

MORETS: Desenvolvimento de uma ferramenta para construção de sistemas tipográficos web responsivos

MORETS: Development of tool for building web responsive typographic systems

TARGINO, Mateus; Graduando; Universidade Federal do Ceará-UFC

mateusntargino@gmail.com

NOVAIS, Carlos; Doutor; Universidade Federal do Ceará-UFC

eduardonovais@virtual.ufc.br

O uso de sistemas tipográficos na criação de interfaces digitais é uma prática comum e recomendada. Se tratando de websites, há uma preocupação recorrente em manter os layouts confortáveis e consistentes para diferentes tamanhos de tela considerando a variedade de dispositivos presentes no mercado. Um dos principais desafios neste contexto são as mudanças feitas em características dos textos presentes nestes ambientes, como tamanhos de fonte e altura de linha. Aliado a isso, nos últimos tempos, as fontes variáveis vêm ganhando cada vez mais destaque e visibilidade por parte dos estudiosos e entusiastas de design e tipografia por suas potencialidades, por exemplo, mudar a largura dos tipos de forma dinâmica. Diante disto, este trabalho apresenta o processo de pesquisa e desenvolvimento de uma solução que visa criar sistemas tipográficos web responsivos. Para tanto, a ferramenta web aqui proposta se inspira em cânones do design com tipos, utiliza de técnicas de desenvolvimento web e as possibilidades das *variable fonts*, permitindo que os usuários criem sistemas *type systems* fluidos e customizáveis com comportamento responsivo.

Palavras-chave: Tipografia; Web Design Responsivo; Design de interfaces; Design editorial; Fontes Variáveis

Keywords: *Typography; Responsive Web Design; User Interface Design, Book Design; Variable fonts*

The use of typographic systems while creating digital interfaces is a common and recommended practice. When it comes to websites, there is a recurring concern to keep layouts comfortable and consistent for different screen sizes considering the variety of devices on the market. One of the main challenges in this context are the changes in the characteristics of the texts present in these environments, like font sizes and line heights. Furthermore, in recent times, variable fonts have been gaining emphasis and visibility from scholars and enthusiasts of design and typography for their potential, for example, changing character width dynamically. This work presents the research and development of a solution that aims to generate responsive typographic systems. The web tool proposed is inspired by canons of typography, uses web development techniques and the possibilities of variable fonts, allowing users to create fluid and customizable type systems with responsive behavior.

1 Introdução

A tipografia é um conjunto de práticas subjacentes à criação e à utilização de símbolos visíveis relacionados às letras, números e sinais de pontuação para fins de reprodução, independentemente do modo como foram criados (a mão livre, por meios mecânicos) ou reproduzidos (impressos em papel, gravados em um documento digital). Dentre as práticas do design tipográfico, temos a distinção entre design de tipos (em inglês, *type design*) e design com tipos (Farias, 2010), que é um dos principais tópicos tratados neste trabalho.

Um de seus principais usos práticos se dá através da aplicação de composição de blocos de textos. Decisões tipográficas, quando bem utilizadas, podem fazer grande diferença no resultado final do projeto editorial. Neste sentido, alguns recursos simples quando empregados já trazem grandes impactos na leitura. Para Lupton (2018, p.87), "desde de um simples recuo (sinalizando a entrada de uma nova ideia) até um link destacado (anunciando um salto para outro lugar), a tipografia ajuda os leitores a navegar pelo fluxo de conteúdo".

Uma das mídias mais tradicionais a carregar conteúdos textuais são os livros. Tschichold (1991) argumenta que para a construção de bons projetos de livros, deve-se trabalhar com o resgate histórico e a aplicação de regras de diagramação. Desta maneira, encontra-se em Bringhurst (2016) um compilado de diversas recomendações tipográficas baseadas na história.

Para além do livro impresso, as possibilidades abertas para a distribuição e consumo de textos pelos meios digitais apresentam novos desafios para a construção de layouts na medida em que novos meios e formatos de telas surgem. Considerando tantas possibilidades, o web design responsivo está associado a uma abordagem de design e desenvolvimento para criar websites onde o conteúdo e os elementos de design possam se adaptar em todos os dispositivos (Pamental, 2014).

As tecnologias tipográficas tiveram avanços consideráveis nos últimos anos. Mudanças surgiram a fim de melhorar e facilitar a composição com os tipos, por exemplo o formato *OpenType* desenvolvido pela Adobe e pela Microsoft com objetivo de aprimorar o formato *TrueType*, permitindo maior flexibilidade de uso e capacidade de armazenamento, além de outras possibilidades como a adição de ligaturas, suporte a grafemas Unicode e compatibilidade com os principais sistemas operacionais.

Já as *web font*, foram desenvolvidas por um consórcio de empresas como Mozilla, Type Supply, LettError entre outras organizações, com o intuito de permitir que o usuário visualize as fontes usadas em um website sem que elas estejam instaladas em seu computador, resolvendo assim as questões de licenciamento que limitavam o uso de fontes nos websites (W3SCHOOLS, 2019).

Em 2016 a Google, Apple e Microsoft anunciaram uma extensão para o *OpenType*, a *OpenType Variable Fonts* (Victionary 2019). Segundo Pamental (2018) as fontes variáveis consistem em uma tecnologia na qual diferentes larguras, pesos, inclinações, ascendentes, descendentes, itálicas e características adicionais são incorporadas em um único arquivo. Seu dispositivo pode exibir texto com todas estas variações estilísticas utilizando um único arquivo de fonte com menos espaço e largura de banda (Google, 2016).

Esta tecnologia abre espaço para mudanças no projeto gráfico digital. Um levantamento de Woloszyn, Meürer & Gonçalves (2019), destaca entre as potencialidades de *variable fonts* levantadas entre os *designers* consultados: o carregamento de fontes em web pages e seu uso em ambientes digitais responsivos.

A linguagem convencional para a estilização web é CSS, que aliada ao HTML possibilita a criação de sites. No contexto de responsividade, a W3C (2012) recomenda o uso de *media queries*, um recurso que facilita a escolha de pontos de quebra baseados no tamanho da tela do navegador para mudanças no estilo de uma página. Assim o desenvolvedor consegue criar layouts diferentes e adaptados para diferentes dispositivos.

Avançando neste conceito, Brown (2016) descreve o uso de uma técnica chamada *CSS Locks*, utilizada para alterar dinamicamente elementos tipográficos baseados na largura atual do navegador web ou do dispositivo. Motivado pelo início do suporte nos principais navegadores às *variable fonts*, Pamental (2018) desenvolve uma demonstração de página web utilizando o método, mudando também características variáveis da fonte (peso e largura) de acordo com a largura da tela, demonstrando algumas das possibilidades do uso da tecnologia na internet.

A criação de sistemas tipográficos é uma técnica comum durante o desenvolvimento de identidades visuais e interfaces digitais. Alguns exemplos são os presentes no *Material Design da Google* (2014), orientações do *FLuent Design System* da Microsoft (2017) e no *Carbon Design System* utilizado pela IBM (2015). Estes possuem regras de uso prático, semântica para os estilos, bem como definição de hierarquia e exemplos de uso para diferentes dispositivos.

Tais investidas, todavia ainda não aparentam ter absorvido as tecnologias e técnicas apresentadas, observa-se então que há espaço a ser explorado tais como o uso da tipografia fluida e variação de características de acordo com o tamanho da tela.

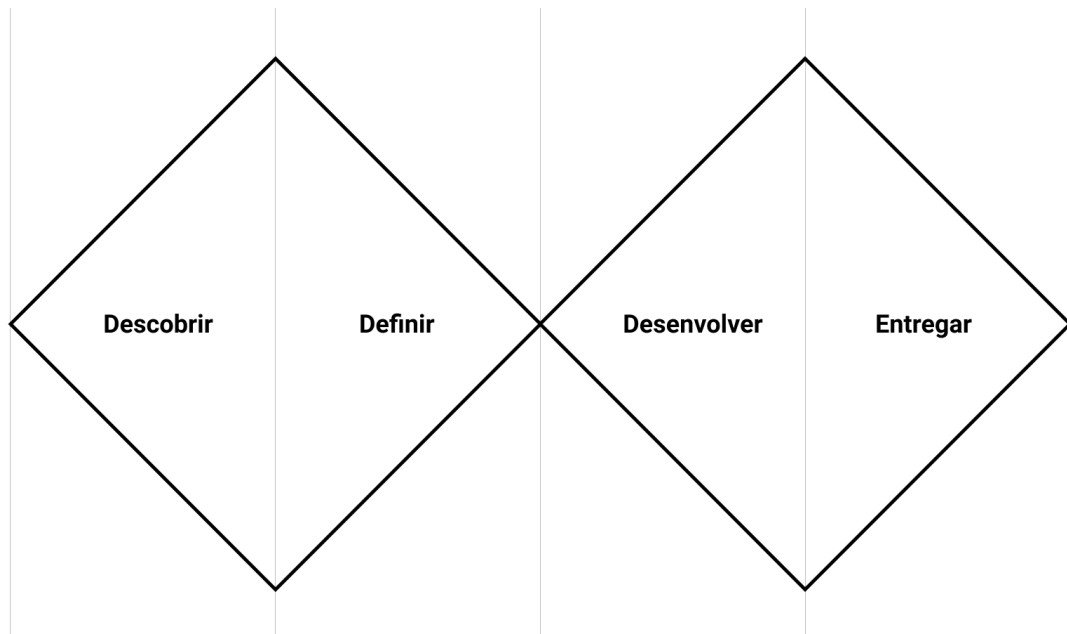
Diante das questões apresentadas, este artigo apresenta o processo de desenvolvimento de uma interface de construção de sistemas tipográficos responsivos e editáveis para utilização na criação de sites ou aplicativos para navegadores, incorporando novas tecnologias e possibilidades por elas geradas.

