



**CENTRO UNIVERSITÁRIO DO DISTRITO FEDERAL – UDF  
COORDENAÇÃO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**AUGUSTO LIBERATO MARQUES FERREIRA**

**PROFIDINA ÁGIL  
PLATAFORMA DE GERENCIAMENTO DE EQUIPES**

**BRASÍLIA  
2025**

**AUGUSTO LIBERATO MARQUES FERREIRA**

**PROFIDINA ÁGIL**

Software para apoio de atividades práticas em sala de aula

Trabalho de conclusão de curso apresentado à Coordenação dos Cursos de Tecnologia, do Centro Universitário do Distrito Federal - UDF, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profa. Dr. Kerlla Luz

Coorientador: Profa. Me. Kadidja Valéria  
Reginaldo de Oliveira

**BRASÍLIA**

**2025**

OBS.: Esta folha deverá ser impressa no verso da folha de rosto (folha anterior)

Sobrenome, Nome.

Título : subtítulo / Nome Sobrenome. -- Brasília, 2021.

xx f. (*quantidade de folhas da monografia*)

Orientador: XXXXXX.

Trabalho de conclusão de curso (Graduação – Direito) -- Centro Universitário do Distrito Federal – UDF. Coordenação de Direito, Brasília, DF, 2021.

1. Assunto. 2. Assunto. 3. Assunto. I. Título.

DU: (consultar na biblioteca)

## **AUGUSTO LIBERATO MARQUES FERREIRA**

### **PROFIDINA ÁGIL**

Software para apoio de atividades práticas em sala de aula

Trabalho de conclusão de curso apresentado à Coordenação dos Cursos de Tecnologia, do Centro Universitário do Distrito Federal - UDF, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profa. Dr. Kerlla Luz.

Coorientador: Profa. Me. Kadidja Valéria Reginaldo de Oliveira.

Brasília, \_\_\_\_\_ de \_\_\_\_\_ de 2021.

#### **Banca Examinadora**

---

**NOME DO EXAMINADOR**  
Titulação  
Instituição a qual é filiado

---

**NOME DO EXAMINADOR**  
Titulação  
Instituição a qual é filiado

---

**NOME DO EXAMINADOR**  
Titulação  
Instituição a qual é filiado

NOTA: \_\_\_\_\_



Dedico este trabalho à minha família pelo  
apoio na realização de tal.

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus, por mais uma conquista; ao meu orientador, pela dedicação e correções.

Durante a elaboração deste trabalho, foram utilizadas ferramentas de Inteligência Artificial Generativa (Claude, Gemini, ChatGPT, entre outras...) exclusivamente para apoio na revisão textual e na organização didática do conteúdo. O uso foi restrito a aspectos formais, não substituindo a autoria intelectual nem a análise crítica dos resultados. A responsabilidade integral pelo conteúdo é dos autores.

“Se a educação sozinha não transforma a sociedade, sem ela tampouco a sociedade *muda*.”

Paulo Freire

## **RESUMO**

Este trabalho apresenta as etapas para o desenvolvimento de um software para auxílio de atividades práticas em sala de aula. A ideia é desenvolver o software utilizando ferramentas gratuitas como ide Visual Studio Code, banco de dados PostgreSQL, JavaScript HTML e CSS. O software agilizará a organização dos estudantes, o ingresso em uma sala de aula criada pelo professor dentro da plataforma é simples e rápido, é o próprio estudante quem ingressa na sala e responde um questionário de perfil, que atribui pontos a ele de acordo com as respostas, e a organização dos grupos é feita pela própria plataforma de uma maneira equilibrada, priorizando que estudantes com uma pontuação ruim fique em um mesmo grupo de estudantes com uma pontuação maior. É o professor quem estabelece um limite de estudantes por grupo, o objetivo é que a organização de grupos em turmas muito grandes não consuma muito tempo da aula. Diminuir o tempo que se leva para organizar os grupos evita de haver pouco tempo para a realização de atividades práticas em sala de aula.

**Palavras-chave:** agilizará. ingresso. equilibrada

## **ABSTRACT**

This work presents the steps for developing software to assist with practical activities in the classroom. The idea is to develop the software using free tools such as Visual Studio Code IDE, PostgreSQL database, JavaScript, HTML, and CSS. The software will streamline student organization; joining a classroom created by the teacher within the platform is simple and quick. Students themselves join the room and answer a profile questionnaire that assigns points to them based on their responses, and group organization is performed by the platform itself in a balanced manner, prioritizing that students with lower scores be placed in the same group as students with higher scores. It is the teacher who establishes a limit of students per group; the objective is that group organization in very large classes does not consume too much class time. Reducing the time taken to organize groups prevents there from being little time for carrying out practical activities in the classroom.

Keywords: streamline. enrollment. balanced

## LISTA DE ILUSTRAÇÕES

Figura 01 - Imagem do site do TBL Active.....	17
Figura 02 - Imagem do site do DreamShaper.....	18
Figura 03 - Imagem do Kahoot.....	19
Figura 04 - imagem sobre benefícios de metodologia ativas.....	22
Figura 05 - Seta representando menos custos.....	24
Figura 06 - Modelo cascata.....	26
Figura 07 - Imagem de gestão empresarial integrada.....	27
Figura 08 - Imagem de código JavaScript.....	29
Figura 09 - Imagem de comando Node.....	31
Figura 10 - Imagem de código VUE 3.....	32
Figura 11 - Diagrama de caso de uso.....	35
Figura 12 - Diagrama Entidade Relacionamento.....	39
Figura 13 - Tela inicial.....	43
Figura 14 - Tela de cadastro.....	44
Figura 15 - Tela de confirmação de e-mail.....	46
Figura 16 - Tela de salas.....	47
Figura 17 - Modal de confirmação.....	49
Figura 18 - Modal para ingressar na sala.....	49
Figura 19 - Tela para envio do nome e rgm.....	50
Figura 20 - Questionário de perfil.....	52
Figura 21 - Tela para formação e gerenciamento de grupo.....	54

## LISTA DE TABELAS E QUADRO

Quadro 01 - Estudo de funcionalidades.....	20
Tabela 01 - Especificação dos casos de uso.....	36-38
Tabela 02 - Dicionário de dados da tabela Professor.....	40
Tabela 03 - Dicionário de dados da tabela Sala.....	40
Tabela 04 - Dicionário de dados da tabela Organizacoes.....	41
Tabela 05 - Dicionário de dados da tabela Sala_alunos.....	42
Tabela 06 - Campos da tela inicial.....	43
Tabela 07 - Comandos da tela inicial.....	44
Tabela 08 - Campos da tela de cadastro.....	45
Tabela 09 - Comandos da tela de cadastro.....	45
Tabela 10 - Campos da tela de confirmação de email.....	46
Tabela 11 - Comandos da tela de confirmação de email.....	47
Tabela 12 - Comandos da tela de salas.....	48
Tabela 13 - Comandos da modal de confirmação.....	49
Tabela 14 - Comandos da modal para ingressar na sala.....	50
Tabela 15 - Campos da tela para envio do nome e rgm.....	51
Tabela 16 - Comando para validar os dados digitados.....	51
Tabela 17 - Campos para o estudante selecionar.....	51
Tabela 18 - Comandos da tela do questionário.....	53
Tabela 19 - Campos da sala.....	54
Tabela 20 - Comandos da sala.....	54

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	15
<b>1.1 OBJETIVOS.....</b>	16
<b>1.1.1 OBJETIVO GERAL.....</b>	16
<b>1.1.2 OBJETIVOS ESPECÍFICOS.....</b>	16
<b>1.2 TRABALHOS CORRELATOS.....</b>	17
1.2.1 TBL ACTIVE.....	17
1.2.2 DREAMSHAPER.....	18
1.2.3 Kahoot.....	19
<b>1.3 SOLUÇÃO PROPOSTA.....</b>	20
1.3.1 IDEIA.....	20
<b>2.0 REFERENCIAL TEÓRICO.....</b>	21
2.1 ENGENHARIA DE SOFTWARE.....	22
2.1.1 Benefícios da Engenharia de Software.....	23
2.1.2 Ciclo de vida do desenvolvimento de software.....	24
2.1.3 Gerenciamento de projetos de software.....	27
2.2 ALGORITMOS DE ORDENAÇÃO.....	27
<b>3.0 REFERENCIAL TECNOLÓGICO.....</b>	28
3.1 JAVASCRIPT.....	28
3.2 NODE JS.....	30
3.3 VUE JS.....	31
3.4 POSTGRESQL.....	32
3.5 HTML.....	32
3.6 CSS.....	33
<b>4.0 MODELAGEM DO SISTEMA.....</b>	33

4.1 UNIFIED MODELING LANGUAGE (UML).....	33
4.2 LEVANTAMENTO DE REQUISITOS.....	33
4.2.1 Diagrama de caso de uso.....	34
4.2.2 Especificação do caso de uso.....	36
4.3.2 Modelagem de dados.....	38
4.3.3 Diagrama de classes.....	38
4.3.4 Dicionário de dados.....	39
4.4 APRESENTAÇÃO DA PLATAFORMA.....	42
4.4.1 Tela inicial.....	43
4.4.2 Tela de cadastro.....	44
4.4.3 Tela de confirmação de e-mail ..,,.....	45
4.4.4 Tela de salas.....	47
4.4.5 Modal de confirmação de exclusão.....	48
4.4.6 Modal para ingressar na sala.....	49
4.4.7 Tela de identificação.....	50
4.4.8 Questionário de perfil.....	51
4.4.9 Tela de uma sala.....	53

## **1. INTRODUÇÃO**

O Team Based Learning (TBL) é uma metodologia adotada em sala de aula que é ativa e colaborativa, atividades em equipe e individuais fazem parte dessa metodologia e os estudantes são avaliados individualmente e coletivamente. O foco está em compreender conceitos chaves para a resolução de problemas e aplicar esses conceitos chaves em situações envolvendo problemas reais. O professor atua como um facilitador, interferindo nos grupos somente quando necessário, além de ser responsável por garantir que todos os alunos participem ativamente, também contribui para a organização dos grupos e na retirada de dúvidas. (GIACOMELLI, 2020)

Um exemplo prático dessa metodologia é uma pesquisa quali-quantitativa do tipo intervenção que foi feita sobre as contribuições da utilização da metodologia TBL usando a plataforma gratuita TBL Active, essa plataforma foi criada para ajudar no uso dessa metodologia ativa. Os resultados mostram um caso de uma professora que utilizou essa plataforma e teve resultados positivos como aprendizados colaborativos por incentivar discussões dos estudantes sobre os conceitos estudados na aula. A conclusão foi de que a metodologia com o uso da tecnologia foi bem aceita e teve bons resultados. (GIACOMELLI, 2020)

PROFIDINA Ágil é uma plataforma de gerenciamento de equipes voltada para ambientes educacionais que adotam metodologias ativas e ágeis, como o Team-Based Learning (TBL). A solução utiliza algoritmos para organizar os estudantes em grupos de forma equilibrada, fazendo com que estudantes com uma pontuação maior estejam no mesmo grupo de estudantes com uma pontuação baixa, os pontos são atribuídos considerando critérios como (habilidade técnica, perfil, tempo de ingresso na instituição),

A solução tem o objetivo de, além simplificar, agilizar o processo de formação de grupos, tendo em vista que os estudantes podem consumir muito do tempo da aula para fazer a formação de grupos que estejam de acordo com o limite mínimo e máximo estabelecido pelo professor. É a própria plataforma que possui a responsabilidade de fazer a organização dos estudantes.

