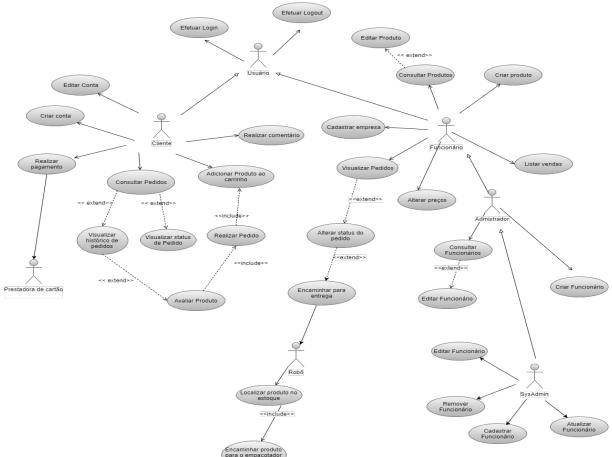
Natural Concept

Bárbara Lima -bclo Carolina Carneiro - ccrs João Ricardo - jrs4 Lucas Artur - lagos

Descrição

O Natural Concept é um sistema de comércio eletrônico de produtos naturais para beleza.

Diagrama de Casos de Uso



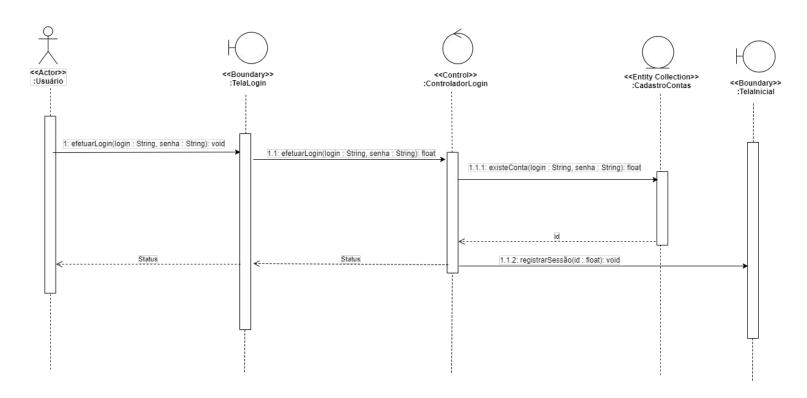
Análise

- 1. Efetuar Login
- 2. Realizar Pagamento
- 3. Adicionar Produto ao Carrinho
- 4. Criar Produto
- 5. Avaliar Produto

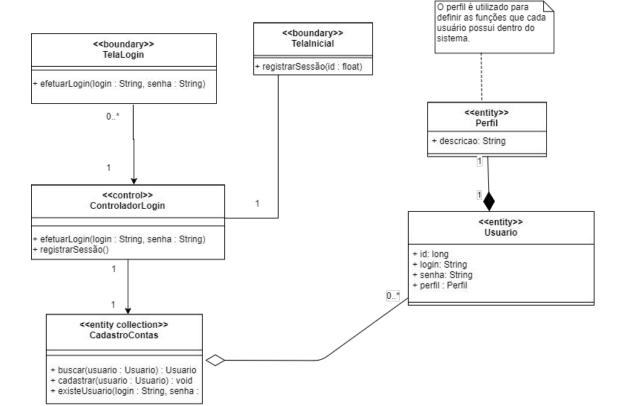
Efetuar Login : Descrição

[UC001]	Efetuar Login	
Descrição	Caso de uso responsável pelo login do usuário no sistema. Deve haver a validação de login e senha já previamente cadastrados.	
Atores	Usuário	
Prioridade	Essencial	
Pré -condições	Estar cadastrado no sistema	
Pós-condições	Ter acesso ao sistema	
Fluxo de eventos principais	 O usuário preenche os campos de login e senha O usuário clica no botão de entrar O sistema valida senha e login 	
Fluxo de eventos secundários	No passo 1, caso o usuário erre seu login ou senha, não conseguirá ter acesso ao sistema .	

