
Grupo 3

E-Classroom
Documento de Arquitetura de Software

Versão 3.9

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

Histórico da Revisão

Data	Versão	Descrição	Autor
15/05/2022	1.0	Elaboração inicial do documento e Visões da Arquitetura (Diagramas de VCP)	Luisa Dirce, Daniel Fernandes
17/05/2022	1.1	Introdução e Qualidade	Lucas Pimenta, Lucas Couto
17/05/2022	1.2	Suposições e dependências	Marcos Roberto, Amanda Marques
17/05/2022	1.3	Metas e restrições da arquitetura; Requisitos arquiteturalmente significantes	Philippe Simões, Alexandre Calmon
17/05/2022	1.4	Decisões, Restrições e justificativas	Fábio Gil e Lucas Pimenta, Philippe Simões
18/05/2022	1.5	Visões da Arquitetura	Marcos Roberto e Daniel Fernandes
18/05/2022	1.6	Diagrama de casos de uso	Fábio Gil, Yuan Stewart
01/06/2022	2.0	Justificativa da escolha do padrão de arquitetura MVC	Lucas Pimenta, Luisa Dirce
02/06/2022	2.1	Definição de novos casos de uso e requisitos funcionais	Daniel Fernandes, Yuan Stewart
02/06/2022	2.2	Ajuste dos Diagramas VCP	Luisa Dirce, Daniel Fernandes
14/06/2022	2.3	Diagrama de Classe	Lucas Pimenta, Luisa Dirce
14/06/2022	2.4	Diagramas de Sequência (2, 4 e 6)	Fabio Gil, Yuan Stewart
14/06/2022	2.4.1	Diagramas de Sequência (1)	Marcos Roberto
14/06/2022	2.4.2	Diagramas de Sequência (3, 5 e 7)	Amanda Marques
29/06/2022	3.0	Padrão de projeto Abstract Factory	Amanda Marques e Fabio Gil
12/07/2022	3.1	Padrão de projeto Singleton	Luisa Dirce
12/07/2022	3.2	Padrão de projeto Chain of Responsibility	Daniel Fernandes
13/07/2022	3.3	Padrão de projeto Adapter	Lucas Pimenta
13/07/2022	3.4	Padrão de projeto Strategy	Lucas Couto
13/07/2022	3.5	Padrão de projeto Observer	Philippe Simões

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

13/07/2022	3.6	Padrão de projeto Command	Alexandre Calmon
13/07/2022	3.7	Padrão de projeto State	Marcos Roberto
13/07/2022	3.8	Padrão de projeto Observer (segundo cenário)	Yuan Stewart
13/07/2022	3.9	Padrão de projeto Abstract Factory (segundo cenário)	Fábio Gil

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
2.	Metas e Restrições da Arquitetura	4
3.	Suposições e Dependências	4
4.	Requisitos Arquiteturalmente Significantes	5
5.	Decisões, Restrições e justificativas	5
6.	Mecanismos Arquiteturais	5
7.	Camadas da Arquitetura	5
8.	Visões da Arquitetura	6
8.1	Diagrama de Casos de Uso	11
8.2	Visões de Classes Participantes (VCP)	13
8.3	Diagrama de Classes	18
8.4	Diagrama de Sequência	19
9.	Qualidade	16

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

Documento de Arquitetura de Software

1. Introdução

O presente documento tem como objetivo apresentar as características da arquitetura do projeto E-Classroom. Este sistema tem como objetivo a criação de um ambiente propício para a aquisição de cursos, para o desenvolvimento acadêmico e profissional, de maneira a facilitar o acesso ao conhecimento.

1.1 Finalidade

A finalidade do documento aqui apresentado, é oferecer todas as informações de forma clara e objetiva de acordo com as visões arquiteturais as quais são vinculadas ao sistema E-Classroom. Assim, apresentando características responsáveis pelo controle das atividades arquiteturais, dessa maneira trilhando o caminho o qual será percorrido para o desenvolvimento do sistema.

1.2 Escopo

Este documento ajuda os envolvidos no projeto e possíveis novos membros da equipe a compreender aspectos arquiteturais do sistema que são necessários para o desenvolvimento de uma solução satisfatória que atenda às necessidades dos usuários finais do E-Classroom. Portanto, também é útil para permitir maior facilidade no desenvolvimento de melhorias e correções através de revisões do documento em questão.

1.3 Definições, Acrônimos e Abreviações

MVC – Padrão de arquitetura de software onde M significa modelo sendo responsável pela parte de regras de negócio, V a visualização responsável pela parte de interfaces e C a parte de controle dos dados

2. Metas e Restrições da Arquitetura

Existem algumas metas e restrições de requisito e de sistema principais que têm uma relação significativa com a arquitetura, sendo elas:

- Utilização do paradigma Orientado a Objetos para o desenvolvimento;
- Estrutura MVC;
- Linguagem de programação Python
- Framework Flask;
- O sistema em questão deverá ser responsivo;
- Bancos de dados PostgreSQL.

3. Suposições e Dependências

▪ Experiência com o Framework Escolhido

Apesar da escolha do micro framework Flask ter envolvido a experiência de alguns integrantes com o mesmo, a maioria dos integrantes possui maior familiaridade com outro framework (Django), porém, as características e pontos positivos pesaram na decisão final. Com isso, há uma certa dependência no aspecto do desenvolvimento do software em relação aos integrantes que possuem maior experiência com Flask.

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

4. Requisitos Arquiteturalmente Significantes

- O sistema deve armazenar as senhas nos bancos de dados de forma segura
- O sistema deve estar disponível para os usuários a qualquer hora do dia
- O sistema deve ser implementado na linguagem Python
- O sistema deve verificar automaticamente a conclusão do curso através do monitoramento do tempo assistindo as aulas
- O sistema deve estar de acordo com os princípios SOLID
- O sistema deve utilizar autenticação por OAuth 2.0 Bearer Token
- O sistema deve ser responsivo, sendo funcional em desktops, notebook, celulares e tablets

5. Decisões, Restrições e Justificativas

▪ Escolha da linguagem Python

A decisão de utilizar a linguagem de programação Python para o desenvolvimento desse sistema ocorreu a partir de uma votação realizada por todos os integrantes do Grupo 3. Após a votação, o fato da maioria dos membros da equipe terem tido contado com esta linguagem, além de alguns serem experientes com frameworks Python (Django ou Flask), fez com que o resultado da votação fosse definitivo.

▪ Escolha do micro framework Flask

A decisão de utilizar Flask para o desenvolvimento do sistema ocorreu como complemento à decisão da escolha de Python como sendo a linguagem de programação, visto que alguns integrantes do grupo possuem experiência considerável com o framework em questão. Além disso, as características de flexibilidade, controle e ferramentas que o Flask possui, são adequadas aos requisitos e metas definidas para o desenvolvimento do sistema.

6. Mecanismos Arquiteturais

Padrões de Arquitetura

Padrão de arquitetura MVC

Considerando os requisitos de arquitetura, a decisão de utilizar o padrão de arquitetura MVC consiste no fato de se tratar de um padrão que facilita a resolução de problemas devido à sua organização, o que facilita a garantia de diversos requisitos não-funcionais do sistema, como a garantia de consistência de dados armazenados, ser adaptável à linguagem utilizada etc. Por se tratar de um projeto com muitos integrantes, o padrão de arquitetura MVC é benéfico ao favorecer a especialização do trabalho de desenvolvimento, ou seja, permitindo que cada integrante do grupo tenha um papel bem definido sobre suas responsabilidades envolvendo a implementação. Além disso, existem outros fatores que auxiliaram na decisão de utilizar esse padrão arquitetural, como o fato de permitir que classes de Modelo sejam usadas por diferentes Visões, além de favorecer a testabilidade.

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

Padrão de arquitetura em Três Camadas

O padrão de arquitetura em Três Camadas foi escolhido para dividir o projeto em camadas diferentes, e com isso, cada camada pode ser desenvolvida simultaneamente pela equipe de desenvolvimento, já que cada camada executa sua própria estrutura. Dessa forma, a plataforma E-classroom possui o servidor web como a camada de apresentação, o servidor de aplicativos como a camada do meio, que faz as consultas no banco de dados, e o servidor de banco de dados como a camada de dados (back-end) que é executada no MySQL.

Padrão de arquitetura Publish/Subscribe

O padrão de arquitetura Publish/Subscribe foi escolhido pensando nas funcionalidades assíncronas do projeto, como o envio de confirmação da compra de um curso para o usuário e a notificação sobre a confirmação da compra para o sistema para que disponibilize o curso para o usuário.

Padrão de arquitetura Cliente-Servidor

A arquitetura cliente servidor foi escolhida devido à necessidade dos clientes acessarem, por meio do navegador, os recursos armazenados e providos pelo servidor, que são os cursos postados na plataforma. Além disso, essa arquitetura facilita manutenções e fornece mais segurança.

Padrões de projeto

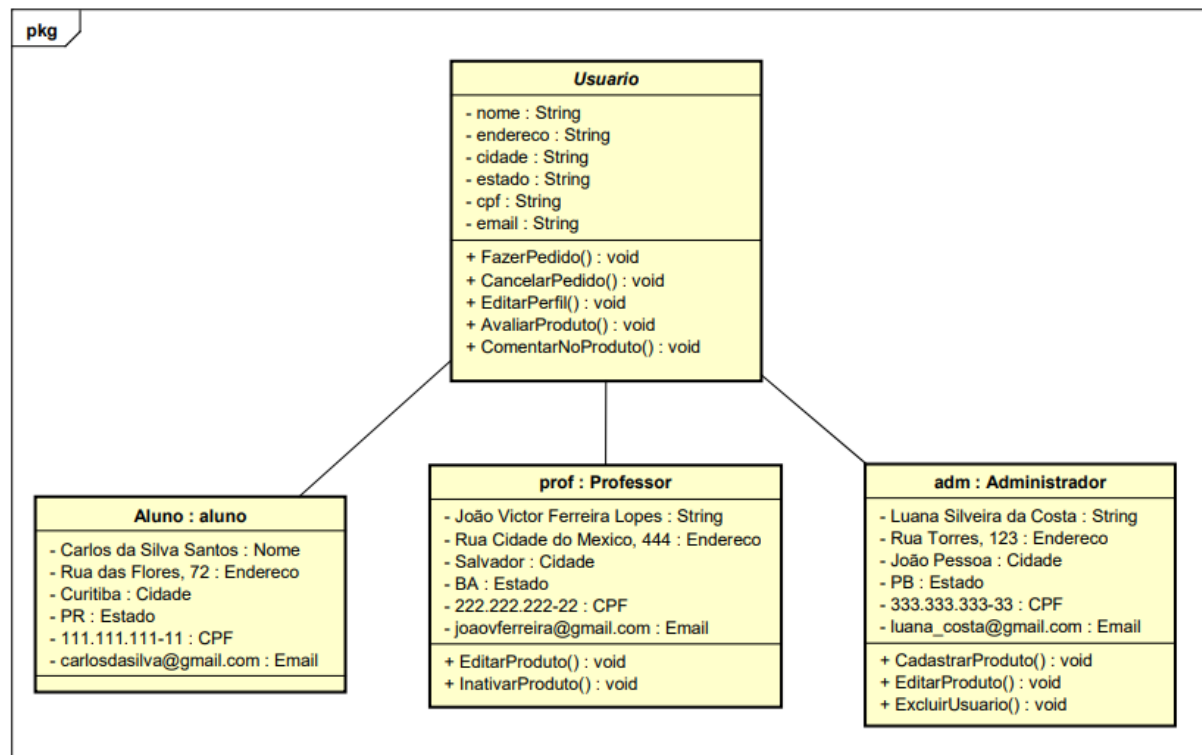
Abstract Factory

Motivação da escolha do padrão

A escolha do padrão Abstract Factory foi motivada pelas três variações de tipos de usuário dentro da plataforma: professor, aluno e administrador. Sendo assim, a utilização de uma classe abstrata que possui todos os atributos comuns a todos os usuários e a criação de subclasses herdando seus métodos, e possuindo seus respectivos métodos exclusivos, faz sentido na hora da escalabilidade do código.

