

UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Bacharelado em Engenharia de Software - 4º Período

DISCIPLINA: *Oficina de Integração 1 - ES63G-ES41*

PROFESSOR: *Adriano Rivolli*

Documento de Projeto de Software

Aluno Tutor

Nelson Antonio Neto Toneze

Vanderson Oliveira

Lucas Malheiros

Cornélio Procópio

2019

SUMÁRIO

1. Introdução	4
1.2 Contexto	4
1.3 Justificativa	4
1.4 Proposta	5
1.5 Equipe de desenvolvimento	5
1.6 Organização do Documento	6
2. Descrição Geral do Sistema	7
2.1 Objetivos (Gerais e Específicos)	7
2.2 Limites e Restrições	7
2.3 Descrição dos Usuários do Sistema	8
3. Desenvolvimento do Projeto	9
3.1 Tecnologias e ferramentas	9
3.2 Metodologia de desenvolvimento	9
3.3 Cronograma previsto	13
4. Requisitos do Sistema	17
4.1 Requisitos Funcionais	17
4.2 Requisitos Não-funcionais	18
4.3 Diagramas de Casos de Uso	19
5. Análise do Sistema	24
6. Implementação	48
6.1 Protótipos de Telas	48
6.2 Descrição do código	52
6.3 Testes	55
7. Considerações Finais	58
8. Bibliografia	59

1 Introdução

1.1 Contexto

Em diversas universidades no meio acadêmico, existe uma modalidade onde os alunos podem se tornar monitores de determinadas matérias. O processo para o aluno se tornar monitor diversas vezes é ineficiente, isso acontece devido à demora do processo de seleção. O estudante que é selecionado para ser o monitor, fica responsável por auxiliar os acadêmicos registrados em tal matéria com as possíveis dúvidas que eventualmente surgirem. Na parte de organização, algumas salas da instituição são reservadas em horários pré-estabelecidos para que a monitoria ocorra. O aluno que deseja sanar uma dúvida, deve então deslocar-se até a sala onde o monitor se encontra.

Nesse tipo de situação também existem os alunos que possuem conhecimento em algumas disciplinas e que não participaram de editais para se tornarem monitor, ou que já estão inscritos como monitor, mas ainda possuem conhecimentos em outras matérias. Por conta disso, muitas vezes conhecimento que poderia vir a ser útil para seus colegas não é compartilhado, o que seria de muita ajuda tanto para quem tem dificuldades quanto para quem está ensinando, uma vez que ajuda na fixação de conteúdo.

1.2 Justificativa

Em algumas instituições de ensino, como a Universidade Tecnológica Federal do Paraná, são oferecidas aulas de monitoria em determinadas disciplinas como uma forma de ajuda para os alunos que possuem dificuldades. Entretanto, nem todos tem acesso a esse recurso pois os dias e horários são fixos e sem ajuste conforme a necessidade do aluno. Devido a isso alguns problemas são ocasionados como, por exemplo, o aluno não mora na cidade em que estuda e frequenta a faculdade apenas no horário noturno ou trabalha durante o dia ficando impossibilitado de usufruir de monitorias no período diurno. Além disso, há falta de informação e ausência de algumas disciplinas, pois muitas monitorias são canceladas devido à falta de busca, o que resulta em um déficit de matérias ofertadas.

Diante de todos os problemas apresentados, quem sente dificuldade e precisa de ajuda fica desamparado o que resulta na escolha de uma opção: desistir da disciplina e fazê-la novamente em outro semestre ou recorrer a vídeo aulas online, mas nem todos os discentes obtém o rendimento desejado estudando a distância e sabe-se que na universidade existem muitos alunos que possuem conhecimento e disposição para ajudar outros, mas não há nenhuma ferramenta que impulsiona essa interação, sendo assim os alunos que oferecem ajuda em uma matéria se anunciam por grupos nas redes sociais ou por mural na própria universidade.

E foi pensando em facilitar o encontro entre esses alunos, ou seja, quem desejam ensinar e quem deseja aprender que se criou a ideia do Aluno Tutor, uma plataforma web

que possibilitaria a realização de reuniões entre alunos que queiram se amparar. O aluno que se cadastrar poderá tanto ajudar, ou seja, se tornar tutor, quanto ser ajudado.

1.3 Proposta

A solução proposta é a criação de uma aplicação web que funcionará como uma plataforma de auxílio organizacional para aulas entre alunos. Os alunos serão capazes de criar um pedido de tutoria em uma matéria específica em uma espécie de lista de pedidos disponível para todos os tutores, procurar tutores separados em matérias ou tutores em especial, se inscrever em aulas ofertadas por tutores no site, conversar com o tutor por uma janela de chat para melhor entendimento de datas ou de matéria a ser ensinada e escrever uma análise do tutor.

O impacto esperado por esse sistema será notado principalmente por aqueles alunos que costumam procurar por ajuda em grupos de redes sociais, como o Facebook. Dessa forma, esses discentes terão um lugar mais prático para encontrar aqueles que estão dispostos a ajudá-los, além de ser organizado de uma maneira que facilitará uma seleção melhor de assuntos e qualificações.

Levando em consideração que o sistema pode proporcionar a opção do usuário que se cadastrar como aluno, também se tornar tutor, aumentam-se as funcionalidades do sistema para poder criar ofertas de aulas. Após combinado uma aula, aluno e tutor registrarão a aula, que será realizada em algum lugar a ser decidido por ambas as partes. Terminada a aula, os alunos poderão escrever uma análise do professor, e estes poderão escrever um relatório sobre a aula dada, assim como sua impressão do aprendizado do aluno.

O projeto pode ser encontrado no GitHub:
<https://github.com/AbyssalSire/oficina-integracao-1>.

1.4 Equipe de desenvolvimento

A equipe designada para projetar esta aplicação é composta por alunos de engenharia de software, sendo dois deles ex-alunos do curso de engenharia de computação.

Lucas Malheiros, responsável pela coordenação do código do projeto possui 21 anos, tem experiência adquirida em projetos da faculdade e estudos pessoais de C, C++, C#, Java, Javascript, HTML, MySQL e PostgreSQL. Foi o idealizador do projeto, tendo preferência por fazer a aplicação Web em HTML, CSS e Javascript/Node.js, mas depois de algumas deliberações com o grupo acabaram escolhendo Python para o backend.

Nelson Antonio Neto Toneze, aluno do quarto semestre de engenharia de software responsável pela análise dos diagramas e pela organização do documento, tem experiência

adquirida em disciplinas e projetos da faculdade e conhecimento em C, C++, C#, JAVA, MySQL e PostgreSQL. Contribuirá para a refinação do projeto no que se diz respeito à UML e documentação.

Vanderson Oliveira Silva Siqueira, aluno de engenharia de software responsável por gerenciar o cronograma do projeto e organização dos documentos. Experiência adquirido no setor de educação, conhecimento em JAVA, MySQL, C, C++, Python e documentos burocráticos relacionados às empresas ME, EPP e MEI. Em relação ao desenvolvimento se responsabilizará pelo o banco de dados, o teste de software e auxiliará os demais membros minimizando falha de erro durante o ciclo de produção do software.

1.5 Organização do Documento

Sendo este um documento provido pelo professor da disciplina, a organização deve seguir o padrão proposto. O padrão proposto é dividido em sete partes, divididas entre documentação, projeto de aplicação, montagem de aplicação e análise da aplicação montada.

A primeira parte, referente à introdução, aborda o cenário atual onde a ideia da aplicação surgiu, seguido por uma justificativa para sua idealização, passando então pela proposta do documento, uma breve apresentação dos membros do grupo e por fim a organização do documento.

Já a segunda parte abrange a descrição geral do sistema, sua documentação, um começo de tecnicidades. Inicialmente foca nos objetivos gerais e específicos do sistema, seus limites e restrições assim como uma descrição dos usuários do sistema.

O desenvolvimento do projeto apresentará as metodologias, tecnologias e ferramentas utilizadas assim como o cronograma previsto para o desenvolvimento.

A quarta parte irá focar na documentação para a estrutura do sistema iniciando com seus requisitos, tantos funcionais quanto não-funcionais que são os geradores dos casos de uso.

Continuando a documentação para desenvolvimento, na quinta parte, denominada Análise do Sistema, conterá o modelo do banco de dados, seus diagramas de classes e atividade.

Terminada a documentação, será então iniciada a parte de implementação, uma representação do programa criado. Protótipos de tela, descrição de telas e testes são o conteúdo esperado para esta seção.

Para finalizar haverá uma seção de considerações finais, onde serão discutidos as dificuldade de documentação, codificação e implementação, assim como uma auto-crítica do

resultado final e como o sistema se encaixaria no cenário descrito na primeira parte. E como de praxe a bibliografia está no final do documento.

2 Descrição Geral do Sistema

2.1 Objetivos (Gerais e Específicos)

Por meio do cenário mostrado na introdução deste documento, a plataforma web a ser desenvolvida tem como objetivo ajudar os alunos da Universidade Tecnológica Federal do Paraná a terem uma facilidade para encontrarem aulas particulares ensinadas por outros alunos.

O foco do projeto é auxiliar os estudantes da Universidade Tecnológica Federal do Paraná a ter acesso a um acervo catalogado de aulas particulares por meio de uma plataforma web. Nesta plataforma é possível a criação, por tutores, de ofertas de aulas através de reuniões. Qualquer estudante poderá tanto procurar quanto se inscrever em aulas, e caso se cadastre como tutor, poderá criar aulas para auxiliar outros alunos. A plataforma terá uma separação por categorias em seu layout, sendo estas categorias: grupos de estudo, aulas ofertadas por tutores e aulas requisitadas por alunos.

Com base em tudo o que foi visto até então, a aplicação deve:

- Ser capaz de cadastrar alunos e adicionar status de tutor para quem requisitar;
- Ser capaz de apresentar de forma satisfatória resultados de busca para tutores, matérias e grupos;
- Permitir que aluno e tutor possam interagir entre si;
- Possibilitar ao aluno avaliar um tutor com o qual tenha tido uma aula;
- Permitir com que aluno requisiite a participação em aulas de tutores, assim como cancelar com um dia de antecedência;
- Permitir que o tutor crie aulas categorizada por matéria e data;
- Permitir que aluno faça um pedido aberto de tutor de determinada matéria;
- Permitir que tutor se ofereça para ensinar um aluno que fez um pedido aberto;
- Ser capaz de executar as funções de forma adequada de acordo com o manuseio do usuário;
- Gerenciar os administradores que serão responsáveis pela manipulação da aplicação;

2.2 Limites e Restrições

2.2.1 Limites

- 2.2.1.1 Todas as pessoas que acessarem o site poderão ver as ofertas de aulas abertas, mas não terão a opção de se inscrever sem estar logado no sistema, visto que essa é uma função direcionada apenas para discentes que possuem cadastro;

2.2.1.2 Apenas alunos que requisitarem adicionar o status de tutor para suas contas poderão criar aulas;

2.2.1.3 Os desenvolvedores não serão responsabilizados pelas mensagens enviadas durante a interação virtual entre alunos e tutores que gerarem algum tipo de problema, assim como não serão responsáveis pelas ações dos mesmos fora do sistema, cabendo aos mesmos denunciarem pessoas que cometerem alguma ofensa, onde os desenvolvedores irão analisar a denúncia e tomar as medidas cabíveis;

2.2.1.4 O sistema não tem acesso ao banco de dados dos discentes da universidade, portanto para autenticação de registro acadêmico, irá validar apenas a quantidade de números contidos no mesmo.

2.2.1.5 O sistema garante que o registro acadêmico informado pelo discente possui formatação de um registro padrão, mas não consegue verificar a existência de tal aluno nem se o registro corresponde ao nome informado.

2.2.1.6 O sistema não se responsabiliza pela inadiplência de seus usuários em compromissos marcados;

2.2.1.7 Para propósitos de demonstração, o sistema não terá domínio registrado na rede mundial de computadores, sendo uma aplicação web hospedada no próprio computador que será usado para fazer a demonstração.

2.2.2 Restrições

2.2.2.1 O software funcionará em ambiente web, executado a partir de um navegador qualquer, sendo gerado por meio da extensão do VS Code chamada de Live Server Extension.

2.2.2.2 Como o sistema não estará vinculado à internet, a base de dados utilizada estará armazenada fisicamente em um dispositivo local.

2.3 Descrição dos Usuários do Sistema

Aluno: Público composto por estudantes que precisam de ajuda com disciplinas de seus respectivos cursos. Em específico os discentes da Universidade Tecnológica Federal do Paraná.

Tutor: Discentes da Universidade Tecnológica Federal do Paraná que tem conhecimento em alguma área e gostariam de compartilhar com algum aluno.

Administrador: Responsável pela manutenção geral do sistema e modificações feitas no mesmo. Apenas uma conta de administrador será concebida.

3 Desenvolvimento do Projeto

3.1 Tecnologias e ferramentas

- **Organização:** GoogleDrive, WhatsApp, GitHub;
- **Linguagens:** HTML, CSS, JavaScript, Node.js;
 - Bibliotecas: Bootstrap, Express.js;
- **Prototipagem:** Moqups.com;
- **Modelagem:** Astah Community, Astah, PostgreSQL;
- **Ferramenta de desenvolvimento:** Microsoft VS Code.
- **Banco de dados:** PostgreSQL.

3.2 Metodologia de desenvolvimento

O modelo SCRUM é conhecido por ser uma metodologia ágil, no qual ajuda a equipe monitorar, planejar e organizar para que o objetivo seja alcançado e cumprido no prazo estabelecido. Uma característica muito forte para a utilização do modelo SCRUM é a rotatividade de funções entre o time; uma pessoa da equipe não ficará designada apenas em uma parte do projeto, mas fará parte de todos os papéis e artefatos estabelecido no projeto. O modelo SCRUM é bastante usado em equipe com poucos integrantes melhorando a comunicação entre as partes envolvidas.

Para esse projeto o modelo SCRUM sofrerá uma mudança em relação ao original levando em considerações disponibilidade da equipe, tempo e conhecimento, alguns papéis serão incorporados por mais de um membro, o DAILY SCRUM sofrerá uma mudança em relação a frequência das reuniões.

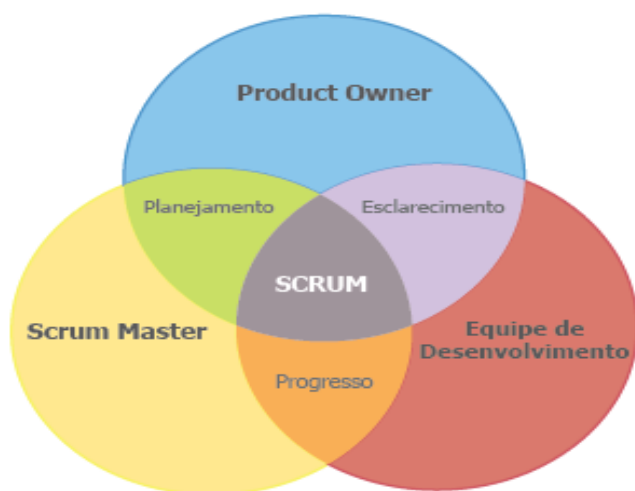


Figura 1 - Diagrama de VENN dos Stakeholders.

3.2.1 Scrum: Papéis, Eventos e Artefatos

3.2.1.1 PAPÉIS

- 3.2.1.1.1 **Product Owner:** Lucas Malheiro responsável por criar a visão do produto. Gerenciar o Backlog do Produto, priorizar as necessidades e compreender expectativas. É a voz dos possíveis clientes.
- 3.2.1.1.2 **Scrum Master:** Vanderson Oliveira, líder servidor, facilitador, removedor de impedimentos do Time Scrum e demais stakeholders. Garante que os eventos Scrum sejam executados. Responsável por manter o Scrum fluindo na organização.
- 3.2.1.1.3 **Time de Desenvolvimento:** Formado por equipe auto-organizada, responsável por realizar o trabalho que irá gerar a entrega da Sprint, possivelmente resultando em uma versão usável do produto. Seu tamanho será composto por quatro (4) integrantes sendo que o Product Owner e Scrum Master irá assumir papel no Time de Desenvolvimento. Nelson Antonio, responsável por orientar a equipe com relação às UML; Lucas Malheiros, responsável por orientar a equipe com relação ao desenvolvimento backend do sistema; Lara Dantas, responsável por orientar a equipe com relação ao desenvolvimento frontend; Vanderson Oliveira, responsável por orientar a equipe em relação ao banco de dados.

3.2.1.2 ARTEFATOS

- 3.2.1.2.1 **Product Backlog:** Lista dinâmica e ordenada de tudo que deve ser necessário no produto, incluindo todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões
- 3.2.1.2.2 **Sprint Backlog:** Itens do Product Backlog que foram selecionados para a Sprint. O Backlog da Sprint retrata os itens que estão sendo trabalhados e que estarão disponíveis no próximo incremento.
- 3.2.1.2.3 **Definition of Done:** Ao definir que um item está pronto, é preciso que todos tenham o mesmo conceito do que é pronto na determinada Sprint. O Time deve ter um entendimento

compartilhado do que significa o trabalho estar completo, assegurando a transparência. A definição de pronto é capaz de impactar diretamente na qualidade efetuada pelo time.

