



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**BRUNO DE CARVALHO SOBRINHO**

**DESENVOLVIMENTO DE UM SISTEMA WEB PARA SOLICITAÇÃO E  
ACOMPANHAMENTO DO PROCESSO DE APROVEITAMENTO DE  
ESTÁGIO-EMPRESA DA UFC QUIXADÁ**

**QUIXADÁ**

**2022**

BRUNO DE CARVALHO SOBRINHO

DESENVOLVIMENTO DE UM SISTEMA WEB PARA SOLICITAÇÃO E  
ACOMPANHAMENTO DO PROCESSO DE APROVEITAMENTO DE ESTÁGIO-EMPRESA  
DA UFC QUIXADÁ

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Engenharia de  
Software da Universidade Federal do Ceará,  
como requisito parcial à obtenção do grau de  
bacharel em Engenharia de Software.

Orientador: Prof. Dr. Emanuel Ferreira  
Coutinho

QUIXADÁ

2022

## RESUMO

Estágio Supervisionado é uma forma de preparação dos egressos para os ambientes onde estes irão desempenhar seus papéis após formados. Na Universidade Federal do Ceará (UFC) - Campus Quixadá, para alguns dos cursos de graduação, tal atividade é um componente obrigatório para a colação de grau. O estudante pode optar por fazer esse estágio no próprio campus - Desenvolvendo atividades no Núcleo de Práticas em Informática (NPI) - ou em empresas conveniadas da instituição. Quando o aluno opta por ter essa experiência em uma instituição parceira, ele é responsável por além de solicitar a matrícula no componente curricular utilizando o sistema oficial do campus Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), também enviar uma documentação ao professor orientador responsável pelos estágios no curso na qual o estudante está matriculado. Para o aluno ter sua matrícula no componente curricular efetivada, é necessário que o professor orientador faça a análise e confirmação de que está tudo certo entre a faculdade e a instituição onde o aluno deseja realizar o estágio, as atividades a serem desenvolvidas pelo aluno vai de encontro ao plano de ensino do curso, a documentação enviada pelo aluno está em conformidade com as regras. Somente após essa confirmação, o professor orientador envia uma confirmação para a coordenação de ensino informando que a matrícula do aluno pode de fato ser efetivada. Atualmente a UFC não dispõe de uma ferramenta na qual possa ser centralizado o envio dessa documentação, verificação da situação desse processo de aproveitamento, notificação de novidades nesse processo (Quando por exemplo o professor orientador confere a documentação e constata que algo está faltando e que o aluno precisa enviar mais alguma coisa), notificação aos atores envolvidos (Professor orientador, aluno, servidor técnico-administrativo). Atualmente, na UFC - Campus Quixadá, todo esse processo é realizado através de uma adaptação do *Google Classroom*, embora por muito tempo tenha sido através da troca de *e-mail* entre os atores citados anteriormente, o que se tornava um problema devido a existirem vários professores orientadores de estágio no campus, e cada um fazendo a sua maneira a execução desses passos. O presente trabalho tem como objetivo propor uma solução e desenvolvimento de um sistema WEB que permita centralizar todo esse processo, tornando-o mais simples, intuitivo e padronizado.

**Palavras-chave:** Estágio supervisionado. Sistema WEB. Engenharia de software. Aplicação. Acompanhamento.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de casos de uso do SIGESTAGIOS . . . . .	13
Figura 2 – Tela de listagem de alunos matriculados no SIGESTAGIOS . . . . .	13
Figura 3 – Diagrama de casos de uso do sistema controle de estágios . . . . .	15
Figura 4 – Tela de cadastro de defesa do sistema controle de estágios . . . . .	15
Figura 5 – Diagrama de casos de uso do sistema Akademic . . . . .	17
Figura 6 – Dashboard do sistema Akademic . . . . .	17
Figura 7 – Tela da página administrativa do SGE . . . . .	19
Figura 8 – Diagrama de fluxo de processo para Estágio Supervisionado da UFC . . . . .	24
Figura 9 – Modelo em cascata . . . . .	27
Figura 10 – Fluxo Geral de funcionamento da <i>World Wide Web (WEB)</i> . . . . .	30
Figura 11 – Arquitetura <i>MongoDB, Express, Vue.js, and Node.js (MEVN)</i> . . . . .	32
Figura 12 – Fluxo de funcionamento para envio de <i>WEB Push Notifications</i> . . . . .	35
Figura 13 – Fluxo de configuração OneSignal de <i>WEB push</i> em um <i>site</i> . . . . .	36
Figura 14 – Passos metodológicos . . . . .	37

## **LISTA DE QUADROS**

Quadro 1 – Quadro comparativo de trabalhos . . . . .	21
Quadro 2 – Cronograma de execução das atividades . . . . .	39

## LISTA DE ABREVIATURAS E SIGLAS

UFC	Universidade Federal do Ceará
NPI	Núcleo de Práticas em Informática
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
WEB	<i>World Wide Web</i>
MEVN	<i>MongoDB, Express, Vue.js, and Node.js</i>
MEC	Ministério da Educação
P&D	Pesquisa e Desenvolvimento
PPC	Projeto Pedagógico de Curso
TCE	Termo de Compromisso de Estágio
CLT	Consolidação das leis do trabalho
PJ	Pessoa Jurídica
AC	Atividades Complementares
IES	Instituição de Ensino Superior
UML	<i>Unified Modeling Language</i>
MER	Modelo Entidade-Relacionamento
DER	Diagrama Entidade-Relacionamento
JSON	<i>JavaScript Object Notation</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
HTTP	<i>Hyper Text Transfer Protocol</i>
MEAN	<i>MongoDB, Express, Angular, and Node.js</i>
MERN	<i>MongoDB, Express, React, and Node.js</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>8</b>
<b>1.1</b>	<b>Objetivos . . . . .</b>	<b>11</b>
<b>1.1.1</b>	<b><i>Objetivos Gerais . . . . .</i></b>	<b>11</b>
<b>1.1.2</b>	<b><i>Objetivos Específicos . . . . .</i></b>	<b>11</b>
<b>2</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>12</b>
<b>2.1</b>	<b><i>Plataforma WEB para Gestão dos Estágios Supervisionados nos Cursos de Graduação (BALUZ et al., 2020) . . . . .</i></b>	<b>12</b>
<b>2.2</b>	<b><i>Sistema para Controle de Estágio Supervisionado do Departamento Acadêmico de Informática (LEONCIO, 2018) . . . . .</i></b>	<b>14</b>
<b>2.3</b>	<b><i>Implementação de um Sistema Multi-plataforma para Gerenciamento de Atividades Complementares em Cursos Superiores utilizando Banco de Dados noSQL (LIMA et al., 2018) . . . . .</i></b>	<b>16</b>
<b>2.4</b>	<b><i>SGE - Sistema de Gerenciamento de Estágios (BRITO; OLIVEIRA, 2016)</i></b>	<b>18</b>
<b>2.5</b>	<b>Análise Comparativa . . . . .</b>	<b>20</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>22</b>
<b>3.1</b>	<b>Estagio supervisionado . . . . .</b>	<b>22</b>
<b>3.1.1</b>	<b><i>Estagio supervisionado na UFC - Campus Quixadá . . . . .</i></b>	<b>23</b>
<b>3.2</b>	<b>Processo de Desenvolvimento de Software . . . . .</b>	<b>25</b>
<b>3.2.1</b>	<b><i>Modelo de desenvolvimento de software em cascata . . . . .</i></b>	<b>26</b>
<b>3.3</b>	<b>Sistema WEB . . . . .</b>	<b>28</b>
<b>3.3.1</b>	<b><i>Arquitetura MEVN . . . . .</i></b>	<b>30</b>
<b>3.4</b>	<b><i>WEB Push Notifications . . . . .</i></b>	<b>33</b>
<b>3.4.1</b>	<b><i>One Signal . . . . .</i></b>	<b>35</b>
<b>4</b>	<b>PROCEDIMENTOS METODOLÓGICOS . . . . .</b>	<b>37</b>
<b>4.1</b>	<b>Pesquisar soluções utilizadas em outras universidades para o gerenciamento de estágios . . . . .</b>	<b>37</b>
<b>4.2</b>	<b>Checar possíveis funcionalidades desse sistema que possam ser trazidos para a solução a ser desenvolvida . . . . .</b>	<b>38</b>
<b>4.3</b>	<b>Definição dos requisitos a serem implementados . . . . .</b>	<b>38</b>
<b>4.4</b>	<b>Modelar o sistema a fim de atender os requisitos definidos . . . . .</b>	<b>38</b>

<b>4.5</b>	<b>Implementação dos requisitos definidos . . . . .</b>	<b>39</b>
<b>4.6</b>	<b>Validação do sistema . . . . .</b>	<b>39</b>
<b>4.7</b>	<b>Cronograma . . . . .</b>	<b>39</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>40</b>



## 1 INTRODUÇÃO

De acordo com o Parecer CNE/CES nº 136/2012, aprovado em 8 de março de 2012, que institui as diretrizes Curriculares Nacionais para os cursos de graduação em computação, como forma de preparação dos egressos para os ambientes onde esses irão desempenhar seus papéis após formados, é fortemente recomendado a realização de estágio nos cursos dessa área (MEC, 2012).

O estágio supervisionado deverá ter um acompanhamento efetivo pelo professor orientador da instituição de ensino e por supervisor da parte concedente (BRASIL, 2008). Segundo o Ministério da Educação (MEC), essa experiência tem o objetivo de “consolidar e articular as competências desenvolvidas ao longo do curso por meio das demais atividades formativas, de caráter teórico ou prático, e permitir o contato do formando com situações, contextos e organizações próprios da atuação profissional” (MEC, 2012, p. 26).

Nesse contexto, a UFC - Campus Quixadá, em conformidade com a lei anteriormente citada, apresenta nos projetos pedagógicos de alguns dos cursos do campus, o estágio supervisionado como requisito obrigatório para a colação de grau e obtenção da graduação do estudante. É facultativo ao aluno a modalidade em que o estágio será cumprido, sendo possível a realização de **Estágio Interno**, podendo ser realizado no NPI, iniciativa empreendedora ou projeto de Pesquisa e Desenvolvimento (P&D) ou o **Estágio Externo**, quando realizado em empresas conveniadas a universidade ou através de participação em projeto de P&D desenvolvido em outras instituições. Independente da modalidade escolhida, é obrigatório que a atividade a ser desempenhada pelo aluno em seu estágio supervisionado obrigatório, seja relacionada com seu curso e realizadas em áreas de atuação afins com o perfil de egresso previsto no Projeto Pedagógico de Curso (PPC).

O NPI é um programa de extensão do campus Quixadá que visa o desenvolvimento de sistemas para a sociedade, e a vivência em um ambiente semelhante a uma empresa, no modelo de fábrica de *software*, com processo definido e projetos com clientes reais (COUTINHO *et al.*, 2021). Para a realização de estágio nesse ambiente, ocorre semestralmente um processo seletivo na qual os alunos interessados e que estejam aptos a realização do estágio - tendo cursado as disciplinas que são pré-requisitos para essa atividade, variando de acordo com o curso no qual o aluno está matriculado - podem se inscrever e caso selecionado, estagiar ali durante todo o semestre letivo.

Para estágio externo em empresas conveniadas (Estágio-Empresa), é necessário que a empresa na qual o aluno deseja realizar o estágio esteja devidamente conveniada e regularizada junto a agência de estágio da universidade. Para tal, se faz necessário a celebração de um Termo de Convênio entre a UFC e a empresa, bem como a assinatura do Termo de Compromisso de Estágio (TCE) e do Plano de Trabalho (UFC, 2009). A empresa também deverá designar uma pessoa para orientar e supervisionar o estagiário.

Assim como para todas as outras disciplinas da estrutura curricular, no período de matrícula para o semestre letivo, o aluno que deseja realizar o estágio curricular obrigatório, deve realizar sua matrícula no componente curricular através do SIGAA, que é o sistema oficial utilizado pela UFC para várias atividades acadêmicas, entre elas para matrícula dos alunos em disciplinas a serem cursados durante o semestre. Para a maioria dos componentes curriculares, a efetivação da solicitação de uma matrícula é realizada tão logo o período de matrículas termine e o aluno tenha conseguido uma vaga na turma solicitada.