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



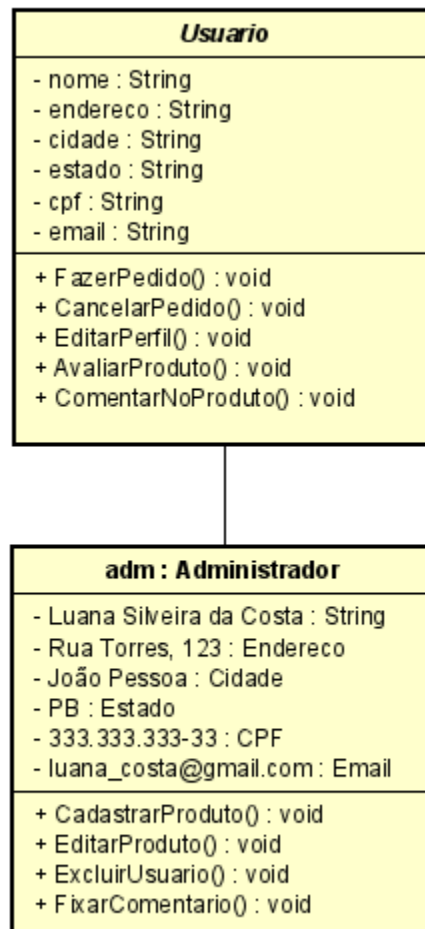
Singleton

Motivação da escolha do padrão

Escolhemos o padrão Singleton para instanciar a classe Professor apenas uma vez para garantir que só ele pode fixar um comentário de um ou mais usuários nos cursos criados por ele no topo da seção de comentários, semelhante ao que acontece no Instagram, por exemplo.

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



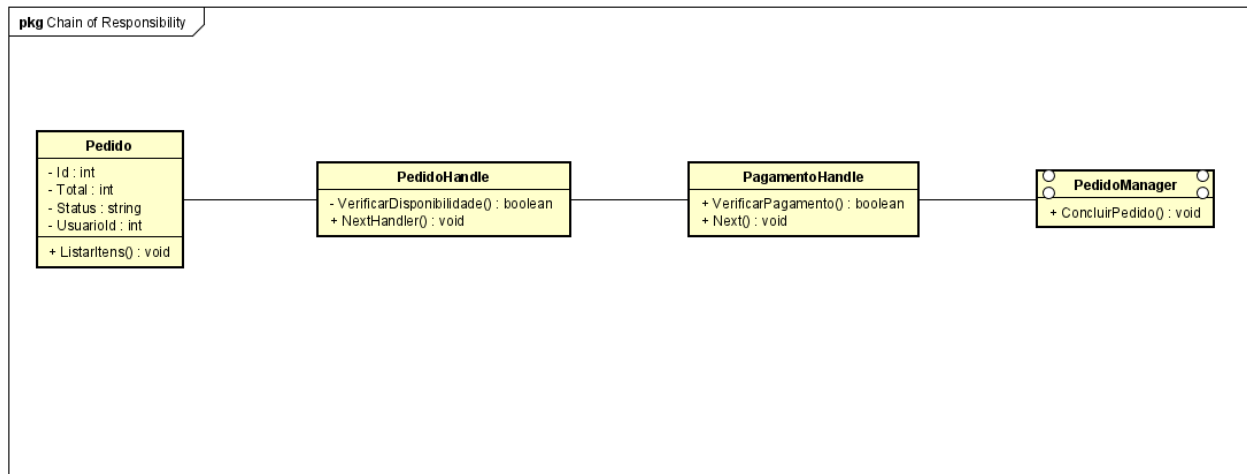
Chain of Responsibility

Motivação da escolha do padrão

A escolha do padrão Chain of Responsibility foi motivada pelo requisito de segurança no momento da compra, é necessário haver as verificações sequencialmente, caso alguma dessas verificações falhem, o processo é interrompido e tratado de forma correta, mantendo assim a segurança da compra.

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

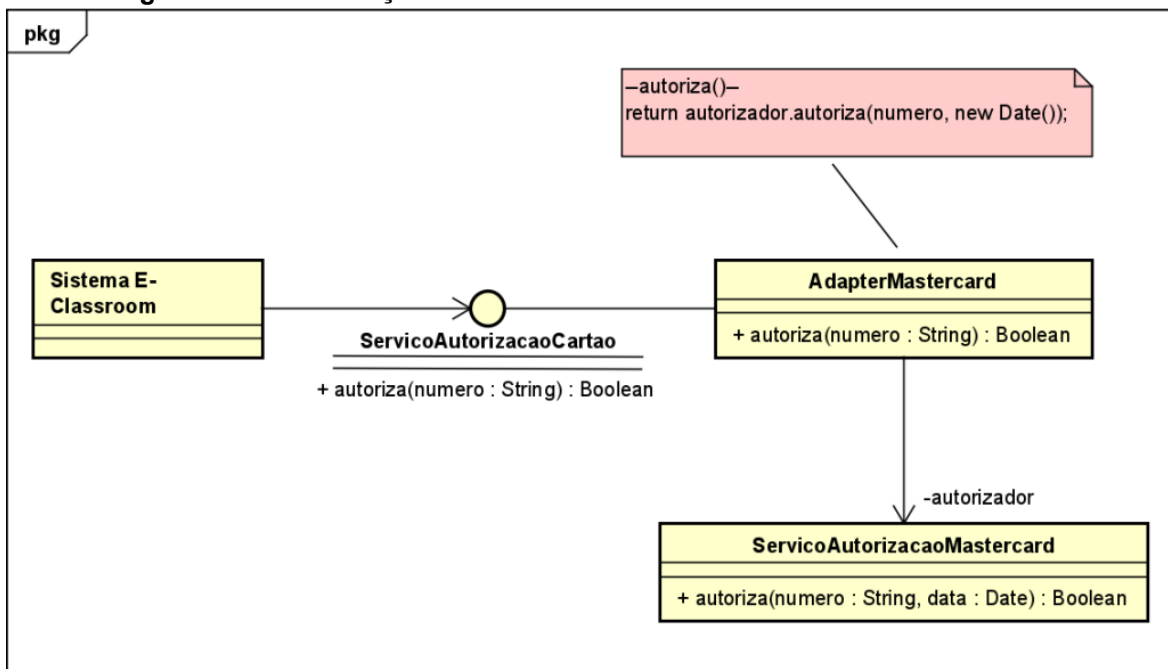


Adapter

Motivação da escolha do padrão

A escolha do padrão Adapter foi motivada levando em consideração um possível cenário em que ocorre uma incompatibilidade de classes quando um usuário precisa realizar o registro do cartão bancário no sistema (considerando um cenário em que há essa opção a fim de que o usuário possa realizar os pagamentos dos cursos comprados com maior facilidade). Portanto, para solucionar esse problema, o padrão Adapter permitiria que tais classes incompatíveis trabalhassem em conjunto.

Diagrama de instanciação



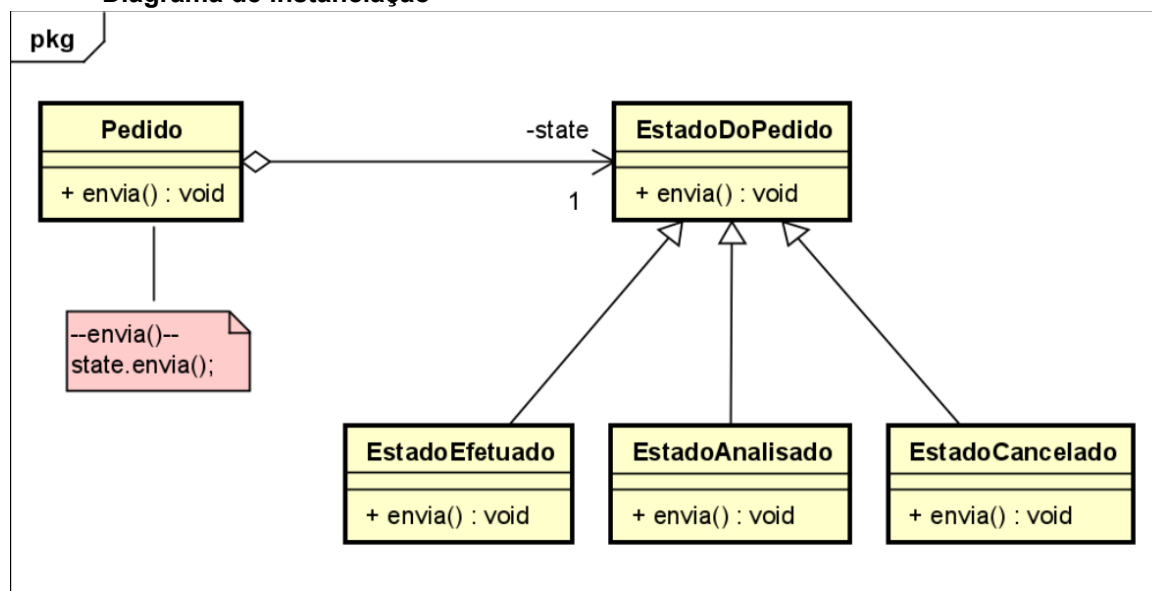
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

State

Motivação da escolha do padrão

A escolha do padrão State foi motivada pelo cenário de uma pessoa realizar um pedido de 1 ou mais produtos no curso e após isso o pedido poderia ter diversos estados (Efetuado, Análise, Cancelado). Nesse sentido, seria criada uma classe abstrata que teria definido todos os métodos de transição de estados do pedido. Assim, o código não ficará amarrado em ifs e elses, deixando a manutenção mais simples e não irá atrapalhar o funcionamento do que já estava validado.

