

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE MOGI MIRIM

Ana Letícia Silvério
Emiliano Callegari Correia de Souza
Karina Caporali D'Avila
Raphael do Carmo

SGTG: SISTEMA DE GERENCIAMENTO DE TRABALHO DE
GRADUAÇÃO

MOGI MIRIM
2023

Ana Letícia Silvério
Emiliano Callegari Correia de Souza
Karina Caporali D'Avila
Raphael do Carmo

SGTG: SISTEMA DE GERENCIAMENTO DE TRABALHO DE GRADUAÇÃO

Trabalho de Graduação apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia Arthur de Azevedo de Mogi Mirim como pré-requisito para a obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Marcio Rodrigues Sabino

Mogi Mirim
2023

Ana Letícia Silvério
Emiliano Callegari Correia de Souza
Karina Caporali D'Avila
Raphael do Carmo

SGTG: SISTEMA DE GERENCIAMENTO DE TRABALHO DE GRADUAÇÃO

Este trabalho de Graduação foi julgado e aprovado para obtenção do título de Tecnólogo, no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Faculdade de Tecnologia de Mogi Mirim “Arthur de Azevedo”

Mogi Mirim, xx de xxxxx de 2023.

Prof. Me. Cesar Eduardo Soares Bagnolo
Coordenador do Curso Superior de Tecnologia
em Análise e Desenvolvimento de Sistemas

BANCA EXAMINADORA

Prof. Me. Marcio Rodrigues
Sabino
Orientador

Prof. XXXXXXXXXXXXXXXX
XXXXXXX
Membro

Prof. XXXXXXXX
XXXXXXXXXXXXXX
Membro

RESUMO

A Fatec Arthur de Azevedo em Mogi Mirim, no ano de 2023, para os cursos de Análise e Desenvolvimento de Sistemas, Gestão Empresarial e Mecatrônica, exige uma atividade científica, nomeada como Trabalho de Graduação (TG), tendo como principal objetivo a unificação e apresentação dos conhecimentos adquiridos durante todo o período do curso. Atualmente, o processo de formalização dos grupos e temas de projeto, escolha de orientador, entrega de documentações, marcação de bancas de defesa e registros são realizados em processos via e-mail, planilhas eletrônicas e pastas disponibilizadas na nuvem. O objetivo deste trabalho foi dar continuidade ao desenvolvimento de um sistema de gerenciamento dos trabalhos de graduação entregues semestralmente. Foram utilizadas as metodologias *design thinking* e a pesquisa bibliográfica para o levantamento de problemáticas, requisitos e tecnologias viáveis para a solução. Para o desenvolvimento *backend* utilizou-se o Node.js e Typescript e no *frontend* o *framework* do React.js, o Next.js. Para o banco de dados, adotou-se o Firebase que é NoSQL e para tornar o projeto robusto e escalável, foi decidido incorporar o PostgreSQL como banco de dados relacional. Com isso, foi possível o desenvolvimento de um sistema que poderá ser disponibilizado para a comunidade discente, docente e coordenação, com o ideal de permitir o envio dos trabalhos, registro e organização de grupos e entregas desenvolvidas na unidade de ensino, além da visualização, controle e até mesmo o agendamento de datas para a banca de defesa.

Palavras-chave: Conclusão de curso; Sistema; Trabalho de graduação.

ABSTRACT

Fatec Arthur de Azevedo in Mogi Mirim, to the courses of Systems Analysis and Development and Mechatronic requires a scientific activity named Graduation Work (GW). Having as the main objective the unification and presentation of acquired skills during the period of the course. The objective of this course is giving continuation to the development of a management system of graduation works biannually delivered, available to the student body, teaching staff, and coordination, a software that will allow the sending of work, an impression of minutes, registry and organization of groups and deliveries developed in the teaching unit, besides the visualization and control of them, and the appointment of dates to the defense jury.

Keywords: *Course conclusion; System; Graduation work.*

LISTA DE FIGURAS

Figura 1 – Etapas do <i>Design Thinking</i>	15
Figura 2 – O conceito do diamante.....	16
Figura 3 – Fluxo de processo linear	20
Figura 4 – Fluxo de processo iterativo.....	21
Figura 5 – Fluxo de processo evolucionário	21
Figura 6 – Fluxo de processo paralelo	21
Figura 7 – O Processo Unificado.....	22
Figura 8 – O paradigma da prototipação	24
Figura 9 – Arquitetura do sistema completo	34
Figura 10 – Tela inicial do projeto.....	36
Figura 11 – Tela de cadastro.....	37
Figura 12 – Tela de verificação de e-mail.....	38
Figura 13 – Modelo de e-mail recebido para confirmar o acesso	38
Figura 14 – Tela de Login.....	39
Figura 15 – Tela para completar o cadastro	40
Figura 16 – Dashboard principal do aluno	41
Figura 17 – Tela de cadastro de grupo.....	42
Figura 18 – Tela de perfil.....	43
Figura 19 – Dashboard da secretaria/coordenação.....	43
Figura 20 – Tela de convite de grupo	44
Figura 21 – Dashboard do orientador	45
Figura 22 – Tela de agendamento.....	46
Figura 23 – Tela de confirmação do agendamento	46
Figura 24 – DCU 001 - Login.....	53
Figura 25 – DCU 002 - Inativar.....	54
Figura 26 – DCU 003 - Termo_formalização	55
Figura 27 – DCU 004 - Aceitar_grupo	56
Figura 28 – DCU 005 - Criar_grupo.....	57
Figura 29 – DCU 006 - Cadastro	58
Figura 30 – DCU 007 - Aceitar_formalização	59
Figura 31 – DCU 008 - Marcar_banca.....	60

Figura 32 – DA 001 - Cadastro de Aluno/Orientador	61
Figura 33 – DA 002 - Inativar aluno	62
Figura 34 – DA 003 - Criar Grupo.....	63
Figura 35 – DA 004 - Incluir Professor	64
Figura 36 – DA 005 - Alterar Professor	65
Figura 37 – DA 006 - Aceitar Grupo	66
Figura 38 – DA 007 - Aceitar Orientados.....	67
Figura 39 – DA 008 - Consultar Aluno.....	68
Figura 40 – DA 009 - Agendar Banca.....	69

LISTA DE QUADROS

QUADRO 1 – Cronograma de pesquisa da fase 2.....	18
---	----

SUMÁRIO

1. INTRODUÇÃO	11
1.1. Objetivos	13
1.2. Hipótese	13
1.3. Estrutura do documento	13
2.1. <i>Design Thinking</i>	15
2.2. Etapas do <i>Design Thinking</i>	16
2.3. Cronograma de Planejamento	18
3. REFERENCIAL TEÓRICO	20
3.1. Engenharia de <i>Software</i>	20
3.1.1. Modelo de processo genérico	21
3.2. RUP	24
3.3. Prototipação	25
3.4. Banco de Dados	26
3.4.1. Banco de Dados Relacional	27
3.4.2. Banco de Dados Não Relacional	27
3.5. Aplicação <i>Web</i>	28
3.6. HTML	28
3.7. CSS	29
3.8. <i>JavaScript</i>	29
3.8.1. React	29
3.8.2. NextJS	30
3.8.3. Express.JS	30
3.9. TypeORM	30
3.10. Docker	31
3.11. UML	32
3.11.1. História da UML	32
3.11.2. Importância da modelagem	33
3.11.3. Tipos de Diagrama	33
3.12. Identidade Visual	34
4. DESENVOLVIMENTO	35
4.1. Imersão	35
4.2. Ideação	35
4.3. Prototipação	36

4.3.1. <i>Backend</i>	36
4.3.2. <i>Frontend</i>	37
5. RESULTADOS.....	39
6. DISCUSSÕES FINAIS	50
REFERENCIAL BIBLIOGRÁFICO	52
APÊNDICE	57
1. Diagrama de Caso de Uso.....	57
2. Diagrama de Atividades.....	65
ANEXOS	74
1. DECLARAÇÃO DE INEXISTÊNCIA DE PLÁGIO	74
2. AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE GRADUAÇÃO ...	75

1. INTRODUÇÃO

O emprego da tecnologia para rotinas do dia a dia já se tornou algo indispensável. O acesso aos seus requintes, por meio da aquisição de dispositivos foi impulsionado pelo advento do desenvolvimento e evolução dos transistores e circuitos integrados, cuja evolução trouxe uma progressiva redução de custos de fabricação que outrora tornavam os dispositivos eletrônicos pouco acessíveis, o que também era potencializado pela baixa variedade de modelos, frequência de lançamentos, e limitação de funcionalidades e *design*.

Por conta desse movimento evolutivo nas recentes décadas, que possibilita a rápida atualização de dispositivos, e fortalecido pela ampla expansão da internet, métodos e ferramentas de desenvolvimento, a tecnologia se consolida como recurso essencial, transformando e criando conceitos, como solicitar um “táxi”, refeições, compartilhar conteúdo, comunicar-se instantaneamente e automatizar processos manuais. Observa-se então constantes evoluções e lançamentos de novos produtos, com novas funcionalidades e melhor desempenho, com destaque à criação de equipamentos “inteligentes”, ou seja, conectados à Internet e outros dispositivos que podem desempenhar tarefas de formas diferentes das convencionais das quais foram concebidas. A exemplo, os *Smart Watches*, as *Smart Tvs*, e carros com painel integrado, entre outros artefatos que fazem parte do cotidiano (SANTAELLA, 2013).

Ainda como fato relevante decorrente da expansão da Internet para grande parte da população, e a, agora trivial, aquisição de dispositivos, a fluída comunicação e entretenimento mitiga os efeitos da distância, estabelecendo o conceito de que não existe mais “perto ou longe”, e sim “conectado”. E paralelamente ao entretenimento, recentes e promissoras tecnologias, como o uso da Inteligência Artificial e demais automatizações, estabelecem a integração entre ferramentas e sistemas, fortalece o compartilhamento e integridade de dados, bem como traz eficiência operacional e financeira pelo aumento de produtividade e mitigação de erros, padronizando assim produtos e serviços e maior transparência e confiança nos negócios (OLIVEIRA, 2017).

Em linha com esse advento, a Faculdade de Tecnologia Arthur de Azevedo de Mogi Mirim (FATEC MM), no estado de São Paulo, estabelece ao final dos cursos superiores de Análise e Desenvolvimento de Sistemas e Mecatrônica, que os discentes devem elaborar um projeto intitulado Trabalho de Graduação (TG), sendo

esta uma atividade científica prevista no Projeto Pedagógico, obrigatória para alunos de Mecatrônica Industrial e de Análise e Desenvolvimento de Sistemas para obtenção do certificado de conclusão do curso como tecnólogo. Sua estrutura é pautada na síntese criativa utilizando os conhecimentos obtidos por meio das disciplinas desenvolvidas ao longo do curso.

O estabelecimento dos grupos, bem como orientadores e banca examinadora cabem ao coordenador do curso. O professor orientador é o responsável pela transmissão do conteúdo pedagógico, cronograma dos eventos relacionados e demais tópicos relacionados ao seu respectivo grupo.

Alguns dados, como os endereços de e-mails dos alunos, são importados do SIGA (Sistema Integrado de Gestão Acadêmica), sistema usado pela secretaria para gestão acadêmica, e armazenados em planilhas do Excel.

Identifica-se dificuldades e retrabalhos devido à integridade desses dados, que são inseridos manualmente e passivos de erros de digitação, o que gera desperdícios operacionais e financeiros pelo uso de insumos extras, haja vista a necessidade de impressões de documentos físicos, bem como não possibilita uma boa experiência ao operador responsável, onerando sua produtividade e consequentemente o atendimento ao aluno (cliente final) por conta de atrasos decorrentes da falha na comunicação. Nesse sentido, observa-se uma oportunidade na reestruturação desse negócio, por meio de revisão de processos e novos sistemas.

No final do primeiro semestre de 2022, o aluno Pedro Luis Berbel dos Santos sob orientação do professor Marcio Rodrigues Sabino realizou a sua defesa de trabalho de graduação intitulada “SISTEMA DE GERENCIAMENTO DE TRABALHOS DE GRADUAÇÃO”. Neste trabalho, o autor idealizou as seguintes fases de desenvolvimento:

FASE 1: organização e desenvolvimento de todo o fluxo de criação dos grupos de usuários definidos por alunos, orientadores, coordenação/secrretaria e administrador, bem como seus cadastros e forma de acesso ao sistema, diferenciando por tipo de usuário. Criação de uma listagem de orientadores que deverá ser cadastrada pelo administrador, assim como as credenciais de acesso de usuários da coordenação/secrretaria.

Ainda nessa fase, a criação de grupos de trabalho, convite aos alunos e orientadores envolvidos.

FASE 2: organização e desenvolvimento das funcionalidades de *upload* e *download* dos arquivos finais do Trabalho de Graduação, alteração de dados pessoais, alteração de detalhes de grupos já cadastrados, bem como a possibilidade de um aluno deixar um grupo ao qual faz parte. Também poderá compreender todo o fluxo em que o orientador irá agendar as datas de qualificação e defesa, e convite a outros orientadores para participar da banca

de determinado grupo de trabalho. Dessa forma, na dashboard do orientador, haverá um novo módulo para gerenciamento das bancas e seus respectivos convites e datas agendadas.

FASE 3: organização e desenvolvimento do módulo para que a coordenação/secretaria se comunique por meio de um sistema de mensagens, podendo assim enviar comunicados gerais ou para determinado grupo ou aluno específico. Fica ainda previsto para essa fase a implementação do controle de versões do trabalho escrito, dividido nas seis principais partes da estrutura, descritas anteriormente. Por fim, deverá ser desenvolvida a área para indicar a situação de um grupo, após qualificação e/ou defesa, ou seja, se ele foi aprovado ou reprovado e a geração da Ata de aprovação pela coordenação/secretaria.

Como objetivo principal concluído, a fase 1 foi finalizada com sucesso, restando as fases seguintes para que outros grupos pudessem realizar.

1.1. Objetivos

- Identificar as necessidades dos alunos, secretaria e orientadores de trabalho de graduação mediante a administração;
- Estruturar os requisitos que um sistema de gerenciamento deva ter para sanar as necessidades do público-alvo;
- Concluir a fase 2, a qual consiste no desenvolvimento de melhorias de uma aplicação web, focando na união de alunos e orientador em um grupo de trabalho.

1.2. Hipótese

Espera-se com a conclusão da segunda fase desse projeto, apresentar uma aplicação mais completa em relação ao gerenciamento de trabalhos, possibilitando emitir documentos formalizados, enviar e-mail de notificações para orientadores, orientados e, se necessário, para a coordenação de maneira automatizada e com todas as informações necessárias, facilitando a utilização para todos os usuários.

1.3. Estrutura do documento

Neste trabalho, está organizado o Capítulo 2, no qual serão mostradas as metodologias usadas para nortear este projeto.

No Capítulo 3, é desenvolvido o referencial teórico, no qual são abordados temas como a Engenharia de *Software*, a história da UML e banco de dados relacionais e não relacionais. Outro tema a ser discutido nesse capítulo são as tecnologias que serão utilizadas na implementação do *backend* e *frontend* para o desenvolvimento dessa aplicação web.

Com foco no desenvolvimento, o Capítulo 4 tratará dos conhecimentos obtidos através dos estudos realizados, das pesquisas e suas aplicações.

No capítulo 5 será destinado aos resultados obtidos, abordando as telas desenvolvidas durante o projeto e a funcionalidade de cada uma.

O Capítulo 6 é dedicado a discussões finais, serão levantados os próximos passos a serem desenvolvidos, os objetivos atingidos e a busca pelo sistema eficaz na próxima etapa.

No capítulo final são apresentadas as referências bibliográficas que foram usadas como consulta para esse projeto.

2. METODOLOGIA DE PESQUISA

Neste capítulo, será apresentada a metodologia de pesquisa utilizada para o desenvolvimento desse trabalho.

2.1. *Design Thinking*

Projetos e iniciativas diversas, baseados ou não em tecnologias, devem atender a um escopo ou necessidade do cliente final para se tornarem viáveis, produtivas e sustentáveis, sendo este um desafio potencializado pela evolução da tecnologia, que permite a implementação de um vasto leque de recursos.

Nesse sentido, o *Design Thinking*, metodologia de inovação pautada no Design e que foca na criatividade, se mostra como uma poderosa ferramenta para se identificar, propor, testar e implantar soluções em linha com as “dores” do cliente ou contexto, sendo, portanto, uma metodologia para resolver problemas de natureza humana, ou seja, em contextos em que a figura humana contida nos processos, mas que também pode coexistir com sistemas ou demais rotinas automatizadas, é onerada em algum sentido (sente “dores” – demoras, inconsistências, incertezas, etc) (DAMASCENO, 2018).

Apesar de estar pautada em *Design*, a principal e desejada habilidade para se executar o *Design Thinking* é a empatia, pois no caso, mais importante do que quantificar os clientes envolvidos no processo de transformação, é preciso qualificá-los para entender os seus reais motivadores e necessidades. Exemplificando de forma prática, em um determinado contexto de resolução de um problema, tão importante quanto elencar que 70% de um grupo reclama da morosidade da navegação na tela de um sistema, é identificar que esse grupo pertence à faixa etária de 15 a 24 anos, geração que se consolidou em meio à tecnologia e que é ávida por soluções fáceis e rápidas.

Conforme Pagani (2018), com o *design* é possível trazer uma percepção mais humana na resolução de problemas, preocupando-se depois com as ferramentas. Como já dito anteriormente, o *Design* foca nos indivíduos para poder solucionar um determinado problema, estabelecendo novas formas de pensamento para buscar compreender a seguinte série de fatores contextuais acerca de uma solução: quem são os indivíduos a quem estamos atendendo? Quais são as suas necessidades e seus desejos? Como eles vivem? Quais são as suas experiências pessoais?

Em resumo, o *Design Thinking* tem como base o processo dos designers de entender quais são as reais necessidades do público-alvo e os recursos disponíveis para integrá-los na criação de produtos que atendam às suas necessidades de um modo tecnologicamente e comercialmente viável (BROWN, 2010).

2.2. Etapas do *Design Thinking*

A metodologia é faseada em etapas que estabelecem a postura dos envolvidos quanto ao pensamento para identificar, propor, testar e implantar as soluções para as dores do cliente.

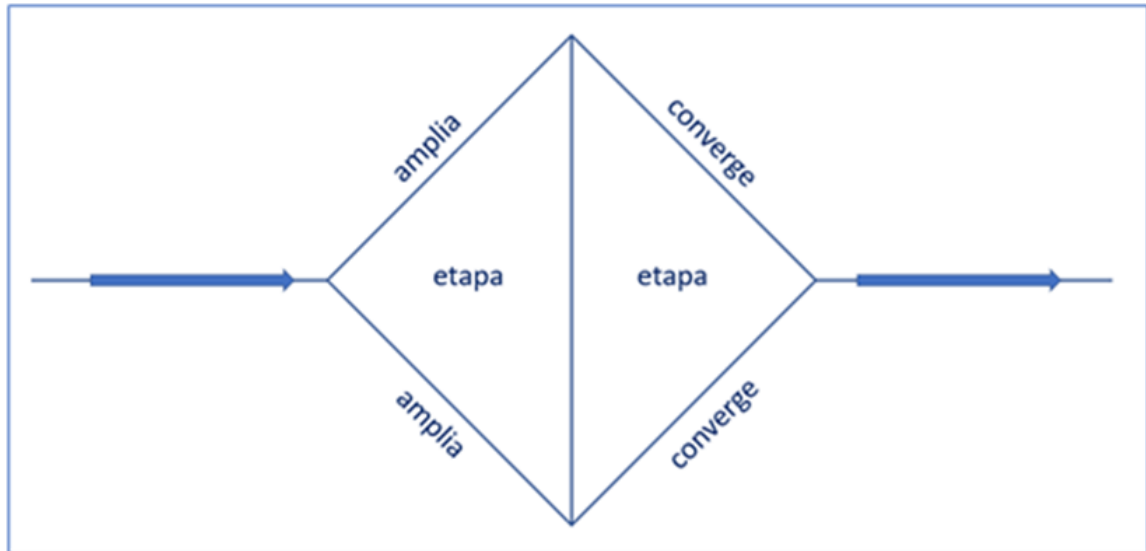
Figura 1 – Etapas do *Design Thinking*.



Fonte: Objetivo (2020).

