

**CENTRO PAULA SOUZA**  
**ETEC ZONA LESTE**  
**Habilitação Profissional Técnica de Nível Médio de Técnico em**  
**Desenvolvimento de Sistemas**

**Beatriz de Oliveira Santos**  
**Vinícius Santana de Carvalho**

**SISTEMA DE MANUTENÇÃO E OCORRÊNCIAS**

**São Paulo**  
**2022**

**Beatriz de Oliveira Santos**  
**Vinícius Santana de Carvalho**

## **SISTEMA DE MANUTENÇÃO E OCORRÊNCIAS**

Trabalho de Conclusão de Curso  
apresentado à Etec Zona Leste, como  
requisito parcial para a obtenção do título  
de Técnico em Desenvolvimento de  
Sistemas. Orientador (a): Prof. Tamara  
Cristina Ferreira.

**São Paulo**  
**2022**

## **AGRADECIMENTOS**

Agradeço primeiramente aos nossos familiares e professores que sempre estiveram dispostos a auxiliar e contribuir no aprendizado em especial a orientadora Prof. Tamara e ao antigo orientador Prof. Jefferson.

A minha instituição por ter me dado à chance e as ferramentas que permitiram chegar ao fim desse período de forma satisfatória.

Ao administrador Patrick e ao Auxiliar Docente do DTI Ralf Gerônimo por nos disponibilizar seu tempo, informações e conselhos, para assim, termos a oportunidade de colocar em prática os conhecimentos que adquirimos, alcançando o final do projeto.

E por fim como dito por Paulo Freire “Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção”, e a pessoa que melhor representa esta frase é o queridíssimo Professor Wagner França Marques, agradecemos por todo o apoio e auxílio que nos prestou desde o início, sem seu apoio este projeto não teria existido.

“Pensar no desenvolvimento e manutenção do colaborador é garantir o clima organizacional saudável para a empresa.”

**Fábio Santos**

## RESUMO

Os *Softwares* caminham no aperfeiçoamento de métodos, técnicas e ferramentas na pretensão de trazer qualidade e produtividade no desenvolvimento das atividades diárias, sejam elas, profissionais ou não. Neste contexto, este trabalho tende a apresentar um *software* que possa auxiliar a gestão da área de manutenções e ocorrências do departamento administrativo e do departamento de Tecnologia da Informação (DTI) da instituição de ensino ETEC Zona Leste. Tais departamentos são incumbidos a dar suporte no setor predial (instalações elétricas e hidráulicas, segurança contra incêndio etc.), e no tecnológico (parte eletrônica). Para tal, foi desenvolvido um *software*, utilizando da tecnologia *web* com o uso de *frameworks*. Com a funcionalidade de reportar, monitorar e solicitar chamados, gerenciando as demandas e reportando os *status* das solicitações.

**Palavras-chave:** Manutenção; Predial; *Web*; *Frameworks*.

## ABSTRACT

*Software trucks in the improvement of methods, techniques, and tools to bring quality and productivity in the development of daily activities, whether professional or not. In this context, this work tends to present a software that can help the management of the area of maintenance and occurrences of the administrative department and the department of Information Technology (DTI) of the educational institution ETEC Zona Leste. These departments are responsible for providing support in the building sector (electrical and hydraulic installations, fire safety etc.), and in the technological sector (electronics). For this, a software was developed, using web technology with the use of frameworks. With the functionality to report, monitor and request tickets, managing demands and reporting the status of requests.*

*Keywords: Maintenance; building; web; frameworks.*

## LISTA DE FIGURAS

Figura 1 - Diagrama de Atividades.....	29
Figura 2 - Diagrama de Caso de Uso .....	30
Figura 3 - Diagrama de Classe .....	31
Figura 4 - Protótipo da tela de login.....	32
Figura 5 - Protótipo Tela Inicial .....	32
Figura 6 - Protótipo Tela de solicitação de Manutenção DTI .....	33
Figura 7 - Protótipo Tela de Solicitação de Manutenção Predial .....	33
Figura 8 - Protótipo Tela de Formulário de Ocorrência .....	34
Figura 9 - Protótipo Tela Inicial Administrador DTI .....	34
Figura 10 - Protótipo Tela de Chamadas de Manutenção DTI.....	35
Figura 11 - Protótipo Tela de Ocorrências DTI .....	35
Figura 12 - Protótipo Tela Inicial Administrador Predial.....	36
Figura 13 - Protótipo Tela de Chamados de Manutenção Predial .....	36
Figura 14 - Protótipo Tela de Ocorrências Predial.....	37
Figura 15 - Tela Inicial .....	38
Figura 16 - Tela Inicial mobile .....	39

## LISTA DE ABREVIATURAS OU SIGLAS

CEETPS	Centro Estadual de Educação Tecnológica Paula Souza
ETEC	Escola Técnica Estadual
DTI	Departamento de Tecnologia da Informação
ICT	Instituto de Ciência e Tecnologia
CONSIP	Conselho das Instituições de Pesquisa do Estado de São Paulo
FATEC	Faculdade de Tecnologia do Estado de São Paulo
EAD	Ensino a Distância
Semi EAD	Semipresencial
CSS	<i>Cascading Style Sheet</i> (Folha de Estilo em Cascata)
HTML	<i>Hyper Text Markup Language</i> (Linguagem de Marcação de Hiper Texto)
Tags	Criador de elementos nas páginas web
Browser	Navegador de (rede, web, internet).
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
URL	<i>Uniform Resource Locator</i> (Localizador Uniforme de Recursos)
PHP	Hypertext Preprocessor



## SUMÁRIO

1.	INTRODUÇÃO .....	14
1.1.	Justificativa .....	14
1.2.	Objetivos .....	15
1.2.1.	Objetivo Geral .....	15
1.2.2.	Objetivos Específicos.....	15
2.	REFERENCIAL TEÓRICO .....	16
2.1.	Centro Paula Souza.....	16
2.1.1.	ETEC .....	17
2.2.	<i>Web Site</i> .....	17
2.3.	CSS.....	18
2.4.	HTML.....	19
2.5.	<i>Javascript</i> .....	20
2.6.	PHP.....	21
2.7.	Node.js.....	21
2.8.	Frameworks.....	22
2.8.1.	<i>Laravel</i> .....	23
2.8.2.	<i>Bootstrap</i> .....	25
2.8.3.	<i>Firebase</i> .....	26
2.8.4.	<i>Express.js</i> .....	26
2.9.	Tipos de Manutenção .....	27
2.9.1.	Manutenção Corretiva .....	27
2.9.2.	Manutenção Preventiva .....	27
2.9.3.	Manutenção Preditiva .....	28
3.	METODOLOGIA.....	29
4.	RESULTADOS E DISCUSSÃO .....	31
4.1.	Modelo Pronto .....	37
5.	CONCLUSÃO .....	39
	REFERÊNCIAS .....	41

## **1. INTRODUÇÃO**

As edificações são o suporte físico para realização direta e indireta de todas as atividades da instituição, tendo portanto, um valor fundamental para o perfeito atendimento aos usuários por muitos anos, e ao longo deste tempo deve resistir aos agentes ambientais e de uso, que alteram suas propriedades técnicas iniciais (Associação Brasileira de Normas Técnicas, 2012). Atualmente existem muitas formas de se gerenciar os processos de manutenção e ocorrências, dentre eles, o mais utilizado é o uso de softwares.

A qualidade no atendimento do setor de TI e predial pode evitar com que serviços essenciais da instituição fiquem parados por falha de hardware, software ou estruturais. Fantini (2001), ressalta que quanto maior a união entre qualidade e agilidade no atendimento, maior será o reconhecimento por parte do cliente, proporcionando, assim, total confiabilidade no suporte fornecido.

Conforme apontado por Cremonini (1988), as edificações necessitam passar por vistorias periódicas de manutenção, devido a inúmeros materiais e componentes das construções que são degradados de forma diferenciada ao longo do tempo, necessitando de restaurações com o objetivo de conservar as condições de uso e funcionamento. Seguindo essa linha pensamento, pode-se dizer que a manutenção de qualquer equipamento é importante, para garantir as condições de uso.

### **1.1. Justificativa**

O Tribunal de Contas da União (TCU), afirma no acórdão 504/2013 que a implantação de um sistema de gestão da qualidade proporciona mecanismos para garantir que o sistema de manutenção de um órgão público atenda aos requisitos estabelecidos na NBR 5674 (Associação Brasileira de Normas Técnicas, 2012) e que a eficiência do sistema traga condições para ser continuamente avaliado pelos gestores e também pelos órgãos de auditoria e controle (BRASIL, 2013).