Diagrama de instanciã o



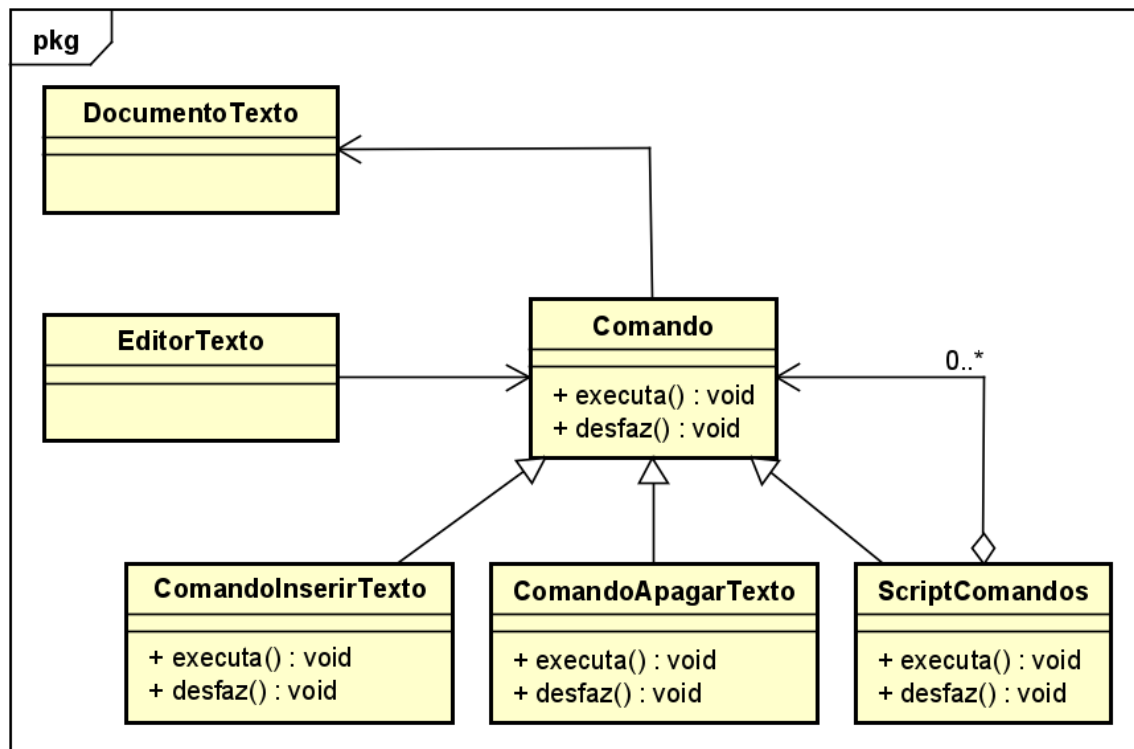
Command

Motivação da escolha do padrão

A escolha do padrão Command foi pensada considerando um cenário em que o professor autor de um curso, ao ter acesso a uma edição dos textos que se referem a atributos do curso (descrição, título, autor(es), etc), tivesse a possibilidade de desfazer uma ou mais ações efetuadas. Além de procurar permitir que o editor possa funcionar de forma não iterativa, através de um roteiro de comandos.

Diagrama de instanciã o

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



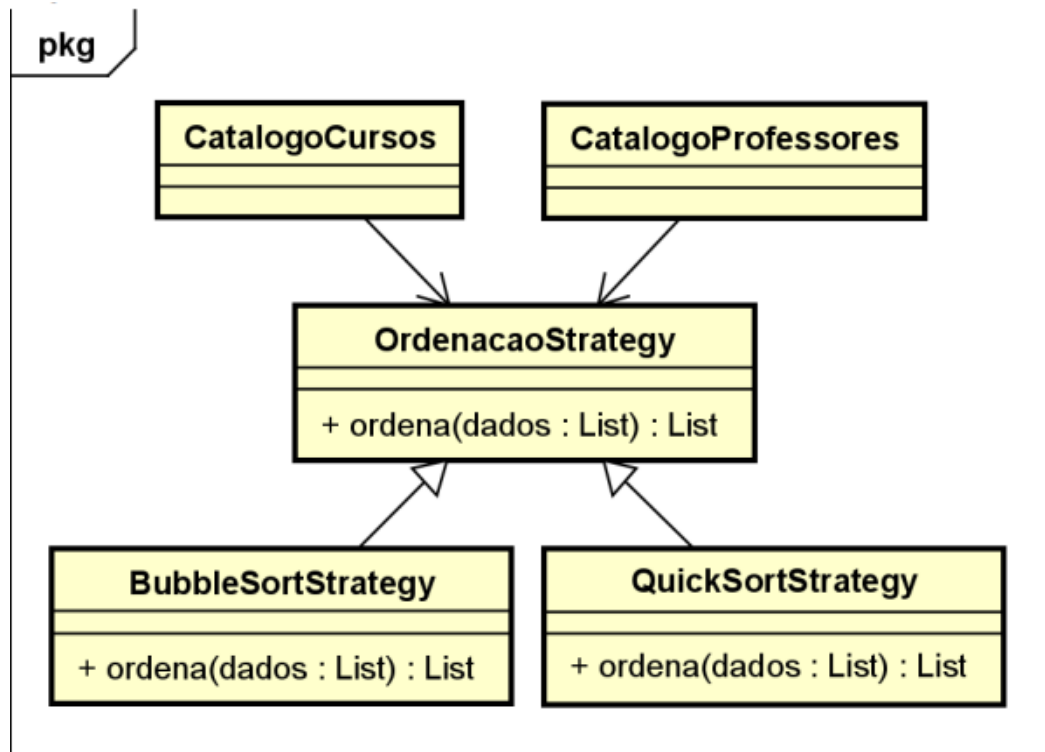
Strategy

Motivação da escolha do padrão

A escolha do padrão Strategy foi motivada através de situações em que algoritmos se repetem com frequência, como nas situações em que se quer ordenar objetos da classe 'CatalogoCursos' ou objetos da classe 'CatalogoProfessores', por exemplo, o que torna necessária uma certa frequência da utilização de algoritmos de ordenação. Para solucionar isso, o padrão Strategy procura isolar tais algoritmos a fim de facilitar a manutenção.

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



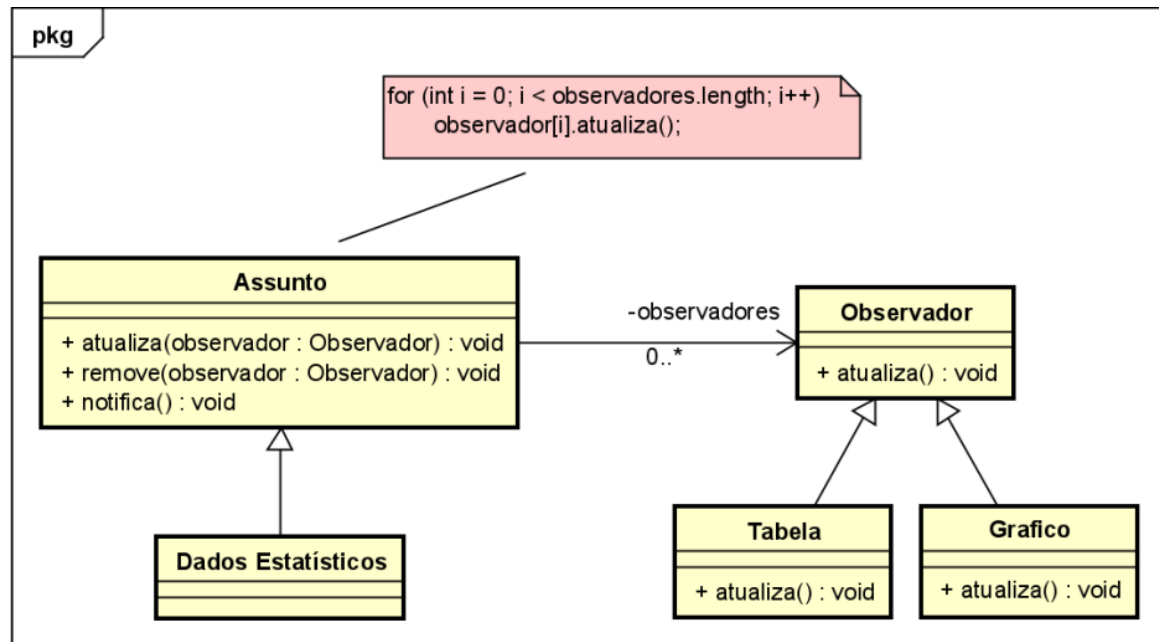
Observer

Motivação da escolha do padrão

A escolha do padrão Observer foi motivada através de um cenário em que as informações de um curso pudessem ser visualizadas pelo usuário de maneiras diferentes no catálogo de cursos da plataforma e na página específica do curso (exemplo: visualização de nível de aprovação exibido em "5 estrelas" no catálogo, enquanto na página específica essa informação é exibida com mais detalhes).

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



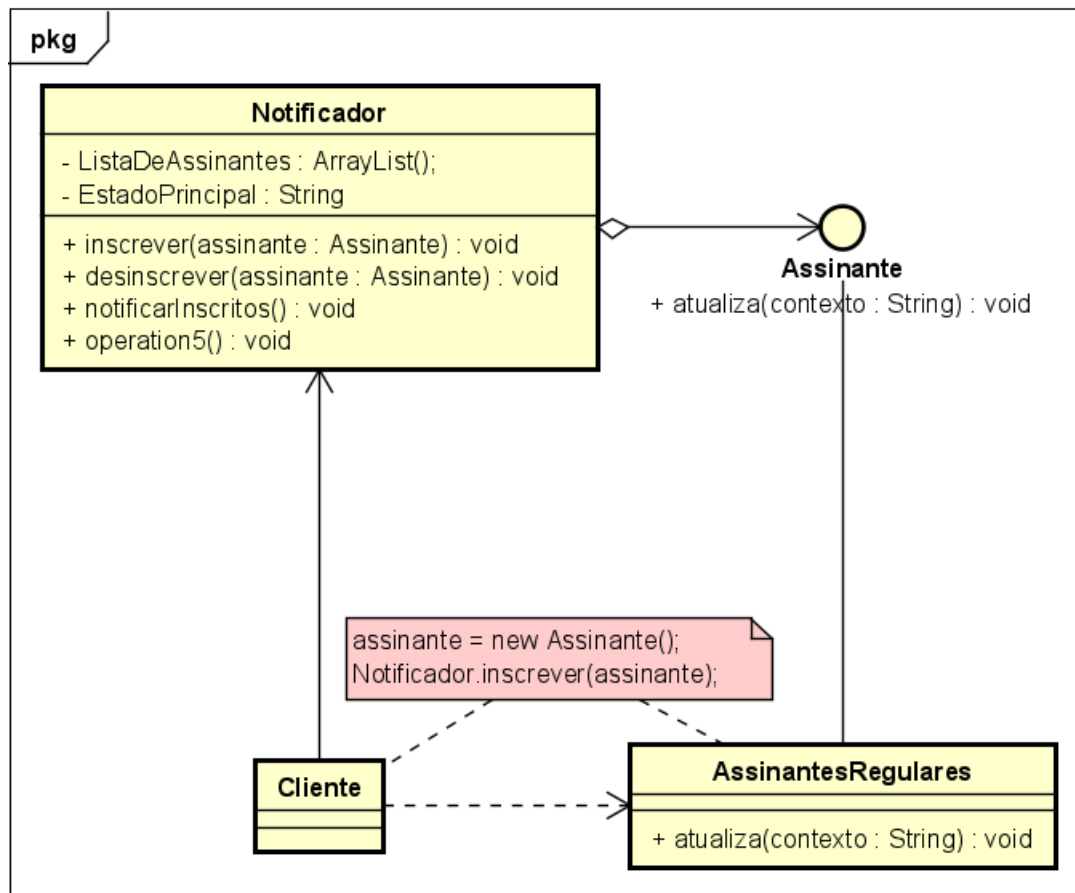
Observer (segundo cenário)

Motivação da escolha do padrão

A escolha do padrão se deve à necessidade identificada em relação aos possíveis usuários, em relação aos materiais disponibilizados na plataforma, para que esses possam ser notificados sobre materiais de seu interesse e quando esses materiais são disponibilizados na plataforma como cursos, ebooks, entre outros.