O aluno que opta por realizar um estágio no NPI, por se tratar de um ambiente controlado pela própria universidade, tem sua matrícula no componente curricular de Estágio Supervisionado efetivada dentro do SIGAA logo que é aprovado no processo seletivo para o estágio nesse ambiente.

Para os alunos que optam por realizar um Estágio-Empresa, esse processo de efetivação da matrícula no componente de Estágio Supervisionado é um pouco diferente e mais burocrático. O aluno precisa enviar ao professor orientador de estágios do seu curso o TCE ou Contrato de trabalho (caso o aluno esteja aproveitando um contrato de trabalho celebrado via Consolidação das leis do trabalho (CLT) ou Pessoa Jurídica (PJ)). O professor orientador então precisa conferir a documentação, garantir que as atividades previstas estão alinhadas com o curso do aluno e se a empresa tem convênio ativo com a UFC. Dependendo da situação, pode haver várias idas e vindas de mensagens entre os orientadores e alunos até que tudo esteja de acordo com o exigido pela instituição e que o aluno possa de fato aproveitar o estágio como o estágio-obrigatório de seu curso. A matrícula do aluno só é de fato efetivada no SIGAA após uma sinalização positiva do professor orientador para a secretaria acadêmica.

Essa etapa até a efetivação da matrícula componente curricular de estágio supervisionado no SIGAA dos alunos solicitantes de Estágio-Empresa se torna ainda mais desafiadora devido a inexistência de uma ferramenta própria para a gestão e acompanhamento desse processo.

Por muito tempo todo esse fluxo foi realizado de forma bem manual e braçal, através de trocas de e-mails entre as pessoas interessadas no processo: Aluno, Professor Orientador e secretaria de estágio. Mesmo atendendo ao objetivo final, o uso de e-mail como meio para esse fluxo era problemático devido ao propósito geral de uma caixa de e-mail e por exemplo solicitações serem esquecidas no meio de outros e-mails na caixa de entrada, algumas vezes ser classificadas como *SPAM*, orientador deixar de enviar um e-mail a uma parte interessada - por exemplo ter uma documentação conferida e dar a confirmação apenas ao aluno e esquecer a secretaria acadêmica, e vice versa - a pouca transparência no processo - não sendo possível saber exatamente o estado da solicitação. Um outro problema com a ausência de uma ferramenta própria para a gestão desse fluxo é que com a existência de diferentes professores orientadores de estágio na instituição esse fluxo não era uniforme e padronizado entre todos eles. Embora a partir do semestre 2021.2 tenha sido adaptado a ferramenta *Google Classroom* para esse fim, por se tratar de uma ferramenta genérica, a existência de uma ferramenta específica e de gerenciamento próprio pela universidade para esse fim continuaria sendo uma contribuição válida.

Dentro desse fluxo existem vários problemas que podem ser minimizados com a utilização de um sistema cujo foco principal seja a automatização e uniformização desse processo, como por exemplo: aumento da agilidade no processo, garantia de as informações serem enviadas e recebidas pelas pessoas necessárias, notificações aos interessados sempre que algum novo evento ocorrer nesse fluxo, padronização do processo feito por diferentes professores orientados de estágio, padronização de como as mensagens são recebidas e enviadas dentro do processo.

O presente trabalho visa o desenvolvimento de um sistema *WEB* onde o aluno seja capaz de realizar a solicitação de aproveitamento de estágio-empresa como componente curricular, preenchendo e anexando todas as informações obrigatórias a serem enviadas para validação, possibilitando ao professor orientador o acesso de forma simplificada a todas as solicitações já submetidas a ele, sendo possível a realização do processo de validação da documentação e/ou solicitações necessárias, e por fim notificando a secretaria acadêmica quando estiver tudo certo com a documentação concedendo então a permissão para efetivação da matrícula do aluno para Estágio-Supervisionado dentro do SIGAA. Além disso, que possibilite que todas as partes envolvidas possam ter uma clara visão da situação atual da solicitação, recebendo sempre as notificações de atualizações pertinentes. E por fim, tornar o processo padronizado, ágil e transparente para solicitação e respostas entre alunos, orientadores e secretaria acadêmica tendo como objetivo final o aproveitamento do estágio como componente curricular para o aluno.

O presente trabalho está dividido nos seguintes capítulos, além deste capítulo introdutório: Capítulo 2 apresenta os trabalhos relacionados a esse projeto, Capítulo 3 apresenta uma fundamentação teórica com conceitos e temas importantes ao projeto, e por fim, o Capítulo 4 apresenta a metodologia deste trabalho.

## **1.1 Objetivos**

Nesta seção, serão apresentados os objetivos gerais e específicos do trabalho.

### ***1.1.1 Objetivos Gerais***

Desenvolver um sistema específico para o fluxo de solicitação, envio de documentação e acompanhamento do processo de aproveitamento de estágio-empresa da UFC Quixadá, possibilitando a padronização desse fluxo e notificação das partes interessadas sobre novas atualizações no processo.

### ***1.1.2 Objetivos Específicos***

- Desenvolver uma solução arquitetural que permita às partes interessadas (Secretaria de estágio, professores orientadores de estágio e estudantes de graduação da UFC Quixadá que estão no processo de aproveitamento de estágio-empresa) a centralização e padronização do processo de aproveitamento de estágio curricular em empresas
- Desenvolvimento de um sistema *WEB* para solicitação e acompanhamento do processo de aproveitamento de estágio-empresa da UFC Quixadá
- Definir um fluxo da operação com a solução arquitetural proposta para auxiliar o acompanhamento do processo de aproveitamento de estágio-empresa

## 2 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos relacionados com o presente trabalho. Foram selecionados dois trabalhos que se assemelham bastante ao objetivo do presente trabalho, onde também se é proposto ferramentas que automatizam o processo de gerenciamento de estágios supervisionados em Instituições de Ensino Superior. Por fim se é apresentado uma seção de comparação e diferenças entre tais trabalhos e a solução proposta no presente trabalho.

### 2.1 *Plataforma WEB para Gestão dos Estágios Supervisionados nos Cursos de Graduação (BALUZ et al., 2020)*

No trabalho de Baluz *et al.* (2020) apresenta-se o contexto do curso de Ciência da Computação da Universidade Estadual do Piauí (UESPI) - Campus Parnaíba, uma instituição de ensino que estava em busca de melhorar as atividades educacionais com o auxílio da TI e que não possuía um sistema informatizado para controle de estágios supervisionados.

Nesse trabalho objetiva-se sistematizar por meio de um *software* em plataforma *WEB* o processo de estágios supervisionados obrigatórios para o curso de Ciência da computação daquela instituição. Nessa sistematização, inclui-se: organizar e armazenar de forma centralizada os documentos referentes ao estágio, evitar o manuseio de papéis pelos atores envolvidos no processo e disponibilização de uma biblioteca dinâmica de documentos e relatórios.

Para o desenvolvimento dessa ferramenta, nomeada de SIGESTAGIOS, o trabalho foi dividido em algumas etapas: Definição da tecnologia a ser utilizada, Definição dos requisitos a serem implementados, Modelagem do sistema a ser desenvolvido, Desenvolvimento do sistema e Testes do sistema.

A figura 1 ilustra todas as funcionalidades elicitadas e que o sistema deve apresentar, enquanto que a figura 2 mostra uma interface desse sistema após implementado.

Ao final do desenvolvimento do sistema, foram convidados 13 voluntários para a validação do sistema. Dentre esses voluntários, haviam diferentes tipos de partes interessadas do sistema, como: 1 Coordenador de curso, 7 alunos de graduação, 2 professores orientadores e 3 supervisores ligados a empresas concedentes de estágio. Após esses voluntários testarem e validarem o sistema desenvolvido, foi aplicado um questionário *SUS - System Usability Scale* para a avaliação de usabilidade do sistema, na qual foi possível medir de maneira quantitativa a satisfação dos usuários nas tarefas desempenhadas pelo sistema.

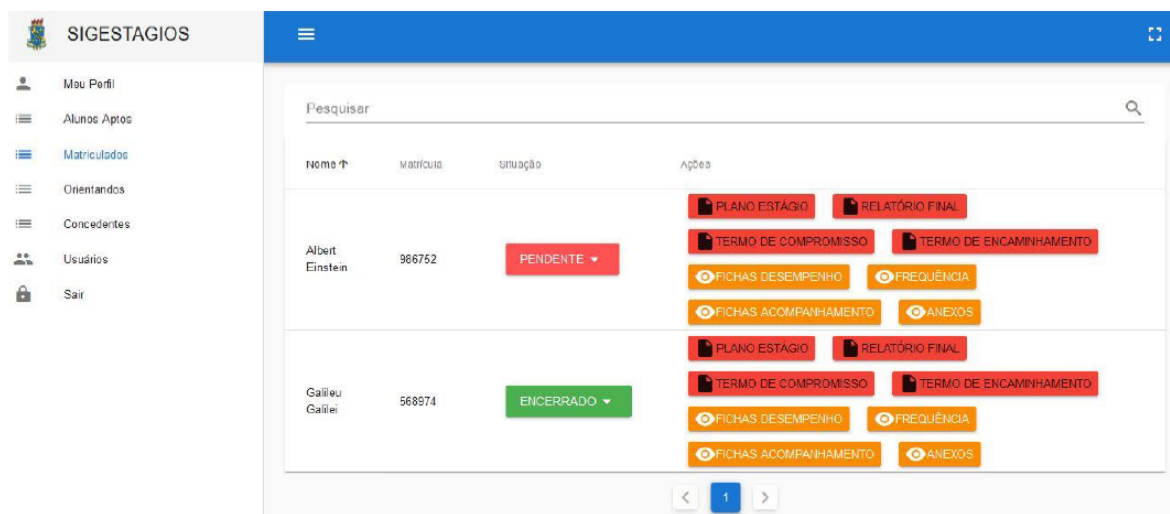
Dentre os resultados coletados, foi possível identificar que 92,31% dos voluntários usariam o sistema caso ele fosse implementado na instituição, e a média de pontuação final sendo de 81,92, o que é acima dos 68 pontos definidos pelo questionário SUS como uma boa avaliação.

Figura 1 – Diagrama de casos de uso do SIGESTAGIOS



Fonte: Baluz *et al.* (2020)

Figura 2 – Tela de listagem de alunos matriculados no SIGESTAGIOS



Fonte: Baluz *et al.* (2020)

## **2.2 Sistema para Controle de Estágio Supervisionado do Departamento Acadêmico de Informática (LEONCIO, 2018)**

No trabalho de Leoncio (2018) apresenta-se o contexto do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná (UTFPR), uma instituição que promove e incentiva a utilização de novas tecnologias, em especial de sistemas voltados ao gerenciamento de atividades educacionais.

Através de pesquisas realizadas, foi constatada a ausência de um sistema para controle de Estágio no Departamento Acadêmico de Informática da UTFPR, Campus Ponta Grossa e que todo armazenamento de informação relacionado a esse controle era realizado utilizando-se planilhas eletrônicas, tornando a manipulação dos dados um tanto quanto inseguras e muitas vezes inviáveis no contexto de eficiência. Ademais, essa forma de armazenamento acabava deixando esses dados suscetíveis a problema de integridade devido a possibilidade de um arquivo ser corrompido ou de ocorrência de falhas durante a realização de operações de escrita nesse arquivo.

Para melhorar esse processo, o trabalho propõe a melhoria no armazenamento das informações dos estágios supervisionados no departamento citado anteriormente através de uma aplicação *WEB* utilizando técnicas de desenvolvimento de aplicações computacionais com o uso de *frameworks* e foco no usuário.

