



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

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C321d Carvalho Sobrinho, Bruno de.

Desenvolvimento de um sistema web para solicitação e acompanhamento do processo de aproveitamento de estágio-empresa da UFC Quixadá. / Bruno de Carvalho Sobrinho. – 2022.
90 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2022.

Orientação: Prof. Dr. Emanuel Ferreira Coutinho.

1. Estágio supervisionado. 2. Aplicações Web. 3. Engenharia de software. 4. Desenvolvimento. I. Título.
CDD 005.1

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.

Aprovada em: ____/____/____

BANCA EXAMINADORA

Prof. Dr. Emanuel Ferreira Coutinho (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

Prof. Dr. David Sena Oliveira
Universidade Federal do Ceará (UFC)

Em memória da minha avó, Raimunda Apolônia de Carvalho, mais conhecida como dona Doca, que já acamada, na noite anterior a minha mudança para Quixadá para iniciar a graduação, pediu-me para que eu não esquecesse dela. Não esqueci da senhora vovó, sua ausência é sentida.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado saúde, coragem e perseverança para enfrentar essa longa trajetória em Quixadá, 637 km de distância de toda minha família.

À minha mãe, Luísa de Carvalho Sobrinho, uma batalhadora, uma lutadora, um exemplo, que sempre fez o possível e o impossível ao seu alcance para me proporcionar a melhor educação possível. Te amo mãe, obrigado por tudo, por todas as orações, conselhos e força.

Aos meus tios, Francisco Ramos Mourão Neto e Kleber Alves de Carvalho, por terem me incentivado e terem feito todo o possível para que a minha jornada em Quixadá fosse possível de ser realizada, sem vocês nada disso teria sido possível.

À minha esposa Joyce Cleide Araujo de Carvalho por todo incentivo, carinho e ajuda para que eu não desistisse de obter minha graduação, mesmo com todos os percalços.

Às minhas tias, Raimundinha, Rozalia, Eliana, Xixica por sempre estarem torcendo, rezando e me apoiando em tudo que era necessário e que estava aos seus alcances.

A toda a minha família, que sempre me apoiou e torceu pelas minhas conquistas.

Aos meus grandes amigos que fiz nessa trajetória, que considero irmãos, em especial Isaac James Mangureira do Nascimento, Isac Moura Gomes e Rodrigo Adamski. Além desses, João Henrique, Jonas Lopes, Jonas Deyvid, Maike Bezerra, Rosembergson e Darliene, todos vocês me ajudaram a encarar a realidade em uma cidade longe de toda minha família de uma forma mais leve e que eu me sentisse acolhido e menos sozinho.

Ao Prof. Dr. Emanuel Coutinho, por toda a sua paciência e compreensão durante essa orientação de TCC.

Ao Prof. Dr. Régis Pires, por ser uma das minhas referências e exemplos no campo de tecnologia desde os tempos de IFPI - Campus Parnaíba e por ter me dado o incentivo e apoio para que eu fizesse a minha matrícula na UFC Quixadá.

À professora Diana Braga e Paulyne Jucá por todos os conselhos e apoios dados durante toda a graduação, e a todos os outros professores da UFC campus Quixadá, por compartilhar toda experiência e conhecimento e ajudar formar.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

“Educação gera conhecimento, conhecimento gera sabedoria, e, só um povo sábio pode mudar seu destino”

(Samuel Lima)

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 a 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 vão 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 de estágio e alunos que realizarão o estágio). 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 à 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; aplicações web; engenharia de software; desenvolvimento.

ABSTRACT

Supervised Internship is a way of preparing students for the environments where they will perform their roles after finishing their graduation. At UFC (Campus Quixadá), for some of the undergraduate courses, such activity is a mandatory component for graduation. The student can choose to have this internship within the campus - developing activities at NPI - or in partner companies of the institution. When the student chooses to have this experience at a partner institution, he is responsible for, in addition to requesting enrollment in the curricular component using the official campus system SIGAA, also sending documentation to the coordinator responsible for internships in the course, in which the student is enrolled. For the student to have his registration in the curricular component confirmed, it is necessary that the coordinator analyze and confirm that everything is right between the college and the company where the student wants to carry out the internship, the activities to be developed by the student are adequate to the course teaching plan, the documentation sent by the student is in compliance with the rules. Only after confirmation, the coordinator sends a confirmation to the teaching coordination informing that the student's enrollment can in fact be carried out. Currently, the UFC does not have a tool in which the sending of this documentation can be centralized, verification of the status of such process, notification of updates in this process (when, for example, the coordinator checks the documentation and verifies that something is missing and the student needs to send something else), notification to the actors involved (internship coordinator and students who will carry out the internship). Currently, at UFC - Campus Quixadá, this whole process is carried out through an adaptation of *Google Classroom*, although for a long time it was through the exchange of *e-mail* between the aforementioned actors previously, which became a problem due to the existence of several professors guiding the internship on the campus, and each one carrying out these steps in their own way. The present work aims to propose a solution and development of a WEB application that allows to centralize this whole process, making it simpler, more intuitive and standardized.

Keywords: supervised internship; web applications; software engineering; development.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de casos de uso do SIGESTAGIOS	22
Figura 2 – Tela de listagem de alunos matriculados no SIGESTAGIOS	22
Figura 3 – Diagrama de casos de uso do sistema controle de estágios	24
Figura 4 – Tela de cadastro de defesa do sistema controle de estágios	24
Figura 5 – Diagrama de casos de uso do sistema Akademic	26
Figura 6 – Dashboard do sistema Akademic	26
Figura 7 – Tela da página administrativa do SGE	28
Figura 8 – Diagrama de fluxo de processos para estágio supervisionado da UFC	33
Figura 9 – Modelo em cascata	36
Figura 10 – Fluxo geral de funcionamento da <i>World Wide Web (WEB)</i>	39
Figura 11 – Arquitetura <i>MongoDB, Express, Vue.js, and Node.js (MEVN)</i>	41
Figura 12 – Fluxo de funcionamento para envio de <i>WEB push notifications</i>	44
Figura 13 – Fluxo de configuração OneSignal de <i>WEB push</i> em um <i>site</i>	45
Figura 14 – Passos metodológicos	46
Figura 15 – Módulo de estágios do SIGAA - UFPI	49
Figura 16 – Diagrama de sequência para funcionalidade de login no sistema	54
Figura 17 – Diagrama de atividades - notificações do sistema	56
Figura 18 – Bibliotecas utilizadas no desenvolvimento do sistema	58
Figura 19 – Organização do projeto	58
Figura 20 – Modelagem do banco de dados	59
Figura 21 – Diagrama de atividades - geração e armazenamento de tokens JSON Web Token (JWT)	60
Figura 22 – Tela de <i>login</i> do sistema	63
Figura 23 – Gerenciamento de orientadores de estágio	64
Figura 24 – Módulo perfil para orientadores	64
Figura 25 – Consultas de empresas conveniadas	65
Figura 26 – Diagrama de atividades - cadastrar e gerar Termo de Compromisso de Estágio (TCE)	66
Figura 27 – Tela de cadastro e geração TCE do sistema	67
Figura 28 – Tela inicial dos orientadores de estágio	68
Figura 29 – Notificações do sistema	70

Figura 30 – 1ª etapa do processo - confirmação de matrícula no SIGAA	71
Figura 31 – 2ª etapa do processo - documentação	71
Figura 32 – 3ª etapa do processo - plano de estágio	72
Figura 33 – 4ª etapa do processo - avaliação de desempenho	72
Figura 34 – 5ª etapa do processo - relatório final de estágio	73
Figura 35 – 6ª etapa do processo - seminário de apresentação de estágios	73
Figura 36 – Visualização de processos pelo orientador	74
Figura 37 – Tela de visualização de mensagens em processo - orientador	74
Figura 38 – Tela de visualização de mensagens em processo - aluno	75
Figura 39 – Processo de acompanhamento de estágio	76
Figura 40 – Passos para cumprir a atividade obrigatória de estágio supervisionado 1 ou 2	76
Figura 41 – Cronograma de estágio em 2022.2	77
Figura 42 – Diagrama de implantação contínua da aplicação	79

LISTA DE QUADROS

Quadro 1 – Quadro comparativo de trabalhos	30
Quadro 2 – Quadro de notificações as partes interessadas de atualizações no processo .	56
Quadro 3 – <i>Middlewares</i> de controle de acesso a recursos no <i>back-end</i>	61

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Exemplo de configuração de notificações	69
--	----

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>
JWT	JSON Web Token
TCE	Termo de Compromisso de Estágio
MEC	Ministério da Educação
P&D	Pesquisa e Desenvolvimento
PPC	Projeto Pedagógico de Curso
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>
UFPI	Universidade Federal do Piauí
CGE	Coordenadoria Geral de Estágios
STI	Superintendência de Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivos	20
1.1.1	<i>Objetivos Gerais</i>	20
1.1.2	<i>Objetivos Específicos</i>	20
2	TRABALHOS RELACIONADOS	21
2.1	<i>Plataforma WEB para Gestão dos Estágios Supervisionados nos Cursos de Graduação (BALUZ et al., 2020)</i>	21
2.2	<i>Sistema para Controle de Estágio Supervisionado do Departamento Acadêmico de Informática (LEONCIO, 2018)</i>	23
2.3	<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>	25
2.4	<i>SGE - Sistema de Gerenciamento de Estágios (BRITO; OLIVEIRA, 2016)</i>	27
2.5	<i>Análise Comparativa</i>	29
3	FUNDAMENTAÇÃO TEÓRICA	31
3.1	Estágio supervisionado	31
3.1.1	<i>Estágio supervisionado na UFC - Campus Quixadá</i>	32
3.2	Processo de Desenvolvimento de Software	34
3.2.1	<i>Modelo de desenvolvimento de software em cascata</i>	35
3.3	Sistema WEB	37
3.3.1	<i>Arquitetura MEVN</i>	39
3.4	WEB Push Notifications	42
3.4.1	<i>One Signal</i>	44
4	PROCEDIMENTOS METODOLÓGICOS	46
4.1	Pesquisar soluções utilizadas em outras universidades para o gerenciamento de estágios	46
4.2	Checar possíveis funcionalidades desse sistema que possam ser trazidos para a solução a ser desenvolvida	47
4.3	Definição dos requisitos a serem implementados	47
4.4	Modelar o sistema a fim de atender os requisitos definidos	47

4.5	Implementação dos requisitos definidos	47
4.6	Validação do sistema	47
5	RESULTADOS	49
5.1	Pesquisar soluções utilizadas em outras universidades para o gerenciamento de estágios	49
5.2	Checar possíveis funcionalidades desse sistema que possam ser trazidos para a solução a ser desenvolvida	50
5.3	Definição dos requisitos a serem implementados	50
5.3.1	<i>Realizar Login no sistema utilizando credenciais da Base Centralizada</i>	50
5.3.2	<i>Cadastrar Orientadores de Estágio dos Cursos</i>	51
5.3.3	<i>Visualizar Empresas Conveniadas para realização de estágio</i>	51
5.3.4	<i>Cadastrar e Gerar de Termo de Compromisso de Estágio</i>	51
5.3.5	<i>Solicitação e Acompanhamento do Processo de Aproveitamento de Estágio-Empresa</i>	52
5.3.6	<i>Apresentar aos alunos informações relevantes sobre o processo de estágio</i>	53
5.4	Modelar o sistema a fim de atender os requisitos definidos	53
5.4.1	<i>Realizar Login no Sistema</i>	53
5.4.2	<i>Cadastrar Orientadores de Estágio dos Cursos</i>	54
5.4.3	<i>Visualizar Empresas Conveniadas para realização de estágio</i>	55
5.4.4	<i>Cadastrar e Gerar de Termo de Compromisso de Estágio</i>	55
5.4.5	<i>Solicitação e Acompanhamento do Processo de Aproveitamento de Estágio-Empresa</i>	55
5.4.6	<i>Apresentar aos alunos informações relevantes sobre o processo de estágio</i>	57
5.5	Implementação dos requisitos definidos	57
5.5.1	<i>Realizar Login no sistema</i>	60
5.5.2	<i>Cadastrar Orientadores de Estágio dos Cursos</i>	63
5.5.3	<i>Visualizar Empresas Conveniadas para realização de estágio</i>	64
5.5.4	<i>Cadastrar e Gerar de Termo de Compromisso de Estágio</i>	65
5.5.5	<i>Solicitação e Acompanhamento do Processo de Aproveitamento de Estágio-Empresa</i>	68
5.5.6	<i>Apresentar aos alunos informações relevantes sobre o processo de estágio</i>	75
5.6	Hospedagem da aplicação	77

5.7	Validação do sistema	79
6	CONCLUSÕES E TRABALHOS FUTUROS	81
6.1	Considerações Finais	81
6.2	Contribuições e Dificuldades	81
6.3	Trabalhos Futuros	82
	REFERÊNCIAS	83
	APÊNDICES	86
	APÊNDICE A– <i>EXEMPLO DE TERMO DE COMPROMISSO DE ES-</i> <i>TÁGIO GERADO PELO SISTEMA</i>	86
	APÊNDICE B– <i>TEMPLATE UTILIZADO PELA BIBLIOTECA DOCX-</i> <i>TEMPLATER PARA GERAÇÃO DE TCE</i>	88

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 recomendada 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 pelo 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 Universidade Federal do Ceará (UFC) - Campus Quixadá, em conformidade com a lei anteriormente citada, apresenta nos projetos pedagógicos de alguns dos cursos da instituição, 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 Núcleo de Práticas em Informática (NPI), projeto de Pesquisa e Desenvolvimento (P&D) ou o *Estágio Externo*, quando realizado em empresas conveniadas à 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 onde o aluno deseja realizar o estágio, esteja devidamente conveniada e regularizada junto à 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 Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), que é o sistema oficial utilizado pela UFC para várias atividades acadêmicas, entre elas para matrícula dos alunos em disciplinas a serem cursadas 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 discente possa, de fato, aproveitar o estágio como o estágio-obrigatório de seu curso. A matrícula do aluno só é efetivada, portanto, 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 à 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 recurso para esse fluxo era problemático, devido ao propósito geral de uma caixa de e-mail e, por exemplo, para solicitações serem esquecidas entre tantas outras 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 gerenciada pela própria 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 *World Wide Web (WEB)* em que 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 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 sistema desenvolvido nesse trabalho encontra-se disponível em um repositório público no GitHub¹.

Além deste capítulo introdutório, o presente trabalho está dividido nos seguintes capítulos: Capítulo 2 - apresenta os trabalhos relacionados a esse projeto, Capítulo 3 - apresenta uma fundamentação teórica com conceitos e temas importantes ao projeto, o Capítulo 4 - apresenta a metodologia deste trabalho, o Capítulo 5 - apresenta os resultados do trabalho, e por fim, o Capítulo 6 apresenta as conclusões e trabalhos futuros.

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 (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.

¹ <https://github.com/ibituruna/Sistema-Estagio-Quixada>

2 TRABALHOS RELACIONADOS

Neste capítulo, serão apresentados alguns trabalhos relacionados a este. Foram selecionados dois trabalhos que se assemelham bastante ao objetivo do presente trabalho, em que também são propostas ferramentas que automatizam o processo de gerenciamento de estágios supervisionados em Instituições de Ensino Superior. Por fim, é apresentada 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 elicítadas e que o sistema deve apresentar, enquanto 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, havia diferentes 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.

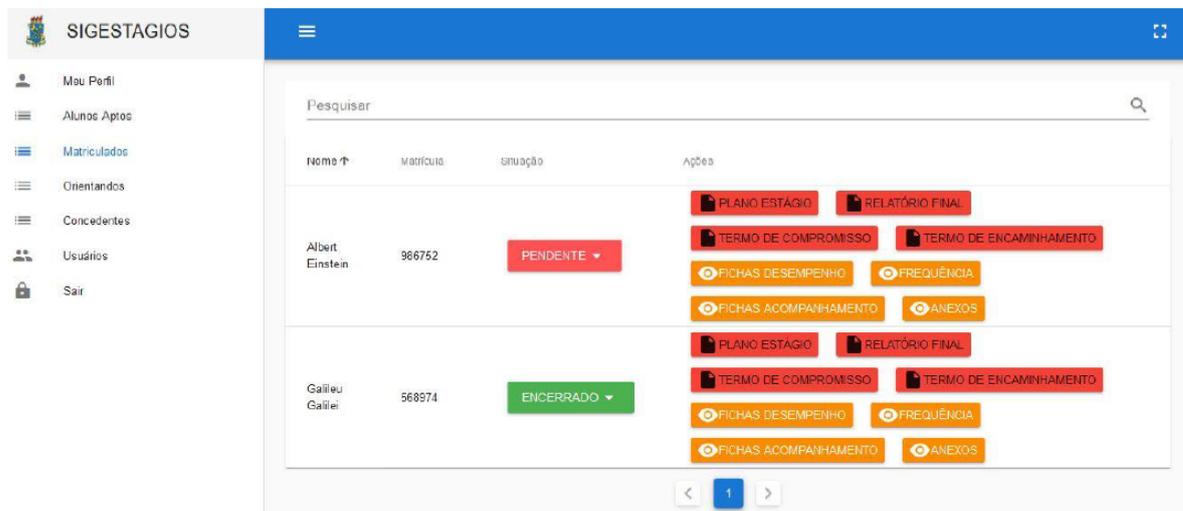
Dentre os resultados coletados, foi possível identificar que 92,31% dos voluntários usariam o sistema, caso ele fosse implementado na instituição, e a média de pontuação final sendo de 81,92, o que é acima dos 68 pontos definidos pelo questionário SUS como uma boa avaliação.

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 aprimorar 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.

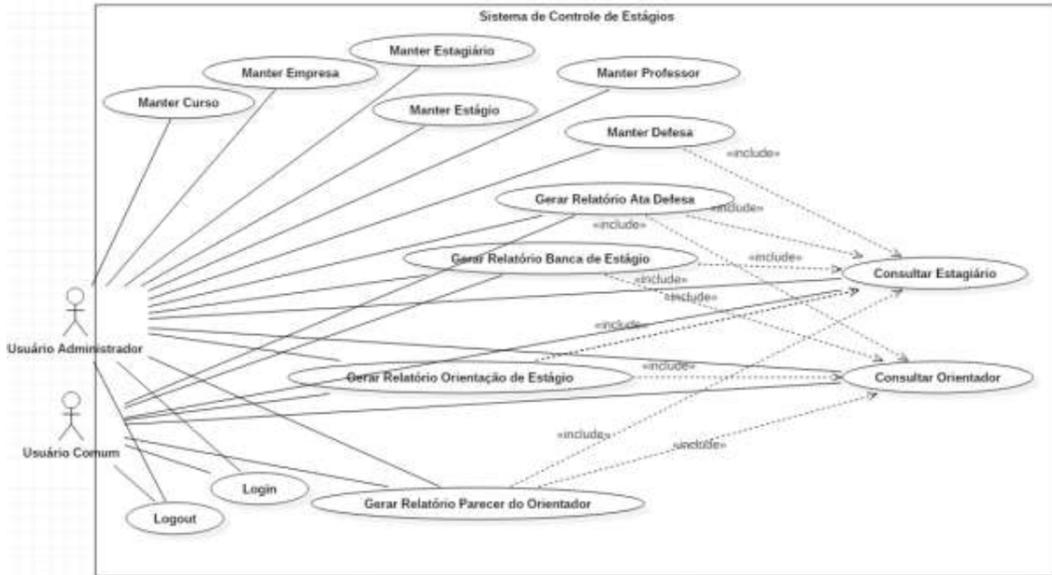
Para o desenvolvimento dessa aplicação, o trabalho foi dividido nas etapas de: Revisão da literatura, Definição das ferramentas a serem utilizadas para o desenvolvimento da aplicação, Definição dos requisitos do sistema, Modelagem do sistema e por fim, Testes do sistema pelos usuários finais.

A Figura 3 ilustra todas as funcionalidades elicitadas e que o sistema deve apresentar, enquanto 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.