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



Abstract Factory (segundo cenário)

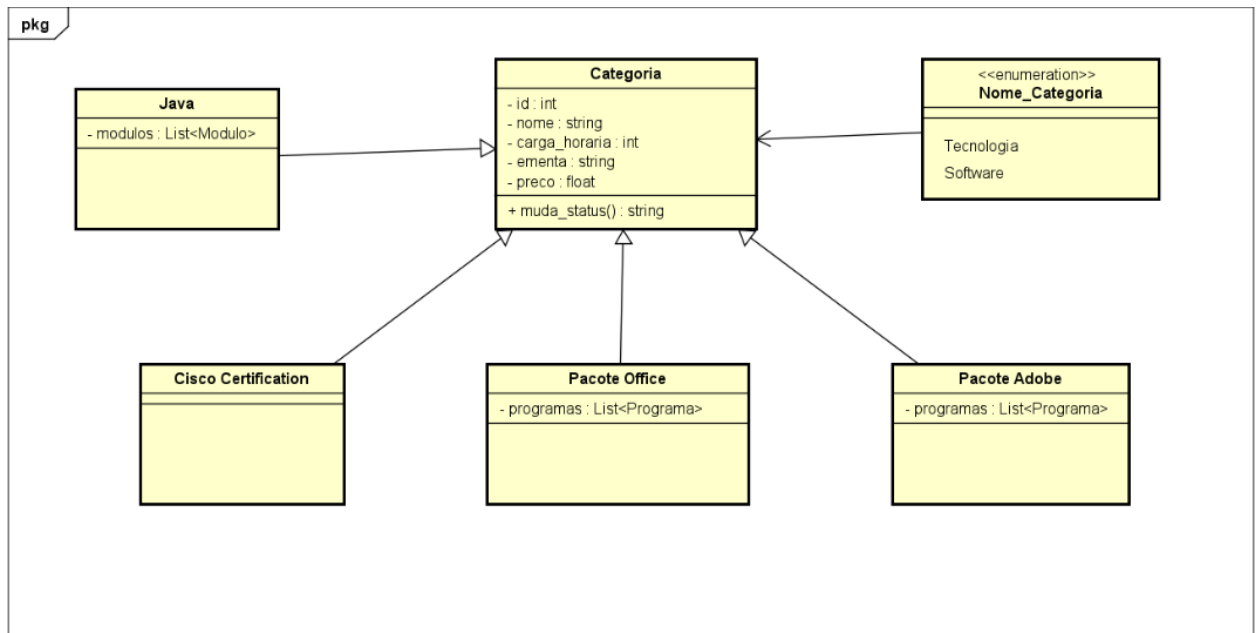
Motivação da escolha do padrão

A escolha do padrão Abstract Factory no modelo em questão foi motivada baseada na economia de classes, que gerará economia no processamento da aplicação. Não faz sentido repetir os mesmos dados em inúmeras classes, apenas em uma, que será herdada pelas demais necessárias.

A necessidade de categorizar e separar os cursos em temas e assuntos nos levou ao uso deste padrão utilizando o enum para auxiliar na separação o que aumenta o desempenho da aplicação gerando uma satisfação maior no usuário final.

Diagrama de instanciação

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



7. Camadas da Arquitetura

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC.

View: Componente que contém as visões do projeto, ou seja, as interfaces, formulários e etc.;

Controller: Componente que recebe as informações e requisições do pacote view e os despacha para devida classe de controle, o controller se comunica com classes BusinessLogic que contém as regras de negócio do sistema;

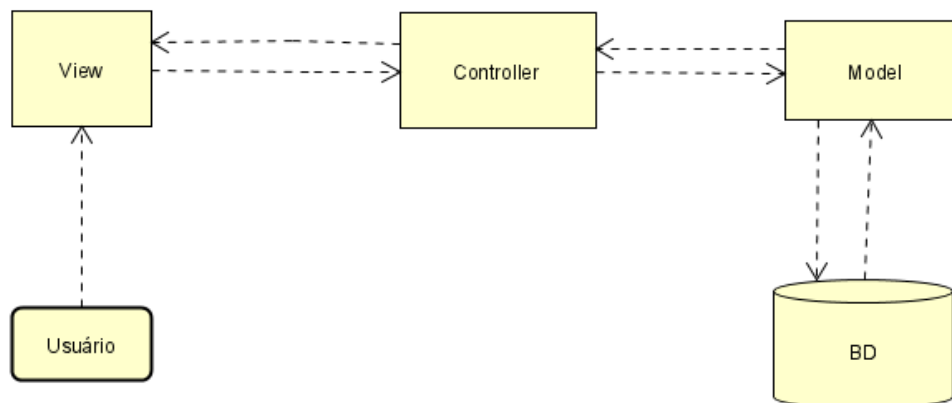
Action - Model: pacote que recebe as informações e requisições do pacote controller e os atribui às respectivas classes do pacote bean e do pacote persistence, utilizando dados de seus modelos;

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

8. Visões da Arquitetura

Visões Recomendadas:

- **Lógica:**



- **Lista de requisitos:**

ID	Descrição
RF-01	Acessar área privada por meio de login
RF-02	Registra produto no sistema
RF-03	Cadastrar outros administradores
RF-04	Alterar atributos do produto
RF-05	Realizar cadastro no sistema
RF-06	Fazer comentários da aula
RF-07	Realizar compra de produto
RF-08	Editar seu perfil
RF-09	Adicionar o produto no carrinho
RF-10	Poder realizar o pagamento via cartão ou boleto
RF-11	Assistir ao curso/mentoria
RF-12	Excluir curso

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

RF-13	Gerar certificado do curso/mentoria
RF-14	Fazer comentários nas aulas
RNF-01	O sistema deve funcionar em desktops, notebook, celulares e tablets
RNF-02	O sistema deve armazenar as senhas no banco de dados de forma segura
RNF-03	O sistema deve ter estar disponível a qualquer hora do dia
RNF-04	O sistema não deve permitir compras sem um login
RNF-05	O sistema será implementado na linguagem Python
RNF-06	O sistema deve verificar automaticamente a conclusão do curso através do monitoramento de tempo assistindo as aulas
RNF-07	O sistema deve utilizar os princípios de SOLID
RNF-08	O sistema deve utilizar autenticação por OAuth 2.0 Bearer Token

• **Casos de Uso:**

Nome do caso de uso	UC1 - Realizar cadastro
Ator(es)	Usuário
Descrição	O usuário preenche um formulário de cadastro que requisita dados básicos para a autenticação e dados úteis para a finalidade do sistema
Referências	-
Gatilho	Na página inicial do sistema, preencher o formulário e, em seguida, confirmar cadastro
Pré-condições	-
Pós-condições	Permitir que o usuário consiga logar no sistema com suas credenciais
Fluxo principal (cenário típico)	1 - O usuário preenche o formulário e confirma o cadastro

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

	2 - O sistema aceita os dados inseridos pelo usuário e o registra no sistema
Fluxo alternativo (cenário alternativo)	2.1 - O sistema não aceita os dados inseridos pelo usuário, que retorna para o passo 1

Nome do caso de uso	UC2 - Realizar login
Ator(es)	Administrador e usuário
Descrição	O usuário preenche um formulário de login que requisita dados básicos para a autenticação.
Referências	-
Gatilho	Na página inicial do sistema o usuário preenche o formulário de login e, em seguida, envia os dados
Pré-condições	Possuir cadastro no sistema
Pós-condições	Os usuários têm acesso ao dashboard, tendo acesso às outras funcionalidades que envolvem seu respectivo tipo de usuário.
Fluxo principal (cenário típico)	1 - O usuário preenche o formulário e confirma o login 2 - O sistema aceita os dados inseridos pelo usuário e o autentica no sistema
Fluxo alternativo (cenário alternativo)	2.1 - O sistema não aceita os dados inseridos pelo usuário, que retorna para o passo 1

Nome do caso de uso	UC3 - Realizar compra de produto
Ator(es)	Usuário

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

Descrição	O usuário pode adicionar o produto ao carrinho e fazer a compra colocando os dados necessário da compra e selecionando o método de pagamento
Referências	RF-07, RF-09, RF-10
Gatilho	Na barra de navegação, acessar "Carrinho de compras", preencher os dados da compra, e finalizar a compra.
Pré-condições	<ul style="list-style-type: none"> • O carrinho de compras deve ter ao menos um produto adicionado • O usuario precisa estar logado
Pós-condições	Permitir o acesso do curso ao usuário, após a confirmação do pagamento
Fluxo principal (cenário típico)	1 - O usuário preenche os dados da compra 2 - O sistema verifica que o pagamento aprovado pelo banco 3 - O acesso ao curso é liberado para o usuário
Fluxo alternativo (cenário alternativo)	2.1 - O sistema verifica o que o pagamento foi recusado pelo banco. Retornar para o passo 1.

Nome do caso de uso	UC4 - Cadastrar produto
Ator(es)	Administrador
Descrição	O administrador pode cadastrar produtos através da página de cadastro que o administrador tem acesso. O administrador deve preencher o formulário com as informações do produto.
Referências	RF-02

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

Gatilho	Na página de “Gerenciar Cursos”, acessar “Novo Curso”, preencher os dados do curso, e finalizar cadastro.
Pré-condições	Estar logado no sistema
Pós-condições	O administrador será redirecionado para a página de gerenciamento de produtos.
Fluxo principal (cenário típico)	1 - O administrador preenche os dados do produto. 2 - O produto é aprovado após avaliação. 3 - O produto fica disponível na loja.
Fluxo alternativo (cenário alternativo)	2.1 - O produto é reprovado após avaliação. 2.2 - O produto pode ser editado para uma nova avaliação.

Nome do caso de uso	UC5 - Assistir aula
Ator(es)	Usuário
Descrição	O usuário pode assistir as aula do curso que ele possui dentro da plataforma
Referências	RF-011
Gatilho	Na barra de navegação, acessando “Meus Cursos” e selecionando o curso desejado, o usuário será redirecionado para sua página do Curso onde poderá assistir às aulas.
Pré-condições	O usuário deverá ter comprado o curso
Pós-condições	O vídeo da aula será reproduzido
Fluxo principal (cenário típico)	1 - O usuário seleciona o curso 2 - O usuário assiste às aulas do curso
Fluxo alternativo (cenário alternativo)	1.1 - O usuário deve comprar o curso antes e realizar o UC4.