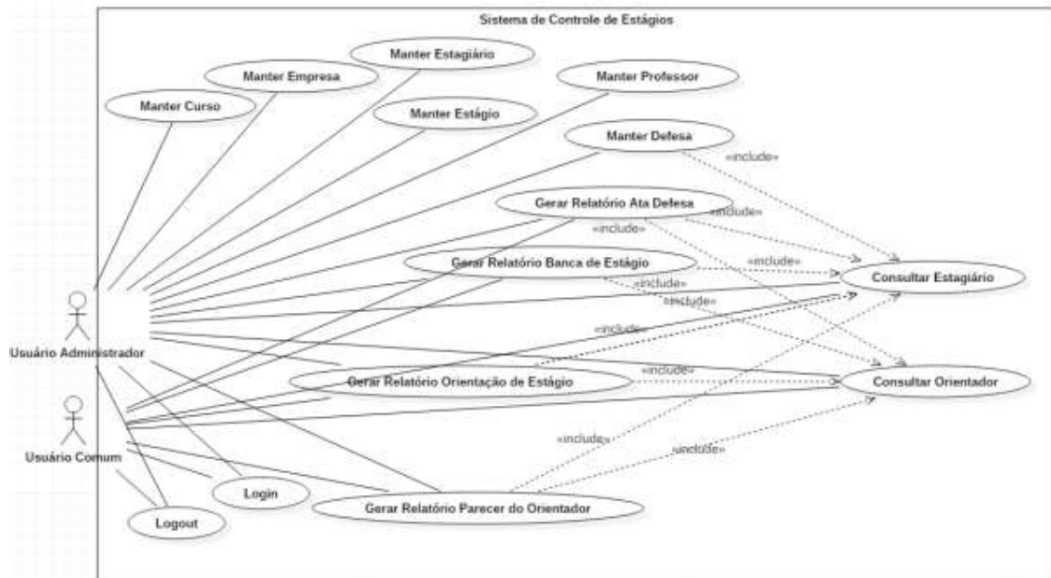
Para o desenvolvimento dessa aplicação, o trabalho foi dividido nas etapas de: Revisão da literatura, Definição das ferramentas a serem utilizadas para o desenvolvimento da aplicação, Definição dos requisitos do sistema, Modelagem do sistema e por fim, Testes do sistema pelos usuários finais.

A figura 3 ilustra todas as funcionalidades elicitadas e que o sistema deve apresentar, enquanto que a figura 4 mostra uma interface desse sistema após implementado.

Os testes realizados pelos usuários finais mostraram que os requisitos definidos para o sistema de manter os dados de estágio e gerar documentos referentes ao processo de defesa de estágio da UTFPR - tais como: Ata de defesa, Banca de estágio, Orientação de Estágio e Parecer do Orientador - foram atendidos e que através do sistema será possível controlar as operações de controle de estágios da instituição. Os testes realizados também mostraram ganho no armazenamento das informações de estágio e que por meio da tecnologia utilizada foi possível melhorar o desempenho e segurança dos dados.

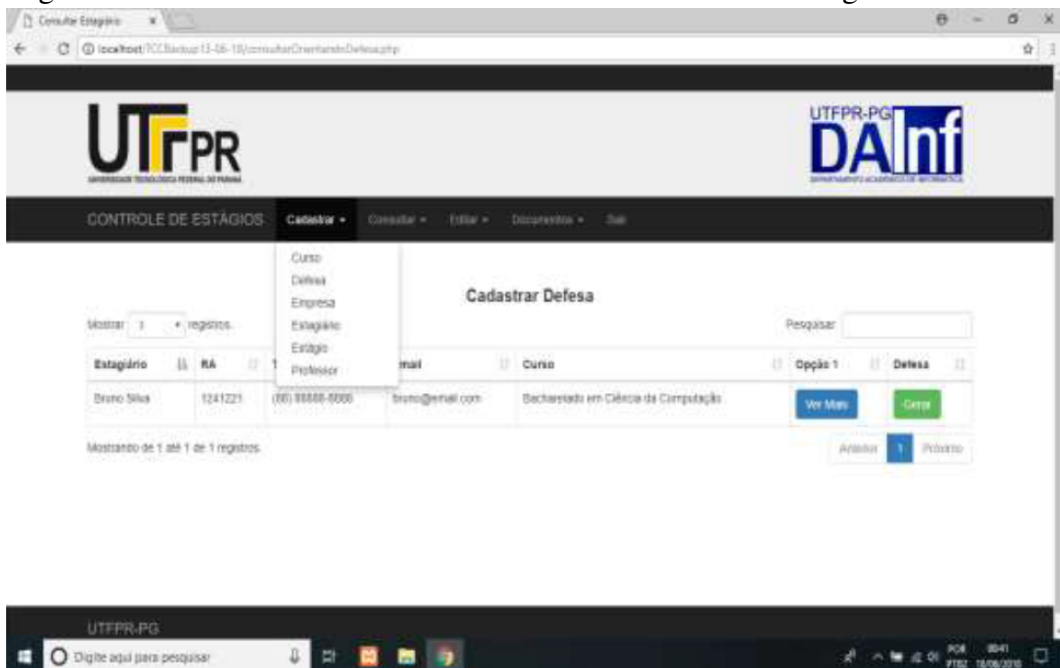
O desenvolvimento dessa aplicação possibilitou a melhoria em uma atividade que necessitava do emprego da tecnologia para que assim houvesse melhor desempenho na realização das atividades administrativas referentes ao estágio curricular.

Figura 3 – Diagrama de casos de uso do sistema controle de estágios



Fonte: Leoncio (2018)

Figura 4 – Tela de cadastro de defesa do sistema controle de estágios



Fonte: Leoncio (2018)



### **2.3 Implementação de um Sistema Multi-plataforma para Gerenciamento de Atividades Complementares em Cursos Superiores utilizando Banco de Dados noSQL (LIMA *et al.*, 2018)**

No trabalho de Lima *et al.* (2018), os autores tem como foco resolver a problemática em torno de um tema que assim como estágio supervisionado é comum nas instituições de ensino superior: Atividades Complementares (AC)

Os autores identificaram um transtorno no gerenciamento dessas atividades quando o aluno precisa fazer a solicitação de aproveitamento dessas AC, uma vez que em muitas instituições não existe um sistema computacional que gerencie a solicitação de aproveitamento de AC e se faz necessário por exemplo a entrega física dos comprovantes por parte dos estudantes, o preenchimento de formulários manuscritos realização de entregas digitais por correio eletrônico ou em plataformas de ensino a distância, tornando assim esse fluxo confuso e moroso tanto para o aluno quando para a Instituição de Ensino Superior (IES), uma vez que esse processo acaba gerando uma intensa carga administrativa devido a quantidade de alunos e comprovantes a serem analisados.

Lima *et al.* (2018) então propuseram um sistema multi-plataforma, intitulado de Akademic, adaptável a diversas IES e cursos superiores, para o gerenciamento dessas atividades complementares, tendo como foco principal, mas não exclusivo, a solicitação para aproveitamento e controle posterior dessas atividades por parte dos alunos e coordenadores de curso. Essa solução buscava também, dentre outras coisas, engajar seus usuários por meio do envio de Notificações *Push*.

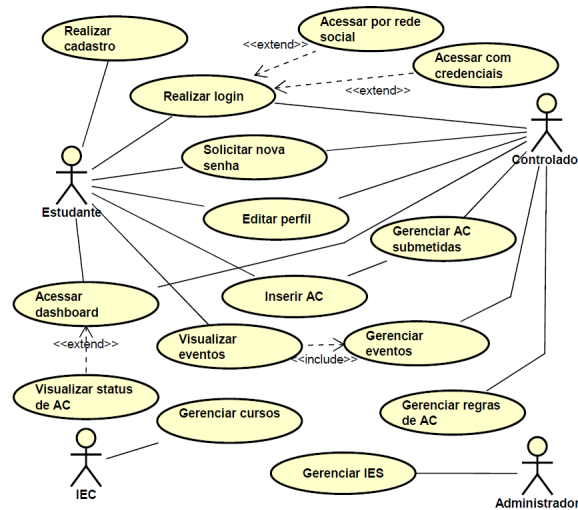
Os passos metodológicos para o projeto foram baseados no manifesto ágil, sendo eles: Levantamento de requisitos, definição de uma arquitetura para o sistema, definição dos modelos de dados, implementação do sistema.

Como resultado, foi desenvolvido um sistema possível de ser acessado em diferentes plataformas sem necessidade de alteração de código-fonte, através do uso da tecnologia Angular, utilizando banco de dados noSQL para permitir um maior escalonamento caso o trabalho vire um produto e seja utilizado por várias IES e que possui as seguintes funcionalidades: cadastro de eventos, realizar envio de AC para validação, visualização das atividades já enviadas e ou pendentes ou já validadas, bem como relatórios de aproveitamento de AC por grupo de atividades. Para os coordenadores existem as funcionalidades de gerenciar e validar as atividades encaminhadas pelos estudantes, bem como o cadastro de tipos de atividades e regras de cada

tipo. O coordenador também dispõe de gráficos que permitem ter de forma simplificada gráficos de acompanhamentos. A figura 5 ilustra todas as funcionalidades elicítadas e que o sistema deve apresentar, enquanto que a figura 6 mostra uma interface desse sistema após implementado.

Após o sistema implementado, Lima *et al.* (2018) destacam que foi possível verificar que o Akademic apresentava funcionalidades primordiais que permitia aos estudantes gerenciarem suas AC de maneira mais simples e controlada. Aos coordenadores, a facilidade e agilidade na verificação e validação das atividades se mostrou de grande ajuda, pois eliminou a necessidade de recebimento de solicitações que antes eram realizadas manualmente.

Figura 5 – Diagrama de casos de uso do sistema Akademic



Fonte: Lima *et al.* (2018)

Figura 6 – Dashboard do sistema Akademic



Fonte: Lima *et al.* (2018)

## **2.4 SGE - Sistema de Gerenciamento de Estágios (BRITO; OLIVEIRA, 2016)**

No trabalho de Brito e Oliveira (2016) apresenta-se o contexto do Instituto Federal do Norte de Minas Gerais (IFNMG) - Campus Januária. O estágio supervisionado é uma atividade curricular nesse campus.

Brito e Oliveira (2016) identificaram que apesar de toda a tecnologia disponível no mercado, a coordenação de estágios do campus ainda realizava seu trabalho de uma forma praticamente manual, através do uso de planilhas eletrônicas que atendiam apenas parcialmente as necessidades da coordenação.

Sabendo de todos os benefícios que um sistema de informação gerencial poderia trazer para a coordenação de estágios, como por exemplo: redução de tempo nas operações, melhoria no acesso as informações, propiciando relatórios mais precisos e rápidos, entre outras coisas, os autores desse trabalho propuseram um sistema *WEB* com o intuito de informatizar todo o processo de gestão de estágios realizado por essa coordenação, substituindo assim o uso das planilhas eletrônicas. O sistema proposto, que foi denominado como SGE - Sistema de Gerenciamento de Estágios, possui a integração das funcionalidades de gerenciamento de estágios e vagas de estágios em uma única ferramenta, além de possuir a capacidade de gerenciar inscrições de candidatos á vagas de estágios e empregos disponíveis nas empresas concedentes da região.

Os autores dividiram o trabalho em: análise de sistemas similares ao sistema a ser desenvolvido, modelagem da arquitetura e tecnologias a serem utilizadas e posteriormente aplicação da metodologia ágil scrum para as fases de desenvolvimento e implementação do sistema.

Como resultado, o sistema *WEB* desenvolvido conta com uma interface gráfica agradável e de fácil manuseio e que qualquer pessoa que disponha de *internet* possa acessar. O sistema possui uma área pública onde é possível qualquer usuário visualizar diversas informações referentes aos estágios e vagas de estágio. O SGE conta ainda com uma seção administrativa, onde somente o coordenador de estágios possui acesso, destinada ao gerenciamento de todas as informações pertinentes aos estágios e vagas de estágios. A área administrativa dispõe de diversos outros cadastros, tais como: Cadastro de alunos, Empresas concedentes, Cursos, Supervisores e Orientadores dos estágios. O SGE dispõe ainda de relatórios uteis, como por exemplo: boletim de estágios, declaração de orientações, relatórios gerenciais. Na seção destinada aos alunos, o sistema permite a candidatura a vagas de estágio, disponibilizando o currículo para empresas no

ato da candidatura. Os alunos também podem acompanhar as informações dos estágios em que estão matriculados e das vagas de estágios em que são candidatos. E por fim o sistema conta com uma seção a ser utilizada por empresas concedentes de estágio de modo que essas empresas possam verificar as candidaturas dos alunos as vagas, visualizando seus currículos e classificando ou desclassificando esses alunos após a análise das informações. A figura 7 mostra uma interface desse sistema após implementado.