2 Metodologia

Analisando as necessidades e objetivos do projeto, foram buscadas metodologias existentes que compreendessem de maneira mais abrangente o processo de design para o desenvolvimento de softwares, onde fosse possível a adaptação e seleção de processos específicos para as necessidades do projeto. Foi selecionada a metodologia Design Thinking, um conjunto de técnicas e ferramentas centrado no usuário que suporta um processo iterativo para produzir e analisar soluções para problemas reais (Junior et al., 2013). Ele oferece modelos de processos para serem conduzidos não apenas por designers, mas por equipes multidisciplinares e em quaisquer tipos de organizações (Tschimmel, 2012).

O modelo de processo de design escolhido foi o chamado de duplo diamante (Design Council 2012), que é constituído por quatro etapas, denominadas: Descobrir, Definir, Desenvolver e Entregar. Em cada uma destas etapas é possível empregar uma ou mais técnicas para amadurecer o conhecimento da equipe de desenvolvimento acerca do produto em construção (Figura 1).

Figura 1 – Versão simplificada do duplo diamante.



Fonte: Adaptado pelo autor com base no modelo do Design Council

O modelo descreve quatro etapas para o desenvolvimento de um projeto. A primeira etapa é a de descoberta, uma etapa divergente, na qual o designer procura novas oportunidades, mercados, informações, tendências e insights (Tschimmel, 2012). Esta fase é também chamada de imersão preliminar, pois objetiva o entendimento inicial do problema (e Silva et al., 2012). Este momento pode ser utilizado para levantar as informações, criando um repertório teórico e técnico para o projeto, bem como justificativas para o desenvolvimento.

Em seguida, tem-se o estágio de definição, um tipo de filtro no qual os primeiros insights são revisados e selecionados (Tschimmel, 2012). Também chamada de imersão em profundidade, pois identifica as necessidades e oportunidades para a geração de soluções nas próximas etapas (e Silva et al., 2012).

Avançando, a terceira representa o desenvolvimento de potenciais soluções para a etapa anterior (Tschimmel 2012). Também chamada de análise e síntese, pois os envolvidos usam todos os dados obtidos, agrupando os que são em comum. Isto faz com que os problemas mais recorrentes sejam identificados (e Silva et al., 2012). No contexto deste projeto, a materialização das primeiras versões seriam produzidas nesta fase, possibilitando discussões sobre o possível uso das ideias.

Por fim, na quarta etapa, segundo diamante, visa a convergência da solução e marca a entrega, que por sua vez centra todas as etapas com a validação do que foi produzido (Tschimmel, 2012). Neste momento, foi possível dedicar o tempo para desenvolver o software e torná-lo funcional, assim possibilitando pesquisas para entender se o produto realmente entrega valor aos possíveis usuários.

3 Desenvolvimento

Nesta seção apresentaremos o processo de desenvolvimento baseado nas etapas da metodologia Design Thinking apresentada anteriormente, descrevendo cada processo e ferramenta selecionada.

3.1 Descoberta

Durante a fase da descoberta, foram executadas três ferramentas. A primeira delas foi uma pesquisa exploratória que tem por objetivo conhecer a variável de estudo tal como se apresenta, seu significado e o contexto onde ela se insere (Piovesan & Temporini 1995). No contexto deste trabalho, visa encontrar embasamento bibliográfico e melhor compreensão do problema.

Foram buscados trabalhos envolvendo os principais temas de estudo; Para tipografia, compreendemos que, por se tratar de um tema de cunho histórico e cultural, o ideal seria compreender melhor sua aplicação prática, os livros *Pensar com tipos* (Lupton, 2020) e *Elementos do estilo tipográfico* (Bringinghurst, 2013) foram utilizados como guias. Sobre responsividade, o livro *Responsive Typography* (Pamental, 2014) seria utilizado como fonte de informação. O artigo “Fontes variáveis: um estudo prospectivo” (Meürer & Gonçalves, 2019) foi importante para compreender dados sobre o uso, desejo e dificuldades sobre o assunto no mercado de design brasileiro, bem como o lançamento do site *variablefonts.io* (Pamental, 2020) que detalha conceitos e possíveis usos da tecnologia, reunindo também materiais externos para consulta.

Para complementar o aprofundamento teórico, foi feita uma pesquisa secundária, que visa reservar um tempo para explorar uma série de informações publicadas sobre seus clientes, seus concorrentes e tendências políticas, sociais e econômicas. É vital explorar e entender o contexto em que se está trabalhando e se manter atualizado com os desenvolvimentos mais recentes. Podem ser feitas online, ou através de sua biblioteca de referências pessoais (Design Council, 2012).

Um esforço foi feito para encontrar mais informações sobre fontes variáveis, artigos informais sobre o uso da tipografia na web e utilização de técnicas de desenvolvimento como *CSS Locks* (Tim Brown, 2012), *Fluid Typography* (Graham, 2019) e *The evolution of typography with variable fonts: an introduction* (Jason Pamental). Ferramentas para o uso da tipografia na web também foram encontradas e seriam comparadas com ideias futuras, através de uma análise competitiva, são elas: *Gridlover*¹, *Typeshift*², *Typetura*³, *Typescale*⁴, *A Typographic Scale Calculator*⁵, *Adaptive Modular Scale*⁶. Bem como bibliotecas existentes que auxiliam uma aplicação mais adequada ou facilitada de características de fontes na web.

Após o entendimento do contexto a ser trabalhado e com material teórico e construção de um repertório necessário para tratar da temática a ser desenvolvida no produto, se fez necessária a sondagem de possíveis funcionalidades e inovações a serem produzidas. Para tal, foi conduzido um *brainstorming*, que consiste em reunir pessoas que se concentram em um determinado assunto, proporcionando a dinâmica em grupo para coletar informações sob diversas perspectivas. O objetivo é garantir que a discussão não limite ideias ou que ideias tenham que ter sentido imediatamente, mas que pessoas trabalhem juntas para descobrir o máximo possível de ideias (Liikkanen, 2011).

¹ Disponível em: <<https://www.gridlover.net/>> Acesso em: 02 de fevereiro de 2022

² Disponível em: <<http://typeshiftapp.com/>> Acesso em: 02 de fevereiro de 2022

³ Disponível em: <<https://typetura.com/>> Acesso em: 02 de fevereiro de 2022

⁴ Disponível em: <<https://type-scale.com/>> Acesso em: 02 de fevereiro de 2022

⁵ Disponível em: <<https://www.layoutgridcalculator.com/typographic-scale/>> Acesso em: 02 de fevereiro de 2022

⁶ Disponível em: <<https://codepen.io/getflourish/full/vXqewy/>> Acesso em: 02 de fevereiro de 2022

O processo foi feito através da seleção e síntese das ideias apresentadas em webconferências, baseando-se em relevância e aderência ao tema trabalhado. Das propostas feitas, as que mais se destacaram foram:

1. A criação de uma ferramenta que possibilitasse designers e desenvolvedores a criar sistemas tipográficos web responsivos e facilitar seu uso.
2. Um ambiente de aprendizado sobre conceitos de fontes variáveis e responsividade tipográfica de forma prática. A primeira ideia se tornou o principal foco após os refinamentos, mas a segunda seria implementada como parte do artefato final, dando dicas de uso e explicações aos usuários.

3.2 Definição

Dois métodos foram utilizados para a etapa de definição. No primeiro momento, foi utilizada uma *assessment criteria*, um método de seleção das ideias mais promissoras para desenvolver, são úteis para levar em consideração as preocupações de várias partes interessadas ao decidir as melhores ideias a serem levadas adiante (Design Council, 2015). Com uma ideia principal selecionada, tem de se colocar em prática a definição das principais características presentes no produto digital a ser desenvolvido. Para isto, o método foi conduzido considerando limitações técnicas, tempo necessário e relevância para o projeto. Através desta análise, foi construída uma lista com as funcionalidades iniciais do sistema como descritas no Quadro 1 com suas respectivas justificativas.

Quadro 1 – Definição das funcionalidades do sistema

Funcionalidade	Descrição
Adaptação responsiva da largura da fonte variável	Alteração da característica de largura da fonte variável de forma fluida. Para que isto seja possível, todas as fontes disponíveis precisam possuir o eixo de largura dos tipos.
Adaptação responsiva dos tamanho do corpo de texto componentes tipográficos	Alteração do tamanho da fonte de forma fluida, isto é, de acordo com o tamanho da tela, seguindo a proposta de Graham (2017).
Ativação da opção de hifenização do texto principal	Tornar opcional ao usuário a utilização de hifenização automática do texto.
Ativação de pontuação suspensa automática	Opção de utilizar ou não a pontuação suspensa nos componentes de texto corrido. De acordo com Lupton (2018), a pontuação suspensa impede que as aspas e outros sinais comam a parte da borda esquerda bem alinhada de um bloco de texto.
Seleção de fontes variáveis para corpo de texto e título	Disponibilização de fontes tipográficas que possam ser alteradas e testadas para diferentes componentes. Pela proposta, se faz necessário o uso de fontes variáveis, para utilizar suas características.
Ativação da recomendação de largura de linha ideal.	Opção para o usuário selecionar a definição automática de uma largura de linha de texto confortável para leitura. Qualquer coisa de 45 a 75 caracteres é amplamente considerada como um

	comprimento de linha satisfatório para uma página de coluna única definida em uma face de texto serifada em um tamanho de texto. A linha de 66 caracteres (contando letras e espaços) é amplamente considerada ideal. Para trabalhos em várias colunas, uma média melhor é de 40 a 50 caracteres (Bringhurst, 2016).
Ativação do ajuste automático de entrelinhas de forma responsiva	Para acompanhar as mudanças de tamanho da fonte e largura da tela, a altura das entrelinhas deveria ser alterada de forma adequada. Quando uma composição responsiva encontra uma <i>viewport</i> , existem diferentes maneiras de preencher o espaço. O que mais interessa aqui é uma relação triádica fundamental na composição tipográfica – a do tamanho da fonte de um texto, altura da linha e comprimento da linha. Ajustar qualquer um desses elementos sem também ajustar os outros é uma receita para uma leitura desconfortável, que é uma das razões pelas quais os designers têm tanta dificuldade com o layout web fluido (BROWN, 2012).
Adaptação responsiva do tamanho óptico da fonte variável	Para facilitar a leitura em tamanhos diferentes, a proposta seria diminuir o tamanho óptico em dispositivos menores e aumentar para telas maiores. Para Pamental (2018) um bom tamanho óptico torna o tipo mais legível em tamanhos menores. Adaptá-lo ao tamanho em que é usado faz uma diferença notável.
Mudança da cor do fundo e texto para experimentação	Para que o designer possa visualizar a forma sua criação de maneira prática, uma área onde seja possível experimentar diferentes cores de texto e fundo com os componentes de texto já modificados.
Disponibilização do código gerado	Para que o usuário possa utilizar o sistema tipográfico criado, se faz necessária a disponibilização dos códigos gerados através das variáveis alteradas através de download ou cópia do material.
Seleção de escala hierárquica tipográfica	Possibilidade do usuário alterar a escala a ser utilizada para a hierarquia tipográfica. No século XVI, uma série de tamanhos comuns se desenvolveu entre os tipógrafos europeus, e a série sobreviveu com poucas mudanças e poucas adições por 400 anos. [...] Use a velha escala familiar, ou use novas escalas de sua própria criação, mas limite-se, a princípio, a um conjunto modesto de intervalos distintos e relacionados (Bringhurst, 2013).

