

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia de Software

# **SAMI: Sistema para análise de métricas de indicadores de aprendizagem**

Autor: João Vitor de Moura Rosa Silva  
Orientador: Prof. Dr. Sergio Antônio Andrade de Freitas

Brasília, DF  
2022





João Vitor de Moura Rosa Silva

# **SAMI: Sistema para análise de métricas de indicadores de aprendizagem**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Sergio Antônio Andrade de Freitas

Brasília, DF

2022

---

João Vitor de Moura Rosa Silva

SAMI: Sistema para análise de métricas de indicadores de aprendizagem/ João Vitor de Moura Rosa Silva. – Brasília, DF, 2022-  
82 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Sergio Antônio Andrade de Freitas

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA , 2022.

1. Palavra-chave01. 2. Palavra-chave02. I. Prof. Dr. Sergio Antônio Andrade de Freitas. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. SAMI: Sistema para análise de métricas de indicadores de aprendizagem

CDU 02:141:005.6

---

João Vitor de Moura Rosa Silva

## **SAMI: Sistema para análise de métricas de indicadores de aprendizagem**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 14 de fevereiro de 2023:

---

**Prof. Dr. Sergio Antônio Andrade de Freitas**  
Orientador

---

**Prof. Dr. Ricardo Ajax Dias Kosloski**  
Convidado 1

---

**Profa. Msc. Cristiane Soares Ramos**  
Convidado 2

Brasília, DF  
2022



# Agradecimentos

Agradeço a minha família por todo apoio e toda motivação durante todoo meu processo de formação.

Aos meus amigos e amigas, em especial Gabriel Álves, Flávio Almeida, Érico Bandeira, Rafael Teodosio, Thais Gomes, Guilherme Machado e Leonardo Machado, e a minha namorada Letícia Ribeiro por me ajudarem e me alegrarem em meus momentos de desmotivação.

Ao meu chefe Murillo Marques pelo apoio e compreensão.

Ao meu orientador Sergio Antônio Andrade de Freitas por me auxiliar na realização desse trabalho, sempre me auxiliar e se importando com minha educação e saúde.

Ao professor Edson Alves da Costa Júnior por me motivar desde o começo da minha jornada na faculdade a ser a melhor versão de mim mesmo e lutar para alcançar meus objetivos de vida.





# Resumo

O objetivo deste trabalho é projetar e implementar um sistema para análise de métricas de indicadores de aprendizagem referente ao programa Aprendizagem para o Terceiro Milênio (A3M). O sistema é a porta de entrada para todos os usuários que desejam usar essas métricas de indicadores. Destaca-se que este é um sistema *web*. Para o desenvolvimento desse sistema foi realizada uma revisão bibliográfica referente aos conceitos de engenharia de *software*, em especial as metodologias para desenvolvimento, bem como a execução de atividades para elicitação de requisitos, e desenvolvimento e priorização dos mesmos, assim como a modelagem das necessidades para o desenvolvimento do sistema. Na primeira fase foi realizado o planejamento de todas as ações necessárias para o projeto do SAMI, em especial modelagem, prototipação e priorização dos requisitos. Com esses dados levantados, foi possível caminhar para a próxima fase, sendo a implementação e testes.

**Palavras-chaves:** Indicadores de aprendizagem. Sistema *web*. Metodologia de desenvolvimento de software.



# Abstract

The objective of this work is to design and implement a system for analyzing learning metrics indicators related to the "Learning for the Third Millennium" (A3M) program. The system is the entry point for all users who wish to use these indicator metrics. It should be noted that this is a web system. A literature review was conducted for the development of this system regarding the concepts of software engineering, especially the development methodologies, as well as the execution of activities to elicit requirements and development and prioritization of these requirements, as well as modeling of the needs for the development of the system. In the first phase, planning was carried out for all the necessary actions for the SAMI project, especially modeling, prototyping, and prioritization of requirements. With the data collected, it was possible to move on to the next phase, which was implementation and testing.

**Key-words:** Learning indicators. Web system. Software development methodologies.



# Lista de ilustrações

Figura 1 – Visão em espiral do processo de engenharia de requisitos (SOMMERVILLE, 2011). . . . .	24
Figura 2 – Processo de elicitação e análise de requisitos (SOMMERVILLE, 2011). . . . .	26
Figura 3 – Página de login com imagem lateral e logotipo original e logotipo da Universidade de Brasília e do A3M em baixo. . . . .	32
Figura 4 – Página de login com imagem lateral e logotipo original e logotipos da Universidade de Brasília e do A3M em cima. . . . .	33
Figura 5 – Página de login com logotipo original centralizada e logotipos da Universidade de Brasília e do A3M em baixo. . . . .	34
Figura 6 – Página de login com logotipo original centralizada e logotipos da Universidade de Brasília e do A3M em cima. . . . .	35
Figura 7 – Página de login com logotipo original centralizada sem descrição e logotipos da Universidade de Brasília e do A3M em cima. . . . .	36
Figura 8 – Página de login com imagem lateral e logotipo da Universidade de Brasília e do A3M. . . . .	37
Figura 9 – Página principal de formulários em cartões. . . . .	38
Figura 10 – Página principal de formulários em cartões com menus abertos. . . . .	39
Figura 11 – Página principal de formulários em lista. . . . .	40
Figura 12 – Página de edição de formulários com menu da questão aberto. . . . .	41
Figura 13 – Página de edição de formulários com menu de adição de questão aberto. . . . .	42
Figura 14 – Página de edição de indicador. . . . .	43
Figura 15 – Página de edição de indicador com menus abertos. . . . .	44
Figura 16 – Modal de adição de aplicação à formulário do indicador. . . . .	45
Figura 17 – Página de edição de indicador. . . . .	46
Figura 18 – Página de edição de indicador com menus abertos. . . . .	47
Figura 19 – Modal de adição de aplicação à formulário do indicador. . . . .	48
Figura 20 – Modelo de relação geral. . . . .	58
Figura 21 – Modelo de relação dos usuários. . . . .	59
Figura 22 – Modelo de relação do formulário. . . . .	61
Figura 23 – Modelo de relação do indicador. . . . .	64
Figura 24 – Tela de login. . . . .	73
Figura 25 – Tela de nova senha. . . . .	73
Figura 26 – Tela de lista de formulários. . . . .	74
Figura 27 – Tela de cadastro de professores. . . . .	74
Figura 28 – Tela de lista de professores. . . . .	75
Figura 29 – Tela de cadastro de formulário. . . . .	75

Figura 30 – Tela de visualizar formulário. . . . .	76
Figura 31 – Tela de cadastro de indicador. . . . .	76
Figura 32 – Tela de edição de indicador. . . . .	77
Figura 33 – Tela de detalhes do indicador. . . . .	77
Figura 34 – Tela de cadastro de turma. . . . .	78
Figura 35 – Tela de lista de turma. . . . .	78

# Lista de tabelas

Tabela 1 – Tabela de permissionamento. . . . .	54
Tabela 2 – Tabela de priorização. . . . .	56
Tabela 3 – Dicionário de dados do professor. . . . .	60
Tabela 4 – Dicionário de dados da turma. . . . .	60
Tabela 5 – Dicionário de dados do grupo. . . . .	61
Tabela 6 – Dicionário de dados do formulário. . . . .	62
Tabela 7 – Dicionário de dados da questão. . . . .	62
Tabela 8 – Dicionário de dados do indicador. . . . .	65
Tabela 9 – Dicionário de dados da aplicação. . . . .	65
Tabela 10 – Cronograma de implementação e testes do SAMI. . . . .	70
Tabela 11 – Tabela de requisitos para atender o menor produto viável. . . . .	70
Tabela 12 – Tabela de requisitos implementados de forma adicional. . . . .	71
Tabela 13 – Tabela de requisitos parcialmente atendidos. . . . .	71
Tabela 14 – Tarefas realizadas por <i>sprint</i> . . . . .	72
Tabela 15 – Tabela de requisitos quem podem ser atendidos em melhorias futuras. . . . .	79





# Lista de abreviaturas e siglas

DEG            Decanato de Ensino de Graduação

DPI            Decanato de Pesquisa e Inovação

DPG            Decanato de Pós-Graduação

DEX            Decanato de Extensão



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
	Introdução	19
1.1	Motivação	19
1.2	Objetivo	20
1.3	Metodologia	20
1.4	Estrutura da monografia	20
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>21</b>
2.1	Programa Aprendizagem para o Terceiro Milênio (A3M)	21
2.2	Indicadores de aprendizagem	21
2.3	Requisitos de software	22
2.3.1	Elicitação de requisitos	24
2.4	Modelagem de sistemas	28
2.5	MoSCoW	29
2.6	Scrum	30
<b>3</b>	<b>ELICITAÇÃO DE REQUISITOS</b>	<b>31</b>
3.1	Prototipagem	31
3.2	Entrevista	48
3.3	Requisitos funcionais	50
3.4	Requisitos não funcionais	52
3.5	Permissionamento	53
3.6	Priorização	55
<b>4</b>	<b>MODELO</b>	<b>57</b>
4.1	Visão Geral	57
4.2	Usuários	59
4.3	Formulários	61
4.4	Indicadores	63
<b>5</b>	<b>IMPLEMENTAÇÃO DO SAMI</b>	<b>67</b>
5.1	Histórico da escolha da arquitetura do Software	67
5.2	Arquitetura de Software	67
5.2.1	Tecnologias utilizadas	69
5.3	Hospedagem	69
5.4	Cronograma	69

<b>5.5</b>	<b>Processo de desenvolvimento</b>	<b>70</b>
5.5.1	Definição do menor produto viável	70
5.5.2	Planejamento	71
5.5.3	Versão final do SAMI	72
5.5.4	Telas da versão final	72
5.5.4.1	Tela de login	73
5.5.4.2	Tela de nova senha	73
5.5.4.3	Tela de lista de formulários	74
5.5.4.4	Tela de cadastro de professores	74
5.5.4.5	Tela de lista de professores	74
5.5.4.6	Tela de cadastro de formulário	75
5.5.4.7	Tela de visualizar formulário	75
5.5.4.8	Tela de cadastro de indicador	76
5.5.4.9	Tela de edição de indicador	76
5.5.4.10	Tela de detalhes do indicador	77
5.5.4.11	Tela de cadastro de turma	77
5.5.4.12	Tela de lista de turma	78
<b>6</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>79</b>
<b>6.1</b>	<b>Trabalhos futuros</b>	<b>79</b>
	<b>REFERÊNCIAS</b>	<b>81</b>

# 1 Introdução

O desenvolvimento das tecnologias da informação e comunicação está cada vez mais acelerado e implica em transformações sociais e econômicas refletidas também no cenário educacional. Com as mudanças observadas no processo de ensino e aprendizagem e no perfil do estudante, novas exigências para a educação surgem. (MELO et al., 2017)

Com foco em valorizar e promover ações educacionais inovadoras com ou sem a utilização de tecnologias, nasce o Programa Aprendizagem para o Terceiro Milênio (A3M). Com o apoio dos decanatos acadêmicos (DEG, DPI, DPG e DEX) e da reitoria da Universidade de Brasília o Programa busca atrair pessoas que compartilham da mesma ideia e trabalhar na perspectiva de construção de um portfólio de processos e produtos educacionais inovadores. (MELO et al., 2017)

Na concretização do programa, estão previstos momentos de socialização e divulgação de conhecimentos, tais como seminários, *workshops* e produções científicas (livros, artigos e outros). Entre as ações está o lançamento de editais, no intuito de identificar e promover projetos de apoio ao desenvolvimento de inovações educacionais, e a construção de um portfólio incluindo processos e produtos educacionais inovadores. (MELO et al., 2017)

Para o auxílio na análise das métricas dos indicadores de aprendizagem desses projetos foi idealizado o SAMI (Sistema de Análise de Métricas de indicadores).

## 1.1 Motivação

Este trabalho se propõe a construir um sistema para análise de métricas de indicadores de aprendizagem, que utiliza formulários como forma principal de quantificar os fatores a serem analisados. Esses formulários são utilizados para recolher dados que são contextualizados por indicadores de aprendizagem. Esse sistema é uma ferramenta de apoio ao Programa Aprendizagem para o Terceiro Milênio, a partir dos indicadores que foram propostos para serem utilizados pelos projetos de inovação metodológica definidas através do seu plano de ação. O atual sistema se propõe a automatizar o uso dos indicadores definidos pelo A3M.

Segundo a Profa Msc Cristiane Soares Ramos, em uma entrevista para o Consórcio STHEM Brasil (STHEM, 2021):

*Os indicadores são importantes para ajudar os professores a medir a aprendizagem e quanto esses projetos inovadores estão de fato alcançando o objetivo,*

que é para que alguns possam aprender melhor qual é o efeito dessa inovação sobre o aluno, o quanto ele aprende mais. É para isso que estamos inovando na educação, e os indicadores nos dão essas respostas e servem para apoiar os professores nesse sentido.

## 1.2 Objetivo

O objetivo geral deste trabalho é construir um sistema computacional, via *web*, para análise de métricas de indicadores de aprendizagem.

Os objetivos específicos, que contribuem com o objetivo geral são:

- Desenvolver um sistema de aplicação de formulários
- Desenvolver um sistema capaz de salvar os alunos que podem responder aos formulários
- Desenvolver um componente capaz de conter, gerenciar e representar visualmente os dados dos formulários referentes a um indicador de aprendizagem
- Desenvolver um sistema capaz de prover a um gerenciador da aplicação a oportunidade de registrar *templates* de formulários e novos professores

## 1.3 Metodologia

Para o desenvolvimento do software foi utilizado a metodologia Scrum. Ela foi escolhida pela utilização de *sprints*, que permite um melhor planejamento durante o desenvolvimento, e proximidade com o cliente visando melhorar o resultado final.

## 1.4 Estrutura da monografia

No capítulo 2 é feita a revisão da literatura sobre elicitação de requisitos e modelagem de sistemas. No capítulo 3 são apresentadas as técnicas de elicitação de requisitos utilizadas, além dos requisitos do SAMI, bem como a priorização dos mesmos. No capítulo 4 são apresentados os modelos e as definições das entidades e relacionamentos para o SAMI, e, por fim, no capítulo 5, são apresentadas as ferramentas que serão utilizadas no desenvolvimento do sistema assim como o cronograma de execução do TCC 2.

## 2 Revisão da literatura

Neste capítulo são definidos conceitos necessários para a realização deste trabalho.

### 2.1 Programa Aprendizagem para o Terceiro Milênio (A3M)

No artigo [Melo et al. \(2017\)](#) O Programa Aprendizagem para o Terceiro Milênio (A3M) foi criado com o intuito de enfrentar resistências e obstáculos que surgem com mudanças e inovações, e com o foco em valorizar e promover ações educacionais inovadoras com ou sem a utilização de tecnologias. O programa procura potencializar as iniciativas dos professores que promovam aprimoramento no processo de ensino e na interação com os alunos.

Segundo o artigo [Melo et al. \(2017\)](#):

*Na Universidade de Brasília identifica-se uma série de ações educacionais isoladas as quais certamente poderiam se transformar em metodologias inovadoras e, mais ainda, em produtos institucionais inovadores. O Programa A3M atua junto à instituição na identificação, valorização e promoção dessas ações.*

*As ações inovadoras buscam alinhar-se com as demandas do novo perfil da comunidade acadêmica. Neste contexto, o A3M lança mão de tecnologias, metodologias e novas formas de interação entre os agentes envolvidos no processo de ensino aprendizagem num cenário presente e futuro: o milênio que se inicia. Com o programa A3M, espera-se alcançar inovações na UnB que proporcionem aprimoramentos no processo educacional com reflexos na aprendizagem, satisfação do aluno, satisfação do professor e integração com a sociedade.*

### 2.2 Indicadores de aprendizagem

De um modo geral, o SAMI considera que um indicador é um conjunto de formulários aplicados a uma turma.

Os indicadores de aprendizagem são os que estão sendo utilizados pelo programa Aprendizagem para o Terceiro Milênio, e estão elencados no anexo B do edital [Oliveira \(2022\)](#). São eles:

- **Aquisição de novos conceitos:** identifica tanto o conhecimento prévio dos(as) alunos(as) quanto à formação de novos conceitos. É utilizado ministrando um for-

mulário no início e no fim do semestre ou antes e após a finalização de um módulo do conhecimento novo a ser adquirido.

- **Enriquecimento da rede de conexões:** identifica o enriquecimento dos engramas tornando a memória a longo prazo mais complexa e a capacidade de evocar os conceitos. É aplicado ao utilizar um único termo chave e pedir para o discente listar toda a rede de informações que são evocadas com esse único termo.
- **Motivação do estudante:** mensura e avalia a motivação dos(as) estudantes com respeito a uma disciplina ou curso. É utilizado aplicando um questionário no meio e ao final do semestre em que os estudantes devem indicar o quanto consideram que as afirmativas são verdadeiras.
- **Avaliação entre os pares:** mensura e avalia trabalhos de grupo, colocando o(a) estudante na posição de avaliador(a) do resultado de sua aprendizagem e dos seus pares. É utilizado com uma avaliação técnica, onde um(a) estudante avalia o trabalho entregue por estudantes de uma outra equipe.
- **Utilização de notas em atividades avaliativas:** mensura e avalia diferentes atividades avaliativas por meio de notas ou conceitos atribuídos a essas atividades. É utilizado ao professor definir os critérios de correção de suas atividades avaliativas e informar aos alunos, assim o(a) estudante saberá qual a expectativa do(a) professor(a) antes de iniciar a produção do trabalho.

## 2.3 Requisitos de software

Os requisitos de *software* são as descrições das funções, serviços oferecidos e restrições do sistema. (SOMMERVILLE, 2011) Esses requisitos são o reflexo do que é esperado do sistema para atender a finalidade de sua criação, seja controlar, buscar, criar, documentar, verificar ou modificar algo.

O termo requisito não é usado de forma consistente pela indústria de *software*. Sendo utilizado em alguns casos para definições mais abstratas sobre alguma característica que o sistema deve possuir e em outros casos podendo ser uma descrição detalhada de alguma função do sistema. (SOMMERVILLE, 2011)

Segundo Davis (1993) essa diferença existe pelo motivo de:

*Se uma empresa pretende fechar um contrato para um projeto de desenvolvimento de software de grande porte, deve definir as necessidades de forma abstrata o suficiente para que a solução para essas necessidades não seja predefinida. Os requisitos precisam ser escritos de modo que vários contratantes possam concorrer pelo contrato e oferecer diferentes maneiras de atender às*



*necessidades da organização do cliente. Logo que o contrato tenha sido adjudicado, o contratante deve escrever para o cliente uma definição mais detalhada do sistema, para este entender e possa validar o que o software fará. Ambos os documentos podem ser chamados documentos de requisitos para o sistema.*

Para [Sommerville \(2011\)](#) as classificações dos requisitos são frequentemente dadas por:

Requisitos Funcionais - São declarações dos serviços fornecidos pelo sistema, como o sistema deve reagir a certas entradas e como o sistema deve se comportar em certas situações, além de também poder definir coisas que o sistema não deve fazer.

