

## **FRAMEWORK PARA PADRONIZAÇÃO E UNIFICAÇÃO DA IDENTIDADE VISUAL DE UMA EMPRESA**

Vinícius Willig Schlickmann<sup>1</sup>

Débora Cristina Engelmann<sup>2</sup>

### **Resumo**

O presente projeto tem como assunto a falta de padronização de componentes web de uma empresa do Vale dos sinos e os problemas que isto pode gerar. Após identificados todos estes problemas, optou-se pela construção de um framework React com componentes validados pela própria organização para solucionar tais equívocos, o que trouxe um ganho de performance em desenvolvimento, auxílio para a manutenibilidade e aumento da fidelidade visual das páginas web desta empresa. A construção do projeto foi guiada por uma versão adaptada da metodologia Scrum combinada com elementos do método Kanban, e seguiu princípios da engenharia de software.

**Palavras-chave:** Componentização. Padronização. Acessibilidade.

## **FRAMEWORK FOR STANDARDIZING AND UNIFYING A COMPANY'S VISUAL IDENTITY**

### **Abstract**

*This project has as its subject the lack of standardization of web components of a company from Vale dos Sinos and the problems that this can generate. After identifying all these problems, we opted for the construction of a React framework with components validated by the organization itself to resolve such mistakes, which brought a gain in development performance, aid for maintenance and increased visual fidelity of this company's web pages. The construction of the project was guided by an adapted version of Scrum methodology combined with elements of the Kanban method, and followed principles of software engineering.*

**Keywords:** *Componentization. Standardization. Accessibility.*

---

<sup>1</sup> Acadêmico do curso de Sistemas para Internet das Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil. E-mail: [viniciuswillig@gmail.com](mailto:viniciuswillig@gmail.com)

<sup>2</sup> Professora Orientadora do curso de Sistemas para Internet das Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil. E-mail: [deboraengelmann@faccat.br](mailto:deboraengelmann@faccat.br)

## 1 INTRODUÇÃO

Com o avanço tecnológico cada vez mais evidente, as empresas necessitam estar atualizadas e usufruindo de tudo que a tecnologia lhes possa oferecer. “Este mercado competitivo exige das empresas a utilização de informações para auxiliar nos processos decisórios e também na sua gestão organizacional” (LÖBLER; BOBSIN; VISENTINI, 2008). Com isso, o número de websites sob domínio empresarial é algo muito relevante.

A utilização dos mesmos vai desde simples controles internos ou organização de vagas na garagem até complexos sistemas para controle de ponto, férias e abonos. É extremamente valioso para as companhias poderem usufruir da tecnologia a seu favor, pois com ela, podem aliar redução de tempo para a realização das atividades e controle sobre as mesmas. Löbler, Bobsin e Visentini (2008) ainda destacam que a TI utilizada como um recurso estratégico das organizações pode transformar as informações que estão dispersas no mercado em novos conhecimentos.

Contudo, existem boas práticas na utilização dessas páginas, que são técnicas identificadas como as melhores em um contexto de tornar a experiência de uso mais satisfatória. Dentre elas, podemos citar o uso de padrões nos componentes web. O uso de padrões, pode trazer inúmeras vantagens para a empresa, dentre elas pode-se citar: facilitar a manutenibilidade das páginas, permitir que os usuários reconheçam visualmente à quem a página pertence, de acordo com cores, fontes e demais elementos contidos no manual da marca, e um dos pontos mais relevantes é referente a experiência do usuário ao utilizar tal página.

Essa experiência relacionada ao usuário é algo de suma importância e está em ascensão na sociedade principalmente em âmbito virtual. Conforto e Santarosa (2002) enaltecem a construção de uma sociedade que renuncia as lógicas de exclusão, aproximando-se da utopia de inserção para todos os atores sociais. Há inúmeros elementos que englobam esse contexto no desenvolvimento web, como por exemplo, acessibilidade e suas técnicas de contraste de cores, tamanho de fontes e o atributo “*alt*” nas imagens.

Com base nos pontos anteriormente citados, o presente projeto tem como objeto de estudo as páginas web internas de uma empresa do Vale dos Sinos que atua na fabricação de softwares. Essas páginas foram analisadas, com base em informações coletadas em conversas informais com usuários de áreas distintas que as utilizam em seu dia-a-dia, e perante validação com o setor de marketing, foi proposta a criação de um *framework* construído em React, para a solução dos problemas encontrados. Esse *framework* engloba

componentes validados pela empresa, disponibilizados para os desenvolvedores responsáveis pelas páginas internas os utilizarem.

O presente artigo descreve o processo de desenvolvimento deste *framework*, que permite realizar a instalação da biblioteca que contém os componentes, bem como a construção de uma página web que permite consulta-los para a utilização em um código fonte. Ilustra, também, os resultados obtidos após a implantação desta e a adoção como instrumento de trabalho. O resultado do trabalho foi originado a partir do seu objetivo principal, que é desenvolver uma aplicação que auxilie o desenvolvimento componentizado, visando sempre os benefícios que isso trará para a empresa e para os usuários dos seus sistemas.

Este artigo apresenta a seguinte ordem, respectivamente: na seção 2, um referencial teórico ilustra os assuntos abordados; A metodologia de desenvolvimento do *software* proposto é explicada na seção 3; Os resultados do estudo são apresentados na seção 4; E, por fim, as conclusões acerca do trabalho são expostas na seção 5.

## **2 REFERENCIAL TEÓRICO**

Nesta seção, os conceitos que envolvem este projeto de pesquisa, bem como as principais fontes e referências utilizadas na criação do *framework* são apresentados.

### **2.1 Tecnologia da Informação e sua utilização nas empresas**

As mudanças nos conceitos dos negócios e as inovações no campo da tecnologia da informação têm gerado uma reviravolta na forma de atuar das empresas, que devem se preparar e se ajustar a essa nova realidade emergente, sob a pena de perder terreno em termos de competitividade (TAPSCOTT; CASTON, 1995).

Em tempos onde o crescimento da influência da tecnologia é cada vez mais evidente, essa premissa se acentua. As empresas estão buscando se manter atualizadas, na busca por melhorias constantes em seus processos, utilizando esse cenário de consolidação tecnológica a seu favor. É muito comum que empresas de pequeno ou grande portes tenham páginas disponíveis na web para facilitar ou aprimorar ações internas. Tendo isso em vista, é interessante que o uso dessas páginas web sigam certos padrões para dinamizar a relação com os usuários (PRATES; OSPINA, 2004).

Esses padrões são essenciais para permitir a manutenção e promover reuso (TIBONI;

LISBOA; MOTA, 2009). Para Santos (2008) o uso de padrões facilita a troca de dados, a manipulação das informações e também a reutilização ao longo do tempo. Direcionando a análise para questões de acessibilidade, Leal Ferreira, Chauvel e Ferreira (2006) afirmam que acessibilidade é uma temática que não trata apenas de acesso por pessoas com deficiência.

Portanto, inúmeras outras situações e características do usuário devem ser consideradas durante o desenvolvimento de páginas web. A seguir, são listados três problemas que podem estar presentes em ambientes cuja padronização e componentização não estejam alinhadas junto ao time de desenvolvimento:

## **2.2 Dificuldade de manutenção**

A manutenção de software pode ser necessária mesmo em sistemas que acabaram de ser entregues ao usuário, porém a manutenção de softwares legados é a atividade que normalmente apresenta os maiores desafios (DE ESPINDOLA, 2004). Quando uma empresa possui muitas páginas a manutenção delas acaba se tornando um problema, que se acentua pelo fato de não ter sido utilizado nenhum tipo de componentização.

A estratégia de fazer uso de componentes facilita muito quando se diz respeito a manutenibilidade das páginas, quando todas essas páginas pertencem a uma mesma empresa e estão sob um mesmo guia visual. Essa questão de componentização é muito explorada na linguagem de programação React. Por permitir e facilitar esse uso, entende-se que a utilização da linguagem é vantajosa, pois permite que o desenvolvedor importe apenas os componentes que irá utilizar, deixando seu trabalho mais consistente e permitindo a criação de aplicações mais leves (FRIAS, 2019).

