

# **Instituto Politécnico de Gestão e Tecnologia - GAIA**

## **Mestrado em Engenharia de Tecnologias e Sistemas Web**

### **Dissertação / Relatório de Projeto final de curso**

A importância e implementação de um Design System num contexto multimarca,  
multiproduto e multidimensão

Daniel Moreira

Vila Nova de Gaia

Ano 2023

# **Instituto Politécnico de Gestão e Tecnologia - GAIA**

## **Mestrado em Engenharia de Tecnologias e Sistemas Web**

### **Dissertação / Relatório de Projeto final de curso**

A importância e implementação de um Design System num contexto multimarca,  
multiproduto e multidimensão

Daniel Moreira

Dissertação submetida para satisfação parcial dos requisitos do grau de Mestre em  
Engenharia de Tecnologias e Sistemas Web sob a orientação da Prof.<sup>a</sup> Doutora  
Mónica Santos

Vila Nova de Gaia

Ano 202



**INSTITUTO POLITÉCNICO DE GESTÃO E TECNOLOGIA**

**A importância e implementação de um Design System num contexto multimarca,  
multiproduto e multidimensão**

Daniel Filipe Oliveira Moreira

Aprovado em 23/10/2023

Composição do Júri

Prof. Doutor Firmino Oliveira Silva

Presidente

Prof. Doutor Sandro Carvalho

Arguente

Prof.<sup>a</sup> Doutora Carla Mónica Santos

Orientador/a

Vila Nova de Gaia

2023

## AGRADECIMENTOS

Com a conclusão de mais uma etapa acadêmica, quero expressar a minha grande gratidão a todas as pessoas e organizações que tornaram a realização deste trabalho possível.

Primeiramente, quero expressar os meus agradecimentos a minha orientadora, Professora Mónica Santos, pela sua orientação, apoio constante e por estar sempre disposta a ajudar sempre que tivesse dúvidas sobre a minha pesquisa. A sua experiência e dedicação foram fundamentais para a finalização deste trabalho.

Gostaria de agradecer também a minha orientadora de estágio, Fátima Ferreira, em que sempre me ajudou ao decorrer do estágio e no relatório em si. As suas contribuições e apoio foram essenciais para a melhoria do meu percurso durante o estágio.

Não posso deixar de agradecer aos meus colegas de equipa UX Chapter, pela colaboração e por abraçarem-me na equipa desde o primeiro dia. Compartilhar ideias e desafios com todos enriqueceu tanto a minha jornada acadêmica como a minha jornada profissional.

À minha instituição acadêmica, ISLA Gaia, e organização de estágio, MC Sonae, agradeço pela oportunidade de realizar esta pesquisa e trabalho, e pelos recursos disponíveis que foram fundamentais para este projeto.

Por fim, quero agradecer este trabalho aos meus pais e amigos, que sempre me incentivaram e apoiaram-me em todas as fases deste trabalho. A conclusão desta etapa acadêmica não seria possível sem o vosso apoio.

## RESUMO

Um dos fatores críticos de sucesso das organizações é a sua capacidade de entrega de soluções num *time to market* cada vez menor, dado o contexto volátil e complexo em que atualmente vivemos e a clientes cada vez mais informados e exigentes. Adicionalmente, e face ao exposto, um dos maiores desafios no desenvolvimento de produtos digitais, principalmente em organizações com vários produtos e marcas, é manter a consistência de identidade, a consistência de marcas e de experiência.

Para além do crescimento das empresas e equipas de produto, as próprias equipas mudam e crescem, o que torna este processo ainda mais complexo, dada a necessidade de *handover* e *onboarding* das pessoas envolvidas naquelas que são as *guidelines* para desenvolvimento. Sendo assim, é importante estabelecer regras e sugestões para a utilização dos fundamentos, componentes e elementos nos produtos, de forma a respeitar e garantir aquilo que é a identidade das marcas, produtos e sobretudo a consistência da experiência do cliente enquanto peça central em cada decisão de desenvolvimento.

O Design System apresenta-se assim como um produto que serve outros produtos, que guia o desenvolvimento, que permite a reutilização de forma consistente e sobretudo um elemento que vive da colaboração de todas as equipas que trabalham na centralidade do cliente.

A metodologia aplicada neste estudo tem como objetivo explicar e analisar a importância de um Design System e como implementar com as melhores práticas de *UX/UI*. A revisão da literatura aborda a introdução de todas as tecnologias usadas e todas as referências de Design Systems, o papel e responsabilidades dos designers e desenvolvedores, demonstrando as métricas usadas para a definição de regras e a implementação.

No âmbito deste trabalho foi igualmente desenvolvido um projeto de criação de um Design System para a empresa MC SONAE, em que são mostrados todos os processos de desenvolvimento do código para componentes que compõem o DS, os processos de *brainstorming* e a realização dos componentes em design e criação da documentação.

Palavras-chave: UI/UX Design; Design System; CSS; Figma; Outsystems; MC SONAE

## ABSTRACT

One of the critical success factors of organizations is their ability to deliver solutions in an increasingly shorter time to market, given the volatile and complex context we currently live in and increasingly informed and demanding customers. Additionally, one of the biggest challenges in the development of digital products, especially in organizations with several products and brands is to maintain consistency of identity, consistency of brands and experience.

In addition to the growth of companies and product teams, the teams themselves change and grow, which makes this process even more complex, given the need for handover and onboarding of the people involved in those guidelines for development. Therefore, it is important to establish rules and suggestions for the use of foundation, components, and elements in products, in order to respect and guarantee what is the identity of brands, products and, above all, the consistency of the customer experience as a central part of each decision.

The Design System thus presents itself as a product that serves other products, that guides development, that allows consistent reuse and, above all, an element that lives from the collaboration of all the teams that work in customer centricity.

The methodology applied in this study aims to explain and analyze the importance of a Design System and how to implement it with the best UX/UI practices. Starting with the introduction of all technologies used and all Design Systems references, the role and responsibilities of designers and developers, demonstrating the metrics used for rule definition and implementation.

Within the scope of this project, a project was also developed a project to create a Design System for the company MC SONAE, in which are shown all the code development processes for components that make up the DS, the brainstorming processes and the realization of components in design and the creation of documentation.

