ESCOLA SENAI

“PROF. DR. EURYCLIDES DE JESUS ZERBINI”

Nícolas Corrêa de Souza

Samuel Moreira da Silva

Gustavo dos Santos Guimarães

**StudyLink**

Campinas, SP

2024

Nícolas Corrêa de Souza

Samuel Moreira da Silva

Gustavo dos Santos Guimarães

**StudyLink**

Projeto apresentado à Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini” para obtenção do certificado de conclusão do Curso Técnico de Desenvolvimento de Sistemas.

Orientador: Renato Mattos Onofre, Paulo Henrique Pansani

Campinas, SP

2024

Nícolas Corrêa de Souza

Samuel Moreira da Silva

Gustavo dos Santos Guimarães

**StudyLink**

Trabalho de conclusão de curso aprovado como requisito parcial para obtenção do grau de técnico, do curso Técnico de Desenvolvimento de Sistemas da Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini”.

**BANCA EXAMINADORA**

1º Examinador

2º Examinador

3º Examinador

Campinas 2024

**DEDICATÓRIA**

Dedicamos este trabalho aos nossos familiares, professores e amigos, cuja paciência, brincadeiras, incentivo e o apoio incondicional foram essenciais para essa conquista.

**AGRADECIMENTOS**

Agradecemos imensamente ao SENAI Prof. Dr. Euryclides de Jesus Zerbini por nos proporcionar um ambiente de aprendizado de excelência, com todos os recursos necessários para o nosso desenvolvimento profissional e pessoal.

Gostaríamos também de expressar nossa gratidão aos professores Paulo Henrique Pansani, Renato de Mattos Onofre, Douglas de Cássio Quinzani Gaspar, Oscar Junior, Allan Crasso, Astarote Borrego e Paulo da Cruz Cardim por compartilharem seus conhecimentos conosco, sempre nos motivando a sermos melhores.

Por fim, mas não menos importante, agradecemos aos nossos familiares, que nos apoiaram incondicionalmente em todos os momentos. Sem vocês, nada disso seria possível.

**EPÍGRAFE**

**“Se a educação sozinha não transforma a sociedade,**

**sem ela tampouco a sociedade muda”.****Paulo Freire**

**SUMÁRIO**

[1 INTRODUÇÃO 9](#_Toc184389953)

[2 JUSTIFICATIVA 10](#_Toc184389954)

[3 OBJETIVOS 11](#_Toc184389955)

[3.1. Objetivos Gerais 11](#_Toc184389956)

[3.2. Objetivos Específicos 11](#_Toc184389957)

[4 PRODUCT BACKLOG 12](#_Toc184389958)

[5 REQUISITOS NÃO FUNCIONAIS 13](#_Toc184389959)

[6 PREMISSAS 13](#_Toc184389960)

[7 RESTRIÇÕES 13](#_Toc184389961)

[8 ANÁLISE DE RISCOS DE UM PROJETO 14](#_Toc184389962)

[8.1. Analise de risco em ordem decrescente 14](#_Toc184389963)

[8.2. Nível e Planos de Ação para os Riscos 14](#_Toc184389964)

[8.3. Planos de ação 15](#_Toc184389965)

[9 SPRINTS 16](#_Toc184389966)

[9.1. Primeiro Sprint 16](#_Toc184389967)

[9.1.1. Product Backlog 16](#_Toc184389968)

[9.1.2. Sprint Backlog 16](#_Toc184389969)

[9.1.3. Burn Down Chart 17](#_Toc184389970)

[9.1.4. Diagramas 17](#_Toc184389971)

[9.1.5. Plano de testes 17](#_Toc184389972)

[9.1.5.1. Resultados 17](#_Toc184389973)

[9.1.6. Kanban e Retrospectiva 18](#_Toc184389974)

[9.2. Segundo Sprint 22](#_Toc184389975)

[9.2.1. Product backlog 22](#_Toc184389976)

[9.2.2. Sprint Backlog 22](#_Toc184389977)

[9.2.3. Burn Down Chart 22](#_Toc184389978)

[9.2.4. Diagramas 23](#_Toc184389979)

[9.2.4.1. Diagrama de caso de uso 23](#_Toc184389980)

[9.2.4.2. Diagrama de classes 24](#_Toc184389981)

[9.2.4.3. Diagrama de narrativa de caso de uso 25](#_Toc184389982)

[9.2.5. Plano de testes 28](#_Toc184389983)

[9.2.5.1. Resultados 29](#_Toc184389984)

[9.2.6. Kanban e Retrospectiva 30](#_Toc184389985)

[9.3 Terceiro Sprint 34](#_Toc184389986)

[9.3.2 Sprint Backlog 34](#_Toc184389987)

[9.3.3 Burn Down Chart 35](#_Toc184389988)

[9.3.4 Diagramas 36](#_Toc184389989)

[9.3.4.1 Diagrama de classe 36](#_Toc184389990)

[9.3.5 Plano de teste 36](#_Toc184389991)

[9.3.6 Resultado 36](#_Toc184389992)

[9.3.7 Kanban e Retrospectiva 37](#_Toc184389993)

[9.4 Quarto Sprint 41](#_Toc184389994)

[9.4.1 Product Backlog 41](#_Toc184389995)

[9.4.2 Sprint Backlog 41](#_Toc184389996)

[9.4.3 Burn Down Chart 42](#_Toc184389997)

[9.4.4 Diagramas 43](#_Toc184389998)

[9.4.5 Plano de Teste 43](#_Toc184389999)

[9.4.6 Resultado 43](#_Toc184390000)

[9.4.7 Kanban e Retrospectiva 44](#_Toc184390001)

[10 Modelo de Dados 48](#_Toc184390002)

[10.1. Modelo conceitual do banco de dados 48](#_Toc184390003)

[10.2. Modelo lógico do banco de dados 49](#_Toc184390004)

[10.3. Dicionário de dados 50](#_Toc184390005)

[11 PRINCIPAIS TELAS DO SISTEMA 52](#_Toc184390006)

[12 CONCLUSÃO 54](#_Toc184390007)

[12.1. Escreva os resultados obtidos 54](#_Toc184390008)

[12.2. Constatações 54](#_Toc184390009)

[12.3. Sugestões de possíveis aperfeiçoamentos técnicos 54](#_Toc184390010)

[13 REFERÊNCIAS 55](#_Toc184390011)

[14 GLOSSÁRIO 56](#_Toc184390012)

# INTRODUÇÃO

A educação no Brasil tem sido marcada por desafios estruturais e um histórico de desigualdade, o que reflete diretamente nos baixos índices de qualidade e proficiência dos estudantes. O ranking do PISA (Programa Internacional de Avaliação de Estudantes), organizado pela Organização para a Cooperação e Desenvolvimento Econômico (OCDE), é uma das principais métricas internacionais para avaliar a educação. Na última edição de 2022, que avaliou alunos de 15 anos de 81 países nas áreas de matemática, leitura e ciências, o Brasil ocupou a 60ª posição. Um dado alarmante é que, segundo avaliações nacionais, apenas 5% dos alunos de escolas públicas se formam com proficiência em matemática básica. Esse cenário evidencia a necessidade urgente de melhorar o sistema educacional brasileiro.

Pensadores como Anísio Teixeira, defensor de uma educação pública, gratuita e laica, e Paulo Freire, com sua pedagogia crítica e revolucionária, desempenharam papéis fundamentais na história da educação no Brasil. Freire, em especial, ressaltava a importância da educação como um ato libertador e dialógico, defendendo a necessidade de transformar o aluno em sujeito ativo no processo de aprendizado. Outro importante nome é Darcy Ribeiro, cujas ideias sobre a universalização do acesso à educação e a criação dos Centros Integrados de Educação Pública (CIEPs) no Rio de Janeiro buscavam promover a inclusão educacional.

Apesar das contribuições de figuras históricas, o Brasil ainda enfrenta obstáculos no cumprimento dessas visões. As dificuldades do presente tornam indispensável a busca por alternativas que potencializem o acesso ao conhecimento e melhorem o desempenho dos estudantes. Nesse sentido, o uso da tecnologia é uma estratégia fundamental para atingir esses objetivos. Plataformas como o StudyLink oferecem soluções inovadoras ao democratizar o conhecimento, permitindo que os estudantes encontrem materiais de estudo atualizados e acessíveis, além de estimular uma rede colaborativa de aprendizado. Essas ferramentas podem ser vistas como uma continuação do legado deixado por grandes educadores, ao buscar, de novas formas, a transformação da realidade educacional no Brasil.

# JUSTIFICATIVA

Com o avanço acelerado da tecnologia e o acesso cada vez mais facilitado à internet, as formas tradicionais de ensino e aprendizado estão sendo desafiadas e progressivamente superadas. Nesse cenário, o modelo de educação tradicional e passivo, muitas vezes, não acompanha as demandas contemporâneas de flexibilidade, interatividade e acessibilidade. Diante dessa realidade, nós, da VisionWork, criamos o StudyLink, uma plataforma web intuitiva, acessível e voltada para transformar a experiência de estudar em algo mais dinâmico, prático e colaborativo.

A grande premissa do StudyLink é a democratização do conhecimento. Acreditamos que o saber não deve estar restrito a ambientes formais ou materiais inacessíveis. Assim, nossa plataforma permite que qualquer usuário não apenas consuma conteúdo educacional de qualidade, mas também atue como produtor e disseminador de conhecimento. Nessa perspectiva, criamos uma área dedicada à compartilhamento de materiais de estudo, onde os usuários podem criar, adicionar e organizar conteúdos, promovendo uma troca constante de saberes. Isso resolve um dos principais desafios do aprendizado autônomo: a dificuldade de encontrar materiais atualizados, de fácil compreensão e que acompanhem as tendências mais recentes do mercado e da academia.

Outro ponto forte do StudyLink é a integração do Método Feynman, uma técnica de ensino e aprendizado altamente eficaz que incentiva a prática de explicar conceitos complexos de maneira clara e acessível. Esse método, aplicado de forma colaborativa, permite que os criadores de conteúdo na plataforma não apenas contribuam para a educação de outros usuários, mas também solidifiquem seu próprio aprendizado, uma vez que ensinar é uma das formas mais poderosas de reforçar o conhecimento adquirido.

Dessa forma, o StudyLink vai além de ser apenas uma ferramenta de estudo; ele se posiciona como uma rede de aprendizado colaborativo, onde todos ganham. Quem busca conhecimento, encontra uma comunidade engajada e materiais adequados. E quem compartilha seu saber, aprimora sua compreensão e fortalece sua capacidade de comunicação. Essa dinâmica de troca sendo benéfica para ambos.

# OBJETIVOS

Segue abaixo os objetivos gerais do sistema:

# Objetivos Gerais

O StudyLink visa oferecer uma plataforma inovadora e interativa para auxiliar estudantes na organização de seus estudos. O sistema permite que os usuários criem agendas para gerenciar suas tarefas e compromissos acadêmicos, promovendo uma rotina de estudos mais estruturada e eficiente. Além disso, o StudyLink permite que os usuários criem uma área de compartilhamento de materiais de estudo, onde outros estudantes podem visualizar e baixar esses conteúdos, incentivando a colaboração e o aprendizado coletivo.

# Objetivos Específicos

* Desenvolver a documentação na norma ABNT;
* Desenvolver o backend do sistema utilizando Java com o framework Spring Boot;
* Desenvolver o front-end utilizando TypeScript com o framework Angular;
* Utilizar corretamente a metodologia ágil Scrum;
* Realizar o design no Figma;
* Criar um manual para o usuário;
* Armazenar dados no MySQL;
* Usaremos a IDE Intellij e Visual Studio Code;
* Usaremos o Github para o versionamento do projeto;
* Hospedar o sistema na web.

