O software deve ser capaz de gerar um relatório com as informações de transferência entre locais de armazenagem - produto transferido, de/para de local de armazenagem, quantidade transferida e data
Referências	
Gatilho	-
Pré-condições	Usuário cadastrado e logado
Pós-condições	
Fluxo principal	1 - O usuário escolhe a opção 'transferências' no menu 2 - O usuário escolhe a opção 'relatório' 3 - O sistema redireciona para uma página que lista o relatório de transferências dos produtos por local de armazenamento
Fluxo alternativo	

4.0 Diagramas

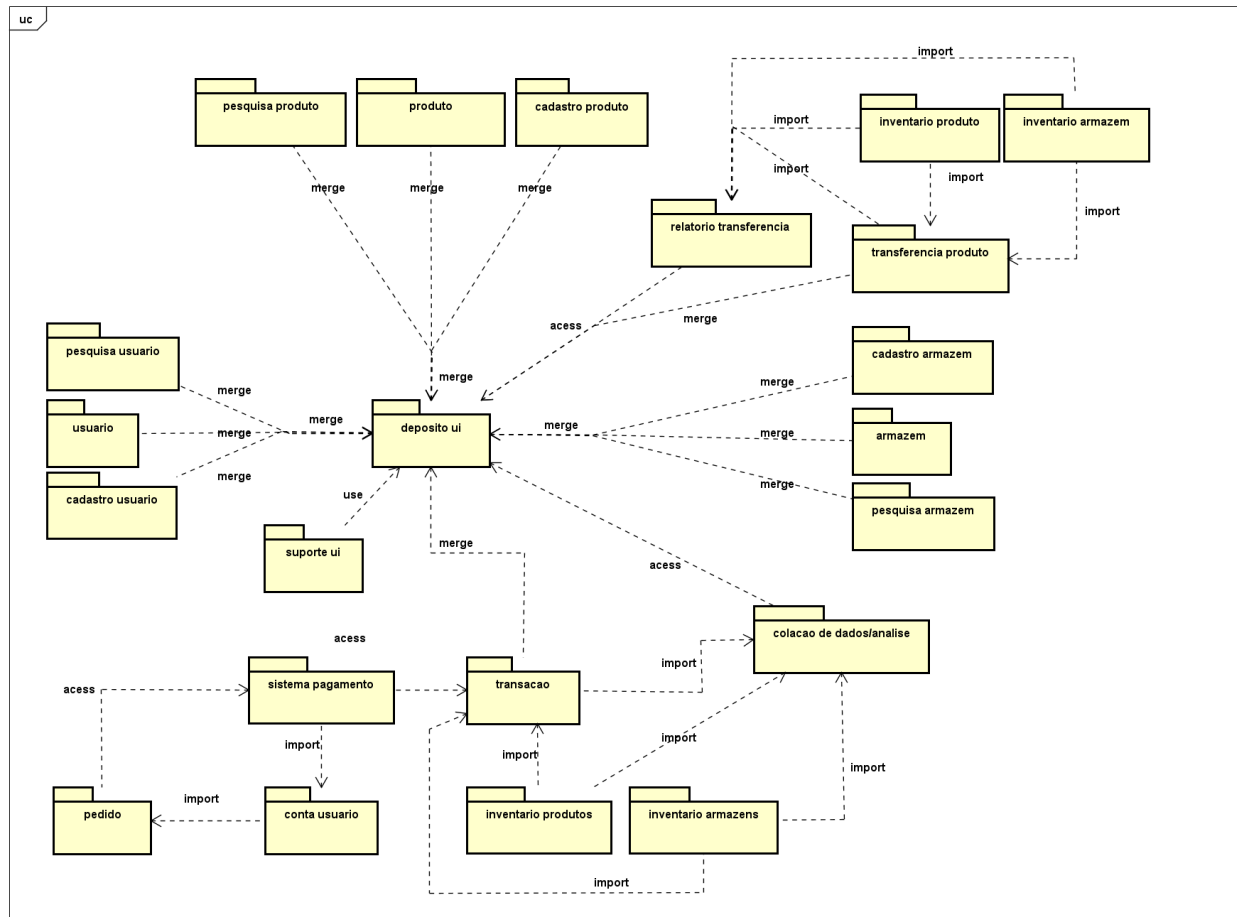
4.1 Diagrama de Casos de uso



4.2 Diagrama de Classes

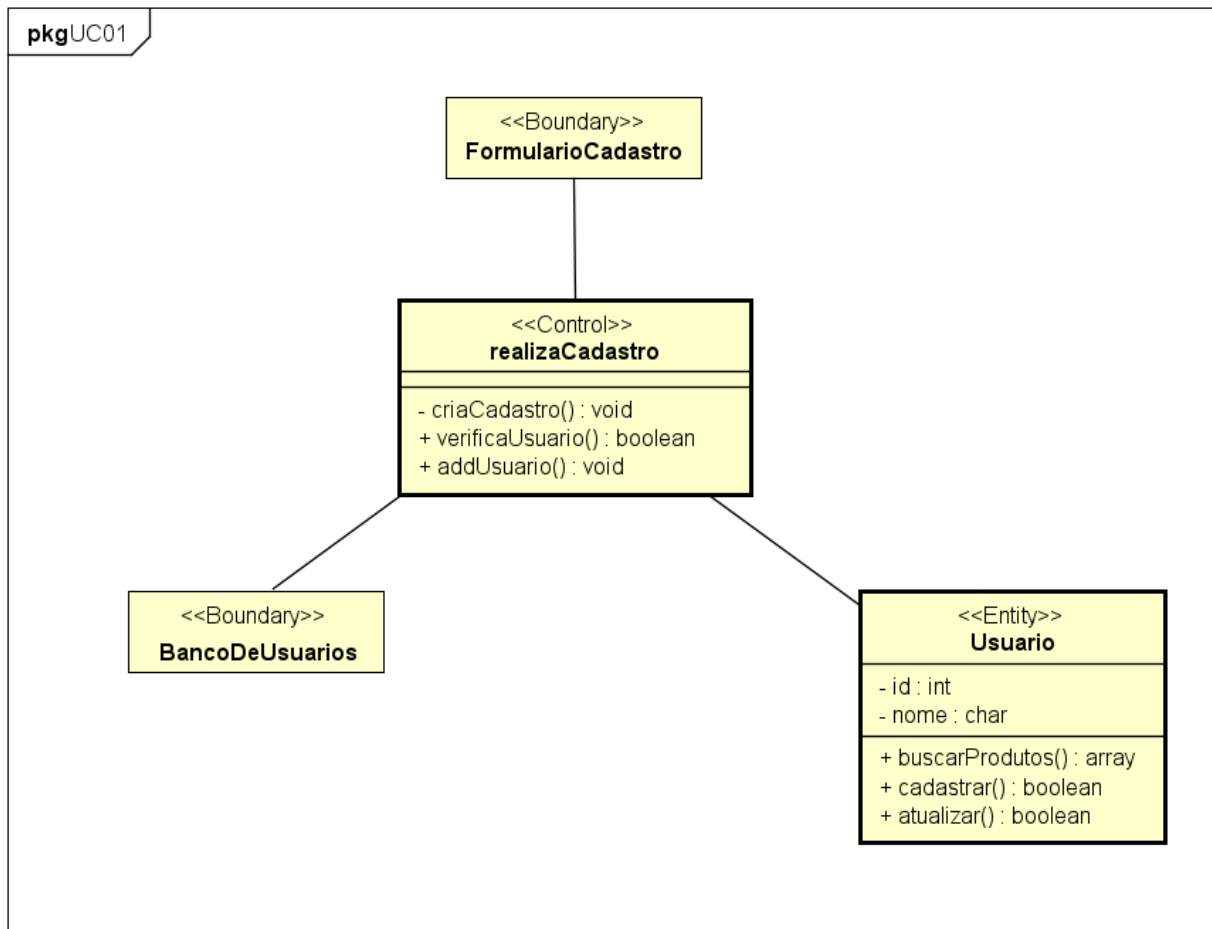


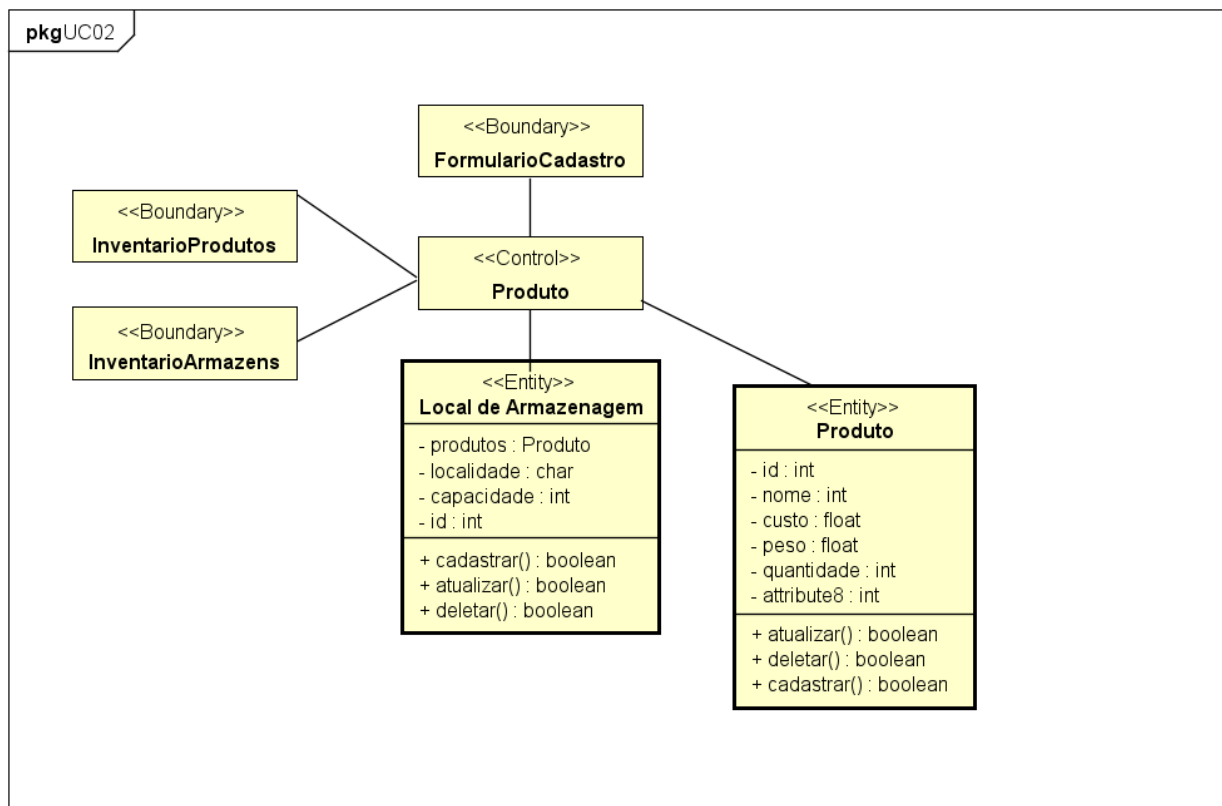
