

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA –
UDESC CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
GRADUAÇÃO – CIÊNCIA DA COMPUTAÇÃO**

**Maurício Nascimento Cunha
Thiago Pimenta Barros Silva**

**PROJETO DE ENGENHARIA DE SOFTWARE:
GetHelp!**

**JOINVILLE-SC
2022**

Maurício Nascimento Cunha

Thiago Pimenta Barros

**PROJETO DE ENGENHARIA DE SOFTWARE:
GetHelp!**

Parte 1 do Trabalho apresentado como requisito para obtenção de nota parcial da disciplina de Engenharia de Software do curso de Ciência da Computação pertencente ao Centro de Ciências Tecnológicas – CCT, da Universidade do Estado de Santa Catarina – Udesc.

Orientadora: Profa. Dra. Rebeca Schroeder Freitas.

JOINVILLE

2022

Sumário

0. Introdução	4
1. Descrição do Problema	
1.1 Contexto	4
1.2 Objetivos e Justificativa	4-5
1.3 Escopo	5-6
1.4 Stakeholders	
1.4.1 Internos	6
1.4.2 Externos	6
1.5 Visão Geral	6-7
2. Documento de Requisitos:	
2.1 Requisitos Funcionais e Não Funcionais Associados	7-18
2.2 Requisitos Suplementares	19
2.3 Organização de requisitos em consultas	20
2.4 Processos de Negócio	20
2.5 Manutenção de Conceitos	21
2.6 Casos de Uso Expandidos	21-22
2.6.1 Exibir aula ou aula de um curso	21
2.6.2 Avaliar aula ou curso	21
2.6.3 Comprar aula ou curso	21-22
2.6.4 Venda aula ou curso	22
2.6.5 Denunciar Aluno ou Professor	22
2.7 Contratos	23-29
2.7.1 CO1: [RS] O sistema exibe a aula selecionada	23
2.7.2 CO2: [EV1] O sistema realiza a compra	23-24
2.7.3 CO3:[EV2] O cliente deve realizar seu cadastro	24
2.7.4 CO4:[EV3] O sistema registra a avaliação	25
2.7.5 CO5: [EV4] Será solicitado a mudança desse trecho do comentário	25-26
2.7.6 CO6:[EV5] O usuário escolhe a opção de quitar os débitos e continuar a compra	26

2.6.7 CO7:[EV6] O usuário cadastra um novo cartão	27
2.7.8 CO8: [EV7] O sistema vai solicitar nova forma de pagamento	27-28
2.7.9 CO9:[EV8] O sistema realiza o pagamento	28
2.7.10 CO10:[EV9] O usuário visualiza o conteúdo impróprio e clica em denunciar	29
2.7.11 CO11:[EV10] Caso seja identificado uma infração	29-30
3. Estrutura Analítica do Projeto (EAP)	
3.1 Projeto no diagrama da imagem 1.	31
3.2 Sequência de Atividades será detalhada	32
4. Estimativa de duração do projeto	
4.1 Análise de Pontos de Função	33-34
5. Diagrama de Classes do Projeto	35-36
6. Considerações Finais	37

Introdução

Planeja-se construir uma aplicação para gerenciar uma plataforma de ensino a distância. Essa plataforma tem como objetivo principal a tentativa de resolver as lacunas que existem nas plataformas como adquirir apenas algumas aulas ou apenas um curso específico. Além disso, os usuários poderão ser de dois tipos distintos: alunos e professores, no caso dos professores, poderão tanto ver aulas como colocar seus conteúdos na plataforma de maneira paga ou gratuita.

A ideia é que os cursos possam ter diversas disciplinas, lecionadas por um ou mais professores. Os alunos terão acesso a qualquer conteúdo do curso em que estarão matriculados. A partir desses serviços disponibilizados aos usuários, procura-se propor melhorias ao ensino a distância, como também torná-lo mais acessível.

1.1 CONTEXTO

O ensino a distância devido a sua recente história como método de aprendizagem ainda apresenta algumas dificuldades em sua aplicação. Dentre elas, encontrar professores qualificados com conteúdo de qualidade, pois embora haja ampla oferta disponível online, faltam plataformas que unifiquem essa busca e oferta de ensino. Outro problema encontrado nas diversas plataformas atuais é a possibilidade de compra de conteúdos específicos e de curta duração. Muitas vezes os usuários precisam comprar cursos inteiros para ter acesso a uma parte específica do conteúdo contido nesse curso.

1.2 OBJETIVOS E JUSTIFICATIVA

A aplicação proposta neste projeto tem como meta principal a tentativa de resolver ou amenizar os problemas citados no tópico anterior. A partir da criação de uma plataforma online de ensino que contenha conteúdos de todas as áreas de ensino em todos os níveis de educação. Sendo assim essa aplicação busca ter conteúdos de ensino básicos como alfabetização como também aulas sobre conteúdos estudados por doutores em instituições de ensino superior. Além de possibilitar o aluno adquirir todo pacote de aulas de um curso,

também será disponibilizada a possibilidade de obter apenas algumas aulas sobre conteúdos de cursos por valores mais acessíveis.

Buscando atingir esses dois objetivos principais o projeto irá possibilitar que professores certificados de todos os níveis de ensino ofereçam seus serviços de forma gratuita ou remunerada. Como também, alunos assistirem a conteúdos gratuitos ou adquirirem cursos pagos, além de permitir que cursos sejam adquiridos por completo ou apenas algumas aulas de seus conteúdos. Outras duas funcionalidades disponibilizadas seriam a avaliação dos alunos sobre os conteúdos ministrados e uma avaliação do professor a partir da média da avaliação de suas aulas e cursos.

A partir desses serviços disponibilizados aos usuários procura-se melhorar e facilitar o ensino a distância, como também torná-lo mais acessível. Além de incentivar que mais profissionais e alunos se interessem pela educação remota fomentando a ingresso das pessoas nessa nova modalidade de ensino.

1.3 ESCOPO

Entre as funcionalidades que a aplicação irá ter se encontram:

- Cadastro de Usuários (Alunos e Professores)
- Cadastro de Aulas e Cursos (Gratuitos e Pagos)
- Host de Vídeo (Aulas Gravadas)
- Compra (Aulas e Cursos)
- Venda (Aulas e Cursos)
- Feedback de Alunos sobre Aulas de Professores

Entre as funcionalidades que não serão incluídas:

- Disponibilizar Aulas e Professores Presenciais: não há possibilidade de atender questões infraestruturais reais em questão de espaço físico ou segurança de usuários.
- Processar Compras ou Pagamentos: todo e qualquer processo que envolva transação monetária será terceirizado e apenas disponibilizado no site, tanto por questão de complexidade quanto por segurança.

1.4 STAKEHOLDERS

1.4.1 Internos

São os que estão dentro do ambiente da empresa. Alguns exemplos de stakeholders internos são:

- Funcionários;
- Gestores;
- Gerentes;
- Usuários
- Investidores

1.4.2 Externos

São os que estão fora do ambiente da empresa, mas que interagem com o projeto de alguma forma indireta. Alguns exemplos de stakeholders externos são:

- Parceiros;
- Concorrentes;
- Fornecedores;
- Estado

1.5 Visão Geral do Sistema

Para resolver o problema do ensino citado e atender aos objetivos propostos uma aplicação chamada “Get|Help” será criada, a aplicação teria todas as funcionalidades descritas no escopo e também seria acessível via web por qualquer navegador. O objetivo do sistema é facilitar o processo de ensino a distância e garantir maior acesso da população a conteúdos educativos. Sendo possível adquirir apenas algumas aulas de cursos completos e assistir aulas disponíveis gratuitamente, ao mesmo tempo que possibilita um melhor controle do aluno sobre sua formação. Isso é possível pois ele terá acesso desde conteúdos básicos a avançados em sua área de ensino.