Keywords: UI/UX Design; Design System; CSS; Figma; Outsystems; MC SONAE

## ÍNDICE

<b>AGRADECIMENTOS .....</b>	<b>I</b>
<b>RESUMO.....</b>	<b>III</b>
<b>ABSTRACT.....</b>	<b>IV</b>
<b>ÍNDICE.....</b>	<b>V</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>VII</b>
<b>GLOSSÁRIO.....</b>	<b>1</b>
<b>INTRODUÇÃO.....</b>	<b>2</b>
MC SONAE .....	4
LEITURA .....	6
<i>Atomic Web Design</i> .....	6
<i>Fundamentals of Design Systems</i> .....	7
DESIGN SYSTEM .....	8
<i>Definição</i> .....	8
<i>Qual a diferença entre Design System, Style Guide e Pattern Library?</i> .....	9
<i>Vantagens</i> .....	10
<i>Princípios</i> .....	11
<i>Elementos base</i> .....	12
<i>Exemplos</i> .....	14
<i>Atomic Design</i> .....	23
<i>Design Tokens</i> .....	25
FERRAMENTAS UTILIZADAS .....	27
<i>Figma</i> .....	27
<i>OutSystems</i> .....	29
<i>Zeroheight</i> .....	31
<b>MÉTODO .....</b>	<b>33</b>
PROJETO – DEFINIÇÃO E OBJETIVOS .....	33
REQUISITOS .....	34

<i>Requisitos Funcionais</i> .....	34
<i>Requisitos Não Funcionais</i> .....	34
COMO SE MATERIALIZA .....	35
ORGANIZAÇÃO DA PASTA FIGMA.....	35
<i>Foundations</i> .....	37
<i>Components</i> .....	39
<i>Antes VS Depois - Organização do ficheiro Components</i> .....	41
<i>Patterns</i> .....	43
<i>Templates</i> .....	44
<i>Graphics</i> .....	45
<i>Icons</i> .....	46
<i>Helpers</i> .....	47
<i>Rules</i> .....	48
PROCESSOS PARA INSERÇÃO DE NOVAS FUNCIONALIDADES.....	49
criação da biblioteca figma.....	50
TRANSIÇÃO DE FIGMA PARA CÓDIGO .....	52
UTILIZAÇÃO DO MANTRA NO FIGMA .....	54
ZEROHEIGHT .....	57
<i>Atualização do Zeroheight para uma melhor aparência visual</i> .....	61
EXEMPLOS PRÁTICOS DO MANTRA .....	63
<b>RESULTADOS</b> .....	<b>66</b>
<b>CONCLUSÕES E PERSPETIVAS FUTURAS</b> .....	<b>67</b>
<b>BIBLIOGRAFIA</b> .....	<b>69</b>



## ÍNDICE DE FIGURAS

Figura 1 - Organograma da empresa.....	5
Figura 2 - Fotografias da entrada da BIT.....	5
Figura 3 - Imagem ilustrativa de uma organização de um Design System.....	8
Figura 4 - Imagem ilustrativa das diferenças entre Design System, Pattern Library e Style Guide. ....	9
Figura 5 - Capturas de ecrã tirados do site Material Design ( <a href="https://m3.material.io/">https://m3.material.io/</a> ) .....	16
Figura 6 - Capturas de ecrã tirados do site Carbon Design System ( <a href="https://carbondesignsystem.com/">https://carbondesignsystem.com/</a> ).....	18
Figura 7 - Capturas de ecrã tirados do site Decathlon Design System ( <a href="https://www.decathlon.design/726f8c765/v/0/p/71b8e3-decathlon-design-system">https://www.decathlon.design/726f8c765/v/0/p/71b8e3-decathlon-design-system</a> ) .....	20
Figura 8 - Capturas de ecrã tirados do site Primavera Design System ( <a href="https://design.primaverabss.com/">https://design.primaverabss.com/</a> ).....	22
Figura 9 - Modelo de Atomic Design .....	23
Figura 10 - Representação da aplicação dos design tokens. Imagem retirada do website oficial do Zeroheight ( <a href="https://zeroheight.com/">https://zeroheight.com/</a> ).....	25
Figura 11 - Logotipo do Figma e o seu software com a funcionalidade de vários designers e programadores estarem no mesmo ficheiro.....	28
Figura 12 - Logotipo do Outsystems e o seu software .....	29
Figura 13 - Logotipo do Zeroheight e o seu software.....	31
Figura 14 - Logotipo do Mantra e a sua página de documentação .....	33
Figura 15 - Esquema de Materialização do Mantra.....	35
Figura 16 - Pasta do Figma do Mantra .....	36
Figura 17 - Imagens dos Logotipos e Cores das Foundations .....	38
Figura 18 - Imagem da organização de um componente do Mantra.....	40

Figura 19 - Antes e depois da organização do <i>template</i> de botões .....	42
Figura 20 - Imagem do <i>Pattern Floating Button</i> .....	43
Figura 21 - Imagem dos <i>Templates Avisos</i> .....	44
Figura 22 - Imagem dos <i>Graphics Illustrations</i> .....	45
Figura 23 - Imagem dos <i>Icons Streamline</i> .....	46
Figura 24 - Imagem dos <i>Assets Keyboards</i> .....	47
Figura 25 - Imagem das <i>Rules Nomenclaturas</i> .....	48
Figura 26 - Imagem demonstrativa dos componentes do Mantra com um exemplo de um <i>Component</i> e um <i>Pattern</i> .....	51
Figura 27 - Imagem demonstrativa dos estilos do Mantra com as cores e tipografia.....	51
Figura 28 - Imagem demonstrativa dos componentes do Mantra numa inspeção do CSS..	52
Figura 29 - Imagem demonstrativa da exportação de <i>assets</i> visuais .....	53
Figura 30 - Imagem demonstrativa da organização do ficheiro “Template Mobile” .....	54
Figura 31 - Imagem demonstrativa da secção “ <i>Assets</i> ” .....	55
Figura 32 - Imagem demonstrativa da atualização de Componentes do Mantra nos Projetos .....	56
Figura 33 - Homepage do Mantra.....	57
Figura 34 - Secção Marcas do Mantra .....	58
Figura 35 - Secção Design do Mantra .....	59
Figura 36 - Secção “O Mantra” do Mantra.....	60
Figura 37 - Comparação entre Variações .....	61
Figura 38 - Diferentes áreas do Zeroheight Mantra, com a sua descrição e ilustrações.....	62
Figura 39 - Antes e depois da aplicação Continente SIGA .....	63
Figura 40 - Antes e depois da aplicação Cartão Continente .....	64
Figura 41 - Antes e depois da aplicação Better Together .....	65

Figura 42 - Nova Célula de UX e Design Tokens .....68

## GLOSSÁRIO

Nesta secção são explicados alguns termos utilizados ao longo deste trabalho, e que são frequentemente utilizados na MC Sonae.