Requisitos Não Funcionais - Define restrições aos serviços e funções fornecidos pelo sistema, e também podem definir restrições que se apliquem ao sistema na totalidade.

Já para [Valente \(2020\)](#), as classificações são um pouco mais diretas:

Requisitos Funcionais - Descreve o que um sistema deve fazer

Requisitos Não Funcionais - Descreve as restrições a que o sistema está submetido

Sommerville também faz outra distinção entre os requisitos definindo-os em:

Requisitos de usuário - descrições, normalmente em linguagem natural com diagramas, de quais serviços e funções serão oferecidos aos usuários pelo sistema e suas restrições.

Requisitos de sistema - descrições, normalmente em mais detalhes, das funções, serviços e restrições operacionais do sistema, definindo exatamente o que deve ser implementado.

Os requisitos funcionais são normalmente descritos de forma abstrata para melhor entendimento dos usuários do sistema quando descritos como requisitos de usuário. Já requisitos funcionais descritos como de sistema, possuem uma descrição mais detalhada das funções e serviços como entradas e saídas, exceções, etc. ([SOMMERVILLE, 2011](#))

Os requisitos não funcionais devem, sempre que possível, ser descritos de forma quantitativa, para poderem ser testados com mais facilidade. Algumas metas podem ser difíceis de chegar a uma definição que possa ser quantificada, como manutenibilidade. Na prática, clientes de um sistema normalmente consideram difícil mensurar tais requisitos. Em alguns casos quantificar um requisito não funcional pode acarretar custos adicionais, que os clientes que pagam pelo sistema podem não achar justificados. ([SOMMERVILLE, 2011](#))

### 2.3.1 Elicitação de requisitos

Os requisitos definidos acima, são levantados durante a elicitação de requisitos. Nessa atividade, os engenheiros de *software* em conjunto com os usuários finais e clientes do sistema trabalham em conjunto de modo a obter informações sobre as funções, serviços e restrições do sistema, assim gerando os requisitos. (SOMMERVILLE, 2011)

Na figura 1 é possível visualizar uma visão em espiral do processo de engenharia de requisitos.

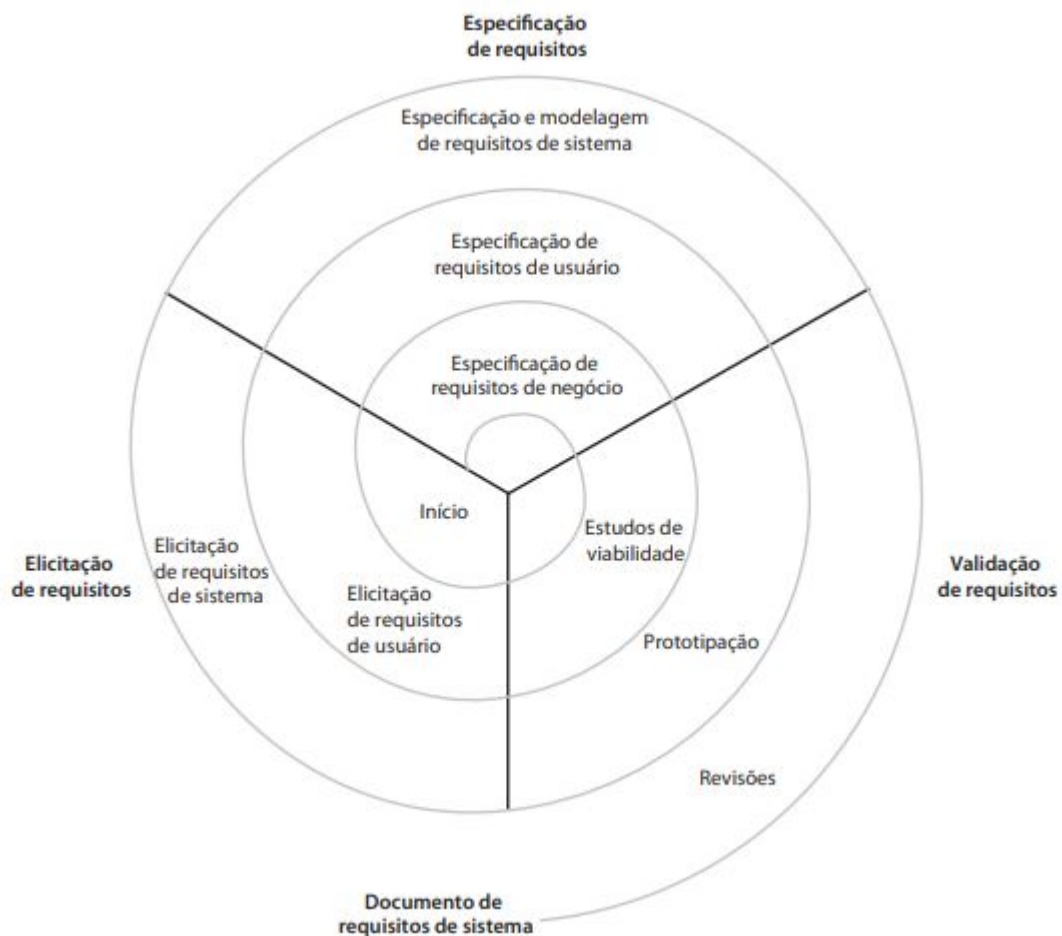


Figura 1 – Visão em espiral do processo de engenharia de requisitos (SOMMERVILLE, 2011).

Uma definição necessária para o entendimento do processo de elicitação de requisitos, é a do que é um *stakeholder*. Para Sommerville, *stakeholders* incluem os usuários finais do sistema e qualquer outra pessoa em uma organização que será afetada por ele. Também pode incluir engenheiros que estão desenvolvendo ou mantendo outros sistemas relacionados a este, como também gerentes de negócios, especialistas de domínio e representantes sindicais.

Na figura 2 é possível visualizar um modelo de processo de elicitação de requisitos fornecido por [Sommerville \(2011\)](#). Esse modelo pode sofrer modificações de acordo com características da organização que está o aplicando.



Figura 2 – Processo de elicitação e análise de requisitos (SOMMERVILLE, 2011).

As atividades do processo exemplificado na figura acima são descritas por Sommerville como:

1. Descoberta de requisitos - Essa atividade de interação com os *stakeholders* serve para descobrir os requisitos do sistema.
2. Classificação e organização de requisitos - Essa atividade consiste em agrupar os requisitos gerados, geralmente utilizando um modelo de arquitetura de sistema para identificar subsistemas e associar cada requisito a um deles.
3. Priorização e negociação de requisitos - Essa atividade está ligada a priorização dos requisitos e resolução de conflitos, geralmente gerados pelas divergências de opiniões dos *stakeholders* envolvidos, através da negociação.
4. Especificação de requisitos - Essa atividade consiste na documentação dos requisitos para poder avançar no processo geral.

O processo de elicitação de requisitos com os *stakeholders* pode ser complicado por várias razões segundo Sommerville:

- Exceto em termos gerais, os *stakeholders* costumam não saberem o que querem de um sistema computacional.
- Os *stakeholders* comumente expressam requisitos utilizando seus próprios termos e experiências, o que pode dificultar o entendimento dos requisitos por parte dos engenheiros de requisitos envolvidos.
- *Stakeholders* diferentes possuem requisitos diferentes e expressam cada um deles de formas diferentes, fazendo com que os engenheiros de requisitos envolvidos tenham que descobrir todas as potenciais fontes, suas semelhanças e conflitos.

- Fatores políticos podem influenciar os requisitos do sistema. Para alguns *stakeholders* requisitos específicos podem aumentar ou diminuir suas influências na organização.
- O ambiente econômico e empresarial onde a análise ocorre é dinâmico. Isso quer dizer que a importância assim como a viabilidade dos requisitos podem mudar, além da possibilidade de criação de novos requisitos a partir de *stakeholders* não consultados.

Para auxiliar os engenheiros de requisitos a lidar com as dificuldades desse processo, algumas técnicas de elicitação de requisitos surgiram temporalmente.

De acordo com Prado (2021) às dez técnicas mais utilizadas são:

- Entrevista - Entrevistas podem ser feitas um a um ou com um grupo pequeno de *stakeholders*. É uma maneira efetiva de elicitar requisitos, pois permite que não seja necessário discutir sobre todos os requisitos elicitados, e sim apenas os relacionados àquele *stakeholder*. (WIEGERS; BEATTY, 2013)
- Questionário - São normalmente utilizados para elicitação de requisitos em grandes grupos ou grupos distribuídos de modo a determinar o que é necessário a cada um. (WIEGERS; BEATTY, 2013)
- Prototipação - Em casos de incertezas sobre alguns requisitos, podem ser construídos protótipos, sendo uma representação possível ou preliminar da aplicação, para tornar alguns conceitos e possibilidades mais tangíveis.(WIEGERS; BEATTY, 2013)
- Caso de uso - Representa um possível uso do sistema por um ator usando algum serviço ou função. O caso de uso narra a interação entre o sistema e os atores envolvidos. (PRADO, 2021)
- *Brainstorming* - É uma reunião onde cada participante pode expressar livremente os requisitos do sistema. (PRADO, 2021)
- Cenários - Representam as interações dos usuários com o sistema. Permitem descrições do processo atual e futuro, necessário para o desenvolvimento do projeto de *software*. (PRADO, 2021)
- *Feedback* - Permite que o usuário final comunique problemas, necessidades e opções ao usar um produto de *software*(ORIOLE et al., 2018)
- *Workshop* - É uma espécie de seminário ou grupo de discussão em que um palestrante apresenta um conteúdo. (PRADO, 2021) específico e incentiva a reflexão no grupo

- Grupo focal - É uma discussão objetiva que introduz um tópico a um grupo de participantes e direciona sua discussão sobre o tema, de uma maneira não-estruturada. (PRADO, 2021)
- Histórias de usuários - Refletem as características que devem ser incluídas no sistema. É realizada em conjunto com os *stakeholders* no início do projeto. (SOMMERVILLE, 2011)

## 2.4 Modelagem de sistemas

Modelagem de sistema é o desenvolvimento de um modelo representado com algum tipo de notação gráfica, em que a mais comum é a notação UML (linguagem de modelagem unificada, do inglês *Unified Modeling Language*), baseado em uma visão ou perspectiva do sistema. (SOMMERVILLE, 2011)

Sommerville também afirma ter uma relação direta da modelagem com os requisitos:

*Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam; e, após isso, são usados para documentar a estrutura e a operação do sistema.*

Com essa informação, ele afirma existirem dois modelos do sistema, sendo eles o modelo do sistema existente e o modelo do sistema a ser desenvolvido:

- Modelo do sistema existente - são normalmente usados durante a engenharia de requisitos e ajudam a esclarecer o que o sistema já faz e pode revelar seus pontos fortes e fracos.
- Modelo do novo sistema - São usados para explicar e exemplificar os requisitos propostos aos *stakeholders* do sistema. Também é utilizado para discutir propostas e na documentação do sistema para implementação.

O aspecto mais importante de um modelo de sistema é a objetividade do seu conteúdo, deixando de fora os detalhes. Um modelo é uma representação abstrata do sistema, e não uma representação alternativa. O intuito é apresentar o sistema de forma simples, salientando as características mais importantes.

Para Sommerville (2011), a partir de perspectivas diferentes, podem ser desenvolvidos diversos modelos para representação do sistema:

- Perspectiva externa - é modelado o contexto ou ambiente do sistema.
- Perspectiva de interação - são modeladas as interações do sistema com seu ambiente ou entre os componentes existentes no sistema.
- Perspectiva estrutural - é modelada a organização do sistema ou a estrutura dos dados processados pelo sistema.
- Perspectiva comportamental - é modelado o comportamento do sistema e como ele reage aos eventos.

## 2.5 MoSCoW

Segundo [Oliveira \(2014, p. 1\)](#) na concepção do seu criador MoSCoW é "um *time-boxing*, o prazo é fixado de modo que o foco de execução acontece sobre os requisitos mais importantes e como tal, é visto como um aspecto central do Desenvolvimento Rápido de Aplicativos (RAD) e dos processos de desenvolvimento de *software*, tais como, DSDM e das técnicas de desenvolvimento ágil de *software*."

Como técnica é usada em contextos como análise de negócios e desenvolvimento de *software*. É utilizada em empreendimentos conduzidos por projetos, podendo ser usado como uma maneira de avaliação na forma de critérios de aceitação, e como uma forma de priorização. ([OLIVEIRA, 2014](#))

De acordo com [IIBA \(2011\)](#) esta técnica consiste em 4 diferentes classificações para cada tarefa, sendo elas:

- Deve ter (*Must have*): Essa é a classificação para os requisitos que devem ser atendidos na solução final para que a mesma seja considerada um sucesso.
- Deveria ter (*Should have*): Essa é a classificação para os requisitos possuem alta prioridade que deveriam ser incluídos na solução, caso possível.
- Poderia ter (*Could have*): Essa é a classificação para requisitos que são considerados desejáveis, mas não necessários, e que serão incluídos caso o tempo e os recursos permitam.
- Não terá (por enquanto) (*Wouldn't have*): Essa é a classificação para requisitos que as partes interessadas concordaram em não implementar em uma determinada entrega, mas que pode ser considerado no futuro.

## 2.6 Scrum

Scrum é uma metodologia ágil com foco na gestão do desenvolvimento iterativo em vez de abordagens técnicas específicas à engenharia de software ágil (SOMMERVILLE, 2010). Ele é dividido em 3 partes:

- Planejamento - onde se estabelecem os objetivos gerais para o projeto e se concebe a arquitetura do software
- *Ciclos ou sprints* - onde cada ciclo desenvolve um incremento do sistema
- Encerramento - onde é completa a documentação necessária, tais como quadros de ajuda do sistema e manuais do utilizador, e avalia as lições aprendidas com o projeto

A característica central e que diferencia o *Scrum* das outras metodologias são as *sprints*. Uma *sprint* é uma unidade de planeamento na qual o trabalho a ser feito é avaliado, as características são selecionadas para desenvolvimento, e essa funcionalidade é implementada. No final de um *sprint*, a funcionalidade completa é entregue aos interessados. (SOMMERVILLE, 2010).

De acordo com (SOMMERVILLE, 2010), as características principais deste processo são as seguintes:

- As *sprints* são de comprimento fixo, normalmente 2-4 semanas.
- O ponto de partida para o planeamento é o *backlog* do produto, que é a lista do trabalho a ser feito no projeto. Durante a fase de avaliação da *sprint*, este é revisto, e são atribuídas prioridades e riscos. O cliente está estreitamente envolvido neste processo e pode introduzir novos requisitos ou tarefas no início de cada *sprint*.
- A fase de seleção envolve toda a equipa de projeto que trabalha com o cliente para selecionar as características e funcionalidades a serem desenvolvidas durante a *sprint*.
- Uma vez que estas sejam acordadas, a equipa organiza-se para desenvolver o software. Pequenas reuniões diárias envolvendo todos os membros da equipe são realizadas para analisar o progresso e, se necessário, redefinir as prioridades do trabalho. Durante esta fase a equipe é isolada do cliente e da organização, sendo todas as comunicações canalizadas através do chamado *Scrum master*. O papel do *Scrum master* é proteger a equipe de desenvolvimento de distrações externas. A forma como o trabalho é feito depende do problema e da equipe.
- No final da *sprint*, o trabalho realizado é revisto e apresentado às partes interessadas.



## 3 Elicitação de requisitos

Na fase de levantamento de requisitos é onde são elicitadas as especificações do produto que será desenvolvido, assim como os critérios para que essas especificações sejam aceitas, sejam essas especificações de requisitos funcionais ou não funcionais. Requisitos não funcionais podem ser considerados os que não tem um impacto direto em como o usuário irá utilizar o sistema mas em como o sistema irá se comportar. Já os requisitos funcionais são referentes a utilização direta do usuário.

Neste tópico são apresentadas as técnicas de elicitação de requisitos utilizadas no trabalho, que foram entrevista e prototipagem.

### 3.1 Prototipagem

A técnica de prototipagem consiste no desenvolvimento de protótipos que são apresentados aos usuários, e com a utilização e visualização destes protótipos, os usuários dão o *feedback* tanto de usabilidade quanto técnicos acerca das experiências obtidas na sua utilização.

Usando a técnica de prototipagem foram criados alguns protótipos de tela e foram apresentados para os usuários. Os protótipos foram desenvolvidos a partir das reuniões realizadas com o cliente, no caso o Prof. Dr. Sergio Antônio Andrade de Freitas, onde foram trocadas informações desde as funcionalidades até os requisitos que já estavam em fase de levantamento.

As figuras de 3 a 8 foram apresentadas para definir como será a página de login do sistema, sendo decidido pela opção representada na figura 3.

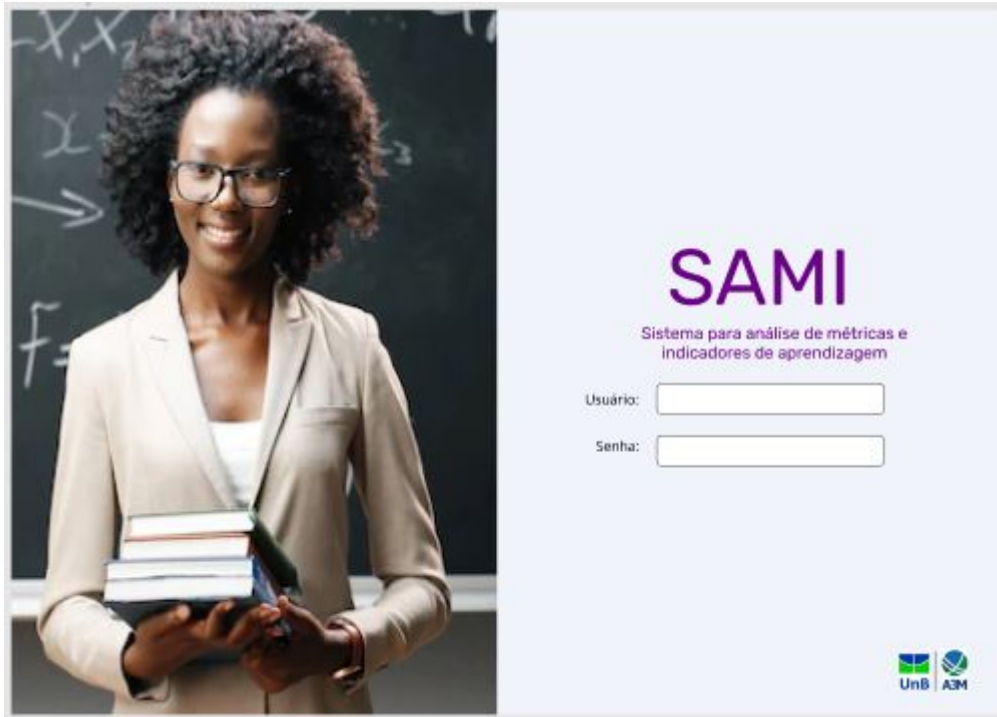


Figura 3 – Página de login com imagem lateral e logotipo original e logotipo da Universidade de Brasília e do A3M em baixo.