O sistema deverá calcular automaticamente o valor das aulas e cursos que foram adquiridos, além de quanto cada professor irá receber por suas aulas pagas e sistemas terceirizados realizarão o pagamento aos professores. Possibilitando assim professores a terem mais uma fonte de renda e de expor o seu trabalho para novos alunos de diferentes lugares do Brasil. A cada conjunto de compra de aulas ou cursos corresponderá a um

pagamento do aluno, sendo usado sistemas terceirizados de crédito, boleto e pix, para realizar a compra. A impossibilidade de efetuar um pagamento deve deixar o cliente aluno suspenso, ou seja, impossibilitado de adquirir novas aulas ou cursos até saldar a dívida. Caso o professor seja denunciado por conteúdo indevido ou não tenha certificação adequada, as suas novas aulas não serão postadas e as antigas bloqueadas pelo sistema.

2. Documento de Requisitos:

2.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS ASSOCIADOS

F1: Cadastrar aluno		Oculto ()		
Descrição: Tem por objetivo cadastrar um aluno no sistema. O sistema deve registrar Nome Completo, CPF Válido, Data de Nascimento, Endereço, Telefone, Usuário, Senha e Email.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Requisitos legais	O sistema deve seguir todas as normas legais previstas na constituição.	Confiabilidade Segurança		X
NF 1.2 Requisitos éticos	O sistema deve garantir a integridade dos dados pessoais de seus usuários.	Confiabilidade	X	X
NF 1.3 Confirmação	O sistema deverá enviar uma mensagem de confirmação após um cadastro, via e- mail.	Interoperabilidade		X
NF 1.4 Verificação	O sistema deve verificar se CPF informado é válido	Confiabilidade Segurança Verificação		X

F2: Cadastrar Professor		Oculto ()		
Descrição: Tem por objetivo cadastrar um professor no sistema. O sistema deve registrar Nome Completo, CPF, Data de Nascimento, Endereço, Telefone, Usuário, Senha, Email, conta bancária, área de atuação e Comprovante de qualificação(diploma ou certificado na área,).				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 2.1 Verificação do comprovante de qualificação.	O sistema deve verificar a qualificação do professor. Impedindo a operação caso for comprovado que não há qualificação necessária ou dados insuficientes. Tando sobre o nível quanto sobre a área de atuação.	Confiabilidade Segurança Verificação		X

NF 2.2 Requisitos éticos	O sistema deve garantir a integridade dos dados pessoais de seus usuários.	Confiabilidade	X	X
NF 2.3 Confirmação	O sistema deverá enviar uma mensagem de confirmação após um cadastro, via e-mail.	Interoperabilidade		X
NF 2.4 Verificação	O sistema deve verificar se CPF informado é válido	Confiabilidade Segurança Verificação		X

F3: Cadastrar Aula		Oculto ()		
Descrição: Tem por objetivo cadastrar uma vídeo aula no sistema. Informando nome, nível de ensino, área de ensino, professor proprietário, se é gratuita ou valor.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 3.1 Acesso ao sistema pelo professor proprietário.	Solicitar dados: Login e Senha.	Confiabilidade Segurança		X
NF 3.2 Usuário Informa os Dados	Valida os dados de entrada. Caso corretos, o professor terá acesso às funcionalidades do sistema para cadastrar a aula.	Confiabilidade		X
NF 3.3 Usuário informa a área de ensino	O usuário deverá cadastrar a aula de acordo com sua área de atuação. Caso o Usuário cadastre uma aula fora de sua área e for identificado, retornará um aviso e a aula não será registrada.	Confiabilidade Verificação	X	X
NF 3.4 Usuário informa o nível de ensino	O usuário deverá cadastrar a aula de acordo com seu nível de formação. Caso o Usuário cadastre uma aula fora de seu nível de formação, retornará um aviso e a aula não será registrada.	Confiabilidade Verificação	X	X

F4: Cadastrar Curso		Oculto ()		
Descrição: Tem por objetivo cadastrar um curso no sistema, cursos podem ter múltiplos professores proprietários e possuir diversas aulas, se é gratuito ou valor, cursos não precisam ser únicos, mas professores não podem disponibilizar o mesmo curso mais de uma vez durante o mesmo período.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 4.1 Acesso ao sistema por um professor proprietário.	Solicitar dados: Login e Senha.	Confiabilidade Segurança		X
NF 4.2 Usuário informa os dados	Valida os dados de entrada. Caso corretos, o professor terá acesso às funcionalidades do sistema para cadastrar o curso.	Confiabilidade		X
NF 4.3 Verificar se os usuários atendem aos requisitos propostos da área de atuação	O sistema deve verificar se os professores têm formação naquela área para disponibilizar o curso.	Confiabilidade Segurança Verificação	X	X
NF 4.4 Verificar se usuário tem nível de formação adequado	O sistema deve verificar se todas as aulas do curso são compatíveis com o nível de formação dos professores.	Confiabilidade Verificação		X
NF 4.5 Verificar aulas	O sistema deve verificar se as aulas adicionadas ao curso são de propriedade do professor ou professores que estão criando o curso	Confiabilidade Verificação		X

F5: Atualizar aluno		Oculto ()		
Descrição: Tem por objetivo atualizar um aluno já cadastrado no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 5.1 Requisitos de autorização	O usuário que desejar atualizar o aluno precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 5.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após atualizar o aluno.	Interoperabilidade		X

F6: Atualizar Professor		Oculto ()		
Descrição: Tem por objetivo atualizar um professor já cadastrado no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 6.1 Requisitos de autorização	O usuário que desejar atualizar o professor precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 6.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após atualizar o professor .	Interoperabilidade		X

F7: Atualizar Aula		Oculto ()		
Descrição: Tem por objetivo atualizar uma vídeo aula já cadastrada no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 7.1 Requisitos de autorização	O usuário que desejar atualizar a aula precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 7.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após atualizar a aula.	Interoperabilidade		X

F8: Atualizar Curso		Oculto ()		
Descrição: Tem por objetivo atualizar um curso já cadastrado no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 8.1 Requisitos de autorização	O usuário que desejar atualizar o curso precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 8.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após atualizar o curso .	Interoperabilidade		X

F9: Deletar aluno		Oculto ()		
Descrição: Tem por objetivo deletar um aluno já cadastrado no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 9.1 Requisitos de autorização	O usuário que desejar deletar o aluno precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 9.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após deletar o aluno.	Interoperabilidade		X

F10: Deletar Professor		Oculto ()		
Descrição: Tem por objetivo deletar um professor já cadastrado no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 10.1 Requisitos de autorização	O usuário que desejar deletar o professor precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 10.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após deletar o professor .	Interoperabilidade		X