# PRODUCT BACKLOG

1. Usuário deve conseguir criar suas tarefas­;

RF01.1 – Poderá alterar as tarefas;

RF01.2 – Poderá criar novas tarefas;

RF01.3 – Poderá listar suas tarefas por período;

RF01.4 – Poderá deletar suas tarefas;

1. Sistema de cadastro de usuário­;

RF02.1 – Deve ser feito utilizando nome, e-mail, senha e confirmar senha;

1. Sistema de login de usuário com e-mail e senha;
2. O sistema front-end deve realizar requisições ao back-end

RF04.1 – O login (front) dever envias uma solicitação HTTP POST para o back-end e receber um token e nome do usuário

RF04.2 – O cadastro (front) deve enviar um HTTP POST para o back-end e receber uma mensagem de sucesso quando bem-sucedido;

1. Iniciar uma sessão no navegador;
2. – A sessão deve conter token de autenticação e nome do usuário;
3. Materiais de estudo;

RF07.1 – Poderá alterar seu material;

RF07.2 – Poderá adicionar material;

RF07.3 – Poderá pesquisar o material pelo título;

RF07.4 – Poderá deletar o material de estudos;

RF07.5 – Deixa-lo visível ou não, para outros usuários;

1. Usuário adicionar foto de perfil depois de logado;
2. Quando habilitado deve guardar o usuário nos cookies do navegador;
3. Dever ser capaz de realizar login utilizando o Google;
4. O sistema deve mandar um e-mail para o usuário conseguir recuperar senha.

# REQUISITOS NÃO FUNCIONAIS

**RNF01** – O sistema deve possuir uma interface intuitiva para o usuário, facilitando a navegação e o uso;

**RNF02** – O sistema deve ser uma aplicação web responsiva, funcionando corretamente em computadores e dispositivos móveis;

**RNF03** – Todas as senhas devem ser criptografadas antes de serem armazenadas no banco de dados, garantindo a segurança das informações;

**RNF04** – O front-end deve ser desenvolvido utilizando a abordagem de componentização, para proporcionar um desenvolvimento mais ágil e um código de fácil manutenção;

**RNF05** – O sistema deve exibir mensagens de erro e sucesso de forma clara para o usuário, informando o status das operações realizadas;

RNF06 – O sistema estará disponível na web.

# PREMISSAS

PRE01– O projeto será desenvolvido por 3 alunos;

PRE02– Cada aluno terá o seu computador para realizar o projeto;

PRE03– Será utilizada metodologias ágeis: Scrum/Kanban;

PRE04 – O projeto será entregue em pequenas partes (sprint);

PRE04.1 – Cada sprint terá 4 semanas.

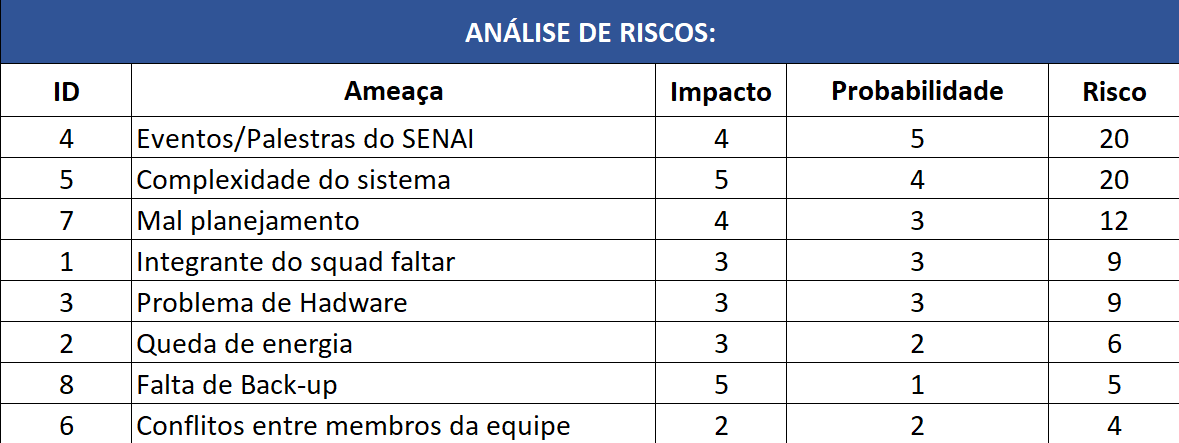
# RESTRIÇÕES

RES01 – O projeto deverá ser entregue após 4º Sprints;

RES01 – Somente softwares de uso livre poderão ser utilizados.

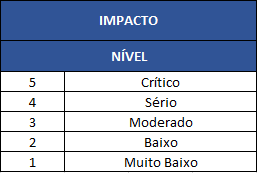
# ANÁLISE DE RISCOS DE UM PROJETO

# Analise de risco em ordem decrescente



# Nível e Planos de Ação para os Riscos

Definimos uma hierarquia do nível dos riscos, do mais grave para o menos grave. Assim, damos uma maior atenção às ameaças com maior impacto e probabilidade de acontecer.



# Planos de ação

Planos de ação para os riscos referente à tabela de riscos geraisTabela

Descrição gerada automaticamente

# SPRINTS

Apresenta-se nesta seção a documentação referente ao desenvolvimento do software.

# Primeiro Sprint

No primeiro Sprint, foram separadas as tarefas a serem feitas por cada aluno do grupo. Foi planejado fazer o design no Figma das telas da web, sendo elas: Tela login/cadastro, tela da agenda onde as tarefas irão aparecer.

Em relação ao Banco de dados de dados foi planejado fazer o modelo lógico e conceitual, em seguida criamos o banco usando o MySQL Workbench.

# Product Backlog

[Não](http://www.devmedia.com.br/curso/introducao-a-uml/128) houve alterações no backlog

# Sprint Backlog

RF01.1 – Poderá alterar as tarefas;

RF01.2 – Poderá criar novas tarefas;

RF01.3 – Poderá listar suas tarefas por período;

RF01.4 – Poderá deletar suas tarefas;

RF02 – Sistema de cadastro de usuário­;

RF03 – Sistema de login de usuário com e-mail e senha;

**RNF01** – O sistema deve possuir uma interface intuitiva para o usuário, facilitando a navegação e o uso;

# Burn Down Chart

# Diagramas

Não realizamos diagramas nesse sprint

# Plano de testes

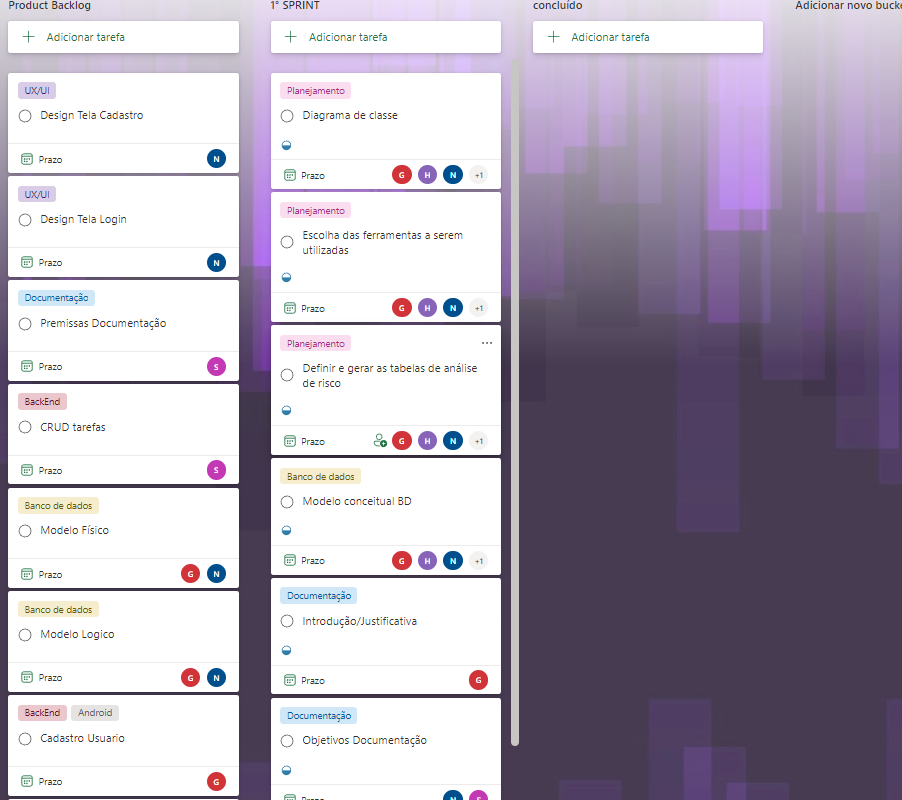
Nesse Sprint não realizo nenhum teste.

# Resultados

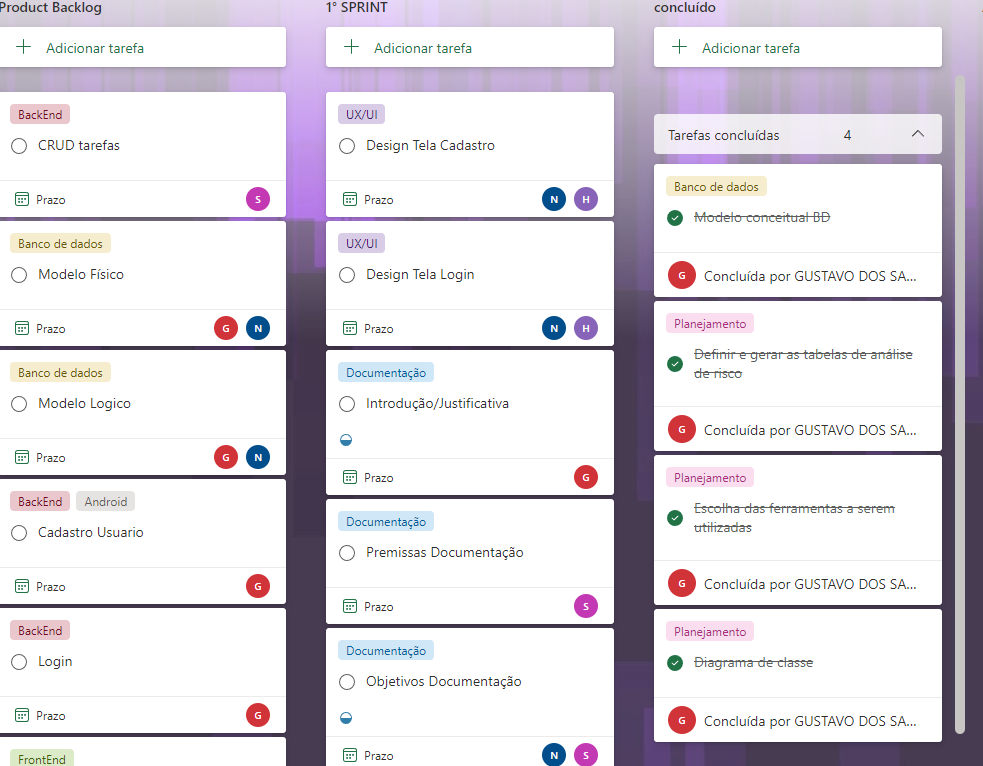
Não obtivemos nenhum resultado, pois não realizamos teste nesse Sprint