Para cada etapa, há a representação gráfica de um “diamante”, que estabelece o momento em que as equipes poderão abstrair ou convergir para determinar as soluções, a exemplo do contexto do problema da reclamação dos usuários com a morosidade da tela: em caráter de *brainstorming* e pensando de forma ampla, uma possível solução seria a implementação de biometria de voz, que identificaria a voz do usuário estabelecendo a sua autenticação e obedeceria aos seus comandos, todavia, conforme eventuais restrições de custos, em caráter de convergência, a solução mais palpável seria revisar e enxugar as funcionalidades em tela para aprimorar sua navegação (SOUZA et al. 2020).

Figura 2 – O conceito do diamante.



Fonte: Próprios Autores.

Segundo SOUZA (2018), existem três macros etapas que estabelecem a dinâmica da ferramenta, as quais estão descritas a seguir:

a) Descoberta

Na primeira etapa, há uma imersão no contexto do problema a ser resolvido. Por meio da empatia, os clientes e demais envolvidos no processo são ouvidos, e o conceito de “Persona” é elencado, dessa forma declaram suas reais dores, anseios, expectativas e demais sentimentos relacionados para se entender o cenário. A Persona, portanto, será considerada durante todas as etapas da metodologia para que haja alinhamento das soluções propostas com suas necessidades.

b) Ideação

Na segunda etapa, o objetivo é gerar ideias inovadoras por meio de brainstormings para que haja o atendimento da expectativa da Persona. Conforme mencionado anteriormente, não há limites e restrições para a etapa em seu momento de ampliação.

c) Experimentação

Na terceira etapa, o momento de convergência estabelece que as soluções elencadas sejam consideradas conforme o crivo do projeto e suas restrições, tais quais custos, prazos, culturas etc. A partir desse momento, as soluções são construídas em caráter de protótipo, ou seja, com funcionalidades sólidas, mas em caráter preliminar de utilização e passiva de ajustes. A etapa dos testes estabelecerá as alterações necessárias até se chegar no produto, cuja satisfação será medida pelo cliente.

Conforme Vianna et al. (2012), o processo de *Design Thinking* é formado por três etapas descritas por imersão, ideação e prototipação e que, segundo o autor, não devem ser realizadas de forma linear, mas sim como partes entrelaçadas de um todo em que cada etapa permeia a outra. Stickdorn (2014) complementa que a cada passo, pode ser necessário retroceder uma etapa ou até começar tudo do zero. Como pode ser visto, cada autor tem uma nomenclatura diferente para cada passo do design, porém apesar dessa diferença de nomes todos seguem o mesmo processo.

2.3. Cronograma de Planejamento

Para um melhor progresso das tarefas, foi criado um cronograma para realização dos tópicos abordados no trabalho. A separação das etapas foi mensal, em cada mês realiza-se uma tarefa.

QUADRO 1 – Cronograma de pesquisa da fase 2

ETAPAS DA FASE 2	Meses referentes ao Ano de 2022							Meses referentes ao Ano de 2023						
	6	7	8	9	10	11	12	1	2	3	4	5	6	7
Conversas iniciais para formação do novo grupo														
Reunião com coordenador de TG e escolha do tema														
Entrevistas com público-alvo e levantamento de requisitos														
Pesquisa Teórica														
Escrita TG														
Correções														
Qualificação														
Correções da banca														
Implementação do <i>frontend</i> e <i>backend</i>														
Testes com o público-alvo														
Escrita do TG														
Defesa TG2														
Correções e entrega da versão final														

Fonte: Próprios Autores (2022).

3. REFERENCIAL TEÓRICO

Neste segmento serão abordados os principais conteúdos pesquisados para sustentação teórica e construção do *software*.

3.1. Engenharia de *Software*

Conforme Sommerville (2011), a Engenharia de *Software* é basicamente um princípio da Engenharia que zela por todas as fases de um processo de desenvolvimento de sistemas. Seu objetivo é ajudar no âmbito profissional, utilizando de técnicas que ajudam no parâmetro, no planejamento e no avanço desse projeto. Essa metodologia é aplicada em todo processo de *software*, desde o levantamento de requisitos até o programa entregue ao cliente.

O autor ainda mostra que a Engenharia é necessária por dois motivos:

- Com a crescente mudança para o ambiente digital, sistemas seguros, acessíveis e rápidos devem, cada vez mais, fazer parte do dia a dia, pois há uma grande dependência deles por parte da sociedade atual.
- Além de ter um custo baixo, a metodologia é usada para um projeto completo de criação de *software* e não algo pessoal. O custo é referente a uma alteração de um sistema por outro.

Para Falbó (2014), a Engenharia de *Software* é uma área da Ciência da Computação que gerencia projetos através de técnicas teóricas e práticas visando a qualidade no processo de *software*, aparelhado com a produtividade e a organização.

O autor também aborda que o problema para ser solucionado deve ser separado e avaliado em pequenas partes, e para cada problema, uma solução deve ser proposta. Já com cada parte solucionada, é fundamental juntar as soluções em uma só, juntamente com uma arquitetura estabelecida para o apoio da resolução como um todo, tendo nessa arquitetura várias ferramentas.

A Engenharia de *Software* “é uma tecnologia em camadas”, sendo sustentada com enfoque na qualidade, desenvolvendo o sistema de forma lógica e no tempo correto. As ferramentas utilizadas suportam para o processo e para o método de forma integradas, as informações elaboradas são usadas entre elas, garantindo um suporte ao desenvolvimento de programas (PRESSMAN, 2011).

Para Pressman (2006, p. 13) “o intuito da engenharia de *software* é fornecer uma estrutura para a construção de *software* com alta qualidade”.

Pressman (2011, p. 32) complementa também que:

O objetivo da engenharia de software moderna é o de “elaborar metodologias baseadas na noção de evolução”; isto é, na noção de que os sistemas de software modificam-se continuamente, novos sistemas são construídos a partir dos antigos e... Todos devem interoperar e cooperar um com o outro”.

De acordo com Sommerville (2011), existem algumas bases que todo tipo de desenvolvimento de sistema utiliza na Engenharia de *Software*:

- A elaboração do projeto deve ser gerenciada e compreendida, contendo ideias transparentes do que será feito e um prazo de entrega. Ressaltando que em *softwares* distintos são usados processos também distintos.
- O *software* a ser desenvolvido deve ser performático e confiável, protegido de invasões externas, fácil usabilidade e intolerante com as falhas.
- É essencial descobrir juntamente ao cliente quais os requisitos e especificações o *software* deve conter, para ser útil e entregue no prazo estipulado.
- Utilização dos recursos com o máximo proveito, propondo reescrever um código já pronto ou desenvolver um novo.

Esses fundamentos são bastante relevantes para se conhecer as vertentes básicas de um projeto de *software*.

3.1.1. Modelo de processo genérico

A metodologia de processo é o fundamento da engenharia de *software*, são pequenas porções de atividades estruturais aplicadas em todo projeto de sistemas, “independentemente de tamanho e complexidade”. Esse processo genérico é dividido em cinco atividades (PRESSMAN, 2011):

- Comunicação: descobrir quais as necessidades das partes interessadas para definição das funcionalidades e parâmetros do *software*.
- Planejamento: processo simplificado em forma de mapa, ou seja, um plano de projeto para ajudar a equipe a seguir um caminho confiável, buscando recursos necessários, possíveis riscos e trabalhar em cima de um cronograma.

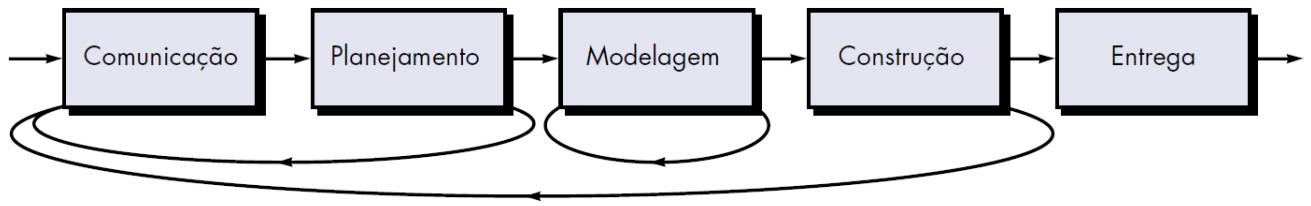
- Modelagem: criação de um esboço, para se ter uma ideia do produto como um todo, para descobrir suas características, arquitetura e se atenderá a necessidade do *software*.
- Construção: o desenvolvimento em si, criação de códigos e validação por meio de testes.
- Emprego: *software* pronto e entregue ao cliente, recebendo dele um *feedback* contendo informações e possíveis mudanças.

Figura 3 – Fluxo de processo linear.



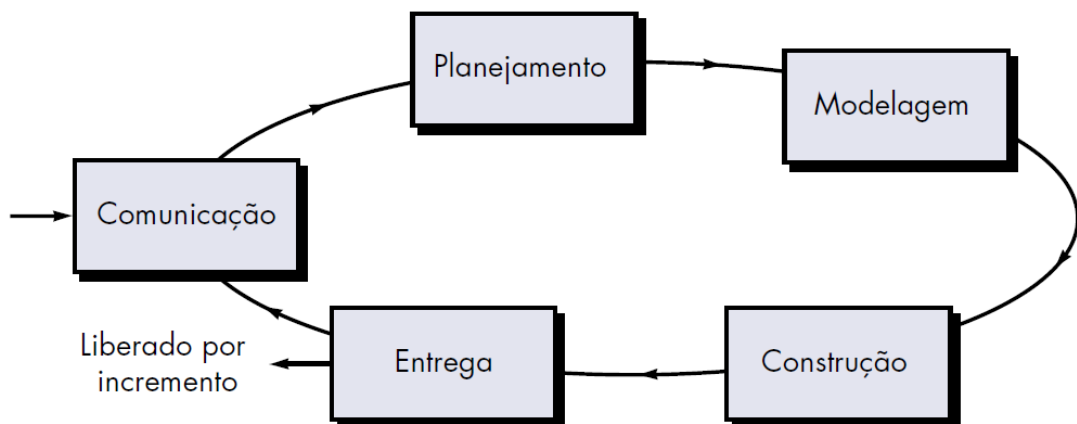
Fonte: Pressman (2011).

A figura 3 cita o processo em sequência, começando pela comunicação e terminando com o emprego.

Figura 4 – Fluxo de processo iterativo.

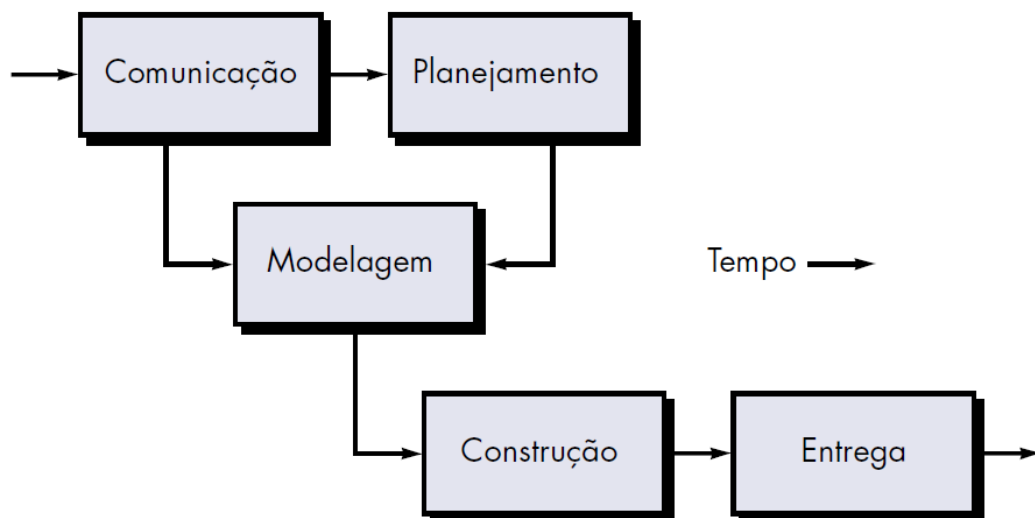
Fonte: Pressman (2011).

Na figura 4 é o mesmo processo, porém iterativo, ou seja, há a repetição de uma ou mais atividades antes de ir para a próxima atividade.

Figura 5 – Fluxo de processo evolucionário.

Fonte: Pressman (2011).

Na figura 5 o fluxo passa a ser de processo evolucionário, passa a executar as atividades de forma “circular”.

Figura 6 – Fluxo de processo paralelo.

Fonte: Pressman (2011).

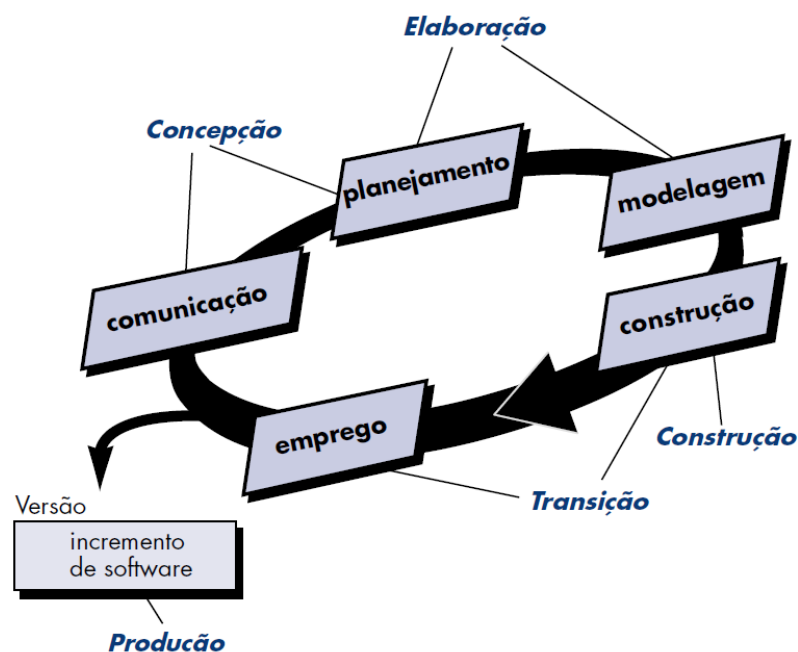
E por último na figura 6, são realizadas uma ou mais atividades em paralelo, ou seja, duas ou mais atividades podem ser executadas ao mesmo tempo.

3.2. RUP

De acordo com Sommerville (2011), o *Rational Unified Process* – RUP é um modelo de processo moderno, reunindo elementos dos processos genéricos, boas práticas na identificação e no projeto, apoiando a prototipação e a entrega incremental. Para Pressman (2011), é uma metodologia para a engenharia de software orientada a objetos utilizando a UML – uma linguagem de modelagem unificada.

Na fase da concepção, há a identificação de todas as entidades externas que irão interagir com o sistema e definir as interações. Na fase de elaboração, o objetivo é desenvolver uma compreensão do maior problema, tendo ao fim, um modelo de requisitos para o sistema. Na terceira fase de construção, o projeto, a programação e os testes são envolvidos, as partes do *software* são desenvolvidas em paralelo e integradas. Na fase de transição, há uma transferência do *software* da comunidade que desenvolve para a comunidade de usuários, funcionando em um ambiente real (SOMMERVILLE, 2011). Conforme Pressman (2011), ele inclui a fase de produção, que é a monitoração do *software* continuamente, disponibilizando suporte operacional, solicitações de mudanças e um relatório de falhas.

Figura 7 – O Processo Unificado.



Fonte: Pressman (2011).

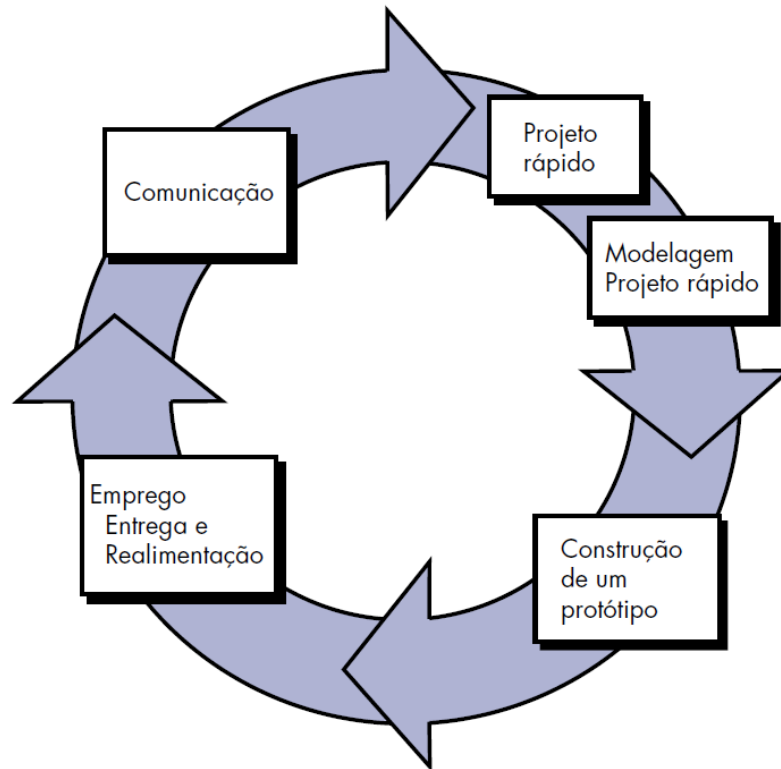
Na figura 7 é mostrado o processo de RUP e todas as suas fases.

3.3. Prototipação

A maioria das partes interessadas de um projeto, acha complicado visualizar o produto em si quando o levantamento de requisitos é feito. Por isso, nesses casos, um protótipo é desenvolvido para ficar mais fácil encontrar problemas e sugerir como os requisitos podem ser melhorados. Um protótipo é uma versão inicial do processo de desenvolvimento, ele torna os requisitos mais reais e diminui as lacunas de entendimento. Ele também permite capturar as reações iniciais do usuário em relação ao sistema (FALBÓ, 2014).

Segundo Pressman (2011), a prototipação auxilia os interessados a compreender melhor o que está para ser construído, representação dos aspectos de um *software* em um projeto rápido visualizados pelo usuário final. Esse protótipo é empregado e avaliado pelos envolvidos, que oferecerão *feedbacks* para aprimoramento dos requisitos. O protótipo pode servir como um “primeiro sistema” atuando como um mecanismo para identificar os requisitos do *software*. A maioria dos protótipos, num primeiro momento, são descartáveis, mas há a possibilidade de aprimoramento e devolver uma versão redesenhada, podendo se tornar o sistema real.

Figura 8 – O paradigma da prototipação.



Fonte: Pressman (2011).

O paradigma da prototipação começa na fase da comunicação, realiza-se uma reunião com as partes interessadas para determinar os objetivos gerais do sistema, apontar os requisitos já conhecidos e mapear os lugares que necessitam de uma melhor descrição (PRESSMAN,2011).

Na opinião de Sommerville (2011), a prototipação é:

Nessa abordagem para validação, um modelo executável do sistema em questão é demonstrado para os usuários finais e clientes. Estes podem experimentar o modelo para verificar se ele atende as suas reais necessidades.

3.4. Banco de Dados

Para Date (2004), o banco de dados opera como um repositório em que uma coleção de dados é mantida. É um sistema informatizado no qual os dados são registrados e condicionados ao armazenamento.

Date (2004, p. 26) define que:

Basicamente nada mais é do que um sistema de armazenamento de dados baseado em computador; isto é, um sistema cujo objetivo global é registrar e manter informação. Esta informação pode ser qualquer uma considerada significativa à organização servida pelo sistema – em outras palavras, qualquer uma necessária ao processo de decisão da gerência daquela organização.

Conforme Elmasri e Navathe (2011), todos os dados para serem condicionados ao armazenamento precisam ter uma significação implícita e concernente entre si, isso faz com que se forme um banco de dados.

3.4.1. Banco de Dados Relacional

Segundo Silberschatz; Korth; Sudarshan (2006) o banco de dados relacional é constituído por uma coleção de tabelas nas quais se armazenam as informações e a relação entre elas. Tabelas em que cada uma contém diversas colunas com nomes únicos.

Esse modelo é baseado em registro e Silberschatz; Korth; Sudarshan (2006, p. 7) explicam que:

Os modelos baseados em registros possuem esse nome porque o banco de dados é estruturado em registros de formato fixo de vários tipos. Cada tabela contém registros de um tipo específico. Cada tipo de registro define um número fixo de campo, ou atributos. As colunas da tabela correspondem aos atributos do tipo de registro.