Mantra Design System (Mantra) – Design system da MC

Product Owner (PO) – Responsável por determinado produto digital

MC Sonae (MC) – Empresa em que foi realizado o estágio

BIT – Área de sistemas de informação que atende os negócios de retalho alimentar e especializado da Sonae

*Customer Centric* – Fluxos e experiências do utilizador iguais em todas as aplicações, dando ênfase sempre a melhor usabilidade para todos os utilizadores

Favicons – Ícones simplificados usados para representar uma marca ou website no browser, o mais comum são logotipo da empresa ou o tema do website

Equipa RAD – Equipa de programação, em que trabalha com a linguagem Outsystems

Produto Digital – Aplicação Mobile ou Web

*Open-source* – Software com Código aberto, possibilitando os utilizadores de alterarem o código da ferramenta.

Biblioteca *master* – Biblioteca principal de derivada aplicação ou ferramenta

Revamp – Mudança completa de uma aplicação, tanto em termos de experiência de utilizador como visual

## INTRODUÇÃO

Os produtos digitais da BIT, área de sistemas de informação que atende os negócios de retalho alimentar e especializado da Sonae, têm sempre uma organização de equipas em que tem Desenvolvedores e UX Designers diferentes, cada um com as suas ideias e formas de trabalho. Este facto resulta em **inconsistências** no trabalho final se forem comparados os diversos produtos, em termos de interface e experiência de utilizador. Portanto, se um utilizador fosse usar diferentes aplicações, poderiam não sentir que eles pertenciam a mesma companhia.

Para além disso, a equipa tinha sempre de refazer elementos simples, e a maior parte das vezes repetidos, como por exemplo, *layouts* e botões, em que pode levar a pequenos erros de execução, perda de tempo e perde uniformidade entre aplicações, tirando valor ao produto final.

Para a resolução desses problemas convergiu-se para a construção de um *Design System*, um sistema vivo, um projeto que nunca acaba, sempre em constante evolução diante as necessidades dos projetos da empresa, que alimenta todos os projetos feitos pela equipa de UX Chapter, a equipa a que pertenci durante o estágio de seis meses na BIT, que consegue fornecer aos elementos da equipa todos os componentes previamente feitos e configurados para serem usados nos projetos.

Isto origina **consistência** e rapidez nos trabalhos entregues, pois os desenvolvedores têm também a facilidade de reconhecer facilmente o que é cada componente porque têm uma biblioteca com os mesmos e *layouts* definidos.

Para além da rapidez de entrega, melhora a comunicação entre designer e programador, por conseguirem falar de componentes em que ambos estão familiarizados, conseguindo focar-se somente na experiência do utilizador.

Além de aumentar a **consistência**, cria também a identidade da marca, para quando os utilizadores forem usar aplicações consigam ter a mesma experiência de utilização e consigam sentir que estão a utilizar uma aplicação da MC.

O *Design System* é chamado “Mantra Design System” e é atualmente o *Design System* da MC Sonae. O seu maior objetivo é tornar a experiência do utilizador *Customer Centric*,

fazendo que todos os projetos da MC tenham a mesma experiência do utilizador. Antes do Mantra, a experiência do utilizador era *User Centric*, cada aplicação tinha os seus fluxos e experiências diferentes.

Atualmente a equipa UX Chapter é uma equipa que trabalha com praticamente todas as soluções digitais da empresa, como o Cartão Continente, Continente Siga, e diversas aplicações internas. Tem, portanto, a responsabilidade de, para além alimentar dezenas de projetos, criar um *Design System* em que **todos os colaboradores da empresa** consigam aceder e perceber o seu propósito.

### **Objetivos gerais e específicos**

- Perceber as vantagens das empresas em ter um Design System;
- Estudar melhores princípios da criação de um Design System;
- Desenvolver um Design System multiproduto e multimarca;
- Aplicação de Atomic Design;
- Aplicação de Design Tokens