4.3 Diagrama de pacotes

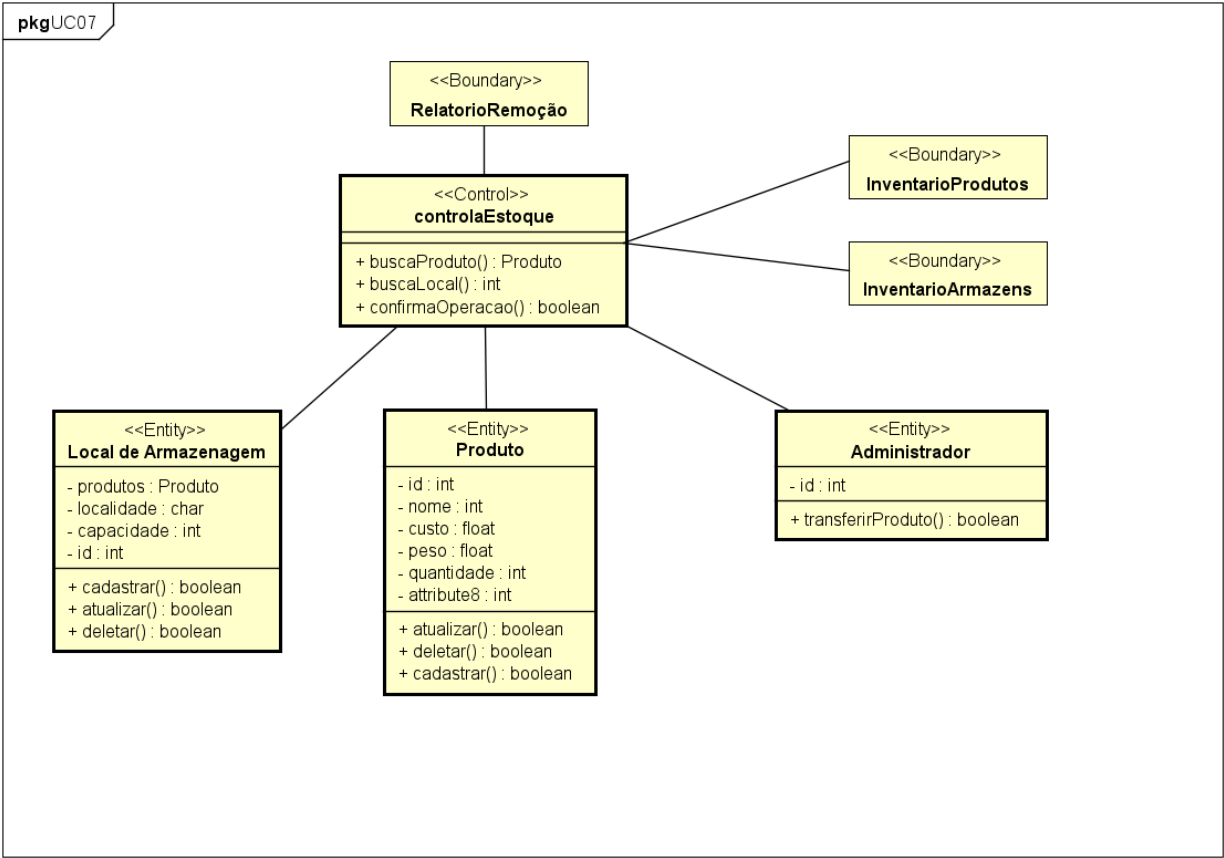


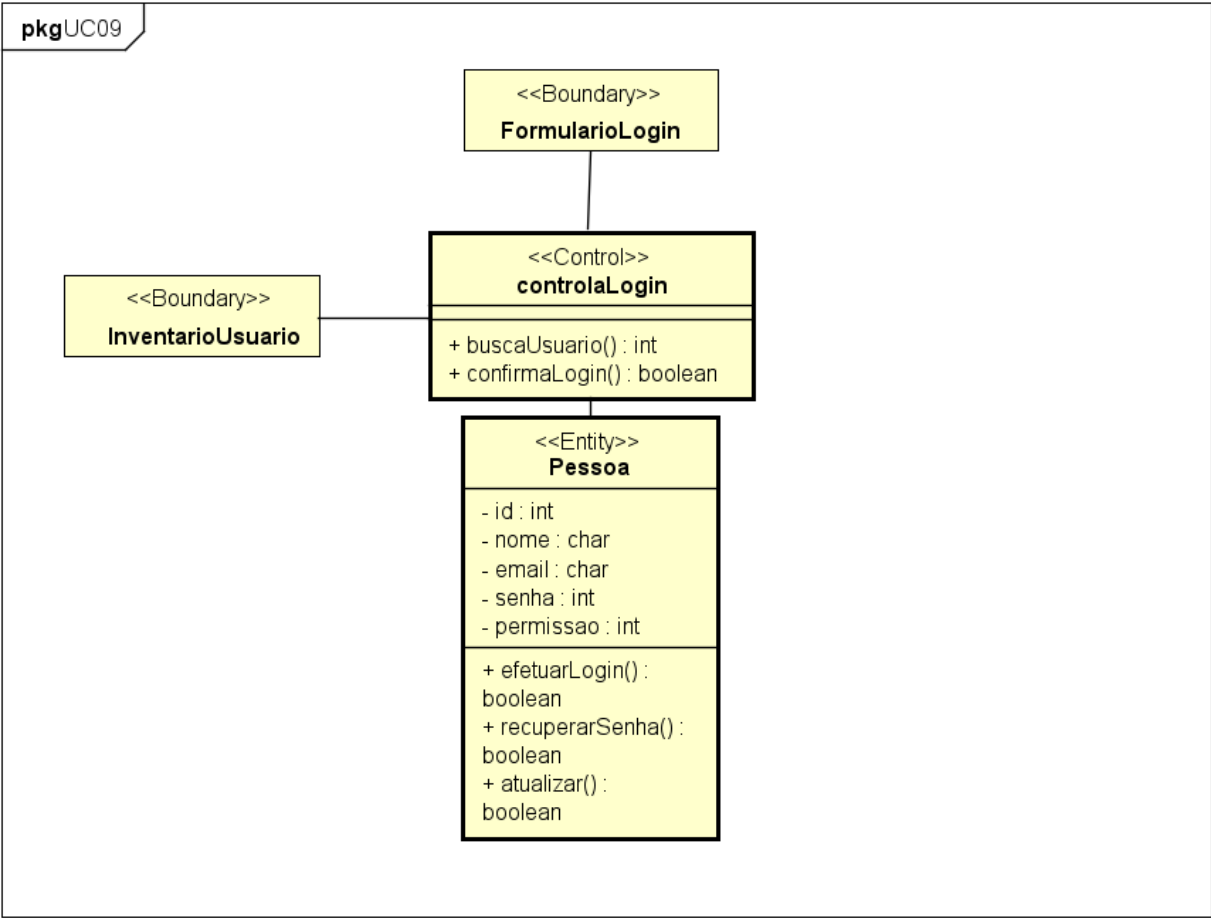
4.4 VCPs

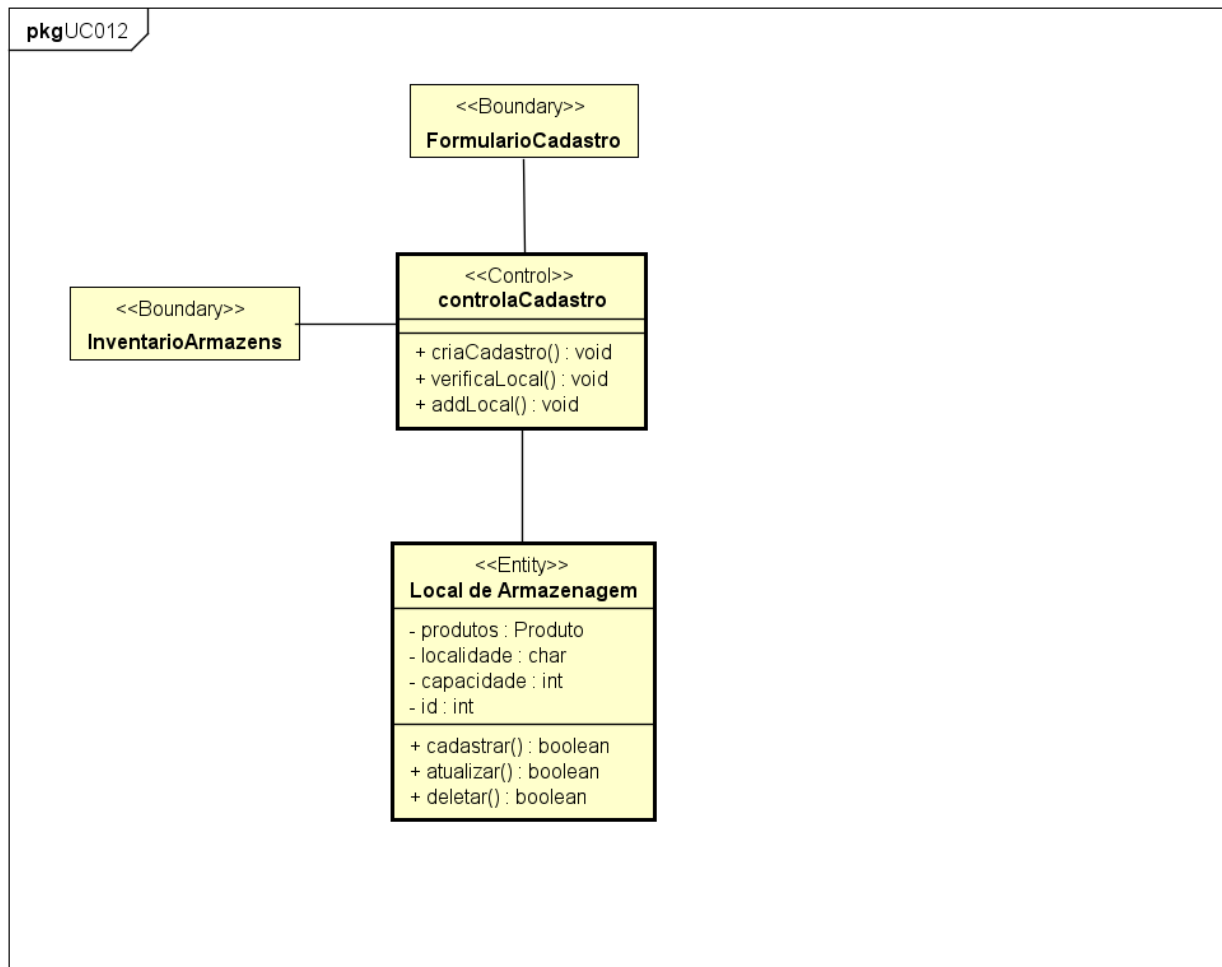
Fizemos diagramas VCPs de 6 casos de uso de nosso projeto.