Kalil e Pujadas (2011) demonstram através de estudo realizado pela câmara de inspeção predial do Ibape/SP que 66% das prováveis causas e origens dos acidentes, em edificações com mais de 10 anos, são relacionadas à deficiência com a manutenção, perda precoce de desempenho e deterioração acentuada. Constata-se

que não basta à indústria da construção civil investir na qualidade de projetos, sistemas executivos e tecnologia, se os usuários, proprietários e gestores não se conscientizarem do papel fundamental que exercem sobre a performance de desempenho das edificações.

A ETEC Zona Leste oferece atualmente 6 cursos técnicos, atendendo os alunos, além de abrigar professores, técnicos administrativos e trabalhadores terceirizados em suas atividades produtivas de ensino e administração. Sendo que a instituição não dispõe de um sistema que gerencia as ocorrências de manutenções prediais e estas são feitas por meio de solicitações verbais e tratadas de forma pontual.

## **1.2. Objetivos**

### **1.2.1. Objetivo Geral**

Este projeto visa apresentar um sistema de manutenção e ocorrências desenvolvido para gerenciar os processos de manutenção corretiva e preventiva do setor administrativo e DTI da ETEC Zona Leste. Sugerindo a sua implementação na gestão da manutenção predial e dos equipamentos eletrônicos na ETEC Zona Leste.

O sistema é dedicado, porém, ele pode ser adaptado para uso em outras instituições.

### **1.2.2. Objetivos Específicos**

- Representar o objeto de estudo, visualizando como as atividades de manutenção são realizadas na ETEC Zona Leste;
- Sugerir um modelo de ações que se destinam a atender as necessidades relativas à gestão da manutenção;

## **2. REFERENCIAL TEÓRICO**

### **2.1. Centro Paula Souza**

O Centro Estadual de Educação Técnica e Tecnológica Paula Souza (CEETPS) iniciou suas atividades no ano de 1969 – na época, vinculado para fins administrativos à Secretaria da Educação e, para medidas financeiras, à Secretaria da Fazenda com a premissa de organizar os primeiros cursos superiores de tecnologia (MORAES et. al, 2022).

No ano de 1976, o CEETPS tornou-se uma autarquia vinculada à Secretaria de Desenvolvimento Econômico, Ciência, Tecnologia e Inovação do Estado de São Paulo que é responsável por administrar a rede de escolas médio/ técnicas e faculdades tecnológicas (SCHWARTZMAN, 2014). De acordo com Trevisan e Veloso (2007), no estado de São Paulo, ela é responsável por uma rede composta de mil escolas de formação profissionalizante que atende mais de 1 milhão de alunos em todas as regiões.

Como principal objetivo, o CEETPS atua diretamente na formação e no desenvolvimento profissional dos alunos no Estado de São Paulo, disponibilizando cursos nas mais diversas áreas e com fácil acessibilidade para pessoas mais carentes. Ele desenvolve os estudantes diretamente ao mercado de trabalho, assim suprimindo a grande demanda das empresas e tornando-se um grande pilar para a infraestrutura do Estado.

Não somente no âmbito profissional, o CEETPS recentemente em setembro de 2021 foi reconhecido em uma votação por unanimidade pelo Conselho das Instituições de Pesquisa do Estado de São Paulo (CONSIP) como um Instituto de Ciência e Tecnologia (ICT), o qual se tem por objetivo o desenvolvimento de pesquisas científicas e tecnológicas sem fins lucrativos.

Após a regulamentação da Lei Federal nº 9394/96, a partir do Decreto nº2208/97, que regulamentou o Ensino Técnico, a Coordenadoria do Ensino Técnico

do Centro Estadual de Educação Tecnológica "Paula Souza" definiu um projeto pedagógico não só para atender à imposição legal, mas que acompanhasse as mudanças ocorridas no mundo do trabalho e as transformações vivenciadas pela sociedade contemporânea. (LDB 9349/96).

Os cursos oferecidos pelo CEETPS de nível técnico destinados as Escolas Técnicas (ETEC) que também atuam com o Ensino Médio, compõem um leque de 224 opções para áreas principalmente produtivas, sendo elas públicas ou privadas. Os cursos oferecidos de nível superior com graduação tecnológica voltados as Faculdades de Tecnologia (FATEC), são um total de 86 que abrangem diversas áreas. Não só essas opções, o CEETPS também oferece cursos de pós-graduação, atualização tecnológica e extensões. Todos esses cursos estão presentes em diferentes modalidades, sendo elas: O presencial, ensino a distância (EAD), semipresencial (Semi EAD).

#### **2.1.1. ETEC**

De acordo com Schwartzman (2014) as ETEC's são instituições de ensino, que dispõe de 201 cursos técnicos e 25 cursos de especialização técnica.

Tais cursos são classificados como ensino profissionalizantes, sendo eles um curso voltado mais diretamente para o mercado de trabalho, possibilitando muitas vezes um certificado específico.

A instituição que iremos abordar é a Etec da Zona Leste, tal foi fundada em fevereiro de 2002, localizada na Av. Águia de Haia, 2633. A instituição disponibiliza 6 cursos técnicos, sendo eles: Técnico em administração, contabilidade, desenvolvimento de sistemas, logística, recursos humanos, e serviços jurídicos.

#### **2.2. Web Site**

*Website* é a junção de dois termos do inglês, *web* (rede) e *site* (lugar). No virtual o termo é usado para se referir a uma página ou a um acervo de páginas, nas quais são interconexão, acessadas por via da internet por meio de endereços específicos (SIGNIFICADOS, 2022).

E desde que a Internet surgiu, seu desígnio essencial é a publicação e troca de informações. Tal proposta é assentada mediante a navegação em documentos virtuais arranjados na rede de forma descentralizada. A Internet preferenciava acesso às publicações científicas. Conforme o transcorrer do tempo, este quadro foi alterando-se e hoje identifica-se que a Internet hospeda mais do que publicações científicas, guias, notícias, tutoriais, imagens, vídeos, serviços, entre outros.

Mais também há estruturas, que conforme o W3C (*World Wide Web Consortium*), são os 3 pilares da *web* (1) a definição de nomenclaturas para o encontro de fontes de informação na *Web*, tal nomenclatura é denominada como URL; (2) a utilização de protocolo para controle de acesso a estas fontes de informações, sendo o protocolo HTTP; e (3) a linguagem de Hipertexto, para tornar a locomoção em meio as fontes de informação mais fáceis: o HTML (TANENBAUM, 2003).

Para o desenvolvimento de *web sites* usa-se variadas linguagens de programação. Tais linguagens se decompõem em duas categorias: linguagens *client-side* e *server-side*. Linguagens *client-side* são desempenhadas adjacentes ao cliente, por via do navegador, comumente são empregadas a fim de adquirir uma melhor experiência do usuário, disponibilizando recursos que melhoram a utilização e navegação, autenticação de formulário e outras funcionalidades. *Javascript* é a linguagem *client-side* mais aplicada no desenvolvimento de conteúdos dinâmicos e abastado em recursos (HAVERBEKE, 2014). Quanto as linguagens *server-side* são desempenhadas no servidor e aprovisiona a lógica principal da aplicação, processando dados com maior agregação e perdurando informações em banco de dados.

### 2.3. CSS

O CSS (*Cascading Style Sheets*) é uma linguagem de estilização incumbido de determinar a apresentação visual de um documento HTML, complementados elementos como cores, sombras e animações (STARK, 2010). O CSS engloba seletores que se encontram associados às *tags* do HTML. Através dos seletores torna-se possível referenciar trechos particulares do documento HTML, estilizando-os individualmente ou coletivamente. Além disso há seletores pré-definidos como *div*,

*span*, *table* e *p*, também é viável criar seletores, tais são denominados como *id* e *class*. Os seletores *id*'s deveriam ser manuseados por elementos HTML que não se reprisam na página, em outras palavras, elementos únicos. Os seletores *class* é capaz de ser concedidos a mais de um elemento HTML (DURANGO, 2015).

O CSS tem propriedades que são pré-definidas e não são capazes de ser adicionadas e/ou removidas. Cada propriedade aguarda um valor apropriado, já outras subsidiam diversos valores e formatos.