Conforme Garcia e Sotto (2019) os bancos de dados que utilizam do modelo relacional têm impasses atrelados substancialmente atrelados à escalabilidade necessária em que os sistemas são distribuídos, sistemas esses que geram uma quantidade volumosa de informações diariamente através de sistemas que não são unicamente fechados.

Aquino e Mello (2021) por causa dos gastos gerados com o controle e densidade dessas informações, o modelo relacional de banco de dados não é a melhor solução para lidar com esses tipos de dados que são altamente heterogêneos.

3.4.2. Banco de Dados Não Relacional

Segundo Scardoelli e Pinto (2020), o Banco de Dados não Relacional foi criado para reparar o aumento da performance, a escalabilidade e disponibilidade no espaço da Internet por conta da larga diversidade de velocidade de conexão e segurança das Redes.

A palavra NoSQL (*Not Only SQL*), foi uma solução para Banco de Dados firmada em arquitetura relacional, mas sem conter a interface SQL (*Structured Query Language*). Essa solução buscava disponibilizar um modelo escalável mais acessível, regular sistemas distribuídos e armazenar dados sem uma estrutura definida. (SILVA; FERREIRA 2017).

Havendo uma escalabilidade horizontal, os custos e dificuldades desse modelo são relativamente mais baixos comparados ao modelo relacional, pois se baseiam em sistemas distribuídos. Como existem vários modelos de banco de dados NoSQL, como por exemplo o DynamoDB da Amazon, o MongoDB e o Cassandra, há uma certa dificuldade em se comparar cada um, pois cada um é fundamentado em um diferente tipo de aplicação (SCARDOELLI; PINTO, 2020).

3.5. Aplicação Web

Operando de forma a conectar diversos computadores ao redor do mundo à rede mundial, a *web* faz alusão a uma teia de aranha interligando essas máquinas à rede.

Segundo Silva (2007) em 1990, Sir Tim Berners-Lee, elaborou seu primeiro protótipo de navegador que visava rodar em computadores. Em 1992, com o objetivo de colaborar com outros cientistas em escala mundial, de maneira a compartilhar seus textos e trabalhos eletronicamente, Tim, criou a *web*.

Para Santos (2021) uma aplicação *web* é gerada em duas partes distintas que se conversam. A interface, pela qual são apresentadas as páginas interativas na qual o usuário irá fazer a navegação, é a primeira parte, o *frontend*. A segunda parte, o *backend*, é responsável por toda a lógica, regras de negócios do sistema e o tratamento de dados.

3.6. HTML

De acordo com Silva (2011) o HTML (*Hypertext Markup Language*), conectando-se à rede por HiperTexto, tem por finalidade oferecer documentos por meio da internet sendo ele uma linguagem de marcação de Hipertexto aplicada para navegadores da *web*.

Conforme Faulkner et al. (2014), o HTML funciona empregando marcações chamadas Tags, em que o conteúdo exibido pode ser organizado de maneira mais proveitosa para o cliente *web*. Utilizando o HTML é permitido que na interface seja disposta a interação como acionamento de botões, publicação de dados como tabelas, arquivos de vídeo, arquivos de áudio e textos.

3.7.CSS

CSS é a abreviação do termo em inglês *Cascading Style Sheet*, que é traduzido para o português como folhas de estilo cascata. Essa linguagem é focada na customização de *desings* dos elementos presentes de uma página (ALURA, 2018).

Em 1994, Hakim Wiuw Lie o criou com a problemática de se ter páginas *web* com fácil manutenção e que fossem mais dinâmicas e leves, fornecendo maior acessibilidade aos usuários (GOMES, 2010).

Para Silva (2013), somente com a estruturação feita pelo HTML, sem estilizações providas dessa linguagem, é fundamental a utilização do CSS nas aplicações *web*, com a funcionalidade de colocar toda a parte visual em serviços *web*.

3.8. JavaScript

Segundo Alura (2018) e Silva (2010), a linguagem foi criada com o intuito de que a possibilidade de desenvolver páginas *web* com mais interatividade fosse uma verdade. Foi em 1995, que a primeira versão do *JavaScript* surgiu. Criada pela Netscape em conjunto com a Sun Microsystem, a linguagem foi implementada em 1996.

Conforme Silva (2010) o *JavaScript* foi criado para rodar do lado do cliente, e não do lado servidor, ou seja, as funcionalidades e interpretação desses *scripts* desenvolvidos nessa linguagem ficam por conta da necessidade de apenas um navegador e do interpretador de *JavaScript* hospedado no mesmo.

Para Flanagan (2013), o *JavaScript* se tornou tão popular, que hoje se tem a utilização dessa linguagem em diversos aparelhos que possuem o interpretador *JavaScript* e isso levou essa tecnologia a ser uma das mais utilizadas da história dentre as linguagens de programação *web*.

Flanagan (2013, p. 1), define que:

JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e *JavaScript*, para especificar o comportamento delas.

3.8.1. React

Segundo Li (2016), o Facebook/Instagram foi responsável pela criação do componente de *User Interface (UI)*, com código aberto que foi extremamente bem recebido pela comunidade de desenvolvedores em *JavaScript*, que por sua vez

passavam pelos mesmos problemas, problemas estes que a biblioteca de React havia sido projetada para solucionar.

Para Li (2016), o primeiro processo de otimização de tempo para a execução do React é realizado com a *Document Object Model (DOM)*. Após este processo de enviar componentes da interface web para o DOM virtual do React, em um momento mais adequado para otimizar desempenho de modificações feitas, é realizada a sincronização da UI com a DOM real do navegador.

3.8.2. NextJS

Johansson (2021) diz que o NextJS é um *Framework* de React com o principal objetivo de facilitar o desenvolvimento de aplicações estáticas e fornecer o roteamento automático do servidor, utilizando nome dos arquivos e as páginas que contêm esses componentes.

Segundo sua documentação (NEXTJS, 2021), esclarece os principais objetivos do *framework* listando as soluções para algumas das principais dificuldades em um desenvolvimento *web* como sistema de roteamento básico, páginas com renderização por parte do servidor, ambiente de desenvolvimento com respostas imediatas às mudanças.

3.8.3. Express.JS

Conforme Mardan (2014), o Express.js é um *framework web* responsável pela conexão dos componentes, além de diminuir códigos semelhantes para realização de tarefas padrões, como o gerenciamento de sessões, análise de cookies ou requisições HTTP.

Seu funcionamento está na tarefa de identificar novas requisições, e processá-las para cada rota que esteja preparada para receber, permitindo assim que seja feita a ação com base na informação recebida, segundo Mardan (2014). Um exemplo bem clássico de como é utilizado esse recurso é para identificar usuários autenticados na aplicação e assim permitir a continuação de sua utilização.

3.9. TypeORM

De acordo com o TypeORM (2021), o TypeORM é um ORM que pode ser executado em várias plataformas, incluindo *Browser*, *Cordova*, *Ionic* e *PhoneGap*. Além dessas, ele também pode ser executado em outras plataformas, como *Electron*, *Expo*, *NativeScript* e *React Native*. Ele pode ser utilizado tanto com *TypeScript* quanto

com *JavaScript*, desde que seja ES5, ES6, ES7 ou ES8, pois tem como objetivo oferecer suporte às funcionalidades mais recentes do JavaScript, permitindo o uso de recursos adicionais que auxiliam no desenvolvimento de qualquer tipo de aplicativo que utilize banco de dados. Isso inclui desde aplicativos menores com algumas tabelas até aplicações corporativas maiores com escala em vários bancos de dados.

Segundo o TypeORM (2021), o TypeORM suporta tanto o padrão Active Record quanto o padrão Data Mapper, diferenciando-se de outros ORMs *JavaScript* disponíveis no mercado. Isso possibilita escrever aplicativos de alta qualidade com um acoplamento simples, escalável e sustentável de forma mais produtiva. O TypeORM utiliza a biblioteca Sequelize como seu ORM, que simplifica a execução de consultas nos bancos de dados.

De acordo com Kriger (2023), trabalhar com bancos de dados relacionais pode ser altamente complexo e demorado. No entanto, o TypeORM resolve esse problema permitindo que os dados sejam trabalhados de forma mais abstrata, ou seja, sem a necessidade de escrever consultas SQL manualmente, pois oferece recursos de migração. Além disso, ele permite utilizar um modelo de programação orientado a objetos, fornecendo métodos de mapeamento de entidades que mapeiam as propriedades das entidades para colunas do banco de dados. Isso permite uma escrita de consultas mais dinâmica e descritiva, tornando o trabalho mais direto.

Kriger (2023) destaca que existem várias maneiras de instalar o TypeORM em um projeto Node.js, permitindo escolher a instalação de acordo com as necessidades. É possível utilizar o NPM, que instalará o TypeORM e suas dependências, ou o Yarn, que permite gerenciar projetos e executar os comandos "*npm install typeorm reflect-metadata*" e "*yarn add typeorm*" respectivamente. Dessa forma, o TypeORM melhora o processo de acesso aos dados, facilitando a escrita de consultas e o gerenciamento de relacionamentos.

3.10. Docker

De acordo com Gomes (2017), o Docker é uma plataforma de código aberto que permite criar, implantar e executar aplicativos em contêineres. Um contêiner é uma unidade isolada que contém todos os recursos necessários para executar um software, incluindo o código, as bibliotecas, as dependências e as variáveis de ambiente.

A ideia central do Docker é criar contêineres leves e portáteis que possam ser executados em qualquer ambiente, independentemente das diferenças nas configurações do sistema. Os contêineres do Docker são baseados na tecnologia de virtualização do sistema operacional, o que significa que eles compartilham o kernel do sistema operacional subjacente, mas são isolados e possuem recursos próprios.

Essa abordagem traz diversos benefícios, tais como:

- Portabilidade: Os contêineres Docker podem ser executados em qualquer ambiente compatível com Docker, seja um laptop, um servidor físico ou na nuvem.
- Consistência: O ambiente de execução dentro do contêiner é altamente consistente, independentemente do ambiente de hospedagem, o que facilita a implantação e evita problemas de compatibilidade.
- Escalabilidade: O Docker permite dimensionar facilmente seus aplicativos, criando e executando múltiplos contêineres simultaneamente.
- Isolamento: Cada contêiner Docker é isolado dos demais, o que evita conflitos entre aplicativos e melhora a segurança. (GOMES, 2017)

3.11. UML

Linguagem de Modelagem Unificada ou *Unified Modeling Language (UML)* é o nome dado para uma modelagem de *softwares* com paradigmas de orientação a objetos (ALURA, 2022).

Conforme Guedes (2009), deve ficar claro que UML não é uma linguagem de programação, e sim uma linguagem de modelagem com o principal objetivo de auxiliar na definição de características do sistema que será criado, assim como requisitos, comportamentos e a estrutura lógica, a dinâmica e as necessidades físicas para a implantação do *software*. Lembrando também que sua utilização não é um processo de desenvolvimento de sistema, estando aberta para ser utilizada do jeito que for mais adequado para o engenheiro responsável.

3.11.1. História da UML

Mencionado por Guedes (2009), o surgimento da UML aconteceu devido a consolidação de três tipos de métodos, o de Grady Booch, Ivar Jacobson e de James Rumbaugh, sendo cada um responsável respectivamente pelos métodos de Booch,

OMT (*Object Modeling Technique*) e OOSE (*Object-Oriented Software Engineering*). Até a década de 1990, as modelagens orientadas a objetos utilizavam algum desses três métodos, por serem os principais na área de desenvolvimento.

Inicialmente, conforme Guedes (2009), a junção entre métodos começou com o de Booch e o OMT, surgindo assim no final de 1995 o Método Unificado. Pouco tempo depois, Rumbaugh juntou-se Booch e Jacobson na Rational Software, incorporando o OOSE à nova metodologia. Os três amigos, como eram conhecidos popularmente, fizeram oficialmente o lançamento da primeira versão da UML.

Logo depois, muitas empresas da área começaram a sugerir para melhorar a linguagem e em 1997, a OMG (*Object Management Group* ou Grupo de Gerenciamento de Objetos) adotou como linguagem padrão para modelagem (GUEDES, 2009, p. 20).

3.11.2. Importância da modelagem

Para Guedes (2009) por mais simples que um projeto seja, para iniciar a criação de um sistema, é necessária a modelagem, isso porque *softwares* tendem a ser algo dinâmico e manter um crescimento contínuo em tamanho, complexidade e abrangência. Alguns dos principais fatores para essas frequentes atualizações, são principalmente solicitações de clientes para melhorias ou modificações, novas estratégias para adequar-se ao mercado e até mesmo alterações de leis.

Segundo Guedes (2009, p. 21) são por estes motivos que a documentação de um sistema precisa estar muito bem detalhada, precisa e atualizada, para que em momentos de correções ou modificações, não apareçam novos erros.

3.11.3. Tipos de Diagrama

Para Guedes (2009), o motivo de haver muitos diagramas, é pelo fato de que assim um sistema pode ser modelado e analisado em diversos aspectos, tendo uma visão mais ampla e fazendo com que cada um se complete. Permitindo assim que um *software* seja analisado em níveis externos, seja apresentada uma visão ou utilização da ferramenta pelo usuário, como níveis mais profundos, apresentando detalhamento de processos e informações técnicas.

Existem duas classificações para diagramas, sendo estruturais e comportamentais. Para Booch; Rumbaugh; Jacobson (2005, p. 96), os diagramas estruturais existem com o propósito de visualizar, especificar, construir e documentar

os pontos estáveis do *software*. Diagramas comportamentais têm o mesmo objetivo, mas para todos os aspectos dinâmicos de um sistema, sendo todos aqueles que podem sofrer alterações.

3.12. Identidade Visual

Segundo Strunck (1989) a identidade visual consiste no conjunto de elementos que formalizam a personalidade de um nome, ideia, produto ou serviço. Para Strunck (2001) o símbolo é um sinal gráfico com o intuito de identificar um nome, ideia, produto ou serviço. Os mesmos podem ser classificados em dois grandes grupos, abstratos e os figurativos. Símbolos abstratos não possuem nenhum significado à primeira vista. Os símbolos figurativos por sua vez, podem ser de três naturezas. Baseados em ícones: são aqueles que representam fielmente o que desejam apresentar. Fonogramas: formados apenas por letras. Ideogramas: representam ideias ou conceitos pelo desenho.

Segundo Raposo (2009), a representação visual de uma marca depende totalmente do foco que ela possui, trazendo a qualidade de um produto ou serviço prestado pela organização. Para Raposo (2009), é de extrema importância que seja criada uma identidade visual da marca com cores, símbolos, logotipo e alfabeto. Conseguindo trazer uma identidade única para aquilo que se refere.

Segundo Farina, Perez, Basto (2011), as cores possuem muita influência em um ser humano, podendo criar sentimentos, impressões, atração e até mesmo sedução para a apresentação de produtos, embalagens, logotipos, cartazes etc. Para Farina, Perez, Basto (2011), a cor é utilizada para criar sensações conscientes por uma energia radiante que estimula a retina.

4. DESENVOLVIMENTO

Nesta seção, seguindo as metodologias *Design Thinking*, suportadas pela pesquisa bibliográfica, serão apresentadas as etapas realizadas para o desenvolvimento deste projeto.

4.1. Imersão

O projeto foi determinado pelo professor e orientador Marcio Rodrigues Sabino, pois percebeu-se que todo processo de gerenciamento dos trabalhos de graduação da Fatec Mogi Mirim, Arthur de Azevedo, eram realizados de forma manual, e para torná-lo automatizado, o sistema foi dividido em partes, sendo elaborada a primeira fase pelo aluno Pedro Luis Berbel dos Santos.

Foram realizadas reuniões com o orientador para compreender a necessidade e os principais motivos desse problema, sendo identificado o retrabalho, causado por erros humanos na elaboração das etapas, mensagens dispersas e falta de comunicação entre as partes, desperdiçando tempo e recursos, provocando atrasos.

Durante esse período também tiramos as dúvidas de como procedermos com o desenvolvimento da segunda fase do trabalho juntamente com o orientador Marcio, e com o aluno Pedro, separadamente, buscando um entendimento completo para elaborarmos a continuação do sistema visando realizar as tarefas abordadas com qualidade, acessibilidade e o máximo de performance desse processo automatizado.

4.2. Ideação

Após realizadas reuniões com o aluno Pedro e o orientador Marcio, foi possível a identificação de quais funcionalidades seriam abordadas na segunda fase do projeto. Com isso, um levantamento de requisitos foi elaborado apresentando a usabilidade diária de cada função.

As funcionalidades identificadas foram:

- Acrescentar o cadastro de Instituição para que seja possível a utilização nas demais Fatecs do Estado de São Paulo;
- Realizar o *upload* da ficha de formalização no sistema para possível consulta posterior;
- Adicionar a lista de grupos orientados para a visualização do orientador;
- Realizar o *upload* do Trabalho de Graduação, na versão final, e documentos extras, como a autorização para publicação do trabalho;

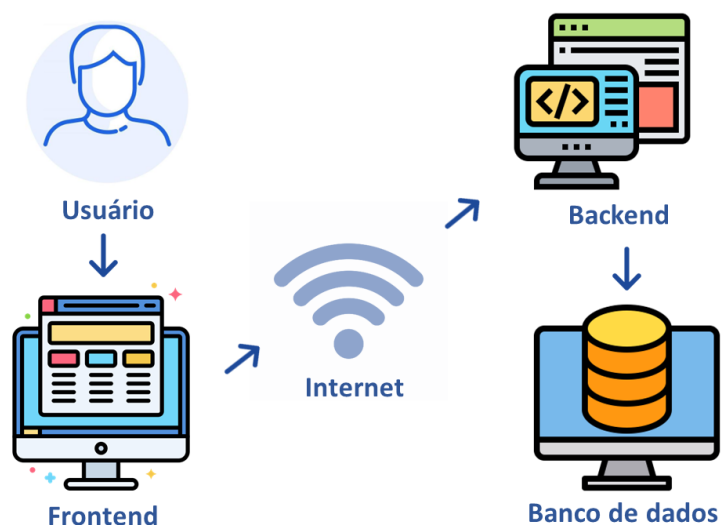
- O sistema precisa mostrar ao orientador a opção de aceite ou recusa de ser orientador de um certo grupo;
- O orientador definirá um cronograma ao grupo orientado;
- O sistema deverá realizar o download dos documentos finais;
- O orientador poderá marcar a banca, escolhendo os professores, o grupo e a data.

No tópico abaixo será apresentado a prototipação da segunda fase do projeto para alcançar o objetivo deste trabalho.

4.3. Prototipação

Essa etapa de prototipação foi dividida em *backend* e o *frontend*, seu funcionamento em conjunto formam o que conhecemos da aplicação, como é representado na figura 9.

Figura 9 – Arquitetura do sistema completo.



Fonte: Próprios Autores.

A figura 9 mostra a arquitetura do sistema, o usuário acessa o site do projeto por um protocolo de comunicação e, por meio da internet, ele busca os “bastidores” desse site e mostra os dados solicitados pelo usuário armazenados em um Banco de Dados.

4.3.1. Backend

Neste projeto o desenvolvimento *backend* foi por meio da tecnologia *open source*, Node.js. Por ser uma tecnologia de código aberto, a quantidade de desenvolvedores ativos torna o avanço tecnológico mais avançado, além de soluções

de problemas e falhas de segurança serem mais rápidas. No desenvolvimento, foi usada a linguagem de programação Typescript, que se assemelha ao Javascript em múltiplos pontos, mas pela definição do tipo das variáveis fazer-se necessária, a assertividade desse ponto pode fazer com que erros relacionados ao que uma variável sem definição suporta sejam evitados.

Sua arquitetura é composta das rotas, *controller* e *services*. No *service* se encontram as regras de negócio e as funções de acesso ao banco de dados, buscando, salvando e alterando informações. As rotas fornecem acesso ao backend através das requisições REST. O fluxo basicamente é definido em cada rota aponta um *controller*, que receberá as requisições carregadas com as informações de entrada, e enviará para os *services* todos os dados relevantes para que determinada atividade possa ser executada. Essa camada também manda de volta ao *frontend* as respostas geradas.