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

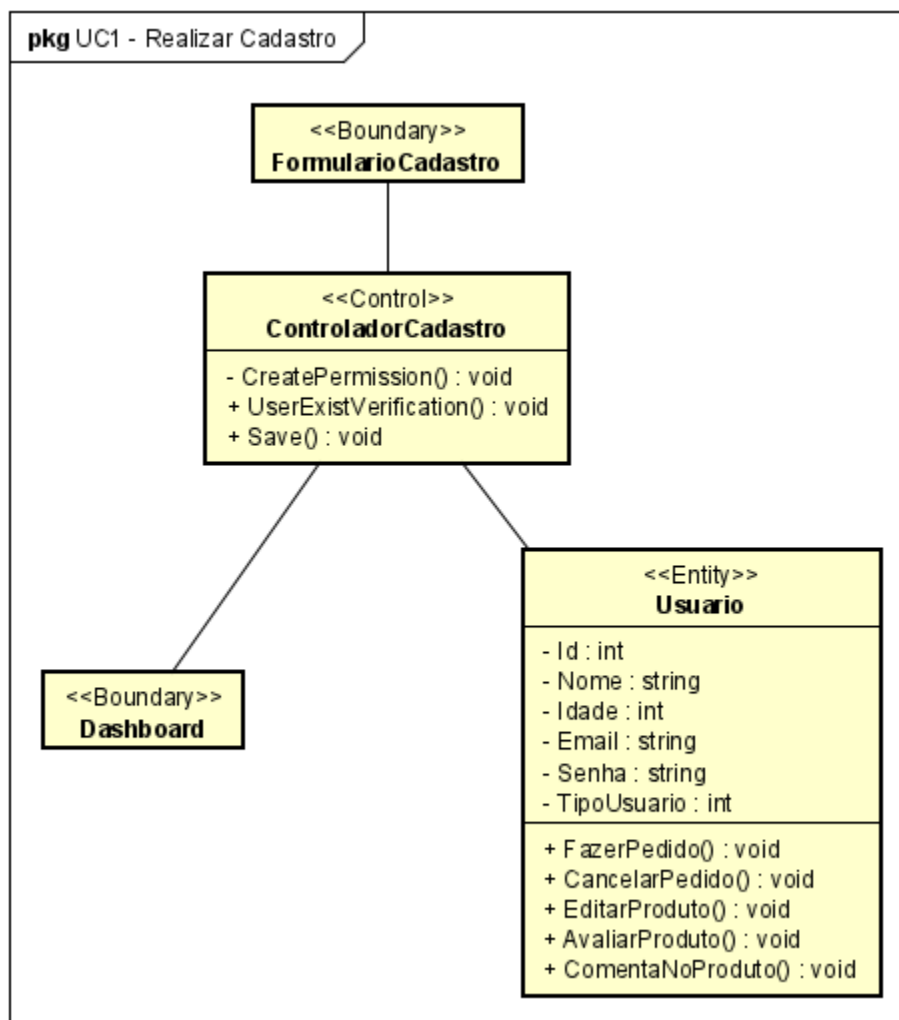
Nome do caso de uso	UC6 - Editar o Perfil
Ator(es)	Usuário
Descrição	O usuário pode alterar seus dados cadastrados
Referências	RF-08
Gatilho	Na barra de navegação o usuário pode clicar em sua foto e ir para seu perfil e acessar a edição de perfil
Pré-condições	Estar logado em uma conta
Pós-condições	-
Fluxo principal (cenário típico)	<ol style="list-style-type: none"> 1. Alterar os dados desejado do perfil 2. Clicar no botão “Salvar”
Fluxo alternativo (cenário alternativo)	2.2. Sair da página sem salvar, não irá salvar as alterações

Nome do caso de uso	UC7 - Listar produtos do usuário
Ator(es)	Usuário
Descrição	O usuário acessa a uma lista de seus produtos
Referências	-
Gatilho	Na barra de navegação o usuário pode clicar em “Meus Cursos” e ir para página de Listagem de Cursos
Pré-condições	Estar logado no Sistema
Pós-condições	-
Fluxo principal (cenário típico)	<ol style="list-style-type: none"> 1. Clicar em “Meus Cursos” 2. Visualizar todos os Cursos comprados

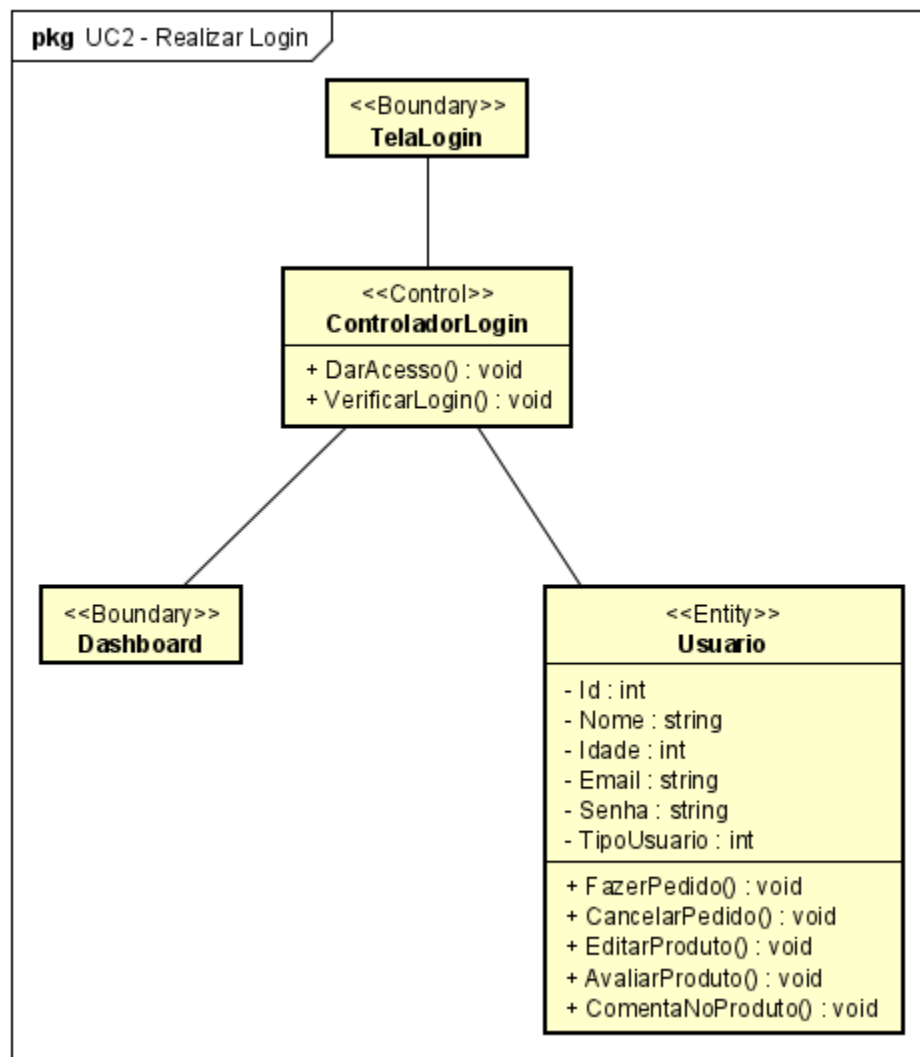
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

- Inicialmente, representamos nos diagramas algumas funções que tinham várias responsabilidades. Por isso, separamos as funções para que tenham apenas uma responsabilidade.
- Inicialmente, fizemos apenas uma classe "Usuario" para todos os tipos de usuário do sistema, impossibilitando a disponibilização de interfaces diferentes para o Admin, o Professor e o Usuário que apenas assiste os cursos. Por isso, separamos essas classes, fazendo uma especialização para o Admin e o Usuário e colocamos o atributo "TipoUsuario" em "Usuario".

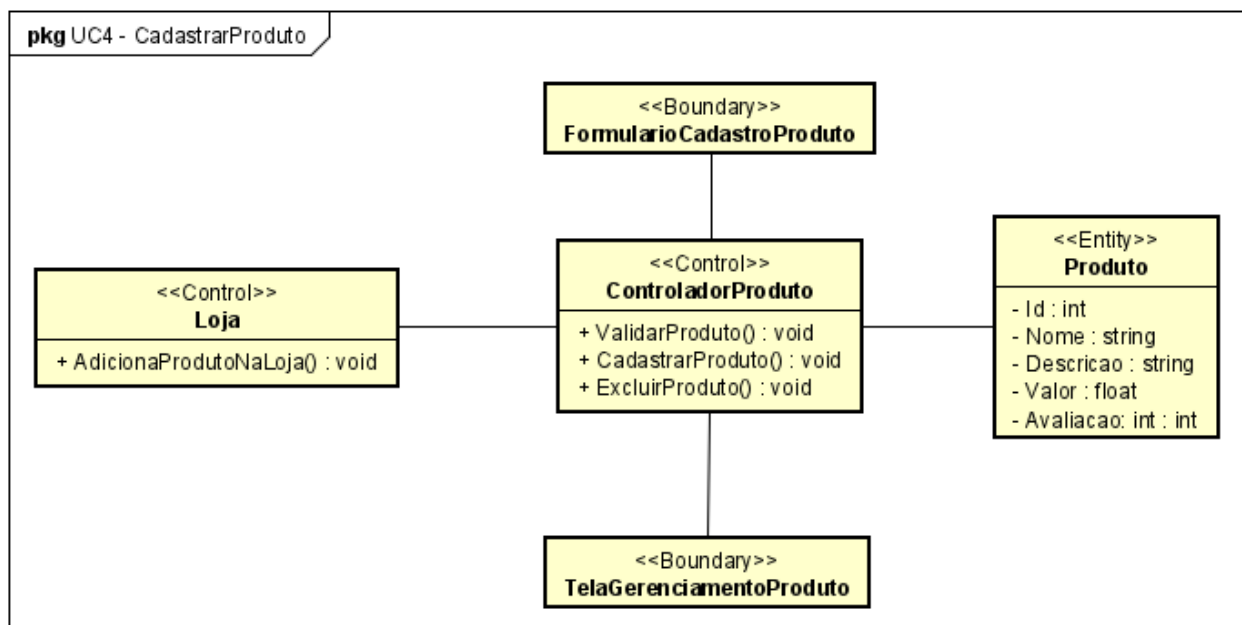
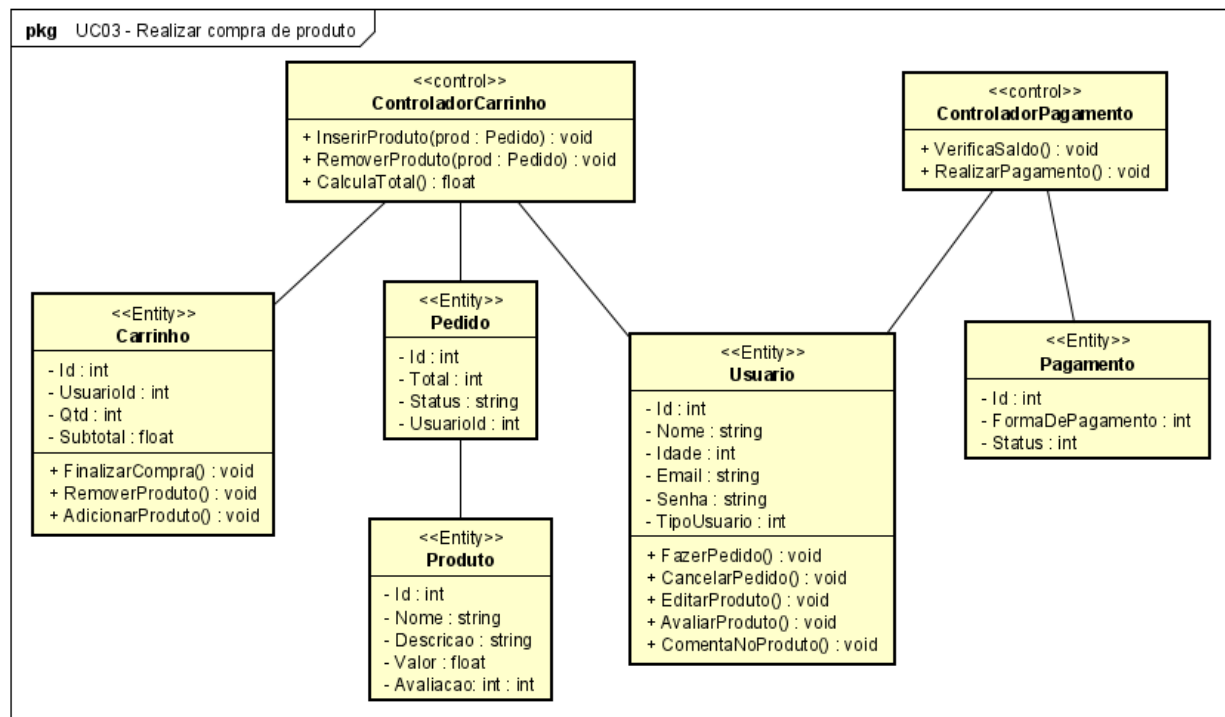
Os seguintes diagramas correspondem a cada caso de uso previamente definido.



