

UNIVERSIDADE FEDERAL DO PARANÁ
ALBERTO SUSSUMU KATO JUNIOR
IVONETE ROQUE
WELYNGTON JOSÉ VINICIUS DAL PRÁ

COAVALEITOR: SISTEMA DE SUPORTE PARA AVALIAÇÃO POR PARES

CURITIBA

2016

ALBERTO SUSSUMU KATO JUNIOR
IVONETE ROQUE
WELYGTON JOSÉ VINICIUS DAL PRÁ

COAVALIEITOR: SISTEMA DE SUPORTE PARA AVALIAÇÃO POR PARES

Trabalho apresentado como requisito parcial à obtenção de grau de Tecnólogo no curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientadora: Profa. Msc. Andreia de Jesus

CURITIBA

2016

AGRADECIMENTOS

Agradecemos à nossa orientadora, Profa. Andreia de Jesus, pelos ensinamentos, dedicação e paciência para com a nossa equipe. Ao Prof. Alexander Robert Kutzke por confiar a nós o desenvolvimento desse projeto tão interessante e desafiador. A todos os professores do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná por compartilhar seu precioso conhecimento. Aos colegas do TADS, pelos momentos inesquecíveis, tanto os trágicos como os cômicos. E, por último, mas não menos importante, dedicamos aos nossos familiares pela compreensão e carinho em momentos tão difíceis.

*“Educação não transforma o mundo.
Educação muda as pessoas. Pessoas
transformam o mundo.”*

(Paulo Freire)

RESUMO

Avaliar é uma tarefa que demanda muito tempo e planejamento dos professores. A avaliação por pares consiste em um processo no qual os alunos avaliam os trabalhos realizados por outros alunos, baseando-se em métricas pré-estabelecidas pelo professor. Essa prática de avaliação ajuda o professor a economizar tempo na correção de atividades, além de melhorar as habilidades metacognitivas dos estudantes. Tendo em vista o cenário apresentado, foi desenvolvido o COAVALEITOR – Sistema de Suporte de Avaliação por Pares, uma ferramenta web que tem por objetivo facilitar a execução desse método de avaliação. Para levantamento de requisitos, foram utilizadas as seguintes técnicas: Entrevista, Prototipação de Telas e Análise de Sistemas Correlatos. Na gerência do projeto foi adotado a metodologia SCRUM. Entre as principais funcionalidades do sistema proposto estão: criação de turmas e avaliações, visualização das soluções e correções realizadas, possibilidade de dar *feedback*, gerar relatórios, entre outras. COAVALEITOR destaca-se em relação aos concorrentes por possuir o código aberto e ser desenvolvido em Português.

Palavras-chave: Avaliação por Pares; Teorias educacionais; Sistema Web.

ABSTRACT

Evaluating is a time-consuming and planning task for teachers. Peer Assessment consists of a process in which students evaluate the work done by other students, based on metrics pre-established by the instructor. This practice is employed, as it helps the teacher to save time in correcting activities, as well as improving students' metacognitive abilities. In view of the scenario presented, the COAVALIEITOR - Peer Assessment Support System was developed, a web tool that aims to facilitate this evaluation method. For requirements elicitation, the following techniques were used: Interview, Screen Prototyping and Analysis of Correlated Systems. In the project management the SCRUM methodology was adopted. Among the main functionalities of the proposed system are: creation of classes and evaluations, visualization of solutions and corrections made, possibility of giving feedback, generating reports, among others. COAVALIEITOR stands out in relation to the competitors for having an open source code and for being developed in Portuguese.

Key words: Peer Assessment, Educational theories, Web System.

LISTA DE FIGURAS

Figura 1 – “Atividades dos Professores X Horas Despendidas”	14
Figura 2 – Esquema de Avaliação pelos Pares.	30
Figura 3 – Tela inicial CrowdGrader	34
Figura 4 – Tela inicial PeergradeIO	36
Figura 5 – Funcionalidades PeergradeIO	37
Figura 6 – Tela inicial Teammates	38
Figura 7 – Fluxograma da metodologia do trabalho.	42
Figura 8 – Caminho de Teste Básico.	57
Figura 9 – Erro ao logar no Sistema.....	58
Figura 10 – Fluxograma do funcionamento do sistema Coavalleitor.	62
Figura 11 – Tela de Login.	63
Figura 12 – Tela de Recuperação de senha.	63
Figura 13 – Tela de Cadastro de Usuário.....	64
Figura 14 – Tela de Configurações da Conta.	64
Figura 15 – Tela de Listar Instituições.....	65
Figura 16 – Tela de Adicionar Instituição.	65
Figura 17 – Tela de Buscar Cursos.....	66
Figura 18 – Tela de Adicionar Curso.....	66
Figura 19 – Tela de Buscar Disciplinas.	67
Figura 20 – Tela de Editar Disciplina.....	67
Figura 21 – Tela de Adicionar Disciplina.	68
Figura 22 – Dashboard.	69
Figura 23 – Tela de Buscar Turmas.	70
Figura 24 – Tela de Editar Turma.....	70
Figura 25 – Tela de Adicionar Turma.	71
Figura 26 – Tela de Adicionar Aluno em uma Turma.	71
Figura 27 – Tela de Listar avaliações de uma turma.....	72
Figura 28 – Tela de Adicionar Avaliação.	73
Figura 29 – Tela de Listar soluções dos alunos.	74
Figura 30 – Tela de Detalhe da Solução.	74
Figura 31 – Tela de listar Correções dos Alunos.....	75
Figura 32 – Tela de Detalhe de Correção.	75

Figura 33 – Tela de Avaliar Alunos.	76
Figura 34 – Tela de Especificar nota.	76
Figura 35 – PDF alunos turma / Tela de Buscar Relatórios.	77
Figura 36 – Tela de Listar Turmas.	78
Figura 37 – Tela de Inscrição.	78
Figura 38 – Tela de Buscar Avaliações.	78
Figura 39 – Tela de Avaliações por Turma.	79
Figura 40 – Tela de Resolver Avaliação.	80
Figura 41 – Tela de Listar Soluções para Correção.	81
Figura 42 – Tela de Corrigir Avaliação.	82
Figura 43 – Tela de Buscar Notas.	83
Figura 44 – PDF Notas Aluno.	83
Figura 45 – Protótipo “Recuperar senha”.	95
Figura 46 – Protótipo “Login”.	95
Figura 47 – Protótipo “Cadastro de usuário”.	96
Figura 48 – Protótipo “Instituições”.	96
Figura 49 – Protótipo “Cadastrar curso”.	97
Figura 50 – Protótipo “Cadastrar disciplina”.	98
Figura 51 – Protótipo “Cadastrar turma”.	99
Figura 52 – Protótipo “Criar avaliação”.	100
Figura 53 – Protótipo “Listar alunos”.	101
Figura 54 – Protótipo “Listar correções de uma tarefa”.	102
Figura 55 – Protótipo “Lista turmas de um professor”.	102
Figura 56 – Protótipo “Listar estudantes de uma tarefa”.	103
Figura 57 – Protótipo “ <i>Dashboard</i> ”.	103
Figura 58 – Protótipo “Relatórios”.	104
Figura 59 – Protótipo “Listar tarefas de uma turma”.	104
Figura 60 – Protótipo “Listar turmas aluno”.	105
Figura 61 – Protótipo “Responder a uma avaliação”.	105
Figura 62 – Protótipo “Realizar correções”.	106
Figura 63 – Protótipo “Avaliações por aluno”.	107

LISTA DE TABELAS

Tabela 1 – Comparativo entre aplicativos similares ao Coavaleitor.	39
Tabela 2 – Requisitos funcionais Coavaleitor.	44
Tabela 3 – Requisitos não funcionais Coavaleitor.	46
Tabela 4 – Processos representados pelos diagramas de sequência.	49
Tabela 5 – Planilha de entregas SCRUM.	52
Tabela 6 – Caso de teste Caixa-Branca.	57
Tabela 7 – Caso de teste Caixa-Preta.	58

LISTA DE QUADROS

Quadro 1 – Exemplo de Avaliação pelos Pares.	27
Quadro 2 – Exemplo de teste Caixa-Branca.	56

SUMÁRIO

1 INTRODUÇÃO	14
1.1 PROBLEMA.....	15
1.2 OBJETIVO GERAL.....	16
1.3 OBJETIVOS ESPECÍFICOS.....	16
1.4 JUSTIFICATIVA.....	16
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1. TEORIAS EDUCACIONAIS.....	18
2.2 APRENDIZAGEM ATIVA.....	20
2.3 AVALIAÇÃO NA EDUCAÇÃO.....	22
2.3.1. AUTOAVALIAÇÃO E AVALIAÇÃO PELOS PARES.....	24
2.4. INFORMÁTICA COMO FERRAMENTA DE AUXÍLIO PARA AVALIAÇÃO.....	31
3 TRABALHOS CORRELATOS	33
4 METODOLOGIA DE TRABALHO	41
4.1 REQUISITOS.....	42
4.1.1 LEVANTAMENTO DE REQUISITOS.....	42
4.1.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA.....	43
4.2 MODELAGEM DOS DADOS DO SISTEMA.....	47
4.3 MODELAGEM DAS FUNCIONALIDADES DO SISTEMA.....	47
4.3.1 DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO.....	48
4.3.2 DIAGRAMA DE CLASSES.....	48
4.3.3 DIAGRAMA DE SEQUÊNCIA.....	49
4.3.4 DIAGRAMA DE ATIVIDADE.....	50
4.4 METODOLOGIA DE GERENCIAMENTO DO PROJETO.....	50
4.5 TESTES REALIZADOS.....	54

4.5.1 EXEMPLO DE TESTE CAIXA-BRANCA	55
4.5.2 EXEMPLO DE TESTE CAIXA-PRETA	58
4.6 TECNOLOGIAS E LINGUAGENS UTILIZADAS NO DESENVOLVIMENTO DO SISTEMA.....	59
5 APRESENTAÇÃO DO SOFTWARE.....	62
5.1 ACESSO AO SISTEMA	62
5.2 MANTER CADASTROS.....	64
5.2.1 PROFESSOR COM PERFIL DE ADMINISTRADOR	65
5.3 MÓDULO DO PROFESSOR.....	68
5.3.1 TURMA.....	69
5.3.2 AVALIAÇÃO	71
5.3.3 SOLUÇÃO	73
5.3.4 CORREÇÃO	74
5.3.5 AVALIAR ALUNO	75
5.3.6 RELATÓRIOS	77
5.4 MÓDULO DO ALUNO.....	77
5.4.1 AVALIAÇÃO	78
5.4.2 SOLUÇÃO	79
5.4.3 CORREÇÃO	80
5.4.4 NOTAS	82
6 CONSIDERAÇÕES FINAIS	84
REFERÊNCIAS	86
APÊNDICE 1 – ENTREVISTA	92
APÊNDICE 2 – PROTOTIPAÇÃO DE TELAS.....	95
APÊNDICE 3 – DIAGRAMA DE CASO DE USO	108
APÊNDICE 4 – ESPECIFICAÇÃO DE CASO DE USO	109
APÊNDICE 5 – DIAGRAMA DE CLASSE DE NEGÓCIOS	131

APÊNDICE 6 – DIAGRAMA DE CLASSE DE IMPLEMENTAÇÃO	132
APÊNDICE 7 – DIAGRAMA DE ENTIDADE RELACIONAMENTO	133
APÊNDICE 8 – DIAGRAMAS DE SEQUÊNCIA.....	134
APÊNDICE 9 – DIAGRAMA DE ATIVIDADE	145

1 INTRODUÇÃO

Com os avanços da tecnologia, muitos processos que eram realizados de forma manual, passaram a ser automatizados por meio da informática. Isso permitiu que informação organizacional se tornasse mais confiável, precisa e acessível. Não podendo esquecer do fator tempo, uma vez que sistemas informatizados proporcionam agilidade na realização de tarefas repetitivas e monótonas.

Com mais de 37 milhões de estudantes matriculados somente na rede pública de ensino básico em 2015 (CENSO ESCOLAR 2015), percebe-se a dimensão do setor da educação no nosso país. Sem a utilização da tecnologia da informação, seria muito difícil gerenciar de forma eficaz uma instituição de ensino. Uma vez que se torna necessário controlar e armazenar as matrículas dos alunos, notas, frequências, os índices de evasão e aprovação de forma quase que automática.

Outro aspecto importante para enfatizar é o processo de correção de provas, que segundo estudo realizado pela OECD - Organização para a Cooperação e Desenvolvimento Econômico em 2014, cada professor gasta em média cerca de 5 horas por semana na tarefa de corrigir provas (Figura 1). Tempo que poderia ser melhor empregado em sala de aula.

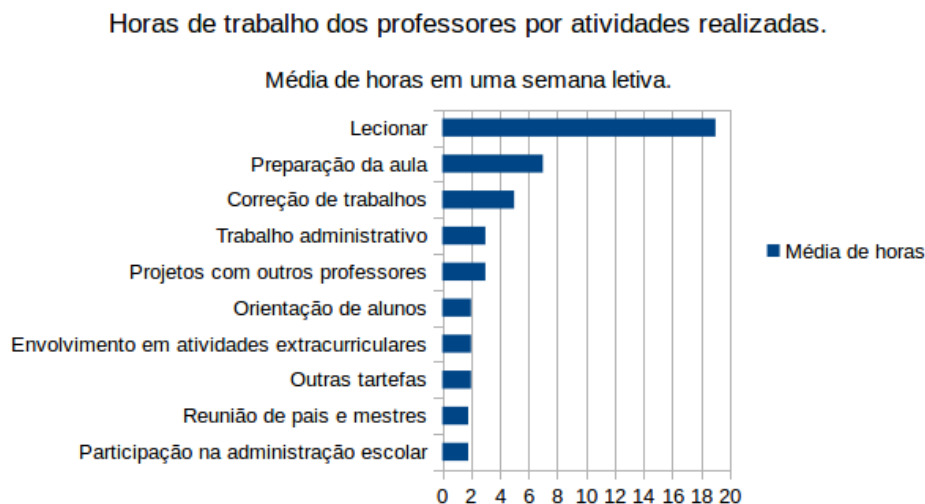


Figura 1 – “Atividades dos Professores X Horas Despendidas”

FONTE: Adaptado de OECD (2014).

O **Peer assessment**, literalmente traduzido para **avaliação pelos pares**, é uma metodologia de ensino na qual estudantes comentam e julgam os trabalhos e provas realizados pelos seus colegas, seguindo métricas preestabelecidas pelo professor, ou seja, trata-se de um processo de coavaliação entre os estudantes. Tal prática é empregada para otimizar o processo de correção das avaliações, poupar o tempo gasto pelo professor, ajudar o aluno a compreender melhor o conteúdo lecionado em sala de aula e melhorar sua habilidade metacognitiva (SANDLER e GOOD, 2006).

É nesse contexto de informatização e automatização de processos voltados para o mundo educacional, que o Prof. Dr. Alexander Robert Kutze do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) do Setor de Educação Profissional e Tecnológica (SEPT) da Universidade Federal do Paraná (UFPR), idealizou um *software* para auxiliar na execução da metodologia *Peer assessment* de forma automatizada, uma vez que o processo manual é muito custoso e trabalhoso, o que acaba por desmotivar muitos professores a utilizar tal metodologia.

Portanto, esse trabalho de conclusão de curso tem como objetivo sistematizar e executar os passos necessários para a construção desse *software* gratuito que automatiza a metodologia *Peer Assessment*. É abordado desde o processo inicial de levantamento de requisitos até a entrega final do produto.

1.1 PROBLEMA

Como já mencionado na introdução, a execução manual da metodologia **Peer assessment** é *bastante custosa* e trabalhosa, o que acaba por desmotivar muitos professores a utilizar essa metodologia.

As correções das atividades, trabalhos e provas é realizada pelos próprios alunos através de papel e caneta. Isso acaba por dificultar a supervisão e avaliação realizada pelo professor. Além disso, é um processo muito lento e pouco eficiente, pois sem a informatização e automação do processamento das correções é possível que se tenha uma perda significativa de acurácia e confiança na metodologia aplicada.

Outro problema detectado foi a dificuldade de encontrar *softwares* que auxiliam na execução da metodologia **Peer Assessment** no idioma Português, de código aberto e 100% gratuito. Adiciona-se a isso questões culturais, pois não é muito comum os alunos participarem de forma efetiva do processo de correções das avaliações (SANTOS, 2002). Já nos EUA, existe até que um número considerável de *softwares*, para instituições de ensino, que auxiliam nos processos de coavaliação entre os estudantes, contudo em sua grande maioria essas ferramentas são pagas.

Logo, o objetivo é disponibilizar um sistema baseado em software livre que seja gratuito, em português e que tenha requisitos que corroborem com a metodologia **Peer Assessment**.

1.2 OBJETIVO GERAL

O objetivo geral deste trabalho é criar uma sistema WEB que dê suporte ao processo de avaliação *Peer Assessment* - avaliação pelos pares.

1.3 OBJETIVOS ESPECÍFICOS

- Levantar requisitos junto a um profissional da educação – professor.
- Relacionar educação e tecnologia, a fim de compreender o contexto em que o *software* proposto será utilizado.
- Definir usuários e suas respectivas ações no sistema.
- Automatizar o processo de avaliação de atividades/tarefas pelos estudantes, com base na metodologia **Peer Assessment**.
- Criar um módulo de estatística das atividades realizadas no sistema, a fim de facilitar a análise do professor.

1.4 JUSTIFICATIVA

Considerando a complexidade das atividades realizadas no contexto educacional e a facilidade que as tecnologias proporcionam no que diz respeito a automatização, sistematização e simulação de processos, Alcântara coloca que:

...diante do avanço das novas tecnologias, o professor tem como auxílio um novo recurso que torna suas aulas mais estimulantes e diferenciadas. Esta é

uma forma de mostrar que o aluno pode sim obter um bom desempenho perante as máquinas, com softwares educacionais que enriquece sua melhor maneira de crescer.(ALCÂNTARA, 2012, p.1)

Essa citação justifica a importância de se investir na criação de *softwares* voltados para educação. O objetivo do *software* nunca será substituir o professor, mas sim servir como instrumento pedagógico. É exatamente essa proposta que o Prof. Dr. Alexander Robert Kutzke estabeleceu, a criação de uma ferramenta que possa auxiliar suas aulas, e como consequência melhorar o aprendizado dos seus estudantes.

Para Sadler e Good (2006), a metodologia de *Peer Assessment* proporciona vantagens tanto para alunos como para professores, como por exemplo:

- Logística: Uma vez que as correções das atividades de uma sala inteira podem ser realizadas de forma simultânea. O que acaba por proporcionar tempo extra para o professor aprofundar o conteúdo.
- Pedagógica: Quando um aluno avalia a resposta de outro estudante, o mesmo acaba por aprimorar seu ponto de vista e aprofundar seu conhecimento em um determinado tópico.
- Metacognição: Desmistificação das provas. Estudantes se tornam mais conscientes dos seus pontos fortes e fracos. Criação do senso de auto avaliação.
- Afetiva: Mudanças afetivas podem tornar os alunos mais cooperativos e produtivos

Logo, com base no que foi exposto. É justificável a criação de um sistema para automatizar a metodologia de ensino *Peer Assessment*. Uma vez que existem várias vantagens, tanto para alunos como para os professores. É importante ressaltar que essa ferramenta será gratuita, de código aberto e disponível para todos aqueles que queiram aplicar essa metodologia em suas aulas e desejem utilizar o sistema como ferramenta de auxílio.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo abordar de forma sucinta os conceitos que embasaram o desenvolvimento desse projeto, que são: metodologias educacionais; aprendizagem ativa; avaliação na educação; avaliação pelos pares.

2.1. TEORIAS EDUCACIONAIS

As diversas teorias sobre educação tem relação com estudos sobre a teoria do conhecimento, psicologia cognitiva, sociologia e outras influências. O termo pedagogia deriva do grego *παιδαγωγία* (*paidagōgia*) e era utilizado para denominar o escravo que acompanhava as crianças até a escola (Pedagogue." Online Etymology Dictionary, 2016). A pedagogia é a ciência da teoria e prática da educação.

Na idade média, a educação era regida principalmente pela igreja, que fundou as universidades, foi um período de muita censura e rigor no ensino, o qual foi encerrado pelas reformas protestantes que separam a igreja do ensino e aprendizagem. Esse período influencia o ensino até os nossos dias. É dessa época a visão do ensino autoritário, em que o aluno somente ouve e não participa do processo de aprendizagem (VAIDERGORN, 2008). Mais tarde esse tipo de pensamento seria questionado por diversos pesquisadores da pedagogia, entre eles Paulo Freire que defendia um ensino democrático, crítico e libertário (DULLO, 2014).

No século XVIII, a epistemologia, o estudo do conhecimento, trouxe novas discussões para a educação, principalmente influenciado pelo filósofo alemão Immanuel Kant. Este pensador foi uma grande referência para os cientistas do século XX, sobretudo os pesquisadores da teoria do conhecimento, educação e sociologia. Nesta mesma época a doutrina dominante de ensino era a de condicionamento, pois acreditava-se que o aprendizado e o comportamento eram respostas a estímulos.

Esse pensamento foi a base da teoria behaviorista cuja teoria determinava que o comportamento é definido a partir de respostas a estímulos (WATSON & JOHN, 2013). O termo vem do inglês *behavior* e pode ser entendido como o estudo do comportamento. Pavlov, filósofo russo, é um dos mais notáveis nomes desta ciência, para ele a psicologia humana baseia-se no comportamento, e este seria consequência de um reflexo condicionado (PAVLOV, 1913).

A teoria behaviorista foi superada pela verificação de que o comportamento, aprendizagem e desenvolvimento humano são amplamente influenciados pela sociedade, interações e o meio. Uma das correntes de pensamento mais críticas do behaviorismo foi o construtivismo. (PIAGET, 1970).

A metodologia que ainda é a mais influente atualmente é conhecida como Pedagogia Tradicional, cuja influência ainda remete a antiga e autoritária educação religiosa. Nessa concepção o professor é maior autoridade, e a ele cabe passar o conhecimento. Dessa forma sua tarefa se limita a transmitir o conhecimento sem levar em conta a individualidade de cada aluno. Já aos alunos cabe a tarefa de "escutar e memorizar" o conteúdo passado pelo professor, não havendo participação (LIBANÊO 1985). Essa metodologia, embora amplamente adotada, é muito criticada devido as suas limitações, sobretudo pelos teóricos do construtivismo, da pedagogia da libertação, entre outros.

O construtivismo baseia-se na ideia de que o conhecimento deve ser desenvolvido pelo próprio educando (PIAGET, 1970), ou seja, ao invés do aluno adotar um comportamento passivo em relação ao processo de aprendizado ele tem sua própria interpretação de mundo e é ativo no processo educacional. Jean Piaget é considerado o criador da teoria do construtivismo social, conhecida apenas por construtivismo. Para ele as pessoas aprendem de forma ativa influenciadas pelo meio onde vivem (PIAGET, GRUBER, & VONECHE, 1977).

Vygostsky foi outro pesquisador considerado construtivista apesar de ser anterior a Piaget, e que trouxe importantes contribuições para a educação. Para ele a aprendizagem é um processo social (YASNITSKY, 2011). O aluno nunca vai aprender completamente sozinho, mas sim de forma interativa e coletiva. Assim como a nossa própria linguagem é adquirida socialmente, todo o aprendizado também é obtido relacionando-se com a cultura, e o meio do aprendente desde o nascimento (YASNITSKY, 2011).

Paulo Freire ainda reforça que a autonomia do educando é fundamental no processo de ensino, a liberdade e o conhecimento do aluno devem ser respeitados. Em oposição ao behaviorismo, citado anteriormente, Freire afirma que "Formar é muito mais do que puramente treinar o educando no desempenho de destrezas" (FREIRE 1996, p.15). Assim como o respeito ao estudante Freire defende a democracia na sala

de aula baseada no diálogo e respeito entre todos os envolvidos no processo e a tomada de decisão de forma horizontal. (FREIRE, 1981). Ou seja, para Freire professor e aluno, diferentemente do que ocorria na pedagogia tradicional destaca:

... Não há docência sem discência, as duas se explicam. e seus sujeitos, apesar das diferenças que os conotam, não se reduzem à condição de objeto um do outro. Quem ensina aprende ao ensinar, e quem aprende ensina ao aprender. (FREIRE., 1996, p.25)

Enfim, foram utilizadas diversas referências teóricas para este trabalho, partindo da teoria construtivista de Piaget que considera a contextualização do ensino de acordo com cada estudante e o respeito a sua individualidade, seu desenvolvimento e evolução (PIAGET, 1970),

Também foram consideradas as contribuições de Vygostsky sobre as interações sociais no desenvolvimento humano, pois não é possível aprender de forma isolada ou sozinho, estamos sempre em interação com o meio, com a sociedade, o próprio processo de aprendizagem é permeado por diversas relações, seja entre professores e alunos, ou dos próprios alunos com seus pares (YASNITSKY, 2011)

Ainda foi adotada como referência a teoria da pedagogia libertária de Paulo Freire que leva em conta as relações de poder na sala de aula e critica o autoritarismo e a opressão na figura do professor. Em contrapartida Freire incentiva a democracia em sala de aula, a educação participativa e a horizontalidade nas discussões (FREIRE, 1996).

Portanto a teoria construtivista de Jean Piaget baseada na adaptação do aprendente ao meio em que está inserido, as observações de Vygostsky sobre as interações sociais no desenvolvimento humano e a crítica de Paulo Freire a democratização do ensino e autonomia do estudante foram as referências teóricas utilizadas para embasar o desenvolvimento deste trabalho, auxiliando na definição e fundamentação dos requisitos do sistema proposto.

2.2 APRENDIZAGEM ATIVA

Uma metodologia de ensino baseada na teoria construtivista é o aprendizado ativo que busca envolver mais diretamente o aluno no processo de ensino, baseia-se

no princípio da participação ativa do aluno ao invés de simplesmente ouvir. (BARNES, 1989). Além de assistir a aulas tradicionais, cabe ao aluno dessa metodologia interpretar, analisar, sintetizar, classificar, relacionar e comparar, trabalhando em conjunto com seus pares.

Barnes (1989) sugere os princípios da aprendizagem ativa:

- Propositiva: A relevância da tarefa com o que importa ao aluno.
- Reflexiva: A reflexão do aluno com o significado do que ele aprende.
- Negociada: Negociação dos objetivos e métodos entre professores e estudantes.
- Crítica: Estudantes apreciam diferentes modos de aprendizado.
- Complexa: Estudantes comparam as tarefas com as complexidades da vida real realizando análise reflexiva.
- Dirigida-a-situação: A necessidade da situação ser considerada para estabelecer tarefas.
- Engajada: Tarefas da vida real refletem as atividades conduzidas no aprendizado.

Ainda, segundo Barnes (1989) a aprendizagem ativa exige um ambiente de aprendizado apropriado, baseado na estratégia construtivista, promovendo pesquisa baseada em investigação, encorajamento de habilidades de liderança nos alunos e atividades autodesenvolvidas, criação de atmosfera colaborativa, cultivo de um ambiente dinâmico e interdisciplinar. A aprendizagem ativa também considera os princípios do construtivismo já que o conhecimento sempre é construído a partir do que o estudante já sabe, como um processo contínuo.

Alguns exemplos de metodologias de aprendizagem ativa são: debates; grupos de discussões; jogos em sala de aula; exercícios rápidos com *feedback*, podendo ser auxiliado por meio de tecnologias como vídeos, slides ou mesmo *softwares* educacionais; atividades em que os alunos ensinam e aprendem entre si.

A autoavaliação interativa é uma forma de aprendizagem ativa, muito utilizada em cursos à distância, permite que o aluno avalie seu próprio desempenho. Existem diversas técnicas de autoavaliação, desde respostas a questionários de múltipla escolha até jogos eletrônicos

Já a avaliação pelos pares, em que os colegas avaliam uns aos outros, também utiliza conceitos da metodologia ativa como a interatividade e o trabalho em grupo, possibilitando reflexão sobre o que é aprendido para assim poder avaliar seus colegas. Ainda há mais possibilidades de explorar a aprendizagem ativa em conjunto com a avaliação pelos pares utilizando os princípios definidos por Barnes (1989) como a negociação envolvida no processo de correção e feedback e o aumento do engajamento do aluno ao envolvê-lo na avaliação - atividade em que ele não faz parte ativa normalmente.

Logo, o sistema proposto neste trabalho, denominado *Coavaliador*, é uma ferramenta criada com base na metodologia de aprendizagem ativa, permitindo o aluno a se autoavaliar ou avaliar seus pares. Possibilitando, assim, o fornecimento de *feedback* e interações.

2.3 AVALIAÇÃO NA EDUCAÇÃO

A avaliação, a despeito de diversos debates em torno de suas formas, é parte essencial do processo de aprendizagem. A partir desta pode-se quantificar os resultados do aprendizado e inclusive direcionar os objetivos do ensino.

Segundo Luckeski (2005), apesar de forte influência de heranças autoritárias do Estado e religião, na cultura em que o exame é utilizado para gerar medo e castigo psicológico, este também pode ser utilizado de forma libertadora e democrática atenta ao estabelecimento da autonomia do educando. Ou seja:

... propomos que a avaliação do aproveitamento escolar seja praticada como uma atribuição de qualidade aos resultados da aprendizagem dos educandos, tendo por base seus aspectos essenciais e, como objetivo final, uma tomada de decisão que direcione o aprendizado e, conseqüentemente, o desenvolvimento do educando (CIPRIANO, 2005, p. 95)

Ainda segundo Luckeski (2005), o erro e o insucesso são partes do caminho necessário para o aprendizado, mas sobre eles não deve se acrescentar culpa ou castigo, mas sim deles devem ser retiradas lições e devem ser considerados como obstáculos para a evolução e a superação no aprendizado. A avaliação nesse sentido pode ser utilizada como:

- a. Medida do aproveitamento escolar.

- b. Transformação da medida em nota ou conceito.
- c. Utilização dos resultados identificados.

Visto que a avaliação tem também um caráter pedagógico e que pode ser instrumento de formação, são diversas as possibilidades de uso da avaliação como meio de motivação e de complementação do ensino. E, considerando esse caráter pedagógico, pode-se colocar que a avaliação da aprendizagem tem por objetivo auxiliar o educando no seu crescimento e desenvolvimento, e assim, na sua integração consigo mesmo, ajudando-o na apropriação dos conteúdos significativos do aprendizado (conhecimento, habilidades, hábitos, convicções) (LUCKESKI, 2005)

Dentro da pedagogia são possibilidades de uso da avaliação segundo Luckeski (2005):

- Coletar, analisar e sintetizar as condutas, produzindo uma configuração do efetivamente aprendido.
- Atribuir uma qualidade a essa configuração da aprendizagem .
- Definir as condutas docentes e discentes a serem seguidas, partir da qualificação.

Também Luckeski (2005) define três funções para a avaliação que podem ser resumidas como:

a) Função de propiciar a autocompreensão tanto do educando quanto do educador, por meio de atos de avaliação.

b) Função de motivar o crescimento, na medida que ocorre o reconhecimento do limite e da amplitude de onde se está, existe a motivação para prosseguir o percurso de crescimento. A avaliação motiva na medida em que se diagnostica e cria o desejo de obter resultados mais satisfatórios.

c) Função de aprofundamento. Quando se faz um exercício para que a aprendizagem seja manifestada. Esse mesmo exercício já é uma oportunidade de aprender o conteúdo de uma forma mais aprofundada, de fixá-lo de forma mais adequada, de aplicá-lo, etc. Portanto o próprio exercício da avaliação apresenta-se como uma oportunidade de aprendizado.

Devemos considerar também que a avaliação consiste na retirada de amostras de conhecimento do avaliado, estimando um valor para essa estimativa. Essas amostras são embasadas em conquistas pessoais, potencial, atitudes, inteligência, motivações. Todas as formas de avaliação fornecem uma estimativa do atual status do avaliado.

Segundo Mogey & Watt (1996) antes de se avaliar alguém deve-se considerar o contexto e objetivos da avaliação. Esta deve ser baseada em critérios e padrões que permitam ranquear, medir e analisar o desenvolvimento do aluno ao longo de um curso. Considerando esses critérios, é possível definir três tipos de avaliação:

- a) Diagnóstica: para identificar forças e fraquezas;
- b) Formativa: para prover *feedback* aos estudantes;
- c) Somativa: para estimar formalmente a performance ao fim de um curso ou de uma unidade de estudo.

A partir dessa visão de que a avaliação pode ser proveitosa, sendo inclusive parte essencial do processo de ensino, partimos da premissa de que uma ferramenta para auxiliar o processo de autoavaliação de alunos pode estimular uma prática mais frequente da aprendizagem ativa, que é parte da metodologia construtivista de educação.

2.3.1. AUTOAVALIAÇÃO E AVALIAÇÃO PELOS PARES

Uma metodologia que pode ser considerada como forma de aprendizado ativo é a autoavaliação e avaliação pelos pares, tema central deste trabalho. Neste método várias habilidades da aprendizagem ativa podem ser desenvolvidas, como a reflexão, crítica, engajamento, etc. Pois, tanto na autoavaliação, como na avaliação pelos pares, o estudante deve adotar um senso de responsabilidade por seu próprio aprendizado e de seus colegas, atuando de forma protagonista no processo de ensino (BARNES, 1989).