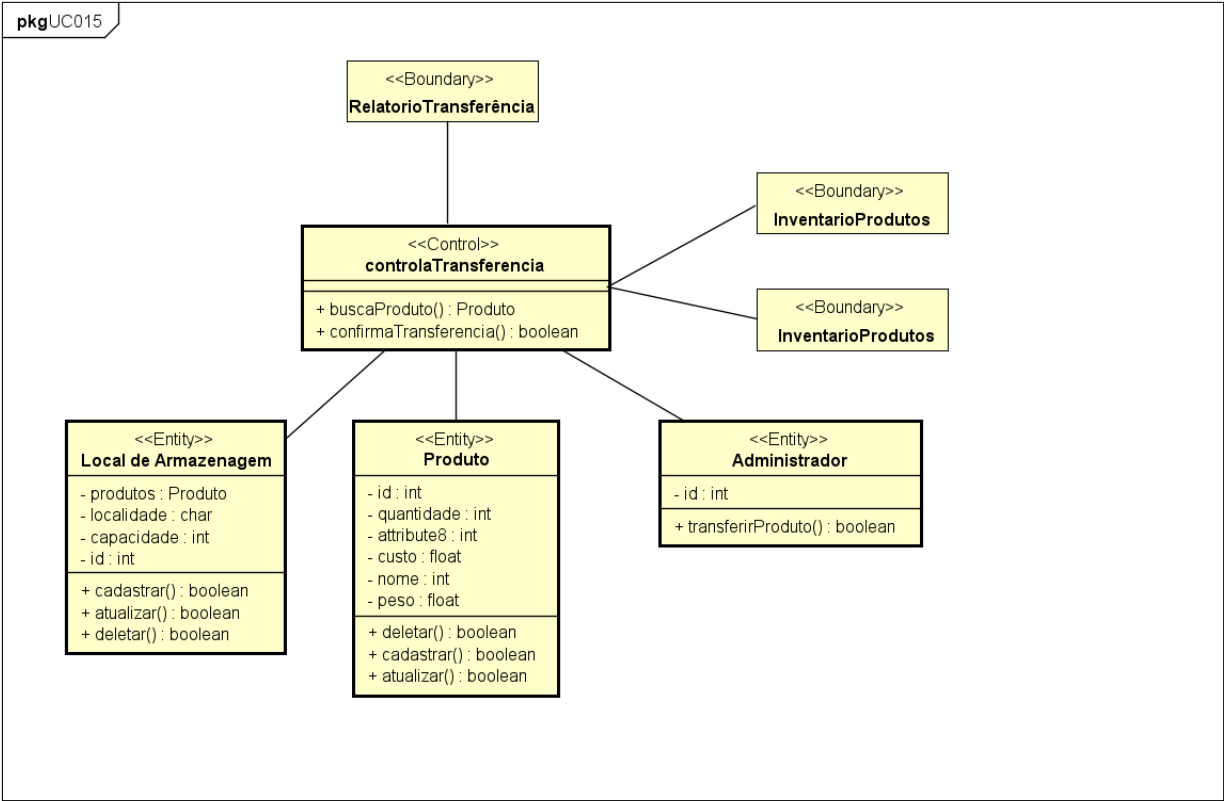




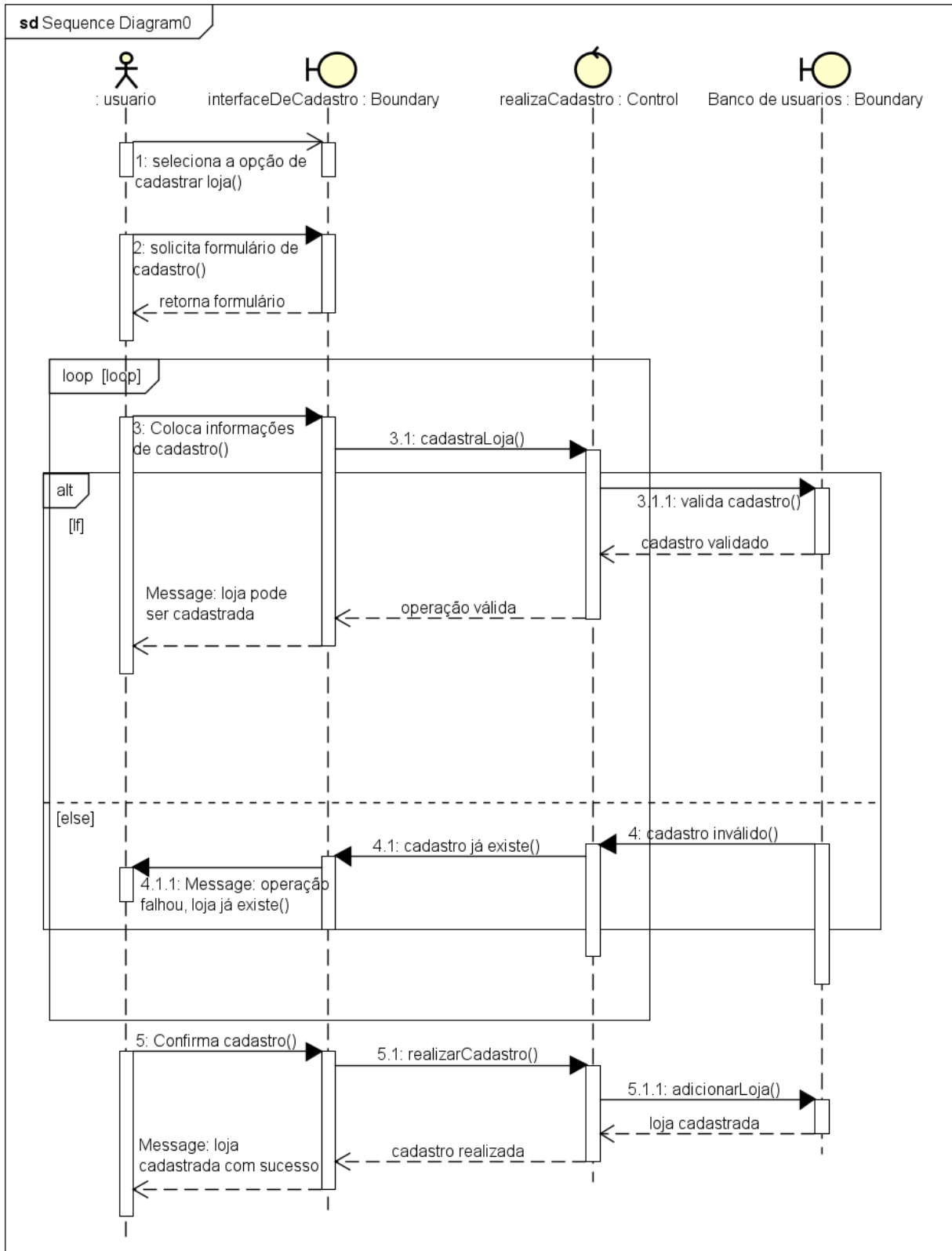


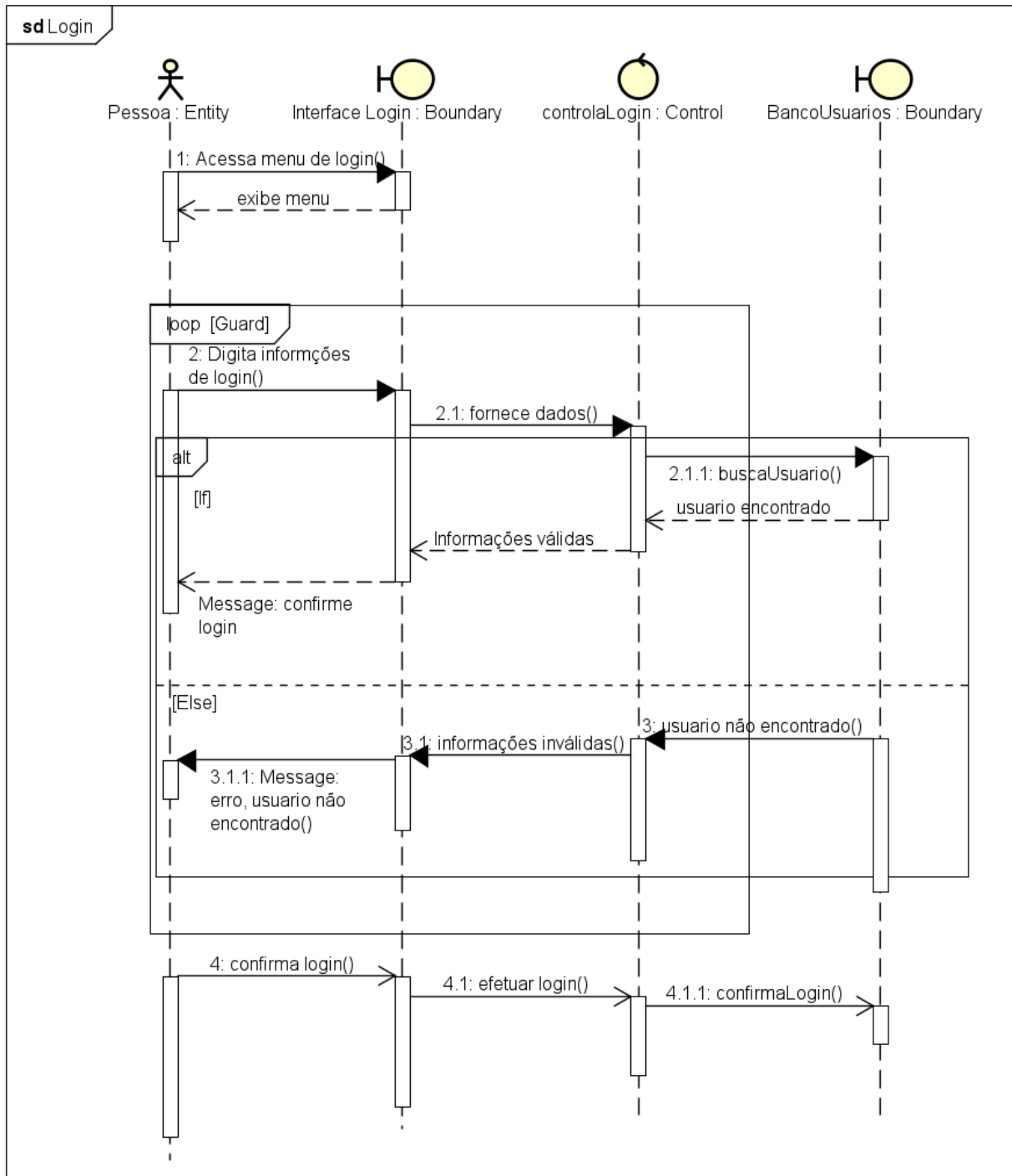




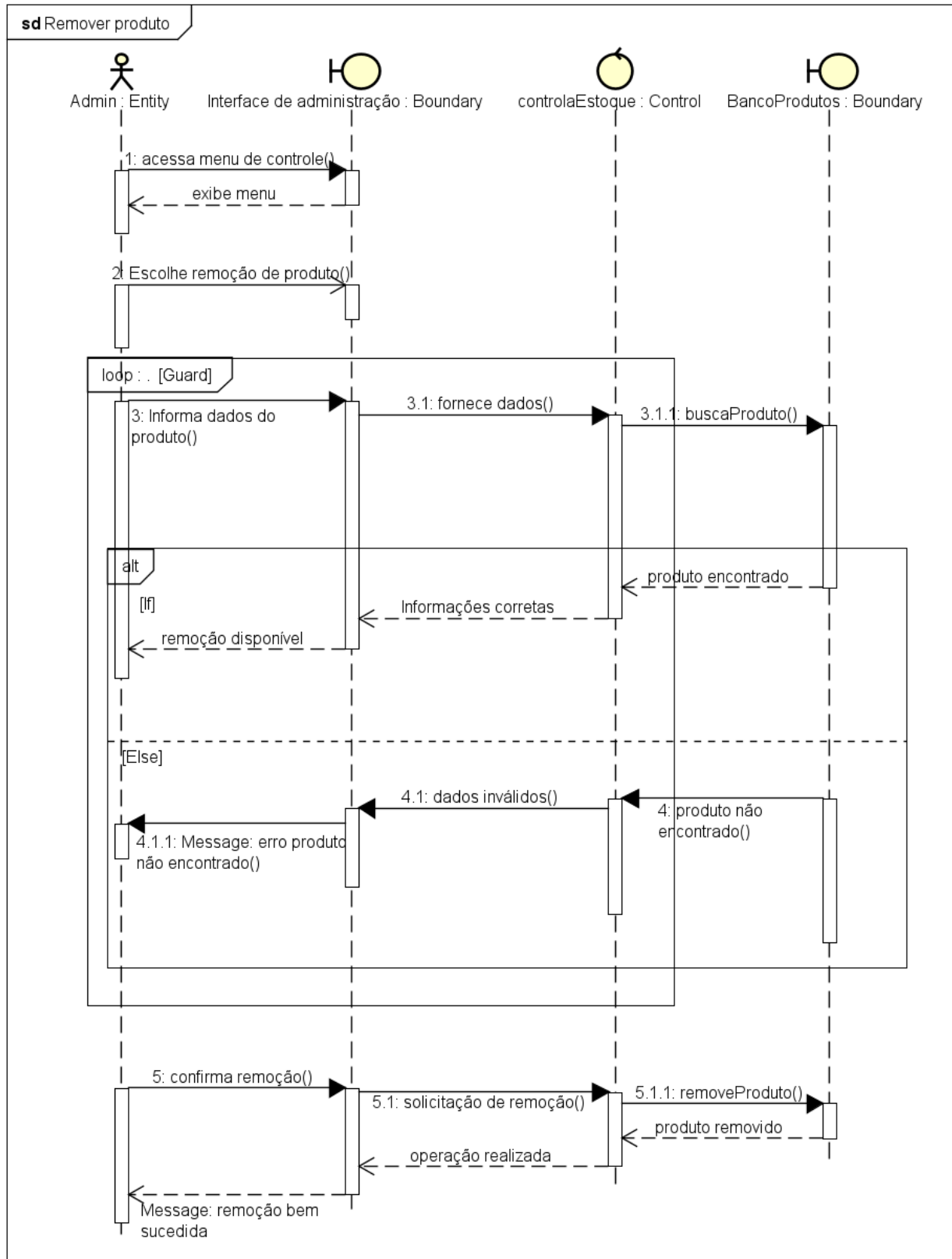


4.5 Diagramas Sequenciais





Grupo 02



Grupo 02

Controle de estoque	Version: <1.0>
Documento de Arquitetura de Software	Date: <dd/mmm/yy>
<document identifier>	

5. Camadas da Arquitetura

5.1 Padrões Arquiteturais

- **MVC (Model-view-controller):** O padrão MVC separa o projeto do software em três camadas independentes: o modelo (manipulação da lógica de dados), a visão (a interface do usuário) e o controlador (fluxo de aplicação). Esta separação permite o devido foco em cada camada do desenvolvimento, especialização, além de facilitar os testes e a manutenção do código, que pode ser reutilizado em outros projetos.
- **Layers (Camadas):** Os módulos e componentes do software são organizados em camadas de funcionalidades, que podem ser desconstruídas em diferentes serviços, permitindo também o desenvolvimento em paralelo por diferentes membros da equipe.
- **Microservices (microserviços):** O padrão se baseia em múltiplos serviços e componentes para desenvolver uma estrutura modular. Permite escalabilidade e independência dos módulos, que podem usar diferentes linguagens, o que seria adequado para o nosso projeto, uma vez que terão várias funcionalidades diferentes dentro de um mesmo software, armazenamento e manipulação de dados, transação e controle de mercadorias, rastreo, busca e etc.
- **Client-server (cliente-servidor):** Fornece a feature de recursos centralizados, já que o servidor está no centro da rede, ele pode gerenciar recursos comuns a todos os usuários como, por exemplo, um banco de dados centralizado, a fim de evitar os problemas de redundância e contradição. Esse modelo oferece também maior segurança, gerenciamento do servidor e uma rede evolutiva. Graças a esta arquitetura, é possível remover ou adicionar clientes sem perturbar o funcionamento da rede e sem modificar o essencial.