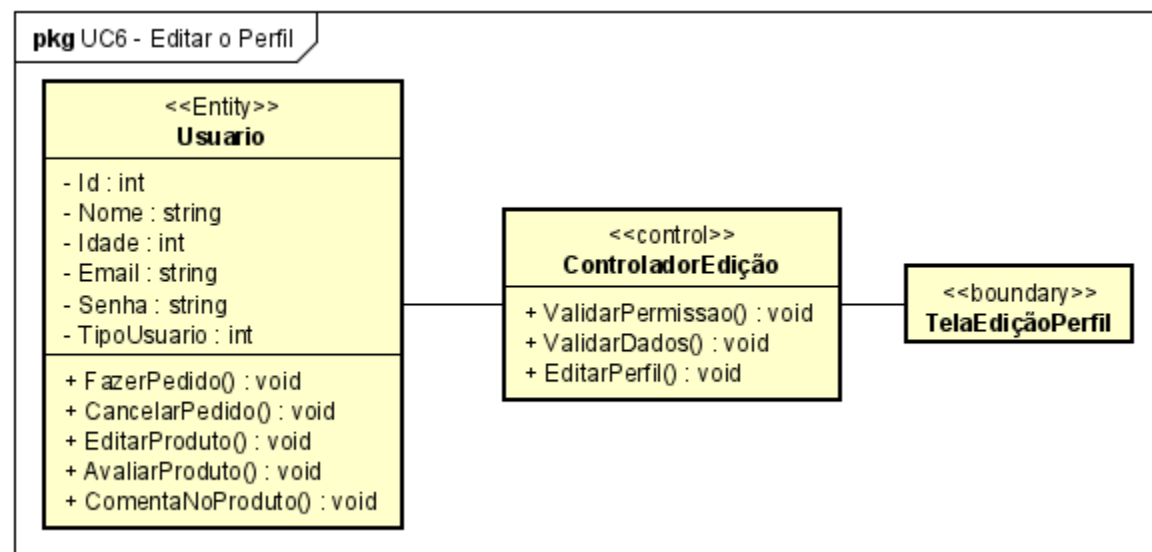
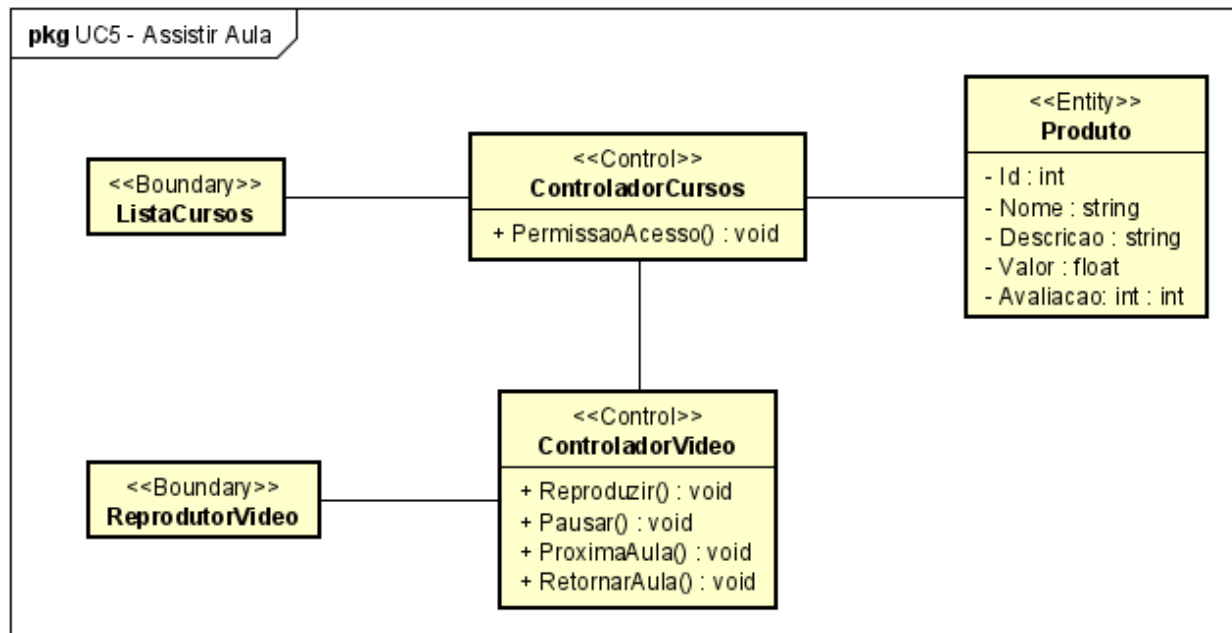
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



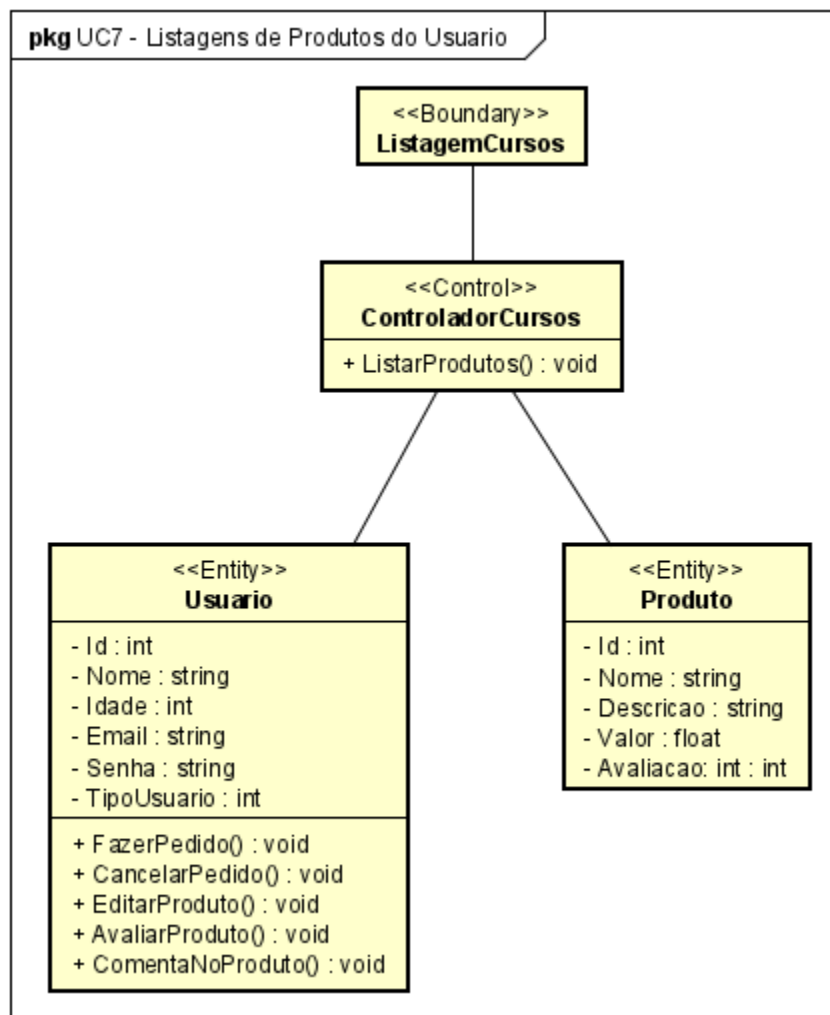
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

