**E-Classroom**

**Documento de Arquitetura de Software**

**Versão 1.0**

**Histórico da Revisão**

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 15/05/2022 | 1.0 | Elaboração inicial do documento (Introdução; diagramas de VCP; qualidade; suposições e dependências; metas e restrições da arquitetura) | Luisa Dirce, Daniel Fernandes, Lucas Pimenta, Philipe Simões |
| 18/05/2022 | 1.1 | Atualização da definição da arquitetura e diagramas UML; decisões, restrições e justificativa | Marcos Roberto, Fábio Gil, Amanda Marques, Alexandre Calmon, Lucas Couto, Yuan Stewart |
|  |  |  |  |
|  |  |  |  |

**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) 11

9. Qualidade 16

**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. **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. **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. **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

1. **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.

1. **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.

1. **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

1. **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.

1. **Mecanismos Arquiteturais**

*[Liste os mecanismos arquiteturais, como mecanismos de persistência, comunicação e tratamento de erros, por exemplo, e descreva ocorrente estado de cada um. Inicialmente, cada mecanismo pode ser somente um nome e uma breve descrição. Eles evoluirão até que o mecanismo se torne um padrão ou uma colaboração de elementos de projeto que possam ser aplicados diretamente em algum aspecto do projeto.]*

**Mecanismo Arquitetural 1**

*[Descreva a finalidade, os atributos e funções do mecanismo arquitetural.]*

**Mecanismo Arquitetural 2**

*[Descreva a finalidade, os atributos e funções do mecanismo arquitetural.]*

1. **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.

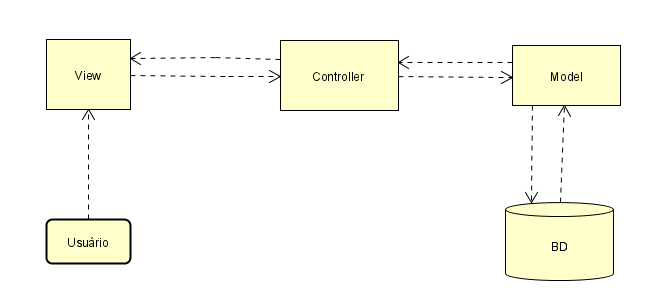
View: Componente que contém as visões do projeto, ou seja, as interfaces, formulários e etc.;

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;

1. **Visões da Arquitetura**

Visões Recomendadas:

* **Lógica:** 
* **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  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 |
| 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 | O carrinho de compras deve ter ao menos um produto adicionado |
| 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. |

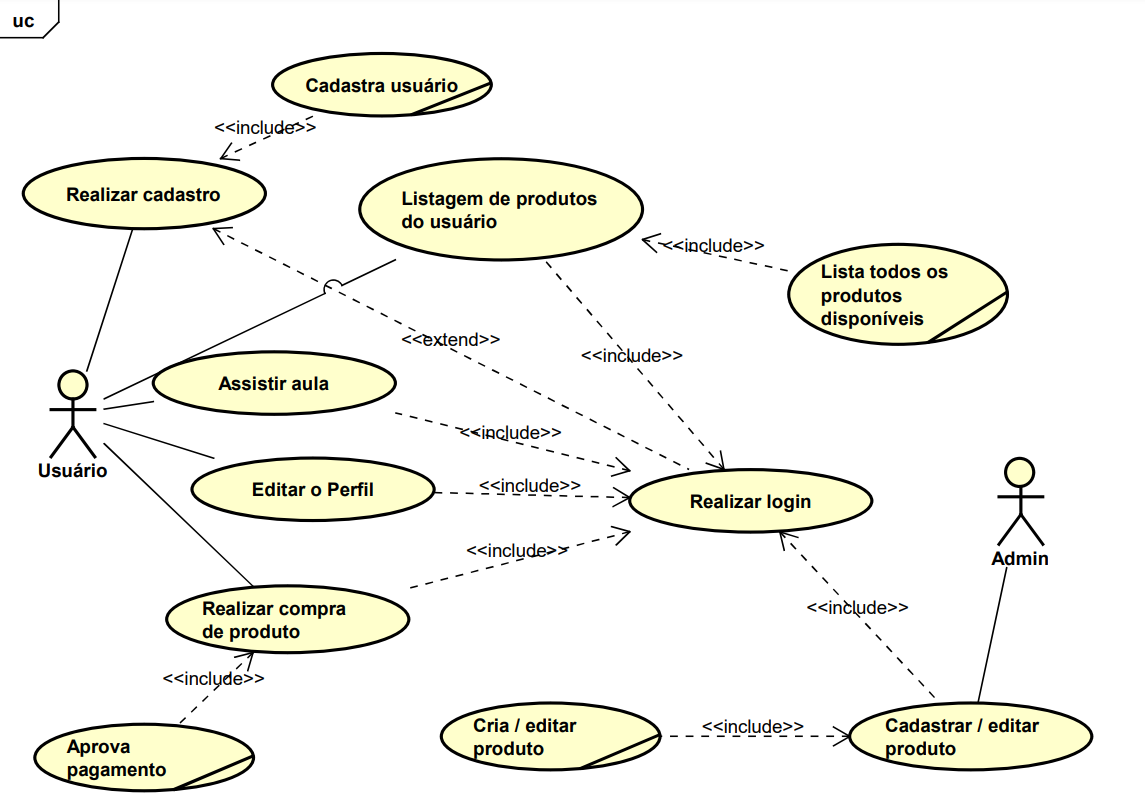
| Nome do caso de uso | UC4 - Cadastrar produto |
| --- | --- |
| Ator(es) | Administrador |
| Descrição | O administrador pode cadastrar produtos através da página de cadastro que o administrador tem acesso. O administrador deve preencher o formulário com as informações do produto. |
| Referências | RF-02 |
| Gatilho | Na página de “Gerenciar Cursos", acessar "Novo Curso", preencher os dados do curso, e finalizar cadastro. |
| Pré-condições |  |
| 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. |