Efetuar Login : Diagrama de Sequência



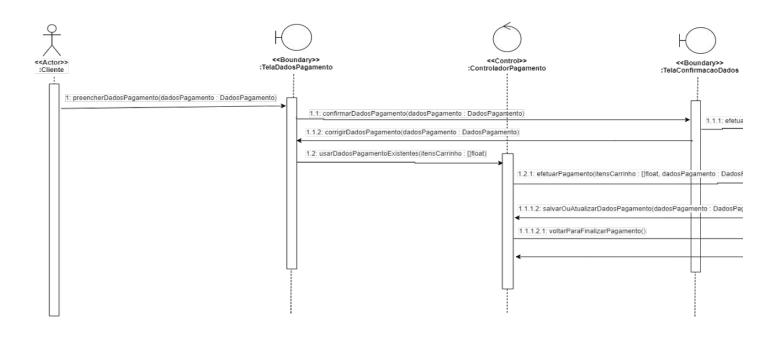
Efetuar Login : Diagrama de Classes



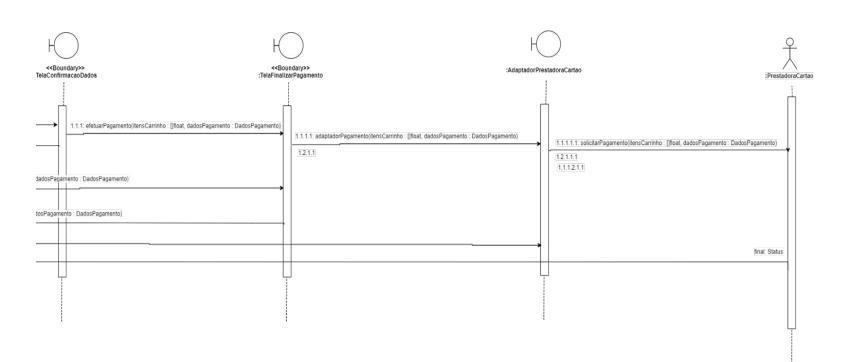
Realizar Pagamento : Descrição

[UC002]	Realizar Pagamento
Descrição	Caso de uso responsável pelo pagamento da compra dos itens no carrinho do Cliente.
Atores	Cliente, Prestadora do Cartão
Prioridade	Essencial
Pré -condições	Cliente faz login no sistema Cliente cria o pedido Dados do cartão Dados do endereço de entrega
Pós-condições	Recebe um retorno da situação do seu pagamento
Fluxo de eventos principais	O usuário logado clica em realizar pagamento, na janela do carrinho Preenche os dados da forma de pagamento Confirma os dados registrados Salva os dados, caso queira para uma próxima compra Recebe um retorno da situação do seu pagamento
Fluxo de eventos secundários	No item 3, caso queira Editar o registro, volta ao item 2, com os campos preenchidos com os dados. No item 5, caso a confirmação de situação venha negativada, para uso de cartão na compra, aparece mensagem para contatar a operadora do seu cartão, para ter mais detalhes. Ou escolher outra forma de pagamento.

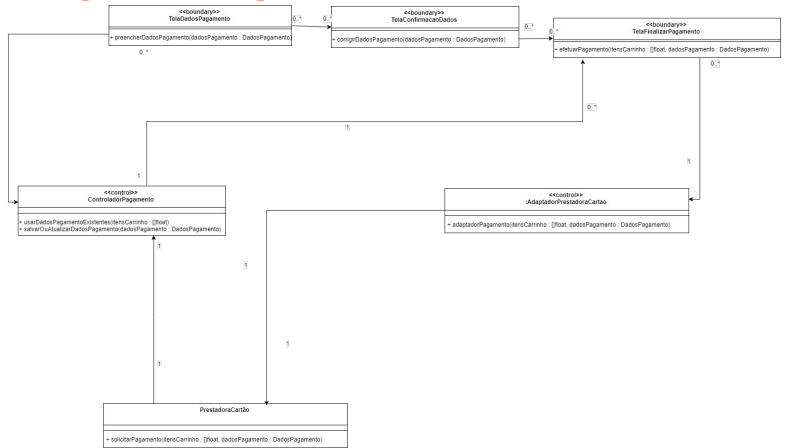
Realizar Pagamento : Diagrama de Sequência