## MC SONAE

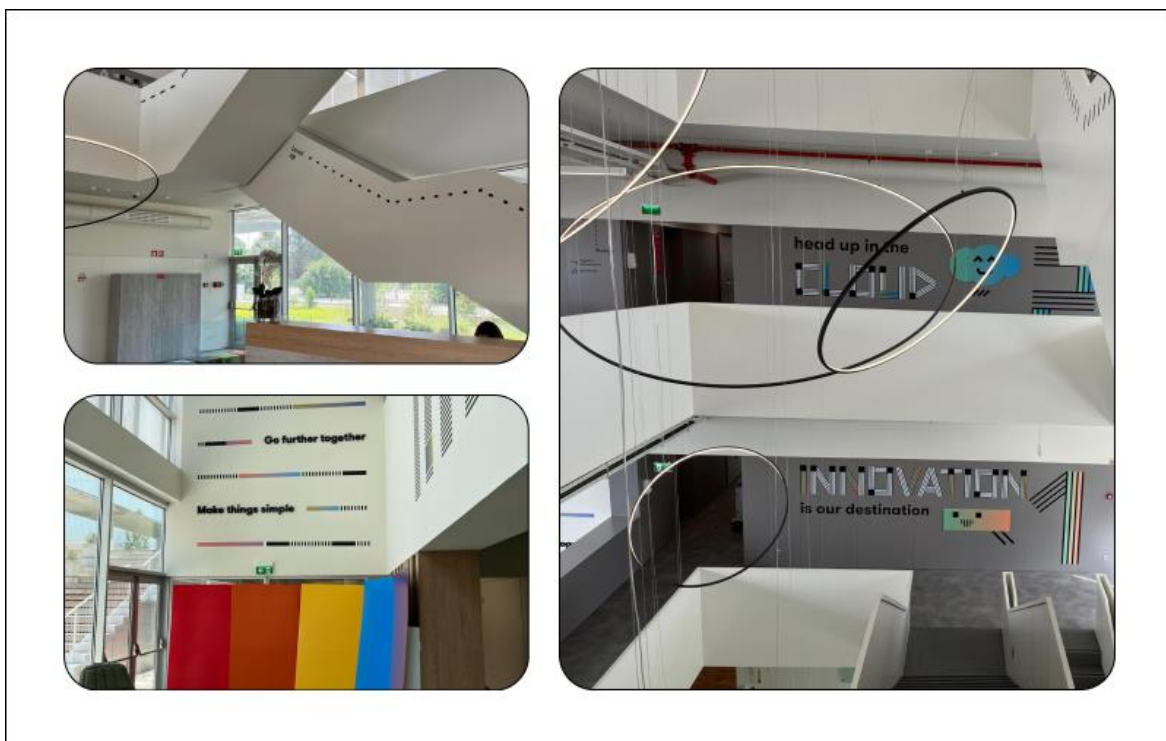
A MC é responsável pelo negócio de retalho alimentar da Sonae e líder de mercado em Portugal, com um grupo de segmentos de negócio distintos, apresentando mais de 35 anos de experiência. A Sonae MC é líder do mercado nacional com um conjunto de formatos de retalho distintos: no setor do retalho alimentar com o Continente (hipermercado urbano), Continente Modelo (supermercado) e Continente Bom dia (supermercados de conveniência), o Continente Online (plataforma de E-commerce) e Meu Super (lojas de proximidade em formato franchising); no setor da saúde, bem-estar e beleza com marcas como a Wells (saúde, bem-estar, beleza e ótica), Arenal (parafarmácia e perfumaria), Dr. Wells (medicina odontológica e estética) e Go Natural (supermercados e restaurantes orgânicos), em relação ao conjunto de negócios complementares de crescimento tem marcas como a Bagga (cafetarias), Note! (papeleria, livros e presentes), ZU (petcare e serviços veterinários), Washy (lavandarias self-service) e Home Story (móveis e acessórios para a casa).

A BIT é a área de sistemas de informação que atende os negócios de retalho alimentar e especializado da Sonae, com cerca de 40 mil colaboradores e um volume de negócios de cerca de 5 mil milhões de euros. Com soluções para as diferentes referências da cadeia de valor, para mercados tão variados como clínicas, parafarmácias, passando por aplicações de supermercado e de moda, a BIT compromete-se a desenvolver novas soluções de IT para responder aos desafios diários dos consumidores, mantendo-se na linha da frente correspondendo à forte concorrência existente no setor.

Na figura 1, mostra o organograma da empresa com a representação das diferentes insígnias do grupo, sendo a área em foco nesta dissertação a Sonae MC, em particular a BIT, sendo esta a área de sistemas de informação da MC, com interação com diferentes equipas e departamentos, sendo o presente apresentado na equipa de Segurança Operacional.



**Figura 1** - Organograma da empresa



**Figura 2** - Fotografias da entrada da BIT



## REVISÃO DA LITERATURA

### Leitura

#### **Atomic Web Design**

**Retirado em:** 17-04-2023, <https://bradfrost.com/blog/post/atomic-web-design/>

**Nome do artigo:** Brad Frost, *Atomic Design*, USA, 2013

**Palavras-chave:** design system, atomic design, atoms

**Contribuição para a investigação:** Este artigo apresenta um modelo de design system baseado em átomos, moléculas, organismos, templates e páginas, mais conhecido como Atomic Design. Esta abordagem permite que os designers e programadores criem interfaces de forma modular e mais escalável.

O autor conclui que com a abordagem Atomic Design, as equipas conseguem trabalhar de forma mais colaborativa e eficiente, criando componentes reutilizáveis e reduzindo a necessidade de recriar interfaces do zero.

O maior desafio é manter uma documentação clara e consistente para garantir a implementação do modelo.

## Fundamentals of Design Systems

**Retirado em:** 11-04-2023

**Nome do artigo:** Marouen Abdi, *Fundamentals of Design Systems*, Escola Superior de Tecnologia e Gestão, Leiria, Portugal, 2020

**Palavras-chave:** Design System, Design, Experiência de utilizador

**Contribuição para a investigação:** Este artigo apresenta uma visão geral dos conceitos básicos de um design system eficaz, explica que um design system consiste num conjunto de padrões, diretrizes e recursos compartilhados que as equipas de design e desenvolvimento usam para criar produtos e serviços de forma consistente e eficiente.

O autor conclui que com uma introdução clara e acessível aos conceitos e elementos básicos, um design system melhora a eficiência, a redução de erros de design e a entrega de uma experiência de utilizador mais consistente e coesa.

Os maiores desafios são:

- Manter a ferramenta escalável e adaptável, para evoluir com as constantes necessidades de novos projetos
- Conseguir manter a colaboração e comunicação entre as equipas de design e desenvolvimento para garantir a melhor implementação possível.

### **Bibliografia retirada da leitura:**

1. Brad Frost, *Atomic Design*, Pittsburgh, Pennsylvania, USA, 2016.
2. Diana MacDonald, *Practical UI Patterns for Design System Fast-Track Interaction Design for a Seamless User Experience*. Victoria, VIC, Australia, 2019.
3. Churchill Elizabeth, *Scaling UX with Design Systems*, ACM, NY, USA, 2019.

## Design System

### Definição

Design System é um sistema “vivo” que contém elementos, componentes, regras e princípios que orientam os diversos projetos de uma empresa.