3.2.1.3 EVENTO

3.2.1.3.1 SPRINT

- **O que é:** Sprint é o coração do Scrum. Onde um incremento de produto, potencialmente utilizável é criado.
- **Duração:** No máximo 4 semanas.
- **Quem participa:** Time Scrum.

3.2.1.3.2 SPRINT PLANNING

- **O que é:** Reunião colaborativa do Time Scrum, onde é definido pelo Time de Desenvolvimento, os itens de Backlog do Produto que serão entregues ao final da Sprint. Estes itens selecionados pelo Time de Desenvolvimento são chamados da Sprint Backlog. Durante esta etapa de planejamento, é definido também a meta da sprint, ou seja, o objetivo definido para a Sprint que pode ser satisfeito através da implementação dos itens da Sprint Backlog.
- **Duração:** No máximo 1 horas.
- **Quem participa:** Time Scrum.

3.2.1.3.3 DAILY SCRUM

- **O que é:** Reuniões semanais no qual será feita durante as quartas-feiras com início às 21:20 horas e quintas-feiras com inícios às 19:00 horas, está sendo modificada de acordo com Reuniões diárias do modelo SCRUM ORIGINAL; executada pelo Time de Desenvolvimento e Scrum Master, onde objetiva-se o sincronismo das tarefas e planejamento para as próximas semana. Algumas questões importantes devem ser levadas para a Daily Scrum, como por exemplo: O que foi feito na semana anterior, o que será feito na semana atual e possíveis impedimentos que estão dificultando a conclusão de tarefas. A equipe poderá fazer um Daily Scrum no decorrer da semana caso se necessário.
- **Duração:** No máximo 15 minutos.
- **Quem participa:** Time de Desenvolvimento e Scrum Master.

3.2.1.3.4 SPRINT REVIEW

- **O que é:** Reunião realizada no último dia da Sprint, onde é demonstrado para o Product Owner e demais stakeholders, o incremento do produto criado. Focado no aceite ou rejeição do trabalho realizado. O resultado da Revisão da Sprint é um Backlog de Produto revisado, que define os itens prováveis de Backlog da próxima Sprint.

- **Duração:** No máximo 1 horas.
- **Quem participa:** Time Scrum e demais stakeholders que o Product Owner julgar necessário.

3.2.1.3.5 SPRINT RETROSPECTIVE

- **O que é:** Reunião que ocorre após a Revisão da Sprint. Focado em rever o processo de trabalho e o que poderá ser melhorado. Está relacionada a garantia da qualidade.
- **Duração:** No máximo 30 minutos.
- **Quem participa:** Time Scrum.

SCRUM

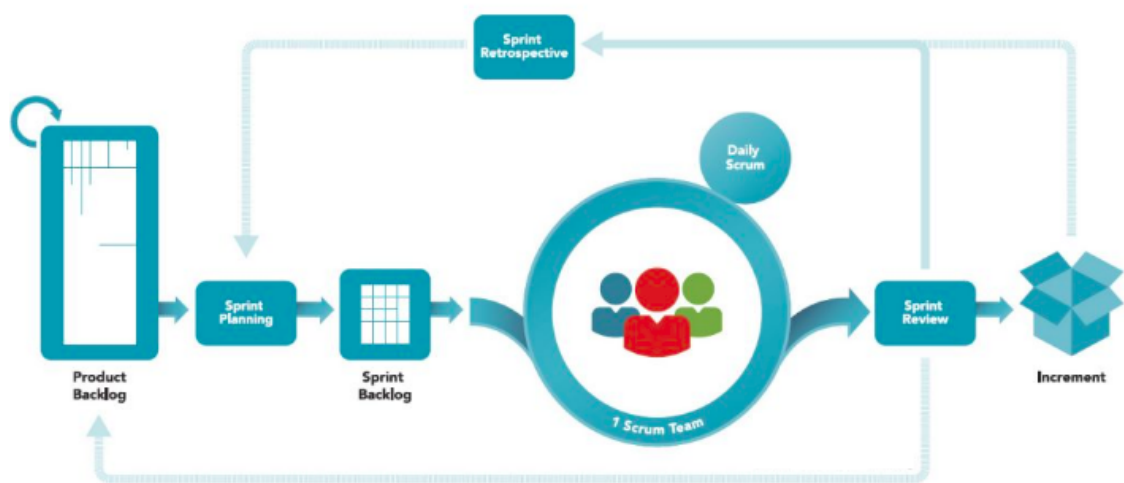


Figura 2 - Papéis, Eventos e Artefatos

3.3 Cronograma previsto

1ª S P R I N T	Objetivo	Elaboração do Tópico 1 Introdução	Data
			15/08/2019
	Responsável	Sprint Backlog	
	Lucas	Definição do Projeto	
	Nelson	Contexto	
	Nelson	Justificativa	
	Vanderson	Organização do Projeto	
	Sprint Review		21/08/2019
	Sprint Retrospective		22/08/2019

2ª S P R I N T	Objetivo	Elaboração do Tópico 2 Descrição Geral do Sistema	Data
			22/08/2019
	Responsável	Sprint Backlog	
	Nelson	Objetivos (gerais e específicos)	
	Lucas	Limites e Restrições	
	Nelson	Descrições do Usuário	
	Vanderson	Desenvolvimento do Projeto	
	Sprint Review		28/08/2019
	Sprint Retrospective		29/08/2019

3ª S P R I N T	Objetivo	Elaboração do Tópico 3 Desenvolvimento do Projeto	Data
			29/08/2019
	Responsável	Sprint Backlog	
	Time Scrum	Tecnologia e Ferramentas	
	Vanderson	Desenvolvimento do Projeto Metodologia de desenvolvimento	
	Sprint Review		04/09/2019
	Sprint Retrospective		05/09/2019

4ª S P R I N T	Objetivo	Elaboração do Tópico 4 Desenvolvimento do Projeto	Data 05/09/2019
	Responsável	Sprint Backlog	
	Time Scrum	Requisitos Funcionais	
	Time Scrum	Requisitos Não-Funcionais	
	Nelson	Diagrama de Caso de Uso Diagrama de Sequência	
	Nelson	Análise de Caso de Uso	
	Lucas	Diagrama de Atividades	
	Vanderson	Análise de Atividades	
	Sprint Review		18/09/2019
	Sprint Retrospective		19/09/2019

5ª S P R I N T	Objetivo	Elaboração do Tópico 5 Análise de Sistema Implementação I Banco de Dados	Data 19/09/2019
	Responsável	Sprint Backlog	
	Lucas	Modelo Conceitual do Banco de Dados	
	Nelson	Modelo Lógico do Banco de Dados	
	Vanderson	Modelo Físico do Banco de Dados	
	Nelson	Documentação do Modelo de Banco de Dados, Diagrama de Classe, Diagrama de Atividades.	
	Sprint Review		02/10/2019
	Sprint Retrospective		03/10/2019

6ª S P R I N T	Objetivo	Elaboração do Tópico 6 Análise de Sistema Implementação II Banco de Dados Prototipação da Tela Desenvolvimento Backend Desenvolvimento Frontend	Data 03/10/2019
	Responsável	Sprint Backlog	
	Nelson	Estudo de padrão de telas (prototipação) Programação Frontend (Tela de Login)	
	Nelson	Declarar Classes no Programa Programação Frontend (Tela Inicial e Tela de Cadastro de Aluno)	

	Vanderson	Conectividade de Banco de Dados Conforme Telas do Sistema Programação Backend (Solicitação de Aulas Disponíveis)
	Lucas	Código do Protótipo de Tela Programação Backend (Cadastro dos Alunos, Modificação de Conta para Tutor, Opção Para oferecer Tutela, Interação entre os Usuários)
	Sprint Review	
	16/09/2019	
	Treinamento de Capacitação da Equipe	
	07 á 09	
Sprint Retrospective (Reorganização do Projeto)		
17/10/2019		

*Treinamento de Capacitação da equipe será para que os alunos busquem conhecimento, ensine os demais colaboradores e demonstre quais dúvida sobre as tecnologias e ferramentas a serem utilizada.

7 ^a S P R I N T	Objetivo	Documentação Tópico 6 Implementação do Sistema III Desenvolvimento Frontend Desenvolvimento Backend Implementação da Tela	Data
			17/10/2019
	Responsável	Sprint Backlog	
	Nelson	Documentação Inicial do Tópico 6 Programação Frontend (Tela de Histórico de Tutela) Programação Backend (Função Gerar Histórico de Tutelas Para o Aluno) Estudo de padrão de telas (usabilidade)	
	Nelson	Programação Frontend (Filtro de Busca de Disciplina, Filtro de Busca por Tutor, Filtro por Categoria)	
	Vanderson	Programação Backend (Função Matéria por Categoria, Avaliação do Tutor, Sistema de Busca)	
	Lucas	Programação Backend (Função Alterar Aulas por Parte do Tuto, Função Cancelar pedidos de Aulas do Aluno, Função Cancelamento de Oferta Pelo Tutor, Função para o Usuário Criar Grupo de Estudos, Função Para o Usuário entrar no Grupo de Estudos)	
	Sprint Review		06/11/2019
	Sprint Retrospective		07/11/2019

8 ^a S P R I N T	Objetivo	Implementação do Sistema IV Desenvolvimento Frontend Desenvolvimento Backend Implementação da Tela Testes	Data
			07/11/2019
	Responsável	Sprint Backlog	
	Time Scrum	Testes iniciais	

	Nelson	Programação Backend (Função de Denunciar um Usuário, Opção de Filtrar Busca)	
	Nelson Vanderson Lucas	Pesquisa e Feedback com a comunidade Acadêmica (mostrar para alguns alunos o sistema)	
	Sprint Review		13/11/2019
	Sprint Retrospective		14/11/2019

9ª S P R I N T	Objetivo	Implementação do Sistema V Desenvolvimento Frontend Desenvolvimento Backend Ajustes da Tela Testes	Data 14/11/2019
	Responsável	Sprint Backlog	
	Time Scrum	Testes intermediários (Backend, Frontend e Telas)	
	Sprint Review		20/11/2019
	Sprint Retrospective		21/11/2019

1 0 ª S P R I N T	Objetivo	Implementação do Sistema VI Finalização Front End Finalização Backend Finalização da Tela Testes	Data 21/11/2019
	Responsável	Sprint Backlog	
	Nelson Lucas Vanderson	Testes finais Correção bugs e revisão Revisão dos Documentos	
	Sprint Review		27/11/2019
	Sprint Retrospective		28/11/2019

4 Requisitos do Sistema

4.1 Requisitos Funcionais

ID	Funcionalidade	Prioridade
RF01	O sistema deve permitir que o usuário faça seu cadastro	Essencial
RF02	O sistema deve possuir matérias por categorias	Importante
RF03	O sistema deve permitir que haja interação entre usuários	Essencial
RF04	O sistema deve permitir que o usuário possa modificar a sua conta para se tornar tutor	Essencial
RF05	O sistema deve conceder a opção de oferecer tutelas	Essencial
RF06	O sistema deve permitir que o aluno solicite aulas disponíveis	Essencial
RF07	O sistema deve permitir alterações nas aulas ofertadas por parte dos tutores	Importante
RF08	O sistema deve permitir que o aluno cancele pedidos de aulas	Importante
RF09	O sistema deve permitir que o tutor cancele sua aula ofertada	Importante
RF10	O sistema deve possibilitar que o aluno avalie um tutor	Importante
RF11	O sistema deve possuir um sistema de busca	Importante
RF12	O sistema deve permitir que o aluno faça um chamado de procura de aulas	Desejável
RF13	O sistema deve permitir que o aluno que está em processo de chamado de procura de aulas negue uma oferta de aula do tutor	Importante
RF14	O sistema deve permitir que o aluno que está em processo de chamado de procura de aulas aceite uma oferta de aula do tutor	Importante
RF15	O sistema deve permitir que o usuário crie um grupo de estudo	Importante
RF16	O sistema deve permitir que o usuário entre em um grupo de estudos	Importante
RF17	O sistema deve permitir que o usuário saia de um grupo de estudos	Importante
RF18	O sistema deve possibilitar a opção de filtragem de busca	Importante
RF19	O sistema deve permitir a opção de denunciar um usuário	Desejável
RF20	O sistema deve possibilitar a geração de um histórico de tutelas para o aluno	Importante

RF2 1	O sistema deve possibilitar a geração de um histórico de tutelas para o tutor	Importante
RF2 2	O sistema deve permitir que o administrador faça alterações na aplicação	Importante

4.2 Requisitos Não-funcionais

Descrever os requisitos não-funcionais do sistema, que especificam restrições sobre os serviços ou funções providas pelo sistema, categorizando de acordo com a característica envolvida, como: Usabilidade, Padronização, Ambiente, Compatibilidade, Recursos, etc.

ID	Requisito	Categoria
RNF0 1	O sistema só pode ser acessado via aplicação web	Ambiente
RNF0 2	O sistema deve ser desenvolvido utilizando HTML, CSS, JavaScript e Node.js	Implementação
RNF0 3	O sistema deve ser implementado na data estipulada pelo professor da disciplina	Entrega

4.3 Diagramas de Casos de Uso

UC1 - Gerenciamento do Administrador

Requisitos atendidos: RF01, RF19, RF22.

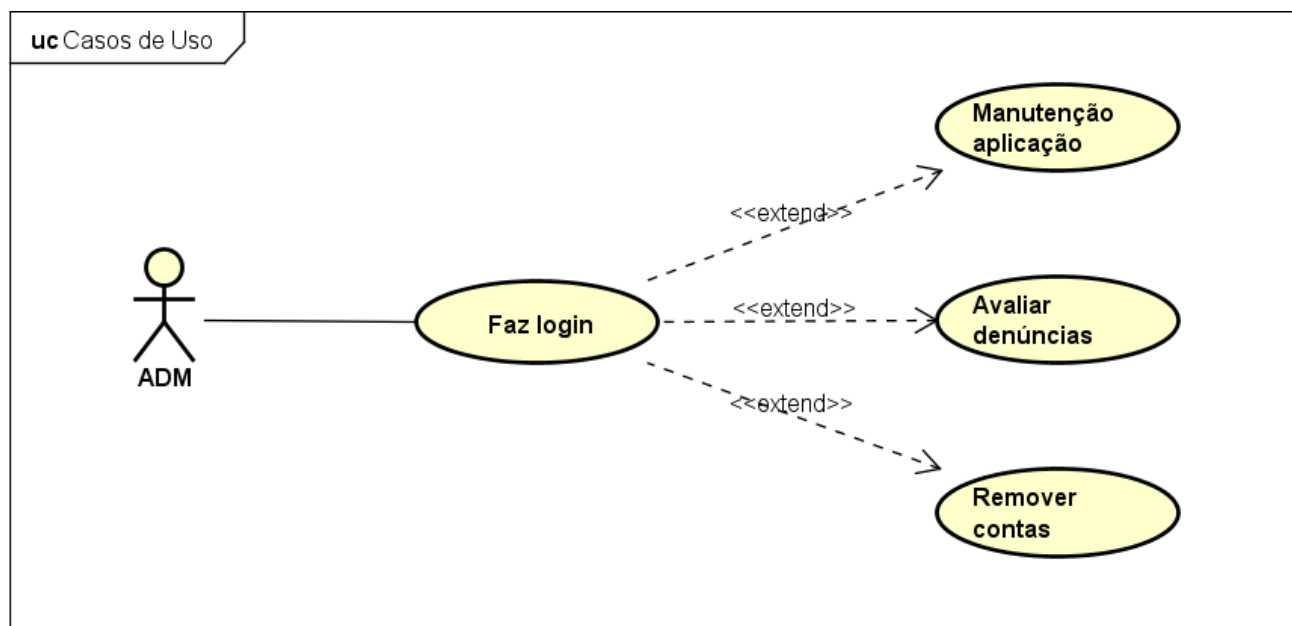


Figura 3 - Caso de uso da manutenção do sistema

ESPECIFICAÇÃO DO CASO DE USO - Manutenção aplicação

Cenário 1- Fazer manutenção geral da aplicação.

Descrição	Deverá dar as opções para o administrador fazer as devidas modificações no sistema.
Autor Principal	Administrador.
Autor Secundário	Sistema.
Pré- Condições	O sistema deve estar ligado ao Banco de dados; O administrador estar logado em conta com privilégios para modificações.
Pós- Condições	
Ações Normais	<ol style="list-style-type: none"> 1. O usuário (administrador) clica na aba de manutenção. 2. O sistema deve apresentar as opções disponíveis para a realização das modificações desejadas. 3. As modificações feitas devem ser implementadas pelo sistema e guardadas no banco de dados.
Observação	

UC2 - Geral Tutor

Requisitos atendidos: RF02, RF03, RF05, RF07, RF09.