Realizar Pagamento : Diagrama de Sequência



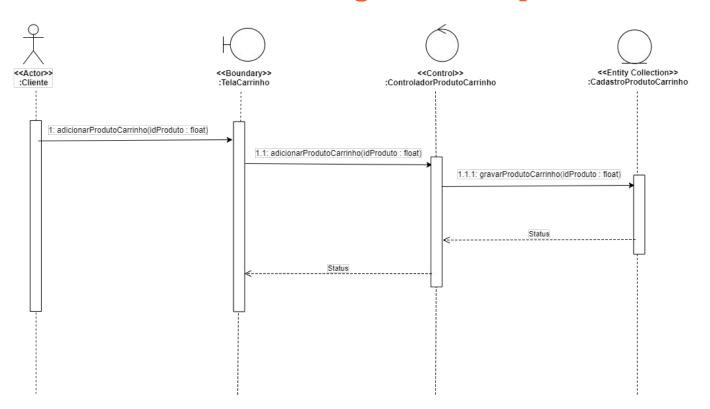
Realizar Pagamento : Diagrama de Classes



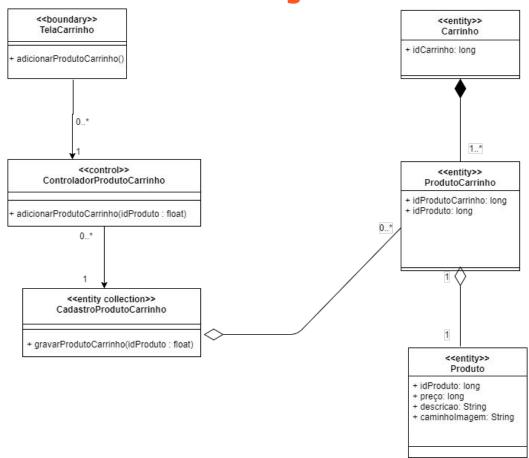
Adicionar Produto ao Carrinho : Descrição

[UC003]	Adicionar produto ao carrinho	
Descrição	Caso de uso responsável pela adição de produtos desejados pelo cliente ao carrinho de compras, que podem ser transformados em um pedido.	
Atores	Cliente	
Prioridade	Importante	
Pré -condições	Fazer login no sistema Produto pré cadastrado	
Pós-condições	Produtos adicionados no carrinho	
Fluxo de eventos principais	Cliente faz login no sistema Cliente procura produto Cliente na página de catálogos Cliente clica em adicionar produto no carrinho Produto adicionado no carrinho	
Fluxo de eventos secundários	No passo 3 o cliente pode cliente pode clicar em voltar sem adicionar produto no carrinho No passo 4 o produto pode está em falta e não é adicionado ao carrinho	