Fonte: O autor.

O escopo inicial foi criado a partir da etapa anterior, mas ainda existia a necessidade de validar os diferenciais e relevância da criação do MORETS⁷, a fim de evitar esforço desnecessário para seu desenvolvimento. Para esta tarefa, foi feita uma análise competitiva, em que são exploradas empresas em um determinado setor da indústria ou nicho de mercado que estão competindo com os produtos ou serviços de sua empresa por participação de mercado. A análise pode ser uma exploração em profundidade dos cinco principais concorrentes, ou um número maior de concorrentes pode ser examinado (normalmente com menos profundidade na análise) (Withrow, 2006). Os fatores selecionados para a comparação foram os seguintes:

⁷ MORETS é uma sigla para Modular Responsive Typographic System

1. Gratuidade se refere a aplicação possuir valor para sua utilização.
2. Fontes variáveis possuem suporte pelo projeto.
3. Fluidez tipográfica se dá pela capacidade do sistema alterar características do texto em resposta ao tamanho da tela do navegador.
4. Escala tipográfica consiste na possibilidade de testar escalas hierárquicas predefinidas para o sistema tipográfico.

Os serviços comparados foram selecionados com base na presença de uma ou mais características almejadas que poderiam solucionar as questões levantadas. O resultado é demonstrado no Quadro 2, onde temos que nenhum dos produtos se encaixa em todos os requisitos, um indicativo de que a proposta pode resolver o problema de uma forma diferente.

Quadro 2 – Análise competitiva

Competidor	Gratuidade	Fontes variáveis	Fluidez tipográfica	Escala tipográfica
Adaptive Modular Scale	Sim	Não	Sim	Sim
Gridlover	Sim	Não	Não	Sim
Typeshift	Não	Sim	Sim	Não
Typetura	Sim	Sim	Sim	Não
Typescale	Sim	Não	Não	Sim
Typographic Scale Calculator	Sim	Não	Não	Sim

Fonte: O autor.

3.3 Desenvolvimento

Com as principais funcionalidades descritas, seria necessário um momento de materialização das ideias. Para isto foram utilizados dois tipos de protótipos. Os de média fidelidade são utilizados depois de trabalhar com alguns modelos iniciais e resolver os problemas mais óbvios e flagrantes, deve sair do modo divergente da fase de prototipagem em direção ao refinamento e testar detalhes um pouco mais sutis. Para avançar para o próximo estágio de prototipagem, é necessário adicionar mais detalhes e refinamentos aos protótipos, tornando-os mais parecidos com os produtos finais. (Dam & Siang, 2021). Protótipos de baixa fidelidade não foram utilizados considerando a natureza exclusivamente digital do processo de validação entre os autores.

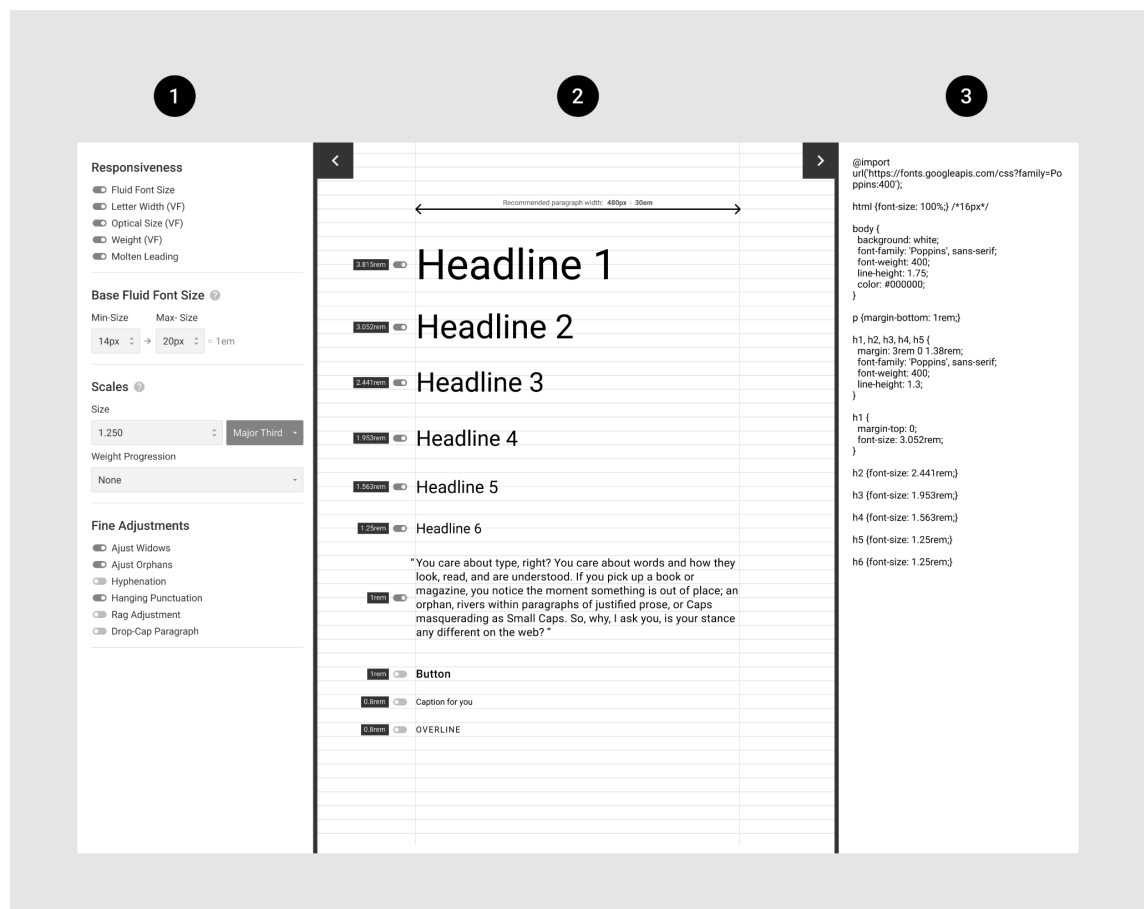
A primeira versão da interface digital foi criada com foco em sua utilização desktop, a qual seria provavelmente a mais comum devido ao teor profissional da aplicação. Os módulos desta opção são os seguintes:

1. Menu de edição das variáveis, definido como o principal elemento a ser utilizado para a modificação do sistema tipográfico.
2. Demonstração dos componentes, que se dá pela exibição do conjunto de elementos tipográficos. Os componentes foram inspirados pelo *Type System* do Material Design 2, um sistema de design criado pelo Google para ajudar as equipes a criar experiências digitais de alta qualidade para *Android*, *iOS*, *Flutter* e a *Web* (Google, 2022). Tal decisão

foi tomada considerando que o Design System da Google representa grande parte das convenções de design de interfaces e em prol de sua relevância. Uma diferença presente é a utilização de apenas um tamanho para o corpo de texto, já que este deve ser alterado de acordo com o tamanho da tela e tal prática é recomendada por Bringhurst (2013).

3. Menu de *handoff*, um espaço para os usuários acessarem o código gerado pela ferramenta. Na figura 2 é possível visualizar o material produzido, nesta opção, o utilizador poderia optar por mostrar ou esconder os módulos 1 e 3 através de botões.

Figura 2 – Primeira versão da interface com sinalização dos módulos



Fonte: O autor.