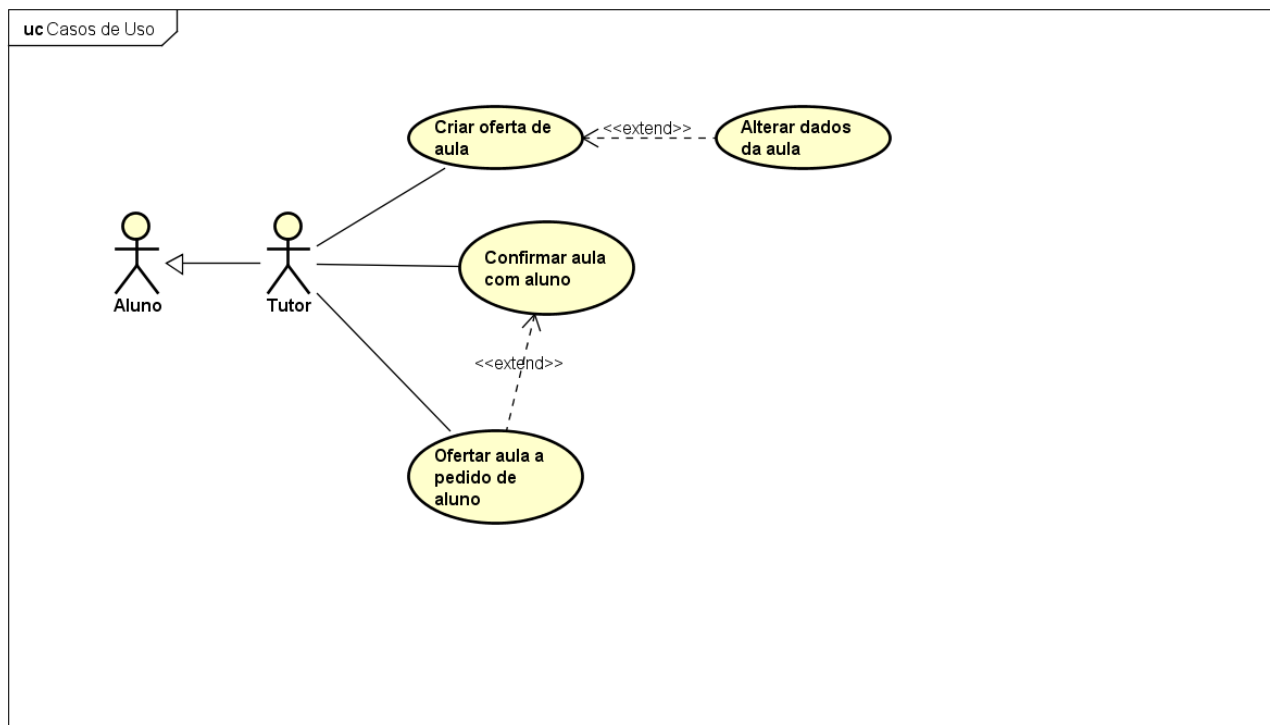


Figura 4 - Caso de uso das funcionalidades de tutor

ESPECIFICAÇÕES DOS CASOS DE USO- Criar oferta de aula

Cenário 1- Criar oferta de aula.

Descrição	Deverá criar uma aula ofertada por um tutor com os detalhes necessários.
Autor Principal	Tutor
Autor Secundário	Sistema
Pré- Condições	O sistema deve estar ligado ao Banco de dados; Aluno estar logado com conta modificada para tutoriar.
Pós- Condições	
Ações Normais	<ol style="list-style-type: none"> 1. O tutor clica na aba de tutoria. 2. O sistema deve apresentar a opção para ofertar aula.
Observações	

Cenário 2- Inserir detalhes da aula ofertada.

Descrição	Deverá dar a opção de detalhar a aula que está sendo ofertada.
Autor Principal	Tutor
Autor Secundário	Sistema
Pré- Condições	O sistema deve estar ligado ao Banco de dados; Aluno estar logado com conta modificada para tutoriar.
Pós- Condições	
Ações Normais	<ol style="list-style-type: none"> 1. O tutor clica na aba de tutoria. 2. O sistema deve apresentar as aulas que o tutor está ofertando. 3. O sistema deve mostrar os campos para preenchimento dos detalhes da aula ofertada.
Observações	

ESPECIFICAÇÕES DOS CASOS DE USO- Ofertar aula a pedido de aluno

Cenário 1- Ofertar aula para um pedido de aluno.

Descrição	Deverá ofertar uma aula para um pedido de aluno
Autor Principal	Tutor
Autor Secundário	Sistema
Pré- Condições	O sistema deve estar ligado ao Banco de dados; Aluno estar logado com conta modificada para tutoriar.
Pós- Condições	
Ações Normais	<ol style="list-style-type: none"> 1. O tutor clica na aba de tutoria. 2. O sistema deve apresentar a opção para ofertar aula para um pedido de aluno existente.
Observações	

UC3 - Geral Aluno

Requisitos atendidos: RF01, RF02, RF03, RF04, RF06, RF08, RF10, RF11, RF12, RF13, RF14, RF18, RF19, RF20, RF21.

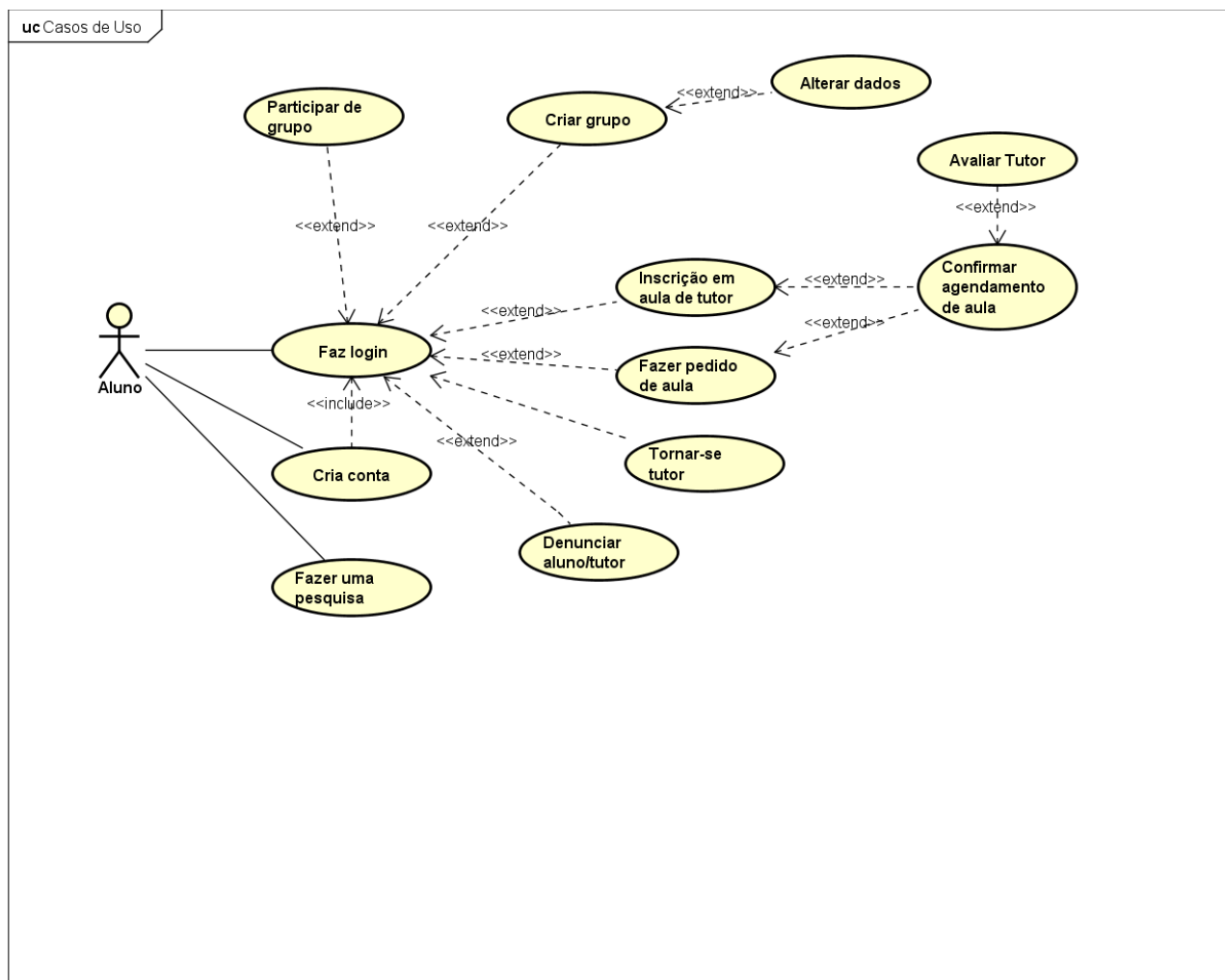


Figura 5 - Caso das funcionalidades de tutor

ESPECIFICAÇÕES DOS CASOS DE USO- Inscrição em aula de tutor

Cenário 1- Se inscrever em uma aula disponível.

Descrição	Deverá inscrever o aluno em uma aula que esteja disponível.
Autor Principal	Aluno
Autor Secundário	Sistema
Pré- Condições	O sistema deve estar ligado ao Banco de dados; Aluno estar logado com sua conta.
Pós- Condições	

Ações Normais	<ol style="list-style-type: none"> 1. O aluno deve clicar na aba aulas ofertadas. 2. O aluno deve clicar em uma das aulas para fazer a inscrição.
Observações	

ESPECIFICAÇÕES DOS CASOS DE USO- Fazer pedido de aula

Cenário 1- Criar um pedido de uma aula.

Descrição	Deverá permitir o aluno a criar um pedido de uma determinada aula.
Autor Principal	Aluno
Autor Secundário	Sistema
Pré- Condições	O sistema deve estar ligado ao Banco de dados; Aluno estar logado com sua conta.
Pós- Condições	
Ações Normais	<ol style="list-style-type: none"> 1. O aluno deve clicar na opção fazer pedido de aula. 2. O sistema deverá mostrar as opções para detalhar o pedido de aula. 3. O sistema deverá criar um pedido de aula.
Observações	

ESPECIFICAÇÕES DOS CASOS DE USO- Tornar-se tutor

Cenário 1- Modificar a conta para ser tutor.

Descrição	Deverá permitir o aluno ter uma conta de tutor.
Autor Principal	Aluno
Autor Secundário	Sistema
Pré- Condições	O sistema deve estar ligado ao Banco de dados; Aluno estar logado com sua conta.
Pós- Condições	
Ações Normais	<ol style="list-style-type: none"> 1. O aluno deve clicar na opção tornar-se tutor. 2. O sistema deve modificar a conta do aluno para tutor.
Observações	

5 Análise do Sistema

- Modelo do Banco de Dados

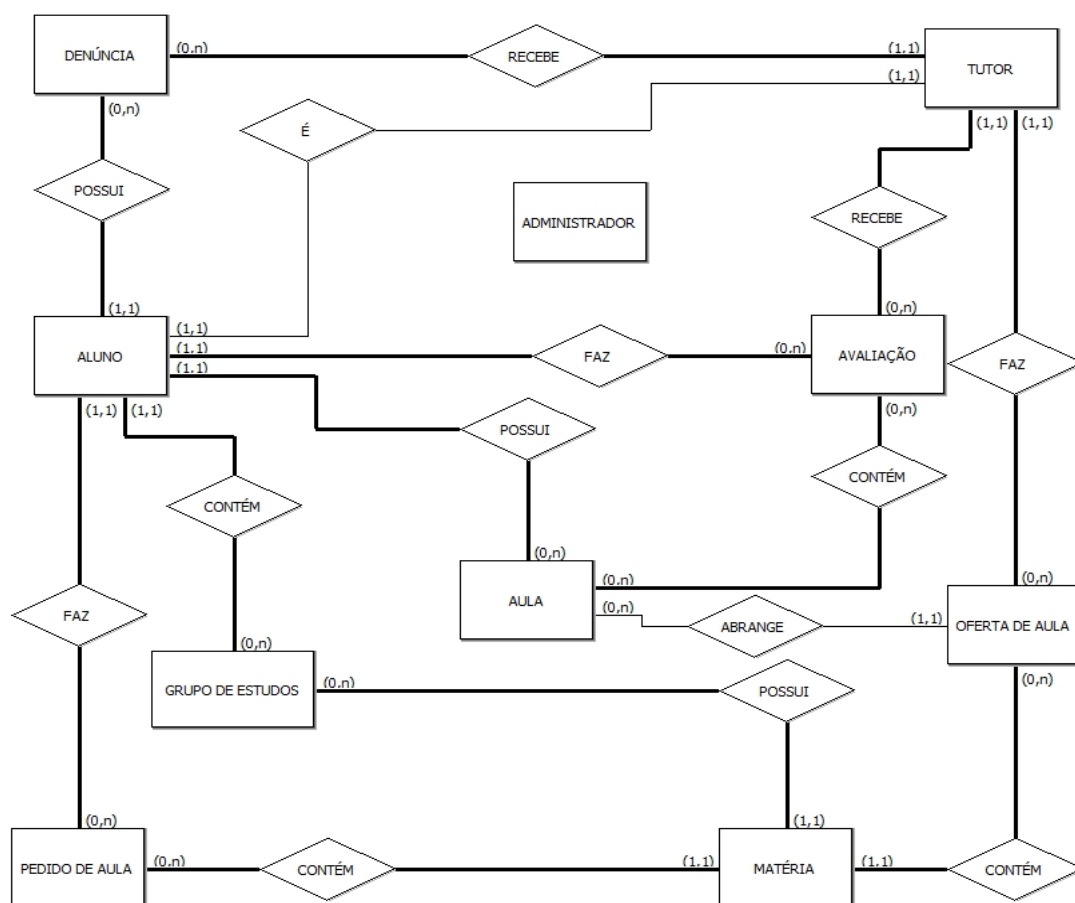


Figura 6 - Modelo Conceitual do banco de dados

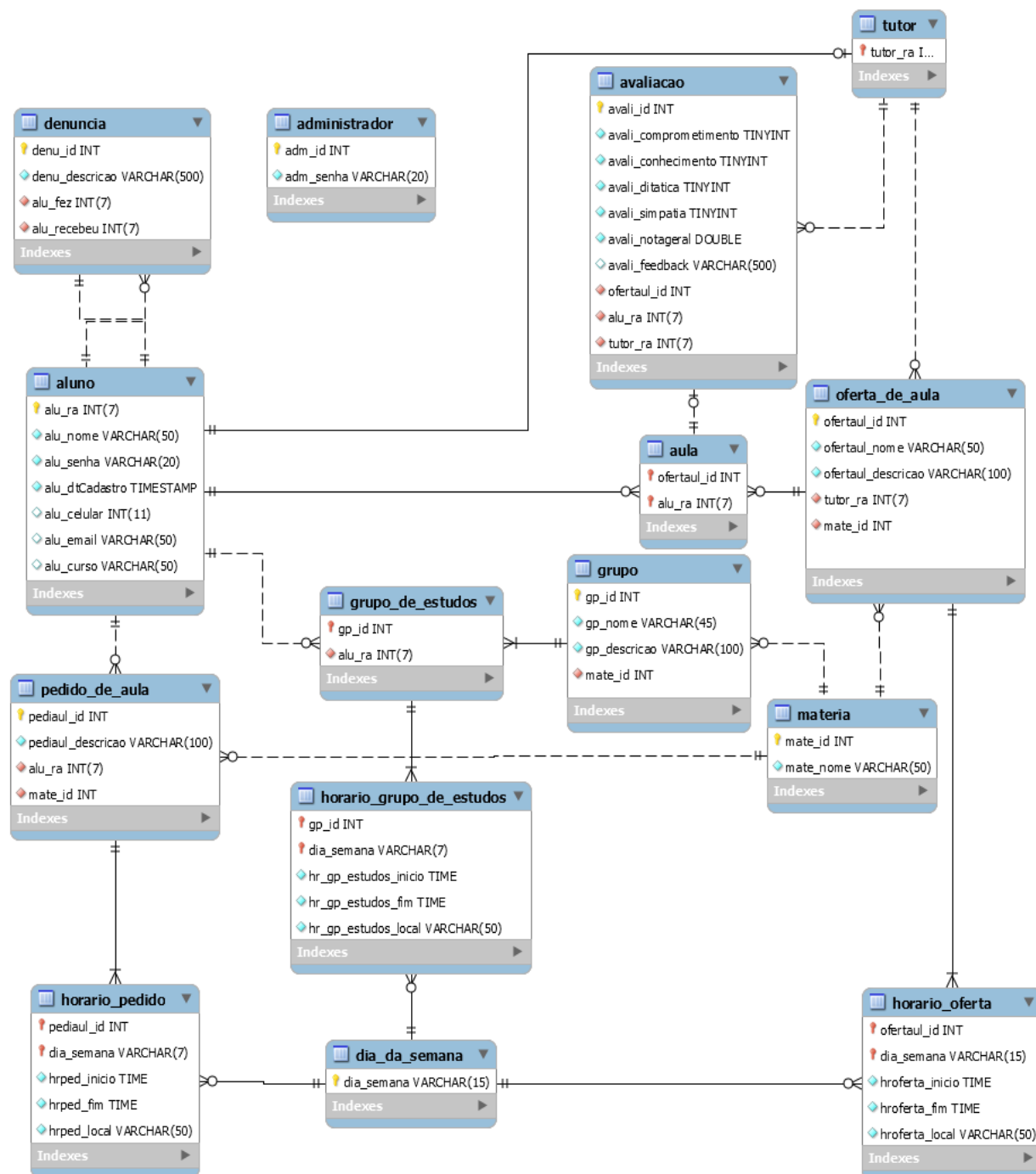


Figura 7 - Modelo Lógico do Banco de Dados

Tabela		Administrador				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
adm_id	INT		PK		Sim	Identificação do administrador
adm_senha	VARCHAR	20			Sim	Senha do administrador