Dentro da sala de aula cada indivíduo tem características únicas e diversas habilidades e potenciais que podem ser descobertos ou estimulados. Para isso, é importante que haja espaço no modelo pedagógico utilizado de forma a dar mais liberdade a cada estudante.

O educando que exercita sua liberdade ficará tão mais livre quanto mais eticamente vá assumindo a responsabilidade de suas ações. (FREIRE, 1996, p.91)

Diversas metodologias educacionais modernas mostram como o conhecimento pode ser construído em conjunto com o aluno e que a participação deste no processo de aprendizagem é fundamental, inclusive na fase da avaliação.

Nesse contexto iremos considerar o método de avaliação pelos pares e a autoavaliação como um meio de aumentar a motivação e engajamento dos estudantes na fase de avaliação da aprendizagem. Esse método ainda permite incrementar a eficiência do processo de avaliação.

Os educandos fazem neste modelo participação ativa no processo de aprendizagem, podendo ser auto avaliadores do que aprenderam. Assim, ao participarem, surgem necessidades tais como: a de precisar entender o conteúdo para poder avaliar o colega, além de auxiliar o estudante a enxergar seus próprios erros ao se comparar com as respostas de outros estudantes (BARREIRO & NASCIMENTO, 2000).

O professor ao utilizar essa metodologia passaria a ser não apenas uma figura de autoridade que ensina e avalia os estudantes, mas um orientador que os auxilia em suas autocorreções fornecendo apoio e *feedback*, libertando-o também de uma atividade repetitiva para que possa dedicar mais tempo a análise do aprendizado e ao tratamento personalizado dado a cada indivíduo da sua turma.

Ao se utilizar desse modelo a privacidade do aluno deve ser considerada, visto que pode haver constrangimento ao se revelar as identidades. O anonimato também contribui para que a avaliação seja imparcial. Como colocava Paulo Freire:

O bom clima pedagógico democrático é o em que o educando vai aprendendo à custa de sua prática mesma que sua curiosidade como sua liberdade deve estar sujeita a limites, mas em permanente exercício. Limites eticamente assumidos por ele. Minha curiosidade não tem o direito de invadir a privacidade do outro e expô-lo aos demais.(FREIRE, 1996, p. 85)

As habilidades desenvolvidas por meio da autoavaliação e da avaliação pelos pares auxilia os estudantes a traçar objetivos e aprender por conta própria além de

aumentar a eficiência do processo de avaliação, trazendo a figura do professor como um conselheiro que pode fornecer *feedbacks* ao invés de apenas um corretor de provas. Além disso, conforme coloca Sadler (1989), há indícios de que estudantes se tornam mais cooperativos, produtivos e apresentam atitude positiva frente a uma avaliação por *feedback*, ao invés de ser avaliados simplesmente por nota (SADLER, 1989).

Diversas vantagens foram observadas no ensino universitário ao se substituir a visão de professor como ensinante e aluno como aprendiz, para uma visão de professor como gestor do ensino e o aluno como auto avaliador, e de seus pares, do seu próprio aprendizado (HANRAHAN & ISAACS, 2001).

Um exemplo de utilização de avaliação pelos pares é realizada em revistas científicas. Os artigos submetidos são revisados pelos próprios colegas pesquisadores de forma anônima. Essa técnica é conhecida como *peer review* e é adotada por revistas renomadas como a *Nature*. Em pesquisas realizadas com cientistas a maioria avalia a revisão cega pelos pares como a mais efetiva (MULLIGAN, *et al.*, 2009).

Este tipo de avaliação, pelos pares, se baseia na verificação da qualidade de seu trabalho pelos seus próprios colegas. Para que funcione é preciso um claro entendimento dos papéis de cada avaliador e um treinamento prévio. Uma forma de facilitar a implantação do método é realizar uma sessão de treinamento, explicar os critérios de avaliação e como pode ser realizada a correção. Para facilitar podem ser usadas listas com os critérios de avaliação ou *checklists* para orientar a correção e também exemplos de acertos e erros.

Para atingir os objetivos da avaliação pelos pares é necessário que os estudantes se sintam confiantes e confortáveis uns com os outros para se conseguir um *feedback* honesto e construtivo. Uma forma de alcançar essa confiança é utilizar a avaliação pelos pares em conjunto com atividades em grupo, deixando os alunos mais confortáveis uns com os outros. (ZUNDERT & MERRIÊNBOER, 2010) dão algumas indicações de como utilizar a avaliação pelos pares:

- Identificar atividades nas quais os estudantes podem tirar proveito do *feedback* dos pares;

- Considerar a quebra de uma grande avaliação em várias menores, por exemplo, planejamento, primeira entrega, segunda entrega, etc;
- Definir critérios e guias de avaliação para orientar os estudantes;
- Explicar a utilização dos guias de avaliação em exemplos de exercícios durante o ensino;
- Ajudar os estudantes a corrigir de forma crítica, construtiva e a fornecer *feedbacks* descritivos e objetivos;

Já a autoavaliação, denominada também de *selfassessment*, é a avaliação do aluno por ele mesmo. Como este não pode guiar seu próprio caminho sozinho, geralmente este tipo de avaliação é auxiliada por um *software* educacional, que segue as orientações do tutor. Autoavaliação é baseada em três pilares (ZUNDERT & MERRIËNBOER, 2010):

1. Objetivos: parte essencial da autoavaliação, o objetivo será sempre confrontado pelo estudante com o seu desempenho e seu aprendizado.
2. Ferramentas de apoio: Para guiar e orientar os alunos é necessária uma estratégia de ensino. Existem diversas técnicas de autoaprendizado e autoavaliação que podem ser utilizadas e as ferramentas são fundamentais para o sucesso desse método.
3. Portfólio: O portfólio são conquistas que o estudante obtém ao seguir com o seu aprendizado em direção ao objetivo. O portfólio descreve uma coleção de realizações do estudante que contam uma história do seu esforço em direção ao objetivo.

Existe a possibilidade inclusive de combinar as duas estratégias de avaliação, por exemplo, estudantes que se autoavaliaram podem compartilhar as suas respostas com seus pares e receber um *feedback* do orientador da sua autocorreção e da correção dos pares. O Quadro 1 demonstra um exemplo de aplicação do conceito de avaliação pelos pares com *feedback* do professor para uma atividade de lógica de programação.

Quadro 1 – Exemplo de Avaliação pelos Pares.

FONTE: OS AUTORES (2016).

Exercício Fatorial (Valor 10 pontos)

Um número fatorial é definido por

* Valor 1 se $n = 0$ ou $n = 1$

* O produto de todos os números naturais de n até 1 para qualquer $n > 1$.

Indicamos o fatorial de n por $n!$

Faça um algoritmo na linguagem C que calcula o valor de $n!$

Possíveis Soluções

```
a)  int fat( int n) {
      if ((n == 0 ) || (n == 1) )
          return 1;
      return fat ( n - 1) * n;
  }
```

```
b)  int fat (int n) {
      int fat = 1;
      if n <= 1
          return fat;
      for (n;n>1;n--)
          fat:=fat*n;
      return fat;
  }
```

Critérios de avaliação do exercício "Fatorial" (Guia de correção):

1 - Algoritmo funcional - ou seja, se fosse compilado o código retornaria o valor de $n!$ corretamente. (5 pontos)

2 - Elegância - o código está indentado, não utiliza variáveis nem códigos desnecessários. (3 pontos)

3 - Sintaxe incorreta - o algoritmo tem erros de sintaxe como falta de "{" ou ";" . (-1 ponto por erro)

4 - Bônus - o algoritmo utiliza recursão para calcular o resultado. (2 pontos)

Exemplos de correção

Possível correção para a resposta a)

Re: a solução está correta, funcional, elegante, parabéns!

Possível correção para a resposta b)

Re: a solução está correta e funcional, no entanto o código também poderia ser otimizado para utilizar recursão.

Possível feedback para a resposta b)

Re: Caro Aluno, a solução está funcional, mas segue uma dica: o primeiro IF poderia ser descartado já que o FOR da linha 5 já basta para calcular o fatorial para $n=1$ ou $n=0$

Dessa forma a avaliação pelos pares pode ser esquematizada conforme a Figura 2.

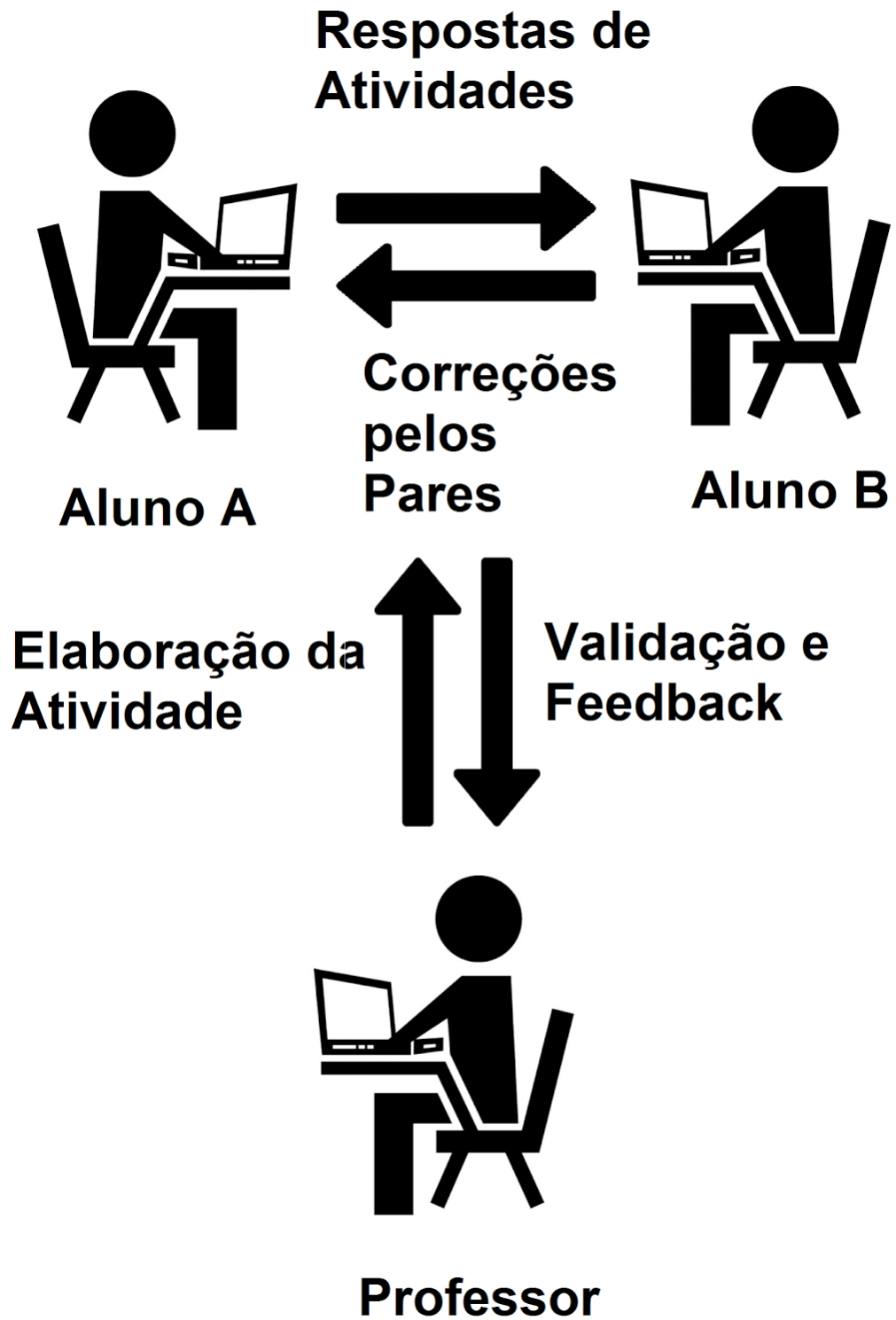


Figura 2 – Esquema de Avaliação pelos Pares.

FONTE: OS AUTORES (2016).

2.4. INFORMÁTICA COMO FERRAMENTA DE AUXÍLIO PARA AVALIAÇÃO

O uso da tecnologia na avaliação pode ter diversos propósitos, desde o gerenciamento da informação sobre os exames com auxílio de *Business Intelligence*¹ até um sistema completamente automatizado de avaliação. Os benefícios ao se utilizar o computador como ferramenta auxiliar na avaliação do ensino são inúmeros como redução de custos, aumento da produtividade e eficiência, aumento da agilidade nas correções e *feedback* do instrutor, entre outras (HENSLEY, 2015).

Conforme coloca Mogey & Watt (1996), a estratégia de avaliação utilizada *deve ser considerada na criação de um curso e as seguintes questões devem ser consideradas:*

- Que tipo de coisas queremos que os estudantes aprendam?
- Que oportunidades serão fornecidas?
- Quais tarefas avaliativas serão cobradas?
- Que métodos de avaliação serão utilizados?

Com o auxílio de diversas ferramenta simples de operar como *MS Excel*², *MS Access*³, *MATLAB*⁴, é possível incluir os dados de respostas de testes e retirar informações relevantes sobre desvio padrão, média, análise de regressão, etc. Ainda é possível realizar a análise dessa informação com muito mais facilidade do que utilizando o processo manual de avaliação. Processadores de texto, softwares de apresentação de slides, processadores de planilhas eletrônicas, podem ser utilizados como ferramentas de auxílio no ensino e aprendizado. (HENSLEY, 2015)

A segurança deve ser considerada, visto que as respostas, conceitos e *feedbacks* fornecidos são informações privadas. O acesso a esses dados deve ser protegido com o uso de senha. Outra situação que envolve a confiabilidade de testes eletrônicos é a possibilidade de plágio nas respostas, nesse quesito o teste eletrônico não difere do risco do teste manual, porém no teste eletrônico podem haver diversas

¹ Business Intelligence ou BI: em tradução direta: inteligência de negócio é um conjunto de técnicas e ferramentas voltados a extração e transformação de dados brutos de forma automatizada para suporte da análise e auxílio a tomada de decisão.

² MS Excel é uma ferramentas de planilhas eletrônicas da Microsoft

³ MS Access é um banco de dados da Microsoft que conta com interface intuitiva

⁴ MATLAB é um software da MathWorks com ampla possibilidade de utilização sobretudo na área de matemática.

ferramentas que aumentem a segurança de testes como comparação de respostas, geração aleatória de respostas para testes.

Outra vantagem da utilização do computador como ferramenta de auxílio na avaliação é a integração dos sistemas de avaliação com os sistemas escolares, que gerenciam o cadastro e desenvolvimento do estudante. Além disso, há vantagens óbvias do uso de testes computadorizados, incluindo aumento da motivação dos estudantes, incremento da precisão na obtenção de dados, melhor possibilidade de combinação para populações especiais (HENSLEY, 2015).

O computador também permite utilizar funções de acessibilidade tais como, conversores de texto para voz e voz para texto, legendas, etc., ajudando alunos que tenham necessidades especiais e assim possibilitar condições mais justas em testes competitivos. Ainda o computador permite incluir ferramentas auxiliares como calculadoras, dicionários, validação de vocabulário, tradutores.

Ao contrário dos testes baseados em papel, os testes baseados em computador dão a possibilidade de o instrutor criar um *feedback* para os exames. Ferramentas simples e intuitivas como o *Google Forms* podem ser utilizadas para a criação de formulários, testes, questionários, repositório de trabalhos, etc. Com maior velocidade na avaliação, coleta de dados de testes e correção, sobra mais tempo para o professor tomar decisões de ensino que corrijam problemas no processo de aprendizado.

3 TRABALHOS CORRELATOS

Existem alguns softwares que abordam essa metodologia de *Peer Assessment*. A maior dificuldade foi encontrar softwares relacionados ao assunto na língua portuguesa, pois aqui no Brasil não utilizamos com frequência este tipo de metodologia de avaliação, e nós acreditamos que esta metodologia de ensino será muito promissora em nosso país. (BROOKE e ANDRADE, 2013). Porém, para a análise comparativa das características, encontramos *softwares* relacionados ao assunto em sites fora do país, todos em inglês. Na América do Norte esta forma de avaliação é bem comum no ensino de um modo geral (SANDLER e GOOD, 2006), é uma questão de cultura, diferentemente do que acontece aqui no Brasil.

De todos os softwares que encontramos baseados na metodologia de *Peer Assessment*, nenhum deles disponibiliza o código fonte aberto para realização de estudos e pesquisas. Apenas um dos exemplos encontrados é 100% gratuito, os demais são todos pagos com uma versão *trial* para testes.

Nesta seção serão apresentados sistemas considerados concorrentes, por possuírem os mesmos objetivos do software desenvolvido neste trabalho.

1) **Crowdgrader**

Ferramenta estadunidense, utilizada em várias universidades e em diferentes localidades, com a fundamentação de ensino em pares. Inteiramente em Inglês e gratuita, nos módulos básicos, se necessário um melhor desempenho da ferramenta acessos específicos, deve realizar a compra do software mais completo.

A segurança das informações dos usuários nesse sistema é questionável, pois o *login* é feito por intermédio do *Gmail* (e-mail do *Google*). Nessa ferramenta *online*, os professores conseguem criar uma avaliação, os estudantes submetem suas soluções e avaliam os trabalhos realizados pelos colegas, e por fim o professor decide a nota final de cada aluno. A Figura 3 apresenta a tela inicial do sistema.

albertokatojr@gmail.com

crowdgrader

MANAGE ▾ HELP ▾ WHAT'S NEW

Your Recent Assignments

ALL ASSIGNMENTS
 CALENDAR
 DOWNLOAD COURSE GRADES
 CREATE ASSIGNMENT

Assignment	Current phase	Your role	Submissions	Reviews	Grades
sasasas	Accepting review feedback	manager	Submissions	Reviews	Grades
Copy of sasasas	Accepting review feedback	manager	Submissions	Reviews	Grades
Copy of teste	Accepting review feedback	manager, student	Submissions	Reviews	Grades
teste	Accepting review feedback	manager, student	Submissions	Reviews	Grades
Demo Assignment	Reviews available	manager	Submissions	Reviews	Grades
Demo Assignment	Reviews available	manager	Submissions	Reviews	Grades
Demo Assignment	Reviews available	manager	Submissions	Reviews	Grades

COPYRIGHT © 2016 CROWDGRADER LLC -- BLOG -- TERMS OF USE -- PRIVACY POLICY
 POWERED BY WEB2PY

Figura 3 – Tela inicial CrowdGrader

FONTE: CROWDGRADER(2016)

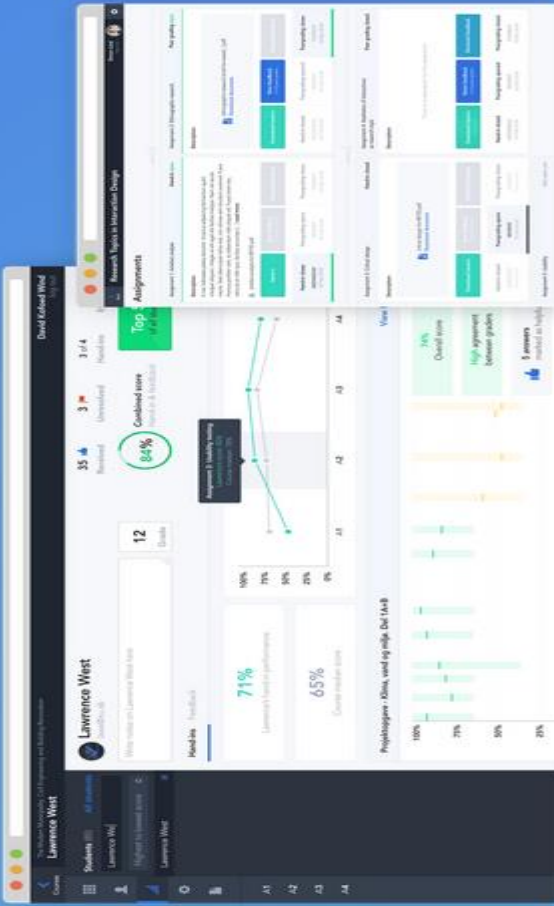
2) PEERGRADE.IO: Segue a mesma ideia da avaliação em pares. Relação entre aluno e professor e mostra o desempenho dos alunos em determinada matéria. Já na tela inicial ele possui uma apresentação geral do *software* de forma muito simples e moderna, como se fosse um manual para o usuário; mostra quais os criadores do software; disponibiliza um calendário que serve como agenda, podendo armazenar nome e contatos; pode ser criado um grupo de contatos; e, em casos de dúvidas em relação ao sistema, é possível enviar email para o suporte. Trata-se de uma ferramenta multiplataforma, funcionando perfeitamente em *desktop*, *tablets* e celulares, por exemplo. Porém, esta versão é inteiramente em inglês; e é disponibilizado, inicialmente, apenas uma versão trial com o básico, que serve mais para conhecimento da ferramenta. Caso necessitar de acessos adicionais é necessário realizar compra da licença da ferramenta. As Figuras 4 e 5 apresentam a tela de acesso desse sistema e algumas funcionalidades.

peergrade Features Pricing About Jobs Contact Help Blog

Book a demo Login

Empowering teachers to create better students

Get started ... It's free!



The screenshot displays the Peergrade dashboard for a user named Lawrence West. The interface includes a navigation menu with 'Features', 'Pricing', 'About', 'Jobs', 'Contact', 'Help', and 'Blog'. A 'Book a demo' button and a 'Login' button are positioned in the top right. The main dashboard area shows a 'Students' section with 12 students, a 'Combined score' of 86%, and a 'Top Assignments' table. A line graph shows 'Learning Quality Score' trends, and a bar chart displays 'Psychographics' data. The 'Top Assignments' table lists various assignments with their respective scores and completion rates.

Assignment	Score	Completion Rate
Assignment 1	85%	100%
Assignment 2	78%	95%
Assignment 3	72%	90%
Assignment 4	68%	85%
Assignment 5	62%	80%
Assignment 6	58%	75%
Assignment 7	55%	70%
Assignment 8	52%	65%
Assignment 9	48%	60%
Assignment 10	45%	55%
Assignment 11	42%	50%
Assignment 12	38%	45%

Figura 4 – Tela inicial PeergradeIO

FONTE: PEERGRADEIO(2016)

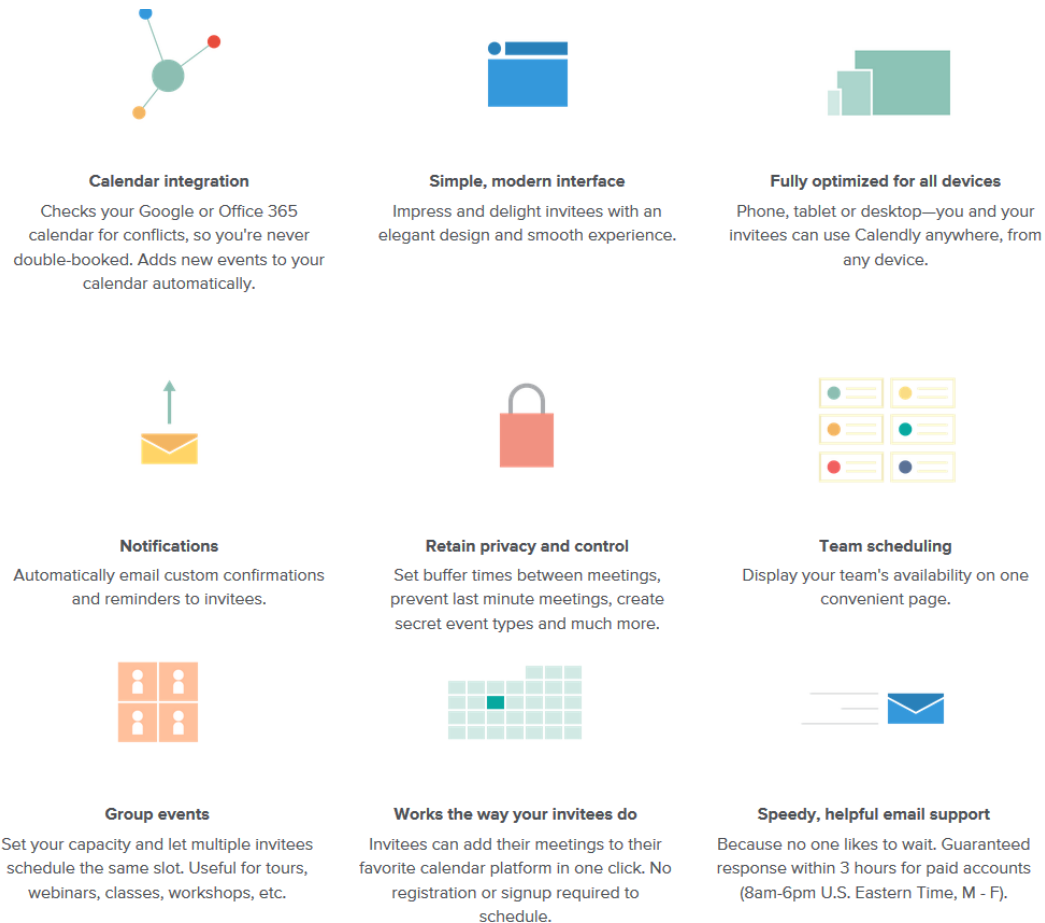
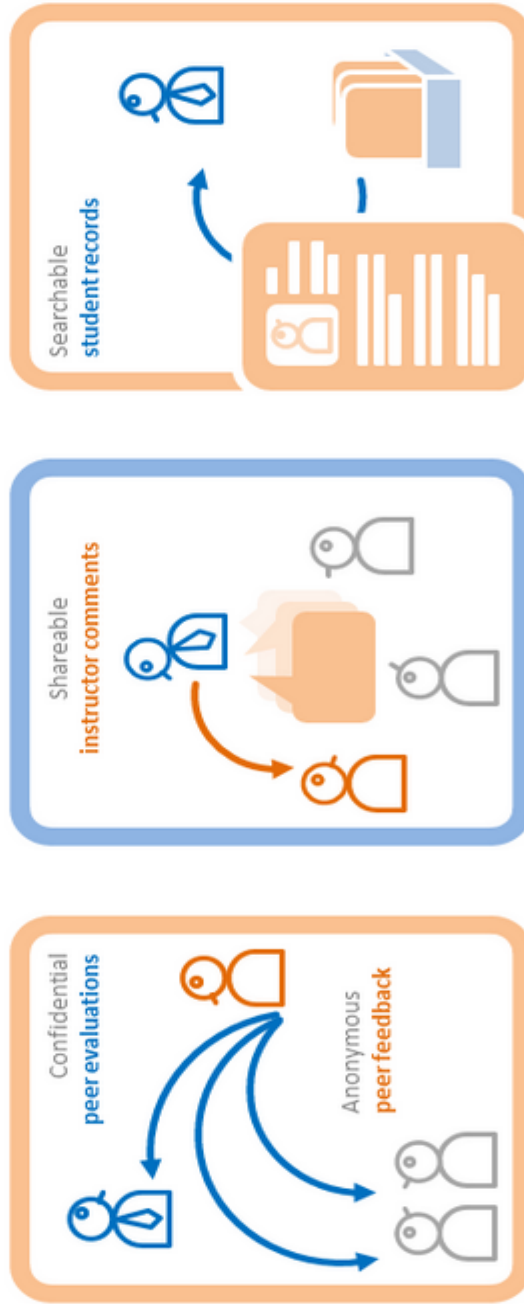


Figura 5 – Funcionalidades PeergradeIO

FONTE: PEERGRADEIO(2016)

3) TEAMMATES: Ferramenta criada por uma equipe de professores e alunos, para professores e alunos, com alto grau de flexibilidade, totalmente em inglês. A tecnologia baseada em pares vem a ser muito bem aplicada, em especial nesta ferramenta. Ela possui uma interface muito simples e de fácil aprendizado, disponibiliza um vídeo inicial, que demonstra a praticidade de utilizar o *software*. Em seu *website*, encontramos *feedback* de seus usuários, com muitos elogios em relação a sua usabilidade e respostas positivas de aproveitamento de conhecimento de alunos. Ferramenta gratuita. Na tela inicial possui um botão de acesso ao professor ,onde ao clicar é solicitado a requisição de preenchimento de um formulário, como suas permissões de acesso. A Figura 6 apresenta a tela inicial desse sistema.

Student peer evaluations/feedback, shareable instructor comments, and more...



[Request a FREE Instructor Account](#)

[Video Tour](#)

3,708,483 feedback entries submitted so far ...

Figura 6 – Tela inicial Teammates

FONTE: TEAMMATES(2016)

A análise dos *softwares* apresentados acima, serviu como uma das formas de levantamento de requisitos para o sistema proposto neste trabalho. Na *Tabela 1* é apresentada uma comparação entre esses *softwares* e o *software* proposto, o *Covaliator*.

Tabela 1 – Comparativo entre aplicativos similares ao Coavaliator.

FONTE: OS AUTORES (2016).

	CROWDGRADER	PEERGRADEIO	TEAMMATES	COAVALIATOR
Plataforma Web	x	x	x	x
Avaliação em Pares	x	x	x	x
Confidencial	x	x	x	x
<i>Feedback</i>	x	x	x	x
<i>Dashboard</i>		x		x
<i>Login</i> próprio		x	x	x
Visão Aluno	x	x	x	x
Visão Professor	x	x	x	x
Visão Por instituição		x		x
Visão por Curso	x	x	x	x
Visão por disciplina				x
Visão por turma				x
Idioma Português				x
Gratuito	x	x	x	x
Código Aberto				x

Com base na Tabela 1, concluímos que o desenvolvimento do *software Coavaliator* justifica-se pela ausência de opções de aplicações com as

funcionalidades necessárias para atender os requisitos do cliente deste projeto. Além disso, a proposta de ser um *software* de *Peer Assessment* totalmente em Português, elimina a dificuldade imposta por outro idioma, o Inglês, o qual a maioria dos brasileiros não domina. Portanto, acreditamos que o *software* desenvolvido é único pela combinação de tais funções.

4 METODOLOGIA DE TRABALHO

Todo processo de construção do projeto, análise e desenvolvimento foi baseado nos requisitos do cliente, o Prof. Dr. Alexander Robert Kutze do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) do Setor de Educação Profissional e Tecnológica (SEPT) da Universidade Federal do Paraná (UFPR).

Durante a especificação dos requisitos foi colocado em prática toda ideia inicial do projeto, através da prototipação das telas, visando a melhor forma de desenvolver, para que não tivéssemos nenhum tipo de retrabalho.

Por se tratar de uma ferramenta totalmente nova para os usuários, foram projetadas interfaces simples e eficientes, para facilitar o acesso as funcionalidades do sistema. Testes foram realizados e a interface sem dúvida ajudou na execução deste processo, tornando as funcionalidades evidentes e fáceis de serem executadas.

A forma como foi desenvolvido, realizando levantamento de requisitos, entrevista com o cliente, modelagem do banco de dados de forma clara e sucinta, utilizando para o desenvolvimento a linguagem de programação orientada a objetos (JAVA), utilizando o MVC , vem à ajudar a solucionar futuras correções ou erros que venham a surgir. Tornando simples todo o processo, e utilizando o SCRUM como método de gerenciamento de projeto. A Figura 7, demonstra toda a metodologia de trabalho utilizada para o desenvolvimento do sistema.

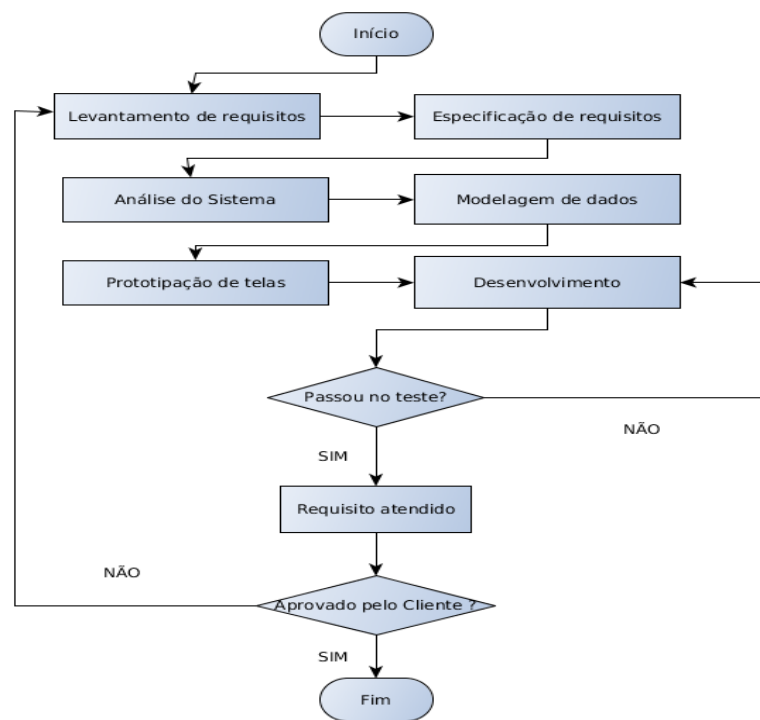


Figura 7 – Fluxograma da metodologia do trabalho.

FONTE: OS AUTORES (2016).