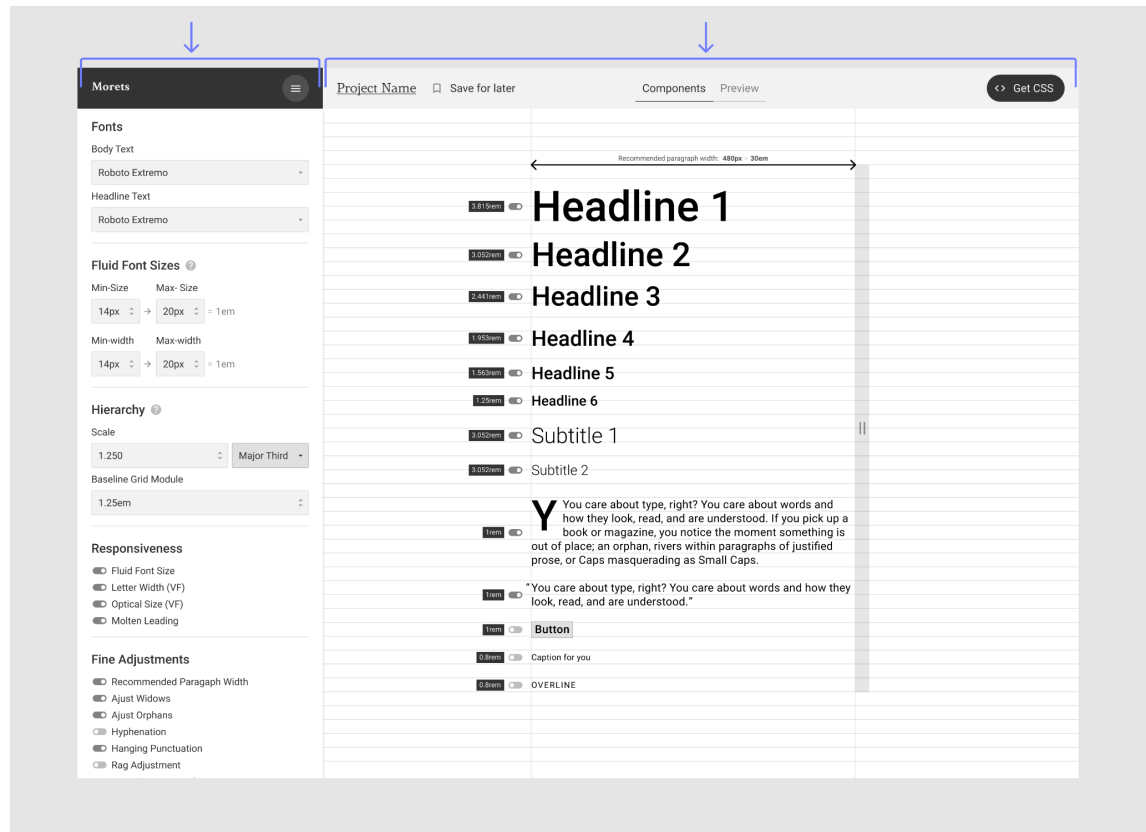
Na segunda variante, alguns detalhes foram adicionados e soluções de componentes foram modificados. Um cabeçalho foi criado para conter novos botões e opções de interação como demarcado na Figura 3.

O usuário poderia alterar no nome do projeto que seria criado no campo "Project Name", deixar aquela versão salva em sua sessão para alterar posteriormente com o botão "Save for later", alternar entre o ambiente de visualização dos componentes tipográficos na aba "Components" e pré-visualização através de "Preview" e clicar para visualizar o código no botão "Get CSS".

Além disso, uma modal foi adicionada em detrimento do “Menu de handoff” na primeira versão e agora o utilizador poderia copiar e fazer o download do arquivo CSS neste espaço destacado na Figura 4.

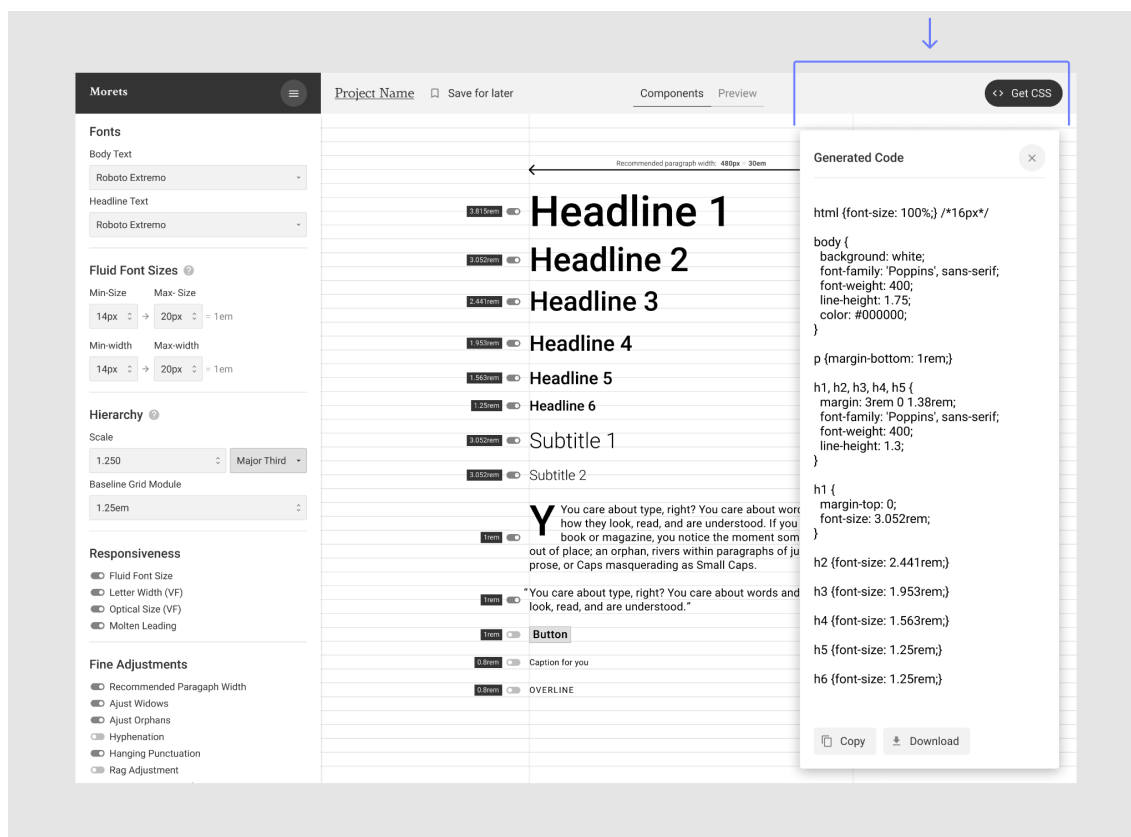
O “menu de edição das variáveis” foi mantido, porém com modificação da diagramação e reformulação dos campos.

Figura 3 – Segunda versão da interface



Fonte: O autor.

Figura 4 – Segunda versão a modal de handoff exibida



Fonte: O autor.

Após o refinamento da proposta nos protótipos de média fidelidade, seria necessário desenhar a ferramenta com definições visuais e seguindo uma visão final do produto a ser entregue posteriormente. Foram utilizados então, protótipos de alta fidelidade, que são a última linha de testes antes de passar para a execução das soluções. Eles permitem fornecer uma representação precisa de como a solução pode ser com detalhes finos; melhor ainda, eles podem incluir muitas das funcionalidades esperadas. Os modelos de alta fidelidade podem até conter todos os elementos funcionais necessários, embora possamos executá-los usando mecanismos ou tecnologia abaixo do ideal (Dam & Siang, 2021).

Seguindo com a prototipação, nesta versão, o “menu de edição das variáveis” manteve a categorização dos campos, mas seria possível colapsar ou esconder estes de acordo com a necessidade do usuário. Demais mudanças foram feitas em um âmbito estético que podem ser visualizadas na Figura 5, mantendo boa parte das ideias da segunda versão. A principal adição neste momento do desenvolvimento foi a demonstração do ambiente de pré-visualização, presente na Figura 6.

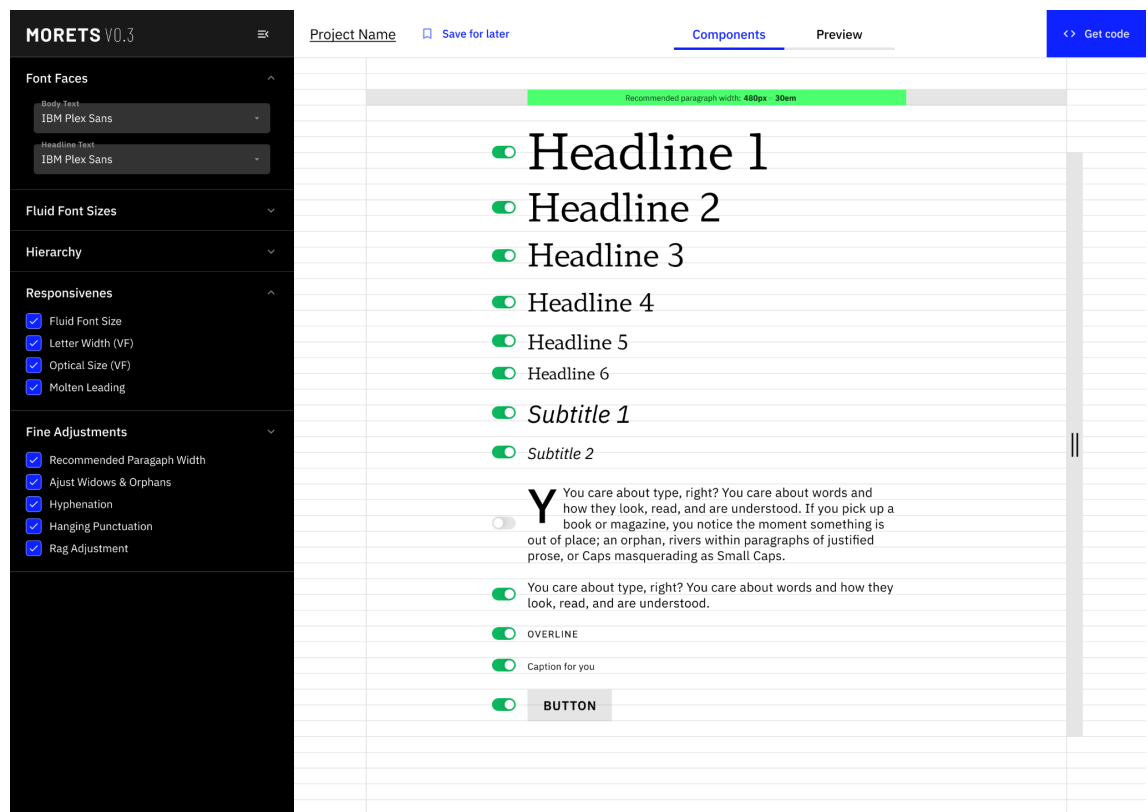
O conto *The Variable Man*⁸ de *Philip K. Dick* foi selecionado para o ambiente de demonstração do sistema tipográfico e está presente no site Project Gutenberg, uma biblioteca online de ebooks gratuitos. O texto foi escolhido considerando o uso de uma obra curta, em língua inglesa, para manter a consistência em relação à interface, com bastante variação dos

⁸ Disponível em: <<https://www.gutenberg.org/files/32154/32154-h/32154-h.htm>> Acesso em: 02 de fevereiro de 2022

parágrafos, utilização de aspas nas falas e o fato de não haver restrições de direitos autorais. Se tratando apenas de uma demonstração, não a exibição do conteúdo em si, alguns cortes no texto foram feitos para não tornar a página muito extensa.