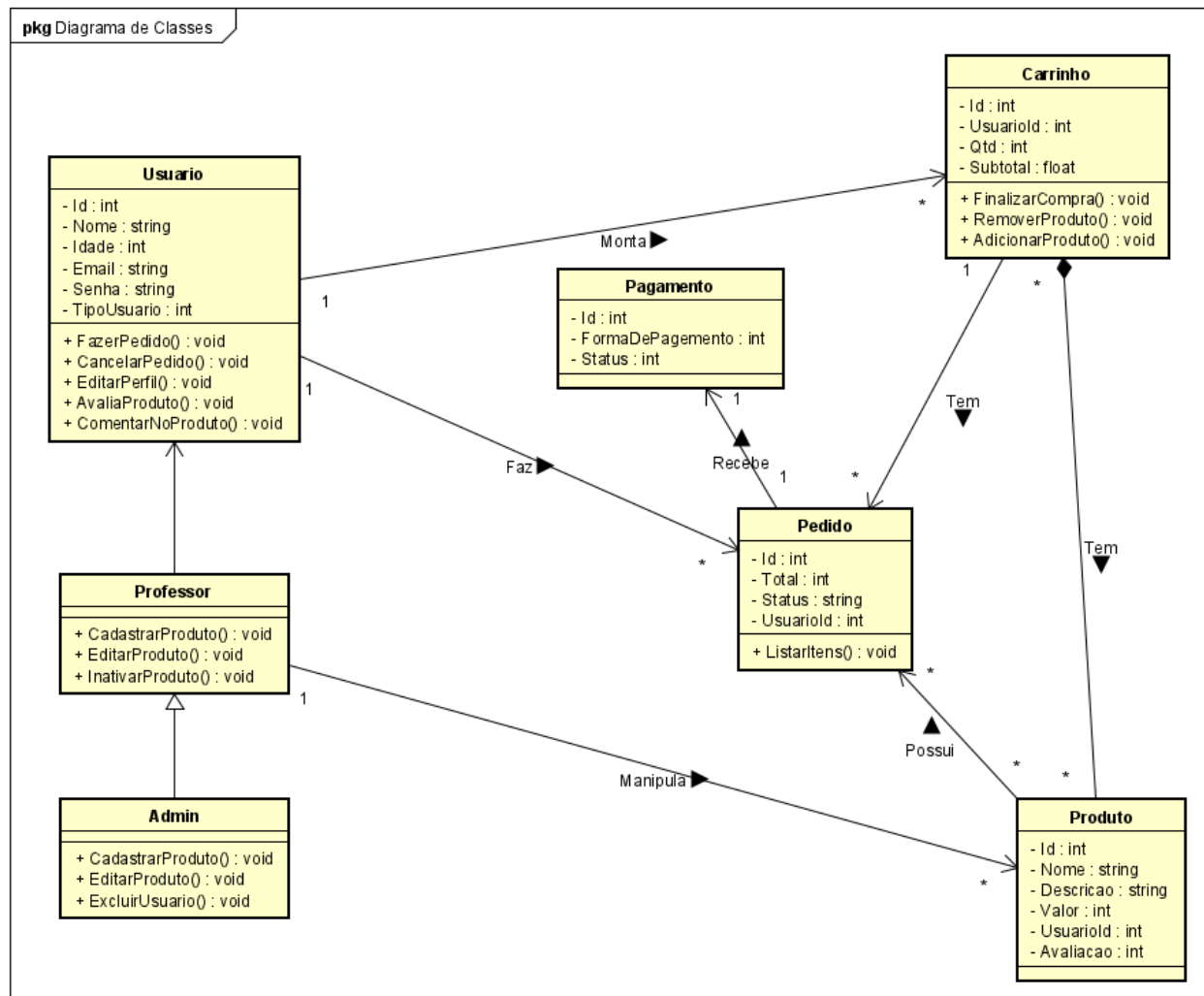


E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



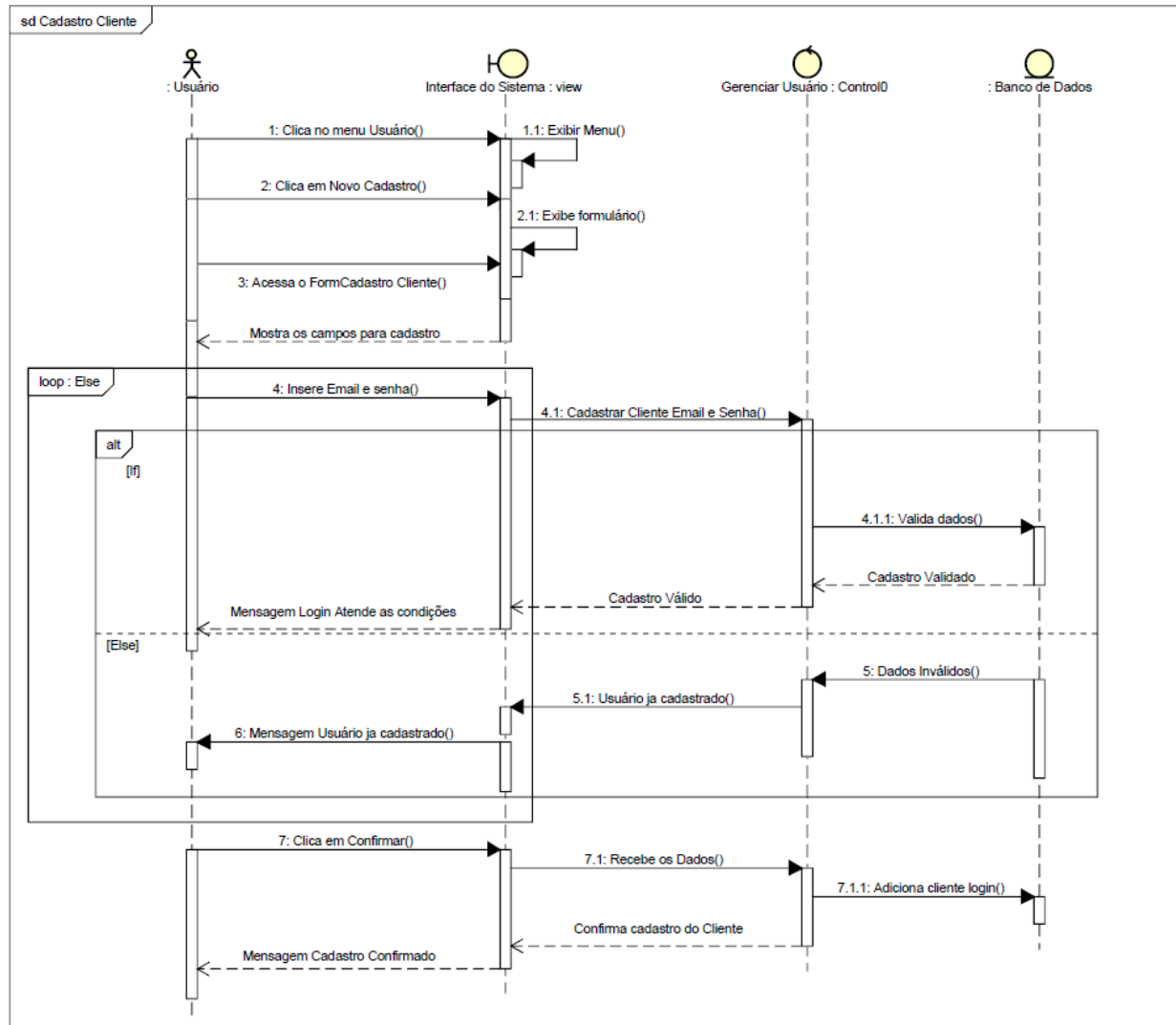
8.3 Diagrama de Classes

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

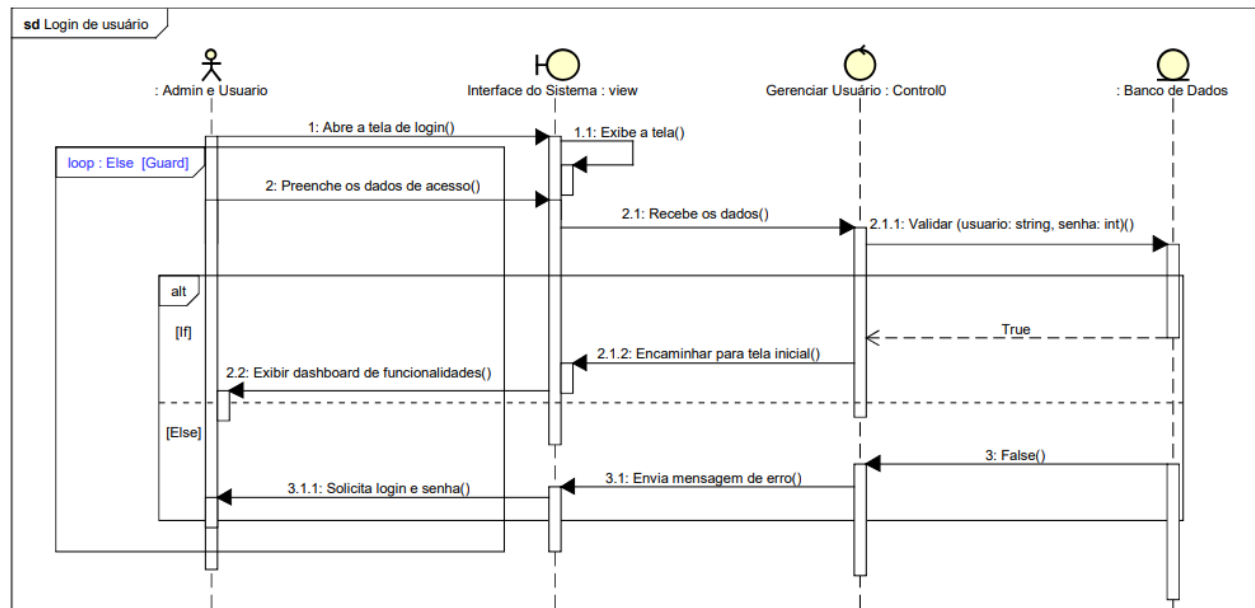


8.4 Diagramas de Sequências

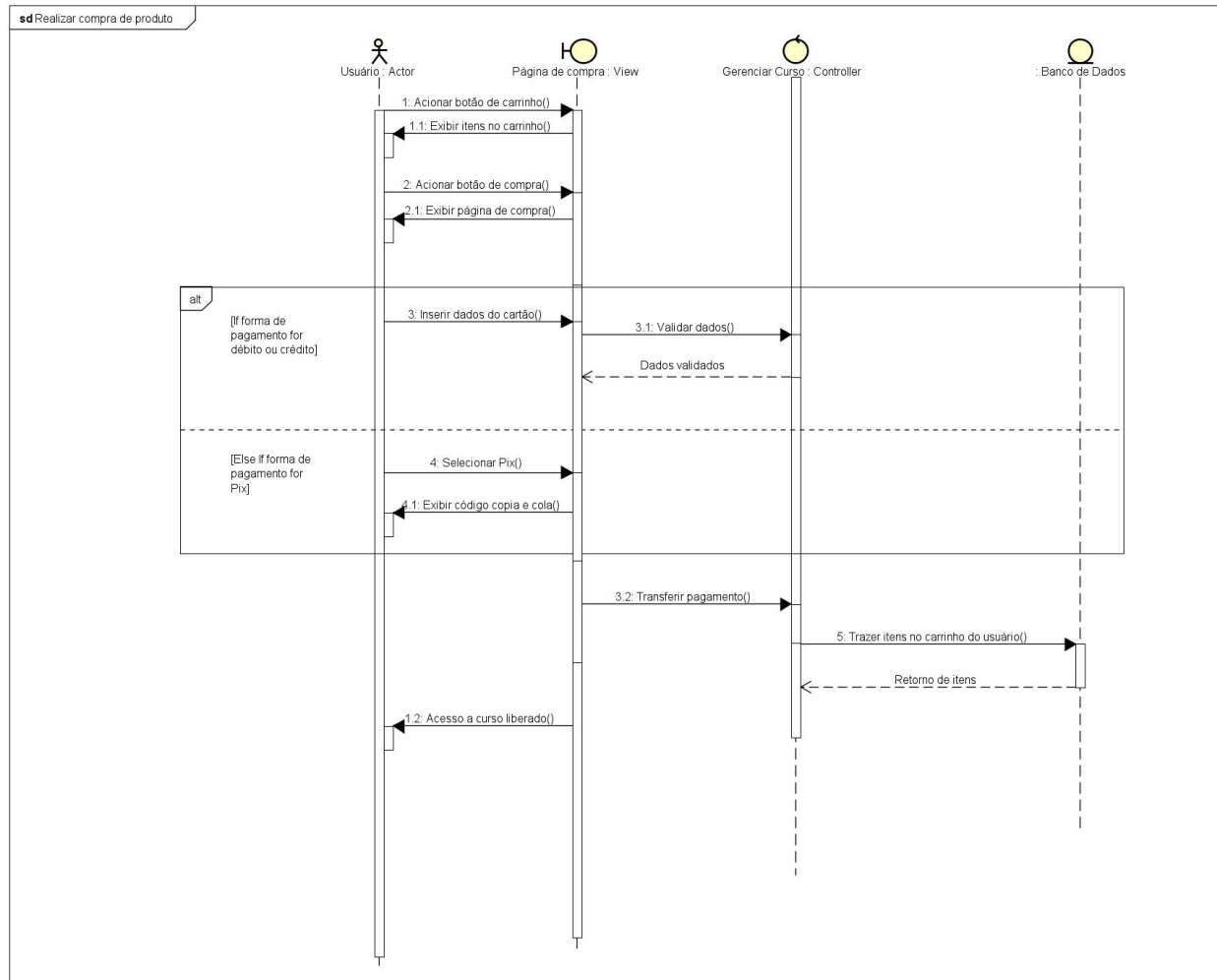
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