4.1 REQUISITOS

4.1.1 LEVANTAMENTO DE REQUISITOS

Para a etapa de levantamento de requisitos, buscou-se, primeiramente, compreender em detalhes a ideia do cliente, os benefícios e as restrições que o projeto traria. Conforme os requisitos foram sendo especificados nas primeiras reuniões com o cliente, o Prof. Dr. Alexander Robert Kutze do curso TADS do Setor de Educação Profissional e Tecnológica da UFPR, protótipos de telas foram implementados e discutidos com ele. Desta forma foi possível compreender perfeitamente a real necessidade do professor e refinar os requisitos já levantados.

Para tanto, o levantamento dos requisitos foi elaborado utilizando três técnicas: (a) Entrevistas com o cliente; (b) Análise de sistemas existentes e suas correlações com o sistema proposto; (c) Prototipação de telas.

(a) **Entrevista:** a Entrevista é uma técnica de levantamento de requisitos muito usada em toda a engenharia de *software*, por ser muito simples e eficiente para o levantamento de requisitos funcionais e não funcionais que um *software* deverá

apresentar em sua entrega (REISSWITZ,2013). Esta técnica deve ser levada muito a sério e deve-se ter um cuidado extremo com o entrevistado para não fugir do tema abordado ou o entrevistador não induzir o entrevistado aos requisitos aos quais ele acredita contribuir mais para a construção do novo *software*. O trabalho de análise e desenvolvimento do sistema proposto, foi elaborado de acordo com os requisitos e necessidades do cliente, o professor Alexander. No processo de coleta dos requisitos foram elaboradas entrevistas abertas com o mesmo e apresentação de protótipos de telas do sistema, e desta forma foi possível elencar os requisitos funcionais e não funcionais que o sistema deve ter para atender a metodologia *Peer assessment*. Foi necessário realizar uma entrevista com nosso cliente, professor Alexander, a qual foi dividida em 3 reuniões. Os questionários aplicados na entrevista encontram-se no Apêndice 1.

(b) **Análise de sistemas correlatos:** analisar *softwares* existentes, verificar e entender técnicas já aplicadas e comparar com uma nova proposta de sistema passada pelo seu cliente, trata-se de uma técnica de suma importância para o levantamento de requisitos (SCHACH,2010). Podemos encontrar muitas ideias, informações que podem vir agregar muito valor ao seu projeto, ou mesmo verificar fatores com pouca relevância ao todo do projeto. No sistema proposto conseguimos inserir funcionalidades inéditas em relação ao *softwares* analisados. (Tabela 1. Comparativo entre aplicativos similares ao *Coavaliator*).

(c) **Prototipação de interface:** a prototipação de telas envolve a criação de versões iniciais de um sistema futuro, nele podemos realizar alguns experimentos antes do *software* ser de fato desenvolvido (MARTINS, 2007). Os protótipos das telas do *software Coavaliator*, foram desenvolvidos com base nos requisitos levantados e na modelagem dos dados, em seguida agendadas reuniões com cliente para demonstração dos mesmos e refinamento dos requisitos. Os protótipos foram desenvolvidos na ferramenta *Balsamiq* e encontram-se no Apêndice 2.

4.1.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA

(a) **Funcionais:** são aqueles que definem uma função ou método a ser executado pelo sistema que será desenvolvido, eles definem o comportamento do sistema e são documentados por meio de casos de uso (GILLEANES,2010). Durante o processo de entrevista com nosso cliente, utilizamos questionários (Apêndice 1), com o objetivo de entender qual a real necessidade do sistema e sua usabilidade contínua no âmbito acadêmico. Com isso, identificamos as principais funcionalidades do sistema que refletem a metodologia *Peer Assessment*. A Tabela 2 apresenta os principais requisitos funcionais do sistema proposto.

Tabela 2 – Requisitos funcionais Coavaleitor.

FONTE: OS AUTORES (2016).

Módulo	Requisito	Descrição
Estudante	Inscriver em turma	O estudante deve se inscrever em uma turma.
	Realizar <i>Login</i>	O estudante deve realizar <i>Login</i> com seu perfil Aluno
	Atualizar Cadastro	O estudante realiza e mantém atualizado seu cadastro no sistema.
	Visualizar atividades.	O estudante visualiza todas as atividades que ele tem pendente.
	Realizar atividades	O estudante realiza todas as atividades que tem pendente.
	Visualizar correção das atividades	O estudante visualizar correções de suas atividades.
	Avaliar/comentar as atividades de outros alunos	O estudante pode avaliar e corrigir atividade de outros alunos; a quantidade de correções permitidas é definida pelo professor
	Visualiza <i>FeedBack</i> de Professor.	O estudante visualiza <i>FeedBack</i> de seu professor.

Módulo	Requisito	Descrição
Professor	Atualizar Cadastro	O professor pode atualizar seu cadastro quando achar necessário, assim como de seus alunos.
	Manter Atividade	O professor mantém atividades de alunos, que serão visíveis no portal do aluno.
	Manter correção	O professor é o responsável por manter as correções de atividades dentro do prazo estipulado.
	Manter turmas	O professor é o responsável por cadastrar e manter as turmas no sistema.
	Visualizar Relatório	O professor pode visualizar os relatórios que desejar, de cada aluno, por turma ou por instituição todos previamente cadastrados no sistema.
	Visualizar <i>Dashboard</i> .	O professor pode visualizar os <i>Dashboards</i> das atividades que foram realizadas por seus alunos.
Professor Administrador	Manter instituição	O professor administrador é o responsável por cadastrar as instituições no sistema, onde cada professor e aluno possuem relacionamentos em comum.
	Manter Curso	O professor administrador é o responsável por cadastrar os cursos no sistema.
	Manter disciplina	O professor administrador é o responsável por cadastrar as disciplinas no sistema.

(b) Não Funcionais: requisito não funcional de *software* não é aquele que descreve o que o sistema fará, mas sim como ele fará, eles não estão relacionados diretamente com a funcionalidade do *software* (GILLEANES,2010). Estes são aplicados ao sistema como um todo e são mais imprevisíveis do que os requisitos funcionais. Para o sistema proposto foram levantados os seguintes requisitos funcionais, apresentados na Tabela 3.

Tabela 3 – Requisitos não funcionais Coavaleitor.

FONTE: OS AUTORES (2016).

Tipo de Requisito	Requisito	Descrição
Requisitos de Produto	Velocidade de Resposta	O tempo de resposta não deve ultrapassar 45 segundos.
	Implementação	Desenvolvimento em camadas (MVC).
	Confiabilidade	Segurança de dados dos professores e aluno. Senha criptografada em MD5.
	Disponibilidade	Sistema deve rodar pelo menos no Chrome e Firefox.
	Tecnologias envolvidas	Utilizar tecnologias livres no desenvolvimento do projetos
Requisitos de Interface	Ajuda e documentação	<i>Tooltips</i> explicativos serão utilizados em telas que realizam procedimentos complexos.
	Consistência	Ícones seguem um padrão. Ex. Deletar sempre é vermelho.
	Reconhecer, diagnosticar e sanar erros.	Mensagens de erro serão apresentadas de forma objetiva para ajudar o usuário a diagnosticar e sanar seus erros.

4.2 MODELAGEM DOS DADOS DO SISTEMA

A modelagem dos dados de um sistema pode ser representada por diferentes níveis de abstração. (MARTINS, 2007).

- **Conceitual:** é uma descrição de banco de dados de forma independente de implementação em um Sistema Gerenciador de Banco de Dados (SGBD). O modelo conceitual pode ser descrito de forma textual.
- **Lógico:** é uma descrição das estruturas que serão armazenadas no banco de dados e que resulta numa representação gráfica dos dados de uma maneira lógica, inclusive nomeando os componentes e ações que exercem uns sobre os outros.
- **Físico:** é a descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, esse modelo depende do SGBD que está sendo usado e é nele que são detalhados os componentes da estrutura física do banco, como tabelas, campos, tipos de valores e índices.

Para a descrição da modelagem dos dados do sistema foi utilizado o DER – Diagrama Entidade Relacionamento, encontra-se no Apêndice 7. Durante a modelagem dos dados do sistema *Coavaliator* foi considerado, principalmente, a integridade dos dados, a fácil manipulação e a manutenção dos dados. Procurou-se generalizar o menos possível o diagrama, para torná-lo de fácil entendimento e ao mesmo tempo versátil, atendendo todos os requisitos.

4.3 MODELAGEM DAS FUNCIONALIDADES DO SISTEMA

Para a modelagem das funcionalidades do sistema foi utilizada a UML – *Unified Modelling Language* ou Linguagem de Modelagem Unificada. Esta é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação objetos.

É uma linguagem de modelagem de propósito geral, que pode ser aplicada a todos os domínios de aplicação. É bom frisar que UML não é uma linguagem de

programação é uma linguagem de modelagem, para definir as características do sistema, o seu comportamento e sua estrutura lógica (GILLEANES, 2011).

Os seguintes diagramas da UML foram modelados neste projeto: Caso de Uso, Diagrama de Classe, Diagrama de Sequência e Diagrama de Atividade, os quais serão apresentados nas próximas seções.

4.3.1 DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

Este é um diagrama mais geral e informal da UML, utilizado na fase de levantamento e análise de requisitos do *software*, embora venha ser consultado durante todo processo e pode vir a servir como base para a criação de outros diagramas (GILLEANES,2010).

O Diagrama de caso de uso do sistema proposto define três perfis de usuário: Administrador, Professor e Aluno. Encontra-se no Apêndice 3.

As especificações dos casos de uso encontram-se no Apêndice 4 desta documentação.

4.3.2 DIAGRAMA DE CLASSES

Diagrama de Classe é com certeza o mais utilizado e dos mais importantes da UML. Ele define a estrutura das classes que serão utilizadas pelo *software*, determinando os atributos e métodos que cada classe irá possuir, além de definir entre as classes quais serão seus relacionamentos (GILLEANES, 2011). Todas as classes de negócio definidas para o *software* proposto e seus relacionamentos encontram-se no Diagrama de Classes de Negócios, Apêndice 5.

É importante colocar que a estrutura utilizada para a modelagem do sistema foi o MVC. Esta estrutura está dividida em três partes: *Model*, *View* e *Controller*. Na *MODEL* estão as classes *DAO*, são essas que carregam as regras de negócio, utilizadas como uma ponte por onde passam as informações ou requisições feitas a partir da interface do usuário. Na camada *VIEW* estão as interfaces do sistema, todas

as telas e funcionalidades são aqui desenvolvidas. Por fim, na camada *CONTROLLER* são implementadas as classes que enviam comandos entre as camadas *VIEW* e *MODEL*.

O Diagrama de Classe de Implementação, que apresenta as classes das três camadas (*Model*, *View* e *Controller*) e os seus relacionamentos encontra-se no Apêndice 6 desta documentação.

4.3.3 DIAGRAMA DE SEQUÊNCIA

O Diagrama de sequência, é um diagrama comportamental, que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo (GILLEANES,2011). Ele se “apoiar” no diagrama de classe para determinar os objetos das classes envolvidas em um processo. Com isso, ele consegue identificar o evento gerador do processo modelado, bem como o ator responsável pelo evento, e determinar a melhor forma do processo ser desenvolvido, por meio de chamadas de métodos enviados entre os objetos.

Os seguintes processos foram modelados (Tabela 4):

Tabela 4 – Processos representados pelos diagramas de sequência.

FONTE: OS AUTORES (2016).

Ator	Processos
ALUNO	<ul style="list-style-type: none"> ● Submeter Solução ● Corrigir Avaliação ● Visualizar Notas ● Recuperar Senha (Aluno e Professor) ● Manter Matrícula (Aluno e Professor) ● Manter Usuário (Aluno e Professor) ● Login (Aluno e Professor)
PROFESSOR	<ul style="list-style-type: none"> ● Manter Turma ● Manter Avaliação ● Visualizar Soluções ● Visualizar Correções ● Avaliar Aluno

Ator	Processos
PROFESSOR ADMINISTRADOR	<ul style="list-style-type: none"> ● Manter Instituição ● Manter Disciplina ● Manter Curso

Os Diagramas de Sequência do sistema encontram-se no Apêndice 8 desta documentação.

4.3.4 DIAGRAMA DE ATIVIDADE

O Diagrama de Atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo ser representado por métodos com certos graus de complexidade, podendo ser um algoritmo ou por processo completo (GILLEANES,2011). A aplicação desse diagrama torna-se interessante quando se deseja representar uma atividade que é executada por 2 ou mais atores e que suas ações se intercalam durante o processo.

A atividade de Visualizar Notas enquadra-se nesse quesito, ou seja, uma atividade sendo utilizada de forma intercalada por dois atores. Este diagrama encontra-se no Apêndice 9 desta documentação.

4.4 METODOLOGIA DE GERENCIAMENTO DO PROJETO

Para o gerenciamento do projeto foi aplicada uma metodologia adaptada do SCRUM, para a realidade da equipe.

O foco no SCRUM é controlar o tempo de desenvolvimento no menor tempo possível. Nele os projetos são divididos em *Sprints* que são ciclos repetitivos de atividades, que podem variar entre duas a quatro semanas (CRUZ, 2015). No início de cada *Sprint* é feito uma reunião de planejamento onde é discutido e verificado quais atividades terão prioridades e quais os prazos para cada uma.

A implementação do *Scrum* em um projeto de desenvolvimento de *software* está baseada em três elementos de sustentação: (1) transparência, (2) inspeção e (3)

adaptação. A transparência garante que todos os envolvidos no projeto, desenvolvedores e os responsáveis por inspecionar os resultados, compartilhem da mesma definição de produto pronto. Inspeção diz respeito a capacidade de se inspecionar o processo a qualquer momento, desde que a frequência não seja tamanha a ponto de atrapalhar na execução. E, por fim, a adaptação, que refere-se a possibilidade de correção de variações ao longo do processo de inspeção (CRUZ, 2015).

O *Scrum* define três papéis principais: *Product Owner*, *Scrum Master* e *Team Member*, conforme descrevem Schwaber & Beedle (2002).

- *Product Owner*: comunica a visão do produto para a equipe de desenvolvimento e representa os interesses do cliente através de requisitos e priorização. Este representado pelo nosso cliente, o professor Alexander Robert Kutze.
- *Scrum Master*: atua como uma ligação entre o *Product Owner* e a equipe. Ele não gerencia a equipe, mas trabalha para ajudar a equipe a alcançar os objetivos das *sprints*, removendo obstáculos. Este representado por nossa orientadora, professora Andreia de Jesus.
- *Team Members*: pessoas que executam o trabalho do projeto. O time normalmente consiste de engenheiros de software, arquitetos, analistas e testadores. Este representado pelos três membros da equipe desse projeto.

No *Scrum* também há quatro eventos formais que auxiliam nas ações de inspeção e adaptação:

- Reunião de planejamento da *sprint*;
- Reunião diária;
- Reunião de revisão da *sprint*;
- Retrospectiva da *sprint*.

Para o projeto não foi possível a realização de reuniões presenciais diárias, entretanto foi estabelecido contato entre a equipe diariamente por meio de mensagens, por diferentes plataformas de comunicação para controle do andamento

das atividades. As reuniões com o cliente foram feitas semanalmente ou quinzenalmente, com a apresentação da evolução do trabalho para nosso cliente e orientador.

Ao todo, foram feitas seis *sprints* ao longo do projeto, conforme apresentado na Tabela 5, com as entrega das fases de análise e desenvolvimento de cada *sprint*.

Inicialmente o foco foi a modelagem do sistema. Na *Sprint 1 e Sprint 2* foi realizada a entrevista com o cliente e, desta forma, conseguimos levantar os requisitos necessários do sistema. Em seguida foi feita a elaboração dos diagramas de casos de uso e físico do banco de dados e, também, o desenvolvimento de módulos de base para o sistema. Já na *Sprint 3 e Sprint 4*, o desenvolvimento das funcionalidades foi priorizado e iniciou-se o desenvolvimento da documentação do sistema.

Nas últimas duas *sprints (Sprint 5 e Sprint 6)* foram finalizadas as funcionalidades dos módulos do aluno e professor e o desenvolvimento do restante da documentação. Nestas *sprints* também foi incluído os testes finais do sistema.

Por fim, a equipe realizou reuniões periódicas para acompanhar o andamento e inspecionar os resultados do desenvolvimento. As variações eram levantadas e discutidas em conjunto visando sempre atender as expectativas e os requisitos do cliente e qual a melhor e mais eficiente forma de solucioná-lo.

Tabela 5 – Planilha de entregas SCRUM.

FONTE: OS AUTORES (2016).

Reunião	Tarefas	Data Inicial	Data Final
Sprint 1	Elaboração de questões para entrevista com o cliente.	08/08/2016	12/08/2016
Sprint 1	Reunião com Cliente e levantamentos dos requisitos	08/08/2016	12/08/2016
Sprint 1	Discussão interna sobre reunião com cliente.	08/08/2016	12/08/2016
Sprint 1	Escolhas de framework para desenvolvimento do projeto	08/08/2016	12/08/2016

Reunião	Tarefas	Data Inicial	Data Final
Sprint 2	Pesquisa Trabalhos Correlatos.	13/08/2016	26/08/2016
Sprint 2	Definição de tecnologias de desenvolvimento.	13/08/2016	26/08/2016
Sprint 2	Reunião interna da equipe.	13/08/2016	26/08/2016
Sprint 3	Término capítulo 1. Introdução do projeto.	27/08/2016	05/09/2016
Sprint 3	Prototipação das telas dos 3 módulos do sistema(Aluno, Professor Administrador).	27/08/2016	05/09/2016
Sprint 3	Revisão Orientador Capítulo 1.	27/08/2016	05/09/2016
Sprint 4	Criação do Banco de Dados	06/09/2016	16/09/2016
Sprint 4	Escrever e Revisar capítulo 2	06/09/2016	16/09/2016
Sprint 4	Escrever e Revisar capítulo 3	06/09/2016	16/09/2016
Sprint 4	FrontEnd Login Aluno	06/09/2016	16/09/2016
Sprint 4	Frontend Professor	06/09/2016	16/09/2016
Sprint 4	FrontEnd Portal Aluno	06/09/2016	16/09/2016
Sprint 4	FrontEnd Tela Portal Professor	06/09/2016	16/09/2016
Sprint 4	FrontEnd Tela Portal Administrador	06/09/2016	16/09/2016
Sprint 5	Cap. 4 Metodologia de Trabalho	17/09/2016	07/10/2016
Sprint 5	Cap.4.1 Requisitos Funcionais e não funcionais	17/09/2016	07/10/2016
Sprint 5	Cap. 4.1.2 Modelagem das Funcionalidades do Sistema	17/09/2016	07/10/2016
Sprint 5	Backend Portal do Aluno	17/09/2016	07/10/2016
Sprint 5	Backend Portal do Professor	17/09/2016	07/10/2016
Sprint 5	Backend Portal do Administrador	17/09/2016	07/10/2016
Sprint 5	Primeira entrega para revisão parcial do Projeto para professora	17/09/2016	07/10/2016

Reunião	Tarefas	Data Inicial	Data Final
	Orientadora		
Sprint 6	Desenvolvimento do DashBoard e Relatórios	08/10/2016	23/11/2016
Sprint 6	Ajustes Caso de Uso	08/10/2016	23/11/2016
Sprint 6	Criação Diagrama Classe	08/10/2016	23/11/2016
Sprint 6	Criação Diagrama Sequência	08/10/2016	23/11/2016
Sprint 6	Criação Diagrama Atividade	08/10/2016	23/11/2016
Sprint 6	Ajustes no DER	08/10/2016	23/11/2016
Sprint 6	Cap. 5 O Sistema Coavalíator.	08/10/2016	23/11/2016
Sprint 6	Considerações Finais.	08/10/2016	23/11/2016
Sprint 6	Entrega final e Revisão pela professora Orientadora.	08/10/2016	23/11/2016

4.5 TESTES REALIZADOS

Esta etapa de testes é muito importante no desenvolvimento de qualquer projeto de *software*, que consiste em um processo dinâmico do projeto desenvolvido, a fim de eliminar ou diminuir erros persistentes. Estes testes, ajudam a identificar problemas de usabilidade e aumenta consideravelmente confiabilidade com relação a utilização e manutenção do sistema desenvolvido.

Foram realizados testes de caixa-branca para a verificação das funcionalidades implementadas e testes de caixa-preta para a validação dos testes do *software*, de acordo com os requisitos levantados.

A técnica de teste de caixa-branca é conhecida como teste estrutural, em que se realiza testes diretamente no código fonte (COSTA,2014). O responsável pelo teste deve conhecer plenamente toda a estrutura do *software*, e elabora os testes de forma a verificar todas as possibilidades possíveis de validações das informações; concentra-se apenas no código fonte e não verifica a lógica da especificação.

Já os testes de caixa-preta são conhecidos como testes funcionais ou comportamentais, pois são baseados nos requisitos funcionais do sistema, ou seja, nas ações que o sistema deve desempenhar (COSTA, 2014). São chamados de caixa-preta porque não é analisada a estrutura de codificação, não é visível o código na hora de aplicar esse teste, utilizando somente a parte da interface para a realização do teste.

4.5.1 EXEMPLO DE TESTE CAIXA-BRANCA

O Quadro 2, ilustra o trecho de código da tela de *login*, selecionado para realizar o teste de caixa branca. É possível verificar o caminho básico desse teste na Figura 8 e os casos de testes gerados na Tabela 6.

Quadro 2 – Exemplo de teste Caixa-Branca.

FONTE: OS AUTORES (2016).

Exemplo Testes Caixa-Branca.

Método do Caminho Básico:

```

<% @include file="include/header.jsp" %>
    <div class="wrapper">
        <form class="form-signin" action="LoginController?logar=true"
method="POST">
            
            <% @include file="include/mensagem.jsp"%>

```

1

2

3 <button class="btn btn-lg btn-primary btn-block" type="submit">Login</button>

4 Esqueceu sua senha?

5 Ainda não possui uma conta? Cadastre-se já

```

        </form>
    </div>
<% @include file="include/footer.jsp" %>

```

$V(G) = E - N + 2$

$V(G) = 4 - 5 + 2$

$V(G) = 1$

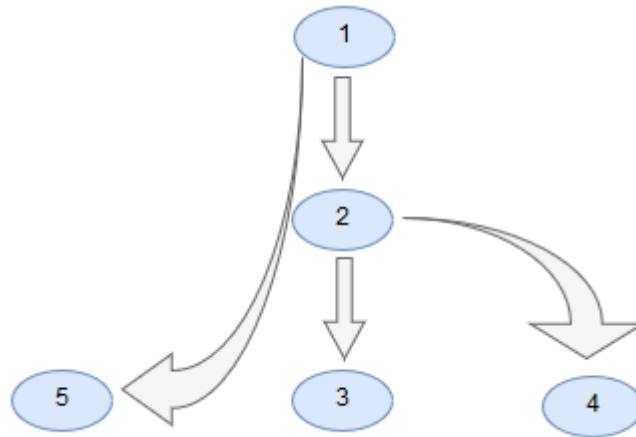


Figura 8 – Caminho de Teste Básico.

FONTE: OS AUTORES (2016).

Tabela 6 – Caso de teste Caixa-Branca.

FONTE: OS AUTORES(2016).

#	Caminho	Descrição	Procedimento	Resultado esperado
1	1,2,3	Email e senha Correspondentes	Inserir e-mail no padrão, e com senha correta.	Não deve aparecer nenhuma mensagem de advertência e usuária acessa sistema como esperado.
2	1,2,4	Usuário esqueceu senha cadastrada.	Usuário é direcionado para nova tela, nesta irá informar seu e-mail cadastro onde será redirecionada uma nova senha para e-mail informado.	Aparecerá uma mensagem, que nova senha foi enviada para e-mail informado.
3	1,5	Usuário ainda não possui acesso ao sistema	Ao clicar na mensagem “Ainda não possui acesso, Deseja se cadastrar agora?”. Usuário é redirecionado para tela de cadastro inicial.	Usuário realizar login no sistema sem nenhuma mensagem de erro.

4.5.2 EXEMPLO DE TESTE CAIXA-PRETA

Nesse exemplo utilizamos o método de partição de equivalência na tela de *login*, Figura 9. Foi criado um caso de teste, ilustrado na Tabela 7.

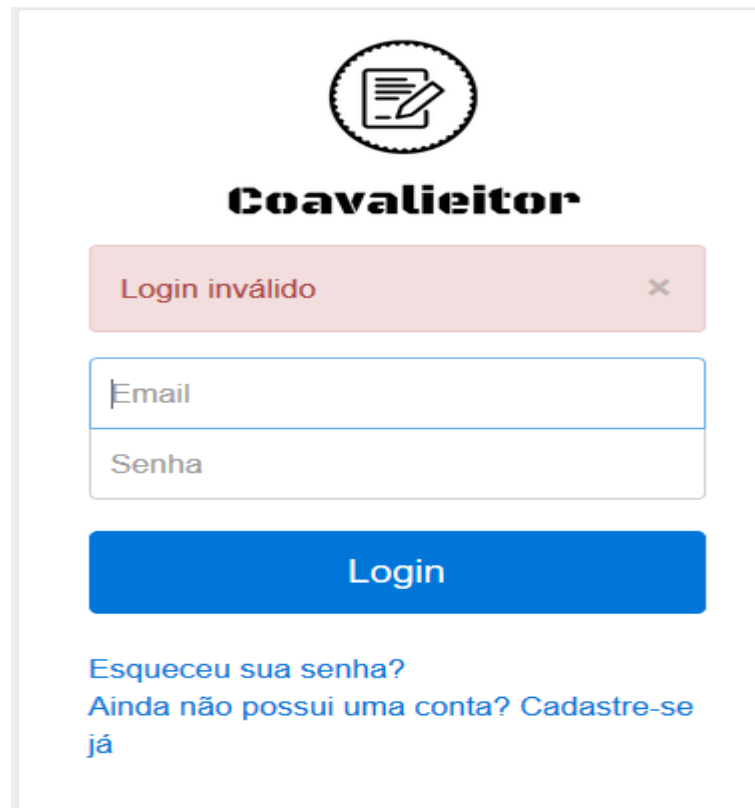


Figura 9 – Erro ao logar no Sistema.

FONTE: OS AUTORES (2016).

Tabela 7 – Caso de teste Caixa-Preta.

FONTE: OS AUTORES (2016).

#	Descrição	Procedimento	Resultado esperado
1	E-mail e senha não cadastrados são informados em seus respectivos locais.	Após usuário informar os dados deve clicar em botão "Login".	Deve aparecer mensagem de erro em vermelho informando "Login Invalido".

4.6 TECNOLOGIAS E LINGUAGENS UTILIZADAS NO DESENVOLVIMENTO DO SISTEMA

Para o desenvolvimento do projeto foram utilizadas as ferramentas a seguir apresentadas, as quais auxiliaram diretamente na construção, codificação e interação da equipe em relação ao desenvolvimento do projeto.

a) Modelagem do sistema

- *MySql*

Sistema de gerenciamento de banco de dados que utiliza a linguagem SQL. Segundo o *DB-Engines Ranking* é o segundo SGBD mais popular do mundo em Junho de 2016, ficando atrás apenas do *Oracle*. O *MySQL* suporta qualquer plataforma atual, possui excelente desempenho e compatibilidade com outros sistemas.

- *Balsamiq*

Basalmiq Mockup (BALSAMIQ, 2011) da *Basalmiq Studio* é uma aplicação desenvolvida na linguagem de programação *ActionScript*, que executa *Adobe AIR* (*Adobe Integrated Runtime*). Essa aplicação é utilizada para desenvolver protótipos ou modelos (*mockups*), como as telas de um sistema *desktop* ou sistema/páginas web ou mobile.

- *Astah Community*

Ferramenta gratuita de modelagem de diagramas UML. O software foi desenvolvido no Japão na plataforma Java em 2006 pela empresa *Change Vision Inc* (VISION, 2006).

- *yEd Graph Editor*.

Ferramenta gratuita utilizada para o desenvolvimento dos fluxogramas. Desenvolvida na linguagem JAVA.

b) Implementação do Sistema e do Banco de Dados

- *NetBeans 8.1*

O *NetBeans* é um ambiente de desenvolvimento integrado e gratuito, o qual atende diversas linguagens de programação (JAVA, PHP entre outras). Roda

em diferentes sistemas operacionais, como Windows e Linux. Atualmente está na versão 8.1.

- *GlassFish Server Open Source Edition 4.1.1*

GlassFish é um servidor web Java, que suporta todas as especificações da API Java EE, conectores, *servelets*, JSF e diversas tecnologias de *WebService*. É distribuído como *software* livre e liderado pela *Oracle*.

- *Bootstrap*

Bootstrap é uma ferramenta gratuita de código aberto para criação de *websites* e aplicações web. É um *framework* de *front-end*, que facilita o desenvolvimento de sites dinâmicos. Ele é hospedado, desenvolvido e mantido no *Github*. Está atualmente na versão 4.0.0, lançada em agosto de 2015.

- Linguagem programação Java Web (JSP, *Servlet*, EL - *Expression Language*)

Java é uma linguagem de programação orientada à objetos. É uma linguagem que possibilita a utilização de *frameworks* e ambientes para facilitar o desenvolvimento das aplicações. Possui uma comunidade de desenvolvedores muito vasta, onde facilita a troca de conhecimentos entre os desenvolvedores.

- *HighCharts 4.2.7*

Utilizado para criar os gráficos do *Dashboard* do sistema. O *HighCharts* é um conjunto de bibliotecas *JavaScript* de uso gratuito mantido pela empresa *HighSoft*.

c) Gerenciamento e Documentação do Projeto

- GitHub

Github é um serviço de hospedagem de sites compartilhado que utiliza o controle de versionamento *git*¹⁹. Possui uma interface web gráfica, controle de acesso e várias características como rastreamento de erros, gerenciamento de tarefas e tutoriais para projetos. Oferece planos para projetos privados e livres. Possui mais de 11 milhões de usuários e mais de 29.4 milhões de repositórios, 20 ao redor do mundo.

- Arquitetura MVC

MVC (*Model-View-Controller*) é um padrão de arquitetura de *software*, que divide a aplicação em três partes: modelo, visão e controlador. Essa estrutura permite que as informações referentes às regras de negócio fiquem separadas da interface com a qual o usuário interage.

A equipe optou pelo uso das ferramentas e tecnologias citadas acima pelo fato de possuir conhecimento prévio quanto ao uso de grande parte delas e, também, pela popularidade das mesmas no ramo em que atuam. Além disso, todas são disponibilizadas gratuitamente, mesmo que em versões com funcionalidades reduzidas. A criação deste software ajudará a tornar essa forma de metodologia de aprendizado, Avaliação por pares, conhecida no Brasil, por este motivo, iremos disponibilizar este software gratuitamente para escolas públicas que estejam dispostas a inovar e aprender com essa nova ferramenta.

5 APRESENTAÇÃO DO SOFTWARE

Neste capítulo apresentaremos o sistema COAVALIEITOR por meio de sua interface gráfica. A apresentação será dividida em quatro seções: Acesso ao Sistema, Manter Cadastros, Módulo do Professor; Módulo do Aluno. Um fluxograma (Figura 10) também foi elaborado para melhor compreender o funcionamento do sistema.

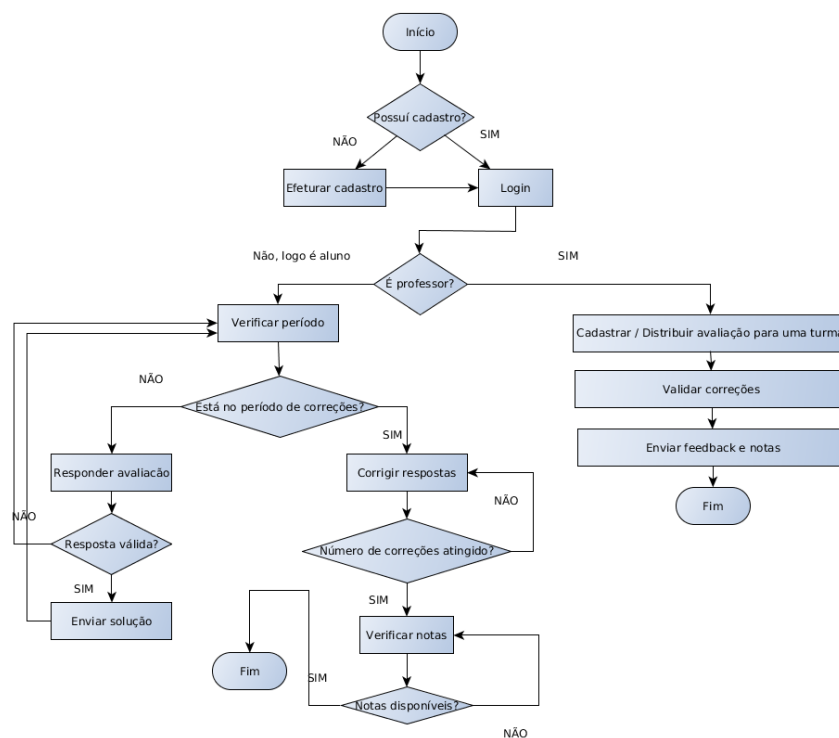


Figura 10 – Fluxograma do funcionamento do sistema Coavaleitor.

FONTE: OS AUTORES (2016).

5.1 ACESSO AO SISTEMA

O Tela de login (Figura 11) é uma etapa obrigatória para professores e alunos acessarem o sistema.