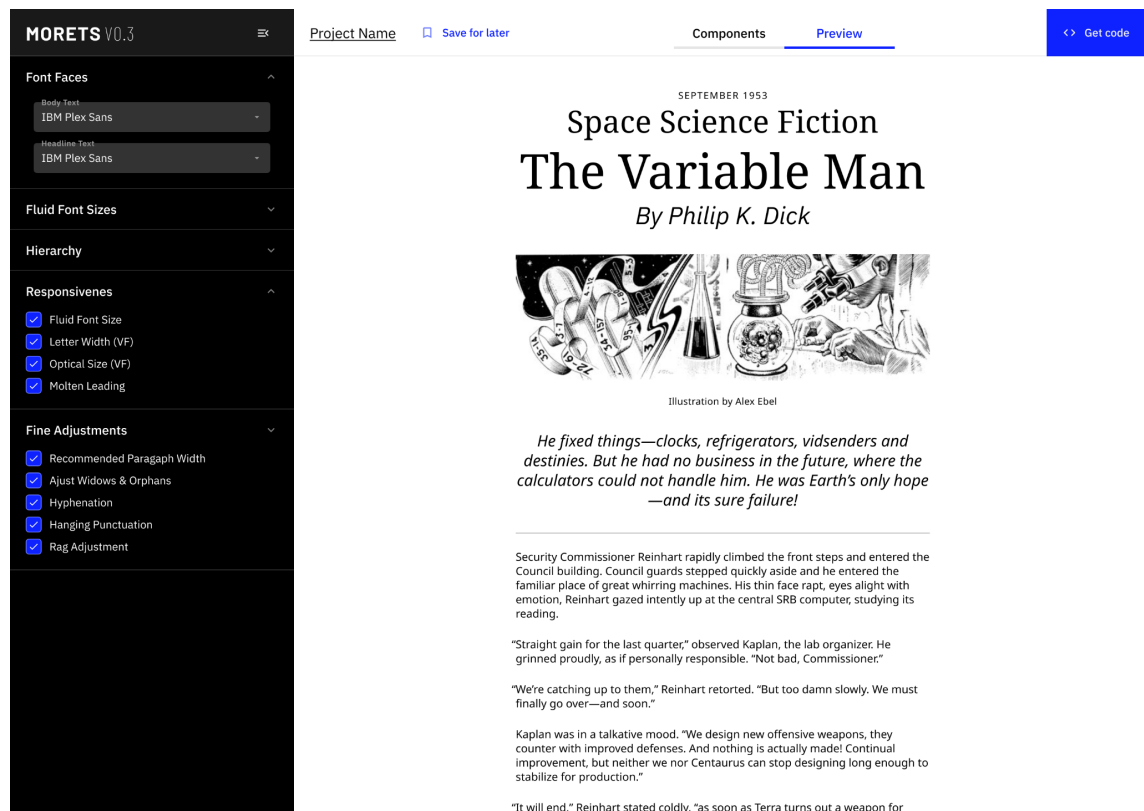
Todas as versões foram criadas através do Figma, um software focado no desenvolvimento de sistemas de design gráfico, prototipagem de interface gráfica de usuário e desenvolvimento de UI/UX (user interface experience ou experiência da interface com o usuário) (Gonzalez, 2020).

Figura 5 – Versão de alta fidelidade da interface na aba Components



Fonte: O autor.

Figura 6 – Versão de alta fidelidade da interface na aba Preview



Fonte: O autor.

3.4 Entrega

Com as primeiras versões dos protótipos de alta fidelidade finalizados, foi iniciada a fase de entrega do produto, onde o produto seria devidamente produzido a fim de ser publicado e disponibilizado para uso. A seguir são descritas as etapas e detalhes técnicos do processo. Apesar de seguir com uma fidelidade razoável as definições dos protótipos, algumas mudanças foram feitas e serão descritas durante o processo de codificação.

Por se tratar de uma ferramenta a ser utilizada em navegadores web, seu desenvolvimento necessitava de linguagens que possibilitassem criar um sistema neste meio. As tecnologias utilizadas foram:

1. HTML é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web, usa "Marcação" para anotar texto, imagem e outros conteúdos para exibição em um navegador da Web (Mozilla, 2022).
2. CSS é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML (incluindo várias linguagens em XML como SVG, MathML ou XHTML). O CSS descreve como elementos são mostrados na tela (Mozilla, 2022).
3. Javascript é uma linguagem de programação leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web. O JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional) (Mozilla 2022).

4. Pug é um motor de templates para Node e para o navegador. Ele compila para HTML e possui uma sintaxe simplificada, o que pode tornar o trabalho mais produtivo e o código mais legível (Hibbard, 2019).
5. Sass é um pré-processador de CSS, uma ferramenta que gera CSS com uma sintaxe mais rica e expressiva (Mozilla, 2022).
6. Gulp é um kit de ferramentas que ajudam a automatizar tarefas dolorosas ou demoradas no fluxo de trabalho de desenvolvimento (Github, 2022).
7. Node é um ambiente de execução multi-plataforma em JavaScript que permite aos desenvolvedores produzirem aplicações para rede e server-side usando o JavaScript.
8. Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência (Git, 2022).

Para que o site pudesse ser salvo, publicado e atualizado sem custo adicional, foram utilizadas duas plataformas de desenvolvimento sendo elas:

1. Github, que é uma plataforma de hospedagem de código para controle de versão e colaboração. Ele permite que você e outras pessoas trabalhem juntos em projetos de qualquer lugar (Github, 2022) e foi utilizado para o armazenamento e atualização dos arquivos através de um repositório privado.
2. Netlify fornece hospedagem para sites cujos arquivos de origem são armazenados no sistema de controle de versão Git e, em seguida, gerados em arquivos de conteúdo da web estáticos (Camden & Rinaldi 2017) e serviu para disponibilizar o site em uma URL aberta. Os arquivos atualizados e aprovados na ramificação de arquivos principal do Github são o código fonte para a última versão publicada. Isto foi feito através de uma funcionalidade de conexão entre as ferramentas descritas.

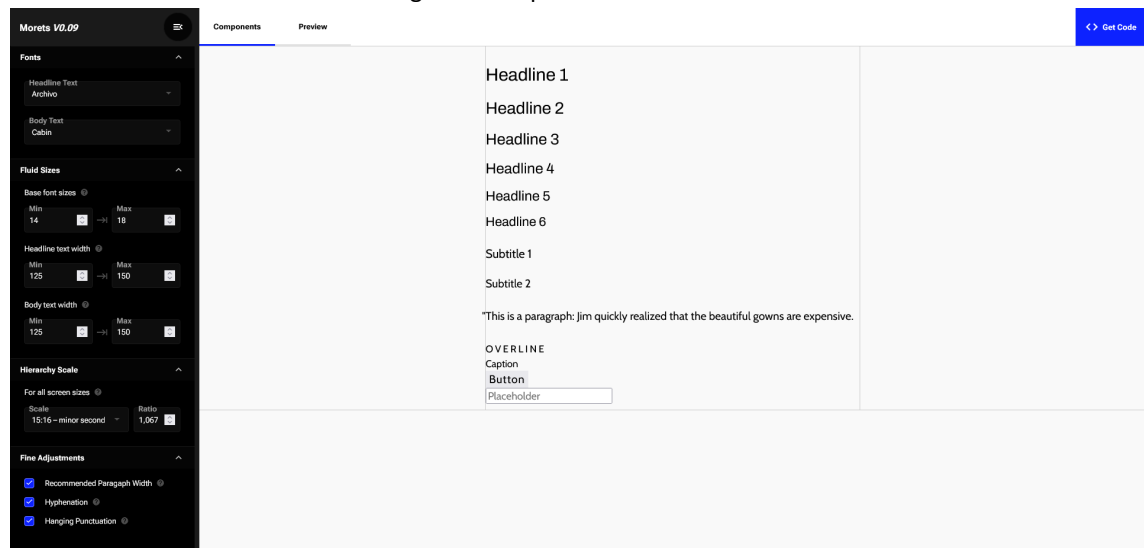
A interface foi produzida seguindo como base o protótipo de alta fidelidade, seguindo os mesmos módulos já definidos e adicionando a modal da função "Get code", exibida na Figura 8, que diferentemente da opção da Figura 4, não contém o botão de download por limitações técnicas e apresenta abas para a cópia de códigos HTML e Javascript ao invés de apenas CSS, necessidade que foi percebida posteriormente. O header possui os elementos já presentes na Figura 5 porém foi necessário remover as funcionalidades de alteração do nome do projeto e opção de salvar para mais tarde por falta de conhecimento técnico para produzir tal função. Na figura 10 é possível visualizar um exemplo da versão *mobile* do software, onde o menu de controles inicia colapsado e pode ser ativado como nos exemplos. O menu de controles localizado à esquerda da interface na figura 7 possui todas as possibilidades de mudança no sistema tipográfico na versão publicada até o dia 30 de janeiro de 2022. As opções são listadas em campos de formulário dos tipos *option*, *number* e *checkbox*. Este espaço da interface também pode ser colapsado de acordo com o interesse do utilizador, esta mudança pode ser vista na captura de tela da Figura 8.

Os campos de controle são:

1. *Headline Text* no qual o usuário pode selecionar a fonte a ser utilizada para os títulos.
2. *Body Text* onde é possível escolher a família tipográfica presente em subtítulos e de corpo de texto.
3. *Base font sizes (min & max)* em que o utilizador seleciona os tamanhos mínimos (relacionado com uma largura de tela de 320px) e máximos (relativo a telas com largura de 1600px) para o corpo do texto.

