**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**

**Lara Rodrigues Dantas**

**Nelson Antonio Neto Toneze**

**Vanderson Oliveira**

**Lucas Malheiros**

**Cornélio Procópio**

**2019**

**Sumário**

[**1**](#_heading=h.30j0zll) **Introdução 2**

[1.1](#_heading=h.1fob9te) Contexto 2

[1.2](#_heading=h.3znysh7) Justificativa 2

[1.3](#_heading=h.2et92p0) Proposta 2

[1.4](#_heading=h.3dy6vkm) Organização do Documento 2

[**2**](#_heading=h.1t3h5sf) **Descrição Geral do Sistema 3**

[2.1](#_heading=h.4d34og8) Objetivos (Gerais e Específicos) 3

[2.2](#_heading=h.2s8eyo1) Limites e Restrições 3

[2.3](#_heading=h.17dp8vu) Descrição dos Usuários do Sistema 3

[**3**](#_heading=h.3rdcrjn) **Desenvolvimento do Projeto 4**

[3.1](#_heading=h.26in1rg) Tecnologias e ferramentas 4

[3.2](#_heading=h.lnxbz9) Metodologia de desenvolvimento 4

[3.3](#_heading=h.1pxezwc) Cronograma previsto 4

[**4**](#_heading=h.1ksv4uv) **Requisitos do Sistema 5**

[4.1](#_heading=h.44sinio) Requisitos Funcionais 5

[4.2](#_heading=h.2jxsxqh) Requisitos Não-funcionais 5

[4.3](#_heading=h.z337ya) Diagramas de Casos de Uso 5

[**5**](#_heading=h.3j2qqm3) **Análise do Sistema 6**

[**6**](#_heading=h.1y810tw) **Implementação 7**

[6.1](#_heading=h.4i7ojhp) Protótipos de Telas 7

[6.2](#_heading=h.2xcytpi) Descrição do código 7

[**7**](#_heading=h.3whwml4) **Considerações Finais 8**

[**8**](#_heading=h.2bn6wsx) **Bibliografia 9**

# Introdução

## 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.

## 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.

## 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>.

## 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.

Lara Rodrigues Dantas, 20 anos de idade, aluna do quarto semestre de engenharia de Software, juntamente com Nelson é responsável pela análise dos diagramas e pela organização do documento. Tem experiência adquirida em C, Java, MySQL.

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.

## 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 segundo parte abrange a descrição geral do sistema, sua documentação, um começo de tecnicalidades. 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.

# Descrição Geral do Sistema

## 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 requisite 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;

## Limites e Restrições

* + 1. Limites
       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. Apenas alunos que requisitarem adicionar o status de tutor para suas contas poderão criar aulas;
       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 mesmo fora do sistema, cabendo aos mesmos denunciarem pessoas que cometerem alguma ofensa, onde os desenvolvedores irão analisar a denúncia e tomar as medidas cabíveis;
       4. O sistema não tem acesso ao banco de dados dos discentes da universidade, portanto para autenticação de registro acadêmico, irá validar número inseridos com módulo de 9.
       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.
       6. O sistema não se responsabiliza pela inadimplência de seus usuários em compromissos marcados;
       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. Restrições
       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. Como o sistema não estará vinculado à internet, a base de dados utilizada estará armazenada fisicamente em um dispositivo local.

## Descrição dos Usuários do Sistema

**Interessados:** Público composto por estudantes que precisam de ajuda com disciplinas de seus respectivos cursos, pessoas que tem conhecimento em alguma área e gostariam de compartilhar. Em específico os discentes da Universidade Tecnológica Federal do Paraná.