Para o banco de dados, que por possuir autenticação por e-mail e senha e o registro de dados em si, adotou-se o Firebase, que é NoSQL. O banco de dados está na nuvem, que abrange a capacidade de proporcionar múltiplos acessos simultâneos e aumenta sua disponibilidade, além da sua administração poder ser feita pela internet, em qualquer lugar. Com o objetivo de tornar o projeto mais robusto e escalável, foi decidido incorporar o PostgreSQL como banco de dados relacional. Essa escolha estratégica permitirá o armazenamento eficiente dos dados de configuração, como tipos de usuários, instituições e cursos. O PostgreSQL oferece recursos avançados, como consultas complexas, integridade referencial e transações, que serão fundamentais para a gestão e organização dessas informações. Além disso, a escalabilidade proporcionada pelo PostgreSQL, por meio de recursos como particionamento e replicação, garantirá que o projeto possa lidar com o aumento contínuo do número de usuários e requisitos de armazenamento. A adoção do PostgreSQL como banco de dados relacional é uma medida estratégica que contribuirá para a flexibilidade e o crescimento sustentável do projeto.

4.3.2. Frontend

Para o *frontend* desse projeto utilizamos um framework do React.js, o Next.js. Essa escolha se deu pela simplicidade que é ofertada em relação ao desenvolvimento da navegação entre páginas diferentes e por facilitar a estruturação da página com a possibilidade de criação dos componentes visuais serem incluídos à estrutura do

HTML. Por conta deste código do componente ficar encapsulado, permite-se que tais componentes possam ser reaproveitados em diferentes pontos do código e serem representados como tag o que, visualmente, provoca a ideia de que esses componentes são do próprio HTML.

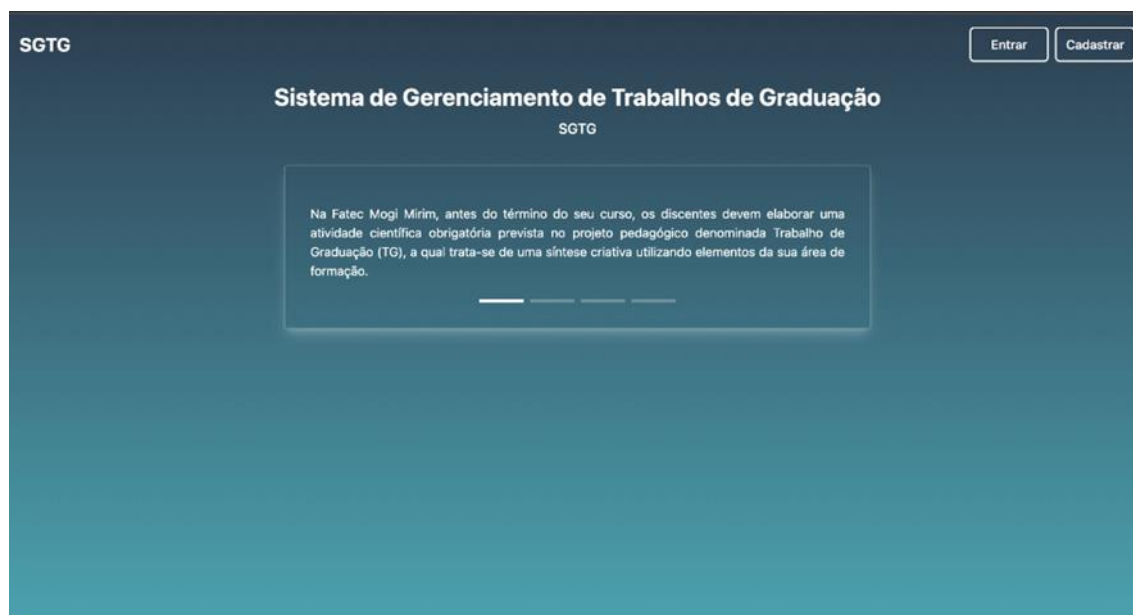
Foram usadas também entre as tecnologias web a linguagem JavaScript, HTML5 e o CSS3.

5. RESULTADOS

Neste capítulo, serão mencionados os resultados alcançados durante o desenvolvimento do projeto. Serão exibidas telas que demonstram o fluxo de utilização pelos usuários.

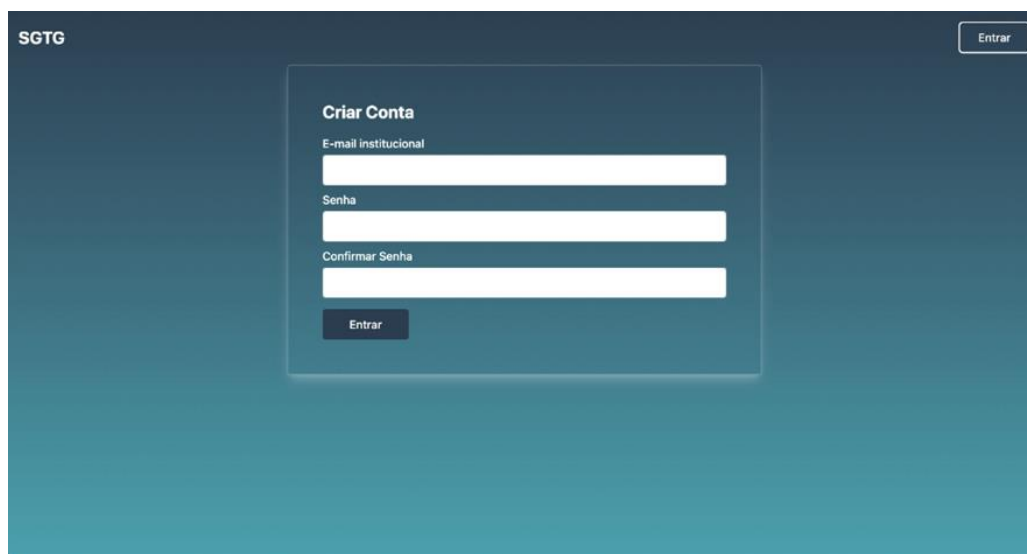
Na figura 10, é mostrado a tela inicial do projeto.

Figura 10 – Tela inicial do projeto.



Fonte: Próprios Autores.

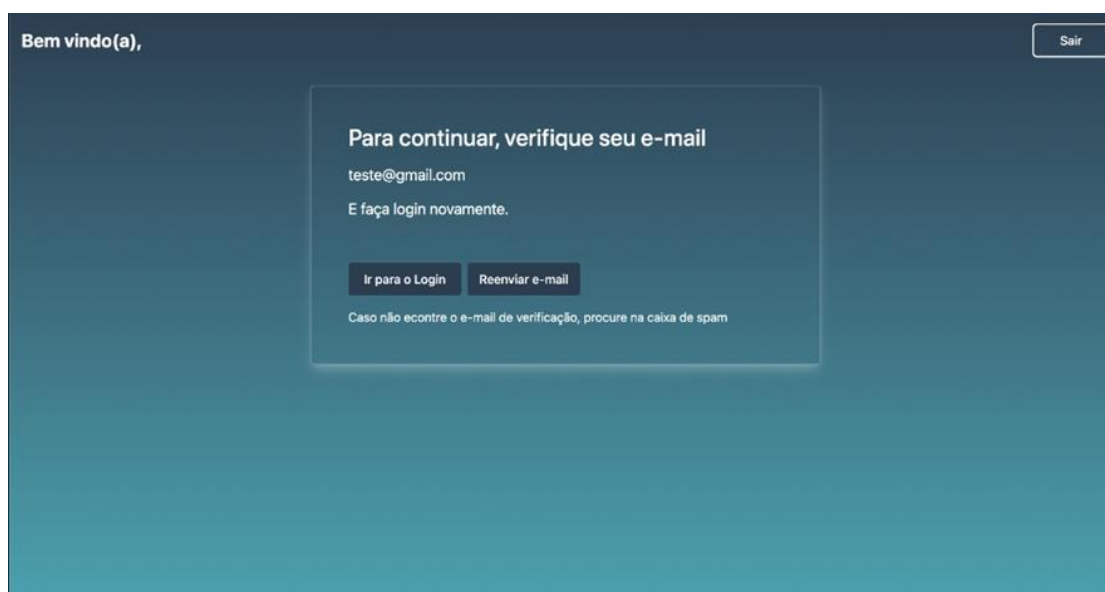
A figura 10 possui uma breve explicação sobre o sistema de gerenciamento de trabalhos de graduação. Composta por um pequeno banner centralizado, e dois botões na parte superior, no qual direciona para a tela de acesso do sistema ou a tela de registro de usuários. Todos os usuários iniciam sua experiência de acesso pela mesma tela, mas o que diferencia são os painéis de visualização conforme o tipo de usuário, seja ele alunos, orientadores, secretaria e administrativo. Na figura 11, é mostrado a tela de cadastro.

Figura 11 – Tela de cadastro.

The screenshot shows a registration form titled 'Criar Conta' on a dark blue background. The form is centered and contains three input fields: 'E-mail institucional', 'Senha', and 'Confirmar Senha'. Below the 'Confirmar Senha' field is a dark blue button labeled 'Entrar'. In the top left corner of the page, the text 'SGTG' is visible, and in the top right corner, there is a small button labeled 'Entrar'.

Fonte: Próprios Autores.

A Tela de cadastro deverá ser utilizada para a criação de conta de alunos e orientadores. Para identificar que o atual cadastro pertence ao orientador, é necessário que um administrador do sistema faça o primeiro registro pelo seu acesso (**Tela do administrador**), vinculando assim o e-mail desejado. Para o cadastro de alunos, nenhum processo prévio deve ser executado. Basta preencher os campos e seguir com o processo de verificação de e-mail conforme a figura 12. Na figura 12 é apresentado a tela de verificação de e-mail.

Figura 12 – Tela de verificação de e-mail.

The screenshot shows an email verification screen. At the top left, it says 'Bem vindo(a),'. At the top right, there is a button labeled 'Sair'. The main content is centered and includes the heading 'Para continuar, verifique seu e-mail', followed by the email address 'teste@gmail.com' and the instruction 'E faça login novamente.'. Below this are two buttons: 'Ir para o Login' and 'Reenviar e-mail'. At the bottom, a small note reads 'Caso não encontre o e-mail de verificação, procure na caixa de spam'.

Fonte: Próprios Autores.

Após realizar o registro, o usuário é direcionado para esta tela, onde exige que seja feita a confirmação do login no e-mail cadastrado. Caso o e-mail não seja recebido pelo usuário, poderá clicar na opção de reenviar e-mail para que seja encaminhado novamente. É recomendado seja verificada toda a caixa de e-mail, inclusive lixos eletrônicos e spam.

Na figura 13 é apresentado a tela de modelo de e-mail recebido para confirmação do acesso.

Figura 13 – Modelo de e-mail recebido para confirmar o acesso.

Olá, teste@gmail.com

Parece que uma verificação de senha para sua conta foi solicitada.

Se foi você, então clique no link abaixo para escolher uma nova senha:

[Verificar e-mail](#)

Se não foi você então descarte esse e-mail.

Obrigado!
Equipe SGTG

Fonte: Próprios Autores.

Este será o modelo de e-mail encaminhado pelo sistema, e para confirmação por parte do usuário, será necessário clicar em “Verificar e-mail”, e assim poderá retornar ao sistema e seguir com o login normalmente. Na figura 14 é mostrado a tela de Login.

Figura 14 – Tela de Login.

Fonte: Próprios Autores.

Após verificar o e-mail, o usuário deverá ir até a página de login do sistema, e preencher o e-mail e senha cadastrados anteriormente. Após preencher os campos e seguir, pressionando no botão de entrar, será direcionado para a tela de completar o cadastro.

Na figura 15 é apresentado a tela para completar o cadastro.

Figura 15 – Tela para completar o cadastro.

A interface de usuário para completar o cadastro no sistema SGTG. No topo, há o logotipo "SGTG", o texto "Bem vindo(a)," e um botão "Sair". O formulário principal, intitulado "Complete seu Cadastro", contém os seguintes campos e opções:

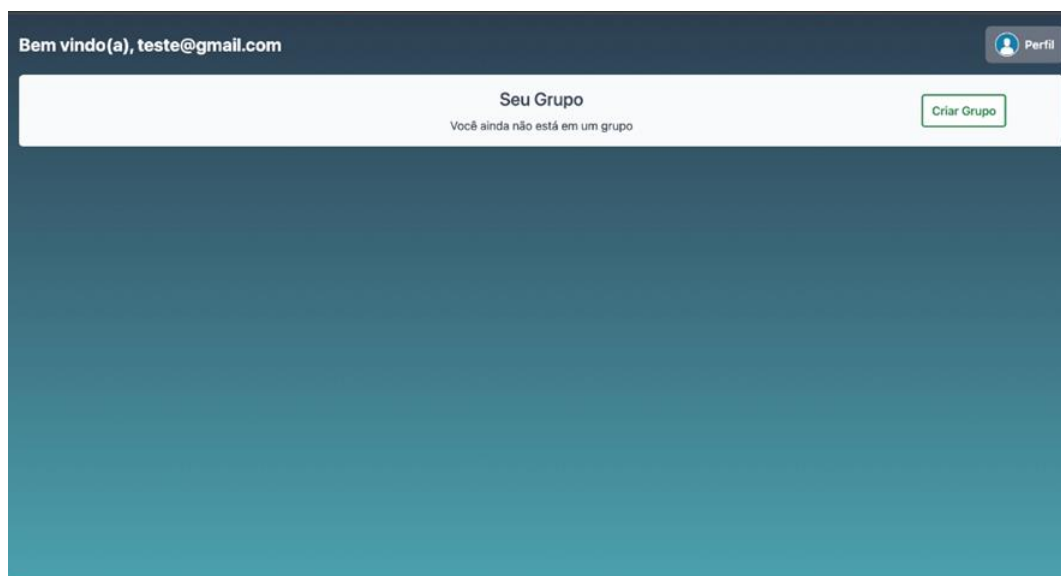
- Nome completo:** Campo de texto.
- Username:** Campo de texto com um ícone de @.
- E-mail institucional:** Campo de texto com o exemplo "teste@gmail.com".
- E-mail Secundário (Opcional):** Campo de texto com o exemplo "nome@example.com".
- Unidade de Ensino:** Menu suspenso com a opção "Selecione..." e uma seta para baixo.
- Curso:** Menu suspenso com a opção "Selecione..." e uma seta para baixo.
- Disciplina:** Menu suspenso com a opção "Selecione..." e uma seta para baixo.
- Celular (Opcional):** Campo de texto com o exemplo "(xx) x 3000-3000".
- Informações adicionais:**
 - Texto: "O seu número de celular não será exibido de forma pública, apenas para o seu grupo de trabalhos, caso permita."
 - ☐ Este celular é whatsapp.
 - ☐ Quero que meu celular fique visível para o meu grupo de trabalho.
 - ☐ Estou ciente e permito que estes dados sejam armazenados e compartilhados com a minha instituição de ensino.
- Botão:** "Completar o Cadastro".

No rodapé, há o copyright "© 2022 SGTG".

Fonte: Próprios Autores.

Fazendo o primeiro acesso, o usuário será direcionado para tela obrigatório de conclusão de cadastro. Nela será solicitado alguns dados extras, como o nome completo, um nome de identificação, um e-mail secundário, unidade de ensino da Fatec, o curso, a disciplina e um telefone celular. Na figura 16 é mostrado o Dashboard principal do aluno.

Figura 16 – Dashboard principal do aluno.



Fonte: Próprios Autores.

Após finalizar o complemento do cadastro, o aluno será direcionado para o dashboard principal onde será possível criar o grupo para iniciar o registro do trabalho de graduação conforme a figura 17.

A figura 17 mostra a tela de cadastro de grupo.

Figura 17 – Tela de cadastro de grupo.

Crie seu Grupo

Alunos

Aluno 1

E-mail

Preencha o e-mail institucional dos integrantes do seu grupo

Aluno 2 Opcional

E-mail Opcional

Aluno 3 Opcional

E-mail Opcional

Aluno 4 Opcional

E-mail Opcional

Trabalho

Tema

Proposta do Tema

Justificativas

1: Obrigatória

2: Obrigatória

3: Obrigatória

4: Opcional

5: Opcional

Resumo - Explicação do Tema (150 a 300 caracteres) 0

Descrição/Resumo

Documento do grupo

Upload or drag a file right here

Monografia

Upload or drag a file right here

Orientador

Professor

☐ Declaro estar ciente e concordar com as regras estabelecidas pela coordenação do Curso Superior de Tecnologia.

Criar Grupo

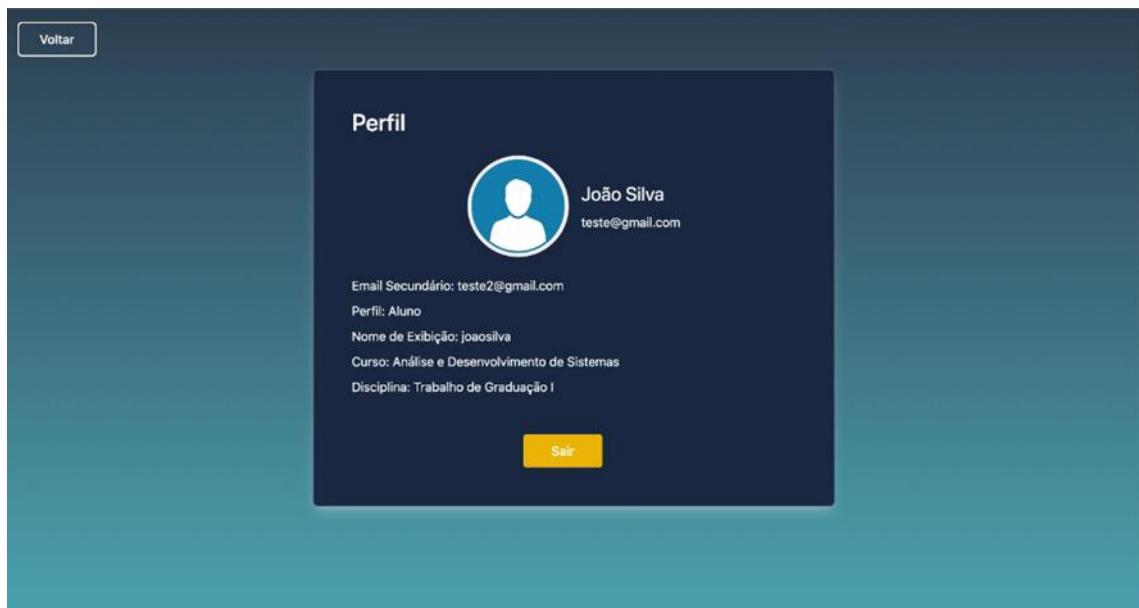
© 2022 SGT0

Fonte: Próprios Autores.

Nesta tela, será informado o e-mail dos integrantes do grupo e uma breve descrição sobre o trabalho, informando o tema, suas justificativas, o resumo e a formalização preenchida. Para que os integrantes do grupo recebam o convite (Conforme a figura 20), será necessário que todos os integrantes do grupo estejam com o cadastro completo no sistema.

Na figura 18, é mostrada a Tela de perfil.

Figura 18 – Tela de perfil.

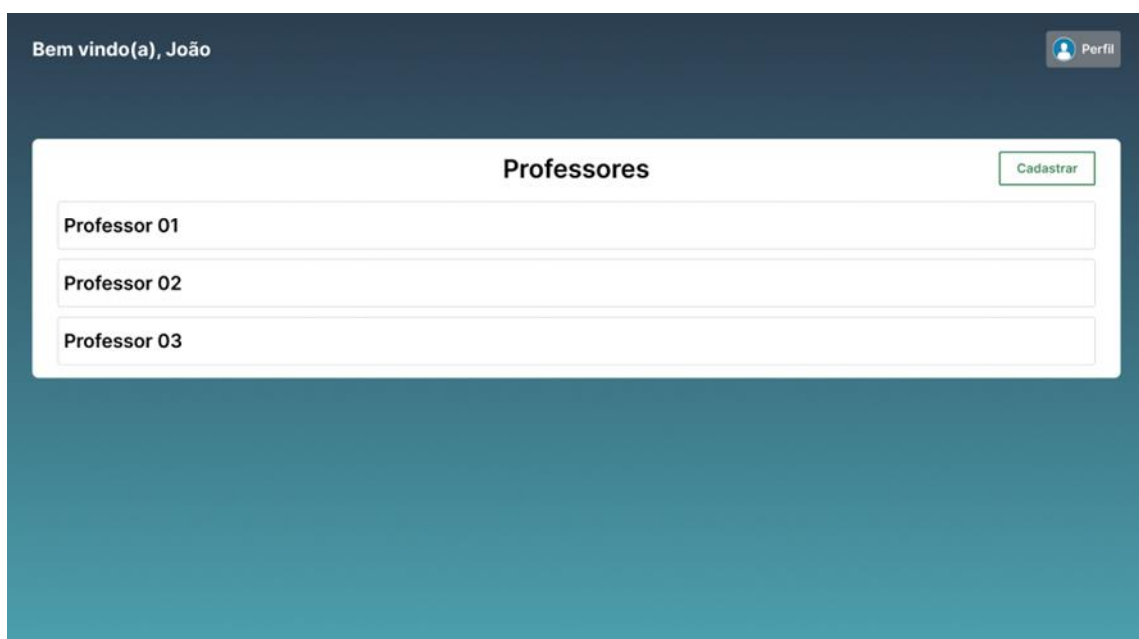


Fonte: Próprios Autores.