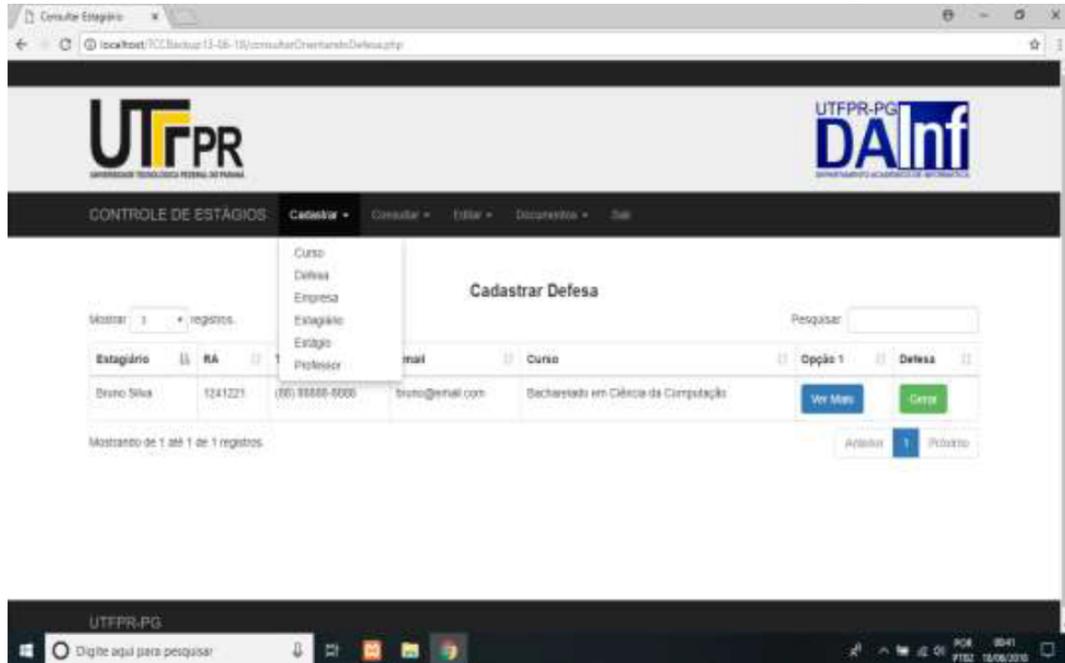
O desenvolvimento dessa aplicação possibilitou a melhoria em uma atividade que necessitava do emprego da tecnologia para que assim houvesse melhor desempenho na realização das atividades administrativas, referentes ao estágio curricular.

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 têm 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 à distância, tornando assim esse fluxo confuso e moroso, tanto para o aluno quando para a Instituição de Ensino Superior (IES), tendo em vista que esse processo acaba gerando uma intensa carga administrativa, devido à quantidade de alunos e comprovantes a serem analisados.

Lima *et al.* (2018) então propuseram um sistema Multiplataforma, 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*.

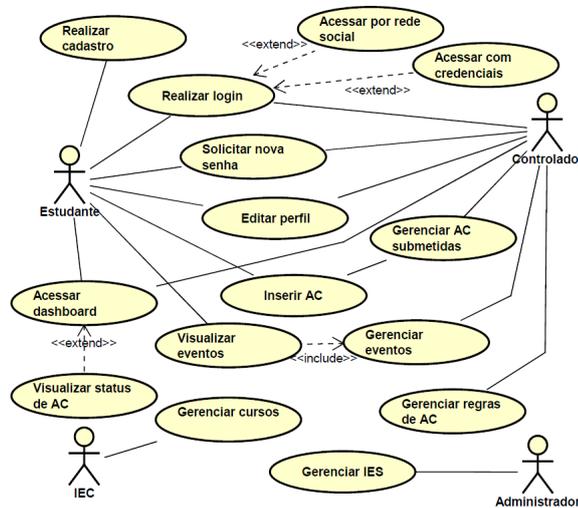
Os passos metodológicos para o projeto foram baseados no manifesto ágil, sendo eles: Levantamento de requisitos, definição de uma arquitetura para o sistema, definição dos modelos de dados, implementação do sistema.

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 elicidadas e que o sistema deve apresentar, enquanto 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 às 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 a 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 úteis, 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 para 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 à informação ali cadastrada.

Figura 7 – Tela da página administrativa do SGE

Sistema Gerenciamento de Estágios

Bem Vindo Warley Mota!

Novo Cadastro

Estagiario	Tipo de Estagio	Status	Carga Horaria Concluída	Inicio	Termino	Empresa Concedente	Curso	Opcoes
Thales Vinicius Alves de Brito	Obrigatorio	Concluido	1200	03/02/2014	03/02/2016	Emater	Analise de Sistemas	Visualizar

Filtros

Digite o nome do Estagiario para Filtrar

Estagiario:

Empresa Concedente:

Limpar Filtros Filtrar

Copyright 2015 - TADS IFNMG Januária

Fonte: Brito e Oliveira (2016)

2.5 Análise Comparativa

Esta seção traz 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 têm 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 definidos 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 à 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, têm 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 as 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 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

Trabalho	Contexto	Envio de documentação	Tipo de Notificação	Fluxo modelado no sistema
Baluz <i>et al.</i> (2020)	Estágios Supervisionados	Relatório de Estagio, Seguro Obrigatório, Ficha de inscrição e Ficha de acompanhamento	Nenhuma Notificação	Estágios supervisionados no curso de Ciência da Computação da Universidade Estadual do Piauí (UESPI) - Campus Parnaíba
Leoncio (2018)	Estágios Supervisionados	Nenhum	Nenhuma Notificação	Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná (UTFPR)
Lima <i>et al.</i> (2018)	Atividades Complementares	Certificado de Atividades Complementares	<i>Web Push Notifications</i>	Fluxo Genérico para Instituições de Ensino Superior para o envio de atividades complementares
Brito e Oliveira (2016)	Estágios Supervisionados	Documentos de estágio	Nenhuma Notificação	Estágio Supervisionado - Instituto Federal do Norte de Minas Gerais (IFNMG) - Campus Januária
Este trabalho	Estágios Supervisionados	Termo de Contrato de Trabalho, Contrato de Trabalho, Plano de Estágio, Relatório Final de Estágio, Relatório de Avaliação de Desempenho	<i>In-app e Web Push Notifications</i>	Estágio supervisionados em Empresa - UFC (Campus Quixadá)

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 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 o acompanhamento efetivo do professor orientador da instituição de ensino e do 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 uma das mais importantes e significativas, pois é nesse momento que o futuro profissional tem oportunidade de entrar

em contato direto com a realidade profissional na 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 relacionada com seu curso e realizadas em áreas de atuação afins com o perfil de

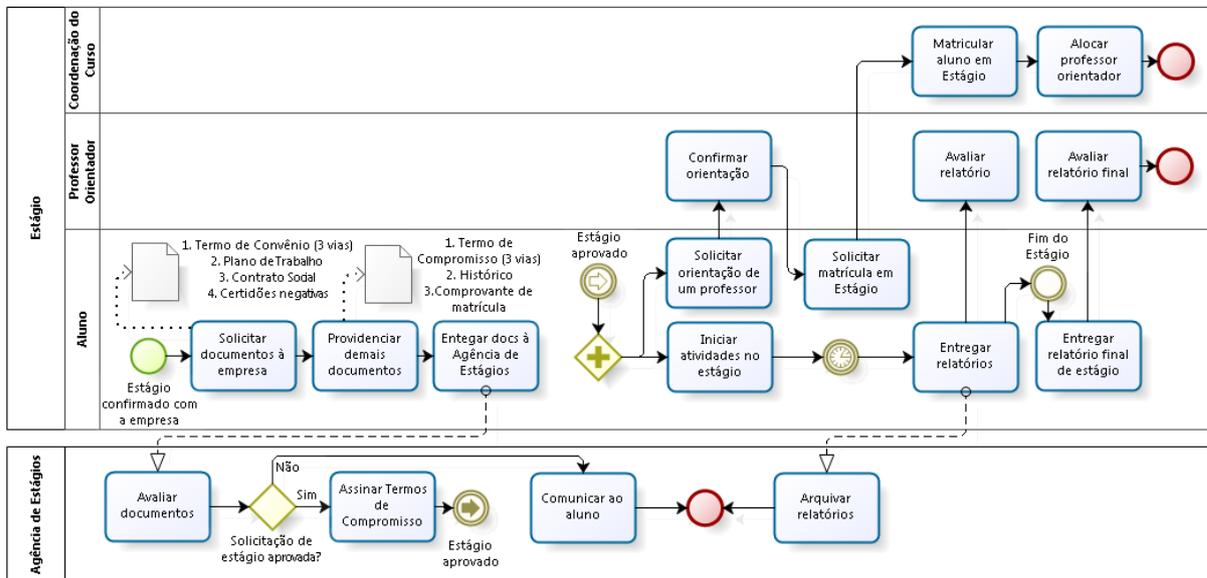
ingresso previsto no PPC.

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 à 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 sejam selecionados, 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 à 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 com os processos necessários para a realização de estágio supervisionado na UFC.

Figura 8 – Diagrama de fluxo de processos 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 definidas 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.

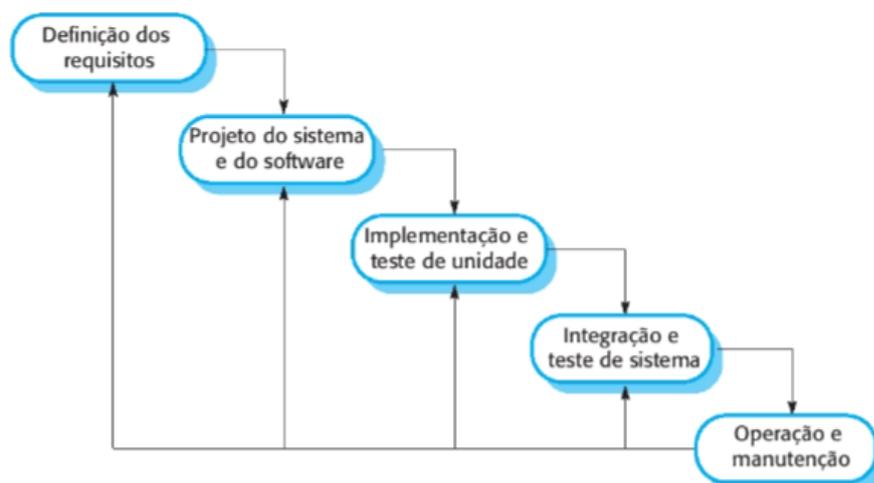
3.2.1 Modelo de desenvolvimento de software em cascata

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 estabelecidas 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 removida 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). Comunicar-se 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 Noletto (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, dentre 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 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

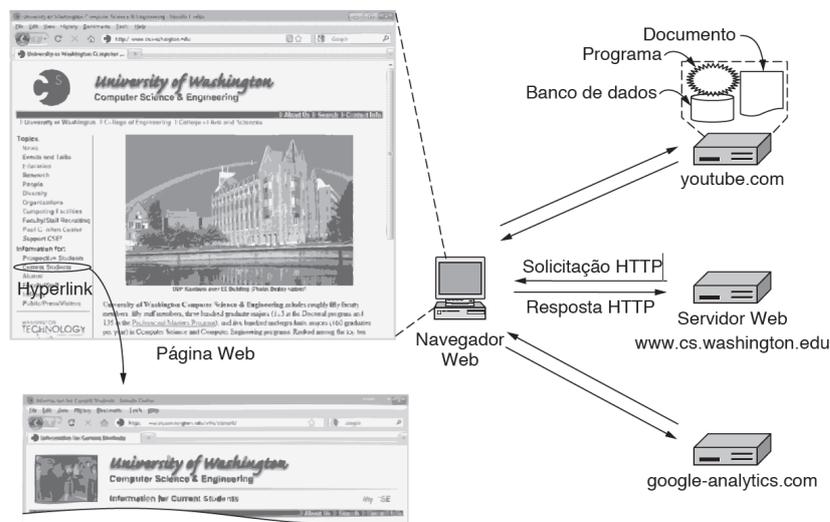
conteúdo, renderizado o conteúdo de uma forma amigável ao usuário final. Atualmente se encontra na versão 5.0 (TANENBAUM; WETHERALL, 2011).

- **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 página *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, JavaScript, 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)*, *MongoDB, Express, Vue.js, and Node.js (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 eles se comunicam entre si. O propósito dessa forma é facilitar o desenvolvimento, a implantação, operação e manutenção do sistema de *software* contido nela (MARTIN, 2019).

A arquitetura *MEVN* é um 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 dinâmicos (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 V8¹, 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

¹ <https://v8.dev/>

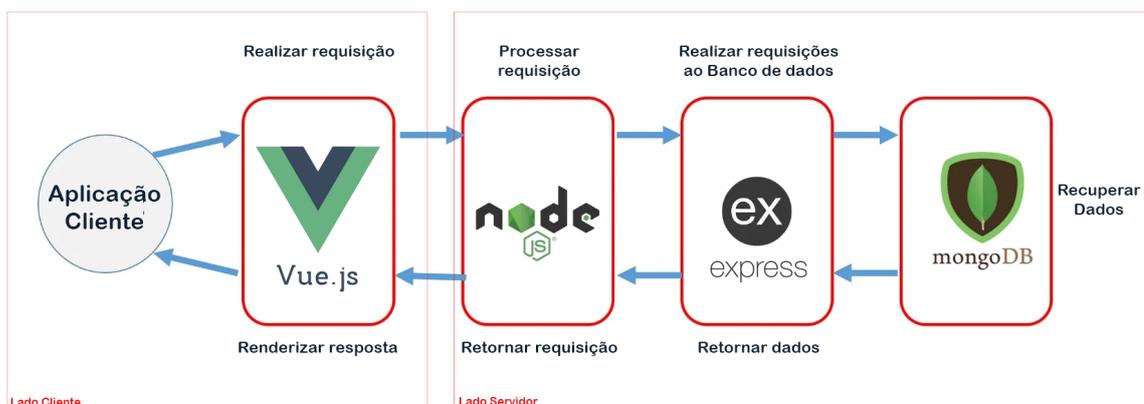
permite que muitas operações sejam executadas em paralelo, o que é essencial ao escalar aplicações. O Node.js também inclui vários módulos que fornecem recursos essenciais para a implementação de aplicativos *WEB*, como por exemplo protocolos de rede como *HTTP* (MORGAN, 2020).

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* React², e a arquitetura *MEAN* utiliza o *framework* Angular³.

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).

Essa arquitetura será utilizada no sistema a ser desenvolvido nesse trabalho devido a menor curva de aprendizagem necessária, uma vez que ela já fora utilizada outras vezes pelo autor desse trabalho.

Figura 11 – Arquitetura *MEVN*



Fonte: Adaptado de Oluyege (2019)

² <https://pt-br.reactjs.org/>

³ <https://angular.io/>

3.4 WEB Push Notifications

Segundo 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, independentemente 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:

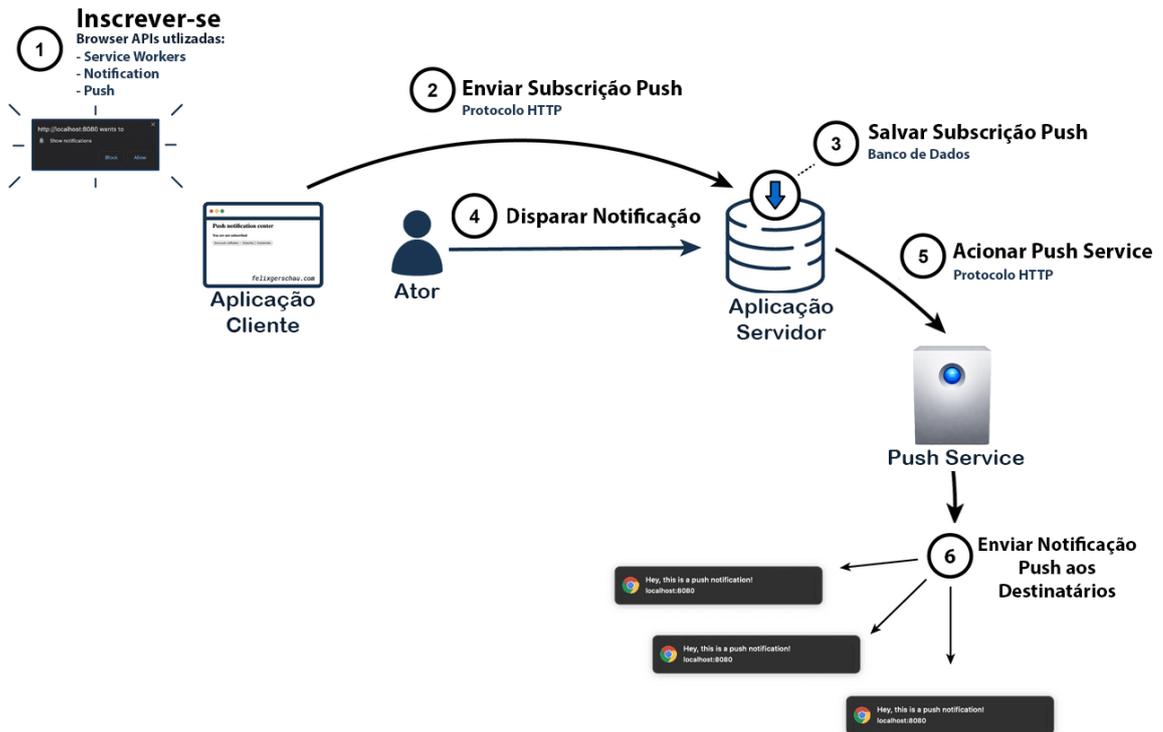
- **Notifications**: Permite que páginas *WEB* mostrem notificações do sistema para o usuário final. Por estar vinculado a um contexto mais geral do navegador, essas notificações podem ser exibidas mesmo que o usuário tenha mudado de aba ou até mesmo quando tenha mudado para um outro aplicativo (MDN 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. **Inscrever-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. **Accionar *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 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 OneSignal⁴ 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

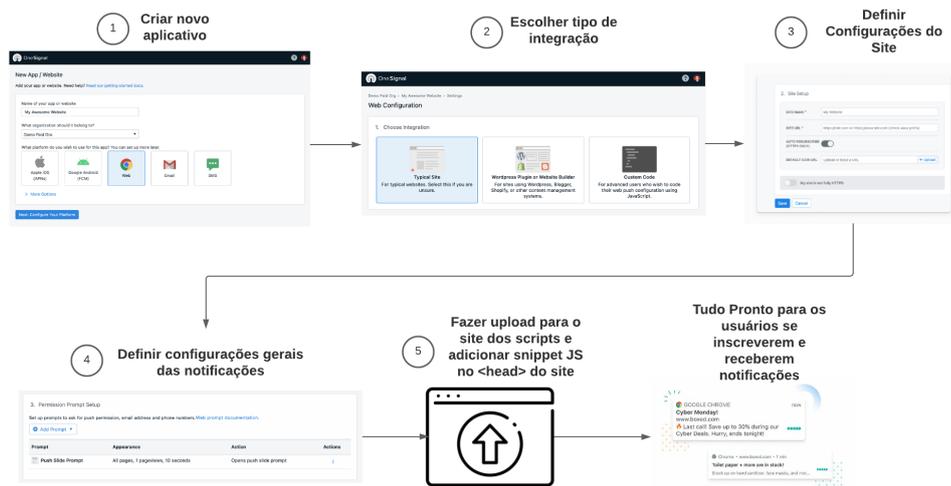
⁴ <https://onesignal.com/>

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. Em seguida, se faz necessário a configuração de algumas preferências relacionadas às 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*

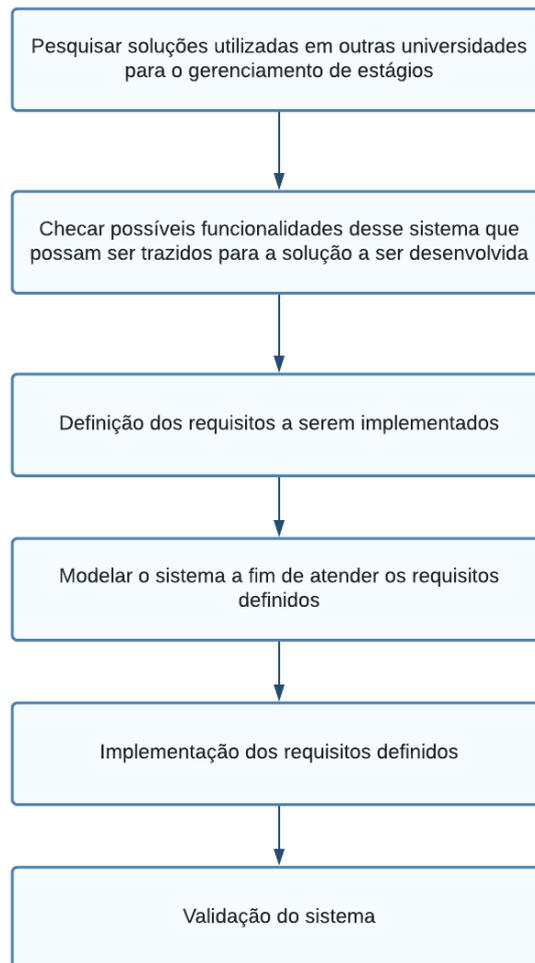


Fonte: Adaptado de OneSignal (2021)

4 PROCEDIMENTOS METODOLÓGICOS

Nesta seção serão apresentadas as etapas necessárias 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 como essas aplicações funcionam e quais tipos de funcionalidades elas tem.

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 poderiam ser acrescentadas à 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 através da análise das etapas do processo de solicitação e acompanhamento de estágio em empresas conveniadas da UFC Quixadá.

