



Universidade Federal do ABC
Pós-Graduação em Tecnologias e Sistemas de Informação

DESENVOLVIMENTO DE UMA
FERRAMENTA PARA PADRONIZAÇÃO
DE RELATÓRIOS CIRCUNSTANCIADOS
UTILIZANDO O FRAMEWORK LARAVEL

Jorge Miguel Teodoro de Oliveira

Santo André - SP, 2024

Jorge Miguel Teodoro de Oliveira

**DESENVOLVIMENTO DE UMA FERRAMENTA
PARA PADRONIZAÇÃO DE RELATÓRIOS
CIRCUNSTANCIADOS UTILIZANDO O
FRAMEWORK LARAVEL**

Monografia apresentada à Especialização em Tecnologias e Sistemas de Informação, área de concentração: Desenvolvimento Web, como parte dos requisitos necessários para conclusão do curso.

Universidade Federal do ABC – UFABC
Especialização em Tecnologias e Sistemas de Informação

Orientador: Mario Alexandre Gazziro

Santo André - SP
2024

Agradecimentos

Dedico este trabalho a Deus, pela força e sabedoria concedidas ao longo dessa caminhada.

Agradeço ao meu orientador, Mario Alexandre Gazziro, pela orientação e apoio inestimáveis durante todo o processo. Agradeço ao meu chefe, Daniel, pela compreensão e suporte durante esta jornada. À minha família, pelo amor e incentivo constantes.

*“Deixem que o futuro diga a verdade e avalie
cada um de acordo com o seu trabalho e realizações.
O presente pertence a eles, mas o futuro pelo qual
eu sempre trabalhei pertence a mim.”*
(Nikola Tesla)

Resumo

Este trabalho propõe o desenvolvimento de uma ferramenta web para automatizar e padronizar a criação de relatórios circunstanciados, utilizando o framework Laravel, na instituição CRECI-SP. A ferramenta foi projetada para melhorar a eficiência no gerenciamento de relatórios, eliminando o processo manual, sujeito a erros e inconsistências. A metodologia adotada envolveu análise de requisitos, modelagem de banco de dados e desenvolvimento de uma interface intuitiva para o usuário. Os resultados demonstraram uma significativa redução no tempo de elaboração de relatórios e maior controle sobre os processos administrativos. A solução mostrou-se flexível e escalável, podendo ser adaptada a outros contextos organizacionais com demandas similares.

Palavras-chaves: Laravel, Automatização de Relatórios, Padronização, Desenvolvimento Web, MVC

Abstract

This work proposes the development of a web tool to automate and standardize the creation of circumstantial reports, using the Laravel framework, at the CRECI-SP institution. The tool was designed to improve efficiency in report management by eliminating the manual process, which is prone to errors and inconsistencies. The adopted methodology involved requirements analysis, database modeling, and the development of a user-friendly interface. The results demonstrated a significant reduction in the time spent on report generation and greater control over administrative processes. The solution proved to be flexible and scalable, making it adaptable to other organizational contexts with similar demands.

Keywords: Laravel, Report Automation, Standardization, Web Development, MVC

Lista de ilustrações

Figura 1 – Estrutura básica do HTML5	3
Figura 2 – Formas de usar o CSS	4
Figura 3 – Exemplo de código JavaScript no próprio documento HTML	7
Figura 4 – Benchmarks compilados das principais plataformas de PHP	9
Figura 5 – Exemplo da classe PostPolicy criada	14
Figura 6 – Página 1 do modelo de relatório circunstanciado	20
Figura 7 – Página 2 do modelo de relatório circunstanciado	21
Figura 8 – Diagrama de bloco simplificado da criação de relatórios circunstanciados	24
Figura 9 – Estrutura Analítica do Projeto	30
Figura 10 – Modelo Entidade-Relacionamento (MER)	31
Figura 11 – Arquitetura MVC no Laravel	33
Figura 12 – Exemplo da página principal	34
Figura 13 – Exemplo da página secundária	34
Figura 14 – Página principal do kit tabler.io	35
Figura 15 – Página de login do sistema Gestor de Relatórios Circunstanciados . . .	36
Figura 16 – Menu de navegação lateral com todas as rotas disponíveis	37
Figura 17 – Painel administrativo com a visão geral	38
Figura 18 – Adicionar template para o relatório circunstanciado	39
Figura 19 – Adicionar novo relatório circunstanciado	39
Figura 20 – Itens selecionados para o preenchimento	40
Figura 21 – Campos necessários para a confecção do relatório	41
Figura 22 – Página para fazer o download em .docx do relatório	42
Figura 23 – Relatórios confeccionados	43
Figura 24 – Histórico de pagamentos	44
Figura 25 – Pendencias de pagamentos	45
Figura 26 – Comparativo de confecção dos relatórios dos funcionários antes e depois da implementação do software (média)	47
Figura 27 – Comparativo da quantidade de relatórios refeitos antes e depois da implementação do software	48

Lista de tabelas

Tabela 1 – Diferença entre CSS embutido, interno e externo	6
Tabela 2 – Tabela atual com as PSR's e seus estados	11
Tabela 3 – Comandos básicos do composer	12
Tabela 4 – Desempenho coletado antes da implementação do software	26
Tabela 5 – Requisitos e detalhes do sistema	27
Tabela 6 – Desempenho coletado após implementação do software	46

Sumário

1	INTRODUÇÃO	1
1.1	PANORAMAS DO DESENVOLVIMENTO WEB	2
1.2	TECNOLOGIAS FRONT-END	3
1.2.1	HTML	3
1.2.2	CSS	4
1.2.3	JavaScript	7
1.3	TECNOLOGIAS BACK-END	8
1.3.1	PHP	8
1.3.2	PHP Standards Recommendations	10
1.3.3	Composer – gerenciador de dependências	12
1.3.4	MySql	12
1.4	LARAVEL FRAMEWORK	13
1.4.1	Autenticação	13
1.4.2	Autorização	14
1.4.3	Mapeamento objeto-relacional	14
1.4.4	Migrações de banco de dados	15
1.4.5	Validações	15
1.4.6	Notificações e e-mails	15
1.4.7	Armazenamento de arquivos	15
1.4.8	Filas de trabalhos	15
1.4.9	Agendamento de Tarefas	15
1.4.10	Testes	16
1.4.11	Eventos e WebSockets	16
1.5	JUSTIFICATIVA	17
2	OBJETIVOS DA PESQUISA	18
2.1	OBJETIVO GERAL	18
2.2	OBJETIVO ESPECÍFICO	18
3	PROBLEMA	19
4	ORGANIZAÇÃO DO TRABALHO	23
5	METODOLOGIA	24
5.1	REUNIÃO COM OS INTERESSADOS	24
5.2	AUDITORIA DE PROCESSO	24

5.2.1	Entrada de dados e elaboração do relatório	25
5.2.2	Revisão e acesso ao ERP da empresa	25
5.2.3	Tramitação do protocolo	25
5.3	INDICADORES DE PRODUTIVIDADE	25
5.4	ANÁLISE DE REQUISITOS	26
5.5	EAP	30
5.6	MODELAGEM INICIAL DA BASE DE DADOS	31
6	DESENVOLVIMENTO	32
6.1	VSCODE E VERSIONAMENTO DE CÓDIGO	32
6.2	BASE DE DADOS	32
6.3	BACKEND	33
6.4	FRONT-END	35
7	RESULTADOS	36
7.1	PRINCIPAIS TELAS DO SISTEMA	36
7.2	INDICADORES DE PRODUTIVIDADE APÓS IMPLEMENTAÇÃO DO SOFTWARE	46
8	CONCLUSÃO	49
	APÊNDICES	50
	APÊNDICE A – REPOSITÓRIO DO PROJETO PUBLICADO NO GITHUB	51
	APÊNDICE B – ENTIDADE RELACIONAMENTO FINAL	52
	REFERÊNCIAS	53

1 Introdução

Em rotinas diárias, a geração de relatórios é uma atividade crucial para o sucesso de muitas organizações em diversos ramos. Através de relatórios bem elaborados, estruturados e padronizados, empresas podem tomar decisões mais assertivas, acompanhar desempenho das atividades e otimizar processos. Ainda assim, as elaborações desses relatórios podem ser demoradas, mesmo que seja modelo pronto, pode auxiliar no processo, mas nem sempre conseguem atender às necessidades específicas de cada organização. Rotinas manuais para a elaboração, podem acarretar falta de padronização, esforço desnecessário entre outros problemas, como em alguns casos, a procrastinação por parte do colaborador.

Documento apresentado por comissão parlamentar de inquérito ao finalizar seus trabalhos de investigação, contendo suas conclusões e sugestões. Deverá ser encaminhado às autoridades competentes nos termos regimentais. (NACIONAL, 2024)

O trabalho propõe o desenvolvimento de uma ferramenta web para auxiliar na criação de relatórios padronizados, utilizando o framework Laravel. O Framework teve sua primeira versão em junho de 2011 por Taylor B. Otwell, como uma tentativa de fornecer algo mais avançado do que o framework CodeIgniter, visto que o CodeIgniter na época não fornecia recursos de suporte integrado para autenticação e nem autorização de usuários. Esse kit de desenvolvimento utiliza o padrão arquitetural MVC com a linguagem de programação PHP e atualmente, o Laravel não é apenas um simples framework, mas sim uma robusta entidade, que mantém vários outros projetos para turbinar o desenvolvimento. Dentre seus projetos, atualmente existem ferramentas para: Ambiente de desenvolvimento, Painel de administração, Autenticação via redes sociais, depurador com interface para usuário, Autenticação de API e aplicativos móveis Websockets.

Em 12 de março de 2024, o Laravel lançou sua 11^a versão com suporte de correção de segurança até 12 de março de 2026 e sem dúvidas, o sucesso dessa ferramenta e o auxílio de seus 647 colaboradores no repositório GitHub com a comunidade ativa ao redor do mundo, contribuíram com a evolução e agilidade do desenvolvimento de aplicações seguras para todos os tipos de organizações. A estrutura do software de código aberto é licenciada sob a licença MIT.

1.1 PANORAMAS DO DESENVOLVIMENTO WEB

Durante muito tempo considerou-se que o uso de Tecnologia da informação era exclusivo das grandes organizações. Com o avanço tecnológico atual, grande parte das pequenas e médias empresas estão defasadas nessa área. Organizações que não trabalham o SI ou até desenvolvem mais de forma equivocada acabam sofrendo com vários fatores no decorrer do processo produtivo.

"Sistema é um conjunto de partes interagentes e interdependentes que, conjuntamente, formam um todo unitário com determinado objetivo e efetuam função específica. (OLIVEIRA, 2000)

A internet continua a expandir e evoluir a um ritmo acelerado. Novas tecnologias estão surgindo constantemente, e o crescimento do número de páginas web e usuários conectados tem sido notável ao longo dos últimos anos.

O termo desenvolvimento web, é utilizado para descrever a criação de websites através de processos de construção e testes de softwares específicos para a web, que tem o principal objetivo de obter um conjunto de programas para satisfazer funções pretendidas. A web evoluiu muito desde os seus primórdios, de texto e hipertexto apenas, passamos de imagens e vídeos para sites totalmente interativos carregados de conteúdo multimídia e, infelizmente, scripts e cookies de rastreamento e inúmeras solicitações que tornam o carregamento de tudo mais pesado e lento.

1.2 TECNOLOGIAS FRONT-END

1.2.1 HTML

Criado por Tim Berners-Lee, em 1991, a intenção era interligar computadores do laboratório a outras instituições de pesquisas para exibir documentos de cunho científico de forma simples e fácil de acessar. Após a criação da World Wide Web deu-se a construção de vários browsers (navegadores) WWW e servidores que viabilizaram a WEB.

Mozilla Developer Network define o HTML da seguinte forma:

HTML não é uma linguagem de programação; é uma linguagem de marcação, usada para definir a estrutura do seu conteúdo. HTML consiste de uma série de elementos, que você usa para delimitar ou agrupar diferentes partes do conteúdo para que ele apareça ou atue de determinada maneira. As tags anexas podem transformar uma palavra ou imagem num hiperlink, pode colocar palavras em itálico, pode aumentar ou diminuir a fonte e assim por diante. (NETWORK, 2023)

Portanto, não devemos confundir a linguagem de marcação de textos, com linguagem de programação. Nesses mais de 30 anos, houveram muitas mudanças, embora tenha sido lançado em 1992, demorou um pouco para a rede de computadores alavancar. Antes de 1996 praticamente era utilizado por alguns programadores e entusiastas da área de tecnologia. Os anos de 1995 e 1996 foram cruciais para o sucesso que a internet é hoje, pois floresceram navegadores, ferramentas para construção de páginas.

A estrutura básica de um documento HTML sofreu poucas alterações no decorrer da sua evolução até a atual, porém essa evolução mudou bastante a forma de diagramar um site/sistema, visto que antigamente era apenas o HTML e páginas estáticas, sites criados através de tabelas sem cores e sem dinamismo. Ao passar dos anos, foram criados o CSS que basicamente é a folha de estilo do website e o JavaScript. O HTML, CSS E JavaScript atualmente é uma das três principais tecnologias da World Wide Web.

Figura 1 – Estrutura básica do HTML5

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8"/>
5     <title>Estrutura básica do HTML5</title>
6   </head>
7   <body>
8   </body>
9 </html>
```

Fonte: Elaborado pelo próprio autor (2024).

1.2.2 CSS

CSS é a sigla de Cascading Style Sheets, em português traduzido para Folhas de estilo em cascata. O CSS é utilizado em conjunto em linguagens de marcação de textos, para dar estilo à página web. Dificilmente encontramos uma página web sem estilização CSS atualmente, pois todas páginas web costumam ter cores, tamanhos de fontes, backgrounds, posições de seções.

Mozilla Developer Network define o HTML da seguinte forma:

Assim como o HTML, o CSS não é realmente uma linguagem de programação. Também não é uma linguagem de marcação — é uma linguagem de folhas de estilos. Isso significa que o CSS permite aplicar estilos seletivamente a elementos em documentos HTML. (NETWORK, 2022))

Existem três maneiras de inserir uma folha de estilo:

- CSS externo: Utilizando uma folha de estilos externa, é possível modificar a aparência de todo um site com apenas uma edição em um arquivo. Cada página HTML deve conter uma referência ao arquivo de folha de estilos externo dentro do elemento `<link>` localizado na seção `head`.
- CSS interno: Uma folha de estilos interna pode ser útil em websites onepage, quando uma única página HTML requer um estilo exclusivo. O estilo interno é definido dentro do elemento `<style>` na seção `head`.
- CSS em linha: Para usar estilos em linha, usa-se o atributo `style` ao elemento HTML. Essa forma sem dúvidas é a mais arcaica e menos utilizada.

Figura 2 – Formas de usar o CSS

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8"/>
5     <title>Estrutura básica do HTML5</title>
6     <link rel="stylesheet" href="style.css" <!--CSS externo-->
7     <style>
8       /*CSS interno */
9       div {
10        background-color: orange;
11      }
12    </style>
13  </head>
14  <body>
15    <div>
16      <h1 style="color: blue; font-size: larger;">Olá mundo!</h1> <!--CSS embutido-->
17    </div>
18  </body>
19 </html>
```

Fonte: Elaborado pelo próprio autor (2024).

Cada um tem sua vantagem e desvantagem, que dependerá do tipo de projeto. No caso do CSS Externo a grande vantagem é a reutilização de estilo para múltiplas páginas, utilização de cache de navegador, mesmo que seja feito uma requisição HTTP adicional.

CSS interno e em linha, costumam ser mais rápidos para o carregamento.

A Tabela 1 apresenta as diferenças.