# Kanban e Retrospectiva

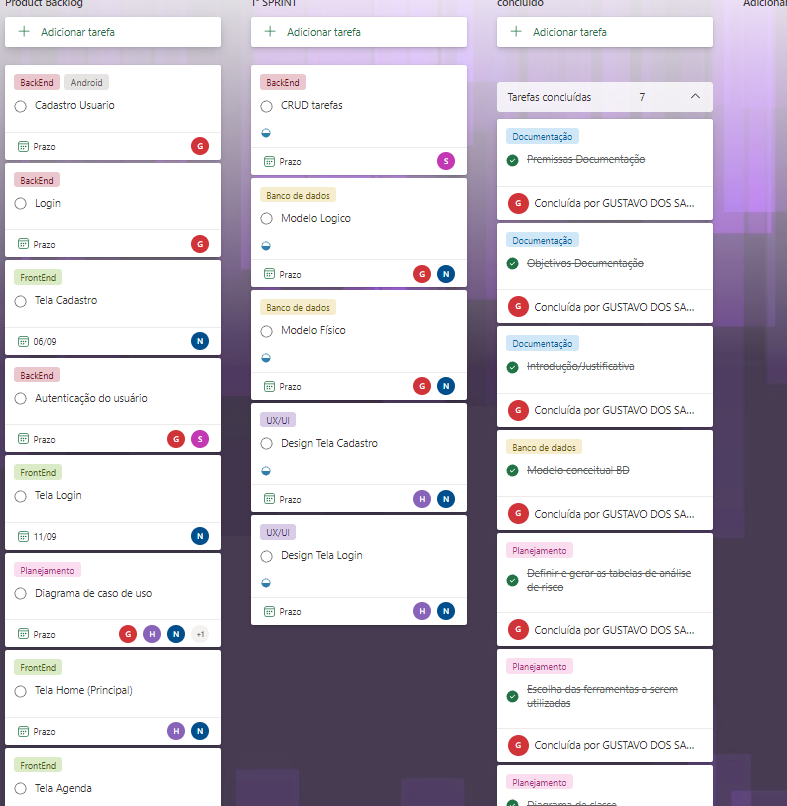
**Primeira semana: 12/08/2024 a 16/08/2024**



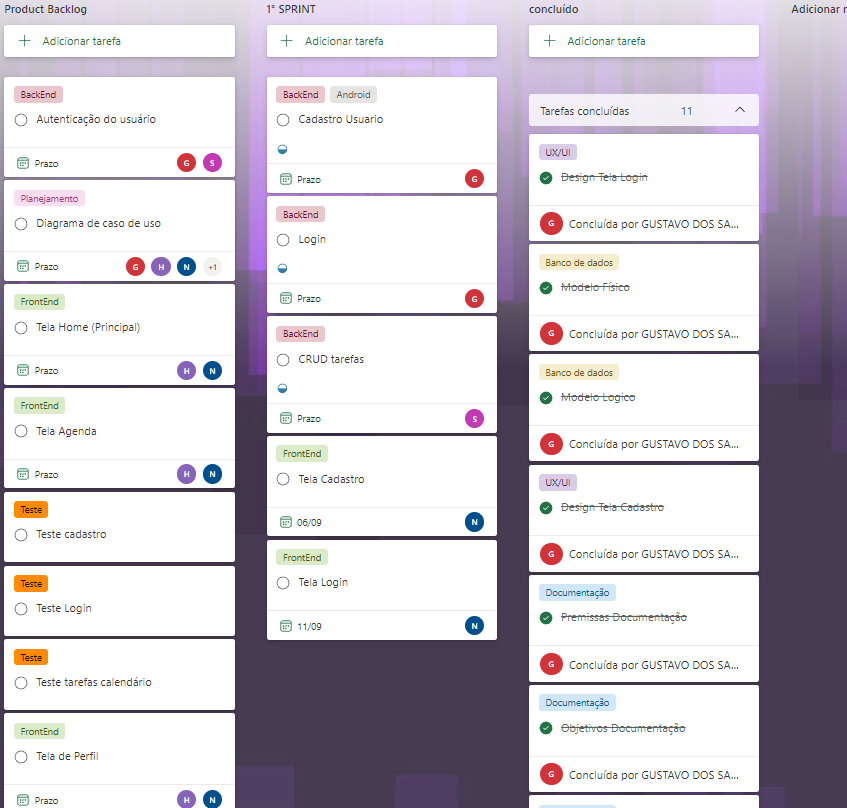
**Segunda semana: 19/08/2024 à 23/08/2024**



**Terceira semana: 26/08/2024 a 30/08/2024**



**Quarta semana: 02/09/2024 a 06/08/2024**



# Segundo Sprint

Nesta sprint, continuamos o desenvolvimento do back-end e decidimos também mudar o design das telas. Portanto, foi necessário alterar as telas do front-end que já estavam prontas no primeiro sprint. Conseguimos fazer essas alterações com sucesso e integramos todo o processo de login e cadastro. Ou seja, o banco de dados, o back-end e o front-end agora funcionam em conjunto.

# Product backlog

Não foram feitas alterações ao product backog

# Sprint Backlog

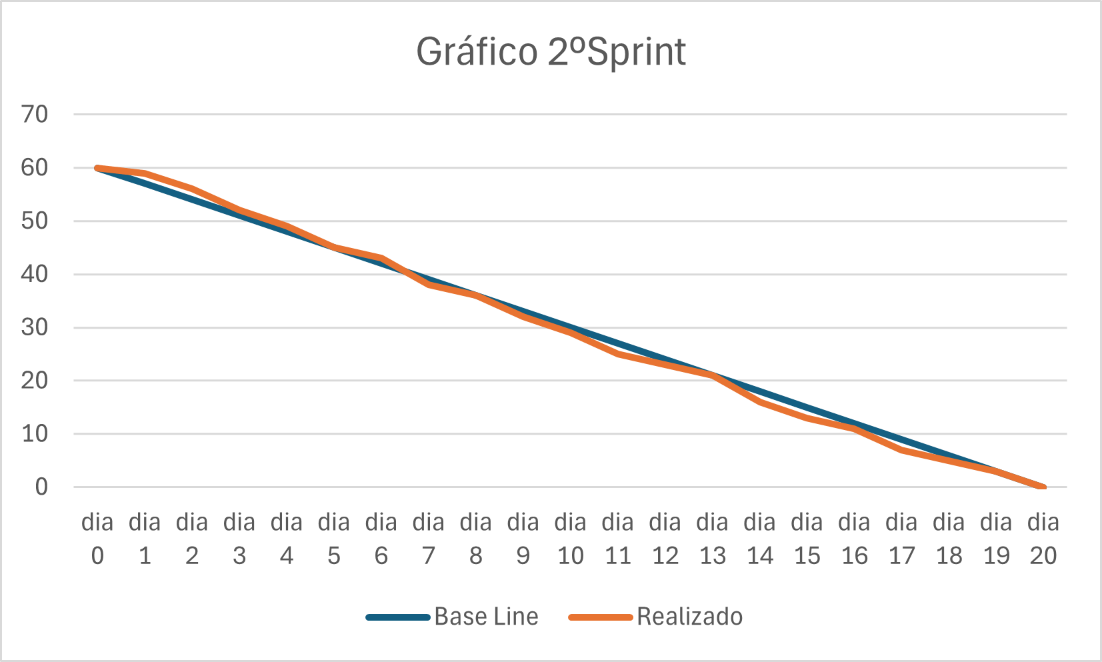
RF04.1 – O login (front) dever envias uma solicitação HTTP POST para o back-end e receber um token e nome do usuário

RF04.2 – O cadastro (front) deve enviar um HTTP POST para o back-end e receber uma mensagem de sucesso;

RNF02 – O sistema deve ser uma aplicação web responsiva, funcionando corretamente em computadores e dispositivos móveis;

RNF03 – Todas as senhas devem ser criptografadas antes de serem armazenadas no banco de dados, garantindo a segurança das informações;

# Burn Down Chart

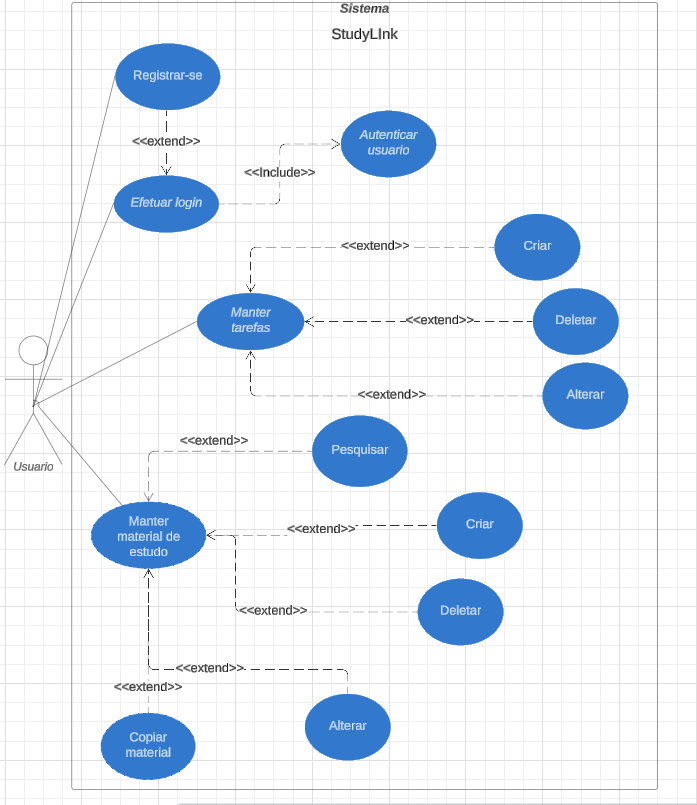


# 

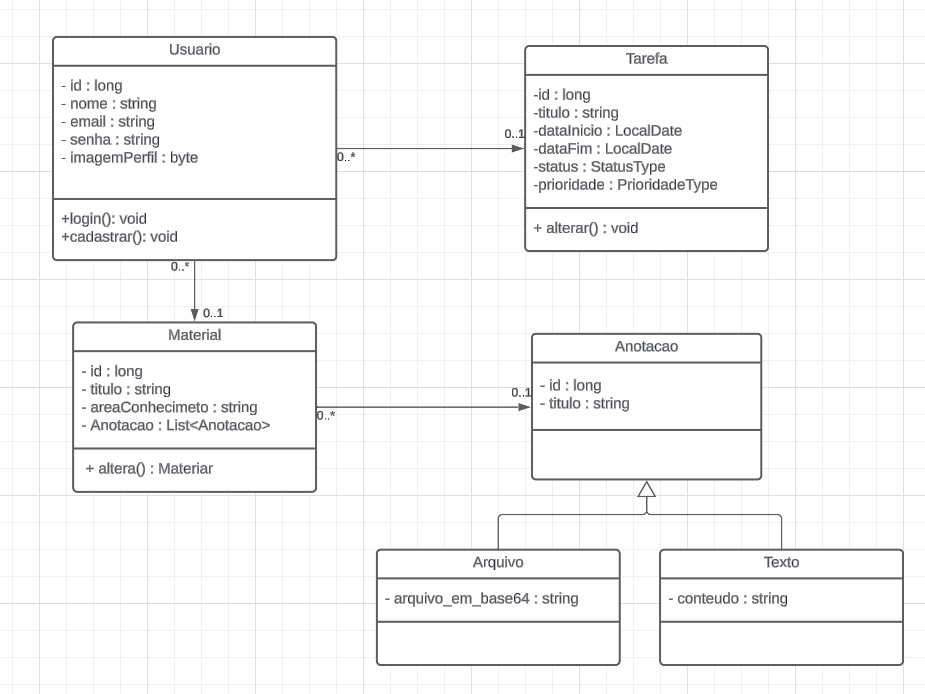
# Diagramas

Todos os diagramas feitos durante o sprint.