Junior (2019) complementa citando a maior característica da linguagem, afirmando que os padrões React utilizam a estrutura de componentes para granular o software, ou seja, dividi-lo em partes menores. O foco nesse caso é criar componentes independentes e reutilizáveis no decorrer da aplicação ou em diversas aplicações. Sendo assim, a componentização aliada com a linguagem pode diminuir significativamente os ruídos provenientes da manutenção às páginas internas da empresa.

## **2.3 Experiência do usuário prejudicada**

De acordo com o que afirma Souza (2005), quando o usuário acessa a uma página

web, procura por interatividade. Logo, não há apenas uma busca por informações, mas sim o anseio por interagir com elas. Toda organização tem seu público alvo, e os websites devem atender prioritariamente às necessidades desse segmento. Além disso, os usuários buscam padrões visuais que lhes ajudem, indicando de forma intuitiva onde se deve clicar, tornando a ação praticamente involuntária.

Segundo Kalbach (2009), um site pode ser atraente e bonito, mas ineficaz se as pessoas não conseguirem encontrar facilmente aquilo que procuram. Por isso é tão importante ajudar visualmente ao usuário, utilizando padrões que ele possa reconhecer facilmente, bem como cores e tonalidades que identifiquem ocasiões e status de momento. “A usabilidade ou a experiência do usuário são determinantes críticos para o sucesso de aplicações web, sobretudo porque interfaces mal projetadas aumentam a chance de erro por parte do usuário” (SOUZA, 2005).

#### **2.4 Falta de unificação na identidade visual**

A falta de parâmetros para o desenvolvimento com base na identidade visual resulta em projetos incompreensíveis, e os equívocos no uso de cores, tonalidades e formas resultam na sobrecarga informacional, dentre outros problemas, conforme ressalta Silva (2012). Os equívocos citados podem se somar quando a identidade visual deixa de ser utilizada em websites. Normalmente, organizações de diversos portes possuem um manual das suas marcas, elaborado por profissionais da área do design, que visa a harmonia entre o utilizador do produto final e o item a ser consumido. Silva (2012) também afirma que esses manuais estabelecem normas e critérios técnicos na produção da marca nos mais variados suportes que podem ser aplicados a ela, pretendendo atingir de forma positiva o consumidor, seja por meio de material físico ou virtual.

Teixeira (2012) corrobora dizendo que existem alguns itens básicos obrigatórios em um manual de identidade visual, que são: especificação do uso de cores, recomendações e proibições na divulgação da marca, criação de tipografia para o uso em todas as aplicações, posições para assinatura, entre outras. “A identidade de marca de uma empresa envolve sua relação com seus públicos, sejam eles internos ou externos à organização, abrangendo áreas diversas como a financeira, institucional, social, política, entre outras” (VÁSQUEZ, 2007). Por isso, a identidade visual de uma organização é de suma importância, e deve estar presente em todas as suas fontes consumíveis pelo usuário, inclusive nas suas páginas web.








## 2.5 Trabalhos relacionados

Embora haja certa resistência sobre a utilização de *frameworks* por parte de alguns desenvolvedores, a utilização dos mesmos pode ser algo muito útil, podendo aliar desempenho e praticidade. Os *frameworks* segundo Santoro, Da Silva Borges e Santos (1999), surgem como importantes quadros conceituais, que fornecem diretrizes para a pesquisa e também para o desenvolvimento. Com base em pesquisas preliminares ao desenvolvimento do presente projeto, foi feita uma análise em *frameworks* que se assemelham ao proposto. Dentre eles, podemos citar os seguintes:

- Pure: é um *framework* muito focado em CSS. Segundo o site oficial purecss.io (2014), ele se baseia em Normalize.css e fornece *layouts* e estilos para elementos HTML nativos, além de demais componentes mais comuns.
- Material UI: conforme apresentado no site oficial do *framework*, é uma “biblioteca de componentes React para um desenvolvimento ágil e fácil” (MATERIAL UI, 2020).
- Semantic UI: possui bases totalmente ligadas ao HTML, que procura utilizar a “sintaxe da linguagem natural como substantivos, ordem das palavras e pluralidade para relacionar os conceitos intuitivamente” (SEMANTIC UI, 2020).
- Materialize: foi criado e projetado pelo Google, e teve por objetivo “desenvolver um sistema de design que permita a experiência do usuário unificada em todos os seus produtos em qualquer plataforma” (MATERIALIZECSS, 2020).
- Foundation: é um *framework* baseado em HTML e CSS, trabalha com alguns temas prontos já para a escolha do usuário, e segundo o site, possui uma base semântica, é legível, flexível e totalmente personalizável (FOUNDATION, 2020).
- Bootstrap: provavelmente o mais famoso desta lista, o bootstrap se denomina “o kit de ferramentas de *front-end open source* mais popular do mundo” (BOOTSTRAP, 2020), que apresenta sistemas de grade responsivo, componentes pré-construídos e *plug-ins JavaScript*.

Para refinar o elemento comparativo, foi construída uma tabela, cujos atributos e características foram colocados lado a lado. A tabela 1, traz como atributos: a linguagem em que o *framework* está baseado; possuir ou não *plug-ins JavaScript*, que são necessários para a realização de animações e efeitos nos componentes; ser ou não modular, o que significa que, sendo modular, o usuário necessita instalar ou importar apenas o conteúdo que irá utilizar, sendo que o não modular necessita instalar toda biblioteca, mesmo que use apenas alguns componentes dela; a questão de instalação, se necessita ou não; e por fim, a facilidade de uso, que foi consultada em fóruns oficiais e comentários de usuários nas próprias páginas.

Tabela 1 - Comparativo entre frameworks

Framework	Linguagem Base	Plug-ins JS	Molular	Facilidade de uso	Necessita instalação
 PURE	HTML e CSS	Não	Sim	Muito fácil	Não
 MATERIAL UI	Java Script	Sim	Não	Fácil	Sim
 SEMANTIC UI	HTML e CSS	Sim	Sim	Muito Fácil	Sim
 MATERIALIZE	HTML e CSS	Sim	Não	Fácil	Sim
 FOUNDATION	HTML e CSS	Sim	Não	Muito fácil	Sim
 BOOTSTRAP	HTML e CSS	Sim	Não	Muito fácil	Sim
 CWI COMPONENTS	Java Script	Sim	Não	Muito fácil	Sim

Fonte: Elaborada pelo autor (2020).

Como pode ser observado, as características dos *frameworks* são bastante semelhantes. A grande maioria deles tem como linguagem base o HTML e o CSS e possuem plug-ins para JavaScript. A modularização não é uma prática muito adotada nesses casos, provavelmente pelo fato de todos serem leves, não havendo um problema relacionado ao tamanho dos mesmos, já que, como também pode-se ver na tabela, a maioria dos frameworks necessita ser instalado para ser utilizado. Visto as principais características, o que torna o projeto CWI Components único é o fato de conter elementos específicos de uma empresa, moldados e planejados exclusivamente para o uso dela.

### 3 METODOLOGIA

Esta seção detalha o embasamento literário da arquitetura, da metodologia e das tecnologias utilizadas para a realização do projeto.

Para a realização de um diagnóstico mais preciso, foram analisadas as páginas web de uma empresa do Vale dos Sinos, que atua na criação de software. O empreendimento possui em torno de cinquenta páginas internas para o controle de ativos, colaboradores, organização e administração. O que pôde ser constatado após a análise é que não há uma padronização geral na utilização de componentes web, tendo em vista que eles diferem em suas cores, elementos, fontes e inclusive na navegação.

Com isso, a metodologia foi definida e modelada para que pudesse direcionar o encontro de possíveis soluções e nortear todo desenvolvimento do projeto.

### 3.1 Metodologia de pesquisa

A metodologia aplicada para a realização deste trabalho foi uma versão adaptada da Metodologia Scrum, por meio da qual as tarefas para a conclusão do estudo foram estimadas e divididas em sprints de duas semanas. Conforme cita Sabbagh (2014), o Scrum pode permitir a redução dos riscos de insucesso, entregar valor mais rápido e lidar com possíveis e, muitas vezes inevitáveis, mudanças de escopo, transformando-as em vantagens competitivas.