Tabela 1 – Diferença entre CSS embutido, interno e externo

Característica	CSS em linha	CSS interno	CSS externo
Localização	Ele é usado dentro da tag HTML usando o atributo <i>style</i>	Ele é usado dentro da seção <i><head></i> do documento HTML	Ele é usado em um arquivo .css separado
Escopo do seletor	Afeta um único elemento ou um grupo de elementos	Afeta vários elementos dentro do mesmo elemento HTML	Afeta vários documentos HTML ou um site inteiro
Reutilização	Não reutilizável. Os estilos precisam ser repetidos para cada elemento	Pode ser reutilizado em vários elementos dentro do mesmo documento HTML	Pode ser reutilizado em vários documentos HTML ou em um site inteiro
Prioridade	Prioridade máxima. Substitui estilos internos e externos	Prioridade média. Substitui estilos externos, mas pode ser substituído por estilos embutidos	Prioridade mais baixa. Pode ser substituído por estilos embutidos e internos
Tamanho do arquivo	Os estilos embutidos aumentam o tamanho do arquivo HTML, o que pode afetar o tempo de carregamento da página	Os estilos internos fazem parte do arquivo HTML, o que aumenta o tamanho do arquivo	Os estilos externos estão em um arquivo separado, o que reduz o tamanho do arquivo HTML e pode ser armazenado em cache para carregamentos de página mais rápidos
Manutenibilidade	Não é fácil de manter. As alterações precisam ser feitas manualmente em cada elemento	Relativamente fácil de manter. As alterações precisam ser feitas em um só lugar na seção <i><head></i> do documento HTML	Mais fácil de manter. As alterações precisam ser feitas em um só lugar no arquivo de .css externo

Fonte: Adaptado e traduzido de (SATYA, 2024)

1.2.3 JavaScript

Diferente do HTML e CSS, mencionados nos tópicos anteriores, o JavaScript é uma linguagem de programação leve. Entretanto, é importante não confundir o JavaScript com a linguagem de programação Java, pois cada uma possui suas próprias sintaxes e semânticas distintas. Embora seja conhecido como uma linguagem de script para páginas da web, o JavaScript também pode ser executado em ambientes que não possuem navegador, como o Adobe Acrobat, Node.js e outras ferramentas.

Na verdade, o nome “JavaScript” é um pouco enganoso. A não ser pela semelhança sintática superficial, JavaScript é completamente diferente da linguagem de programação Java. E JavaScript já deixou para trás suas raízes como linguagem de script há muito tempo, tornando-se uma linguagem de uso geral robusta e eficiente. (FLANAGAN, 2013).

A linguagem é muito poderosa, simples para aprender e roda no client side da web. Difícilmente encontramos um website sem a tecnologia JavaScript, pois ela é muito útil e dá para executar incontáveis ações e eventos. Alguns exemplos de utilização, são: Manipulação de DOM, manipulação de eventos, requisições assíncronas, geolocalização, validação de formulários, animações e efeitos especiais, armazenamento local, entre outras finalidades. Como outras linguagens de programação orientada a objeto, existem os conceitos de encapsulamento, polimorfismo e herança. Como mostrado no CSS, é possível usar o JavaScript de forma interna (No mesmo documento HTML), ou de forma externa (Em um arquivo a parte com a extensão.js).

Figura 3 – Exemplo de código JavaScript no próprio documento HTML

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8"/>
5     <title>Exemplo JavaScript - Somar</title>
6   </head>
7   <body>
8     <input type="number" id="num1" placeholder="Número 1">
9     <input type="number" id="num2" placeholder="Número 2">
10    <button onclick="somar()">Somar</button>
11    <p>Resultado: <span id="resultado"></span></p>
12    <script>
13      function somar() {
14        var num1 = parseFloat(document.getElementById("num1").value);
15        var num2 = parseFloat(document.getElementById("num2").value);
16        var resultado = num1 + num2;
17        document.getElementById("resultado").textContent = resultado;
18      }
19    </script>
20  </body>
21 </html>
```

Fonte: Elaborado pelo próprio autor (2024).

Com a popularidade, existe uma grande e crescente criação de frameworks baseadas no JavaScript que facilitam o desenvolvimento. Existem algumas razões para a utilização de frameworks e bibliotecas, ao desenvolver aplicativos que visam a produtividade, eficiência, padronização e segurança. Pesquisas realizadas na plataforma de hospedagem de código fonte, o GitHub mostra que as bibliotecas React, Vue e Bootstrap são as mais populares framework front-end, algumas são mantidas por Big Techs, como Google e Facebook. (BUILD..., 2024)

1.3 TECNOLOGIAS BACK-END

Como o termo sugere, backend é o que trabalha na parte de trás da aplicação. Pode estar relacionada à linguagem de programação, de servidores, de banco de dados, estrutura, seguranças e atualizações. Essa estrutura desempenha um papel crucial para as aplicações, pois é a parte responsável pelo processamento e armazenamento de dados.

1.3.1 PHP

O PHP (Hypertext Preprocessor) é uma linguagem de script popular para websites dinâmicos. Criada em 1994, sua simplicidade a torna amigável para iniciantes. Há anos, existem boatos na comunidade de que a linguagem PHP teria pouco tempo de vida útil, existem muitas críticas a respeito. Ao contrário do que dizem, o PHP continua sendo uma ferramenta viável e eficaz para o desenvolvimento web. Em constante evolução, atualmente está em sua versão 8.3 com muitas novidades.

Com a linguagem podemos escrever códigos com o paradigma procedural, que atualmente não é recomendável, tanto como podemos utilizar outras formas como a programação orientada a objetos. Programação Orientada a Objetos (POO) no PHP é uma maneira de escrever código que se concentra em "objetos- coisas que possuem características e fazem coisas. No PHP, isso significa criar classes que definem como os objetos devem ser, com suas próprias características (como nome, idade, etc.) e comportamentos (coisas que podem fazer, como correr, pular, etc.).

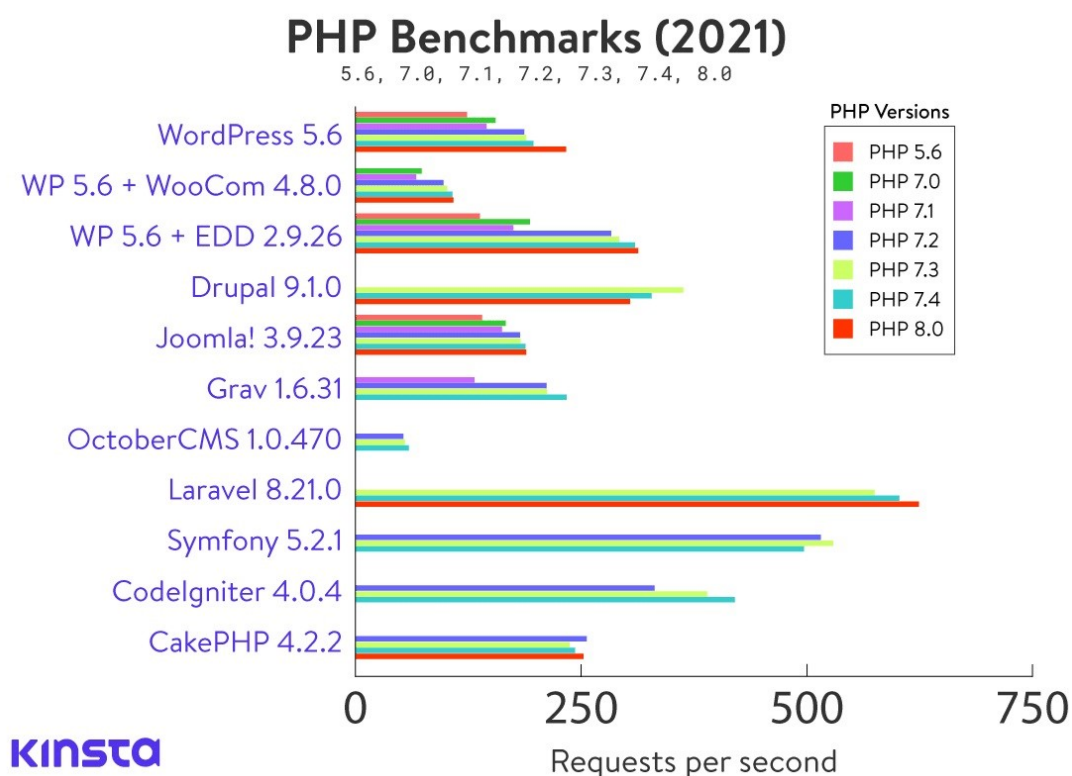
Quando escrevemos código em POO, começamos criando uma classe que serve como um modelo para os objetos que queremos criar. Então, podemos criar objetos a partir dessa classe, cada um com suas próprias características e comportamentos únicos. Um dos benefícios legais da POO é a herança. Isso significa que podemos criar uma classe que "herda" características e comportamentos de outra classe. Isso economiza tempo e torna nosso código mais organizado e fácil de entender.

Outro conceito importante é o encapsulamento. Isso basicamente significa que podemos esconder detalhes complicados sobre como nossos objetos funcionam e só mostrar o que é importante para quem está usando esses objetos. É como ter um controle remoto de TV - você não precisa saber como funciona por dentro, apenas como usá-lo.

E tem o polimorfismo, que é um jeito chique de dizer que podemos tratar diferentes objetos de maneira parecida, desde que eles façam coisas parecidas. Isso torna nosso código mais flexível e poderoso. No geral, a POO no PHP nos ajuda a escrever código mais organizado, fácil de manter e reutilizável. É uma habilidade fundamental para qualquer programador PHP que queira construir coisas legais na web. Muitas ferramentas foram construídas com PHP, como gerenciadores de conteúdo, lojas virtuais, portais de notícias, blogs, frameworks, sendo assim, é uma das linguagens de programação mais utilizadas no mundo.

“O PHP 8.0 se destacou como o vencedor na maioria das plataformas que o suportam, incluindo WordPress e Laravel.” (DANIELE, 2020)

Figura 4 – Benchmarks compilados das principais plataformas de PHP



Fonte: Carlo Daniele.

1.3.2 PHP Standards Recommendations

A linguagem PHP infelizmente sofre um grande preconceito na comunidade do desenvolvimento, por ser uma tecnologia que muitos utilizam para entregar soluções rápidas, com códigos de baixa qualidade e difícil manutenção. Em sua versão 5x era muito precária e somente após a versão 7x a comunidade amadureceu entregando soluções com mais qualidade de código.

Existe um grupo formado por desenvolvedores chamado PHP-FIG, que é responsável por falar sobre pontos em comuns no projeto e encontrar uma melhor maneira de trabalhar juntos. O principal objetivo era definir recomendações que fossem aplicadas aos frameworks PHP. Ao passar do tempo, o grupo percebeu que seria útil em outros tipos de projetos, como em grandes gerenciadores de conteúdo. Assim o grupo ganhou novos membros colaboradores, por exemplo: WordPress, Magento, Joomla, Drupal e diversos outros.

As PSR são as recomendações para o estilo de desenvolvimento,

“As recomendações de estilo possuem a função de padronizar o desenvolvimento de uma aplicação para que outros desenvolvedores possam realizar manutenção ou dar continuidade à implementação do software de maneira rápida e fácil” (CARLOS, 2017).

Cada PSR tem um status de como está sendo trabalhado, uma vez que uma proposta de nova PSR tenha passado em votação, ela será listada como rascunho. O rascunho pode mudar para revisão e posteriormente poderá ser aceito ou não. Os status são: Aceito, Rascunho, Abandonado e Obsoleto. Abaixo segue a tabela com todas as recomendações, com seus números de identificação, título, mantenedores/editores e qual é a situação atual.

Tabela 2 – Tabela atual com as PSR's e seus estados

Num	Título	Editor(es) / Man- tenedores	Estado
0	Carregamento automático padrão		Depreciado
1	Padrão de codificação básico	vago	Aceitado
2	Guia de Estilo de Codificação		Depreciado
3	Logger Interface	Jordi Boggiano	Aceitado
4	Carregamento automático padrão	vago	Aceitado
5	Padrão PHPDoc	Chuck Burgess	Rascunho
6	Interface de cache	Larissa Garfield	Aceitado
7	Interface de mensagem HTTP	Mateus Weier O'Phinney	Aceitado
8	Huggable Interface	Larissa Garfield	Abandonado
9	Avisos de Segurança	Michael Hess	Abandonado
10	Processo de emissão de relatórios de segurança	Michael Hess	Abandonado
11	Interface de contêiner	Matthieu Napoli, David Négrier	Aceitado
12	Guia de estilo de codificação estendida	Korvin Szanto	Aceitado
13	Links de hipermídia	Larissa Garfield	Aceitado
14	Despachante de Eventos	Larissa Garfield	Aceitado
15	Manipuladores HTTP	Gilk amadeirado	Aceitado
16	Cache simples	Paulo Dragoonis	Aceitado
17	Fábricas HTTP	Gilk amadeirado	Aceitado
18	Cliente HTTP	Tobias Nyholm	Aceitado
19	Tags do PHPDoc	Chuck Burgess	Rascunho
20	Relógio	Chris Seufert	Aceitado
21	Internacionalização	Navarr Barnier	Rascunho
22	Rastreamento de aplicativos	Adão Allport	Rascunho

Fonte: Traduzido de Framework Interop Group ([PHP-FIG, 2024](#))

1.3.3 Composer – gerenciador de dependências

O Composer é amplamente reconhecido como o principal gerenciador de dependências para projetos em PHP. Sobre a ferramenta,

"trata-se de um gerenciador de dependências para PHP, que é uma ferramenta simples e considerada confiável, nos quais os desenvolvedores utilizam para o gerenciamento e integração de pacotes ou bibliotecas externas de projetos baseados em PHP". (CARDOSO, 2021)

O Composer simplifica o processo de integração e gerenciamento de bibliotecas externas, permitindo que os desenvolvedores instalem, removam e atualizem pacotes de terceiros com facilidade. As dependências são declaradas no arquivo `composer.json`, onde também são especificadas as versões mínimas e máximas necessárias para cada pacote. Ao utilizar o Composer, os desenvolvedores podem garantir que as bibliotecas utilizadas estejam sempre atualizadas e compatíveis com o código em desenvolvimento. Além disso, o Composer fornece um autoloader que simplifica o processo de carregamento das classes necessárias, tornando-o automático conforme elas são utilizadas no código.

Em resumo, o Composer é uma ferramenta essencial para o desenvolvimento em PHP, proporcionando eficiência e confiabilidade no gerenciamento de dependências e integração de pacotes externos nos projetos.

Tabela 3 – Comandos básicos do composer

Comandos CLI	Ação
<code>composer require vendor/pacote</code>	Adicionar uma nova dependência
<code>composer install</code>	Instalar todas as dependências da lista
<code>composer update</code>	Atualizar as dependências
<code>composer remove vendor/pacote</code>	Remover uma dependência específica
<code>composer show</code>	Mostrar informações das dependências instaladas

Fonte: Elaborado pelo próprio autor (2024).

1.3.4 MySQL

O programa MySQL é um sistema de gerenciamento de banco de dados relacional que utiliza a linguagem de consulta estruturada SQL como interface de acesso e extração de informações do banco de dados em uso. O MySQL é um dos sistemas de gerenciamento de bancos de dados mais populares e usados no mundo. É rápido, multitarefa e multiusuário. (AUGUSTO, 2011)

Banco de dados é um sistema para armazenar e gerenciar grandes quantidades de dados de forma eficiente. Com ele é possível inserir, recuperar, atualizar e excluir dados de forma rápida e segura. O banco de dados é composto por várias partes principais. Dados, SGBD, Schema, linguagem de consulta, Índices. Os bancos de dados são utilizados em inúmeros softwares e desempenham um papel fundamental na organização e manutenção de dados para aplicações de softwares.

Os bancos de dados relacionais surgiram a partir do modelo relacional, desenvolvido por Codd em 1970 (CODD, 1970), e, até os tempos atuais, são os bancos de dados mais utilizados no mercado. Eles são conhecidos por serem sistemas de propósito geral, utilizarem a linguagem SQL (Structured Query Language; ou Linguagem de Consulta Estruturada, em português) para consulta e manipulação de dados e estruturas de dados, além das propriedades ACID para transações. (SILVA, 2021)

1.4 LARAVEL FRAMEWORK

O framework se destaca para o desenvolvimento web, com sua arquitetura MVC é possível criar aplicações modernas e escaláveis com uma comunidade bastante ativa e uma documentação impecável. Pronto para o uso, o laravel possui soluções elegantes para recursos comuns necessários para todas aplicações. O principal foco de utilização de frameworks, é evitar a perda de tempo e parar de reinventar a roda. Boa parte da aplicação, é executada no terminal, desde a criação do projeto, de tabelas no banco de dados, limpeza de cache, criação de modelos, policies, eventos, controladores, para verificar todos os comandos úteis basta utilizar o comando `php artisan list`.

Para a criação do projeto via composer, usa-se o comando: `composer create-project laravel/laravel meu-projeto`

Os tópicos abaixo são os itens que tornam o Laravel Framework incrível.

1.4.1 Autenticação

É disponibilizado uma camada completa de autenticação, desde login, registro de usuário, recuperação de senha, validação de registro enviando mensagem ao email de cadastro, limites de tentativas de acesso. O bom de ter tudo isso pronto, é porque não há a necessidade de se preocupar na segurança, pois a ferramenta está bem madura e segura nesse quesito. Sobre a senha de usuário, é armazenada em banco de dados com criptografia bcrypt, que por sinal é extremamente segura. Para obter o usuário autenticado na aplicação, usa-se o seguinte código:

```
$user = Auth::user();
```

Com as atualizações do framework, houve mudanças no pacote de autenticação, antes usava o kit `laravel/ui` ainda com suporte para a última versão, porém o CEO Taylor Otwell recomenda que utilize o `laravel/jetstream` ou `laravel/breeze` que também é um kit de pré-definições de autenticação que tem as opções de adicionar a camada front-end com React/Vue, Livewire, Blade ou Next.js/API. A diferença do `laravel/jetstream` para o `laravel/breeze`, é que o `jetstream` é mais completo, com a opção de controle de times/acessos.

1.4.2 Autorização

É muito importante para qualquer tipo de sistema verificar se o determinado usuário tem autorização para executar uma tarefa, é muito simples fazer a verificação. Em resumo, os guards gerenciam a autenticação, os gates definem regras gerais de controle de acesso e as policies fornecem uma maneira estruturada de gerenciar regras de autorização específicas para os modelos do seu aplicativo. Esses conceitos funcionam juntos para ajudá-lo a controlar o acesso e as permissões. As policies são mais organizadas e há código específico para ser executado no cli do php, basta executar o código

```
$php artisan make:policy PostPolicy --model=Post
```

Será criado uma classe PostPolicy que é possível determinar cada ação padrão de usuário.

Figura 5 – Exemplo da classe PostPolicy criada

```
class PostPolicy
{
    /**
     * Determine if the given post can be updated by the user.
     */
    public function update(User $user, Post $post): bool
    {
        return $user->id === $post->user_id;
    }
}
```

Fonte: Documentação laravel.

1.4.3 Mapeamento objeto-relacional

Para minimizar o problema de comunicação entre o banco de dados relacional e a programação orientada a objetos, utilizam-se ferramentas de Mapeamento Objeto Relacional, cujo objetivo é possibilitar uma transparência na comunicação entre a aplicação e o seu banco de dados. (ALCARAZ, 2016)

Cada framework famoso possui seu ORM. Assim como o Django (Python) utiliza o Django ORM, Ruby on Rails (Ruby) utiliza o ActiveRecord, Spring Boot (Java) utiliza o Spring Data JPA (Java Persistence API), Express.js (Node.js) utiliza Sequelize (também pode usar outras opções como Mongoose para MongoDB), ASP.NET Core (C#) usa Entity Framework Core e o Laravel utiliza o Eloquent ORM. No eloquent ORM, basicamente para cada tabela há uma classe modelo.

1.4.4 Migrações de banco de dados

Sem dúvidas, é uma boa prática utilizar as migrações para controlar versões do banco de dados. No diretório `database\migrations` é armazenado todas as tabelas do projeto, onde é possível criar novas tabelas, alterar atributos, adicionar novos campos com facilidade. Cada arquivo está nomeado com base em sua característica, no exemplo da tabela usuários está assim:

```
0001_01_01_000000_create_users_table.php
```

1.4.5 Validações

O Laravel disponibiliza um sistema robusto de validação, simplificando a verificação dos dados fornecidos pelos usuários antes de serem processados pelo aplicativo. Através da utilização das classes de validação, torna-se possível definir de maneira precisa as regras de validação, como campos obrigatórios, formatos de dados e limites de tamanho.

1.4.6 Notificações e e-mails

O Laravel apresenta uma funcionalidade refinada para o envio de e-mails e notificações aos usuários do aplicativo. Através de uma API expressiva, é possível enviar mensagens de forma síncrona ou assíncrona, integrando-se a diversos serviços de e-mail renomados.

1.4.7 Armazenamento de arquivos

O Laravel oferece suporte nativo para o armazenamento de arquivos em diversas plataformas, incluindo sistemas de arquivos locais e serviços de armazenamento na nuvem. Através da API de armazenamento, é possível manipular arquivos de forma eficiente, executando operações como upload, download e exclusão.

1.4.8 Filas de trabalhos

As filas de trabalhos constituem uma ferramenta poderosa para o processamento assíncrono de tarefas em segundo plano no Laravel. Com o uso das filas, é possível executar tarefas demoradas sem impactar a experiência do usuário no aplicativo.

1.4.9 Agendamento de Tarefas

O agendamento de tarefas desempenha um papel fundamental na automação de tarefas recorrentes no Laravel. Através da utilização do Laravel Scheduler, é possível agendar a execução de tarefas como envio de e-mails e geração de relatórios em intervalos regulares.

1.4.10 Testes

O Laravel facilita a escrita e execução de testes automatizados para garantir a qualidade e estabilidade do aplicativo. Utilizando o Laravel PHPUnit, é possível desenvolver testes unitários, de integração e funcionais para validar o comportamento do aplicativo em diferentes cenários.

1.4.11 Eventos e WebSockets

O Laravel oferece suporte nativo para o desenvolvimento de aplicativos em tempo real através de eventos e WebSockets. Com o uso do Laravel Echo e do Laravel WebSockets, é possível criar aplicativos interativos que respondem em tempo real às ações dos usuários.

1.5 JUSTIFICATIVA

Oferecer uma solução que automatiza tarefas repetitivas, simplificando processos e promover a colaboração entre os membros que são responsáveis por gerenciar esses relatórios. Além disso, a implementação dessa ferramenta pode resultar em economia de tempo e recursos, além de melhorar a precisão e a confiabilidade dos relatórios produzidos.

2 OBJETIVOS DA PESQUISA

2.1 OBJETIVO GERAL

Na instituição CRECISP: Conselho Regional de Corretores de Imóveis 2ª Região Conselho, alguns funcionários são designados para fiscalizar os contratos de serviços e gerar relatórios circunstanciados. Esse relatório tem como objetivo apontar se o serviço foi prestado de forma satisfatória e atendeu ao período de cobrança, ou em caso de necessidade de apontamentos para que seja providenciado abatimentos ou penalizações ao contratado, detalhando os serviços pelo seu período.

O objetivo da pesquisa é levantar detalhes da rotina para a geração dos relatórios e desenvolver uma ferramenta capaz de automatizar o processo. Seguindo a documentação do framework, devemos desenvolver algo amigável e eficiente para suporte na rotina específica. Desde a criação do modelo padrão, até a integração com o ERP atual. A utilização do framework foi escolhida por ser uma ferramenta open-source, simples, flexível e robusta.

2.2 OBJETIVO ESPECÍFICO

- Entender e mapear o fluxo atual de geração dos relatórios circunstanciados
- Levantar requisitos para a modelagem da aplicação
- Modelar o banco de dados
- Aplicar as melhores práticas e seguir a documentação do Framework
- Fazer os devidos testes
- Trazer os resultados após a criação da ferramenta

3 PROBLEMA

A falta de sistemas informatizados para pequenas rotinas administrativas é um problema comum em muitas empresas brasileiras, esse problema aumenta, quando trata-se de rotinas específicas. A carência por falta de informatização nessas áreas, podem ter um impacto significativo na produtividade e eficiência e falta de integração com outras ferramentas. Na empresa em questão, os relatórios criados manualmente, são armazenados em pastas do explorador do Windows e posteriormente armazenado no ERP da empresa. Não há um controle eficiente para a consulta rápida de determinado mês referência. Há uma certa dificuldade para rever determinado relatório, obter o número de protocolo gerado no ERP da empresa, visto que não há uma ferramenta de apoio para armazenar esses detalhes.

Figura 6 – Página 1 do modelo de relatório circunstanciado

RELATÓRIO CIRCUNSTANCIADO

Modalidade da licitação: Pregão Nº: 0001/XXXX Processo: 0001/YYYY
 Empresa Contratada: Empresa Fictícia
 Objeto: Serviços de internet
 Localidade: SEDE
 Nota de Empenho: 0001/AAAA
 Relatório elaborado por: José Bento Renato Monteiro Lobato

Apresento Relatório Circunstanciado de Execução de Contrato conforme informações discriminadas abaixo:

1. INFORMAÇÕES DO CONTRATO E SUA EXECUÇÃO

RAZÃO SOCIAL: Empresa Fictícia		CNPJ Nº: 00.000.000/0000-00
OBJETO CONTRATADO: Serviços de internet		
DATA DA CONTRATAÇÃO: 01/01/2024	VIGÊNCIA DO CONTRATO: 31/12/2024	
VALOR TOTAL DO CONTRATO: R\$ 1.440,00	VALOR MENSAL: R\$ 120,00	
DATA DE VENCIMENTO DA FATURA: 30/04/2024	VALOR DA FATURA APRESENTADA: R\$ 120,00	
MÊS DE REFERÊNCIA DA FATURA: Abril/2024		
GLOSA PROPOSTA (VALOR E JUSTIFICATIVA DA GLOSA):		
CERTIDÕES DE REGULARIDADE FISCAL, SOCIAL E TRABALHISTA – VÁLIDAS E/OU OUTROS DOCUMENTOS ANEXOS:		
EVENTUAIS OCORRÊNCIAS DETECTADAS		
EVENTUAIS OCORRÊNCIAS/FALHAS NA EXECUÇÃO DO CONTRATO		
SUGESTÕES DE MEDIDAS A SEREM IMPLEMENTADAS:		

Fonte: Adaptado pelo autor com base no relatório atual da empresa (2024).

Figura 7 – Página 2 do modelo de relatório circunstanciado

2. ACOMPANHAMENTO DO SALDO CONTRATADO

3.1. Valor Total do Contrato: **R\$ 1.440,00**

3.2. Planilha de Acompanhamento do Saldo Contratual:

Número da Nota Fiscal/Fatura	Período de execução/referência	Valor da Nota Fiscal/Fatura – R\$	Vencimento	Valor do Saldo Contratual – R\$
01001	Janeiro/2024	R\$ 120,00	31/01/2024	R\$ 1.320,00
02002	Fevereiro/2024	R\$ 120,00	31/02/2024	R\$ 1.200,00
03003	Março/2024	R\$ 120,00	31/03/2024	R\$ 1.080,00
04004	Abril/2024	R\$ 120,00	31/04/2024	R\$ 960,00

3. INFORMAÇÕES GERAIS QUE O FISCAL JULGUE PERTINENTE:

Diante do exposto, segue a Fatura/Nota Fiscal/Recibo nº \${fatura}, devidamente atestada, tendo em vista a efetiva prestação dos serviços contratados, considerando os apontamentos acima registrados, para conhecimento, análise e posterior pagamento da despesa, ressaltando que (✓) HÁ ou (X) NÃO HÁ a proposição de aplicação de GLOSAS.

São Paulo, 03 de abril de 2024.

José Bento Renato Monteiro Lobato
Departamento de Tecnologia da Informação.

2

Fonte: Adaptado pelo autor com base no relatório atual da empresa (2024).

É importante observar na figura 7 , a planilha de acompanhamento do saldo contratual, pois ali é lançado o valor de cada boleto mensal, ou único. O maior problema da elaboração desses relatórios manuais, é quando o prestador de serviço é recorrente quando disponibiliza o serviço para mais de uma unidade operacional. A empresa VIVO S.A por exemplo, serve o CRECISP em média 18 endereços, neste caso, deverão ser elaborados 18 relatórios manuais apenas desse prestador de serviço.

4 ORGANIZAÇÃO DO TRABALHO

O trabalho se dará em duas etapas. A primeira etapa é a escolha do tema de estudos que envolverá: Capítulos e Revisão da Literatura, Levantamento de requisitos e escolha das tecnologias que serão utilizadas. A segunda etapa será uma breve abordagem do framework e das tecnologias relacionadas no desenvolvimento, colocando em prática a implementação do sistema a fim de demonstrar resultados do que foi produzido e a conclusão do projeto.

5 METODOLOGIA

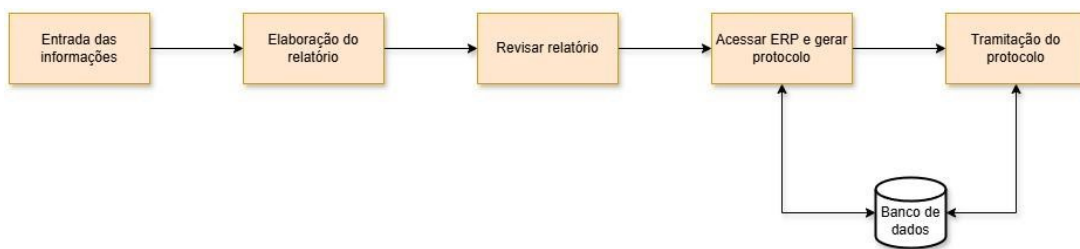
Para resolver os problemas identificados, inicialmente foram estabelecidos alguns requisitos para a concepção de sistema e sua integração no cotidiano operacional. Em seguida, foi delineada uma proposta de solução. As metodologias empregadas consistem na coleta e análise de informações relacionadas a organização CRECISP, visando compreender aspectos pertinentes ao gerenciamento de relatórios circunstanciados. Nesse contexto, a abordagem adotada visa a aplicação direta dos conhecimentos práticos na resolução de problemas reais, resultando no desenvolvimento de um sistema para o gerenciamento eficaz desses documentos. Ao longo deste trabalho, será apresentado o processo de concepção, desenvolvimento e implementação do sistema proposto, detalhando sua funcionalidade e impacto na operação da organização.

5.1 REUNIÃO COM OS INTERESSADOS

Durante a reunião com os interessados, foram delineados os procedimentos para a elaboração de um relatório circunstanciado, os quais foram divididos em etapas distintas. Com essa reunião, foi possível ter uma visão geral e para demonstrarmos neste trabalho de forma simples, foi elaborado o diagrama de blocos simplificado.

Diagramas de blocos simples são ferramentas valiosas para representar sistemas de forma clara e concisa. Através de blocos, setas e símbolos padronizados, eles facilitam a compreensão de como os componentes do sistema interagem e funcionam em conjunto.

Figura 8 – Diagrama de bloco simplificado da criação de relatórios circunstanciados



Fonte: Elaborado pelo próprio autor (2024).

5.2 AUDITORIA DE PROCESSO

A auditoria de processo envolve a revisão sistemática de um processo ou sistema, com o objetivo de avaliar sua conformidade com os requisitos estabelecidos, identificar áreas de melhoria e garantir a eficácia e eficiência do processo. Ela pode incluir entrevistas,

observações diretas, revisão de documentos e análise de dados para fornecer uma visão abrangente do desempenho do processo.

5.2.1 Entrada de dados e elaboração do relatório

Cada prestador de serviço emite boletos e notas fiscais relativos aos serviços prestados. Alguns desses prestadores, como no caso dos provedores de internet, oferecem serviços para múltiplas unidades operacionais.

Para cada unidade operacional, é necessário gerar um relatório circunstanciado, que pode abranger um ou mais boletos mensais. Por exemplo, serviços como "Serviço de IP Dedicado" e "Monitoramento de Dados".

O relatório circunstanciado inclui informações como os dados do prestador de serviço, dados da nota de empenho, informações específicas da instituição (no caso, CRECISP), detalhes da unidade operacional, eventuais observações sobre o desempenho do contrato, dados do boleto e informações do gestor responsável pelo contrato, incluindo sua assinatura e o acompanhamento do saldo contratual inicialmente estabelecido na nota de empenho.

5.2.2 Revisão e acesso ao ERP da empresa

É essencial acessar o ERP da empresa para a criação de um protocolo para cada relatório elaborado. Um protocolo deve ser gerado para cada relatório produzido. Após a geração do protocolo, este deve ser anexado ao relatório correspondente.

5.2.3 Tramitação do protocolo

É necessário encaminhar todos os protocolos que foram feitos no dia para o Departamento de Superintendência. Essas etapas compõem o processo de criação e registro de relatórios circunstanciados, os quais são fundamentais para o controle e monitoramento eficaz dos serviços prestados.

5.3 INDICADORES DE PRODUTIVIDADE

Para medir a eficiência e produtividade da rotina de criação de relatórios consideramos algumas abordagens e métricas:

- Tempo médio por relatório: O tempo que o funcionário leva para criar cada relatório individualmente. Isso pode ajudar a identificar padrões e oportunidades de melhoria na eficiência do processo.

- Número de relatórios por período: Quantos relatórios o funcionário é capaz de criar em uma manhã ou tarde. Isso pode ser uma medida de produtividade geral.
- Taxa de erros ou retrabalho: Quantos relatórios precisam ser revisados ou corrigidos devido a erros ou problemas de qualidade.
- Satisfação do funcionário: Pesquisas de satisfação para entender como o funcionário se sente em relação a ferramentas utilizadas.

Na tabela 4 estão as estatísticas coletadas durante o período.

Tabela 4 – Desempenho coletado antes da implementação do software

	Funcionário 1	Funcionário 2	Funcionário 3	Média
Tempo Médio por Relatório	Entre 6 a 8 minutos	Entre 7 a 9 minutos	Entre 6.5 a 8.5 minutos	7.5 minutos
Número de Relatórios por Período	20	20	20	-
Qualidade dos Relatórios (re-feitos)	5 (25%)	4 (20%)	2 (10%)	18.33%
Interrupções Durante o Processo	5	7	3	5
Satisfação do Funcionário	Achou ruim o procedimento	Achou ruim o procedimento	Achou ruim o procedimento	-

Fonte: Elaborado pelo próprio autor (2024).

5.4 ANÁLISE DE REQUISITOS

Este documento apresenta a análise de requisitos para o desenvolvimento do sistema que auxilia na elaboração de relatórios circunstanciados. O sistema deve gerar relatórios no formato Word, utilizando dados armazenados em um banco de dados interno. É importante envolver todos os stakeholders (partes interessadas) do projeto na definição dos requisitos para garantir que o sistema atenda às necessidades reais da organização.

Tabela 5 – Requisitos e detalhes do sistema

Categoria	Número	Requisito e Detalhes
Gerenciamento de Usuários	RF01	Cadastro de Usuários: Nome completo, Endereço de e-mail, Senha, Foto de perfil (opcional), ID da conta Google (opcional), Data de criação, Data de atualização
	RF02	Autenticação de Usuários: Login utilizando e-mail e senha
	RF03	Gerenciamento de Perfis: Visualização e edição de perfil, controle de acesso baseado em níveis de permissão
Gerenciamento de Empresas	RF04	Cadastro de Empresas: Nome da empresa, Razão social, CNPJ, ID do usuário que cadastrou, Data de criação, Data de atualização
	RF05	Consulta de Empresas: Pesquisa por nome, razão social ou CNPJ
	RF06	Edição de Empresas: Edição das informações cadastradas
Gerenciamento de Notas de Empenho	RF07	Cadastro de Notas de Empenho: Ano, Número do processo, Modalidade, Processo da modalidade, Serviço prestado, Valor total, Valor mensal, Descrição do serviço, Situação, Data de início e fim
	RF08	Consulta de Notas de Empenho: Pesquisa por ano, número do processo, modalidade, serviço prestado, situação, data de início/fim
	RF09	Edição de Notas de Empenho: Edição das informações cadastradas
Gerenciamento de Localizações	RF10	Cadastro de Localizações: Nome da unidade operacional
Continua na próxima página		

Tabela 5 – continuação da página anterior

Categoria	Número	Requisito e Detalhes
	RF11	Consulta de Localizações: Pesquisa por nome
Gerenciamento de Relatórios	RF12	Criação de Relatórios: Utilização de dados de Empresas, Localizações, Notas de Empenho e Usuários, geração no formato Word, campos específicos para gestor, departamento, data de geração, dados das empresas, localizações e notas de empenho
	RF13	Filtros de Relatórios: Filtro por período de tempo, empresas, unidades operacionais, modalidades, situação
	RF14	Visualização de Relatórios: Visualização no formato Word, download do relatório
Gerenciamento de Pagamentos (Opcional)	RF15	Cadastro de Pagamentos: Identificador único (UUID), Número da fatura, Referência do pagamento, Preço do serviço, Data de pagamento, Data de elaboração
	RF16	Associação de Pagamentos a Relatórios: Vinculação de pagamentos a relatórios específicos
Requisitos Não-Funcionais	RNF01	Segurança: Autenticação segura, criptografia de dados, controle de acesso baseado em permissões
	RNF02	Performance: Resposta ágil e eficiente às solicitações dos usuários
	RNF03	Usabilidade: Interface intuitiva e fácil de usar
	RNF04	Manutenção: Facilidade de manutenção e atualização
Continua na próxima página		

Tabela 5 – continuação da página anterior

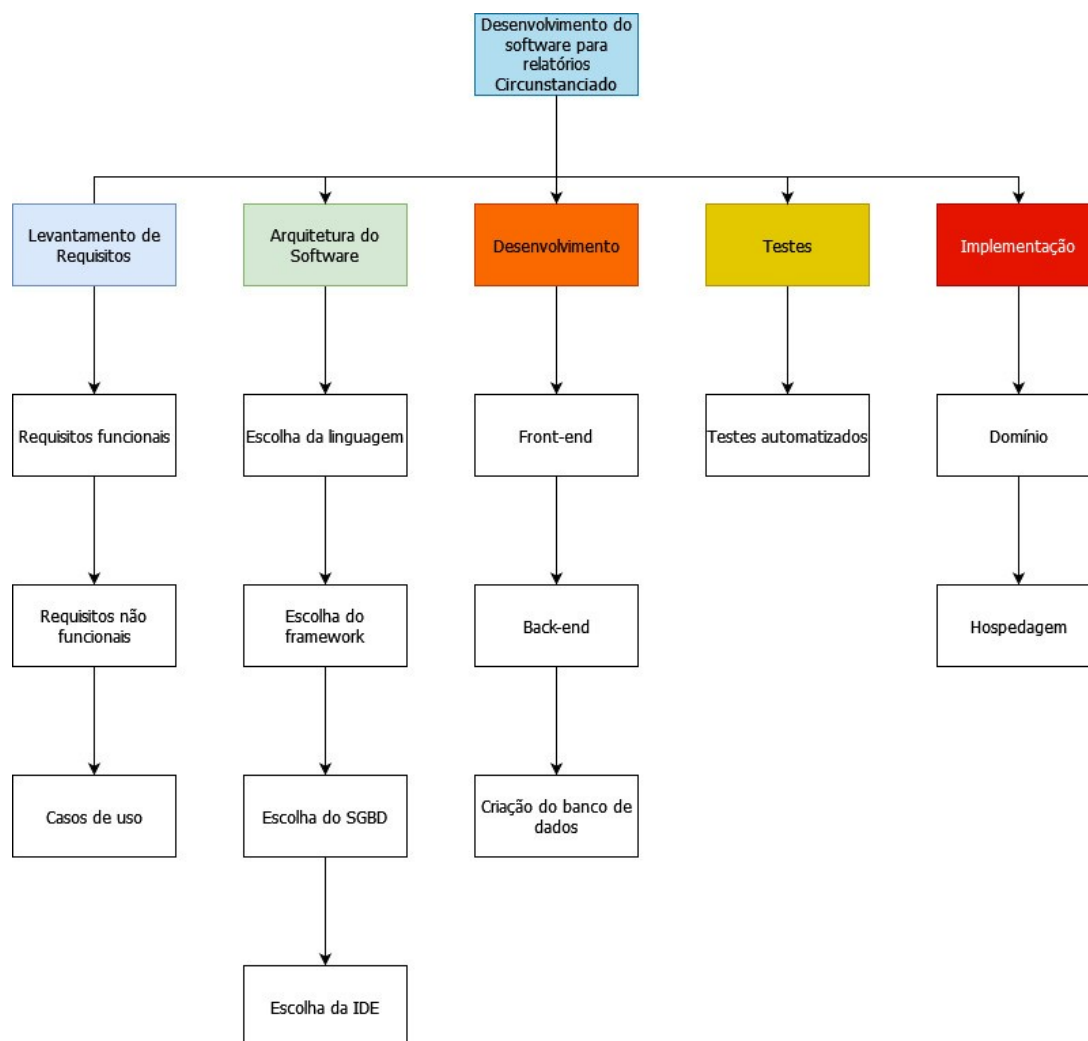
Categoria	Número	Requisito e Detalhes
Considerações Futuras	CF01	Integração com outros sistemas: Avaliação da integração com sistemas de contabilidade ou ERP
	CF02	Relatórios customizáveis: Análise para permitir customização dos relatórios, inclusão/exclusão de campos específicos
	CF03	Painel de visualização: Consideração para implementação de um painel de visualização de dados resumidos e gráficos

Fonte: Elaborado pelo próprio autor (2024).

5.5 EAP

Estrutura Analítica do Projeto (EAP) é uma expressão traduzida do inglês Work Breakdown Structure (WBS). Todo o trabalho do projeto deve estar, de alguma forma, representado na EAP; caso não esteja, não é um trabalho que poderá ser realizado no projeto. (CAMARGO, 2018)

Figura 9 – Estrutura Analítica do Projeto

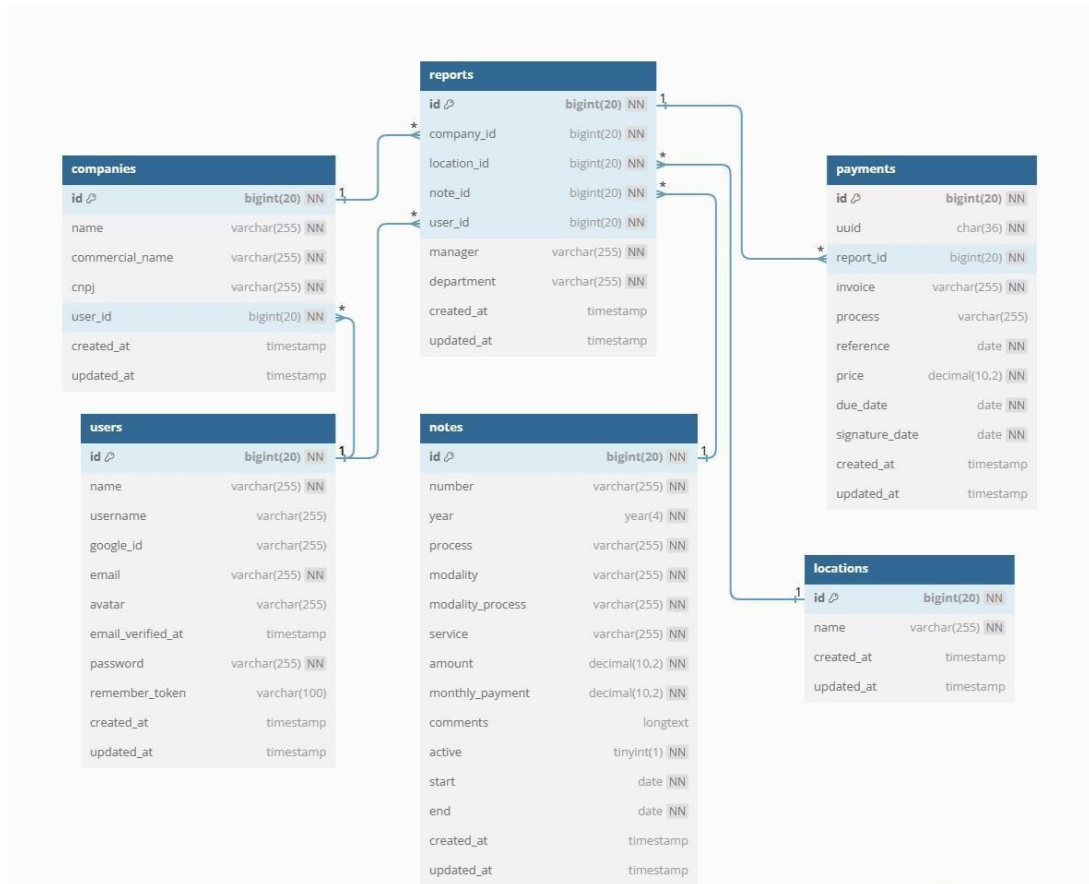


Fonte: Elaborado pelo próprio autor (2024).

5.6 MODELAGEM INICIAL DA BASE DE DADOS

Com base nos requisitos levantados, foi realizada a modelagem inicial da base de dados necessária para armazenar as informações dos relatórios, prestadores de serviço, notas fiscais, entre outros dados relevantes.

Figura 10 – Modelo Entidade-Relacionamento (MER)



Fonte: Elaborado pelo próprio autor (2024).

6 DESENVOLVIMENTO

Após reuniões com as partes interessadas e análise da estrutura do projeto, chegou a hora de selecionar as tecnologias a serem utilizadas. O framework e a linguagem de programação escolhidos foram Laravel/PHP, devido à ampla comunidade ativa na internet e aos conteúdos/tutoriais atualizados, conforme discutido nos capítulos 1 e 1.4.

6.1 VSCODE E VERSIONAMENTO DE CÓDIGO

No desenvolvimento do software criado, optou-se pelo editor de texto Visual Studio Code (VSCODE) como ferramenta de trabalho. Esse editor, criado há quase dez anos pela Microsoft, é atualmente bastante utilizado por sua leveza, facilidade de uso, suporte a extensões e integração com ferramentas de versionamento de código.

Extensões utilizadas:

- PHP Intelephense
- Lavavel Blade Snippets
- HTML CSS Support
- GitHub Pull Requests

O desenvolvimento da ferramenta foi gerenciado com GIT e disponibilizado no GitHub. Essas são ferramentas essenciais para o desenvolvedor, mas têm funções diferentes.

O controle de versão é um sistema que registra alterações em um arquivo ou conjunto de arquivos ao longo do tempo, permitindo lembrar versões específicas mais tarde.

Enquanto o GIT funciona de forma offline para computadores centralizados, o GitHub é uma plataforma de hospedagem de código que usa o GIT para controle de versão. Ele fornece uma interface web e ferramentas adicionais para facilitar a colaboração, a revisão de código e a integração contínua. O link do software desenvolvido está disponível no repositório online do autor. Disponível no APÊNDICE A.

6.2 BASE DE DADOS

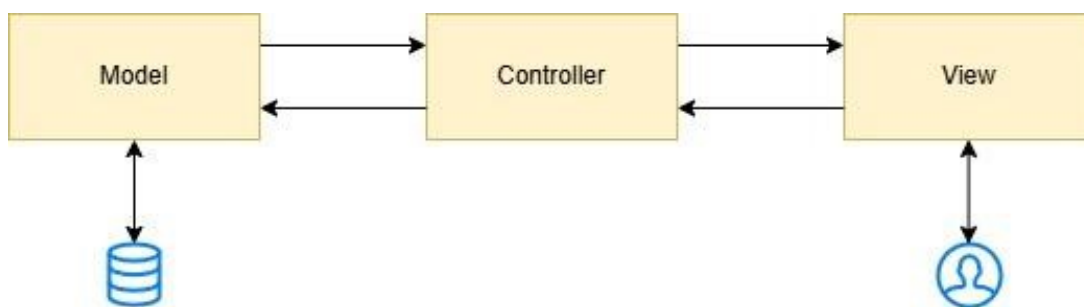
Para o banco de dados, foi selecionado o MySQL, um sistema de gerenciamento de banco de dados relacional estruturado com licença de código aberto e gratuito. Uma das vantagens do MySQL é a sua facilidade de acesso, pois pode ser utilizado em diferentes sistemas operacionais. O MySQL proporciona segurança eficiente, permitindo apenas acesso de usuários com permissões, além de usar criptografia de dados. No contexto do MySQL, estamos usando a interface phpMyAdmin, uma ferramenta de administração baseada na web.

A utilização conjunta de MySQL e phpMyAdmin oferece uma sinergia que potencializa as capacidades de gestão e administração de bancos de dados. Enquanto MySQL fornece a infraestrutura robusta e eficiente para o armazenamento e processamento de dados, phpMyAdmin simplifica a interação com essa infraestrutura, tornando a administração acessível mesmo para usuários com conhecimento limitado em SQL.

6.3 BACKEND

Utilizando o paradigma do framework, alinhado com a arquitetura MVC. Essa abordagem propiciou o desenvolvimento rápido, seguro e performático da aplicação, resultando em um código limpo e de fácil manutenção. Trabalhar com códigos sem qualquer arquitetura é uma tarefa árdua, principalmente quando o projeto é de grande porte e não possui uma estrutura para gerenciamento.

Figura 11 – Arquitetura MVC no Laravel



Fonte: Elaborado pelo próprio autor (2024).

Ao mencionar o Modelo-Visão-Controlador (MVC) é essencial esclarecer a estrutura do projeto, indicando onde cada parte do código se encaixa.

A camada de modelo é essencialmente onde as operações relacionadas ao banco de dados são realizadas, tais como inserir, editar e atualizar dados. A pasta contendo os arquivos de modelo está localizada em: `app/Models`. Esta organização foi introduzida no Laravel a partir da versão 8. Anteriormente, os modelos eram geralmente colocados diretamente na pasta `app`. A mudança foi efetuada com o objetivo de organizar melhor a estrutura do projeto e separar claramente os modelos do restante da aplicação. Como mencionado já neste trabalho, cada tabela no banco de dados representa uma classe modelo. Ou seja, foi criada uma classe para Empresas, Arquivos, Localizações, Notas de Empenho, Pagamentos, Relatórios e Usuários.

A camada de visualização foi responsável pela exibição de dados e pela renderização da interface do usuário. O caminho dos itens de visualização neste projeto é o diretório `resources/views/dashboard`. Foram criados nove arquivos com a extensão `*.blade.php`, cada um com sua responsabilidade específica. Para fins de reutilização de código, existe a view `app.blade.php`, que abrange o cabeçalho, o menu e o rodapé. As demais views estendem-se a partir da view principal, conforme os trechos abaixo:

Figura 12 – Exemplo da página principal