A figura 3 representa a opção de tela de login na qual, a mesma é dividida em uma imagem a esquerda, que pode ser definida pelos donos do produto, e na outra metade os campos referentes ao login do usuário com o logotipo do sistema centralizado e os logotipos da Universidade de Brasília e do A3M no canto inferior direito.

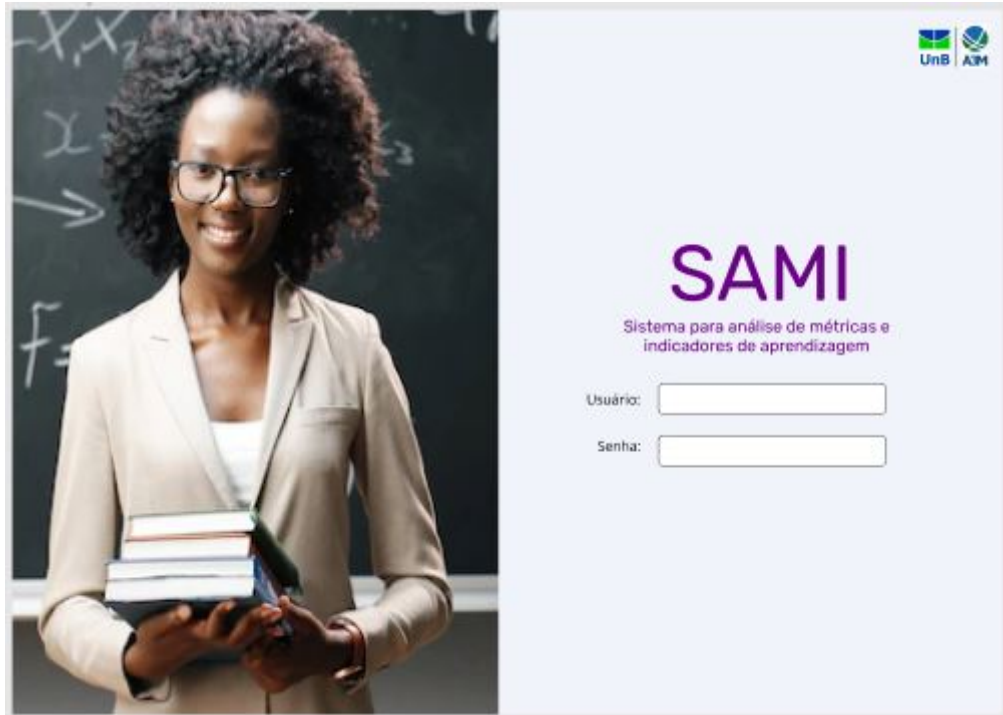


Figura 4 – Página de login com imagem lateral e logotipo original e logotipos da Universidade de Brasília e do A3M em cima.

A figura 4 representa a opção de tela de login na qual a mesma é dividida em uma imagem a esquerda, que pode ser definida pelos donos do produto, e na outra metade os campos referentes ao login do usuário com o logotipo do sistema centralizado e os logotipos da Universidade de Brasília e do A3M no canto superior direito.

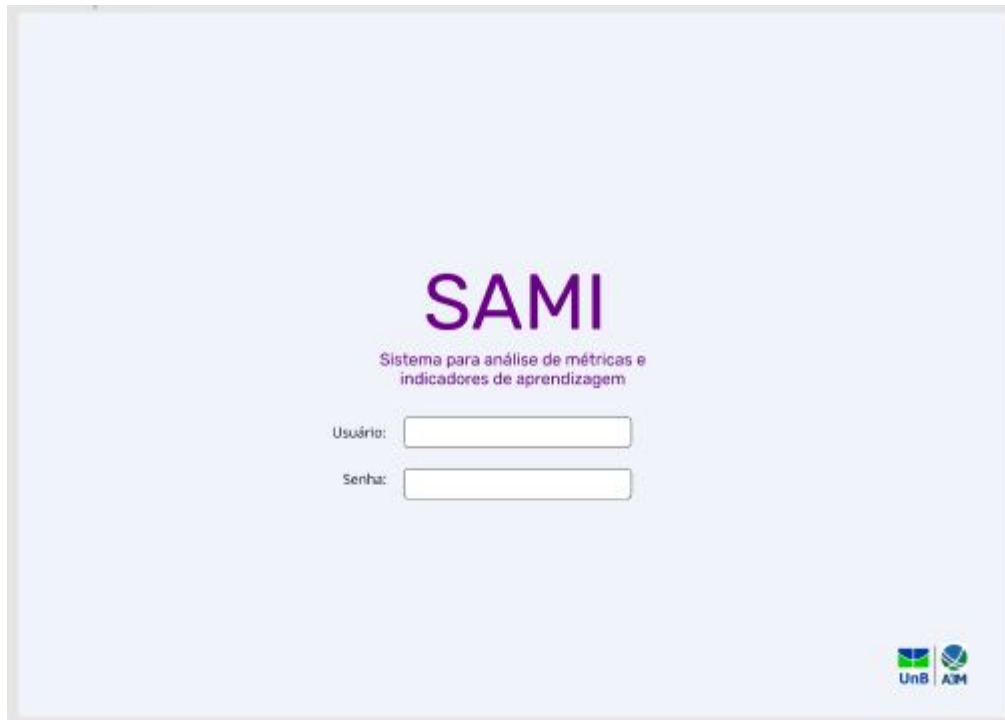


Figura 5 – Página de login com logotipo original centralizada e logotipos da Universidade de Brasília e do A3M em baixo.

A figura 5 representa a opção de tela de login na qual os campos referentes ao login do usuário com o logotipo do sistema estão centralizadas e os logotipos da Universidade de Brasília e do A3M no canto inferior direito.

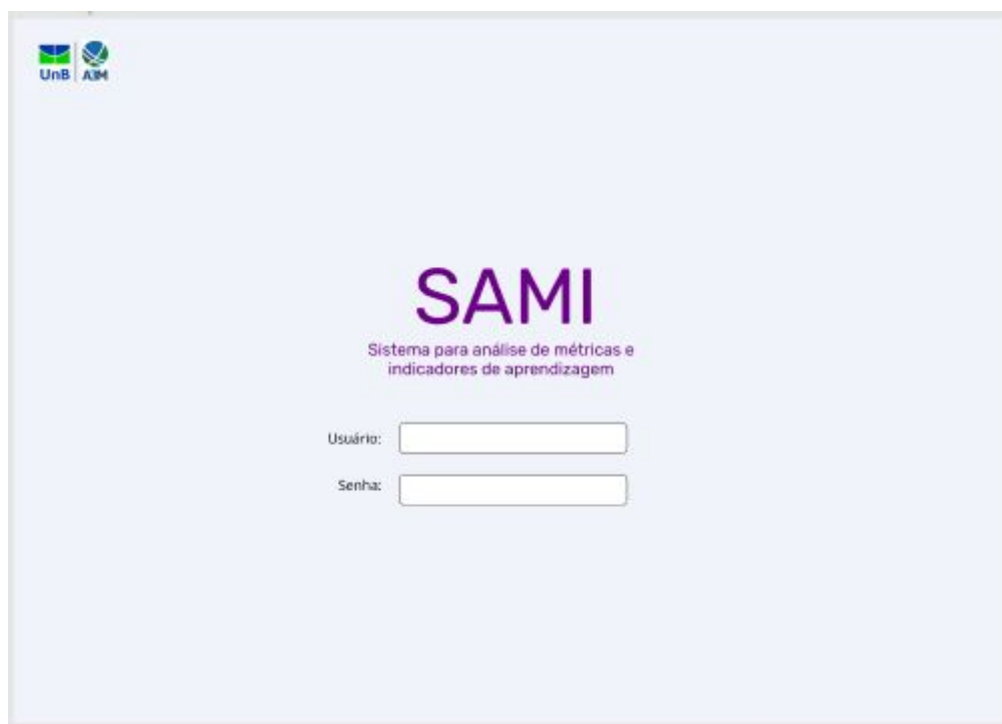


Figura 6 – Página de login com logotipo original centralizada e logotipos da Universidade de Brasília e do A3M em cima.

A figura 6 representa a opção de tela de login na qual os campos referentes ao login do usuário com a logotipo do sistema estão centralizadas e os logotipos da Universidade de Brasília e do A3M no canto superior esquerdo.



Figura 7 – Página de login com logotipo original centralizada sem descrição e logotipos da Universidade de Brasília e do A3M em cima.

A figura 7 representa a opção de tela de login na qual os campos referentes ao login do usuário com a logotipo do sistema sem sua descrição estão centralizadas e os logotipos da Universidade de Brasília em conjunto com da A3M no canto superior direito.



Figura 8 – Página de login com imagem lateral e logotipo da Universidade de Brasília e do A3M.

A figura 8 representa a opção de tela de login na qual a mesma é dividida em uma imagem a esquerda que pode ser definida pelos donos do produto e na outra metade os campos referentes ao login do usuário e os logotipos da Universidade de Brasília e do A3M centralizada.

As figuras de 9 a 13 foram apresentadas para definir como seriam as páginas de gestão de formulários sendo todas aceitas, com definição de que a versão apresentada na figura 11 seria a forma de visualização principal e a versão em cartões representada nas figuras 9 e 10 a forma de visualização secundária.

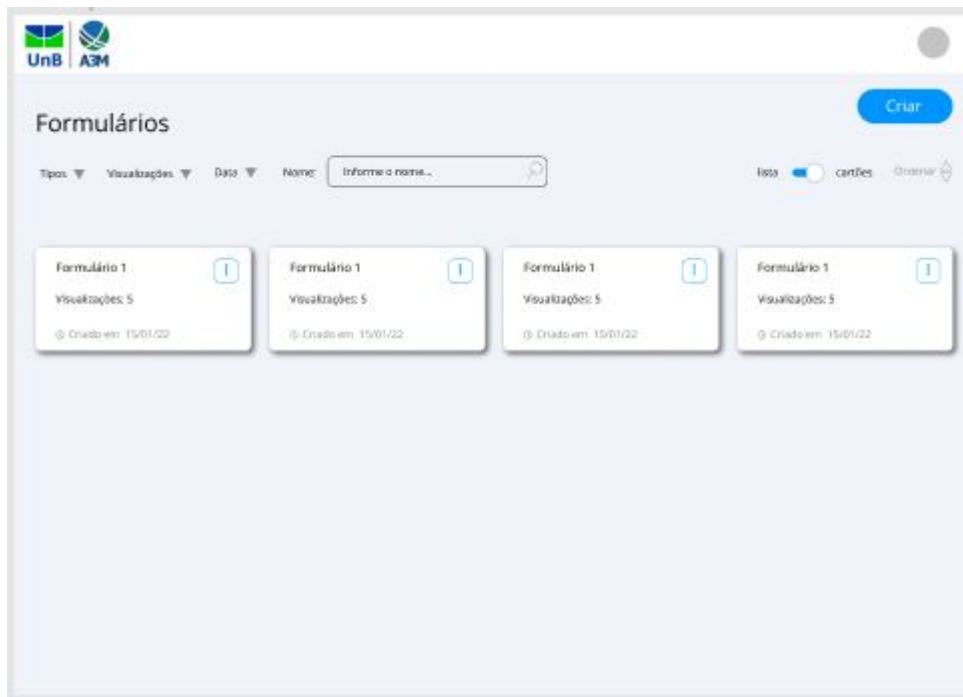


Figura 9 – Página principal de formulários em cartões.

A figura 9 representa a opção de tela de visualização de formulários apresentados em formato de cartões.



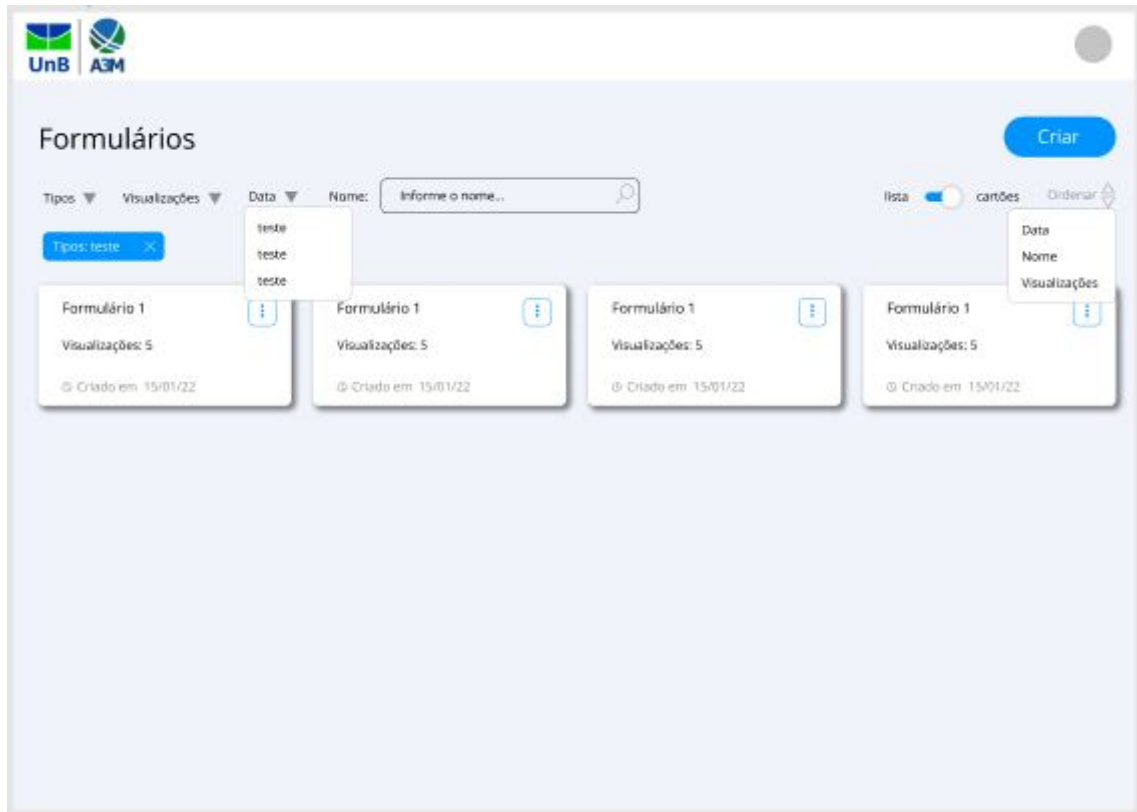
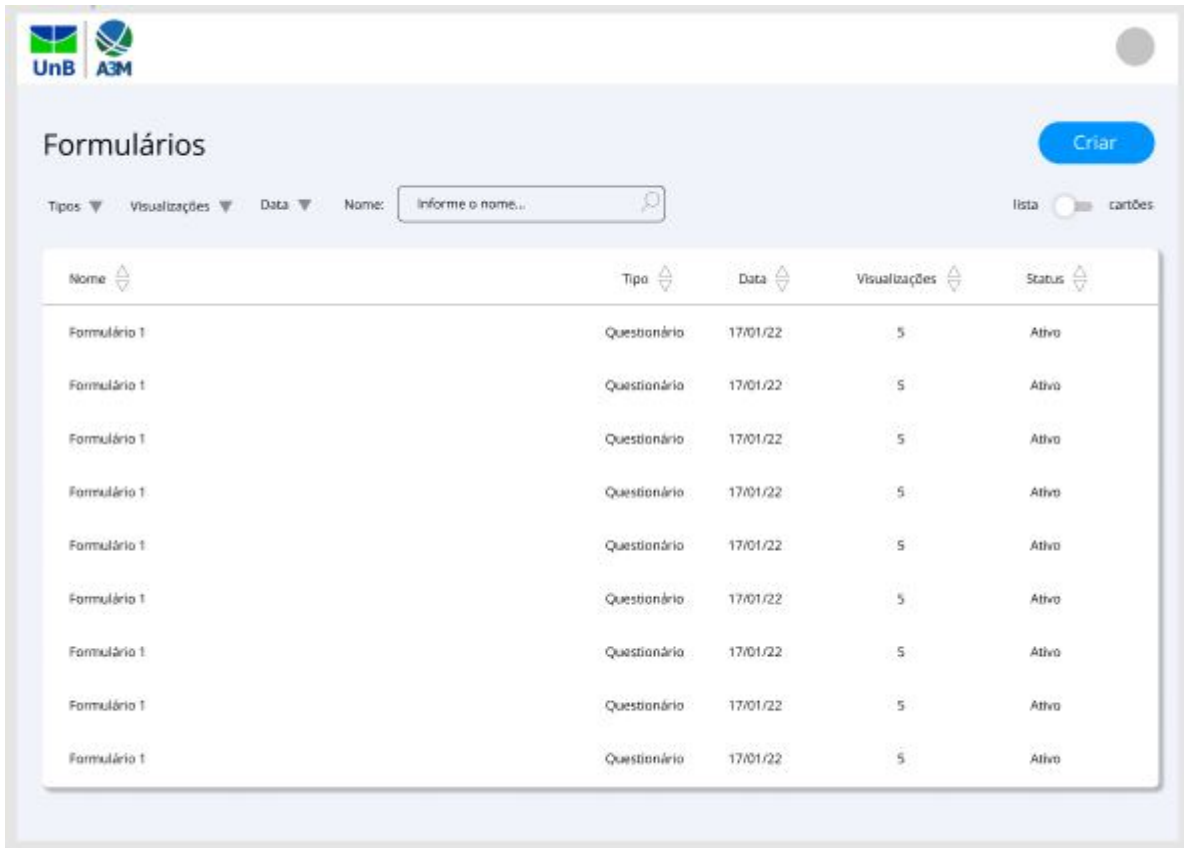


Figura 10 – Página principal de formulários em cartões com menus abertos.

A figura 10 representa a opção de tela de visualização de formulários apresentados em formato de cartões com as opções de seus menus reveladas.



UnB A3M

## Formulários

Tipo Visualizações Data Nome: Informe o nome... lista  cartões

Nome	Tipo	Data	Visualizações	Status
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo
Formulário 1	Questionário	17/01/22	5	Ativo

Figura 11 – Página principal de formulários em lista.

A figura 11 representa a opção de tela de visualização de formulários apresentados em formato de lista.