O sistema após implementado se encontrava ainda em fase de implantação e pendente de ser submetido a um teste de usabilidade, visando avaliar as principais funcionalidades do sistema. Os testes devem ser realizados por partes interessadas do sistema. O sistema desenvolvido no entanto, possibilitava a gestão dos estágios e acesso automatizado a informação ali cadastrada.

Figura 7 – Tela da página administrativa do SGE

**Sistema Gerenciamento de Estágios**

Bem Vindo Warley Motat

Novo Cadastro

Estagiário	Tipo de Estágio	Status	Carga Horária Concluída	Início	Término	Empresa Concedente	Curso	Opções
Thales Vinicius Alves de Brito	Obrigatorio	Concluido	1200	03/02/2014	03/02/2016	Emater	Análise de Sistemas	Visualizar

Filtros

Digite o nome do Estagiário para Filtrar

Estagiário:

Empresa Concedente:

Limpar Filtros Filtrar

Copyright 2015 - TADS IFNMG Januária

Fonte: Brito e Oliveira (2016)

## 2.5 Análise Comparativa

Esta seção apresenta uma análise comparativa entre os trabalhos apresentados neste capítulo e a solução proposta neste trabalho. O quadro 1 apresenta o resumo dessa análise.

Os trabalhos que foram apresentados tem em comum otimizar através do uso de um sistema de informação temas comuns em IES. Cada um desses trabalhos, apesar de resolverem problemas parecidos, usam diferentes abordagens, com soluções com diferentes funcionalidades, sendo algumas delas semelhantes aos do trabalho que está sendo desenvolvido. Dessa forma foram definido alguns critérios de comparação entre esses trabalhos e o presente trabalho:

- **Ferramenta *WEB* para Estágios Supervisionados:** Utilizado para identificar se a ferramenta *WEB* desenvolvida tem como foco fluxos relacionados a estágios supervisionados.
- **Envio de documentação:** Utilizado para verificar se a solução proposta permite o envio de algum tipo de documentação para análise e aprovação de coordenadores dentro do sistema.
- **Notificações *Push*:** Utilizado para verificar se a solução proposta permite algum tipo de notificação de atualização de estado em uma solicitação através do uso de notificações *Push*.
- **Adaptado para o fluxo da UFC Quixadá de Estágio-Empresa:** Utilizado para verificar se a solução é adaptável a problemática do campus previamente descrita nesse trabalho.

Todos os trabalhos apresentados nesse capítulo propõem como solução uma ferramenta *WEB*. Com exceção do trabalho de Lima *et al.* (2018), cujo foco é o uso dessa ferramenta para gerenciamento de AC, todos os demais trabalhos, assim como o presente trabalho, tem como foco usar essa ferramenta para o gerenciamento de Estágio Supervisionado.

Com exceção do trabalho de Leoncio (2018), todos os demais trabalhos, assim como o presente trabalho, permitem o envio de algum tipo de documentação para análise e aprovação de coordenadores dentro do sistema, removendo assim a necessidade de manuseio de documentos físicos e possibilitando o acesso a essas informações para posteriores consultas de forma prática e simplificada.

Diferentemente dos trabalhos de Baluz *et al.* (2020), Leoncio (2018) e Brito e Oliveira (2016), este trabalho permite que os diferentes partes interessadas no processo de estágio supervisionado possam ser notificadas através de notificações *Push*, a cada nova mudança de estado em um processo de aproveitamento de estágio-empresa de um aluno da UFC Quixadá até que o aluno tenha sua matrícula efetivada no sistema SIGAA.

Por fim, é importante destacar que o presente trabalho visa solucionar o fluxo específico da UFC Quixadá, que já possui o SIGAA como o sistema informatizado para a gestão das matrículas dos alunos, porém não possui um para a solicitação e acompanhamento do processo de aproveitamento de estágio-empresa do campus, o que é diferente de todos os outros trabalhos citados anteriormente.

Quadro 1 – Quadro comparativo de trabalhos

<b>Trabalho</b>	<b>Ferramenta WEB para Estágios Supervisionados</b>	<b>Envio de documentação</b>	<b>Notificações <i>Push</i></b>	<b>Adaptado para o fluxo da UFC Quixadá de Estágio- Empresa</b>
Baluz <i>et al.</i> (2020)	Sim	Sim	Não	Não
Leoncio (2018)	Sim	Não	Não	Não
Lima <i>et al.</i> (2018)	Não	Sim	Sim	Não
Brito e Oliveira (2016)	Sim	Sim	Não	Não
Este trabalho	Sim	Sim	Sim	Sim

Fonte: Elaborado pelo autor.

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os principais conceitos para o desenvolvimento deste trabalho. Na seção 3.1 são tratados os conceitos de estágio supervisionado de forma genérica e também como funciona esse tipo de estágio na UFC - Campus Quixadá. Na seção 3.2 são tratados os conceitos relacionados a processos de desenvolvimento de um *software* bem como o modelo de desenvolvimento de *software* em cascata. Na seção 3.3 são tratados os conceitos de Sistema *WEB* bem como sobre uma arquitetura de desenvolvimento desse tipo de sistemas, conhecido como MEVN. Por último, na seção 3.4 são tratados os conceitos de *WEB Push Notifications* bem como a apresentação de uma plataforma de envio desses tipos de notificações, denominada OneSignal.

#### 3.1 Estágio supervisionado

A lei federal nº 11.788, de 25 de setembro de 2008, que dispõe sobre o estágio de estudantes, define estágio como:

**Art. 1º:** Estágio é ato educativo escolar supervisionado, desenvolvido no ambiente de trabalho, que visa à preparação para o trabalho produtivo de educandos que estejam frequentando o ensino regular em instituições de educação superior, de educação profissional, de ensino médio, da educação especial e dos anos finais do ensino fundamental, na modalidade profissional da educação de jovens e adultos. (BRASIL, 2008).

Segundo essa mesma lei, o estágio pode ser classificado em: ***Estágio obrigatório***, que é aquele definido como tal no PPC, cuja carga horária é requisito para aprovação e obtenção de diploma, e ***Estágio não-obrigatório***, que é uma atividade desenvolvida como opcional, acrescida à carga horária regular e obrigatória. O estágio, como ato educativo escolar supervisionado, deverá ter acompanhamento efetivo pelo professor orientador da instituição de ensino e por supervisor da parte concedente (BRASIL, 2008).

O estágio propicia uma formação acadêmica associada ao contexto real de atuação no mercado em que será inserido o estudante, possibilitando a construção do conhecimento científico a partir da vivência de situações práticas e do diálogo com profissionais mais experientes. No estágio, o estudante tem a oportunidade de aproximar sua formação conceitual das características de sua comunidade (UFC, 2015a).

Para os profissionais ainda em formação, o estágio é um dos momentos mais importantes e significativos, pois é nesse momento que o futuro profissional tem oportunidade de entrar

em contato direto com a realidade profissional no qual será inserido, além de pôr em prática pressupostos teóricos adquiridos pela observação de determinadas práticas específicas e do diálogo com profissionais mais experientes, assim obtendo a troca de experiências (TORCATE *et al.*, 2018).

O estágio supervisionado geralmente ocorre em empresas, para que a vivência seja mais plena, relacionando-se com outras pessoas, áreas de atuação e clientes, fora do ambiente de sala de aula (COUTINHO *et al.*, 2021).

Coutinho *et al.* (2021, p. 2) também ressaltam que “diversos benefícios tanto para a academia quanto para a indústria são percebidos por experiências de um estágio supervisionado, tais como: interação com ambientes mais reais de empresas, mesmo que simulados; lidar com responsabilidades de entregas, prazos e pressão; e desenvolvimento de habilidades em equipe”.

### **3.1.1 Estágio supervisionado na UFC - Campus Quixadá**

De acordo com UFC (2009), o Estágio Supervisionado, como atividade acadêmica, deve assegurar integração entre teoria e prática, em situação real de vida e trabalho, com vistas à formação profissional e pessoal do estudante. Ademais, toda e qualquer atividade de Estágio assumida pela UFC será curricular e supervisionada.

Com o objetivo de disciplinar o Programa de Estágio Curricular Supervisionado em todas as unidades acadêmicas, a UFC dispõe de um setor denominado Agência de Estágios que é responsável pela articulação, agenciamento e formalização dos estágios obrigatórios e não obrigatórios, bem como pela celebração de convênios para concessão de estágios (UFC, 2015a).

UFC (2009) também dispõe que os estágios na instituição serão realizados mediante a celebração de um Termo de Convênio entre a UFC e a Instituição/Empresa interessada com assinatura do TCE e do Plano de Trabalho, sendo a Agência de Estágios a responsável por parte da UFC.

Para os cursos em que o estágio obrigatório é pré-requisito para a colação de grau, é facultativo ao aluno a modalidade em que o estágio será cumprido, sendo possível a realização de **Estágio Interno**, podendo ser realizado no NPI, iniciativa empreendedora ou projeto de P&D ou o **Estágio Externo**, quando realizado em empresas conveniadas a universidade ou participação em projeto de P&D desenvolvido em outras instituições. Independente da modalidade escolhida, é obrigatório que a atividade a ser desempenhada pelo aluno em seu estágio supervisionado obrigatório, seja relacionado com seu curso e realizadas em áreas de atuação afins com o perfil



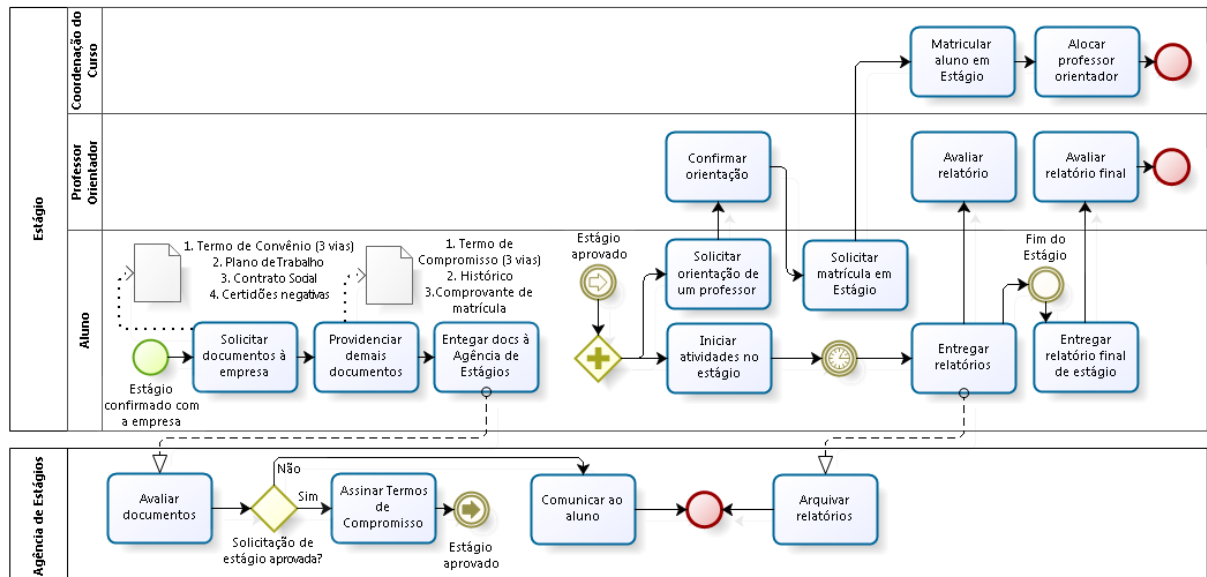
de egresso previsto no PPC.