Tabela		Aluno				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
alu_ra	INT	7	PK		Sim	Registro acadêmico do aluno, servirá como chave primária pois é único dado a cada aluno pela instituição.
alu_nome	VARCHAR	50			Sim	Nome completo do aluno
alu_senha	VARCHAR	12			Sim	Senha criada pelo usuário, preferencialmente a mesma utilizada pelo mesmo para acessar os domínios da faculdade
alu_dtCadastro	TIMESTAMP					Data de cadastro do aluno
alu_celular	INT	11			Não	DDD + Celular do aluno
alu_email	VARCHAR	50			Não	Email do usuário, preferencialmente o e-mail acadêmico
alu_curso	VARCHAR	50			Não	Curso cursado pelo aluno

Tabela		Denúncia				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição

denu_id	INT		PK		Sim	Identificação da denúncia
denu_descricao	VARCHAR	500			Sim	Descrição escrita da denúncia
alu_fez	INT	7			Sim	Registro acadêmico do aluno que recebeu a denúncia
alu_recebeu	INT	7			Sim	Registro acadêmico do aluno que fez a denúncia

Tabela		Tutor				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
tutor_ra	INT	7	PK	FK	Sim	Mesmo número de alu_ra, chave estrangeira herdada

Tabela		Oferta de aula				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
ofertaul_id	INT		PK	FK	Sim	Identificação da oferta de aula
Ofertaul_nome	VARCHAR	50			Sim	Nome da oferta de aula
mate_id	INT				Sim	Identificação da disciplina da aula
tutor_ra	INT	7			Sim	Registro acadêmico do tutor
ofertaul_descricao	VARCHAR	100			Sim	Descrição da oferta de aula

Tabela		Horário_oferta				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição

ofertaul_id	INT		PK	FK	Sim	Mesma Identificação da oferta_de_aula
dia_semana	VARCHAR	7		FK	Sim	Mesma identificação da tabela dia_da_semana
hrofert_inicio	TIME				Sim	Horário de início da aula
hrofert_fim	TIME				Sim	Horário de fim da aula
hrofert_local	VARCHAR	50			Sim	Local da oferta de aula

Tabela		Dia_da_semana				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
dia_semana	VARCHAR	7	PK		Sim	Identificação dos dias da semana (segunda, terça, quarta, quinta, sexta, sábado e domingo)

Tabela		Aula				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
ofertaul_id	INT			FK	Sim	Identificação da oferta de aula
alu_ra	INT	7		FK	Sim	Registro acadêmico do aluno

Tabela		Avaliação				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
avali_id	INT		PK		Sim	Número ID da avaliação, gerado pelo sistema por meio de auto incremento
avali_compro	INT				Sim	Nota de avaliação

metimento						conferida ao tutor por seu comprometimento durante a aula
avali_conhecimento	INT				Sim	Nota de avaliação conferida ao tutor por seu conhecimento durante a aula
avali_notageral	INT				Sim	Nota de avaliação geral para a aula
avali_didatica	INT				Sim	Nota de avaliação pela didática apresentada pelo tutor
avali_simpatia	INT				Sim	Nota pela simpatia do tutor
avali_feedback	VARCHAR	500			Não	Avaliação escrita da aula, também serve para dar críticas construtivas para melhora da aula do tutor
alu_ra	INT	7		FK	Sim	RA do aluno que teve a aula
tutor_ra	INT	7		FK	Sim	RA do tutor que deu a aula
ofetaul_id	INT			FK	Sim	ID da oferta de aula que gerou a aula em questão

Tabela		Matéria				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
mate_id	INT		PK		Sim	Identificação da disciplina
mate_nome	VARCHAR	50			Sim	Nome da disciplina

Tabela		Grupo				
Campo	Tipo	Tamanho	PK	FK	Obrigat	Descrição

		o			ório	
gp_id	INT		PK		Sim	ID do grupo de estudos
mate_id	INT			FK	Sim	ID da matéria a qual o grupo em questão abordará em suas sessões
gp_nome	VARC HAR	50			Sim	Nome do grupo
Gp_descricao	VARC HAR	100			Sim	Descrição geral do grupo

Tabela		Grupo_de_estudos				
Campo	Tipo	Tam anh o	PK	FK	Obrigat ório	Descrição
gp_id	INT			FK	Sim	Mesma identificação da tabela Grupo
alu_ra	INT	7			Sim	Identificação dos alunos no grupo

Tabela		Horario_grupo_de_estudos				
Campo	Tipo	Tam anh o	PK	FK	Obrigat ório	Descrição
gp_id	INT			FK	Sim	Mesma Identificação da tabela grupo_de_estudos
dia_semana	VARC HAR	7		FK	Sim	Mesma identificação da tabela dia_da_semana
hr_gp_estudos_inicio	TIME				Sim	Hora de inicio dos estudos em grupos
hr_gp_estudos_fim	TIME				Sim	Hora de termino dos estudos em grupos
hr_gp_estudos_local	VARC HAR	50			Sim	Local de estudos em grupos

Tabela		Pedido de aula				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
pediaul_id	INT		PK		Sim	Identificação do pedido de aula
pediaul_descricao	VARCHAR	100			Sim	Descrição do pedido de aula
mate_id	INT				Sim	Identificação da matéria
alu_ra	INT	7			Sim	Registro acadêmico do aluno

Tabela		Horario_pedido				
Campo	Tipo	Tamanho	PK	FK	Obrigatório	Descrição
pediaul_id	INT			FK	Sim	Mesma Identificação da tabela pedido_de_aula
dia_semana	VARCHAR	7		FK	Sim	Mesma Identificação da tabela dia_da_semana
hrped_inicio	TIME				Sim	Horário de início de disponibilidade do aluno
hrped_fim	TIME				Sim	Horário de término de disponibilidade do aluno