4. *Headline text width (min & max)* onde é possível selecionar as larguras mínimas (relacionada com uma largura de tela de 320px) e máximas (relativa a telas com largura de 1600px) para a fonte nos títulos.
5. *Body text width (min & max)* no qual são escolhidas as larguras mínimas (relacionada com uma largura de tela de 320px) e máximas (relativa a telas com largura de 1600px) para a fonte no corpo de texto e subtítulos.
6. *Scale/Ratio* com o intuito de escolher entre escalas pré-definidas ou um valor numérico customizado.
7. *Recommended paragraph width* para optar entre utilizar uma largura de linha de texto recomendada ou não. Esta opção segue a recomendação de Bringhurst (2013) para manter as linhas entre 45 e 75 caracteres e variando de acordo com o tamanho da tela.
8. *Hyphenation* no qual o usuário pode escolher hifenizar o texto do corpo.
9. *Hanging punctuation* habilitando a pontuação pendurada para os textos de corpo. Alguns destes campos possuem explicações sobre seu funcionamento e justificativas escondidas que podem ser acessadas através do clique em ícones de interrogação ao lado dos mesmos.

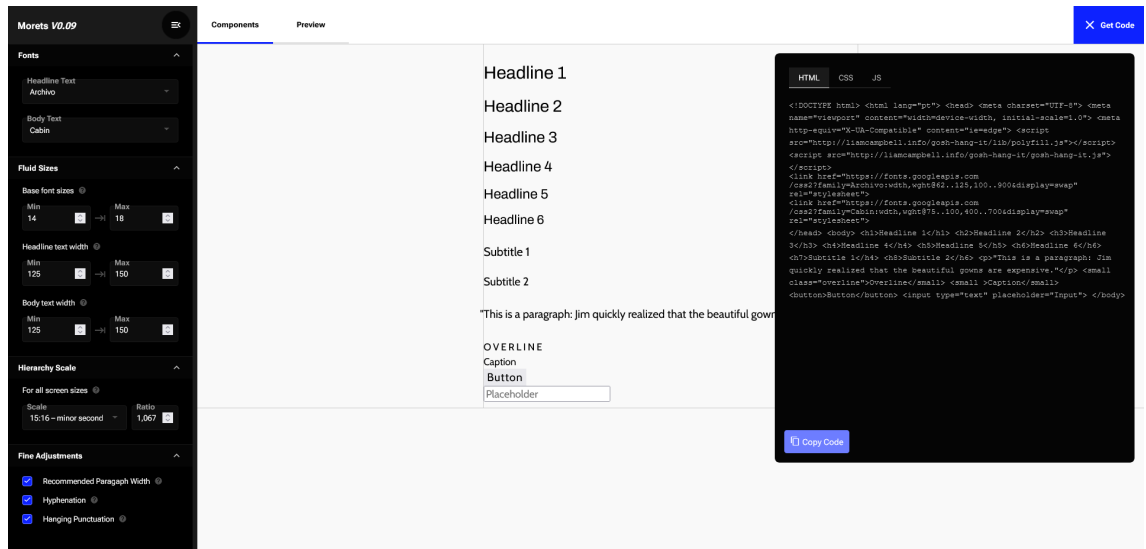
Figura 7 – Captura de tela do site⁹



Fonte: O autor.

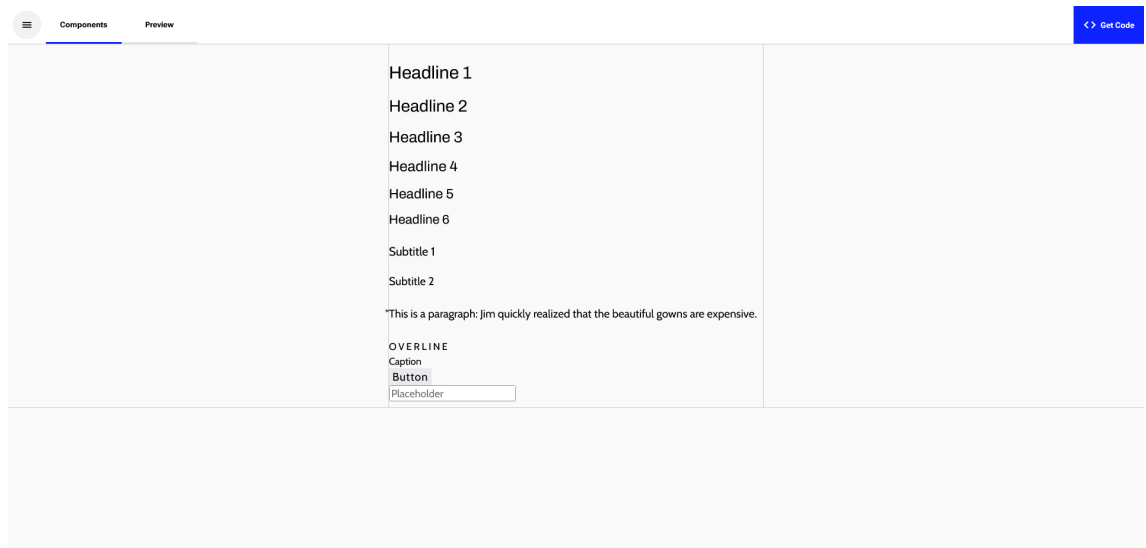
Figura 8 – Captura de tela do site com a modal de Get Code

⁹ Disponível em: <<https://morets.netlify.app/>> Acesso em: 02 de fevereiro de 2022



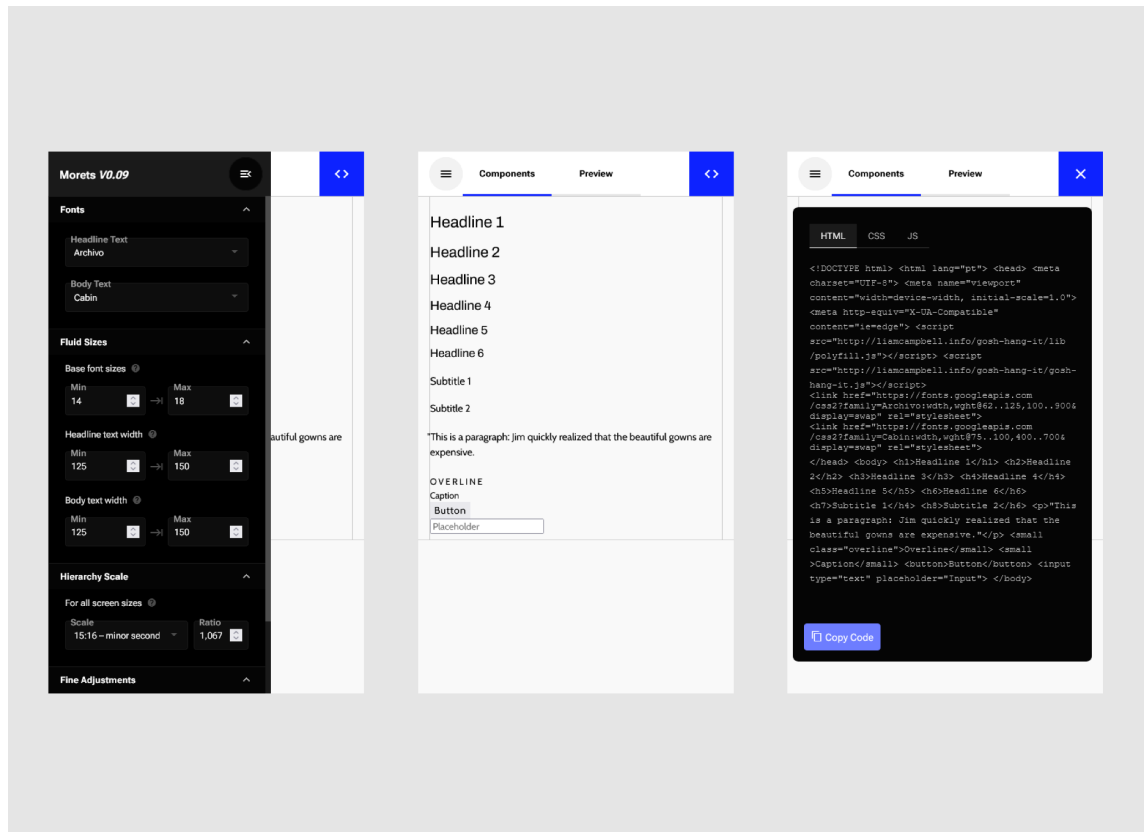
Fonte: O autor.

Figura 9 – Captura de tela do site com o menu de controles colapsado



Fonte: O autor.

Figura 10 – Capturas de tela do site em sua versão mobile



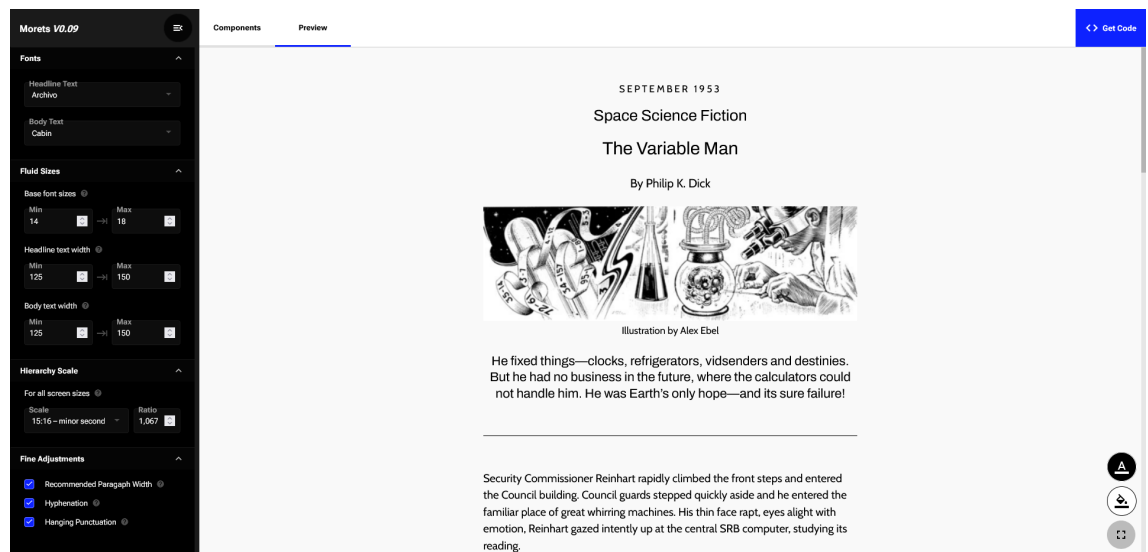
Fonte: O autor.