Dos discentes, 81,5% concordaram totalmente que o trabalho em equipe proporcionou a aprendizagem colaborativa e 85,2% concordaram que o uso da tecnologia em sala proporcionou facilidade e praticidade às aulas. O TBL foi validado por discentes e docente que destacou sua pretensão em usá-lo novamente. (GIACOMELLI, 2020)

## **1.1 OBJETIVOS**

Este estudo trata-se de descrever, de forma geral, os resultados que se pretende obter com o uso do PROFIDINA e como será seu desenvolvimento

### **1.1.1 OBJETIVO GERAL**

Com a finalidade de Desenvolver um sistema inteligente de gerenciamento de equipes para ambientes educacionais que utilizam metodologias ágeis de aprendizagem baseada em equipe, aplicando algoritmos de ordenação para otimizar a formação de grupos e promover o protagonismo dos estudantes, esse estudo visa apresentar, o uso da solução de gerenciadores de grupos de trabalhos para a execução de atividades didáticas por grupos, como a metodologia do Team Based Learning onde professor não é mais o centro da aula, o único responsável pelo aprendizado (GIACOMELLI, 2020).

### **1.1.2 OBJETIVOS ESPECÍFICOS**

Será necessário para o êxito no desenvolvimento desse software

- **Levantamento de requisitos funcionais e não funcionais:** Identificar as necessidades dos usuários (professores e alunos) e definir os critérios de ordenação para formação de grupos.
- **Modelagem de dados e arquitetura do sistema:** Criar diagramas UML e banco de dados relacional que suportem a lógica de agrupamento dinâmico.
- **Construção de protótipos interativos:** Desenvolver interfaces intuitivas com ferramentas como Figma, priorizando usabilidade e acessibilidade.
- **Desenvolvimento da aplicação:** Utilizar tecnologias gratuitas como Node.js, Vue.js, PostgreSQL, HTML, CSS e JavaScript para construir o sistema.
- **Implementação de algoritmos de ordenação:** Integrar algoritmos como Bubble Sort, Quick Sort e Selection Sort para ordenar alunos com base na sua pontuação

## 1.2 TRABALHOS CORRELATOS

Este tópico pretende mostrar exemplos de softwares já existentes e que serão a inspiração para a plataforma de gerenciamento de grupos PROFIDINA ÁGIL.

### 1.2.1 TBL Active

TBL Active é uma plataforma para auxiliar na aplicação da metodologia ativa Team Based Learning que pode ser usada tanto em computadores como em smartphones. Nessa plataforma o professor pode criar dois tipos de questionários, o primeiro tipo é um que é respondido individualmente pelo estudante, nele, o professor escolhe um nome para o questionário, decide quantas alternativas cada pergunta terá, das alternativas que o professor criou ele escolhe qual será a correta, elabora as questões do questionário podendo fazer umas perguntas mais fáceis e outras mais difíceis por exemplo.

O professor, depois de ter criado o questionário, terá que decidir quanto valerá a avaliação individual, e quanto valerá a avaliação feita em grupo com relação a nota máxima. Depois será criada uma sala virtual com um número gerado pelo próprio software, os estudantes usarão esse número para entrar na sala e responder ao questionário. A plataforma foi desenvolvida por Ana Paula Ambrósio Zanelato Marques e William Alves Marques para auxiliar na aplicação da metodologia TBL (Team Based Learning) e possui uma versão grátis e duas versões pagas com melhorias como cronômetro e editor de texto.

A figura 01 demonstra uma imagem do site do TBL Active, esse software é um dos utilizado na metodologia Tem Based Learning

**FIGURA 01** - Imagem da plataforma TBL Active



Fonte: <https://www.tblactive.com.br/> (2025)

### 1.2.2 DreamShaper

O DreamShaper é uma ferramenta online que tem como foco ser utilizada na aplicação de metodologias que são não só ativas, mas também baseada em projetos, essa ferramenta é utilizada por instituições de ensino e por professores, DreamShaper é uma ferramenta que potencializa o protagonismo do estudante aumentando sua autonomia e motivação com trilhas de aprendizagem que, além de serem práticas, são adaptativas, isso faz com que a aprendizagem seja ativa e participativa.

No DreamShaper o professor pode estar associado a várias turmas, e as turmas também podem ter mais de um professor, o aluno pode estar em mais de um projeto e a ferramenta mostra em porcentagem o quanto do projeto está concluído. A ferramenta também mostra quais alunos estão dentro do mesmo projeto, o Dreamshaper oferece opções sobre qual tipo de projeto o estudante quer fazer, o aluno tem opções como projeto de pesquisa, projeto de cidadania entre outros.

O DreamShaper foi criado por pessoas que tinham uma ONG chamada Acredita Portugal, o objetivo dessa ONG era promover o empreendedorismo, para criar essa ferramenta tiveram ajuda de professores das universidades de Harvard e Stanford para criar o desenho do produto, o lançamento piloto foi no Brasil com o apoio da Fundação Lemann, hoje a ferramenta é utilizada tanto no Brasil como no Exterior, o DreamShaper é atualmente uma ferramenta paga, e utilizada por diversas instituições de ensino.

A figura 02 mostra uma imagem do site do DreamShaper

**FIGURA 02 - Imagem da ferramenta DreamShaper**



Fonte: <https://dreamshaper.com/pt-BR> (2025)

### 1.2.3 Kahoot

O Kahoot é uma plataforma muito útil em atividades em sala de aula, nele é possível fazer quizz que o estudante responde e ao final tem uma pontuação dos acadêmicos que tiveram o melhor desempenho, é uma excelente plataforma baseada em jogos. A plataforma Kahoot tem o objetivo de tornar fácil a criação de sessões de aprendizagem, nessas sessões uma pessoa ou empresa faz as perguntas e as demais respondem com tempo limite para cada pergunta. No Kahoot além de ser possível criar sessões de aprendizagem, é possível compartilhar essas sessões com outras pessoas,

No Kahoot é possível hospedar uma sessão de aprendizagem, a hospedagem torna não só possível poder aplicar novamente essas sessões de aprendizagem sem a necessidade de ter que criar uma nova, mas também facilita no compartilhamento com quem tiver interesse. A facilidade de criar, compartilhar e hospedar sessões de aprendizado impulsiona o aprendizado e acaba tornando o aprendizado algo mais envolvente.

O Kahoot foi fundado por Morten Versvik, Johan Brand, e Jamie Brooker, o projeto foi feito em conjunto com a Universidade Norueguesa de Ciência e Tecnologia (NTNU). Morten Versvik foi aluno de mestrado na NTNU e teve como professor Alf Inge Wang, as pesquisas feitas por Morten Versvik serviram de base para a criação do Kahoot, o professor Wang e o empreendedor Åsmund Furuseth se juntaram a equipe para contribuir com o desenvolvimento do Kahoot.

A figura 03 demonstra uma imagem do Kahoot

**FIGURA 03 - Imagem do Kahoot**



Fonte: [kahoot.com](https://kahoot.com) (2025)

Quadro 1 – Estudo de funcionalidades

Funcionalidade	DreamShaper	TBL Active	Kahoot	PROFIDINA
Formação de grupos somente pelo sistema				X
Questionário/Projetos/Teste	X	X	X	
Organização manual feita pelo professor	X	X	X	
ingresso manual em um grupo pelo aluno	X	X	X	
Foco exclusivo na organização de grupos				X
Múltiplas salas com múltiplo grupos	X	X		X
Priorizar uma organização equilibrada de grupos				X

Fonte: Autor (2025)

### 1.3 SOLUÇÃO PROPOSTA

Este capítulo tem como objetivo explicar a proposta do sistema de software PROFIDINA ÁGIL.

#### 1.3.1 Ideia

A ideia do PROFIDINA é que seja de fácil uso, que não seja necessário dar muitas informações para se fazer um login, que o usuário possa, de forma fácil, organizar os estudantes em sala de aula em atividades práticas, a organização será feita de forma rápida, o usuário vai gerar um QR Code e o estudante, com o celular, poderá entrar na sala criada pelo usuário. Existem estudantes que não gostam de trabalhar em

grupo, isso atrapalha no seu desenvolvimento como profissional, estudantes que estão em turmas grandes e que ninguém se conhece podem acabar levando muito tempo para eles mesmos fazerem a organização dos grupos, principalmente estudantes com jornadas de trabalho exaustivas

O PROFIDINA ÁGIL pretende facilitar a organização de grupos em sala de aula, o software irá ser o único responsável pela organização dos grupos. A plataforma terá uma funcionalidade oferecida para o usuário para a formação de grupos, e será de forma equilibrada, a ideia é que o usuário, quando escolher organizar os estudantes de forma heterogênea, poderá escolher, de forma simples, entre as opções de organizações que serão oferecidas pelo software PROFIDINA ÁGIL, limite de grupos e qual método de ordenação utilizar.