F11: Deletar Aula		Oculto ()		
Descrição: Tem por objetivo deletar uma aula já cadastrada no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 11.1 Requisitos de autorização	O usuário que desejar deletar a aula precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 11.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após deletar a aula .	Interoperabilidade		X

F12: Deletar Curso		Oculto ()		
Descrição: Tem por objetivo deletar um curso já cadastrado no sistema, pedindo um usuário e senha válidos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 12.1 Requisitos de autorização	O usuário que deseja deletar o curso precisa ter credenciais válidas.	Confiabilidade Segurança Autorização		X
NF 12.2 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após deletar o curso .	Interoperabilidade		X

F13: Buscar Aluno		Oculto ()		
Descrição: Tem por objetivo realizar a busca de um aluno já cadastrado no sistema, a partir do seu nome ou usuário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 13 .1 Requisitos de validação	O aluno pesquisado deve ter nome ou usuário encontrado no banco de dados.	Confiabilidade Segurança Validação	X	X
NF13 .2 Velocidade de apresentação	O sistema deve fazer a busca no banco de dados de maneira eficiente.	Eficiência	X	
NF 13.3 Apresentação	O sistema deverá mostrar as informações do aluno pesquisado .	Interoperabilidade	X	X

F14: Buscar Professor		Oculto ()		
Descrição: Tem por objetivo realizar a busca de um professor já cadastrado no sistema, a partir do seu nome, usuário, aulas, área de ensino, nível de ensino ou avaliação.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 14.1 Requisitos de validação	O professor pesquisado deve ter nome, usuário, aulas, área de ensino, nível de ensino ou avaliação, encontrado no banco de dados.	Confiabilidade Segurança Validação	X	X
NF 14.2 Velocidade de apresentação	O sistema deve fazer a busca no banco de dados de maneira eficiente	Eficiência	X	
NF 14.3 Apresentação	O sistema deverá mostrar as informações do professor pesquisado .	Interoperabilidade	X	X

F15: Buscar Aula		Oculto ()		
Descrição: Tem por objetivo realizar a busca de uma aula já cadastrada no sistema, a partir do seu nome, área de ensino, nível de ensino, valor, avaliação ou professor proprietário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 15.1 Requisitos de validação	A aula pesquisada deve ter nome, área de ensino, nível de ensino, valor, avaliação professor proprietário encontrado no banco de dados.	Confiabilidade Segurança Validação	X	X
NF 15.2 Velocidade de apresentação	O sistema deve fazer a busca no banco de dados de maneira eficiente	Eficiência	X	
NF 15.3 Apresentação	O sistema deverá mostrar as informações da aula pesquisada .	Interoperabilidade	X	X

F16: Buscar Curso		Oculto ()		
Descrição: Tem por objetivo realizar a busca de um curso já cadastrado no sistema, a partir do seu nome, área de ensino, nível de ensino, valor, avaliação ou professor proprietário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 16.1 Requisitos de validação	O curso pesquisado deve ter nome, área de ensino, nível de ensino, valor, avaliação ou professor proprietário encontrado no banco de dados.	Confiabilidade Segurança Validação	X	X
NF 16.2 Velocidade de apresentação	O sistema deve fazer a busca no banco de dados de maneira eficiente	Eficiência	X	
NF 16.3 Apresentação	O sistema deverá mostrar as informações do curso pesquisado .	Interoperabilidade	X	X

F17: Compra de Aula		Oculto ()		
Descrição: Tem por objetivo realizar a compra de uma aula já cadastrada no sistema por um aluno ou professor, pode ser comprada uma ou várias aulas. O sistema deve calcular o valor e chamar o serviço de compra terceirizada.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 17.1 Confirmação do Pagamento	Após chamar o serviço de compra, o sistema deve verificar a confirmação da compra.	Confiabilidade Segurança		X
NF 17.2 Verificação de pendência	O sistema deve verificar se o usuário que está comprando não tem nenhuma outra compra feita no sistema com o pagamento pendente	Verificação		X
NF 17.3 Velocidade de compra	O sistema deve fazer a compra de maneira eficiente	Eficiência	X	
NF 17.4 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após a compra ser realizada.	Interoperabilidade		X

F18: Compra de Curso		Oculto ()		
Descrição: Tem por objetivo realizar a compra de um curso já cadastrada no sistema por um aluno ou professor, pode ser comprada um ou vários cursos. O sistema deve calcular o valor e chamar o serviço de compra terceirizada.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 18.1 Confirmação do Pagamento	Após chamar o serviço de compra, o sistema deve verificar a confirmação da compra.	Confiabilidade Segurança		X
NF 18.2 Verificação de pendência	O sistema deve verificar se o usuário que está comprando não tem nenhuma outra compra feita no sistema com o pagamento pendente	Verificação		X
NF 18.3 Velocidade de compra	O sistema deve fazer a compra de maneira eficiente	Eficiência	X	
NF 18.4 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após a compra ser realizada.	Interoperabilidade		X

F19: Pagamento de Professor		Oculto ()		
Descrição: Tem por objetivo realizar o pagamento ao professor ou aos professores proprietários de uma aula ou curso comprado no sistema. O sistema deve calcular o valor e chamar o serviço de pagamento terceirizado.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 19.1 Confirmação do Pagamento	Após chamar o serviço de pagamento, o sistema deve verificar a confirmação do pagamento.	Confiabilidade Segurança		X
NF 19.2 Velocidade de pagamento	O sistema deve fazer o pagamento de maneira eficiente	Eficiência	X	

F20: Avaliação de Aula		Oculto ()		
Descrição: Tem por objetivo realizar a avaliação de uma aula vista por um aluno ou professor.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 20.1 Confirmação da visualização do aluno ou professor	O sistema deve confirmar que o aluno ou professor viu a aula que está avaliando.	Confiabilidade		X
NF 20.2 Calcular avaliação de professor proprietário	O sistema deve recalcular a avaliação de professores proprietários após a nova avaliação ser registrada.	Confiabilidade		X
NF 20.3 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após a avaliação ser realizada.	Interoperabilidade		X

F21: Avaliação de Curso		Oculto ()		
Descrição: Tem por objetivo realizar a avaliação de um curso visto por um aluno ou professor, para poder avaliar o aluno ou professor precisa ter assistido pelo menos três aulas do curso.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 21.1 Confirmação da visualização do aluno ou professor	O sistema deve confirmar que o aluno ou professor viu pelo menos três aulas do curso que está avaliando.	Confiabilidade		X
NF 21.2 Calcular avaliação de professor proprietário	O sistema deve recalcular a avaliação de professores proprietários após a nova avaliação ser registrada.	Confiabilidade		X
NF 21.3 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após a avaliação ser realizada.	Interoperabilidade		X

F22: Avaliação do Professor		Oculto ()		
Descrição: Tem por objetivo realizar a avaliação de um professor, a avaliação do professor é feita a partir do cálculo da média das avaliações de suas aulas e cursos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 22.1 Velocidade de cálculo de média	O sistema deve fazer o cálculo da média de maneira eficiente	Eficiência	X	