# Diagrama de caso de uso



# Diagrama de classes



# Diagrama de narrativa de caso de uso

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** CSU01-Registrar-se.  **Objetivo:** Permitir um registro de um usuário no site.  **Frequência estimada:** Sempre que um novo usuário querer se integrar ao site  **Ator Principal:** Usuário |
| **CENÁRIO PRINCIPAL** |
| 1. O caso de uso começa quando o usuário acessa o site e escolhe a opção cadastrar. 2. O site exibe a tela que comporta nome de usuário, e-mail, senha e confirmar senha. 3. O usuário preenche os campos com seus dados. 4. O site valida os dados para o registrá-lo no banco de dados. |
| **CENÁRIO ALTERNATIVO** |
| **CENÁRIO EXCESSÃO** |
| 1. **Falha na Validação** 2. Caso um ou todos os campos estejam vazios. 3. Caso o campo senha esteja com caracteres insuficientes. 4. Caso o campo e-mail esteja sendo registrado um e-mail que não existe. 5. Não existe nenhum limite quanto ao número de tentativas de acesso. |
| **PRÉ-CONDIÇÕES** |
| **PÓS-CONDIÇÕES** |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** CSU02-Login.  **Objetivo:** Permitir o login para o usuário no site.  **Frequência estimada:** Sempre que o usuário entrar no site  **Ator Principal:** Usuário |
| **CENÁRIO PRINCIPAL** |
| 1. O caso de uso começa quando o usuário entra no site e seleciona o login. 2. O site exibe a tela de login que será pedido: e-mail e senha. 3. O usuário ele escreverá suas informações nos campos. 4. O site valida o seu login para ver se a conta está registrada. |
| **CENÁRIO ALTERNATIVO** |
| **CENÁRIO DE EXECESSÃO** |
| 1. **Falha na Validação** 2. Caso um ou todos os campos estejam vazios 3. Caso o campo senha esteja com caracteres insuficientes 4. Caso o campo e-mail esteja sendo registrado um e-mail que não existe 5. Não existe nenhum limite quanto ao número de tentativas de acesso |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve tem que estar registrado primeiro |
| **PÓS-CONDIÇÕES** |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** CSU03- Manter tarefas.  **Objetivo:** Controlar o cadastro de tarefas, permitindo inclusão, alteração, exclusão e consulta pelas datas e títulos.  **Frequência estimada:** 1 a 5 vezes ao dia.  **Ator Principal:** Usuário |
| **CENÁRIO PRINCIPAL** |
| 1. O caso de uso começa quando o usuário vai na tela home e com o botão direto ele cria uma tarefa. 2. O sistema apresenta uma janela para controle de tarefas contendo as seguintes opções: Cadastro, Alteração, Excluir, Listagem (pela descrição) e voltar à tela anterior. 3. Após término da opção escolhida, o sistema irá voltar ao passo 2. 4. Caso o usuário escolha voltar, o caso de uso é encerrado e o sistema volta para a tela anterior. |
| **CENÁRIO ALTERNATIVO** |
| 1. Se o usuário clicar no botão direito do mouse no calendário abra uma janela para ele adicionar uma tarefa. 2. Se ele clicar com o botão esquerdo do mouse em uma tarefa que já existe ele pode alterar ou excluir.      1. **Cadastro** 2. O usuário seleciona a opção para cadastrar uma tarefa. 3. O sistema apresenta uma tela contendo os campos referentes aos dados da tarefa (título, descrição, dtInicio, dtFinalização e prioridade) 4. O usuário preenche os dados da nova tarefa e pressiona o botão para adicionar 5. O sistema valida os dados e solicita a confirmação do usuário 6. O usuário confirma o cadastro. 7. O sistema realiza o cadastro da tarefa no banco de dados 8. O usuário pressiona o botão voltar e o caso de uso retorna ao passo 2      1. **Alteração** 2. O atendente seleciona a opção para alterar medicamento 3. O sistema apresenta uma lista contendo diversas tarefas e uma barra de pesquisa para realiza a pesquisa por meio da descrição. 4. O usuário seleciona a tarefa a ser alterado. 5. O sistema apresenta os dados da tarefa a ser alterado 6. O usuário altera os dados desejados e pressiona o botão atualizar 7. O sistema realiza a validação dos dados e atualiza o banco de dados 8. O usuário pressiona o botão voltar e o caso de uso retorna ao passo 2        1. **Exclusão**      1. O usuário seleciona a opção para excluir a tarefa 2. O sistema apresenta uma lista contendo diversas tarefas se uma barra de pesquisa para que o usuário possa realizar uma pesquisa por meio do código. 3. O usuário seleciona a tarefa para ser excluído e pressiona o botão excluir. 4. O sistema solicita a confirmação da exclusão.      1. Se o usuário confirmar a exclusão o sistema exclui o medicamento do banco de dados; caso contrário, retorna ao passo 3.3.2. 2. O usuário pressiona o botão voltar e o caso de uso retorna ao passo 2.        1. **Listagem** 2. O usuário seleciona a opção para listar a tarefa. 3. O sistema apresenta uma lista contendo diversas tarefas e barra de pesquisa para que o usuário possa realizar uma pesquisa por meio da descrição. 4. O usuário seleciona a tarefa que deseja ver. 5. O sistema apresenta todos os dados referentes as tarefas. 6. O usuário pressiona o botão voltar e o caso de uso retorna ao passo 2. |
| **CENÁRIO DE EXECESSÃO** |
| 1. **Falha de cadastro** 2. Se uma tarefa for criada no mesmo horário que a outra já está, dará errado. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve estar registrado pelo sistema. |
| **PÓS-CONDIÇÕES** |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** CSU04- Manter material de estudos.  **Objetivo:** Controlar o cadastro de matérias.  **Frequência estimada:** 1 a 5 vezes ao dia.  **Ator Principal:** Usuário |
| **CENÁRIO PRINCIPAL** |
| 1. O caso de uso começa quando o usuário seleciona a opção de material no menu. 2. O site apresenta uma tela com um botão onde o usuário irá adicionar um material. 3. Irá abrir uma janela para cadastrar a matéria áreas com os campos (título, área de conhecimento). 4. Na hora que clicar na matéria adicionada irá aparecer as opções de PDF, foto/vídeo, áudio e anotações. 5. Após isso é adicionado na tela de material |
| **CENÁRIO ALTERNATIVO** |
|  |
| **CENÁRIO DE EXECESSÃO** |
| 1. **Falha de tamanho(img)** 2. Caso a imagem colocada seja maior do que o sistema comporta dará uma falha e não carregara a imagem. 3. **Falha de limite de caracteres(anotação)** 4. Caso a anotação passar de um limite de caracteres previsto irá dar erro no sistema. |
| **PRÉ-CONDIÇÕES** |
| 1. O usuário deve estar registrado pelo sistema. |
| **PÓS-CONDIÇÕES** |

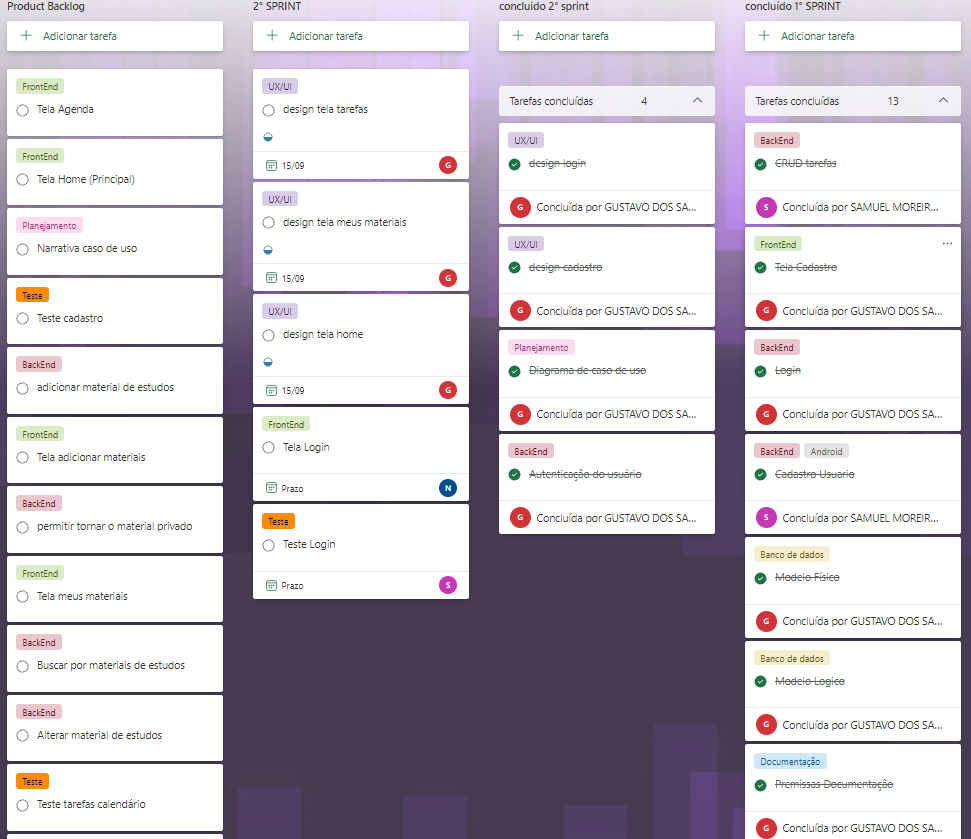
# Plano de testes

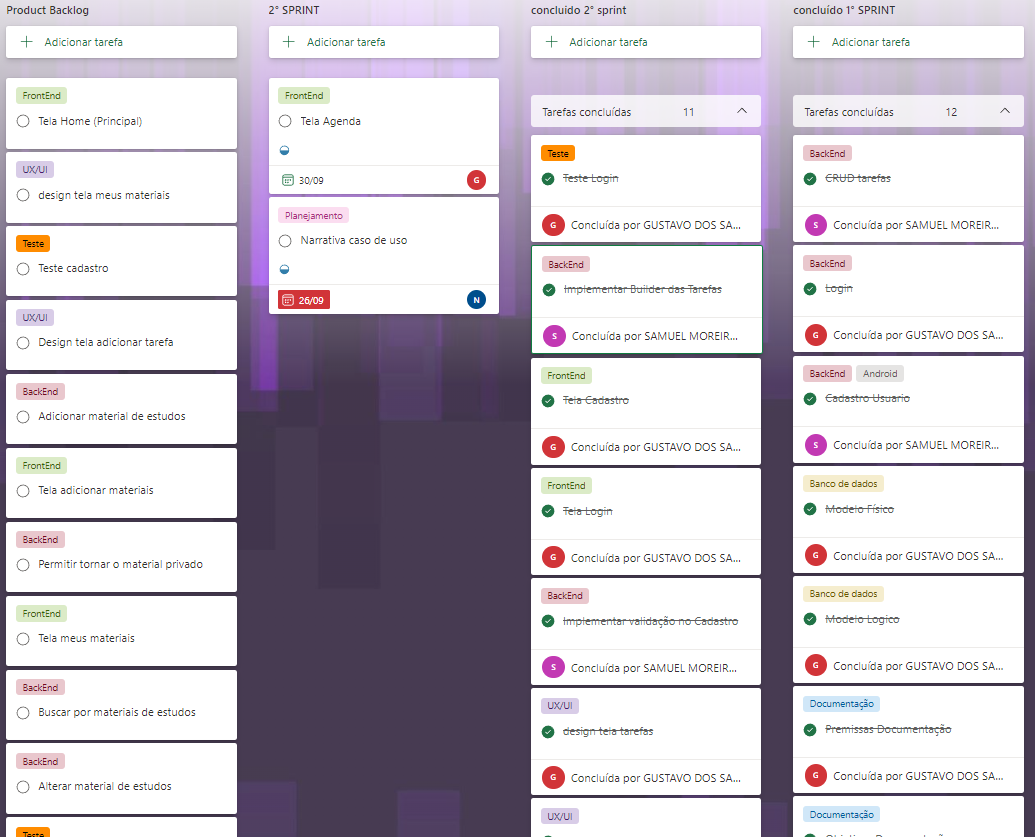
|  |
| --- |
| **Plano de teste login** |
| **Nome:** Gustavo Dos Santos Guimaraes Data: 23/09/2024 |
| **Objetivo:**  Este teste tem como objetivo verificar se o login de um usuário está funcionando corretamente no aplicativo. Ao enviar dados válidos via requisição HTTP POST o sistema deve receber da API o nome do usuário e seu token de autenticação. |
| **Descrição do Cenário:**  O usuário insere suas credenciais (e-mail e senha) no formulário de login e o envia. A aplicação envia uma requisição ao back-end para autenticação. Caso as credenciais sejam válidas, a aplicação deve:   * Receber uma resposta da API com um token de autenticação e o nome do usuário. |
| **Pré-requisitos:**   * A aplicação back-end deve estar em execução * A porta <http://localhost:8084/auth/login>deve estar acessível. * O usuário deve estar cadastrado com um e-mail e senha válidos. * A URL <http://localhost:4200/login> estar disponível |
| **Passos:**   1. Acessar <http://localhost:4200/login>; 2. Inserir no campo e-mail: [senai@aluno.com](mailto:senai@aluno.com) e no campo senha: 12345678; 3. Clicar no botão entrar |