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, diagrama de atividades 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 à validação do sistema desenvolvido. Será utilizado a ferramenta postman para testar a *API back-end* da aplicação e testes manuais na aplicação realizados pelo próprio autor do trabalho para garantir que as funcionalidades desenvolvidas

estejam funcionando. Ao fim do desenvolvimento de cada módulo será realizado um teste de regressão, também manual, em todo o sistema.

5 RESULTADOS

Nesta seção serão apresentados os resultados da execução das etapas descritas na seção de procedimentos metodológicos.

5.1 Pesquisar soluções utilizadas em outras universidades para o gerenciamento de estágios

Foram pesquisadas quais soluções são utilizadas em algumas IES para gerenciamento de estágios. As características aqui descritas de cada um desses sistemas foram retiradas diretamente do manual de uso desses sistemas, que são disponibilizados em seus respectivos sites institucionais.

A Universidade Federal do Piauí (UFPI) utiliza o Módulo de Estágios do SIGAA, com o objetivo principal de atender às demandas de estágio obrigatório dos cursos de graduação e da Coordenadoria Geral de Estágios (CGE). Com a utilização desse módulo, é possível, dentre outras coisas: Garantir a padronização das atividades de estágio; Automatizar a geração de documentos dos convênios; Integração dos dados dos alunos na plataforma SIG; Gerar dados quantitativos de forma automática e distribuir as atividades de acompanhamento do estágio. A Figura 15 mostra alguns papéis e funcionalidades desse módulo, de acordo com o manual de uso do mesmo fornecido pela Superintendência de Tecnologia da Informação (STI) da instituição.

Figura 15 – Módulo de estágios do SIGAA - UFPI



Fonte: Elaborado pelo autor.

A Universidade Federal do Sul da Bahia, UFSB, também faz uso do Módulo de Estágios do sistema SIGAA. No módulo da UFSB o orientador tem a possibilidade de cadastrar um modelo de relatório final a ser preenchido pelos alunos, bem como os alunos podem preencher e submeter tal relatório diretamente pelo sistema.

A Universidade Federal de Lavras, UFLA, utiliza um módulo de estágio dentro de um sistema de gerenciamento das informações da instituição chamado SIG. Atores: Aluno, Orientador de estágio, Coordenador de estágio, PROEC (Pró-Reitoria de Extensão e Cultura). As funcionalidades são bem similares as que o sistema utilizado pela UFPI possui.

5.2 Checar possíveis funcionalidades desse sistema que possam ser trazidos para a solução a ser desenvolvida

Após análise dos sistemas descritos anteriormente, foi decidido implementar no sistema sendo desenvolvido nesse trabalho a funcionalidade de o aluno poder preencher e gerar o TCE diretamente no sistema.

Dessa forma, os alunos que estiverem realizando um estágio convencional (que não possua um contrato de Trabalho CLT ou PJ), pode preencher e posteriormente gerar o TCE no modelo disponibilizado pela UFC. O Aluno por sua vez, precisará somente imprimir o documento gerado, realizar as assinaturas necessárias e enviar posteriormente para aprovação da coordenação de estágio do seu curso.

5.3 Definição dos requisitos a serem implementados

5.3.1 Realizar Login no sistema utilizando credenciais da Base Centralizada

A UFC (Campus Quixadá) possui um sistema unificado de cadastros de informações de alunos e servidores do campus, denominado Base Centralizada ¹. De posse do login e senha cadastrado ali, o aluno é capaz de acessar diferentes sistemas internos da instituição, como por exemplo, SIPPA, Moodle, Contest e Sistema de gerenciamento de processos do auxílio moradia.

Como todos os alunos da instituição já possuem cadastro nessa base, será implementada uma funcionalidade que possibilite aos alunos acessarem o sistema desenvolvido nesse trabalho com essas mesmas credenciais, sem a necessidade de criação de uma conta para acesso específico a este sistema. Dessa forma, o usuário terá maior praticidade para realizar tal acesso.

¹ <https://identidadepessoa.quixada.ufc.br/>

Utilizar as credenciais dessa base trará também maior segurança e confiabilidade ao sistema, uma vez que como o gerenciamento das contas é realizada pela própria instituição, os usuários que realizarem login com essas credenciais necessariamente são vinculados à UFC, removendo assim a necessidade de o sistema de estágio validar se o usuário que está se cadastrando/logando no sistema de fato é uma pessoa vinculada à instituição.

5.3.2 Cadastrar Orientadores de Estágio dos Cursos

Todo curso possui necessariamente um orientador responsável pelos estágios. São os orientadores os responsáveis por analisar e aprovar, ou não, todas as documentações relacionadas ao estágio dos alunos daquele curso.

O sistema possuirá uma funcionalidade onde um usuário administrador, que já é criado por padrão no sistema, possa cadastrar novos orientadores de estágios, assinalando estes para seus respectivos cursos no qual serão responsáveis.

Após o cadastro, o orientador poderá editar as informações do seu próprio perfil. O administrador poderá também remover e editar as informações do orientador.

5.3.3 Visualizar Empresas Conveniadas para realização de estágio

Para que os alunos possam realizar estágio em uma determinada empresa, esta deverá obrigatoriamente possuir um convênio ativo juntamente à agência de estágios da UFC.

Com a finalidade de que o aluno possa saber se pode estagiar em uma determinada empresa, ou se será necessária uma formalização dessa empresa juntamente à UFC, o sistema deverá fornecer uma funcionalidade onde possa ser possível verificar as empresas que possuem convênios com a universidade.

5.3.4 Cadastrar e Gerar de Termo de Compromisso de Estágio

Os alunos que estiverem realizando um estágio convencional (que não é através de um contrato de trabalho CLT ou PJ), devem obrigatoriamente preencher e enviar à coordenação de estágios um TCE.

A UFC disponibiliza um modelo² de TCE para que os alunos possam fazer *download* e preencher.

² <https://estagios.ufc.br/wp-content/uploads/2021/11/termo-de-compromisso-de-estagio-obrigatorio.odt>

De modo a ter o processo o mais centralizado possível dentro do sistema, sem que o aluno tenha necessidade de ficar acessando diferentes recursos, o sistema deverá fornecer uma funcionalidade onde o aluno possa preencher todos os dados presentes no TCE e posteriormente gerar o documento, criado a partir do modelo fornecido pela universidade, já preenchido com as informações fornecidas pelo aluno.

5.3.5 Solicitação e Acompanhamento do Processo de Aproveitamento de Estágio-Empresa

O processo para cumprir a atividade obrigatória de estágio supervisionado da UFC consiste em 6 etapas:

1. **Solicitação de Matrícula no SIGAA;**
2. **Envio da documentação de estágio; (TCE ou Contrato de Trabalho (PJ ou CLT));**
3. **Envio do plano de estágio;**
4. **Envio da avaliação de desempenho do estagiário;**
5. **Envio do relatório Final de Estágio;**
6. **Apresentação no seminário de estágio.**

O sistema possuirá uma funcionalidade que permita ao aluno o envio dos documentos para aprovação do orientador de seu curso para cada etapa do processo.

Para o papel de orientador, o sistema possuirá uma funcionalidade que permita a visualização e aprovação desses documentos.

O sistema possuirá também um sistema de **Web Push Notifications** e **In-App Notifications** (Listagem de notificações dentro do próprio aplicativo) onde a parte interessada receberá notificações sempre que um novo evento ocorrer dentro de um processo como, por exemplo, o aluno receberá uma notificação sempre que o orientador aprovar uma documentação do seu processo. De forma similar, o orientador deverá receber uma notificação sempre que um aluno de um curso, na qual ele é responsável, enviar uma documentação para sua análise.

O módulo de processos do sistema também deverá possuir uma funcionalidade onde possa ser viável a troca de mensagens entre orientador e aluno, caso algo dentro da documentação necessite ser ajustado. Assim como na atualização de processos, a parte interessada também deverá receber uma **Web Push Notification** e uma **In-App Notification** sempre que uma nova mensagem for recebida.

5.3.6 *Apresentar aos alunos informações relevantes sobre o processo de estágio*

O sistema deverá apresentar aos alunos informações que possam ser úteis sobre o processo de estágio, tais como uma breve descrição sobre cada etapa necessária no processo e um cronograma detalhado dessas etapas.

5.4 Modelar o sistema a fim de atender os requisitos definidos

5.4.1 *Realizar Login no Sistema*

O sistema possui 3 tipos de usuários: **Aluno**, **Orientador(Coordenador)** e **Administrador**.

O usuário administrador será criado na inicialização do sistema. As credenciais desse usuário poderão ser definidas através das variáveis de ambientes **SUPER_ADMIN_LOGIN** e **SUPER_ADMIN_PASSWORD**. Caso não exista um usuário no banco de dados cujo login é aquele que está presente na variável de ambiente, **SUPER_ADMIN_LOGIN**, e a variável de ambiente **SUPER_ADMIN_PASSWORD** esteja preenchida, então será inserido um usuário no banco de dados com permissão de administrador com essas credenciais.

O usuário administrador será o responsável por criar os usuários do tipo orientador. Tanto o login dos usuários administradores, como de orientadores, será realizado através da verificação das credenciais informadas pelo usuário com aquelas informações cadastradas no banco de dados.

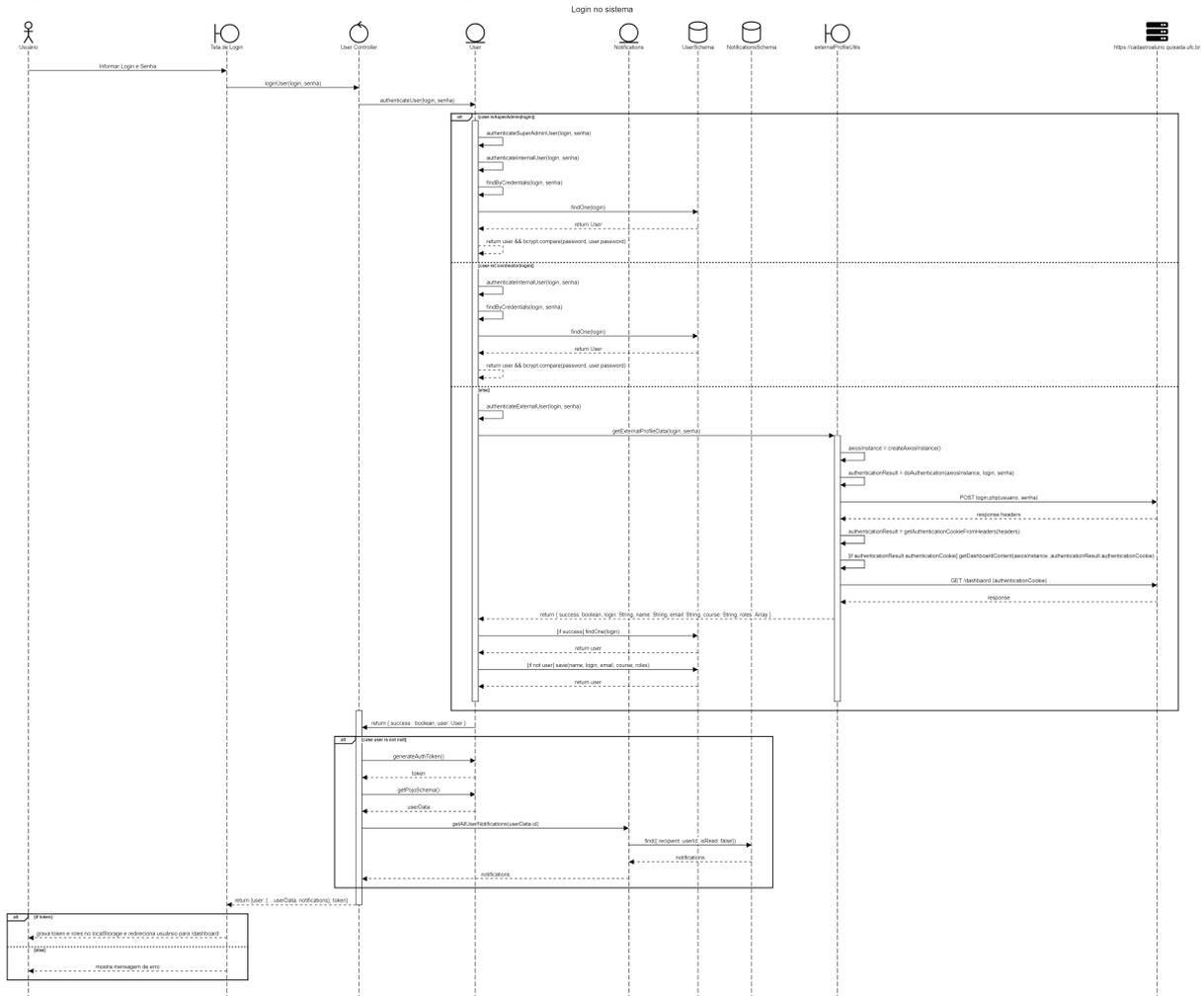
O usuário do tipo estudante, diferente dos outros dois tipos de usuários, não precisa ter um cadastro prévio no sistema. Será realizado uma consulta no sistema Base Centralizada de estudantes da UFC Quixadá para checar se o *login* e senha informados são válidos dentro daquele sistema. Para realização de tal consulta será desenvolvido um *middleware* no *back-end* que possa fazer uma requisição *HTTP* para o sistema Base Centralizada. A partir da resposta dessa consulta será possível obter as informações sobre o aluno, tais como nome e curso.

Quando o aluno realizar *login* pela primeira vez no sistema, será salvo no banco de dados as seguintes informações retornadas pelo sistema base centralizada: Nome, login, email e curso. Isso é necessário para que seja possível ter uma referência daquele aluno dentro do sistema. Nenhuma senha será salva. A cada novo login do aluno no sistema de estágios, a base centralizada sempre será consultada para garantir que as credenciais estão corretas e que o aluno

possui vínculo com a universidade.

A Figura 16 mostra como foi modelada a funcionalidade de Login no sistema.

Figura 16 – Diagrama de sequência para funcionalidade de login no sistema



Fonte: Elaborado pelo autor.

5.4.2 Cadastrar Orientadores de Estágio dos Cursos

Conforme descrito na subseção anterior, o usuário administrador será o responsável por criar no sistema os orientadores de estágio.

Um módulo denominado “Coordenadores” somente estará disponível para usuários administradores. Quando registrando um novo orientador de estágio, o administrador terá que informar os seguintes dados: Nome, login, email e cursos cujo aquele orientador é responsável.

5.4.3 Visualizar Empresas Conveniadas para realização de estágio

A UFC possui um módulo público do SIGAA onde é possível realizar consultas as instituições e os profissionais já conveniados, bem como acompanhar os convênios em tramitação ou renovação³.

Como descrito em uma subseção anterior, de modo a evitar que os usuários (alunos e orientadores) tenham que ficar acessando diferentes recursos, o sistema deverá fornecer uma funcionalidade em que seja possível a realização de consultas para aquela página pública do SIGAA sem necessidade de sair do sistema de estágios.

5.4.4 Cadastrar e Gerar de Termo de Compromisso de Estágio

Os alunos na etapa de documentação ao escolher o tipo de estágio como convencional, deverá ser capaz de preencher e gerar o TCE para posterior impressão, assinaturas das pessoas necessárias e envio para análise da coordenação de estágio do seu curso.

Ao iniciar o cadastro de um TCE, o sistema já deverá trazer preenchido as informações que ele conhece sobre o aluno: Nome, CPF, Curso, *E-mail*. Também deve trazer preenchido as informações sobre o orientador de estágio, uma vez que o curso do aluno já é conhecido, dessa forma o nome do orientador e o número do Siape deverão estar preenchidos.

Tal informação sobre o TCE do aluno deverá ser salvo dentro da coleção *tces* do banco de dados. A Figura 20 ilustra a estrutura dessa coleção.

5.4.5 Solicitação e Acompanhamento do Processo de Aproveitamento de Estágio-Empresa

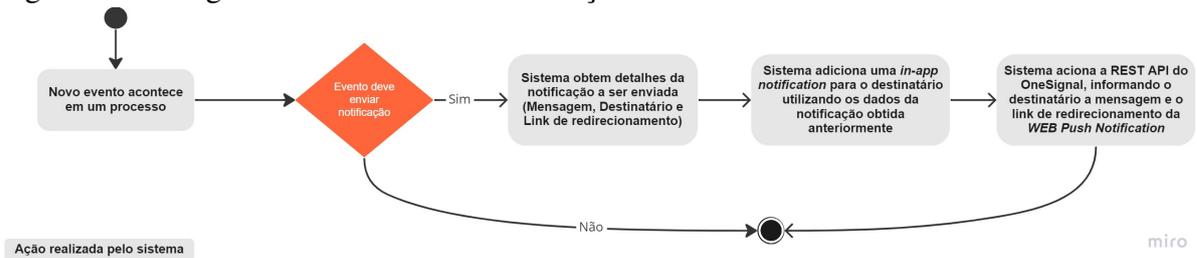
Conforme ilustrado na Figura 20, uma coleção no banco de dados chamada *processes* será utilizada para armazenar todos os processos criados no sistema (incluindo as mensagens enviadas entre orientadores e alunos).

O processo só poderá ser visualizado e atualizado pelas partes interessadas nele (aluno e o orientador de estágio). O sistema deverá ter uma funcionalidade que garanta esse controle de acesso, assim como da documentação anexada a ele. Somente o orientador pode aprovar uma documentação, assim como somente o aluno pode anexar.

Com relação à funcionalidade de notificações que o sistema deverá possuir, a Figura 17 ilustra como o sistema deverá lidar com essa funcionalidade,

³ <https://si3.ufc.br/sigaa/public/estagio/lista.jsf/>

Figura 17 – Diagrama de atividades - notificações do sistema



Fonte: Elaborado pelo autor.

O Quadro 2 mostra todas as notificações que o sistema deverá enviar às partes interessadas quando acontecer um determinado evento dentro de um processo. Todas as notificações deverão ser enviadas em forma de **WEB Push Notifications** e de **In-App Notifications**.

Quadro 2 – Quadro de notificações as partes interessadas de atualizações no processo

Evento	Etapa	Notificação	Quem notificar
Aluno confirma que realizou matrícula no SIGAA	Matrícula SIGAA	Nova confirmação de matrícula no SIGAA.	Orientador
Aluno envia TCE para análise do orientador	Documentação	Novo TCE enviado para sua análise.	Orientador
Aluno envia Contrato de Trabalho (CLT ou PJ) para análise do orientador	Documentação	Novo contrato de trabalho enviado para sua análise.	Orientador
Orientador aprova TCE enviado para análise	Documentação	Seu TCE foi aprovado pela coordenação de estágio.	Aluno
Orientador aprova Contrato de Trabalho (CLT ou PJ) enviado para análise	Documentação	Seu contrato de trabalho foi aprovado pela coordenação de estágio.	Aluno
Aluno envia Plano de Estágio para análise do orientador	Plano de Estágio	Novo plano de estágio enviado para sua análise.	Orientador
Orientador aprova Plano de Estágio enviado para análise	Plano de Estágio	Seu plano de estágio foi aprovado pela coordenação de estágio.	Aluno
Aluno envia relatório de Avaliação de Desempenho para análise do orientador	Avaliação de Desempenho	Novo relatório de avaliação de desempenho enviado para sua análise.	Orientador
Orientador aprova relatório de Avaliação de Desempenho enviado para análise	Avaliação de Desempenho	Seu relatório de avaliação de desempenho foi aprovado pela coordenação de estágio.	Aluno
Aluno envia Relatório Final de Estágio para análise do orientador	Relatório Final	Novo relatório final de estágio enviado para sua análise.	Orientador
Orientador aprova Relatório Final de Estágio enviado para análise	Relatório Final	Seu relatório final de estágio foi aprovado pela coordenação de estágio. Agora falta somente a apresentação do seminário de estágio, dia 12/12/2022. Lembre-se que essa apresentação é obrigatória.	Aluno
Orientador confirma que o seminário de estágio foi apresentado pelo aluno	Seminário	Parabéns, todas as etapas do seu processo foram concluídas.	Aluno
Orientador envia mensagem dentro de um processo para um aluno	Todas	Nova mensagem recebida no processo de estágio. Clique aqui para ler.	Aluno
Aluno envia mensagens dentro de um processo para o orientador de estágio	Todas	Nova mensagem recebida no processo de estágio. Clique aqui para ler.	Orientador

Fonte: Elaborado pelo autor.

5.4.6 Apresentar aos alunos informações relevantes sobre o processo de estágio

As informações gerais sobre estágio deverão aparecer logo na primeira página após a realização do login por parte do aluno. O fluxo geral do estágio na UFC Quixadá pode ser encontrado no site oficial do campus⁴ e na página específica sobre informações de estágio do campus⁵.

Para uma melhor organização dentro do projeto, essas informações deverão ser armazenadas em forma de documentos, em formato *Markdown*, uma linguagem de marcação leve das mais populares do mundo, e que pode ser usada para adicionar elementos de formatação a documentos de texto simples (MARKDOWN, 2022).