```
resources > views > layouts > app.blade.php > html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10   @yield('content')
11 </body>
12 </html>
```

Fonte: Elaborado pelo próprio autor (2024).

Figura 13 – Exemplo da página secundária

```
resources > views > dashboard > page.blade.php > ...
1 {{-- conteúdo das demais páginas --}}
2 @extends('layouts.app')
3 @section('content')
4   <h2>conteúdo das demais páginas </h2>
5 @endsection
```

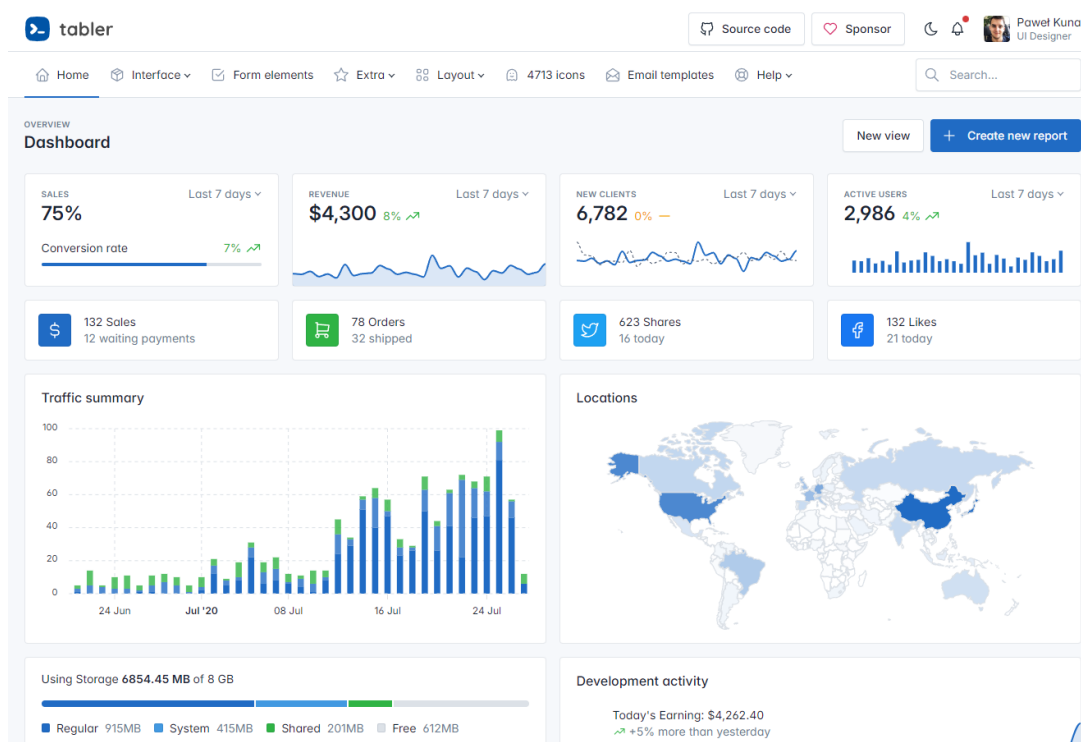
Fonte: Elaborado pelo próprio autor (2024).

A camada de controlador, por padrão, é armazenada no diretório `app/Http/Controllers`. Essa camada é responsável por definir a lógica de tratamento das solicitações. Cada solicitação possui um arquivo único, por questão de boas práticas. Consequentemente, o `UserController.php` lida apenas com ações relacionadas aos usuários, o `ReportController.php` lida apenas com ações relacionadas aos relatórios e assim por diante. Por padrão, todas as classes de controladores possuem os métodos `index`, `create`, `store`, `show`, `edit`, `update` e `destroy`. Esses métodos são típicos de ações para os famosos CRUD e o framework facilita essa chamada de ações na rota, com uma simples declaração: `Route::resource(users, UserController::class);` Esse modelo de roteamento já mapeia todos os métodos padrão do controlador. Uma boa prática que, além disso, mantém o código mais limpo e enxuto.

6.4 FRONT-END

O projeto empregou a tecnologia `tabler.io` para o front-end, cujo lema é fornecer um modelo de painel premium e de código aberto com uma interface de usuário responsiva de alta qualidade. Este tema foi instalado no projeto por meio do gerenciador de pacotes para Node.JS (`npm`), conforme o commit de código `cae3dc3`. O kit de interface de usuário `tabler.io` é gratuito e de código aberto, é baseado no `Bootstrap 5` e fornece centenas de componentes responsivos e vários layouts. O desenvolvimento da ferramenta foi caracterizado por uma agilidade considerável, uma vez que foi possível aproveitar componentes pré-existent, tais como páginas de login, redefinição de senha, tabelas `html`, modais e `cards`, entre outros.

Figura 14 – Página principal do kit `tabler.io`



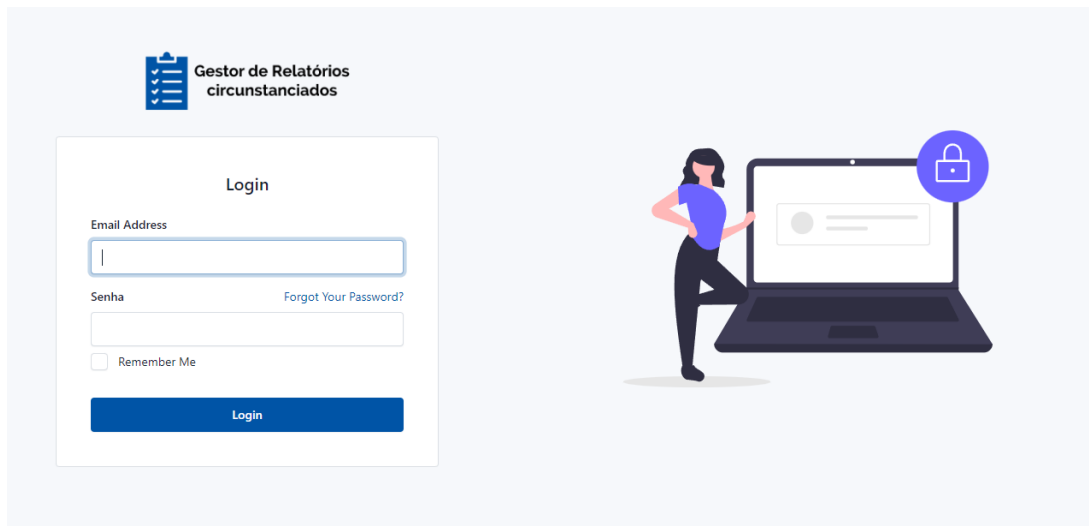
Fonte: Capturado pelo próprio autor (2024).

7 RESULTADOS

7.1 PRINCIPAIS TELAS DO SISTEMA

O kit front-end `tabler.io` fornece uma variedade de layouts, entre os quais os mais significativos são as páginas de autenticação e redefinição de senha. Nesta primeira versão, optamos por não incluir a autenticação via `Oauth Google`, devido ao curto prazo de desenvolvimento e a existência de outras funcionalidades mais importantes. A figura 15 demonstra a tela de autenticação, para o login, é necessário informar apenas o endereço de email corporativo e a senha, que é fornecida pelo administrador do sistema. Foi necessário adaptar o tema front-end com o `laravel`, visto que

Figura 15 – Página de login do sistema Gestor de Relatórios Circunstanciados

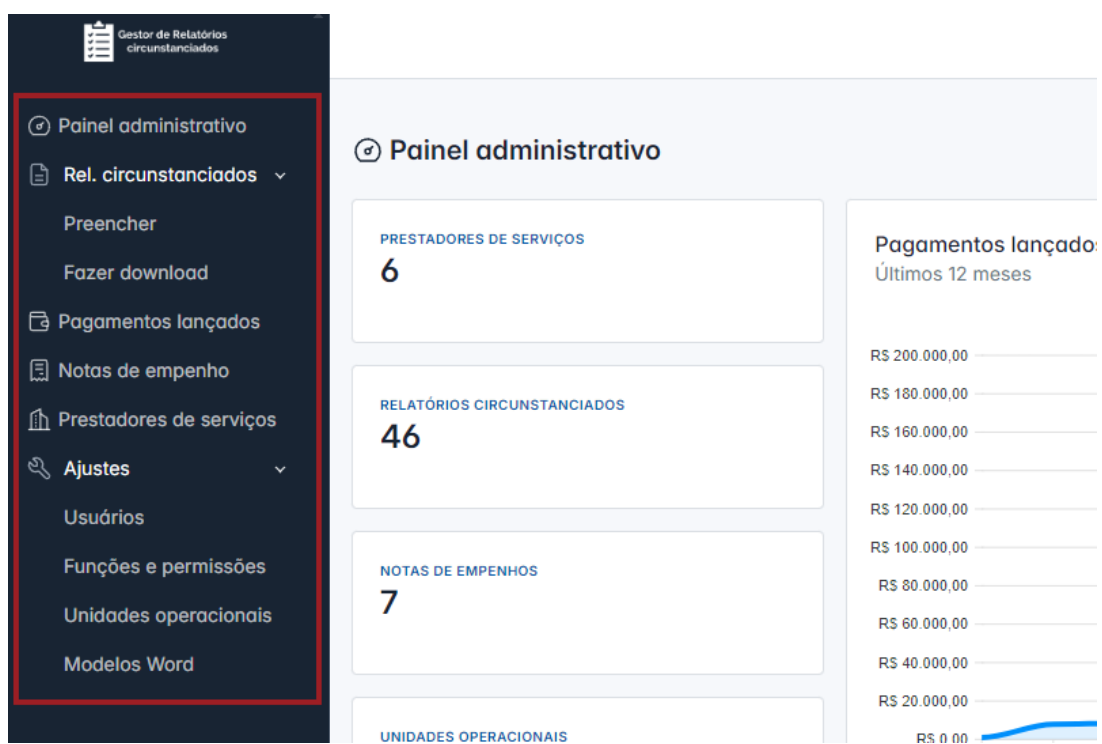


Fonte: Capturado pelo próprio autor (2024).

o `tabler.io` é apenas um conjunto `HTML5`, `CSS3` e `JavaScript` sem integração com a renderização `blade/laravel`.

Ao realizar o acesso à área restrita, conforme figura 16, o usuário encontra em sua lateral esquerda, o menu de navegação principal. Esse menu, é de acordo com os níveis de acesso. O usuário de perfil administrador, tem acesso a todas as funcionalidades disponíveis do sistema.

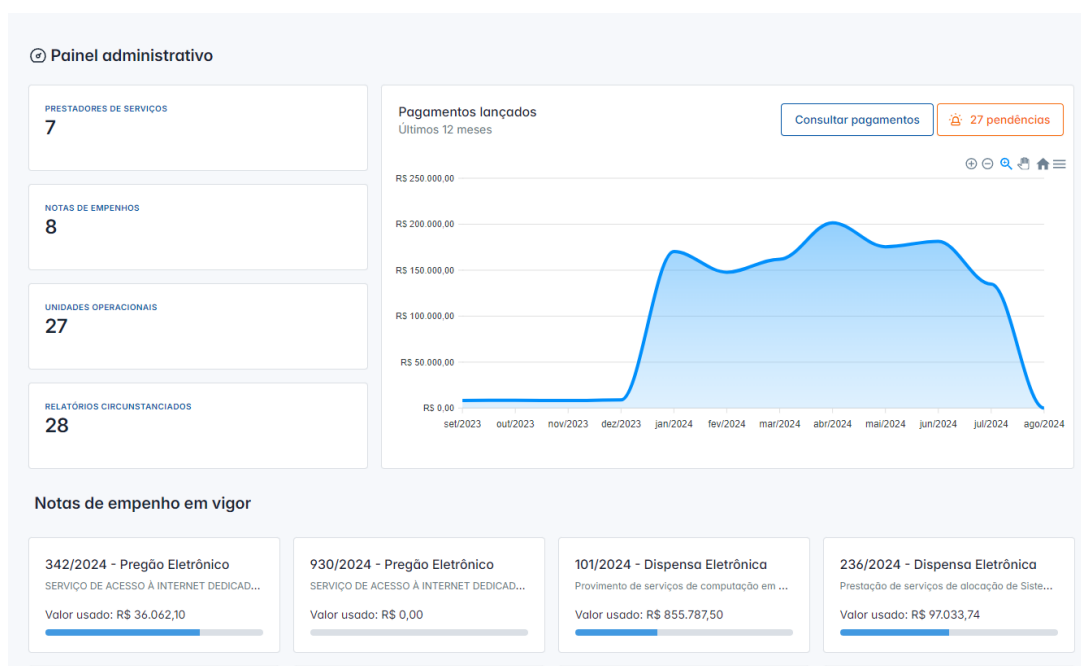
Figura 16 – Menu de navegação lateral com todas as rotas disponíveis



Fonte: Capturado pelo próprio autor (2024).

O acesso ao painel administrativo conforme figura 17, contém as principais informações úteis, com o histórico de lançamento de pagamentos dos últimos 12 meses, sendo possível obter detalhes mês a mês, além de mostrar a quantidade de pendências, que são meses em que não houveram pagamentos. Cada widget disponível na tela, traz detalhes da quantidade de prestadores de serviços, relatórios circunstanciados, notas de empenho e unidades operacionais.

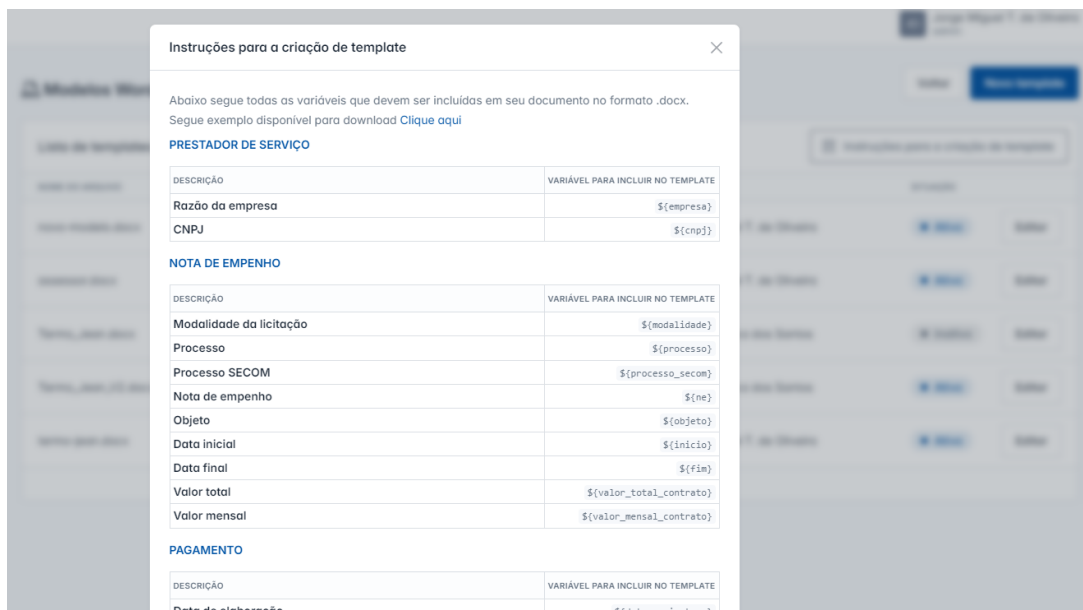
Figura 17 – Painel administrativo com a visão geral



Fonte: Capturado pelo próprio autor (2024).

No que se refere ao modelo word, no próprio sistema de gerenciamento de relatórios circunstanciados, há instruções para confeccionar um modelo padrão para os relatórios. Inclusive, há um modelo pronto para ser baixado e utilizado como base. Ele está bem documentado com todas as variáveis possíveis de cada objeto do banco de dados. Para utilizá-lo, é necessário utilizar a variável neste formato: $\${foo}$. Na área ilustrada na Figura 18, o gestor tem a possibilidade de criar um modelo personalizado para um relatório específico.

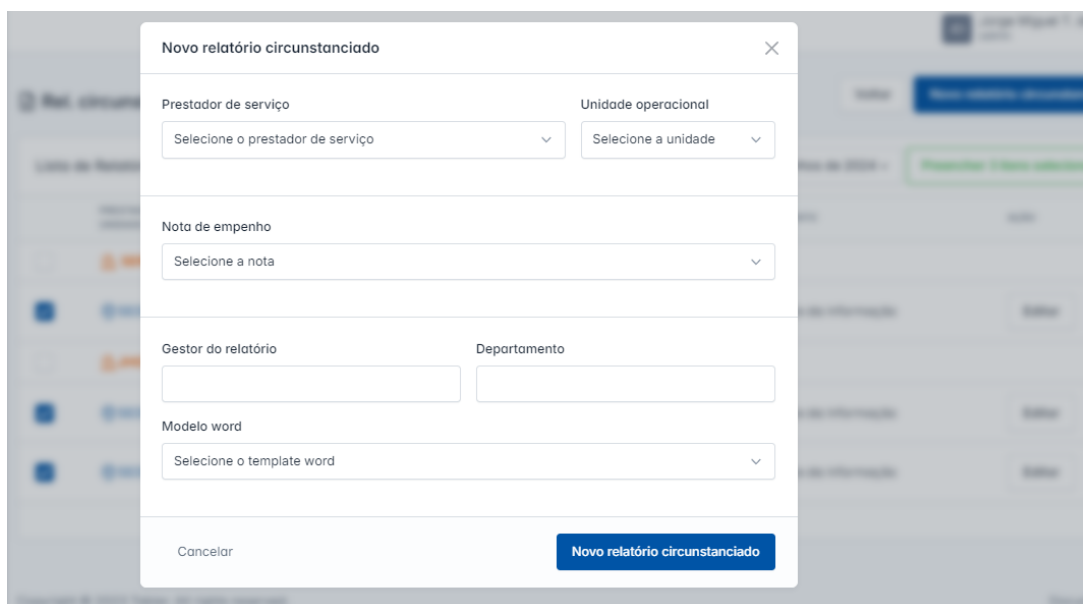
Figura 18 – Adicionar template para o relatório circunstanciado



Fonte: Capturado pelo próprio autor (2024).

A figura 19 demonstra como incluir um novo relatório circunstanciado. É necessário que os seguintes itens já estejam registrados no sistema: prestador de serviços, unidade operacional, nota de empenho dedicada à prestação de serviços, modelo word, que é basicamente um modelo com variáveis, conforme mencionado no parágrafo anterior, nome do gestor do relatório e departamento.

Figura 19 – Adicionar novo relatório circunstanciado

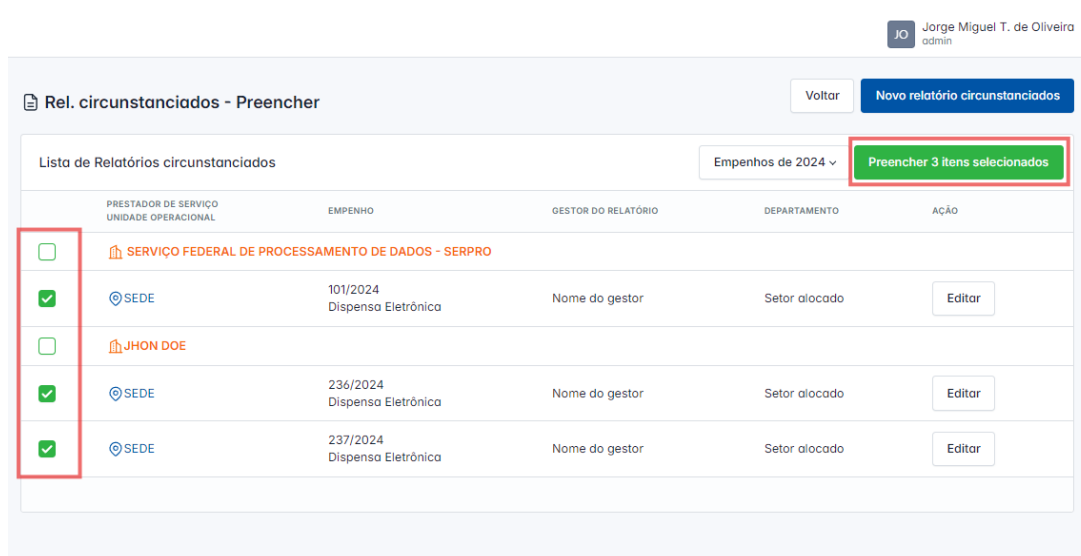


Fonte: Capturado pelo próprio autor (2024).

Após a inclusão do relatório, é possível adicionar informações de pagamento pelo serviço prestado, que pode ser único ou recorrente, como contas de consumo. Antes da implementação do sistema informatizado na organização, esses detalhes eram inseridos manualmente no relatório

circunstanciado, individualmente. Após a implementação, é possível selecionar vários fornecedores simultaneamente e gerar diversos relatórios ao mesmo tempo. A figura a seguir apresenta a lista de relatórios circunstanciados disponíveis para a inclusão de informações de pagamento.

Figura 20 – Itens selecionados para o preenchimento



PRESTADOR DE SERVIÇO	UNIDADE OPERACIONAL	EMPENHO	GESTOR DO RELATÓRIO	DEPARTAMENTO	AÇÃO
<input type="checkbox"/>	SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS - SERPRO				
<input checked="" type="checkbox"/>	SEDE	101/2024 Dispensa Eletrônica	Nome do gestor	Setor alocado	Editar
<input type="checkbox"/>	JHON DOE				
<input checked="" type="checkbox"/>	SEDE	236/2024 Dispensa Eletrônica	Nome do gestor	Setor alocado	Editar
<input checked="" type="checkbox"/>	SEDE	237/2024 Dispensa Eletrônica	Nome do gestor	Setor alocado	Editar

Fonte: Capturado pelo próprio autor (2024).

A Figura 20 apresenta a relação de relatórios circunstanciados organizada de modo a ser agrupada pelo prestador dos serviços. No texto de cor laranja, é identificado o prestador de serviços, enquanto no texto de cor azul, é identificada a unidade operacional na qual o serviço está sendo prestado. Esta lista contém detalhes resumidos da nota de empenho, bem como a indicação do gestor responsável pela fiscalização do contrato. Ao lado do texto de fornecedor de serviços, decidimos incluir uma caixa de seleção que, ao ser marcada, seleciona automaticamente os itens abaixo na sequência. Isso simplifica o processo de preenchimento de vários relatórios simultaneamente. Após selecionar, basta clicar em “Preencher x itens selecionados”, que na próxima tela haverá os campos necessários para preencher os dados do pagamento que está sendo feito. É importante ressaltar que a ferramenta Gestor de Relatórios Circunstanciados não possui integração com plataformas de pagamento. A ferramenta é estritamente um aplicativo que auxilia na elaboração de Relatórios Circunstanciados, com a finalidade de gerar um relatório referente ao serviço prestado.

Figura 21 – Campos necessários para a confecção do relatório

Rel. circunstanciados - Preencher Voltar

SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS - SERPRO (verifique os 4 campos) JHON DOE Preencher relatórios em massa

SEDE - Descrição do serviço prestado, conforme nota de empenho

NOTA DE EMPENHO 236/2024	PROCESSO SECOM 157/2023	MODALIDADE DA LICITAÇÃO E PROCESSO Dispensa Eletrônica - 157/2023	ELABORADO POR: Nome do gestor - Setor alocado
VALOR TOTAL R\$ 12.000,00	VALOR MENSAL R\$ 1.000,00	PERÍODO 12/01/24 até 12/01/25	Adicionar nova linha

NÚMERO DA NOTA FISCAL/FATURA	PERÍODO DE EXECUÇÃO	VALOR DA NOTA FISCAL/FATURA - R\$	VENCIMENTO	DATA DE ELABORAÇÃO	AÇÃO
123456 ✓	julho de 2024 ✓	1.000,00 ✓	07/07/2024 ✓	01/07/2024 ✓	

SEDE - Descrição do serviço prestado, conforme nota de empenho

NOTA DE EMPENHO 237/2024	PROCESSO SECOM 0157/2023	MODALIDADE DA LICITAÇÃO E PROCESSO Dispensa Eletrônica - 0157/2023	ELABORADO POR: Nome do gestor - Setor alocado
VALOR TOTAL R\$ 24.000,00	VALOR MENSAL R\$ 2.000,00	PERÍODO 12/01/24 até 12/01/25	Adicionar nova linha

NÚMERO DA NOTA FISCAL/FATURA	PERÍODO DE EXECUÇÃO	VALOR DA NOTA FISCAL/FATURA - R\$	VENCIMENTO	DATA DE ELABORAÇÃO	AÇÃO
654321 ✓	julho de 2024 ✓	2.000,00 ✓	07/07/2024 ✓	01/07/2024 ✓	

Adicionar pagamentos

Fonte: Capturado pelo próprio autor (2024).

A figura 21 exibe a tela de preenchimento das informações dos pagamentos. Os dados de pagamentos são os seguintes:

- Número da nota ou fatura
- Período de execução
- Valor
- Vencimento
- Data de elaboração do preenchimento

Todos os campos com possibilidade de preenchimento estão sujeitos à validação. No caso de não preenchimento ou preenchimento não conforme, o campo será destacado na cor vermelha ao tentar adicionar informações de pagamento. Quando o gestor for elaborar um relatório envolvendo múltiplos prestadores simultaneamente, a visualização será organizada em abas, conforme ilustrado na figura acima. Ao clicar no nome do fornecedor, serão exibidos os dados de pagamento específicos daquele fornecedor. Na área de ação, existem dois botões: um para adicionar eventuais ocorrências (quando o serviço não foi prestado de forma adequada e houve falhas) e o botão para remover o pagamento. Ao submeter o preenchimento, o sistema exibe a mensagem “Pagamento(s) adicionado(s) com sucesso!”. Em seguida é direcionado a nova página de download.

Figura 22 – Página para fazer o download em .docx do relatório

JO Jorge Miguel T. de Oliveira
admin

Rel. circunstanciados - Fazer download Voltar

Busca rápida

Prestador de serviço: Todos

Período de execução: ----- de ----

Período de criação do RC (intervalo): 07/07/2024 a 13/07/2024

Filtrar

Histórico de pagamentos Download dos 14 itens selecionados

PAGAMENTO	PROTOCOLO	DIGITALIZAÇÃO
<input checked="" type="checkbox"/> Relatórios circunstanciados gerados em 12/07/2024		
<input checked="" type="checkbox"/> TELEFÔNICA BRASIL S/A R\$ 328,00 - SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO - ARAÇATUBA - JUNHO/2024	ADM-2024/257047	<input type="button" value="Incluir anexo"/>
<input checked="" type="checkbox"/> TELEFÔNICA BRASIL S/A R\$ 328,00 - SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO - CARAGUATATUBA - JUNHO/2024	ADM-2024/257049	<input type="button" value="Incluir anexo"/>
<input checked="" type="checkbox"/> TELEFÔNICA BRASIL S/A R\$ 328,00 - SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO - FRANCA - JUNHO/2024	ADM-2024/257050	<input type="button" value="Incluir anexo"/>
<input checked="" type="checkbox"/> TELEFÔNICA BRASIL S/A R\$ 328,00 - SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO - ITU - JUNHO/2024	ADM-2024/257051	<input type="button" value="Incluir anexo"/>
<input checked="" type="checkbox"/> TELEFÔNICA BRASIL S/A R\$ 316,00 - SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO - JUNDIAÍ - JUNHO/2024	ADM-2024/257052	<input type="button" value="Incluir anexo"/>
<input checked="" type="checkbox"/> TELEFÔNICA BRASIL S/A		

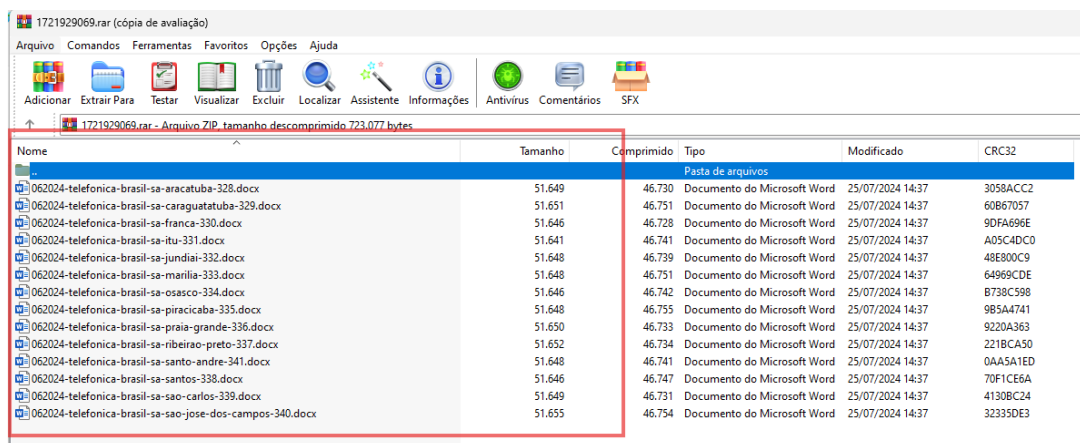
Fonte: Capturado pelo próprio autor (2024).

Após o preenchimento das informações referentes aos pagamentos, a etapa seguinte consiste em efetuar o download dos relatórios detalhados. Nesta página, incluímos um filtro para pesquisas, permitindo a busca por fornecedor, mês de referência (período de execução) e intervalo

de criação do relatório. Essa funcionalidade de busca rápida foi de grande utilidade, pois anteriormente não havia um controle sistemático, o que demandava um tempo considerável para localizar qualquer pagamento contestado. Após selecionar os relatórios conforme indicado na Figura 22, há um botão para efetuar o download dos itens específicos. Basicamente, o software, em sua regra de negócio, realizará o processo de montagem de cada relatório e confeccionará de acordo com o modelo relacionado. Após a execução do clique, será iniciado o download dos relatórios compactados em um único arquivo com a extensão RAR.

A figura 23 exhibe os relatórios compactados em um arquivo *.rar. Com o intuito de facilitar a identificação do relatório antes de sua abertura, o sistema registrou o nome dos arquivos com o mês de referência, o prestador de serviço, a localização e um código identificador, que é auto incremento na base de dados. É importante ressaltar que cada relatório possui um modelo específico e características próprias.

Figura 23 – Relatórios confeccionados

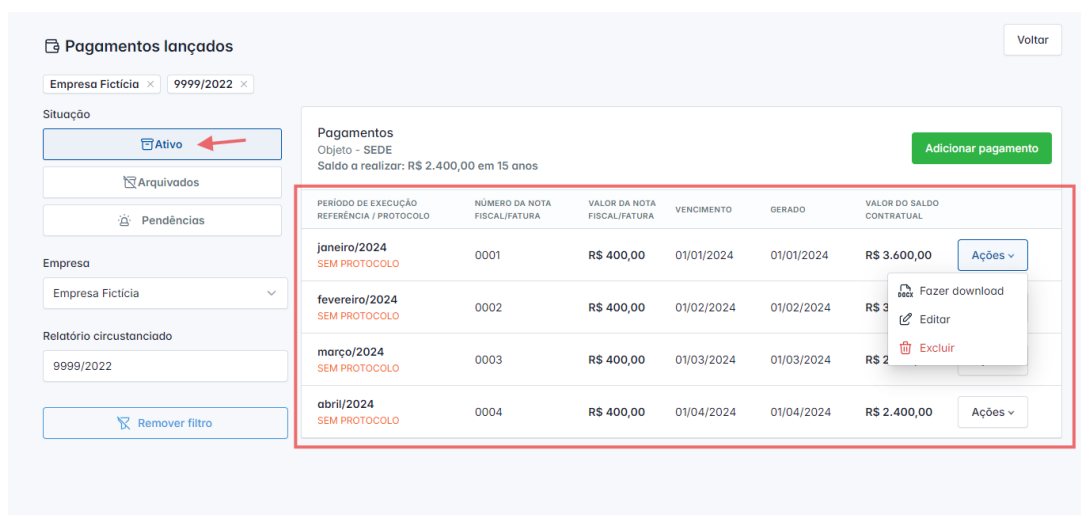


Nome	Tamanho	Comprimido	Tipo	Modificado	CRC32
Pasta de arquivos					
062024-telefonica-brasil-sa-aracatuba-328.docx	51.649	46.730	Documento do Microsoft Word	25/07/2024 14:37	3058ACC2
062024-telefonica-brasil-sa-caraguatuba-329.docx	51.651	46.751	Documento do Microsoft Word	25/07/2024 14:37	60B67057
062024-telefonica-brasil-sa-franca-330.docx	51.646	46.728	Documento do Microsoft Word	25/07/2024 14:37	9DFA696E
062024-telefonica-brasil-sa-itu-331.docx	51.641	46.741	Documento do Microsoft Word	25/07/2024 14:37	A05C4DC0
062024-telefonica-brasil-sa-jundiiai-332.docx	51.648	46.739	Documento do Microsoft Word	25/07/2024 14:37	48E800C9
062024-telefonica-brasil-sa-marilia-333.docx	51.648	46.751	Documento do Microsoft Word	25/07/2024 14:37	64969CDE
062024-telefonica-brasil-sa-osasco-334.docx	51.646	46.742	Documento do Microsoft Word	25/07/2024 14:37	B738C598
062024-telefonica-brasil-sa-piracicaba-335.docx	51.648	46.755	Documento do Microsoft Word	25/07/2024 14:37	9B5A4741
062024-telefonica-brasil-sa-praia-grande-336.docx	51.650	46.733	Documento do Microsoft Word	25/07/2024 14:37	9220A363
062024-telefonica-brasil-sa-ribeirao-preto-337.docx	51.652	46.734	Documento do Microsoft Word	25/07/2024 14:37	221BCA50
062024-telefonica-brasil-sa-santo-andre-341.docx	51.648	46.741	Documento do Microsoft Word	25/07/2024 14:37	0AA5A1ED
062024-telefonica-brasil-sa-santos-338.docx	51.646	46.747	Documento do Microsoft Word	25/07/2024 14:37	70F1CE6A
062024-telefonica-brasil-sa-sao-carlos-339.docx	51.649	46.731	Documento do Microsoft Word	25/07/2024 14:37	4130BC24
062024-telefonica-brasil-sa-sao-jose-dos-campos-340.docx	51.655	46.754	Documento do Microsoft Word	25/07/2024 14:37	32335DE3

Fonte: Capturado pelo próprio autor (2024).

A figura 24 apresenta o histórico de pagamentos, possibilitando a filtragem por situação (ativo/arquivado), prestador de serviço e relatório circunstanciado (nota de empenho/localidade). A área destacada em vermelho representa o histórico, detalhando as informações referentes aos pagamentos. Esses detalhes são impressos no relatório mensalmente.

Figura 24 – Histórico de pagamentos



Pagamentos lançados Voltar

Empresa Fictícia x 9999/2022 x

Situação

- Ativo
- Arquivados
- Pendências

Empresa

Empresa Fictícia

Relatório circunstanciado

9999/2022

Remover filtro

Pagamentos
Objeto - SEDE
Saldo a realizar: R\$ 2.400,00 em 15 anos Adicionar pagamento

PERÍODO DE EXECUÇÃO REFERÊNCIA / PROTOCOLO	NÚMERO DA NOTA FISCAL/FATURA	VALOR DA NOTA FISCAL/FATURA	VENCIMENTO	GERADO	VALOR DO SALDO CONTRATUAL	Ações