O NPI é um programa de extensão do campus Quixadá que visa o desenvolvimento de sistemas para a sociedade, e a vivência em um ambiente semelhante a uma empresa, no modelo de fábrica de *software*, com processo definido e projetos com clientes reais (COUTINHO *et al.*, 2021). Para a realização de estágio nesse ambiente, ocorre semestralmente um processo seletivo na qual os alunos interessados e que estejam aptos a realização do estágio - tendo cursado as disciplinas que são pré-requisitos para essa atividade, variando de acordo com o curso no qual o aluno está matriculado - podem se inscrever e caso selecionado, estagiar ali durante todo o semestre letivo.

Para estágio externo em empresas conveniadas (Estágio-Empresa), é necessário que a empresa na qual o aluno deseja realizar o estágio esteja devidamente conveniada e regularizada junto a agência de estágio da universidade, bem como a assinatura do TCE e do Plano de Trabalho (UFC, 2009). A empresa também deverá designar uma pessoa para orientar e supervisionar o estagiário.

A figura 8 mostra o fluxo de processo de estágio supervisionado na UFC.

Figura 8 – Diagrama de fluxo de processo para Estágio Supervisionado da UFC



Fonte: UFC (2015b)

### 3.2 Processo de Desenvolvimento de *Software*

Pressman e Maxim (2016) definem processo de *software* como:

Processo é um conjunto de atividades, ações e tarefas realizadas na criação de algum artefato. Uma atividade se esforça para atingir um objetivo amplo (por exemplo, comunicar-se com os envolvidos) e é utilizada independentemente do campo de aplicação, do tamanho do projeto, da complexidade dos esforços ou do grau de rigor com que a engenharia de *software* será aplicada. Uma ação (por exemplo, projeto de arquitetura) envolve um conjunto de tarefas que resultam em um artefato de *software* fundamental (por exemplo, um modelo arquitetural). Uma tarefa se concentra em um objetivo pequeno, porém bem-definido (por exemplo, realizar um teste de unidades), e produz um resultado tangível. (PRESSMAN; MAXIM, 2016).

De acordo com Sommerville (2019), embora existam diferentes processos de desenvolvimento de *software*, todos eles devem incluir de alguma forma as quatro atividades fundamentais da engenharia de *software*:

1. **Especificação:** A funcionalidade do *software* e as restrições sobre sua operação devem ser definidas.
2. **Desenvolvimento:** O *software* deve ser produzido para atender à especificação.
3. **Validação:** O *software* deve ser validado para garantir que atenda ao que o cliente deseja.
4. **Evolução:** O *software* deve evoluir para atender às mudanças nas necessidades dos clientes.

Dentre os processos de desenvolvimento de *software* mais conhecidos pode-se citar: Modelo cascata, Modelo RAD, Modelo em V, Modelo Incremental, Modelo Evolutivo, Modelo Espiral, entre outros

Para a atividade de especificação, a engenharia de *software* dispõe de uma disciplina denominada Engenharia de Requisitos. De acordo com Pressman e Maxim (2016), essa disciplina estabelece uma base sólida para o projeto e construção de um *software*. Sem ela, o *software* resultante tem grande probabilidade de não atender às necessidades do cliente. É nesse processo que ocorre a comunicação entre o cliente e o analista da equipe de desenvolvimento.

O presente trabalho fará uso das seguintes técnicas da engenharia de requisitos para a etapa de especificação do projeto: Casos de uso, Documento de requisitos, Prototipação de sistema e Modelagem de Banco de dados.

Os casos de uso são uma maneira de descrever as interações entre usuários e um sistema usando um modelo gráfico e um texto estruturado. Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e nomeia o tipo de interação. O conjunto de casos de uso representa todas as interações possíveis que serão descritas nos requisitos de

sistema. Os atores no processo, podem ser humanos ou outros sistemas. Os diagramas de caso de uso estão definidos dentro da *Unified Modeling Language (UML)* (SOMMERVILLE, 2019).

O documento de requisitos de *software* é uma declaração oficial do que os desenvolvedores do sistema devem implementar. Ele pode incluir os requisitos de usuário para um sistema e uma especificação detalhada dos requisitos de sistema. O nível de detalhe que deve ser incluído em um documento de requisitos depende do tipo de sistema que está sendo desenvolvido e do processo de desenvolvimento utilizado (SOMMERVILLE, 2019).

De acordo com Vasquez e Simões (2016) a prototipação é uma técnica que busca simular para o usuário o funcionamento dos seus requisitos antes que o produto final esteja pronto. É utilizado para entre outras coisas, melhorar a experiência do usuário, avaliar opções de *design* e montar uma base para o desenvolvimento final do produto. A prototipação pode ser classificada como sendo de **baixa fidelidade**, onde os protótipos são desenhados na forma de esboço e sem muitos detalhes não havendo a preocupação prioritária com as características estéticas, e de **alta fidelidade**, onde é apresentado um protótipo de alta fidelidade e com um nível alto de detalhes do ponto de vista da aparência, sendo o usuário capaz de enxergar de modo mais realista possível como será o produto final.

A Modelagem de Banco de dados é a etapa onde a partir dos requisitos do sistema são definidos as entidades a serem criadas no banco de dados e seus relacionamentos entre si. O artefato gerado nessa modelagem pode ser um Modelo Entidade-Relacionamento (MER) ou Diagrama Entidade-Relacionamento (DER). Nesse trabalho será realizado uma adaptação a esses artefatos, uma vez que a aplicação será construída usando um banco não relacional baseado em documentos *JavaScript Object Notation (JSON)* através da utilização do MongoDB.

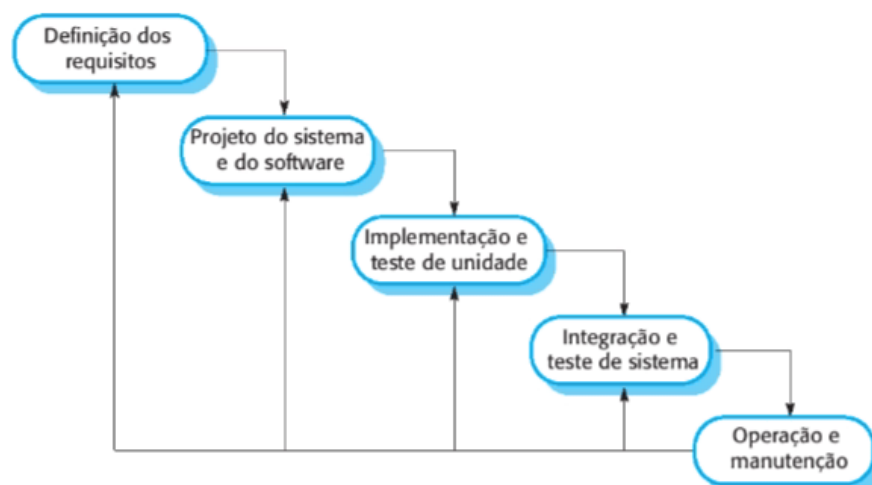
### 3.2.1 Modelo de desenvolvimento de software em cascata

O modelo de desenvolvimento de *software* em cascata, é um modelo que apresenta o desenvolvimento de *software* como uma série de estágios. Ele recebe essa denominação devido à cascata de uma fase para outra. Esse modelo é um exemplo de processo dirigido por plano, e que a princípio, pelo menos, é necessário planejar e criar um cronograma de todas as atividades de processo antes de começar o desenvolvimento de *software*. O resultado de cada fase no modelo gera um artefato e a próxima etapa não pode iniciar enquanto a anterior não estiver concluída. (SOMMERVILLE, 2019). A figura 9 ilustra as etapas do Modelo de desenvolvimento de *software* em cascata.

Sommerville (2019) define esse modelo em 5 etapas:

1. **Análise e definição dos requisitos:** Os serviços, as restrições e as metas do sistema são estabelecidos por meio de consulta aos usuários. Depois, eles são definidos em detalhes e servem como uma especificação do sistema.
2. **Projeto do sistema e do *software*:** O processo de projeto do sistema reparte os requisitos entre requisitos de sistemas de hardware e de *software*, e estabelece uma arquitetura global do sistema. O projeto de *software* envolve a identificação e a descrição das abstrações fundamentais do sistema de *software* e seus relacionamentos.
3. **Implementação e teste de unidade:** Durante essa etapa, o projeto do *software* é realizado como um conjunto de programas ou unidades de programa. O teste de unidade envolve a verificação de cada unidade, conferindo se satisfazem a sua especificação.
4. **Integração e teste de sistema:** As unidades de programa ou os programas são integrados e testados como um sistema completo a fim de garantir que os requisitos de *software* tenham sido cumpridos. Após os testes, o sistema de *software* é entregue ao cliente.
5. **Operação e manutenção:** Normalmente, essa é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em uso. A manutenção envolve corrigir os erros que não foram descobertos nas primeiras fases do ciclo de vida, melhorar a implementação das unidades do sistema e aperfeiçoar os serviços do sistema à medida que novos requisitos são descobertos.

Figura 9 – Modelo em cascata



Fonte: Sommerville (2019)

Para a execução desse projeto será utilizado uma adaptação desse modelo de desenvolvimento de *software*. Devido ao escopo desse trabalho, nessa adaptação foi removido a etapa

de “Operação e manutenção”, além de algumas outras pequenas mudanças nas demais etapas, de modo que esse modelo se tornasse apropriado para esse trabalho

### 3.3 Sistema *WEB*

A *WEB* foi concebida em 1989, buscando criar um meio de troca de dados utilizando a infraestrutura da internet e se caracteriza como uma vasta coleção mundial de documentos, geralmente chamados páginas *WEB* e tendo como principal característica o hipertexto, que significa que cada página pode conter links para outras páginas em qualquer lugar do mundo e os usuários podem seguir esses links através de um simples clique (TANENBAUM; WETHERALL, 2011).

A partir da sua criação, a *WEB* se popularizou e passou em poucos anos a estar presente no cotidiano das pessoas, fazendo com que em grande parte das atividades desenvolvidas por um indivíduo, a *WEB* se encontra atuante (VIDOTTI *et al.*, 2019). Se comunicar com outras pessoas, assistir filmes, escutar músicas, ler livros, participar de reuniões de trabalho, tudo isso são exemplos de atividades que hoje em dia podem ser desenvolvidas por um indivíduo através do uso de sistemas *WEB*.

Segundo Noleto (2020), um sistema *WEB* é um tipo de aplicação que pode ser executado diretamente no navegador de internet do usuário, como por exemplo Google Chrome, Mozilla Firefox, Microsoft EDGE, Safari, entre outros, não havendo a necessidade de uma instalação daquela aplicação no computador do usuário. Esses navegadores são responsáveis por buscarem a página solicitada, interpretar seu conteúdo e exibir a página, formatada de modo apropriado na tela do dispositivo do usuário (TANENBAUM; WETHERALL, 2011).

A principal diferença de um sistema *WEB* para um *site*, é que naquele, há uma possibilidade de um usuário ter uma interação com o sistema, enquanto que um *site* tende a ter uma informação mais estática, sendo apenas um conjunto de páginas que normalmente informam o usuário sobre alguma coisa (NOLETO, 2020).

As principais tecnologias necessárias para a construção de um sistema *WEB* são:

- **HTML:** *HyperText Markup Language (HTML)* (Linguagem de marcação de hipertexto), foi introduzida com a *WEB* e que permite que sejam produzidas páginas que incluem texto, gráficos, vídeos, links para outras páginas na *WEB* e muito mais. Essa linguagem, por ser de marcação permite descrever como os documentos (páginas) devem ser formatados. O navegador então é responsável por interpretar os comandos de marcação e aplicá-los ao

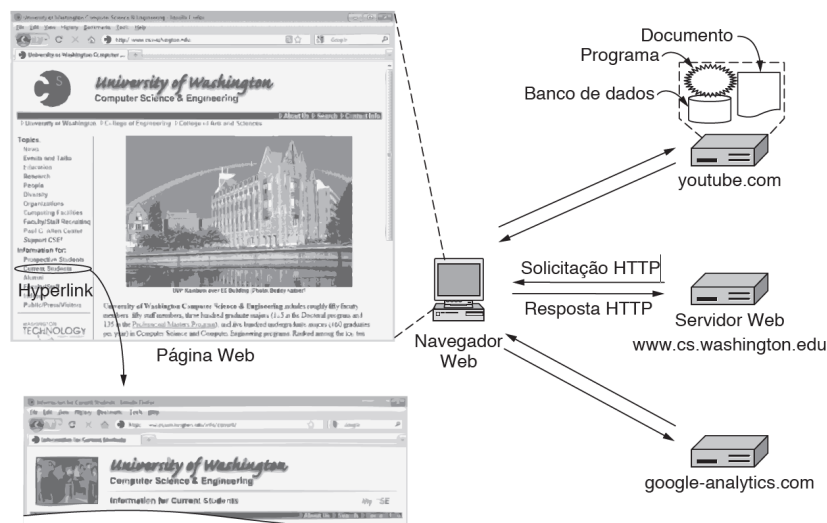
conteúdo, renderizado o conteúdo de uma forma amigável ao usuário final. Atualmente se encontra na versão 5.0 (TANENBAUM; WETHERALL, 2011).