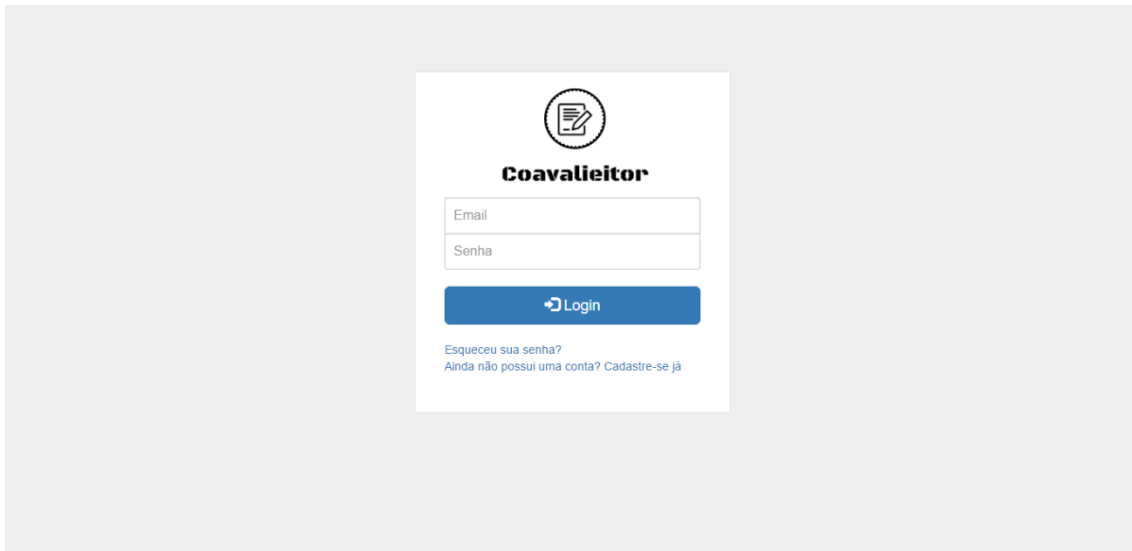


Figura 11 – Tela de Login.

FONTE: OS AUTORES (2016).

Caso seja necessário recuperar uma senha esquecida, o sistema fornece um mecanismo de recuperação, localizado na tela de recuperação de senha (Figura 12). Uma nova senha será enviada para o e-mail previamente cadastrado.

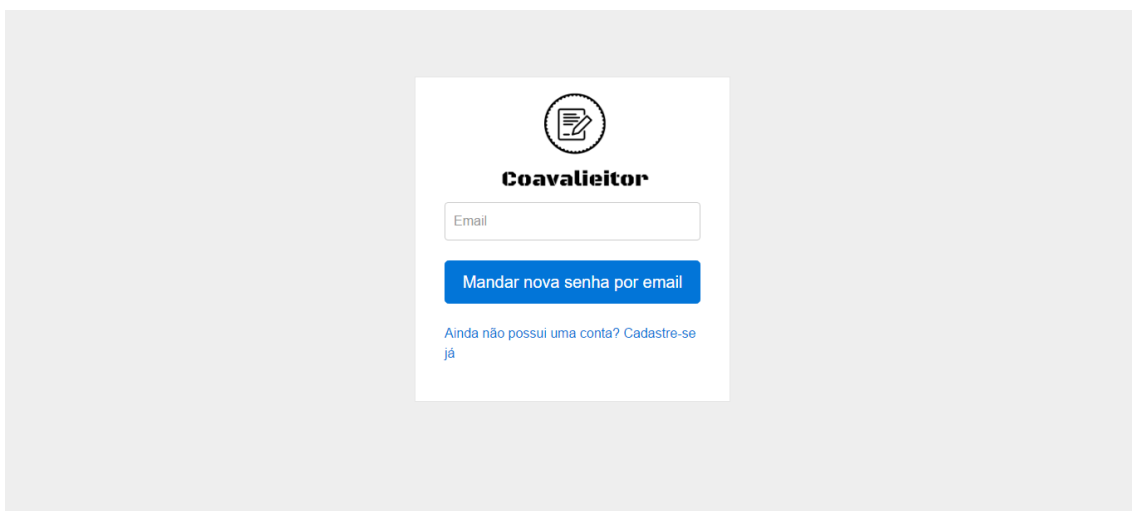
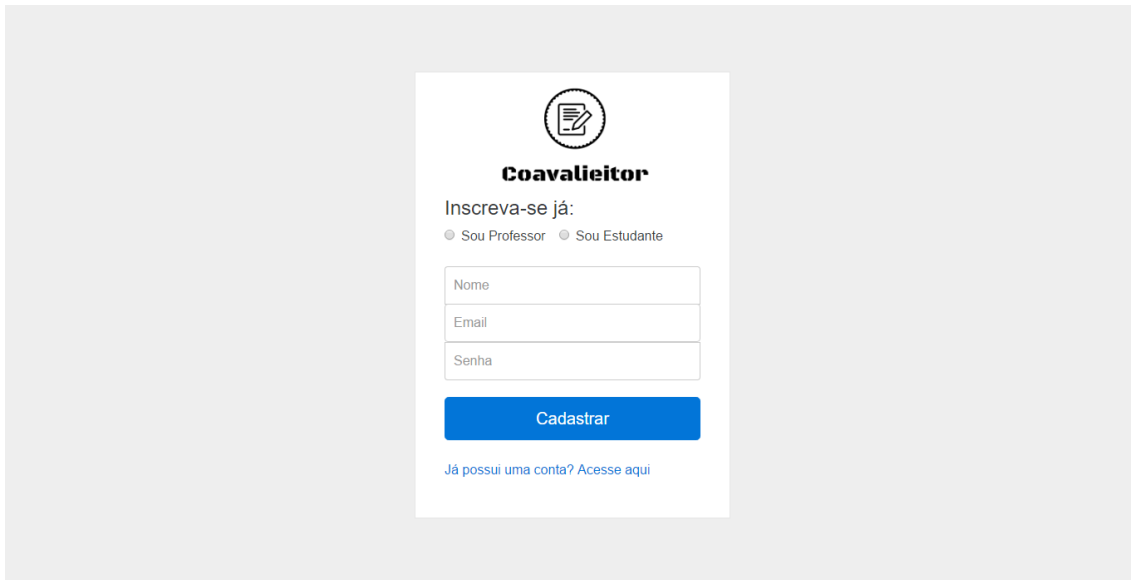



Figura 12 – Tela de Recuperação de senha.

FONTE: OS AUTORES (2016).

Para acessar o sistema pela primeira vez, é necessário preencher um cadastro (Figura 13) e definir seu perfil como Professor, ou Aluno.




Coavalleitor
 Inscreva-se já:
 Sou Professor Sou Estudante

Cadastrar

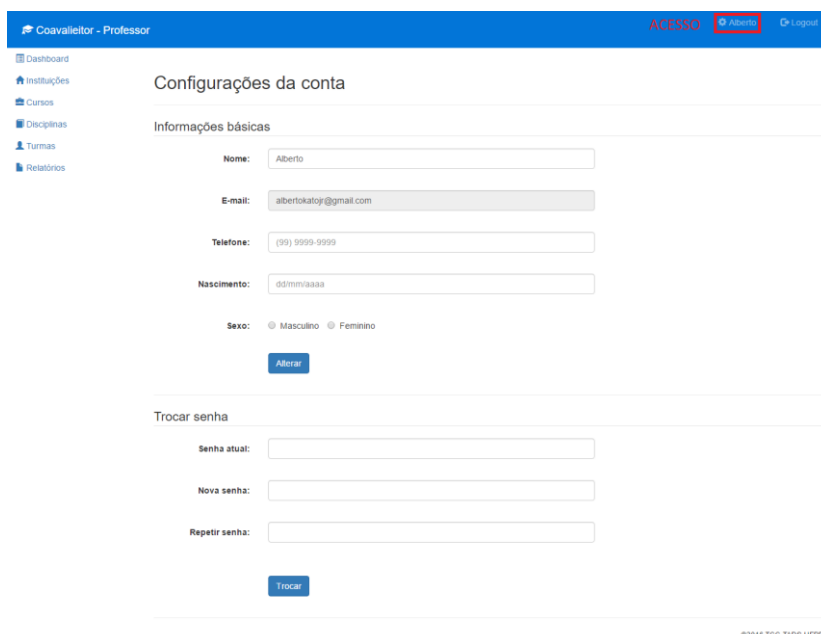
[Já possui uma conta? Acesse aqui](#)

Figura 13 – Tela de Cadastro de Usuário.

FONTE: OS AUTORES (2016).

5.2 MANTER CADASTROS

Professores e Alunos conseguem atualizar suas informações cadastrais na tela de configurações da conta (Figura 14), localizada no canto superior esquerdo, no nome do usuário logado.



Coavalleitor - Professor ACESSO **alberto** Logout

[Dashboard](#)
[Instituições](#)
[Cursos](#)
[Disciplinas](#)
[Turmas](#)
[Relatórios](#)

Configurações da conta

Informações básicas

Nome:
 E-mail:
 Telefone:
 Nascimento:
 Sexo: Masculino Feminino

Alterar

Trocar senha

Senha atual:
 Nova senha:
 Repetir senha:

Trocar

©2016 TCC-TADS-UFPR

Figura 14 – Tela de Configurações da Conta.

FONTE: OS AUTORES (2016).

5.2.1 PROFESSOR COM PERFIL DE ADMINISTRADOR

Somente o professor que possui um perfil de administrador no sistema, consegue realizar operações do tipo CRUD (*Creat, Read, Update, Delete*), nos seguintes elementos acadêmicos: Instituição, Curso, Disciplina.

Ao clicar em Instituições, no menu lateral esquerdo, será apresentado a tela de listar instituições (Figura 15). É possível excluir, editar e adicionar uma nova Instituição (Figura 16). Tanto o editar como o adicionar compartilham da mesma tela, só diferindo no título.

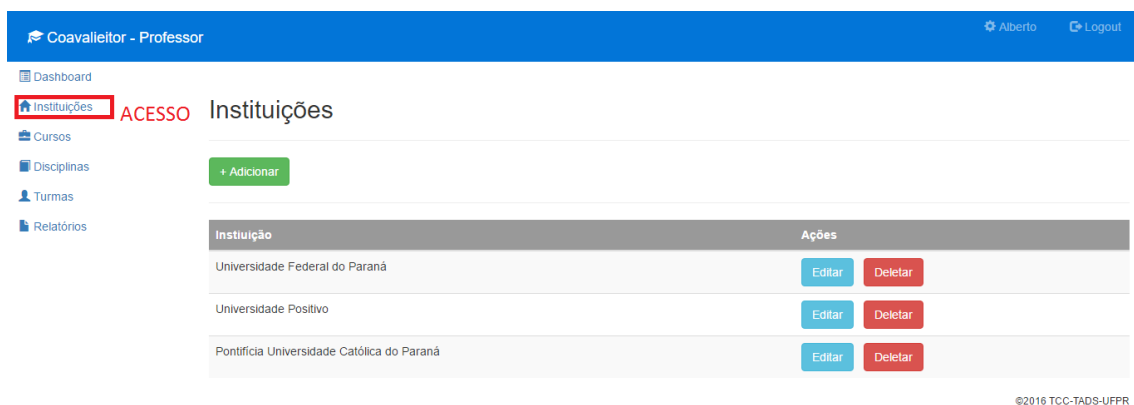


Figura 15 – Tela de Listar Instituições.

FONTE: OS AUTORES (2016).

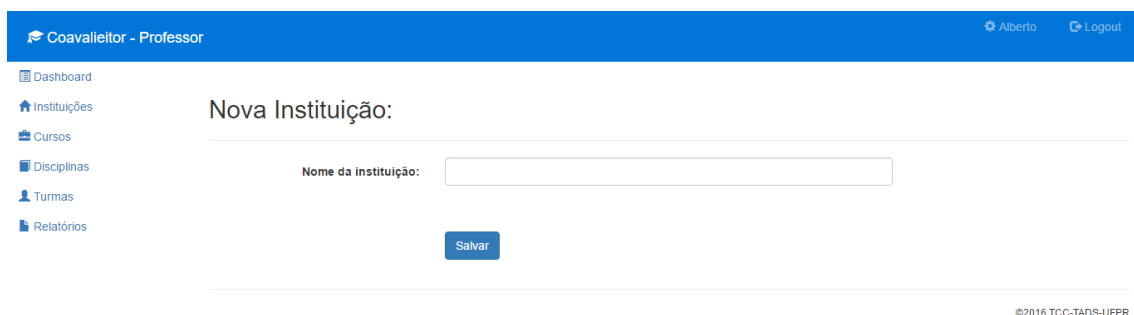


Figura 16 – Tela de Adicionar Instituição.

FONTE: OS AUTORES (2016).

Ao clicar em cursos, menu lateral esquerdo, será apresentado uma tela que busca os cursos já previamente cadastrados para uma determinada instituição (Figura 17). O professor deverá selecionar uma instituição e clicar em “Listar cursos”. É possível excluir, editar e adicionar um novo curso (Figura 18). Tanto o editar como o adicionar compartilham da mesma tela, só diferindo no título.

Figura 17 – Tela de Buscar Cursos.

FONTE: OS AUTORES (2016).

Figura 18 – Tela de Adicionar Curso.

FONTE: OS AUTORES (2016).

Ao clicar em disciplinas, menu lateral esquerdo, será apresentado uma tela que busca as disciplinas de uma determinada instituição e curso (Figura 19). O professor

deve selecionar os parâmetros para instituição e cursos e clicar em listar “Listar disciplinas”. É possível excluir, editar (Figura 20) e adicionar uma nova disciplina (Figura 21).

Coavalleitor - Professor

Dashboard

Instituições

Cursos

Disciplinas ACESSO

Turmas

Relatórios

Disciplinas

+ Adicionar

Nome da instituição: Universidade Federal do Paraná

Nome do curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Listar disciplinas

Universidade Federal do Paraná > Tecnologia em Análise e Desenvolvimento de Sistemas :

Disciplina	Ações
Algoritmo I	Editar Deletar
Banco de dados	Editar Deletar
Estrutura de dados	Editar Deletar
Gerência de projetos	Editar Deletar

©2016 TCC-TADS-UFPR

Figura 19 – Tela de Buscar Disciplinas.

FONTE: OS AUTORES (2016).

Coavalleitor - Professor

Dashboard

Instituições

Cursos

Disciplinas

Turmas

Relatórios

Editar disciplina:

Nome da disciplina: Algoritmo I

Salvar

©2016 TCC-TADS-UFPR

Figura 20 – Tela de Editar Disciplina.

FONTE: OS AUTORES (2016).

The screenshot shows a web interface for a professor. At the top, there is a blue header with the text 'Coavalleitor - Professor' on the left and 'Alberto' and 'Logout' on the right. A sidebar on the left contains a menu with the following items: 'Dashboard', 'Instituições', 'Cursos', 'Disciplinas', 'Turmas', and 'Relatórios'. The main content area is titled 'Nova disciplina:'. It contains three form fields: 'Nome da instituição:' with a dropdown menu showing 'Universidade Federal do Paraná'; 'Nome do curso:' with a dropdown menu showing 'Tecnologia em Análise e Desenvolvimento de Sistemas'; and 'Nome da disciplina:' with a text input field containing the placeholder 'Digite o nome da disciplina'. Below these fields is a blue 'Salvar' button. At the bottom right of the page, there is a small copyright notice: '©2016 TCC-TADS-UFPR'.

Figura 21 – Tela de Adicionar Disciplina.

FONTE: OS AUTORES (2016).

5.3 MÓDULO DO PROFESSOR

Essa seção demonstrará as funcionalidades disponíveis no módulo do professor. As funções apresentadas aqui são comuns tanto para o professor que tem perfil de administrador, como para o professor que não possui tal perfil.

A primeira tela que o sistema apresenta neste módulo, é o Dashboard (Figura 22), que possui estatísticas sobre avaliações, correções, alunos e etc.



Figura 22 – Dashboard.

FONTE: OS AUTORES (2016).

5.3.1 TURMA

Ao clicar em turmas, no menu lateral esquerdo, será apresentado a tela de buscar turmas (Figura 23) de um determinado professor logado, baseando-se nos parâmetros: instituição, curso e disciplina. É possível excluir, editar (Figura 24), adicionar nova turma (Figura 25) e adicionar aluno em uma turma (Figura 26).

Coavalleitor - Professor

Alberto Logout

Dashboard

Instituições

Cursos

Disciplinas

Turmas

Relatórios

Turmas

+ Adicionar

Nome da instituição: Universidade Federal do Paraná

Nome do curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Nome da disciplina: Algoritmo I

Listar turmas

Universidade Federal do Paraná > Tecnologia em Análise e Desenvolvimento de Sistemas > Algoritmo I:

Turma	Ações
Turma Matutina	Editar Deletar Alunos Avaliações
Turma Vespertina	Editar Deletar Alunos Avaliações
Turma Noturna	Editar Deletar Alunos Avaliações

©2016 TCC-TADS-UFPR

Figura 23 – Tela de Buscar Turmas.

FONTE: OS AUTORES (2016).

Coavalleitor - Professor

Alberto Logout

Dashboard

Instituições

Cursos

Disciplinas

Turmas

Relatórios

Editar Turma:

Voltar

Nome da turma: Turma Matutina

Chave de acesso: BATATA1

Salvar

©2016 TCC-TADS-UFPR

Figura 24 – Tela de Editar Turma.

FONTE: OS AUTORES (2016).

Coavalleitor - Professor

Alberto Logout

Dashboard

Instituições

Cursos

Disciplinas

Turmas

Relatórios

Nova Turma:

Nome da instituição: Universidade Federal do Paraná

Nome do curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Nome da disciplina: Algoritmo I

Nome da turma: Digite o nome da turma

Chave de acesso: Digite a chave de acesso

Salvar

©2016 TCC-TADS-UFRP

Figura 25 – Tela de Adicionar Turma.

FONTE: OS AUTORES (2016).

Coavalleitor - Professor

Alberto Logout

Dashboard

Instituições

Cursos

Disciplinas

Turmas

Relatórios

Alunos da turma: Turma Matutina

Voltar

Incluir aluno manualmente:

Incluir aluno

Aluno	E-mail	Opções
Eddie Parks	aluno75@gmail.com	Remover
Harumi Kato	aluno23@gmail.com	Remover

Figura 26 – Tela de Adicionar Aluno em uma Turma.

FONTE: OS AUTORES (2016).

5.3.2 AVALIAÇÃO

Como demonstrado na Figura 23 – Tela de Buscar Turmas, um botão laranja estará disponível para acessar Tela de listar as avaliações de uma turma (Figura 27). É possível adicionar (Figura 28), editar e excluir uma avaliação. Tanto o adicionar como o editar, compartilham da mesma tela, só diferindo o título.

Coavalleitor - Professor Alberto Logout

Dashboard
Instituições
Cursos
Disciplinas
Turmas
Relatórios

Turma Matutina - Avaliações

[Voltar](#) [+ Adicionar](#)

Avaliação	Submissão	Correção	Ações				
Sequência Inversa	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar
Converter temperatura	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar
Múltiplos 7	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar
Calcular o número h	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar
Fatorial	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar
Soma de números	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar
Exercício polimorfismo	🕒	🕒	Editar	Deletar	Listar soluções	Listar correções	Avaliar

Figura 27 – Tela de Listar avaliações de uma turma.

FONTE: OS AUTORES (2016).

The screenshot shows a web interface for creating a new evaluation. The header is blue with the text 'Coavalleitor - Professor' on the left and 'Alberto' and 'Logout' on the right. A sidebar on the left contains navigation links: Dashboard, Instituições, Cursos, Disciplinas, Turmas, and Relatórios. The main content area is titled 'Nova avaliação:' and contains several form fields: 'Nome da avaliação:' (text input), 'Descrição:' (text area), 'Requisitos adicionais:' (text area), 'Período de submissões:' (text input), 'Período de correções:' (text input), 'Número de correções por estudante:' (text input), 'Nota máxima:' (text input), and 'Critérios de correções:' (text area). A blue button labeled 'Criar avaliação' is at the bottom. The footer on the right says '©2016 TCC-TADS-UFPR'.

Figura 28 – Tela de Adicionar Avaliação.

FONTE: OS AUTORES (2016).

5.3.3 SOLUÇÃO

Como demonstrado na Figura 27 – Tela de listar Avaliações, um botão azul estará disponível para acessar a Tela de listar as soluções dos alunos (Figura 29).

Aluno:	Solução	Submetido em:	Status
Joshua Wright		02/11/2016 00:06	OK
Roger Wright		12/10/2016 00:11	OK

Figura 29 – Tela de Listar soluções dos alunos.

FONTE: OS AUTORES (2016).

Para visualizar os detalhes de uma solução de um aluno específico é preciso clicar no ícone do olho da Figura 29, localizado na coluna chamada “Solução”. A Tela de detalhe da solução (Figura 30) irá se abrir:

```

<?php

for($i=100;$i>=0;$i++){
    echo $i . '<br>';
}
?>

```

Figura 30 – Tela de Detalhe da Solução.

FONTE: OS AUTORES (2016).

5.3.4 CORREÇÃO

Como demonstrado na (Figura 27 – Tela de listar Avaliações), um link estará disponível para Tela de listar as correções dos alunos (Figura 31).

Submetido por:	Solução:	Corrigido por:	Correção:
Joshua Wright		Harumi Kato	
Joshua Wright		Lutz Gonçalves da Silva	

Figura 31 – Tela de listar Correções dos Alunos.

FONTE: OS AUTORES (2016).

Para visualizar os detalhes de uma correção de um aluno específico é preciso clicar no ícone do olho da Figura 31, localizado na coluna chamada “Correção”. A Tela de detalhe de correção (Figura 32) irá se abrir:

Corrigido Por: Maria da Silva

Data correção:
02/11/2016 00:55

Comentário:
Excelente

Nota:
100.0

Fechar

Figura 32 – Tela de Detalhe de Correção.

FONTE: OS AUTORES (2016).

5.3.5 AVALIAR ALUNO

Como demonstrado na Figura 27 – Tela de listar Avaliações, um botão laranja estará disponível para acessar a Tela de avaliar alunos (Figura 33).

Coavaleitor - Professor

Dashboard

Instituições

Cursos

Disciplinas

Turmas

Relatórios

Avaliar Estudantes - Sequência inversa

Voltar

Liberar notas

Aluno	Média das correções	Nota final do Professor	Seu feedback	Avaliar
Joshua Wright	100	100	Excelente. Continue assim.	<input checked="" type="checkbox"/>
Roger Wright	56.25	84.82	sfsddfsfsfs	<input checked="" type="checkbox"/>

Figura 33 – Tela de Avaliar Alunos.

FONTE: OS AUTORES (2016).

Para especificar a nota de um determinado aluno específico é preciso clicar no ícone do “Checked” da Figura 33, localizado na coluna chamada “Avaliar”. A Tela de especificar nota (Figura 34) irá se abrir:

Jose da Silva

Média das correções: ?

76.5

Peso da Média das Correções [0% - 100%]: ?

10

Nota final do Professor: ?

76,5

Seu Feedback: ?

Parabéns Pessoa!

Fechar Salvar alterações

Figura 34 – Tela de Especificar nota.

FONTE: OS AUTORES (2016).

5.3.6 RELATÓRIOS

Ao clicar em “Relatórios”, no menu lateral esquerdo, a Tela de buscar relatórios (Figura 35) é apresentada. O professor seleciona os parâmetros pertinentes e clica em “Gerar relatório” e um PDF será gerado.

Universidade Federal do Paraná	
Professor(a): Welyngton Disciplina: Banco de dados	
Aluno	Turma
Alfredo Caballero	Turma BD
Allan	Turma BD

Figura 35 – PDF alunos turma / Tela de Buscar Relatórios.

FONTES: OS AUTORES (2016).

5.4 MÓDULO DO ALUNO

Essa seção apresenta todas as funcionalidades disponíveis para disponíveis para os usuários do sistema com perfil de aluno.

Ao entrar no sistema, é apresentado para o aluno a Tela de listar turmas (Figura 36), que lista todas as informações pertinentes das turmas registradas no sistema.

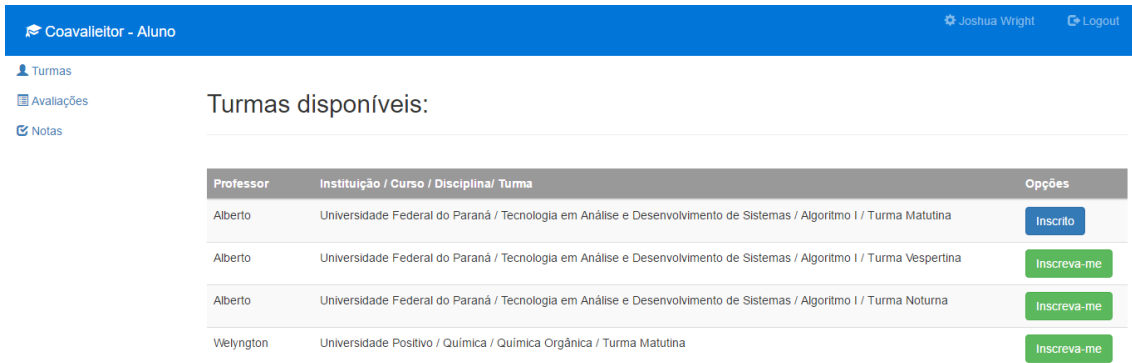


Figura 36 – Tela de Listar Turmas.

FONTE: OS AUTORES (2016).

Para se inscrever em uma turma o aluno precisa clicar no link “Inscriva-me” da Figura 39, fazendo isso a Tela de inscrição (Figura 37) se abrirá.

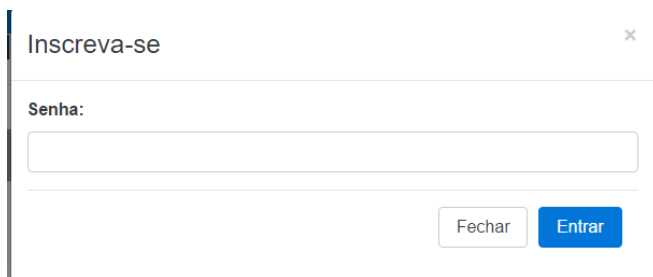


Figura 37 – Tela de Inscrição.

FONTE: OS AUTORES (2016).

5.4.1 AVALIAÇÃO

Ao clicar em “Avaliações”, no menu lateral esquerdo, a Tela de buscar avaliações é apresentada (Figura 38). O aluno seleciona a turma desejada e clica em “Checar Avaliações”, o sistema direciona para Tela de Avaliações por turma (Figura 39).

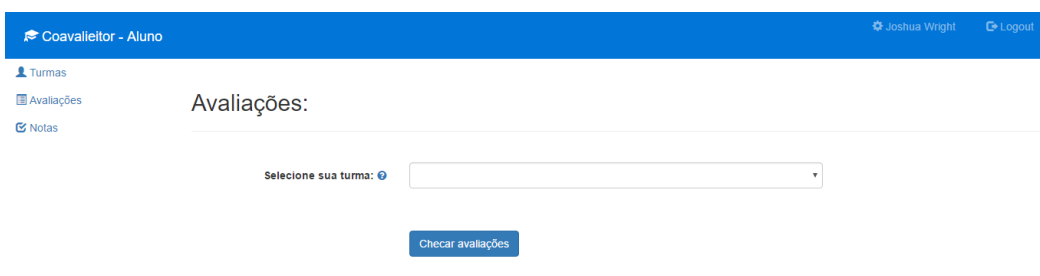


Figura 38 – Tela de Buscar Avaliações.

FONTE: OS AUTORES (2016).

Coavalleitor - Aluno Joshua Wright Logout

Turmas
Avaliações
Notas

Turma Matutina - Avaliações:

Submeter soluções:

Avaliação	Resolver até:	Resolvido em:	Opção
Sequência Inversa	🕒	02/11/2016 00:06	Submetido
Calcular o número h	🕒		Resolver

Corrigir soluções:

Avaliação	Corrigir até:	Opção
Soma de números	🕒	Corrigir

Pontuação e comentários do Professor:

Avaliação	Nota Prof.	FeedBack Prof.	Corrigido em:
Sequência Inversa	100,00	Excelente. Continue assim.	05/11/2016 16:36
Converter temperatura	60,00	Bom trabalho.	05/11/2016 21:12

©2016 TCC-TADS-UFPR

Figura 39 – Tela de Avaliações por Turma.

FONTE: OS AUTORES (2016).

5.4.2 SOLUÇÃO

A avaliação precisa estar na fase de “submissões” para que seja possível enviar uma solução. Para submeter a solução o aluno precisa clicar em no botão “Resolver” da Figura 39. Então o sistema irá apresentar a Tela de resolver avaliação (Figura 40).

Coavalleitor - Aluno

Joshua Wright Logout

Turmas

Avaliações

Notas

Resolver avaliação - Calcular o número h

Problema:

Sendo $h = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$, construa um algoritmo para calcular o número h, sendo o número inteiro N fornecido pelo usuário. Seu algoritmo deve garantir que apenas um valor maior do que zero seja aceito como entrada.

*Linguagem PHP | C#

Sua resposta:

Submeter sua resposta

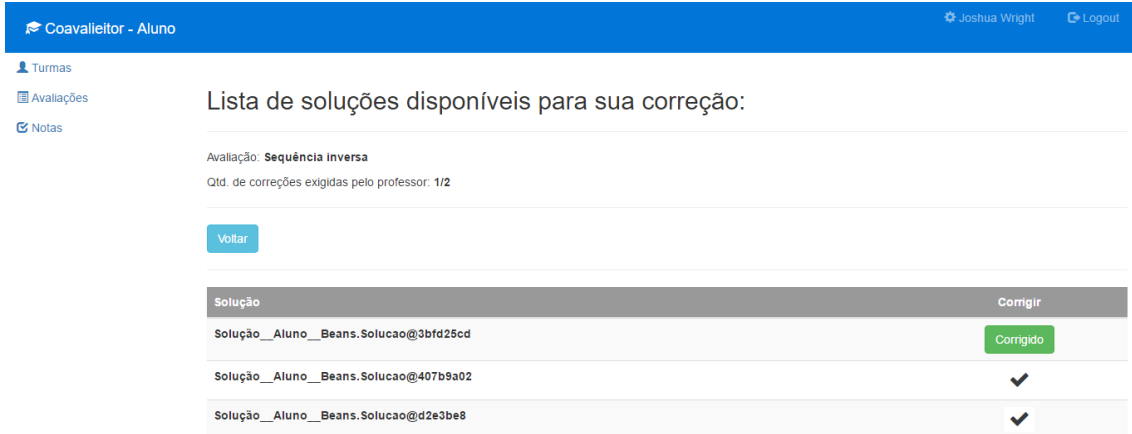
©2016 TCC-TADS-UFPR

Figura 40 – Tela de Resolver Avaliação.

FONTE: OS AUTORES (2016).

5.4.3 CORREÇÃO

A avaliação precisa estar na fase de “correções” para que seja possível efetuar correções das soluções de outros alunos. Para corrigir uma solução o aluno precisa clicar em no botão “Corrigir” da Figura 39. Então o sistema irá apresentar a Tela de listar soluções para correção (Figura 41). Na Figura 41 também é possível verificar o número de correções exigidas pelo professor.



The screenshot shows a web interface for a student named Joshua Wright. The header includes the name 'Coavalleitor - Aluno' and a 'Logout' button. A sidebar on the left contains navigation links for 'Turmas', 'Avaliações', and 'Notas'. The main content area is titled 'Lista de soluções disponíveis para sua correção:'. Below this title, it specifies the evaluation as 'Sequência Inversa' and that 1/2 corrections are required. A 'Voltar' button is present. The core of the interface is a table with two columns: 'Solução' and 'Corrigir'. The table lists three solutions, each with a unique ID and a corresponding 'Corrigir' button or checkmark.

Solução	Corrigir
Solução__Aluno__Beans.Solucao@3bfd25cd	Corrigido
Solução__Aluno__Beans.Solucao@407b9a02	✓
Solução__Aluno__Beans.Solucao@d2e3be8	✓

Figura 41 – Tela de Listar Soluções para Correção.

FONTE: OS AUTORES (2016).

Para enviar uma correção de uma solução, o aluno precisa clicar no botão “checked” para que o sistema redirecione para Tela de corrigir avaliação (Figura 42).

Coavalleitor - Aluno

Joshua Wright Logout

Turmas

Avaliações

Notas

Correção de solução - Sequência inversa

[Voltar](#)

Problema:

Escreva um algoritmo que coloque os números de 1 a 100 na tela na ordem inversa (começando em 100 e terminando em 1).

Requisitos adicionais:

- *Linguagem C++
- *Utilizar estrutura de repetição 'for'

Critérios de correção:

- 20 - Legibilidade do código.
- 20 - Elegância do código.
- 20 - Lógica de programação.
- 20 - Complexidade do algoritmo.
- 20 - Indentação.

Resposta aluno (n):

```
#include <iostream>
using namespace std;

int main() {
    for(int i=100; i>=1; i--){
        cout << i << "\n";
    }
    return 0;
}
```

Seu comentário:

Sua nota:

[Submeter sua correção](#)

©2016 TCC-TADS-UFFR

Figura 42 – Tela de Corrigir Avaliação.

FONTE: OS AUTORES (2016).

5.4.4 NOTAS

Ao clicar em “Notas”, no menu lateral esquerdo, a Tela de buscar notas será apresentada (Figura 43). O aluno seleciona a turma desejada e clica em “Checar Notas”, o sistema irá gerar um PDF com as notas do aluno (Figura 44).