## 2.0 REFERENCIAL TEÓRICO

Tecnologias são mediadores de um novo paradigma educacional em que as metodologias ativas, como Team Based Learning, são pontos de partida para que haja um avanço em processos de reelaboração de novas práticas. É fundamental o uso de tecnologias na área da educação tendo em vista que vivemos, atualmente, em um período de mudanças no cenário educacional propiciadas pelas metodologias ativas que promovem o protagonismo do estudante ao favorecer sua própria aprendizagem através do incentivo a sua autonomia e do seu envolvimento participativo e reflexivo. (Marques et al, 2024)

Novas abordagens pedagógicas que visam além de valorizar o desenvolvimento de práticas em processos educativos, incentivar a aprendizagem desenvolvida por meio do próprio estudante é de suma importância, abordagens pedagógicas que descentralizam o professor no processo de aprendizagem através da adoção de metodologias ativas com apoio de Tecnologias Digitais de Informação e Comunicação (TDIC) podem contribuir para superar a aprendizagem centrada no professor. (Marques et al, 2024).

Tem Based Learning (TBL) ou Aprendizagem Baseada em Equipe é um método de ensino ativo que busca potencializar o aprendizado através do trabalho colaborativo, o TBL tem como objetivo incentivar o aluno a fazer sua própria busca por conhecimento, fazendo com que haja inversão em sala de aula entre estudantes e

professor(a) e tornando possível uma atitude mais ativa por parte do aluno no seu desenvolvimento do aprendizado e na própria construção do conhecimento. (Marques et al, 2024).

A figura 04 é sobre os benefícios das metodologias ativas

**FIGURA 04** - Imagem sobre benefícios de metodologias ativas



Fonte: [masteraulas.com](https://masteraulas.com) (2025)

## 2.1 ENGENHARIA DE SOFTWARE

Engenharia de software é a utilização de princípios da Engenharia para a construção de sistemas de softwares. No campo da Engenharia, que normalmente lida com sistemas físicos, existem princípios que são usados em seus sistemas, aplicar esses princípios para: especificar um sistema de software, fazer o projeto desse sistema de software, iniciar o seu desenvolvimento, testar esse sistema de software várias vezes, fazer a sua implementação e seu gerenciamento é o que chamamos de engenharia de software. (Kalinowski et al, 2023)

A construção de um sistema de software é composta de várias etapas, no processo de desenvolvimento, que é uma delas, a construção de um sistema de software tem como fator importante a programação, mas de forma alguma se resume somente a isso. Investir tempo em entender o que será programado é uma etapa fundamental, que de preferência é melhor que seja feita antes das outras etapas, essa etapa de entendimento do que programar, que é comumente chamada de requisitos, é

fundamental para que não haja um erro de entendimento com relação a como será o sistema a ser desenvolvido. (Kalinowski et al, 2023)

Como será feita a programação é outra etapa que exige trabalho em equipe e maturidade para decidir o projeto e a arquitetura do software. A fase de teste é de suma importância, nela é que realmente se decide se o que foi programado está certo, nessa etapa é feita repetidas revisões e testes de software. Depois que os testes foram feitos no software chega a hora de implantar a solução de forma eficiente em operação, essa última etapa se chama DevOps. (Kalinowski et al, 2023)

Existem muitas variantes em um desenvolvimento de um software, alguns exemplos dessas variantes são os custos de se desenvolver, determinar um prazo que seja possível de ser cumprido e a qualidade do software quando estiver pronto. Fatores como custo, prazo e qualidade ficam mais difíceis de serem determinados com precisão conforme o tamanho do software aumenta. É muito mais fácil, por exemplo, estimar o esforço para o desenvolvimento de uma pequena agenda pessoal do que para um sistema de gestão integrada envolvendo dados de várias áreas de negócio. (Pressman, Maxim, 2021)

### 2.1.1 Benefícios da Engenharia de Software

Benefícios da aplicação de boas práticas da Engenharia de Software são apontados em evidências científicas, benefícios no custo, prazo e qualidade de projetos de software são apontados nessas evidências. Aumento de produtividade e de redução de trabalho foram obtidos no investimento em princípios de engenharia de software, esse resultado foi obtido em uma investigação realizada por oito anos com centenas de empresas desenvolvedoras de software. (KALINOWSKI et al, 2023)

Engenharia de software possui técnicas e conceitos que, se não for aplicado corretamente, no desenvolvimento do projeto, possui efeitos notáveis como o aumento dos custos de manutenção do software crescendo exponencialmente em relação aos custo de desenvolvimento do software que foi desenvolvido sem os conceitos e as técnicas adequadas.(Kechi, 2012) Os custos de manutenção desses projetos de software tendem a ser consideravelmente reduzidos quando esses projetos são desenvolvidos com os cuidados de arquitetura e projeto de solução. (KALINOWSKI et al, 2023)

O apoio na escolha de tecnologias a serem empregadas na indústria fundamentada em conhecimento científico é sem dúvida um dos benefícios da engenharia de software baseada em evidências (SANTOS, TRAVASSOS, 2008). Em mercados competitivos como o da tecnologia, a melhoria contínua da capacidade de desenvolvimento é de suma importância para que empresas de software possam prosperar, não à toa existem modelos de referência que são guias para a melhoria da capacidade de processos de engenharia de software. (KALINOWSKI, 2012)

Na figura 04 há uma seta apontando para baixo, o objetivo dessa imagem é representar a redução nos custos de desenvolvimento ao adotar as técnicas de engenharia de software

**FIGURA 05 - Seta representando menos custos**



Fonte: [tiespecialistas.com](https://tiespecialistas.com) (2025)

## 2.1.2 Ciclo de vida do desenvolvimento de software

O termo Ciclo de Vida surgiu para fazer a descrição de um grupo de atividades e para descrever como essas atividades se relacionam entre si. O termo Ciclo de Vida surgiu de um estudo realizado da década de 70, em um período que é chamado de crise do software devido à alta demanda por sistemas e poucas técnicas de desenvolvimento. Existem várias fases em um desenvolvimento de software são elas análise, projeto, codificação e depuração, testes, implantação, manutenção e evolução do software fazem parte da criação de um sistema. (Polachini, Rocha, 2018)

Fases em um desenvolvimento de software é algo que é de suma importância e que sejam feitas de forma sequencial, caracterizando um ciclo de vida do sistema, e é esse ciclo de vida composto por fases colocadas em sequência que deve ter qualidade. Os motivos pelos quais adota-se um ciclo de vida para desenvolvimento de software é atingir objetivos como, definir as atividades a serem executadas em um projeto, ter coerência entre os vários projetos na mesma organização e ter pontos de checagem para controle de gerência e pontos de checagem para a decisão. (Santos, 2021)

Definir as atividades a serem executadas em um projeto antes de seu desenvolvimento se torna essencial quando novas pessoas estão constantemente entrando no desenvolvimento do sistema seja na gerência ou na programação, analistas de sistemas júnior podem ter dificuldades de entender como que a parte do software que estão desenvolvendo se encaixa no projeto de um modo geral, uma descrição apropriada de todas as fases do projeto ajuda a entender como que aquilo que o analista júnior está se esforçando pra fazer se encaixa no projeto como um todo. (Santos, 2021)

A coerência entre projetos é importante nos níveis mais altos de gerência, é difícil a supervisão de muitos sistemas, principalmente se cada um deles é feito de uma forma diferente. Pontos de checagem tanto para controle de gerência quanto para decisão varia de acordo com o tamanho do projeto, em projetos menores a checagem é o fim do projeto, em projetos sistemas maiores a checagem é feita em estágios para determinar se precisa de recursos adicionais ou se o projeto vai atrasar. (Santos, 2021)

Um dos modelos utilizados no desenvolvimento de software é o cascata, após 10 anos da crise do software nos anos 70 esse modelo ganhou visibilidade, dois fatores contribuíram para a sua ampla adoção, sua rigidez e o fato de não ser tão administrativo comparado com os outros modelos utilizados no desenvolvimento de software. O modelo cascata tem uma proposta de que o início do projeto seja o levantamento dos requisitos do cliente, somente depois que esta etapa de requisitos é totalmente completada é que começa a fazer o projeto de software. (Polachini, Rocha, 2018)

O modelo cascata possui tanto um ponto positivo quanto negativo, a vantagem de utilizar esse modelo é que primeiro uma etapa é totalmente concluída para então

seguir para a próxima fase do projeto, o lado que pode ser considerado negativo é a exigência de que a haja o levantamento de todos os requisitos já no início. Modelos como, prototipagem, modelo espiral, iterativo e incremental tiveram como base o modelo cascata. (Polachini, Rocha, 2018)

A figura 06 demonstra uma imagem do modelo cascata, esse é um dos modelos utilizados na engenharia de software

**Figura 06:** Modelo cascata



Fonte: [kxptech.com](http://kxptech.com) (2025)

Segundo MONDLANE (2023), existem vários modelos no ciclo de vida de desenvolvimento de software, ainda que uns sejam mais tradicionais e outros mais modernos a maioria deles possuem as seguintes fases:

- **Requisitos:** normalmente é etapa inicial, onde se entende as necessidades da organização
- **Design ou Projeto:** especificação detalhada das características do software, organização da equipe e determinar um prazo para o projeto
- **Codificação:** o desenvolvimento é feito com base nas fases anteriores,
- **Teste:** Verifica-se está indo de acordo com os resultados esperados
- **Implementação e manutenção:** Coloca-se o software em ambiente de produção para ser utilizado pelo usuário final, se houver alguma necessidade de mudança o sistema deve passar por uma manutenção

### 2.1.3 Gerenciamento de projetos de software

A figura 07 demonstra os ícones Sults, Monday, Jira e Bitrix24 são softwares utilizados no gerenciamento de projetos. Os ícones ERP, HCM, CRM e SCM, são siglas que correspondem a características de software para gestão empresarial integrada

**Figura 07** - Softwares utilizados no gerenciamento de projetos e características de software para gestão empresarial integrada



Fonte: [economiasc.com](https://economiasc.com) (2025)

O conjunto integrado de tecnologias chamado HCM permite a gestão dos funcionários por parte das empresas, recrutamento e seleção de talentos, folha de pagamentos e administração de benefícios são exemplos de uso do HCM. (ORACLE, 2025) Às ferramentas de planejamento dos sistemas ERP tornam possível a avaliação do impacto, que decisões em áreas estratégicas, tiveram exemplos de áreas estratégicas são suprimentos, finanças, e recursos humanos. (DANTAS, NEVES, 2024) CRM se refere a softwares para gestão de relacionamento com o cliente enquanto SCM é sobre a gestão da cadeia de fornecedores. (SORDI, MARINHO, 2026)