A definição da metodologia foi resultante justamente da análise dessas vantagens que o Scrum proporciona, permitindo que entregas de valor fossem realizadas com boa frequência e ainda auxiliando no acompanhamento do desenvolvimento da ferramenta. Sabbagh (2014) ainda aponta que o Scrum é aplicado em projetos de características muito variadas, desde projetos críticos de milhões de dólares, até projetos internos simples e de baixa complexidade. Porém, a Metodologia é normalmente utilizada por equipes formadas por mais de um profissional. No caso do presente projeto, foi necessária uma adaptação para aproximar os conceitos teóricos de sua realidade prática.

Aliada a essa Metodologia, fez-se também o uso do Método de Kanban, pelo qual as tarefas foram agrupadas em um quadro de atividades no formato de fichas, separadas de acordo com o andamento do projeto como *backlog*, *doing* e *done*. No guia escrito por Boeg (2010), o autor define que o Kanban oferece uma abordagem menos prescritiva e tem se tornado uma extensão popular dos métodos ágeis tradicionais, como o XP e o Scrum.

Além da Metodologia e Método citados anteriormente, o projeto foi construído e guiado de acordo com pilares presentes na engenharia de software, que são: comunicação, planejamento, modelagem, construção e entrega. Em cada uma dessas sessões, técnicas são aplicadas ou atividades são realizadas, para que todas possam convergir nos resultados apresentados. Como Pressmann (2016) explana em seu livro *Engenharia de Software - 8ª edição*, a engenharia de software é uma tecnologia construída em camadas, e esse processo constitui a base para o controle do gerenciamento de projetos de software e estabelece um contexto, no qual são aplicadas técnicas, estabelecidos marcos, a qualidade é garantida e mudanças são geridas de forma apropriada.

### 3.2 Comunicação



O pilar de comunicação é aquele no qual iniciam-se os levantamentos necessários para a construção do projeto. É nessa fase que é feita a coleta de requisitos e as entrevistas com os usuários. Para a construção do presente projeto, foram realizadas conversas informais com 12 colaboradores da empresa pesquisada. Dessas 12 pessoas, sete eram desenvolvedores que trabalhavam diretamente com as páginas internas da organização, três eram designers e dois colaboradores eram da área de gestão.

A partir da análise do conteúdo dessas conversas, foi possível obter um apanhado geral das principais angústias laborais dos colaboradores com os quais houveram as conversas informais, possibilitando a idealização de uma ferramenta realmente útil e eficaz. Para possibilitar esse diagnóstico, foi realizada a criação de duas personas. Alan Cooper (1999), foi o criador desse conceito, e afirma que elas servem para ajudar os profissionais a conhecer melhor e desenvolver empatia pelos usuários.

O resultado das criação das personas está evidenciada na figura a seguir.

**Figura 1- Análise das personas**



**Fonte: Elaborada pelo autor (2020).**

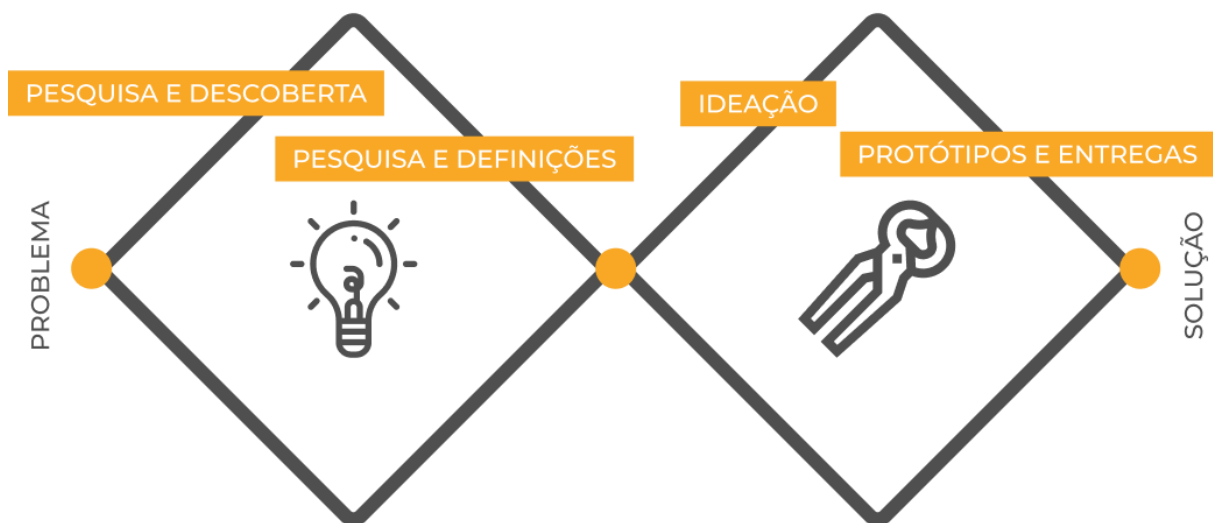
A partir da análise das personas do projeto, evidenciou-se a necessidade de uma padronização dos componentes utilizados para a criação das páginas; técnica essa já adotada pela equipe de design, que agregou bastante valor às tarefas diárias e a sua utilização em

seus protótipos. Além disso, questões fundamentais na web, como é o caso da acessibilidade, também podem ser aprimoradas com essa prática.

### 3.3 Planejamento

Com os requisitos iniciais já analisados, chega a fase de compreender de forma mais abrangente o escopo do projeto, considerando sua totalidade, para que se possa ter uma breve noção de roteiro implementado. Para nortear essa etapa, fez-se o uso de uma técnica muito comum no meio do design de interação, que é o *Double-Diamond*. “O modelo *Double-Diamond* de design orienta a investigação simultânea de diversas soluções antes de convergir para o protótipo final, o que favorece o processo criativo” (MARTINEZ, 2015).

Figura 2 - Double-Diamond



Fonte: Elaborada pelo autor (2020).

Essa técnica consiste em vislumbrar o problema, e a partir dele traçar alguns passos para chegar até as suas possíveis soluções. Conforme a figura 2 ilustra, em um primeiro momento é identificada a demanda e, após esse diagnóstico, parte-se para uma etapa que engloba pesquisas e descobertas de todos os fatores e pessoas às quais o elemento que precisa ser modificado está relacionado.

Portanto, há uma primeira abertura do diamante, em que estão contidas todas essas informações. Foi exatamente nessa etapa em que fez-se a criação das personas citadas na subseção 3.2. Apoiado nessas descobertas iniciais, o passo subsequente é analisar as

demandas para a geração de definições, fazendo com que o primeiro diamante se feche com uma possível solução já definida. No caso do presente projeto, neste ponto ficou definido que se faria o uso de componentes para padronização.

Parte-se então para o segundo diamante, fazendo sua abertura com as ideias envolvendo essa solução. Nessa etapa, a partir de um rápido estudo, ficou definido que a solução seria disponibilizada em forma de uma página web, liberada para todos os colaboradores da empresa pesquisada, a partir da sua autenticação com e-mail organizacional. Portanto, qualquer colaborador da empresa terá acesso à solução, mesmo que não a utilize no projeto em que está inserido atualmente.

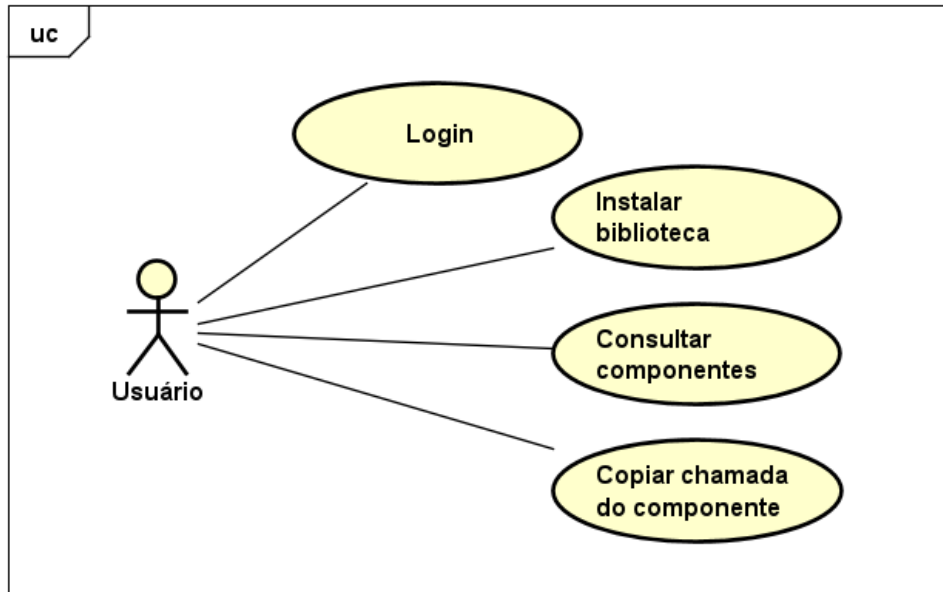
E, por fim, o fechamento do segundo diamante é concretizado com as entregas, para que a solução propriamente dita esteja completamente analisada. Chegou-se à conclusão, nesse ponto, de que a ferramenta ideal seria um framework, em que os componentes validados com a empresa estariam contidos e aptos para uso dos desenvolvedores.