O CSS3 é o parâmetro mais recente da linguagem. Se contraposto a versão anterior, CSS 2.1, o CSS3 acrescentou distintas propriedades que propiciaram a estilização de *Web sites*, fazendo um conteúdo assaz abastado em recursos visuais. O CSS3 concretizou os perfis (*profiles*) para plataforma mobile, TVs e impressões, tais deixam que seja definido os estilos individuais de cada perfil.

## 2.4. HTML

O HTML (*Hyper text Markup Language*) é um agrupamento de instruções, ilustradas por *tags*, que orientam o browser (programa que possibilita ao usuário interagirem com documentos HTML hospedados) na publicação da página *web*, por tanto, o *browser* interpreta as *tags* e desenha a página da *interface*. As instruções estão acopladas em ficheiros de tipo texto, sem qualquer tipo especial de formatação (CALDEIRA, 2015).

Todas as páginas presentes na *Internet* possuem HTML, por ser a estrutura básica dela. Proveniente do SGML (*Standard Generalized Markup Language*) o HTML foi originado no início da década de 1990 com a finalidade de tornar possível o acesso e troca de informações e pesquisas entre universidades, em sua aparição a notável inovação foi a implementação da *tag* `<a>`, concedendo a referência de links entre os documentos.

A partir de 1999, o desenvolvimento da linguagem HTML ficou estagnada na versão 4. Desde então, a W3C esteve concentrada em linguagens como XML e SVG. Ao mesmo tempo, os navegadores ficaram receosos em desenvolver suas funcionalidades, como exibir páginas em abas e disponibilizar a integração com leitores de RSS (SARTI, 2009). Em 19 dezembro de 1999, foi divulgado a versão 4.01

do HTML como uma indicação do W3C. E só em janeiro de 2008 o HTML5 foi anunciado como um novo projeto do W3C.

Em sua última atualização, o HTML5 tornou-se uma especificação avançada em nível de simplicidade, extensibilidade e foco na aplicação. Prove recursos para o uso do CSS e *JavaScript*, novas *tags* vieram a ser incluídas e outras tiveram suas funções alteradas. Com base em suas novas API tornou-se possível modelar as características destes elementos de modo que o *Web site* permaneça leve e funcional. *Canvas*, *localStorage*, geolocalização e *WebGL* são correspondentes dos componentes que consistem no HTML5.

Uma nova funcionalidade inserida é a alternativa de transformar o conteúdo das páginas mais semântico, liberando aos motores de busca, o entenderem a este conteúdo e o indexarem corretamente. As versões antecedentes ao HTML5 não têm um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, *sidebar* e menu, bem como não tinha um diretriz de nomenclatura de *IDs*, classes ou *tags*. Com a utilização das novas 20 *tags header, footer, section, aside, nav* e *article* é capaz de transformar o código HTML mais semântico e "legível" aos mecanismos de busca, assim a indexação será feita de uma maneira mais inteligente e efetiva (PILGRIM, 2010).

A ideia de hipertexto que o HTML traz consigo contém todas as características principal da página, e a possibilidade de interligar páginas *webs*. Ou seja, o hipertexto são os *links* disponibilizados nas páginas dos *sites*, no qual, já nos habituamos a visitar nos momentos de navegação na *internet* (SILVA,2008).

## **2.5. Javascript**

*Javascript* é a linguagem de programação para *web*. Na qual grande parte dos *sites* na atualidade utilizam esta linguagem nos navegadores em jogos, *tablets*, computadores etc. (FLANAGAN, 2004) Sua primeira versão foi lança em 1995 e implementada em 1996, desenvolvido pela *Netscape* em colaboração com a *Sun Microsystems*, a fim de trazer um meio de aplicar interatividade a uma página *web*.



Esta linguagem é de alto nível, dinâmica, interpretada e não tipada, sendo oriunda do *JAVA*, ela trouxe para as páginas *web* o estilo da programação voltada a objeto e suas funcionalidades (SILVA, 2010).

Apesar de ser oriunda do *JAVA* o *Javascript* tem algumas de suas funções baseadas em protótipos de *Self* (linguagem voltada a orientada a objeto, que se baseia no conceito de prototipagem) (SILVA, 2010). Atualmente ela é responsável por um grande acervo deveres, que variam de requisições assíncronas de obtenção de dados a alterações dinâmicas de CSS (ANJOS, 2017).

## 2.6. PHP

O PHP (sigla que é um acrônimo recursivo para *Hypertext Preprocessor*) é uma linguagem de criação de *scripts* voltada à *Internet* que permite o desenvolvimento de *sites* dinâmicos (NIEDERAUER, 2017), ela foi criada para permitir o pré-processamento das páginas HTML. Por tanta, o PHP pode modificar o conteúdo da página *web* antes de encaminhá-la ao navegador (BENTO, 2021).

É uma das linguagens mais usadas na criação de páginas *web*. Tendo milhões de aplicações em *sites* de todo o globo (NIEDERAUER, 2011), isso se dá por conta de alguns fatores, conforme apontado por Chaves, 2008: (1) desenvolvimento de software tanto web quanto desktop; (2) ser gratuito e com código-fonte aberto; (3) ter compatibilidade com a maioria dos sistemas operacionais; (4) criação de páginas dinâmica; (5) ter suporte para grande parte dos atuais servidores *web*; e (6) dispor de suporte para diversos bancos de dados.

Tais fatores trazem uma praticidade para vida do desenvolvedor, como diz Niederauer (2011), imagina fazer a atualização de cem páginas manualmente, seria muito trabalhoso e demorado.

## 2.7. Node.js

Segundo Pereira (2014), o *Node.js* foi criado por Ryan Dahl com a ajuda inicial de 14 colaboradores. O *Node.js* tem um modelo orientado a eventos que utiliza o

acionamento de funções de chamada depois da conclusão de uma tarefa ou geração de erro (SHAH, 2017). Possuindo um sistema inovador: sua arquitetura é completamente *non-blocking thread* (não bloqueante). Se sua aplicação trabalha com processamento de arquivos 1/0, adotar esse tipo de arquitetura vai resultar em uma boa performance com relação ao consumo de memória, pois usa o máximo e de forma eficiente o poder de processamento dos servidores (PEREIRA, 2014). Sistemas que fazem uso do *Node.js* não são afetados por *dead-locks*, já que o *Node.js* trabalha apenas em *single-thread* (única *thread* por processo). Desenvolver sistemas nesse paradigma é simples e prático (PEREIRA, 2014).

O *Node.js* foi desenvolvido desde o começo com o objetivo de manuseio de *Input/Output (I/O)* de forma assíncrona (que não mantém sincronia), enquanto outras linguagens oferecem esse recurso no exterior dos seus núcleos de processamento, e dessa forma terminam por executar de modo mais lento que o *Node.js* (SHAH, 2017).

Por ser um ambiente no qual trabalha-se em tempo de execução baseando-se no interpretador V8 do *Google* que habilita o desenvolvimento de aplicações escaláveis de rede, em linguagem *JavaScript*. Nas aplicações *Node.JS*, várias conexões podem ser controladas ao mesmo tempo. Se houver uma conexão, uma função de retorno é chamada. Caso contrário, o *Node.js* fica inerte. No *Node*, não há impasses de processos, já que o *Node.js* raramente utiliza operações diretas de entrada e saída (NODEJS, 2022). Dүүna (2016) ressalta alguns atributos importantes do *Node.js* no ponto de vista de segurança. Entre elas o fato de o *Node.js* receber e interpretar 16 requisições em apenas uma *thread* (ou tarefa), sendo assim, há apenas um ponto de entrada e saída de eventos, chamado de *loop* de eventos (*event loop*). Essa distinção é demasiadamente importante para segurança da informação, principalmente na condição de disponibilidade, pois se algo bloquear o *loop* de eventos, a aplicação não poderá mais servir os usuários, gerando uma negação de serviço.

## 2.8. Frameworks

São ferramentas genéricas que visam solucionar problemas comuns no processo de desenvolvimento de *software*. Sendo abstrato e unindo códigos comuns

no interior de vários problemas comuns. Utilizando de configurações o *framework* obtém uma funcionalidade específica. Ao contrário de bibliotecas é capaz de articular o fluxo da aplicação, por meio das configurações pré-estabelecidas. Concedendo o reuso de uma arquitetura de aplicação pré-definida e forcem componentes que assessoram na preparação do *software* (SOMMER, 2012).