## 2.2 ALGORITMOS DE ORDENAÇÃO

Uma determinada sequência de etapas que são computacionais e que recebem uma entrada e transformam essa entrada em uma saída é uma das formas de descrever um algoritmo, outra forma é dizer que é um processo sistemático para a

resolução de um problema. Uma maneira mais detalhada de descrever um algoritmo é, uma lista finita de passos que tenham instruções bem definidas para resolver um problema específico, isto é, dada uma função “f” com uma entrada “x” é possível obter  $f(x)$  utilizando um algoritmo que foi escolhido levando em conta sua eficiência e complexidade. (PEDROSO, CINTRA, 2022)

Algoritmos de ordenação são muito importantes, para facilitar e tornar mais rápida a busca por uma informação específica, um exemplo é buscar o número de uma pessoa específicas que está no meio de muitos contatos, é possível tornar a busca por um dado específico menos complexa se os dados estiverem ordenados. O bubble sort é um algoritmo de ordenação que compara dois valores, se o primeiro valor for maior que o segundo, é feita uma troca de posição, o bubble sort repete esse procedimento até que nenhuma troca seja realizada por todos os valores estarem organizados. (PEDROSO, CINTRA, 2022)

O Algoritmo de ordenação chamado de selection sort busca até encontrar o menor valor entre os valores de entrada, coloca esse valor encontrado na primeira posição e volta a buscar o segundo menor valor entre os valores, para colocar na segunda posição, o processo é repetido até que todos os valores estejam ordenados. Quick sort é um algoritmo de ordenação que consiste em escolher o escolher um valor do vetor como pivô, o algoritmo divide o vetor em dois, de um lado, um subvetor que contém apenas valores menores que o pivô, do outro lado, um subvetor que contém valores que são maiores, o algoritmo de ordenação quick sort repete esse procedimento em cada subvetor até que todo o vetor seja ordenado. (PEDROSO, CINTRA, 2022)

### **3.0 REFERENCIAL TECNOLÓGICO**

Este capítulo tem como objetivo a demonstração de tecnologias que serão usadas no desenvolvimento do software PROFIDINA ÁGIL.

#### **3.1 JAVASCRIPT**

JavaScript foi criado por Brendan Eich que além de programador, foi cofundador do Mozilla Firefox, JavaScript foi um pedido da Netscape, o objetivo era tornar os sites mais dinâmicos, uma das utilizações foi a validação de formulários

(PEREIRA, 2024). JavaScript é muito utilizado no desenvolvimento dentro do ambiente da rede mundial de computadores, Além do JS(JavaScript) ser usado pela ampla maioria dos sites modernos, todos os navegadores mais atuais possuem interpretadores para o JS, isso faz com que tenha uma grande presença na web. (GUEDES et al, 2021)

As características do JavaScript incluem ser multiparadigma (estruturado, orientado a objetos e funcional), ser fracamente tipado e interpretado. Sua criação remete a meados de 1990. Com o objetivo de tornar a experiência do usuário a mais rica possível, a principal aplicabilidade da linguagem JavaScript em um sistema web é a sua interatividade, essa característica torna o JS uma peça fundamental na construção de um sistema. (GUEDES et al, 2021)

Esse dinamismo que o JavaScript oferece nas páginas da internet, faz com que seu uso traga infinitas possibilidades. Vale ressaltar que uma característica interessante do JavaScript é o fato de poder ser interpretado tanto do lado do cliente, no navegador da WEB, como também no lado do servidor, por exemplo, utilizando Node.js. A vasta presença do JS faz com que a maioria das páginas da internet execute código JavaScript, no cotidiano das pessoas já é uma realidade a presença do JS como quando se faz uma pesquisa no Google simples, leitura em portais de notícias pela internet ou quando se utiliza redes sociais. (GUEDES et al, 2021)

A figura 08 mostra um código JavaScript feito pelo autor, o código verifica se o usuário é maior de idade ou não com base no valor armazenado na variável

**Figura 08:** código JS

```
let augusto = 20

function checkarIdade(idade) {
    if (idade >= 18) {
        return "maior de idade";
    } else {
        return "menor de idade";
    }
}
console.log(checkarIdade(augusto));
```

Fonte: Autor (2025)

### 3.2 NODE.JS

O Node.js possui uma grande importância para o JavaScript devido ao seu ecossistema node.js e NPM. De baixo custo e altamente escalável, Node JS foi construído sobre o motor do JavaScript, essa plataforma além de permitir que bibliotecas que acessam recursos do sistema operacional sejam utilizadas, proporciona para o desenvolvedor a programação direta com variados protocolos de internet e rede. (GUEDES et al, 2021)

O propósito do Node de transferir a responsabilidade do JavaScript do lado do navegador, que seria o cliente, para um servidor, no caso o Node torna possível, por exemplo, a criação de um servidor web, ou seja, possibilidade de desenvolvimento de aplicações que possuem uma escalabilidade alta, como uma capacidade que pode chegar a milhares de conexões simultâneas e em tempo real.

NPM é o nome do gerenciador de pacotes que o Node.js possui, a popularidade desse gerenciador de pacotes na comunidade foi tão grande que desde a versão 0.6.0 do Node JS o NPM passou a ser o gerenciador padrão, ou seja, se integrou ao instalador, essa integração simplificou a vida dos programadores por ter feito uma grande quantidade de projetos serem convergidos para essa plataforma. O gerenciador de pacotes NPM (Node Package Manager) proporciona ao desenvolvedor uma grande flexibilidade. (GUEDES et al, 2021)

Essa flexibilidade trazida pelo NPM faz com que o Node.js seja uma plataforma que desempenha uma importante função na utilização de bibliotecas e frameworks JavaScript. O fato de o NPM ser um grande repositório de software utilizado no mundo o torna indispensável nos frameworks JS utilizados atualmente, pois hoje em dia, a construção de um sistema de software web com um framework é necessário o gerenciamento de dependências, gerenciamento de versão, instalação de pacotes e outros recursos mais. (GUEDES et al, 2021)

A figura 09 mostra a utilização do node.js para executar o código JavaScript da figura 08, na figura 09 é possível ver o uso do node.js e o output(saída) que é “maior de idade”

**Figura 09:** exemplo simples do uso de node

```
>> node index.js  
maior de idade
```

Fonte: Autor (2025)

### 3.3 VUE.JS

O framework progressivo chamado Vue.js é utilizado para a construção de interfaces de usuário, esse framework JS foi pensado para que sua utilização seja de forma incremental, o principal objetivo do Vue.js é na aplicação front-end, ou seja é atuar na camada de visualização de uma aplicação. A capacidade de construir aplicações robustas SPA (Single Page Application) é uma característica do Vue.js por conseguir facilmente ser integrado à outros projetos e bibliotecas que já existem, outra característica do Vue.js é a sua simplicidade. (GUEDES et al, 2021)

Evan You é o nome da pessoa que criou o Vue.js, Evan trabalhava na Google usando Angular, que é uma tecnologia da própria empresa, mas estava insatisfeito em usar a ferramenta para prototipagem rápida, nessa época o React era uma tecnologia muito recente, JS já estava sendo utilizado em aplicações de larga escala e com arquitetura MVC. Evan pensava que nenhuma dessas tecnologias supria uma necessidade que ele tinha, que era a de uma ferramenta flexível e leve, que seja focada na prototipagem rápida UI (User Interface). (GUEDES, 2021)

Foi a partir dessa situação que Evan You decidiu criar um framework que suprisse esse vácuo criado pela ausência de uma tecnologia rápida e flexível. O objetivo principal com a criação do Vue.js era criar um framework que oferecesse uma forma flexível e fácil de fazer a ligação com dados reativos, componentes reutilizáveis, prototipação rápida e a possibilidade de desenvolver aplicativos escaláveis e reativos na web. (GUEDES, 2021)

A figura 10 traz um exemplo simples de um código feito em Vue 3 em que o componente Item é implementado no App.vue.

**Figura 10:** código em Vue.js

```
<template>
|  
|  <Items />
</template>
<script>
import Items from './components/Items.vue'
export default {
  name: 'App',
  components: [
    Items
  ]
}
</script>
```

Fonte: Autor (2025)

### 3.4 POSTGRESQL

O sistema de gestão de bases de dados relacionais (SGBDR) chamado PostgreSQL é muito parecido com outros bancos de dados como o MySQL por exemplo, a gerência de dados que possuem um alto rendimento necessita de um banco capaz de fazer a gestão desses dados de forma eficiente, PostgreSQL além de ser capaz de ser uma ótima solução funciona em diversos sistemas operacionais como Windows, Linux, Mac OS e Unix, uma vantagem do PostgreSQL é o fato de ser grátis e open source. (CALDEIRA, 2025)

### 3.5 HTML

Tim Beners-Lee era pesquisador da Organização Europeia de Pesquisas Nucleares, A forma que Tim e sua equipe encontraram para não só redigir mas também publicar com facilidade suas pesquisas na World Wide Web, foi desenvolver uma linguagem de marcação de texto chamada HTML (HyperText Markup Language) o início do desenvolvimento web se dá por volta de 1990, antes do desenvolvimento

do HTML, a comunicação era limitada a engenheiros, pesquisadores e funcionários do governo. (PEREIRA et al, 2025)

### 3.6 CSS

Cascading Style Sheets(CSS) tem como função principal no desenvolvimento front-end, determinar como será feita a apresentação do conteúdo de uma página, o CSS é uma linguagem que estiliza o conteúdo de uma página através de cores, fontes e layouts, ao utilizar os css, modifica-se o elemento HTML que o CSS está vinculado por meio de seletores, propriedades e valores. Ter um código CSS bem-organizado é, além de um fator importante, um desafio para os desenvolvedores, uma vez que a programação ocorre com muitas edições de código.(SILVA, 2025)

## 4.0 MODELAGEM DO SISTEMA

Este capítulo tem como objetivo demonstrar de forma detalhada, o que se objetiva com o desenvolvimento do software PROFIDINA ÁGIL, que tem como foco principal, utilizar algoritmos de ordenação para fazer a ordenação de estudantes, agrupá-los de forma equilibrada. O foco é demonstrar como ocorre a interação do usuário com o sistema e a responsabilidade que cada ator terá.