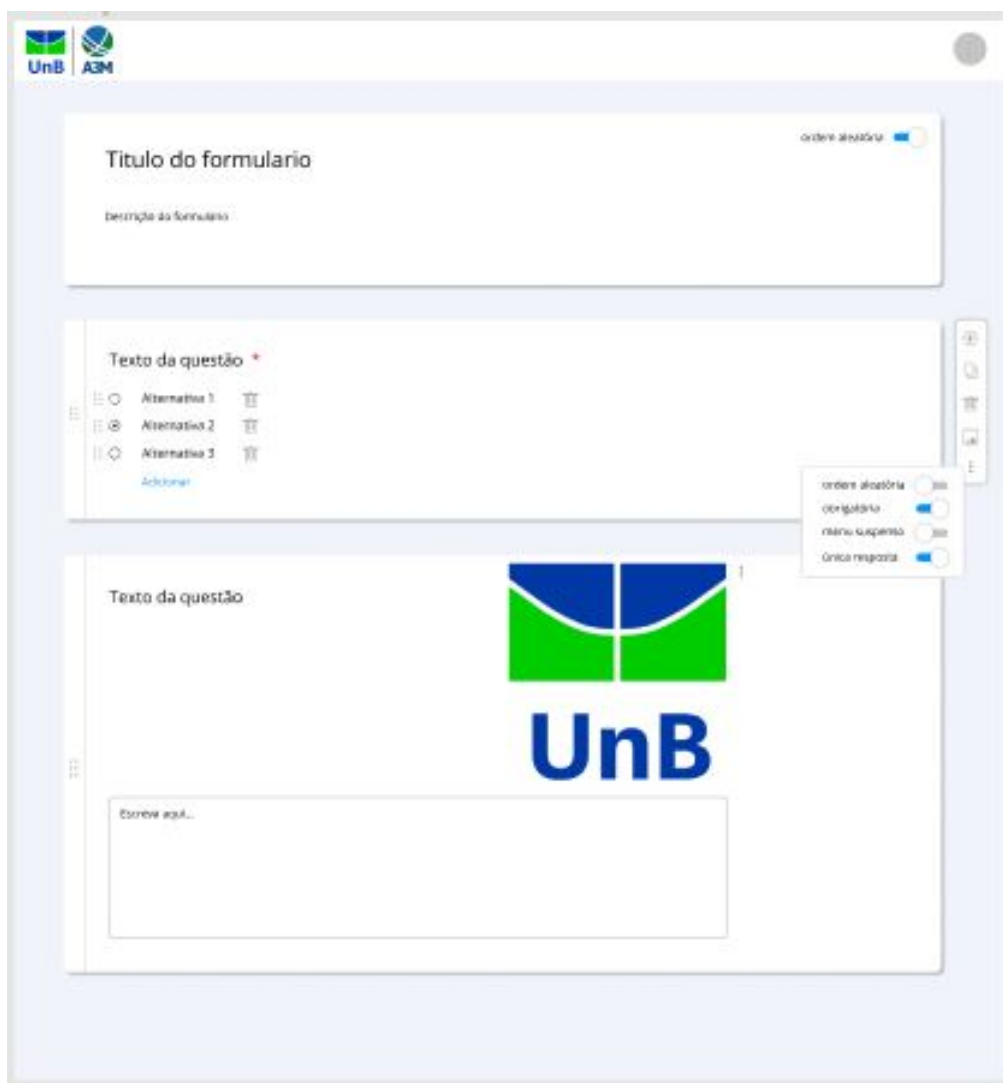


Figura 12 – Página de edição de formulários com menu da questão aberto.

A figura 12 representa a opção de tela de edição de um formulário com menu lateral flutuantes com opções de características da questão reveladas.

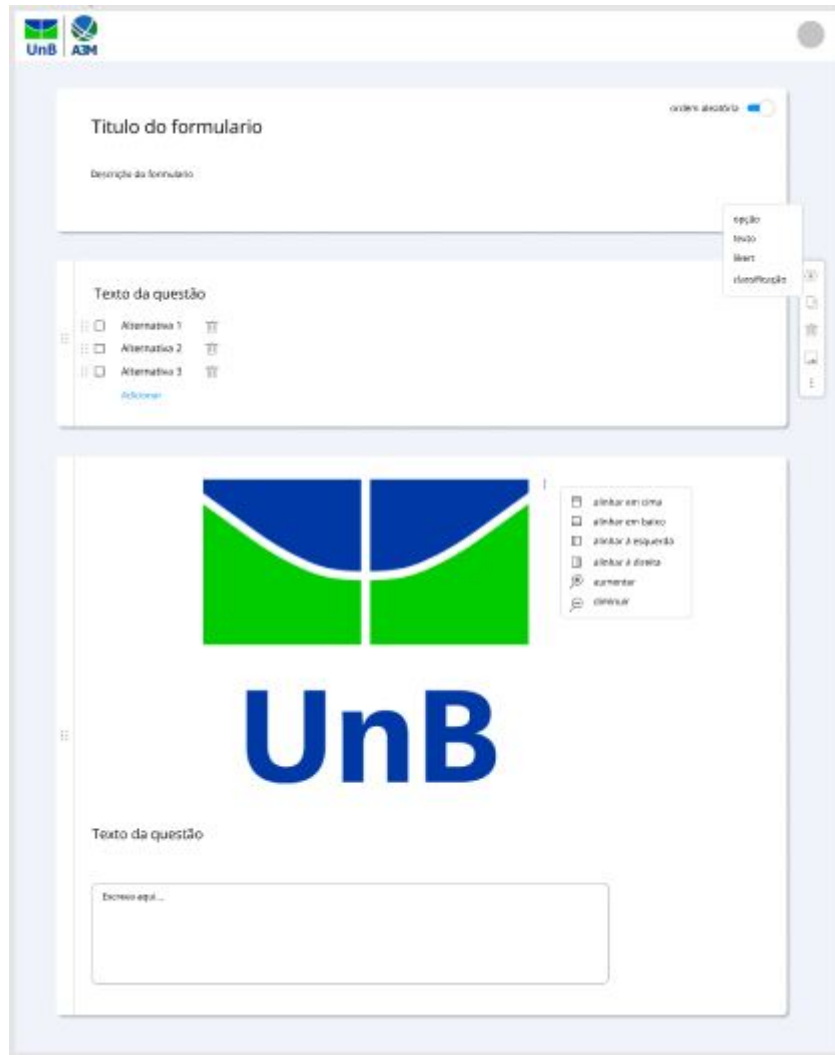


Figura 13 – Página de edição de formulários com menu de adição de questão aberto.

A figura 13 representa a opção de tela de edição de um formulário com menu lateral flutuantes com opções de adição de nova questão reveladas.

As figuras de 14 a 16 foram apresentadas para definir como seriam as páginas de gestão de indicadores sendo todas aceitas.

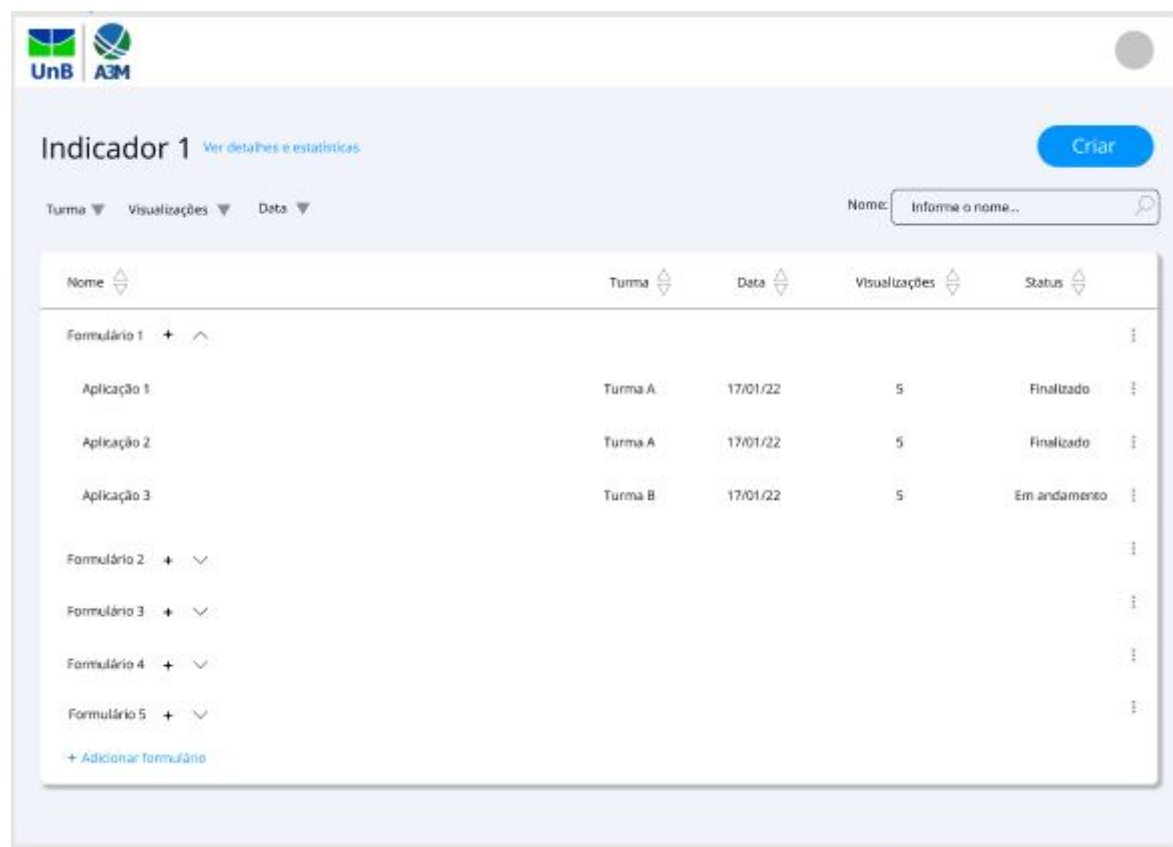


Figura 14 – Página de edição de indicador.

A figura 14 representa a opção de tela de visualização de um indicador com os seus formulários apresentados em formato de lista.

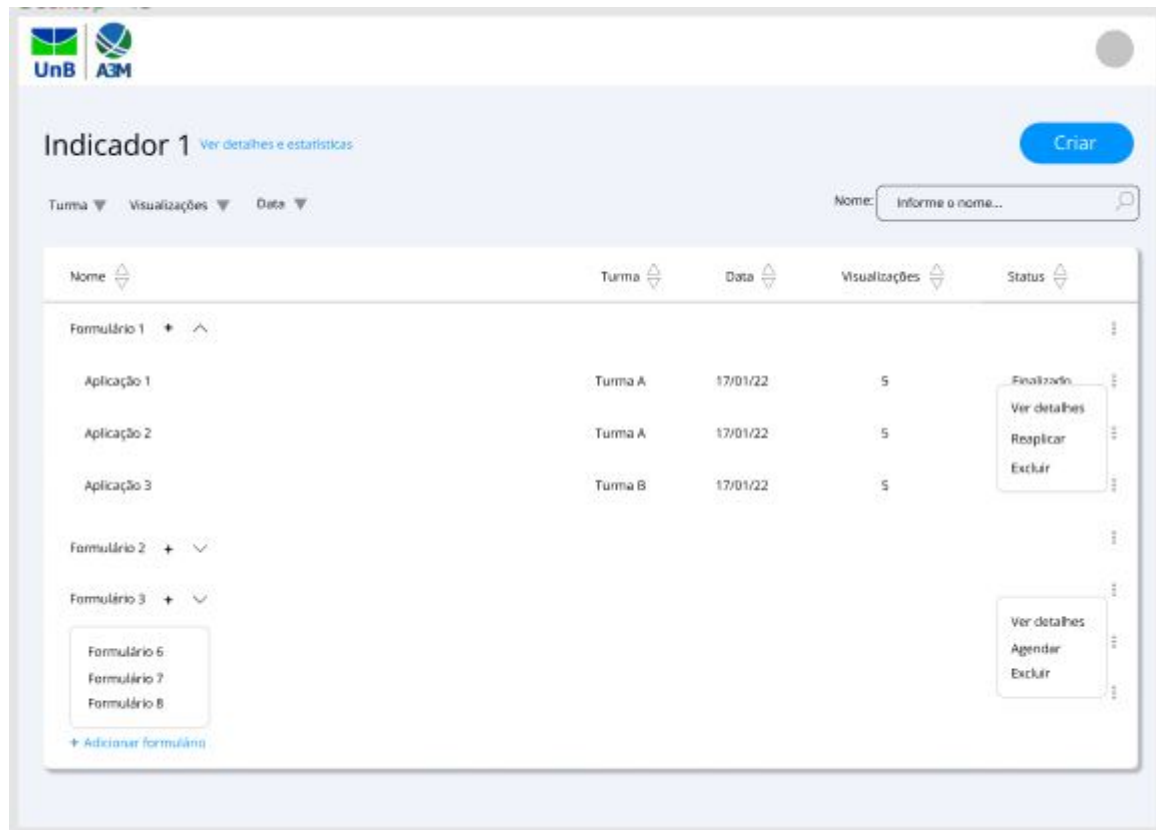


Figura 15 – Página de edição de indicador com menus abertos.

A figura 15 representa a opção de tela de visualização de um indicador com os seus formulários apresentados em formato de lista com suas opções dos menus reveladas.



Figura 16 – Modal de adição de aplicação à formulário do indicador.

A figura 16 representa a modal de adição de uma aplicação a um formulário do indicador.

As figuras de 17 a 19 foram apresentadas para definir como seriam as páginas de gestão de *templates* sendo todas aceitas.

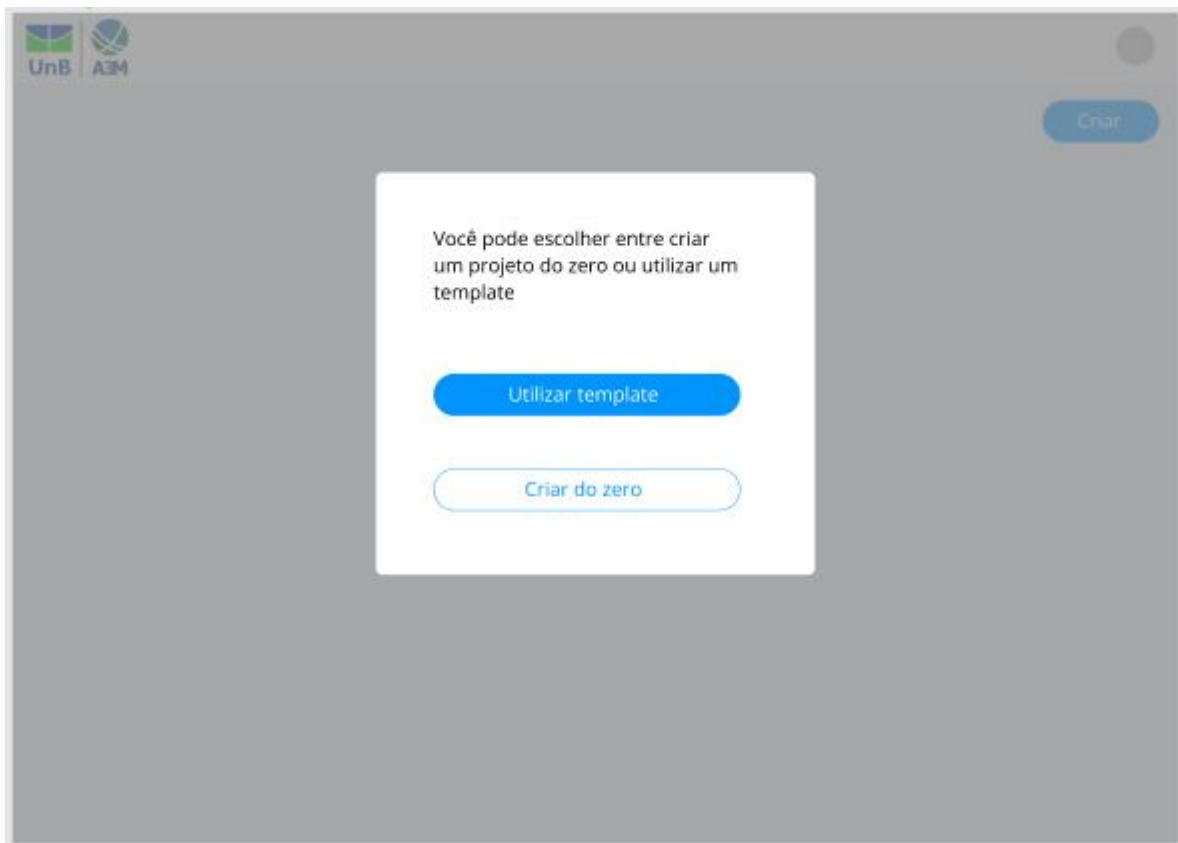


Figura 17 – Página de edição de indicador.

A figura 17 representa a modal que aparece ao clicar na opção de adicionar um formulário, apresentando as opções de criar um do zero ou a partir de um *template*.



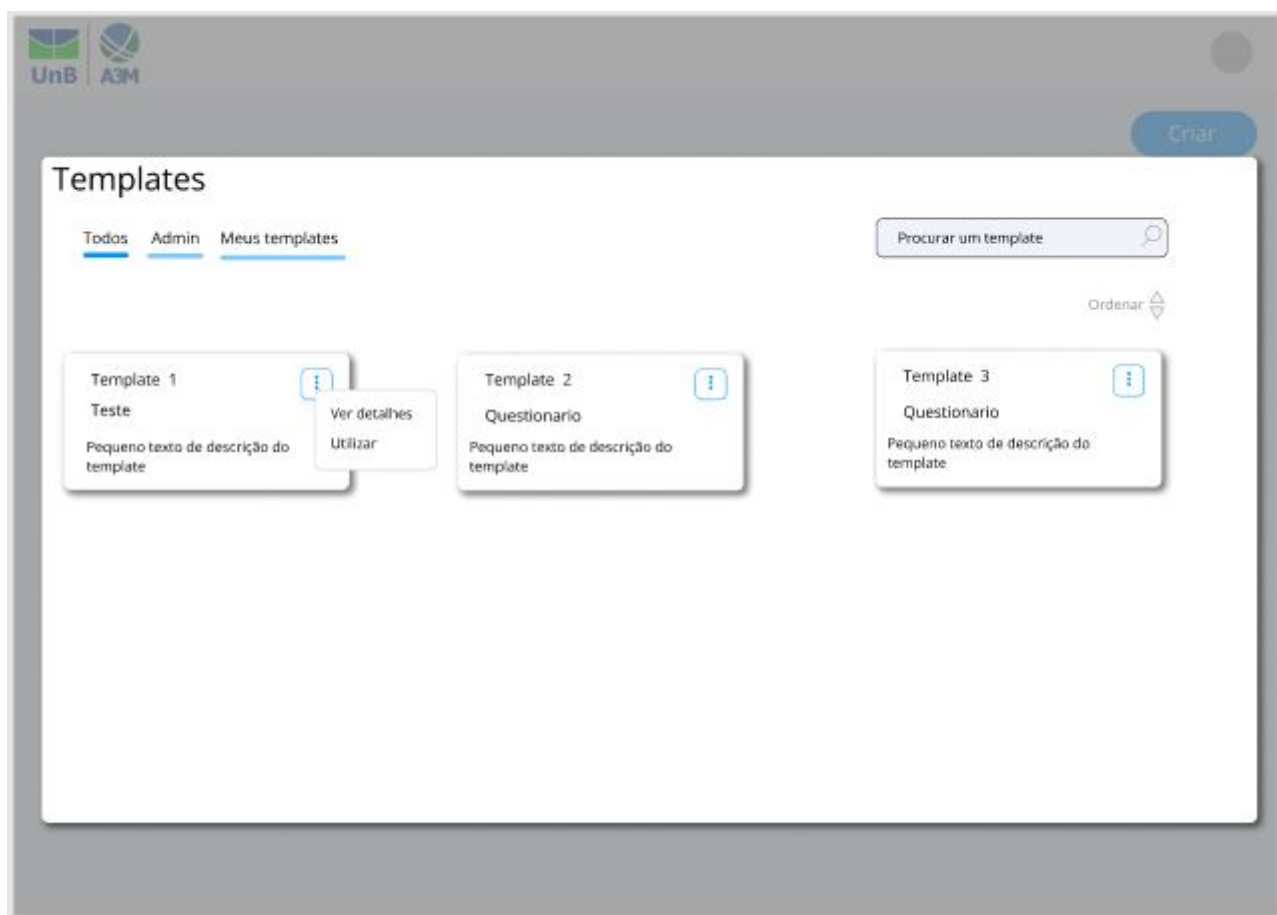


Figura 18 – Página de edição de indicador com menus abertos.