### **3.4 Modelagem**

O pilar de modelagem é o que começa a tornar a aplicação mais palpável. Nele, são realizadas as prototipagens de sistema e também a construção de diagramas, que norteiam o desenvolvimento do início ao fim. Os diagramas são muito úteis em cenários de desenvolvimento de software. Segundo Garcia (2016), eles descrevem abstrações de software ou componentes com especificações de interface. Na construção do presente projeto, três diagramas foram elaborados: o diagrama de caso de uso, o de domínio e o de atividades.

A figura 3 representa o principal caso de uso da ferramenta, exibindo cada interação que o usuário poderá exercer. As principais interações são a instalação da biblioteca de componentes e, após isso, a cópia da chamada para o uso de determinado componente na aplicação em que o usuário estiver trabalhando.

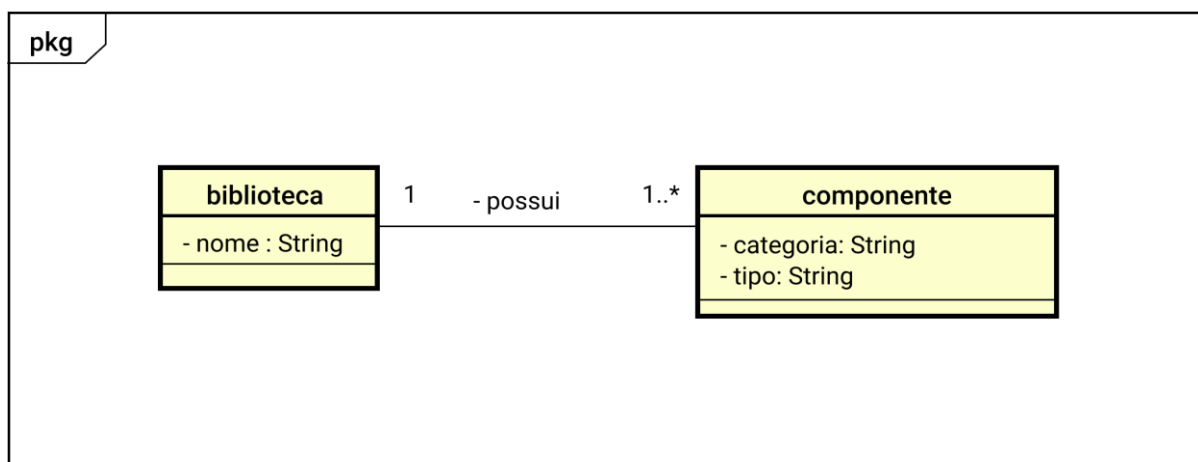
Figura 3 - Diagrama de Caso de Uso



Fonte: Elaborada pelo autor (2020).

O seguinte diagrama, representado na figura 4, diz respeito ao diagrama de domínio. Ele contém a interação entre a biblioteca e os componentes que ela contém. Nesse caso, a biblioteca cujo atributo único é o nome em formato de String, possui ao menos um e no máximo  $n$  componentes. Já os componentes possuem categoria e tipo como atributos, ambos em formato de String, além de comportamento no formato booleano. Para os componentes, a cardinalidade da ligação com a biblioteca é 1, o que significa que cada componente pertence a apenas uma única biblioteca.

Figura 4 – Diagrama de Domínio

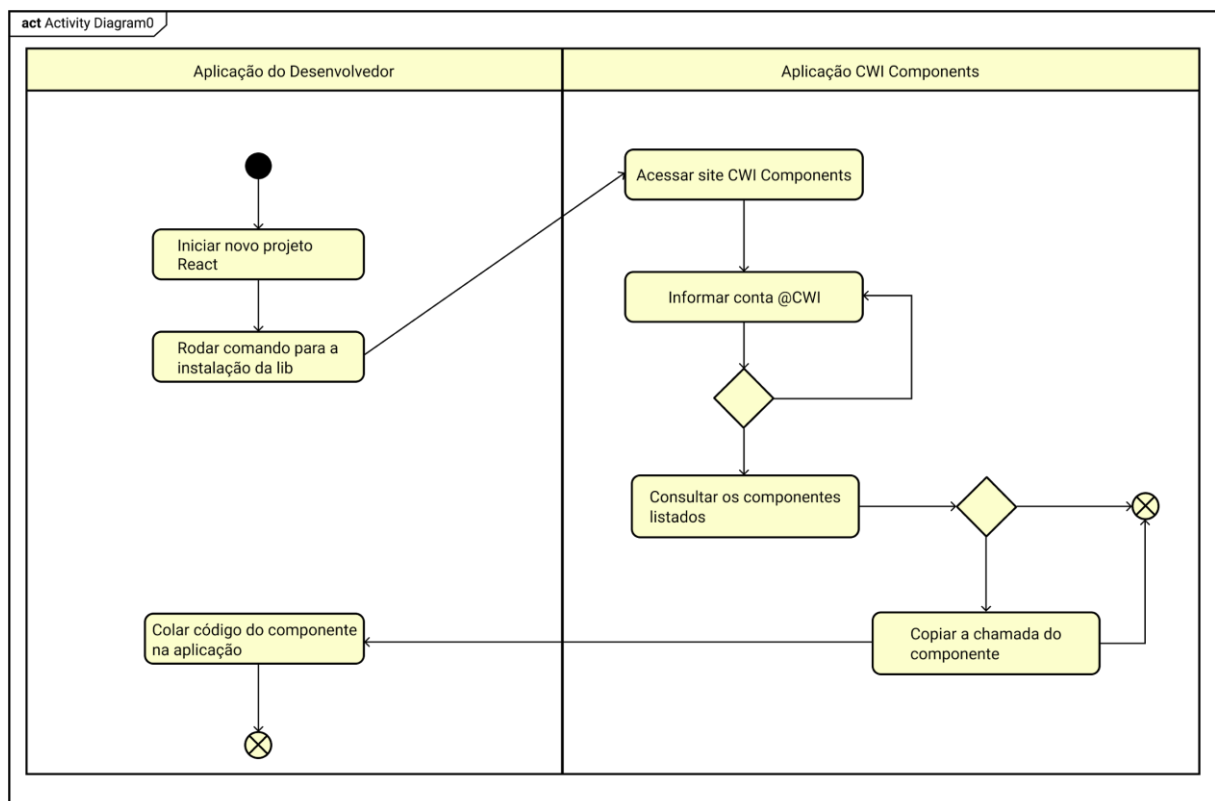


Fonte: Elaborada pelo autor (2020).

Por fim, a figura 5 apresenta o diagrama de atividades. Ele basicamente descreve um caminho feliz na interação usuário-ferramenta. A estrutura do diagrama foi dividida entre ações que o usuário executará a partir da sua área de trabalho e outras que serão executadas na própria ferramenta. Os primeiros passos do usuário são respectivamente iniciar um novo projeto e instalar a biblioteca neste projeto.

A partir disso, as suas ações ocorrem na própria ferramenta. Após acessar a página de apoio, o usuário informa uma conta válida para login, consulta os componentes ali listados e por fim copia o código da chamada de determinado componente. Nesse instante, o usuário volta para a sua área de trabalho e cola a chamada copiada anteriormente. Após essas etapas, o componente já estará em uso na aplicação.