Controle de estoque	Version: <1.0>
Documento de Arquitetura de Software	Date: <dd/mmm/yy>
<document identifier>	

5.2 Padrões de Projeto

- **Singleton (Tiago Espanha)**

Esse padrão serve para garantir que uma classe possui somente uma instância e que essa instancia tenha acesso global. É útil quando queremos garantir que somente teremos um ponto de acesso a algo. Por exemplo, em geral não é proveitoso mais de um ponto de acesso ao mesmo banco de dados em uma aplicação, por isso criamos um único ponto de acesso com o padrão.

```

> Backend > JS InitDatabase.js > ...
...
1  import {Sequelize } from 'sequelize';
2
3  |  sequelize = undefined;
...
4
5  export const initDatabase = async () => {
6
7      if (!sequelize) {
8          sequelize = new Sequelize('postgres', 'root', 'ewqewq321', {
9              host: 'localhost',
10             dialect: 'postgres'
11         });
12
13         try {
14             await sequelize.authenticate();
15             console.log('Connection has been established successfully.');

```


Controle de estoque	Version: <1.0>
Documento de Arquitetura de Software	Date: <dd/mm/yy>
<document identifier>	

- **Observer** (Marcos Victor Ennes)

Intenção:

Os estabelecimentos devem se manter constantemente atualizados sobre a disponibilidade dos produtos disponíveis em cada local de armazenagem. Uma comunicação frequente com estes locais é necessária para se ter informações sobre a disponibilidade dos produtos.

Este padrão de projeto trás a possibilidade de notificar diversos estabelecimentos caso um produto esteja em falta em um local de armazenagem.

Caso se esgote um produto em um determinado local de armazenagem, todos os estabelecimentos que eram fornecidos por aquele local de armazenagem devem ser avisados para que busquem o produto desejado em outro local até que de fato cheguem mais produtos.

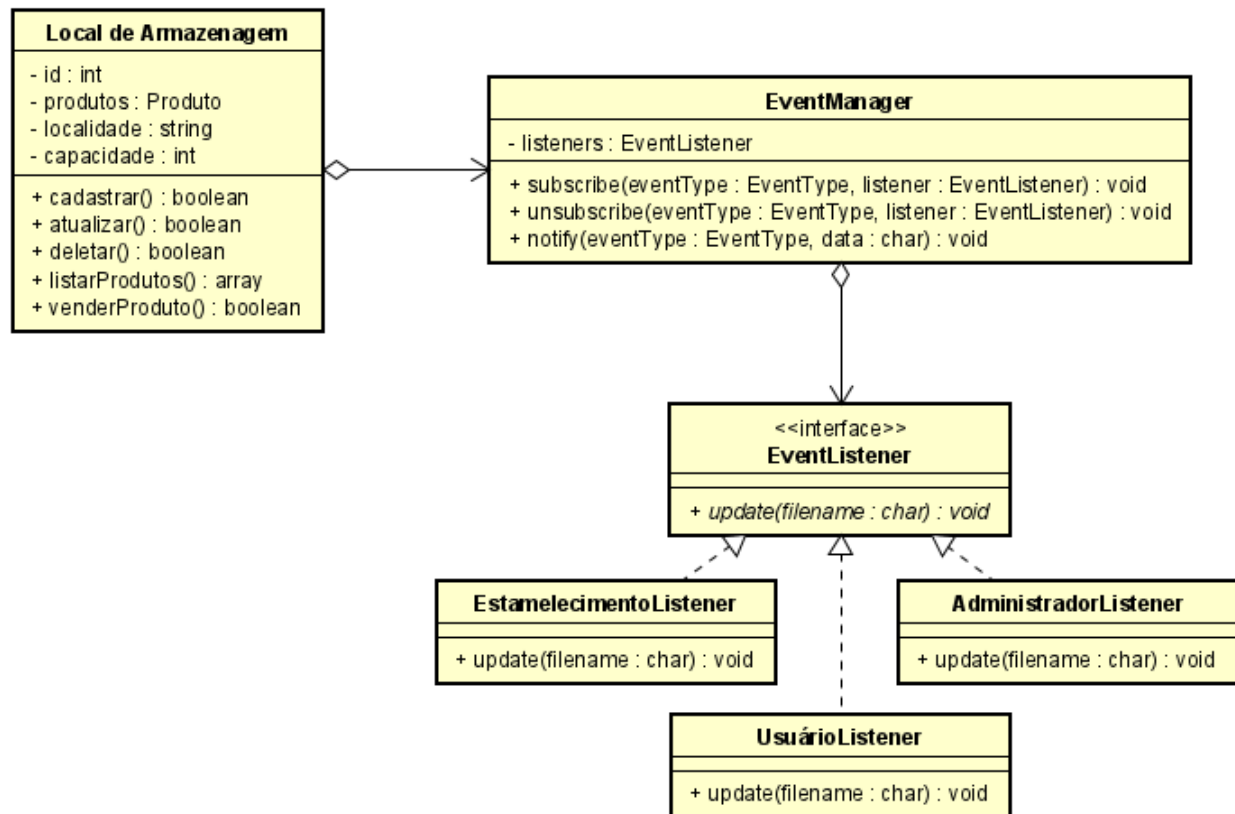
Motivação:

Este mecanismo evita um trabalho desnecessário de comunicação entre a loja e o local de armazenagem, uma vez que a loja não precisa se comunicar com o local de armazenagem constantemente para adquirir informações sobre os produtos.

Além disso, o local de armazenagem também não precisa informar o estabelecimento constantemente sobre a disponibilidade do produto uma vez que o padrão de projeto Observer irá solucionar este problema.

Controle de estoque	Version: <1.0>
Documento de Arquitetura de Software	Date: <dd/mmm/yy>
<document identifier>	

Estrutura:



Participantes e Colaborações:

O objeto Local de Armazenagem é o objeto que sofre as mudanças de estado. Caso haja a falta de um produto específico no local de armazenamento por exemplo.

O Objeto EventManager é responsável por informar os Listeners das alterações de estado que ocorreram com o Inscrito.

Controle de estoque	Version: <1.0>
Documento de Arquitetura de Software	Date: <dd/mmm/yy>
<document identifier>	

A Interface Event Listener define as funções que serão utilizadas pelos Listeners.

Os Listeners são objetos que são notificados e realizam as ações necessárias.