O emprego de um *framework* destaca o reuso de projetos em objeção a reutilização de código, assim entendesse que um conjunto de classes implementadas em uma linguagem específica.

Nesta circunstância, os *frameworks* são repartidos em dois conjuntos: verticais que são utilizados em domínios usuais entre inúmeros projetos, como instância e interfaces gráficas. E horizontais, são chamados de especialistas, criados com base em um problema diante a um domínio específico, complementando funcionalidades específicas de um domínio.

Existem diversos *frameworks* à disposição, fornecendo ferramentas que ajudam nos afazeres de processo de desenvolvimento de *software*, sendo considerável expor a auxílio das comunidades desenvolvidas em prol destes projetos, que cooperam com a compartilhando e disseminando das informações, ocasionando e ampliando mais a comunidade. E como consequência o desenvolvedor possui um vasto número de ferramentas para auxiliar na construção de aplicativos (ROSÁRIO,2015).

### **2.8.1. Laravel**

O *Laravel* é um *framework* PHP desenvolvido por Taylor Otwell no ano de 2011, para a finalidade de facilitar a concepção de projetos de modo rápida tornando o código mais limpo e com uma boa aprendizagem, facilitando assim manutenções. Este *framework* possui arquitetura MVC (*Model-View-Controller*) e possui, como atributo principal, o papel de auxiliar no desenvolvimento de aplicações de forma ágil e simplificada incentivando a boas práticas de programação e utilização de padrões. (STAUFFER, 2017).

Com a finalidade de gerenciar as dependências o *Laravel* dispõe do *composer*, uma ferramenta desenvolvida por Nils Adermann e Jordi Boggiano para

gerenciamento de dependências em PHP. O *composer* permite que o desenvolvedor informe as bibliotecas que precisão em seu projeto e as instalará e atualizará. O *composer* foi inspirado no NPM e *Bundler*, no entanto, ele gerencia cada pacote ou biblioteca em uma base por projeto, instalando-os em um diretório chamado *vendor* dentro do projeto (ADERMANN & BOGGIANO, 2022).

O *Laravel* aliás contém uma ferramenta de linha de comando chamada de *Artisan*, um agrupamento de ações internas com a disponibilidade de inclusão de outras. Com o *artisan*, o desenvolvedor consegue acionar ações, como testes de unidade, tarefas agendadas e execução de migrações do banco de dados (MCCOOL, 2012).

O *Laravel* detém de uma estrutura de arquivos e diretórios projetados para exibição das páginas do *site*. Num arquivo de rotas (*/routes/web.php*), as URLs do *site* são especificadas e mapeadas em *closures* ou para métodos definidos nos controladores. Bem como nos arquivos de rotas é concebível definir os *Middleware*, elementos que o *Laravel* utiliza para autenticar usuários e acesso às funções que eles tenham permissão (STAUFFER, 2017).

Os controladores são *contêiners* para métodos que abrange a lógica da aplicação. As *closures* ou os métodos implementados nos controladores são incumbidos por recepcionar as requisições HTTP recebidas pelo servidor *web*. O código dos controladores é escrito na linguagem PHP e além de controlar o fluxo da aplicação, abastecera dados do banco de dados para a visualização nas *views* (MCCOOL, 2012).

Para exposição do conteúdo, o *Laravel* aplica *Views*, onde o conteúdo de uma página é descrito operando um motor de *templates* chamado *Blade*. Este *template engine* customizado pelo *Laravel* foi inspirado no *engine Razor* da plataforma .NET e contém uma sintaxe limpa e precisa, além de conter um modelo de herança espontâneo e acessível extensão. Os arquivos das páginas encontram-se guardados no diretório *resources/views/* e devem ter a extensão *blade.php*, tais arquivos são compilados em PHP e armazenados em cache, fazendo-os deste modo mais rápidos e possibilitando também o inserimento de códigos PHP nativos nos arquivos *Blade* se lhe apetecer (STAUFFER, 2017, p. 40-79).

Neste sentido o *framework Laravel* dispõe de definições de estrutura do banco de dados facilitando-o através do arquivo de migrações. O vigente arquivo pode ser criado através de um comando da CLI (*Command Line Interface*) e nele consegue-se definir cada nova tabela, coluna, índice, tipo do dado, tudo em um breve período. O *Laravel* utiliza três formas de manipulação de Banco de Dados, sendo, duas delas utilizando a *Facade DB: Raw SQL* (SQL Bruto), e a *Fluent Query Builder* com o emprego através de métodos singulares em qualquer transação requerida no BD. A um modo distinto de manipulação do banco de dados que é através do *Eloquent*, um dos mais populares recursos do *Laravel*. O Eloquent é um ORM (*Object-Relational Mapping*) *ActiveRecord* indicando que ele é uma camada de abstração do banco de dados e dispõem de uma interface única para a interação com o BD. Isto quer dizer que com base em uma classe do *Eloquent* é possível fornecer a interação com a tabela inteira ou até representar uma linha individual da tabela. Tal se baseia no modelo *convention over configuration* para assentir a elaboração de modelos com código mínimo e com grande poder nas funcionalidades. Nisto o ORM oferece ênfase a simplicidade (STAUFFER, 2017, p. 158-190).

### 2.8.2. *Bootstrap*

Criado em 2011 o método *Bootstrap* é muito útil por não precisar de muitas suposições para estimação de parâmetros das distribuições de benefícios (RIZZO, 2006). É um *framework* para desenvolvimento *front-end*, desenvolvido em CSS e *Javascript*, viabilizando um elegante design as suas páginas HTML de forma descomplicada, rápida e prática (DE CERQUEIRA, 2014).

Possuindo um conjunto de ferramentas de código públicos para desenvolvimento de *sites web* (MATOS, 2020). Seu principal objetivo é assessorar no desenvolvimento de sites amigáveis e responsivos. Ademais, oferece uma grande abundância de *plug-ins*, apresenta temas compatíveis com diversos outros *frameworks* e tem suporte para todos os navegadores. Também fornece extensibilidade através do uso de *JavaScript*, com assistência interno para *plug-ins jQuery* e uma API *JavaScript*. O *Bootstrap* é capaz de ser utilizado com qualquer IDE ou editor, como também com qualquer tecnologia ou linguagem *back-end* como ASP.NET, PHP, *Ruby on Rails* ou *Python* (SANTIAGO, 2020). E, é considerável

destacar que, é indubitável que os elementos selecionados não conflitarão entre si e estarão totalmente responsivos e operando tanto para *desktop* quanto para quaisquer outros formatos de tela. Sendo como um quebra-cabeças, com exceção de que cada peça se encaixa igualmente com quaisquer outras peças escolhidas (RIZZO, 2006).

### **2.8.3. Firebase**

O *Firebase* é um de produto gratuito, com um limite de utilização (FIREBASE, 2017). Dentro existem serviços de hospedagem, armazenamento em nuvem e banco de dados. No caso do banco, a utilização gratuita autoriza até 100 acessos simultâneos.

Além da utilização do banco de dados, o *Firebase* disponibilizada outras ferramentas muito utilizadas para o desenvolvimento de aplicativos, e para esse projeto, é utilizado o sistema de *Cloud Messaging* (Mensagem das nuvens). Este sistema pode ser usado de diversas maneiras, mas para o objetivo deste trabalho, foi desenvolvida uma função em Node.js que dispara mensagens que irá aparecer em todos os dispositivos de acesso administrativo, alertando sobre as chamadas (MEZZARI, 2019).

### **2.8.4. Express.js**

De acordo com STRONGLOOP (2015) o *Express* é um *framework* utilizado na construção de aplicações *web*, flexível e pequeno para *Node.js*. Com o *Express* é possível instanciar servidores *web* e receber requisições HTTP de modo trivial, ainda permiti o desenvolvimento de um conjunto de diretórios com uma arquitetura padrão, e também gerencio na as rotas dos arquivos para as *views* da aplicação (BONFIM, 2014). Com uma abundância de métodos utilitários HTTP e *middleware* (é o *software* que fornece serviços e recursos comuns a aplicações) à sua disposição, é possível desenvolver uma API robusta, de maneira rápida e fácil (STRONGLOOP 2015).

## **2.9. Tipos de Manutenção**

De acordo com Kardec & Carvalho (2002) a forma na qual é realizado o tratamento nos equipamentos, sistemas ou instalações equivalem a uma série de tipos de manutenção. Há inúmeras formas de denominações para catalogar a atuação da manutenção.