Figura 43 – Tela de Buscar Notas.

FONTE: OS AUTORES (2016).

Avaliação	Feedback	Nota final	Corrigido em
Sequência inversa	Excelente. Continue assim.	100.0	05/11/16 16:36
Converter temperatura	Bom trabalho.	60.0	05/11/16 21:12

Figura 44 – PDF Notas Aluno.

FONTE: OS AUTORES (2016).

Foram apresentadas todas as funcionalidades do sistema Coavaleitor, tanto no módulo do professor como no módulo do aluno. Com essa apresentação do *software*, acreditamos que o usuário não terá tantas dificuldades para manusear o sistema.

6 CONSIDERAÇÕES FINAIS

Acreditamos que o desenvolvimento deste trabalho de conclusão de curso conseguiu atender todos os requisitos exigidos pelo cliente, o Prof. Dr. Alexander Robert Kutze. O sistema web desenvolvido apresenta-se como uma ótima ferramenta de suporte para o processo de avaliação pelos pares. Uma vez que facilita, devido a informatização, as correções de soluções entre os alunos.

A equipe, de um modo geral, percebeu uma grande melhora nas habilidades pertinentes a formação acadêmica/profissional, tais como: modelagem de dados, UML, programação em JAVA WEB e gerência de projetos. Isso foi decorrente da intensa pesquisa realizada para desenvolver um *software* com alto padrão de qualidade. Além disso, não podemos deixar de citar o aprendizado adquirido em áreas das quais a equipe não possuía um conhecimento prévio, como: pedagogia, o processo de avaliação e aprendizagem de alunos.

Um dos grandes desafios de executar esse projeto foi, sem sombra de dúvidas, o curto prazo para entrega final do sistema. Todos os integrantes da equipe tiveram que se dedicar ao máximo para conseguir elaborar uma documentação coesa com a implementação do sistema. Foi preciso aplicar metodologias de desenvolvimento ágil, como SCRUM, para conseguir entregar o projeto na data estabelecida no cronograma.

Com relação a implementações futuras, acreditamos que seja importante (1) possibilitar o *upload* de arquivos no sistema, para que assim seja possível aprofundar o processo de correções e soluções submetidas pelos alunos. Isso ajudaria estudantes de diferentes cursos e necessidades aproveitarem as funcionalidades do sistema ao máximo; (2) permitir notificações por e-mail, pois auxiliaria o professor no processo de comunicação com seus estudantes; (3) disponibilizar um recurso para o professor escolher o algoritmo de distribuição de soluções; (4) melhorar o editor de respostas e o filtro de turmas no módulo do aluno.

Por fim, confiamos que o sistema alcançou seu propósito inicial. Muitas melhorias podem ser feitas porém, dentro das nossas restrições de tempo, consideramos que conseguimos entregar um projeto completo e funcional, de acordo com os requisitos elencados pelo cliente, que poderá futuramente atender vários

professores e alunos, e por consequência, poderá ajudar na disseminação da Avaliação por Pares.

REFERÊNCIAS

- AGÊNCIA BRASIL, **MEC divulga resultados preliminares do Censo Escolar 2015**, Disponível em <<http://agenciabrasil.ebc.com.br/educacao/noticia/2015-10/mec-divulga-resultados-preliminares-do-censo-escolar-2015>> - Acesso em 18 de setembro de 2016
- BARNES, Douglas. **Active Learning. Leeds University TVEI Support Project**, 1989
- BARREIRO, Aguida Celina de Méo, Nascimento, Otaciro Rangel **A Participação de Alunos na Correção das Provas de uma Disciplina no Ensino Superior**, Cad.Cat.Ens.Fís., v.17, n.3: p.295-306, dez.2000. 295 Disponível em <<https://periodicos.ufsc.br/index.php/fisica/article/download/6762/6230>> Acesso em 12 de novembro de 2016
- BOOCH, Grady. **UML : guia do usuário : o mais avançado tutorial sobre Unified Modeling Language (UML), elaborado pelos próprios criadores da linguagem**. Rio de Janeiro : Campus : Elsevier, c2012.
- BROOKE, Georgia; ANDRADE, Heidi. **Peer Assessment**. Disponível em: <http://www.jff.org/sites/default/files/publications/materials/2_SATC_AssessTools_PeerAssessment_042913.pdf>. Acesso em: 10 de novembro de 2016
- CARLESS, David **Excellence in University Assessment: Learning from award-winning practice**, Routledge, Nova York, 2015, Disponível em <https://books.google.com.hk/books?id=_WShCAAQBAJ&printsec=frontcover#v=onepage&q&f=false> Acesso em 15 de setembro de 2016
- CORNELL University, Center for Teaching Excellence, **Assessing Student Learning**, Disponível em: <<https://www.cte.cornell.edu/teaching-ideas/assessing-student-learning/index.html>> Acesso em 20 de setembro de 2016
- COSTA, PEDRO. **Conceitos: testes de caixa branca e caixa preta, 2014**. Disponível em:<<http://crowdtest.me/teste-caixa-branca-caixa-preta/>> Acesso em: 20 de outubro de 2016.
- CRUZ, Tadeu. **Workflow, A tecnologia que vai revolucionar processos**. São Paulo: Editora Atlas, 2000.
- CRUZ, Tadeu. **Workflow II, A tecnologia que revolucionou processos**. São Paulo: Editora Atlas, 2004.
- SILVA, Delcio Barros da, **As Principais Tendências Pedagógicas na Prática Escolar Brasileira e seus Pressupostos de Aprendizagem** Disponível em

<http://coral.ufsm.br/lec/01_00/DelcioL&C3.htm> Acesso em 20 de setembro de 2016

DICIONÁRIO DE ETIMOLOGIA, disponível em:

<<http://www.etymonline.com/index.php?term=pedagogue>> - Acesso em 13 de novembro de 2016

DULLO E. FREIRE, Paulo, **O testemunho e a pedagogia católica: a ação histórica contra o fatalismo**, Rev. bras. Ci. Soc. vol.29 no.85 São Paulo Junho de 2014, Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-69092014000200004> Acesso em 20 de setembro de 2016

Eusébio André Machado , Marta Abelha , Carlos Barreira & Ana Salgueiro, **Avaliação pelos Pares Percurso Normativo da Avaliação do Desempenho Docente em Portugal**, 2012 - Disponível em <<http://iduc.uc.pt/index.php/rppedagogia/article/viewFile/1783/1132>> - Acesso em 30 de outubro de 2016

FREIRE, Paulo. 1981. **Pedagogia do Oprimido**. 10ª ed. Rio de Janeiro: Paz e Terra.

_____. 1994. **Pedagogia da Esperança: Um reencontro com a Pedagogia do oprimido**. 3ª ed. Rio de Janeiro: Paz e Terra.

_____. 1996. **Pedagogia da Autonomia: Saberes Necessários à Prática Educativa**. 2ª ed. São Paulo: Paz e Terra. (Coleção Leitura)

_____. 2000a. **A Educação na Cidade**. 4.ed. São Paulo: Cortez.

_____. 2000b. **Pedagogia da Indignação: cartas pedagógicas e outros escritos**. São Paulo: Editora UNESP.

_____. 2001. **Política e Educação**. 5ª ed. São Paulo: Cortez. (Coleção Questões da Nossa Época; v.23)

FLOWER, Martin. **UML Essencial: Um breve guia para linguagem-padrão de modelagem de objetos**. São Paulo: Bookman, 2007.

FOUNDATION COALITION, **Peer Assessment and Peer Evaluation**, 2002
https://www.uwstout.edu/soe/profdev/resources/upload/2002peer_assessment.pdf
GILLEANES, T.A Guedes. **UML 2 - Uma Abordagem Prática**. São Paulo, 2010.

GOLDBERG, A. L., & Pedulla, J. J. (2002). **Performance differences according to test mode and computer familiarity on a practice graduate record exam**. **Educational and Psychological Measurement**, Journal of Special Education Technology December 1, 2014 29: 1-13, Disponível em

<<http://epm.sagepub.com/content/62/6/1053.short>> Acesso em 17 de setembro de 2016

HANRAHAN, STEPHANIE J. e ISAACS, GEOFF **Assessing Self- and Peer-assessment: the students' views**, Higher Education Research & Development, Vol. 20, No. 1, 2001 Disponível em <https://pes.concordia.ca/docs/assessing_self.pdf> Acesso em 18 de setembro de 2016

HELGA M. e Jorge P. **Os contributos da coavaliação entre pares, através do feedback, na regulação de aprendizagens**, Disponível em <<https://comum.rcaap.pt/bitstream/10400.26/6842/1/Os%20contributos%20da%20coavalia%C3%A7%C3%A3o%20entre%20pares%20e%20regula%C3%A7%C3%A3o%20das%20aprendizagens.pdf>> - Acesso em 20 de outubro de 2016

HENSLEY, Kiersten Kenning, **Examining the effects of paper-based and computer-based modes of assessment on mathematics curriculum-based measurement**, 2015, Disponível em <<http://ir.uiowa.edu/etd/1627>> Acesso em 15 de setembro de 2016

KANT I., **Crítica da Razão Pura**, Disponível em <<http://www.ebooksbrasil.org/eLibris/critica.html>>, Ebook, Acrópolis, 1781

LIBÂNIO, José C.; **Democratização da Escola Pública a Pedagogia Crítico Social dos Conteúdos**. São Paulo, Loyola, 15ª edição, 1985.

LUCKESKI, Cipriano Carlos, **Avaliação da Aprendizagem Escolar**, Editora Cortez, 19ª Edição, São Paulo, 2005

MARTINS, José Carlos Cordeiro. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML**. Rio de Janeiro: Brasport, 2010

MARTINS, José Carlos Cordeiro **Técnicas para gerenciamento de projetos de software / José Carlos Cordeiro Martins**. Rio de Janeiro : Brasport, 2007.

MARTINS, José Carlos Cordeiro **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**. Rio de Janeiro : Brasport, 2007.

MINISTÉRIO DA EDUCAÇÃO E CULTURA (MEC) - **Tecnologias na Escola**, Disponível em: <<http://portal.mec.gov.br/seed/arquivos/pdf/2sf.pdf>> - Acesso em 02 de outubro de 2016

NASSI-CALÒ, Lilian **Avaliação por pares: modalidades, prós e contras**, 2015 Disponível em <<http://blog.scielo.org/blog/2015/03/27/avaliacao-por-pares-modalidades-pros-e-contras/#.WAV6VFQrLIU>> Acesso em 24 de novembro de 2016

NETO, A. C. D.; Dias, C. **Introdução a teste de software Engenharia de Software Magazine**, v. 1, 2008.

The Organisation for Economic Co-operation and Development (**OECD**), **How much time do teachers spend teaching?**, Disponível em <[http://www.oecd.org/edu/EAG2014-Indicator%20D4%20\(eng\).pdf](http://www.oecd.org/edu/EAG2014-Indicator%20D4%20(eng).pdf)> - Acesso em 13 de novembro de 2016

OLIVEIRA, Marta K. Vygotsky: **Aprendizado e Desenvolvimento, um processo sócio-histórico**. Ed. Scipione, 1993.

OLIVEIRA, Wilson José de. **Segurança da informação : técnicas e soluções**. Florianópolis : Visual Books, 2001

PAVLOV, **Psychology as the Behaviorist views it**, 1913

PAVLOV, I.P. **Resposta de um fisiologista aos psicólogos**. 1932 in: ALVAREZ, Antonio C. (org.) Textos de Ivan Pavlov, fisiologia e psicologia. Lisboa, Estudos Cor, 1976

Peer Assessment, Em: Wikipédia , Disponível em <https://en.wikipedia.org/wiki/Peer_assessment#cite_note-2> - Acesso em 15 de setembro de 2016

PEREIRA, Marta Maximo e Andrade, Viviane Abreu **Autoavaliação como Estratégia para o Desenvolvimento Investigações em Ensino de Ciências – V17(3)**, pp. 663-674, 2012 Disponível em <http://www.if.ufrgs.br/ienci/artigos/Artigo_ID310/v17_n3_a2012.pdf> Acesso em 21 de setembro de 2016

PIAGET, Jean. **Psicologia e pedagogia**. Tradução de Dirceu Accioly Lindoso e Rosa Maria Ribeiro da Silva. São Paulo e Rio de Janeiro: Editora Forense, 1970

PIAGET, J., Gruber, H. (Ed.), & Voneche, J. J. (Ed.). **The essential Piaget** (100th Anniversary Ed.). New York: Jason Aronson.

PINHEIRO, José Q. **Anonimato e avaliação cega por pares**, Estud. psicol. (Natal) vol.9 no.2 Natal Maio/Agosto. 2004, Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-294X2004000200001,

RADAELLI M. R. R. e Fruet F. S. O. **Processo Ensino-Aprendizagem e Interação Entre Alunos e Professores Potencializados pelas Tecnologias da Informação e da Comunicação**, 2011, Disponível em: <<http://seer.ufrgs.br/renote/article/viewFile/21969/12742>> - Acesso em 05 de outubro de 2016

RÉGNIERA, Jean-Claude, **Auto-Avaliação na Prática Pedagógica Revista Diálogo Educacional** - v. 3 - n.6 - p.53-68 - maio/agosto, 2002 1 Disponível em:

<www2.pucpr.br/reol/index.php/dialogo?dd99=pdf&dd1=688> Acesso em 07 de outubro de 2016

REISSWITZ, Flavia. **Análise de Sistemas v.4: Processo de desenvolvimento de software**. Rio de Janeiro: Clube dos Autores,200

RIZZONI, Gisele **A Sala de Aula Sob o Olhar do Construtivismo Piagetiano: Perspectivas e Implicações**, 2010 Disponível em <https://www.ucs.br/site/midia/arquivos/Construtivismo_Piagetiano.pdf> Acesso em 19 de setembro de 2016

JANCO, Robert L. **The use of computer in assessment**, Strategies for Assessment from the 1995 and 2002 COMSEP Curricula Disponível em: <<https://www.comsep.org/educationalresources/viewAssessment.cfm?id=16>> Acesso em 17 de setembro de 2016

SADLER E GOOD, **The Impact of Self- and Peer-Grading on Student Learning**, 2006 Disponível em: <<https://www.cfa.harvard.edu/sed/staff/Sadler/articles/Sadler%20and%20Good%20E.A.pdf>> - Acesso em 15 de setembro de 2016

SANTOS, J. C. **A participação ativa e efetiva do aluno no processo ensino-aprendizagem como condição fundamental para a construção do conhecimento**, Disponível em <<http://hdl.handle.net/10183/2313>> - Acesso em 13 de novembro de 2016

SCHACH, Stephen R. **Engenharia de Software: Os Paradigmas Clássico & Orientado a Objetos**. Porto Alegre: AMGH,2010.

SCHWABER, Ken - **Agile software development with Scrum / Ken Schwaber and Mike Beedle**. Upper Saddle River, NJ : Prentice Hall, 2002.

VAIDERGORN, J. **Ensino religioso, Uma herança do Autoritarismo**, 2008 - Disponível em <<http://www.scielo.br/pdf/ccedes/v28n76/a07v2876.pdf>> - Acesso em 27 de outubro de 2016

VYGOTSKY, Lev (1987), **Pensamento e Linguagem**, SP: Martins Fontes (tradução da versão resumida norte-americana).

VIGOTSKY, Leiv S. **Os métodos de investigação reflexológicos e psicológicos**, 1924 in: VIGOTSKY, Leiv S. Teoria e método em psicologia. SP, Martins Fontes, 2004

WATSON, John B. **Psychology as the behaviorist views it** Psychological Review, Vol 20(2), Mar 1913, 158-177. - Disponível em <<http://dx.doi.org/10.1037/h0074428>>

- Acesso em 30 de outubro de 2016

YASNITSKY, A. **O Vigotski que nós (não) conhecemos: os principais trabalhos de Vigotski e a cronologia de sua composição**, PsyAnima, Dubna Psychological Journal (RU) 4 (4): 62–70, 2011

ZANON, Denise Puglia e Althaus, Maiza Margraf **Instrumentos de Avaliação na Prática Pedagógica Universitária**, 2008, Disponível em <<http://lcc-ead.nutes.ufrj.br/constructore/objetos/InstrumentosdeAvaliacao.pdf>> - Acesso em 20 de outubro de 2016

APÊNDICE 1 – ENTREVISTA

Entrevista 1 - 15/08/2016

Projeto: Fórum educacional

Stackholder entrevistado: Prof. Alex Kutzke.

Orientação: Andreia de Jesus

Entrevistadores: Alberto Kato, Ivonete Roque e Welyngton Del Prá.

Levantamento inicial - Conhecendo o domínio do problema

1) Qual problema que será resolvido com esse projeto a ser desenvolvido? (cuidado para não descrever uma solução, mas sim uma oportunidade ou problema)

R: Automatização do processo de correção de exercícios entre os alunos como reforço pedagógico.

2) Como seria a solução para esse problema? (Descrever a solução da forma mais detalhada possível)

R: Construção de uma ferramenta que ajude nesse processo.

3) Quais são as vantagens em resolver esse problema? (O que você ganha no caso desse projeto ser implementado com sucesso? Economia de tempo?, lucratividade?, eficiência?, eficácia?, etc)

R: Mais um instrumento pedagógico.

4) Cite os principais objetivos do sistema?

R: Potencializar o aprendizado e facilitar a aplicação dessa técnica.

Critérios para definir que o sistema alcançou o objectivo proposto (Sucesso)

5) Qual o principal critério que deve ser alcançado para que se possa dizer que o sistema foi implementado com sucesso?

R: O Processo deve ser simples , automatizado e fácil gerenciamento.

6) Será necessário realizar algum treinamento de usuário para que o sistema funcione como planejado e conseqüentemente alcance o sucesso?

R: Ser intuitivo já basta.

Identificando as funcionalidades do sistemas

7) Cite as principais funcionalidades do sistemas? (Exemplo: Cadastrar professores,etc)

R: Cadastro de atividades. Correção. Feedback etc.

8) É necessário o desenvolvimento de relatórios? Caso positivo, cite o nome dos relatórios e as informações que devem ser apresentadas.

R: Relatórios para professores. Fica a critério da equipe de desenvolvimento.

Identificando usuários e suas devidas funções no sistema

9) Quem irá utilizar o sistema? (Liste todos os usuários do sistema, incluindo, caso necessário, administradores)

R: Professores e Alunos.

10) Qual é o nível de conhecimento dos usuários (Experiência em lidar com sistemas informatizados - Expertise)?

R: Básico.

11) Qual o papel (função) de cada usuário listado no sistema? (Exemplo: Professor: Realizar cadastro de questões, etc)

R: Professor pode : criar atividades, verificar correções, verificar soluções. Aluno pode: submeter soluções e corrigir atividades de outros alunos.

12) É necessário a criação de módulos para os diferentes usuários?

R: Aluno e Professor.

Tecnologias e critérios específicos (requisitos não funcionais)

13) Existe alguma exigência em relação a tecnologia utilizadas pelos desenvolvedores? (Exemplo: linguagem de programação, plataforma, Sistema operacional, Banco de dados, etc)

R: Não, contanto que seja gratuito e na plataforma WEB.

14) Em relação aos critérios de : disponibilidade, segurança, confiabilidade, usabilidade, desempenho e manutenção. Existe alguma exigência específica? Caso positivo, cite o critério e descreva de forma objetiva seu requerimento.

R: No momento não.

APÊNDICE 2 – PROTOTIPAÇÃO DE TELAS

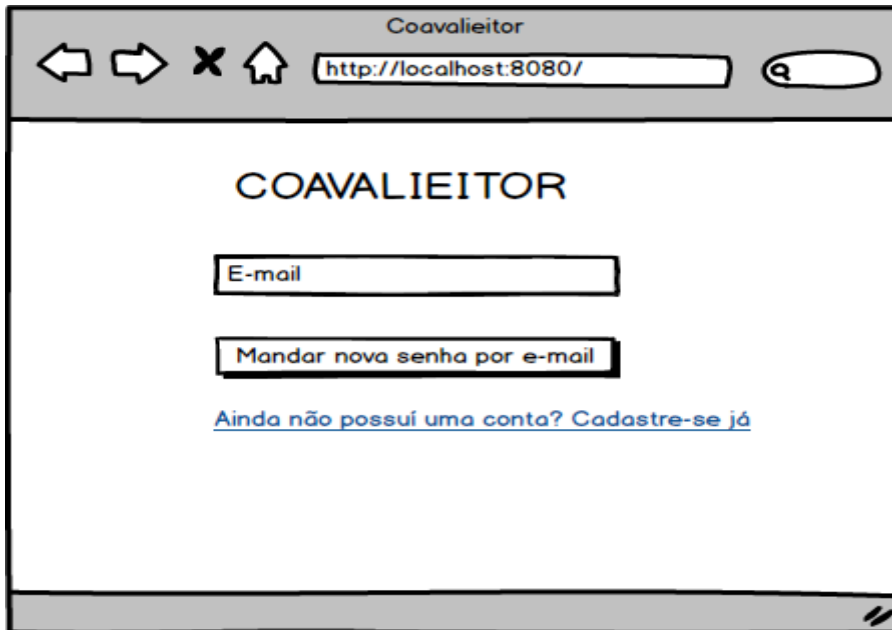


Figura 45 – Protótipo “Recuperar senha”.

FONTE: OS AUTORES (2016).



Figura 46 – Protótipo “Login”.

FONTE: OS AUTORES (2016).

Coavaleitor

http://localhost:8080/Coavaleitor/

COAVALEITOR

Increva-se já

Nome Completo

Email

Sou Professor Sou estudante

Cadastrar

[Já possui uma conta? Acesse aqui](#)

Figura 47 – Protótipo “Cadastro de usuário”.

FONTE: OS AUTORES (2016).

Coavaleitor

http://localhost:8080/Coavaleitor/

- Dashboard
- Relatórios
- Avaliações
- Turmas
- Cursos
- Instituição
- Professores
- Alunos

Nome:

Figura 48 – Protótipo “Instituições”.

FONTE: OS AUTORES(2016).

Coavaleitor

http://localhost:8080/Coavaleitor/

Cadastro de Curso

Nome

Instituição... ▾
UFPR
UTFPR

Coordenador... ▾
Andreia
Alex

Salvar

Figura 49 – Protótipo “Cadastrar curso”.

FONTE: OS AUTORES (2016).

Coavaleitor

http://localhost:8080/Coavaleitor/

Cadastro de Disciplina

Nome

Instituição...
UFPR
UTFPR

Curso...
TADS
Me. em Bioinformática

Prof. Responsável...
Andreia
Alex

Salvar

Figura 50 – Protótipo “Cadastrar disciplina”.

FONTE: OS AUTORES (2016).

Coavaleitor

← → × 🏠 🔍

Cadastro de Turma

Prof. Responsável... ▾

Disciplina... ▾

Nome	e-mail	Curso	Turmas	Avaliações	Correções	Remove
alberto	albertinho@bol.com.br	TADS	Turmas }	Avaliações }	Correções }	<input checked="" type="checkbox"/>
Welyngton	wel@uol.com.br	TADS	Turmas }	Avaliações }	Correções }	<input checked="" type="checkbox"/>
Nety	roque@oi.com.br	TADS	Turmas }	Avaliações }	Correções }	<input checked="" type="checkbox"/>

Figura 51 – Protótipo “Cadastrar turma”.

FONTE: OS AUTORES (2016).

Coavaleitor

← → ✕ 🏠 🔍

Criar Tarefa (Avaliação) - Menu Professor

Nome tarefa:

Turma: ▼

Descrição:

Faça um programa que calcule o fatorial de um número n de maneira recursiva e iterativa. Comente sobre as vantagens e desvantagens de cada solução.

Requisitos adicionais:

*Todas a soluções devem ser submetidas na linguagem C ANSI.

*Na solução iterativa , resolver com DO-WHILE

Período de submissões: 📅 até: 📅

Período de correções: 📅 até: 📅

Num. Correções por estudantes:

Nota máxima:

Critérios de correção:

0 - O estudante não submeteu solução alguma.
 1 - O estudante resolveu de forma incompleta somente uma solução.
 2 - O estudante resolveu de forma incompleta as duas soluções.
 3 - O estudante desenvolveu 2 soluções, mas não atendeu aos requisitos adicionais.
 4 - O estudante desenvolveu 2 soluções e atendeu a todos os requisitos.
 5 - O estudante resolveu os problemas de forma elegante.

Figura 52 – Protótipo “Criar avaliação”.

FONTE: OS AUTORES (2016).

Coavaleitor

http://localhost:8080/Coavaleitor/

Listar alunos - Módulo professor

Nome

Curso...

Turma...

Nome	e-mail	Cursc	Turmas	Avaliações	Correções
Alberto	albertinho@bol.com.br	TADS	Turmas	Avaliações	Correções }
Welyngton	wel@uol.com.br	TADS	Turmas	Avaliações	Correções }
Nety	roque@oi.com.br	TADS	Turmas	Avaliações	Correções }

Figura 53 – Protótipo “Listar alunos”.

FONTE: OS AUTORES (2016).

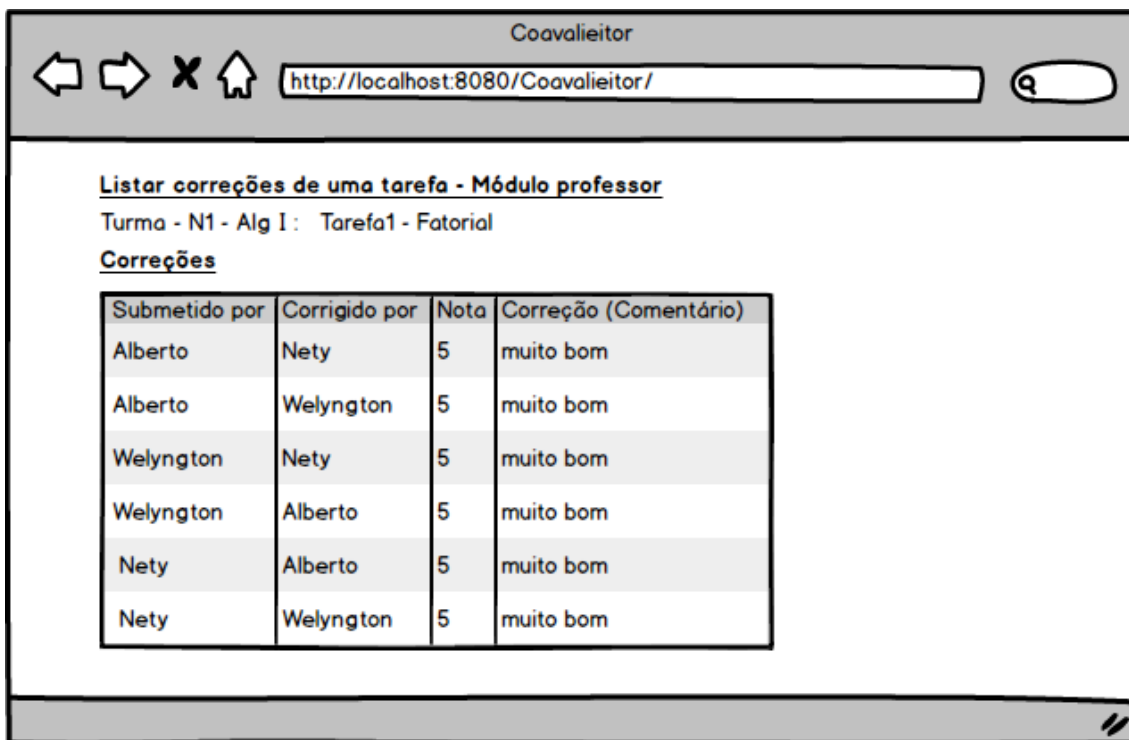


Figura 54 – Protótipo “Listar correções de uma tarefa”.

FONTE: OS AUTORES (2016).

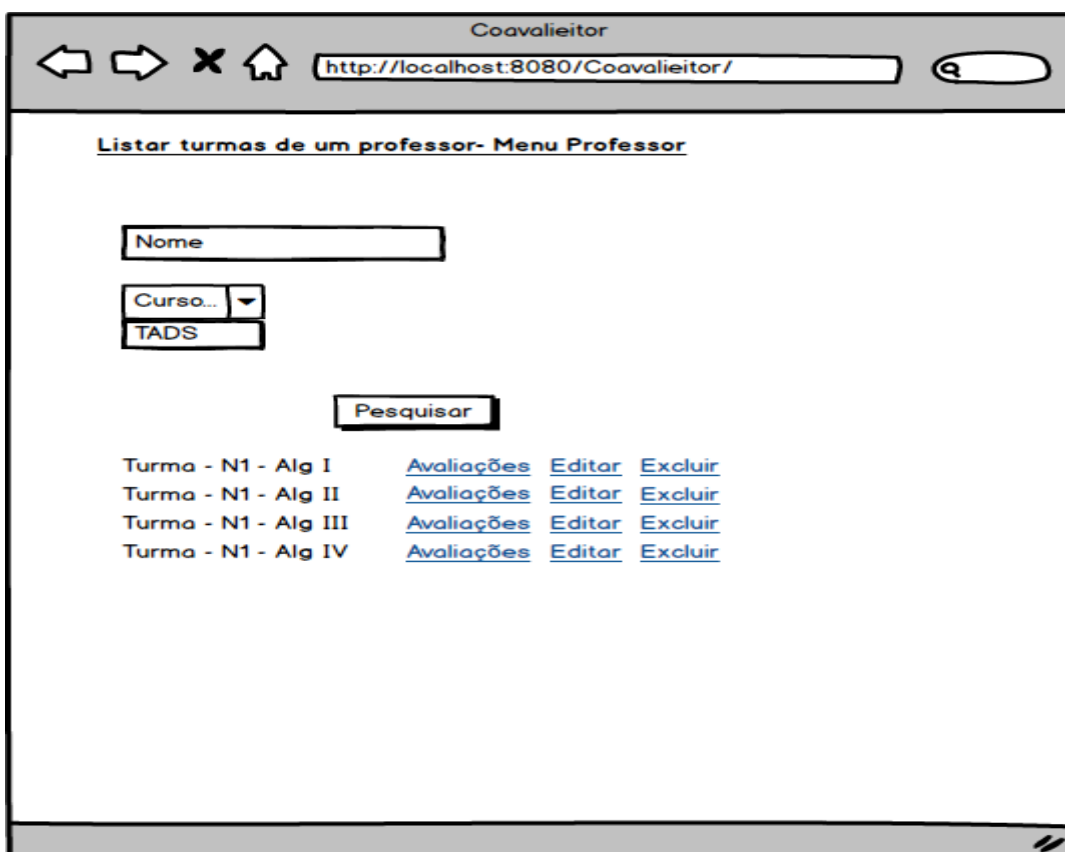


Figura 55 – Protótipo “Lista turmas de um professor”.

FONTE: OS AUTORES (2016).

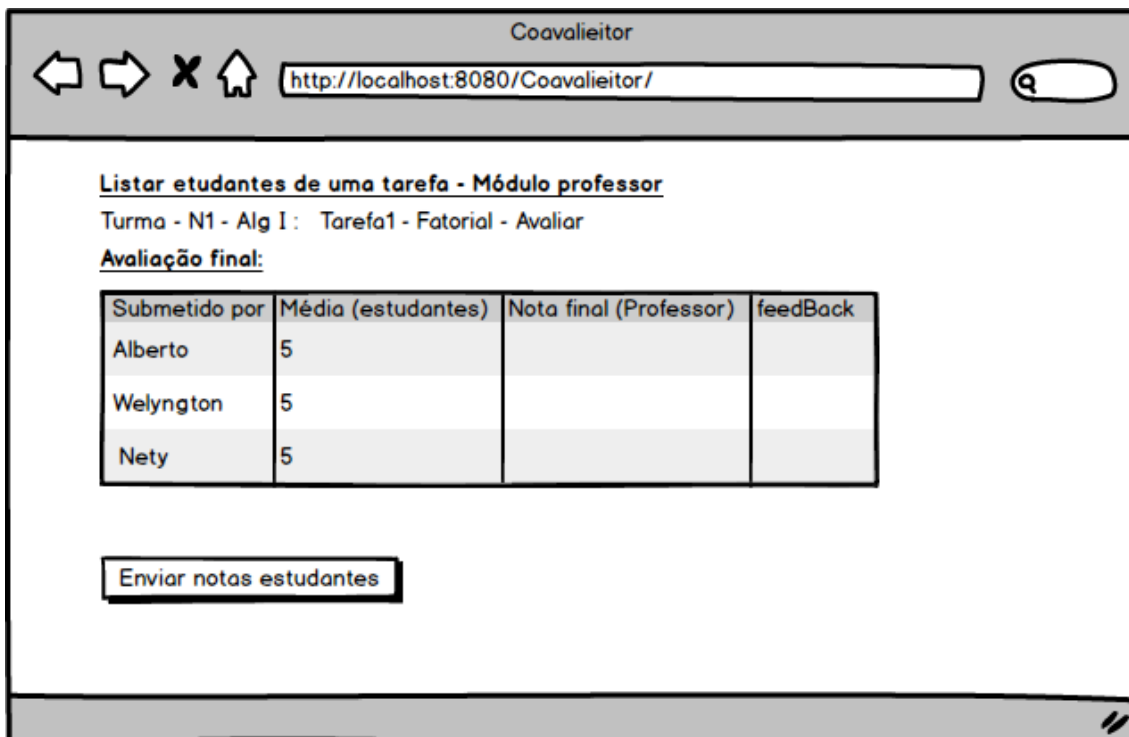


Figura 56 – Protótipo “Listar estudantes de uma tarefa”

FONTE: OS AUTORES (2016).