- **CSS: *Cascading Style Sheets* (CSS)** (Folhas de estilo em cascata) são estilos que definem a aparência dos elementos *HTML*. Foi criada para dar suporte a *HTML*, separando, de modo adequado, o estilo do conteúdo da página. Os estilos podem ser armazenados em folhas de estilos dentro do próprio arquivo *HTML* ou em folhas de estilos externas: arquivos *CSS*. O nome *Cascading* foi utilizado porque um estilo definido é aplicado para todos os elementos dentro de um documento (CASAS, 2008).
- **JavaScript:** Linguagem de programação interpretada que é responsável por adicionar interatividade às páginas *HTML* e que são muito utilizados por exemplo para a validação de formulários no cliente, antes de serem enviados ao servidor (CASAS, 2008). Com a criação do Node.js, agora é possível a execução de javascript fora de uma páginas *WEB*, sendo possível o desenvolvimento de um sistema *WEB* de ponta a ponta - do cliente ao banco de dados - apenas utilizando essa tecnologia.
- **Linguagem de programação utilizada na aplicação no lado servidor:** Responsável pelo processamento e geração de páginas e conteúdos dinâmicos de acordo com o recurso solicitado e os dados de entrada enviados na requisição. Diversas linguagens podem ser utilizadas para esse propósito, como por exemplo: Java, Go, Python, PHP, Node.js, entre outros.
- **Banco de dados:** Responsável por agrupar dados sobre um mesmo assunto, armazenando assim diversos dados como documentos, endereços, serviços, clientes, entre outros. Desta forma, um banco de dados serve para que estes dados sejam armazenados e consumidos posteriormente. Para seu gerenciamento, é necessário a utilização de um Sistema de Gerenciamento de Banco de Dados (SGBD), são eles os *softwares* responsáveis por armazenar, listar, pesquisar, remover e qualquer outra manipulação feita em um banco de dados. Existem dois tipos de SGBD: Os relacionais e os não relacionais (NoSQL), onde os relacionais modelam os dados no formato de tabelas, que podem se relacionar entre si, e os não relacionais, onde é utilizado diversos modelos de dados incluindo documentos, gráficos, chave-valor e colunares. (ANDRADE, 2021)

Em um sistema *WEB*, é necessário um servidor para efetuar a comunicação entre as requisições e as respostas a essas requisições, possibilitando que o usuário solicite um recurso, através de um navegador, que retorne para ele documentos em formato PDF, imagens e gráficos

ou páginas em *HTML* após o servidor encontrar e responder o recurso requisitado (GUANAIS, 2008 apud BORGES, 2013). Essa comunicação é realizada através de um protocolo chamado de *Hyper Text Transfer Protocol (HTTP)* e as páginas *HTML* retornadas podem ser **estáticas**, quando apresentam o mesmo documento toda vez que for exibida, ou **dinâmicas**, quando a página é gerada sob demanda por um programa ou se contém um programa (TANENBAUM; WETHERALL, 2011). A figura 10 ilustra esse fluxo geral de funcionamento da *WEB*.

Figura 10 – Fluxo Geral de funcionamento da *WEB*



Fonte: Tanenbaum e Wetherall (2011)

Diversas arquiteturas podem ser utilizadas para o desenvolvimento de sistemas *WEB*, como por exemplo: *MongoDB, Express, Angular, and Node.js (MEAN)*, *MongoDB, Express, React, and Node.js (MERN)*, *MEVN*, *XAMPP*, *LAMP*, entre outras. Como esse trabalho irá utilizar a arquitetura *MEVN*, maiores detalhes sobre esse tipo de arquitetura serão informados na seção 3.3.1.

### 3.3.1 Arquitetura *MEVN*

A arquitetura de um sistema de *software* é a forma dada a esse sistema pelos seus criadores. Essa forma está na divisão desse sistema em componentes, na organização desses componentes e nos modos como esses componentes se comunicam entre si. O propósito dessa forma é facilitar o desenvolvimento, implantação, operação e manutenção do sistema de *software* contido nela (MARTIN, 2019).

A arquitetura *MEVN* é uma conjunto de tecnologias baseadas em javascript para desenvolvimento de aplicações *WEB*, que combinadas possibilitam o desenvolvimento de uma

aplicação que utiliza javascript de ponta a ponta, da aplicação cliente ao banco de dados. Essa arquitetura recebe esse nome pois se trata de um acrônimo para cada tecnologia que a compõe:

- **MongoDB:** Banco de dados de documentos de código aberto, não relacional, que provê persistência para os dados de uma aplicação e que foi projeto com escalabilidade e agilidade do desenvolvedor em mente. Em vez de armazenar dados em linhas e colunas como faria um banco de dados relacional, o MongoDB armazena documentos *JSON* em coleções com esquemas dinâmico (MORGAN, 2020).
- **Express:** *Framework* javascript de código aberto, que oferece um grande conjunto de funcionalidades para aplicações *WEB* e *mobile* desenvolvidas com Node.js. Esse *framework* estende as capacidades do servidor padrão do Node.js, adicionando/melhorando diversas características, tais como: Maior facilidade de configuração, adição de *middlewares*, possibilidade de trabalhar com diferentes *engines* de *templates* para a construção de interfaces, maior facilidade para trabalhar com rotas, *cookies* e sessão e por fim melhoria na segurança. Esse *framework* alcançou uma posição dominante na indústria como *framework* voltado para aplicações servidoras desenvolvidas em Node.js (OLUYEGE, 2019).
- **Vue.js:** *Framework* progressivo javascript para a construção de interfaces de usuário. Ao contrário de outros *frameworks* monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal é focada exclusivamente na camada visual (*view layer*), sendo fácil adotar e integrar com outras bibliotecas ou projetos existentes. Uma das principais características desse *framework* é a possibilidade de reutilização de componentes de uma forma simplificada. O Vue.js nasceu como uma ferramenta para rápida prototipação, mas agora pode ser usado para construir complexas e escaláveis aplicações *WEB* reativas. (VUE.JS, 2021).
- **Node.js:** Programa de código-aberto construído a partir do motor javascript utilizado pelo Google Chrome, chamado V8<sup>1</sup>, que permite a execução de código javascript fora de um *browser* e que consiste em bibliotecas javascript que são implementados na forma de módulos. O ambiente de servidor multiplataforma é leve e eficiente porque usa um modelo de Entrada e Saída de dados sem bloqueio e orientado a eventos (OLUYEGE, 2019). Essa orientação a eventos significa que o aplicativo faz uma solicitação e continua trabalhando em outras tarefas úteis, em vez de parar e aguardar uma resposta. Após a conclusão da tarefa solicitada, o aplicativo é informado dos resultados por meio de um *callback*. Isso

---

<sup>1</sup> <https://v8.dev/>

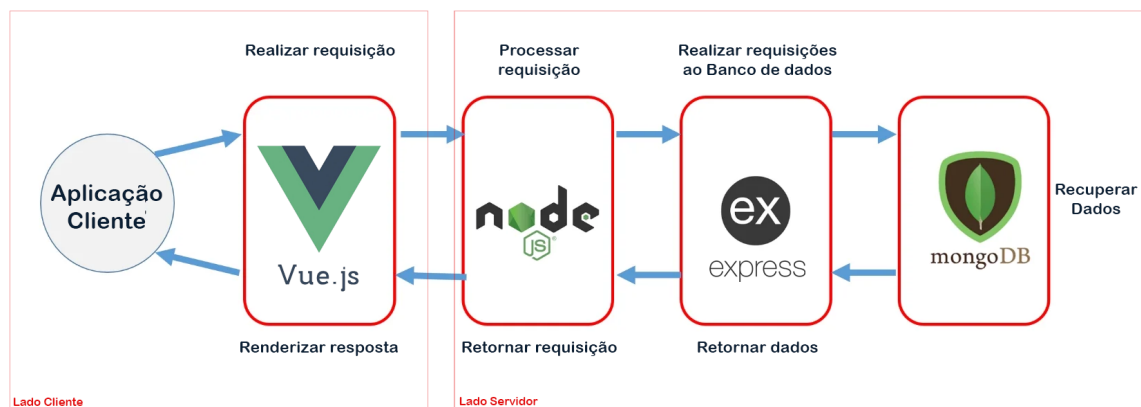


permite que um grande número de operações seja executado em paralelo, o que é essencial ao escalar aplicações. O Node.js também inclui vários módulos que fornecem recursos essenciais para a implementação de aplicativos *WEB*, como por exemplo protocolos de rede como *HTTP* (MORGAN, 2020).

Existem duas variações a essa arquitetura que também são muito conhecidas e utilizadas: *MEAN* e *MERN*. Todas elas com o mesmo princípio de usar tecnologias baseadas em javascript para o desenvolvimento de aplicações *WEB* de ponta a ponta, tendo apenas o *framework* usado para a construção das interfaces da aplicação cliente como diferencial entre elas: A arquitetura *MERN* utiliza o *framework* React<sup>2</sup>, e a arquitetura *MEAN* utiliza o *framework* Angular<sup>3</sup>.

A figura 11 mostra como essa arquitetura funciona de ponta a ponta: O usuário primeiro interage com a interface do usuário, que é construída usando a estrutura de *front-end* Vue.js. As solicitações realizadas na interface são passadas para o Node.js, o mecanismo do lado do servidor. Em seguida, a estrutura de *middleware* conhecida como Express.js faz uma solicitação ao MongoDB, que é o banco de dados. O Express.js recupera a resposta na forma de dados do banco de dados. Por fim, essa resposta é retornada pelo Node.js ao cliente por meio da exibição dessa resposta para na interface do usuário (OLUYEGE, 2019).

Figura 11 – Arquitetura *MEVN*



Fonte: Adaptado de Oluyege (2019)

<sup>2</sup> <https://pt-br.reactjs.org/>

<sup>3</sup> <https://angular.io/>

### 3.4 WEB Push Notifications

Segundo Google Developers (2019), uma notificação é uma mensagem que aparece no dispositivo de um usuário. Essas notificações podem ser acionadas localmente, ou podem ser enviadas de um servidor para o usuário, mesmo quando o aplicativo não está em execução. Esse último tipo, enviado por um servidor, é chamado de *Push notification*.

Essa funcionalidade de *Push notification* por depender de um serviço que fica em execução em segundo plano para receber e lidar com uma mensagem enviada por um servidor, independente de ter o sistema destino dessa mensagem em execução, por muito tempo só esteve presente em aplicativos *mobile* nativos.

Com a evolução dos navegadores de internet modernos e implementação de algumas *APIs* para esses navegadores, surgiu então as *WEB push notifications*, que é um tipo de notificação que pode ser enviado diretamente para sistemas *WEB* rodando diretamente no *browser*, mesmo em computadores.

Esse tipo de notificação permite que os usuários optem por atualizações oportunas de *sites* que eles utilizam frequentemente e permitem que um sistema *WEB* possa reengajar seus usuários com conteúdo personalizado e relevante (MEDLEY, 2019).