Os tipos de manutenção mais habituais e aplicáveis para as atividades prediais e de equipamentos são a (1) manutenção corretiva não planejada; (2) manutenção corretiva planejada; (3) manutenção preventiva; e (4) manutenção preditiva.

### **2.9.1. Manutenção Corretiva**

Manutenção corretiva é a mais simples. Trata-se do ato de concerta o que se quebrou, está inoperante, em outras palavras essa manutenção acarreta da suspensão das funções do equipamento (NETO, 2002).

A manutenção corretiva pode ser dividida em das classes: Manutenção Corretiva Não Planejada; e Manutenção Corretiva Planejada.

A manutenção corretiva não planejada é aquela onde a falha ou o desempenho aquém do esperado já ocorreu e traz como problema os altos custos, porque geralmente causa perdas de produção, qualidade e custos indiretos de manutenção (TAVARES, 1999).

A manutenção planejada é sempre mais barata que a não planejada. Por mais que se deixe o equipamento funcionar até a quebra, o impacto pode ser mais bem administrado. A manutenção corretiva planejada traz como benefícios: (1) intervenções com menores impactos na operação; (2) diminuição de acidentes; (3) tempo para aquisição do material necessário.

### **2.9.2. Manutenção Preventiva**

A manutenção preventiva tem como princípio a redução percentual das falhas, além de atuar de forma a se evitar a ocorrência destas e eventuais quedas de rendimento através de listas de verificação (*check list*) inseridas em um cronograma periódico (PINTO, 2003).

A periodicidade deste tipo de manutenção pode ser de diferentes formas: por ciclos de tempo (semanal, bimestral etc.), valores definidos em horímetros ou contadores, paradas programadas nos equipamentos, entre outras mais.

Basicamente são realizados ajustes e substituições de peças, mesmo que às vezes sem necessidade.

Os intervalos entre as manutenções podem ser ajustados por meio de análises no histórico das preventivas, estes ajustes podem tornar a manutenção mais assertiva, evitando-se desperdícios (SLACK, CHAMBERS e JOHNSTON, 2002).

Para Pinto e Xavier (2003), a manutenção preventiva traz os seguintes benefícios: garante o funcionamento contínuo de máquinas, funcionamento este, interrompido somente em horas de manutenção programada; ajuda no cumprimento das metas de produção ou tempo de operação ininterrupta; e torna possível a previsibilidade de consumo de materiais e sobressalentes.

### **2.9.3. Manutenção Preditiva**

A definição de manutenção preditiva consiste na análise do melhor momento para uma intervenção via manutenção preventiva no equipamento, ou seja, o ponto onde a probabilidade de falha no equipamento aumenta (TAVARES, 1996).

Dentre os exemplos mais comuns deste tipo de manutenção estão as análises de viscosidade do óleo em transformadores e da vibração dos rolamentos de motores. Esta prática faz com que as ações sejam antecipadas evitando às falhas, aumentando a disponibilidade dos equipamentos e reduzindo os custos com manutenção (PINTO E XAVIER, 2003).



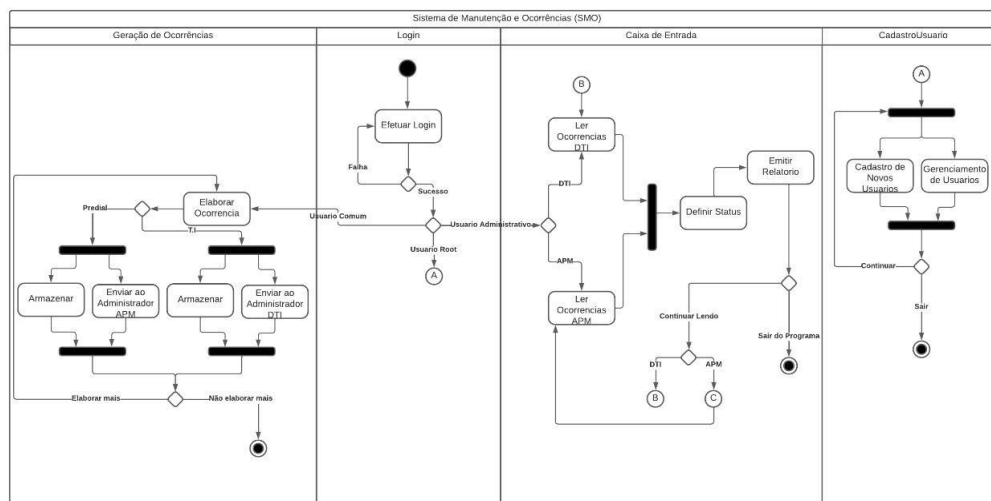
### 3. METODOLOGIA

A situação problemática a ser entendida nesta pesquisa é o processo de manutenção e ocorrência utilizado atualmente nas edificações do centro de ensino ETEC Zona Leste, propondo a implementação de uma ferramenta de gestão.

Seguindo uma abordagem estimatória, quantitativa e de pesquisa, no qual a função do pesquisador na atual situação, segundo Ganga (2012) foi adquirir informações através a visão dos indivíduos, do mesmo jeito que observando e coletando evidências que permitem elucidar o ambiente em que a problemática sucede e proceder com os levantamentos bibliográficos que excitam o e do problema.

Para tal foi feita uma observação do método que está em utilização, com o objetivo de compreender e extrair os diagramas (conforme a Figura 1, 2 e 3), dessa foram poder realizar de modo efetivo o plano de melhoria.

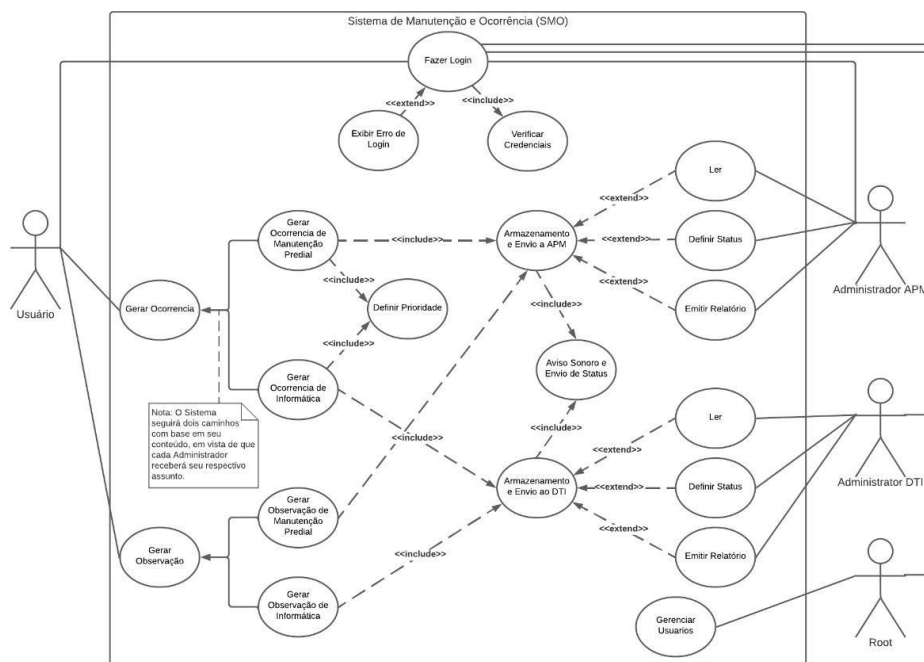
*Figura 1 - Diagrama de Atividades*



**Fonte:** Autoria Própria (2022)

A Figura 1 trata do diagrama de atividades para modelagem do sistema que será desenvolvido. E sua função é ilustrar uma sequência de atividades que foram modeladas a fim de representar de forma dinâmica os aspectos de um processo computacional (SILVA, 2014).