F23: Denúncia de conteúdo		Oculto ()		
Descrição: Tem por objetivo realizar a denúncia de um aluno ou professor sobre o conteúdo de uma aula ou de um curso. Caso seja confirmada a denúncia a aula ou curso será bloqueado, caso seja um caso grave ou recorrente todas as aulas e cursos do professor ou professores proprietários serão bloqueadas. Além do professor ou professores não poderem colocar novas aulas ou cursos. A denúncia pode ser ou de conteúdo indevido ou de ministrar conteúdo sobre assuntos que não são da formação ou do nível de formação do professor.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 23.1 Velocidade de análise da denúncia	O sistema deve fazer a análise da denúncia de maneira eficiente	Eficiência	X	
NF 23.1 Verificar gravidade	O sistema deve identificar violação grave ou não	Confiabilidade Segurança Verificação	X	X
NF 23.2 Verificar Recorrência	O sistema deve verificar se o professor ou professores tem denúncias recorrentes	Confiabilidade Segurança Verificação	X	X
NF 23.4 Confirmação	O sistema deverá mostrar uma mensagem na tela de confirmação após a denúncia ser realizada.	Interoperabilidade		X

F24: Exibir aula		Oculto ()		
Descrição: Tem por objetivo exibir uma aula selecionada por um usuário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 24.1 Velocidade de exibição	O sistema deve exibir o vídeo de maneira eficiente.	Eficiência	X	

F25: Bloquear CPF		Oculto ()		
Descrição: Caso um professor seja configurado com uma denúncia grave ou com denúncias recorrentes suas aulas além de serem bloqueadas e depois deletadas se não houver devida defesa, seu CPF será registrado como bloqueado para que não faça adição de novos conteúdos.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 25.1 Integridade	O sistema deve garantir que seja sigiloso e seguro de ataques a lista dos CPF bloqueados.	Segurança Confiabilidade		X

2.2 REQUISITOS SUPLEMENTARES

Nome	Restrição	Categoria	Desejável	Permanente
S1 Legal	O sistema deve seguir todas as normas legais previstas na constituição.	Legal	()	(X)
S2 Interface	As interfaces do sistema devem ser implementadas de forma a rodar em navegadores Web que suportem html	Interface	()	(X)
S3 Armazenamento de dados	A camada de persistência deve ser realizada de forma a permitir integração com diferentes bancos de dados	Persistência	()	(X)
S4 Confiabilidade	O sistema deverá ter alta disponibilidade 24 horas	Segurança	()	(X)
S5 Ferramentas	O sistema deve usar para modelar páginas a ferramenta Figma e para implantação no Frontend será usado React e no Backend Rails. Os testes serão feitos com a biblioteca Jest feita em JavaScript.	Ferramenta	()	(X)
S6 Ético	O sistema deve garantir a integridade dos dados pessoais de seus usuários.	Ético	()	(X)
S7 Eficiência	O sistema deve realizar todos seus procedimentos de maneira eficiente evitando lentidão e travamento da reprodução das aulas.	Eficiência	()	(X)
S8 Perfil de usuários	Alunos: acesso apenas às suas aulas e cursos Professor: acesso apenas às suas aulas e cursos, pode fazer modificações em aulas e cursos que é proprietário Administrador: usuário da plataforma que tem acesso a todas as aulas e cursos, podendo efetuar qualquer operação do sistema Convidado: pode acessar o site e ver três aulas grátis sem se cadastrar	Segurança	()	(X)

2.3 Organização de requisitos em consultas:

Nome	Referências Cruzadas
Atualizar usuário, aulas e/ou cursos	F1, F2, F5, F6, F8, F7
Adicionar aulas gravadas	F2, F4, F3
Buscar usuário, aula e/ou curso	F13, F14, F15, F16, F24
Compras de cursos e/ou aulas	F13, F14, F15, F16, F17, F18, F19
Vendas de cursos e/ou aulas	F4, F3 F18, F17, F19
Avaliação do conteúdo	F21, F20, F22, F24
Deletar usuário, aula e/ou curso	F9, F10, F11, F12
Suporte para denúncia	F3, F4, F10, F11, F12, F23, F25

2.4 Processos de Negócio

Nome	Atores	Descrição	Referências Cruzadas
Exibir	Aluno ou professor	O usuário irá navegar pela interface da aplicação selecionar uma aula ou curso que será exibido.	F13, F14, F15, F16, F24
Avaliar	Aluno ou professor	O usuário vai assistir. Depois disso poderá avaliar a aula ou curso.	F20, F21, F22, F24
Compra	Aluno ou professor	O usuário irá navegar pela interface, selecionar as aulas ou cursos que deseja comprar, o cálculo será feito pelo sistema. Caso haja pendências a compra será negada, se não o sistema irá realizar a compra. Após isso, um pagamento aos professores proprietários será disparado.	F13, F14, F15, F16, F17, F18, F19
Venda	Professor	O professor irá cadastrar a aula ou curso e a partir disso caso haja uma compra ele irá receber o pagamento.	F3, F4, F17, F18, F19
Denúncia	Aluno ou Professor e Administrador do Sistema	Após uma denúncia ser feita e confirmada a aula ou comentário será bloqueado e caso não houver defesa por parte do usuário o comentário ou aula será deletado após 3 meses e o usuário bloqueado temporariamente. Se a denúncia for grave ou recorrente todo o conteúdo desse usuário será deletado após 6 meses sem devida defesa. Além do CPF cadastrado pelo usuário ser bloqueado.	F3, F4, F10, F11, F12, F23, F25

2.5 Manutenção de Conceitos

Conceito	I	A	E	C	Observação	Ref. Cruzadas
Aluno	X	X	X	X	Só é possível criar se tiver CPF válido.	F1
Professor	X	X	X	X	Só é possível criar se tiver CPF válido e não tiver CPF na lista de bloqueados.	F2, F25
Aula	X	X	X	X	Só é possível ser criada por professores que tenham diplomas na área e no nível de conhecimento necessário.	F3
Curso	X	X	X	X	Só é possível ser criada por professores que tenham diplomas na área e no nível de conhecimento necessário.	F4

2.6 Casos de Uso Expandidos

2.6.1 Exibir aula ou aula de um curso

Fluxo Principal:

1. O usuário irá navegar pela interface e escolher qual aula ele quer ver
2. [RS] O sistema exibe a aula selecionada

Tratamento de Exceções:

- 1a. O usuário não possui a aula ou o curso que contém a aula que deseja ver
 - 1a. 1 O cliente escolhe se compra a aula ou o curso que contém a aula
 - 1a. 2 [EV1] O sistema realiza a compra
 - 1a. 3 Retorna ao fluxo principal no passo 2
- 1b. O usuário não possui cadastro
 - 1b. 1 [EV2] O cliente deve realizar seu cadastro
 - 1b. 2 Retorna ao fluxo principal no passo 2

2.6.2 Avaliar aula ou curso

Fluxo Principal:

1. O usuário deve assistir a aula completa ou curso completo
2. O usuário vai escolher uma nota de 1 a 5
3. Caso o usuário quiser ele pode deixar um comentário sobre a aula ou curso
4. [EV3] O sistema registra a avaliação

Tratamento de Exceções:

- 2a. O usuário já tinha avaliado a aula ou curso antes
 - 2a. 1 O sistema vai confirmar se o usuário quer avaliar a aula ou curso novamente
 - 2a. 2 Retorna ao fluxo principal no passo 4
- 4a. O usuário utilizou vocabulário impróprio no comentário
 - 4a. 1 [EV4] Será solicitado a mudança desse trecho do comentário
 - 4a. 2 Retorna ao fluxo principal no passo 5