### 4.1 UNIFIED MODELING LANGUAGE (UML)

A Linguagem Unificada de Modelagem (UML), é uma linguagem gráfica que é usada em artefatos de sistemas complexos de software, a UML é utilizada na visualização, especificação, construção e documentação do que chamamos de artefatos. A UML permite padronizar a preparação dos planos de arquitetura de projeto de sistemas, processos de negócio, funções do sistema, classes e esquemas de banco de dados que são conceitos que fazem parte da UML (BOOCH, 2006)

### 4.2 LEVANTAMENTO DE REQUISITOS

As expectativas e necessidades do usuário em relação ao sistema de software precisam ser, além de identificadas, documentadas e verificadas, para isso se utiliza o levantamento de requisitos para entender não só o contexto em que deve ser

considerado durante o desenvolvimento do software, mas também os desejos com relação as funcionalidades do sistema e as restrições que devem ser levadas em conta durante o desenvolvimento do sistema de software. (SILVA, 2023)

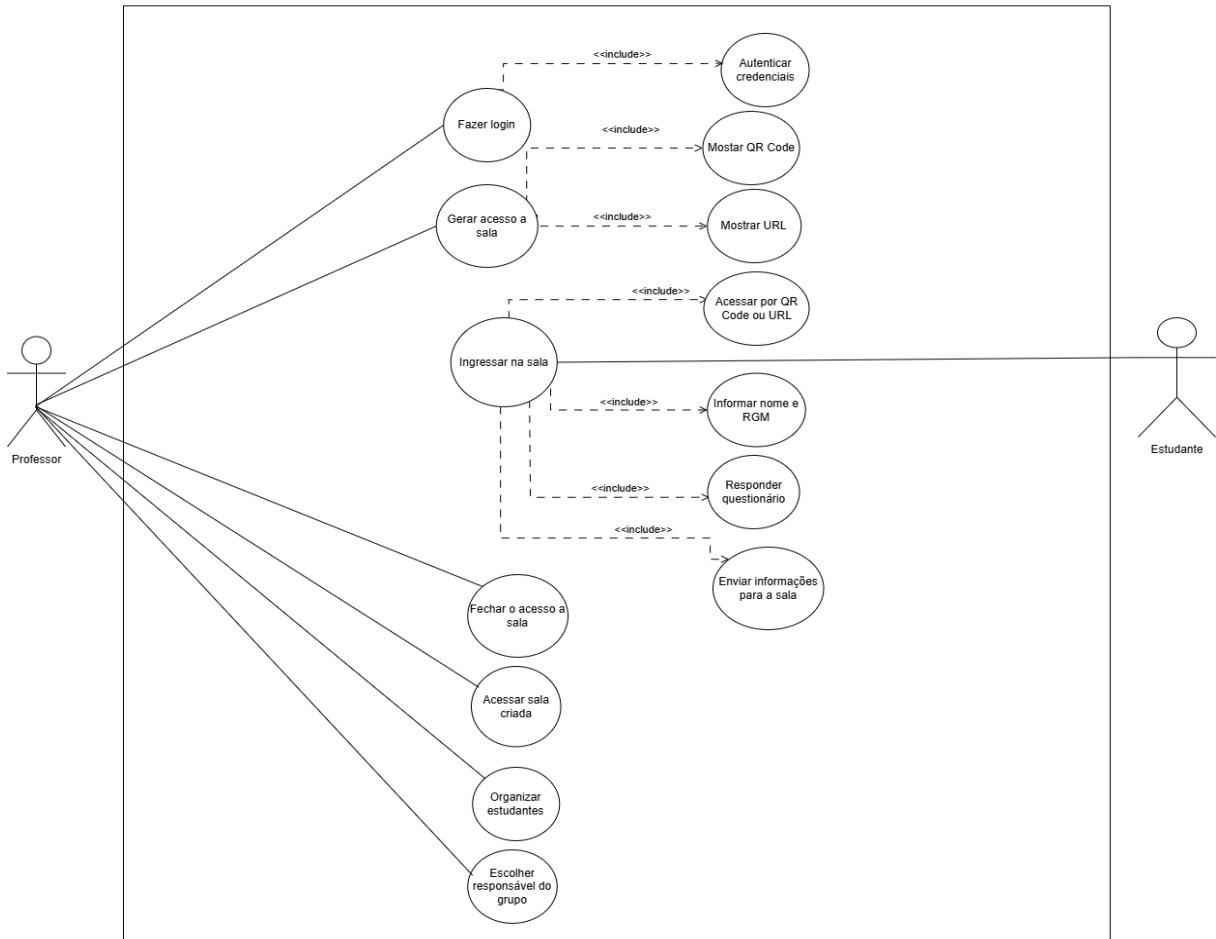
O software PROFIDINA ÁGIL deve ajudar o professor em sala de aula na organização de grupos, o usuário não terá a possibilidade de decidir em qual grupo determinado estudante fará parte, os grupos de estudantes serão feitos de forma equilibrada, e será de responsabilidade do software fazer essa organização. Para fazer a organização dos grupos, algoritmos de ordenação serão de suma importância para o sistema.

#### 4.2.1 Diagrama de caso de uso

O diagrama de caso de uso possui nos seus relacionamentos dependência, associação e generalização com o objetivo de trazer uma visão estática dos casos de uso que existem no sistema. O elemento sintático “ator” é utilizado para representar a forma como os elementos externos interagem com as funcionalidades do sistema de software. Ter essa visão estática que o diagrama de caso de uso fornece é fundamental para o entendimento do comportamento do sistema, principalmente para compreender as funcionalidades que são externamente visíveis oferecidas pelo sistema. (da Silva et al, 2017)

A seguir, a figura 11 apresenta o Diagrama de Caso de uso que mostra a interação de atores externos com as funcionalidades do sistema de software PROFIDINA ÁGIL. A figura 11 também demonstra a interação entre funcionalidades.

**Figura - 11 Diagrama de caso de uso**



Fonte: Autor (2025)

No diagrama de caso de uso acima é possível observar que é o professor quem faz o login na plataforma de gerenciamento de equipes, e é quem gera o acesso a sala para os estudantes poderem ingressar na sala criada pelo próprio professor. Os acadêmicos, após escanear o QR Code ou utilizar a URL, poderão ingressar no sistema, informando seu nome, RGM e respondendo o questionário para a avaliação de perfil se feita.

O professor, após os estudantes terem enviado seus dados juntamente com sua pontuação resultante do questionário, pode fechar a tela que mostra tanto o QR Code quanto a URL para o estudante ingressar na sala, depois do professor fechar a tela, acessa a sala onde pode decidir a organização dos grupos em sala de aula,

estabelecendo limite de estudantes por grupo e o algoritmo de ordenação a ser utilizado.

Após definir como os grupos devem ser feitos, a plataforma faz a organização dos estudantes, ou seja, a própria plataforma de gerenciamento de equipes formará os grupos. Com os grupos formados, o próprio professor escolhe um líder de cada grupo, somente um estudante pode ser selecionado.

#### 4.2.2 Especificação do caso de uso

O diagrama de caso de uso foi apresentado no item anterior, a seguir serão mostradas as interações dos atores externos com os casos de uso internos do sistema da plataforma de gerenciamento de equipes PROFIDINA ÁGIL.

**Tabela 01:** Especificação dos casos de uso

<b>Nome do caso de uso</b>	Sistema de grupos PROFIDINA ÁGIL
<b>Atores principais</b>	Professor e estudante
<b>Atores secundários</b>	Sistema de software
<b>Resumo</b>	Este caso de uso demonstra a interação dos atores externos com as funcionalidades internas do sistema para criar salas, definir a formação de grupos, e participar dos grupos
<b>Pré-condições</b>	
<b>Pós-condições</b>	Depois do cadastro, professor poderá logar e ir para a tela de salas
<b>Fluxo principal</b>	<ol style="list-style-type: none"><li>1. Sistema exibe uma tela usada para login ou para ir para tela de cadastro ao clicar em “Cadastre-se”</li><li>2. Usuário seleciona “Cadastre-se”</li><li>3. Sistema exibe uma tela para o usuário informar nome, e-mail, criar uma senha, confirmar a senha e clicar no botão “Enviar Código de Verificação”</li><li>4. Usuário coloca o nome, e-mail, cria uma senha, confirma a senha e seleciona o botão “Enviar Código de Verificação”</li></ol>

5. Sistema exibe uma tela para o usuário informar o código que o sistema enviou ao e-mail informado e depois clicar no botão “Confirmar e Cadastrar”
6. Usuário informa o código que foi enviado para seu e-mail e clica no botão “Confirmar e Cadastrar”
7. Sistema cria uma conta com o nome, e-mail e a senha criada
8. Sistema retorna para a tela de login
9. Usuário entra com os dados cadastrados
10. Sistema vai para a tela de salas
11. Usuário cria uma sala
12. Depois de criada, a sala aparece na tela de sala, com nome, descrição, código, quantidade de alunos, um botão para entrar na sala e um botão para mostrar o QR Code
13. Usuário clica no botão “Gerar Acesso” para os estudantes poderem entrar na sala
14. Sistema exibe uma tela com que mostra o QR Code e uma URL
15. Estudante usa o celular para ler o QR Code e entrar na sala
16. O sistema exibe uma tela com três campos, uma para o código da sala e dois para os estudantes se identificarem, e um botão para continuar para a próxima tela
17. Os acadêmicos preenchem com o código da sala, nome, sobrenome e RGM
18. Os estudantes clicam em “Continuar”
19. A plataforma exibe uma tela com diversas alternativas para avaliar o perfil do estudante, quanto mais alternativas o acadêmico marcar, maior será sua pontuação
20. O Estudante marca as opções que estão de acordo com seu perfil e clica no botão “Enviar informações”
21. O sistema cria os usuários com os dados fornecidos pelos estudantes, RGM, nome e sua pontuação
22. Na tela de salas, o professor, após os estudantes terem enviado suas informações, fecha a tela que mostra o QR Code e a url da sala
23. Professor clica no botão “Entrar na sala” e entra na sala
24. Dentro da sala o professor estabelece um número de estudantes por grupo, seleciona o algoritmo de ordenação e clica no botão “Organizar Grupos”
25. A organização dos grupos é feita
26. Professor seleciona um aluno de cada grupo como o responsável / líder

#### **Fluxo alternativo - informações erradas**

1. O sistema exibe uma tela com um campo para os alunos se identificarem
2. O aluno coloca informações erradas
3. Professor exclui usuário com informações erradas

#### **Fluxo alternativo – e-mail já existe**

1. O sistema exibe uma tela para o professor se cadastrar.
2. Professor informa nome, um e-mail que já foi cadastrado, senha e confirma a senha.
3. Sistema exibe uma mensagem dizendo que o e-mail já está cadastrado

4. Professor informa um e-mail que não foi cadastrado ainda e clica em Enviar código de Verificação.
5. Sistema envia o código de verificação e exibe uma tela para que o professor possa informar o código enviado para seu e-mail.
6. Professor informa o código recebido e o sistema cria o usuário.