A figura 18 representa a opção de tela de visualização de templates em formato de cartões com as opções de seus menus reveladas.

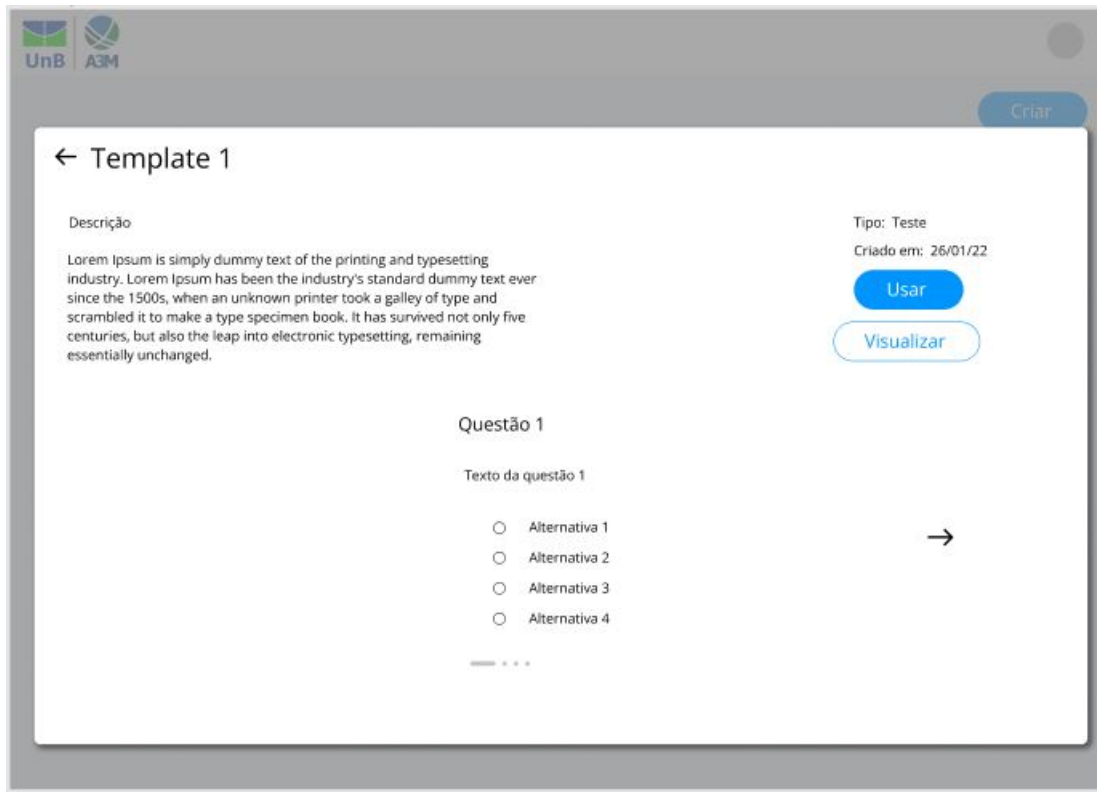


Figura 19 – Modal de adição de aplicação à formulário do indicador.

A figura 19 representa a modal de visualização de um template de formulário, permitindo ao usuário que veja as questões deste template e possa utilizar ou fazer uma visualização de como o aluno veria o mesmo.

## 3.2 Entrevista

A técnica da entrevista consiste em obter dados para o levantamento de requisitos através de perguntas feitas diretamente com o usuário e também de relatos sobre usabilidade que a aplicação deve ter.

Na utilização da técnica da entrevista, futuros usuários foram perguntados sobre os requisitos funcionais da plataforma.

A entrevista aconteceu durante uma reunião realizada remotamente com o Prof. Dr. Sergio Antônio Andrade de Freitas no ano de 2022, no dia 25 de abril.

As perguntas realizadas para entendimento do sistema foram:

Pergunta 1 - Qual é a definição de indicador pro senhor?

Resposta obtida a partir da transcrição da fala do professor Sergio: "Indicador está entre uma métrica e um fator. O que é um fator? Um fator é um elemento conceitual, genérico que você não necessariamente consegue exprimir sobre um formato concreto

ou um formato numérico, por exemplo, fator de aprendizagem. O que é um fator de aprendizagem? Por exemplo, a pessoa está bem de saúde. É um fator. A pessoa está tranquila psicologicamente, bem alimentada. São fatores que influenciam a aprendizagem. Mas como você mede esses fatores? Você não tem como medir um fator. Medir se a pessoa está bem com um número, não tem como. Então você não tem como pegar todos os fatores e transformar em algo que a gente possa quantificar. Então dentre os fatores que podemos encaixar que influenciam a aprendizagem, nós podemos elencar alguns deles que digamos, conduzem a elementos quantificáveis. Esses elementos quantificáveis nós chamamos de indicadores.

Então os indicadores são parte desses elementos de aprendizagem que a gente consegue quantificar. Então como podemos quantificar? Por exemplo, eu consigo quantificar sua auto aprendizagem num determinado conteúdo. Mas como? Eu faço um questionário e posso perguntar pra você por exemplo o quanto você aprendeu fazendo o TCC 1, e coloco uma escala de *likert*, por exemplo, de um a sete, onde um você aprendeu muito pouco e sete aprendeu plenamente. Então você observa que eu consigo com um fator desse quantificar uma condição a partir de um elemento que é subjetivo, que é nesse caso a sua satisfação, sua aprendizagem sobre o TCC 1. Então esse elemento que pode ser quantificável nós chamamos de indicadores. Lembrando que esses indicadores que estamos falando são sempre indicadores de aprendizagem.

Então esse indicador de aprendizagem é um conjunto de elementos que permitem aferir, quantificar, medir um determinado fator de aprendizagem. Que podem ser por exemplo, satisfação, auto aprendizagem, tudo isso. E a partir dessa medida quantificável, obtemos métricas, que são os valores que permitem dar um número para aquele fator. E essas métricas se tornam importantes por, a partir delas, surgir a possibilidade de não só aferir os valores, mas compará-los com valores obtidos de outros indivíduos. E essas comparações podem ser verticais e horizontais. Verticais são para uma comparação por exemplo entre estudantes de uma mesma turma. Já as horizontais são para métricas medidas em vários momentos do mesmo estudante ao longo da disciplina."

Pergunta 2 - O sistema não foi pensado para aplicar provas, certo?

Resposta obtida a partir da transcrição da fala do professor Sergio: "Exatamente. Ele não vai aplicar provas."

Pergunta 3 - Quais são, inicialmente, os usuários do sistema?

Resposta obtida a partir da transcrição da fala do professor Sergio: "Em um primeiro instante a intenção é que possamos fornecer o sistema para cerca de 15 professores, que participam de projetos que o decanato de ensino de graduação vai soltar em um edital referente ao programa Aprendizagem para o terceiro milênio de 2022, pois nesse edital fala que esses professores tem que usar os indicadores que já foram criados para esse

propósito."

Foram realizadas reuniões remotamente com o Prof. Dr. Sergio Antônio Andrade de Freitas no ano de 2022, nos dias 14 e 21 de janeiro, 4, 11 e 18 de fevereiro, 11, 15 e 28 de março, 4, 11, 18, 25 e 28 de abril em que foram tratados aspectos que também corroboraram para o levantamento de requisitos.

### 3.3 Requisitos funcionais

A partir dos resultados da elicitação de requisitos, foram levantados os seguintes requisitos funcionais.

**Login (RF1):** O sistema deve disponibilizar uma tela de login na qual usuário digitará suas credenciais e terá acesso às funcionalidades de acordo com o seu papel.

**Cadastro de professores (RF2):** O sistema deve disponibilizar uma área para que o administrador possa cadastrar novos professores.

**Cadastro de alunos (RF3):** O sistema deve disponibilizar uma área para que o professor possa cadastrar novos alunos.

**Edição de usuário (RF4):** O sistema deve disponibilizar uma área para que o usuário logado modifique seus dados.

**Cadastro de turma (RF5):** O sistema deve disponibilizar uma área para que o professor possa cadastrar uma nova turma.

**Edição de turma (RF6):** O sistema deve disponibilizar uma área para que o professor possa editar uma turma já existente.

**Remoção de turma (RF7):** O sistema deve disponibilizar uma área para que o professor possa remover uma turma já existente.

**Cadastro de grupo (RF8):** O sistema deve disponibilizar uma área para que o professor possa cadastrar um novo grupo.

**Edição de grupo (RF9):** O sistema deve disponibilizar uma área para que o professor possa editar um grupo já existente.

**Remoção de grupo (RF10):** O sistema deve disponibilizar uma área para que o professor possa remover um grupo já existente.

**Criação de formulário (RF11):** O sistema deve disponibilizar uma área para que o professor possa cadastrar um novo formulário.

**Edição de formulário (RF12):** O sistema deve disponibilizar uma área para que o professor possa editar um formulário já existente.

**Remoção de formulário (RF13):** O sistema deve disponibilizar uma área para que o

professor possa remover um formulário já existente.

**Visualização de lista de formulários (RF14):** O sistema deve disponibilizar uma área para que o professor possa visualizar todos os seus formulários cadastrados.

**Adição de questão ao formulário (RF15):** O sistema deve disponibilizar uma área para que o professor possa cadastrar uma nova questão em um formulário já existente.

**Adição de opção à questão (RF16):** O sistema deve disponibilizar uma área para que o professor possa cadastrar uma nova opção à uma questão já existente.

**Adição de template de formulário (RF17):** O sistema deve disponibilizar uma área para que o professor possa cadastrar um novo template de formulário.

**Edição de template de formulário (RF18):** O sistema deve disponibilizar uma área para que o professor possa editar um template de formulário já existente.

**Remoção de template de formulário (RF19):** O sistema deve disponibilizar uma área para que o professor possa remover um template de formulário já existente.

**Adição de template de indicador (RF20):** O sistema deve disponibilizar uma área para que o professor possa cadastrar um novo template de indicador.

**Edição de template de indicador (RF21):** O sistema deve disponibilizar uma área para que o professor possa editar um template de indicador já existente.

**Remoção de template de indicador (RF22):** O sistema deve disponibilizar uma área para que o professor possa remover um template de indicador já existente.

**Criação de indicador (RF23):** O sistema deve disponibilizar uma área para que o professor possa cadastrar um novo indicador.

**Edição de indicador (RF24):** O sistema deve disponibilizar uma área para que o professor possa editar um indicador já existente.

**Remoção de indicador (RF25):** O sistema deve disponibilizar uma área para que o professor possa remover um indicador já existente.

**Visualização de lista de indicadores (RF26):** O sistema deve disponibilizar uma área para que o professor possa visualizar todos os seus indicadores cadastrados.

**Visualização de dados do indicador (RF27):** O sistema deve disponibilizar uma área para que o professor possa visualizar os dados de um indicador a sua escolha.

**Adição de formulário ao indicador (RF28):** O sistema deve disponibilizar uma área para que o professor possa adicionar um de seus formulários a um indicador já existente.

**Agendamento de aplicação de formulário (RF29):** O sistema deve disponibilizar uma área para que o professor possa agendar a aplicação de um formulário de um indicador a uma turma ou grupo.

**Opção para respostas do formulário de forma anônima (RF30):** O sistema deve fornecer a opção de que o formulário a ser aplicado não salve os dados do aluno que o respondeu.

**Visualização dos dados da aplicação do formulário (RF31):** O sistema deve disponibilizar uma área para que o professor visualize os dados referentes a uma aplicação de um formulário.

**Visualização para resposta do aluno ao formulário (RF32):** O sistema deve disponibilizar uma área para que o aluno responda um formulário que está sendo aplicado a ele.

**Visualização de como o formulário será apresentado ao aluno (RF33):** O sistema deve disponibilizar uma área para que o Professor visualize como o aluno verá o formulário.

**Visualização da lista de todos os indicadores (RF34):** O sistema deve disponibilizar uma área para que o administrador visualize todos os indicadores cadastrados no sistema.

**Visualização da lista de todos os formulários pelo administrador (RF35):** O sistema deve disponibilizar uma área para que o administrador visualize todos os formulários cadastrados no sistema.

**Visualização de todos os usuários (RF36):** O sistema deve disponibilizar uma área para que o administrador visualize todos os outros usuários cadastrados no sistema.

**Edição de outros usuários (RF37):** O sistema deve disponibilizar uma área para que o administrador edite um professor cadastrado do sistema.

**Visualização de gráficos com os dados do indicador (RF38):** O sistema deve disponibilizar uma área para que o professor visualize gráficos referentes às respostas dos formulários de um indicador.

## 3.4 Requisitos não funcionais

A partir dos resultados das técnicas de elicitação de requisitos aplicadas foi levantado o seguinte requisito não funcional.

**Integração com serviços da *Microsoft* (RNF1):** O sistema deve ter integração com serviços da *Microsoft* para que seja possível o login automático com a conta da *Microsoft* do usuário.

**Respeitar a Lei Geral de Proteção de Dados Pessoais (RNF2):** o sistema deve respeitar e seguir a Lei Geral de Proteção de Dados Pessoais (LGPD), Lei nº 13.709/2018.

## 3.5 Permissionamento

Os requisitos levantados, possuem restrições de acordo com o usuário que está logado. Essas restrições podem ser visualizadas na tabela 1, onde quando tiver 'Sim' na coluna, significa que aquele usuário pode realizar aquela ação dentro do sistema, e a mesma lógica se aplica para o 'Não'.

Tabela 1 – Tabela de permissionamento.

Requisito	Coordenador	Professor	Aluno
Login (RF1)	Sim	Sim	Sim
Cadastro de professores (RF2)	Sim	Não	Não
Cadastro de alunos (RF3)	Não	Sim	Não
Edição de usuário (RF4)	Sim	Sim	Não
Cadastro de turma (RF5)	Não	Sim	Não
Edição de turma (RF6)	Não	Sim	Não
Remoção de turma (RF7)	Não	Sim	Não
Cadastro de grupo (RF8)	Não	Sim	Não
Edição de grupo (RF9)	Não	Sim	Não
Remoção de grupo (RF10)	Não	Sim	Não
Criação de formulário (RF11)	Não	Sim	Não
Edição de formulário (RF12)	Não	Sim	Não
Remoção de formulário (RF13)	Não	Sim	Não
Visualização de lista de formulários (RF14)	Sim	Sim	Não
Adição de questão ao formulário (RF15)	Não	Sim	Não
Adição de opção à questão (RF16)	Não	Sim	Não
Adição de template de formulário (RF17)	Sim	Sim	Não
Edição de template de formulário (RF18)	Sim	Sim	Não
Remoção de template de formulário (RF19)	Sim	Sim	Não
Adição de template de indicador (RF20)	Sim	Sim	Não
Edição de template de indicador (RF21)	Sim	Sim	Não
Remoção de template de indicador (RF22)	Sim	Sim	Não
Criação de indicador (RF23)	Não	Sim	Não
Edição de indicador (RF24)	Não	Sim	Não
Remoção de indicador (RF25)	Não	Sim	Não
Visualização de lista de indicadores (RF26)	Sim	Sim	Não
Visualização de dados do indicador (RF27)	Não	Sim	Não
Adição de formulário ao indicador (RF28)	Não	Sim	Não
Agendamento de aplicação de formulário (RF29)	Não	Sim	Não
Opção para respostas do formulário de forma anônima (RF30)	Não	Sim	Não
Visualização dos dados da aplicação do formulário (RF31)	Não	Sim	Não
Visualização para resposta do aluno ao formulário (RF32)	Sim	Sim	Sim
Visualização de como o formulário será apresentado ao aluno (RF33)	Sim	Sim	Não
Visualização da lista de todos os indicadores (RF34)	Sim	Não	Não
Visualização da lista de todos os formulários pelo administrador (RF35)	Sim	Não	Não
Visualização de todos os usuários (RF36)	Sim	Não	Não
Edição de outros usuários (RF37)	Sim	Não	Não
Visualização de gráficos com os dados do indicador (RF38)	Não	Sim	Não



## 3.6 Priorização

Os requisitos levantados, possuem prioridades diferentes no que diz respeito a sua importância no sistema, e para definir essas diferentes prioridades, foi utilizada a técnica MoSCoW.

A técnica MoSCoW é uma maneira de classificar e priorizar requisitos para inclusão em um sistema. Este mecanismo é utilizado no desenvolvimento *timebox* no qual o escopo pode precisar ser redefinido de acordo com a evolução de progresso. (OLIVEIRA, 2014)

A tabela 2 apresenta os requisitos classificados de acordo com a técnica MoSCoW.

Tabela 2 – Tabela de priorização.