2.6.3 Comprar aula ou curso

Fluxo Principal:

1. O usuário deve selecionar aula ou curso

2. O sistema adiciona ao carrinho a aula ou curso selecionado
3. O usuário seleciona a finalização da compra
4. O usuário informa o modo de pagamento que deseja utilizar
5. [EV1] O sistema realiza a compra

Tratamento de Exceções:

- 3a. O usuário possui débitos pendentes
 - 3a. 1 [EV5] O usuário escolhe a opção de quitar os débitos e continuar a compra
 - 3a. 2 Retorna ao fluxo principal no passo 4
- 4a. O usuário escolhe pagamento com cartão e não possui um cartão registrado
 - 4a. 1 [EV6] O usuário cadastra um novo cartão
 - 4a. 2 Retorna ao fluxo principal no passo 5
- 5a. Forma de pagamento não autorizada
 - 5a. 1 [EV7] O sistema vai solicitar nova forma de pagamento
 - 5a. 2 Retorna ao fluxo principal no passo 5

2.6.4 Venda aula ou curso

Fluxo Principal:

1. Um usuário realiza a compra de algum aula ou curso
2. [EV8] O sistema realiza o pagamento

Tratamento de Exceções:

- 3a. O sistema não consegue realizar o pagamento de algum proprietário
 - 3a. 1 O sistema informa o usuário sobre essa situação
 - 3a. 2 Após o usuário resolver ele avisa o sistema
 - 3a. 3 Retorna ao fluxo principal no passo 2

2.6.5 Denunciar Aluno ou Professor

Fluxo Principal:

1. [EV9] O usuário visualiza o conteúdo impróprio e clica em denunciar
2. Os administradores analisam a denúncia
3. [EV10] Caso seja identificado uma infração o conteúdo será deletado e o usuário que criou esse conteúdo, seja ele uma aula ou comentário, terá sua conta bloqueada

2.7 Contratos

2.7.1 CO1: [RS] O sistema exibe a aula selecionada

Classe Aula:

Consulta: `exibirAula(idAula: int)`

Pré:

- Existe um `usuarioAtual`
- Existe uma *Aula* com *id* igual a *idAula*

Resultados:

- Uma aula está sendo exibida

Em OCL

Alias: `aula = self.aula -> select(id=idAula)`

Pré:

`self.usuarioAtual -> size() == 1`
`aula -> size() == 1`

Resultados:

Exibe vídeo

2.7.2 CO2: [EV1] O sistema realiza a compra

Classe Compra:

Operação: `realizarCompra(aulasCompra: Aulas[], tipoCompra: int)`

Pré:

- Existe *Aulas* com *ids* igual a *aulasCompra.id*
- Existe um `usuarioAtual`

Pos:

- Foi criada uma nova *Compra*
- O *atributo* `id` foi alterado para um novo `int`
- O *atributo* `usuário` foi alterado para `usuarioAtual.cpf`
- O *atributo* `valor` foi alterado para soma dos valores das *aulas*
- O *atributo* `tipo` foi alterado para `tipoCompra`

Em OCL

Alias: `aulas = self.aulas -> select(aulas.id=aulasCompra.id)`

Pré:


```
self.usuarioAtual -> size() == 1
aulas.id -> size() == 1
```

Pós:

```
compra = Compra.new()
compra.id = novo int
compra.usuario = usuarioAtual.cpf
compra.valor = soma(self.aulas.valor)
compra.tipo = tipoCompra
```

2.7.3 CO3:[EV2] O cliente deve realizar seu cadastro**Classe Usuário:**

Operação: cadastrar(nomeNovoUsuário: string, cpfNovoUsuário: int, emailNovoUsuário: string, idadeNovoUsuário: int)

Pré:

- Não existir um *Usuário* com o *cpf* igual a *cpfNovoUsuário*

Pos:

- Foi criado um novo *Usuário*
- O *atributo* nome foi alterado para *nomeNovoUsuário*
- O *atributo* cpf foi alterado para *cpfNovoUsuário*
- O *atributo* email foi alterado para *emailNovoUsuario*
- O *atributo* idade foi alterado para *idadeNovoUsuário*

Em OCL

Alias: usuario = self.usuario -> select(cpf=cpfNovoUsuário)

Pré: usuario -> size() == 0

Pós:

```
usuario = Usuario.new()
usuario.nome = nomeNovoUsuário
usuario.cpf = cpfNovoUsuário
usuario.email = emailNovoUsuário
usuario.idade = idadeNovoUsuário
```

2.7.4 CO4:[EV3] O sistema registra a avaliação

Classe Avaliação:

Operação: avaliar(idAvaliado: Int, novaNota: Int, novoComentario: String)

Pre:

- Existe um usuarioAtual
- Não existe avaliação *Aula ou Curso* com *id* igual a idAvaliado

Pós-condição:

- Foi criada uma nova Avaliacao
- O *atributo* id foi alterado para um novo int
- O *atributo* usuario foi alterado para usuarioAtual.cpf
- O *atributo* avaliado foi alterado para idAvaliado
- O *atributo* nota foi alterado para novaNota
- O *atributo* comentario foi alterado para novoComentario

Em OCL

Alias: avaliacao = self.avaliacao -> select(avaliacao.avaliado = idAvaliado)

Pré:

avaliacao -> size() == 0
self.usuarioAtual -> size() == 1

Pós:

avaliacao = Avaliacao.new()
avaliacao.id = novo int
avaliacao.usuario = usuarioAtual.cpf
avaliacao.avaliado = idAvaliado
avaliacao.nota = novaNota
avaliacao.comentario = novoComentario

2.7.5 CO5: [EV4] Será solicitado a mudança desse trecho do comentário

Classe Avaliação:

Operação: mudarComentario(avaliacaoAntiga: Avaliacao, novoComentario: String)

Pre:

- Existe um usuarioAtual
- Existe uma *Avaliacao* com *id* igual a *avaliacaoAntiga.id*

Pos:

- O *atributo* comentario foi alterado para novoComentario

Em OCL

Alias: avaliacao = self.avaliacao -> select(avaliacao.id=avaliacaoAntiga.id)

Pré:

self.usuarioAtual -> size() == 1
avaliacao -> size() == 1

Pos:

avaliacao.comentario = novoComentario

2.7.6 CO6:[EV5] O usuário escolhe a opção de quitar os débitos e continuar a compra

Classe Compra

Operação: quitarDebitos(aulasQuitaDebito: Aulas[], tipoQuitarDebito: int)

Pré:

- Existe *Aulas* com *ids* igual a *aulas.id*.
- Existe um *usuarioAtual*

Pos:

- Foi criada uma nova *QuitarDebito*
- O *atributo* id foi alterado para um novo int
- O *atributo* usuario foi alterado para *usuarioAtual.cpf*
- O *atributo* valor foi alterado para soma dos valores das *aulas*
- O *atributo* tipo foi alterado para *tipoQuitarDebito*

Em OCL

Alias: aulas = self.aulas -> select(aulas.id=aulasQuitaDebito.id)