- Diagrama de Classes

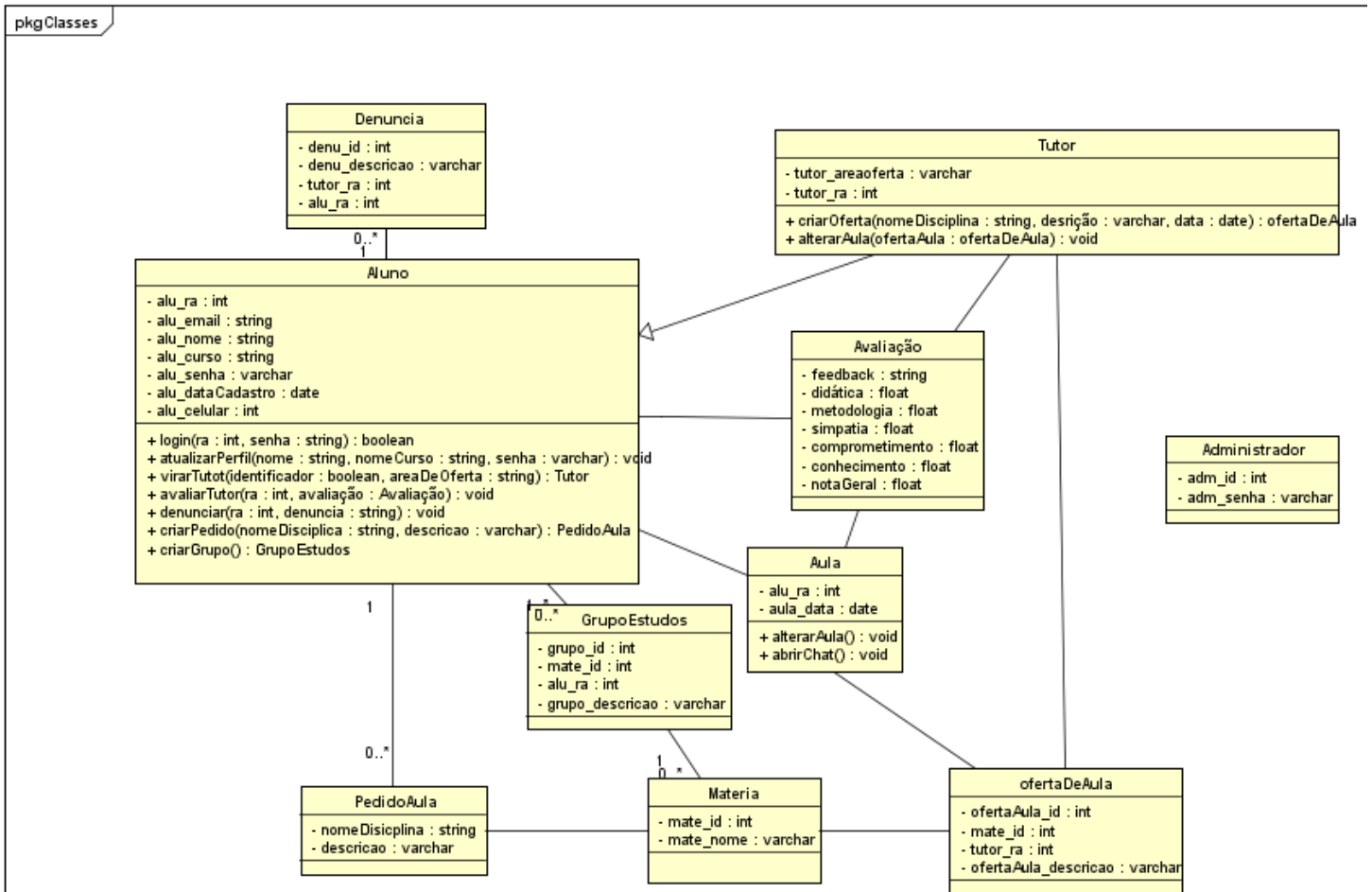


Figura 8 - Diagrama de classes geral

- Diagrama de Atividades

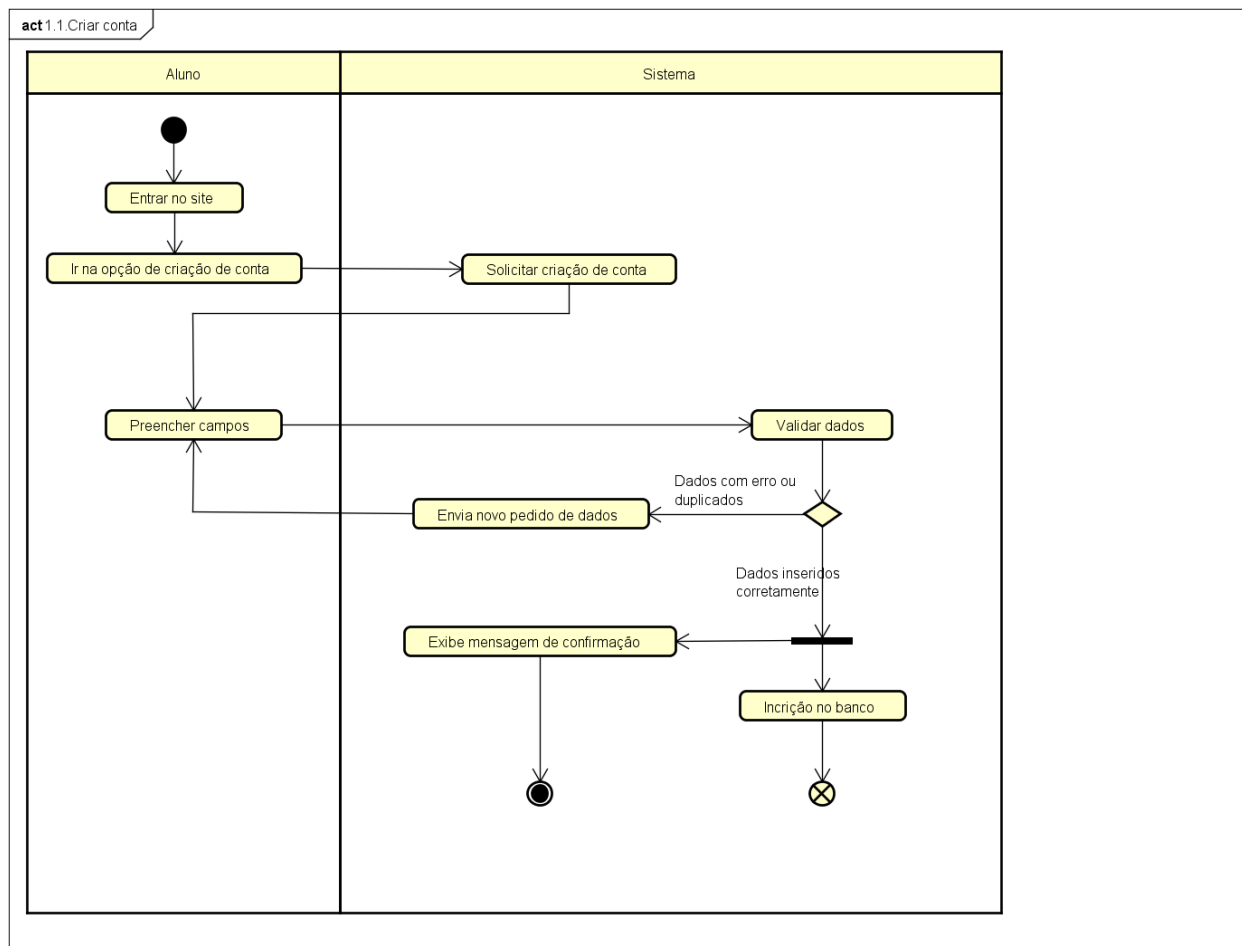
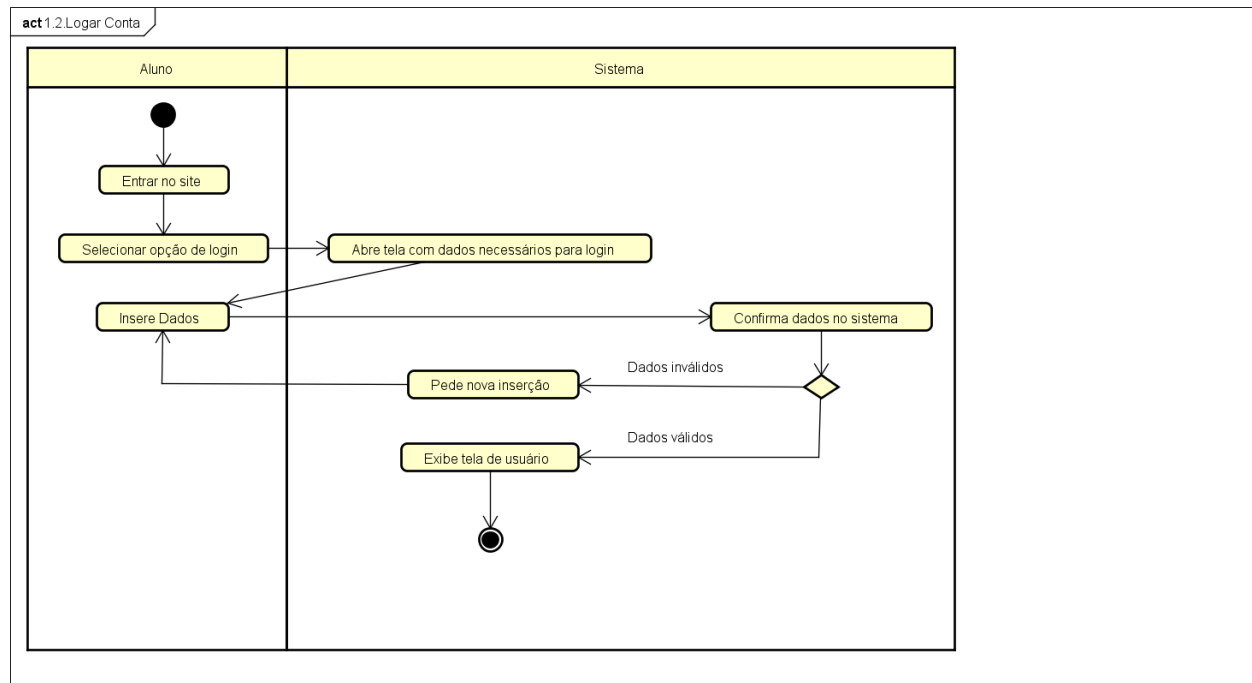
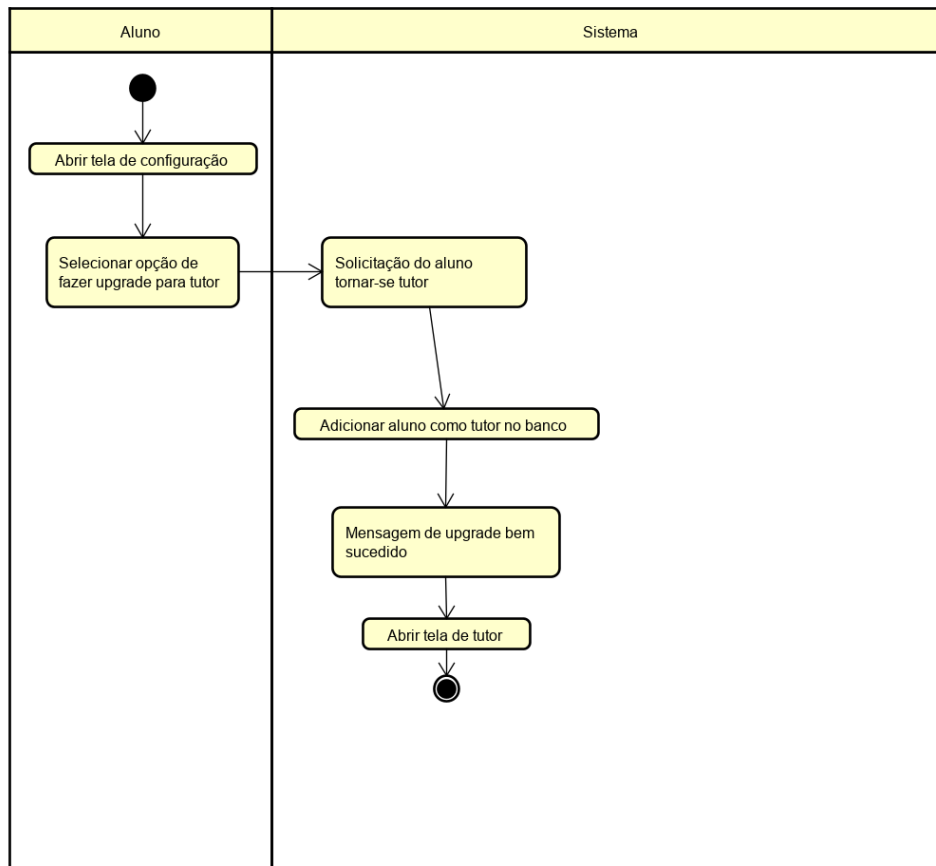
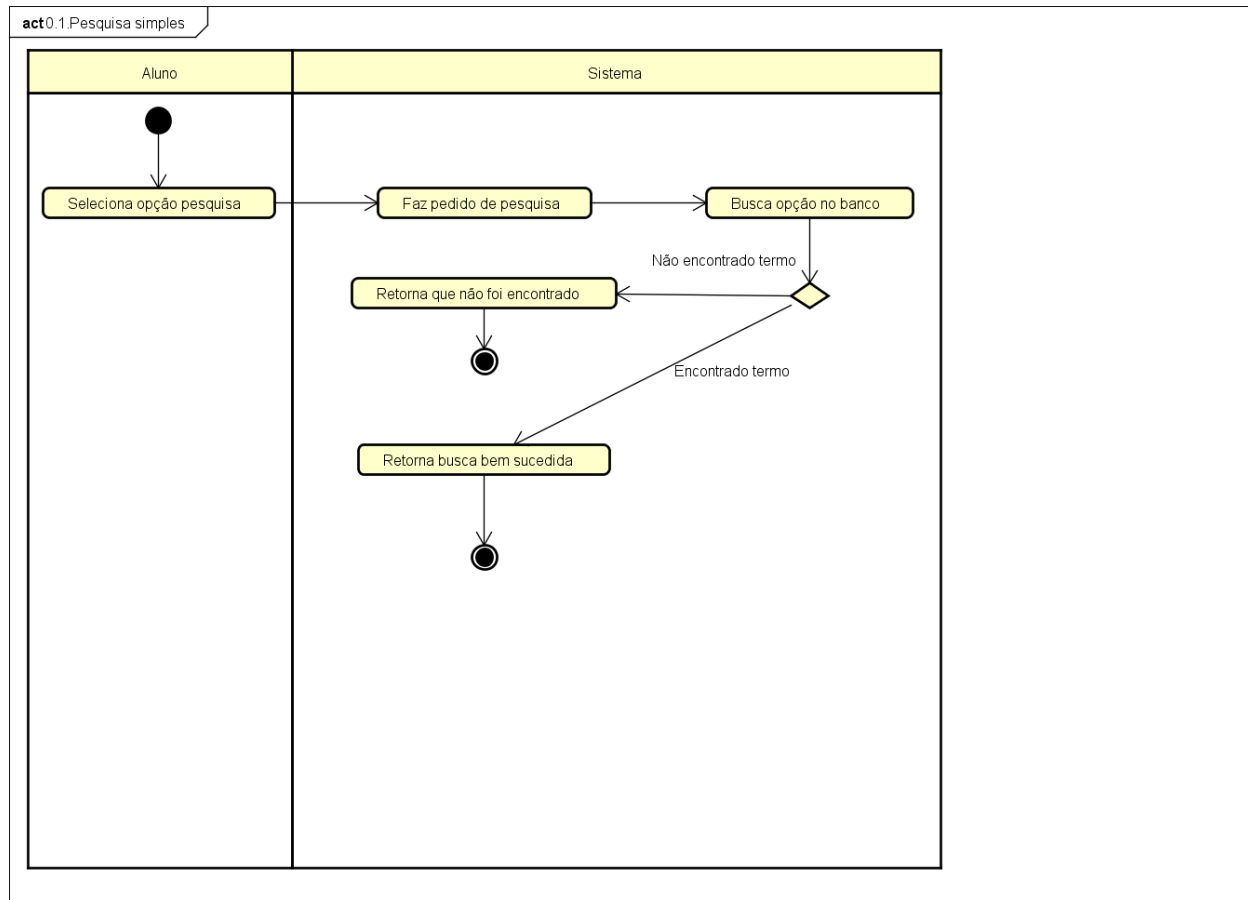


Figura 9 - Diagrama de atividade criar conta

**Figura 9 - Diagrama de atividade logar conta**

act 1.3.Alteração conta para tutor

**Figura 10 - Diagrama de atividade de alteração conta para tutor**

**Figura 11 - Diagrama de atividade de pesquisa simples**

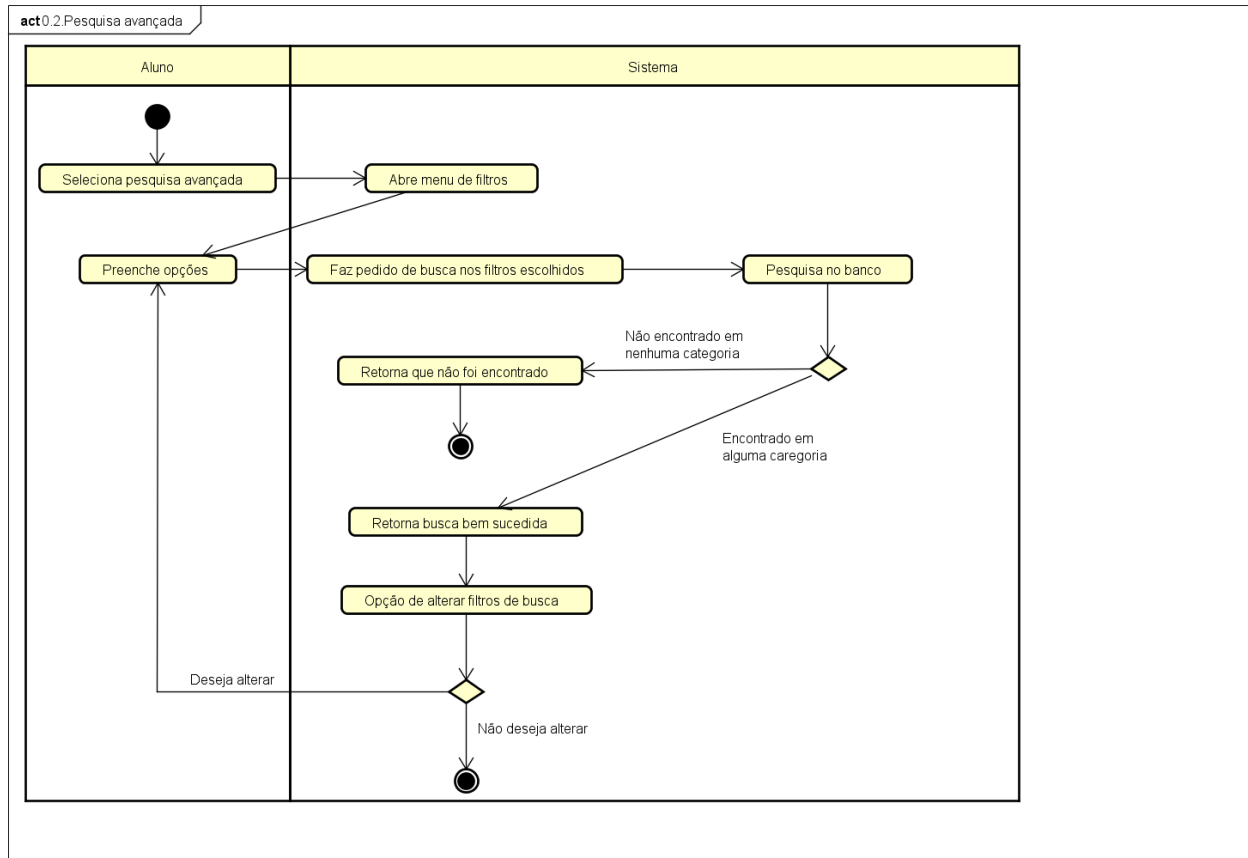
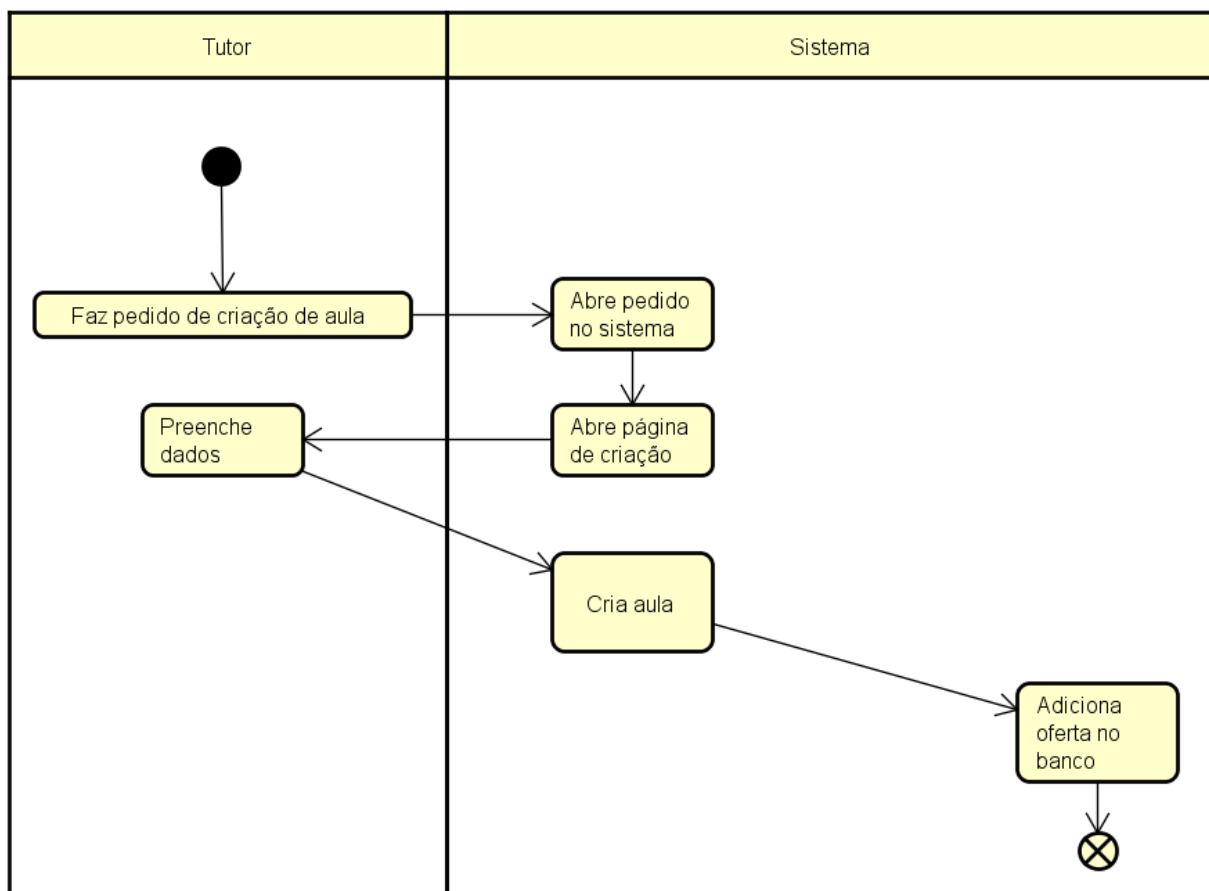
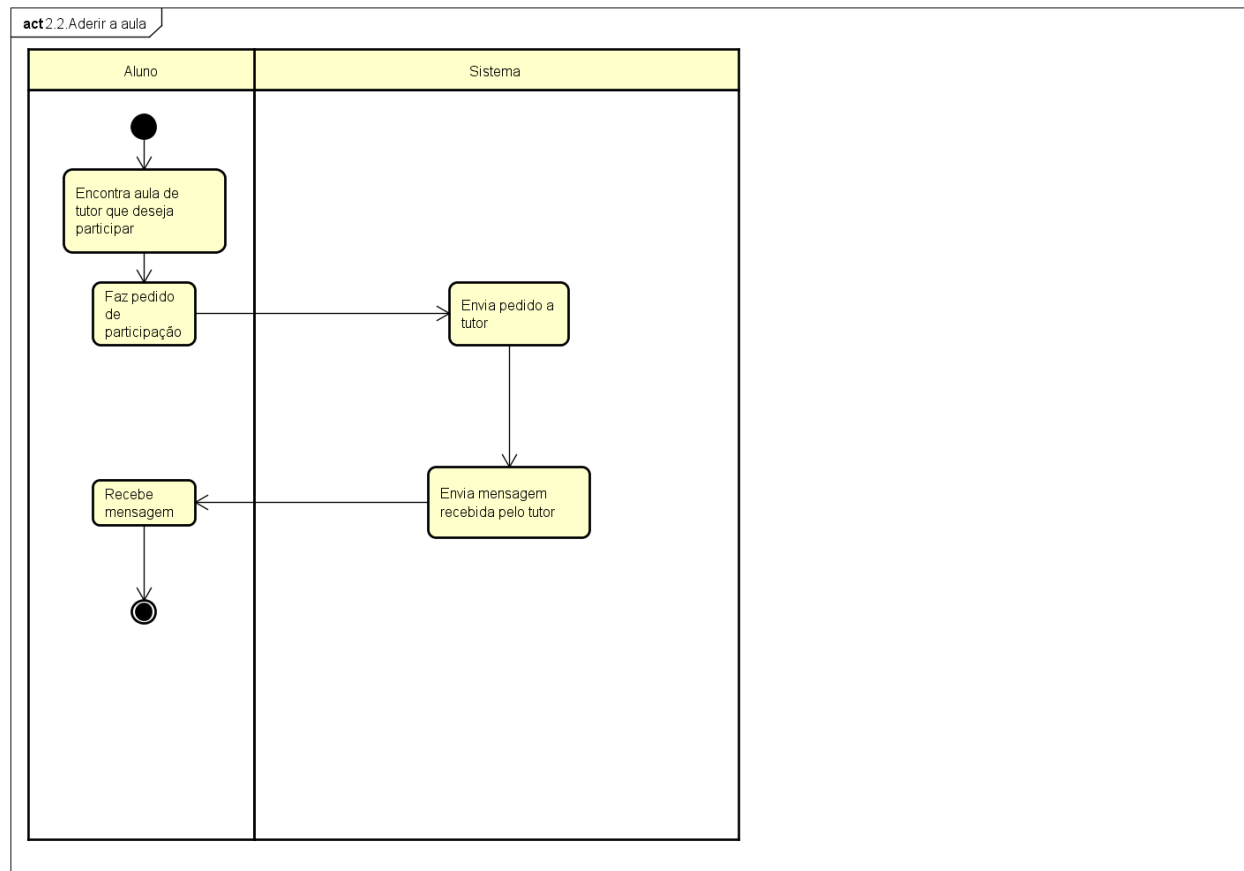