Assim como no protótipo demonstrado anteriormente na figura 5, o site demonstrado na figura 7 apresenta os componentes tipográficos, com seus tamanhos de fonte definidos pelas mudanças feitas pelo usuário no menu de controles. Mudanças foram feitas durante o processo de desenvolvimento do software, a remoção das linhas horizontais marcando as entrelinhas, os botões *switch* para a adoção ou não de determinados componentes para o sistema final foi excluída por limitações de tempo e falta de justificativas para a existência; existia um *handler* com o intuito de testar as modificações para diferentes larguras de tela mas foi descartado por dificuldade nas limitações desta ideia, como alternativa a esta necessidade, o usuário pode alterar o tamanho do navegador ou abrir o site em seu celular e modificar as variáveis equivalentes aos de seu teste no computador no menu de controle.

A aba de pré-visualização se manteve bastante fiel a proposta demonstrada na figura 6 e pode ser comparada com a figura 11 porém com a adição dos botões flutuantes com as seguintes funções:

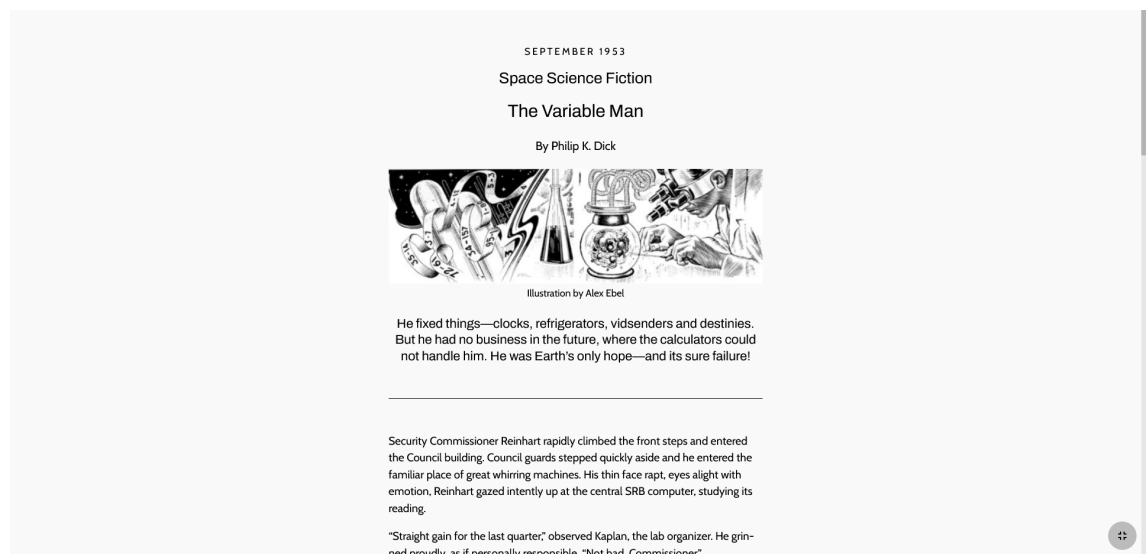
1. Alteração da cor do texto onde o usuário pode testar variações e combinações com o fundo.
2. Alteração da cor de fundo para que o usuário consiga modificar a cor de fundo e experimentar de acordo com a mistura que deseja.
3. Botão de tela cheia em que é possível esconder a interface e visualizar o texto com os componentes aplicados de maneira focada como exemplificado na figura 12. Na figura 13 se encontra uma captura de tela da adaptação *mobile* da aba de *preview*.

Figura 11 – Captura de tela do site na aba Preview



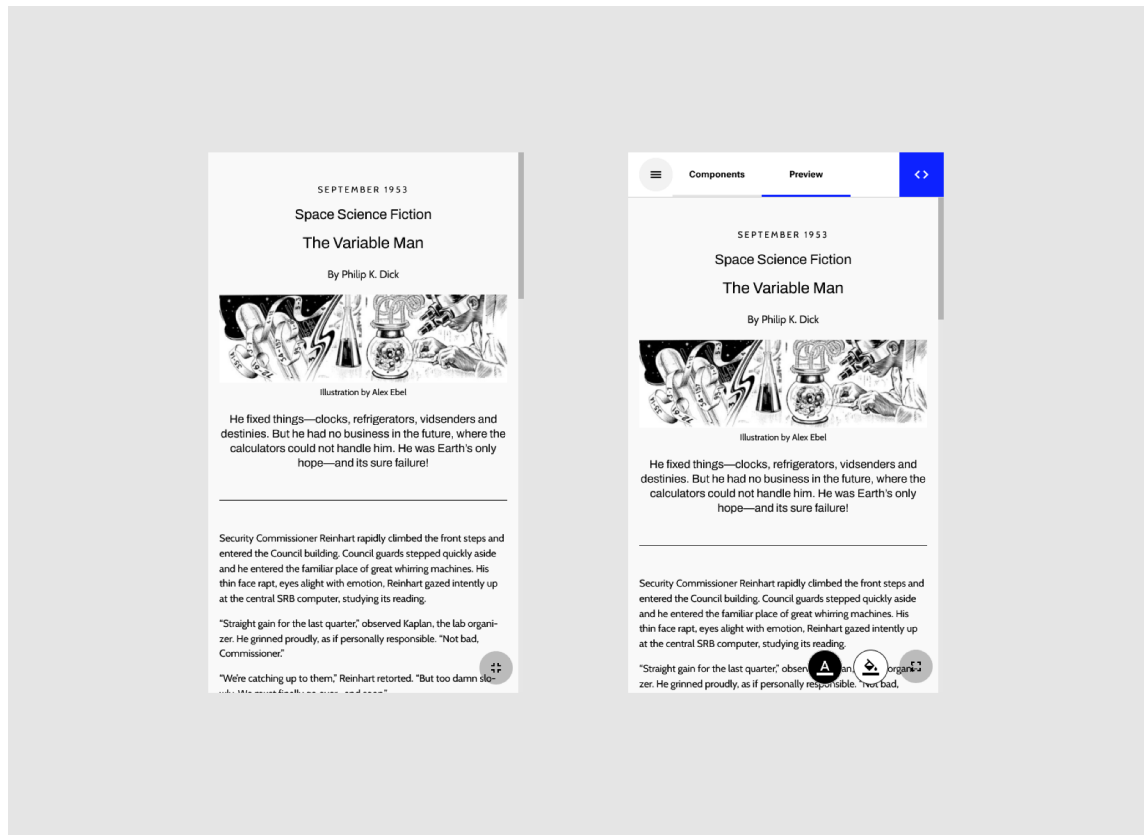
Fonte: O autor.

Figura 12 – Captura de tela do site na aba Preview em modo tela cheia



Fonte: O autor.

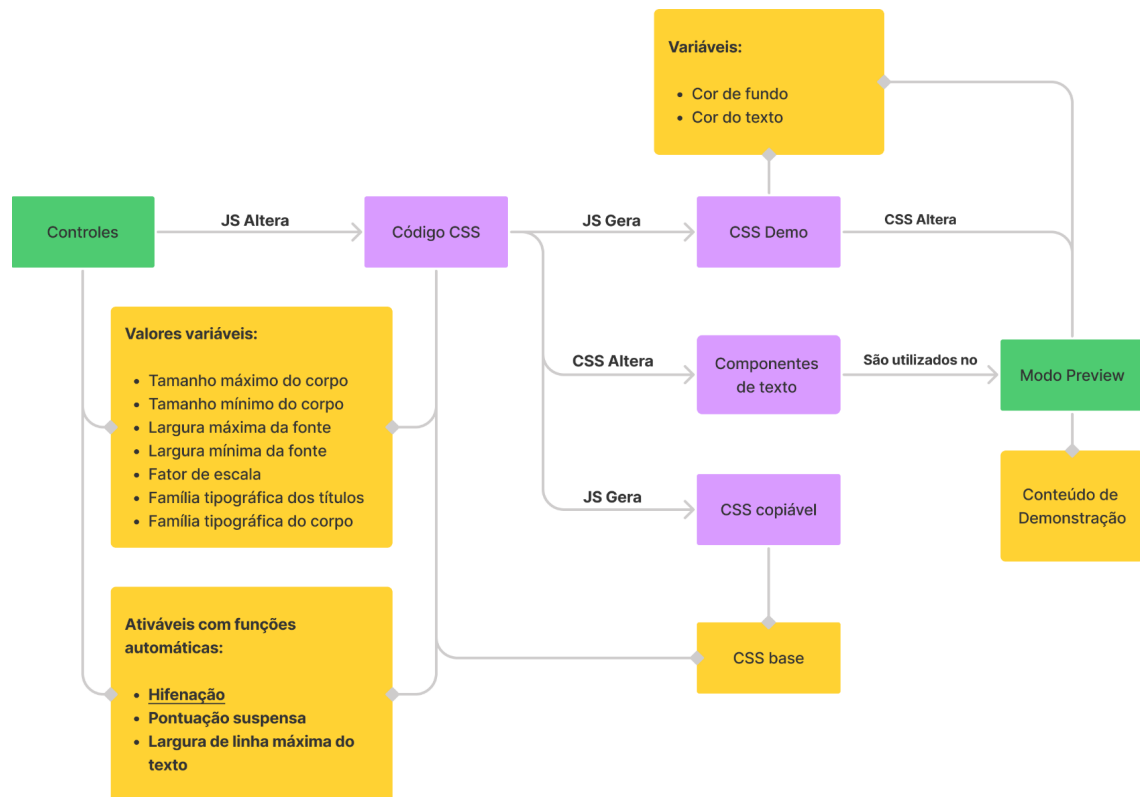
Figura 13 – Capturas de tela do site na aba Preview em sua versão mobile



Fonte: O autor.