Adicionar Produto ao Carrinho : Diagrama de Sequência



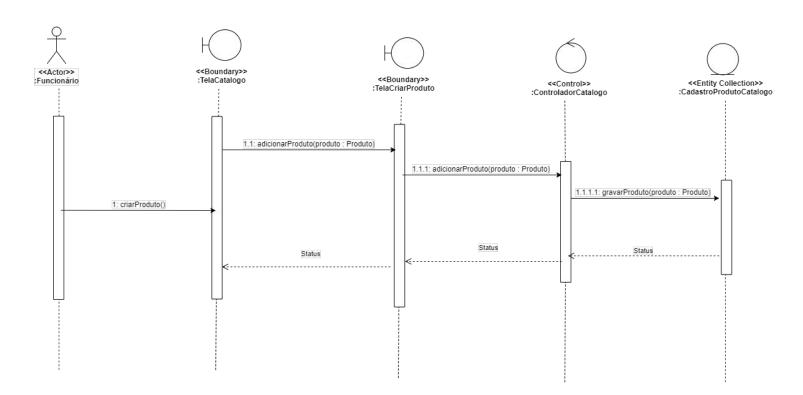
Adicionar Produto ao Carrinho : Diagrama de Classes



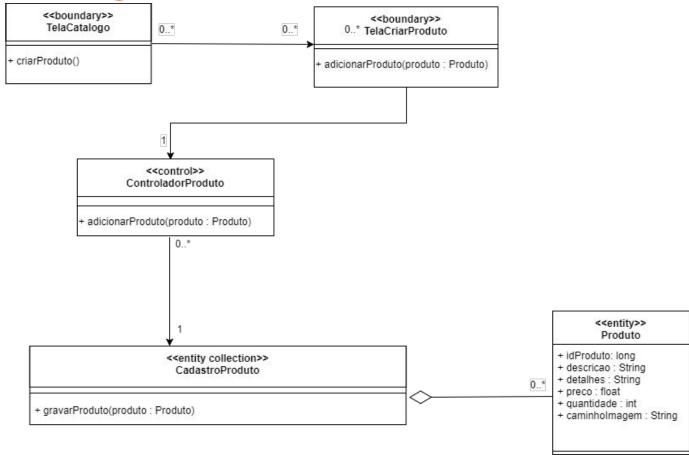
Criar Produto : Descrição

[UC004]	Criar produto	
Descrição	Caso de uso responsável pela criação de produtos pelo funcionário. Nessa etapa, os produtos que foram criados no sistema são adicionados ao catálogo.	
Atores	Funcionário	
Prioridade	Importante	
Pré -condições	Funcionário conectado	
Pós-condições	Lista de produtos no catálogo	
Fluxo de eventos principais	Funcionário faz login no sistema Cria Produto Produto é adicionado ao catálogo automaticamente	
Fluxo de eventos secundários	Sem fluxo registrado	