Com este intuito, o Design System proporciona consistência e padronização a todos os produtos digitais, a ideia deste sistema é facilitar o trabalho e comunicação entre os designers e programadores, e da empresa criar a própria identidade que se irá refletir em todos os projetos. Sem esta ferramenta o trabalho repetitivo iria acontecer com mais frequência e era levado a inconsistências e prejuízo de tempo (Editorial Aela, 2021).

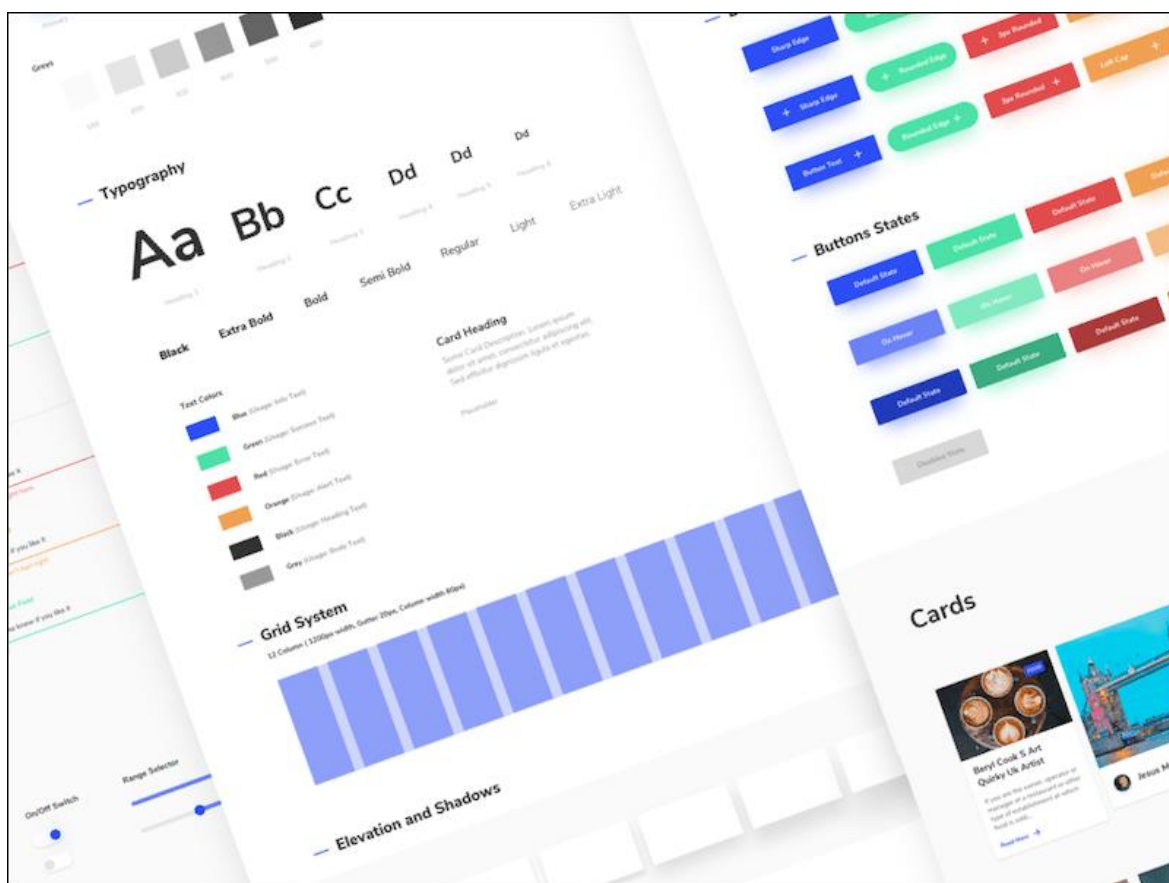


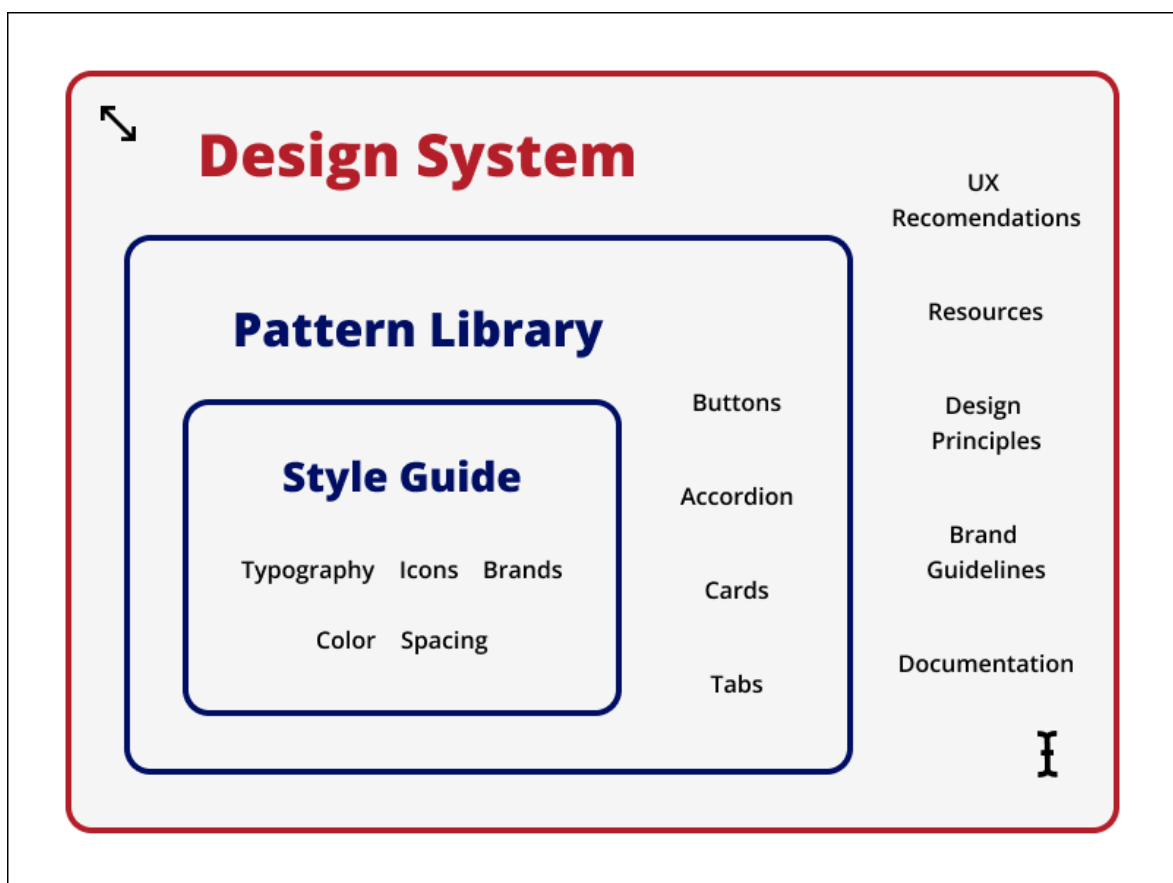
Figura 3 - Imagem ilustrativa de uma organização de um Design System.