Figura 12 - Diagrama de atividade de pesquisa avançada

**Figura 13 - Diagrama de atividade de criar aula**

**Figura 14 - Diagrama de atividade de aderir aula**

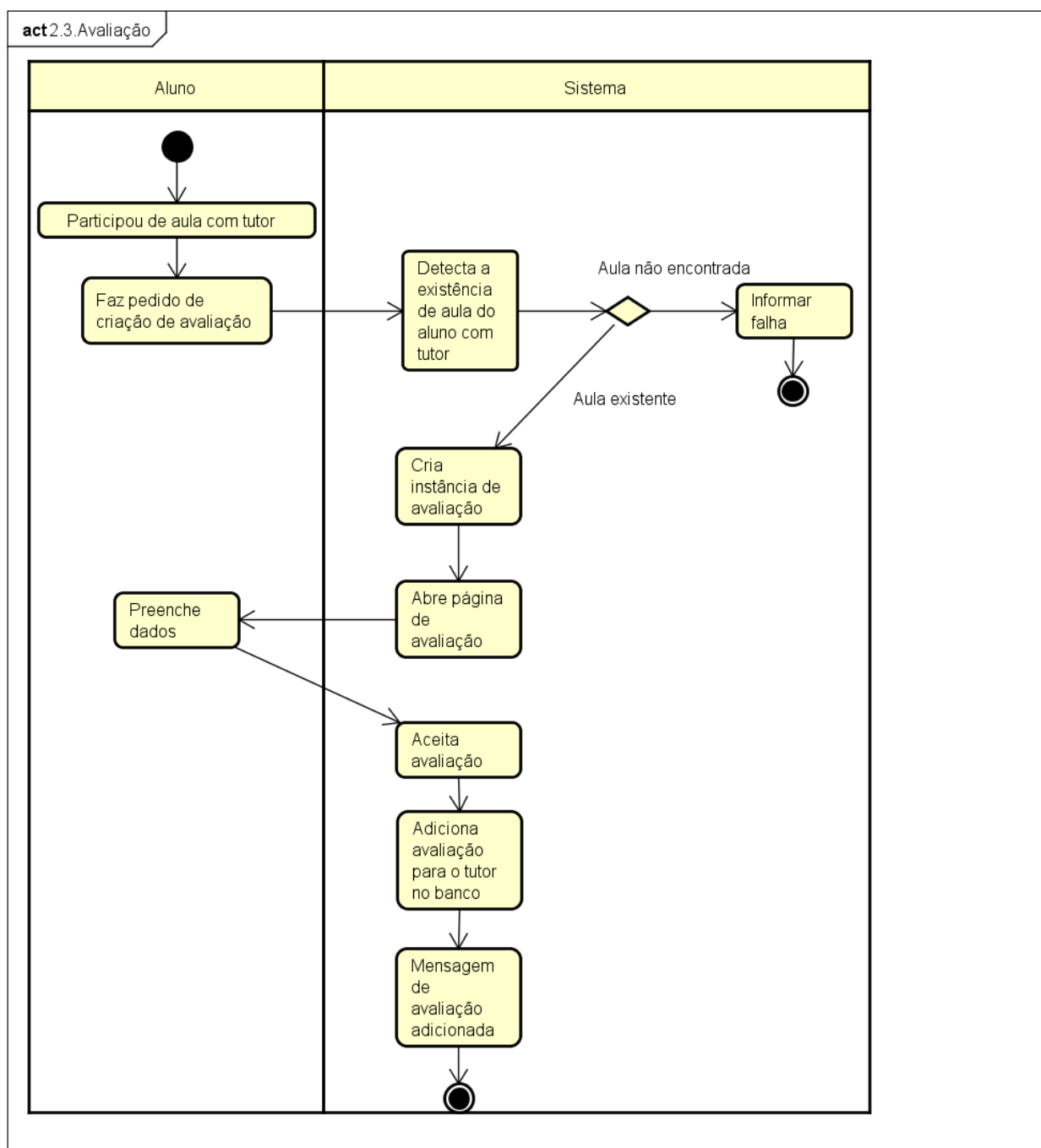


Figura 15 - Diagrama de atividade de avaliação

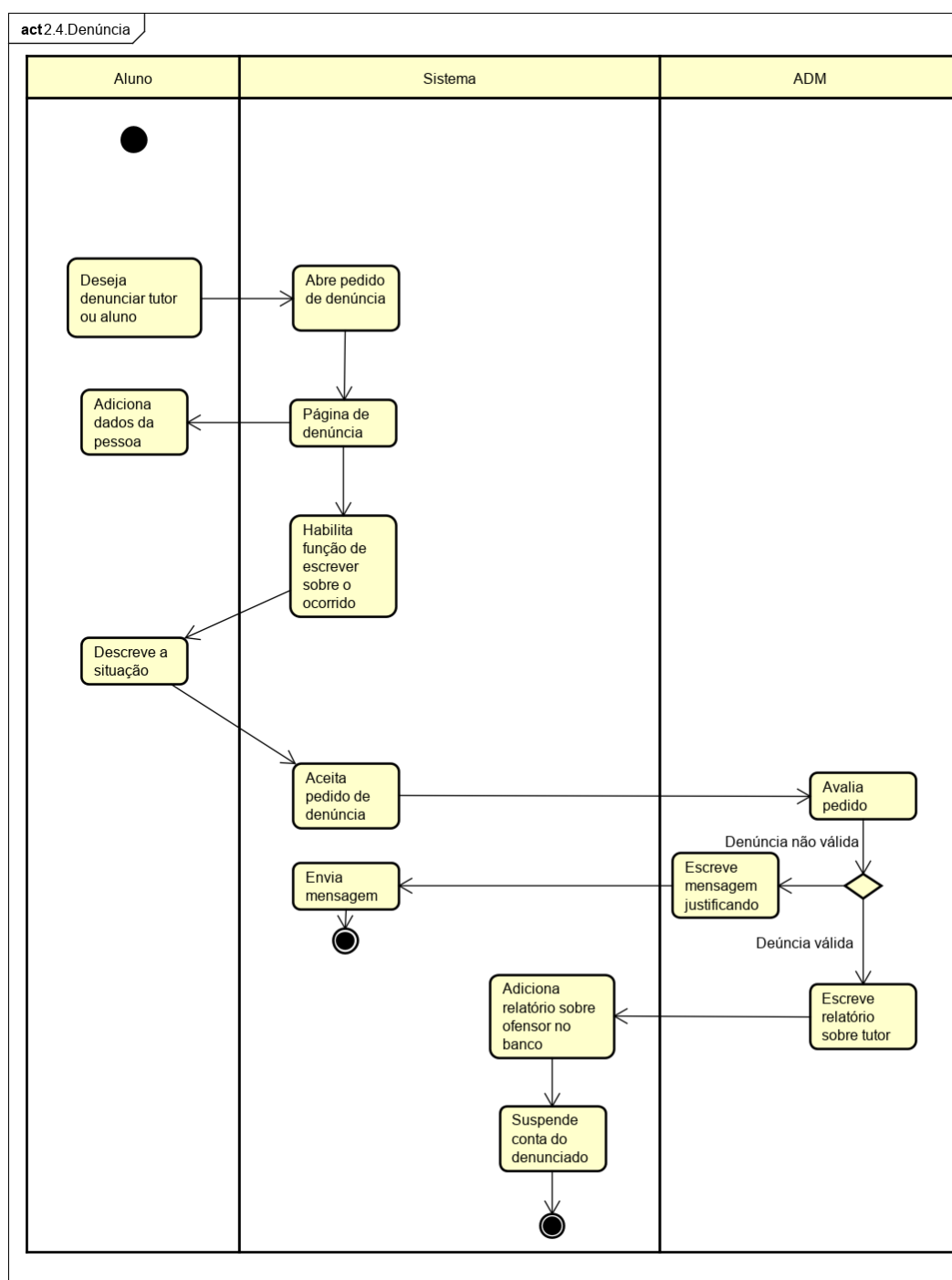


Figura 16 - Diagrama de atividade de denúncia

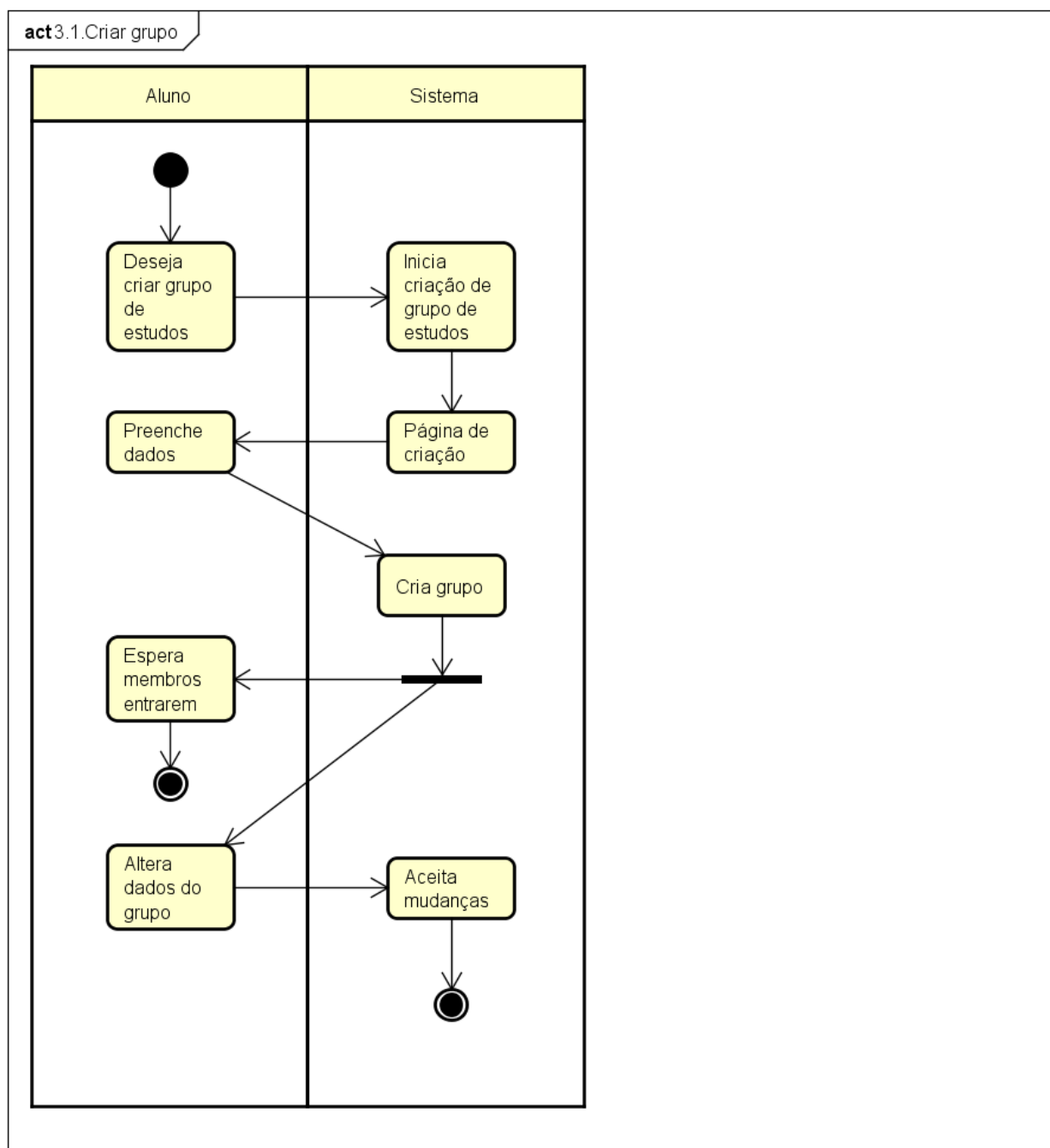
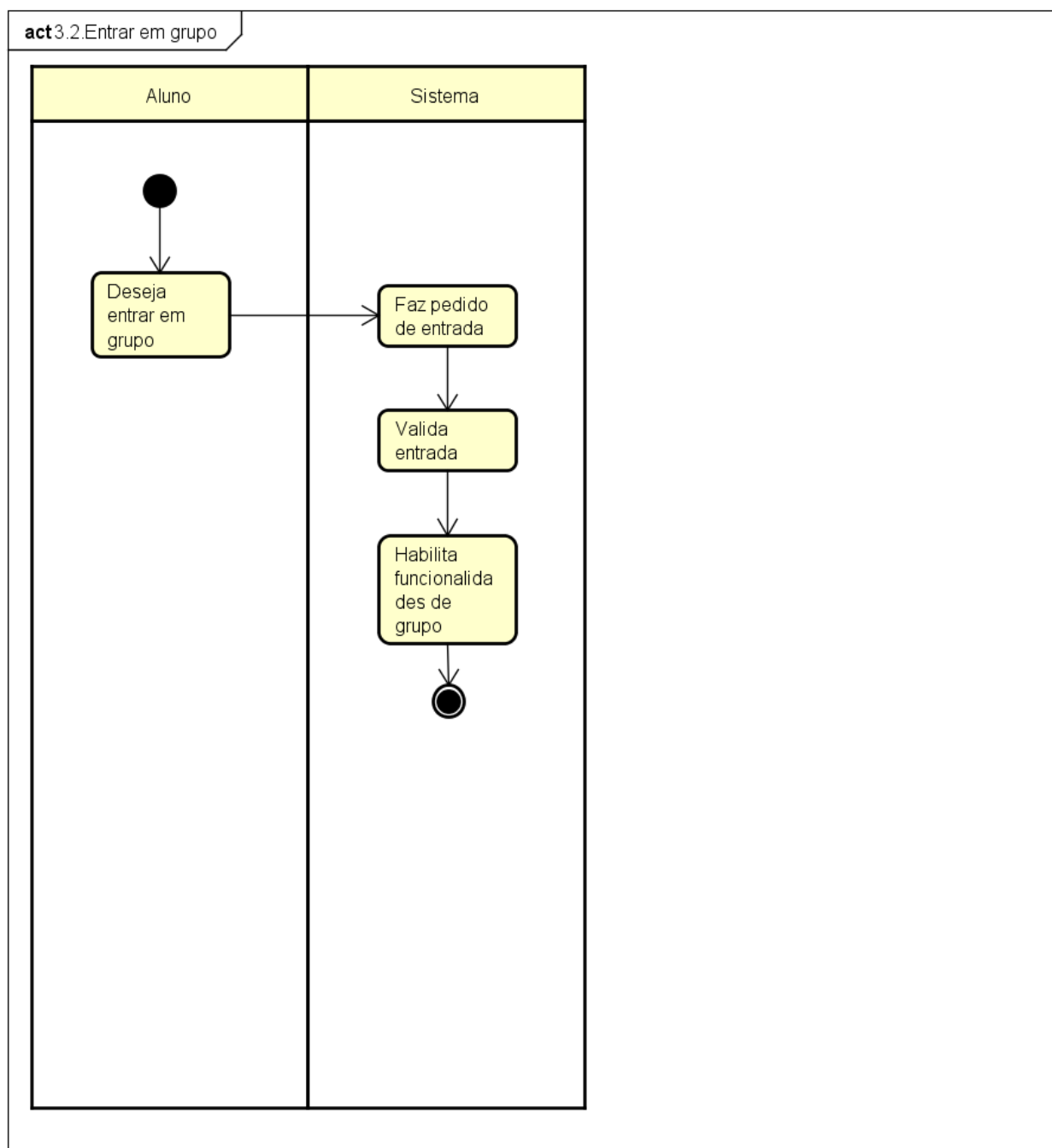
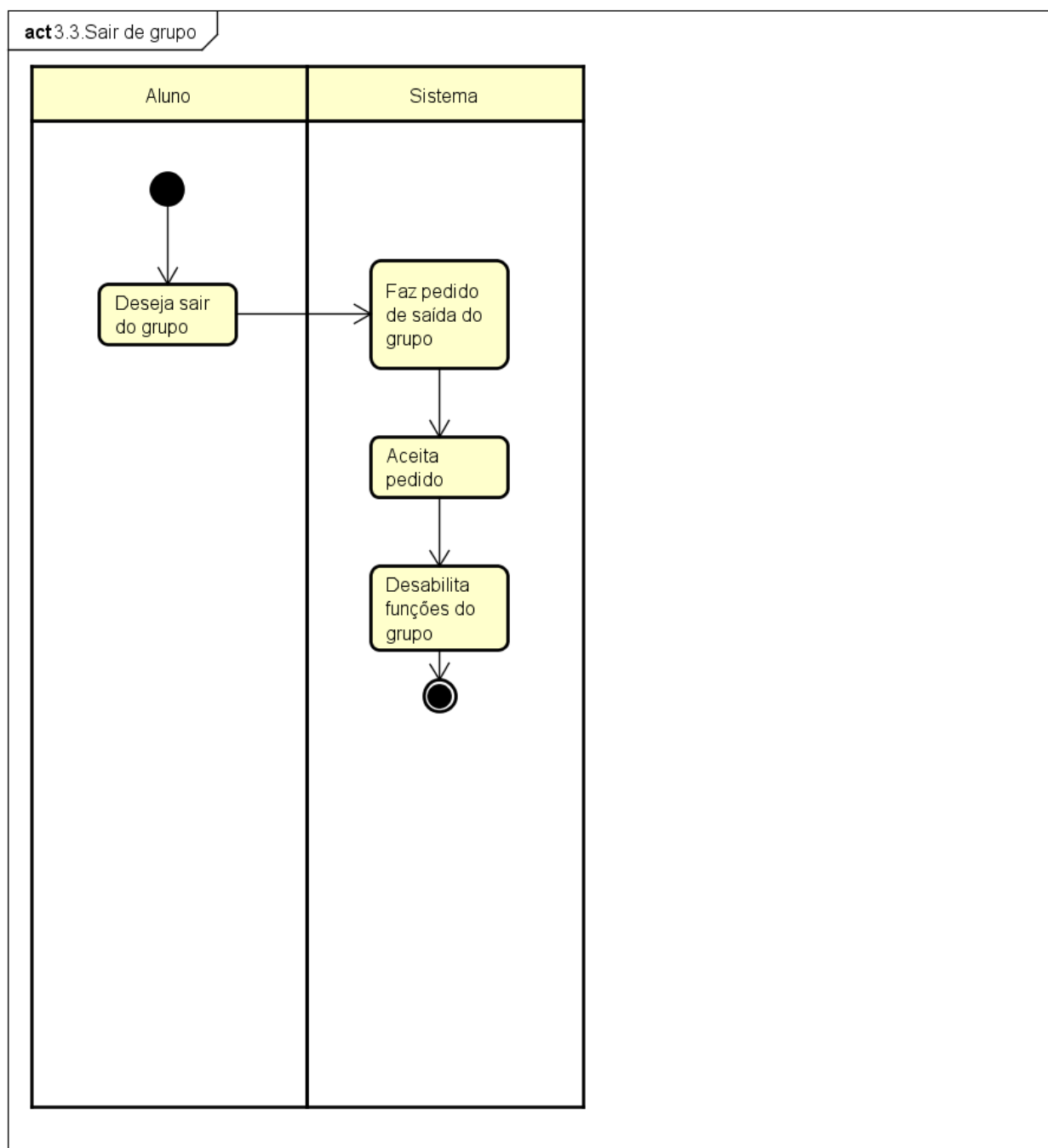
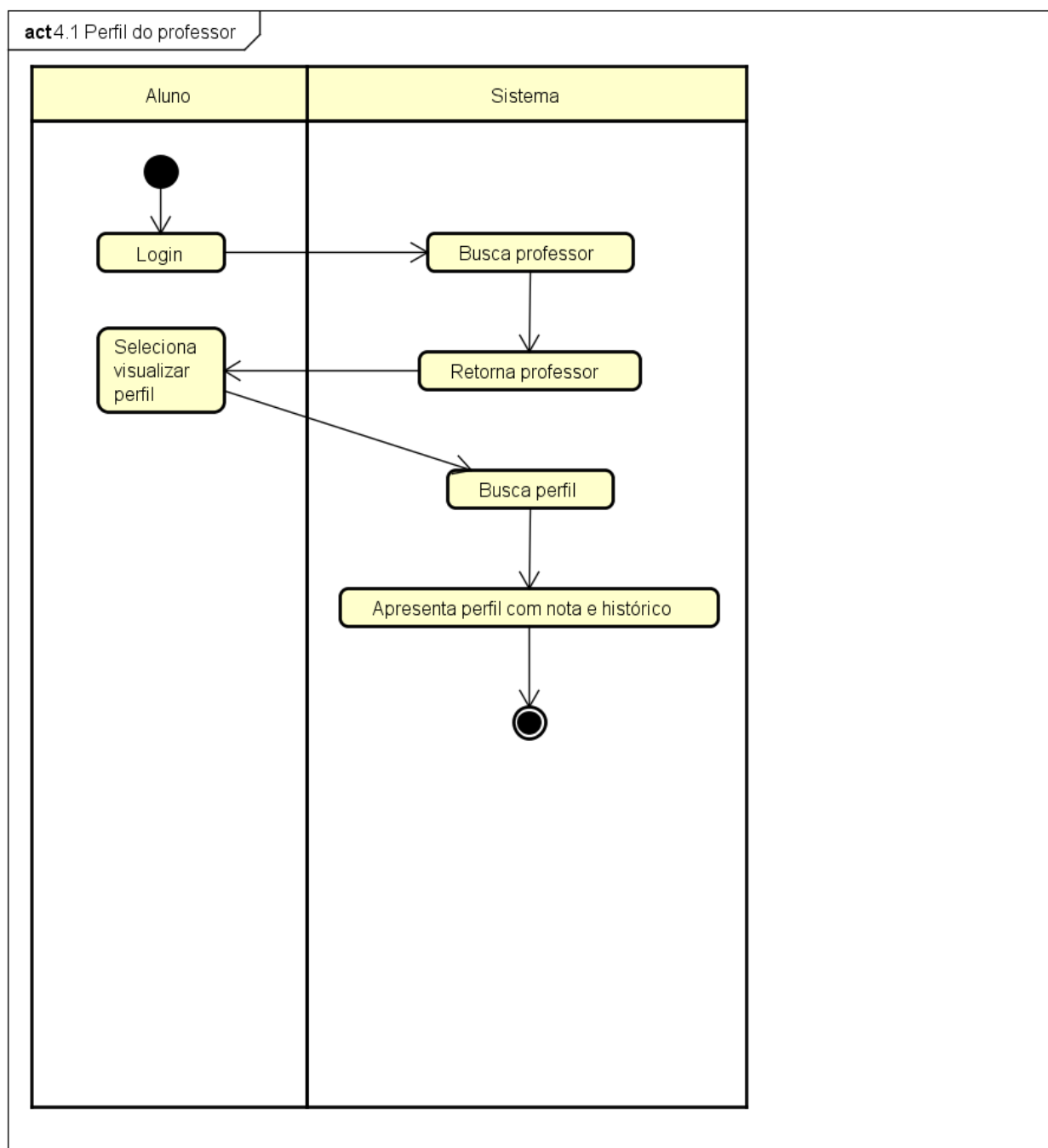
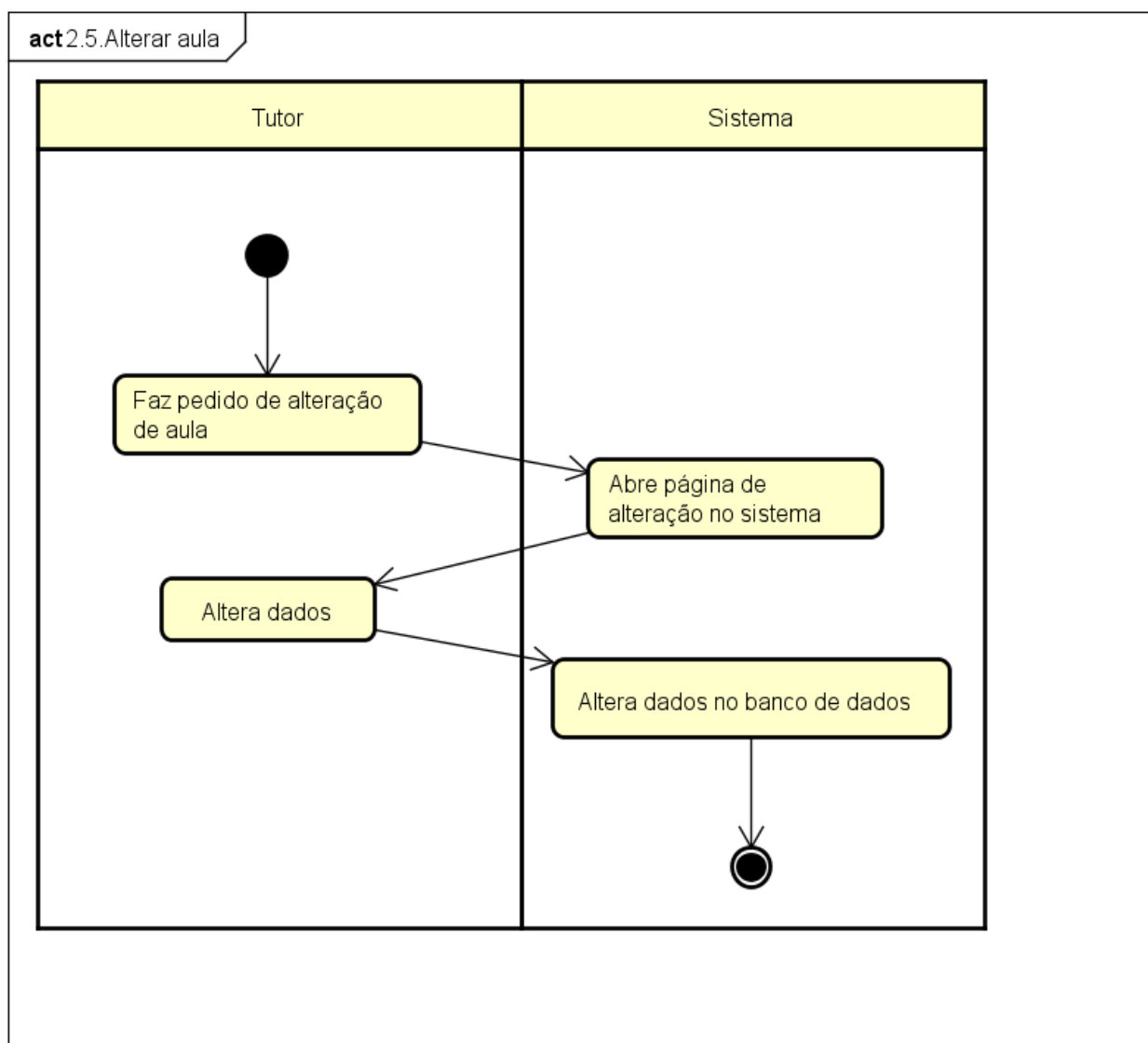


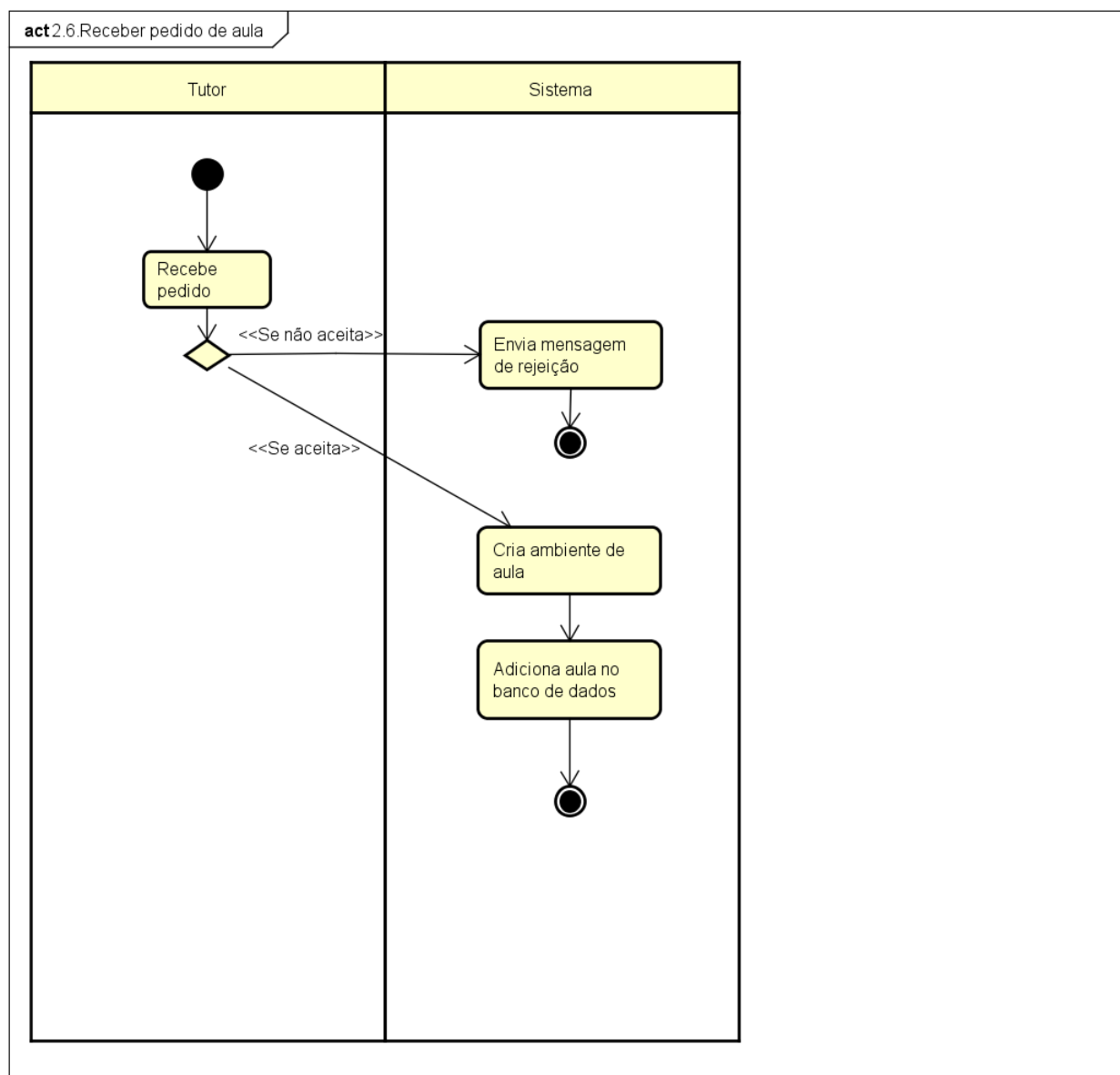
Figura 17 - Diagrama de atividade de criar grupo

**Figura 18 - Diagrama de atividade de entrar em grupo**

**Figura 19 - Diagrama de atividade de sair de grupo**

**Figura 20 - Diagrama de atividade de perfil do professor**

**Figura 20 - Diagrama de atividade de alteração de aula**

**Figura 20 - Diagrama de atividade de receber pedido de aula**

6 Implementação