Pré:

self.usuarioAtual -> size() == 1
aulas.id -> size() == 1

Pós:

quitarDebito = *QuitarDebito*.new()
quitarDebito.id = novo int
quitarDebito.usuario = *usuarioAtual.cpf*
quitarDebito.valor = soma(self.aulas.valor)
quitarDebito.tipo = *tipoQuitarDebito*

2.6.7 CO7:[EV6] O usuário cadastra um novo cartão

Classe Cartao:

Operação: `cadastrarCartao(numeroCartao: String, cvvCartao: Int, vencimentoCartao: String)`

Pré-condições:

- Não existir Cartao com numeroCartao já cadastrado no sistema
- Existe um usuarioAtual

Pós-condição:

- Foi cadastrado um novo *Cartão*
- O *atributo* usuario foi alterado para usuarioAtual.cpf
- O *atributo* numero foi alterado para numeroCartao
- O *atributo* cvv foi alterado para cvvCartao
- O *atributo* vencimento foi alterado para vencimentoCartao

Em OCL

Alias: `cartao = self.cartao -> select(cartao=numeroCartao)`

Pré:

`cartao -> size() == 0`
`self.usuarioAtual -> size() == 1`

Pós:

`cartao = Cartao.new()`
`cartao.usuario = usuarioAtual.cpf`
`cartao.numero = numeroCartao`
`cartao.cvv = cvvCartao`
`cadastro.vencimento = vencimentoCartao`

2.7.8 CO8: [EV7] O sistema vai solicitar nova forma de pagamento

Classe Pagamento:

Operação: `novaFormaDePagamento(tipoNovaForma: Int, dadosNovaForma: DadosFormaPagamento{ })`

Pre:

- Existe um usuarioAtual
- Forma de pagamento nao existir para o usuarioAtual

Pós:

- Foi cadastrada uma nova *FormaPagamento*
- O *atributo* usuario foi alterado para usuarioAtual.cpf
- O *atributo* tipo foi alterado para tipoNovaForma
- O *atributo* dados foi alterado para dadosNovaForma

Em OCL

Alias: formaPagamento = self.formaPagamento ->
select(usuarioAtual.formasPagamento=tipoNovaForma)

Pré:

formaPagamento -> size() == 0
self.usuarioAtual -> size() == 1

Pós:

formaPagamento = FormaPagamento.new()
formaPagamento.usuario = usuarioAtual.cpf
formaPagamento.tipo = tipoNovaForma
formaPagamento.dados = dadosNovaForma

2.7.9 CO9:[EV8] O sistema realiza o pagamento

Classe Pagamento:

Operação: pagar(aulasPagamento: Aulas[], tipoPagamento: Int, usuarioPagamento: Int, porcentagem: Float)

Pré:

- Existe *Aulas* com *ids* igual a *aulasPagamento.id*
- Existe um *Usuario* com *id* igual a *usuarioPagamento*

Pos:

- Foi criado um novo *Pagamento*
- O *atributo* id foi alterado para um novo int
- O *atributo* usuario foi alterado para usuarioPagamento
- O *atributo* valor foi alterado para soma dos valores das *aulas*
- O *atributo* tipo foi alterado para tipoPagamento

Em OCL

Alias:

aulas = self.aulas -> select(aulas.id=aulasPagamento.id)
usuario = self.usuario -> select(usuario.id=usuarioPagamento)

Pré:

usuario.id -> size() == 1
aulas.id -> size() == 1

Pós:

pagamento = Pagamento.new()
pagamento.id = novo int
pagamento.usuario = usuarioPagamento
pagamento.valor = soma(self.aulas.valor*porcentagem)
pagamento.tipo = tipoPagamento

2.7.10 CO10:[EV9] O usuário visualiza o conteúdo impróprio e clica em denunciar

Classe Denuncia:

Operação: reportar(idDenunciado: Int, descricaoDenuncia: String)

Pré:

- Existe uma *Aula* ou *Avaliacao* com *id* igual a idDenunciado
- Existe um usuarioAtual

Pos:

- Foi criada uma nova *Denuncia*
- O *atributo* id foi alterado para um novo int
- O *atributo* usuario foi alterado para usuarioAtual
- O *atributo* descricao foi alterado para descricaoDenuncia
- O *atributo* denunciado foi alterado para idDenunciado

Em OCL

Alias: denunciado = self.denunciado -> select(denunciado.id=idAula || idAvaliacao)

Pré:

self.usuarioAtual -> size() == 1
denunciado -> size() == 1

Pós:

denuncia = Denuncia.new()
denuncia.id = novo int
denuncia.usuario = usuarioAtual.cpf
denuncia.descricao = *descricaoDenuncia*
denuncia.denunciado = idDenunciado

2.7.11 CO11:[EV10] Caso seja identificado uma infração o conteúdo será deletado e o usuário que criou esse conteúdo, seja ele uma aula ou comentário, terá sua conta bloqueada

Classe Denuncia:

Operação: deletarBloquear(idDenunciado: Int, idUsuario: Int)

Pré:

- Existe *Denuncia* com *id* igual a idDenunciado
- Existe um *Usuário* com *id* igual a idUsuario

Pos:

- Foi criada uma nova *DeletaBloquear*
- O *atributo* id foi alterado para um novo int
- O *atributo* usuario foi alterado para idUsuario
- O *atributo* denunciado foi alterado para idDenunciado

- Foi deletado a Aula ou Avaliação com id igual a idDenunciado
- O Usuário com id igual a idUsuario foi bloqueado

.

Em OCL

Alias:

denunciado = self.denunciado -> select(denunciado.id=idAula || idAvaliacao)
usuario = self.usuario -> select(usuario.id=idUsuario)

Pré:

usuario -> size() == 1
denunciado -> size() == 1

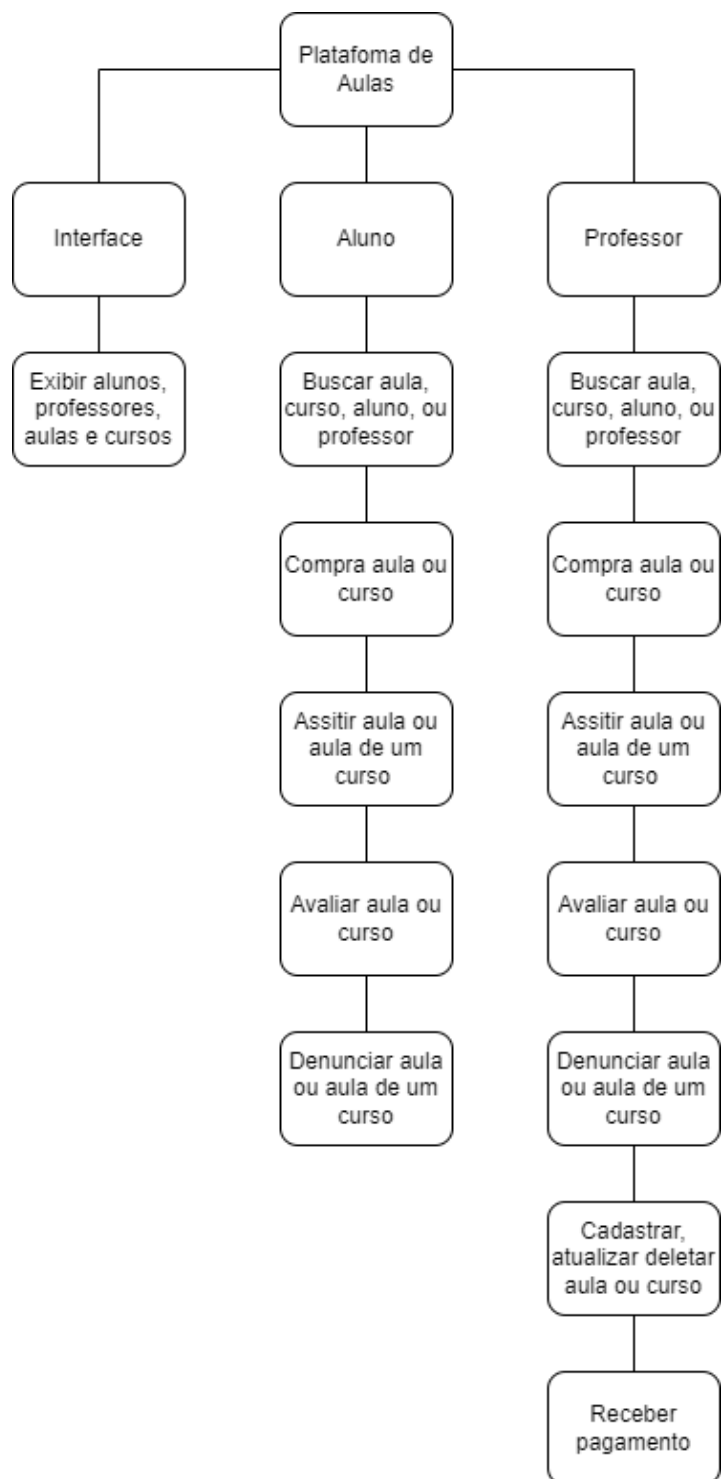
Pós:

deletarBloquear = *DeletaBloquear*.new()
deletarBloquear.id = novo int
deletarBloquear.usuario = idUsuario
deletarBloquear.denunciado = idDenunciado
denunciado -> size() == 0
usuario.bloqueado == 1

3-Estrutura Analítica do Projeto (E

3.1 Definimos o EAP do projeto no diagrama da imagem 1.

Imagem 1. Diagrama EAP



3.2 A seguir a sequência de atividades será detalhada:

- **Exibir alunos, professores, aulas e cursos**
 1. Listar aulas, professores, aulas e cursos 4(SE)
 2. Entidades aluno, professor, aula e curso 4(ALI)
- **Cadastrar, atualizar, deletar aula ou curso**
 1. Cadastrar 2(EF) e mensagem de feedback 2(SE)
 2. Atualizar 2(EF) e mensagem de feedback 2(SE)
 3. Deletar 2(EF) e mensagem de feedback 2(SE)
- **Receber pagamento**
 1. Pedido de pagamento (EF)
 2. Processamento do pagamento (ALI)
 3. Mensagem de feedback (SE)
- **Buscar aula, curso, aluno ou professor**
 1. Busca 4(CE)
- **Compra aula ou curso**
 1. Pedido de compra 2(EF)
 2. Processamento da compra (ALI)
 3. Mensagem de feedback 2(SE)
- **Assistir aula ou aula de um curso**
 1. Pedido de exibição (EF)
 2. Exibir a aula (SE)
- **Avaliar aula ou curso**
 1. Registro de avaliação 2(EF)
 2. Entidade avaliação 2(ALI)
 3. Cálculo da avaliação do professor 2(ALI)
 4. Mensagem de feedback 2(SE)
- **Denunciar aula ou aula de curso**
 1. Registro da denúncia (EF)
 2. Entidade denuncia (ALI)
 3. Verificação das condições e consequências da denúncia (ALI)
 4. Mensagem de feedback (SE)

4-Estimativa de duração do projeto

Para estimar o tempo de duração do projeto, serão utilizadas as estimativas paramétricas COCOMO, em conjunto com o APF.

4.1 Análise de Pontos de Função

O projeto tem no total de atividades 17(SE), 12(ALI), 14(EE), 4(CE) e 0(AIE). Será definido a complexidade do projeto como média devido a quantidade de entidades não ser alta, mas ter diversas relações entre elas. A tabela 1 ilustrada a seguir, será utilizada como base para calcular os pesos de cada atividade.

Elemento/Complexidade	Baixa	Média	Alta
Entrada Externa (EE)	3	4	6
Saída Externa (SE)	4	5	7
Consultas Externas (CE)	3	4	6
Arquivos Lógicos Internos (ALI)	7	10	15
Arquivos de Interface Externos (AIE)	5	7	10

Tabela 1. Tabela de Complexidade das Atividades

Com isso, o valor do PFNA será:

$$\text{PFNA} = 17*5 + 12*10 + 14*4 + 4*4 + 0*7 = 277$$

Questões de Ajuste:

- I. Necessita de backup? 5
- II. Necessita de mecanismos especializados de comunicação? 1
- III. Tem processamento distribuído? 0
- IV. Precisa de alto desempenho? 3
- V. Terá grande número de usuários em paralelo? 5
- VI. Precisar de entrada de dados on-line? 5
- VII. No caso de entradas on-line, existirão múltiplas telas? 5
- VIII. A atualização das entidades será feita on-line? 3
- IX. As entradas e saídas de dados serão complexas? 2
- X. O processamento interno será complexo? 2
- XI. O código será projetado para ser reutilizado? 5

XII. Migração e instalação estarão incluídos? 0

XIII. O sistema será instalado em diversas organizações? 0

XIV. O projeto pretende facilitar mudanças e operação do usuário? 4

Soma: 40

Com o valor do PFNA e a soma das respostas para normalização, podemos calcular o PF.

$$\mathbf{PF} = \mathbf{PFNA} * (0.65 + 0.01 * \Sigma \text{Respostas})$$

$$\mathbf{PF} = 277 * (0.65 + 0.01 * 40)$$

$$\mathbf{PF} = 290.85$$

Cálculo de KLOCS:

O projeto será realizado em JavaScript, então para realização do cálculo a quantidade de LOCS para cada PF será baseada nos valores da linguagem JAVA. Ou seja. 53 LOCS por PF.