Fonte: Autor (2025)

#### 4.3.2 Modelagem de dados

Um banco de dados possui informações conceituais, compreender e representar o mundo real é o objetivo da modelagem conceitual para que se possa obter as características a serem estudadas do mundo real, o principal componente de informações conceituais de um banco de dados é a modelagem de dados. Um fato importante é que, adotar o diagrama de classe UML como alternativa ao modelo ER (Entidade Relacionamento) já é uma realidade em muitas empresas (FRANK et al. 2021)

#### 4.3.3 Diagrama de classes

O diagrama de classe da UML é um dos mais utilizados por conseguir, por meio da representação de suas classes, atributos, operações e relações, fazer a demonstração da estrutura lógica de software orientados a objetos. Na modelagem conceitual os seguintes elementos são tidos como os mais importantes quando se utiliza o diagrama de classe: Nome da classe, atributos com tipos, associações com suas multiplicidades e relações de generalização (Souza Filho,2023).

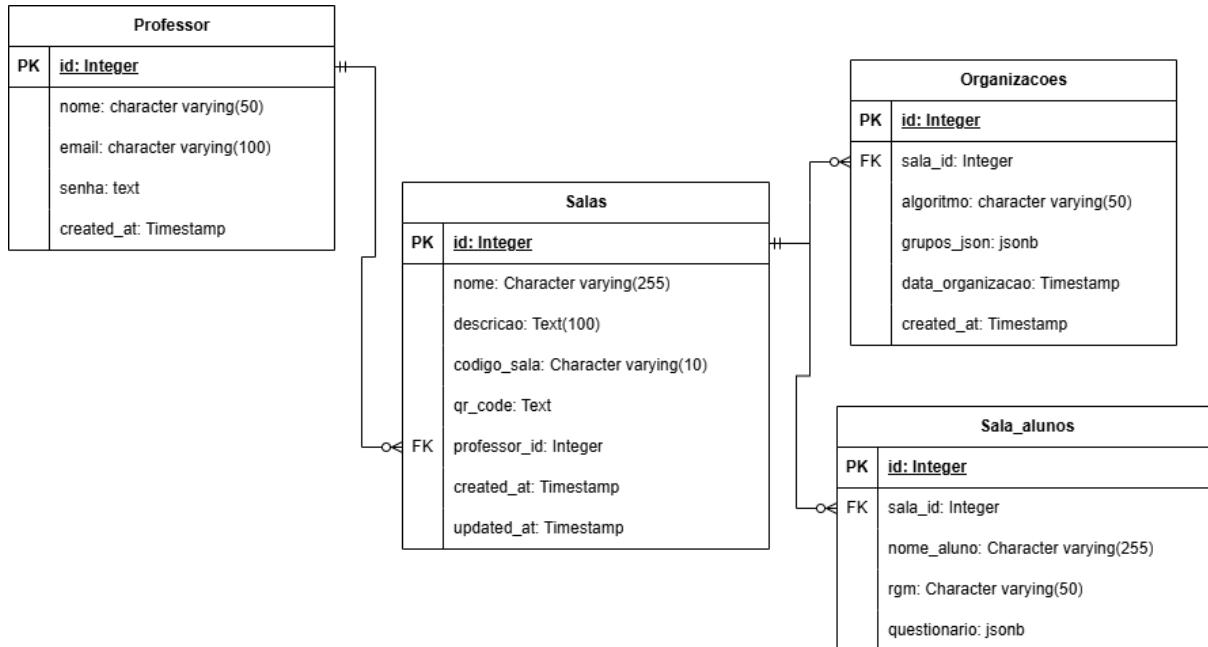
Segundo de Oliveira et al (2025) na UML os mais comumente usados na modelagem dos sistemas são:

- **Diagrama de casos de uso:** modelar as interações entre o usuário e o sistema
- **Diagrama de classes:** representa a estrutura estática do sistema, é usado na modelagem da arquitetura do software
- **Diagrama de sequência:** mostrar como ocorre o fluxo de controle e interação entre componentes do sistema,
- **Diagrama de atividade:** usado para modelar fluxos de trabalho

- **Diagrama de componentes:** descreve a arquitetura de sistemas distribuídos
- **Diagrama de máquina de estados:** Estados que um objeto passa durante o ciclo de vida.
- **Diagrama de implementação:** Usa na modelagem de infraestrutura física do sistema
- **Diagrama de pacotes:** Organizar classes e outros elementos em grupos

Na figura a seguir, será apresentado uma imagem Diagrama Entidade Relacionamento (DER) que será usado no desenvolvimento do sistema de software PROFIDINA ÁGIL

**Figura - 12** Diagrama (DER) do sistema



Fonte: Autor (2025)

#### 4.3.3 Dicionário de dados

No próximo bloco será apresentado o dicionário de dados das tabelas do Diagrama Entidade Relacionamento que se apresenta na figura 12, as tabelas desse diagrama DER são utilizadas no sistema de software PROFIDINA ÁGIL.

A tabela 02 demonstra o dicionário de dados da tabela Professor

**Tabela 02:** dicionário de dados da tabela Professor

Tabela - Professor					
Definição		Armazena dados do professor			
Coluna	Tipo	Tamanho	Restrições	Valor	Descrição
ID	INT	N/A	PK	N/A	Identificador único do professor
nome	CHAR	100	N/A	N/A	Nome do professor

Fonte: Autor (2025)

A tabela 03 demonstra o dicionário de dados da tabela Salas

**Tabela 03:** dicionário de dados da tabela Salas

Tabela - Salas					
Definição		Armazena dados da sala			
Coluna	Tipo	Tamanho	Restrições	Valor Padrão	Descrição
ID	INT	N/A	PK	N/A	Identificador único da sala
nome	CHAR	255	N/A	N/A	Nome da sala
descricao	TEXT	100	N/A	N/A	Descrição da sala
codigo_sala	CHAR	10	N/A	N/A	Código utilizado para que o estudante ingressa na sala
qr_code	TEXT	N/A	N/A	N/A	QR Code utilizado para ir para tela onde o estudante enviará suas informações

professor_id	INT	N/A	FK	N/A	Aponta para a chave primária da tabela Professor.
created_at	TIMESTAMP	N/A	N/A	N/A	Data em que a sala foi criada
updated_at	TIMESTAMP	N/A	N/A	N/A	Data da última atualização da sala

Fonte: Autor (2025)

A tabela 04 demonstra o dicionário de dados da tabela Organizacoes

**Tabela 04:** dicionário de dados da tabela Organizacoes

Tabela – Organizacoes					
Definição		Armazena dados das organizações feitas na sala			
Coluna	Tipo	Tamanho	Restrições	Valor Padrão	Descrição
ID	INT	N/A	PK	N/A	identificador único da organização
sala_id	INT	N/A	FK	N/A	Aponta para a chave primária da tabela Salas
algoritmo	CHAR	50	N/A	N/A	Algoritmo utilizado para fazer a ordenação dos estudantes
grupos_json	JSONB	N/A	N/A	N/A	Armazena os grupos formados em sala
data_organizacao	TIMESTAMP	N/A	N/A	N/A	Data da organização dos grupos enviada pelo frontend
created_at	TIMESTAMP	N/A	N/A	N/A	Data da organização criada no próprio backend

Fonte: Autor (2025)

A tabela 05 demonstra o dicionário de dados da tabela Sala\_aluno

**Tabela 05:** dicionário de dados da tabela Sala\_alunos

Tabela – Sala_alunos					
Definição		Armazena dados que os alunos vincularam a sala ao enviar suas informações e responder o questionário			
Coluna	Tipo	Tamanho	Restrições	Valor Padrão	Descrição
ID	INT	N/A	PK	N/A	Identificador único do aluno
sala_id	INT	N/A	FK	N/A	Aponta para a chave primaria da tabela Salas
nome_aluno	CHAR	255	N/A	N/A	Nome do estudante
rgm	CHAR	50	N/A	N/A	Matrícula do acadêmico
questionário	JSONB	N/A	N/A	N/A	Armazena o resultado do questionário respondido pelo estudante

Fonte: Autor (2025)

#### 4.4 APRESENTAÇÃO DA PLATAFORMA

A seguir, será apresentado os a plataforma de gerenciamento de equipes PROFIDINA ÁGIL, o objetivo é, além de demonstrar as principais funcionalidades que o sistema oferece, mostrar como ocorre a interação do usuário com o software PROFIDINA ÁGIL, também será mostrado as tabelas que demonstram cada funcionalidade dos botões e dos campos que o sistema de software PROFIDINA ÁGIL tem.

#### 4.4.1 Tela inicial

A seguir, Na Figura 13, é possível ver a demonstração da tela onde o usuário pode logar ou ir para a tela de cadastro

**Figura 13:** Protótipo - Tela inicial



Fonte: Autor (2025)

A tabela 06 apresenta os campos da tela de login.

**Tabela 06:** Campos da tela inicial

Nome do Campo	Tipo	Máscara	Obrigatório	Observação
Email	Texto	N/A	Sim	N/A
Senha	Alfanumérico	N/A	Sim	N/A

Fonte: Autor (2025)

A tabela 07 apresenta o Botão entrar, para o usuário poder ir para a tela de salas, e o link Cadastre-se, para poder fazer o cadastro

**Tabela 07:** Comandos da tela inicial

Comando	Ação	Observações
Entrar	Redireciona para a tela de salas	As informações, email e senha, deve já estar cadastradas
Cadastre-se	Redireciona para a tela de cadastro	N/A

Fonte: Autor (2025)

#### 4.4.2 Tela do professor

A figura 14 apresenta a tela em que o usuário poderá fazer seu cadastro na plataforma de gerenciamento de grupo

**Figura 14:** Tela de cadastro

A imagem é uma captura de tela de uma interface de usuário para cadastro. O formulário tem um fundo cinza escuro com uma caixa central branca. No topo, o título "Cadastrar" está escrito em azul. Abaixo dele, há um link "Voltar" em azul. O formulário contém quatro campos de texto: "Usuário", "Email", "Senha" e "Confirmar Senha". Cada campo tem uma barra de texto acima e uma barra de validação abaixo. À direita de cada campo de senha, há ícones de olho para alternar visibilidade. Abaixo dos campos, há um botão "Enviar Código de Verificação" com um fundo azul e textos brancos.