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

# Desenvolvimento do Projeto

## Tecnologias e ferramentas

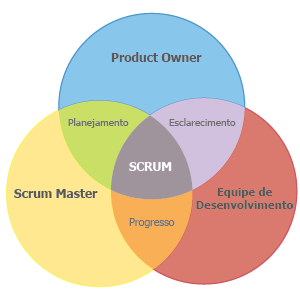
* **Organização:** GoogleDrive, WhatsApp, GitHub;
* **Linguagens:** HTML, CSS, JavaScript, Node.js;
  + Bibliotecas: Bootstrap, Express.js;
* **Prototipagem:** Moqups.com;
* **Modelagem:** Astah Community, Astah, PostgreSQL;
* **Ferramenta de desenvolvimento:** Microsoft VS Code.
* **Banco de dados:**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

## 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.

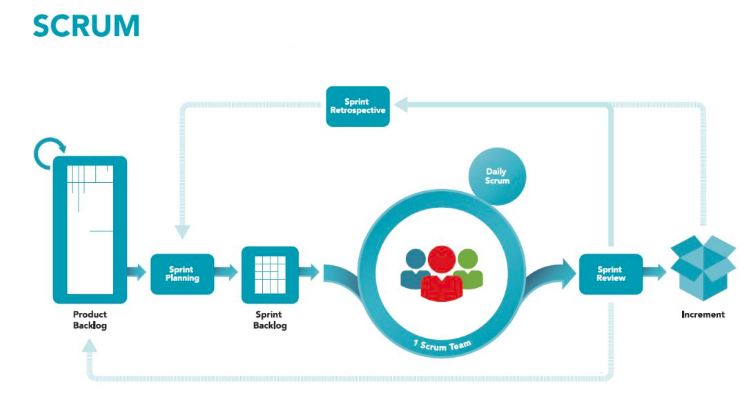
Para esse projeto o modelo SCRUM sofrerá uma mudança em relação ao original levando em considerações disponibilidade da equipe, tempo e conhecimento, alguns papéis serão incorporados por mais de um membro, o DAILY SCRUM sofrerá uma mudança em relação a frequência das reuniões.

**Figura 01 - Diagrama de VENN dos Stakeholders.**

****

* + 1. **Scrum: Papéis, Eventos e Artefatos**
       1. **PAPÉIS**
          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.
          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. **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.
       2. **ARTEFATOS**
          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
          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. **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. **EVENTO**
          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.
  + - * 1. **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.
  + - * 1. **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.
  + - * 1. **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.
  + - * 1. **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.

**Figura 02 - Papéis, Eventos e Artefatos**

****

## Cronograma previsto

|  |  |  |  |
| --- | --- | --- | --- |
| **1ª SPRINT** | **Objetivo** | Elaboração do Tópico 1  Introdução | **Data** |
| 15/08/2019 |
| **Responsável** | **Sprint Backlog** |
| Lucas | Definição do Projeto | |
| Nelson | Contexto | |
| Lara | Justificativa | |
| Vanderson | Organização do Projeto | |
| Sprint Review | | 21/08/2019 |
| Sprint Retrospective | | 22/08/2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| **2ª SPRINT** | **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 | |
| Lara | Descrições do Usuário | |
| Vanderson | Desenvolvimento do Projeto | |
| Sprint Review | | 28/08/2019 |
| Sprint Retrospective | | 29/18/2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| **3ª SPRINT** | **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 SPRINT** | **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 | |
| Lara | Análise de Caso de Uso | |
| Lucas | Diagrama de Atividades | |
| Vanderson | Análise de Atividades | |
| Sprint Review | | 18/09/2019 |
| Sprint Retrospective | | 19/09/2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| **5ª SPRINT** | **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 | |
| Lara | Documentação do Modelo de Banco de Dados, Diagrama de Classe, Diagrama de Atividades. | |
| Sprint Review | | 02/10/2019 |
| Sprint Retrospective | | 03/10/2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| **6ª SPRINT** | **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** |
| Lara | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **7ª SPRINT** | **Objetivo** | Documentação Tópico 6  Implementação do Sistema III  Desenvolvimento Frontend  Desenvolvimento Backend  Implementação da Tela | **Data** |
| 17/10/2019 |
| **Responsável** | **Sprint Backlog** |
| Lara | 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 |