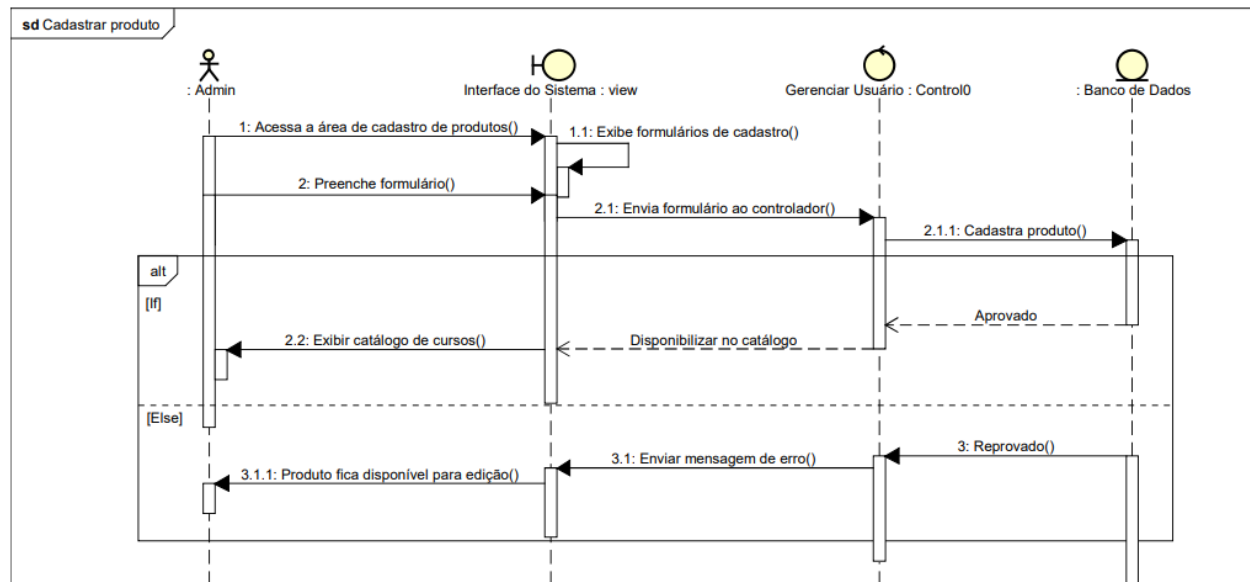
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



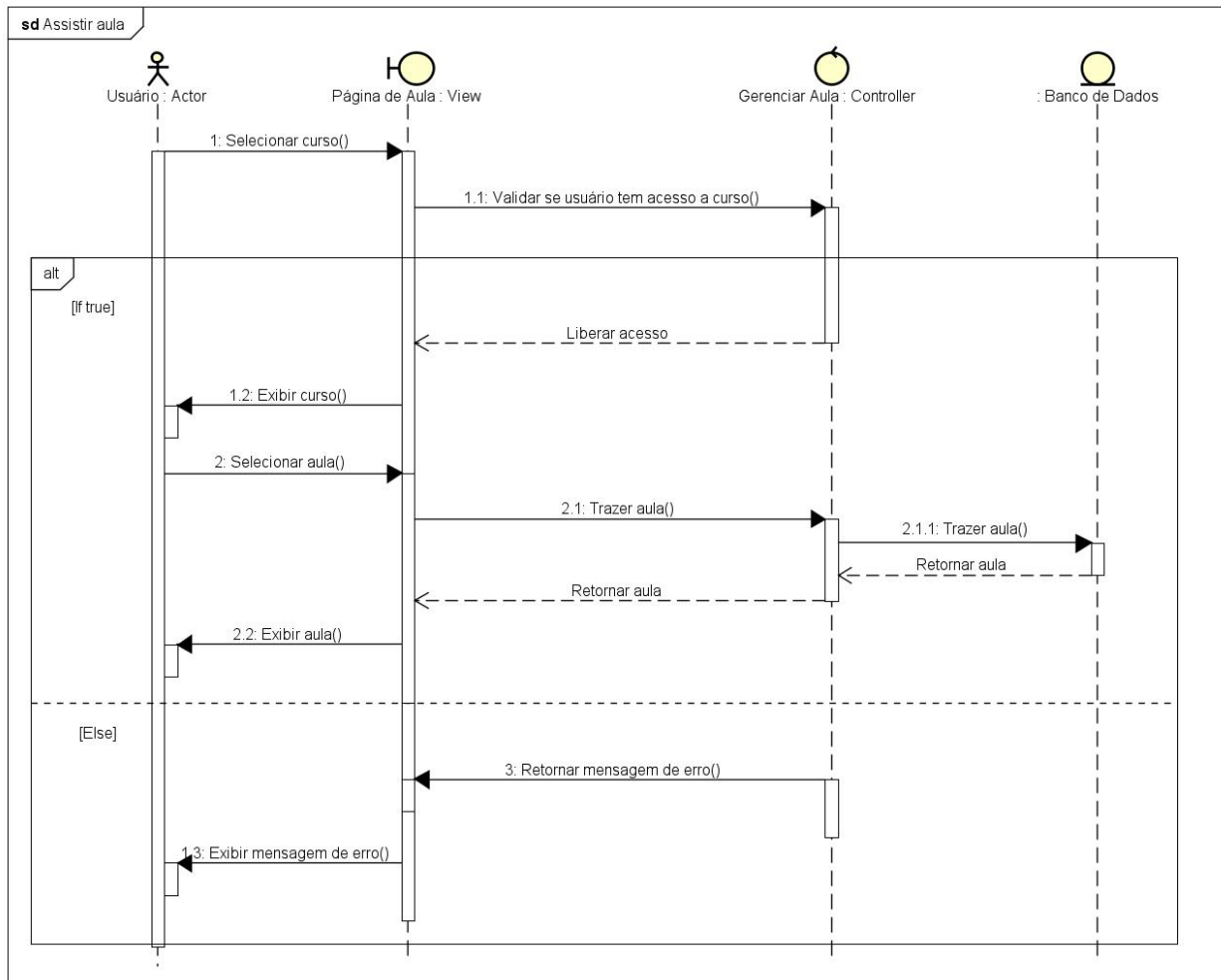
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



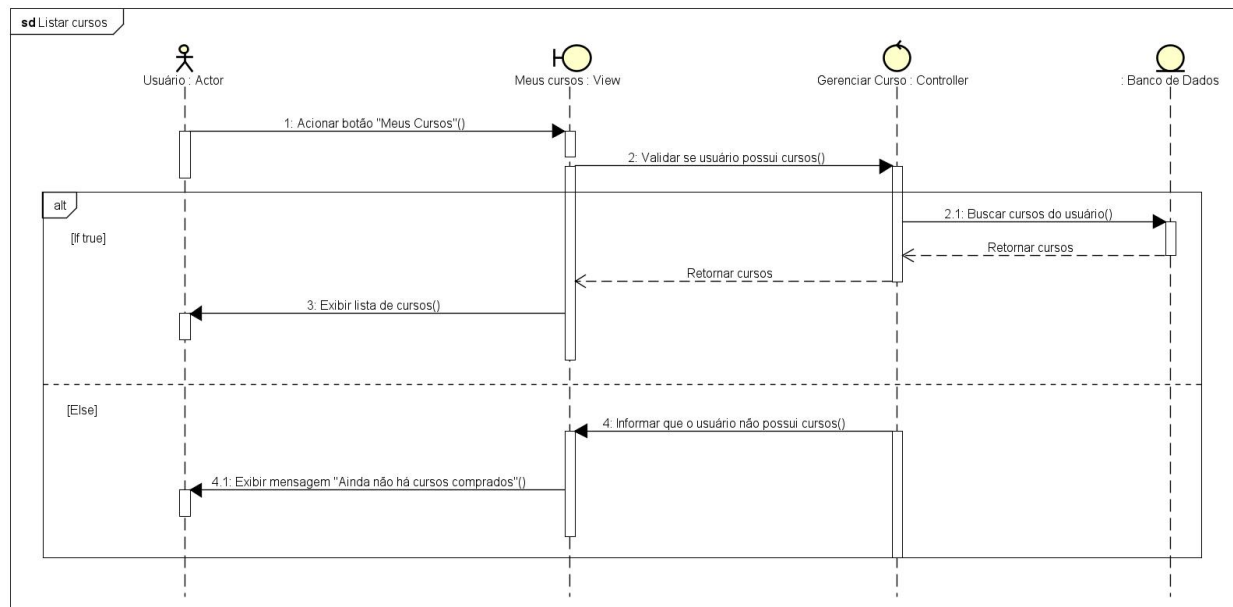
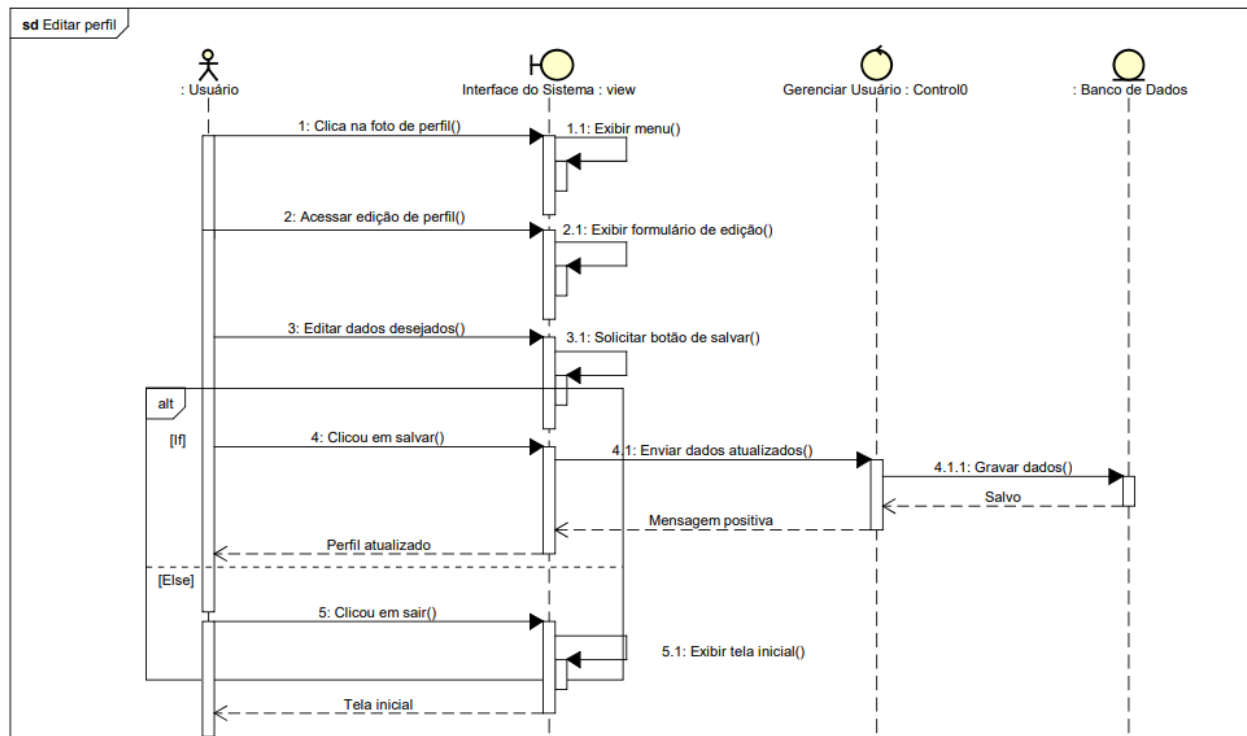
E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	



9. Qualidade

Considerando que utilizaremos o padrão de arquitetura MVC, a arquitetura desse sistema permite dividir as funcionalidades em camadas, facilitando a resolução de problemas maiores. A

E-Classroom	Version: 3.9
Documento de Arquitetura de Software	Date: 29/06/2022
<document identifier>	

utilização de Flask, por ser um micro framework, disponibiliza maior controle e flexibilidade para os desenvolvedores, o que garante um design extensivo à arquitetura do sistema. A constante atualização desse micro framework, a disponibilidade de diversas ferramentas de segurança, entre outros elementos, são alguns fatores que proporcionam confiabilidade e qualidade à arquitetura do sistema.