janeiro/2024 SEM PROTOCOLO	0001	R\$ 400,00	01/01/2024	01/01/2024	R\$ 3.600,00	Ações
fevereiro/2024 SEM PROTOCOLO	0002	R\$ 400,00	01/02/2024	01/02/2024	R\$ 3.200,00	Fazer download Editar
março/2024 SEM PROTOCOLO	0003	R\$ 400,00	01/03/2024	01/03/2024	R\$ 2.800,00	Excluir
abril/2024 SEM PROTOCOLO	0004	R\$ 400,00	01/04/2024	01/04/2024	R\$ 2.400,00	Ações

Fonte: Capturado pelo próprio autor (2024).

Ao clicar em ações, tem as seguintes opções:

- Fazer download: Realizar o download individual do item específico no formato .docx.
- Editar: Editar informação do pagamento, que pode ser protocolo, numero da nota, valor e datas.
- Excluir: Excluir definitivamente um pagamento específico.

A figura 25 destaca-se como uma das principais funcionalidades, pois permite verificar se algum pagamento devido não foi realizado. A falta de pagamento pode ocorrer por esquecimento, não emissão por parte do prestador ou preenchimento divergente.

Figura 25 – Pendências de pagamentos

MÊS	RELATÓRIOS SEM PAGAMENTO
janeiro/2024	Não há pendências
fevereiro/2024	Não há pendências
março/2024	Não há pendências
abril/2024	Não há pendências
maio/2024	1 pendências Empresa Fictícia <input type="checkbox"/> 9999/2022 SEDE Objeto
junho/2024	5 pendências TELEFÔNICA BRASIL S/A <input type="checkbox"/> 342/2024 CENTRO SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO <input type="checkbox"/> 342/2024 DIVIDA ATIVA SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO <input type="checkbox"/> 342/2024 NOVA SEDE SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO <input type="checkbox"/> 342/2024 NOVA SEDE SERVIÇO DE ACESSO À INTERNET DEDICADA IP FIXO Empresa Fictícia <input type="checkbox"/> 9999/2022 SEDE Objeto

Fonte: Capturado pelo próprio autor (2024).

Nesta tabela HTML, são apresentadas as pendências mês a mês. A lógica consiste em verificar, para cada mês de referência, se há pagamentos em todos os relatórios circunstanciados ativos no sistema. Esse recurso trouxe melhorias significativas, pois anteriormente os gestores só eram informados sobre uma pendência após uma nova cobrança por parte do prestador de serviço, o que frequentemente resultava em boletos reenviados com juros.

7.2 INDICADORES DE PRODUTIVIDADE APÓS IMPLEMENTAÇÃO DO SOFTWARE

Após a implementação do software que auxiliou na informatização da rotina de gerenciamento de relatórios circunstanciados, observamos uma melhoria significativa nos procedimentos realizados.

Tabela 6 – Desempenho coletado após implementação do software

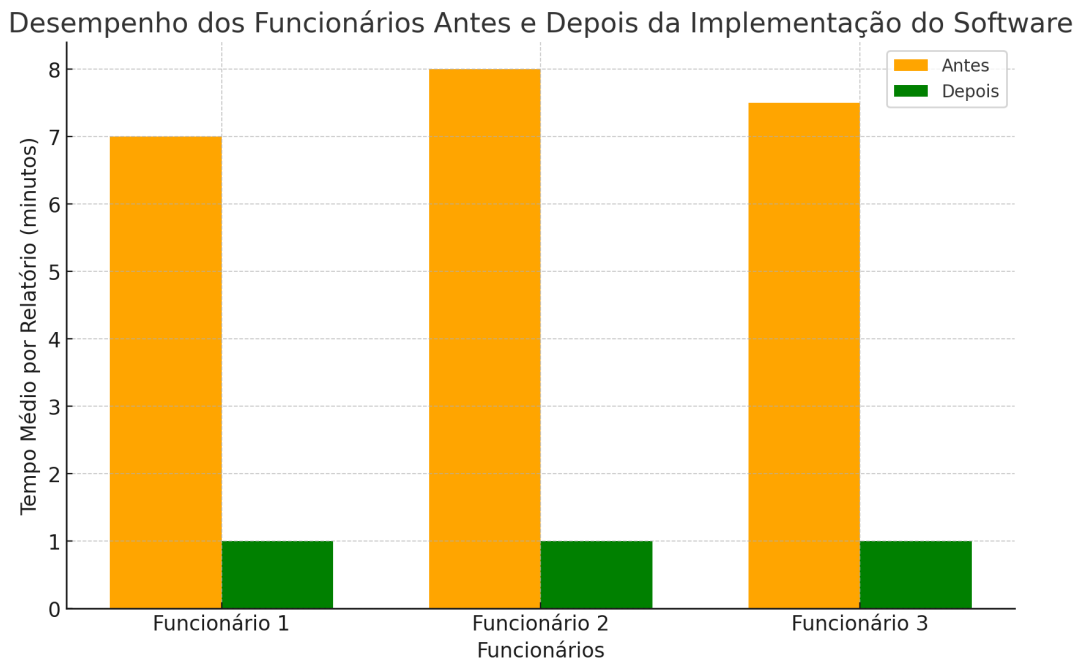
	Funcionário 1	Funcionário 2	Funcionário 3	Média
Tempo Médio por Relatório	1 minuto	1 minuto	1 minuto	1 minuto
Número de Relatórios por Período	20	20	20	-
Qualidade dos Relatórios (re-feitos)	1 (5%)	1 (5%)	0 (0%)	aproximadamente 3.33%
Interrupções Durante o Processo	0	0	0	0
Satisfação do Funcionário	Procedimento eficiente	Procedimento eficiente	Procedimento eficiente	-

Fonte: Elaborado pelo próprio autor (2024).

A análise dos dados coletados antes e após a implementação do software revela mudanças significativas em diversos aspectos do desempenho dos funcionários. As tabelas 4 e 6 fornecem uma visão detalhada sobre o impacto da nova ferramenta nas métricas de tempo, qualidade, interrupções e satisfação dos funcionários.

Para avaliar o impacto da implementação do software, foram coletados dados de desempenho antes e após sua introdução. A seguir, apresentam-se os resultados obtidos.

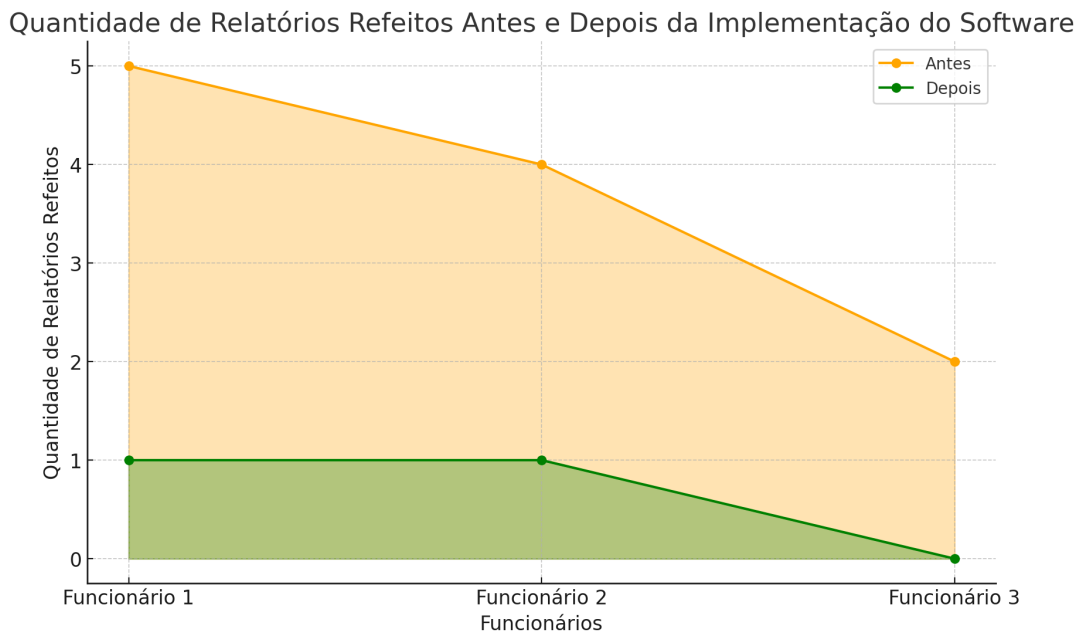
Figura 26 – Comparativo de confecção dos relatórios dos funcionários antes e depois da implementação do software (média)



Fonte: Capturado pelo próprio autor (2024).

A implementação da ferramenta Gestor de relatório circunstanciado resultou em uma expressiva melhoria na eficiência das atividades executadas pelos gestores. Antes da implementação, o tempo médio necessário para completar cada atividade variava entre 6 a 9 minutos, conforme os dados coletados na tabela 4. Após a adoção do software, esse tempo foi drasticamente reduzido para uma média de 1 minuto por atividade. Esta redução representa uma significativa otimização dos processos, evidenciando o impacto positivo do software na produtividade.

Figura 27 – Comparativo da quantidade de relatórios refeitos antes e depois da implementação do software



Fonte: Capturado pelo próprio autor (2024).

A implementação do novo software resultou em uma melhoria significativa na qualidade dos relatórios circunstanciados. A redução do número de relatórios refeitos, de 18,33% para 3,33%, evidencia a eficácia do software em padronizar e melhorar o processo de criação de documentos. Notavelmente, o terceiro funcionário não teve nenhum relatório refeito após a mudança, o que destaca ainda mais o impacto positivo da nova ferramenta. Este aprimoramento não só aumentou a eficiência dos funcionários, mas também assegurou a produção de relatórios de alta qualidade, minimizando a necessidade de retrabalho e fortalecendo a confiabilidade dos processos operacionais.

8 CONCLUSÃO

O presente trabalho teve como objetivo principal o desenvolvimento e a implementação de uma aplicação na CRECI-SP, visando melhorias no desenvolvimento de relatórios circunstanciados. A avaliação dos resultados obtidos antes e após a implementação, com base em uma amostra de 3 funcionários e 120 relatórios circunstanciados, evidenciou melhorias significativas nos processos.

Embora a ferramenta tenha sido desenvolvida para atender às necessidades específicas da instituição, sua arquitetura modular permite adaptações para outros contextos, tornando-a potencialmente útil para o controle de informações sobre pagamentos a fornecedores em diversas empresas.

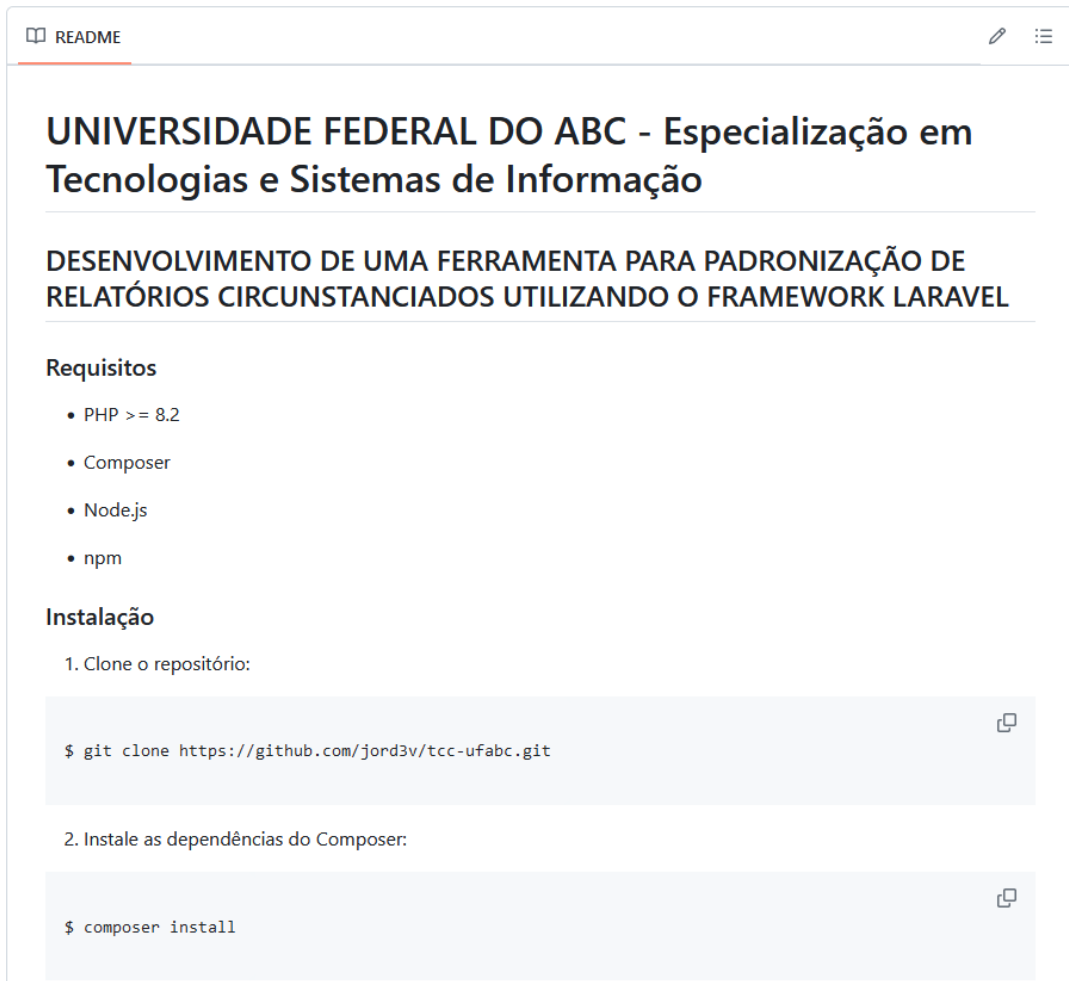
A implementação da ferramenta resultou em uma padronização eficaz dos relatórios, redução do tempo de elaboração e minimização de inconsistências, retrabalho e resultou na satisfação de forma eficiente por parte dos funcionários.

A escolha do framework Laravel para o desenvolvimento desta aplicação foi estratégica, uma vez que sua arquitetura robusta e modular facilitou a organização do código e promoveu o cumprimento de boas práticas de desenvolvimento, como o padrão MVC. A extensa comunidade e a ampla documentação disponível para o Laravel garantiram suporte contínuo durante o desenvolvimento, agilizando a resolução de problemas e a implementação das funcionalidades desejadas.

Apêndices

APÊNDICE A – Repositório do projeto publicado no GitHub

Ao acessar o projeto, você encontrará todo o código necessário. A primeira página fornece detalhes sobre os requisitos mínimos para a execução da aplicação, bem como instruções para instalação, primeiro acesso e colaboração. (MIGUEL, 2024)



The image shows a screenshot of a GitHub README file. At the top left, there is a 'README' tab with a square icon. At the top right, there are icons for editing and a menu. The main content starts with a large heading: 'UNIVERSIDADE FEDERAL DO ABC - Especialização em Tecnologias e Sistemas de Informação'. Below this is a subtitle: 'DESENVOLVIMENTO DE UMA FERRAMENTA PARA PADRONIZAÇÃO DE RELATÓRIOS CIRCUNSTANCIADOS UTILIZANDO O FRAMEWORK LARAVEL'. The 'Requisitos' section lists: PHP >= 8.2, Composer, Node.js, and npm. The 'Instalação' section has two steps: 1. Clone o repositório: followed by a code block containing '\$ git clone https://github.com/jord3v/tcc-ufabc.git' with a copy icon. 2. Instale as dependências do Composer: followed by a code block containing '\$ composer install' with a copy icon.

README

UNIVERSIDADE FEDERAL DO ABC - Especialização em Tecnologias e Sistemas de Informação

DESENVOLVIMENTO DE UMA FERRAMENTA PARA PADRONIZAÇÃO DE RELATÓRIOS CIRCUNSTANCIADOS UTILIZANDO O FRAMEWORK LARAVEL

Requisitos

- PHP >= 8.2
- Composer
- Node.js
- npm

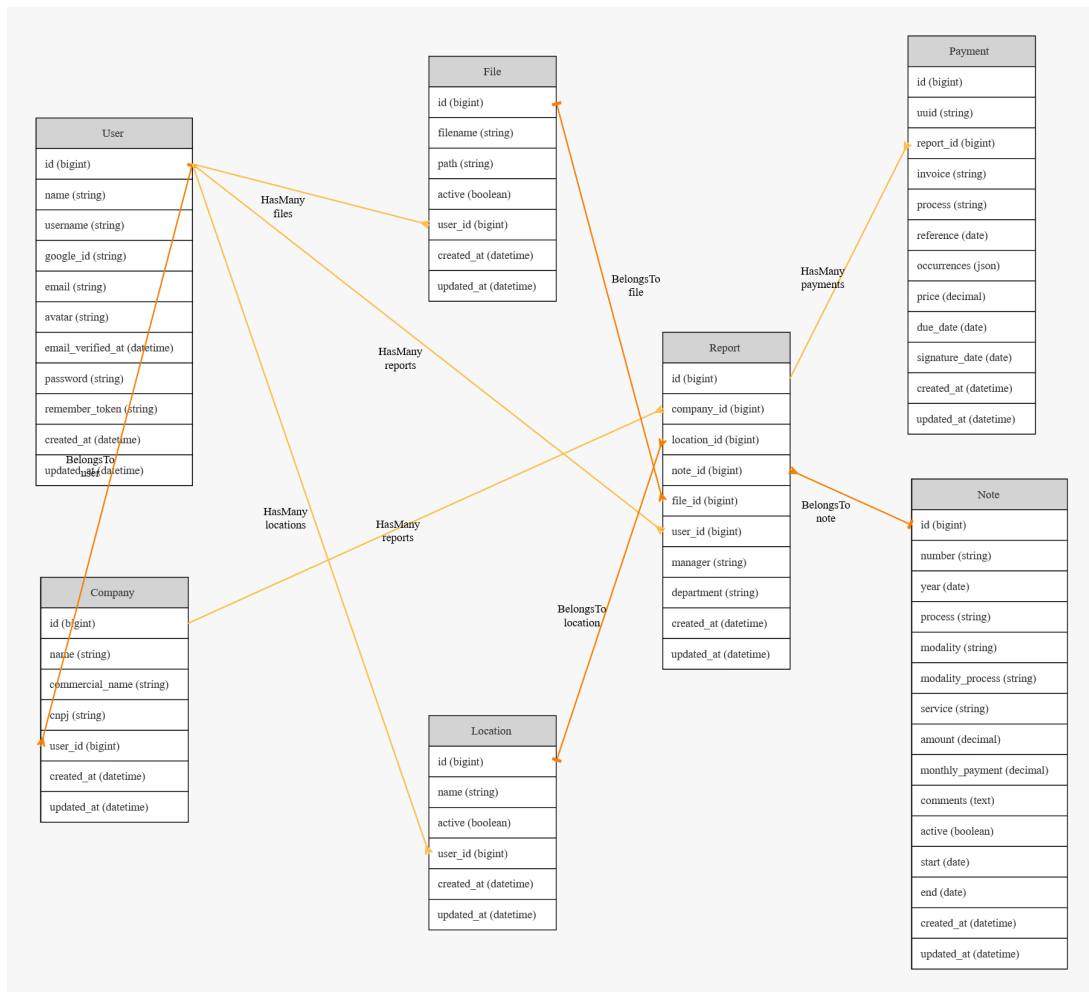
Instalação

1. Clone o repositório:

```
$ git clone https://github.com/jord3v/tcc-ufabc.git
```
2. Instale as dependências do Composer:

```
$ composer install
```

APÊNDICE B – Entidade Relacionamento Final



Referências

- ALCARAZ, P. C. F. **Entity Framework, uma Introdução ao ORM utilizando o Code First**. Medianeira, 2016. 74 p. Citado na página 14.
- AUGUSTO, J. **MySQL 5.5 - Interativo**. 1. ed. São Paulo: Érica, 2011. ISBN 9788536503851. Citado na página 12.
- BUILD software better, together. 2024. Disponível em: <<https://github.com/topics/javascript>>. Acesso em: abril. 2024. Citado na página 8.
- CAMARGO, M. R. **Gerenciamento de Projetos: Fundamentos e Prática Integrada**. 2. ed. Rio de Janeiro, RJ: Atlas, 2018. 74 p. ISBN 978-85-352-8023-4. Citado na página 30.
- CARDOSO, L. D. C. **Frameworks Back End**. 1. ed. São Paulo: Platos Soluções Educacionais S.A., 2021. ISBN 978-65-89965-87-9. Citado na página 12.
- CARLOS, R. N. **Estudo de tecnologias atuais para elaboração de arquitetura de sistemas desenvolvidos em PHP**. Brasília, 2017. 60 p. Citado na página 10.
- DANIELE, C. **O Que Há de Novo no PHP 8 (Características, Melhorias e o Compilador JIT)**. 2020. Disponível em: <<https://kinsta.com/pt/blog/php-8/#desempenho-do-php-8-benchmarks>>. Acesso em: abril. 2024. Citado na página 9.
- FLANAGAN, D. **JavaScript - O guia definitivo**. Porto Alegre, RS: Bookman Companhia Editora Ltda., uma empresa do Grupo A Educação AS, 2013. Citado na página 7.
- MIGUEL, J. **DESENVOLVIMENTO DE UMA FERRAMENTA PARA PADRONIZAÇÃO DE RELATÓRIOS CIRCUNSTANCIADOS UTILIZANDO O FRAMEWORK LARAVEL**. 2024. Disponível em: <<https://github.com/jord3v/tcc-ufabc>>. Acesso em: Agosto. 2024. Citado na página 51.
- NACIONAL, C. **Termo: Relatório Circunstanciado - Glossário de Termos Legislativos - Congresso Nacional**. 2024. Disponível em: <https://www.congressonacional.leg.br/legislacao-e-publicacoes/glossario-legislativo/-/legislativo/termo/relatorio_circunstanciado>. Acesso em: abril. 2024. Citado na página 1.
- NETWORK, M. D. **CSS | MDN**. 2022. Documentação de tecnologias da web. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: abril. 2024. Citado na página 4.
- NETWORK, M. D. **HTML: Linguagem de Marcação de Hipertexto**. 2023. Documentação de tecnologias da web. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: abril. 2024. Citado na página 3.
- OLIVEIRA, D. d. P. R. **Sistemas, organização & métodos: uma abordagem gerencial**. 21. ed. [s.l.]: Editora Atlas S.A., 2000. Citado na página 2.
- PHP-FIG. **PHP Framework Interoperability Group**. 2024. <<https://www.php-fig.org/>>. Acessado em: 17-ago-2024. Citado na página 11.
- SATYA. **Difference between Inline, Internal and External CSS**. 2024. Disponível em: <<https://www.geeksforgeeks.org/difference-between-inline-internal-and-external-css/>>. Acesso em: abril. 2024. Citado na página 6.

SILVA, L. e. a. **Banco de dados não relacional**. Porto Alegre: SAGAH, 2021. ISBN 9786556901534. Citado na página [13](#).