**Figura 5 - Diagrama de Atividades**

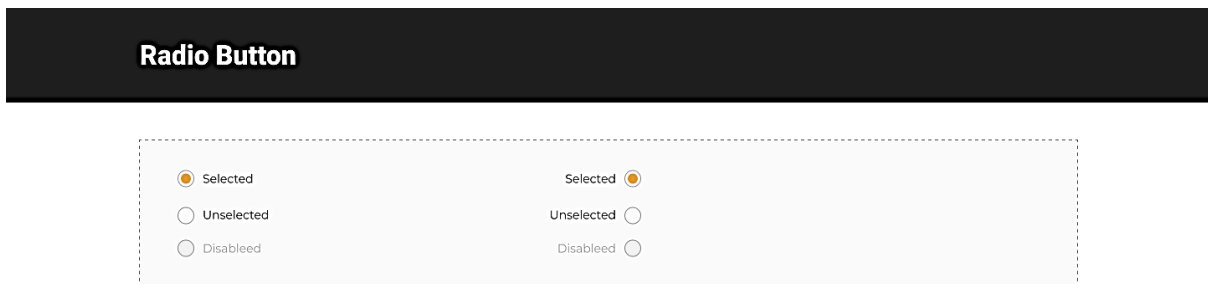


**Fonte: Elaborada pelo autor (2020).**

Findadas as questões referentes a diagramas, o último tópico do pilar é referente a prototipagem. Essa técnica, segundo Soares e Resende (2007) tem por objetivo auxiliar na especificação e validação de requisitos relevantes ou problemas de implementação, permitindo elaborar e testar interfaces com os usuários de maneira visual e interativa. A

validação citada anteriormente foi executada, levando sempre em consideração o usuário. Portanto, foram conferidas questões relevantes como contraste de cores, tamanho de fontes, cores e tonalidades aptas para uso, sempre de acordo com o manual da marca da organização. A figura 6 a seguir mostra um fragmento de prototipagem de um dos componentes contidos na biblioteca.

**Figura 6 - Fragmento de prototipagem**



**Fonte: Elaborada pelo autor (2020).**

### 3.5 Construção

A construção do projeto nesse contexto condiz com requisitos tecnológicos de desenvolvimento e de aplicação. Através de um estudo realizado na empresa onde o presente projeto foi aplicado, ficou definida como linguagem oficial da ferramenta o React. Essa linguagem vem sendo utilizada na elaboração das páginas mais recentes da organização bem como na remodelagem de páginas legadas, e também em dois módulos do projeto interno de capacitação de novos colaboradores.

Logo, as frentes utilizadas para a aplicação do projeto foram os desenvolvedores alocados na construção de páginas internas e também os candidatos inscritos no projeto de capacitação.

### 3.6 Entrega

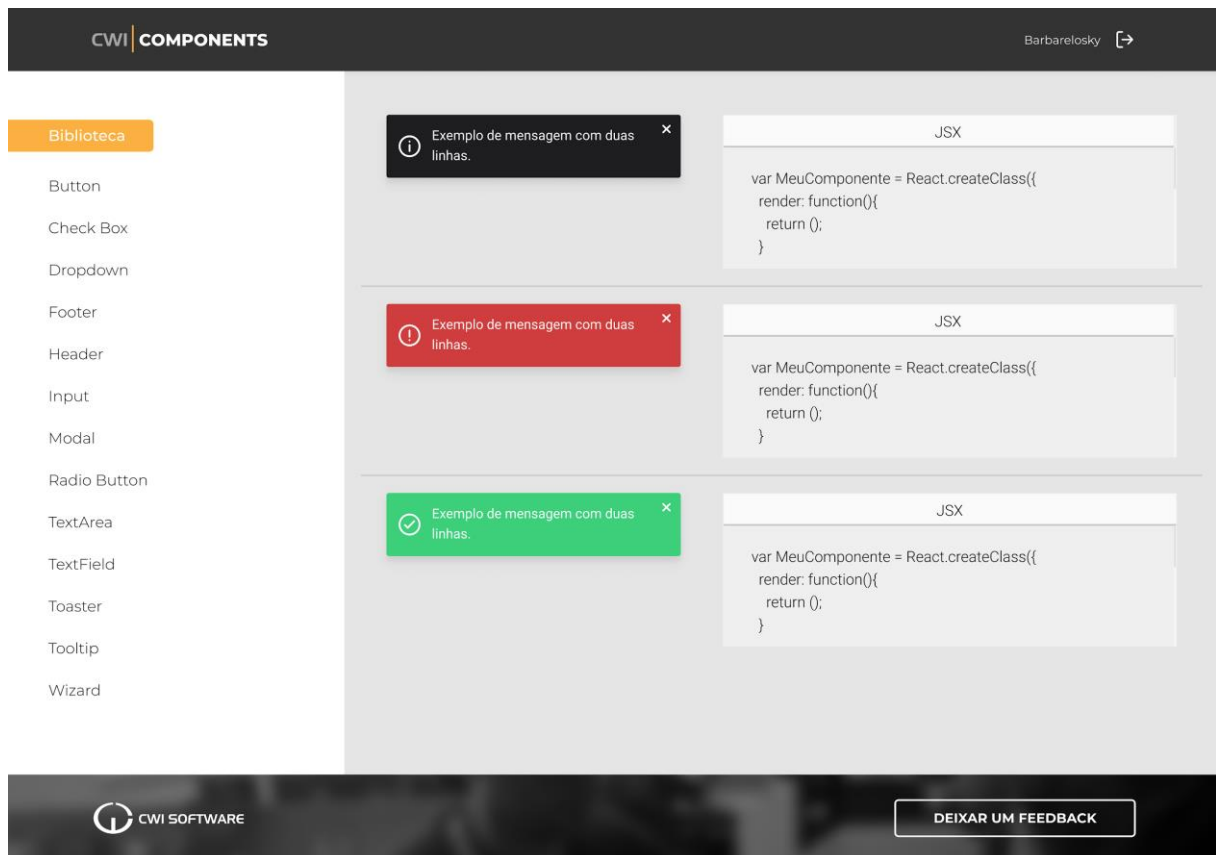
Após todas as análises e validações concluídas, o desenvolvimento e o teste do projeto são as partes que completam o pilar de entrega, que se formaliza na figura de um software, o qual pode ser descrito como um framework para padronização e unificação da identidade visual. Além de um framework, a solução também contempla uma página web para consulta dos componentes nele contidos. A página respeita as orientações contidas no

manual de uso da marca da empresa, implantando adequadamente as questões como cores, ícones e fontes, bem como a utilização da própria marca da empresa seguindo tamanhos e proporções pré-definidos.

Além do manual da marca, o quesito acessibilidade também guiou o desenvolvimento da ferramenta. Todas as combinações de cores, seja entre *background* e texto, entre botão e *label*, ou de forma geral entre primeiro e segundo planos respeitam o valor referência de contraste entre as cores relacionadas. Como Rocha e Duarte (2012) explicam, as coloração de primeiro e segundo planos das páginas precisam ser suficientemente contrastantes para que as pessoas com baixa visão, com daltonismo ou que utilizam monitores de vídeo monocromático possam visualizá-las e distingui-las.

Outro fator levado em consideração foi o tamanho das fontes, para o qual foi seguida uma recomendação de Vechiato e Vidotti (2013) que sugerem o uso de fontes cujo tamanho para o corpo seja de 12 ou 14. Conforme pode ser observado na imagem a seguir, o menu localizado na lateral esquerda da tela está em ordem alfabética, facilitando a busca realizada pelo usuário por categorias.

**Figura 7 - Prototipagem da página**



**Fonte: Elaborada pelo autor (2020).**

Outra característica presente na construção da página foi o uso do atributo “*alt*” nas imagens, que é um recurso de interface utilizado a fim de que deficientes visuais também possam acessar o conteúdo a partir de leitores de tela, conforme ressalta Ferreira (2007).

Ainda falando sobre a página, a figura anterior ilustra a apresentação dos componentes. Todos foram agrupados em categorias, e assim que o usuário escolhe uma delas, os componentes nela contidos são exibidos, com todas as variações possíveis para que possa ser conduzida uma avaliação rápida sobre os componentes. Entre os itens do menu, apenas o primeiro não se trata de um componente, mas sim do código referente à instalação da biblioteca. A ação prevista a ser executada por parte do usuário é a cópia do link responsável pela instalação; portanto, ao lado da caixa de texto que contém o link, há um botão que prevê uma ação de copiar o conteúdo presente ao lado, também com o intuito de facilitar o uso.