5.5 Implementação dos requisitos definidos

Para a implementação do sistema foi utilizado a arquitetura *MEVN*, que já foi detalhada na seção 3.3.1. Dentro dessa arquitetura foi necessário a utilização de bibliotecas complementares no lado cliente e no lado servidor.

No lado cliente, para agilizar o desenvolvimento, foi utilizado a biblioteca **Vuetify**⁶, um framework Vue.js com vários componentes inspirados no padrão Material Design já prontos para uso. Também fora utilizado **Vuex** para gerenciamento de estado das informações no lado do cliente e também a biblioteca **vuex pathify**, bastante útil para facilitar acesso a informações dos objetos gerenciados pelo vuex.

No lado servidor foi utilizada a biblioteca JavaScript **mongoose** para facilitar a modelagem e acesso ao banco de dados MongoDB da aplicação.

A biblioteca JavaScript **Moment.js** foi utilizada em ambos os lados para facilitar a manipulação de datas no sistema.

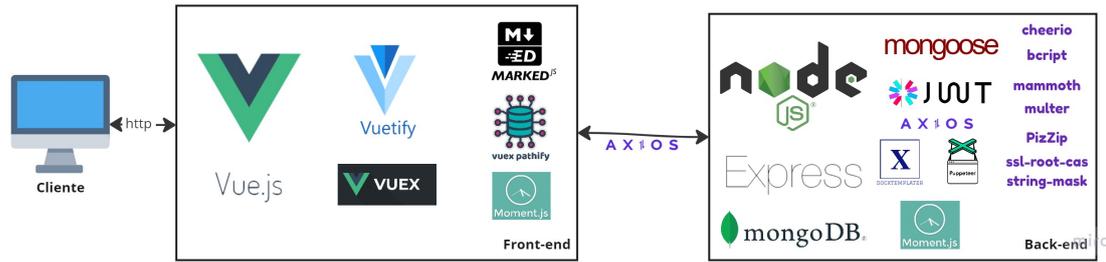
A Figura 18 ilustra todas essas bibliotecas utilizadas.

⁴ <https://www.quixada.ufc.br/estagio/>

⁵ <https://bit.ly/estagio-ufc-quixada>

⁶ <https://vuetifyjs.com/>

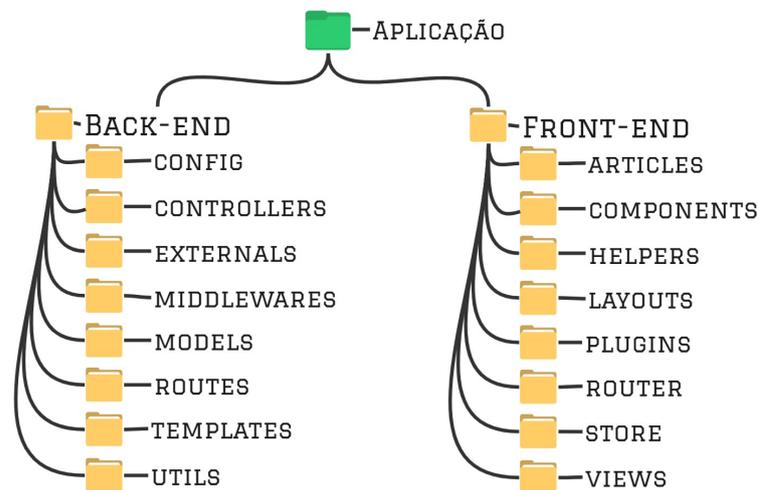
Figura 18 – Bibliotecas utilizadas no desenvolvimento do sistema



Fonte: Elaborado pelo autor.

A Figura 19 mostra como ficou a organização do projeto após a implementação. O código-fonte completo da aplicação encontra-se disponível em um repositório público no GitHub⁷.

Figura 19 – Organização do projeto

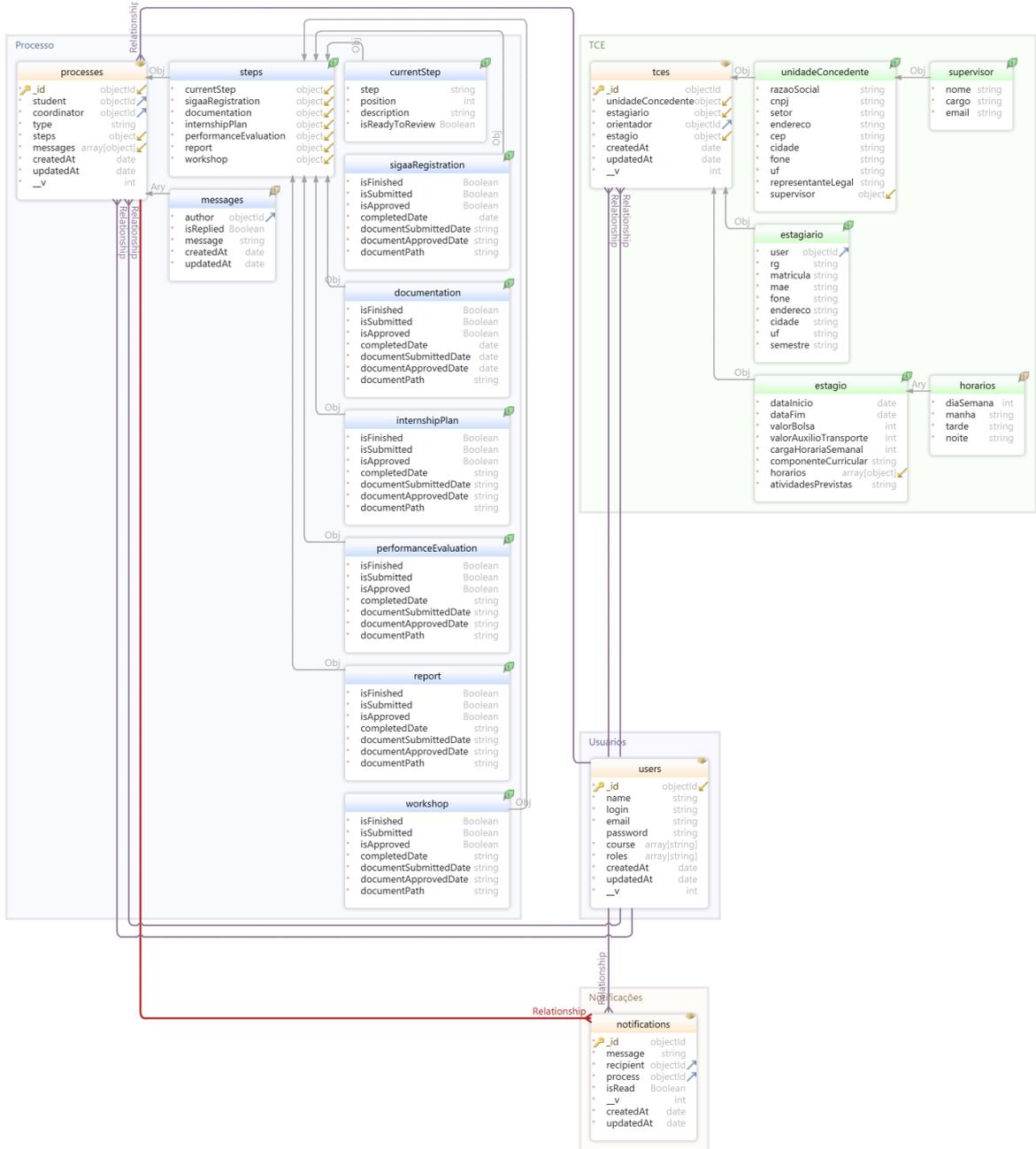


Fonte: Elaborado pelo autor.

A Figura 20 mostra como ficou a modelagem de banco de dados da aplicação.

⁷ <https://github.com/ibituruna/Sistema-Estagio-Quixada>

Figura 20 – Modelagem do banco de dados



Fonte: Elaborado pelo autor.

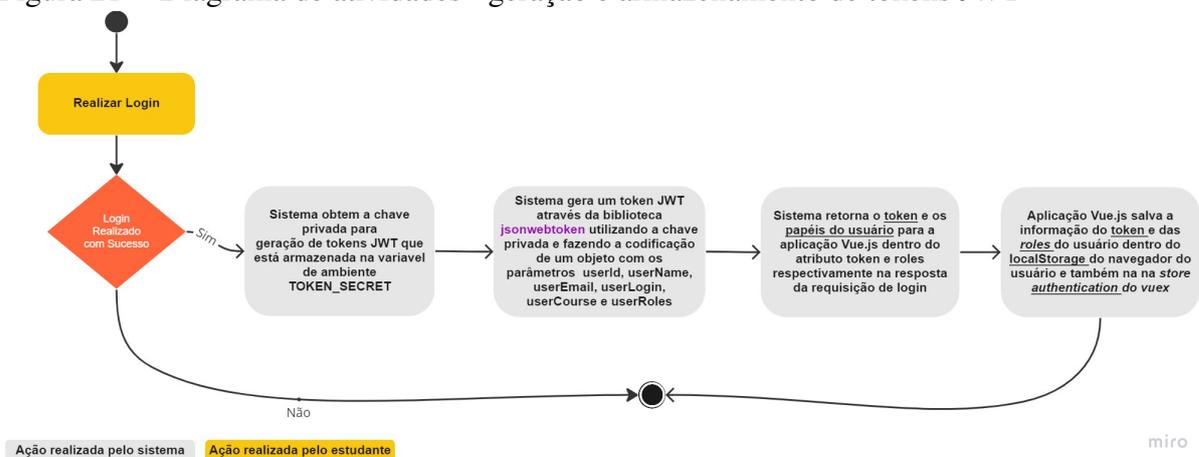
5.5.1 Realizar Login no sistema

Para o módulo de Login e controle de acesso ao sistema, além do que já foi descrito na subseção 5.4, também foi usado o padrão de autenticação entre cliente e servidor utilizado JSON Web Token (JWT). Após o usuário se autenticar usando suas credenciais, o servidor retorna um token de acesso JWT para o cliente que envia essa informação em todas as requisições subsequentes. Foi criado um módulo utilitário no *back-end* chamado *tokenUtils.js* para concentrar todas as operações de geração e decodificação de tokens JWT na aplicação.

Foi utilizado o conceito de *interceptors* da biblioteca JavaScript **axios** (responsável por realizar as requisições entre cliente e servidor na aplicação) para que antes de qualquer requisição seja lido o token salvo no *localStorage* do navegador e adicionado essa informação dentro do cabeçalho *authorization*.

A Figura 21 ilustra como é o fluxo para geração e armazenamento desse token após a autenticação do usuário.

Figura 21 – Diagrama de atividades - geração e armazenamento de tokens JWT



Fonte: Elaborado pelo autor.

Foram implementados alguns middlewares no *back-end* para controle de acesso, de modo a facilitar a proteção de *endpoints* de acesso não autorizado.

Caso o usuário não atenda a todos os pré-requisitos definidos nos *middlewares* que foram adicionados ao *endpoint* que está tentando ser acessado, o servidor retorna um código de resposta **401**, que significa que o usuário não possui autorização para acessar tal recurso, garantindo assim o controle de acesso aos recursos do sistema conforme necessário.

O Quadro 3 descreve todos esses *middlewares* bem como suas aplicabilidades.

Quadro 3 – *Middlewares* de controle de acesso a recursos no *back-end*

Middleware	Descrição
userLoggedIn.validateLoggedIn	Responsável por ler o Token JWT de autorização enviado nos cabeçalhos das requisições, decodificá-lo utilizando a chave privada e então criar um atributo chamado <i>userDecoded</i> dentro do contexto do objeto <i>request</i> do <i>express.js</i> contendo os valores de <i>userId</i> , <i>userName</i> , <i>userEmail</i> , <i>userLogin</i> , <i>userCourse</i> e <i>userRoles</i> . É realizado a inclusão desse <i>middleware</i> em todos os <i>endpoints</i> que não são públicos e que requerem que o usuário esteja autenticado no sistema.
userLoggedIn.isStudent	Responsável por validar se o usuário logado possui o papel de estudante no sistema. É realizado a inclusão desse <i>middleware</i> em todos os <i>endpoints</i> que somente podem ser acessados por alunos.
userLoggedIn.isSuperAdmin	Responsável por validar se o usuário logado possui o papel de administrador no sistema. É realizado a inclusão desse <i>middleware</i> em todos os <i>endpoints</i> que somente podem ser acessados por administradores.
userLoggedIn.isCoordinator	Responsável por validar se o usuário logado possui o papel de orientador no sistema. É realizado a inclusão desse <i>middleware</i> em todos os <i>endpoints</i> que somente podem ser acessados por orientadores.
userLoggedIn.isCoordinatorOrSuperAdmin	Responsável por validar se o usuário logado possui o papel de orientador ou de administrador no sistema. É realizado a inclusão desse <i>middleware</i> em todos os <i>endpoints</i> que somente podem ser acessados por orientadores e administradores.
userLoggedIn.validateOwnResourceAccess	Responsável por validar se o usuário está tentando modificar um recurso que ele realmente pode ler / modificar. É realizado a inclusão desse <i>middleware</i> em todos os <i>endpoints</i> onde é necessário verificar se o usuário é uma parte interessada no recurso que está tentando ser lido / modificado.

Fonte: Elaborado pelo autor.

Para a integração do sistema com a base centralizada da UFC Quixadá foi desenvolvido um *middleware* utilizando a biblioteca JavaScript **axios** no *back-end*. Além do que já fora ilustrado na Figura 16, foi necessário contornar alguns problemas encontrados durante o desenvolvimento.

O primeiro deles foi o fato de o servidor não conseguir reconhecer o certificado SSL do domínio ‘.quixada.ufc.br’ como confiável quando realizado requisições a esse domínio utilizando o **axios**. Para contornar esse problema, foi necessário utilizar a ferramenta **whatsmychaincert** para obter o certificado completo daquele domínio e posteriormente injetar no **axios** esse certificado como confiável. Para realizar essa injeção, foi necessário utilizado a biblioteca JavaScript **ssl-root-cas**⁸.

O próximo problema encontrado foi para obter a resposta se o usuário havia se autenticado com sucesso ou não, na base centralizada com as credenciais informadas. Foi observado que quando o usuário não havia se autenticado com sucesso, o servidor da base

⁸ <https://www.npmjs.com/package/ssl-root-cas>

centralizada retornava um código de resposta 302 redirecionando o usuário para a página **‘/acessonegado’** e quando o usuário se autenticava com sucesso era retornado um *HTTP response header* **‘set-cookie’**, que é o padrão utilizado pelo navegador para salvar automaticamente um cookie no navegador do usuário para ser enviado em posteriores requisições aquele domínio (MDN Web Docs, 2022), e posteriormente o usuário era redirecionado para a página **‘/dashboard’**, que continha as informações do usuário. Quando era realizado requisições no servidor para o *endpoint* **‘/login.php’** da base centralizada, o axios sempre obtinha como resposta o conteúdo do formulário de login inicial do usuário informando que o usuário precisava informar suas credenciais.

Foi observado então que o comportamento que acontecia no navegador (e no Postman) não ocorria no axios, que era o de o redirecionamento para a página **‘acessonegado’** ou para a página **‘/dashboard’**. Isso acontecia devido:

1. **Axios não reconhecer a resposta 302 como uma resposta de sucesso:** Para contornar esse problema, foi realizada a modificação do parâmetro *validateStatus* para que o código 302 fosse interpretado como uma resposta de sucesso. Foi necessário também modificar o parâmetro de configuração do axios *maxRedirects* de modo que o axios não tentasse realizar o redirecionamento do usuário para um outro endpoint e fosse possível verificar o cabeçalho **‘location’** para identificar se o usuário deveria ser redirecionado para a página **acessonegado** ou **dashboard**.
2. **Axios não usar o cookie retornado no parâmetro ‘set-cookie’ nos próximos redirecionamentos dentro da mesma requisição:** Como já havia sido desabilitado a opção de o axios realizar o redirecionamento automático da requisição, para realizar uma correção para esse problema foi apenas necessário fazer a leitura do valor do *cookie* de autenticação retornado quando o usuário se autenticava com sucesso. Com posse dessa informação, foi necessário fazer uma segunda requisição, agora para o *endpoint* **‘/dashboard’**, realizando a inserção desse *cookie* manualmente na requisição.

Por fim, o último problema encontrado foi o de como realizar a leitura das informações do usuário quando esse tivesse autenticado com sucesso. Como mencionado acima, quando isso acontece o usuário é redirecionado para a página **/dashboard**, o que significa que o **axios** tem como resposta o *HTML* completo daquela página. Para resolver esse problema, foi utilizada a biblioteca JavaScript **cheerio**⁹. Com essa biblioteca é possível fazer a leitura de

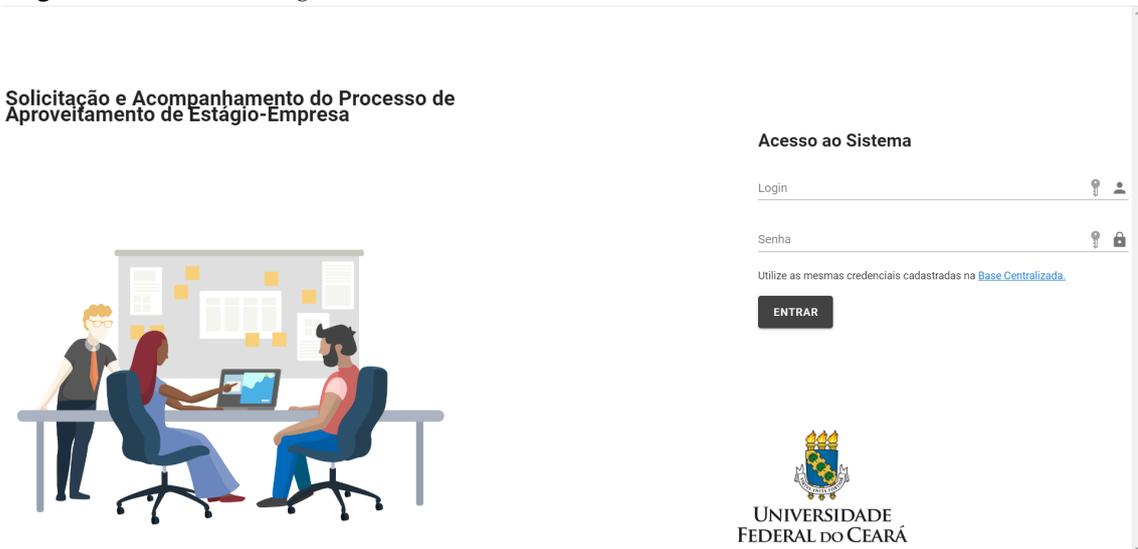
⁹ <https://cheerio.js.org/>

qualquer valor dentro de um *HTML* a partir do *parse* de sua árvore. O acesso à árvore *HTML* é realizado inspirado na biblioteca jQuery¹⁰. Dessa forma só foi necessário mapear em que elementos *HTML* as informações do usuário estavam e realizar a obtenção delas utilizando a biblioteca cheerio.

Para os tipos de usuários no qual o acesso não é realizado através da base centralizada e a senha é salva dentro do próprio banco de dados do sistema, como é o caso de administradores e orientadores, foi utilizado a biblioteca JavaScript **bcryptjs**¹¹ para que seja realizada a criptografia da senha do usuário antes de ser salva no banco de dados. Dessa forma, no banco de dados é armazenado apenas o *hash* dessa senha.

A Figura 22 mostra a tela de login do sistema após a implementação.

Figura 22 – Tela de *login* do sistema



Fonte: Elaborado pelo autor.

5.5.2 Cadastrar Orientadores de Estágio dos Cursos

A Figura 23 mostra o módulo de gerenciamento (cadastro, edição e deleção) dos orientadores de estágio no sistema após a implementação.

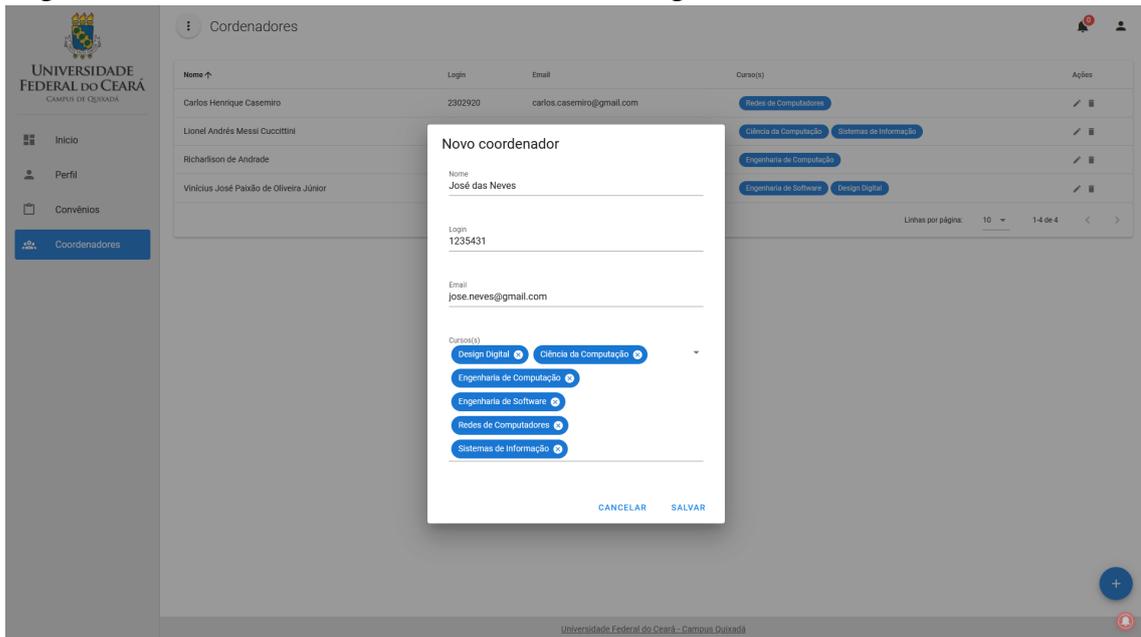
Conforme descrito na subseção 5.4.2, esse módulo somente está disponível para usuários administradores.