| 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) | 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 |

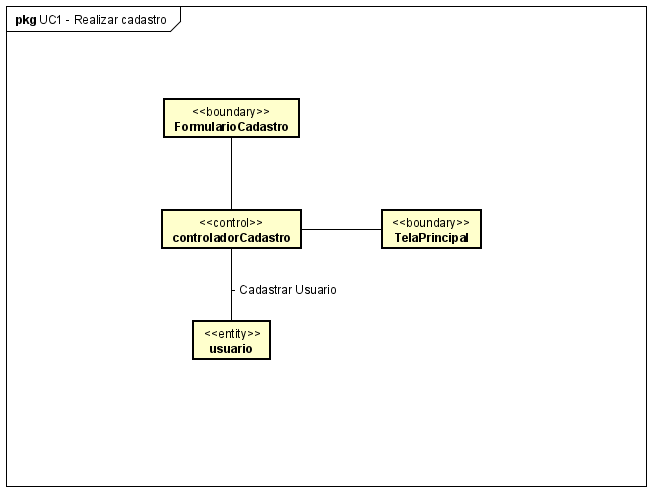
| Nome do caso de uso | UC7 - Listar produtos do usuário |
| --- | --- |
| Ator(es) | Usuário |
| Descrição | O usuário acessa a uma lista de seus produtos |
| Referências | - |
| Gatilho | Na barra de navegação o usuário pode clicar em “Meus Cursos” e ir para página de Listagem de Cursos |
| Pré-condições | Estar logado no Sistema |
| Pós-condições | - |
| Fluxo principal (cenário típico) | 1. Clicar em “Meus Cursos” 2. Visualizar todos os Cursos comprados |
| Fluxo alternativo (cenário alternativo) | 2.2. O usuário não possui nenhum curso comprado |