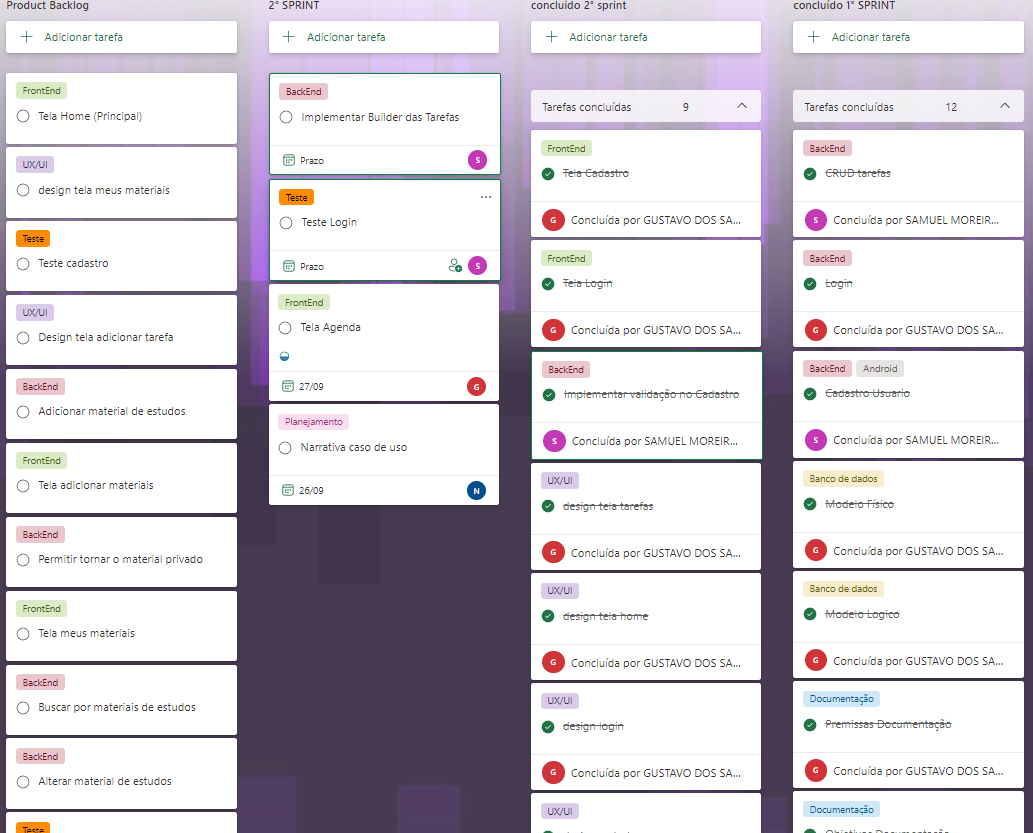
# Resultados

|  |  |  |  |
| --- | --- | --- | --- |
| **Resultado teste login** | | | |
| **Resultados esperados:**   * Um toast verde com uma mensagem “Login efetuado com sucesso”; * Redirecionar para a página home; * Um status 200 OK; * A resposta da API deve conter {token: "aqui o token gerado", nome\_usuario: "Senai"} | | | |
| Tentativa/ID | Responsável | Resultado do teste | Data: |
| 01 | Gustavo Guimaraes | Erro | 23/09/2024 |
| 02 | Gustavo Guimaraes | Sucesso | 25/09/2024 |
| **Resultados obtidos:**   1. Retornou um erro 403 **Forbidden, pois o CORS (Cross-Origin Resource Sharing) não estava configurado corretamente. Para corrigir o erro, foi configurado no backend que o CORS aceite as requisições feitas da origem** <http://localhost:4200> com **os métodos (POST, PUT e DELETE)** 2. Todos os resultados esperados foram alcançados | | | |

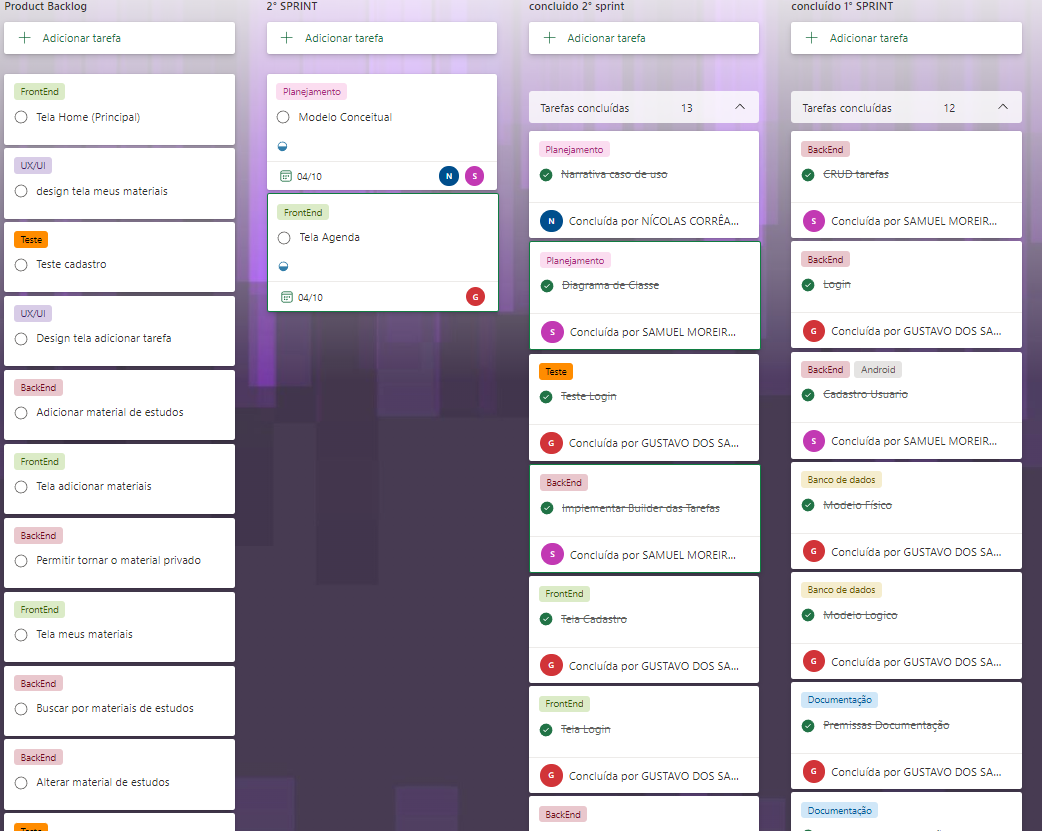
# Kanban e Retrospectiva

**Primeira semana: 09/09/2024 a 13/09/2024**

**Segunda semana: 16/09/2024 à 20/09/2024**

**Terceira semana: 23/06/2024 a 27/09/2024**

**Quarta semana: 30/09/2024 a 04/10/2024**



**Retrospectiva:**

Após analisar os acontecimentos do segundo sprint o grupo chegou as seguintes conclusões:

**Coisas que deram certo:** Conseguimos aprender e compreender o Angular, o que era um dos maiores risco, o grupo conseguiu debater e entrar em um consenso sobre o design das telas, a parte do front-end de login e o cadastro está interagindo com a parte do back-end.

**O que deu errado:** Nos atrasamos pela falta de pratica com novas tecnologias, também organizamos mal o planejamento das tarefas e o que faríamos no sprint.

**O que podemos melhorar:** No próximo sprint iremos organizar melhor o planner e as tarefas atribuídas para cada integrante da equipe, vamos também reservar um tempo maior para cuidar da parte de documentação e ir fazendo junto com as outras tarefas.

# Terceiro Sprint

Durante este sprint, avançamos significativamente no desenvolvimento do back-end, que agora está próximo de ser concluído. No front-end, progredimos de forma notável na construção de telas, restando apenas a duas para completar o design planejado. Esse avanço nos aproxima da finalização da interface deixando o sistema mais integrado e funcional.

* + 1. **Product backlog**

Não houve nenhuma alteração no product backlog.

# Sprint Backlog

RF07– Materiais de estudo;

RF07.1 – Poderá alterar seu material;

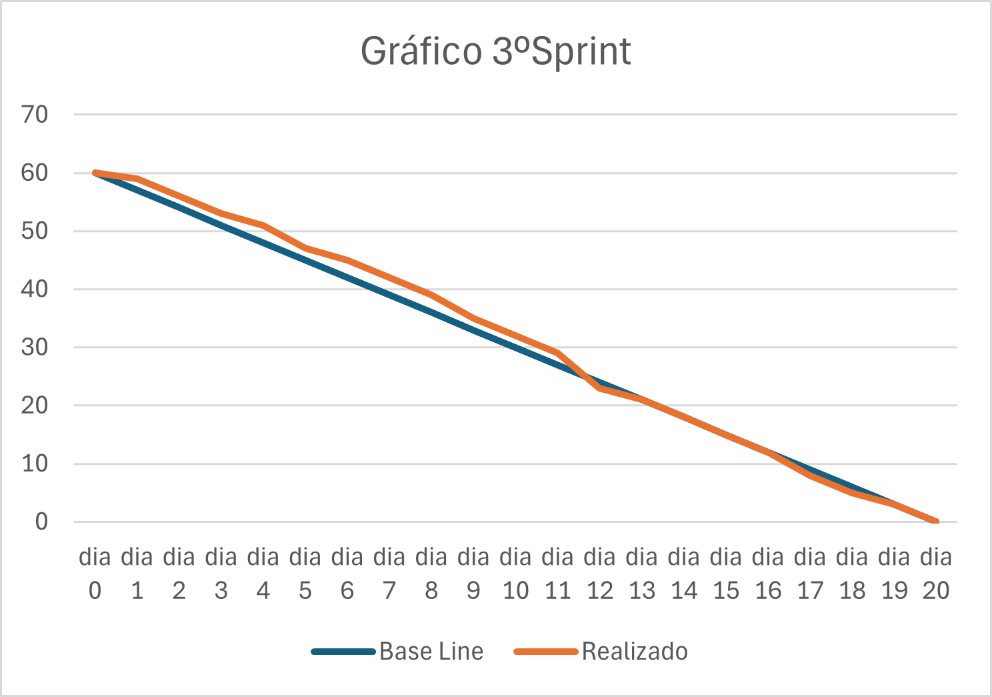
RF07.2 – Poderá adicionar material;

RF07.3 – Poderá pesquisar o material pelo título;

RF07.4 – Poderá deletar o material de estudos;

RF07.5 – Deixa-lo visível ou não, para outros usuários;

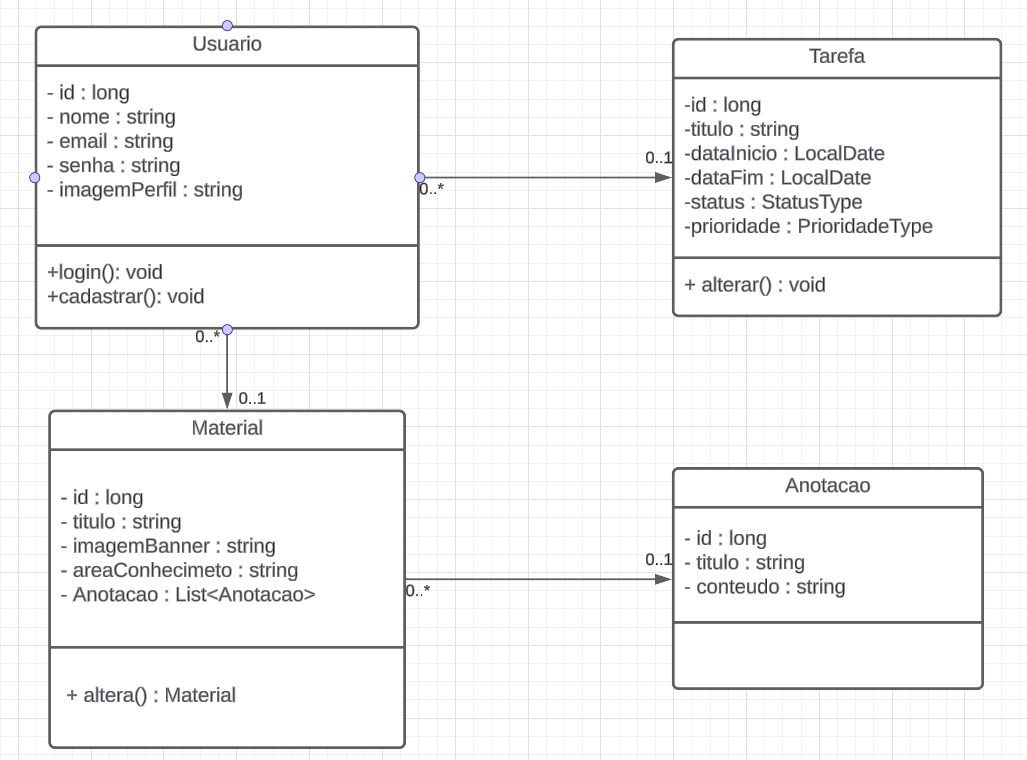
# Burn Down Chart

****

# Diagramas

Todos os Diagramas já foram feitos no 2°Sprint, mas só o diagrama de classe foi alterado, assim será colocado aqui a versão atualizada.