|  |  |  |  |
| --- | --- | --- | --- |
| **8ª SPRINT** | **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) | |
| Lara  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 SPRINT** | **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 | | 21/112019 |

|  |  |  |  |
| --- | --- | --- | --- |
| **10ª SPRINT** | **Objetivo** | Implementação do Sistema VI  Finalização Front End  Finalização Backend  Finalização da Tela  Testes | **Data** |
| 21/11/2019 |
| **Responsável** | **Sprint Backlog** |
| Lara  Nelson  Lucas  Vanderson | Testes finais  Correção bugs e revisão  Revisão dos Documentos | |
| Sprint Review | | 27/11/2019 |
| Sprint Retrospective | | 28/11/2019 |

# Requisitos do Sistema

## Requisitos Funcionais

Apresentar os requisitos funcionais, que especificam ações que o sistema deve ser capaz de executar, ou seja, as funções do sistema. Classifique as funcionalidades quanto a prioridade:

Essencial – deve ser implementado para que o sistema funcione.

Importante – sem este requisito o sistema pode funcionar, mas não da maneira esperada.

Desejável – este tipo de requisito não compromete o funcionamento do sistema.

|  |  |  |
| --- | --- | --- |
| **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 |
| RF21 | O sistema deve possibilitar a geração de um histórico de tutelas para o tutor | Importante |

Criar aqui subitens do capítulo para descrever textualmente, com mais detalhes, as funcionalidades previstas.

## Requisitos Não-funcionais

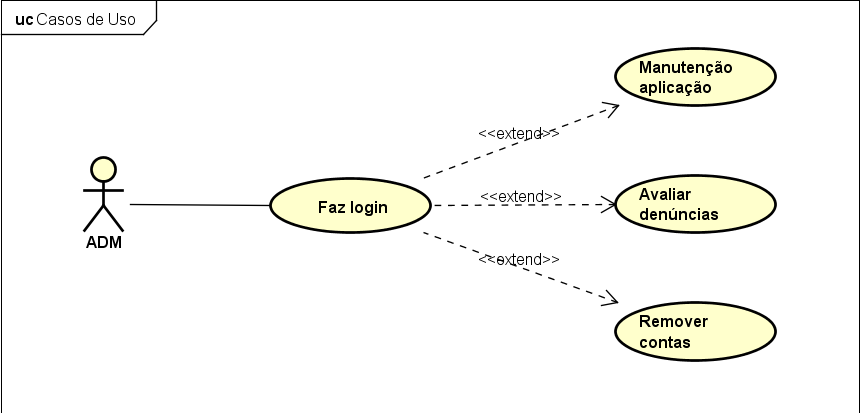
Descrever os requisitos não-funcionais do sistema, que especificam restrições sobre os serviços ou funções providas pelo sistema, categorizando de acordo com a característica envolvida, como: Usabilidade, Padronização, Ambiente, Compatibilidade, Recursos, etc.

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Categoria** |
| RNF01 | O sistema permite apenas cadastro de usuário através da utilização do Registro Acadêmico do usuário. | Padronização |
| RNF02 | O sistema permite apenas números e letras como senha | Padronização |
| RNF03 | O sistema deve ser acessado via aplicação web | Ambiente |
| RNF04 | O sistema deve permitir que os usuários se comuniquem apenas por texto | Padronização |
| RNF05 | O aluno deve cancelar o pedido de aula com no máximo vinte e quatro horas de antecedência | Padronização |
| RNF06 | O sistema de busca deve possuir filtros por matéria, data e avaliação de tutor | Usabilidade |
| RNF07 | O grupo de estudo deve ter um limite máximo de dez participantes | Padronização |
| RNF08 | O histórico de tutelas do aluno deve possuir data da ocasião, local, R.A do tutor que participou e nome do tutor que participou | Padronização |
| RNF09 | O histórico de tutelas do tutor deve possuir data da ocasião, local, R.A do aluno que participou e nome do aluno que participou | Padronização |
| RNF10 | O sistema deve ser desenvolvido utilizando HTML, CSS, JavaScript e Node.js | Implementação |
| RNF11 | Todo registro acadêmico de usuários que estiverem cadastrados no sistema deve ser validado com base no dígito verificador. | Segurança |
| RNF12 | O sistema deve ser implementado na data estipulada pelo professor da disciplina | Entrega |
| RNF13 | O sistema deve permitir que o administrador faça alterações na aplicação | Implementação |