A geração dos códigos a serem copiados na interface através da modal aberta pelo botão *get code* presente na figura 8 foi feita através de scripts na linguagem Javascript, onde o HTML em formato de texto corrido presente nas abas é alterado sempre que o usuário faz mudanças nas variáveis acessadas pelo menu de controles. A figura 14 contém um esquema exemplificando o funcionamento interno do software, incluindo quais variáveis são alteradas e quais são os elementos estáticos. Em roxo estão os conteúdos que circulam no sistema e são alterados pelas variáveis e elementos de código fonte, representados pelas formas amarelas. O propósito final deste espaço do site é possibilitar a utilização dos códigos em arquivos pessoais dos usuários.

Figura 14 – Estrutura básica do funcionamento do sistema



Fonte: O autor.

4 Considerações finais

Neste trabalho, foi apresentado o processo de desenvolvimento de uma ferramenta de criação de sistemas tipográficos responsivos para web, visando trazer uma alternativa aos modelos já estabelecidos para a solução do uso de elementos tipográficos em meios responsivos. O resultado alcançado com a pesquisa e desenvolvimento do projeto traz uma primeira versão com suas funcionalidades básicas, possibilitando assim o teste com usuários reais e uma futura validação do produto referente ao valor que pretende entregar bem como avaliações de usabilidade e experiência do usuário.

A versão descrita neste artigo ainda possui melhorias que podem ser feitas em diversos aspectos, além de limitações técnicas a serem resolvidas. Dentre as melhorias, uma aplicação mais fiel ao uso do eixo de largura das fontes variáveis, que no sistema se limita a valores máximos e mínimos que ponderam de acordo com cada caso suas variações; a aplicação correta da hanging punctuation de forma responsiva e sua devida aplicação na aba preview, onde não é possível utilizar; o código gerado e poderia utilizar novas técnicas CSS como o *clamp*, descrito em um uso semelhante ao do projeto no artigo de Kellum (2021); O uso da hifenização como padrão do sistema deve ser revisitado, já que em conteúdos web é muito difícil aplicar isto de forma adequada como as recomendações de Bringhurst (2013). Uma limitação técnica importante a ser analisada é a compatibilidade das fontes advindas do *Google Fonts*, as quais não são renderizadas corretamente no navegador Firefox da Mozilla.

Como citado durante o texto, o projeto também tem potencial para ser utilizado como uma ferramenta com fins educacionais, utilizando espaços com textos e materiais interativos

durante o uso da interface. Para isto, é importante produzir os materiais textuais e interativos necessários para tal.

Uma revisão da bibliografia deve ser feita para validar os pontos propostos como soluções através da pesquisa, bem como seria importante apresentar o material para especialistas e receber os devidos feedbacks para uma produção mais assertiva da ferramenta.

Por fim, vale destacar a natureza experimental do produto proposto, se utilizando de tecnologias e técnicas ainda pouco exploradas no meio acadêmico brasileiro. O uso de fontes variáveis no contexto de responsividade aparenta ter caminhos a serem construídos e discutidos, mas ainda se encontra em um estágio de pouca adoção como prática de mercado, apesar de haver esforço de entusiastas e empresas interessadas.

5 Referências

ADOBE SYSTEM INCORPORATED. **Designing Multiple Master Typefaces**. Adobe System Incorporated, 1997. Disponível em <https://www.adobe.com/content/dam/acom/en/devnet/font/pdfs/5091.Design_MM_Fonts.pdf>. Acesso em 15/06/2019.

BRINGHURST, R. **The Elements of Typographic Style: Version 4.3** 1.ed. Vancouver: Hartley & Marks Publishers, 2013.

BROWN, T. **Flexible typography with CSS locks**. Adobe Typekit Blog, 2016. Disponível em <<https://blog.typekit.com/2016/08/17/flexible-typography-with-css-locks/>>. Acesso em 30/01/2022.

CAMDEN, R.; RINALDI, B. **Working with Static Sites: Bringing the Power of Simplicity to Modern Sites**. O'Reilly Media, Sebastopol, 2017.

da SILVA, D. M.; BELLO, J. C. S. **O uso de design thinking na concepção de um produto de software**: Auxiliando instrutores de musculação. TCC, PUCRS, Porto Alegre, 2018.

DAM, R. F.; SIANG, T. Y. . **What Kind of Prototype Should You Create?** Interaction Design Foundation, 2020. Disponível em <<https://www.interaction-design.org/literature/article/what-kind-of-prototype-should-you-create>>. Acesso em 30/01/2022.

e SILVA, M. J. V.; e FILHO, Y. V.; ADLER, I. K.; de FIGUEIREDO LUCENA, B.; RUSSO, B. **Design thinking: inovação em negócios**. MJV Press, Rio de Janeiro, 2012.

ESTEVES, R.; FARIAS, P. **O Design Brasileiro de Tipos Digitais: a Configuração de um Campo Profissional** 1.ed. Blucher, São Paulo, 2010.

GIT. **Git**. Git, 2022. Disponível em <<https://git-scm.com/>>. Acesso em 30/01/2022.

GITHUB. **Hello World**. Github Docs, 2022. Disponível em <<https://docs.github.com/en/get-started/quickstart/hello-world>>. Acesso em 30/01/2022.

GITHUB. **What is Gulp**. Github, 2019. Disponível em <<https://github.com/gulpjs/gulp>>. Acesso em 30/01/2022.

GONZALEZ, R. **Figma Wants Designers to Collaborate Google-Docs Style**. Wired, 2017. Disponível em <<https://github.com/gulpjs/gulp>>. Acesso em 30/01/2022.

GOOGLE. **Introduction**. Material Design, 2018. Disponível em <<https://material.io/design/introduction>>. Acesso em 30/01/2022.

- GOOGLE. **Introducing OpenType Font Variations**. Google Open Source Blog, 2016. Disponível em <https://opensource.googleblog.com/2016/09/introducing-opentype-font-variations.html>. Acesso em 30/01/2022.
- GRAHAM, G. **Fluid Typography**. CSS Tricks, 2019. Disponível em <https://css-tricks.com/snippets/css/fluid-typography/>. Acesso em 30/01/2022.
- KELLUM, S. **Intrinsic Typography is the Future of Styling Text on the Web**. CSS Tricks, 2021. Disponível em <https://css-tricks.com/intrinsic-typography-is-the-future-of-styling-text-on-the-web/>. Acesso em 30/01/2022.
- HIBBARD, J. **A Guide to the Pug HTML Template Preprocessor**. Sitepoint, 2019. Disponível em <https://www.sitepoint.com/a-beginners-guide-to-pug/>. Acesso em 30/01/2022.
- JUNIOR, M.; FREITAS, R.; PERES, S.; FANTINATO, M.; STEINBECK, R.; e ARAÚJO, U. **Experimenting with design thinking in requirements refinement for a learning management system**. Simpósio Brasileiro de Sistemas de Informação, João Pessoa, 2013.
- LIKKANEN, L. A.; LAAKSO, M., BJÖRKLUND, T. **Foundations for studying creative design practices**. In: Proceedings of the Second Conference on Creativity and Innovation in Design. ACM, New York, NY, USA 2011. p.309–315.
- LUPTON, E. **Pensando com tipos: Guia para designers, escritores e estudantes**. 1ed. - Editora Gustavo Gili, 2020.
- MOZILLA. **The Web Open Font Format(WOFF)**. MDN Web Docs Mozilla, 2022. Disponível em <https://css-tricks.com/intrinsic-typography-is-the-future-of-styling-text-on-the-web/>. Acesso em 30/01/2022.
- NETLIFY. **About Netlify**. Netlify, 2022. Disponível em <https://css-tricks.com/intrinsic-typography-is-the-future-of-styling-text-on-the-web/>. Acesso em 30/01/2022.
- PAMENTAL, J. **Responsive Typography: Using Type Well on the Web**. 1ed. - O'Reilly Media, Sebastopol, 2014.
- PIOVESAN, A.; TEMPORINI, R. **Pesquisa exploratória: procedimento metodológico para o estudo de fatores humanos no campo da saúde pública**. Scielo Public Health, 1995. Disponível em http://www.scielo.org/scielo.php?pid=S0034-89101995000400010&script=sci_arttext&tling=>. Acesso em 30/01/2022.
- TSCHIHOLD, J. **The Form of the Book: Essays on the Morality of Good Design**. - Hartley & Marks, Vancouver, 1991.
- TSCHIMMEL, K. **Design thinking as an effective toolkit for innovation**. In: Proceedings of the ISPIIM Conference: Action for Innovation from Experience. Barcelona 2012. p.1–20.
- VICITIONARY. **On the road to variable: flexible future of typography**. - Two Points, Hong Kong, 1991.
- W3SCHOOLS. **CSS Web Fonts**. W3Schools, 2022. Disponível em https://www.w3schools.com/css/css3_fonts.asp. Acesso em 30/01/2022.
- MOZILLA. **CSS**. MDN Web Docs, 2021. Disponível em <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em 30/01/2022.

MOZILLA. **HTML:** Linguagem de Marcação de Hipertexto. MDN Web Docs, 2021. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em 30/01/2022.

MOZILLA. **Javascript.** MDN Web Docs, 2021. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em 30/01/2022.

MOZILLA. **Node.js.** MDN Web Docs, 2021. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Glossary/Node.js>>. Acesso em 30/01/2022.

WITHROW, J. **Competitive Analysis:** Understanding the Market Context. Boxes and Arrows, 2006. Disponível em <<https://boxesandarrows.com/competitive-analysis-understanding-the-market-context/>>. Acesso em 30/01/2022.

WOLOSZYN, M.; MEÛRER, M.; GONÇALVES, B. S. **Fontes variáveis:** um estudo prospectivo. In: Anais do 9º CIDI e 9º CONGIC. 14 Folhas. Sociedade Brasileira de Design da Informação – SBDI, 2019.