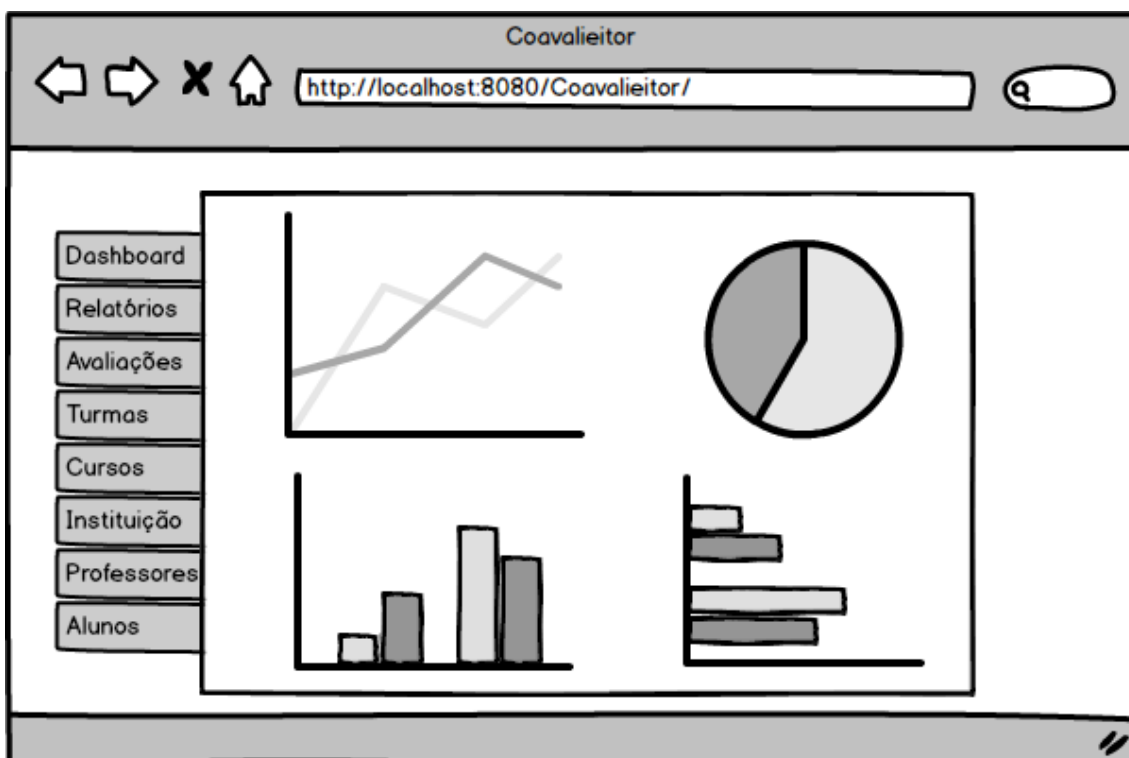


Figura 57 – Protótipo “Dashboard”.

FONTE: OS AUTORES (2016).

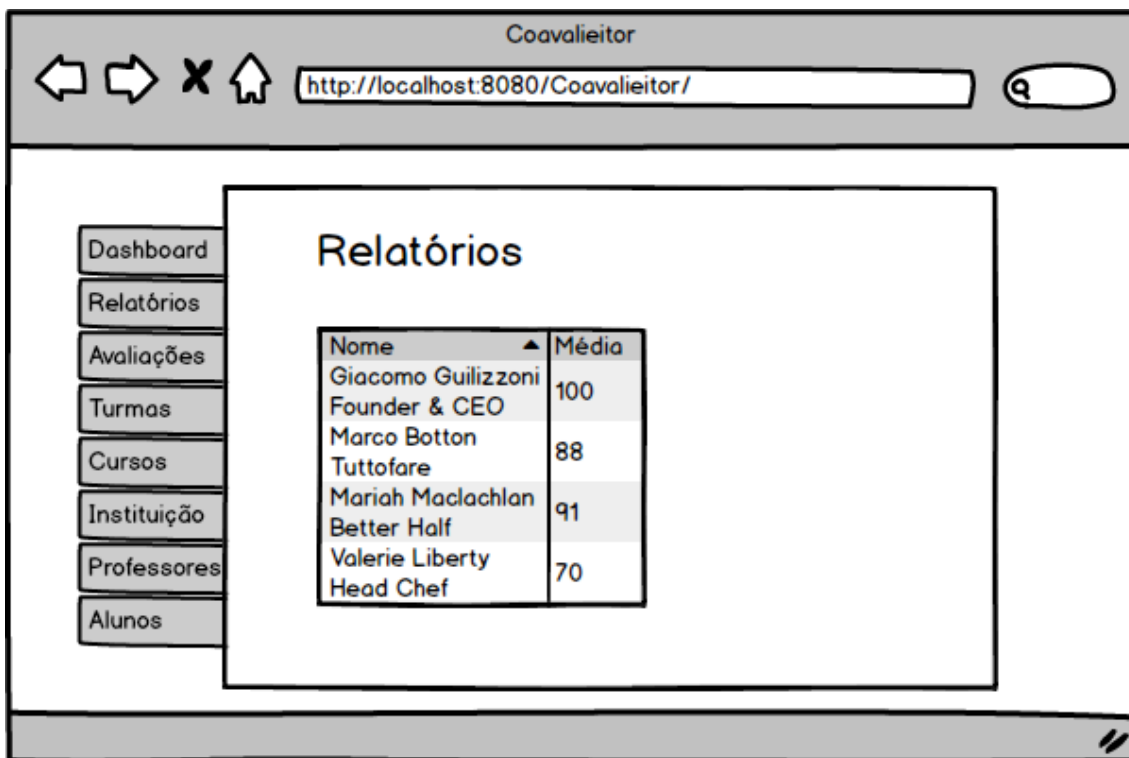


Figura 58 – Protótipo “Relatórios”.

FONTE: OS AUTORES (2016).

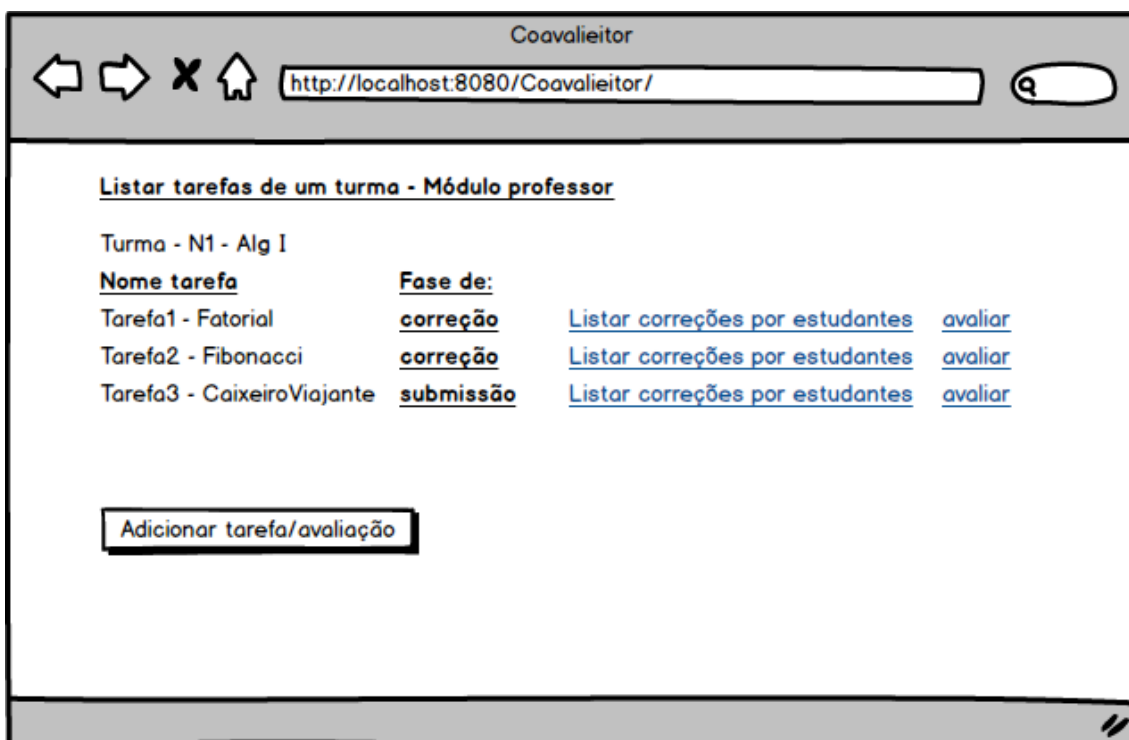


Figura 59 – Protótipo “Listar tarefas de uma turma”.

FONTE: OS AUTORES (2016).

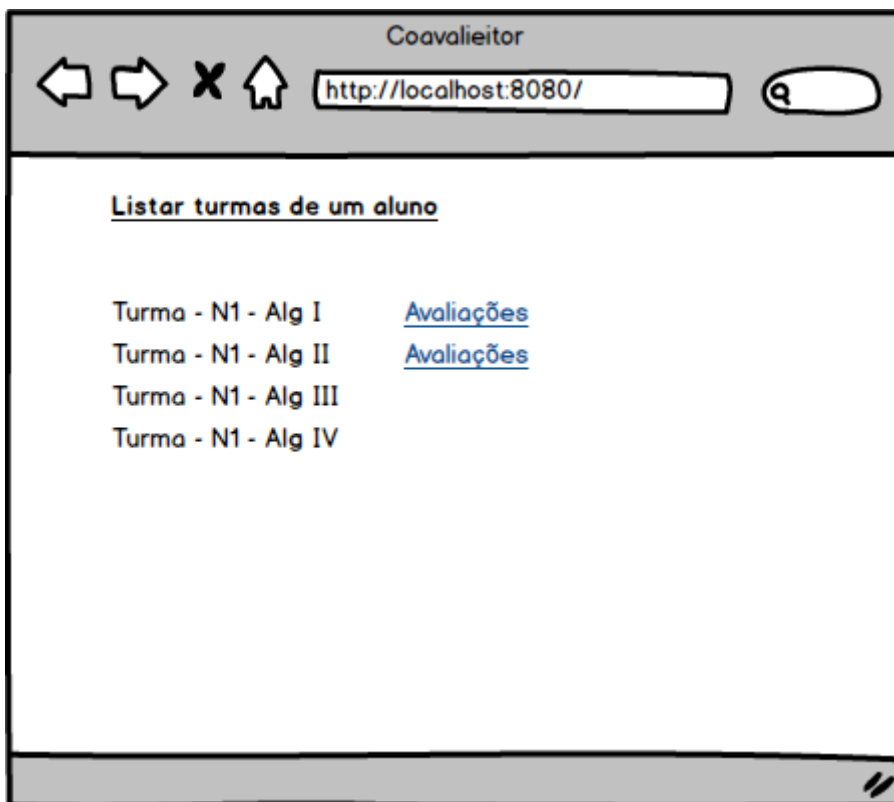


Figura 60 – Protótipo “Listar turmas aluno”.

FONTE: OS AUTORES (2016).

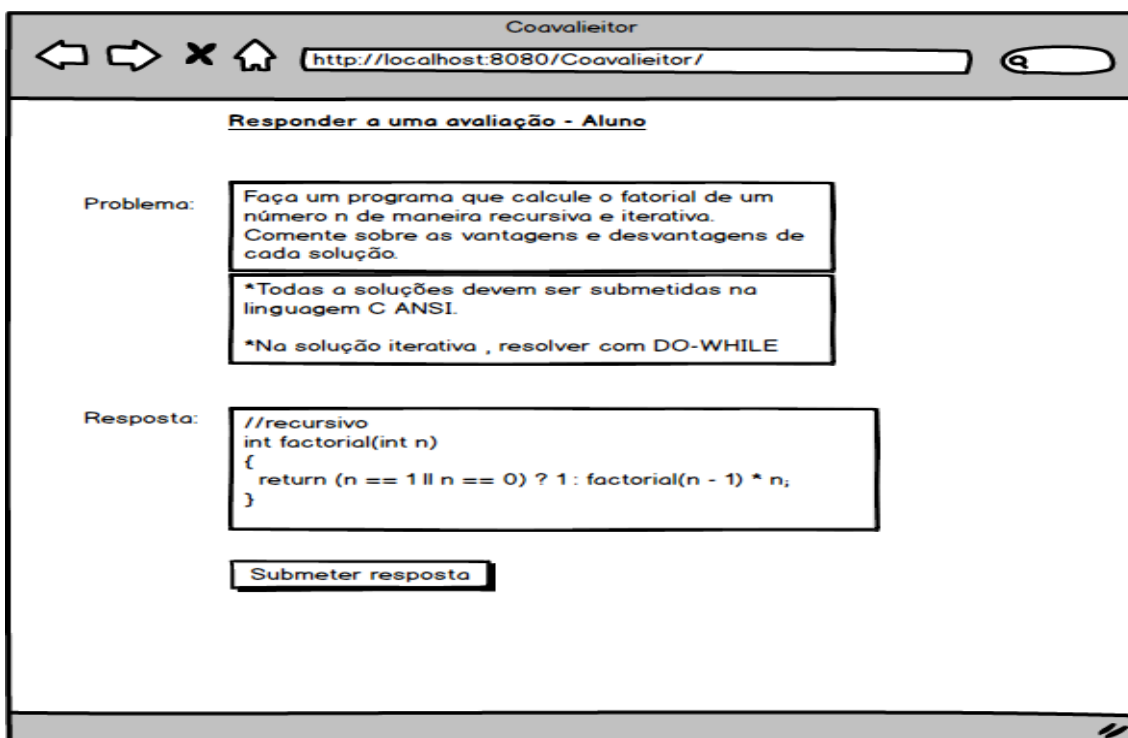


Figura 61 – Protótipo “Responder a uma avaliação”

FONTE: OS AUTORES (2016).

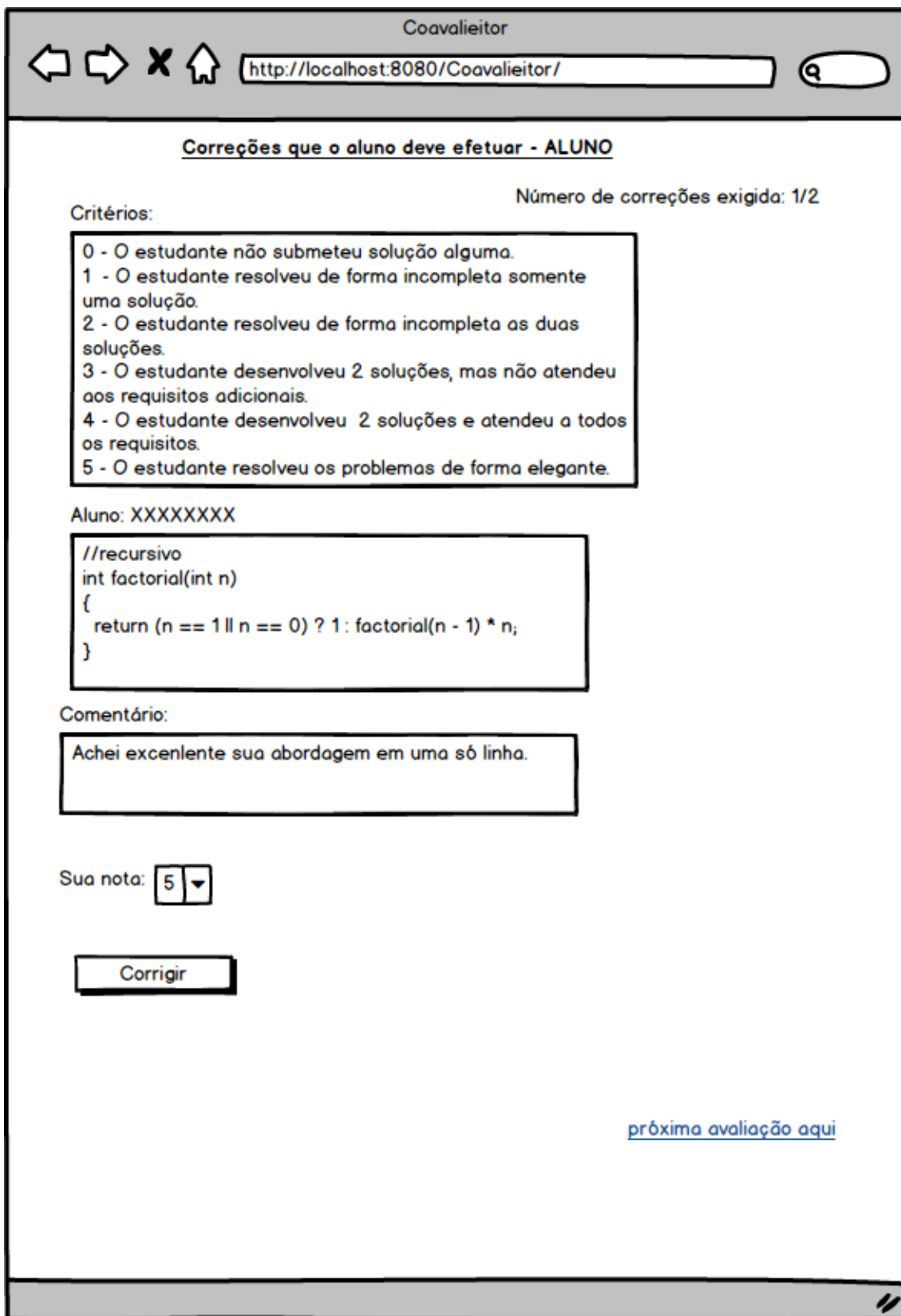
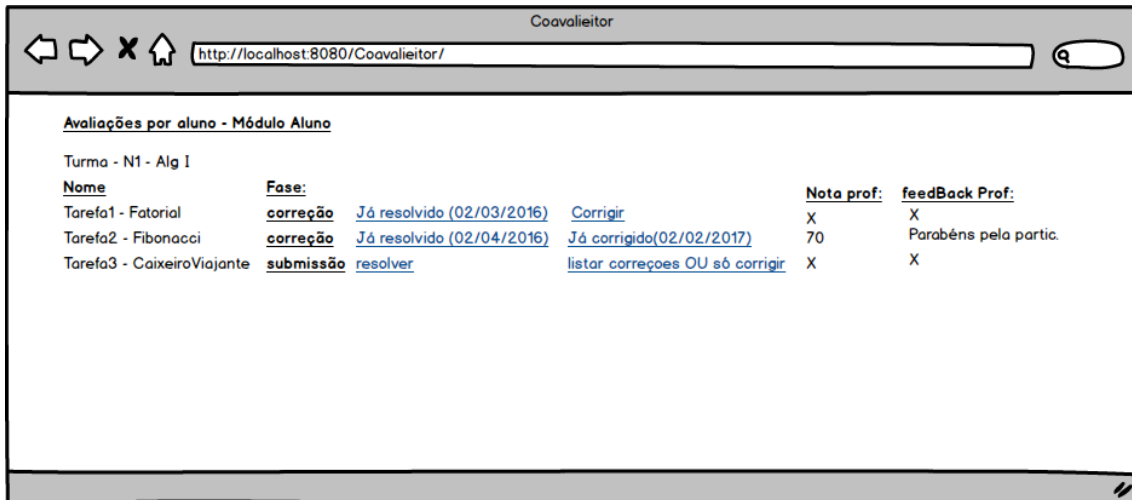


Figura 62 – Protótipo “Realizar correções”.

FONTE: OS AUTORES (2016).



Avaliações por aluno - Módulo Aluno

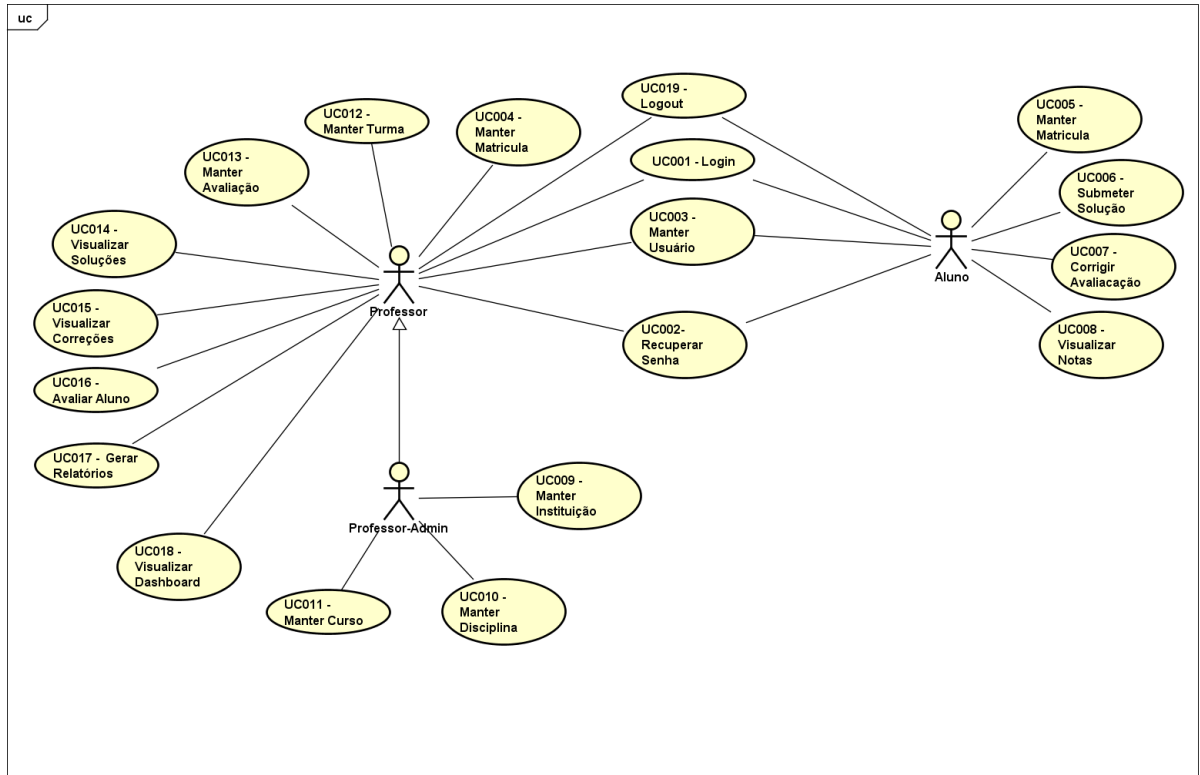
Turma - N1 - Alg I

<u>Nome</u>	<u>Fase:</u>		<u>Nota prof:</u>	<u>feedBack Prof:</u>
Tarefa1 - Fatorial	<u>correção</u>	Já resolvido (02/03/2016) Corrigir	X	X
Tarefa2 - Fibonacci	<u>correção</u>	Já resolvido (02/04/2016) Já corrigido(02/02/2017)	70	Parabéns pela partic.
Tarefa3 - CaixeiroViajante	<u>submissão</u>	resolver listar correções OU só corrigir	X	X

Figura 63 – Protótipo “Avaliações por aluno”.

FONTE: OS AUTORES (2016).

APÊNDICE 3 – DIAGRAMA DE CASO DE USO



APÊNDICE 4 – ESPECIFICAÇÃO DE CASO DE USO

UC001 – Login

Descrição:

Este Caso de Uso descreve as etapas percorridas para realizar o login no software Coavaleitor.

Pré-condições

1. Não se aplica.

Pós-condições

1. Após esse caso de uso o usuário deverá estar devidamente logado nos sistema.

Ator Primário

Professor/Estudante

Fluxo Principal

- 1- O sistema apresenta a tela de Login.
- 2- Aluno/Professor preencha os campos “Email” e “Senha”.
- 3- Aluno/Professor clica no botão “Login”. (A1) (A2)
- 4- O sistema valida os dados.(E1) (E2)
- 5- O sistema redireciona para tela inicial.
- 6- O caso de uso é finalizado.

Fluxos Alternativos

A1. Usuário clica no link “Esqueceu senha”

1. O sistema redireciona para tela de “Esqueceu senha”

A2. Usuário clica no botão “Ainda não possui conta? Cadastre – se já”

1. O sistema redireciona para tela de “Cadastre – se já”

Fluxos de Exceção

E1. Campos requeridos não preenchidos.

1. O sistema emite uma mensagem “Preencher campos requeridos”.

2. O caso de uso é reiniciado.

E2. Login inválido

1. O sistema emite a mensagem “Login inválido”.
2. O caso de uso é reiniciado.

Regras de Negócio: Não se aplica.

UC002 - Recuperar Senha

Descrição:

Este Caso de Uso descreve as etapas percorridas para realizar a recuperação de senha, a mesma será enviada para o e-mail previamente cadastrado no sistema.

Pré-condições

1. Não se aplica

Pós-condições

1. Um e-mail de recuperação de senha será enviado para o e-mail do Professor/Aluno.

Ator Primário

Professor/Estudante

Fluxo Principal

- 1- O sistema apresenta a tela de Recuperar Senha.
- 2- O usuário preenche o campo “E-mail”.
- 3- O usuário clica no botão “Mandar nova senha por e-mail”. (A1).
- 4- O sistema valida o e-mail (E1)(E2).
- 5- O sistema envia um e-mail de recuperação de senha.
- 6- O caso de uso é encerrado com sucesso.

Fluxos Alternativos

A1. Usuário clica no link “*Ainda não possui um conta? Cadastre-se já*”.

1. O sistema redireciona para tela de cadastro de usuário DV003.

Fluxos de Exceção

E1. Campo e-mail vazio.

1. O sistema emite uma mensagem “Esse campo é requerido”.
2. O Caso de uso é reiniciado.

E2. E-mail não cadastrado no sistema.

1. O sistema emite uma mensagem "E-mail inválido"
2. O Caso de Uso é reiniciado.

UC003 -Manter Usuário**Descrição:**

Este Caso de Uso descreve as etapas percorridas para realizar a ação de cadastro de usuário de acordo com o perfil selecionado no software Coavaleitor.

Pré-condições

1. Não se aplica.

Pós-condições

1. O usuário estará devidamente cadastrado no sistema de acordo com seu perfil. (Professor/Estudante).

Ator Primário

Professor/Estudante

Fluxo Principal

- 1- Aluno/Professor seleciona opção "Sou Professor" ou "Sou estudante".
- 2- Aluno/Professor digita seu nome.
- 3- Aluno/Professor digita seu e-mail para acesso.(R1)
- 4- Aluno/Professor digita uma senha para acesso.(R2)
- 5- Aluno/Professor clica no botão "Cadastrar". (A1)
- 6- O sistema valida as informações preenchidas pelo usuário. (E1)(E2)
- 7- O sistema redireciona para tela inicial do Professor, ou Aluno.
- 8- O caso de uso é finalizado.

Fluxos Alternativos

A1. Usuário clica no link "Já possui uma conta? Acesse aqui".

1. O sistema redireciona para tela de Login.

Fluxos de Exceção

E1. Campos requeridos não preenchidos.

1. O sistema emite uma mensagem "Preencher campos requeridos".
2. O caso de uso é reiniciado.

E2. O e-mail já foi previamente cadastrado no sistema.

1. O sistema emite uma mensagem "E-mail já cadastrado no sistema. "
2. O caso de uso é reiniciado

Regras de Negócio

R1. E-mail só poderá ser cadastrado uma única vez no sistema.

R2. A senha deverá ser criptografada na base de dados.

UC004 - Manter Matrícula Professor**Descrição:**

Este Caso de Uso descreve as etapas percorridas pelo professor para matricular um aluno em uma turma.

Pré-condições

O Professor deve estar logado no sistema.

Pós-condições

Um professor cadastrará com sucesso um aluno em uma turma.

Ator Primário

Professor

Fluxo Principal

1. O sistema apresenta a tela de matrículas.
2. O Professor digita o nome do aluno (E1)
3. O Professor clica em "Incluir Aluno" (A1) (A2)
4. O sistema matricula o aluno na turma.
5. O caso de uso é encerrado.

Fluxos Alternativos

- A1. O professor clica em "Remover"
1. O sistema remove o aluno da turma.
- A2. O professor clica em "Voltar"
1. O sistema redireciona o professor para tela de listar turmas.

Fluxo de Exceção

- E1. Aluno não registrado no sistema.
1. O sistema não retorna o nome do aluno procurado.

Regras de Negócios

Não se aplica.

UC005 - Manter Matrícula Aluno**Descrição:**

Este Caso de Uso descreve as etapas percorridas pelo aluno para se matricular em uma turma.

Pré-condições

O Aluno deve estar logado no sistema.

Pós-condições

O Aluno estará devidamente matriculado em uma turma.

Ator Primário

Aluno

Fluxo Principal

1. O sistema apresenta a tela de “Listar turmas”.
2. O aluno clica em “Inscreva-me”.
3. O sistema apresenta uma janela modal, chamada “Inscreva-me”
4. O aluno preenche o campo senha. (E1)
5. O aluno clica em “Entrar” (A1).
6. O sistema matricula o aluno na turma.
7. O Caso de uso é encerrado.

Fluxos Alternativos

A1. O aluno clica em “Fechar”.

1. O sistema redireciona para tela de “Listar turmas”.

Fluxo de Exceção

E1. Senha inválida.

1. O Sistema emite a mensagem “Senha inválida”.
2. O Sistema redireciona para tela de “Listar turmas”.

Regras de Negócio

Não se aplica.

UC006 - Submeter Solução**Descrição**

Este Caso de Uso descreve as etapas percorridas para realizar a ação de solução e submissão de respostas dos alunos ao software Coavaleitor.

Pré-condições

O Estudante deve estar logado no software.

Pós-condições

Correção Avaliação

Ator Primário

Estudante

Fluxo Principal

1. O estudante deve estar logado no sistema
2. O estudante deve ter se inscrito na instituição de ensino (A1)
3. O estudante deve ter se inscrito na disciplina
4. O estudante seleciona turma desejada (A2)
5. O estudante deve clicar no botão "Resolver" (A3)
6. O software abre a tela de Resolver Avaliações
7. O estudante submete sua resposta
- 9- O caso de uso é finalizado.

Fluxos Alternativos

- A1. O software solicita senha de acesso ao estudante
- A2. O software direciona para tela de avaliações.

UC007 Corrigir Avaliação**Descrição**

Este Caso de Uso descreve as etapas percorridas para realizar a ação de correção de respostas dos estudantes, baseado nos questionários passados pelos professores.

Pré-condições

O estudante deve estar na fase de correções

Pós-condições

Não se aplica

Ator Primário

Estudante

Fluxo Principal

1. O estudante deve estar na fase de correções
2. O estudante pode corrigir as questões de outros alunos(A1)
3. O sistema mostra quais soluções estão disponíveis(A2)(R1)(R2)
4. O estudante clica em checar (E1)
5. O sistema redireciona para tela Correção de Soluções
6. O estudante visualiza a questão
7. O estudante verifica os critérios de correção
8. O estudante escreve um comentário sobre solução de outro estudante
9. O estudante fornece sua nota baseado na solução do outro estudante
10. O caso de uso é finalizado.

Fluxos Alternativos

(A1) O estudante deve clicar no botão corrigir

(A2) O sistema irá abrir a tela lista de soluções

Fluxos de Exceção

(E1) O sistema mostra mensagem de quantas soluções podem ser feitas

Regras de Negócio

(R1)O sistema mostra quantas soluções pode ser feita por aluno

(R2) O sistema mostra em qual solução de 'n' o estudante está

UC008 -Visualizar Notas

Descrição

Esta Tela deve ser visualizada pelo Ator Estudante, a fim de verificar sua nota da disciplina à qual está matriculado.

Pré-condições

O ator deve estar logado no sistema

Pós-condições

O Sistema exibe as notas.

Ator Primário

Estudante

Fluxo Principal

1. O estudante acessa a tela de visualização de Notas.
2. O estudante seleciona a matérias desejada e clica em Checar Notas
3. O sistema gera um PDF com as notas do aluno.
4. O caso de uso é finalizado.

Fluxos Alternativos

Não se aplica.

Fluxos de Exceção

Não se aplica.

Regras de Negócio

Não se aplica.

UC009 - Manter Instituição**Descrição**

Este Caso de Uso descreve as etapas percorridas para realizar a ação de cadastro de instituição.

Pré-condições

Estar logado com perfil Professor Administrador

Pós-condições

Não se aplica

Ator Primário

Professor Administrador

Fluxo Principal

1. O sistema mostra a tela cadastro de instituição

2. O professor Administrador clica no botão Adicionar
3. O sistema mostra a tela Nova instituição
4. O professor administrador realiza o cadastro da Instituição
5. O sistema volta para tela de visualização de instituição
6. O professor Administrador clica no botão Editar (A1)(A2)(A3)
7. O professor clica em no botão Deletar (E1)
8. O caso de uso é finalizado.

Fluxos Alternativos

- A1. O Sistema mostra tela para editar nome de instituição
- A2. O professor administrador clica em salvar
- A3. O sistema volta para tópico 5.

Fluxos de Exceção

- E1. O sistema mostra mensagem se realmente deseja excluir instituição.