Ao clicar em perfil na parte superior direita, será redirecionado para essa tela, onde é apresentado os dados do usuário logado e a opção de sair, para caso queira encerrar a sessão.

Na figura 19 é apresentado o Dashboard da secretaria/coordenação.

Figura 19 – Dashboard da secretaria/coordenação.

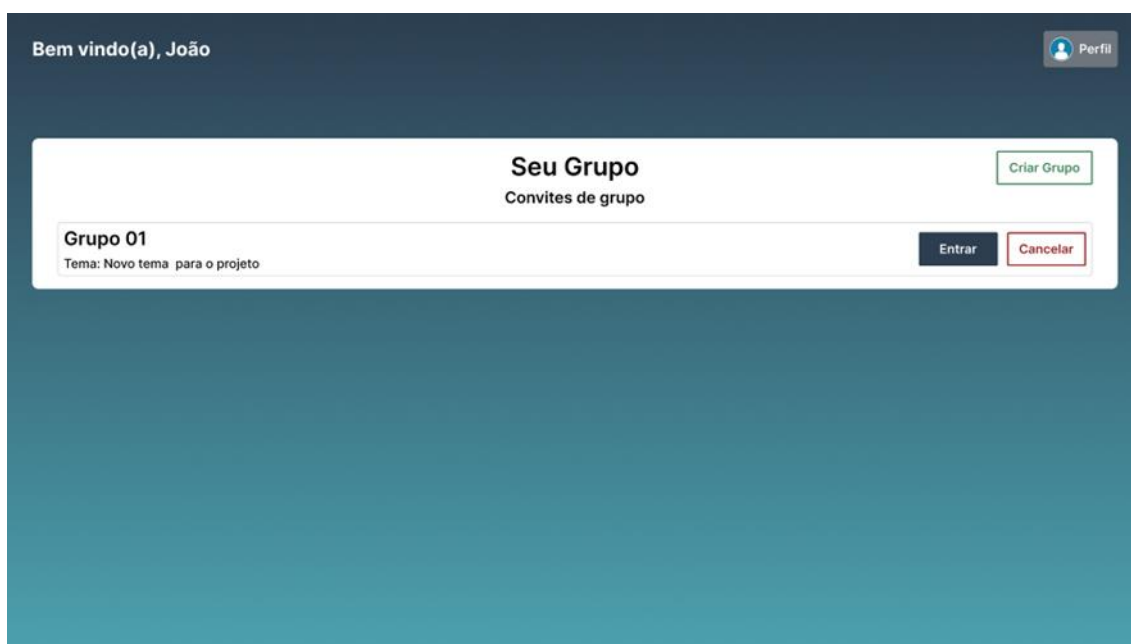


Fonte: Próprios Autores.

Ao acessar o sistema com um usuário da coordenação ou secretaria, será direcionado para o dashboard principal, onde terá todos os professores orientadores ativos. Ao clicar em um professor específico, irá trazer todos os grupos que estão sendo orientados e detalhes sobre. Nesta tela irá aparecer apenas grupos com o status ativo, definido pelo orientador responsável.

Na figura 20, é mostrado a tela de convite de grupo.

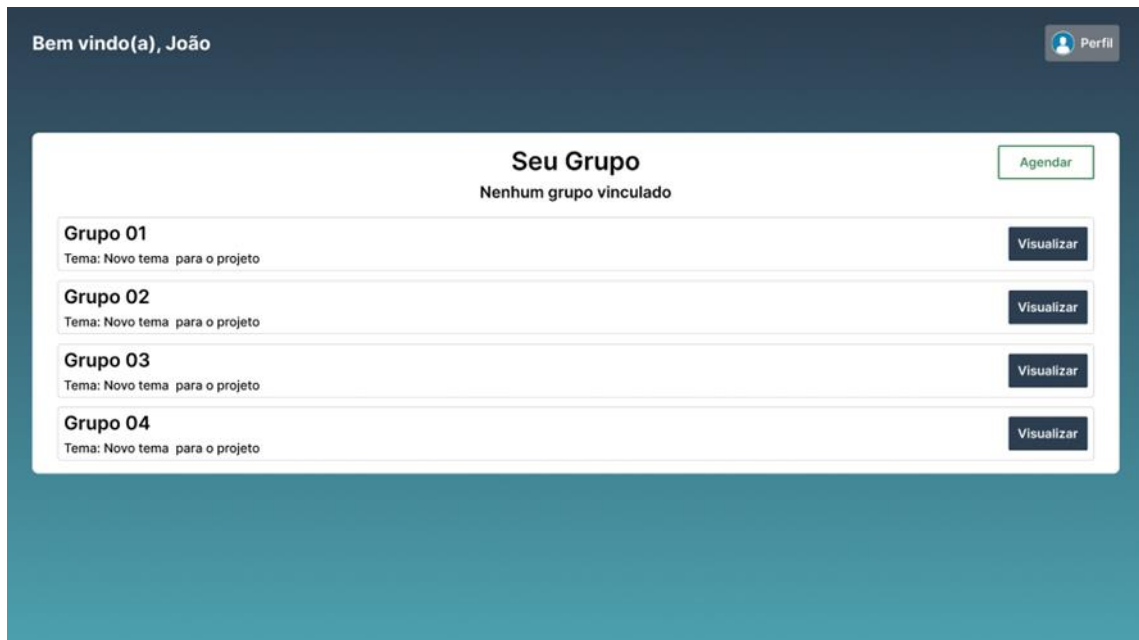
Figura 20 – Tela de convite de grupo.



Fonte: Próprios Autores.

Quando um aluno for convidado para um grupo, será notificado com o tema e a descrição do projeto na tela principal, logo após realizar o login, com a opção de aceitar ou recusar. Case aceite, será possível acessar a página de detalhes do trabalho de graduação. Na figura 21, é apresentado o *Dashboard* do orientador.

Figura 21 – Dashboard do orientador.



Fonte: Próprios Autores.

Ao realizar o login como orientador, será possível verificar todos os grupos orientados. Nesta tela, o professor poderá ver mais detalhes do grupo, clicando no botão de visualizar. Também será possível realizar o agendamento do horário da banca para o grupo, definindo o dia e horário, conforme o fluxo de agendamento na figura 22.

Na figura 22 é apresentado o fluxo de agendamento e figura 23 é mostrado a tela de confirmação do agendamento.

Figura 22 – Tela de agendamento.

Bem vindo(a), João

Perfil

Selecione uma data

Maio 2023

30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Horário

- 09:00
- 10:00
- 11:00
- 12:00
- 13:00
- 14:00

Fonte: Próprios Autores.

Figura 23 – Tela de confirmação do agendamento.

Bem vindo(a), João

Perfil

Selecione uma data

Maio 2023

30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Horário

- 09:00
- 10:00
- 11:00
- 12:00
- 13:00
- 14:00

Fonte: Próprios Autores.

Ao seguir com o agendamento do grupo, será necessário escolher o dia e horário da banca. Após isso será direcionado para a tela de confirmação do

agendamento, onde o orientador irá verificar se está correto para seguir com o processo.

Para uma compreensão geral, seguem dois links para vídeos de apresentação realizados pela equipe. O QR code e *link* a seguir o levará para um vídeo de apresentação da problemática, objetivos e metodologias utilizadas no desenvolvimento do projeto:



https://drive.google.com/file/d/1qcFAcSyrzEMIYOw0kcTt0dzF4JfMtucu/view?usp=share_link

Para assistir ao vídeo explicativo do funcionamento do sistema, basta acessá-lo por meio do seguinte QR code ou pelo *link* na sequência:



<https://www.youtube.com/watch?v=9Ffzd2uwy6U>.

6. DISCUSSÕES FINAIS

A Faculdade de Tecnologia Arthur de Azevedo de Mogi Mirim possui como obrigatório para alguns de seus cursos um projeto intitulado Trabalho de Graduação (TG). Identificou-se dificuldades e retrabalhos devido à integridade de dados das equipes, orientadores, montagem de bancas e arquivamento dos trabalhos.

No cenário atual onde é necessário cada vez mais automatização de processos com agilidade e qualidade faz-se necessário o uso de ferramentas cada vez mais tecnológicas e customizadas para cada segmento, otimizando assim o fluxo destes processos.

Pensando em um projeto de grande escopo, foram definidas 3 fases de desenvolvimento. A **FASE 1**, já implementada em um TG anterior orientado pelo professor Marcio Rodrigues Sabino, realizou-se a organização e desenvolvimento de todo o fluxo de criação dos grupos de usuários definidos por alunos, orientadores, coordenação/secretaria e administrador, bem como seus cadastros e forma de acesso ao sistema, diferenciando por tipo de usuário. Criação de uma listagem de orientadores que deverá ser cadastrada pelo administrador, assim como as credenciais de acesso de usuários da coordenação/secretaria. Ainda nessa fase, a criação de grupos de trabalho, convite aos alunos e orientadores envolvidos.

Para este trabalho atual, os objetivos principais orbitaram na realização da **FASE 2** apresentando a organização e desenvolvimento das funcionalidades de *upload* e *download* dos arquivos finais do Trabalho de Graduação, alteração de dados pessoais, alteração de detalhes de grupos já cadastrados, bem como a possibilidade de um aluno deixar um grupo ao qual faz parte. Um *dashboard* do orientador com um novo módulo para gerenciamento das bancas e seus respectivos convites e datas agendadas para bancas de qualificação e defesa.

Com o foco nas dores do público-alvo definido como alunos, orientadores e coordenação da Fatec Arthur de Azevedo de Mogi Mirim, utilizou-se a metodologia *design thinking* seguindo as etapas de imersão, ideação e prototipação. Para o embasamento teórico utilizou-se a pesquisa bibliográfica. O desenvolvimento foi separado em *frontend* e *backend*: para o *backend* utilizou-se as tecnologias do Node.js e Typescript enquanto no *frontend* o utilizado o Next.js, *framework* do React.js. No banco de dados adotou-se o Firebase o qual é NoSQL e foi incorporado

o PostgreSQL como banco de dados relacional para tornar o projeto robusto e escalável. Foi também realizada a criação de diagramas de caso de uso e diagramas de atividades, readequação e melhorias no código visando performance e sua implementação e testes.

Os objetivos propostos para o projeto foram cumpridos integralmente, possibilitando a entrega de um sistema funcional.

Espera-se com ele possibilitar a integração dos grupos de alunos, orientador e a secretária acadêmica com um sistema de usabilidade simples e seguro, visando uma melhor organização, agilidade nos processos e confiabilidade de resguardo dos dados.

Como projetos futuros, inicializará a **FASE 3**, a qual buscará a organização e desenvolvimento do módulo para que a coordenação/secretaria se comunique por meio de um sistema de mensagens, podendo assim enviar comunicados gerais ou para determinado grupo ou aluno específico. Fica ainda previsto para essa fase a implementação do controle de versões do trabalho escrito. Por fim, deverá ser desenvolvida a área para indicar a situação de um grupo, após qualificação e/ou defesa, ou seja, se ele foi aprovado ou reprovado e a geração da Ata de aprovação pela coordenação/secretaria.

REFERENCIAL BIBLIOGRÁFICO

ALURA. **Desenvolvimento Web com HTML, CSS e JavaScript: curso wd-43**. Curso WD-43. 2018. Disponível em: <https://www.caelum.com.br/apostila-html-cssjavascript/>. Acesso em: 29 out. 2022.

ALURA. **Java e Orientação a Objetos: Curso FJ-11**. Disponível em: <https://www.caelum.com.br/apostila-java-orientacao-objetos/orientacao-a-objetosbasica/#criando-um-tipo>. Acesso em: 29 out. 2022.

AQUINO, Amanda Christoval da Veiga de; MELLO, Ronaldo dos Santos. **Um Levantamento sobre Sistemas de Gerenciamento de Bancos de Dados NoSQL Multimodelo**. Anais do XVI Escola Regional de Banco de Dados. SBC, 2021. p. 31-40.

BROWN, T. ***Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias***. Rio de Janeiro: Elsevier, 2010.

DAMASCENO, Jéssica. ***Design Thinking: empatia, criação e inovação***. 2018. Disponível em: <https://medium.com/blogando/design-thinking-empatia-cria%C3%A7%C3%A3o-e-inova%C3%A7%C3%A3o-f2d238bdbfc8>. Acesso em: 30 out. 2022.

DATE, C. J. **Introdução à sistema de bancos de dados**. 8. ed. Rio de Janeiro: Elsevier, 2004.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. Ed. São Paulo: Addison Wesley, 2011.

FALBÓ, Ricardo de Almeida. **Engenharia de Software: notas de aula**. UFES (Universidade Federal do Espírito Santo), 2014.

FARINA, Modesto; PEREZ, Clotilde; BASTOS, Dorinho. **Psicodinâmica das cores em comunicação**. 6. ed. São Paulo: Blucher, 2011.

FAULKNER, Steve et al. **HTML 5.1**. 2014. Disponível em: <https://www.w3.org/TR/html51/>. Acesso em: 30 out. 2022.

FLANAGAN, David. **JavaScript O Guia Definitivo**. 6. Ed. Porto Alegre: Bookman, 2013.

GARCIA, Vinícius Salles; SOTTO, Eder Carlos Salazar. **COMPARATIVO ENTRE OS MODELOS DE BANCO DE DADOS RELACIONAL E NÃO-RELACIONAL**. Revista Interface Tecnológica, v. 16, n. 2, p. 12-24, 2019.

GOMES, Ana Laura. **XHTML/CSS: criação de páginas web**. São Paulo: Senac, 2010.

GOMES, Rafael. **Docker para desenvolvedores**. Leanpub, 2017.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. São Paulo: Novatec, 2009. Disponível em: <https://s3.novatec.com.br/capitulos/capitulo9788575222812.pdf>. Acesso em: 30 out. 2022.

JOHANSSON, J. **Create React App vs NextJS – A comparison of two ReactJS based web application frameworks**. Mittuniversitetet, 2021.

KRIGER, Brunno. **Typeorm o que é: conceito, para que serve e como instalar?** 2023. Disponível em: <https://kenzie.com.br/blog/typeorm/#:~:text=trabalhar%20com%20dados.-,Para%20que%20serve%20o%20TypeORM%3F,funcionalidades%20para%20o%20nosso%20projeto>. Acesso em: 02 jun. 2023.

LI, Sing. **React: crie componentes de UI de alto desempenho, que podem ser mantidas**. 2016. Disponível em: <https://developer.ibm.com/br/tutorials/wa-react-intro/>. Acesso em: 02 out. 2022.

MARDAN, Azat. **Express.js Guide: The Comprehensive Book on Express.js**. 1. ed. Leanpub, 2014.

OBJETIVO (Sorocaba). **Design Thinking na educação: O que é e a importância dessa tendência educacional**. 2020. Disponível em: <https://blog.objetivosorocaba.com.br/design-thinking-na-educacao/>. Acesso em: 29 out. 2022.

OLIVEIRA, Sérgio. **Internet das Coisas: com esp8266, Arduino e Raspberry pi**. São Paulo: Novatec, 2017. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=kdQnDwAAQBAJ&oi=fnd&pg=PT4&dq=internet+das+coisas&ots=zLCmAf82-O&sig=RIQ-3bu-SOnGI6vd6LI1v8mmU3s#v=onepage&q=internet%20das%20coisas&f=false>. Acesso em: 26 out. 2022.

PAGANI, T. **Design Thinking**. São Paulo: Senac, 2018.

PINHO, José Benedito. **O poder das marcas**. Summus, 1996.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. São Paulo: Mcgraw-hill, 2006.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011.

RAPOSO, D. **Design de Identidade e Imagem Corporativa**. Lisboa/Portugal: IPC, 2009.

SANTAELLA, Lucia et al. **Desvelando a Internet das Coisas**. 2013. Disponível em: <http://www.revistageminis.ufscar.br/index.php/geminis/article/view/141>. Acesso em: 28 OUT. 2022.

SANTOS, Felipe Mota dos et al. **Uma aplicação web para organizar programas de pesquisa em engenharia de software seguindo modelo ABC**. 2021.12 p. Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação)

– Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática, 2021.

SCARDOELLI, Keren Antonielli; PINTO, Giuliano Scombatti. **BANCO DE DADOS NOSQL: uma alternativa para grandes empresas**. Revista Interface Tecnológica, v. 17, n. 2, p. 219-230, 2020.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de dados**. 5. ed. Rio de Janeiro: Elsevier, 2006.

SILVA, Gilmar José da; FERREIRA, Júlio Cesar Oliveira. **Análise Comparativa de Desempenho de Consultas entre um Banco de Dados Relacional e um Banco de Dados Não Relacional**. 2017.

SILVA, Maurício Samy. **Construindo sites com CSS e (X) HTML: sites controlados por folhas de estilo em cascata**. São Paulo: Novatec Editora, 2007.

SILVA, Maurício Samy. **Html 5**. São Paulo: Novatec Editora, 2011.

SILVA, Maurício Samy. **JavaScript Guia do Programador**. São Paulo: Novatec Editora, 2010.

SILVA, Maurício Samy. **JQuery: a biblioteca do programador javascript**. 3. ed. São Paulo: Novatec Editora, 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison Wesley, 2007.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOUZA, Ana Paula de Melo e; CAVASSINI, Gabrielly Alves; SABINO, Marcio Rodrigues. **Design Thinking e Scrum no desenvolvimento de software para Gerência de Petições**. Refas -Revista Fatec Zona Sul, São Paulo, v.7, n.1, p.19-34,

out. 2020. Quadrimestral. Disponível em:
<http://www.revistarefas.com.br/index.php/RevFATECZS/article/view/332/283>. Acesso
 em: 30 out. 2022.

SOUZA, Lucas Ricardo Mendes de. **3 etapas do *Design Thinking***. 2018. Disponível
 em: <https://itforum.com.br/noticias/3-etapas-do-design-thinking/>. Acesso em: 29 out.
 2022.

STICKDORN, M. **Isto é *Design Thinking* de Serviços**. Porto Alegre: Bookman, 2014.

STRUNCK, G. **Como Criar Identidades Visuais Para Marcas de Sucesso**. Rio de
 Janeiro: RIO BOOKS, 2001.

STRUNCK, G. **Identidade visual: a direção do olhar**. Europa, 1989.

TYPEORM. **TypeORM**. 2021. Disponível em:
<http://www.revistarefas.com.br/index.php/RevFATECZS/article/view/332/283>. Acesso
 em: 30 out. 2022.

VIANNA, Mauricio et al. ***Design thinking: inovação em negócios***. 2. ed. Rio de
 Janeiro: MJV Press, 2012.