## Qual a diferença entre Design System, Style Guide e Pattern Library?

Com a definição descrita de o que é um Design System, pode levar a crer que é uma ferramenta igual a um Style Guide ou um Pattern Library, mas tem diferenças em termos de objetivos e complexidades.

O Style Guide concentra-se mais no estilo gráfico (cores, fontes, ilustrações, entre outros) e o Pattern Library é um repositório de componentes, o Design System é muito mais que isso, é uma ferramenta que une as duas ferramentas anteriores, que nunca fica completa, está em constante evolução de acordo com a complexidade e amadurecimentos dos projetos da empresa (Cunningham, 2020). Para além da união, também adiciona mais conteúdo como por exemplo a documentação de todos os componentes e marcas, recomendações de *UX/UI* em todos os componentes desenvolvidos, aonde se deve e não se deve serem utilizados e os princípios de Design utilizados.

A utilização do Design System permite a aplicação dos diferentes elementos de UI de forma coerente e consistente, em conformidade com as ideias da marca (Fanguy, 2019).



**Figura 4** - Imagem ilustrativa das diferenças entre Design System, Pattern Library e Style Guide.

## **Vantagens**

O desenvolvimento de um sistema tão complexo como um Design System pode ser bastante trabalhoso e demorado, mas como resultado, a ferramenta poupa tempo e trabalho dos utilizadores para projetos futuros. Existem outros benefícios como: (IPNET Growth Partner, 2021)

- Manutenção da identidade da marca, com a criação de regras visuais e definição de padrões de comunicação visual;
- Consistência, uma vez que todos os elementos de UI estão padronizados;
- Melhor qualidade de interface, levado pela padronização, os erros tornam-se escassos e a experiência de utilizador melhora em todas as aplicações;
- Melhor comunicação, tanto na equipa de designers como entre os designers e programadores, pois como os elementos base já estão oficializados, existe uma amenização entre as discussões;
- Maior foco em UX, uma vez que o trabalho de UI já está uniformizado, os designers podem gastar mais tempo em somente melhorar a experiência do utilizador final (EDIT, 2018);
- Reutilização de componentes, devido ao desenvolvimento modular (Atomic Design), em que consiste na criação de um componente que pode ser reutilizado em mais que uma plataforma / projeto (Bhawalkar, 2022).

## **Princípios**

Como todos os projetos funcionais, o Design System necessita de algumas regras e tem características básicas que orientam um bom funcionamento de um sistema de design.

Uma delas é que o Design System é um produto digital, como qualquer outro projeto da empresa, necessita de manutenção e atualização constante, é fundamental a melhoria contínua para manter a sua eficiência.

A colaboração também é essencial para manter o Design System atualizado, portanto a comunicação entre os desenvolvedores e designers tem de ser constante para o sistema atender a todas as necessidades e até encontrarem e resolverem possíveis erros da ferramenta.

Outra característica que é importante é a simplicidade ao documentar e organizar, para que as informações sejam claras e fáceis de encontrar. A navegação deve ser simples e as respostas e soluções devem ser facilmente encontradas.

Para finalizar, o Design System tem de ser centralizado, ele é único na empresa, não deve haver mais do que um disponível para os projetos e equipas, portanto, é importante alocar a ferramenta num local de fácil acesso e que todos os utilizadores consigam utilizar, como por exemplo no Figma e no Zeroheight, além disso, as informações de utilização devem estar escritas de modo a todos os leitores não necessitarem de experiência no tema para entender (Editorial Aela, 2021).

## **Elementos base**

Existem alguns elementos base para qualquer Design System funcionar corretamente: (Editorial Aela, 2021)

- **Objetivos e valores**

Os objetivos são importantes para que todos os utilizadores do sistema estejam familiarizados com o propósito da empresa, no entanto estes podem ser alterados ao longo do desenvolvimento da ferramenta levando à necessidade de uma revisão regular deste elemento.

Da mesma forma, os valores refletem as crenças da organização no momento da tomada de decisão, por isso é fundamental que o Design System também traduza esses valores nas ideias da empresa.

- **Regras e Princípios**

Na criação de interfaces existem regras e princípios que guiam os designers a tomar decisões importantes para os seus devidos projetos, portanto é essencial a descrição dessas regras no Design System, como por exemplo as regras de UX (Lei de Hick e Lei de Miller) e as heurísticas de Nielsen.

- **Identidade da Marca**

Como dito anteriormente, o Design System é responsável por ajudar a desenvolver e orientar os utilizadores no desenvolvimento das interfaces gráficas de forma mais rápida e com consistência, mantendo a identidade da empresa. Isto é feito documentando de forma clara todos os elementos visuais da marca como por exemplo: as cores, os logotipos, a tipografia, imagens e as ilustrações.

- Componentes

Os componentes são das peças mais importantes de um Design System, são eles que definem e padronizam as interfaces. Praticamente são a base de todos os produtos digitais da empresa. Alguns exemplos de componentes são botões, *accordion*, *bottom sheet* e *inputs*. Têm de estar bem documentados com tamanho, tipos de uso e anatomia para que o utilizador consiga perceber quando e onde usar cada elemento.