<b>Requisito</b>	<b>Classificação</b>
Login (RF1)	<i>Must</i>
Cadastro de professores (RF2)	<i>Must</i>
Cadastro de alunos (RF3)	<i>Must</i>
Edição de usuário (RF4)	<i>Must</i>
Cadastro de turma (RF5)	<i>Must</i>
Remoção de turma (RF7)	<i>Must</i>
Criação de formulário (RF11)	<i>Must</i>
Edição de formulário (RF12)	<i>Must</i>
Remoção de formulário (RF13)	<i>Must</i>
Visualização de lista de formulários (RF14)	<i>Must</i>
Adição de questão ao formulário (RF15)	<i>Must</i>
Adição de opção à questão (RF16)	<i>Must</i>
Criação de indicador (RF23)	<i>Must</i>
Edição de indicador (RF24)	<i>Must</i>
Remoção de indicador (RF25)	<i>Must</i>
Visualização de lista de indicadores (RF26)	<i>Must</i>
Visualização de dados do indicador (RF27)	<i>Must</i>
Adição de formulário ao indicador (RF28)	<i>Must</i>
Visualização dos dados da aplicação do formulário (RF31)	<i>Must</i>
Visualização para resposta do aluno ao formulário (RF32)	<i>Must</i>
Respeitar a Lei Geral de Proteção de Dados Pessoais (RNF2)	<i>Must</i>
Edição de turma (RF6)	<i>Should</i>
Cadastro de grupo (RF8)	<i>Should</i>
Edição de grupo (RF9)	<i>Should</i>
Remoção de grupo (RF10)	<i>Should</i>
Agendamento de aplicação de formulário (RF29)	<i>Should</i>
Visualização da lista de todos os indicadores (RF34)	<i>Should</i>
Visualização da lista de todos os formulários pelo administrador (RF35)	<i>Should</i>
Visualização de todos os usuários (RF36)	<i>Should</i>
Edição de outros usuários (RF37)	<i>Should</i>
Visualização de gráficos com os dados do indicador (RF38)	<i>Should</i>
Adição de template de formulário (RF17)	<i>Could</i>
Edição de template de formulário (RF18)	<i>Could</i>
Remoção de template de formulário (RF19)	<i>Could</i>
Adição de template de indicador (RF20)	<i>Could</i>
Edição de template de indicador (RF21)	<i>Could</i>
Remoção de template de indicador (RF22)	<i>Could</i>
Opção para respostas do formulário de forma anônima (RF30)	<i>Could</i>
Visualização de como o formulário será apresentado ao aluno (RF33)	<i>Could</i>
Integração com serviços da Microsoft (RNF1)	<i>Wouldn't</i>

## 4 Modelo

Este capítulo apresenta os modelos e as definições das entidades e relacionamentos que foram obtidos na fase de levantamento de requisitos para o sistema SAMI.

### 4.1 Visão Geral

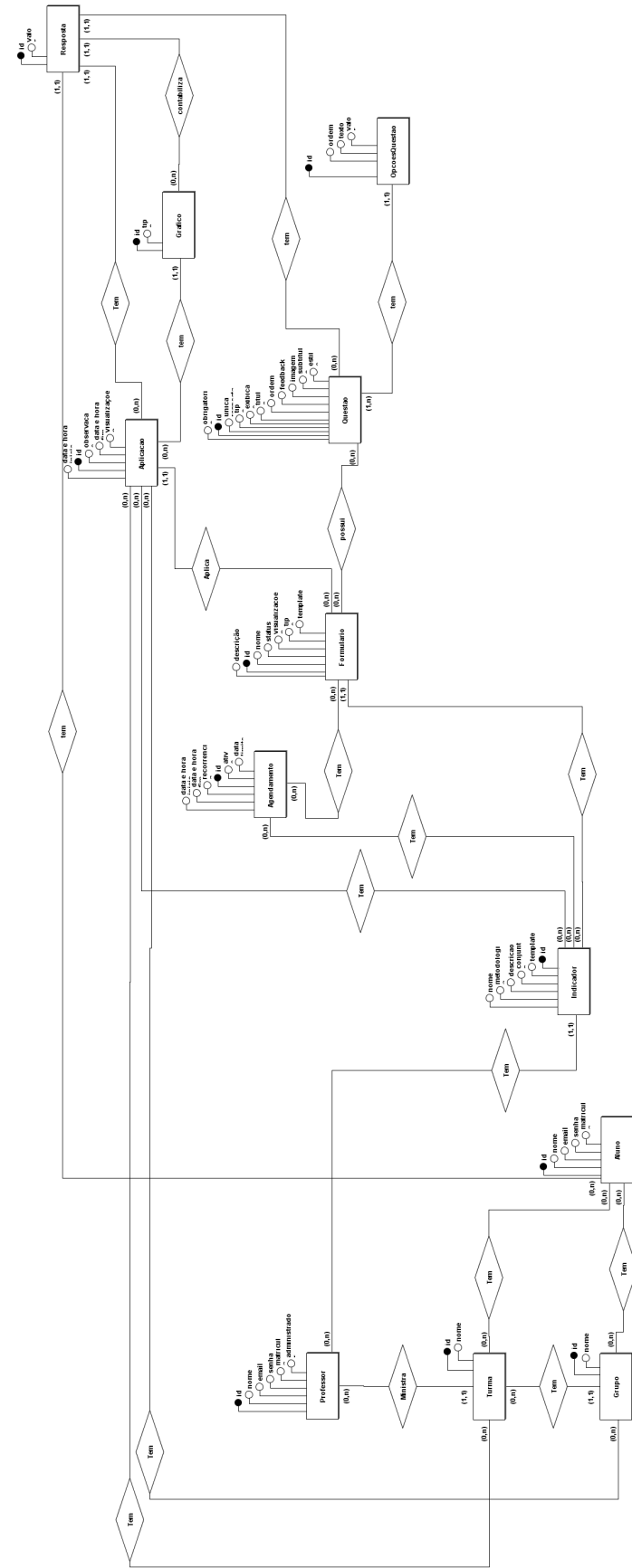


Figura 20 – Modelo de relação geral.

O modelo geral na figura 20, descreve a maneira com que todas as entidades do sistemas estão ligadas. O modelo foi separado em três grupos: usuários, formulário e indicador. Essa separação não reflete com precisão as ligações entre as entidades dentro do sistema, e foi feita para fins representativos e explicativos neste texto.

## 4.2 Usuários

Os usuários são professores e alunos da Universidade de Brasília. Os administradores do sistema serão alguns professores e isso será controlado através de uma variável, e além dessa variável, na representação dos dois não terá nenhuma diferença, portanto ao longo do texto todos os docentes serão chamados apenas de professores.

Após o levantamento de requisitos, foi criado o seguinte modelo de entidade e relacionamento para os usuários:

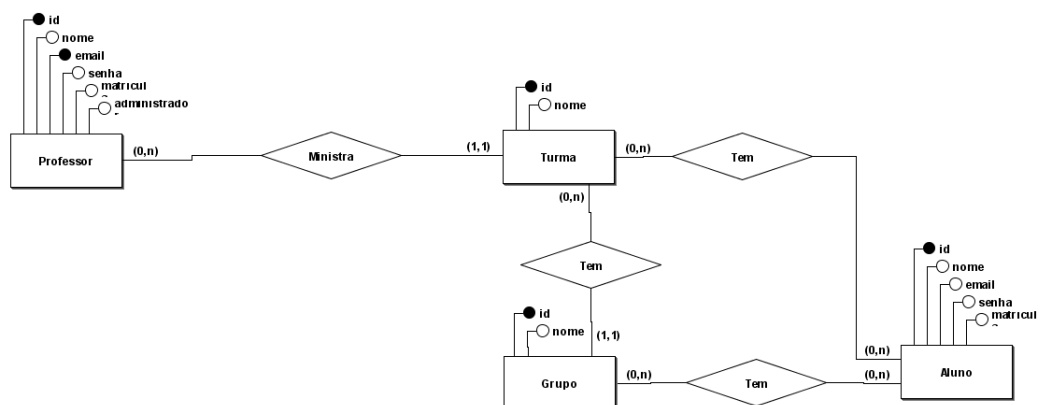


Figura 21 – Modelo de relação dos usuários.

O modelo de usuários na figura 21, descreve a maneira com que a conexão de alunos com professores permite o entendimento dos atributos e permissões necessários à cada um deles, e a forma a qual eles podem ser representados.

O dicionário de dados abaixo explicita os atributos que são atribuídos a um usuário:

Tabela 3 – Dicionário de dados do professor.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador do professor no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Nome	varchar	nome do professor	João Vitor
Email	varchar	email de login do professor	joao@unb.com.br
Senha	varchar	senha de acesso do professor	pass123
Matrícula	varchar	matrícula do professor na UnB	160123456
Token	varchar	variável de controle para acesso do usuário ao sistema	U2FsdGVkX184+bex
Tipo	varchar	variável de controle para identificar o tipo do usuário (professor, aluno ou administrador)	teacher
Novo usuário	booleano	variável de controle para identificar se o usuário é novo (utilizado para primeiro acesso)	true

A ligação entre as entidades professor e aluno acontece na formação de uma turma. A turma é um elemento básico das atividades acadêmicas da graduação na Universidade de Brasília, e representa a materialização da execução de uma disciplina. O dicionário de dados abaixo explicita os atributos que são atribuídos a uma turma:

Tabela 4 – Dicionário de dados da turma.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador da turma no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Nome	varchar	nome da turma como no SIGAA	IHC-01
Nome da disciplina	varchar	nome da disciplina no SIGAA que esta turma faz parte	Introdução humano computador
Código da disciplina	varchar	código da disciplina no SIGAA que esta turma faz parte	FGA-0001
Semestre	varchar	nome do semestre que esta turma faz parte	2022/1

Em termos representativos, uma turma surge para criar uma conexão entre os professores que ministram uma disciplina e alunos que serão matriculados para cursar essa mesma disciplina, e o faz com a definição de que um ou mais professores podem estar em uma ou mais turmas, em que essas turmas podem ter um ou mais alunos.

Dentro da conexão feita da turma com os alunos matriculados na mesma, pode ocorrer uma nova ligação com outra entidade, chamada de grupo, este que é um sub-

conjunto dos alunos matriculados na turma. O dicionário de dados abaixo explicita os atributos que são atribuídos a um grupo:

Tabela 5 – Dicionário de dados do grupo.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador do grupo no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Nome	varchar	nome do grupo	Grupo 2

A existência dessa entidade, que em termos representativos, pertence a uma turma e pode conter um ou mais alunos, não substitui a ligação da turma com os discentes, sendo utilizada para complementar a utilização do sistema, permitindo realizar atividades específicas com grupos selecionados.

### 4.3 Formulários

Os formulários serão a principal forma de coletar dados para a avaliação das métricas e indicadores. Eles são também a principal forma de interação entre os professores e alunos.

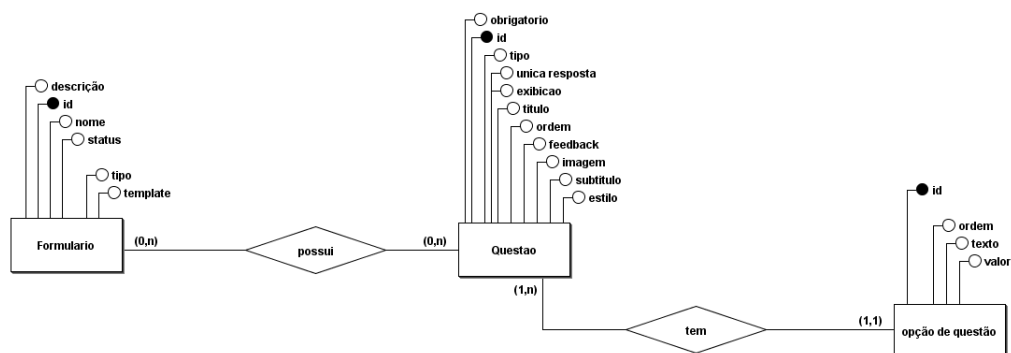


Figura 22 – Modelo de relação do formulário.

O modelo do formulário na figura 22, descreve a maneira com que o mesmo é montado. O dicionário de dados abaixo explicita os atributos que são atribuídos a um formulário:

Tabela 6 – Dicionário de dados do formulário.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador do formulário no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Nome	varchar	nome do formulário criado pelo professor	Form 1
Descrição	varchar	descrição do formulário dada pelo professor	Formulário de acompanhamento
Aleatório	booleano	variável de controle para saber se as questões deste formulário serão exibidas em ordem aleatória	false

A entidade principal que compõe um formulário é a questão. Uma questão para o formulário é um conjunto de dados que definem uma pergunta e como ela é representada. O dicionário de dados abaixo explicita os atributos que são atribuídos a uma questão:

Tabela 7 – Dicionário de dados da questão.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador da questão no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Única resposta	booleano	variável de controle para saber se a questão possui apenas uma resposta	true
Título	varchar	título da questão	como você está hoje?
Subtítulo	varchar	subtítulo da questão	Escolha uma das opções abaixo
Imagem	varchar	url da imagem da questão	<a href="https://imgs.com/questao.png">https://imgs.com/questao.png</a>
Ordem	valor com ponto flutuante	variável para identificar qual a ordem dessa questão em relação às outras	2.0
Tipo	varchar	variável para identificar o tipo da questão	likert
Obrigatória	booleano	variável para identificar se é uma questão obrigatória	false
Aleatória	booleano	variável para identificar as opções dentro da questão serão mostradas em ordem aleatória	true

Em termos representativos, um formulário pode ter zero ou mais questões que por sua vez possui uma ou mais opções de questão.



## 4.4 Indicadores

Os indicadores são as entidades principais do sistema. Com eles será possível comparar dados resultados de formulários, e gerenciar a aplicação dos mesmos, adicionando agendamentos periódicos de aplicações desses formulários e guardar os dados resultantes.

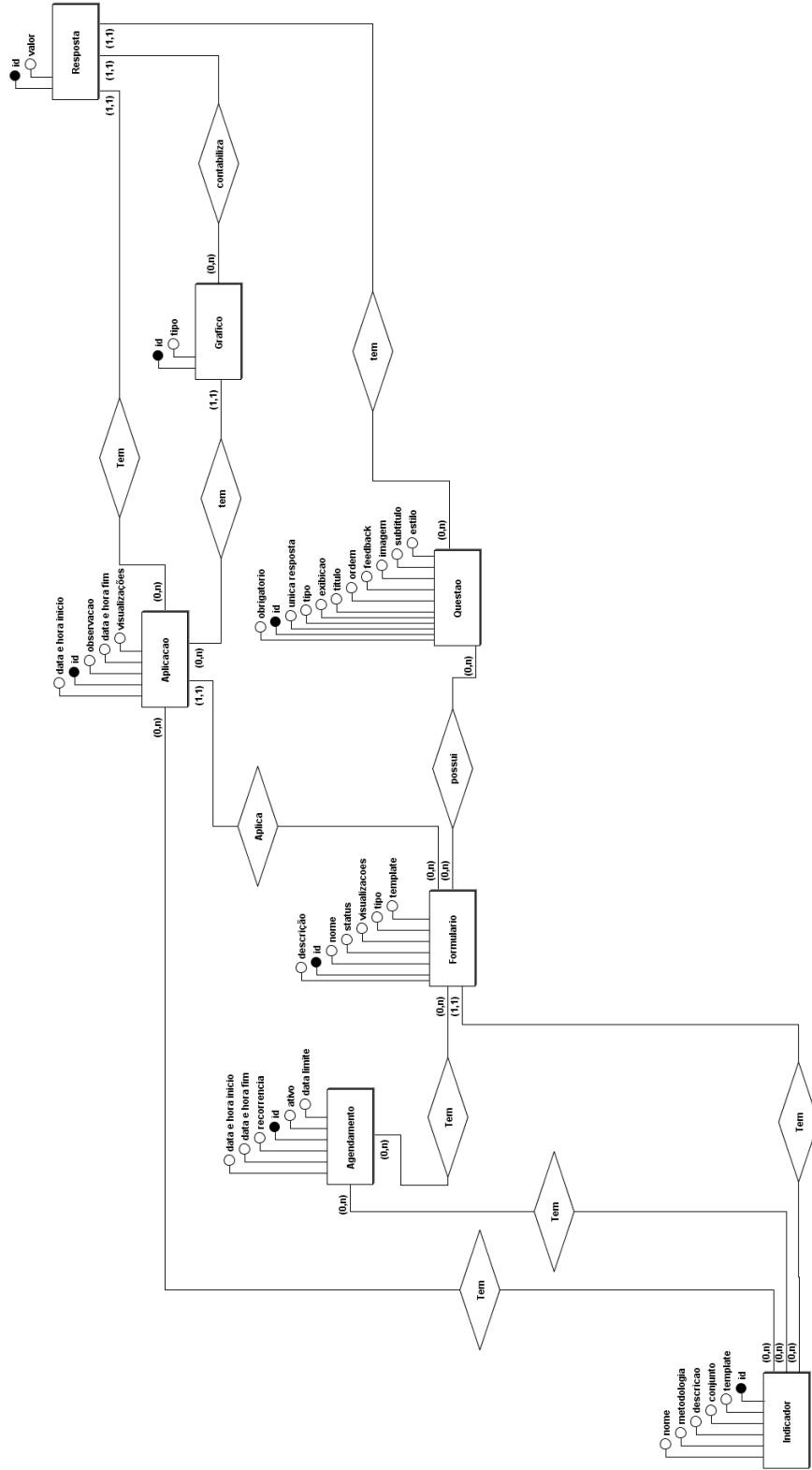


Figura 23 – Modelo de relação do indicador.

O modelo do indicador na figura 23, descreve a maneira com que o mesmo é definido. O dicionário de dados abaixo explicita os atributos que são atribuídos a um indicador:

Tabela 8 – Dicionário de dados do indicador.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador do indicador no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Nome	varchar	nome do indicador criado pelo professor	Ind 1
Descrição	varchar	descrição do indicador dada pelo professor	Indicador de acompanhamento
Metodologia	varchar	metodologia utilizada no indicador	Metodologia 1
Conjunto	varchar	conjunto do indicador dado pelo professor	Conjunto 2

A entidade principal dentro do indicador é a aplicação, que é a aplicação de um formulário a uma turma ou grupo. O dicionário de dados abaixo explicita os atributos que são atribuídos a uma aplicação:

Tabela 9 – Dicionário de dados da aplicação.

Nome	Tipo	Descrição	Exemplo
Id	uuid	identificador da aplicação no sistema	71186d49-819d-4b7b-80a6-b6febf74fdb9
Data e hora de início	data	data e hora em que a aplicação será iniciada	12/12/2022 - 00:00
Data e hora de fim	data	data e hora em que a aplicação será encerrada	13/12/2022 - 23:59
Recorrente	Json	variável para identificar se a aplicação será recorrente em algum dia da semana	segunda: true, terça: true
Deve repetir	Booleano	variável para controle de qual dados serão utilizados na verificação, a repetição por dia ou a data definida	false



## 5 Implementação do SAMI

Neste capítulo são descritas as decisões arquiteturais do sistema, as ferramentas que serão utilizadas para o desenvolvimento do SAMI, a definição do produto final e o cronograma de desenvolvimento do sistema executado no TCC 2.

### 5.1 Histórico da escolha da arquitetura do Software

No início da do processo de desenvolvimento, houve uma tentativa de utilizar a plataforma Power Apps da Microsoft. Essa ferramenta é utilizada para criação de aplicações sem utilização de código, e possui vários *plugins* e integrações com outros serviços da Microsoft.

As vantagens da utilização da ferramenta Power Apps, seria poder utiliza-la em conjunto com outra ferramenta também da Microsoft, o Forms. Essa ferramenta de criação, aplicação de formulários já tem uma usabilidade conhecida e prática, e também a sua utilização traria um ganho de tempo ao desenvolvimento do SAMI, por não ser necessário criar tal tipo de gerenciador de formulários.