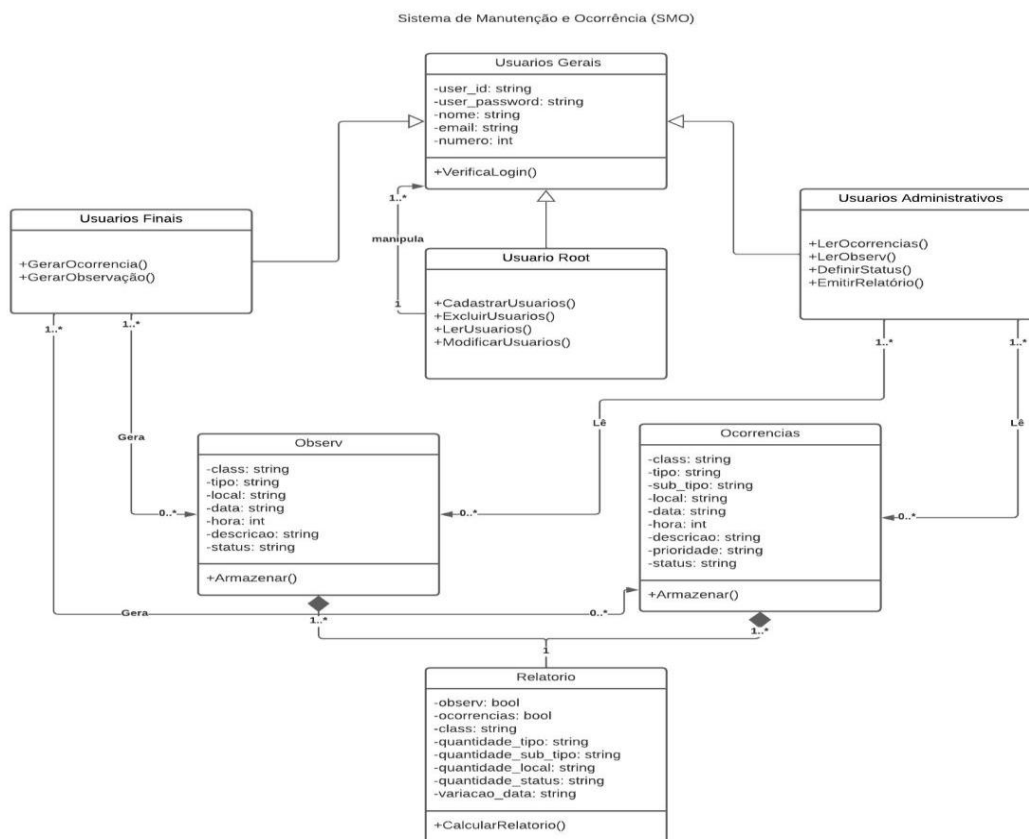
**Figura 2 - Diagrama de Caso de Uso**



**Fonte:** Autoria Própria (2022)

A Figura 2 é o modelo de caso de uso, que é a representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema e, que com ele interagem (BEZERRA, 2007);

**Figura 3 - Diagrama de Classe**



**Fonte:** Autoria Própria (2022)

A figura 3 mostram a estrutura estática dos objetos em um sistema. Segundo Kollmann (2002) e outros, “são geralmente considerados os diagramas mais bem empregados e os mais bem compreendidos dentre os diagramas da UML”.

#### 4. RESULTADOS E DISCUSSÃO

As Figuras 4 a 14, tratam dos protótipos do sistema desenvolvido, para fim de ilustrar como o sistema será, mostrando sua aparência. Para tornar mais fácil a compreensão de como o sistema será para o usuário.

**Figura 4 - Protótipo da tela de login**

SMO

Faça Seu Login

Insira seu AD

Enviar



**Fonte:** Autoria Própria (2022)

**Figura 5 - Protótipo Tela Inicial**

SMO

Departamento T.I	Predial	Ocorrências
		
Solicite apoio técnico em problemas de informática, sejam computadores, periféricos, televisores, projetores, internet e outros.	Crie chamados para questões de manutenção na ambiente predial, envolvendo problemas hidráulicos, elétricos, estruturais e de limpeza.	Notifique questões que considere incorretas ou que necessitam de atenção de seus superiores no seu dia a dia.

**Fonte:** Autoria Própria (2022)

**Figura 6** - Protótipo Tela de solicitação de Manutenção DTI



SMO

[Manutenção Predial](#) [Manutenção DTI](#) [Registro de Ocorrências](#)

## Manutenção DTI

Selecione o Setor ▼

Tipo de Manutenção ▼

Selecione a Prioridade ▼

Informe local

Descrição

Enviar

**Fonte:** Autoria Própria (2022)

**Figura 7** - Protótipo Tela de Solicitação de Manutenção Predial



SMO

[Manutenção Predial](#) [Manutenção DTI](#) [Registro de Ocorrências](#)

## Manutenção Predial

Selecione o Setor ▼

Tipo de Manutenção ▼

Selecione a Prioridade ▼

Informe local

Descrição

Enviar

**Fonte:** Autoria Própria (2022)

**Figura 8 - Protótipo Tela de Formulário de Ocorrência**

O protótipo da tela de formulário de ocorrência apresenta uma interface com uma barra superior azul contendo o logotipo 'SMO' e três links de navegação: 'Manutenção Predial', 'Manutenção DTI' e 'Registro de Ocorrências'. O título principal da seção é 'Ocorrências'. Abaixo dele, há dois menus suspensos para 'Selecione o Setor' e 'Selecione o Tipo'. Segue-se um campo de texto rotulado 'Informe local' e um campo de texto maior rotulado 'Descrição'. No canto inferior direito, há um botão amarelo com o texto 'Enviar'.

**Fonte:** Autoria Própria (2022)

**Figura 9 - Protótipo Tela Inicial Administrador DTI**

O protótipo da tela inicial do administrador DTI possui uma barra superior azul com o logotipo 'SMO'. A interface é dividida em duas colunas principais. A coluna da esquerda, intitulada 'Chamados', contém uma imagem de uma reunião e o texto: 'Cheque as mensagens de solicitação enviadas a você pelos usuários, referente ao Departamento de Tecnologia da Informação(DTI)'. A coluna da direita, intitulada 'Ocorrências', contém uma ilustração de uma pessoa lendo documentos com uma lupa e o texto: 'Cheque as mensagens enviadas a você pelos usuários, notificando ocorrências que envolvam o Departamento de Tecnologia da Informação(DTI)'.

**Fonte:** Autoria Própria (2022)

**Figura 10 - Protótipo Tela de Chamadas de Manutenção DTI**

SMO							
				<a href="#">Caixa de Entrada Manutenção DTI</a>		<a href="#">Caixa de Entrada Ocorrências DTI</a>	
ID	Urgência	Setor	Local	Manutenção	Relatório	Data	Status
1	Alta	Bloco_B	Laboratorio 5.	Teclado	A bia pegou o teclado e agrediu sucessivamente o colega de classe Joao.	2022-04-08 22:35:14+00	
2	Média	Anexo	dasdasd	Mouse	sdasdas	2022-04-08 22:47:43+00	
3	Baixa	Bloco_A	Sala 02.	Janelas	Cortina Rasgada.	2022-04-25 19:37:51+00	
4	Média	Administração	Lab 9 Jurídico	Computador	Precisa habilitar shadow	2022-04-26 00:46:02+00	

**Fonte:** Autoria Própria (2022)

**Figura 11 - Protótipo Tela de Ocorrências DTI**

SMO				
		<a href="#">Caixa de Entrada Manutenção DTI</a>		<a href="#">Caixa de Entrada Ocorrências DTI</a>
ID	Setor	Local	Relatório	Data
3	Biblioteca	dasdasd	asdasdasd	2022-04-08 22:47:57+00

**Fonte:** Autoria Própria (2022)

**Figura 12** - Protótipo Tela Inicial Administrador Predial**Fonte:** Autoria Própria (2022)**Figura 13** - Protótipo Tela de Chamados de Manutenção Predial

SMO		<a href="#">Caixa de Entrada Manutenção Predial</a>				<a href="#">Caixa de Entrada Ocorrências Predial</a>	
ID	Urgência	Setor	Local	Manutenção	Relatório	Data	Status
1	Alta	Administração	Secretaria.	Elétrica	Sem luz.	2022-04-08 18:08:40	
2	Baixa	Administração	Sala do Diretor.	Elétrica	A bia queimou a lampada.	2022-04-08 22:34:25	
3	Média	Bloco_A	dasdasd	Portas	asdasdads	2022-04-08 22:47:27	