Para finalizar a apresentação do *dashboard*, resta listar o *code-box* contido ao lado de cada exemplo ilustrado. Esse elemento contém o fragmento codificado que o usuário fará uso em seu código fonte. Já a chamada contém a importação do componente, bem como o trecho responsável por projetar em tela o componente, seus atributos e eventuais comportamentos.

### **3.7 Tecnologias utilizadas**

Para o desenvolvimento do presente projeto, foram utilizadas ferramentas que auxiliaram na organização bem como nas atividades de construção. Inicialmente, as atividades necessárias tinham grau teórico, como todas as pesquisas e definições elaboradas para que o projeto pudesse ser embasado. Nessa fase, fez-se uso do Trello, que com seus quadros, listas e cartões permitem a organização e a priorização de projetos de uma maneira divertida e flexível (TRELLO, 2020). Foram organizados dois quadros de atividades; O primeiro deles foi preenchido com tarefas complexas, relacionadas a um objetivo para conclusão do projeto.

O segundo quadro lista as tarefas detalhadas necessárias para a construção destes objetivos, onde a cada sprint (de duas semanas cada) eram alocadas algumas destas, conforme prioridade, para sua conclusão. Relacionada com a modelagem do projeto, utilizou-se duas ferramentas, o Figma e o Astah. O Figma, que é um editor gráfico de prototipagem de projetos de design (FIGMA, 2020) foi onde todos os protótipos do presente projeto foram elaborados, tanto dos componentes que compõem a biblioteca, como da página web onde os mesmos são exibidos para consulta. Já o Astah, que é uma ferramenta de modelagem, que permite a



construção de diagramas, mapas mentais, e demais artefatos (ASTAH, 2020) foi onde os três diagramas contidos no projeto foram construídos.

Ainda em caráter teórico, uma pesquisa juntamente com responsáveis pela tecnologia da empresa, gerou insumos que permitiram definir o React, como a linguagem de programação que seria utilizada para a construção do projeto. React vem de encontro com o tema proposto ao projeto, visto que a linguagem faz uso de componentes encapsulados que gerenciam seu próprio estado (REACT, 2020). Para auxílio à linguagem principal, na estilização dos componentes e demais elementos da página e da biblioteca, foi escolhido o CSS, pois segundo a W3C (2020) se trata de um mecanismo simples para adicionar estilo.

Após as fases teóricas, com as definições findadas, partiu-se para uma fase voltada para a prática da construção da ferramenta. Para a codificação do projeto, a interface escolhida foi o Visual Studio Code, que é um editor de código-fonte leve, porém poderoso, que vem com suporte integrado de JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens (VISUAL STUDIO, 2020).

Por fim, utilizou-se o Github, como repositório do projeto, para fins de backup e também para poder disponibilizar a biblioteca para futura instalação. O Github, é uma plataforma *OpenSource* de hospedagem de código-fonte e arquivos com controle de versões usando o Git (GITHUB, 2020).

## 4 RESULTADOS

Após a execução plena do projeto, o mesmo foi integrado junto às equipes de desenvolvimento atuantes em projetos internos da empresa, para que pudessem usar a ferramenta e testar se a contribuição da mesma era realmente efetiva. Neste instante, se viu necessária a utilização de algum artefato que auxiliasse e permitisse um demonstrativo de efetividade. Um fator então corroborou para uma comparação direta, já que o primeiro sistema exposto ao uso do *framework* em seu desenvolvimento, foi a modernização de um projeto legado, que nessa ocasião foi todo remodelado, utilizando os componentes contidos na biblioteca.

Para realizar a medição do desempenho obtido através do *framework*, o uso de métricas e *OKRs* foi adotado como procedimento de definição. Métricas, segundo a definição do SEI (*Software Engineering Institute*) é um conjunto de medidas de um processo de software, cujo processo pode ser visto como um objeto abstrato que evolui de uma instrução inicial para o sistema de software finalizado, incluindo o código-fonte e demais formas de documentação

produzidas durante o desenvolvimento (MILLS, 1988). Mills (1988) também destaca que o objetivo das métricas de software é a identificação e a medição dos principais parâmetros que afetam o desenvolvimento de software. O uso desse tipo de medição vem crescendo, e em projetos de *software* é tido como bastante comum. Tanto que, na bibliografia encontram-se inúmeros exemplos de métricas que podem ser utilizadas em projetos ou processos tecnológicos.

*OKR* segundo Valerim (2020), nada mais é do que uma ferramenta para orientar e executar estratégias dentro de uma empresa, onde se define objetivos e resultados-chave para alcançar esses objetivos de acordo com o tamanho do desafio ou da própria empresa. Portanto, *OKRs* são grandes objetivos a serem concluídos, cujas métricas atribuídas a cada um deles, dirá se a meta foi alcançada e de que forma isso aconteceu.

O embasamento foi obtido através de estudos sobre tais métricas, onde a maior fonte de consulta foi a tese apresentada por Paulo Roberto Miranda Meirelles ao Instituto de Matemática e Estatística da Universidade de São Paulo. Para Meirelles (2013) monitorar a qualidade do *software* é fundamental, e dentre as inúmeras características que fazem-no ser considerado bom, muitas delas são referentes ao código-fonte. Por isso, julga tão importante medir a qualidade do código além de outros fatores relacionados ao desenvolvimento e ao desenvolvedor.

#### **4.1 Diminuir o tempo de desenvolvimento**

O primeiro *OKR* definido foi diminuir o tempo de desenvolvimento. Para verificar o cumprimento deste *OKR*, as seguintes métricas foram construídas:

- Tempo gasto na execução do código-fonte referente ao *front-end* da aplicação.
- Tempo estimado para a construção do código-fonte referente ao *front-end* da aplicação com o uso do *framework*.

Para a construção destes *OKRs*, foram elaboradas algumas perguntas destinadas aos potenciais usuários do sistema. Estas perguntas foram feitas através de um formulário, encaminhado aos desenvolvedores responsáveis pelos projetos internos da empresa, e respondidas através de estimativas, tendo em vista que o projeto onde os componentes propostos foram utilizados, foi uma remodelagem visual de um projeto legado. Essa construção foi iniciada, porém não foi finalizada, devido a demandas relacionadas a pandemia, onde situações precisaram ser adaptadas, portanto a utilização da ferramenta iniciou mas teve que ser pausada por breve período de tempo. Além dos desenvolvedores, os participantes do projeto final do programa de qualificação da empresa também responderam a pesquisa, e os resultados

estão dispostos separadamente.

As perguntas feitas foram as seguintes: “De acordo com sua última experiência com a construção de uma página web interna para a empresa, qual foi o tempo (em horas) gasto para a construção do front-end da mesma? E, “Agora, imagine que para esta mesma página, você pudesse utilizar um framework com os componentes previamente validados pela empresa, quantas horas estimadas você levaria para a construção do mesmo? (Para esta pergunta, foi disponibilizado o link para consulta da ferramenta).”

Os resultados coletados para este objetivo são apresentados na tabela a seguir.

**Tabela 2 - Estimativa de horas gastas em desenvolvimento**

Público Alvo	Horas gastas na última aplicação	Estimativa de horas gastas na mesma aplicação, porém com uso do framework	Redução de horas em %
Desenvolvedores trabalhando em equipes internas	86 horas	61 horas	29,06%
Participantes do projeto final do programa de qualificação	119 horas	94 horas	21,01%
Média	102,5 horas	77,5 horas	24,39%

**Fonte: Elaborada pelo autor (2020).**

Como ilustra a tabela 2, a redução de tempo baseada nas estimativas dos desenvolvedores chega a níveis médios próximos a 25%. Conclui-se então que este objetivo foi atingido com êxito.

#### **4.2 Reduzir tempo de desenvolvimento e teste na manutenção das páginas**

A manutenibilidade das páginas foi um dos grandes problemas encontrados no início do projeto. A quantidade de páginas existentes na empresa corrobora para que esse ponto se agrave ainda mais.

Este *OKR* é bem específico referente a manutenções das páginas que forem construídas com o auxílio do framework. Porém, até o presente momento, nenhum projeto foi entregue por completo fazendo uso da ferramenta, conseqüentemente, ainda não houve um momento onde manutenções fossem necessárias.