Após o cadastro, através do módulo 'Perfil', cada orientador poderá editar suas próprias informações. A Figura 24 mostra esse módulo 'Perfil' do orientador.

¹⁰ <https://jquery.com/>

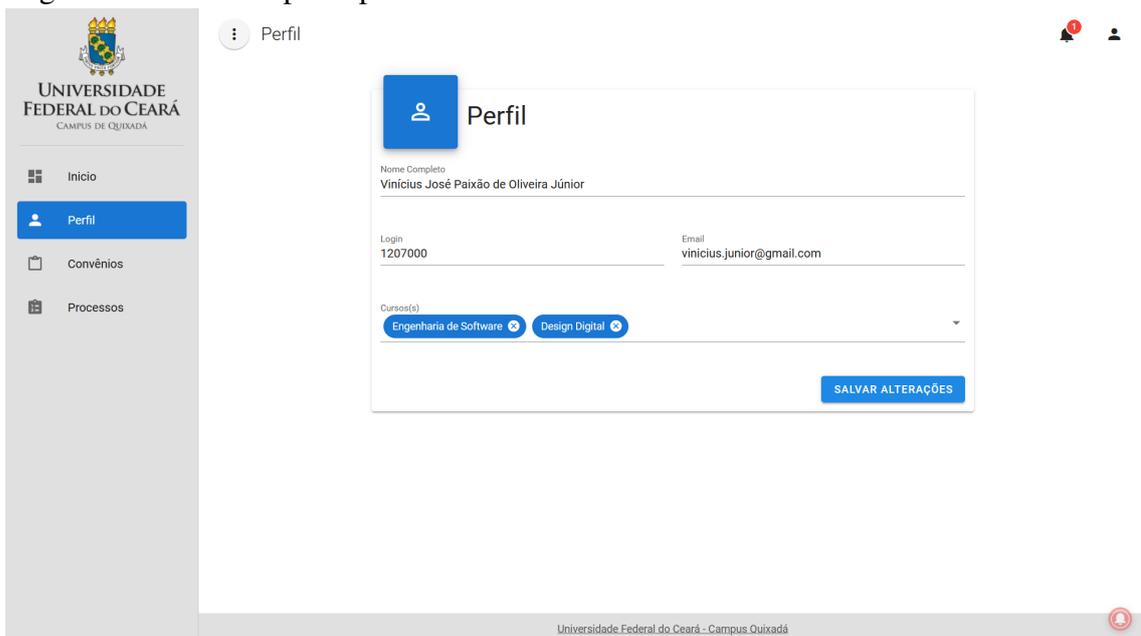
¹¹ <https://www.npmjs.com/package/bcryptjs>

Figura 23 – Gerenciamento de orientadores de estágio



Fonte: Elaborado pelo autor.

Figura 24 – Módulo perfil para orientadores



Fonte: Elaborado pelo autor.

5.5.3 Visualizar Empresas Conveniadas para realização de estágio

Para que fosse possível a pesquisa de empresas conveniadas utilizando o módulo público do SIGAA, foi utilizado um elemento *HTML IFrame*, uma vez que o intuito desse elemento é exatamente tornar possível a renderização de uma página dentro de outra página.

A Figura 25 mostra a tela de visualização e consulta de empresas conveniadas para a realização de estágio.

Figura 25 – Consultas de empresas conveniadas



Fonte: Elaborado pelo autor.

5.5.4 Cadastrar e Gerar de Termo de Compromisso de Estágio

Para a implementação da funcionalidade de geração de TCE foi utilizado um conjunto de diferentes bibliotecas JavaScript: **docxtemplater**¹², **PizZip**¹³, **mammoth**¹⁴ e **puppeteer**¹⁵.

A necessidade do uso dessas diferentes bibliotecas para a realização dessa tarefa se deu pelo fato da ausência de bibliotecas *JavaScript* gratuitas que fossem capazes de gerar um arquivo PDF a partir de um *template* já pronto em formato .odt ou .docx, como é o caso do modelo de TCE disponibilizado pela UFC.

Com a biblioteca **docxtemplater** foi possível ter um arquivo .docx como *template*, contendo textos *placeholders* que posteriormente serão substituídos por valores informados pelo estudante quando cadastradas as informações do TCE. Uma limitação dessa biblioteca é que ela somente é capaz de gerar um arquivo .docx como resultado, não provendo suporte para arquivos PDF. Essa biblioteca requer obrigatoriamente a realização da compactação para .zip do arquivo *template* .docx antes de ser usado pela biblioteca. Tal compactação é realizada através da biblioteca **PizZip**.

Como somente foram encontradas bibliotecas JavaScript gratuitas para geração de arquivos PDF a partir de *HTML*, como é o caso da biblioteca **puppeteer**, foi necessária a realização de um passo extra de pós processamento em cima do arquivo .docx gerado pelo

¹² <https://docxtemplater.com/>

¹³ <https://www.npmjs.com/package/pizzip/>

¹⁴ <https://www.npmjs.com/package/mammoth/>

¹⁵ <https://pptr.dev/>

docxtemplater para converter o conteúdo desse arquivo para o formato *HTML*.

Para isso, foi utilizada a biblioteca **mammoth** que é capaz de receber um arquivo .docx e transformar isso em *HTML*. Uma limitação, no entanto, dessa biblioteca, é que os estilos originais presentes no arquivo .docx são perdidos durante a conversão para *HTML* (como, por exemplo as grades das tabelas) e o cabeçalho das páginas não são convertidos corretamente, havendo perda de referência da posição em que estão.

E por fim foi utilizada a biblioteca **puppeteer**, que provê entre outras coisas, a possibilidade de geração de arquivos PDF a partir de conteúdo *HTML*. Com essa biblioteca foi possível contornar o problema de perda de estilo no documento causada pela biblioteca **mammoth**, uma vez que é possível informar estilos *CSS* como parâmetro nessa biblioteca. Dessa forma, foram criados os estilos *CSS* que fossem capazes de adicionar as formatações perdidas durante a conversão para *HTML*. Um outro problema que essa biblioteca foi capaz de resolver, foi a possibilidade de inserir um cabeçalho nas páginas PDF sendo gerados, uma vez que é possível passar como parâmetro um arquivo *HTML* que é renderizado antes da geração de cada página do PDF.

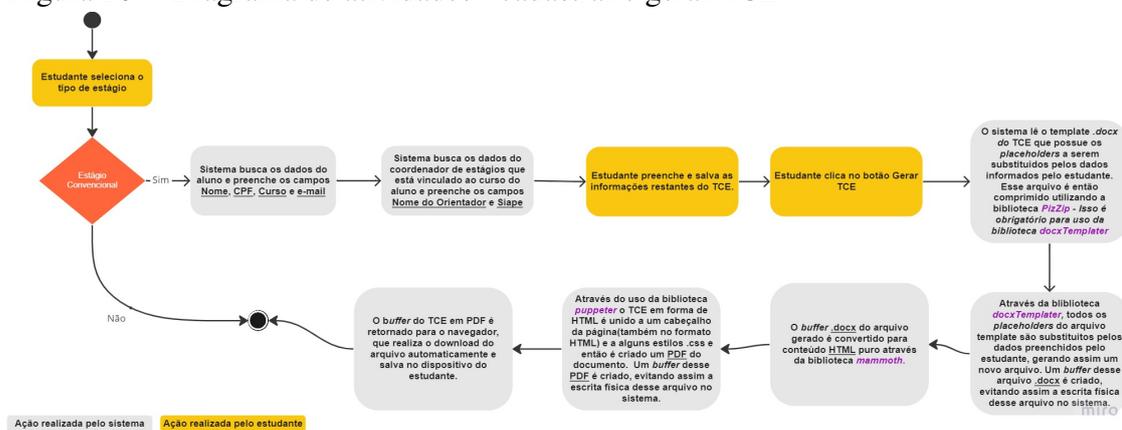
Para evitar uso desnecessário de armazenamento, nenhum arquivo é salvo fisicamente no servidor quando um TCE é gerado. Tudo é realizado através do uso de *buffers* que ao fim do fluxo de geração são liberados da memória.

O Apêndice A mostra um exemplo de TCE gerado pelo sistema.

O Apêndice B mostra o *template* utilizado para a geração de TCE no sistema. Ele foi criado a partir do modelo de TCE disponibilizado pela UFC de TCE gerado pelo sistema.

A Figura 26 ilustra o diagrama *UML* de atividades dessa funcionalidade.

Figura 26 – Diagrama de atividades - cadastrar e gerar TCE



Fonte: Elaborado pelo autor.

A Figura 27 mostra a tela de cadastro e geração de TCE após a sua implementação.

Figura 27 – Tela de cadastro e geração TCE do sistema



**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS DE QUIXADÁ

Processo

1 Matricula SIGAA
Até 22/08/2022

2 Documentação
Até 26/08/2022

1 Preenchimento e Geração do TCE **2** Envio do TCE para Análise da Coordenação **3** Aprovação da Coordenação




U

Unidade Concedente

A unidade concedente deve necessariamente ter convênio ativo com a agência de estágios da UFC.
Clique aqui para verificar a situação do convênio da empresa com a UFC.

Reação Social
Empresa Teste

CNPJ
23.424.242/4242-42

Fone
(23) 32323-2323

Sector
Informática

11 / 255

Endereço
Rua Teste

CEP
23.232-323

Cidade
Parnaíba

U.F.
PI

9 / 255

Representante Legal
Joaquim José da Silva Xavier

U.F.
PI

28 / 255

Nome do Supervisor
Saul Goodman

Cargo do Supervisor
Back-end developer

Email do Supervisor
saugoodman@gmail.com

21 / 100

U

Estagiário

Nome
BRUNO DE CARVALHO SOBRINHO

RG
54588454

CPF
123.456.789-00

26 / 255

Matrícula
111111

Curso
Engenharia de Software

Semestre
13º Semestre

12 / 50

Endereço
Rua Teste

Cidade
Parnaíba

U.F.
PI

9 / 255

Fone
(23) 32323-2323

Email
bruno.carvalho@gmail.com

Nome de Mãe
Jane Doe Carvalho

24 / 100

U

Orientador

Nome
Vinicius José Paixão de Oliveira Júnior

Slope
120700

39 / 255

U

Dados do Estágio

Data de Início
21/11/2022

Data de Fim
22/03/2023

Valor da Bolsa
250

Valor do Auxílio Transporte
250

Carga Horária Semanal
20 h

Componente Curricular
Estágio Supervisionado 2

24 / 100

Horário do estágio

Dia da Semana	Manhã	Tarde	Noite
Segunda-Feira	08:00 h as 12:00h	13:00 h as 17:00h	19:00 h as 22:00h
Terça-Feira	08:00 h as 12:01h	13:00 h as 17:01h	19:00 h as 22:01h
Quarta-Feira	08:00 h as 12:02h	13:00 h as 17:02h	19:00 h as 22:02h
Quinta-Feira	08:00 h as 12:03h	13:00 h as 17:03h	19:00 h as 22:03h
Sexta-Feira	08:00 h as 12:04h	13:00 h as 17:04h	19:00 h as 22:04h
Sábado	08:00 h as 12:05h	13:00 h as 17:05h	19:00 h as 22:04h

Atividades Previstas
Lorem Ipsum atividades previstas

VOLTAR
SALVAR
GERAR TCE
SUBMITER TCE

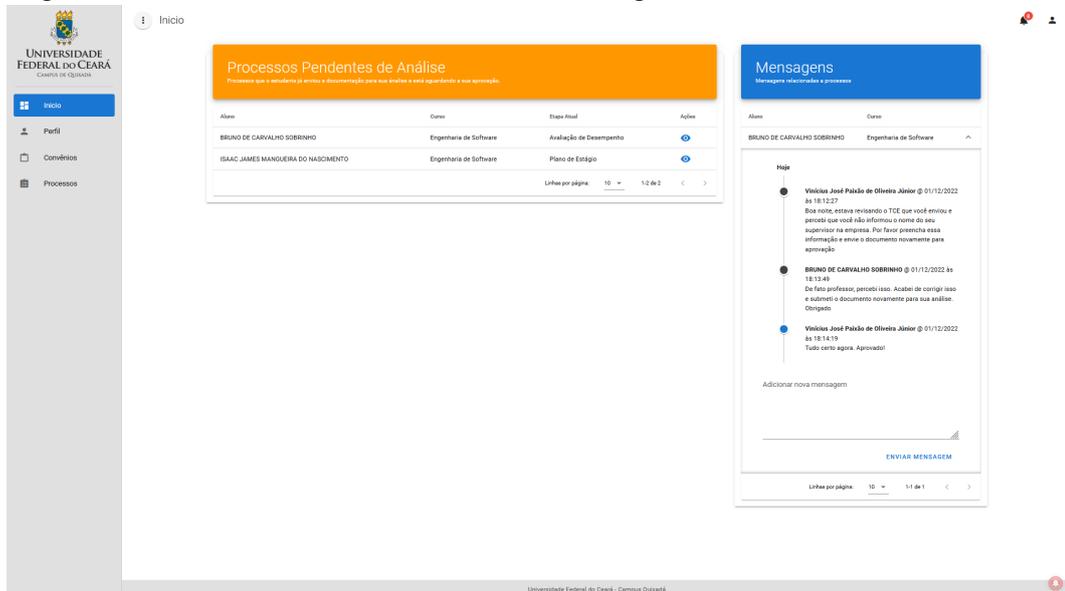
Universidade Federal do Ceará - Campus Quixadá

Fonte: Elaborado pelo autor.

5.5.5 Solicitação e Acompanhamento do Processo de Aproveitamento de Estágio-Empresa

Para a implementação dessa funcionalidade foram criados dois módulos distintos: Um para orientadores de estágio, onde esses são capazes de ter uma visão completa sobre todos os processos de estágio em andamento no seu curso, e um outro módulo para estudante, onde esse pode ter acesso a informações de todas as etapas necessárias até o fim do processo de estágio. A Figura 28 mostra como foi implementada a tela inicial para os orientadores de estágio, na qual foram adicionados dois componentes que mostram respectivamente os processos que estão esperando aprovação de documentação e as mensagens trocadas dentro de processos com alunos.

Figura 28 – Tela inicial dos orientadores de estágio



Fonte: Elaborado pelo autor.

O controle de acesso de qual o módulo vai ser exibido para o usuário, é realizado através de uma checagem de qual papel está assinalado para o usuário logado. Tal controle de acesso é realizado também no servidor da aplicação através de um *middleware* para garantir que somente um usuário logado e que possa ler / modificar aquele processo esteja o acessando, conforme já fora explicado anteriormente na subseção 5.5.1.

Para realizar o *upload* de arquivos da aplicação cliente para o servidor foi utilizado a biblioteca JavaScript **multer**¹⁶. Todos os arquivos são guardados dentro da pasta **uploads/documentation** do servidor e antes de ser salvo no sistema, é realizado a adição de um prefixo no nome do arquivo contendo o código identificador do usuário no sistema.

¹⁶ <https://www.npmjs.com/package/multer>

Para que os arquivos possam ser lidos pelo orientador posteriormente, a pasta **uploads/documentation** foi adicionada como uma pasta `express.js` estática do projeto. Dessa forma o conteúdo que há dentro dela pode ser acessado externamente no navegador. No campo *documentPath* do passo onde a documentação foi anexada, é salvo o caminho completo para que o arquivo possa ser acessado. Dessa forma, quando requisitada a leitura de uma documentação pelo orientador, é realizada uma requisição a *path* do documento e então renderizado o conteúdo dele no navegador do usuário, através da API do navegador *createObjectURL*.

Para ativar o **OneSignal** no sistema, primeiro foi necessário a criação de um aplicativo no dashboard do **OneSignal**. Posteriormente a isso, foi utilizado a biblioteca JavaScript **OneSignal-Vue**. Com o uso dessa biblioteca bastaram apenas poucos passos para que a integração fosse realizada com sucesso. Um passo extra necessário, foi a de enviar para o **OneSignal** qual o id do usuário que está se inscrevendo para receber notificações, isso foi possível de ser realizado facilmente com o uso dessa biblioteca da **OneSignal** específica para Vue.js, através do uso do *Event Listener subscriptionChange*.

Para o envio das **WEB Push Notifications** quando um evento acontece dentro de um processo, foi criado no *back-end* um módulo denominado *oneSignal.js*. Esse módulo é responsável por acionar o *endpoint /notifications* da API REST do OneSignal, passando por parâmetro o identificador do destinatário da mensagem, a mensagem a ser enviada e para qual URL o usuário será redirecionado quando clicar na notificação. A autenticação dessa API ocorre através do envio de um cabeçalho *Authorization* do tipo *Basic* e contendo o valor da API Key da Rest API do aplicativo criado no *dashboard* do OneSignal.

Para que esse módulo pudesse ser acionado, foi definido um *hook* no schema *processes* do banco de dados, executado sempre que uma informação é salva em qualquer documento desse *schema*. Para centralizar quais atributos do *schema* deveria ser verificado se foi alterado, e quando alterado qual o destinatário e a mensagem a ser enviada, foi criado um arquivo que tivesse todas essas informações.

No exemplo mostrado no Código-fonte 1, é definido que na etapa de Plano de estágio (*internshipPlan*), haverá notificações quando os atributos *documentPath* ou *isApproved* forem modificados. Quando *documentPath* for modificado, o orientador deve ser a pessoa notificada com a mensagem **Novo plano de estágio enviado para sua análise**, da mesma forma que quando o atributo *isApproved* for modificado, o estudante deverá ser notificado com a mensagem **Seu plano de estágio foi aprovado pelo orientador de estágio**.

Código-fonte 1 – Exemplo de configuração de notificações

```

1      ....,
2      "INTERNSHIP_PLAN": {
3          ....,
4          "step": "internshipPlan",
5          "notifications": {
6              "documentPath": {
7                  "coordinator": "Novo plano de estagio enviado para sua analise."
8              },
9          "isApproved": {
10             "student": "Seu plano de estagio foi aprovado pelo coordenador de estagio."
11         }
12     }
13 },
14     ....

```

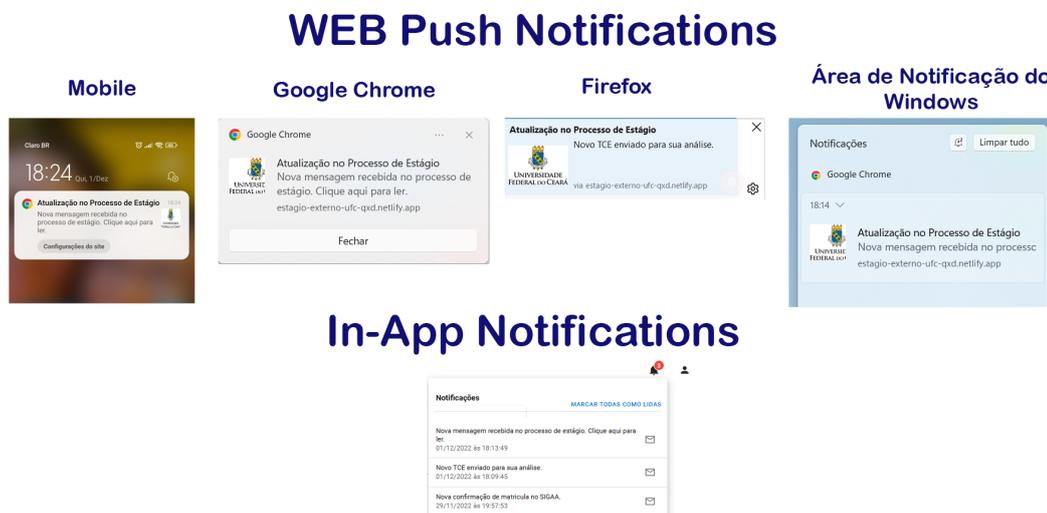
Fonte: Elaborado pelo autor.