**Fonte:** Autoria Própria (2022)



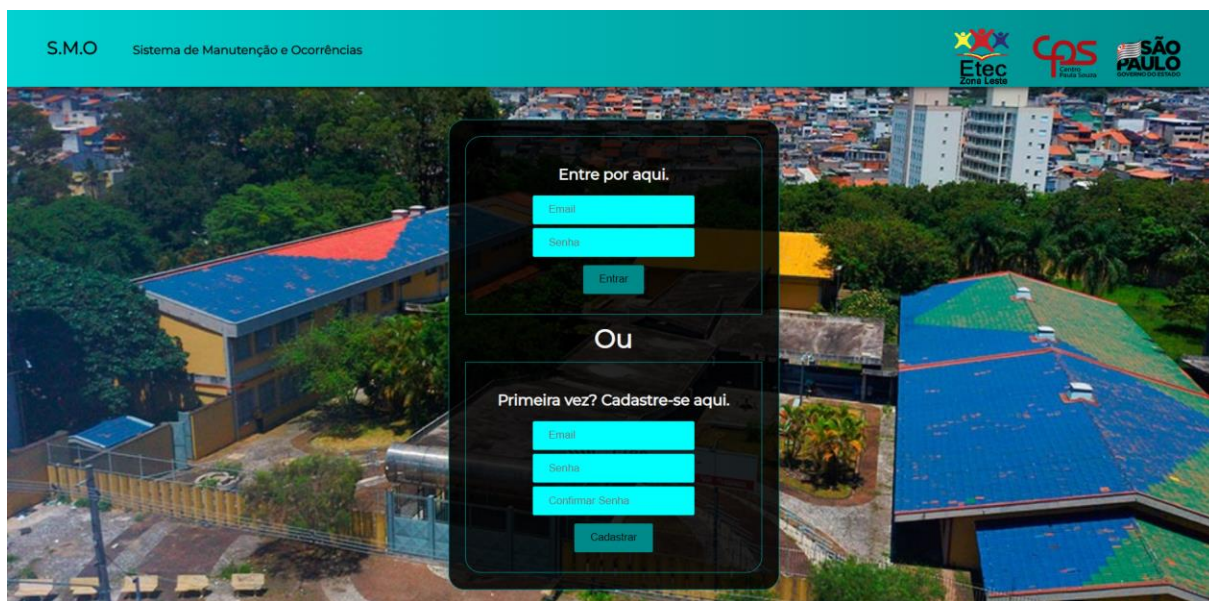
**Figura 14 - Protótipo Tela de Ocorrências Predial**

ID	Setor	Local	Relatório	Data
1	Patio	Cantina	A bia trouxe uma marreta, e bateu no pilar.	2022-04-08 22:35:47+00
2	Anexo	dasdasd	sadasddas	2022-04-08 22:47:51+00

**Fonte:** Autoria Própria (2022)

#### 4.1. Modelo Pronto

Para o projeto final foi realizada alterações no designer para melhorar estéticas. De maneira objetiva e utilizando-se do princípio da psicodinâmica das cores, foram aplicados detalhes em cor azul, um fundo com elemento gráfico referente à funcionalidade do sistema, além de caixas de acesso para adição textual do login e da senha do usuário. Uma vez inseridas as informações necessárias, estas serão verificadas pela central através de um clique no botão, como mostrado na Figura 15 e Figura 16.

**Figura 15 - Tela Inicial**

**Fonte:** Autoria Própria (2022)

Segundo Farina, Clotilde Perez (2006) as cores são um tipo de código fácil de entendimento e assimilação, por conta disso, podem e devem ser usadas estrategicamente como um instrumento didático. Assim sendo uma parte do entendimento da psicodinâmica das cores, que, neste estudo em específico, vinculada aos princípios de unificação, segregação e continuidade, tornou possível a constituição de uma alternativa com qualidade. A psicodinâmica discorre da relação comportamental humana em interações conscientes e inconscientes, num associamento ao design e à psicologia das cores, contribui, de maneira relevante, para a experiência do usuário.

**Figura 16** - Tela Inicial mobile



The image shows a mobile application interface for login and registration. At the top, there is a teal header bar containing four logos: 'S.M.O', 'Etec Zona Leste', 'CPS', and 'SÃO PAULO'. Below the header, the background is a photograph of a cityscape. The main content is a dark, semi-transparent overlay with rounded corners. It features two sections: a login section at the top and a registration section at the bottom, separated by the word 'Ou' (Or). The login section is titled 'Entre por aqui.' and contains two white input fields labeled 'Email' and 'Senha', followed by a teal 'Entrar' button. The registration section is titled 'Primeira vez? Cadastre-se aqui.' and contains three white input fields labeled 'Email', 'Senha', and 'Confirmar Senha', followed by a teal 'Cadastrar' button.

**Fonte:** Autoria Própria (2022)

## 5. CONCLUSÃO

Levando em conta o processo metodológico desta pesquisa, primeiramente, cumpriu-se com a proposta de geração de conteúdo, em virtude de que foi estabelecida a problemática de forma clara e sucinta, o que se observou-se pelo desenrolamento da justificativa e dos objetivos de pesquisa. Procurou-se o aprofundamento literário por via de uma análise do referencial teórico, de modo a

agrupar e investigar os questionamentos relacionados ao problema da pesquisa e aos objetivos esperados.

Com fundamento nas referências respectivas, notou-se a viabilidade do estudo de maneira mais vasta e o quanto a pesquisa poderia agregar tanto para a instituição em questão – ETEC Zona Leste - quanto para os seus usuários finais.

À vista disso, depois da obtenção das informações necessárias na primeira parte desse estudo, partiu-se para a preparação dos materiais que constituem as etapas de coleta de dados. Foram selecionadas as melhores práticas para desenvolvimento e aplicabilidade com foco na usabilidade. Após o preparo dos materiais e pesquisa foram aplicados os instrumentos de definidos.

Tornando-se admissível elaborar e propor uma solução viável para o problema identificado. O sistema proposto, embora ainda esteja em processo de aperfeiçoamento, possuirá usabilidade simples, ágil e eficiente, o que o torna muito relevante no contexto abordado, considerando as necessidades dos usuários finais. Com embasamento na pesquisa produzida, pode-se afirmar que os objetivos definidos no início da pesquisa foram alcançados.

## REFERÊNCIAS

MORAES, Carmen Sylvia Vidigal; REIS, Elydimara Durso dos; ALENCAR, Felipe. **Educação profissional paulista e relações público-privadas na política curricular: Centro “Paula Souza” (1995-2018)**. Revista Brasileira de Educação, v. 27, 2022.

KALIL, M. L.; PUJADAS, F. Z. A. Acidentes prediais: riscos do descaso com a manutenção. In: PINI, M. S. (Org.). **Manutenção Predial**. São Paulo: Pini, 2011, p. 146 – 150.

BRASIL. Tribunal de Contas da União. **Acórdão Plenário 504/2013 – Avaliação do Desempenho da Gestão de Compras e Contratos de Manutenção da Universidade Federal de Pernambuco**. Brasília, 13 mar. 2013. Disponível em: <[https://pesquisa.apps.tcu.gov.br/#/documento/acordaocompleto/\\*/NUMACORDAO%253A504%2520ANOACORDAO%253A2013/DTRELEVANCIA%20desc,%20NUMACORDAOINT%20desc/0/%20?uuid=5c118830-6886-11e9-9a6b7158c8a86983](https://pesquisa.apps.tcu.gov.br/#/documento/acordaocompleto/*/NUMACORDAO%253A504%2520ANOACORDAO%253A2013/DTRELEVANCIA%20desc,%20NUMACORDAOINT%20desc/0/%20?uuid=5c118830-6886-11e9-9a6b7158c8a86983)> Acesso em 08 de jun. 2022.

SCHWARTZMAN, Simon. **O Centro Paula Souza e a educação profissional no Brasil. Educação Básica no Estado de São Paulo: avanços e desafios**. São Paulo: Fundação Seade, p. 187-216, 2014.

TREVISAN, Leonardo; VELOSO, Elza. **Gestão de competitividade e políticas públicas de formação de mão-de-obra: o caso Centro Paula Souza**. Revista de Administração Pública, v. 41, p. 887-908, 2007.

MILETTO, Evandro Manara; DE CASTRO BERTAGNOLLI, Silvia. Desenvolvimento de Software II: Introdução ao **Desenvolvimento Web com HTML, CSS, JavaScript e PHP-Eixo: Informação e Comunicação-Série Tekne**. Bookman Editora, 2014.

CALDEIRA, Carlos. **Introdução ao HTML**. 2015.