# Diagrama de classe



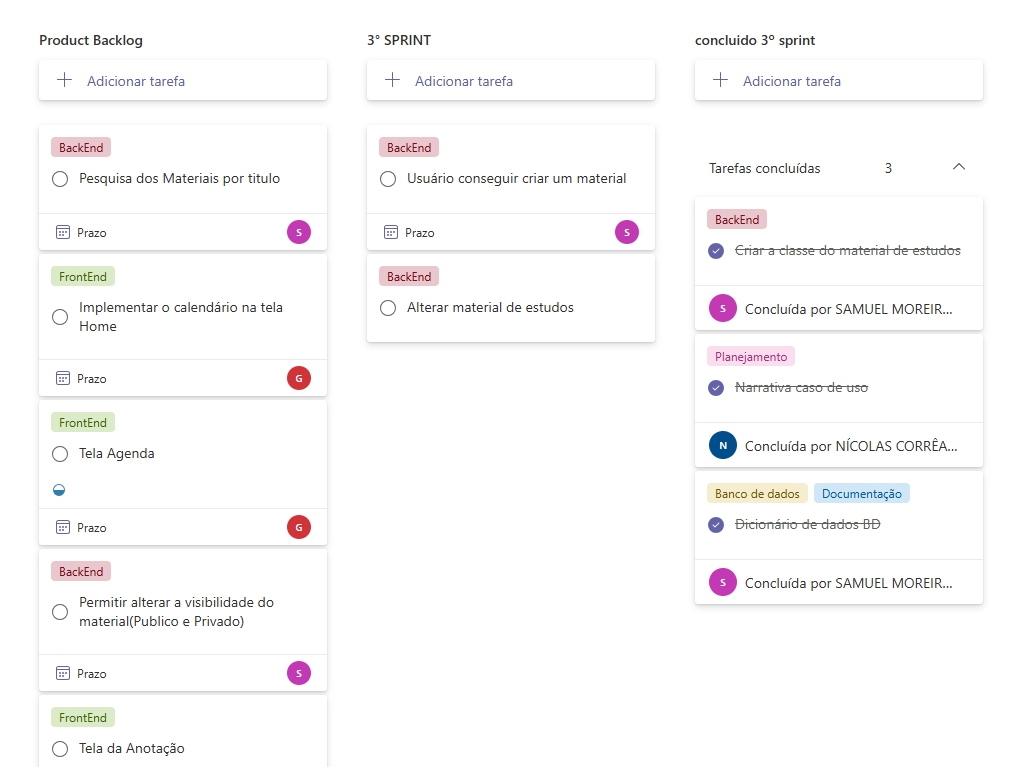
# Plano de teste

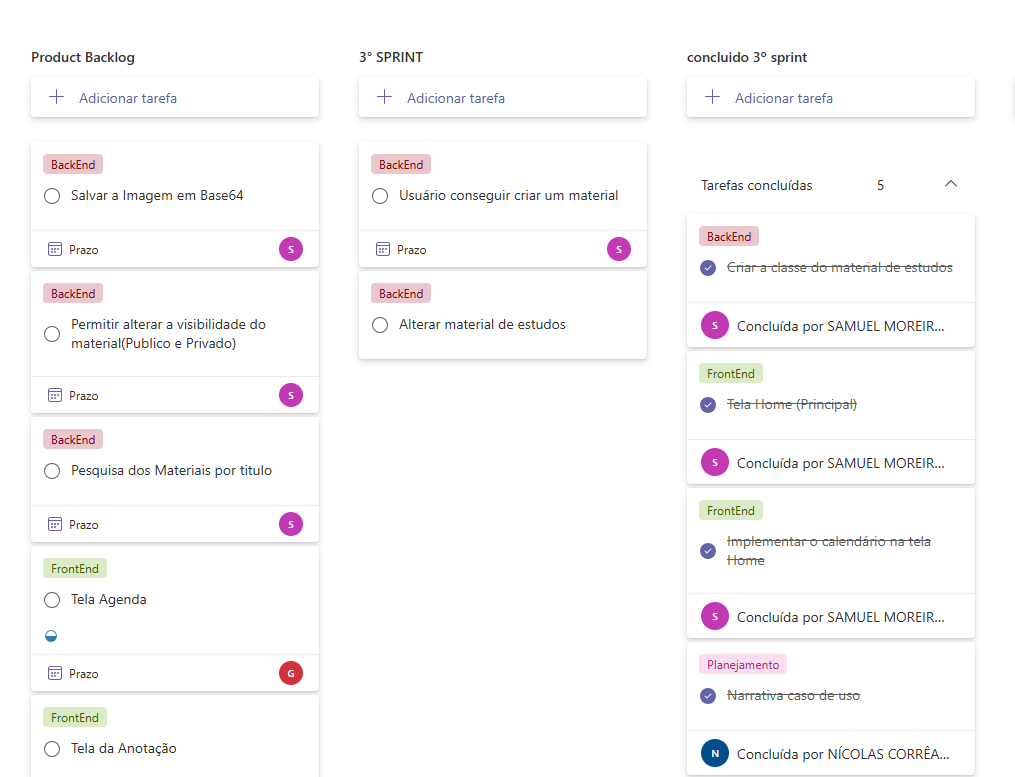
Não foram feitos testes nesse sprint.

# Resultado

Não foi obtido nenhum resultado por não ter tido nenhum teste esse sprint.

# Kanban e Retrospectiva

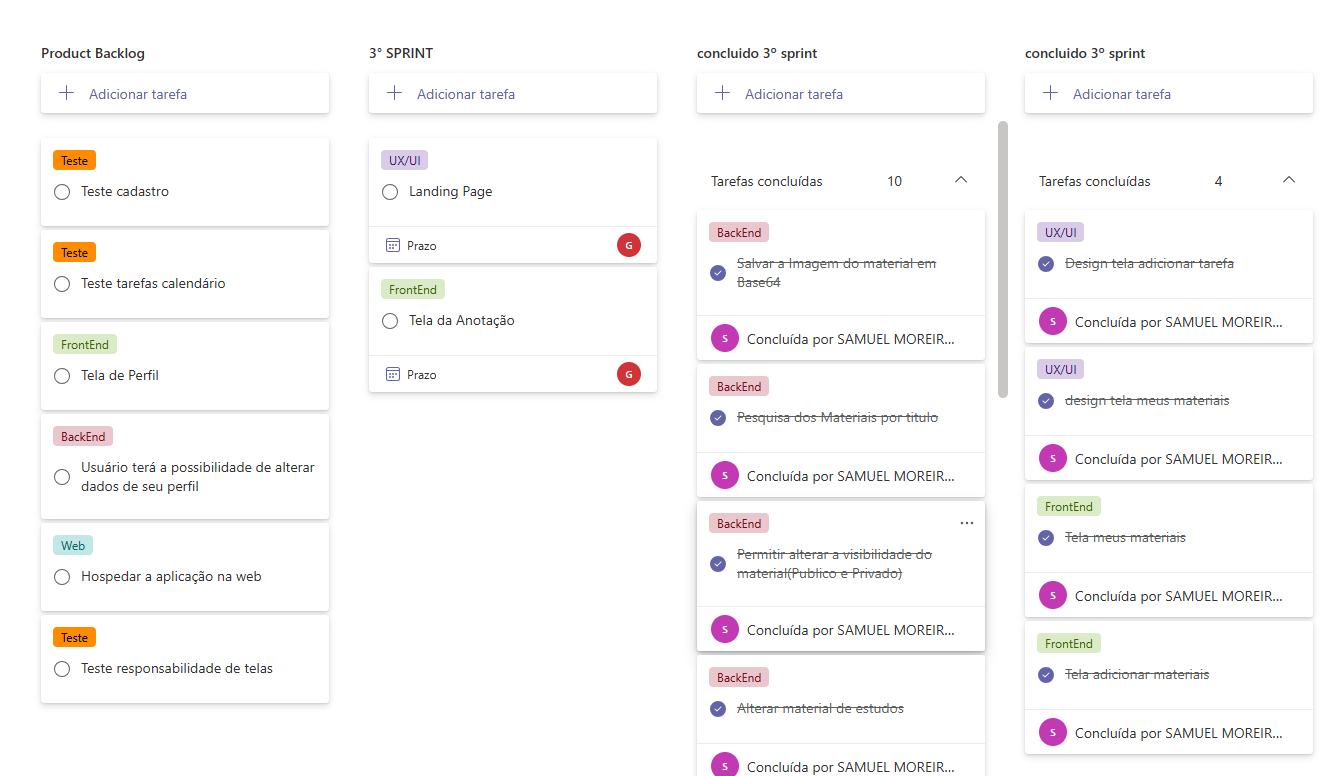
**Primeira semana**

**Segunda Semana**

**Terceira semana**

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

**Quarta Semana**

**Retrospectiva:**

Após analisarmos os acontecimentos do terceiro sprint chegamos à conclusão:

**Coisas que deram certo:** conseguimos avançar muito no projeto, o back-end está quase pronto, e o front está no estágio final.

**O que deu errado:** Nesse sprint o grupo acabou não dando tanta relevância ao front-end, o que consequentemente ocorreu um atraso no desenvolvimento do projeto.

**Oque podemos melhorar:** No próximo sprint o grupo colocara uma atenção melhor para o front e testes.

# Quarto Sprint

Nesse sprint, concluímos as pendencias anteriores, finalizando o desenvolvimento do back-end e do front-end, juntamente com suas funcionalidades. Com isso, conseguimos integrar ambos, garantido que eles se comuniquem de forma eficiente e atendam aos requisitos propostos.

# Product Backlog

Não houve nenhuma alteração no product backlog.

# Sprint Backlog

RF08 –. Usuário adicionar foto de perfil depois de logado;

RF09 –. Quando habilitado deve guardar o usuário nos cookies do navegador;

# Burn Down Chart

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

# Diagramas

Nesse sprint, na maioria dos diagramas não houve nenhuma alteração, somente no digrama de classe que houve alteração.