Conforme pode ser observado na Figura 20, foi criado um *schema* no banco de dados denominado *notifications* para armazenar as *in-app notifications* do sistema. Isso significa que para uma notificação desse tipo seja enviado a um usuário basta que seja criado um documento dentro desse *schema* contendo o destinatário e a mensagem.

O usuário é redirecionado automaticamente para o processo quando clica em qualquer notificação.

A Figura 29 mostra como são exibidas as notificações do sistema - Tanto as **WEB Push Notifications** quanto as **In-App notifications**. A lista com todas as notificações que são enviadas pelo sistema pode ser encontrada no Quadro 2.

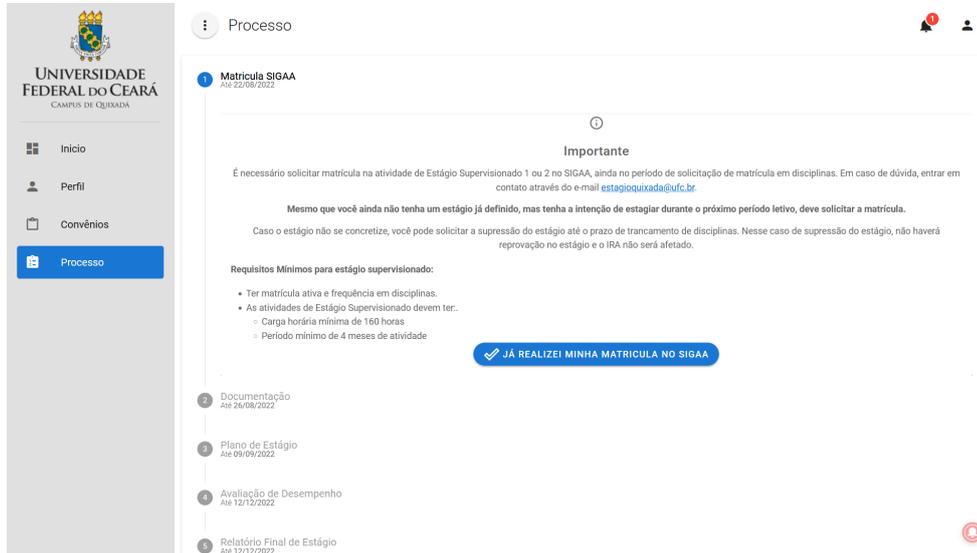
Figura 29 – Notificações do sistema



Fonte: Elaborado pelo autor.

A Figura 30 mostra como ficou a tela da primeira etapa do processo que é mostrada para o aluno após a implementação.

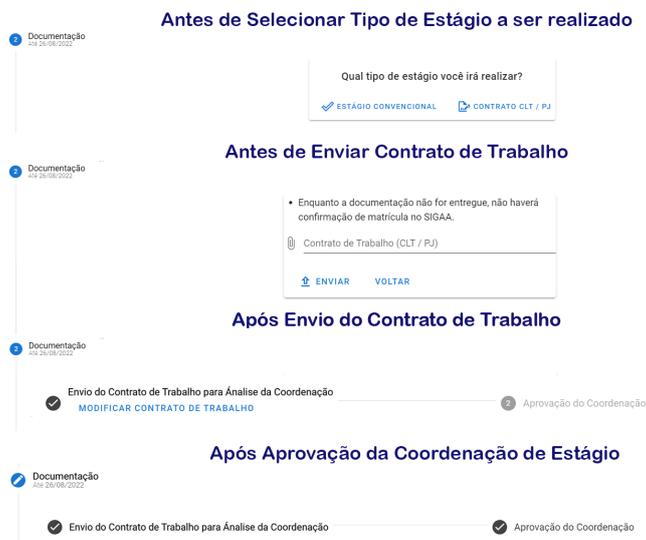
Figura 30 – 1ª etapa do processo - confirmação de matrícula no SIGAA



Fonte: Elaborado pelo autor.

A Figura 31 mostra como ficou a tela da próxima etapa onde o aluno pode selecionar qual tipo de estágio ele vai fazer, e selecionando contrato de trabalho, como o componente será exibido (antes do envio da documentação, após o envio e após a aprovação da coordenação). Caso seja selecionado a opção **Estágio Convencional**, o usuário será direcionado para a tela de cadastro de TCE, que é ilustrado na imagem 27.

Figura 31 – 2ª etapa do processo - documentação



Fonte: Elaborado pelo autor.

A Figura 32 mostra como é exibido o componente para a etapa de plano de estágio (antes do envio, após o envio e após a aprovação da coordenação).

Figura 32 – 3ª etapa do processo - plano de estágio

Antes do Envio do Plano de Estágio

3 Plano de Estágio
Até 09/09/2022

- Preencher e realizar o upload do arquivo em formato PDF.
- Atraso na entrega resultará em redução da nota do Plano de Estágio.
- [Modelo de Plano de Estágio](#)

📎 Contrato de Trabalho (CLT / PJ)

ENVIAR

Após Enviar Plano de Estágio

3 Plano de Estágio
Até 09/09/2022

Envio do Plano de Estágio para Análise da Coordenação [MODIFICAR PLANO DE ESTÁGIO](#) 2 Aprovação da Coordenação

Após Aprovação da Coordenação de Estágio

3 Plano de Estágio
Até 09/09/2022

Envio do Plano de Estágio para Análise da Coordenação Aprovação da Coordenação

Fonte: Elaborado pelo autor.

A Figura 33 mostra como é exibido o componente para a etapa de plano de estágio (antes do envio, após o envio e após a aprovação da coordenação).

Figura 33 – 4ª etapa do processo - avaliação de desempenho

Antes do Envio do Relatório de Avaliação de Desempenho

4 Avaliação de Desempenho
Até 12/12/2022

- A Avaliação de Rendimento deve ser preenchida e assinada pelo Supervisor de Estágio. O supervisor de estágio é o mesmo que está definido no Termo de Compromisso de Estágio.
- Os estagiários de empresas devem enviar o [Modelo de Avaliação de Rendimento](#) para preenchimento e assinatura pelo seu Supervisor de estágio.
- Depois, o estagiário também deve assinar a Avaliação de Rendimento em sua última página.
- Preencher e realizar o upload do arquivo em formato PDF.

📎 Avaliação de Rendimento

ENVIAR

Após Enviar Relatório de Avaliação de Desempenho

4 Avaliação de Desempenho
Até 12/12/2022

Envio da Avaliação de Rendimento para Análise da Coordenação [MODIFICAR AVALIAÇÃO DE RENDIMENTO](#) 2 Aprovação do Coordenação

Após Aprovação da Coordenação de Estágio

4 Avaliação de Desempenho
Até 12/12/2022

Envio da Avaliação de Rendimento para Análise da Coordenação Aprovação do Coordenação

Fonte: Elaborado pelo autor.

A Figura 34 mostra como é exibido o componente para a etapa de relatório final de estágio (antes do envio, após o envio e após a aprovação da coordenação).

Figura 34 – 5ª etapa do processo - relatório final de estágio

Antes do Envio do Relatório Final de Estágio

5 Relatório Final de Estágio
Até 12/12/2022

- Preencher e realizar o upload do arquivo em formato PDF.
- [Modelo de Relatório Final do Estágio](#)

📎 Relatório Final de Estágio

[ENVIAR](#)

Após Enviar Relatório Final de Estágio

5 Relatório Final de Estágio
Até 12/12/2022

Envio do Relatório Final de Estágio para Análise da Coordenação
[MODIFICAR RELATÓRIO FINAL DE ESTÁGIO](#)

2 Aprovação do Coordenação

Após Aprovação da Coordenação de Estágio

6 Relatório Final de Estágio
Até 12/12/2022

Envio do Relatório Final de Estágio para Análise da Coordenação

Aprovação do Coordenação

Fonte: Elaborado pelo autor.

A Figura 35 mostra como é exibido o componente para a última etapa do processo, que é o seminário de apresentação de estágios (antes e após a apresentação).

Figura 35 – 6ª etapa do processo - seminário de apresentação de estágios

Antes da Apresentação do Seminário de Estágios

6 Seminário de Estágio
Em 12/12/2022

- Apresentação dos seminários.
- Cada aluno deve fazer uma apresentação com slides eletrônicos possuindo uma estrutura semelhante à do Relatório Final de Estágio. **Recomenda-se fortemente aos alunos que trabalham no mesmo projeto e apresentem o Seminário no mesmo turno para criarem uma única apresentação de slides para apresentá-la em grupo.**
- **TURNO MANHÃ:**A partir das 8h, somente para os discentes dos cursos de Design Digital e Engenharia de Software. Estagiários que chegarem na sala após 10h não poderão mais apresentar. O evento se encerra imediatamente após a última apresentação dos presentes.
- **TURNO TARDE:**A partir das 13h30, somente para os discentes dos cursos de Ciência da Computação e Sistemas de Informação. Estagiários que chegarem na sala após 15:30h não poderão mais apresentar. O evento se encerra imediatamente após a última apresentação dos presentes.
- Importante assistir às apresentações dos colegas para compreender o panorama geral de estágios e empregos relacionados aos outros estagiários do mesmo curso.
- **ATENÇÃO:** A apresentação do Seminário ocorrerá de forma presencial na sala multiuso (Bloco 2) do Campus da UFC em Quixadá.

Após Apresentação do Seminário de Estágios

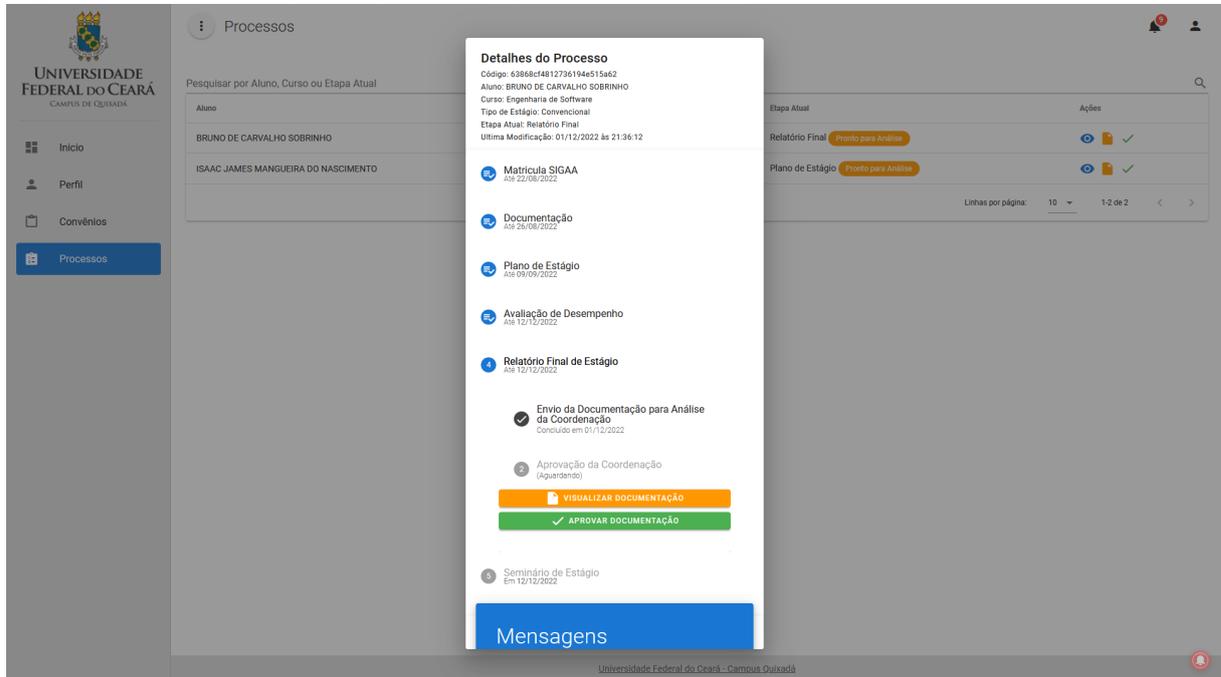
6 Seminário de Estágio
Em 12/12/2022

- Apresentação dos seminários.
- Cada aluno deve fazer uma apresentação com slides eletrônicos possuindo uma estrutura semelhante à do Relatório Final de Estágio. **Recomenda-se fortemente aos alunos que trabalham no mesmo projeto e apresentem o Seminário no mesmo turno para criarem uma única apresentação de slides para apresentá-la em grupo.**
- **TURNO MANHÃ:**A partir das 8h, somente para os discentes dos cursos de Design Digital e Engenharia de Software. Estagiários que chegarem na sala após 10h não poderão mais apresentar. O evento se encerra imediatamente após a última apresentação dos presentes.
- **TURNO TARDE:**A partir das 13h30, somente para os discentes dos cursos de Ciência da Computação e Sistemas de Informação. Estagiários que chegarem na sala após 15:30h não poderão mais apresentar. O evento se encerra imediatamente após a última apresentação dos presentes.
- Importante assistir às apresentações dos colegas para compreender o panorama geral de estágios e empregos relacionados aos outros estagiários do mesmo curso.
- **ATENÇÃO:** A apresentação do Seminário ocorrerá de forma presencial na sala multiuso (Bloco 2) do Campus da UFC em Quixadá.

Fonte: Elaborado pelo autor.

A Figura 36 mostra como é exibido o componente de visualização de processos para os orientadores.

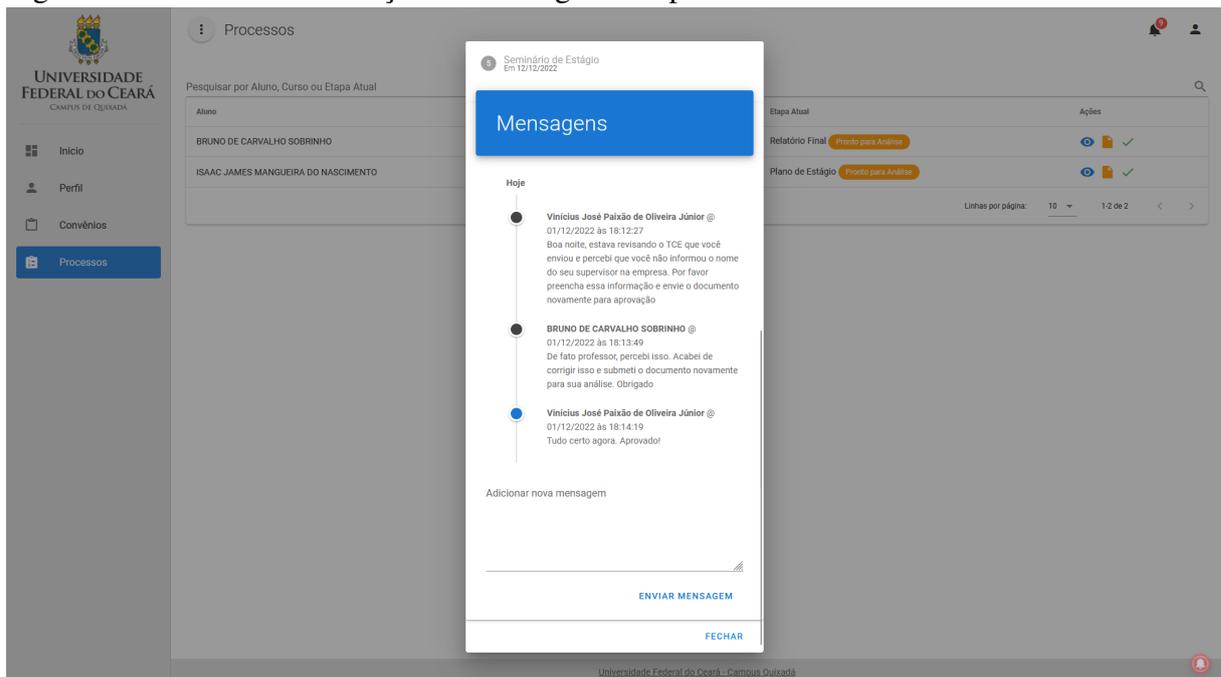
Figura 36 – Visualização de processos pelo orientador



Fonte: Elaborado pelo autor.

A Figura 37 mostra como é exibido o componente de mensagens para os orientadores de estágio.

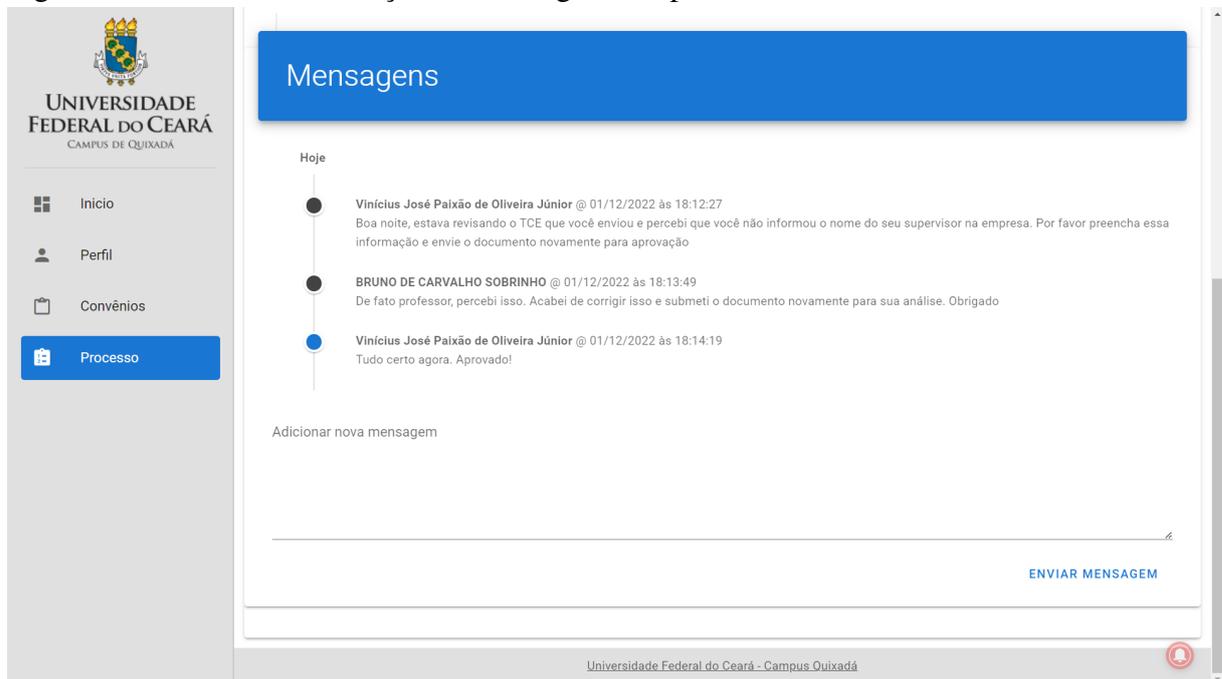
Figura 37 – Tela de visualização de mensagens em processo - orientador



Fonte: Elaborado pelo autor.

A Figura 38 mostra como é exibido o componente de mensagens para os alunos.

Figura 38 – Tela de visualização de mensagens em processo - aluno



Fonte: Elaborado pelo autor.

5.5.6 Apresentar aos alunos informações relevantes sobre o processo de estágio

Conforme pode ser observado na Figura 19, dentro da pasta *front-end* foi criada uma pasta denominada *articles*. Todos os documentos *Markdown* com as informações de estágio que deverá ser exibido aos alunos deverão estar presentes ali.

Foi utilizada a biblioteca JavaScript **marked**¹⁷ na aplicação *front-end* para que fosse realizado o *parse* dos artigos de *Markdown* para *HTML*.

Foram adicionados três artigos sobre estágios para serem apresentados aos alunos: Processo de acompanhamento de estágio, Passos para cumprir a atividade obrigatória de estágio supervisionado 1 ou 2 e Cronograma de Estágio em 2022.2.

As figuras 39, 40 e 41 mostram como esses artigos são apresentados para os alunos.

¹⁷ <https://marked.js.org/>

Figura 39 – Processo de acompanhamento de estágio

Processo de acompanhamento de estágio

Fluxo geral

Antes da confirmação da matrícula no SIGAA:

1. Discente entrega termo de compromisso (ou contrato) de estágio com assinaturas da Empresa/Instituição e Discente.
2. Orientador de estágio assina documentação.
3. Secretaria acadêmica encaminha documentação para homologação pela Agência de Estágios.
4. Após retorno da documentação homologada, Secretaria acadêmica procede a matrícula da atividade no SIS.

Após a confirmação da matrícula no SIGAA:

1. Discente deve verificar se sua matrícula está devidamente realizada.
2. Discente deve entregar o Plano de Estágio ao Orientador.
3. Orientador deve divulgar aos discentes em estágio as datas de entrega de Relatório e de realização de Seminários.
4. Discente deve entregar Relatório e participar de Seminários conforme calendário divulgado.
5. Orientador deve solicitar aos Supervisores de estágio as Avaliações de Rendimento dos estagiários.
6. Orientador deve realizar a avaliação do estágio e lançar notas no SIS.