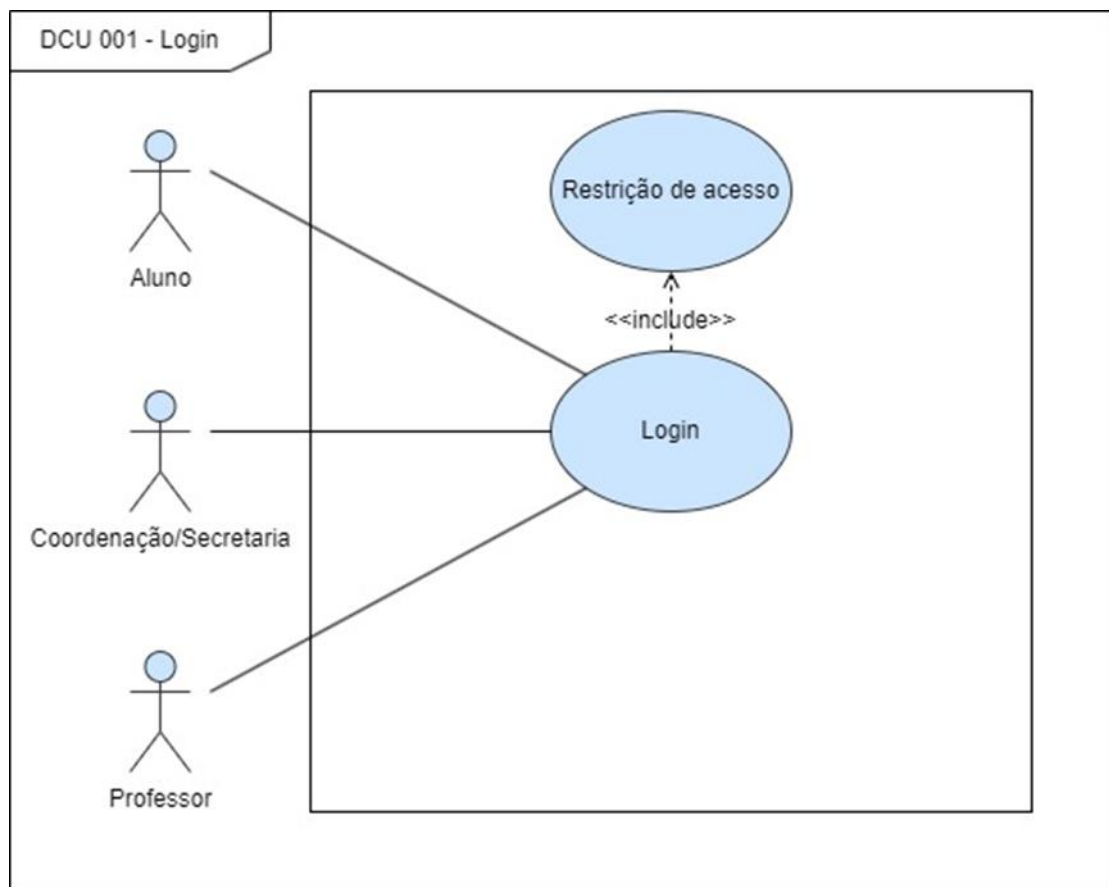
WHEELER, Alina. ***Design de identidade da marca: guia essencial para toda a
 equipe de gestão de marcas***. 5. ed. Porto Alegre: Bookman, 2019.

APÊNDICE

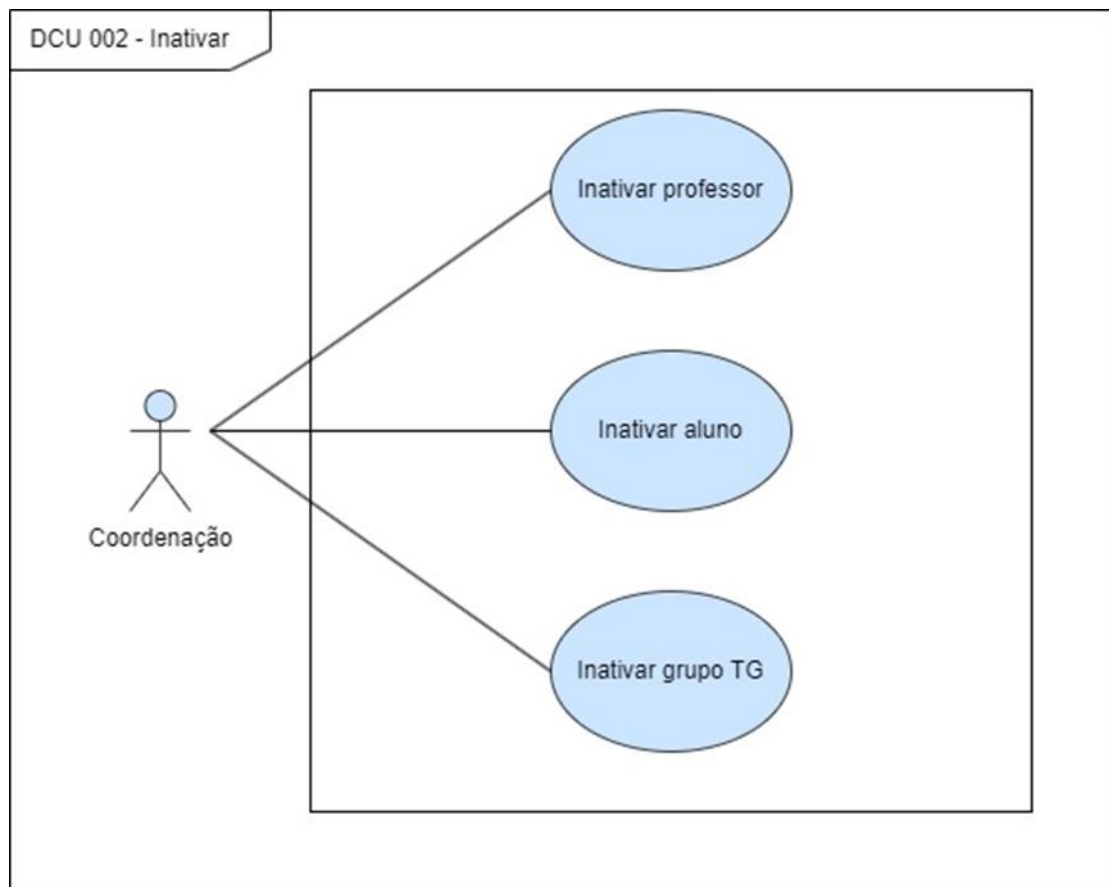
Nesta seção, serão apresentados os diagramas desenvolvidos para a criação do sistema.

1. Diagrama de Caso de Uso

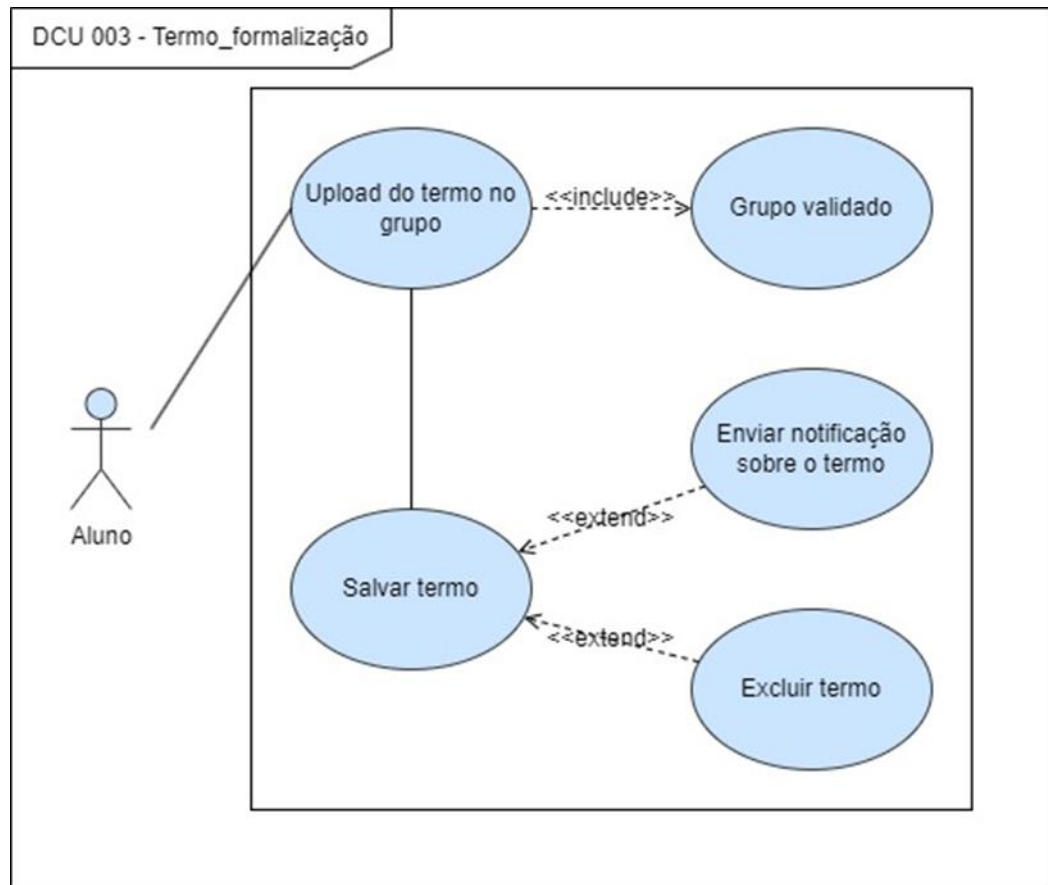
Figura 24 – DCU 001 - Login.



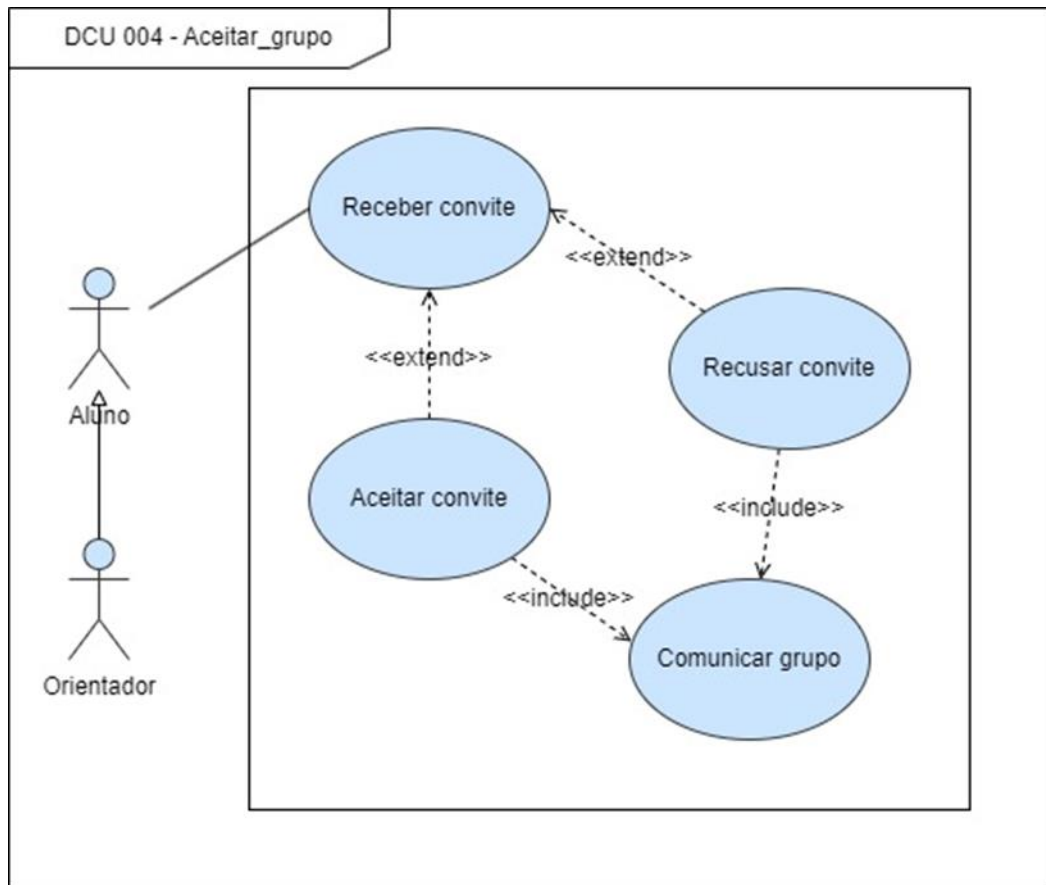
Fonte: Próprios Autores.

Figura 25 – DCU 002 - Inativar.

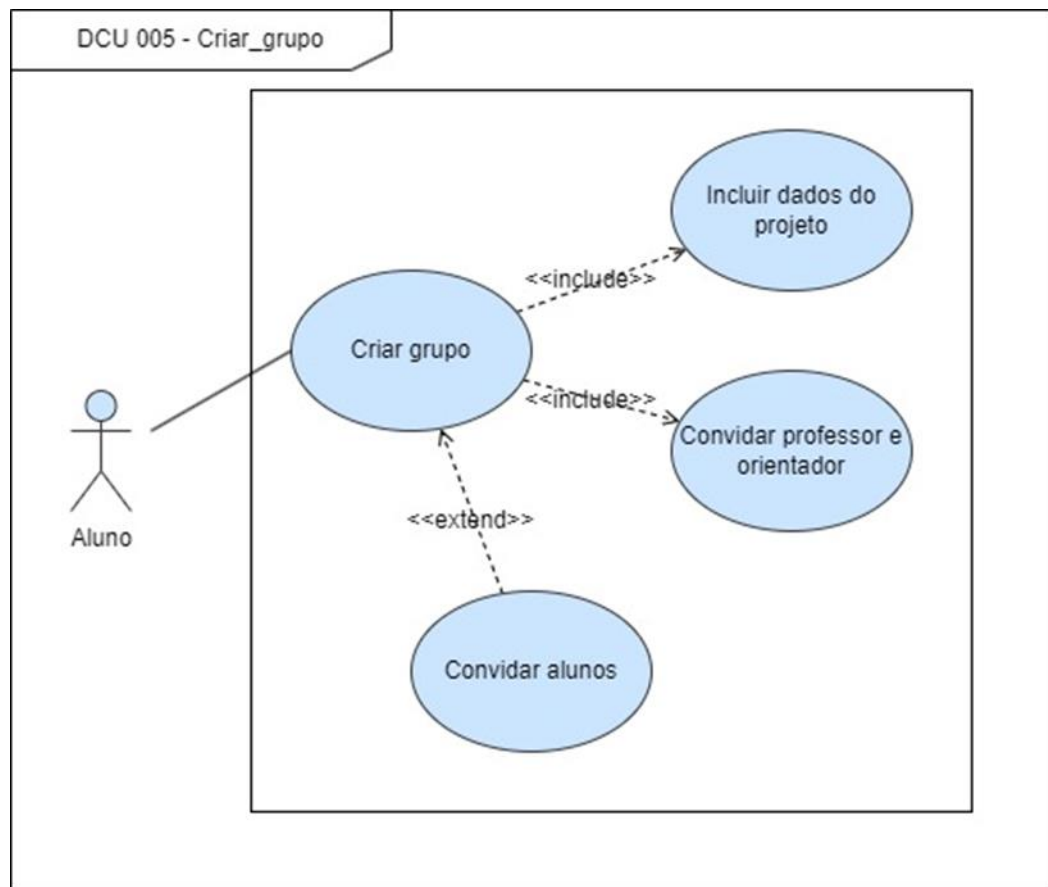
Fonte: Próprios Autores.

Figura 26 – DCU 003 - Termo_formalização.

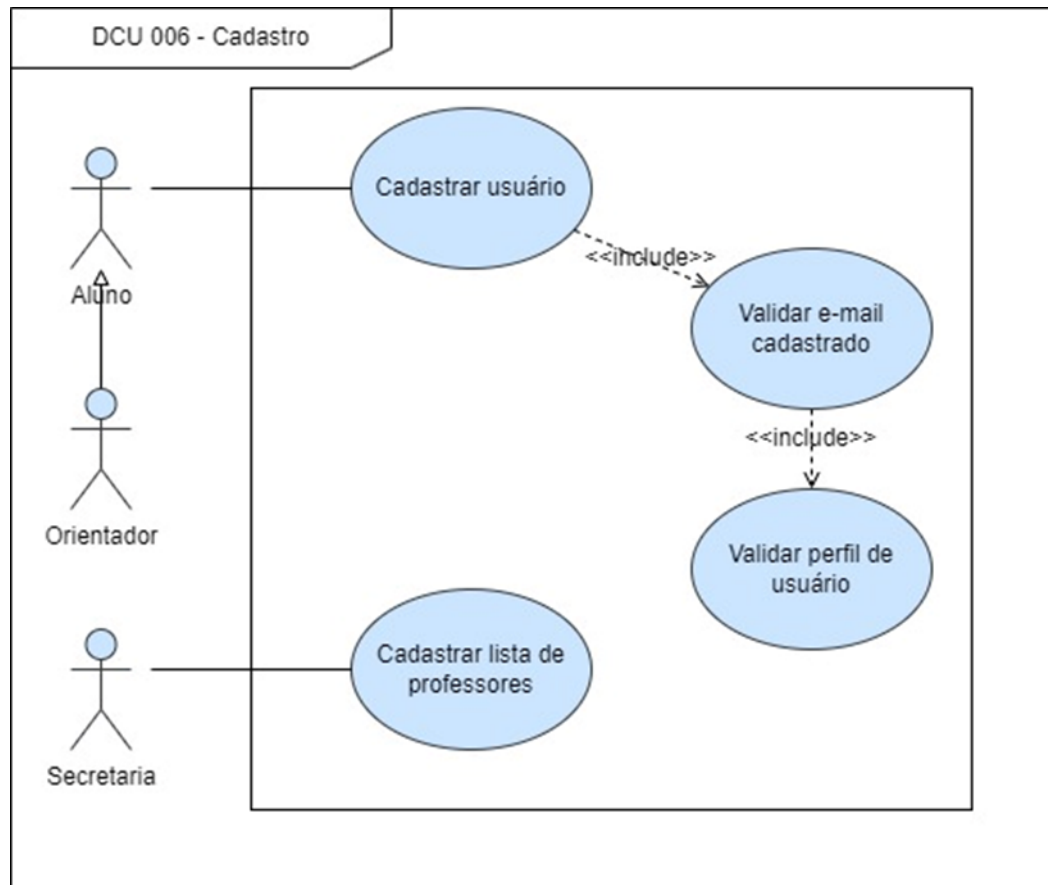
Fonte: Próprios Autores.

Figura 27 – DCU 004 - Aceitar_grupo.

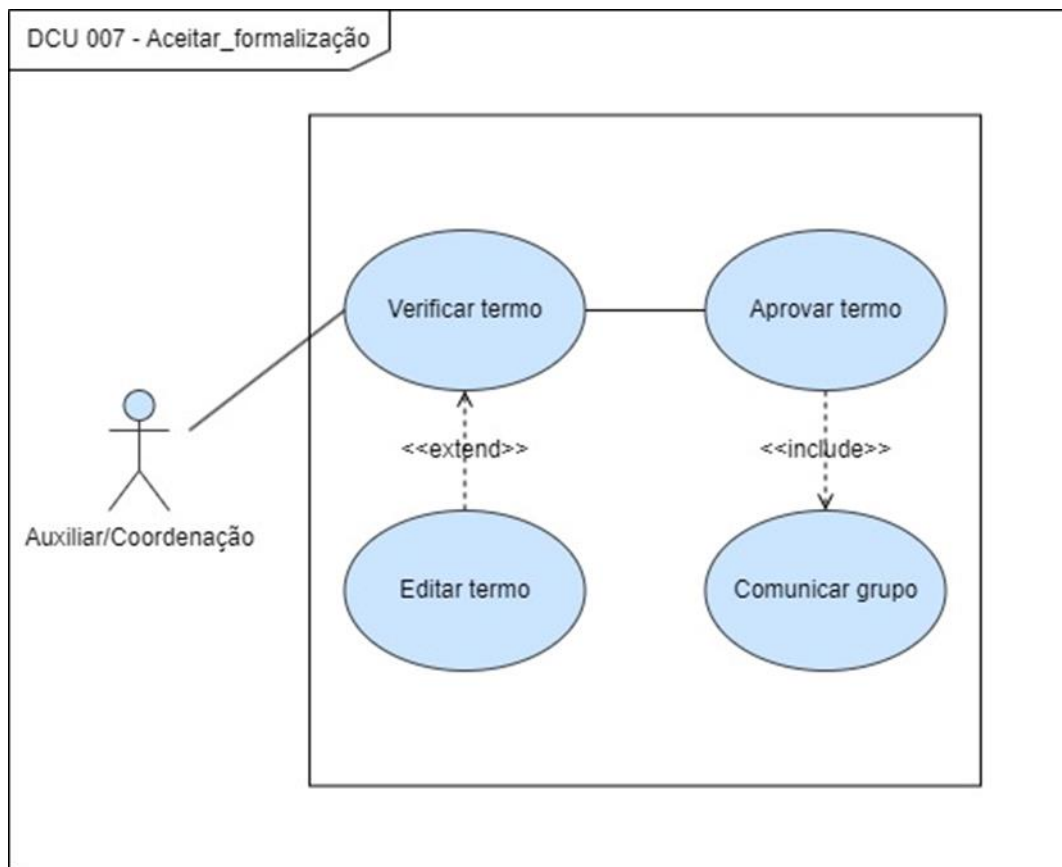
Fonte: Próprios Autores.