Para isso, são utilizadas três *APIs* dos navegadores de internet modernos:

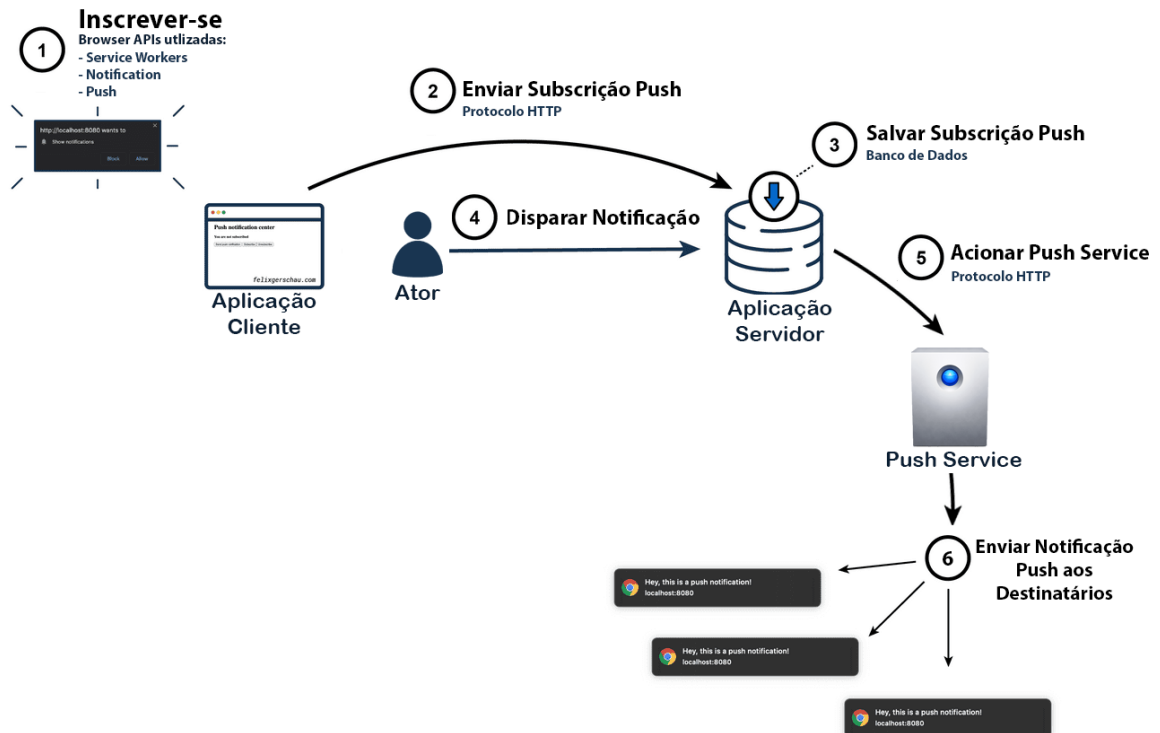
- **Notifications:** Permite que páginas *WEB* mostrem notificações do sistema para o usuário final. Por estar vinculado a um contexto mais geral do navegador, essas notificações podem ser exibidas mesmo que o usuário tenha mudado de aba ou até mesmo quando tenha mudado para um outro aplicativo (MDN Web Docs, 2021a).
- **Push:** Oferece aos sistemas *WEB* a capacidade de receber mensagens enviadas a ele de um servidor, independentemente de o sistema estar ou não executando em primeiro plano, ou mesmo carregado no momento em um navegador. Essa *API* permite que os desenvolvedores forneçam notificações e atualizações assíncronas aos usuários que optarem por participar, resultando em melhor envolvimento com novos conteúdos oportunos (MDN Web Docs, 2021b).
- **Service workers:** *Script* que um navegador executa em segundo plano, separado da página *WEB* e que possibilita recursos que não precisam de uma página da *WEB* ou de interação do usuário. Os *service workers* oferecem o suporte necessário para a inclusão de várias funcionalidades em sistemas *WEB* que só estavam disponíveis em aplicativos nativos, como por exemplo: Experiências *off-line* avançadas, sincronizações periódicas em segundo plano

e as notificações *push* (GAUNT, 2021).

A figura 12 mostra o fluxo de funcionamento para o envio de *WEB Push Notifications* para um usuário, sendo eles:

1. **Inscriver-se:** O *website* solicita permissão para envio de notificações ao usuário. Essa solicitação é feita através da *API* do navegador *Notification* e resulta em uma caixa de confirmação sendo mostrada no navegador na qual o usuário pode aceitar ou negar a solicitação a essa permissão. Caso o usuário aceite a notificação, por meio do contexto do *service worker* em execução na aplicação cliente, é possível realizar o acesso a *API Push* do navegador e obter um objeto de subscrição *push* do usuário, que contém toda a informação obrigatória para que uma notificação *push* possa ser enviada aquele usuário.
2. **Enviar Subscrição Push:** O objeto de subscrição *push* gerado no passo anterior é enviado, através do protocolo *HTTP*, para o servidor do sistema *WEB*.
3. **Salvar Subscrição Push:** O objeto de subscrição *push* é salvo no banco de dados utilizado pelo sistema *WEB* para posterior consulta quando precisar ser enviado notificações para todos os usuários alvo daquela notificação.
4. **Disparar Notificação:** Um ator externo (um usuário) ou interno (o próprio sistema) inicia o fluxo de disparo de uma notificação *push* para um usuário ou grupo de usuários. O servidor *WEB* então recupera no banco de dados todos os objetos de subscrição *push* dos usuários alvos.
5. **Acionar Push Service:** O servidor *WEB* não pode enviar diretamente para o destinatário a notificação *Push*. Ao invés disso, para cada objeto de subscrição *push*, o servidor deve fazer uma chamada *HTTP* para a *URL* contida naquele objeto e também sua chave de encriptação privada.
6. **Enviar Notificação Push aos Destinatários:** Com a posse da chave de encriptação privada, o servidor de *push service* então valida se o remetente realmente pode enviar aquela notificação para aquele destinatário. Caso positivo, então o *push service* faz o envio da notificação *push* para o destinatário. Cabe ressaltar que cada navegador possui seu próprio servidor de *push service*.

Figura 12 – Fluxo de funcionamento para envio de *WEB Push Notifications*



Fonte: Adaptado de Gerschau (2020)

### 3.4.1 One Signal

A OneSignal<sup>4</sup> é uma empresa líder de mercado em mensagens e engajamento de clientes, oferecendo notificações *push* para dispositivos móveis e *WEB*, mensagens internas em aplicativos, SMS e email. A plataforma permite que um milhão de empresas entreguem mais de 10 bilhões de mensagens diariamente. Com uma excelente arquitetura, a plataforma da OneSignal foi projetada para escalar com os negócios de seus clientes e entregar mensagens de forma mais rápida e confiável do que a concorrência (ONESIGNAL, 2022).

Quando se trata de *WEB push notifications*, não há uma uniformidade no processo de configuração dessas notificações entre os diferentes *browsers* no mercado. Através do uso do serviço *WEB push* da OneSignal é possível configurar um *site* para oferecer o serviço de notificações *push* com apenas poucos cliques e sem a necessidade de programar nenhuma linha de código, bastando apenas adicionar no *site* os arquivos javascript indicados durante a configuração do serviço no *dashboard* OneSignal, bem como a inclusão de um bloco de código no *site* que permitirá a correta inclusão e configuração da biblioteca *OneSignal* no *site*. Esse serviço no plano gratuito permite uma quantidade ilimitada de usuários inscritos para recebimento de notificação

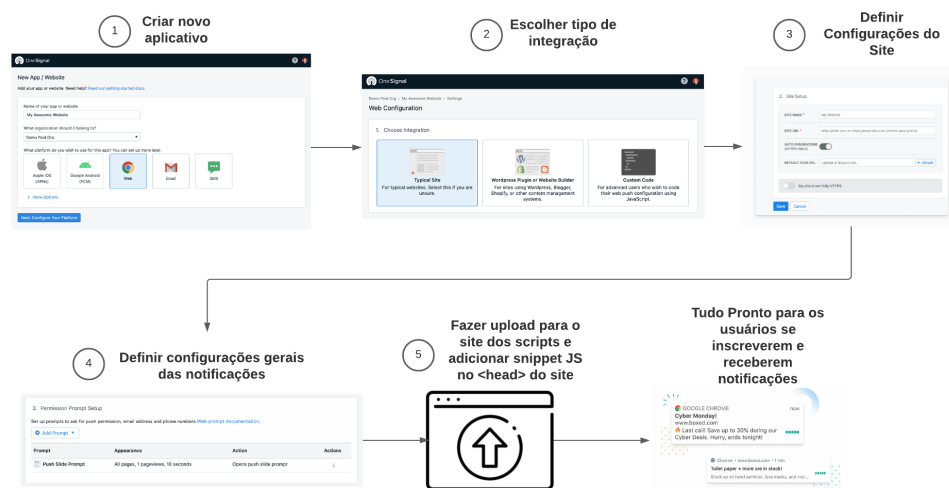
<sup>4</sup> <https://onesignal.com/>

*push*, porém uma notificação é limitada para envio a no máximo 10.000 usuários.

Através do *dashboard*, é possível configurar facilmente modelos de mensagens a serem enviadas, bem como definir segmentos de audiência alvo de uma mensagem. O servidor do sistema *WEB* pode então disparar o envio de uma *push notification* através do uso de uma *API REST* fornecida pela OneSignal.

A figura 13 mostra uma representação gráfica dos passos necessários para configuração do serviço *WEB push* da OneSignal em um *site*. Ao se dirigir ao dashboard da OneSignal, a primeira ação necessária é a criação de um novo aplicativo do tipo *WEB push*. Após isso é necessário a escolha do tipo de integração que será realizado. Após isso é necessário configurar as informações do *site* onde a funcionalidade vai disparar e de onde os usuários vão se inscrever e receber as notificações. Após isso se faz necessário a configuração de algumas preferências relacionadas as notificações a serem exibidas de forma geral, como por exemplo a regra de negócio que fará o *pop-up* de inscrição para recebimento de *push notificações* ser exibido para os usuários no *site*. Por fim a OneSignal irá fornecer os *scripts* necessários a serem incluídos no *site*, bem como um *snippet JS* a ser adicionado a *tag head* do *site* e com isso a configuração estará finalizada. O *pop-up* já estará pronto para ser exibido aos usuários, de acordo com a regra de negócio cadastrado para o aparecimento desse componente.

Figura 13 – Fluxo de configuração OneSignal de *WEB push* em um *site*

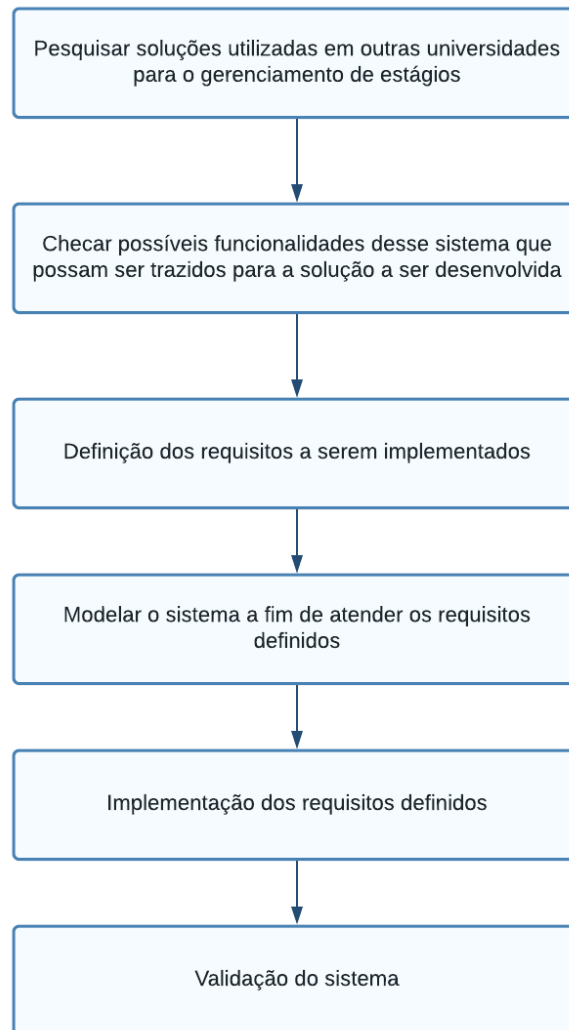


Fonte: Adaptado de OneSignal (2021)

## 4 PROCEDIMENTOS METODOLÓGICOS

Nesta seção serão apresentadas as etapas necessários para a execução do trabalho, além de um planejamento de condução dessas tarefas, para que a solução presente nesse trabalho possa ser modelada, desenvolvida e validada. A figura 14 apresenta a sequência das etapas.