Regras de Negócio

Não se Aplica

UC010 - Manter Disciplina**Descrição**

Este Caso de Uso especifica a ação do Ator Professor Administrador, poderá adicionar, alterar ou remover disciplina.

Pré-condições

O ator deve estar logado no sistema e possuir permissão de administrador.

Pós-condições

Não se aplica

Ator Primário

Professor-Administrador

Fluxo Principal

1. O Ator administrador acessa tela de disciplinas.
2. O sistema apresenta formulário

3. O Ator deve preencher os campos obrigatórios
4. O ator clica em botão adicionar (A1)
5. O Ator Adiciona nova disciplina
6. O sistema apresenta tela como nova disciplina
7. O sistema salva a disciplina no banco de Dados (E01)
8. O sistema mostra na tela disciplina cadastrada
9. O Caso de Uso é finalizado.

Fluxos Alternativos

A01.

- a. O Ator clica em salvar
- b. O ator clica em editar
- c. O ator altera as informações desejadas e clicar em salvar
- d. O sistema salva as informações no banco de dados
- e. O caso de uso é reiniciado

Fluxos de Exceção

E1.

- a. O ator não preencheu todos os campos necessários
- b. O sistema não adiciona a matéria ao banco de dados
- c. O caso de uso reinicia

Regras de Negócio

Não se aplica

UC011 - Manter Curso

Descrição

Este Caso de Uso especifica a ação do Ator Professor Administrador, poderá adicionar, alterar ou remover disciplina.

Pré-condições

Estar logado com perfil Professor Administrador

Pós-condições

Não se Aplica

Ator Primário

Professor Administrador

Fluxo Principal

1. O Sistema mostra a tela Cursos
2. O professor administrador Clica em adicionar
3. O sistema mostra tela Novo Curso
4. O professor administrador seleciona o nome da instituição
5. O professor administrador Digita o nome do Curso
6. O professor administrador clica no botão salvar (A1)
7. O professor administrador clica no botão listar(A2)
8. O Sistema mostra a lista de cursos baseado na instituição selecionada
9. O professor administrador clica no botão editar
10. O sistema direciona para tela editar curso
11. O professor administrador realiza edição e clica em salvar.
12. O professor administrador clica no botão deletar(E1)
13. O caso de uso é finalizado.

Fluxos Alternativos

- A1. O sistema volta para tela inicial
- A2. O sistema mostra mensagem que não existe registro.

Fluxos de Exceção

- E1. O sistema mostra mensagem se realmente deseja excluir curso.

Regras de Negócio

Não se aplica

UC012 - Manter Turma

Descrição

Este Caso de Uso descreve as etapas percorridas para realizar a ação de cadastro de Turma.

Pré-condições

Estar Logado com permissões Professor Administrador

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

1. O sistema mostra a tela Turmas
2. O professor clica no botão adicionar
3. O sistema redireciona para cadastro de nova turma (A1)
4. O professor volta para tela turmas
5. O professor seleciona filtra instituição
6. O professor seleciona filtra nome curso
7. O professor seleciona filtra disciplina
8. O professor clica em listar turmas(A2)
9. O professor clica em Editar (A3)
10. O professor clica em Deletar (A4)
11. O professor clica no botão Aluno(A5)
12. O professor clica no botão Avaliação (A6)
13. O caso de uso é finalizado.

Fluxos Alternativos

- A1.
- a. O professor seleciona instituição
 - b. O professor seleciona curso
 - c. O professor seleciona disciplina
 - d. O professor digita turma
 - e. O professor digita chave de acesso
 - f. O professor clica em botão salvar (E1)
- A2. Lista de turmas é listada em grid
- A3.
- a. O sistema mostra tela Editar Turma
 - b. O professor clica em salvar
 - c. O professor clica em voltar

d. O professor volta para tela Turmas

A4. O sistema mostra mensagem se ele realmente deseja realizar ação de exclusão.

A5.

a. O professor é direcionado para tela alunos turma

b. O professor inclui aluno em turma

c. O professor clica no botão incluir

d. O sistema mostra na tela a lista de alunos desta turma

e. O professor clica no botão voltar

f. O professor volta para tela Turmas

A6. O professor adiciona novas avaliações para alunos.

Fluxos de Exceção

Não se aplica

Regras de Negócio

Não se aplica

UC013 - Manter Avaliação

Descrição

Este Caso de Uso descreve as etapas percorridas para realizar a ação de cadastro de Avaliação.

Pré-condições

Estar Logado com permissões Professor Administrador

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

1. O sistema exibe tela Turma Matutina
2. O Professor clica no botão adicionar
3. O Professor Insere o nome da Avaliação
4. O Professor descreve atividade

5. O Professor descreve os requisitos adicionais
6. O Professor período de submissão (A1)
7. O Professor período de correção(A2)
8. O Professor seleciona quantidade de correções permitida por aluno (A3)
9. O Professor seleciona o valor máximo da nota atividade
10. O professor descreve os critérios de correção da atividade
11. O Professor clica no botão criar atividade
12. O sistema é redirecionado para tela de avaliações(A3)
13. O Professor clica no botão Editar.(A4)
14. O Professor clica no botão Deletar.(E1)
15. O Professor clica no botão Listar submissão (A5)
16. O Professor clica no botão Listar Correções(A6)
17. O professor clica no botão Avaliar(A7)
18. O professor clica no checkbox Liberar notas. (E2)
19. O caso de uso é finalizado.

Fluxos Alternativos

A1.

- a. O Professor seleciona no calendário o período inicial e final de submissão da atividade.
- b. O Professor seleciona hora inicial e final de submissão da atividade.

A2.

- a. O Professor seleciona no calendário o período inicial e final de correção da atividade.
- b. O Professor seleciona hora inicial e final de correção da atividade.

A3.

- a. O sistema mostra lista onde o professor pode editar
- b. O sistema mostra lista onde o professor pode deletar
- c. O sistema mostra lista onde o professor pode listar soluções
- d. O sistema mostra lista onde o professor pode listar correções
- e. O sistema mostra lista onde o professor pode Avaliar

A4.

- a. O Professor é direcionado para tela de Editar avaliação
 - b. O Professor clica em salvar Alterações
 - c. O sistema é direcionado para tela de avaliação
- A5. O Professor visualiza todas as submissões de seus alunos
- A6. O Professor visualiza todas as listas de correções de seus alunos
- A7.
- a. O professor é redirecionado para tela Avaliar Estudante.
 - b. O professor visualiza notas e médias finais de seus alunos.

Fluxos de Exceção

E1. O sistema mostra mensagem para usuário confirmar se realmente deseja excluir avaliação.

E2. Aparece mensagem informando que media só deve ser liberado após fechamento de todas as notas.

Regras de Negócio

Não se aplica

UC014 - Visualizar Solução

Descrição

Este Caso de Uso descreve as etapas percorridas para realizar a ação de Visualizar Solução.

Pré-condições

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

1. O professor deve clicar no botão Listar soluções
2. O sistema mostra tela de Lista de Soluções (A1)
3. O professor clica no botão voltar

4. O caso de uso é finalizado.

Fluxos Alternativos

A1.

- a. O professor deve clicar na imagem do olho para visualizar solução por aluno.
- b. A tela de detalhe de solução é aberta

Fluxos de Exceção

Não se aplica

Regras de Negócio

Não se aplica

UC015 - Visualizar Correção

Descrição

Este Caso de Uso descreve as etapas percorridas para realizar a ação de Visualizar Correção.

Pré-condições

Estar logado com perfil Professor

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

- 1. O professor deve clicar no botão Listar correções
- 2. O sistema mostra tela de Lista de Correções(A1)
- 3. O professor clica no botão voltar
- 4. O caso de uso é finalizado.

Fluxos Alternativos

A1.

- c. O professor deve clicar na imagem do olho na coluna correção para visualizar correção por aluno.
- d. A tela de detalhe de correção é aberta

Fluxos de Exceção

Não se aplica

Regras de Negócio

Não se aplica

UC016 -Avaliar Aluno**Descrição**

Este Caso de Uso descreve as etapas percorridas para realizar a ação da Tela de avaliar alunos.

Pré-condições

Estar logado perfil professor

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

1. O sistema mostra tela Avaliar Estudante
2. O professor clica na coluna checked (avaliar)
3. O sistema mostra a tela especificar nota
4. O sistema mostra a média das correções(A1)
5. O sistema mostra peso das médias das correções(A2) (R1)
6. O sistema mostra nota final do aluno
7. O professor digita feedback baseado na solução
8. O professor clica no botão salvar alterações
9. O professor clica no botão fechar
10. O caso de uso é finalizado.

Fluxos Alternativos

A1.O sistema mostra média baseada nas notas de correções de outros alunos.

A2. O sistema calcula média de aluno baseada em seu peso

Fluxos de Exceção

Não se aplica

Regras de Negócio

R1. O sistema mostra a média do aluno baseada no cálculo da média ponderada do aluno em suas atividades.

UC017 - Gerar Relatório**Descrição**

Este Caso de Uso descreve as etapas percorridas para realizar a ação da Tela de Gerar Relatório.

Pré-condições

Estar logado com perfil Professor

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

1. O sistema mostra tela de relatórios
2. O professor escolhe o tipo de relatório (A1)
3. O professor escolhe Período (A2)
4. O professor clica no botão Gerar Relatório
5. O sistema mostra em PDF Relatório gerado.
6. O caso de uso é finalizado.

Fluxos Alternativos

A1.

- a. O professor escolhe relatório por média
- b. O professor escolhe relatório quem efetuou correção
- c. O professor escolhe relatório agrupados por turma

A2.

- a. O professor escolhe período para gerar relatório

- b. O professor clica em aplicar

Fluxos de Exceção

Não se aplica

Regras de Negócio

Não se aplica

UC018 -Visualizar Dashboard

Descrição

Este Caso de Uso descreve as etapas percorridas para realizar a ação da Tela de Visualizar Dashboard.

Pré-condições

Estar logado com perfil Professor

Pós-condições

Não se aplica

Ator Primário

Professor

Fluxo Principal

1. O sistema mostra a tela de Dashboard.
2. O sistema mostra um gráfico com estatísticas sobre avaliações(A1)
3. O caso de uso é finalizado.

Fluxos Alternativos

- A1.
 - a. O sistema mostra gráfico com estatísticas
 - b. O sistema mostra gráfico com correções

Fluxos de Exceção

Não se aplica

Regras de Negócio

Não se aplica

UC019- Fazer Logout**Descrição:**

Este caso de uso descreve as etapas percorridas por um usuário para sair do sistema Coavaliator.

Pré-condições

O ator deve estar logado

Pós-condições

Encerrar acesso do Usuário

Ator Primário

Professor/Aluno

Fluxo Principal

1. O ator deve acionar o menu LOGOUT
2. O sistema encerra a sessão do usuário
3. O sistema redireciona o usuário para página inicial
4. O caso de uso é finalizado.

Fluxos Alternativos

Não se aplica.

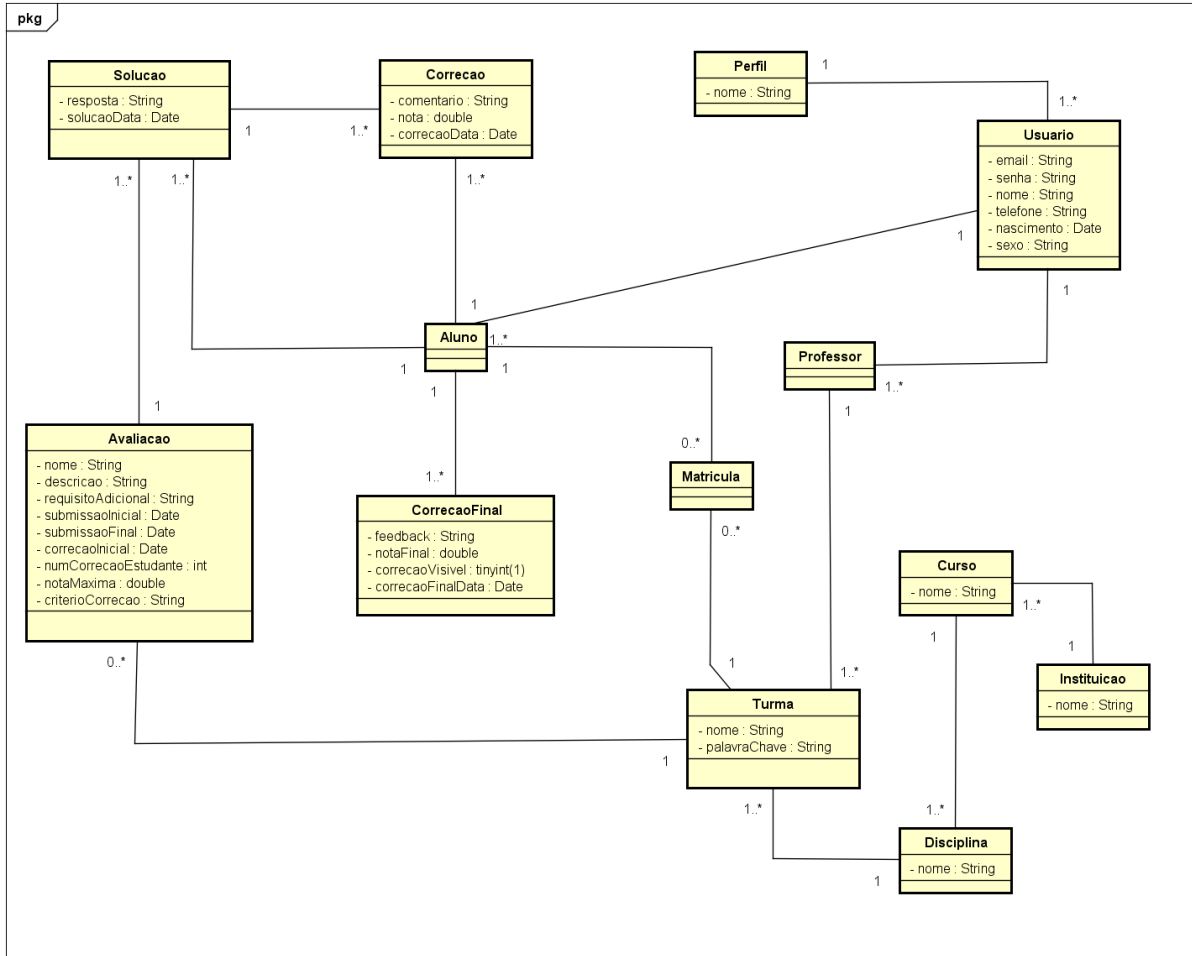
Fluxos de Exceção

Não se aplica.

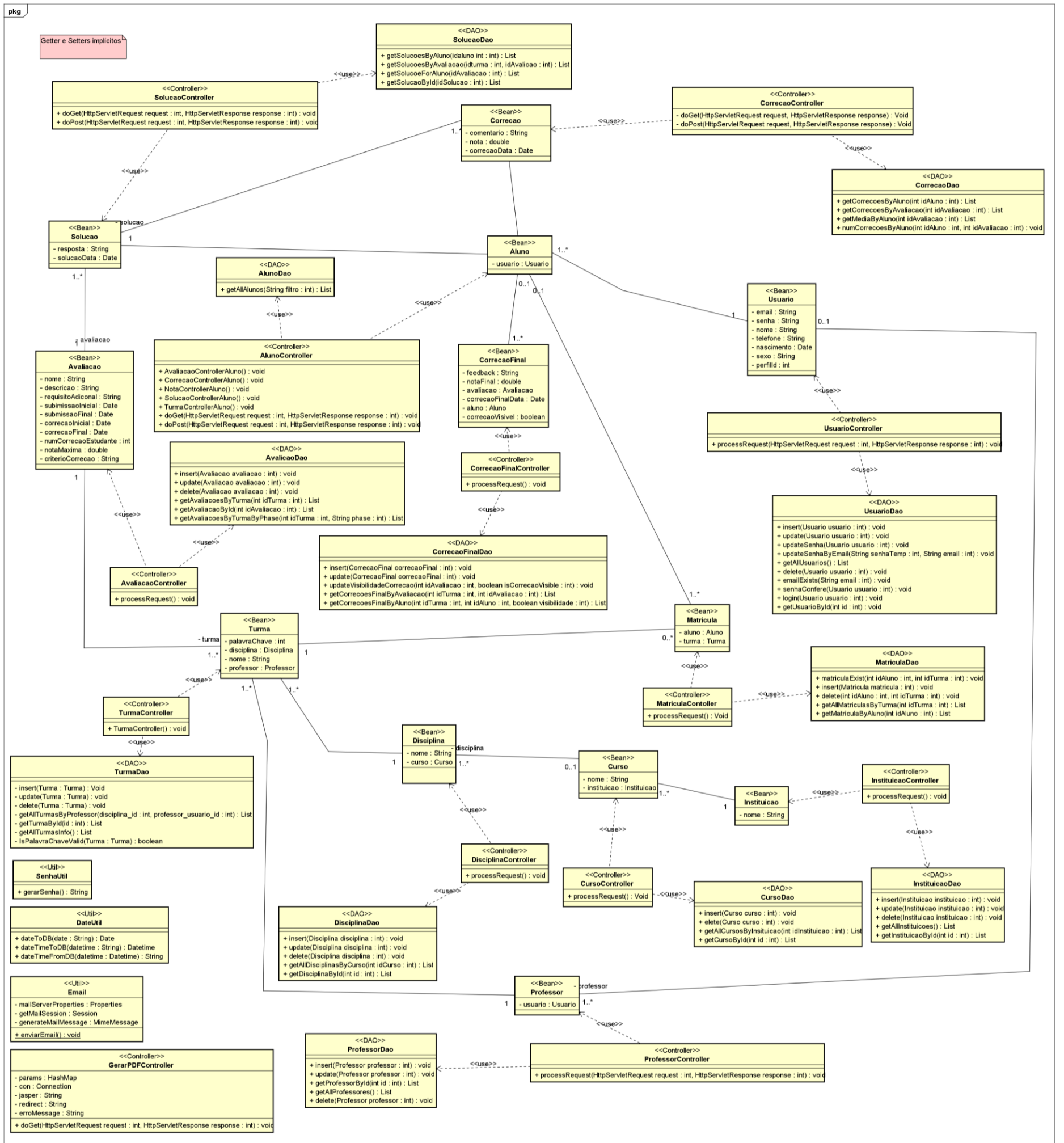
Regras de Negócio

Não se aplica.

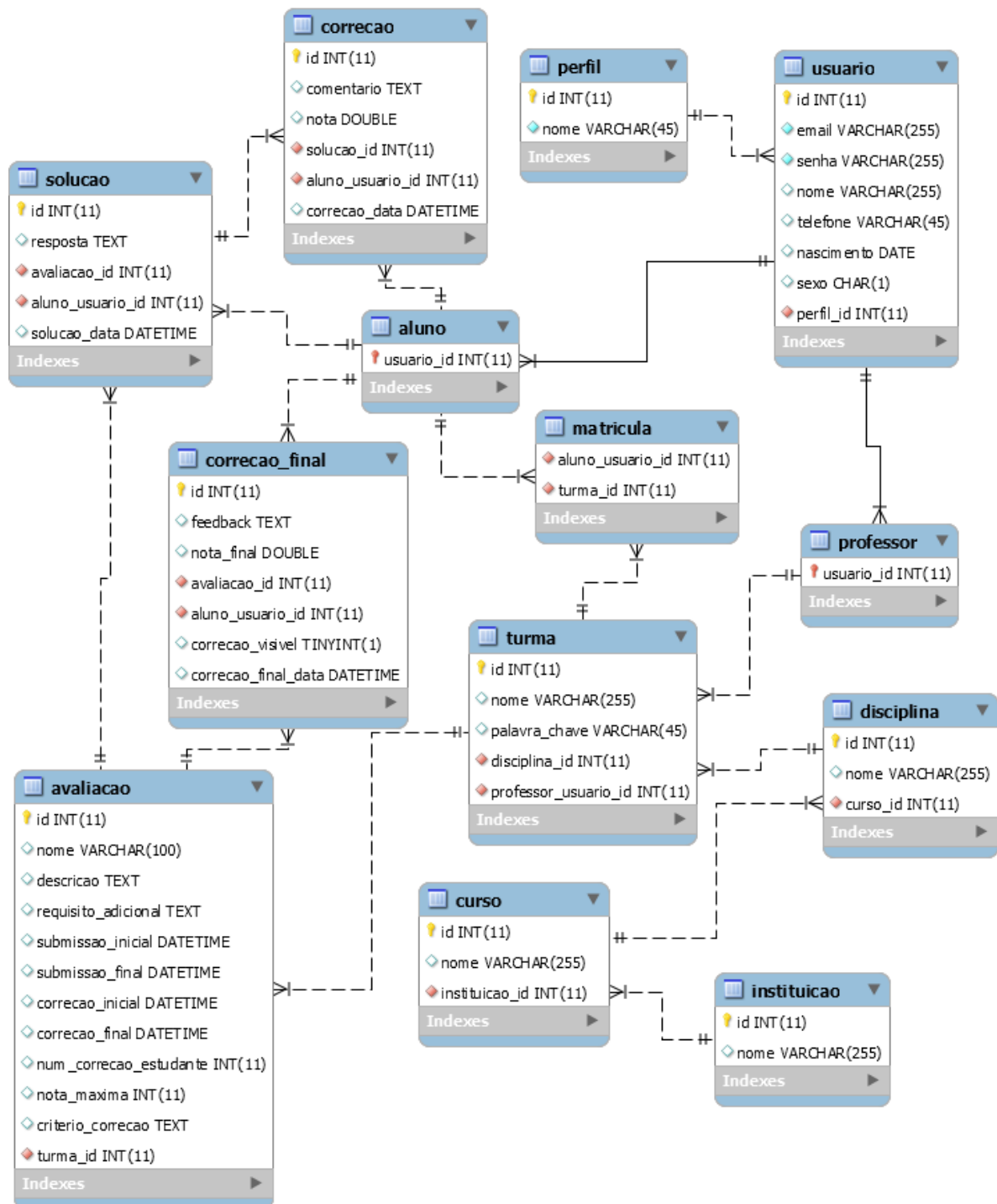
APÊNDICE 5 – DIAGRAMA DE CLASSE DE NEGÓCIOS



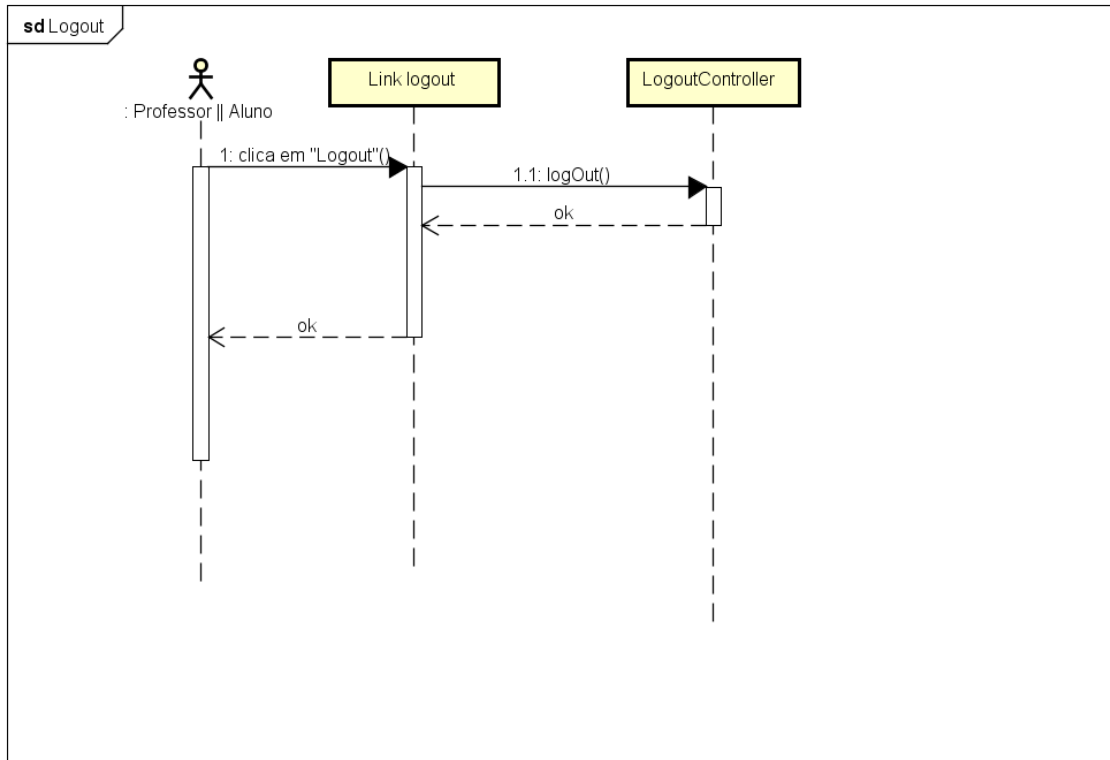
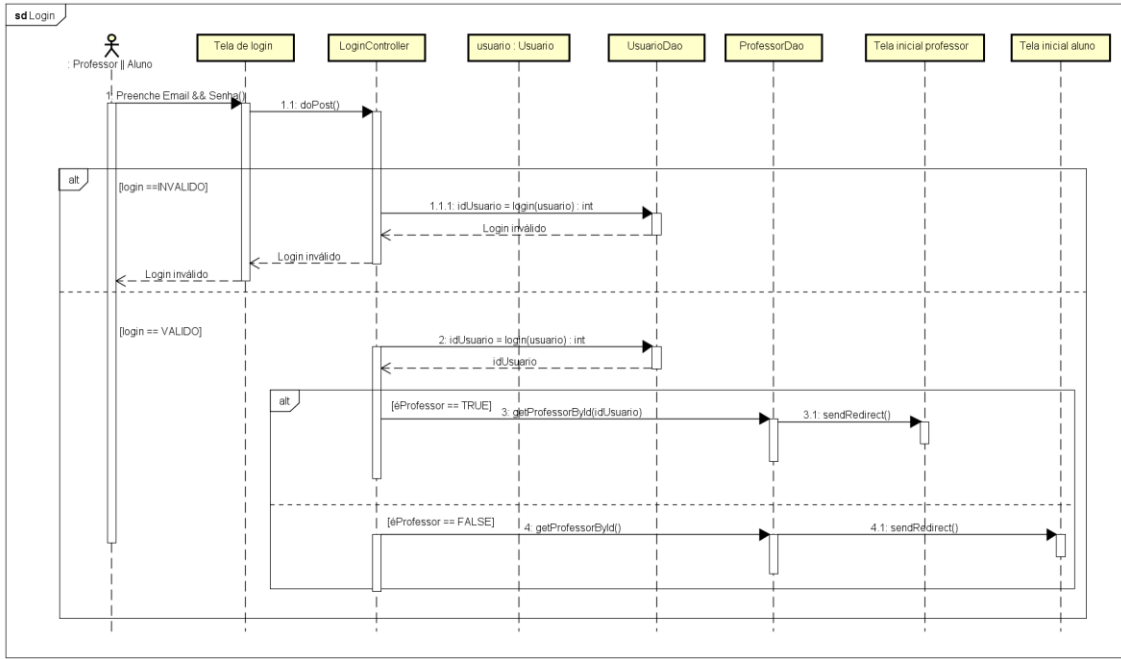
APÊNDICE 6 – DIAGRAMA DE CLASSE DE IMPLEMENTAÇÃO

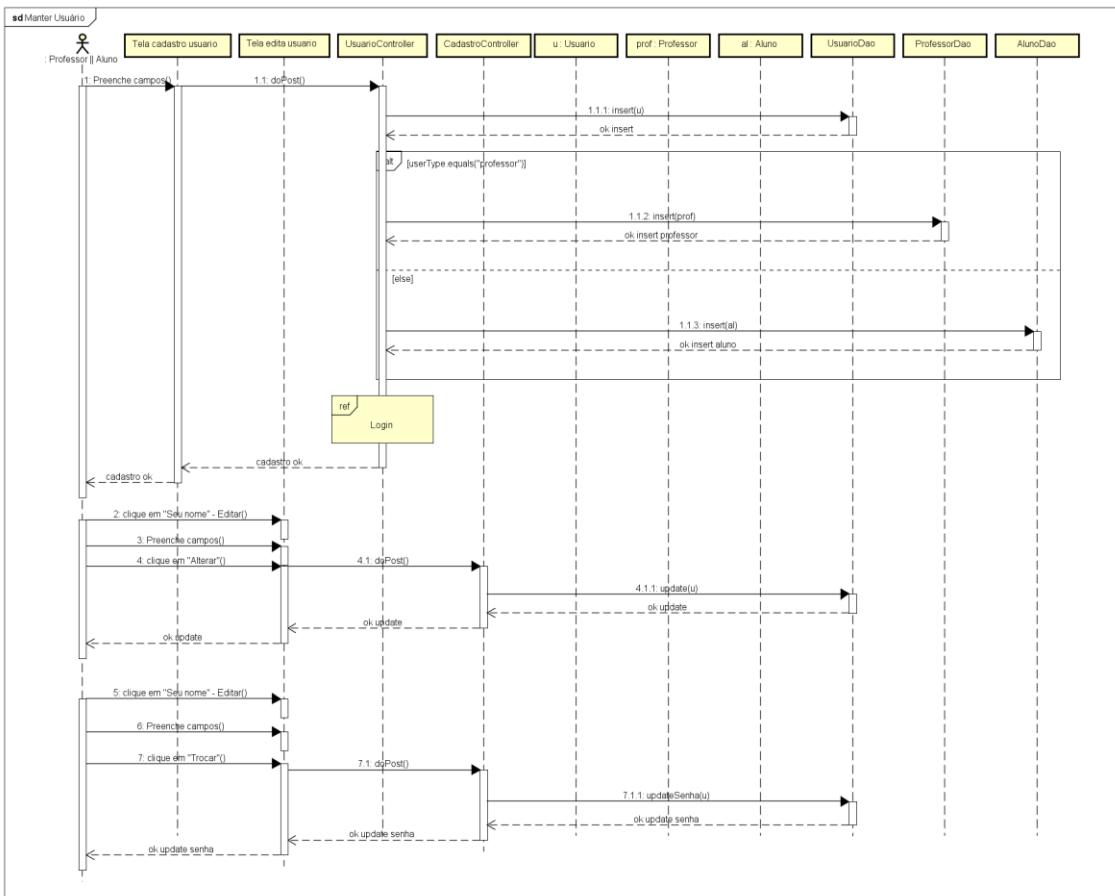
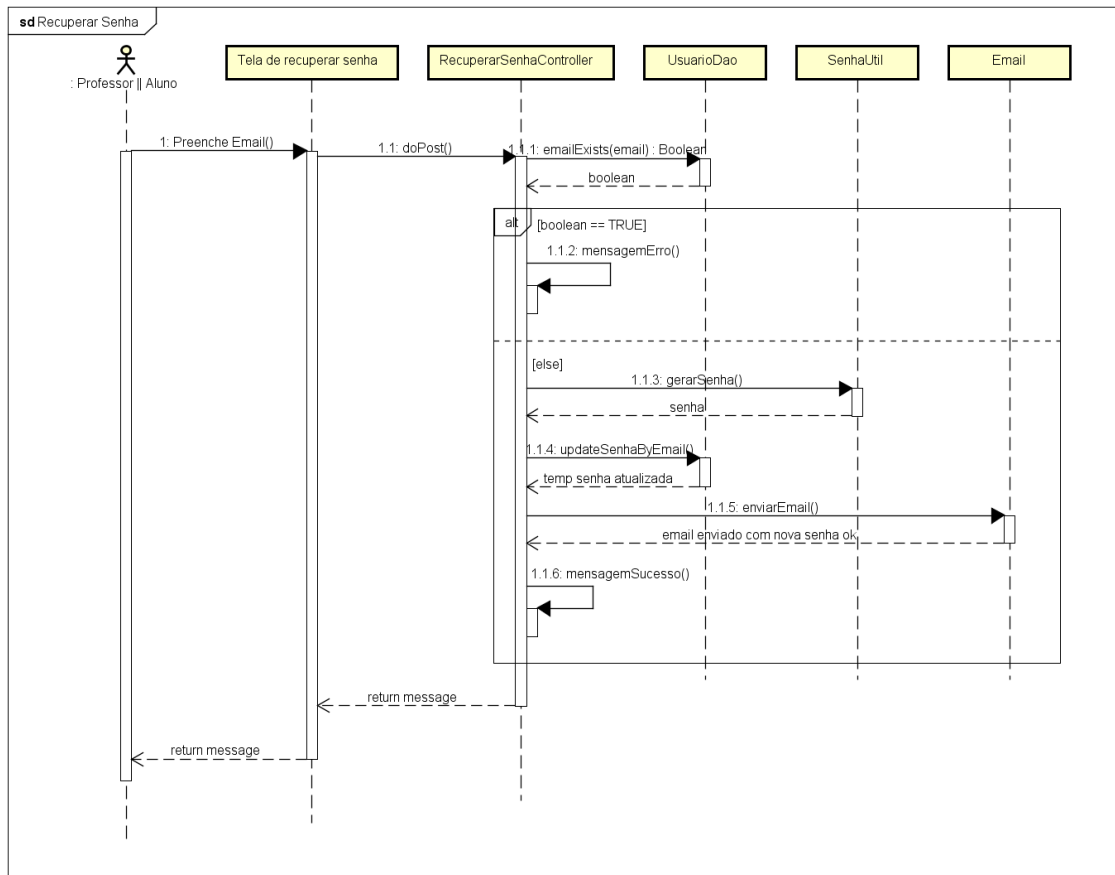