Observações

- Não serão encaminhados para homologação os contratos que estiverem com erros ou de empresas não conveniadas.
- Para saber o que é verificado nos contratos, leia a lista de conferência.
- Os professores orientadores de estágio são:
 - Sistemas de Informação = [Prof. Régis Pires Magalhães](#)
 - Engenharia de Software = [Prof. Bruno Góis Mateus](#)
 - Redes de Computadores = [Prof. Marcos Dantas Ortiz](#)
 - Ciência da Computação = [Prof. Régis Pires Magalhães](#)
 - Design Digital = [Prof. Bruno Góis Mateus](#)
 - Engenharia de Computação = [Prof. Paulo Armando Cavalcante Aguiar](#)
- Eventualmente, discentes de um curso podem ser alocados para orientação por outros docentes não listados. Consulte a Secretaria se tiver dúvida sobre quem é seu orientador.

INICIAR/CONTINUAR PROCESSO DE ESTÁGIO

Universidade Federal do Ceará - Campus Oixadá

Fonte: Elaborado pelo autor.

Figura 40 – Passos para cumprir a atividade obrigatória de estágio supervisionado 1 ou 2

Passos para cumprir a atividade obrigatória de estágio supervisionado 1 ou 2

1. **Solicitar matrícula** na atividade de Estágio Supervisionado 1 ou 2 no SIGAA, ainda no período de solicitação de matrícula em disciplinas. Em caso de dúvida, entrar em contato através do e-mail estagiooixada@ufc.br.
 - Mesmo que você ainda não tenha um estágio já definido, mas tenha a intenção de estagiar durante o próximo período letivo, deve solicitar a matrícula. Caso o estágio não se concretize, você pode solicitar a supressão do estágio até o prazo de trancamento de disciplinas. Nesse caso de supressão do estágio, não haverá reprovação no estágio e o IRA não será afetado.
2. **Preencher, datar e assinar o TCE** - Termo de Compromisso de Estágio. Caso você tenha um contrato CLT ou PJ, é possível solicitar aproveitamento, devendo-se também realizar o item 1 (Solicitar matrícula).
 - O TCE deve ser assinado pelo estagiário e pelo representante da empresa conveniada à agência de estágios da UFC.
 - Ver prazos e modelos de TCE no [Cronograma de Estágio 2022.2](#).
 - Existe TCE Obrigatório e TCE Não Obrigatório. Distinguir a diferença entre eles em:
 - [Diferenças entre Estágio Obrigatório e Estágio Não Obrigatório](#)
3. **Submeter o TCE ou o contrato CLT ou o contrato PJ** para aprovação do coordenador de estágio do seu curso
4. **Submeter Plano de Estágio, Relatório Final de Estágio, Avaliação de Desempenho** a partir dos modelos e prazos definidos no [Cronograma de Estágio 2022.2](#).
 - Contratos CLT e PJ também devem cumprir este requisito.
5. **Realizar apresentação durante o Seminário de Estágio definido** no [Cronograma de Estágio 2022.2](#).
 - Contratos CLT e PJ também devem cumprir este requisito.

INICIAR/CONTINUAR PROCESSO DE ESTÁGIO

Universidade Federal do Ceará - Campus Oixadá

Fonte: Elaborado pelo autor.

Figura 41 – Cronograma de estágio em 2022.2

Documento / Evento	Data ou Prazo Final	Descrição
Termo de Compromisso de Estágio (TCE) Obrigatório (Supervisionado)	até 26/08/2022 (sexta)	- Enquanto o termo não for entregue, não haverá confirmação de matrícula no SIGAA. - Todos os termos de compromisso (TCEs) de estágio obrigatório e não obrigatório devem conter as assinaturas do estagiário e do supervisor (unidade concedente). - Aproveitamento de contratos de trabalho CLT ou PJ não deve ter submissão de termos de compromisso de estágio obrigatório (TCE), mas sim o envio do contrato de trabalho.
Plano de Estágio	até 09/09/2022 (sexta)	- Preencher e realizar o upload do arquivo em formato PDF. - Atraso na entrega resultará em redução da nota do Plano de Estágio.
Avaliação de Rendimento (preenchida pelo Supervisor de Estágio)	até 12/12/2022 (segunda)	- A Avaliação de Rendimento deve ser preenchida e assinada pelo Supervisor de Estágio. O supervisor de estágio é o mesmo que está definido no Termo de Compromisso de Estágio. - Os estagiários de empresas devem enviar o Modelo de Avaliação de Rendimento para preenchimento e assinatura pelo seu Supervisor de estágio. - Depois, o estagiário também deve assinar a Avaliação de Rendimento em sua última página. - A submissão da avaliação de rendimento deve ser feita em formato PDF.
Relatório Final de Estágio	até 12/12/2022 (segunda)	- Preencher e realizar o upload do arquivo em formato PDF. - Modelo de Relatório Final do Estágio
Seminário de Estágio	em 12/12/2022 (segunda)	- Apresentação dos seminários. - Cada aluno deve fazer uma apresentação com slides eletrônicos possuindo uma estrutura semelhante à do Relatório Final de Estágio. Recomenda-se fortemente aos alunos que trabalhem no mesmo projeto e apresentem o Seminário no mesmo turno para criarem uma única apresentação de slides para apresentá-la em grupo. - TURNO MANHÃ: A partir das 8h, somente para os discentes dos cursos de Design Digital e Engenharia de Software. Estagiários que chegarem na sala após 10h não poderão mais apresentar. O evento se encerra imediatamente após a última apresentação dos presentes. - TURNO TARDE: A partir das 13h30, somente para os discentes dos cursos de Ciência da Computação e Sistemas de Informação. Estagiários que chegarem na sala após 15:30h não poderão mais apresentar. O evento se encerra imediatamente após a última apresentação dos presentes. - Importante assistir às apresentações dos colegas para compreender o panorama geral de estágios e empregos relacionados aos outros estagiários do mesmo curso. ATENÇÃO: A apresentação do Seminário ocorrerá de forma presencial na sala multiuso (Bloco 2) do Campus da UFC em Quixadá. Apresentação remota somente poderá ocorrer caso o estagiário realize ESTÁGIO PRESENCIAL fora de Quixadá. As apresentações remotas iniciarão somente após finalizadas todas as apresentações presenciais.

Fonte: Elaborado pelo autor.

5.6 Hospedagem da aplicação

O sistema desenvolvido nesse trabalho encontra-se disponível no endereço <https://estagio-externo-ufc-qxd.netlify.app/>.

Para a hospedagem desse sistema foram escolhidos três diferentes serviços em nuvem:

- **Netlify**¹⁸: Netlify é uma plataforma como serviço (PaaS) com planos gratuitos que possui suporte a vários frameworks front-end JavaScript, entre eles o Vue.js. Após a criação de um projeto, é possível deixá-lo conectado a uma *branch* específica do github e a cada nova atualização nessa *branch*, um novo *deployment* desse código é realizado automaticamente. Para esse projeto o Netlify está sendo utilizado para a hospedagem da aplicação front-end construída em Vue.js versão 2.
- **Heroku**¹⁹: Heroku é uma plataforma como serviço (PaaS) baseada em contêiner. Os desenvolvedores usam o Heroku para implantar, gerenciar e dimensionar aplicativos modernos. É conhecida por ser flexível, fácil de usar e totalmente gerenciado, dando aos desenvolvedores a liberdade de se concentrar em seu produto principal sem a distração de

¹⁸ <https://www.netlify.com/>

¹⁹ <https://www.heroku.com/>

manter servidores, hardware ou infraestrutura. Possui suporte para diversas linguagens e frameworks *server-side*, entre elas Node.js (HEROKU, 2022). Assim como o Netlify, também possui suporte a conectar uma *branch* específica de um repositório github e a cada atualização nessa *branch* um novo deployment ser realizado automaticamente. Para esse projeto o Heroku está sendo utilizado para hospedagem da REST API back-end construída em Node.js.

- **MongoDB Atlas** ²⁰: A plataforma MongoDB Atlas é um serviço de nuvem que se responsabiliza pela hospedagem, gerenciamento e a segurança de um cluster MongoDB, instanciando servidores de nuvem em plataformas como: Amazon Web Services, Google Cloud Platform e Azure. Sendo considerado um serviço de Banco de dados como Serviço (DBaaS), a plataforma tem o intuito de administrar toda a infraestrutura, manutenção e segurança das máquinas e clusters de seus clientes (RIBEIRO *et al.*, 2021). Para esse projeto o Atlas está sendo utilizado para hospedagem do banco de dados MongoDB da aplicação.

Para a configuração do Netlify foi necessário adicionar uma configuração para que fosse utilizada a pasta *front-end* como raiz do projeto a ser implantado ali. Isso se faz necessário pois ambas as aplicações *front-end* e *back-end* se encontram no mesmo repositório, conforme ilustrado na Figura 19.

Para o heroku foram necessários utilizar dois *buildpacks*, termo utilizado para conjuntos de scripts utilizados para compilar aplicações na plataforma, produzido por terceiros além de um padrão. O primeiro, denominado **subdir-heroku-buildpack**²¹ é utilizado para se tornar possível fazer o deploy de um projeto a partir de uma subpasta de uma pasta raiz, no caso desse projeto, era necessário definir a pasta *back-end* como a pasta padrão do projeto a ser implantado no servidor heroku, uma vez que como já foi mencionado acima, na raiz do repositório se encontram duas pastas que contém a aplicação *back-end* e a *front-end*.

Um outro *buildpack* de terceiros necessário foi o **puppeteer-heroku-buildpack**²². Esse foi necessário para que a biblioteca JavaScript **puppeteer** pudesse ser executada com sucesso no servidor. E por fim foi utilizado também o *buildpack* padrão para compilação de projetos node.js, denominado **heroku/nodejs**.

Em ambos os serviços, foi configurado a *branch develop* do repositório *github* como

²⁰ <https://www.mongodb.com/atlas/database>

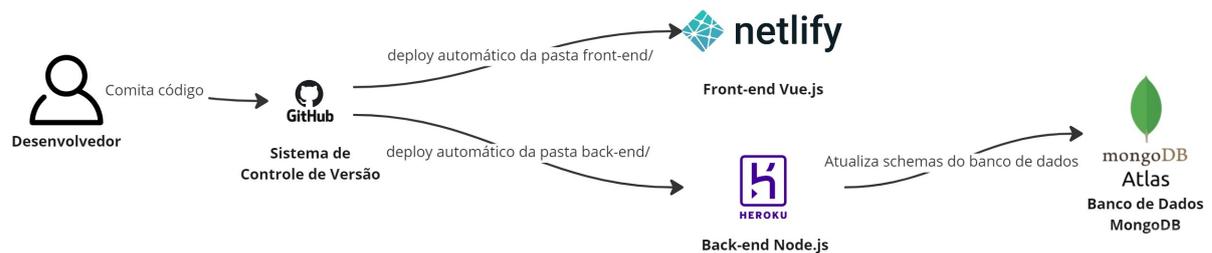
²¹ <https://github.com/timanovsky/subdir-heroku-buildpack>

²² <https://github.com/jontewks/puppeteer-heroku-buildpack#22.0.0-no-xdg-utils>

a padrão a ser implantada automaticamente nos servidores tão logo ocorra alguma atualização nelas. Como a aplicação *back-end* possui a conexão com o serviço MongoDB Atlas, qualquer modificação nos *schemas* do banco de dados identificados durante essa implantação, será automaticamente propagada para o MongoDB Atlas.

A Figura 42 mostra como está estruturado a implantação automatizada (*Continuous deployment*) da aplicação.

Figura 42 – Diagrama de implantação contínua da aplicação



Fonte: Elaborado pelo autor.

5.7 Validação do sistema

A validação do sistema foi realizada apenas pelo desenvolvedor, não foram realizados testes pelos usuários finais, no caso alunos e orientadores de estágio.

Primeiramente foi desenvolvido toda a parte da *API back-end* do sistema. A cada novo *endpoint* criado era realizado um teste utilizando a ferramenta Postman²³ para garantir que o *endpoint* estivesse funcionando e retornando as informações de acordo com o esperado. Uma outra preocupação era de sempre garantir que cada aluno ou orientador só pudesse ter acesso de escrita / leitura a informações que ele realmente pudesse acessar. Por exemplo, um orientador não poderia ver e nem aprovar uma documentação de um aluno de um curso que não é sua responsabilidade.

Posteriormente foi desenvolvida a aplicação *front-end*, bem como realizado a integração dessa aplicação à *API back-end* através da biblioteca JavaScript Axios. Os testes realizados nessa aplicação foram todos manuais tentando-se encontrar *bugs* e consertá-los conforme iam sendo encontrados. A cada fim do desenvolvimento de um módulo completo era realizado testes de regressões em todo o sistema para garantir que nenhuma outra funcionalidade estava sendo danificada pela que acabara de ser desenvolvida.

²³ <https://www.postman.com/>

Como a aplicação front-end utiliza *Vuex* para gerenciamento de estado das informações, foi utilizado a extensão **vue-devtools**²⁴ para acompanhamento de como os dados eram assinaladas nas *stores* do *Vuex* e para garantir que as informações estavam sendo lidas e escritas de forma correta pela aplicação.

Para a realização dos testes de notificações **WEB Push Notifications** foi necessário contornar um problema, uma vez que um navegador *WEB* só está apto a receber as notificações de um único usuário. Para ser possível receber notificações de mais de um usuário foram utilizados diferentes navegadores em dois computadores e um celular. Dessa forma, um computador, por exemplo, estava configurado para receber as notificações de um usuário orientador e outro estava configurado para receber as notificações de um usuário estudante.

Para o teste de integração de login da aplicação utilizando as credenciais da base centralizada foram utilizados os logins de três alunos distintos, sendo um deles o do autor desse trabalho e outros dois de outros estudantes que gentilmente aceitaram ceder as suas credenciais para a realização dos testes.

²⁴ <https://github.com/vuejs/devtools>

6 CONCLUSÕES E TRABALHOS FUTUROS

Nesta seção serão apresentadas as considerações finais desse trabalho, cujo objetivo foi o de desenvolver um sistema para o fluxo de solicitação, envio de documentação e acompanhamento do processo de aproveitamento de estágio-empresa da UFC Quixadá.

Além disso, esse sistema deveria possuir a capacidade de enviar notificações às partes interessadas, sempre que houvesse uma nova atualização dentro de um processo. E por fim, a execução do trabalho consistiu no uso da arquitetura *MEVN*.

6.1 Considerações Finais

Durante a execução desse trabalho, a UFC implementou um módulo de estágios dentro do SIGAA. Com esse módulo, os estudantes podem fazer a consulta de documentos como termos de compromisso, aditivos, rescisões e relatórios de atividades de estágio (UFC, 2022).

Ainda assim esse trabalho consegue ter algumas funcionalidades a mais em relação ao módulo do SIGAA, que é o fato de possuir sistema de *WEB Push Notifications*, Notificações dentro do próprio aplicativo, informações gerais sobre o processo de estágio e a possibilidade de troca de mensagens entre aluno e orientador durante o processo, em caso de algo na documentação precisar ser ajustada. Uma desvantagem desse sistema em relação ao módulo do SIGAA é que por já está dentro da plataforma utilizada para todo o gerenciamento da universidade, o módulo do SIGAA possui uma melhor integração com todos os dados do aluno, estágios e convênios e até mesmo para confirmação de matrícula do aluno posteriormente.

6.2 Contribuições e Dificuldades

Quando desconsiderando esse módulo do SIGAA e observando apenas as formas anteriores utilizadas pela universidade para esse gerenciamento do processo de estágio em empresas conveniadas, que eram troca de e-mails e posteriormente o google classroom, pode se citar o sistema desenvolvido nesse trabalho como contribuição, pois pode ser muito útil para estudantes e orientadores de estágios por permitir o acompanhamento, submissão, revisão e aprovação de documentação por parte de alunos e orientadores, notificações de atualizações, troca de mensagens entre alunos e orientadores e centralização das informações de estágios em um só lugar.

Uma outra contribuição desse trabalho é o fornecimento de um meio de realizar

autenticação e obtenção de dados de usuários direto da base centralizada da UFC Quixadá.

Uma das maiores dificuldades encontradas nesse trabalho foi a implementação da funcionalidade de gerar TCE. Essa dificuldade se deu pela ausência de bibliotecas gratuitas *javascript* que fossem capazes de gerar um arquivo PDF a partir de um arquivo *template* pré-definido, como era o caso do modelo disponibilizado pela UFC. Conforme detalhado na subseção 5.5.4, para contornar tal problema, foi necessário o uso de diferentes bibliotecas que eram capazes de solucionar apenas uma parte do problema e que fazendo o uso delas em conjunto, o arquivo PDF pudesse ser gerado com sucesso.

Uma outra grande dificuldade foi a construção do middleware no *back-end* para a realização de login no sistema utilizando as credenciais da base centralizada. Essa dificuldade e as soluções necessárias para contornar os problemas encontrados foram detalhados na subseção 5.5.1.

6.3 Trabalhos Futuros

Como trabalhos futuros, poderia ser realizado uma rodada de validação do sistema por alunos e orientadores a fim de se obter comentários e possíveis melhorias que possam ser implementados no sistema. Pode-se também implementar dentro do sistema as funcionalidades para preenchimento e geração de Plano de Estágio, Avaliação de Rendimento e Relatório Final de estágio, tal como fora implementado para TCE.

Um outro trabalho futuro poderia ser a integração de login de orientadores de estágios de curso com a base centralizada de alunos, tal como acontece com o login de estudantes, dessa forma não haveria a necessidade de um cadastro por parte do usuário administrador.

Também pode ser feito como trabalho futuro a possibilidade de adição de notificações via e-mail às partes interessadas quando houver uma atualização no processo. Além do uso de um *web scraping* para a obtenção das empresas conveniadas a UFC, dessa forma, poderia ser disponibilizada essa informação no sistema sem necessidade de uso da página pública do SIGAA através de um *Iframe*.

Por fim, pode-se criar uma funcionalidade na qual seja possível a inserção/edição de documentos *markdown* utilizados para apresentar aos alunos instruções gerais sobre o estágio e suas fases através do uso de um editor *wysiwyg*. Pode ser realizado também a melhoria da disposição dos conteúdos da aplicação para acesso em dispositivos móveis

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 jan. 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>. Acesso em: 10 jan. 2022.
- BORGES, A. C. **Desenvolvendo um sistema Web com EXTJS**. 2013. Trabalho de conclusão de curso (Curso de Tecnologia em Análise de Sistemas e Tecnologia da Informação) – Faculdade de Tecnologia de Americana, Americana, 2013. Disponível em: <http://ric.cps.sp.gov.br/handle/123456789/1607>. Acesso em: 15 jan. 2022.
- 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/111788.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. **Tecnologias WEB aplicadas ao ensino de engenharia de estruturas**. 2008. Dissertação (Programa de Pós-Graduação em Engenharia de Estruturas da Escola de Engenharia) – Universidade Federal de Minas Gerais, Belo Horizonte, 2008. Disponível em: <http://hdl.handle.net/1843/PASA-7THGXQ>. Acesso em: 15 jan. 2022.
- 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>. Acesso em: 14 jan. 2022.
- DEVELOPERS, G. **Introduction to Push Notifications**. [S. l.]: Chrome Developers, 2019. Disponível em: <https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications/>. Acesso em: 15 jan. 2022.
- GAUNT, M. **Service Workers: an introduction**. [S. l.]: Chrome Developers, 2021. Disponível em: <https://developers.google.com/web/fundamentals/primers/service-workers/>. Acesso em: 16 jan. 2022.
- GRSCHAU, F. **How to Create Web Push Notifications - Full Tutorial**. [S. l.]: Felix Gerschau, 2020. Disponível em: <https://felixgerschau.com/web-push-notifications-tutorial/>. Acesso em: 17 jan. 2022.
- GUANAIS, K. A. Aplicações ricas de internet. In: **Revista eletrônica de Sistemas de Informação Gestão de Tecnologia**. Belém, PA, Brasil: IESAM, 2008. v. 5. ISSN 2357-9927. Disponível em: <http://www3.iesam-pa.edu.br/ojs/index.php/sistemas/article/view/544>. Acesso em: 14 jan. 2022.
- HEROKU. **What is Heroku?** [S. l.]: Chrome Developers, 2022. Disponível em: <https://www.heroku.com/about/>. Acesso em: 30 nov. 2022.

LEONCIO, D. **Sistema para controle de estágio supervisionado do Departamento Acadêmico de Informática**. 2018. Trabalho de conclusão de curso (Tecnologia em Análise e Desenvolvimento de Sistemas) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2018. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/16809>. Acesso em: 15 jan. 2022.

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**, São Caetano do Sul, v. 14, n. 2, out. 2018. Disponível em: https://seer.uscs.edu.br/index.php/revista_informatica_aplicada/article/view/6935. Acesso em: 30 nov. 2021.