Figura 28 – DCU 005 - Criar_grupo.

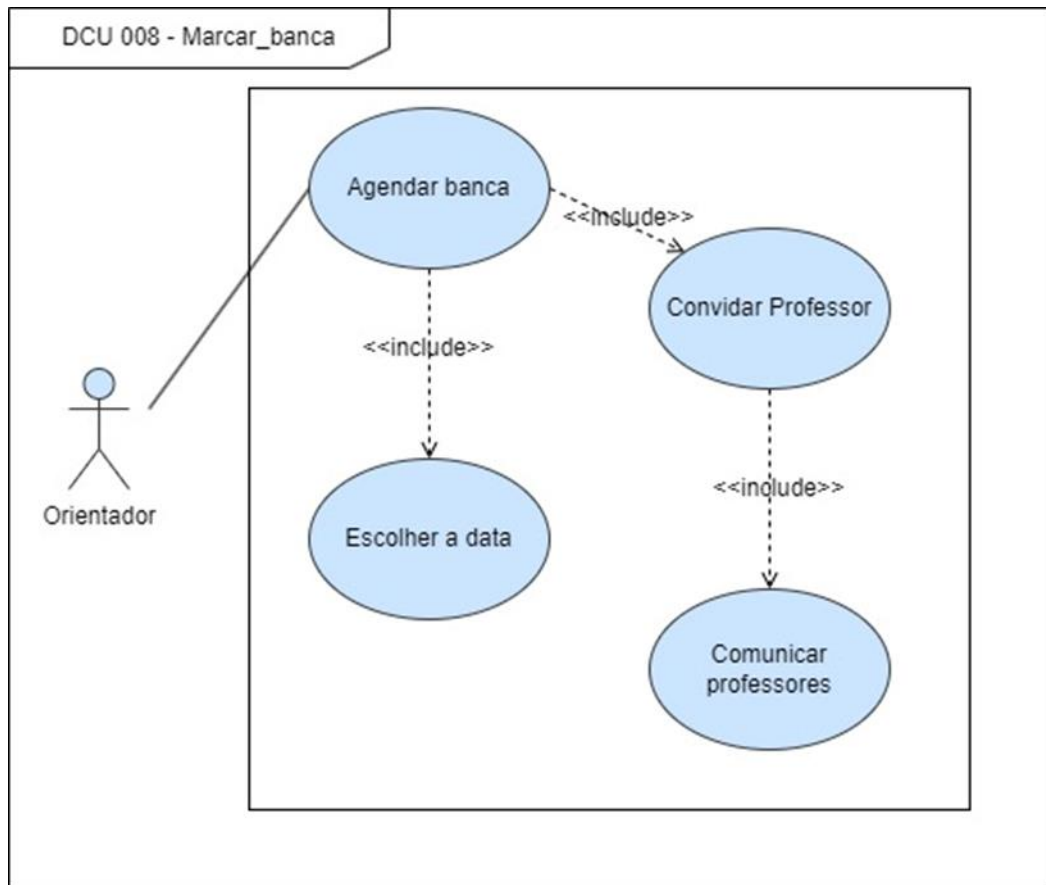
Fonte: Próprios Autores.

Figura 29 – DCU 006 - Cadastro.

Fonte: Próprios Autores.

Figura 30 – DCU 007 - Aceitar_formalização.

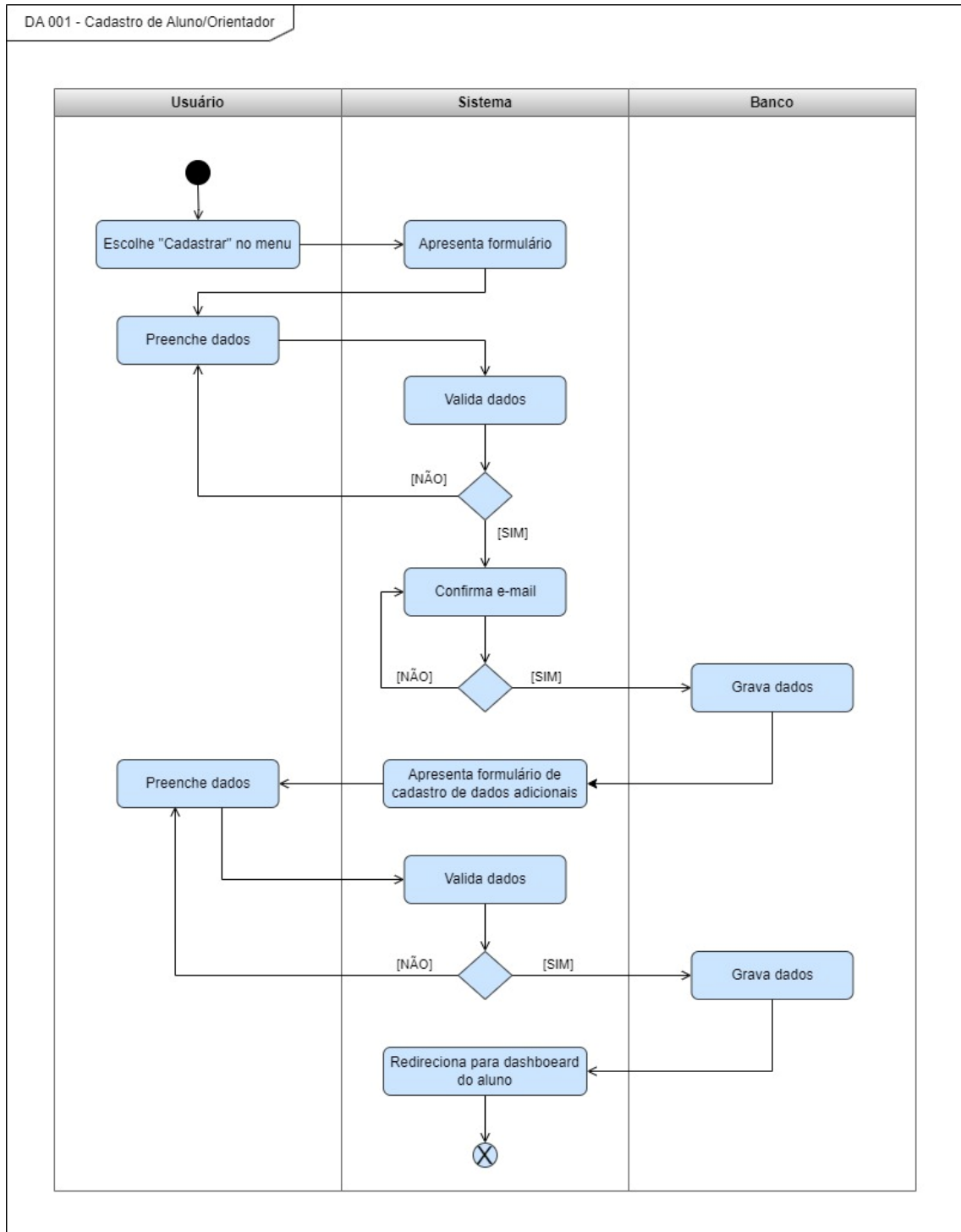
Fonte: Próprios Autores.

Figura 31 – DCU 008 - Marcar_banca.

Fonte: Próprios Autores.

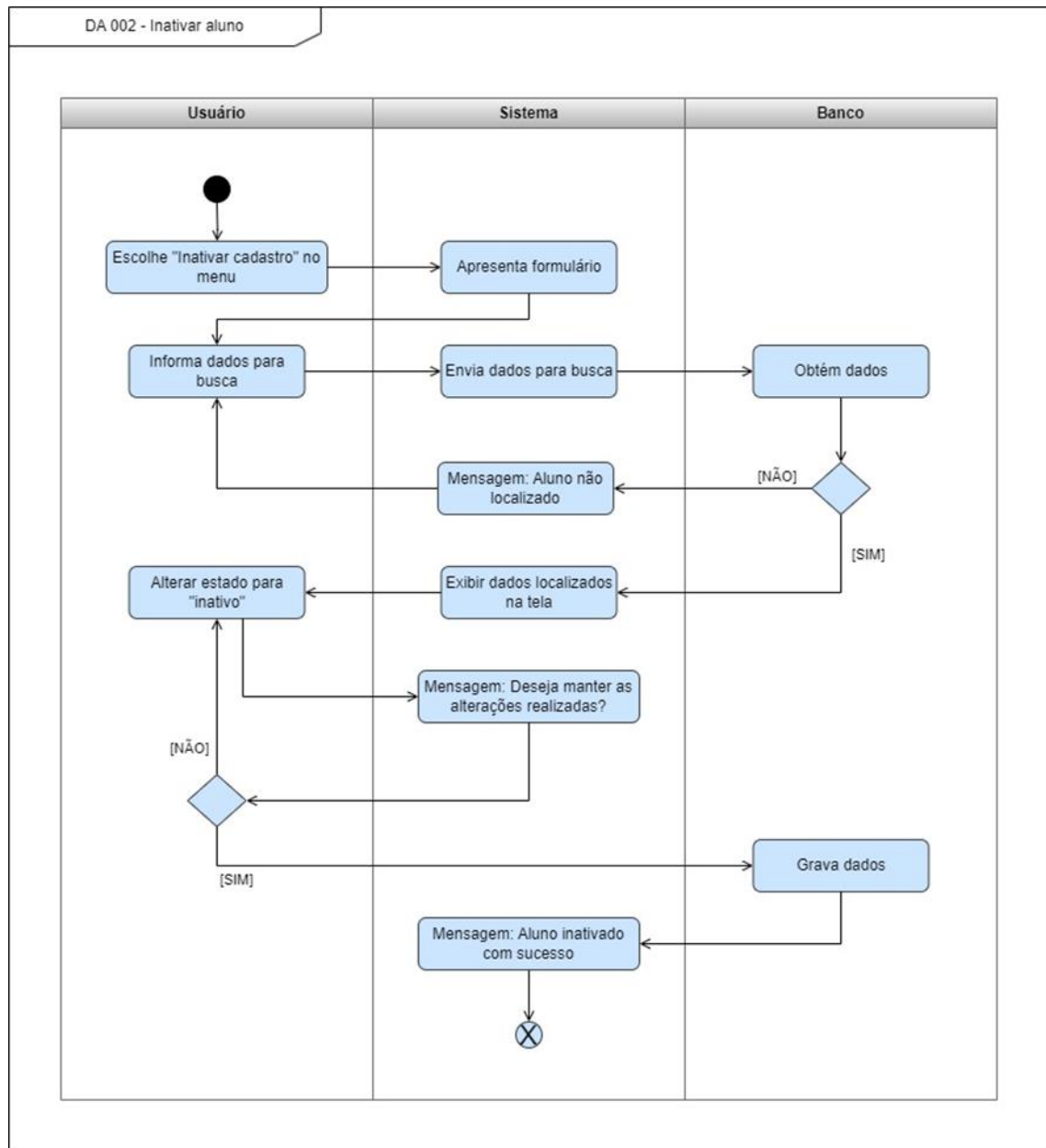
2. Diagrama de Atividades

Figura 32 – DA 001 - Cadastro de Aluno/Orientador.

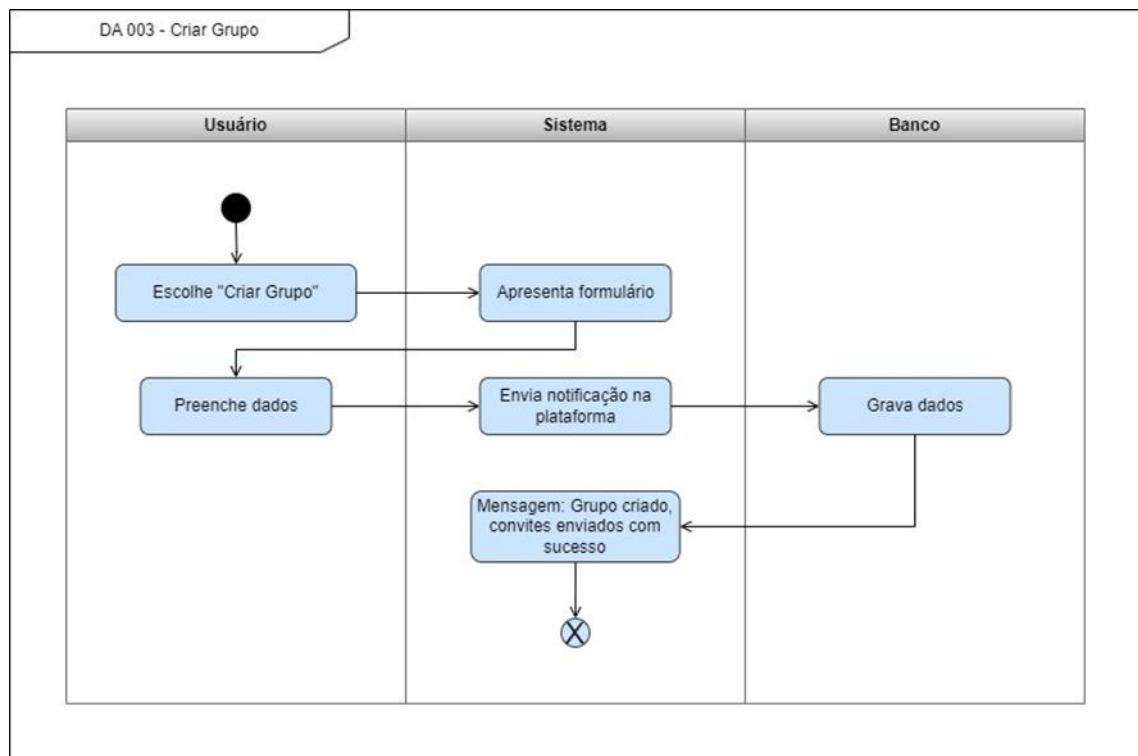


Fonte: Próprios Autores.

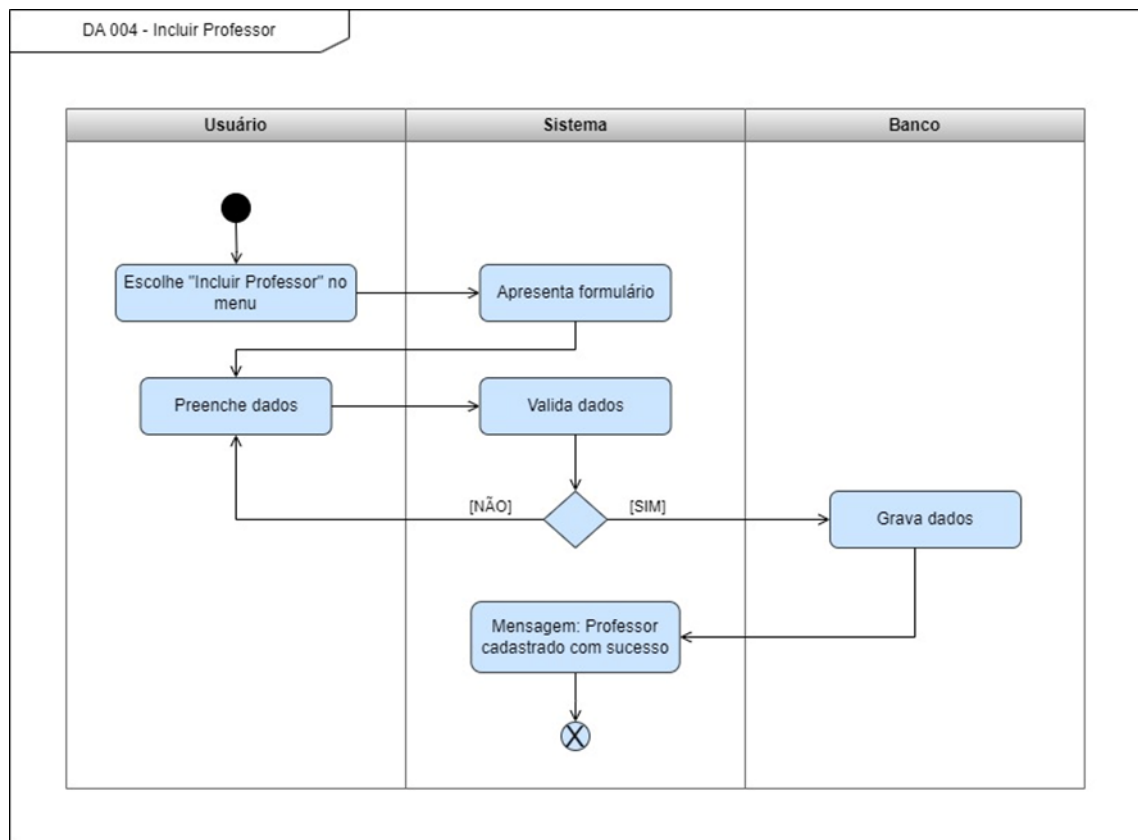
Figura 33 – DA 002 - Inativar aluno.



Fonte: Próprios Autores.

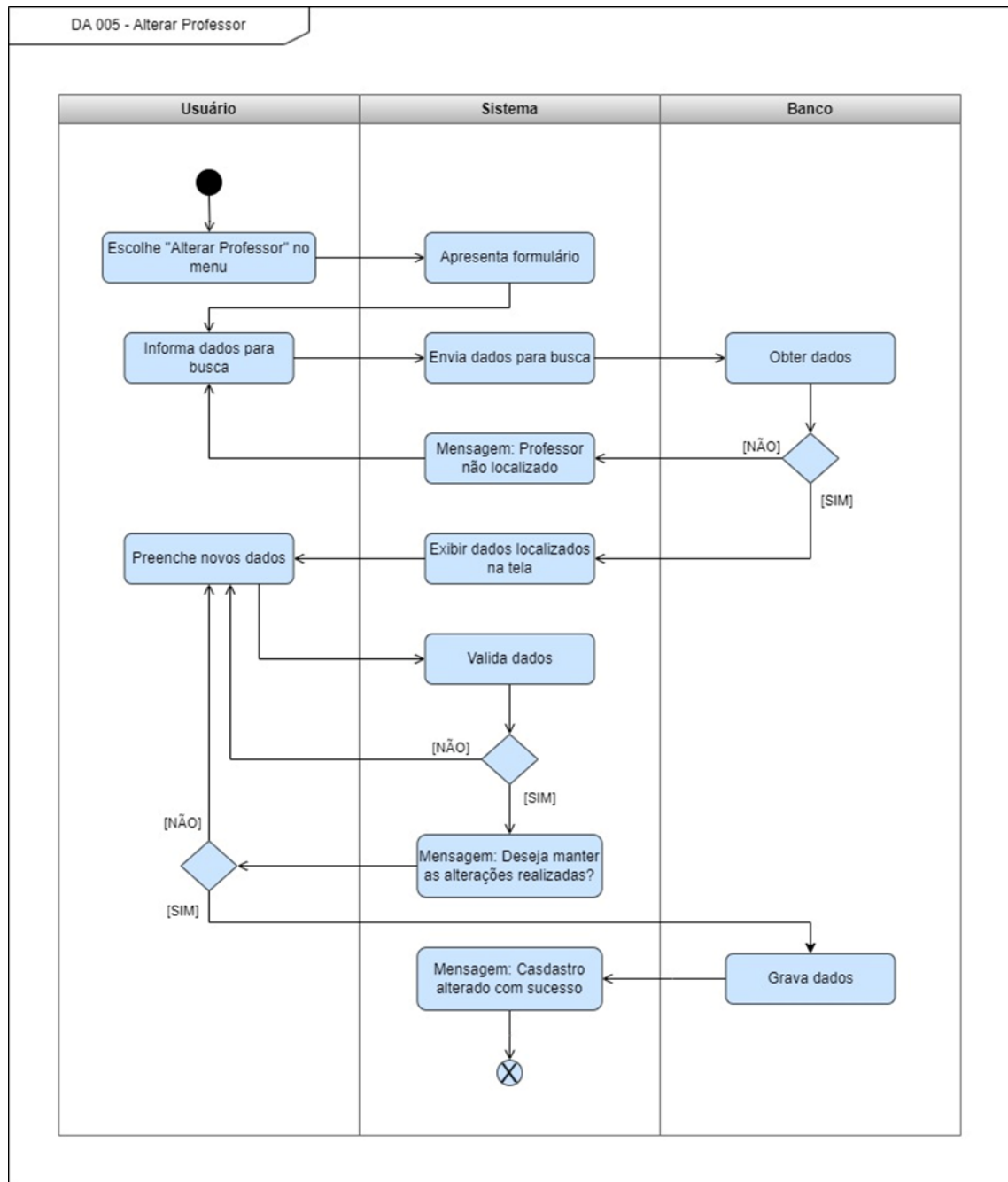
Figura 34 – DA 003 - Criar Grupo.