Fonte: Autor (2025)

A tabela 08 apresenta os campos para a pessoa preencher para realizar o cadastro.

**Tabela 08:** Campos da tela de cadastro

Nome do Campo	Tipo	Máscara	Obrigatório	Observação
Usuário	Texto	N/A	Sim	N/A
Email	Texto	N/A	Sim	Tem que ter um formato válido
Senha	Alfanumérico	N/A	Sim	N/A
Confirmar Senha	Alfanumérico	N/A	Sim	Tem que ser igual a primeira senha

Fonte: Autor (2025)

Na tabela 9 são apresentados o comando “Enviar Código de Verificação” que leva para uma tela de onde é possível confirmar o código recebido no email que foi informado na tela de cadastro

**Tabela 9:** Comandos da tela de cadastro

Comando	Ação	Observações
Enviar Código de Verificação	Redireciona para a próxima tela onde é possível confirmar o email	Verificar a caixa de SPAM

Fonte: Autor (2025)

#### 4.4.3 Tela de confirmação de e-mail

A seguir a figura 15 se refere a tela de confirmação do e-mail, utilizada para informar o código recebido por e-mail e realizar o cadastro.

**Figura 15:** Tela de confirmação de e-mail



Fonte: Autor (2025)

Na tabela 10 é apresentado os campos da tela de confirmação de e-mail do professor

**Tabela 10:** Campos da tela de confirmação de e-mail

Nome do Campo	Tipo	Máscara	Obrigatório	Observação
Código de Verificação	Texto	N/A	Sim	N/A

Fonte: Autor (2025)

Na tabela 11 é mostrado o comando da tela de confirmação do e-mail.

**Tabela 11:** Comandos da tela de confirmação de e-mail

Comando	Ação	Observações
Confirmar e Cadastrar	Confirma o e-mail fornecido	Cadastra o usuário

Fonte: Autor (2025)

#### 4.4.4 Tela de salas

A figura 16 demonstra a tela onde é possível visualizar todas as salas criadas, além da visualização é possível gerar acesso a sala e entrar para saber quantos e quais estudantes ingressaram.

**Figura 16:** Tela de salas

The screenshot shows a user interface for managing rooms. At the top, there's a header with 'Bem Vindo augusto' and an email address 'augusto.liberato10@gmail.com'. Below the header, there are two buttons: 'Ver Salas' (View Rooms) and 'Ver Perfil' (View Profile). The main area is titled 'Minhas Salas' (My Rooms) and contains three cards, each representing a room:

- Logica**: Código: 177ADF, Alunos: 0, Criada em: 29/11/2025. Buttons: Entrar na sala, Gerar Acesso.
- Programação**: Código: CDA7F0, Alunos: 0, Criada em: 20/11/2025. Buttons: Entrar na sala, Gerar Acesso.
- física**: Código: 551171, Alunos: 1, Criada em: 25/10/2025. Buttons: Entrar na sala, Gerar Acesso.

At the bottom right of the card area, there's a button labeled 'Criar Sala' (Create Room).

Fonte: Autor (2025)

Na tabela 12 é mostrado os comandos da tela de salas do usuário que fez seu cadastro com sucesso, conseguiu efetuar o login na conta cadastrada e criou salas vinculadas a sua conta.

**Tabela 12:** Comandos da tela de salas

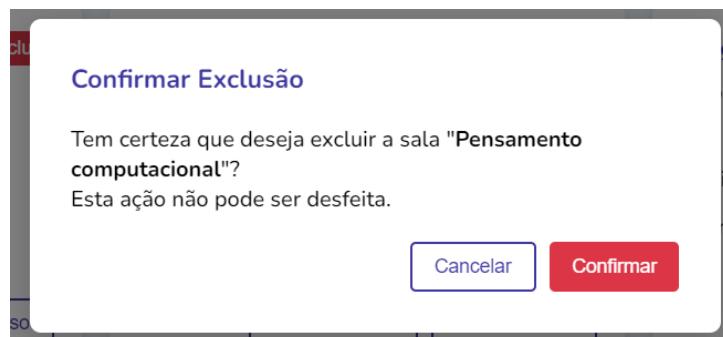
Comando	Ação	Observações
Ver Salas	Exibe a tela de salas	A tela de sala é, por padrão, exibida primeiro ao logar.
Ver Perfil	Vai para a tela onde tem informações do usuário logado	N/A
Sair	Desloga e vai para a tela de login	N/A
Criar Sala	Exibe um modal	Nesse modal é informado nome e descrição da sala
Editar	Abre um modal para editar o nome e a descrição da sala	N/A
Excluir	Exibe um modal para confirmar se quer realmente excluir	N/A
Copiar	Copia o código da sala	Código utilizado para informar em qual sala o estudante envia suas informações
Entrar na sala	Entra na sala criada	N/A
Gerar Acesso	Exibe um modal que exibe um QR Code e um link	Tanto o QR Code quanto o link podem ser utilizados para ingressar na sala

Fonte: Autor (2025)

#### 4.4.5 Modal de confirmação de exclusão

A seguir, na figura 17, é demonstrado na tela um modal de confirmação de exclusão que aparece ao clicar em Excluir para deletar uma sala.

**Figura 17:** modal de confirmação



Fonte: Autor (2025)

Na tabela 13 é mostrado os comandos do modal, onde o usuário poderá decidir se confirma a exclusão ou se cancela a ação de deletar a sala.

**Tabela 13:** Comandos da modal de confirmação

Comandos	Ação	Observações
Cancelar	Fecha o modal de confirmação de exclusão de uma sala	N/A
Confirmar	Confirma a exclusão de uma sala	Ação irreversível.

Fonte: Autor (2025)

#### 4.4.6 Modal para ingressar na sala

Na figura 18 é mostrado o modal utilizado pelos estudantes para escanearem e irem para a tela onde poderão enviar seus dados para a sala

**Figura 18:** Tela para formação e gerenciamento de grupo



Fonte: Autor (2025)

A tabela 14 demonstra os comandos do modal utilizado para ingressar em uma sala.

**Tabela 14:** Comandos da modal para ingressar na sala

Comando	Ação	Observações
QR Code	Escanear link pela câmera do celular	N/A
Fechar	Fecha o modal do acesso a sala	N/A

Fonte: Autor (2025)

#### 4.4.7 Tela de identificação

Na figura 19 é demonstrado a tela para o estudante informar seu nome e rgm, para serem enviados a sala criada pelo professor

**Figura 19:** Tela para envio do nome e rgm

Ingressar na Sala  
Preencha seus dados para entrar na sala

Seus Dados

Código da Sala \*

12312

Nome e Sobrenome \*

Marcos

Seu RGM (Registro Geral de Matrícula) \*

302388

Digite apenas números

Continuar →

Fonte: Autor (2025)

A seguir, a tabela 15 mostra os campos que o estudante usará para digitar o código da sala, seu nome e RGM.

**Tabela 15:** Campos da tela para envio do nome e rgm

Nome do Campo	Tipo	Máscara	Obrigatório	Observação
Código da Sala	Alfanumérico	N/A	Sim	N/A
Nome e Sobrenome	Text	N/A	Sim	N/A
Seu RGM	Text	SIM	SIM	Não permite digitar letras

Fonte: Autor (2025)

A tabela 16 mostra o comando da tela usada para validar o código fornecido pelo aluno.

**Tabela 16:** Comando para validar os dados digitados

Comando	Ação	Observações
Continuar	Validar o dado fornecido e avança para o questionário	N/A

Fonte: Autor (2025)

#### 4.4.8 Questionário de perfil

A seguir, na figura 20, será apresentado a tela usada para avaliar o perfil do estudante, quanto mais opções o acadêmico marcar, maior será sua pontuação. Os pontos obtidos pelo estudante são de suma importância para que a plataforma de gerenciamento de grupos PROFIDINA ÁGIL possa fazer a organização dos grupos de forma equilibrada.

**Figura 20:** Questionário de perfil

The screenshot shows a mobile-style survey form titled "Questionário de perfil". It consists of five numbered questions (1 to 5) with various answer options and a feedback section at the bottom.

- 1** Disponibilidade para reuniões além dos finais de semana?
 

Sim, tenho disponibilidade
- 2** Tem mais de 1 ano na faculdade?
 

Sim, estou há mais de 1 ano
- 3** Papel (pode marcar mais de um)
 

Organização da parte de cada um     Apresentar para a sala  
 Pesquisar     Preparar Apresentação
- 4** Perfil (pode marcar mais de um)
 

Extrovertido     Equilibrado  
 Mão na massa  
Fazer o software, protótipo, editar imagens
- 5** Possui alguma habilidade técnica que possa contribuir com o projeto?
 

Possuo habilidade técnica que contribua com o projeto  
Ex: prototipação, edição de vídeo, imagem ou áudio

Quanto mais respostas, maior a sua pontuação para formação de grupos equilibrados

**Voltar**    **Enviar informações**

Fonte: Autor (2025)

A seguir, a tabela 17 mostra os campos que o estudante usará para digitar o código da sala, seu nome e RGM.

**Tabela 17:** Campos para o estudante selecionar

Nome do Campo	Tipo	Máscara	Obrigatório	Observação
Sim, tenho disponibilidade	Checkbox	N/A	Não	N/A
Sim, estou há mais de 1 ano	Checkbox	N/A	Não	N/A
Organização da parte de cada um	Checkbox	N/A	Não	N/A

Apresentar para a sala	Checkbox	N/A	Não	N/A
Pesquisar	Checkbox	N/A	Não	N/A
Prepara apresentação	Checkbox	N/A	Não	N/A
Extrovertido	Checkbox	N/A	Não	N/A
Equilibrado	Checkbox	N/A	Não	N/A
Mão na massa	Checkbox	N/A	Não	N/A
Possuo habilidade técnica que contribua com o projeto	Checkbox	N/A	Não	N/A

Fonte: Autor (2025)

A seguir, a tabela 18 apresentará os comandos usados tanto para enviar as informações quanto para voltar e revisar os dados.