Contudo, a pesquisa com os desenvolvedores também abordou esse tema, levando em consideração a experiência de cada um e de acordo com estimativas para que as respostas pudessem ser coletadas. A pergunta feita para os mesmos foi a seguinte: “Se todas as páginas web da empresa fossem construídas com o auxílio do *framework*, e por ventura houvesse a necessidade de manutenção de algum componente, o uso da ferramenta facilitaria ou dificultaria esse processo?”

Para esta pergunta foram coletadas respostas que comprovam a eficácia da ferramenta, e dentre as respostas podemos destacar as seguintes: “Acredito que essa biblioteca facilitaria muito o desenvolvimento e a manutenção front-end dentro da empresa, porque o tempo de lidar com a estilização dos componentes seria praticamente eliminado. Além disso, sempre que necessário, novos componentes poderiam ser criados e adicionados, podendo serem reaproveitados facilmente”. Outra resposta que podemos citar foi a seguinte: “Facilitaria muito, uma vez que não seria preciso acessar individualmente cada projeto e aplicar as manutenções, bastaria efetuar uma atualização do pacote dos componentes. Certamente, tende a otimizar muito a produção e padronização”. Todas as perguntas e as respostas das mesmas encontram-se nos apêndices do presente projeto.

Por mais que esta métrica não tenha sido coletada após trabalhos práticos com a ferramenta, pode-se ver que de acordo com a experiência e avaliação dos desenvolvedores, O *framework* irá auxiliar a todos na manutenção das páginas, diminuindo o tempo de desenvolvimento.

### **4.3 Diminuir tamanho do código-fonte**

Para prosseguir com o processo de medição dos resultados, as seguintes métricas relacionadas ao tamanho do código-fonte presentes na tese de Meirelles (2013) foram consideradas coesas para avaliação do presente projeto:

- LOC (*Lines of Code* – Número de linhas de código): é a métrica mais simples relacionada ao tamanho do código. E para realizar essa medição de forma mais precisa, são eliminados os espaços em branco e também os comentários.
- AMLOC (*Average Method LOC* – Média de número de linhas de código por método): essa média indica de forma simples se o código está bem distribuído entre os métodos. Quanto maior e mais pesado forem os métodos, pior para o sistema.

Contudo, como o desenvolvimento do projeto que fazia uso da ferramenta foi paralizado momentaneamente, essas métricas não puderam ser coletadas, porém seguirão em

acompanhamento quando o desenvolvimento for retomado pela equipe.

## 5 CONCLUSÕES

Após a listagem dos problemas encontrados e análise do que isso representa, viu-se uma grande necessidade de propor uma solução para as mesmas. Essa solução então foi prontamente elaborada e validada com quem faria uso dela, bem como com a empresa em questão, onde o sistema foi testado e implantado como ferramenta de trabalho.

Após o período de teste e onde a ferramenta foi exposta para os desenvolvedores, coletaram-se os resultados provenientes, que trouxeram informações conclusivas quanto ao uso da mesma. Estimou-se que o tempo de desenvolvimento com o uso do *framework*, tende a ser em torno de 25% menor. Portanto conclui-se que o ponto referente ao auxílio aos desenvolvedores foi extremamente satisfatório, reduzindo de forma significativa o tempo gasto na construção do *front-end* das páginas.

Do ponto de vista sobre a manutenção das páginas, também obteve-se um resultado positivo, que não pode ser mensurado através de números, porém as conclusões retiradas a partir dos depoimentos dos desenvolvedores sobre o ganho que a ferramenta proporcionará a este item, é extremamente satisfatória. O benefício obtido ao se alterar um componente apenas na biblioteca, e isso refletir em todas as páginas que a utilizam, é extremamente valioso, tanto pela ótica da redução expressiva de tempo que isso retrata, como pela coesão visual que tende a ser respeitada.

Este último ponto citado, que refere-se a identidade visual da empresa, é onde o *framework* consegue mostrar-se mais participativo, visto que muitas páginas sob domínio da companhia, necessitam desta reestruturação visual, e com o uso da ferramenta isso se dará de forma natural. Aos olhos dos usuários, é o ponto de maior ganho, proporcionando a todos uma melhor experiência, guiada por elementos padrões, com comportamentos específicos e facilmente encontrados no universo web da instituição.

Portanto, com os apontamentos citados, conclui-se de forma satisfatória a experiência de desenvolvimento da solução proposta e dos resultados obtidos, com ganhos expressivos, conforme tratamento dado aos problemas encontrados.

O framework seguirá em utilização na organização, podendo ser expandido, de forma a contemplar mais equipes de desenvolvimento, e estará sendo ativamente atualizado, com novos componentes e comportamentos de acordo com novas demandas.

## REFERÊNCIAS

ASTAH, 2020. **Astah**. Disponível em: <https://astah.net/>. Acesso em: 29 de set de 2020.

BOEG, Jesper. **Kanban em 10 passos**. Tradução de Leonardo Campos, Marcelo Costa, Lúcio Camilo, Rafael Buzon, Paulo Rebelo, Eric Fer, Ivo La Puma, Leonardo Galvão, Thiago Vespa, Manoel Pimentel e Daniel Wildt. C4Media, 2010.

BOOTSTRAP, 2020. **Bootstrap**. Disponível em: <https://getbootstrap.com/>. Acesso em: 26 de set de 2020.

CONFORTO, Débora; SANTAROSA, Lucila. Acessibilidade à web: internetpara todos. **Informática na educação: teoria & prática**, v. 5, n. 2, 2002.

COOPER, A. **The inmates are running the asylum**. [s.l.]: Sams, 1999.

DE ESPINDOLA, Rodrigo Santos; MAJDENBAUM, Azriel; AUDY, Jorge Luis Nicolas. **Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software**. In: WER. 2004. p. 226-238.

FERREIRA, Simone Bacellar Leal. **E-acessibilidade: tornando visível o invisível**. Revista Morpheus-Estudos Interdisciplinares em Memória Social, v. 6, n. 10, 2007.

FIGMA, 2020. **Figma**. Disponível em: <https://figma.com/>. Acesso em: 29 de set de 2020.

FOUNDATION, 2020. **Foundation**. Disponível em: <https://get.foundation/index.html>. Acesso em: 26 de set de 2020.

FRIAS, Thiago. **React vs Vue vs Angular: qual escolher**. 2019. Disponível em: <https://blog.geekhunter.com.br/react-vs-vue-vs-angular-qual-escolher/>. Acesso em: 31 ago. 2020.

GARCIA, Rogério Eduardo. **Engenharia de Software I**. Tópicos, v. 19, p. 04, 2016.

GITHUB, 2020. **GitHub. Criado para desenvolvedores.** Disponível em: <https://github.com/>. Acesso em: 26 de set de 2020.

JUNIOR, Anderson Hilario; FARAH, Paulo Roberto. **Avaliação de Code Smells e Padrões de Projeto em Aplicações com Angular e React.** In: Anais da III Escola Regional de Engenharia de Software. SBC, 2019. p. 81-88.

KALBACH, James. **Design de Navegação Web.** Otimizando a experiência do usuário: 1. Ed. Porto Alegre: Bookman Editora, 2009.

LEAL FERREIRA, S. B.; CHAUVEL, M. A.; FERREIRA, MGAL. **e-Acessibilidade: Tornando Visível O Invisível.** Morpheus-Revista Eletrônica em Ciências Humanas, 2007.

LÖBLER, Mauri Leodir; BOBSIN, Debora; VISENTINI, Monize Sâmara. Alinhamento entre o plano de negócio e o plano de tecnologia de informação das empresas: análise comparativa através dos níveis de maturidade e fatores críticos de sucesso. **JISTEM-Journal of Information Systems and Technology Management**, v. 5, n. 1, p. 37-60, 2008.

MARTINEZ, Maria L. **Ensinando design de interação no curso de Editoração.** In: Proceedings of the VI HCI Education Workshop-XIV Brazilian Symposium on Human Factors in Computing Systems (WEIHC 2015). 2015.

MATERIAL UI, 2020. **Material-UI.** Disponível em: <https://material-ui.com/pt/>. Acesso em: 26 de set de 2020.

MATERIALIZECSS, 2020. **Materialize.** Disponível em: <https://materializecss.com/>. Acesso em: 26 de set de 2020.

MEIRELLES, Paulo Roberto Miranda. **Monitoramento de métricas de código-fonte em projetos de software livre.** 2013. Tese de Doutorado. Universidade de São Paulo.