# Diagrama de classe

# Diagrama Descrição gerada automaticamente

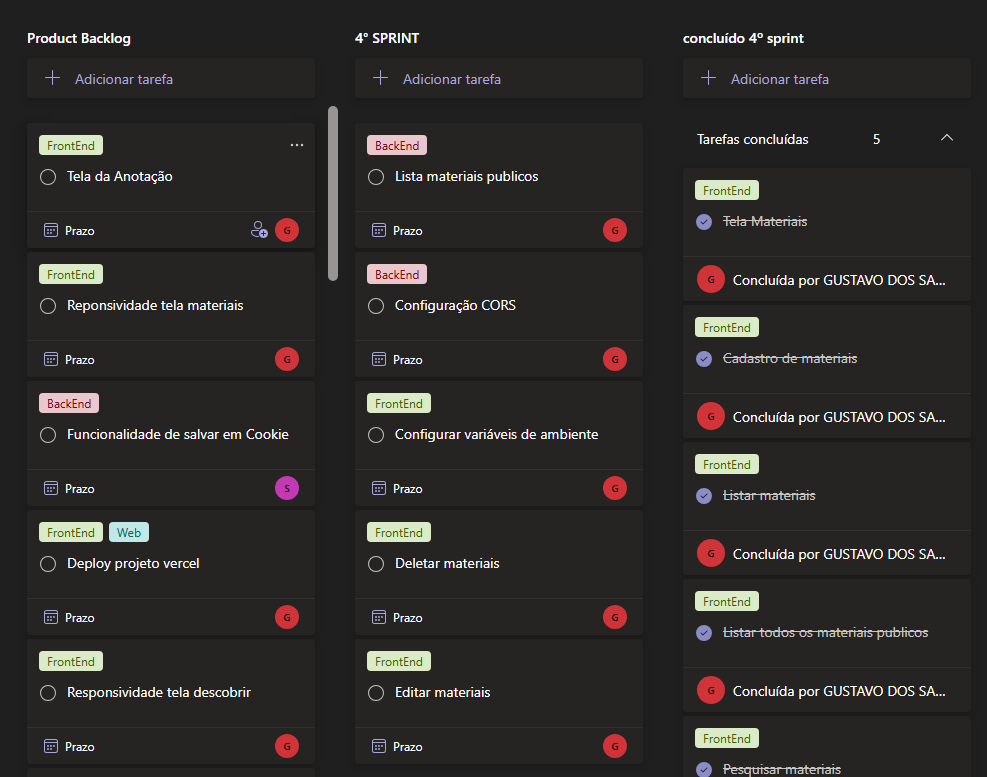
# Plano de Teste

|  |
| --- |
| **Plano de teste cadastro** |
| **Nome:** Nícolas Corrêa de Souza Data: 9/12/204 |
| **Objetivo:**  Este teste tem o objetivo de verificar se o cadastro de um usuário está funcionando corretamente no aplicativo. Ao enviar dados incorretos via requisição HTTP POST o sistema deve receber da API o campo que está incorreto e mensagem de erro. |
| **Descrição do Cenário:**  O usuário inseri suas credenciais (Nome, e-mail, senha e confirmar senha) no formulário de cadastro e o envia. A aplicação envia uma requisição back-end para autenticação. Caso as credenciais sejam validas, a aplicação deve:   * Receber uma resposta da API com o campo que está incorreto e mensagem de erro. |
| **Pré-requisitos:**   * A aplicação back-end deve estar em execução * A porta <http://localhost:8084/api/register>deve estar acessível. * A URL <http://localhost:4200/cadastro> estar disponível |
| **Passos:**   1. Acessar <http://localhost:4200/cadastro>; 2. Inserir no campo Nome: SE, Email: Senai, senha Senai; 3. Clicar no botão cadastrar; |

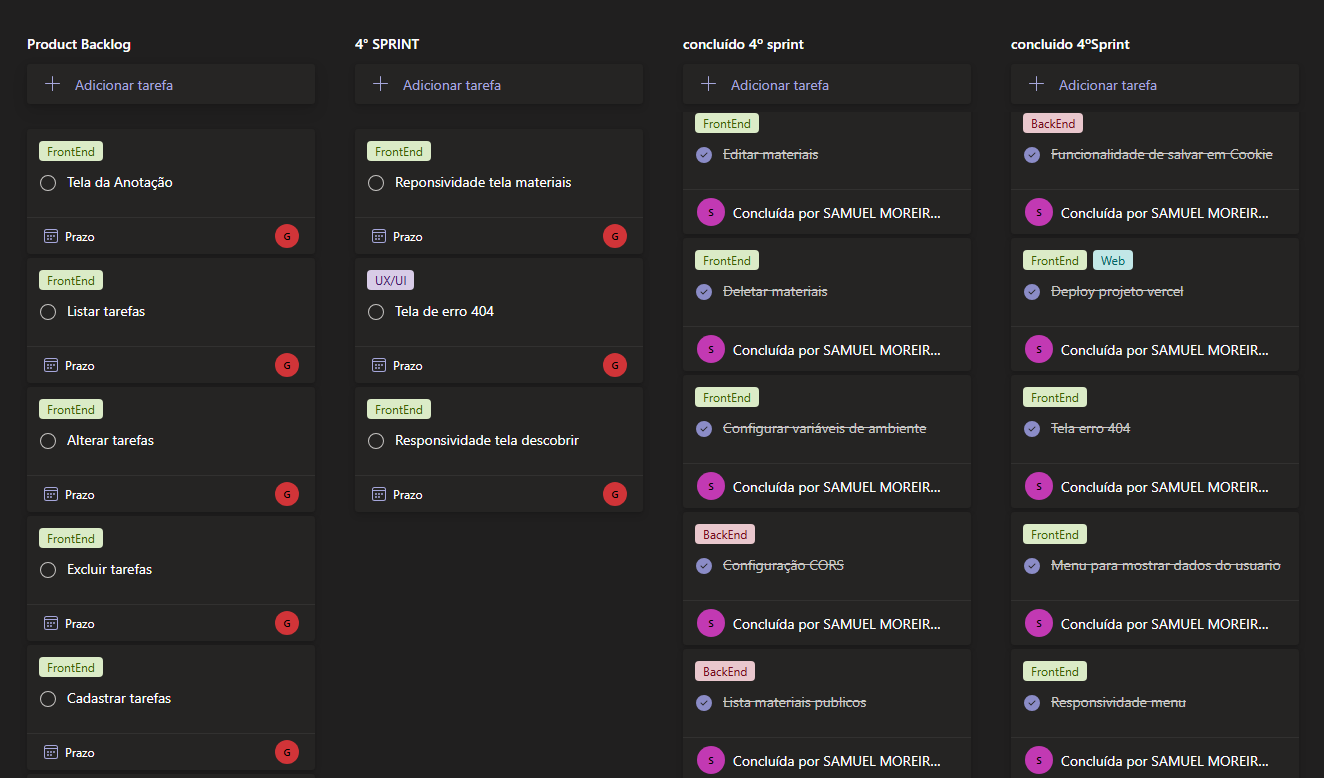
# Resultado

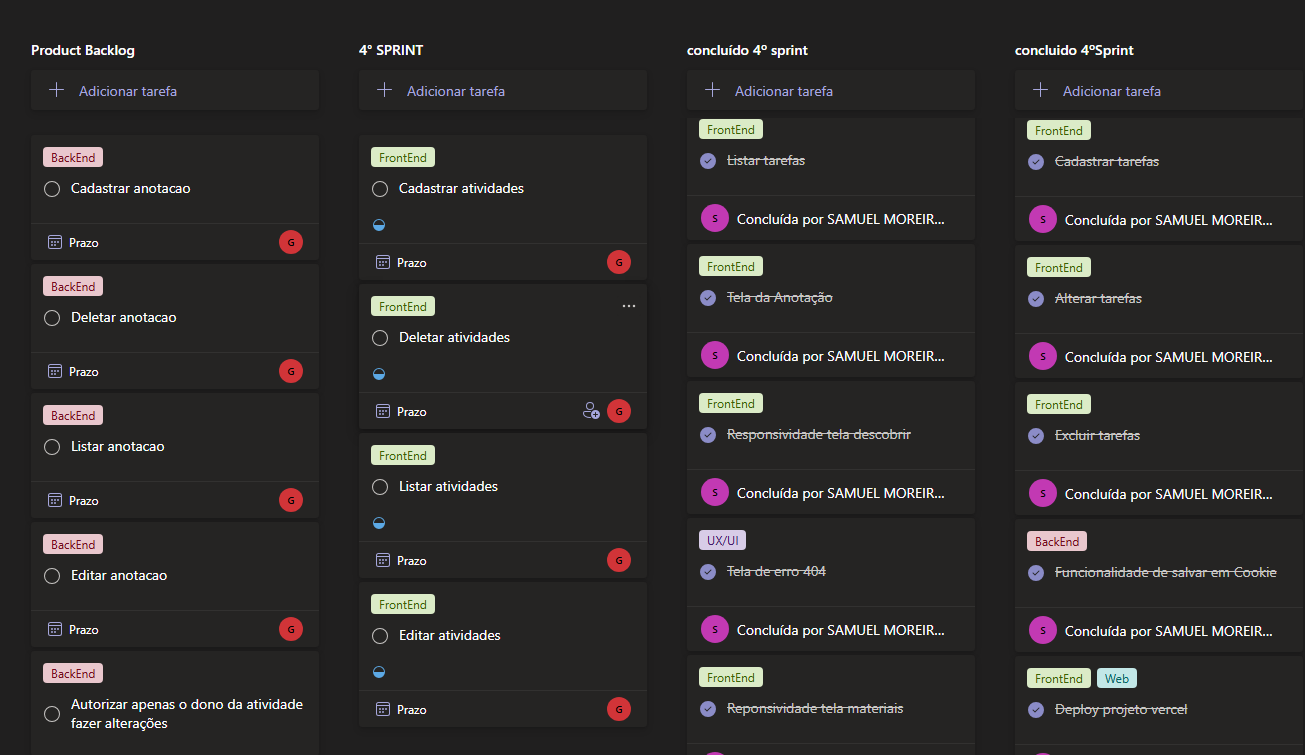
|  |  |  |  |
| --- | --- | --- | --- |
| **Resultado teste cadastro** | | | |
| **Resultados esperados:**   * Um toast laranja com uma mensagem “Por favor, preencha os campos corretamente”; * Um status 400 bad request; * A resposta da API deve conter {campo: "campo errado", mensagem: “mensagem de erro”} | | | |
| Tentativa/ID | Responsável | Resultado do teste | Data: |
| 01 | Nícolas Corrêa | Sucesso | 9/12/2014 |
| **Resultados obtidos:**   1. Todos os resultados esperados foram alcançados. | | | |

# Kanban e Retrospectiva

**Primeira semana**

**Segunda semana**

****

**Terceira semana**

**Quarta semana**

**Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente**

**Retrospectiva:**

Após analisarmos os acontecimentos do quarto sprint chegamos à conclusão:

**Coisas que deram certo:**

Conseguimos finalizar todas as tarefas planejadas para este sprint, alcançando os objetivos estabelecidos. Concluímos o desenvolvimento do sistema web, integrando front-end e back-end com sucesso. Essa conquista marca o encerramento das funcionalidades essenciais do projeto, garantido que os componentes se comuniquem adequadamente e atendam às necessidades definidas.