Criar Produto : Diagrama de Sequência



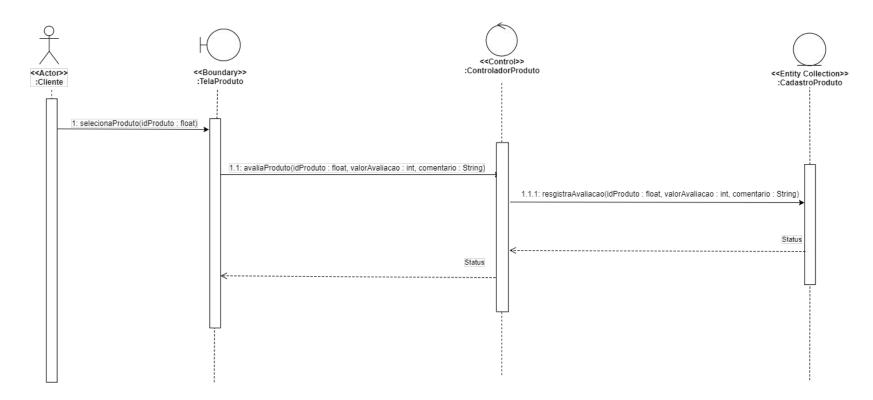
Criar Produto : Diagrama de Classes



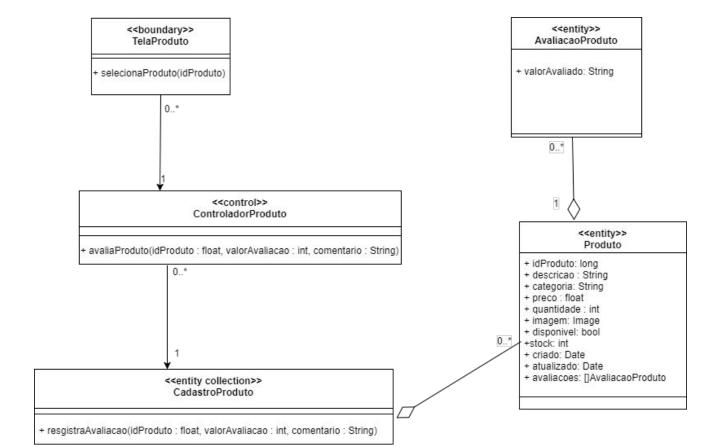
Avaliar Produto : Descrição

[UC005]	Avaliar Produto	
Descrição	Caso de uso responsável por avaliar um produto.	
Atores	Cliente	
Prioridade	Desejável	
Pré -condições	1. Fazer Login	
Pós-condições	Produto avaliado	
Fluxo de eventos principais	Cliente faz login Cliente na página de produto Cliente avalia produto(s)	
Fluxo de eventos secundários	No passo 3 o cliente pode fazer comentário sobre produto	

Avaliar Produto : Diagrama de Sequência



Avaliar Produto : Diagrama de Classes



Projeto

Mapeamento de Classes de Análise

Classes de Análise	Classes de Projeto
Tela Login	TelaLogin TelaLoginControle
Tela Inicial	TelaInicial TelaInicialControle
TelaDadosPagamento	TelaDadosPagamento TelaDadosPagamentoControle
TelaConfirmacaoDados	TelaConfirmacaoDados TelaConfirmacaoDadosControle
TelaCarrinho	TelaCarrinhoControle
TelaCatalogo	TelaCatalogo TelaCatalogoControle
TelaCriarProduto	TelaCriarProduto TelaCriarProdutoControle

Mapeamento de Classes de Análise

Classes de Análise	Classes de Projeto
TelaProduto	TelaProduto TelaProdutoControle
TelaFinalizarPagamento	TelaFinalizarPagamento TelaFinalizarPagamentoControle
	Fachada
ControladorLogin	ControladorLogin
ControladorPagamento	IPaymentSystem
ControladorProdutoCarrinho ControladorProduto	ControladorProduto
CadastroContas	CadastroContas RepositorioUsuarioBDR
CadastroPorduto	CadastroPorduto RepositorioProdutoBDR

Mapeamento de Classes de Análise

Classes de Análise	Classes de Projeto
AdaptadorPrestadoraCartao	AdaptadorPagamento
PrestadoraCartão	PayPalPaymentSystem (subsistema)
Usuário	Usuário
Perfil	Perfil
Produto ProdutoCarrinho Carrinho	Produto
AvaliacaoProduto	AvaliacaoProduto