Fonte: Próprios Autores.

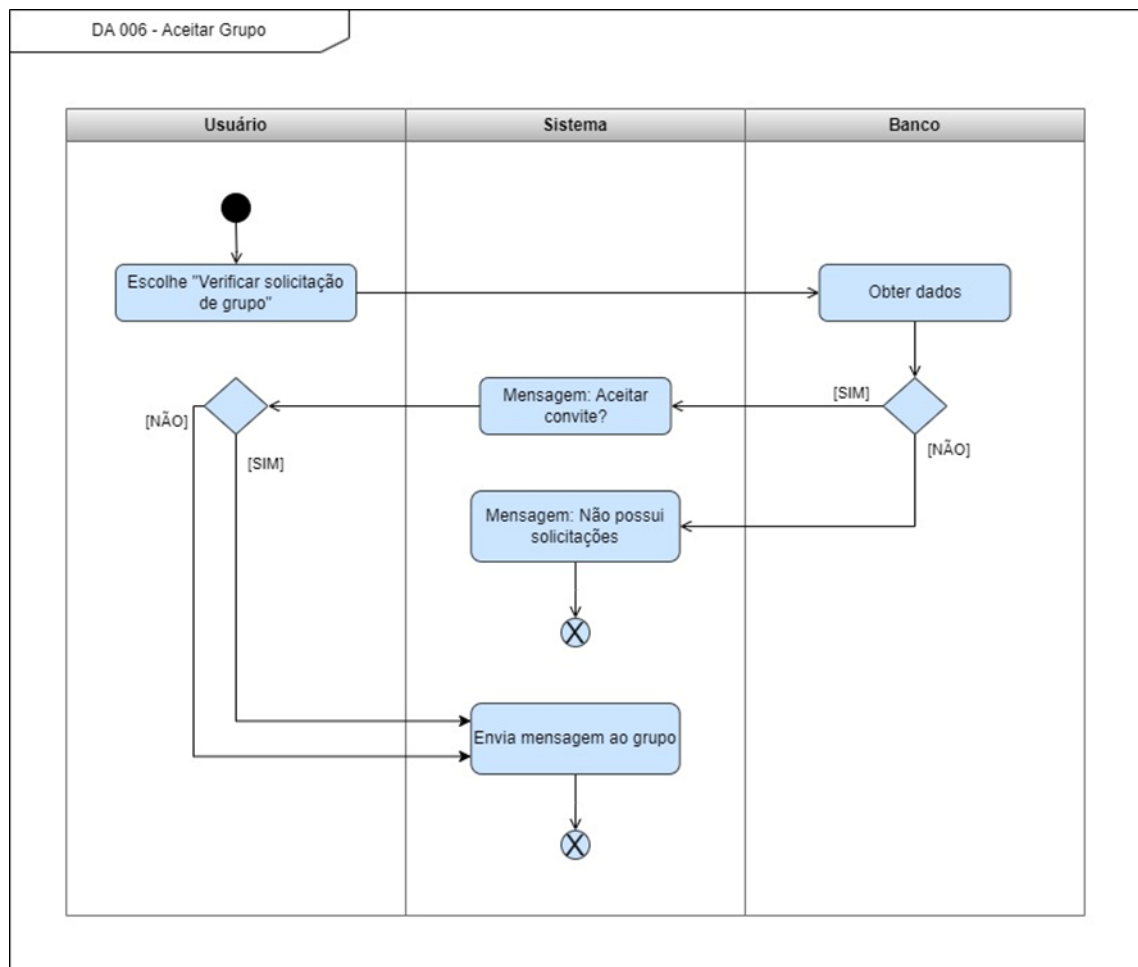
Figura 35 – DA 004 - Incluir Professor.

Fonte: Próprios Autores.

Figura 36 – DA 005 - Alterar Professor.

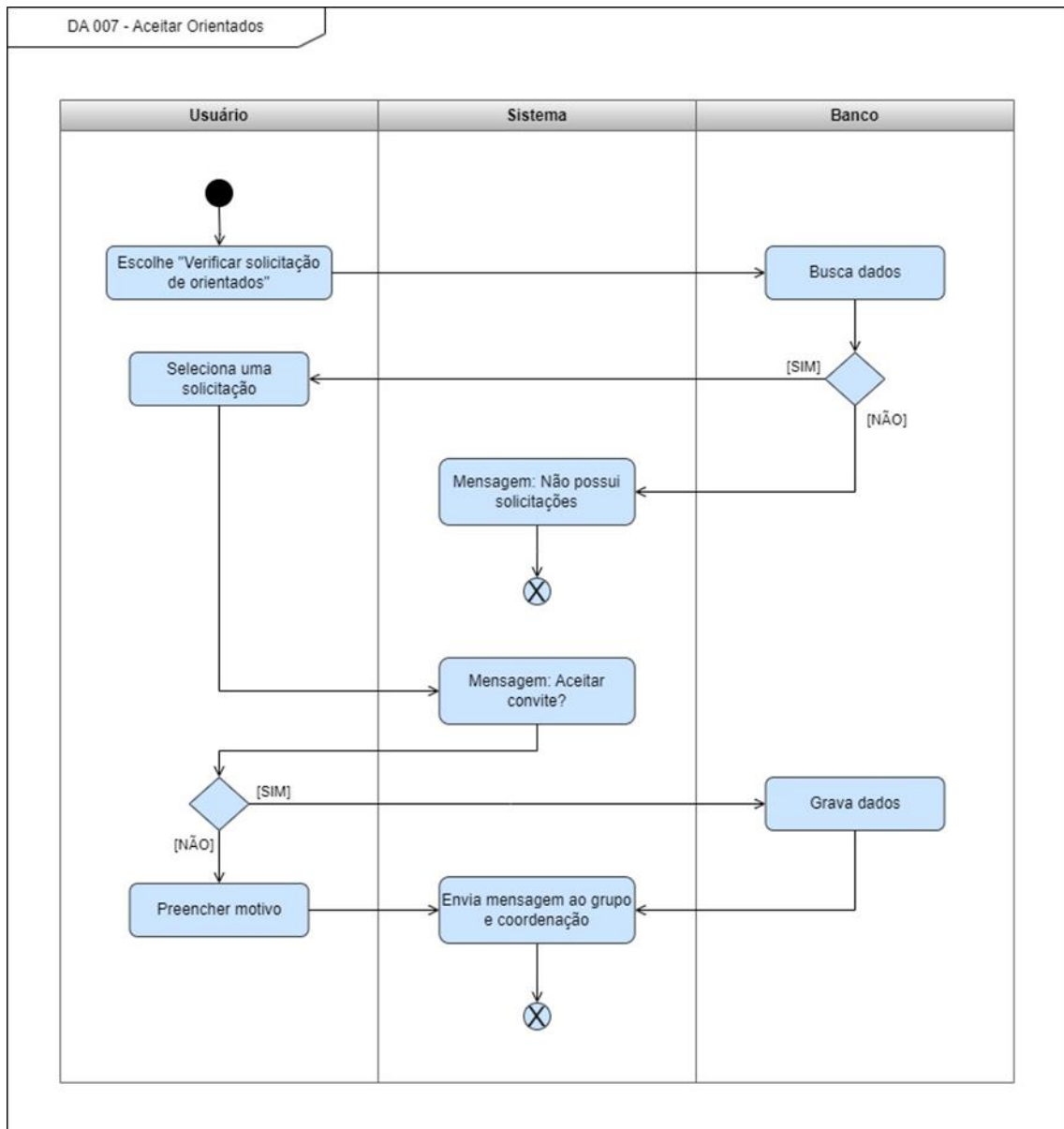


Fonte: Próprios Autores.

Figura 37 – DA 006 - Aceitar Grupo.

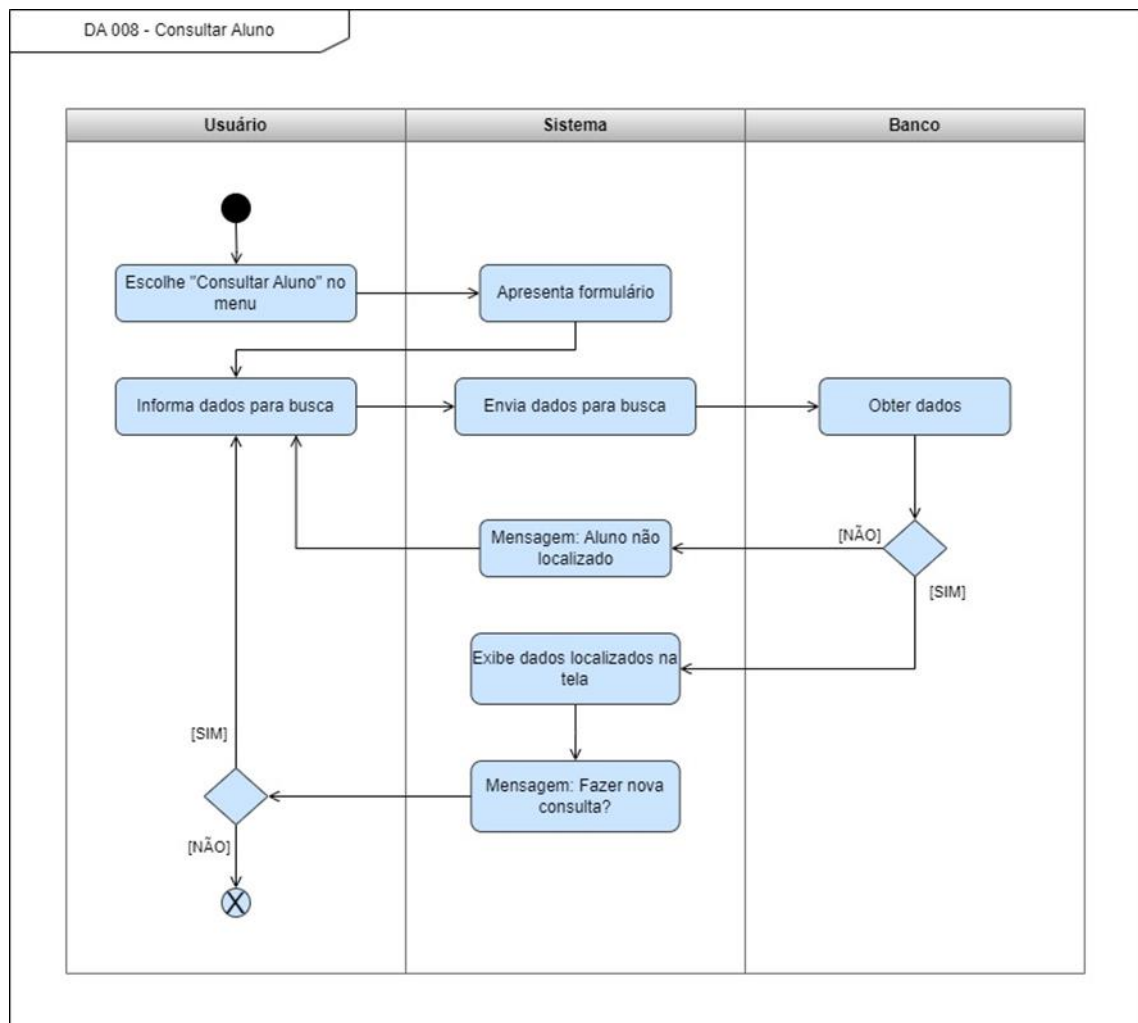
Fonte: Próprios Autores.

Figura 38 – DA 007 - Aceitar Orientados.

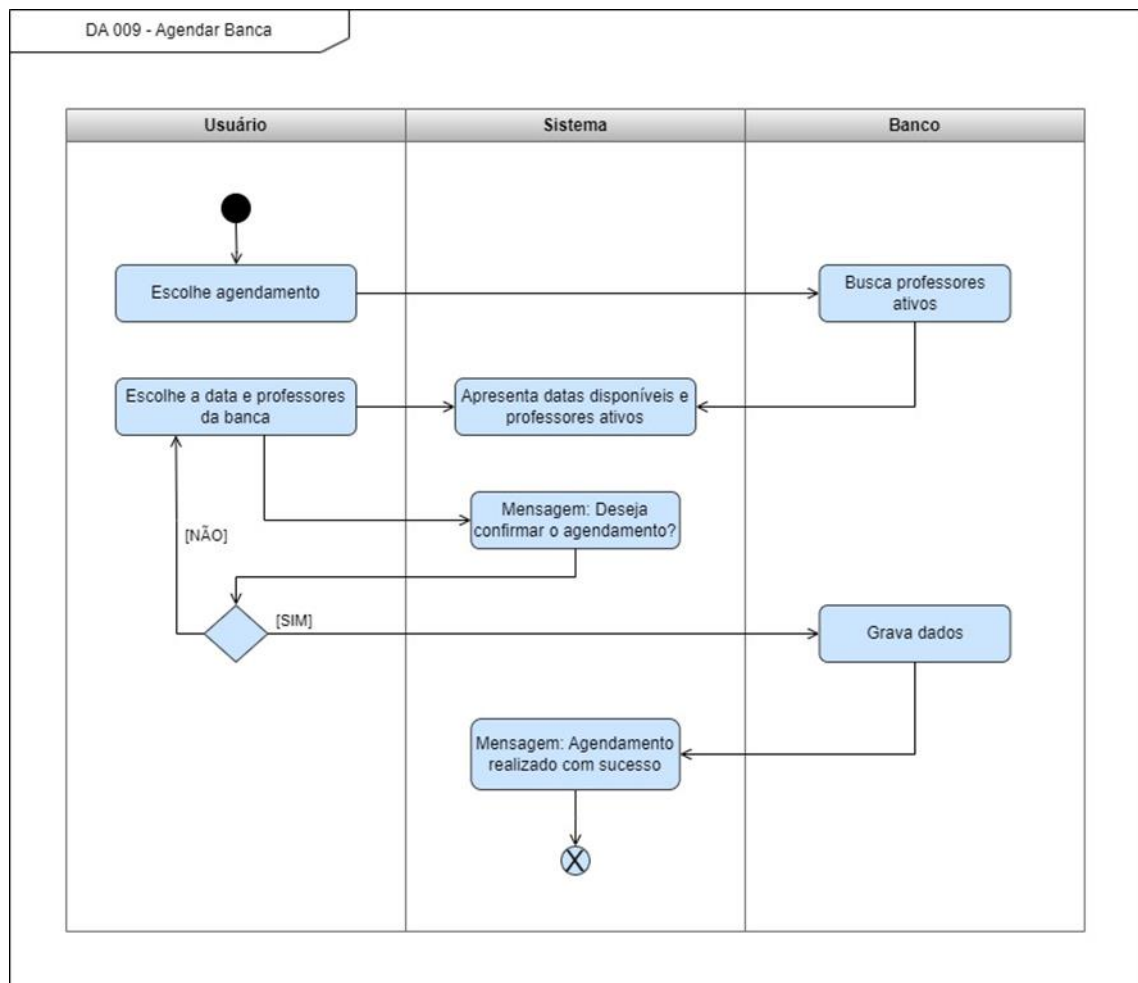


Fonte: Próprios Autores.

Figura 39 – DA 008 - Consultar Aluno.



Fonte: Próprios Autores.

Figura 40 – DA 009 - Agendar Banca.**Fonte:** Próprios Autores.



Fatec Mogi Mirim

ANEXOS

1. DECLARAÇÃO DE INEXISTÊNCIA DE PLÁGIO

TÍTULO DE TRABALHO: SGTG: SISTEMA DE GERENCIAMENTO DE TRABALHO DE GRADUAÇÃO
ALUNOS: ANA LETÍCIA SILVÉRIO
EMILIANO CALLEGARI CORREIA DE SOUZA
KARINA CAPORALI D'AVILA
RAPHAEL DO CARMO

Eu, **ANA LETÍCIA SILVÉRIO**, RA **1630482023012**, eu, **EMILIANO CALLEGARI CORREIA DE SOUZA**, RA **1630482023020**, eu, **KARINA CAPORALI D'AVILA**, RA **1630482023024**, e eu, **RAPHAEL DO CARMO**, RA **1630482023001**, concluintes do curso de Tecnologia em **ANÁLISE E DESENVOLVIMENTO DE SISTEMAS** e que apresentamos nosso trabalho no **1º semestre de 2023**, declaramos que, com exceção das citações diretas e indiretas claramente indicadas e referenciadas, este trabalho foi escrito por nós e portanto não contém plágio. Estamos conscientes que a utilização de material de terceiros incluindo uso de paráfrase sem a devida indicação das fontes será considerado plágio, e estará sujeito a processo administrativo da Faculdade de Tecnologia de Mogi Mirim e sanções legais.

Mogi Mirim, 03 de junho de 2023.

Ana Letícia Silvério

Emiliano Callegari Correia de Souza

Assinatura do aluno1
Karina Caporali D'Avila

Assinatura do aluno2
Raphael do Carmo

Assinatura do aluno3

Assinatura do aluno4



Fatec Mogi Mirim

ANEXOS

2. AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE GRADUAÇÃO

Eu, **ANA LETÍCIA SILVÉRIO**, RA **1630482023012**, eu, **EMILIANO CALLEGARI CORREIA DE SOUZA**, RA **1630482023020**, eu, **KARINA CAPORALI D'AVILA**, RA **1630482023024**, e eu, **RAPHAEL DO CARMO**, RA **1630482023001**, concluintes do curso de Tecnologia em **ANÁLISE E DESENVOLVIMENTO DE SISTEMAS** e que apresentamos nosso trabalho no **1º semestre de 2023**, declaramos que autorizamos a Faculdade de Tecnologia de Mogi Mirim – Arthur de Azevedo a publicar nosso Trabalho de Graduação em seu Site na Internet, integralmente ou parcialmente, bem como a disponibilizar o trabalho impresso na biblioteca da unidade.

Mogi Mirim, 03 de junho de 2023.

Karina Caporali D'Avila

Assinatura do aluno



Versão do CopySpider: 2.1.1

Relatório gerado por: karinacdavila@gmail.com

Modo: web / normal

Arquivos	Termos comuns	Similaridade
TG1_SGTG_VERSÃO_FINAL.docx X http://www.contornospesquisa.org/2012/08/como-referenciar-figuras-imagens-e.html	50	0,35
TG1_SGTG_VERSÃO_FINAL.docx X https://projetoacademico.com.br/fonte-proprio-autor-no-tcc	14	0,14
TG1_SGTG_VERSÃO_FINAL.docx X https://tecnoblog.net/responde/como-colocar-fonte-em-imagens-retiradas-da-internet-nas-normas-abnt	12	0,12
TG1_SGTG_VERSÃO_FINAL.docx X http://fatecmm.edu.br	10	0,11
TG1_SGTG_VERSÃO_FINAL.docx X https://johanna-b.github.io/files/documents/teachingWorkshop16_Beyer.pdf	5	0,03
TG1_SGTG_VERSÃO_FINAL.docx X https://books.google.com/books/about/Engenharia_de_Software_8%C2%AA_Edi%C3%A7%C3%A3o.html?id=wexzCwAAQB AJ	2	0,02

Arquivos com problema de download

https://normas-abnt.espm.br/index.php%3Ftitle%3DFiguras	Não foi possível baixar o arquivo. É recomendável baixar o arquivo manualmente e realizar a análise em conluio (Um contra todos). - sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
https://www.revistageminis.ufscar.br/index.php/geminis/article/view/149	Não foi possível baixar o arquivo. É recomendável baixar o arquivo manualmente e realizar a análise em conluio (Um contra todos). - sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
https://regrasparatcc.com.br/formatacao/fonte-do-proprio-autor	Não foi possível baixar o arquivo. É recomendável baixar o arquivo manualmente e realizar a análise em conluio (Um contra todos). HTTP response code: 301 - 30
https://edisciplinas.usp.br/mod/resource/view.php%3Fid%3D1096878	Não foi possível baixar o arquivo. É recomendável baixar o arquivo manualmente e realizar a análise em conluio (Um contra todos). - Erro: Parece que o documento foi removido do site ou nunca existiu. HTTP response code: 404 - https://edisciplinas.usp.br/mod/resource/view.php%3Fid%3D1096878