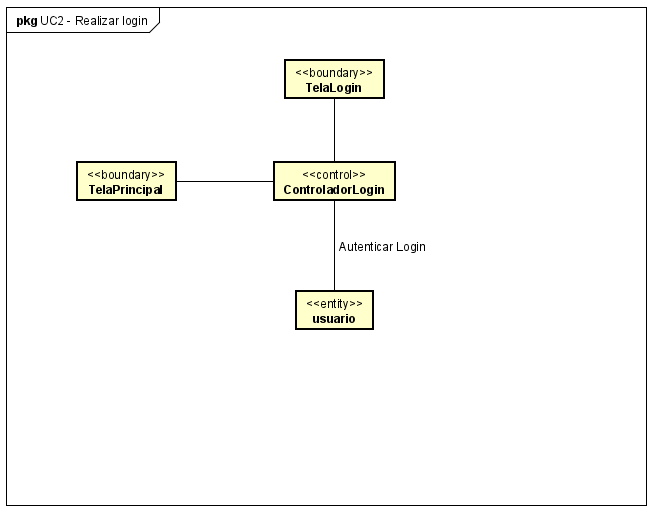
* 1. **Diagrama de Casos de Uso**

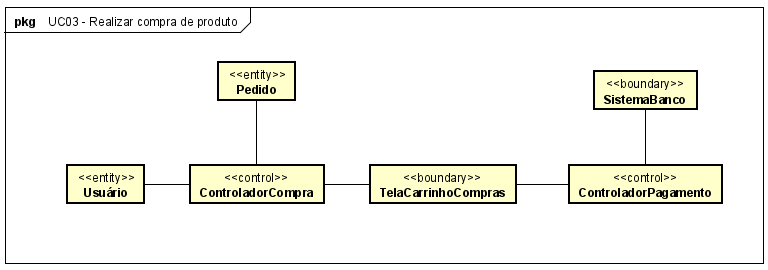
****

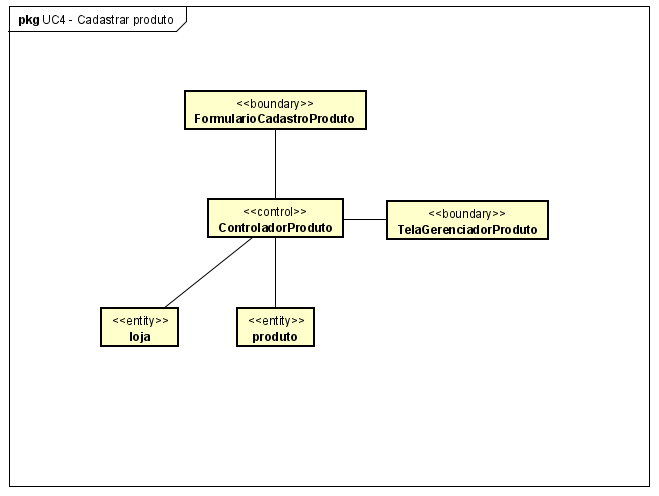
* 1. **Visões de Classes Participantes (VCP)**

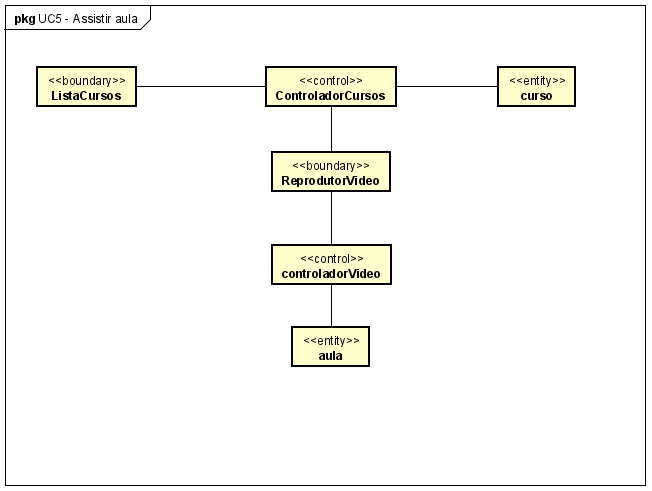
Os seguintes diagramas correspondem a cada caso de uso previamente definido.

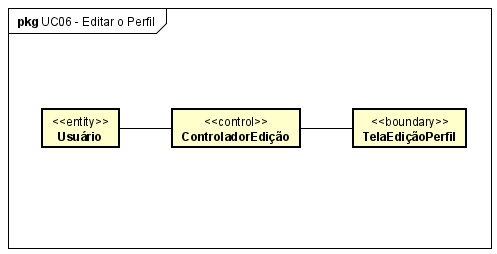


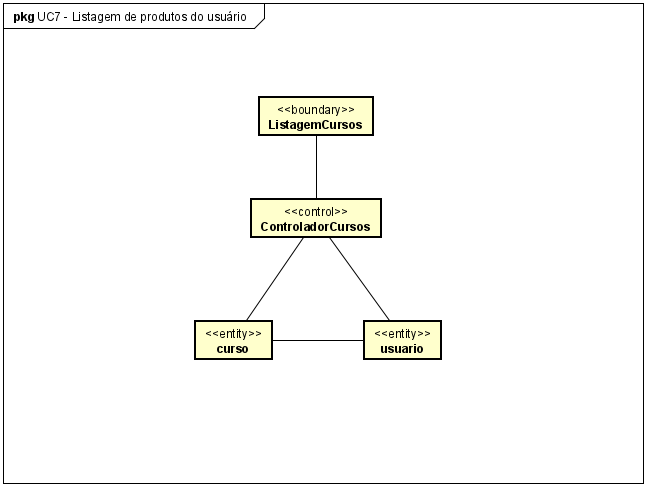


****





****



1. **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 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.