**GOVERNAMENTAL.** CPS: Etec da Zona Leste (Cidade A. E. Carvalho). Disponível em: <<https://www.cps.sp.gov.br/etecs/etec-da-zona-leste-cidade-a-e-carvalho/>>.

Acesso em: 28 de abril de 2022.

SILVA, Maurício Samy. **Criando sites com HTML: sites de alta qualidade com HTML e CSS.** Novatec Editora, 2008.

TOMAZINI, Marcos; LOPES, Luiz Fernando Braga. Web design responsivo-Bootstrap.

SIGNIFICADO. **Significado de website** (O que é. Conceitos. Definição). Disponível em: <https://www.significados.com.br/website/>. Acesso em: 28 de abril de 2022.

FLANAGAN, David. **JavaScript: o guia definitivo.** Bookman Editora, 2004.

SILVA, Maurício Samy. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript.** Novatec Editora, 2010.

HAVERBEKE, M. **Eloquent JavaScript.** Disponível em <[http://eloquentjavascript.net/Eloquent\\_JavaScript.pdf](http://eloquentjavascript.net/Eloquent_JavaScript.pdf)>, Acesso em 03 maio 2015.

ANJOS, Luiz Felipe Rosa dos. **Evolução do Javascript em aplicações multiplataforma.** 2017.

CREMONINI, R. A. **Uso de Levantamento de Campo como Subsídios para Programação da Manutenção de Edifícios.** In: SEMINÁRIO SOBRE MANUTENÇÃO DE EDIFÍCIOS, 1, 1988, Porto Alegre. Anais... Porto Alegre: UFRGS, 1988.

SOOMER, A. Comparison and evaluation of cross-platform frameworks for development of mobile business applications. <[https://www.bruegge.informatik.tumuenchen.de/lehstuhl\\_1/research/paper/sommer2013crossplatform.pdf](https://www.bruegge.informatik.tumuenchen.de/lehstuhl_1/research/paper/sommer2013crossplatform.pdf)>, Acesso em 28 abril.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5674:** Manutenção de

Edificações – Requisitos para o sistema de gestão da manutenção, 2012.

TANENBAUM, Andrew S. **Redes de computadores**. Pearson educação, 2003.

PILGRIM, Mark. **HTML5: up and running: dive into the future of web development**. " O'Reilly Media, Inc.", 2010.

SARTI, Erika. 2009. Introdução ao HTML 5. Disponível em <<http://www.infowester.com/introhtml5.php>>. Acesso em 03 de Maio de 2022.

STARK, Jonathan et al. **Building Android apps with HTML, CSS, and JavaScript**. " O'Reilly Media, Inc.", 2010.

FANTINI, Sérgio Rubens et al. **Aplicação do gerenciamento eletrônico de documentos: estudo de caso de escolhas de soluções**. 2001.

DURANGO, Alicia. **Diseño Web con CSS: 2ª Edición**. IT Campus Academy, 2015.

RIZZO, Ana Lucia Tucci; CYMROT, Raquel. Estudo e aplicações da técnica bootstrap. **II Jornada de iniciação científica. Universidade Presbiteriana Mackenzie, São Paulo**, 2006.

DE CERQUEIRA, Jackson Hélio. **Desenvolvimento de um Portal Web para Viajantes Utilizando Ruby on Rails e Bootstrap**, 2014.

MATOS, Ecivaldo; ZABOT, Diego. **Aplicativos com Bootstrap e Angular: Como Desenvolver Apps Responsivos**. Saraiva Educação SA, 2020.

SANTIAGO, Cynthia Pinheiro et al. Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap. **Sociedade Brasileira de Computação**, 2020.

ADERMANN, N. ; BOGGIANO, J. **Composer**. Disponível em: <<https://getcomposer.org/>>. Acesso em: 12 maio 2022.

ROSÁRIO, Fábio Chassot do. **Desenvolvimento de aplicativos móveis multiplataforma**. 2015. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.o

MCCOOL, S. **Laravel Starter<sup>a</sup> Edição. 1th**. Packt Publishing, 2012.

BRASIL. **Lei nº 9.394**, de 20 de dezembro de 1996. Lei de Diretrizes e Bases da Educação Nacional. Diário Oficial da União, Brasília, DF, 24 dez. 1996. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/Leis/L9394.htm](http://www.planalto.gov.br/ccivil_03/Leis/L9394.htm)>. Acesso em: 12 maio 2022.

NIEDERAUER, Juliano. **Desenvolvendo Websites com PHP: Aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados**. Novatec Editora, 2017.

STAUFFER, M. **Desenvolvendo com Laravel<sup>a</sup> Edição. 1th**. Novatec, 2017.

NIEDERAUER, Juliano. **Desenvolvendo Websites com PHP-2ª Edição: Aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados**. Novatec Editora, 2011.

BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL**. Editora Casa do Código, 2021.

CHAVES, Aline Martins; SILVA, Gabriel. proposta de uma arquitetura de software e funcionalidades para implementação de um ambiente integrado de desenvolvimento para a linguagem php. **I Jornada Científica e VI FIPA do CEFET Bambuí**, v. 31, p. 32-36, 2008.

KARDEC, Alan e CARVALHO, Cláudio. **Gestão Estratégica e Terceirização. Qualitymark, ABRAMAN**. Rio de Janeiro, 2002

TAVARES, L. A. **Administração Moderna de Manutenção**. Rio de Janeiro: Editora Novo Pólo, 1999.



TAVARES, L. A. **Excelência na Manutenção – Estratégias, Organização e Gerenciamento**. Salvador: Casa Da Qualidade Editora Ltda., 1996.

PINTO, Alan Kardec; XAVIER, Julio A. **Manutenção: função estratégica**. Ed. 2. Rio de Janeiro: Editora Qualitymark, 2003.

SLACK, N.; CHAMBERS S.; JOHNSTON R. **Administração da produção**. São Paulo. Editora Atlas 2ª Ed. 2002.

NETO, João Cirilo da Silva; GONÇALVES DE LIMA, A. M. **Implantação do Controle de Manutenção**. Revista Club de Mantenimiento, n. 10, 2002.

PEREIRA, Caio Ribeiro. **Aplicações web real-time com Node.js**. Editora Casa do Código, 2014.

SHAH, Hezbullah e Tariq Soomro: **Node.js challenges in implementation**. Global Journal of Computer Science and Technology: Network, Web and Security, 2017.

NODEJS. NodeJS. **About Node.js** Disponível em <<https://nodejs.org/en/about/>>. Acesso em 08 de jun. 2022.

DÜÜNA, Karl; RASHID, Fahmida Y. (Ed.). **Secure Your Node.js Web Application**. Raleigh: Pragmatic Programmers LLC, 2016.

FIREBASE. Firebase. 2017. **Documentação do Firebase**. Disponível em: <<https://firebase.google.com/docs/>>. Acesso em: 08 jun. 2022.

MEZZARI, Lucas Torres; LEAL, Eduardo Henrique Viva; VIEGAS, Silvio Cesar. INTERNET DAS COISAS: Arduino. Firebase e Android. **REFAQI-Revista Eletrônica em Gestão e Tecnologia**, v. 5, n. 1, p. 50-55, 2019.

GANGA, G. M. D. **Trabalho de Conclusão de Curso (TCC) na Engenharia de Produção**: um guia prático de conteúdo e forma. São Paulo: Atlas, 2012.

SILVA, Williamson; COSTA VALENTIM, Natasha M.; CONTE, Tayana. Projetando diagramas de atividade visando a usabilidade de aplicações interativas. In:

**Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems.** 2014. p. 349-352.

BEZERRA, Eduardo. **Princípios de análise e Projeto de sistemas com UML.** 2. ed., Rio de Janeiro: Elsevier, 2007.

R. Kollmann, P. Selonen, E. Stroulia, T. Systa, and A. Zundorf. **A study on the current state of the art in tool-supported UML-based static reverse engineering.** In Proceedings of the 9th Working Conference on Reverse Engineering, pages 22 – 32, November 2002.

Modesto Farina, Clotilde Perez, D. B. **Psicodinâmica das Cores em Comunicação.** Editora ed. São Paulo: [s.n.], 2006. 189 p.

StrongLoop, I. **Expressfast, unopinionated, minimalist web framework for node.js,** 2015.

BONFIM, F. L. and Liang, M. **Aplicações escaláveis utilizando mean stack.** Ciências Exatas da Universidade Federal do Paraná, 2014.