6.1 Protótipos de Telas



Figura 21 - Protótipo de baixa fidelidade da tela inicial

Menu icon | Search bar | Conta | Sair

CRIAR AULA

Matéria ▾

Dia da semana +

Sala

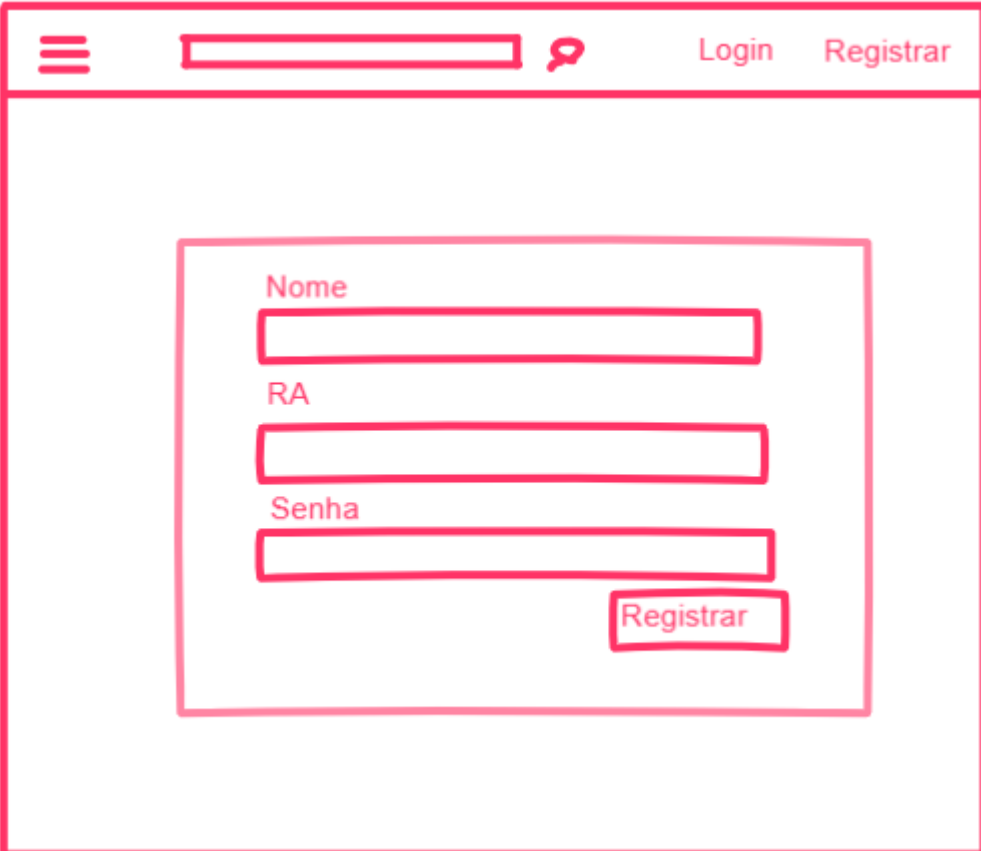
Horário

Descrição

Repetível ☐

Aluno Tutor | Sobre | Nos contate

Figura 22 - Protótipo de baixa fidelidade da tela criar aula



A low-fidelity prototype of a registration screen. The screen is enclosed in a black border. At the top, there is a header bar containing a hamburger menu icon, a search bar, a magnifying glass icon, and the text "Login" and "Registrar". Below the header, the main content area is light gray. In the center, there is a white rounded rectangle containing the registration form. The form has three input fields labeled "Nome", "RA", and "Senha". Below the "Senha" field is a "Registrar" button.