Algumas dificuldades encontradas durante a tentativa, foram a não integração da ferramenta Excel nativamente, sendo necessário utilização de *plugins* externos para sua utilização. No entanto foi encontrado o *plugin* PCFexcel que solucionaria esse problema, mas nesse ponto outro problema foi encontrado, a necessidade de permissão de um administrador para utilização do *plugin*. Tal permissão foi solicitada à STI (Secretaria de Tecnologia da Informação) da UnB, e o pedido foi negado.

Uma segunda tentativa de utilizar o Power Apps foi a de criar um sistema e integra-lo à ferramenta, mas essa abordagem também requeria permissão de um administrador.

### 5.2 Arquitetura de Software

O *front-end* é o setor responsável por garantir a aparência e a funcionalidade de um site ou aplicação para os usuários. Ele é responsável por toda a estrutura, design, conteúdo, comportamento, desempenho e capacidade de resposta da interface, tornando possível a interação dos usuários com a aplicação. (MACHADO, 2022)

O *front-end* do SAMI foi construído usando o framework Next.js, que é baseado em React e oferece uma ampla gama de recursos e plugins para acelerar o desenvolvimento. O Next.js fornece uma estrutura abrangente e pronta para uso, o que facilita o início do

projeto. Além disso, ele combina a eficiência do React com recursos adicionais que tornam a construção de aplicações mais rápida e eficiente. (AURELIO, 2020)

A escolha do Next.js para o *front-end* do SAMI foi baseada na sua facilidade de criação de páginas, amplo suporte de bibliotecas e familiaridade de parte da equipe de desenvolvimento com a linguagem de programação JavaScript. Ele permite a criação rápida e eficiente de páginas, graças às suas ferramentas e recursos prontos para uso. Além disso, a ampla base de usuários e a comunidade ativa de desenvolvedores tornam mais fácil encontrar soluções para eventuais problemas. Tudo isso tornou o Next.js a escolha ideal para o desenvolvimento do front-end do SAMI.

O *back-end* é o setor responsável por todos os aspectos do site ou aplicação que estão ocultos aos usuários e não são acessíveis diretamente. Ele se concentra na lógica, dados e regras da aplicação e gerencia as tarefas importantes que ocorrem fora do alcance dos usuários, como processamento de dados, armazenamento de informações, comunicação com bancos de dados e gerenciamento de sessão. Em resumo, o *back-end* é o responsável pelo funcionamento eficiente e seguro do site ou aplicação. (MACHADO, 2022)

No *back-end*, os *endpoints* são rotas específicas que respondem a solicitações do *front-end*. Quando essas rotas são chamadas, são disparadas rotinas que realizam operações envolvendo os dados armazenados na base de dados. Estas operações incluem as já conhecidas operações CRUD (Create, Read, Update e Delete), que permitem criar, ler, atualizar e excluir dados. Os *endpoints* são uma porta de entrada para os dados armazenados no *back-end* e são cruciais para o funcionamento eficiente da aplicação.

O *back-end* do SAMI foi construído com o *framework* Nest.js, que é um *framework* Node.js de destaque por sua capacidade de desenvolver sistemas eficientes e de alta performance usando Javascript ou Typescript. O Nest.js permite a criação de aplicações de *back-end* de forma fácil e rápida, fornecendo uma estrutura completa e recursos robustos para desenvolver soluções avançadas. A escolha pelo Nest.js reflete a necessidade de ter um *back-end* poderoso, capaz de atender às demandas do SAMI e proporcionar uma experiência excelente aos usuários. (WILLIANS, 2022)

A escolha do Nest.js foi motivada pela familiaridade da equipe com o *framework* e pelo suporte integrado à ferramenta Swagger. O Swagger é uma ferramenta poderosa que permite a documentação automatizada dos *endpoints*, tornando mais fácil o entendimento e o uso da API do *back-end*. Além disso, o Nest.js possui uma comunidade ativa e uma ampla gama de recursos e plugins disponíveis, o que torna possível o desenvolvimento de soluções ainda mais avançadas. A combinação dessas vantagens fez com que o Nest.js fosse a escolha ideal para atender às necessidades do *back-end* do SAMI.

### 5.2.1 Tecnologias utilizadas

As tecnologias que foram utilizadas para o desenvolvimento do sistema são:

- Javascript: é uma linguagem de programação que permite adicionar complexidade como controle de mídias, imagens e informações dinâmicas em páginas web e não apenas informações estáticas.
- React: é uma biblioteca JavaScript para criação de interfaces de usuário.
- Next js: é um *framework* React com foco em produção e eficiência.
- Node.js: é uma tecnologia assíncrona que trabalha em uma única *thread* de execução que utiliza JavaScript e serve para fazer APIs e *backends*.
- PostgreSQL: é um gerenciador de bancos de dados relacionais e permite a implementação da linguagem SQL em estruturas de banco de dados.
- Vercel: é uma plataforma voltada para a hospedagem de aplicações e é a empresa criada do Next js.
- Render: é uma plataforma voltada para a hospedagem de aplicações e bancos de dados na nuvem.
- Imgur: é um servidor de hospedagem de imagem
- Swagger: é uma ferramenta de código aberto que gera uma página da web que documenta *endpoints*.

## 5.3 Hospedagem

Para serem realizados testes pelos usuários durante o desenvolvimento do SAMI, foram utilizadas duas ferramentas para hospedagem do código na nuvem, sendo essas o Vercel e o Render. Ambas as ferramentas foram escolhidas devido a sua facilidade de uso e opção de plano de utilização gratuita.

## 5.4 Cronograma

O cronograma de implementação e testes do SAMI será realizado no TCC2 conforme cronograma na tabela 10.

A implementação terá como base os requisitos priorizados como *Must* e *Should*.

Tabela 10 – Cronograma de implementação e testes do SAMI.

Atividade	Outubro	Novembro	Dezembro	Janeiro	Fevereiro
Implementação do SAMI	X	X	X	X	
Escrita do TCC 2				X	X
Defesa do TCC 2					X

## 5.5 Processo de desenvolvimento

### 5.5.1 Definição do menor produto viável

Para iniciar o desenvolvimento, foi definido quais seriam as funcionalidades que o sistema deveria ter a fim de ser viável para utilização. Essas foram:

Tabela 11 – Tabela de requisitos para atender o menor produto viável.

Requisito	Classificação
Login (RF1)	<i>Must</i>
Cadastro de professores (RF2)	<i>Must</i>
Cadastro de alunos (RF3)	<i>Must</i>
Edição de usuário (RF4)	<i>Must</i>
Cadastro de turma (RF5)	<i>Must</i>
Remoção de turma (RF7)	<i>Must</i>
Criação de formulário (RF11)	<i>Must</i>
Edição de formulário (RF12)	<i>Must</i>
Remoção de formulário (RF13)	<i>Must</i>
Visualização de lista de formulários (RF14)	<i>Must</i>
Adição de questão ao formulário (RF15)	<i>Must</i>
Adição de opção à questão (RF16)	<i>Must</i>
Criação de indicador (RF23)	<i>Must</i>
Edição de indicador (RF24)	<i>Must</i>
Remoção de indicador (RF25)	<i>Must</i>
Visualização de lista de indicadores (RF26)	<i>Must</i>
Visualização de dados do indicador (RF27)	<i>Must</i>
Adição de formulário ao indicador (RF28)	<i>Must</i>
Visualização dos dados da aplicação do formulário (RF31)	<i>Must</i>
Visualização para resposta do aluno ao formulário (RF32)	<i>Must</i>

Como antes citado no planejamento, foram utilizadas para definição do menor produto viável, os requisitos com priorização *Must*. Durante o desenvolvimento foram priorizados também alguns outros requisitos, sendo esses:



Tabela 12 – Tabela de requisitos implementados de forma adicional.

Requisito	Classificação
Edição de turma (RF6)	<i>Should</i>
Agendamento de aplicação de formulário (RF29)	<i>Should</i>
Visualização de gráficos com os dados do indicador (RF38)	<i>Should</i>
Adição de template de formulário (RF17)	<i>Could</i>
Edição de template de formulário (RF18)	<i>Could</i>
Remoção de template de formulário (RF19)	<i>Could</i>
Visualização de como o formulário será apresentado ao aluno (RF33)	<i>Could</i>

Alguns dos requisitos foram parcialmente atendidos durante o desenvolvimento, são esses:

Tabela 13 – Tabela de requisitos parcialmente atendidos.

Requisito	Classificação	O que foi feito
Visualização de todos os usuários (RF36)	<i>Should</i>	É apresentado para o administrador apenas a lista com todos os professores
Edição de outros usuários (RF37)	<i>Should</i>	É apresentada para o administrador apenas a opção de editar professores. Alunos podem ser atualizados pelos professores ao adicionarem novamente o aluno em alguma turma com novos dados

### 5.5.2 Planejamento

Após definido o menor produto viável, foram definidas as *sprints* para implementação do projeto. As *sprints* tinha duração de uma semana com a primeira iniciando no dia 15 de setembro de 2022 e a última finalizando no dia 19 de janeiro de 2023.

Na definição das tarefas das *sprints* além dos requisitos presentes no mínimo produto viável foram listadas tarefas de configuração do ambiente de trabalho, configuração do projeto e a configuração das plataformas responsáveis por hospedar a aplicação na nuvem, que foram Vercel para o *front-end* e Railway e Render para o *back-end*.

A tabela 14 mostra as tarefas realizadas durante o desenvolvimento do SAMI. As tarefas estão em ordem de execução e prioridade, e também é possível notar que o tempo de algumas tarefas ultrapassa o tempo de uma *sprint*, pois a mesma foi executada em mais de uma *sprint*.

A pontuação das tabelas foi feita usando a sequência de Fibonacci para mensurar a dificuldade e esforço referente a cada tarefa. Essa ideia é utilizada definindo o que seria um exemplo de tarefa que é atribuída com o menor valor não nulo, que seria um, e a partir disso, são mensuradas as outras tarefas. Para o desenvolvimento do SAMI, foi definido

como exemplo de tarefa com dificuldade um, o seguinte: Adicionar botão para redirecionar à tela inicial.

As tarefas com pontuação composta, tem essa representação pois tiveram mais de uma *sprint* de desenvolvimento, e cada nota representa sua pontuação em uma diferente.

Tabela 14 – Tarefas realizadas por *sprint*.

Período	Tarefa	Pontuação
15/09 - 22/09	Cronograma das <i>sprints</i>	3
15/09 - 22/09	Iniciar ambiente do <i>front-end</i>	3
15/09 - 22/09	Criar pagina de login (RF01)	3
22/09 - 29/09	Iniciar ambiente do <i>back-end</i>	5
22/09 - 29/09	Criar <i>endpoint</i> para login (RF01)	5
29/09 - 06/10	Criar pagina de cadastro de usuário (RF02, RF03)	3
29/09 - 06/10	Criar <i>endpoint</i> para cadastro de usuário (RF02, RF03)	3
06/10 - 20/10	Criar ambiente de teste	13 + 13
20/10 - 27/10	Gerenciar paginas a serem mostradas de acordo com o tipo de usuário	8
20/10 - 03/11	Gerenciar navegação do usuário	5 + 8
20/10 - 03/11	Criar <i>endpoints</i> para CRUD de turma (RF05,RF07)	5
3/11 - 10/11	Criar pagina de cadastro de turma (RF05)	8
10/11 - 24/11	Finalizar design das telas	8
24/11 - 01/12	Criar <i>endpoints</i> para CRUD de formulário (RF11,RF12, RF13)	8
24/11 - 15/12	Criar pagina de edição de formulários (RF26)	8 + 13 + 13
05/01 - 12/01	Criar pagina de lista de formulários (RF26)	5
05/01 - 12/01	Criar pagina de criação e edição de indicador (RF11, RF12, RF13, RF15, RF16)	5
05/01 - 12/01	Criar <i>endpoints</i> para CRUD de interação do formulário com indicador (RF28)	5
05/01 - 12/01	Criar pagina de resposta do aluno ao formulário (RF32)	5
12/01 - 19/01	Criar pagina de lista de indicadores (RF26)	5
12/01 - 19/01	Criar modal de aplicação do formulário (RF32)	5
12/01 - 19/01	Criar <i>endpoints</i> para CRUD de indicador (RF23,RF24, RF25)	5
12/01 - 19/01	Criar pagina de detalhes do indicador (RF27)	8

### 5.5.3 Versão final do SAMI

O presente código está disponível na plataforma Github na seguinte organização: <https://github.com/SAMI-TCC-Joao>.

### 5.5.4 Telas da versão final

Nesta seção as telas da versão final do SAMI são apresentadas.

#### 5.5.4.1 Tela de login

A figura 24 é a tela de login, utilizada para a autenticação do usuário no SAMI.

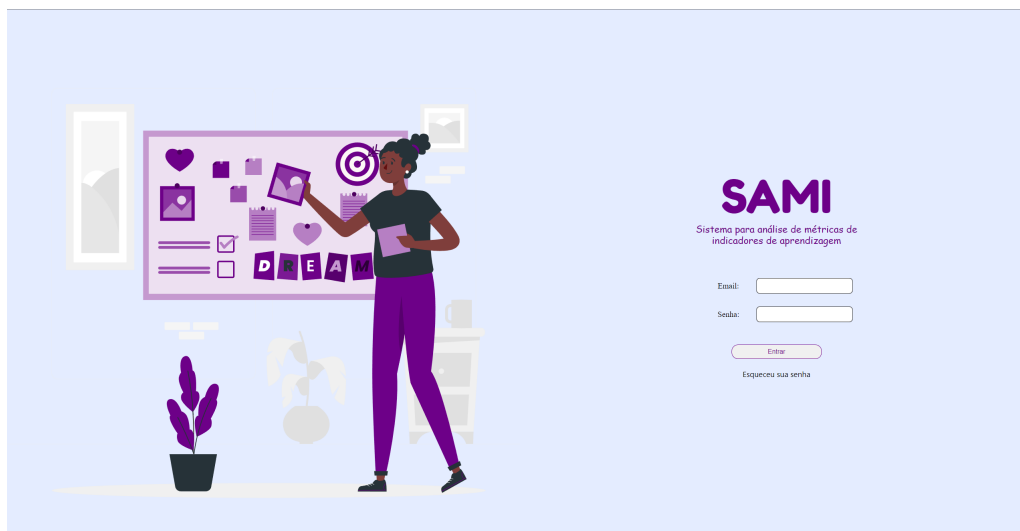


Figura 24 – Tela de login.

#### 5.5.4.2 Tela de nova senha

A figura 25 é a tela de mudança de senha. Nela é possível alterar a senha atual inserindo a senha antiga e a nova nessa ordem. Um link para essa tela é enviado ao solicitar mudança de senha.

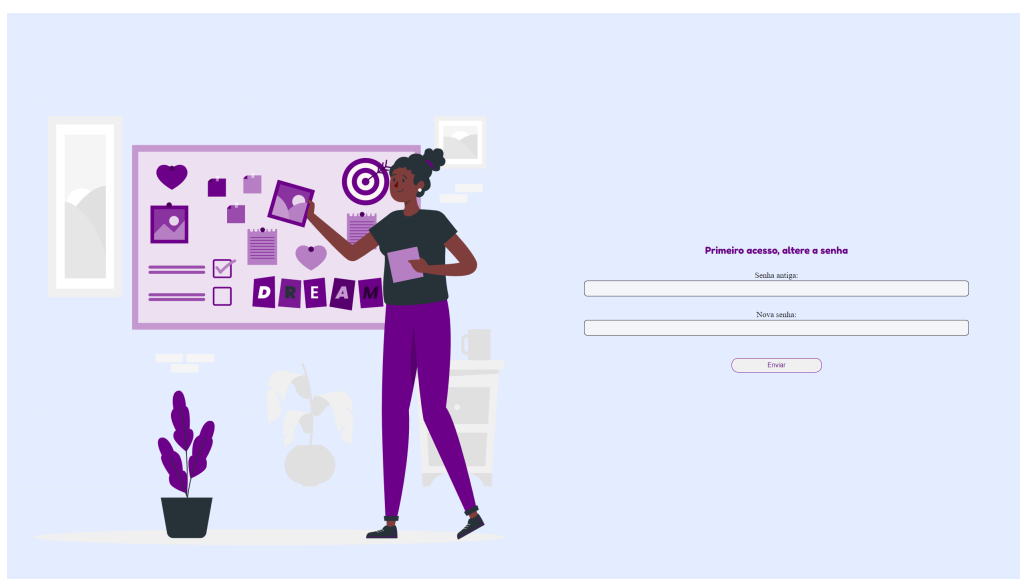


Figura 25 – Tela de nova senha.

#### 5.5.4.3 Tela de lista de formulários

A figura 26 é a tela de lista de formulários, que atua também como a página inicial. Nela é possível alternar entre os formulários do usuário logado e todos os formulários criados pelos administradores como *templates*.

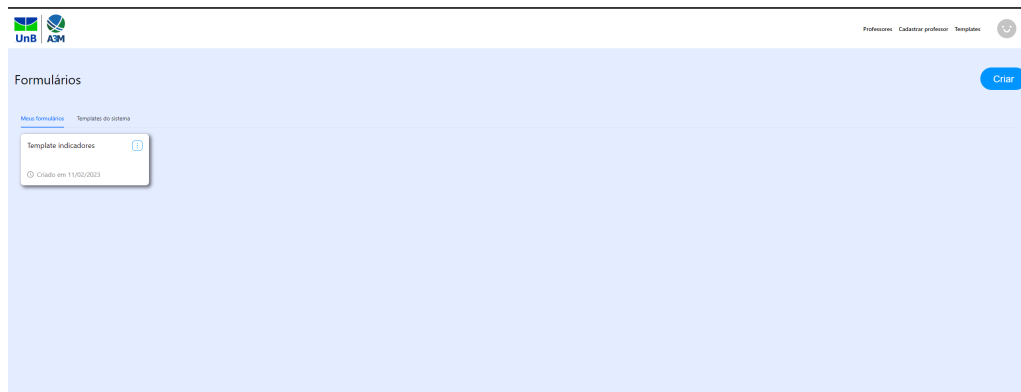


Figura 26 – Tela de lista de formulários.