MILLS, 1988. Everaldo E. Mills. **Software Metrics.** Relatório Técnico, Software Engineering Institute SEI. Disponível em: <https://resources.sei.cmu.edu/library/author.cfm?authorid=4241/>. Acesso em 02 de out de 2020.

PRATES, Gláucia Aparecida; OSPINA, Marco Túlio. Tecnologia da informação em pequenas empresas: fatores de êxito, restrições e benefícios. **Revista de administração contemporânea**, v. 8, n. 2, p. 9-26, 2004.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software-8ª Edição**. McGraw Hill Brasil, 2016.

PURECSS, 2014. **Pure.css**. Disponível em: <https://purecss.io/>. Acesso em: 26 de set de 2020.

React, 2020. **React**. Disponível em: <https://pt-br.reactjs.org/>. Acesso em: 29 de set de 2020.

ROCHA, Janicy Aparecida Pereira; DUARTE, Adriana Bogliolo Sirihal. **Diretrizes de acessibilidade web: um estudo comparativo entre as WCAG 2.0 e o e-MAG 3.0**. *Inclusão Social*, v. 5, n. 2, 2012.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. Editora Casa do Código, 2014.

SANTORO, Flávia Maria; DA SILVA BORGES, Marcos R.; SANTOS, Neide. Um framework para estudo de ambientes de suporte à aprendizagem cooperativa. **Revista Brasileira de Informática na Educação**, v. 4, n. 1, p. 51-68, 1999.

SANTOS, Ernani Marques dos. **Desenvolvimento e implementação de padrões de interoperabilidade em governo eletrônico no Brasil**. 2008. Tese de Doutorado. Universidade de São Paulo.

SEMANTIC UI, 2020. **Semantic UI**. Disponível em: <https://semantic-ui.com/>. Acesso em: 26 de set de 2020.

SILVA, João Carlos Riccó Plácido. **Diretrizes para análise e desenvolvimento de identidade visual: contribuições para o design ergonômico**. 2012.

SOUZA, Osnete Ribeiro de. **Processos de apoio ao desenvolvimento de aplicações web**.



2005. Tese de Doutorado. Universidade de São Paulo.

SOARES, Bruno C.; RESENDE, Rodolfo SF. **Requisitos para utilização de prototipagem evolutiva nos processos de desenvolvimento de software para Web**. Universidade Federal de Minas Gerais (UFMG), Belo Horizonte–MG, 2007.

TAPSCOTT, Dom; CASTON, Art. **Mudança de Paradigma**: 1. Ed. São Paulo: Makron Books, 1995.

TEIXEIRA, Felipe Colvara; SILVA, RDO; BONA, Rafael José. **O processo de desenvolvimento de uma identidade visual**. In: CONGRESSO BRASILEIRO DE CIÊNCIAS DA COMUNICAÇÃO DA REGIÃO SUL. 2012.

TIBONI, Antonio Carlos; DA SILVA LISBOA, Flávio Gomes; MOTA, Luciana Campos. **Uma plataforma livre para padronização do desenvolvimento de sistemas no Governo Federal**. COLIBRI-Colóquio de Informática-Brasil-INRIA, 2009, 2009.

TRELLO, 2020. **Trello**. Disponível em: <https://trello.com/>. Acesso em: 29 de set de 2020.

VALERIM, Gabriel Schossler. **Proposta de implantação do método OKR em uma empresa de pequeno porte do setor alimentício**. 2020.

VÁSQUEZ, Ruth Peralta. **Identidade de marca**, gestão e comunicação. *Organicom*, v. 4, n. 7, p. 198-211, 2007.

VECHIATO, Fernando Luiz; VIDOTTI, Silvana Aparecida Borsetti Gregorio. **Recomendações de usabilidade e de acessibilidade em projetos de ambientes informacionais digitais para idosos**. 2013.

VISUAL STUDIO, 2020. **Código Visual Studio**. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 29 de set de 2020.

W3C, 2020. **O que é CSS?** Disponível em: <https://www.w3.org/Style/CSS/Overview.en.html>. Acesso em: 29 de set de 2020.

## APÊNDICE A

### Perguntas feitas aos desenvolvedores

Pergunta 1:

Pense na sua experiência com a construção de uma página web interna para a CWI. Qual foi o tempo (em horas) que foi gasto para a construção do front-end desta aplicação?

Pergunta 2:

Agora, imagine que para esta mesma página, você pudesse utilizar o *framerowk* com os componentes previamente validados pela CWI, quantas horas estimadas você levaria para a construção do mesmo? (Link para consulta do framework:

<https://5f76193ba4ea866c6ad61e7a--cwi-components.netlify.app/>)

Pergunta 3:

Se todas as páginas web da empresa fossem construídas com o auxílio do framework, e por ventura houvesse a necessidade de manutenção de algum componente, o uso da ferramenta facilitaria ou dificultaria esse processo?

Pergunta 4:

Espaço aberto para comentar e explicar a resposta acima. Se possível, inclua estimativas de tempo na sua resposta.

## APÊNDICE B

### Respostas das perguntas listadas anteriormente

*Desenvolvedor 1:*

Resposta da pergunta 1:

100 horas.

Resposta da pergunta 2:

80 horas.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

Acho que facilitaria bastante, mas dependeria da manutenção necessária, ainda mais com o risco de acabar alterando o comportamento do componente em alguma página, o que causaria mais manutenções.

*Desenvolvedor 2:*

Resposta da pergunta 1:

208 horas.

Resposta da pergunta 2:

184 horas.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

Considerando a aplicação que eu desenvolvi para a empresa (sistema de imagens), acredito que o framework ajudaria nos pontos básicos, mas ainda seria necessário outros itens como card e grid de imagens. Mas facilitaria o desenvolvimento e a centralização de código em

comum entre diferentes aplicações. Por isso considerei que diminuiria o equivalente a 3 dias de trabalho (considerando 8h diárias) de um dos desenvolvedores.

*Desenvolvedor 3:*

Resposta da pergunta 1:

O sistema era bem completo, a equipe era inexperiente e foram várias páginas, a parte do front-end demorou cerca de 100h de trabalho.

Resposta da pergunta 2:

75h, creio que foi gasto bastante tempo na estilização e validação de layout, usando uma lib que contenha isso, teria facilitado bastante.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

Acredito que se fosse bem estruturado, facilitaria na manutenção. Além disso, creio que uma lib ajudaria na padronização do layout em todas as páginas da CWI.

*Desenvolvedor 4:*

Resposta da pergunta 1:

80 horas.

Resposta da pergunta 2:

50 horas.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

Facilitaria muito, uma vez que não seria preciso acessar individualmente cada projeto para aplicar as manutenções, bastaria efetuar uma atualização do pacote dos componentes.

Certamente tende a otimizar muito a produção/padronização.

*Desenvolvedor 5:*

Resposta da pergunta 1:

12 horas.

Resposta da pergunta 2:

7 horas.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

Quando há um padrão já definido, fica mais fácil o trabalho.

*Desenvolvedor 6:*

Resposta da pergunta 1:

150 horas.

Resposta da pergunta 2:

100 horas.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

Acredito que essa biblioteca facilitaria muito o desenvolvimento front-end dentro da empresa, porque o tempo de lidar com a estilização dos componentes seria praticamente eliminado.

Além disso, sempre que necessário novos componentes poderiam ser criados, e reaproveitados facilmente.

*Desenvolvedor 7:*

Resposta da pergunta 1:

50 horas.

Resposta da pergunta 2:

30 horas.

Resposta da pergunta 3:

Facilitaria.

Resposta da pergunta 4:

O uso de componentes prontos facilita muito no desenvolvimento de uma página nova, principalmente quando essa página exige um padrão visual entre as outras páginas da mesma instituição(nesse caso a CWI), pois assim o desenvolvedor não precisa ficar gastando tempo em uma coisa que diversas outras páginas já tenham. Um ponto negativo na utilização de componentes, é na estilização. Se a instituição não tiver um padrão visual para os componentes, ou seja, cada página tiver o seu estilo, seus modelos de componentes, a utilização de um componente pronto apenas dificultaria. Porém, é muito mais provável que uma empresa/instituição tenha sim um padrão visual, fazendo com que a utilização do componente padrão dessa empresa/instituição seja sim um objeto facilitador para o trabalho do desenvolvedor, economizando assim, um valioso tempo.