Menu Search Login Registrar

Nome

RA

Senha

Registrar

Figura 23 - Protótipo de baixa fidelidade da tela de cadastro

Menu icon | Search bar | Speech bubble icon | Conta | Sair

Resultados

Avançado

Física
Vamos aprender sobre fisica 2
Sala X

Segundas 18h
Terças 10h
Lucas

Grupo Supimpa
Cálculo
Encontros na terça

Aluno Tutor | Sobre | Nos contate

Figura 23 - Protótipo de baixa fidelidade da tela de pesquisa

6.2 Descrição do código

6.2.1 Organização dos Arquivos

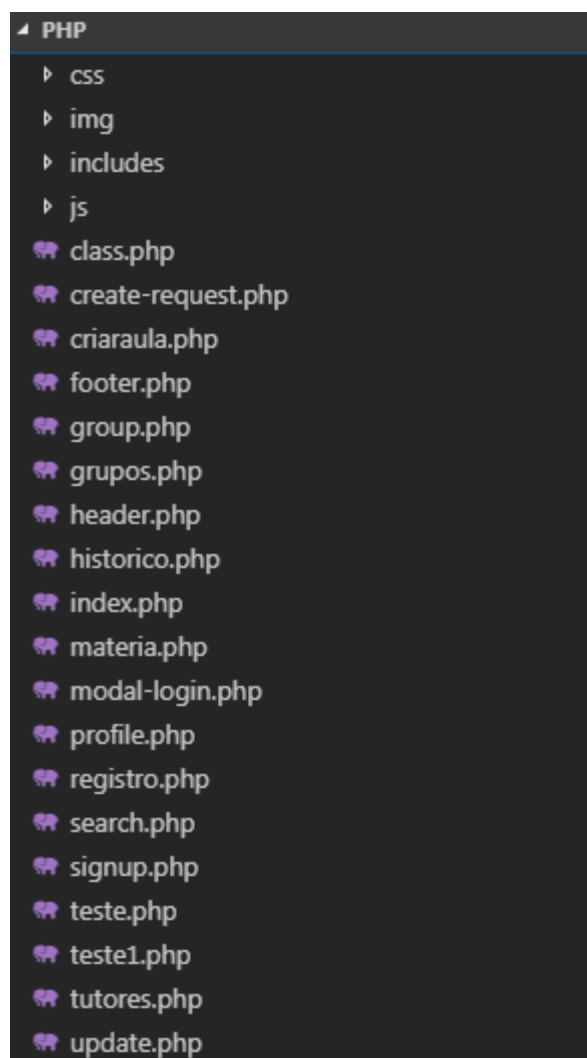


Figura 24 - Estrutura raiz

Pasta raiz onde estão todos os arquivos que formam o projeto. A distribuição é feita em pastas que contém arquivos de código em PHP, css e Javascript. Os principais arquivos estão localizados na raiz.

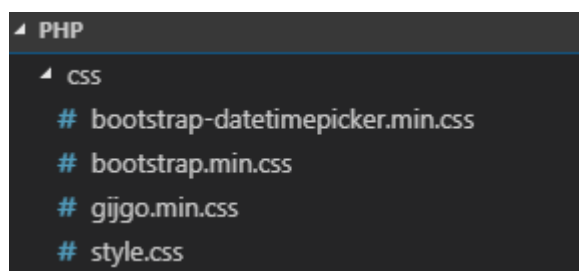


Figura 25 - Pasta do CSS

Nessa pasta ficam os arquivos de bootstrap, que são relacionados aos métodos que foram usados para desenvolver o front-end.

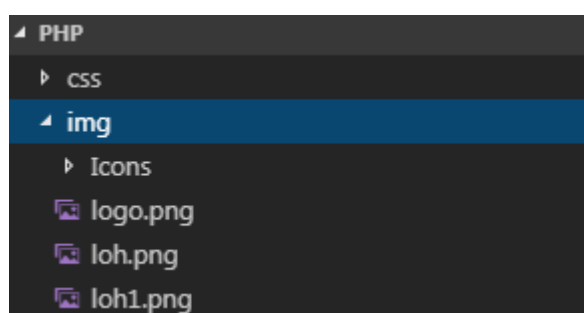


Figura 26 - Pasta das imagens

Nessa pasta ficam as imagens que foram usadas para preencher a interface do projeto.

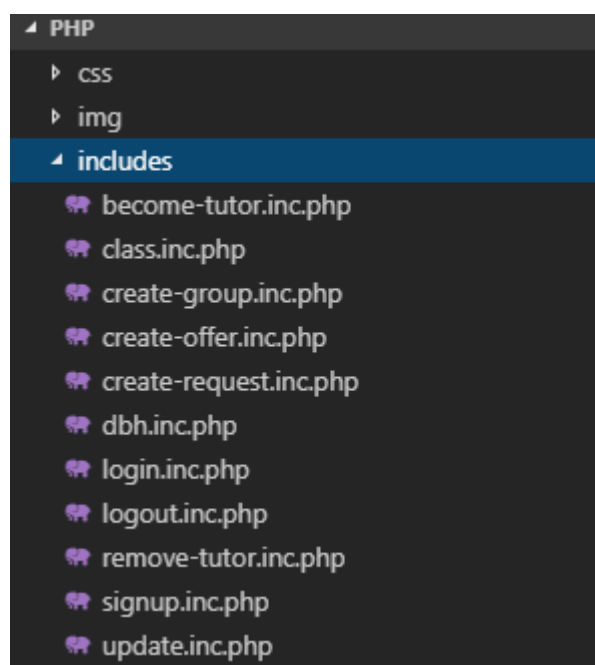


Figura 26 - Pasta de includes

Nessa pasta ficam os arquivos de include do back-end, que são utilizados para quando se quer incluir o mesmo código em várias páginas.

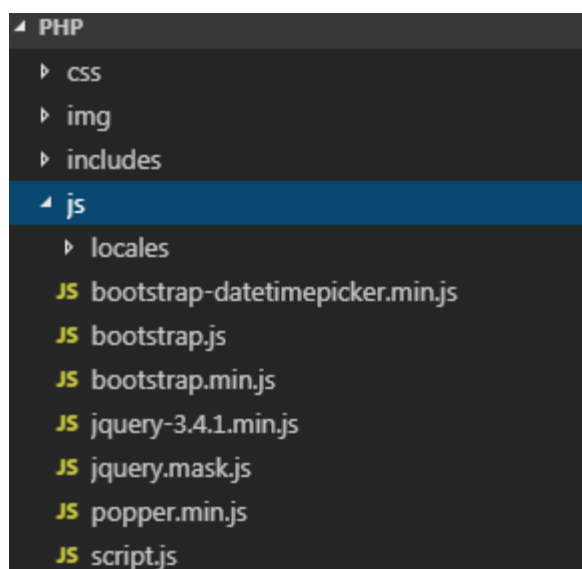


Figura 26 - Pasta dos arquivos Javascript

Nessa pasta ficam os arquivos de Javascript, que contém componentes no formato de plugins jQuery, usados para providenciar elementos de interface do usuário.

6.3 Testes

Os seguintes testes foram feitos usando a semântica Behavior Driven Development (BDD), uma metodologia ágil que busca simplificar a maneira de testar e esclarecer melhor aquela determinada função que está sendo validada.

Teste 1:

Critério de Aceite: O sistema deve permitir a procura de um tutor, uma vez que ele esteja cadastrado no sistema.

Contexto: Algum tutor ou aluno deseja procurar por um determinado tutor.

Cenário: Acesso ao site, entrada dos dados que serão procurados na barra de pesquisa.

Dado: que sou um aluno.

E: tenho cadastro e acesso ao site.

Quando: eu solicitar uma busca por um tutor.

E: o sistema possuir informações sobre esse tutor guardadas no banco de dados.

Então: devo receber o resultado daquilo que estava esperando.

E: deverá aparecer uma lista com o nome do tutor que foi procurado.

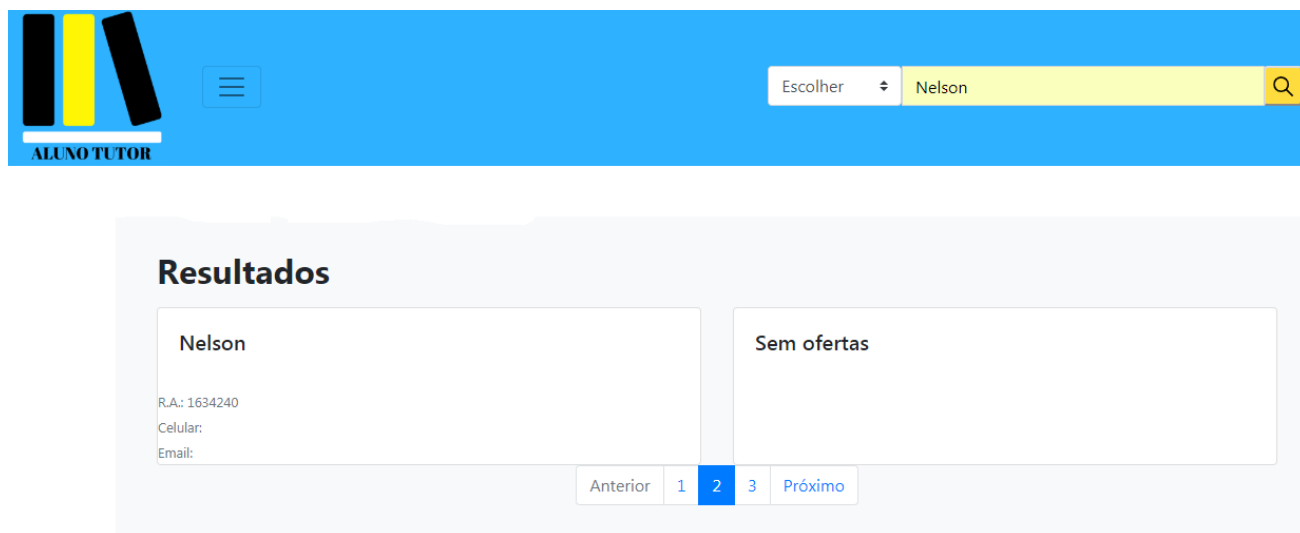


Figura 27 - Imagem do teste da pesquisa realizada

Teste 2:

Critério de Aceite: O sistema deve permitir o cadastro de um usuário, para que ele possa utilizar funcionalidades específicas.

Contexto: Algum usuário deseja ter o seu cadastro realizado.

Cenário: Acesso ao site, entrada dos dados que serão cadastrados ao clicar no botão superior direito login e ir na opção cadastrar.

Dado: que sou um usuário.

E: tenho acesso ao site.

Quando: eu solicitar um cadastro.

E: o sistema possuir as implementações necessárias para tal.

Então: devo conseguir armazenar meus dados no sistema.

E: deverá aparecer uma mensagem de confirmação de cadastro.

A imagem mostra a interface de usuário para o cadastro de um novo usuário no sistema "ALUNO TUTOR". No topo, há uma barra azul com o logo do sistema à esquerda, um ícone de menu hambúrguer no centro e uma barra de busca à direita com o texto "Escolher" e "O que procura?". Abaixo, o formulário de cadastro é intitulado "Cadastrar" e contém os seguintes campos:

- Nome:** Campo de texto com o valor "Nelson".
- Curso:** Campo de seleção com o valor "Engenharia de Software".
- RA:** Campo de texto com o valor "1634240".
- Senha:** Campo de texto com caracteres ocultos por pontos.
- Confirmar senha:** Campo de texto com caracteres ocultos por pontos.

Na base do formulário, há um botão azul rotulado "Criar conta".

Figura 28 - Imagem do teste na tela de cadastro

Teste 3:

Critério de Aceite: O sistema deve permitir a modificação do cadastro de um usuário, para que ele possa mudar as informações desejadas.

Contexto: Algum usuário deseja ter o seu cadastro mudado.

Cenário: Acesso ao site, entrada dos dados que serão cadastrados ao clicar no botão superior direito login, ir na opção conta e depois alterar dados.

Dado: que sou um usuário.

E: tenho acesso ao site.

Quando: eu solicitar uma mudança de informações do cadastro.

E: o sistema possuir as implementações necessárias para tal.

Então: devo conseguir armazenar meus novos dados no sistema.

E: deverá aparecer uma mensagem de confirmação de mudança.

A imagem mostra a interface de usuário de um sistema web. No topo, há uma barra azul com o logo 'ALUNO TUTOR' à esquerda, um ícone de menu hambúrguer no centro e uma barra de busca à direita com o texto 'Escolher' e 'O que procura?'. Abaixo, o formulário 'Editar Dados do Aluno' contém campos para Nome (Nelson), Curso (Engenharia de Software), RA (1634240), Senha (mascarada com pontos), Email (netoneze@hotmail.com) e Celular (43999849996). Um botão azul 'Alterar dados' está na base do formulário.

Figura 29 - Imagem do teste na tela de edição

7 Considerações Finais

Tendo em vista os resultados obtidos, tem-se que foi possível utilizar as orientações dadas durante o decorrer da disciplina, visto que o desfecho geral foi satisfatório. Com as funcionalidades que foram propostas em execução, é possível gerar um ambiente onde as reuniões entre estudantes que desejam compartilhar informações entre si possam acontecer.

Os objetivos foram parcialmente concluídos, sendo que tivemos dificuldades em estabelecer uma interface que fosse suficientemente boa para o propósito do trabalho, visto que a necessidade era de que a interface fosse intuitiva e de fácil utilização. Apesar dos obstáculos, foi possível atingir boa parte das metas estabelecidas no início do projeto.

Existe a possibilidade da utilização do processo no futuro, tanto em futuras disciplinas do curso quanto em execução no âmbito social. É possível ter essa perspectiva devido à natureza do sistema, que é flexível e pode ser modificado e aplicado em vários outros contextos de software. Vale a pena ressaltar que o projeto ainda possui possibilidade de maiores projeções com a aplicação de itens de melhoria, como por exemplo a API do sistema de registro geral dos alunos da universidade.

8 Bibliografia

Microsoft. **Documentação do IDE do Visual Studio**, 2019. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/ide/?view=vs-2019>>. Acesso em: 25 de outubro de 2019.

Atom. **Atom Documentation**, 2019. Disponível em: <<https://atom.io/docs>>. Acesso em: 21 de outubro de 2019.

Bootstrap. **Introduction**, 2019. Disponível em: <<https://getbootstrap.com/docs/4.3/getting-started/introduction/>>. Acesso em: 21 de outubro de 2019.

Astah. **Astah User Guide**, 2019. Disponível em <<http://astah.net/manual>>. Acesso em: 21 de outubro de 2019.

PHP.net. **Sua primeira página PHP**, 2019. Disponível em <https://www.php.net/manual/pt_BR/tutorial.firstpage.php>. Acesso em: 25 de outubro de 2019.

Williams, Sara. **How to Run a PHP File in XAMPP**, 2019. Disponível em <<https://www.techwalla.com/articles/how-to-run-a-php-file-in-xampp>>. Acesso em: 25 de outubro de 2019.

Apache Friends. **XAMPP Documentation**, 2019. Disponível em <<https://www.apachefriends.org/docs/>>. Acesso em: 30 de outubro de 2019.

Chahal, Varnoorjot. **XAMPP phpMyAdmin: How to Create Database and Tables**, 2019. Disponível em <<https://blog.templatetoaster.com/xampp-phpmyadmin/>>. Acesso em: 30 de outubro de 2019.

Postgresql. **Documentation**, 2019. Disponível em <<https://www.postgresql.org/docs/>>. Acesso em: 21 de outubro de 2019.

DevMedia. **Desenvolvimento Orientado por Comportamento**, 2019. Disponível em <<https://www.devmedia.com.br/desenvolvimento-orientado-por-comportamento-bdd/21127>>. Acesso em: 2 de dezembro de 2019.