**Tabela 18:** Comandos da tela do questionário

Comando	Ação	Observações
Enviar informações	Envia as informações junto com a pontuação obtida no questionário	N/A
Voltar	Volta a página anterior	N/A

Fonte: Autor (2025)

#### 4.4.9 Tela de uma sala

A seguir, A figura 21 demonstra uma sala com alunos ingressados, grupos formados e duas opções para o professor escolher como os alunos irão entrar na sala. É nessa sala que é feita a organização dos estudantes e é onde o professor escolhe o líder de cada grupo.

**Figura 21:** Tela para formação e gerenciamento de grupo

Fonte: Autor (2025)

A tabela 19 apresenta os campos que o professor utiliza para fazer a organização dos estudantes.

**Tabela 19:** Campos da sala

Nome do Campo	Tipo	Máscara	Obrigatório	Observação
Limite de estudantes por grupos	Number	N/A	Sim	O maior limite é de 10 estudantes
Algoritmo de ordenação	Select	N/A	Sim	N/A

Fonte: Autor (2025)

A tabela 20 apresenta os comandos utilizado na sala para fazer o gerenciamento de equipes.

**Tabela 20:** Comandos da sala

Comando	Ação	Observações
Organizar Grupos	Forma os grupos	Ordena os estudantes e depois forma os grupos utilizando uma técnica de zig-zag
Voltar	Volta a página anterior	N/A

Escolher	Seleciona o estudante como líder	Só pode selecionar um estudante por grupos
Excluir	Exclui um estudante	N/A
Salvar Organização	Salva a organização dos estudantes	Precisa clicar em salvar para que a organização seja realmente persistida no banco
Reorganizar	Libera a organização dos estudantes	Uma vez que é salva a organização, precisa clicar em reorganizar para poder fazer uma nova organização

Fonte: Autor (2025)

## **REFERÊNCIAS:**

1. ARAÚJO, Ana Clara Souza; DAMASCENO JÚNIOR, José Ademir; ROMEU, Mairton Cavalcante. Introdução à Astronomia no ensino fundamental: análise da Team-Based Learning como estratégia facilitadora de ensino. **Revista Prática Docente**, v. 7, n. 3, e22061, 2022.  
Disponível em: [link](#)  
Acesso em: 26 de fev 2025
2. BOOCH, Grady. **UML: guia do usuário**. Elsevier Brasil, 2006.  
Disponível em: [link](#)  
Acesso em: 31 jun 2025.
3. CALDEIRA, Carlos. PostgreSQL: Guia Fundamental. 2015.  
Disponível em: [link](#)  
Acesso em 17 jun 2025
4. DANTAS, Ana Caroline Martins; NEVES, Ana Paula. Build Fiber: projeto e prototipagem de sistema ERP para gestão e inovação de empresas de instalação de fibra óptica. 2024.  
Disponível em: [link](#)  
Acesso em: 15 jun 2025
5. DOS SANTOS, Paulo Sérgio Medeiros; TRAVASSOS, Guilherme Horta. Colaboração entre academia e indústria: oportunidades para utilização da pesquisa-ação em engenharia de software. In: **Workshop de Engenharia de Software Experimental (ESELAW)**. 2008.  
Disponível em: [link](#)  
Acesso em: 22 maio. 2025
6. Draft. **Site do Draft**. Como a DreamShaper se tornou uma startup que “ensina empreendedorismo” nas escolas  
Disponível em: [aqui](#)  
Acesso em: 21 abril 2025.

7. DE OLIVEIRA, Maria Helena Ferreira; DE SOUSA, Reudismam Rolim; GONÇALVES, Samara Martins Nascimento. UM ESTUDO SOBRE O USO DA UML EM EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE POR PROFISSIONAIS DA TECNOLOGIA DA INFORMAÇÃO. *RECIMA21-Revista Científica Multidisciplinar*-ISSN 2675-6218, 2025, 6.2: e626176-e626176.  
Disponível em: [link](#)  
Acesso em: 04 jun 2025
8. DE CASTRO GUEDES, Pedro Phelipe et al. FRAMEWORKS JAVASCRIPT: ANÁLISE COMPARATIVA DOS PRINCIPAIS CONSTRUTORES DE INTERFACES WEB MODERNAS.  
Disponível em: [link](#)  
Acesso em 15 jun 2025
9. DE SORDI, José Osvaldo; DE LOURDES MARINHO, Bernadete. Análise dos ambientes para integração entre sistemas de informação segundo especialistas. **Revista de Ciências da Administração**, p. 154-177, 2006.  
Disponível em: [Link](#)  
Acesso em: 15 jun 2025.
10. DA SILVA, Rogério Oliveira; MARTINS, Bonny Rodrigues; DINIZ, Walisson Gama. A complexibilidade da UML e seus diagramas. **Tecnologias em Projeção**, v. 8, n. 1, p. 86-99, 2017.  
Disponível em: [link](#)  
Acesso em: 31 jun 2025
11. DA SILVA PEDROSO, Alexandre; CINTRA, Fausto Gonçalves. ESTUDO ANALÍTICO DO DESEMPENHO DE ALGORITMOS DE ORDENAÇÃO.  
Disponível em: [link](#)  
Acesso em: 22 jun 2025

- 12.FRANCK, Kewry Mariobo; PEREIRA, Robson Fernandes; DANTAS FILHO, Jerônimo Vieira. Diagrama Entidade-Relacionamento: uma ferramenta para modelagem de dados conceituais em Engenharia de Software. *Research, Society and Development*, 2021, 10.8: e49510817776-e49510817776.  
Disponível em: [link](#)  
Acesso em: 04 jun. 2025.
- 13.FARIAS, Luciana Vieira; MOHALLEM, Andréa Gomes da Costa. Team-Based Learning online: percepção dos graduandos de saúde e influência do perfil comportamental do estudante. *Revista Brasileira de Educação Médica*, v. 48, n. 1, p. e015, 2024.  
Disponível em: [link](#)  
Acesso em: 26 de fev 2025
- 14.GIACOMELLI, Sandra Cristina Pelegrini et al. O uso da metodologia TEAM-BASED LEARNING (TBL) aliada à tecnologia: percepções sobre a aprendizagem de contabilidade básica no Curso Técnico em Administração. 2020.  
Disponível em: [link](#)  
Acesso em: 31 de mar 2025
- 15.HIRAMA, Kichi. Engenharia de Software. **Qualidade e Produtividade com Tecnologia**, v. 1, 2012.  
Disponível em: [link](#)  
Acesso em: 22 maio. 2025
- 16.KALINOWSKI, Marcos et al. **Engenharia de Software para Ciência de Dados: Um guia de boas práticas com ênfase na construção de sistemas de Machine Learning em Python**. Casa do Código, 2023.  
Disponível em: [link](#)  
Acesso em: 31 de mar 2025

17. KALINOWSKI, Marcos et al. MPS. BR: Promovendo a Adoção de Boas Práticas de Engenharia de Software pela Indústria Brasileira. In: **ClbSE**. 2010. p. 265-278.  
Disponível em: [link](#)  
Acesso em: 22 maio 2025.
18. MONDLANE, Paulo Titos. Proposta de um modelo de ciclo de vida seguro de desenvolvimento de software. caso de estudo: CIUEM. 2023.  
Disponível em: [link](#)  
Acesso em: 26 maio 2025.
19. MALHONE, Mariana Magalhães; FRIGERI, Mônica. A importância do gerenciamento de configuração para o ciclo de vida do software: um estudo de caso baseado nas diretrizes da engenharia de software. **Revista Brasileira em Tecnologia da Informação**, v. 3, n. 1, p. 14-23, 2021.  
Disponível em: [link](#)  
Acesso em: 01 de abril 2025
20. MARQUES, Ana Paula Ambrósio Zanelato, et al. TBL ACTIVE: SOFTWARE PARA APOIO À METODOLOGIA ATIVA TEAM BASED LEARNING. *Anais CIET: Horizonte*, 2022.  
Disponível em: [link](#)  
Acesso em: 3 maio. 2025.
21. ORACLE, O que é gerenciamento de capital humano (HCM)? ,ORACLE, 2025,  
Disponível em: [link](#)  
Acesso em: 15 jun 2025.
22. PELEGRI NI GIACOMELLI, Sandra Cristina; CHRISTINO GITAHY, Raquel Rosan; DE LIMA TERÇARIOL, Adriana Aparecida. A metodología Team-Based Learning (TBL) articulada à plataforma TBL active na aprendizagem de contabilidade no curso técnico em administração. *Actualidades Investigativas en Educación*, v. 21, n. 3, p. 603-630, 2021.  
Disponível em: [link](#)  
Acesso em: 26 fev 2025

23. PEREIRA, Higor Koakovski et al. RELATÓRIO DE ESTÁGIO-BACK-END COM JAVASCRIPT E TYPESCRIPT. 2024.

Disponível em: [link](#)

Acesso em: 15 jun 2025

24. POLACHINI, Maria Noeli; ROCHA, Tiago Rios. Estudo Sobre Modelos de Ciclo de Vida de Software. In: **7<sup>a</sup> MOEPEX**. 2018.

Disponível em: [link](#)

Acesso em: 23 maio 2025

25. PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software-9**. McGraw Hill Brasil, 2021.

Disponível em: [link](#)

Acesso em 22 maio 2025

26. Santos, RR (sd). Ciclo de Vida de Desenvolvimento de Sistemas. Universidade do Estado de Mato Grosso.

Disponível em: [link](#).

Acesso em: 23 maio 2025

27. SILVA, Fernando Guerriero Cardoso da. Levantamento de requisitos aplicado à pesquisa e desenvolvimento de produtos de software. 2023.

Disponível em: [link](#)

Acesso em: 31 jun 2025.

28. SOUZA FILHO, Iderli Pereira de. Uma Proposta de Incorporação do Diagrama de Classes da UML à Linguagem MASRML Adaptado ao Contexto de Sistema Multiagentes. 2023.

Disponível em: [link](#)

Acesso em: 04 jun 2025

29. SILVA, Francisco Alexson Almeida da. Clean CSS: uma plataforma para produção de códigos CSS limpos, performáticos e escaláveis. 2025.

Disponível em: [link](#)

Acesso em 17 jun 2025