$$\mathbf{LOCS} = \mathbf{PF} * 53 = 290.85 * 53 = 15415,05$$

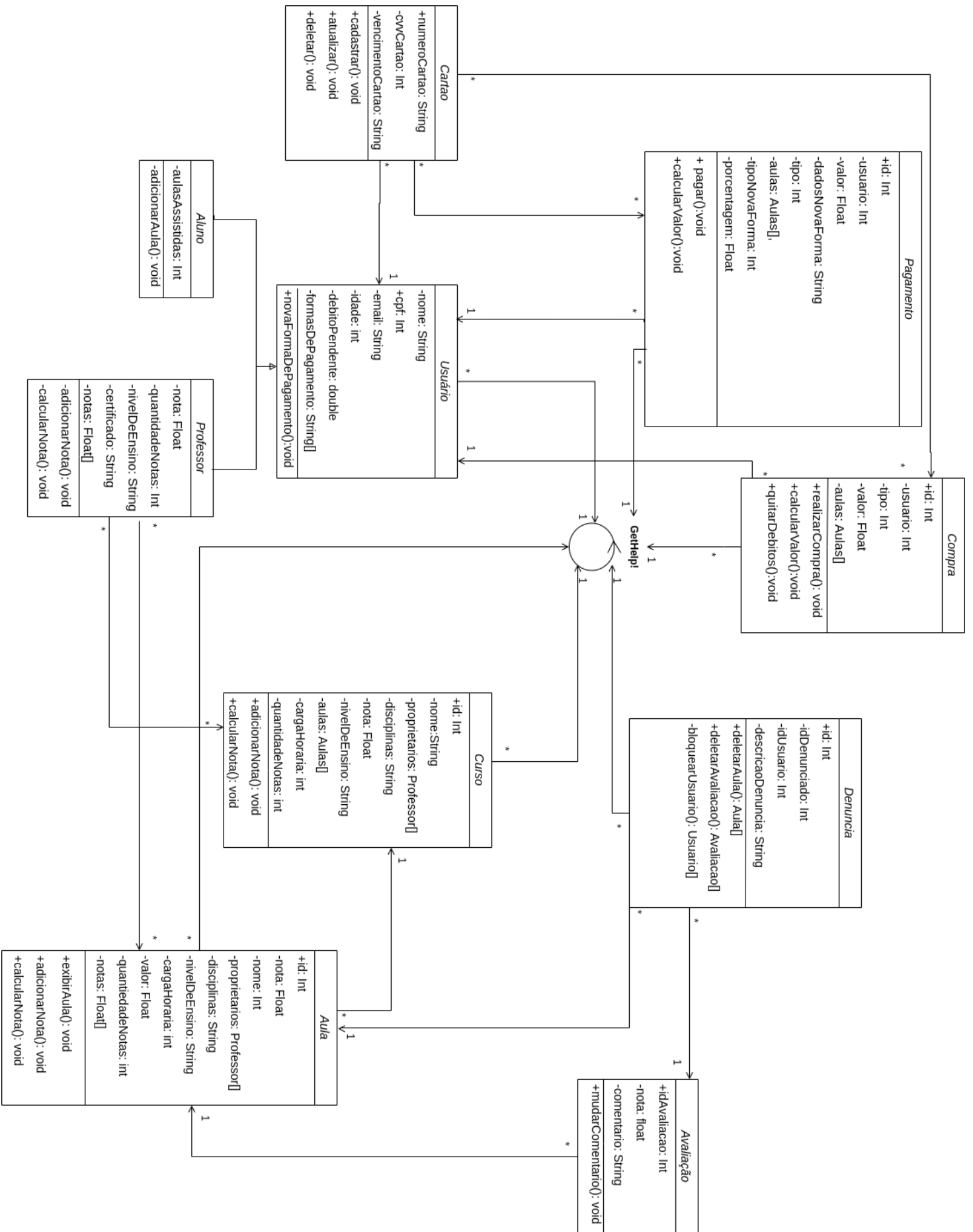
$$\mathbf{KLOCS} = \mathbf{LOCS}/1000 = 15415,05/1000 = 15,41505$$

Cálculo do COCOMO, como o sistema lida com pagamentos e streaming de vídeos será considerado uma complexidade média:

$$\mathbf{Esforço} = 3 * 15,41505^{1,12} = 64,2125$$

$$\mathbf{Duração} = 2,5 * 64,2125^{0,35} = 10,73$$

Com isso, a duração do projeto será de aproximadamente 11 meses.



Na fase 3 do nosso trabalho implementamos todas as classes do Diagrama de Classes do Projeto, mas as classes que consideramos mais relevantes para os testes no nosso trabalho foram: Aluno, Aula, Compra, Curso, Pagamento, Professor e Usuário.

Foco dos testes

Classe AlunoTests: Fizemos o teste com o objetivo de adicionar quantidade de aula.

Classe AulaTests: Fizemos o teste para adicionar nota para determinada aula, o que seria um parâmetro de avaliação pela aula ali disponibilizada. Outro teste feito foi de calcular nota onde nesse teste é passado o valor para tirar média da soma das notas divididas pela quantidade de notas já contabilizadas para aquela determinada aula. Por fim, foi feita uma função para confirmar se a aula está sendo exibida.

Classe CompraTests: Esse teste tem como objetivo calcular o valor de uma compra de aula ou curso, realizar compra de uma aula ou curso e quitar débitos pendentes do usuário.

Classe CursoTests: Nesta classe fizemos o teste para adicionar nota para determinado curso, onde seria um parâmetro de avaliação pela o curso ali disponibilizado. Outro teste feito foi de calcular nota onde nesse teste é passado o valor para tirar média da soma das notas divididas pela quantidade de notas já contabilizadas para aquele determinado curso.

Classe PagamentoTests: Nesse teste é verificado o cálculo do valor a ser pago e é feita a confirmação do pagamento.

Classe ProfessorTests: Fizemos o teste para adicionar nota para determinado professor, o que seria o parâmetro de avaliação dos usuários para aula/curso ministrado. Outro teste feito foi de calcular a nota de avaliação deste professor, onde nesse teste é passado o valor para tirar média. O cálculo é feito pela soma das notas divididas pela quantidade de notas já contabilizadas para aquele determinado professor.

Classe UsuarioTests: É testado adicionar uma nova forma de pagamento.

Trabalhos Futuros

A parte do CRUD decidimos deixar para uma aplicação geral, então não foi feita nesta etapa do trabalho, mas ficará como criação futura. Assim também pensamos em fazer a parte da aplicação geral com adicionar, atualizar, deletar e fazer 100% dos testes.

6. Considerações Finais

O objetivo deste trabalho foi realizar uma solução por meio de uma aplicação que nomeamos de GetHelp para gerenciar uma plataforma de ensino a distância. A modelagem inicial foi feita com a ideia de criar um sistema que torne a conexão entre profissionais do ensino e estudantes algo mais simples e prático.

O primeiro passo do trabalho foi identificar o problema que há nos cursos de ensino a distância como por exemplo: encontrar professores qualificados com conteúdo de qualidade, a falta de plataformas que unifiquem essa busca e oferta de ensino, a impossibilidade de compra de conteúdos específicos e de curta duração, pois identificamos que muitas vezes os usuários precisam comprar cursos inteiros para ter acesso a uma parte específica do conteúdo contido nesse curso.

Para resolver o problema do ensino citado e atender aos objetivos propostos modelamos a aplicação desse trabalho que é o “Get|Help”, onde ao decorrer relatamos o projeto em pôr os conteúdos de todas as áreas de ensino em todos os níveis de educação. Sendo assim, a aplicação buscaria ter conteúdos de ensino básicos como, por exemplo, alfabetização e as aulas sobre conteúdos lecionados por doutores em instituições de ensino superior. Há a ideia de o aluno adquirir todo pacote de aulas de um curso, mas há outra alternativa que seria disponibilizar a possibilidade de obter apenas algumas aulas sobre conteúdos específicos de cursos por valores mais acessíveis.

Em resumo, a plataforma ofereceria cursos e os usuários se matricularam naqueles que lhe interessaram. O aluno também teria acesso vitalício às aulas, ou seja, poderiam estudar no seu ritmo, sem uma data estipulada para terminar o curso e/ou assistir às aulas adquiridas. Além de ser também uma fonte de renda a mais para professores de todos os níveis, e por meio da plataforma poderiam ter alunos em todo o país.