MARKDOWN. **Getting Started with Markdown**. [S. l.]: Markdown Guide, 2022. Disponível em: <https://www.heroku.com/about/>. Acesso em: 30 nov. 2022.

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**. [S. l.]: MDN Web Docs, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API. Acesso em: 16 jan. 2022.

MDN Web Docs. **Push API**. [S. l.]: MDN Web Docs, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/Push_API. Acesso em: 16 jan. 2022.

MDN Web Docs. **Set-Cookie**. [S. l.]: MDN Web Docs, 2022. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie>. Acesso em: 20 nov. 2022.

MEC. **Parecer CNE/CES N° 136/2012**. 2012. Disponível em: 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. [S. l.]: Web dev, 2019. Disponível em: <https://web.dev/notifications/>. Acesso em: 15 jan. 2022.

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

NOLETO, C. **Aplicações web**:: entenda o que são e como funcionam. Belo Horizonte: Trybe, 2020. Disponível em: <https://blog.betrybe.com/desenvolvimento-web/aplicacoes-web/>. Acesso em: 10 jan. 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**. [S. l.]: OneSignal, 2021. Disponível em: <https://documentation.onesignal.com/docs/web-push-quickstart>. Acesso em: 20 jan. 2022.

ONESIGNAL. **About OneSignal**. [S. l.]: OneSignal, 2022. Disponível em: <https://onesignal.com/about>. Acesso em: 20 jan. 2022.

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

RIBEIRO, V. A.; SILVA, V. A. d.; SILVA, L. A. d. Replicação de banco de dados: cluster em ambiente na nuvem com mongodb. **Revista EduFatec: educação, tecnologia e gestão**, Franca, v. 1, n. 4, p. 19–38, jan. / jul. 2021. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2021/10/edufatec-n04v1a02.pdf>. Acesso em: 28 dez. 2021.

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>. Acesso em: 20 jun. 2021.

UFC. **Resolução No 32/CEPE, de 30 de Outubro de 2009**. Fortaleza: UFC, 2009. Disponível em: 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**. Fortaleza: UFC, 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**. Fortaleza: UFC, 2015. Disponível em: <https://producao.ufc.br/wp-content/uploads/2016/10/pop-09-estagio-supervisionado.pdf>. Acesso em: 24 jan. 2022.

UFC. **SIGAA ganha módulo para consulta de documentos e divulgação de vagas de estágio**. Quixadá: UFC Quixadá, 2022. Disponível em: <https://www.quixada.ufc.br/2022/05/19/sigaa-ganha-modulo-para-consulta-de-documentos-e-divulgacao-de-vagas-de-estagio/>. Acesso em: 29 nov. 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. **Informação & Sociedade: Estudos**, Paraíba, v. 29, n. 1, mar. 2019. Disponível em: <https://periodicos.ufpb.br/ojs2/index.php/ies/article/view/44358>.

VUE.JS. **Introdução ao Vue.js**. [S. l.]: Vue.js, 2021. Disponível em: <https://vuejsbr-docs-next.netlify.app/guide/>. Acesso em: 19 jan. 2022.

**APÊNDICE A – EXEMPLO DE TERMO DE COMPROMISSO DE ESTÁGIO GERADO
PELO SISTEMA**



Universidade Federal Do Ceará
Pró-Reitoria De Extensão
Coordenadoria Da Agência De Estágios

TERMO DE COMPROMISSO DE ESTÁGIO OBRIGATÓRIO

Dados da Instituição de Ensino						
Nome: Universidade Federal do Ceará - UFC			CNPJ: 07.272.636/0001-31			
Endereço: Av. da Universidade, 2853, Benfica			Cidade/UF: Fortaleza/CE			
Rep. Legal: José Cândido Lustosa Bittencourt de Albuquerque			Coord. Agência de Estágios: Prof. Rogério Teixeira Mâsih			
Dados da Unidade Concedente						
Razão Social: Empresa Teste						
CNPJ: 23.424.242/4242-42			Setor: Informática			
Endereço: Rua Teste					CEP: 23.232-323	
Cidade/UF: Parnaíba					Fone: (24) 24242-4242	
Nome do Representante Legal: Joaquim José da Silva Xavier						
Nome do Supervisor: Saul Goodman						
Cargo do supervisor: Back-end developer						
E-mail do supervisor: saulgoodman@gmail.com						
Dados do Estagiário						
Nome: BRUNO DE CARVALHO SOBRINHO						
RG: 54588454			CPF: 123.456.789-00		Matrícula: 111111	
Nome da Mãe: Jane Doe Carvalho					Fone: 23323232323	
Endereço: Rua Teste					Cidade/UF: Parnaíba / PI	
Curso: Engenharia de Software					Semestre: 13º Semestre	
Email: bruno.carvalho@gmail.com						
Dados do Professor Orientador						
Nome: Vinícius José Paixão de Oliveira Júnior					Siape: 1207000	
Dados do Seguro contra Acidentes Pessoais						
Empresa Seguradora: Seguros Sura S.A					Apólice: 071.00982.00820-53	
Vigência: 02/12/2021 a 01/12/2022			Morte Acidental: R\$ 10,000.00		Invalidez Permanente: R\$ 10,000.00	
Dados do Estágio:						
Início do estágio: 2022-11-21			Término do estágio: 2023-03-22			
Valor da bolsa mensal ou de outra forma de contraprestação: R\$ 250,00						
Valor do auxílio-transporte: R\$ 250,00			Carga horária semanal: 20 h			
Nome do componente curricular (disciplina/atividade) referente às atividades deste termo de compromisso			Estagio Supervisionado 2			
Horário do estágio:						
Turnos	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
Manhã	08:00 h as 12:00h	08:00 h as 12:01h	08:00 h as 12:02h	08:00 h as 12:03h	08:00 h as 12:04h	08:00 h as 12:05h
Tarde	13:00 h as 17:00h	13:00 h as 17:01h	13:00 h as 17:02h	13:00 h as 17:03h	13:00 h as 17:04h	13:00 h as 17:05h
Noite	19:00 h as 22:00h	19:00 h as 22:01h	19:00 h as 22:02h	19:00 h as 22:02h	19:00 h as 22:03h	19:00 h as 22:04h
Atividades Previstas						
Lorem Ipsum atividades previstas						

As partes firmam o presente Termo de Compromisso de Estágio Obrigatório, observando o disposto na Lei nº 11.788 de 25/09/2008, na Resolução nº 32/CEPE de 30/09/2009, na Portaria nº 123/2018/GR-UFC, no Termo de Convênio já firmado entre a Unidade Concedente e a UFC e nas seguintes cláusulas:

CLÁUSULA PRIMEIRA: Por meio deste Termo, a UNIDADE CONCEDENTE se compromete a oportunizar experiência prática profissional ao ESTAGIÁRIO previamente selecionado e com frequência regular no curso de graduação em que está matriculado na UFC, em conformidade com o Art. 3º, I, da Lei nº 11.788/2008.

CLÁUSULA SEGUNDA: O estágio tem como **objetivo** proporcionar ao estudante a integração entre teoria e prática, a partir da vivência de situações reais e adequadas de trabalho, visando ao aprimoramento profissional e pessoal e obedecendo ao **Plano de Atividades** elaborado acima.

Parágrafo único: As atividades de estágio devem ser compatíveis com o currículo e com os horários escolares do ESTAGIÁRIO, conforme estabelecem o art. 7º, parágrafo único, o art. 3º, III, e o art. 10 da Lei nº 11.788 de



25/09/2008.

CLÁUSULA TERCEIRA: Ficam definidas as seguintes características do estágio:

- a) O **estágio** não pode, em qualquer hipótese, se estender após a conclusão do curso;
- b) A duração do estágio ora firmado não ultrapassará a data prevista para o encerramento do semestre letivo no qual o estudante está matriculado na disciplina de estágio obrigatório, devendo qualquer necessidade excepcional de extensão ser devidamente justificada pelo (a) coordenador (a) do curso ou pelo (a) professor (a) da referida disciplina;
- c) O estudante estagiará respeitando o limite máximo de 6 (seis) horas diárias e 30 (trinta) horas semanais, conforme o art. 10 da Lei nº 11.788 de 25/09/2008;
- d) A **carga horária do estágio será reduzida pelo menos à metade nos períodos de avaliação** do ESTAGIÁRIO, a fim de garantir o bom desempenho do estudante, nos termos do Art. 10, §2º, da Lei nº 11.788 de 25/09/2008;
- e) Para usufruir da redução prevista na alínea acima, o estagiário deverá comunicar previamente à Concedente os dias em que se realizarão as avaliações e apresentar, após o comparecimento, **Declaração de Realização de Avaliação Acadêmica expedida pelo professor da (s) disciplina (s) ou pela Coordenação do Curso;**
- f) A UFC oferece **seguro contra acidentes pessoais** a todos os seus estudantes devidamente matriculados e que estão realizando atividades de estágio obrigatório, contemplando, portanto, o ESTAGIÁRIO parte deste Termo, **durante a vigência do compromisso de estágio;**
- g) O presente termo deve ser registrado na Agência de Estágios da UFC **antes do início** das atividades ou em até **1(um) mês**, a contar da data constante no documento para a **vigência inicial**, conforme Portaria 123/2018/GR-UFC.

CLÁUSULA QUARTA: Compete ao ESTAGIÁRIO:

- a) Cumprir as normas internas da UNIDADE CONCEDENTE, especialmente as de orientação do plano de atividades constante neste Termo, devendo apresentar ao Professor Orientador da UFC, ao final do semestre letivo do estágio obrigatório ou a cada período de 6 (seis) meses, nos casos em que o estágio supervisionado tenha duração superior ao semestre letivo, o **relatório das atividades** desenvolvidas;
- b) Seguir a orientação articulada entre o Supervisor de Estágio e o Professor Orientador designados, respectivamente, pela UNIDADE CONCEDENTE e pela UFC;
- c) Diante da impossibilidade de cumprir o estabelecido neste Termo, comunicar a circunstância à UNIDADE CONCEDENTE, ficando ressalvado, desde logo, que suas obrigações acadêmicas e a falta de pertinência das atividades com a qualificação profissional pretendida serão consideradas motivos justos para a rescisão;
- d) Em caso de desistência do Estágio, comunicar à Empresa com antecedência mínima de 05 (cinco) dias e entregar termo de rescisão contratual à Agência de Estágios da UFC,

CLÁUSULA QUINTA: São motivos para a rescisão imediata deste Termo de Compromisso de Estágio a ocorrência das seguintes hipóteses:

- a) Conclusão, trancamento ou abandono do Curso;
- b) Transferência para curso que não tenha relação com as atividades de estágio previstas no plano acima elaborado;
- c) Descumprimento do convencionado no presente Termo;
- d) Prática comprovada de conduta danosa, não estando o ESTAGIÁRIO isento de arcar com as perdas e os danos decorrentes.

CLÁUSULA SEXTA: O estágio não acarretará vínculo empregatício de qualquer natureza, conforme Art. 3º, *caput* e § 2º, e Art. 2º da Lei nº 11.788 de 25/09/2008.

CLÁUSULA SÉTIMA: O descumprimento das condições estabelecidas neste Termo pela UNIDADE CONCEDENTE caracteriza vínculo de emprego com o ESTAGIÁRIO, para todos os fins da legislação trabalhista e previdenciária, conforme estabelece o art. 15 da Lei nº 11.788 de 25/09/2008.

CLÁUSULA OITAVA: Qualquer alteração do estabelecido neste Termo será feita mediante Aditivo, com a anuência das partes envolvidas.

E, por estarem devidamente cientes das condições aqui estipuladas, bem como das disposições legais vigentes sobre o assunto, firmam a UNIDADE CONCEDENTE e o ESTAGIÁRIO, com interveniência da UFC, o presente TERMO, em 03 (três) vias de igual teor e forma, para que este produza seus devidos efeitos legais.

DECLARO serem exatas e verdadeiras as informações aqui prestadas sob pena de responsabilidade administrativa, cível e penal.

Fortaleza - CE, 30/11/2022.

Estagiário

Unidade Concedente

Professor Orientador UFC

Agência de Estágios UFC

APÊNDICE B – TEMPLATE UTILIZADO PELA BIBLIOTECA DOCXTEMPLATER

PARA GERAÇÃO DE TCE

TERMO DE COMPROMISSO DE ESTÁGIO OBRIGATÓRIO

Dados da Instituição de Ensino						
Nome: Universidade Federal do Ceará – UFC			CNPJ: 07.272.636/0001-31			
Endereço: Av. da Universidade, 2853, Benfica			Cidade/UF: Fortaleza/CE			
Rep. Legal: José Cândido Lustosa Bittencourt de Albuquerque			Coord. Agência de Estágios: Prof. Rogério Teixeira Mâsih			
Dados da Unidade Concedente						
Razão Social: {razaoSocial}						
CNPJ: {cnpj}			Setor: {setor}			
Endereço: {endereco}					CEP: {cep}	
Cidade/UF: {cidade}					Fone: {fone}	
Nome do Representante Legal: {representanteLegal}						
Nome do Supervisor: {nomeSupervisor}						
Cargo do supervisor: {cargoSupervisor}						
E-mail do supervisor: {emailSupervisor}						
Dados do Estagiário						
Nome: {estagiarioNome}						
RG: {estagiarioRg}		CPF: {estagiarioCpf}		Matrícula: {estagiarioMatricula}		
Nome da Mãe: {estagiarioNomeMae}				Fone: {estagiarioFone}		
Endereço: {estagiarioEndereco}				Cidade/UF: {estagiarioCidade} / {estagiarioUf}		
Curso: {estagiarioCurso}				Semestre: {estagiarioSemestre}		
Email: {estagiarioEmail}						
Dados do Professor Orientador						
Nome: {orientadorNome}					Siape: {orientadorSiape}	
Dados do Seguro contra Acidentes Pessoais						
Empresa Seguradora: Seguros Sura S.A				Apólice: 071.00982.00820-53		
Vigência: 02/12/2021 a 01/12/2022		Morte Acidental: R\$ 10,000.00		Invalidez Permanente: R\$ 10,000.00		
Dados do Estágio:						
Início do estágio: {estagioDataInicio}			Término do estágio: {estagioDataFim}			
Valor da bolsa mensal ou de outra forma de contraprestação: {estagioValorBolsa}						
Valor do auxílio-transporte: {estagioValorAuxilio}			Carga horária semanal: {estagioCargaHoraria} h			
Nome do componente curricular (disciplina/atividade) referente às atividades deste termo de compromisso			{estagioComponenteCurricular}			
Horário do estágio:						
Turnos	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
Manhã	{horarioDia1Manha}	{horarioDia2Manha}	{horarioDia3Manha}	{horarioDia4Manha}	{horarioDia5Manha}	{horarioDia6Manha}
Tarde	{horarioDia1Tarde}	{horarioDia2Tarde}	{horarioDia3Tarde}	{horarioDia4Tarde}	{horarioDia5Tarde}	{horarioDia6Tarde}
Noite	{horarioDia1Noite}	{horarioDia2Noite}	{horarioDia3Noite}	{horarioDia4Noite}	{horarioDia5Noite}	{horarioDia6Noite}
Atividades Previstas						
{atividadesPrevistas}						

As partes firmam o presente Termo de Compromisso de Estágio Obrigatório, observando o disposto na Lei nº 11.788 de 25/09/2008, na Resolução nº 32/CEPE de 30/09/2009, na Portaria nº 123/2018/GR-UFC, no Termo de Convênio já firmado entre a Unidade Concedente e a UFC e nas seguintes cláusulas:

CLÁUSULA PRIMEIRA: Por meio deste Termo, a UNIDADE CONCEDENTE se compromete a oportunizar experiência prática profissional ao ESTAGIÁRIO previamente selecionado e com frequência regular no curso de graduação em que está matriculado na UFC, em conformidade com o Art. 3º, I, da Lei nº 11.788/2008.

CLÁUSULA SEGUNDA: O estágio tem como **objetivo** proporcionar ao estudante a integração entre teoria e prática, a partir da vivência de situações reais e adequadas de trabalho, visando ao aprimoramento profissional e pessoal e obedecendo ao **Plano de Atividades** elaborado acima.

Parágrafo único: As atividades de estágio devem ser compatíveis com o currículo e com os horários escolares do ESTAGIÁRIO, conforme estabelecem o art. 7º, parágrafo único, o art. 3º, III, e o art. 10 da Lei nº 11.788 de 25/09/2008.

CLÁUSULA TERCEIRA: Ficam definidas as seguintes características do estágio:

a) O estágio não pode, em qualquer hipótese, se estender após

a conclusão do curso;

b) A duração do estágio ora firmado não ultrapassará a data prevista para o encerramento do semestre letivo no qual o estudante está matriculado na disciplina de estágio obrigatório, devendo qualquer necessidade excepcional de extensão ser devidamente justificada pelo (a) coordenador (a) do curso ou pelo (a) professor (a) da referida disciplina;

c) O estudante estagiará respeitando o limite máximo de 6 (seis) horas diárias e 30 (trinta) horas semanais, conforme o art. 10 da Lei nº 11.788 de 25/09/2008;

d) A carga horária do estágio será reduzida pelo menos à metade nos períodos de avaliação do ESTAGIÁRIO, a fim de garantir o bom desempenho do estudante, nos termos do Art. 10, §2º, da Lei nº 11.788 de 25/09/2008;

e) Para usufruir da redução prevista na alínea acima, o estagiário deverá comunicar previamente à Concedente os dias em que se realizarão as avaliações e apresentar, após o comparecimento, **Declaração de Realização de Avaliação Acadêmica expedida pelo professor da (s) disciplina (s) ou pela Coordenação do Curso;**

f) A UFC oferece **seguro contra acidentes pessoais** a todos os seus estudantes devidamente matriculados e que estão realizando atividades de estágio obrigatório, contemplando, portanto, o ESTAGIÁRIO parte deste Termo, **durante a vigência do compromisso de estágio;**

g) O presente termo deve ser registrado na Agência de Estágios da UFC **antes do início** das atividades ou em até **1(um) mês**, a contar da data constante no documento para a **vigência inicial**, conforme Portaria 123/2018/GR-UFC.

CLÁUSULA QUARTA: Compete ao ESTAGIÁRIO:

a) Cumprir as normas internas da UNIDADE CONCEDENTE, especialmente as de orientação do plano de atividades constante neste Termo, devendo apresentar ao Professor Orientador da UFC, ao final do semestre letivo do estágio obrigatório ou a cada período de 6 (seis) meses, nos casos em que o estágio supervisionado tenha duração superior ao semestre letivo, o **relatório das atividades** desenvolvidas;

b) Seguir a orientação articulada entre o Supervisor de Estágio e o Professor Orientador designados, respectivamente, pela UNIDADE CONCEDENTE e pela UFC;

c) Diante da impossibilidade de cumprir o estabelecido neste Termo, comunicar a circunstância à UNIDADE CONCEDENTE, ficando ressalvado, desde logo, que suas obrigações acadêmicas e a falta de pertinência das atividades com a qualificação profissional pretendida serão consideradas motivos justos para a rescisão;

d) Em caso de desistência do Estágio, comunicar à Empresa com antecedência mínima de 05 (cinco) dias e entregar termo de rescisão contratual à Agência de Estágios da UFC,

CLÁUSULA QUINTA: São motivos para a rescisão imediata deste Termo de Compromisso de Estágio a ocorrência das seguintes hipóteses:

a) Conclusão, trancamento ou abandono do Curso;

b) Transferência para curso que não tenha relação com as atividades de estágio previstas no plano acima elaborado;

c) Descumprimento do convencionado no presente Termo;

d) Prática comprovada de conduta danosa, não estando o ESTAGIÁRIO isento de arcar com as perdas e os danos decorrentes.

CLÁUSULA SEXTA: O estágio não acarretará vínculo empregatício de qualquer natureza, conforme Art. 3º, *caput* e § 2º, e Art. 2º da Lei nº 11.788 de 25/09/2008.

CLÁUSULA SÉTIMA: O descumprimento das condições estabelecidas neste Termo pela UNIDADE CONCEDENTE caracteriza vínculo de emprego com o ESTAGIÁRIO, para todos os fins da legislação trabalhista e previdenciária, conforme estabelece o art. 15 da Lei nº 11.788 de 25/09/2008.

CLÁUSULA OITAVA: Qualquer alteração do estabelecido neste Termo será feita mediante Aditivo, com a anuência das partes envolvidas.

E, por estarem devidamente cientes das condições aqui estipuladas, bem como das disposições legais vigentes sobre o assunto, firmam a UNIDADE CONCEDENTE e o ESTAGIÁRIO, com interveniência da UFC, o presente TERMO, em 03 (três) vias de igual teor e forma, para que este produza seus devidos efeitos legais.

DECLARO serem exatas e verdadeiras as informações aqui prestadas sob pena de responsabilidade administrativa, cível e penal.

Fortaleza - CE, {tceData}.

Estagiário

Unidade Concedente

Professor Orientador UFC

Agência de Estágios UFC