## Diagramas de Casos de Uso

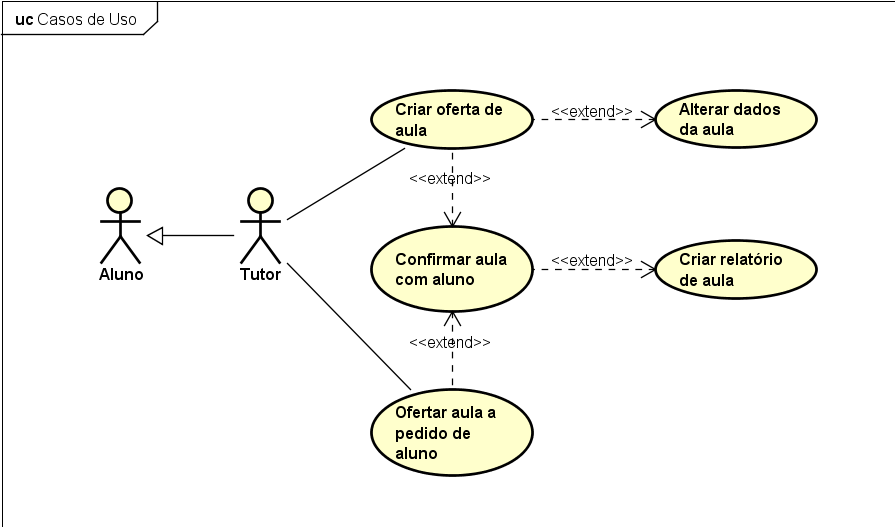
**UC1 - Gerenciamento do Administrador**

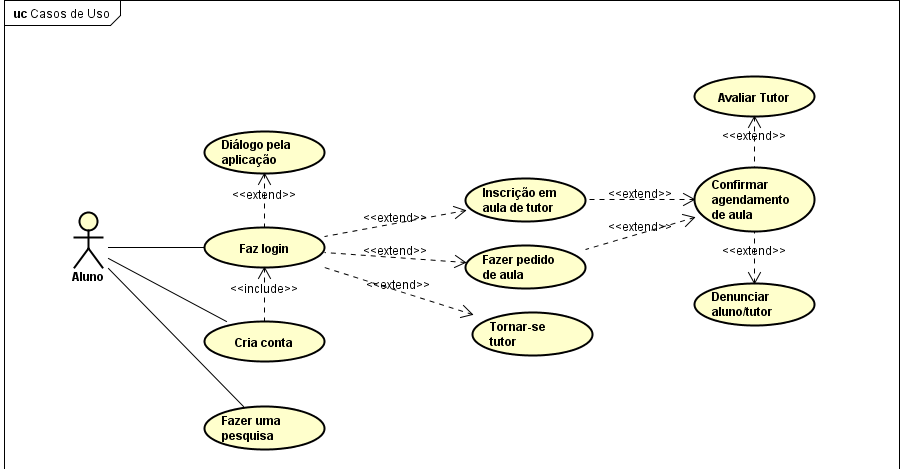
Requisitos atendidos: RF01, RF19.