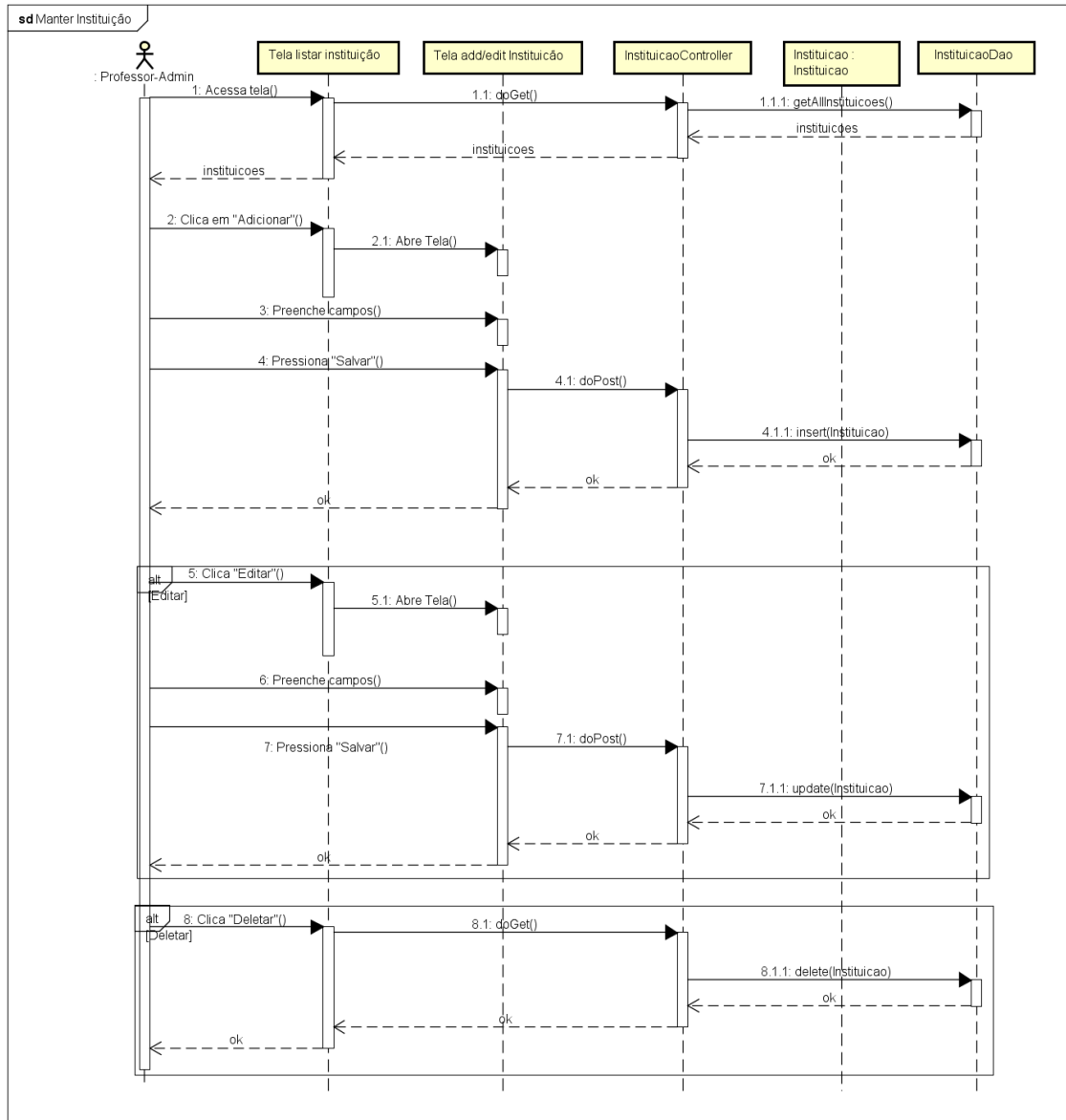
APÊNDICE 7 – DIAGRAMA DE ENTIDADE RELACIONAMENTO

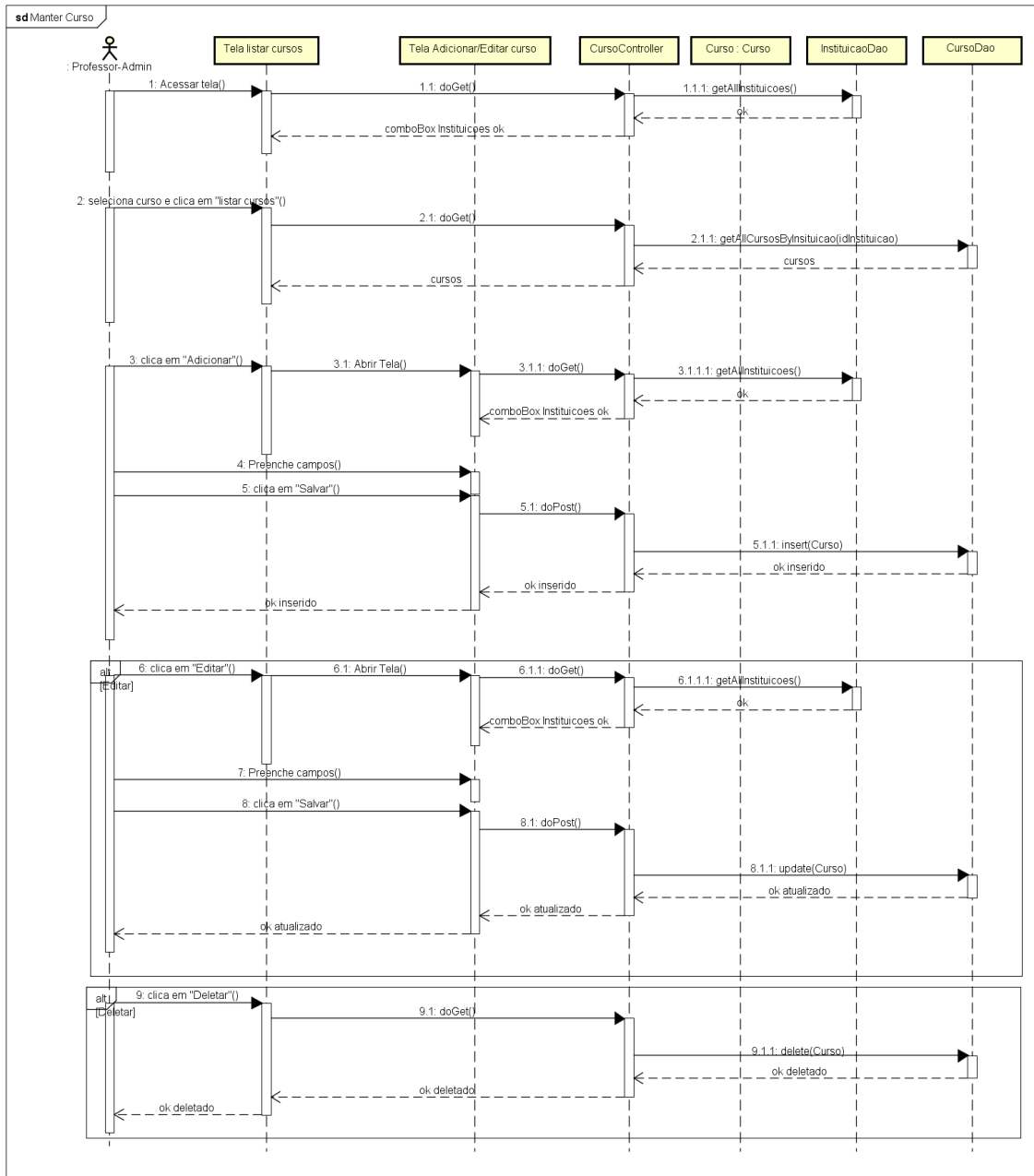


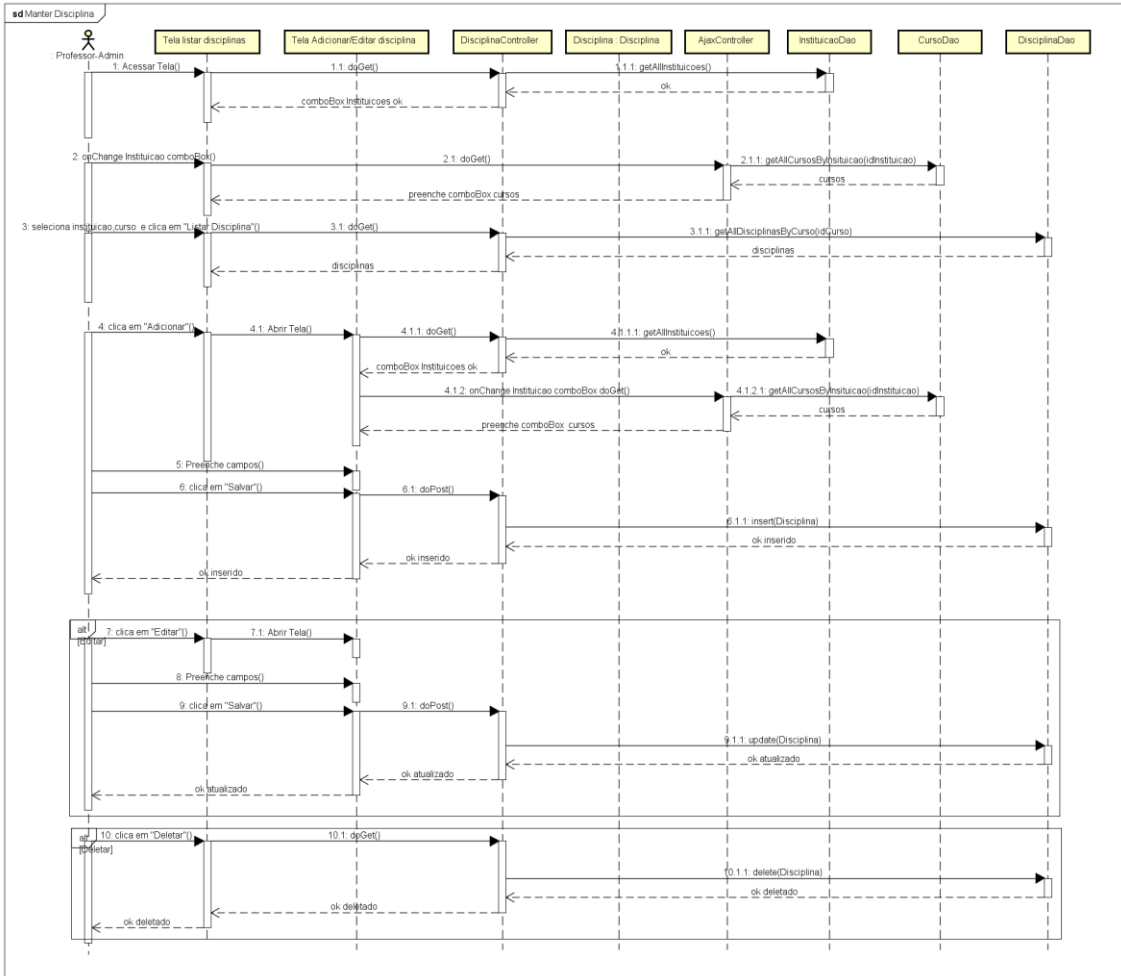
APÊNDICE 8 – DIAGRAMAS DE SEQUÊNCIA

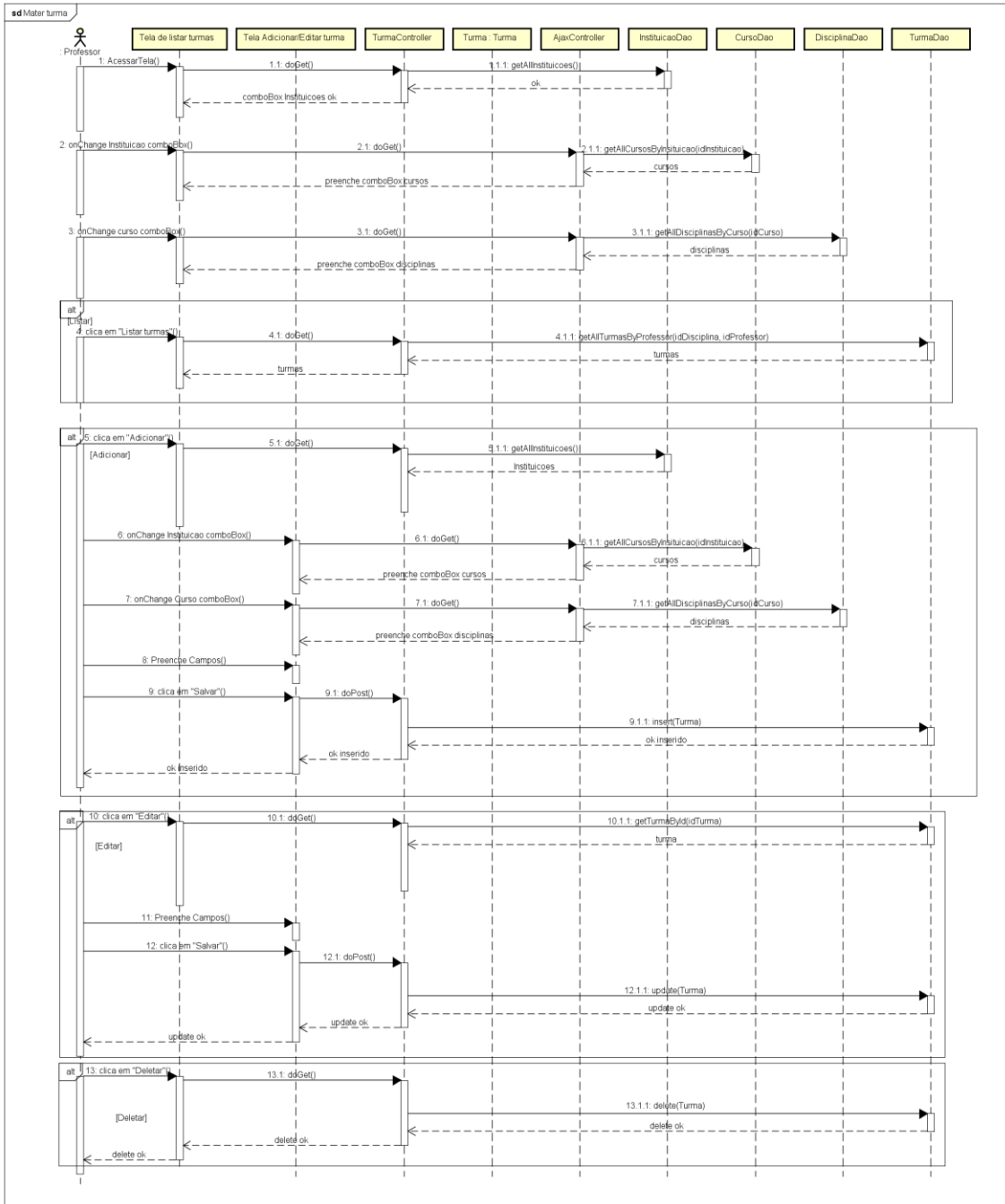


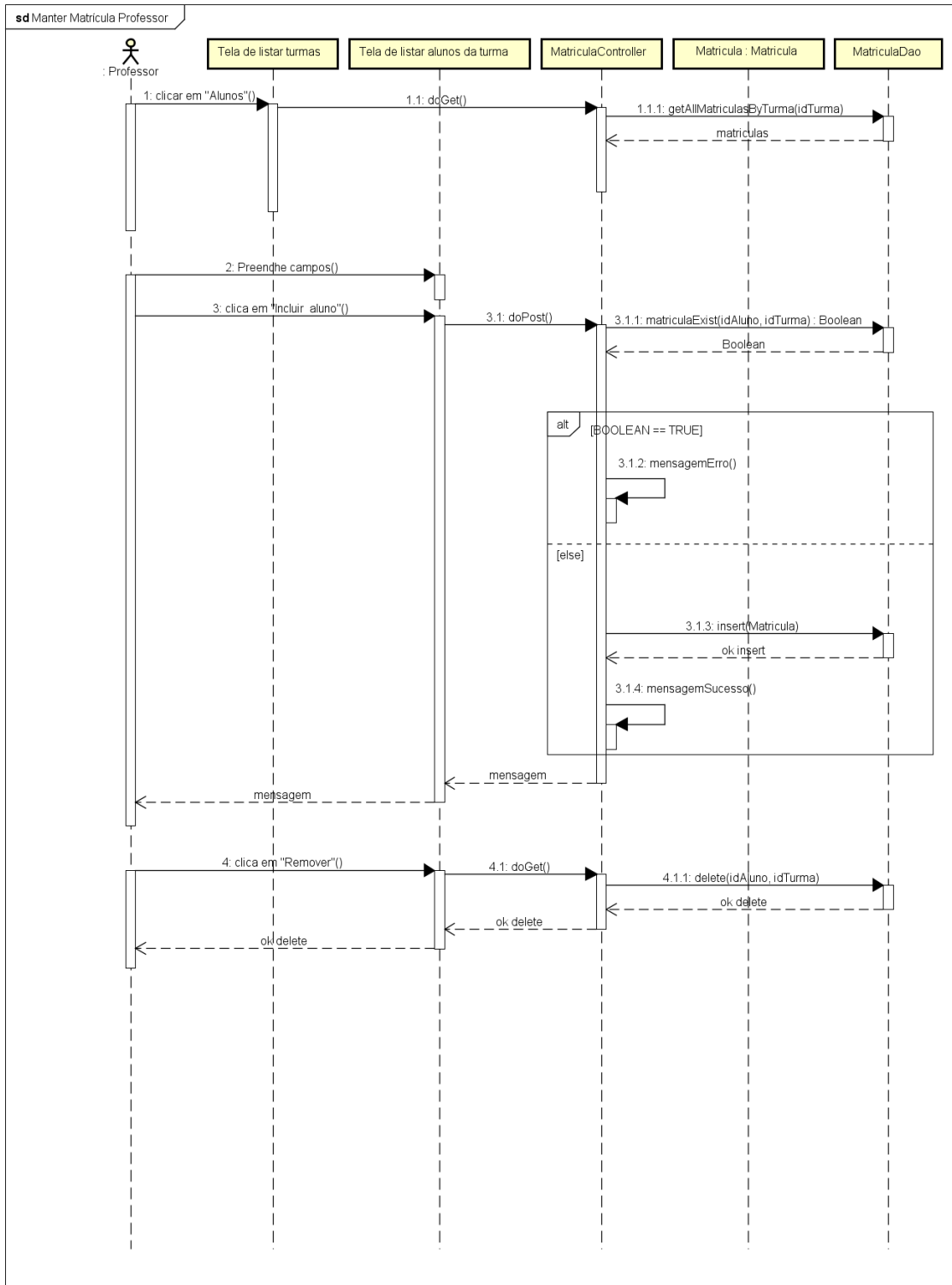


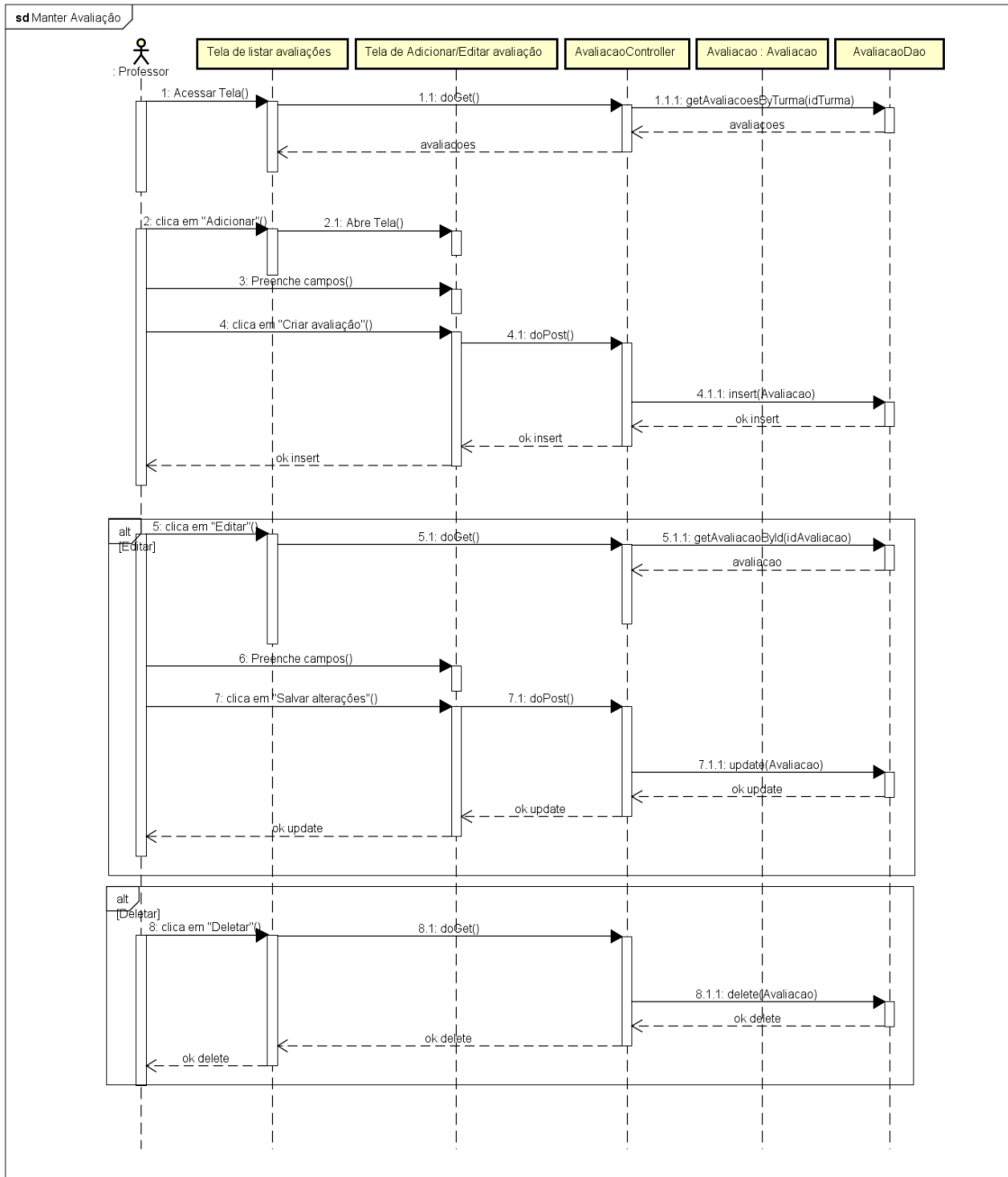


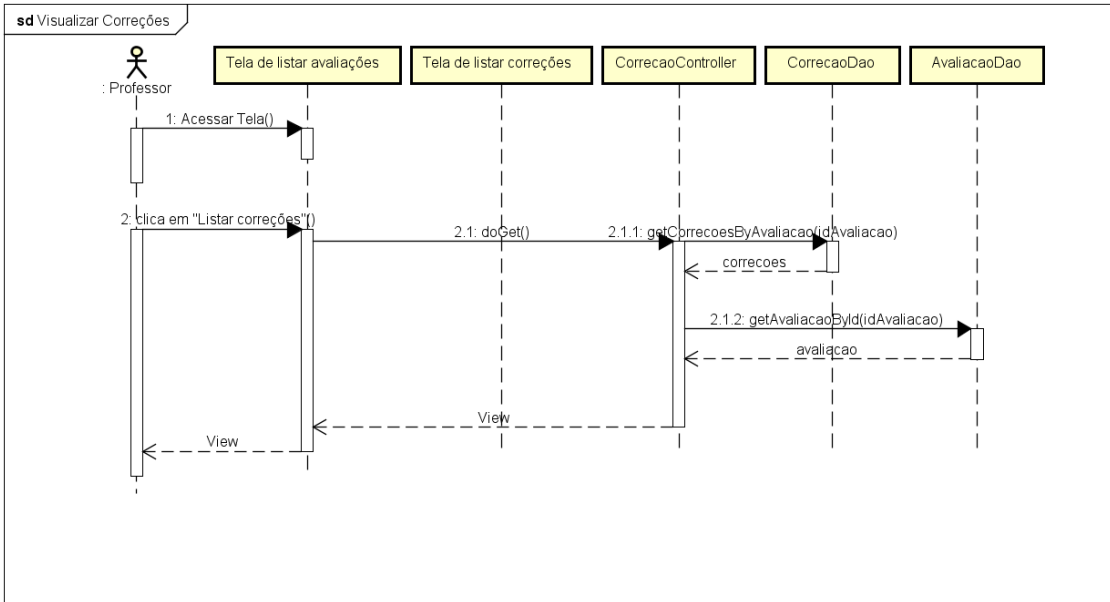
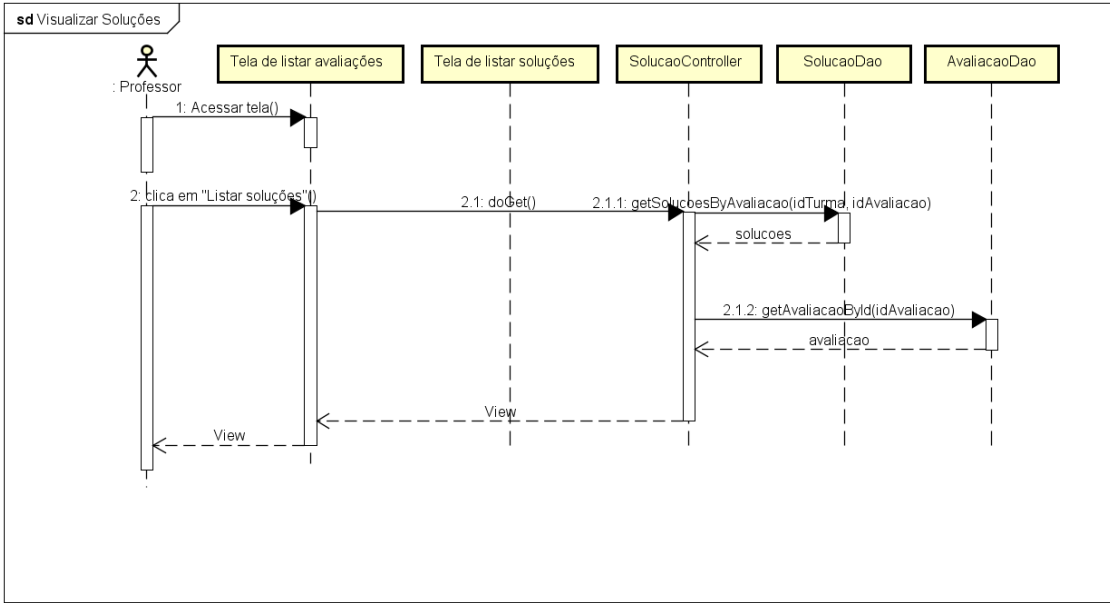


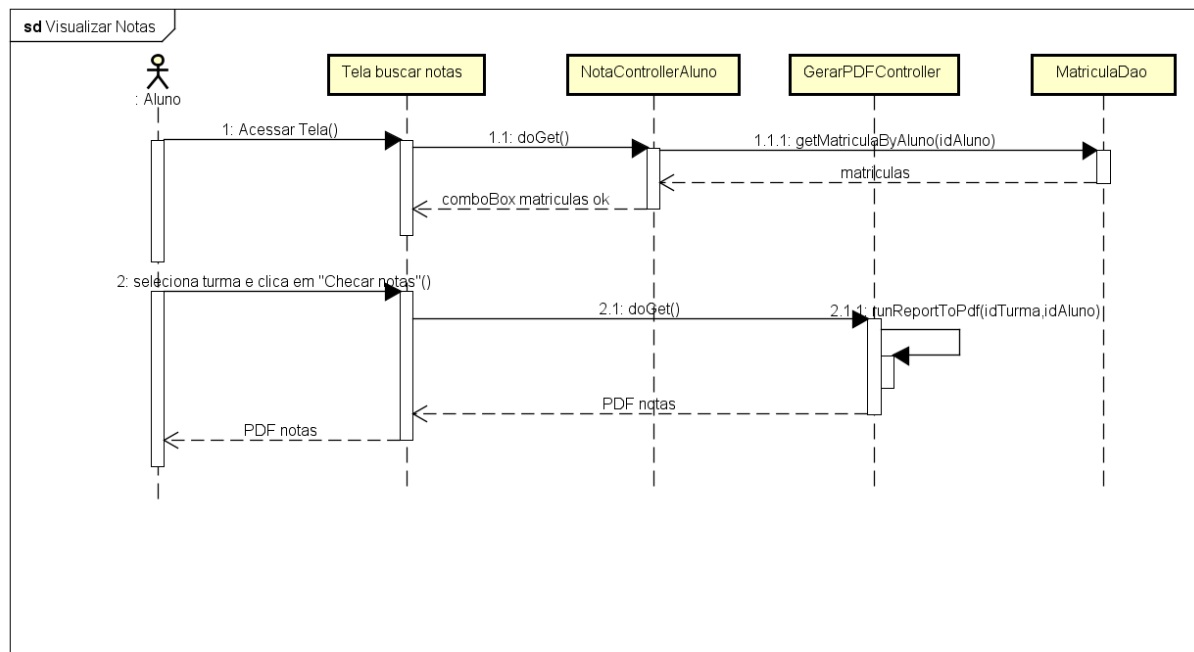
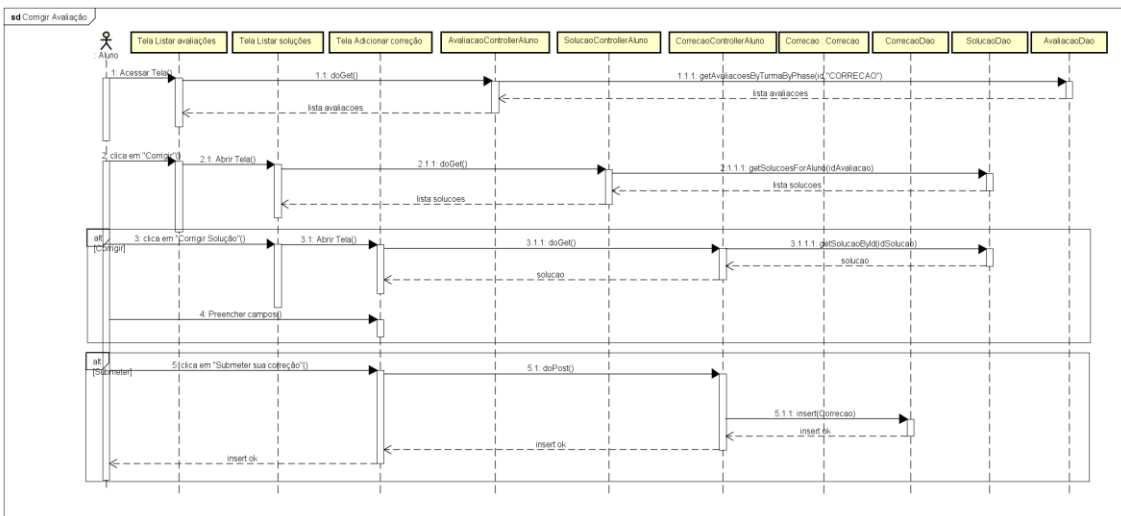
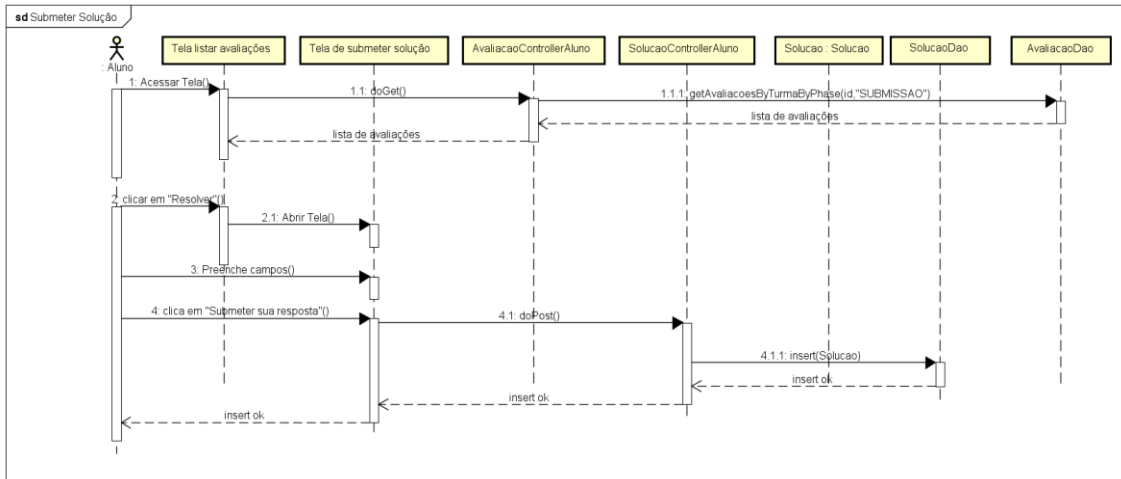












APÊNDICE 9 – DIAGRAMA DE ATIVIDADE