Figura 14 – Passos metodológicos



Fonte: Elaborado pelo autor.

### 4.1 Pesquisar soluções utilizadas em outras universidades para o gerenciamento de estágios

Uma vez que estágio supervisionado é algo que abrange diferentes IES, e não apenas a UFC - Campus Quixadá, a primeira grande tarefa será a de pesquisar quais soluções outras IES utilizam para o processo de gerenciamento de estágios supervisionados de modo que seja

analisado pontos como:

- Como o aluno inicia o processo de matrícula no estágio supervisionado.
- Como o aluno tem acesso a lista de empresas conveniadas com a sua IES.
- Como é realizado o envio de documentação.
- Como ocorre as notificações de atualizações desse processo por parte dos diferentes atores do processo (O coordenador já verificou a documentação enviada? Está faltando alguma coisa? Já foi enviado para a secretaria responsável pela efetivação da matrícula do aluno?)

#### **4.2 Checar possíveis funcionalidades desse sistema que possam ser trazidos para a solução a ser desenvolvida**

Uma vez que as soluções utilizadas em outras IES tiverem sido levantadas, a próxima etapa será realizar uma análise nessas soluções com o objetivo de identificar possíveis funcionalidades que possam ser aplicáveis no contexto do Campus Quixadá e que que poderiam ser acrescentadas a lista de possíveis funcionalidades a serem adicionadas no sistema proposto e que será feito o levantamento adequado na próxima etapa.

#### **4.3 Definição dos requisitos a serem implementados**

Nessa etapa será realizada a junção das funcionalidades úteis e aplicáveis no contexto do Campus Quixadá, que foram identificadas na etapa anterior, com aquelas que serão identificadas juntamente com as partes interessadas envolvidas no processo (secretaria de estágio, professores orientadores de estágio e estudantes de graduação da UFC Quixadá que estão no processo de aproveitamento de estágio-empresa) da UFC Quixadá no processo de levantamento de requisitos.

#### **4.4 Modelar o sistema a fim de atender os requisitos definidos**

Com o conjunto de funcionalidades definidas, o próximo passo será desenhar uma arquitetura de sistema que atenda essas funcionalidades através da realização de uma prototipação do sistema, criação de diagramas *UML* de casos de uso, escrita de um documentos de requisitos e realização da modelagem de banco de dados.

#### 4.5 Implementação dos requisitos definidos

Uma vez que a modelagem estiver pronta, a próxima etapa é a de implementação do sistema com base nesses artefatos gerados na etapa anterior. Conforme fundamentado no capítulo 3, será implementado um sistema *WEB* utilizando a arquitetura *MEVN* e com integração com a plataforma de *WEB push notifications* OneSignal.

#### 4.6 Validação do sistema

E por fim, se faz necessário a validação por meio das partes interessadas envolvidas no processo (secretaria de estágio, professores orientadores de estágio e estudantes de graduação da UFC Quixadá que estão no processo de aproveitamento de estágio-empresa), para se identificar se a solução desenvolvida é válida e se ajuda a resolver ou minimizar o problema.

#### 4.7 Cronograma

No Quadro 2 está descrito o cronograma de atividades a serem executadas para que os objetivos desse trabalho sejam alcançados. O cronograma está disposto em quinzenas dentro de seus respectivos meses.

Quadro 2 – Cronograma de execução das atividades

Atividades	2022													
	Jan		Fev		Mar		Abr		Maio		Jun		Jul	
Defesa TCC1			x											
Pesquisar soluções utilizadas em outras universidades para o gerenciamento de estágios				x	x									
Checar possíveis funcionalidades desse sistema que possam ser trazidos para a solução a ser desenvolvida					x	x								
Definição dos requisitos a serem implementados						x	x							
Modelar o sistema a fim de atender os requisitos definidos						x	x	x						
Implementação dos requisitos definidos								x	x	x	x			
Validação do sistema											x	x		
Escrita do TCC	x	x	x	x	x	x	x	x	x	x	x	x		
Revisão final da monografia													x	x
Defesa do trabalho final														x

Fonte: Elaborado pelo autor.



## REFERÊNCIAS

- ANDRADE, A. P. de. **O que é Banco de Dados?** 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-banco-de-dados>. Acesso em: 20 janeiro 2022.
- BALUZ, R.; ROCHA, F.; SOUZA, I. Plataforma web para gestão dos estágios supervisionados nos cursos de graduação. **Revista Eletrônica de Iniciação Científica em Computação**, v. 18, n. 2, jul. 2020. Disponível em: <https://sol.sbc.org.br/journals/index.php/reic/article/view/1720>.
- BORGES, A. C. Desenvolvendo um sistema web com o extjs. v. 5, dez. 2013. Disponível em: <http://ric.cps.sp.gov.br/handle/123456789/1607>.
- BRASIL. **LEI Nº 11.788, DE 25 DE SETEMBRO DE 2008**. 2008. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/\\_ato2007-2010/2008/lei/l11788.htm](https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11788.htm). Acesso em: 20 jun. 2021.
- BRITO, T. V. A. d.; OLIVEIRA, C. G. Sge - sistema de gerenciamento de estágios. **Anais do IX Simpósio de Informática, IFNMG - Campus Januária, Brasil**, 2016. Disponível em: <http://anais.simposioinformatica.ifnmg.edu.br/ojs/index.php/anaisviiisimposio/article/view/89/73>. Acesso em: 22 nov. 2021.
- CASAS, R. S. L. Tecnologia web aplicadas ao ensino de engenharia de estruturas. dez. 2008. Disponível em: <http://hdl.handle.net/1843/PASA-7THGXQ>.
- COUTINHO, E.; MAGALHÃES, R.; SOUSA, V. Percepções técnicas e sociais de fatores que impactam no desempenho de práticas de estágio supervisionado em tempos de pandemia. In: **Anais do XXVII Workshop de Informática na Escola**. Porto Alegre, RS, Brasil: SBC, 2021. p. 64–74. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/wie/article/view/17835>.
- GAUNT, M. **Service Workers: an Introduction**. 2021. Disponível em: <https://developers.google.com/web/fundamentals/primers/service-workers/>. Acesso em: 16 janeiro 2022.
- GERSCHAU, F. **How to Create Web Push Notifications - Full Tutorial!** 2020. Disponível em: <https://felixgerschau.com/web-push-notifications-tutorial/>. Acesso em: 17 janeiro 2022.
- Google Developers. **Introduction to Push Notifications**. 2019. Disponível em: <https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications/>. Acesso em: 15 janeiro 2022.
- GUANAIS, K. A. Aplicações ricas de internet. v. 5, dez. 2008. Disponível em: <http://www3.iesam-pa.edu.br/ojs/index.php/sistemas/article/view/544>.
- LEONCIO, D. **Sistema para Controle de Estágio Supervisionado do Departamento Acadêmico de Informática**. 2018. Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas), UTFRP (Universidade Tecnológica Federal do Paraná), Paraná, Brasil. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/16809>.
- LIMA, P. G. de; CORREIA, L. H.; SILVA, C. A. G. da. Implementação de um sistema multi-plataforma para gerenciamento de atividades complementares em cursos superiores utilizando banco de dados nosql. **Revista de Informática Aplicada**, v. 14, n. 2, out. 2018. Disponível em: [https://seer.uscs.edu.br/index.php/revista\\_informatica\\_aplicada/article/view/6935](https://seer.uscs.edu.br/index.php/revista_informatica_aplicada/article/view/6935).

MARTIN, R. **Arquitetura Limpa: O guia do artesão para estrutura e design de software**. [S. l.]: Alta Books, 2019. (Robert C. Martin). ISBN 9788550808161.

MDN Web Docs. **Notifications API**. 2021. Disponível em: [https://developer.mozilla.org/en-US/docs/Web/API/Notifications\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API). Acesso em: 16 janeiro 2022.

MDN Web Docs. **Push API**. 2021. Disponível em: [https://developer.mozilla.org/en-US/docs/Web/API/Push\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Push_API). Acesso em: 16 janeiro 2022.

MEC. **Parecer CNE/CES Nº 136/2012**. 2012. Disponível em: [http://portal.mec.gov.br/component/docman/?task=doc\\_download&gid=11205](http://portal.mec.gov.br/component/docman/?task=doc_download&gid=11205). Acesso em: 20 jun. 2021.

MEDLEY, J. **Web Push Notifications: Timely, Relevant, and Precise**. 2019. Disponível em: <https://developers.google.com/web/fundamentals/push-notifications/>. Acesso em: 15 janeiro 2022.

MORGAN, A. **The Modern Application Stack – Part 1: Introducing The MEAN Stack**. 2020. Disponível em: <https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack>. Acesso em: 18 janeiro 2022.

NOLETO, C. **Aplicações web: entenda o que são e como funcionam!** 2020. Disponível em: <https://blog.betrybe.com/desenvolvimento-web/aplicacoes-web/>. Acesso em: 10 janeiro 2022.

OLUYEGE, P. **MongoDB, Express, Angular, and Node.js fundamentals : become a MEAN master and rule the world of web applications**. Birmingham, UK: Packt Publishing, 2019. ISBN 9781789808735.

ONESIGNAL. **Web Push Quickstart**. 2021. Disponível em: <https://documentation.onesignal.com/docs/web-push-quickstart>. Acesso em: 20 janeiro 2022.

ONESIGNAL. **About OneSignal**. 2022. Disponível em: <https://onesignal.com/about>.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software - 8ª Edição**. [S. l.]: McGraw Hill Brasil, 2016. ISBN 9788580555349.

SOMMERVILLE, I. **Engenharia De Software**. [S. l.]: PEARSON BRASIL, 2019. ISBN 9788543024974.

TANENBAUM, A.; WETHERALL, D. **Redes de computadores**. [S. l.]: Pearson Prentice Hall, 2011. ISBN 9788576059240.

TORCATE, A. S.; FARIAS, M. U. F. de; SILVA, S. R. F. da; MARTINS, C. S. A inserção de computação como disciplina no ensino fundamental: Desafios e conquistas em estágio supervisionado. In: **Anais do XXVI Workshop sobre Educação em Computação**. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6175. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/3521>.

UFC. **RESOLUÇÃO Nº 32/CEPE, DE 30 DE OUTUBRO DE 2009**. 2009. Disponível em: [http://www.deecc.ufc.br/Download/Estagio\\_Supervisionado/UFC\\_Resolucao\\_32\\_CEPE\\_30.10.2009\\_Estagio.pdf](http://www.deecc.ufc.br/Download/Estagio_Supervisionado/UFC_Resolucao_32_CEPE_30.10.2009_Estagio.pdf). Acesso em: 20 jun. 2021.

UFC. **Manual de estágios da Universidade Federal do Ceará**. 2015. Disponível em: <https://estagios.ufc.br/wp-content/uploads/2018/11/manual.pdf>. Acesso em: 24 jan. 2022.

UFC. **Procedimento Operacional Padrão 09 - Estágio Supervisionado**. 2015. Disponível em: <https://producao.ufc.br/wp-content/uploads/2016/10/pop-09-estagio-supervisionado.pdf>. Acesso em: 24 jan. 2022.

VASQUEZ, C.; SIMÕES, G. **Engenharia de requisitos: software orientado ao negócio**. [S. l.]: Brasport Livros e Multimídia, 2016. ISBN 9788574527963.

VIDOTTI, S. A. B. G.; CONEGLIAN, C. S.; ROA-MARTÍNEZ, S. M.; VECHIATO, F. L.; SEGUNDO, J. E. S. Web, web semântica e web pragmática: um posicionamento da arquitetura da informação. v. 29, mar. 2019. Disponível em: <https://periodicos.ufpb.br/ojs2/index.php/ies/article/view/44358>.

VUE.JS. **Introdução ao Vue.js**. 2021. Disponível em: <https://vuejsbr-docs-next.netlify.app/guide/>. Acesso em: 19 janeiro 2022.