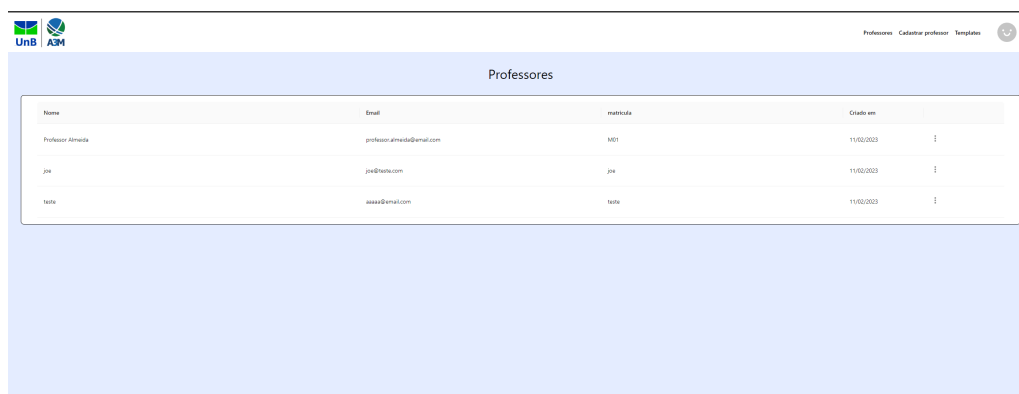
#### 5.5.4.4 Tela de cadastro de professores

A figura 27 é a tela de cadastro de professores. Nela também é possível criar usuários administradores.

Figura 27 – Tela de cadastro de professores.

#### 5.5.4.5 Tela de lista de professores

A figura 28 é a tela de lista de professores. Nela é possível visualizar, editar e excluir os professores cadastrados no sistema.

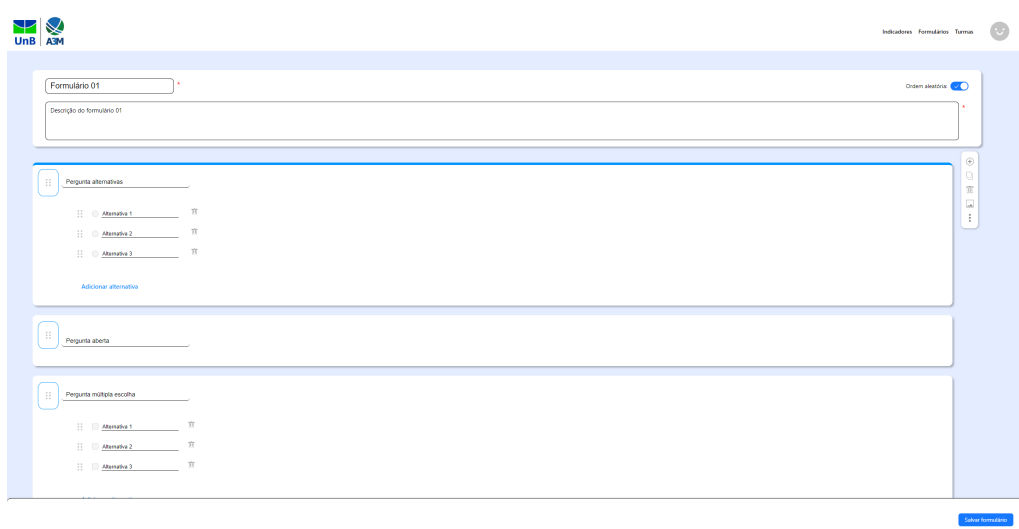


Nome	Email	matricula	Criado em	
Professor Almeida	professor.almeida@email.com	1001	11/02/2023	⋮
João	joao@unb.com	joao	11/02/2023	⋮
teste	teste@email.com	teste	11/02/2023	⋮

Figura 28 – Tela de lista de professores.

#### 5.5.4.6 Tela de cadastro de formulário

A figura 29 é a tela de cadastro de formulário. Nela é possível criar um novo formulário utilizando questões de múltipla escolha, texto livre, alternativas e escala de Likert.



Formulário 01

Descrição do formulário 01

Ordenar perguntas: ON

Pergunta alternativa

- Alternativa 1
- Alternativa 2
- Alternativa 3

Adicionar alternativa

Pergunta aberta

Pergunta múltipla escolha

- Alternativa 1
- Alternativa 2
- Alternativa 3

Salvar formulário

Figura 29 – Tela de cadastro de formulário.

#### 5.5.4.7 Tela de visualizar formulário

A figura 30 é a tela de visualizar formulário. Nela é possível ter uma visualização de como o aluno vai ver aquele formulário ao respondê-lo.

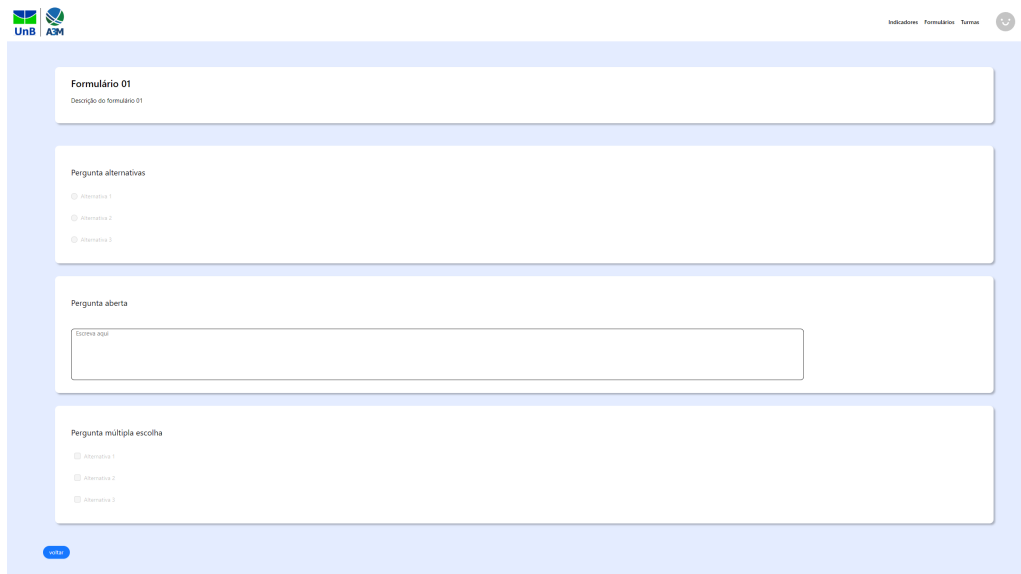


Figura 30 – Tela de visualizar formulário.

#### 5.5.4.8 Tela de cadastro de indicador

A figura 31 é a tela de cadastro de indicador. Nela é possível adicionar um indicador indicando seu conjunto e metodologia.

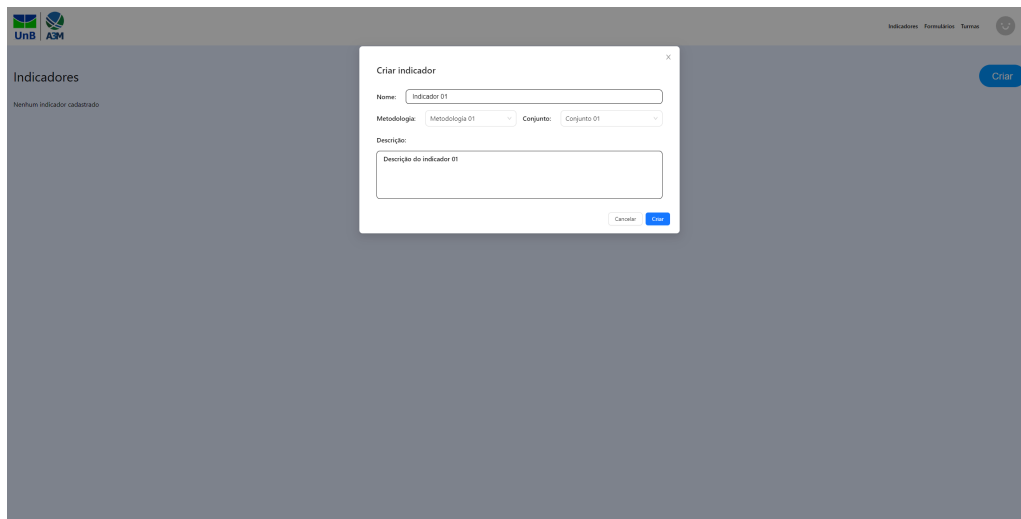


Figura 31 – Tela de cadastro de indicador.

#### 5.5.4.9 Tela de edição de indicador

A figura 32 é a tela de edição de indicador. Nela é possível adicionar, editar e apagar formulários do indicador, adicionar, editar ou apagar aplicações desses formulários à turmas e editar as informações do indicador.

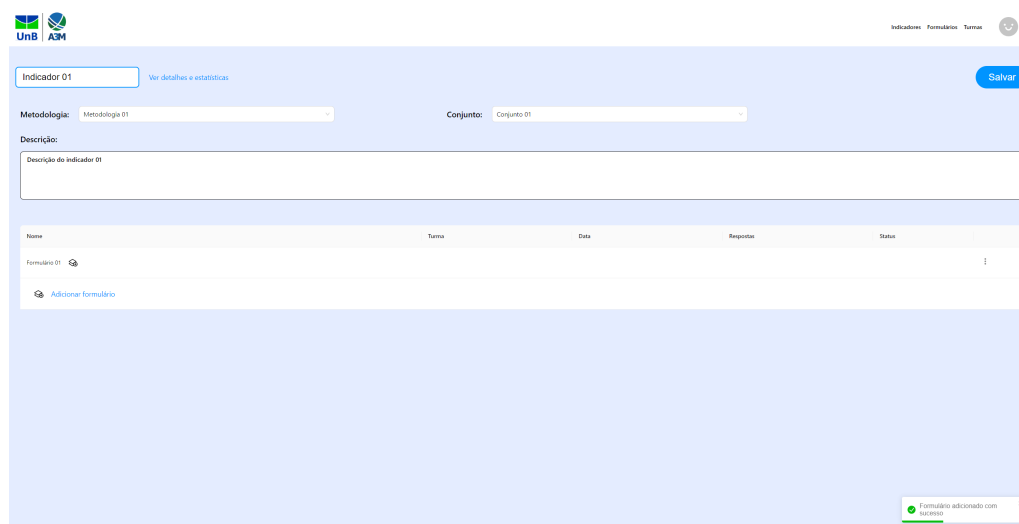


Figura 32 – Tela de edição de indicador.

#### 5.5.4.10 Tela de detalhes do indicador

A figura 33 é a tela de detalhes do indicador. Nela é possível visualizar as respostas das aplicações dos formulários que pertencem àquele indicador, podendo filtrar por turmas e alunos.

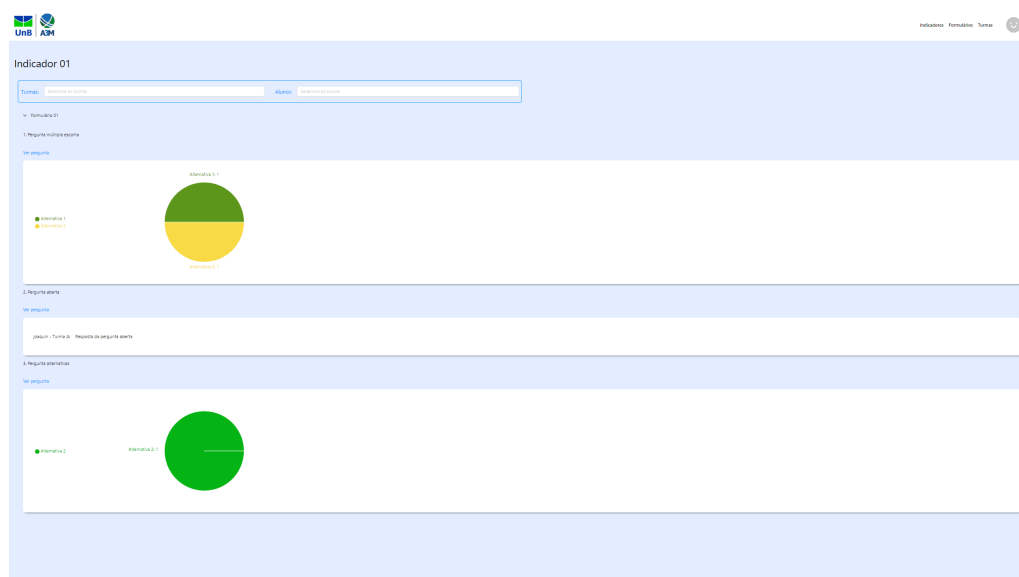


Figura 33 – Tela de detalhes do indicador.

#### 5.5.4.11 Tela de cadastro de turma

A figura 34 é a tela de cadastro de turma. Nela é possível adicionar uma nova turma, também podendo adicionar seus alunos por meio de um arquivo do tipo xlsx.

UnB | ASM

Indicadores Formulários Turmas

### Cadastro Turma

Nome: Turma A

Nome da disciplina: Disciplina A

Código da disciplina: ED1

Semestre: 1

Alunos		
Nome	Email	Matrícula
No data		

[Excluir turma](#)

**ADICIONAR ALUNOS**  
(apenas enquanto está)

[Voltar](#) [Salvar](#)

Figura 34 – Tela de cadastro de turma.

#### 5.5.4.12 Tela de lista de turma

A figura 35 é a tela de lista de turma. Nela é possível visualizar, editar e excluir as turmas cadastradas pelo usuário.

UnB | ASM

Indicadores Formulários Turmas

### Turmas

[Criar](#)

Disciplina	Turma	Semestre	Qtd. de Alunos
Disciplina A	Turma A	1	0

Figura 35 – Tela de lista de turma.



## 6 Conclusão e trabalhos futuros

Neste trabalho foram utilizados os conhecimentos adquiridos durante a graduação no curso de Engenharia de Software na Universidade de Brasília. As disciplinas da graduação mais presentes são as disciplinas de Requisitos de software presente em toda a elicitação de requisitos do SAMI, a disciplina de Métodos de desenvolvimento de software presente na metodologia de software aplicada e a disciplina de Arquitetura e desenho de software presente na decisão arquitetural da aplicação, bem como as tecnologias utilizadas para a sua implementação.

Observando o processo de desenvolvimento planejado e implementado tenho na minha percepção que o processo de desenvolvimento do SAMI teve um bom desempenho, considerando que seu planejamento e implementação foi realizado por uma pessoa com habilidades específicas.

O SAMI é um sistema para coleta de dados para métricas de indicadores de aprendizado que tem o propósito de ser utilizado por professores participantes do projeto A3M e, com o resultado obtido, tem potencial de atender as necessidades dos seus usuários.

### 6.1 Trabalhos futuros

O SAMI é um aplicativo desenvolvido em um curto espaço de tempo, portanto sua implementação teve o escopo reduzido. O SAMI pode ter melhorias já especificadas como:

Tabela 15 – Tabela de requisitos quem podem ser atendidos em melhorias futuras.

Requisito	Classificação
Cadastro de grupo (RF8)	<i>Should</i>
Edição de grupo (RF9)	<i>Should</i>
Remoção de grupo (RF10)	<i>Should</i>
Adição de template de indicador (RF20)	<i>Could</i>
Edição de template de indicador (RF21)	<i>Could</i>
Remoção de template de indicador (RF22)	<i>Could</i>
Opção para respostas do formulário de forma anônima (RF30)	<i>Could</i>
Visualização da lista de todos os formulários pelo administrador (RF35)	<i>Should</i>

Além das melhorias relacionadas aos requisitos ainda não atendidos, algumas outras puderam ser observadas, como:

- Relacionar os dados entre indicadores do mesmo conjunto ou metodologia

- Adicionar filtros nas telas de formulários e indicadores
- Adicionar opção de duplicar formulários e indicadores
- Adicionar opção para editar e remover metodologia e conjunto

# Referências

- AURELIO, I. *O que é e para que serve o Next.js?* 2020. <<https://segredo.dev/o-que-e-next-js/>>. Acessado em: 01/02/2023. Citado na página 68.
- DAVIS, A. *Software Requirements: Objects, Functions, and States*. PTR Prentice Hall, 1993. ISBN 9780138057633. Disponível em: <<https://books.google.com.br/books?id=ZJ1QAAAAMAAJ>>. Citado na página 22.
- IIBA. Um guia para o corpo de conhecimento de análise de negócios™ (guia babok®) versão 2.0. *Toronto, Canadá: International Institute of Business Analysis*, 2011. Citado na página 29.
- MACHADO, A. *Qual a diferença entre front-end e back-end?* 2022. <<https://tecnoblog.net/responde/qual-a-diferenca-entre-front-end-e-back-end/>>. Acessado em: 01/02/2023. Citado 2 vezes nas páginas 67 e 68.
- MELO, L. V. de Sousa e et al. Programa aprendizagem para o terceiro milênio (a3m): Política de valorização e promoção de ações educacionais inovadoras na universidade de Brasília. *XIV Congresso Brasileiro de Ensino Superior a Distância*, 2017. Citado 2 vezes nas páginas 19 e 21.
- OLIVEIRA, D. M. de. Programa de aprendizagem para o 3.º milênio. *SEI/UnB - 8028032 - Edital*, 2022. Citado na página 21.
- OLIVEIRA, R. Prince2: A técnica de priorização moscow. 01 2014. Disponível em: <[https://www.researchgate.net/publication/317539229\\_PRINCE2\\_A\\_Tecnica\\_de\\_Priorizacao\\_MoSCoW](https://www.researchgate.net/publication/317539229_PRINCE2_A_Tecnica_de_Priorizacao_MoSCoW)>. Citado 2 vezes nas páginas 29 e 55.
- ORIOLO, M. et al. Fame: Supporting continuous requirements elicitation by combining user feedback and monitoring. In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*. [S.l.: s.n.], 2018. p. 217–227. ISSN 2332-6441. Citado na página 27.
- PRADO, N. A. e E. V. Técnicas de elicitação de requisitos no desenvolvimento de software: uma revisão sistemática da literatura. *AtoZ: novas práticas em informação e conhecimento*, v. 10, n. 1, p. 39–49, 2021. ISSN 2237-826X. Disponível em: <<https://revistas.ufpr.br/atoz/article/view/77393>>. Citado 2 vezes nas páginas 27 e 28.
- SOMMERVILLE, I. *Software Engineering*. 9. ed. Harlow, England: Addison-Wesley, 2010. ISBN 978-0-13-703515-1. Citado na página 30.
- SOMMERVILLE, I. *Engenharia de software*. Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>. Citado 7 vezes nas páginas 11, 22, 23, 24, 25, 26 e 28.
- STHEM. *UnB cria Programa de Aprendizagem para o 3º Milênio para inspirar professores a inovar*. 2021. <<https://www.sthembrasil.com/unb-cria-programa-de-aprendizagem-para-o-3o-milenio-para-inspirar-professores-a-inovar/>>. Acessado em: 01/02/2023. Citado na página 19.

VALENTE, M. *Modern software engineering: Principles and Practices for Productive Software Development*. [s.n.], 2020. Disponível em: <<http://hdl.handle.net/1843/37905>>. Citado na página 23.

WIEGERS, K. E.; BEATTY, J. *Software Requirements 3*. USA: Microsoft Press, 2013. ISBN 0735679665. Citado na página 27.

WILLIAMS, W. *O grande diferencial do NestJS*. 2022. <<https://fullcycle.com.br/o-grande-diferencial-do-nestjs/>>. Acessado em: 01/02/2023. Citado na página 68.