**ESPECIFICAÇÕES DOS CASOS DE USO- Manutenção aplicação**

|  |  |
| --- | --- |
| **Descrição** | **Realizar a manutenção do sistema** |
| **Autor Principal** | Administrador. |
| **Autor Secundário** | Sistema. |
| **Pré- Condições** | O sistema está ligado no Banco de dados; O administrador estar logado em conta com privilégios para modificações. |
| **Pós- Condições** |  |
| **Ações Normais** | 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 |
| RNF06 | O sistema de busca deve possuir filtros por matéria, data e avaliação de tutor |
| RNF07 | O grupo de estudo deve ter um limite máximo de dez participantes |
| RNF08 | O histórico de tutelas do aluno deve possuir data da ocasião, local, R.A do tutor que participou e nome do tutor que participou |
| RNF09 | O histórico de tutelas do tutor deve possuir data da ocasião, local, R.A do aluno que participou e nome do aluno que participou |
| RNF10 | O sistema deve ser desenvolvido utilizando HTML, CSS, JavaScript e Node.js |
| RNF11 | Todo registro acadêmico de usuários que estiverem cadastrados no sistema deve ser validado com base no dígito verificador. |
| RNF12 | O sistema deve ser implementado na data estipulada pelo professor da disciplina |
| RNF13 | O sistema deve permitir que o administrador faça alterações na aplicação |





# Análise do Sistema

Este item deve apresentar a documentação da análise do sistema conforme o processo ou ciclo de vida descrito no capítulo 3. Organize o capítulo para apresentar os artefatos previstos e o que mais for necessário (protótipos, implementação, versões, telas, etc.), **incluindo no mínimo**:

**- Modelo do Banco de Dados**

Modelo Conceitual: Apresentar o Diagrama Entidade-Relacionamento desenvolvido para o banco de dados do sistema.

Modelo Lógico: Apresentar o esquema relacional (gráfico ou textual) do banco de dados normalizado e apresentando as tabelas com os atributos e restrições (chaves).

Dicionário de dados: Apresentar o dicionário de dados do banco de dados. Documentar cada tabela com seus atributos mostrando nome do atributo, tipo, tamanho, descrição, se é obrigatório ou não, e o que mais for necessário para descrever os dados. Documentar também usuários, stored procedures, funções e qualquer outra implementação ligada ao banco de dados.

**- Diagrama de Classes**

Apresentar o diagrama de classes previsto conforme a fase do projeto.

**- Diagrama de Atividades**

Apresentar o diagrama de atividades, que representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema. Nem todas as tarefas do sistema necessitam de um detalhamento, portanto deve-se considerar no que o diagrama irá auxiliar na implementação do sistema para decidir quais atividades devem ser descritas.

# Implementação

## Protótipos de Telas

Apresentar o protótipo do sistema, que consiste na interface preliminar contendo um conjunto de funcionalidades e telas. O protótipo é um recurso que deve ser adotado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o usuário (usabilidade).

As telas do sistema podem ser criadas na própria linguagem de desenvolvimento ou em qualquer outra ferramenta de desenho. Cada tela deve possuir uma descrição do seu funcionamento, constando pelo menos o objetivo da tela e dinâmica de navegação (de onde é chamada e que outras telas podem chamar). A descrição das telas deve registrar informações que possam ser consultadas para facilitar a implementação e a execução de testes, assim como a que requisitos funcionais se referem.

Se os protótipos de tela foram usados nas fases anteriores, esse item deve ser preenchido conforme os protótipos forem desenvolvidos e concluído com a versão final dos protótipos.

## Descrição do código

Descrever o sistema quanto ao código gerado. Explicar a organização dos arquivos, pacotes, classes ou quaisquer estruturas utilizadas no desenvolvimento do projeto, listando os componentes criados e sua estrutura. Use diagramas (Diagrama de Componentes, Diagrama de Pacotes) para ilustrar a implementação.

Descrever também convenções e padronizações para comentários no código, nomenclatura de classes, objetos, funções, etc. Se necessário, use exemplos.

## Testes

Descrever os procedimentos e métricas empregados para a avaliação e validação do software desenvolvido.

# Considerações Finais

Apresentar e discutir os resultados obtidos e sua aplicabilidade. Abordar o que foi atingido e o que não foi, as limitações, possíveis integrações com outros projetos e continuação do sistema em trabalhos futuros.

# Bibliografia

Apresentar todas as obras (livros, artigos, Internet, revistas, etc...) utilizadas na elaboração da documentação e na implementação do projeto.