**O que deu errado:**

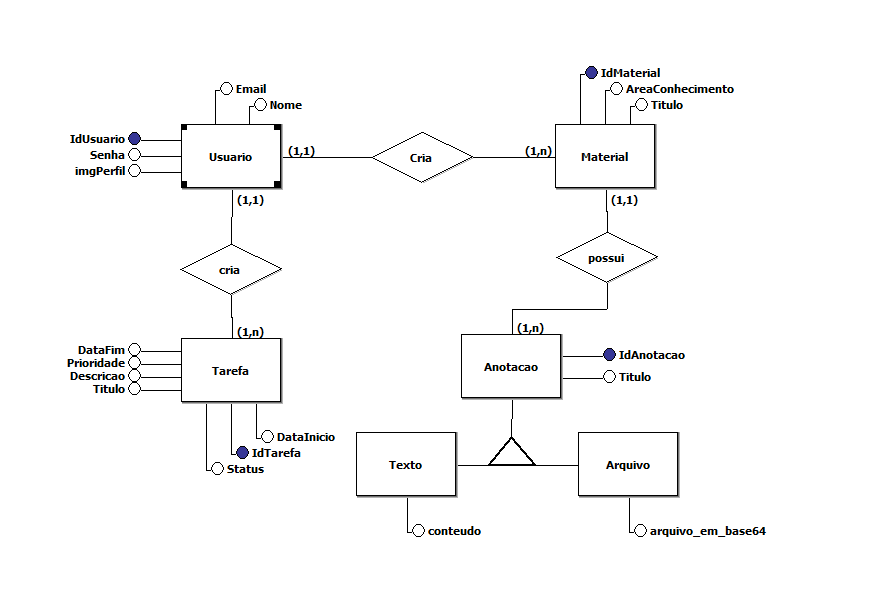
O que prejudicou a gente nesse sprint, foi o rendimento do término de

algumas funcionalidades do front-end e com mudanças repentinas no back-end.

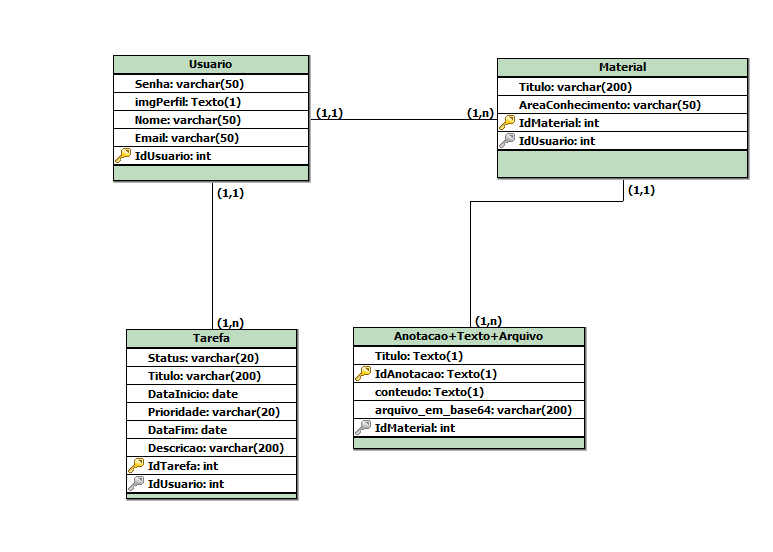
# Modelo de Dados

Esta parte do planejamento traz informações necessárias para a construção de um banco de dados para o Sistema de Gerenciamento de Acessos.

# Modelo conceitual do banco de dados



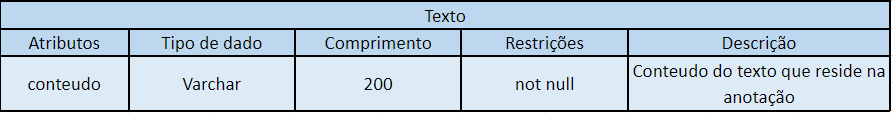
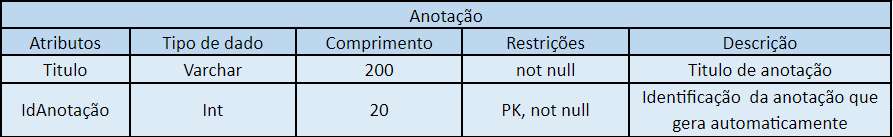
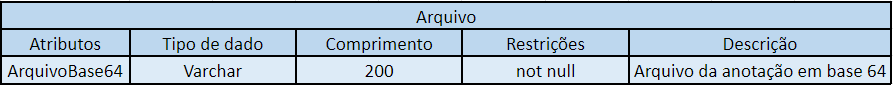
# Modelo lógico do banco de dados



# Dicionário de dados

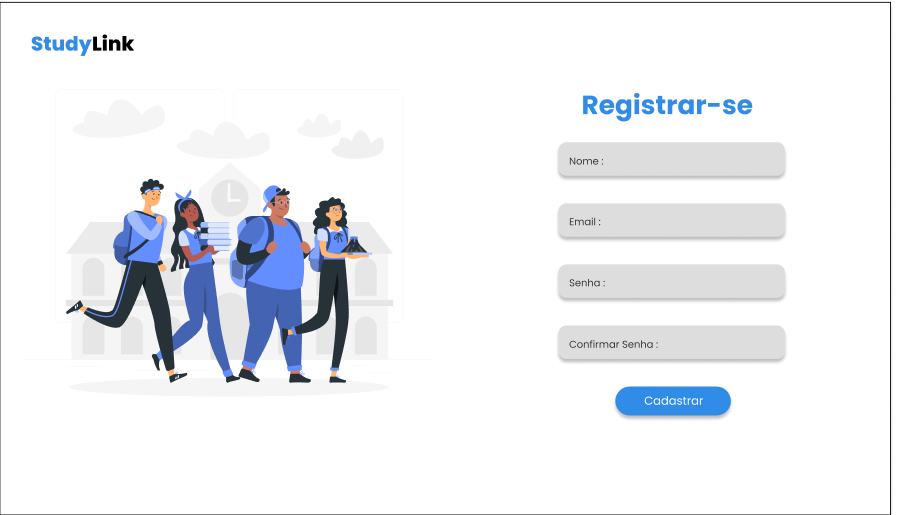
****Tabela

Descrição gerada automaticamente

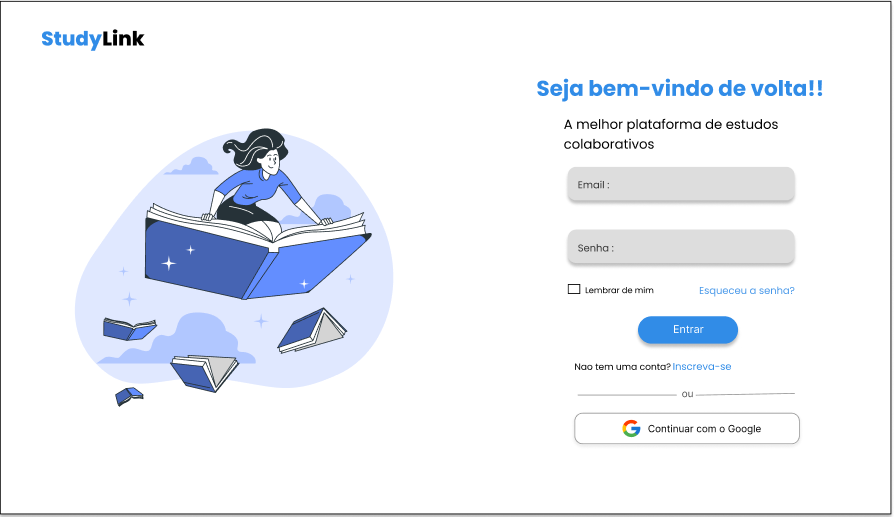


# PRINCIPAIS TELAS DO SISTEMA

Tela de cadastro. Campos: nome, e-mail, senha e confirmar senha.



Tela de Login. Campos: e-mail e senha.



Tela Home, onde fica o calendário e as tarefas**.**

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Tela de material, onde cria, guarda materiais.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Tela de descobrir, achar materiais de outros usuários.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Tela de anotação, anotar o conteúdo das matérias.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

# CONCLUSÃO

# Escreva os resultados obtidos

Nosso projeto foi desenvolvido como uma plataforma web que permite aos usuários agendarem e criar atividades, além de produzir e compartilhar materiais de estudo. Essa funcionalidade possibilita que outros usuários acessem conteúdos de seus interesses, contribuindo para superação do déficit de matérias disponíveis e promovendo um ambiente colaborativo de aprendizado.

Durante os seis meses de desenvolvimento, enfrentamos desafios que exigiram planejamento e adaptação. No entanto, conseguimos manter uma progressão constante ao longo do processo. O resultado reflete os esforços da equipe e atende aos objetivos propostos, consolidando o projeto como uma solução prática e eficiente para facilitar o acesso a conteúdo educacionais.

# Constatações

Ao longo do projeto, exploramos diferentes ideias até definirmos a proposta final, focada em desenvolver uma plataforma que pudesse contribuir significativamente para os estudos. Durante esse processo, buscamos constantemente aperfeiçoar o conceito, ajustando-o para entender melhor às necessidades dos usuários.

Os seis meses de trabalho proporcionaram um grande aprendizado, especialmente ao lidarmos com os erros e desafios enfrentados. Cada obstáculo superado foi uma oportunidade de evolução, permitindo-nos amadurecer como equipe e como desenvolvedores. No final, conseguimos alcançar os objetivos estabelecidos, entregando um projeto alinhando às propostas iniciais e com potencial para impactar positivamente a experiência de aprendizado dos usuários.

# Sugestões de possíveis aperfeiçoamentos técnicos

Para continuar aprimorando e evoluindo a plataforma, uma ideia seria a adição de uma comunidade interativa, onde os usuários possam, se comunicar sobre as matérias de forma mais dinâmica. Esse espaço permitiria debates, troca de conhecimentos e suporte mútuo, fortalecendo o senso de colaboração. Além disso, seria interessante implementar uma funcionalidade que possibilite a resolução conjunta de atividades, incentivando o aprendizado colaborativo e o compartilhamento de estratégias para resolver problemas. Outra proposta seria permitir que os usuários criem e enviem atividades uns aos outros, promovendo um ambiente ainda mais participativo e personalizado. Esses aperfeiçoamentos poderiam ampliar significativamente o engajamento e a utilidade de plataforma tornando-a um recurso educacional ainda mais completo.

# REFERÊNCIAS

Estadão. 95% dos alunos saem do ensino médio sem conhecimento adequado em matemática. **Exame**, 2021. Disponível em: https://exame.com/brasil/95-dos-alunos-saem-do-ensino-medio-sem-conhecimento-adequado-em-matematica/. Acesso em: 29 ago. 2024.

OLIVEIRA, Marco Orlando. PISA 2022: RESULTADOS MOSTRAM UM GRANDE DESAFIO PARA A EDUCAÇÃO BRASILEIRA. **Vídeo e Aprendizagem,** 2024. Disponível em: https://videoeaprendizagem.com/2024/01/17/pisa-2022-resultados-mostram-um-grande-desafio-para-a-educacao-brasileira/. Acesso em: 30 ago. 2024.

MEC, Assessoria De Comunicação Social Do Inep. Divulgados os resultados do Pisa 2022. **Ministério da Educação,** 2023. Disponível em: https://www.gov.br/inep/pt-br/assuntos/noticias/acoes-internacionais/divulgados-os-resultados-do-pisa-2022. Acesso em: 30 ago. 2024.

SANTANDER, Universidade. Método Feynman. **OpenAcademy,** 2024. Disponível em: https://www.santanderopenacademy.com/pt\_br/blog/metodo-feynman.html. Acesso em: 30 ago. 2024.

# GLOSSÁRIO

**API (Application Programming Interface)** - Uma interface que permite a comunicação entre diferentes sistemas de software.

**Componentização** - O processo de dividir um sistema ou aplicação em partes menores e reutilizáveis chamadas componentes. Cada componente encapsula uma funcionalidade específica e pode ser desenvolvido, testado e mantido de forma independente.

**Scrum** - Metodologia ágil de gerenciamento de projetos que utiliza iterações chamadas "sprints" para desenvolvimento incremental. Inclui papéis como Product Owner, Scrum Master e equipe de desenvolvimento, além de reuniões diárias e retrospectivas.

**Kanban** - Abordagem ágil focada na visualização do fluxo de trabalho. Utiliza um quadro com colunas representando etapas do processo, em que tarefas são movidas entre as colunas para facilitar o monitoramento do progresso e identificação de gargalos.

**Planner** - Ferramenta ou aplicativo utilizado para planejamento e organização de atividades, tarefas e metas.