Arquitetura do Sistema **View** *

<b <<pre><<pre>telaCarrinhoControle <
toundary>>
TelaCriarProdutoControle + selecionaProduto(idProduto) + preencherDadosPagamento(dadosPagamento : DadosPagamento) efetuarLogin(login : String, senha : String) registrarSessão() + registrarSessao(id : float) + adicionarProdutororoduto : Produto) Controller 0...* controladorProduto : ControladorProduto controladorLogin : ControladorLogin iPaymentSystem: iPaymentSystem - aducinnerProdutic(produte - Preduto) : Produto
- decignation - Produto - Preduto) : Produto
- removerProduto(derProduto - float) : void
- removerProduto(derProduto - float) : void
- staukszarProduto(produto - float) : Produto
- staukszarProduto(produto - float) : Produto
- staukszarProduto(produto - float) : Produto
- setuant.ogin(login - String, senha - String)
- setuant.ogin(login - String, senha - String) IPayment System preencherDadosPagamento(dadosPagamento: DadosPagamento)
 corrigirDadosPagamento(dadosPagamento: DadosPagamento)
 refetuarPagamento(lensCarrinho: [float, dadosPagamento: DadosPagamento)
 reformoarDatusPagamento(): int adicionarProdutoCarrinho(sessaoUsuario float, idProduto float)
 ineuarindudocCarrinho(sessaoUsuario float)
 ineuarindudocCarrinho(sessaoUsuario float)
 int. comentario floationario float efetuarLogin(login : String, senha : String)
 registrarSessāo()
 usarDadosPagamentoExistentes(itensCarrinho : []float)
 vsalvarOuAtualizarDadosPagamento(dadosPagamento : Dados efetuarPagamento(itensCarrinho : []float, dadosPagamento : DadosPagament Model subsystep <<entity collection>> CadastroProduto availaProduto(idProduto: float, vatorAvailacao: int, comentario: String)
 adicionarProduto(produto: Produto): Produto
 istar(): List-Produto:
 remover(idProduto: float): void
 atualizarProduto: Produto
 atualizarProduto: Produto + efetuarPagamento(itensCarrinho : []float, dadosPagamento : DadosPagamento) : int (Repositorial Isuaria RepositorioProdutoBDR availaProduto(idProduto: float, valorAvailacao: int, comentario: String, addicionarProduto(produto: Produto): Produto istar(): List-Produto istar(): List-Produto remover(idProduto: float): void abualizarProduto: Produto <<entity>> Usuario <<entity>> Produto idProduto: long preço: long descricao: String caminholmagem: String data: Date AvaliacaoProduto ≺<entity>

Diagrama de Classes - View e Controllers

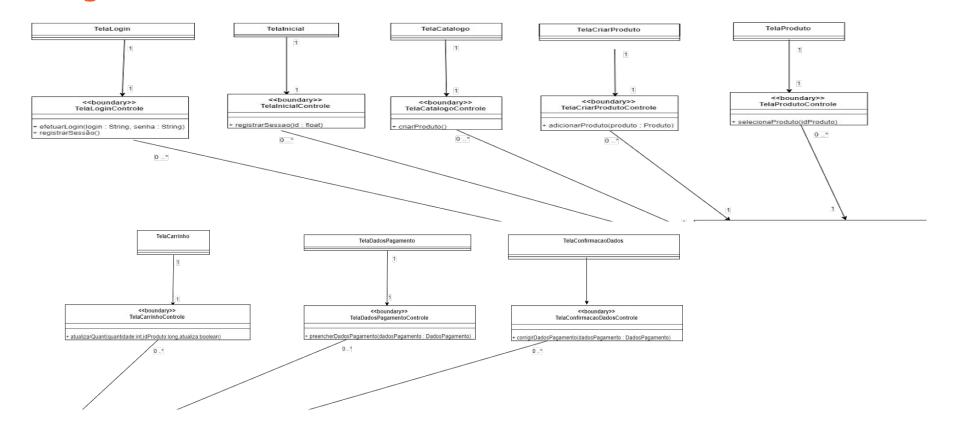


Diagrama de Classes - Model

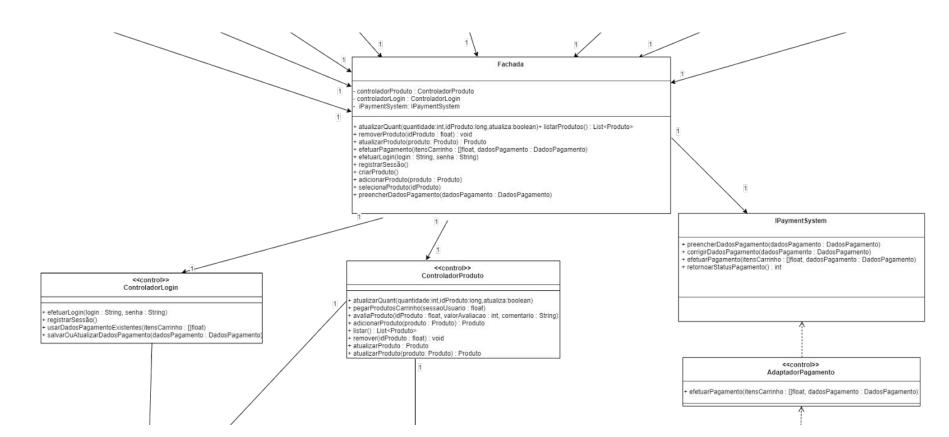
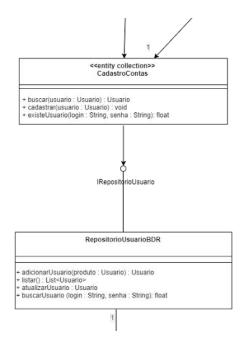
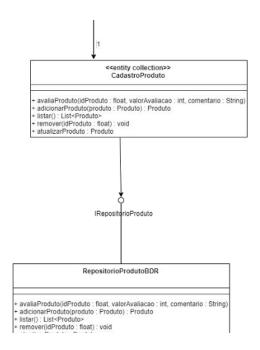


Diagrama de Classes - Model





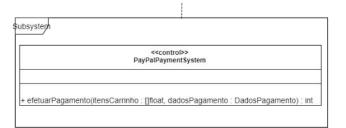
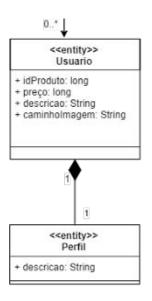


Diagrama de Classes - Model



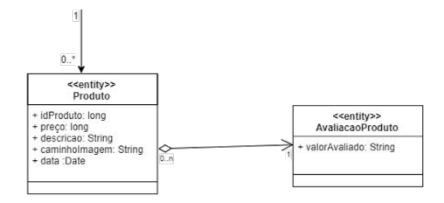
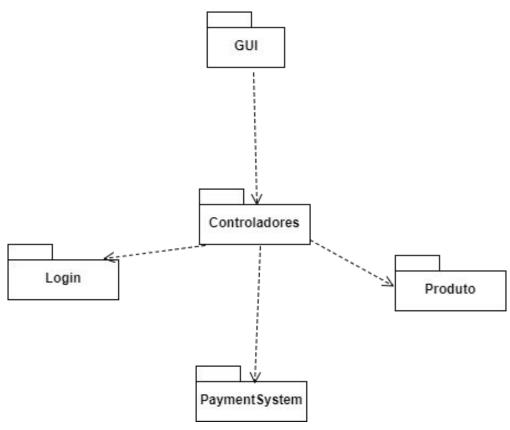
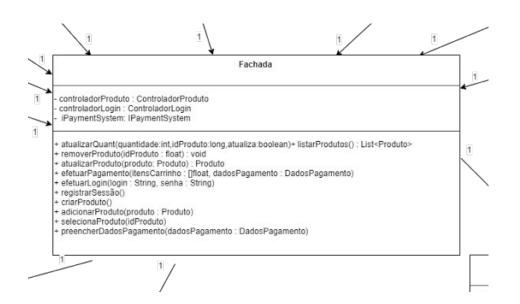
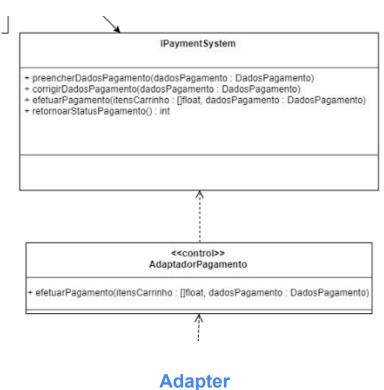


Diagrama de Pacotes



Padrão de Projetos





Fachada