## **Exemplos**

Muitas empresas atualmente já utilizam um Design System para seguir as boas práticas e ajudar os utilizadores a produzir cada vez melhores produtos digitais. Organizações como a IBM, Google, Shopify, Primavera e Apple são alguns dos exemplos.

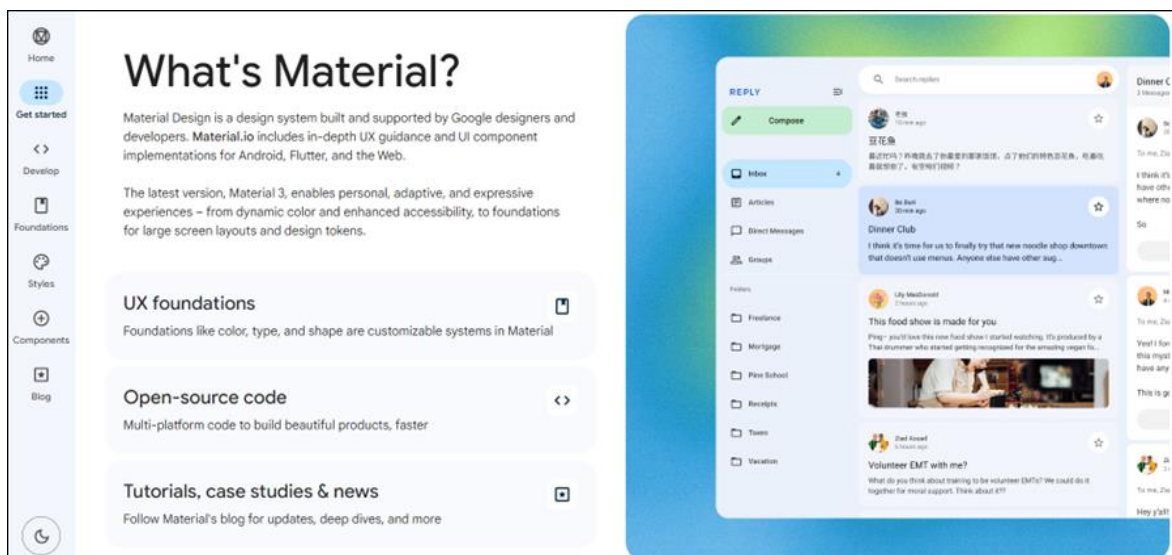
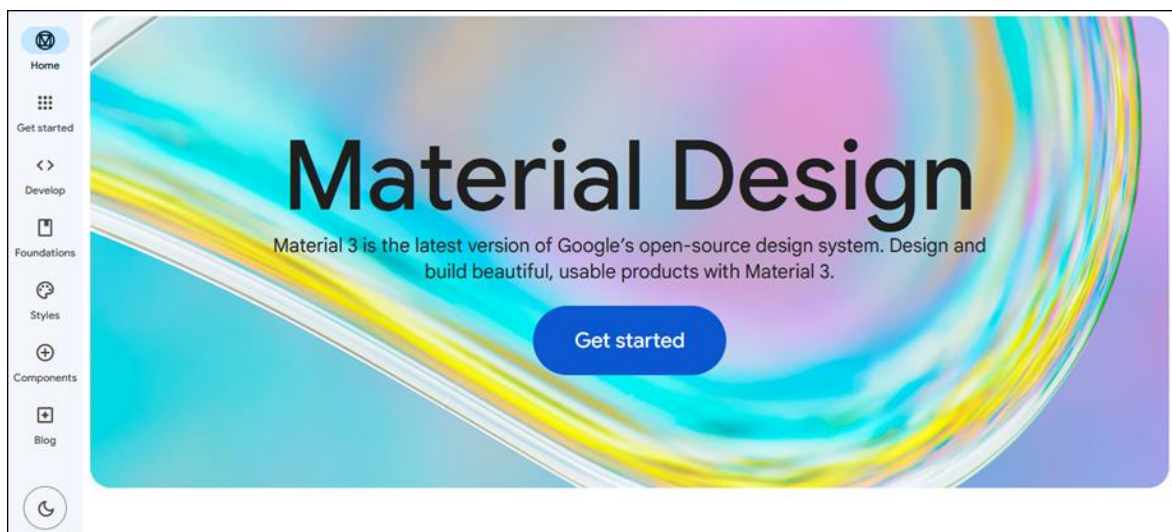
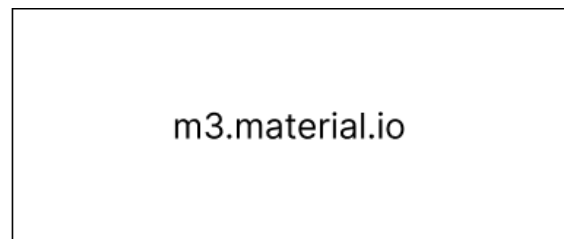
Ao realizar componentes de raiz, foi feita uma pesquisa sobre o funcionamento e utilização de cada componente em diversos Design Systems em que os principais em que foi realizada a pesquisa, pela sua complexidade e documentação detalhada, foram:

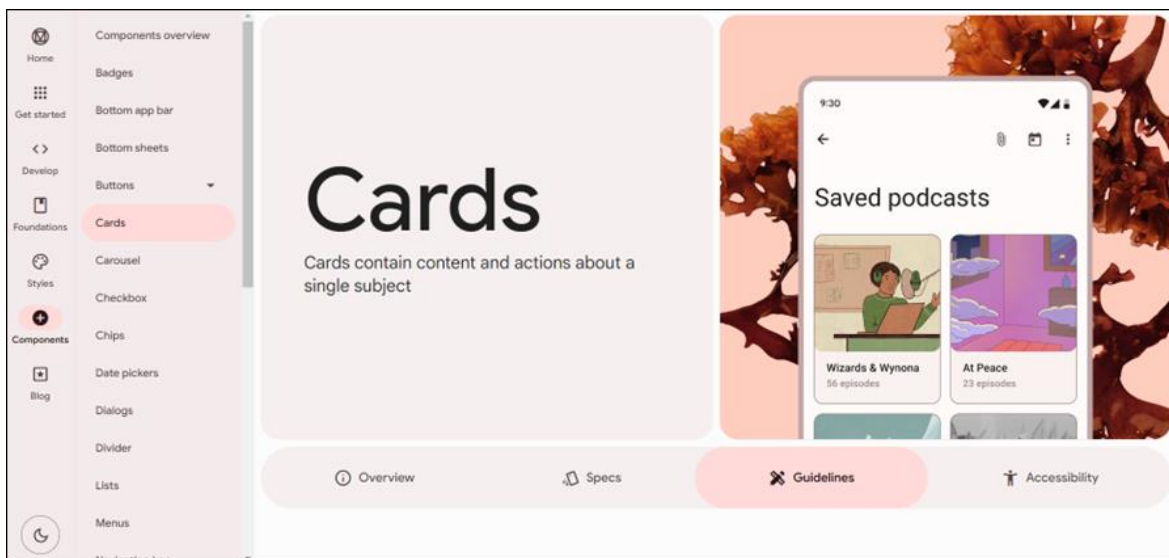
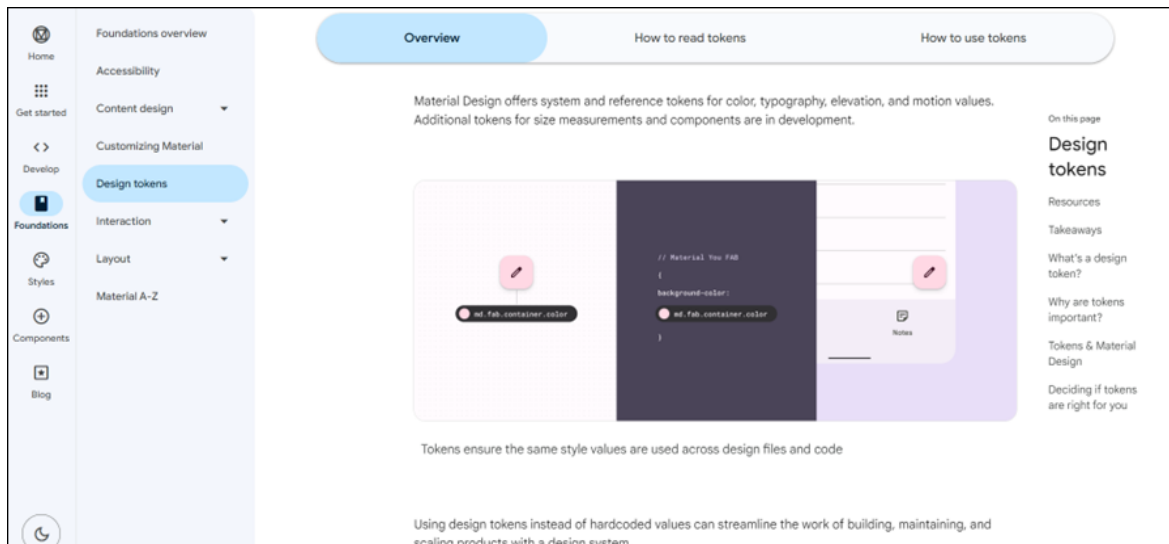
- Material Design
- Carbon Design System
- Decathlon Design System
- Primavera Design System

## Material Design

O Material Design é um Design System mais focado na vertente visual, mantendo uma linguagem simples e dinâmica, contendo a documentação mais completa de todos os exemplos. É um dos Design System mais antigos, lançado em 2014, e foi a principal inspiração para diversos Design Systems.

Interessante destacar que diversas páginas têm diferentes paletes de cores para exemplificar a variedade e consistência da ferramenta nas diversas situações (Dribbble, 2022).

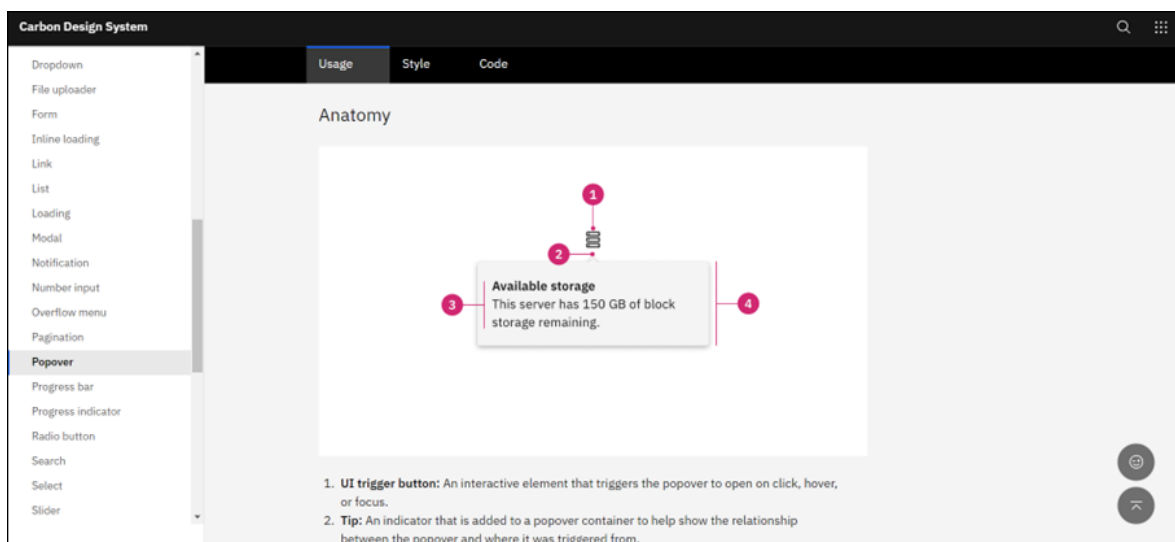
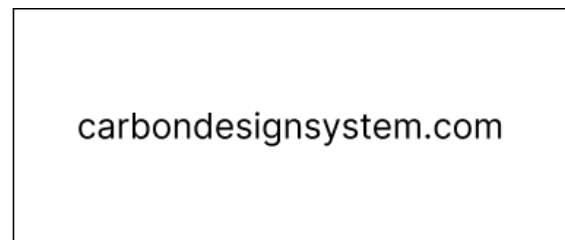


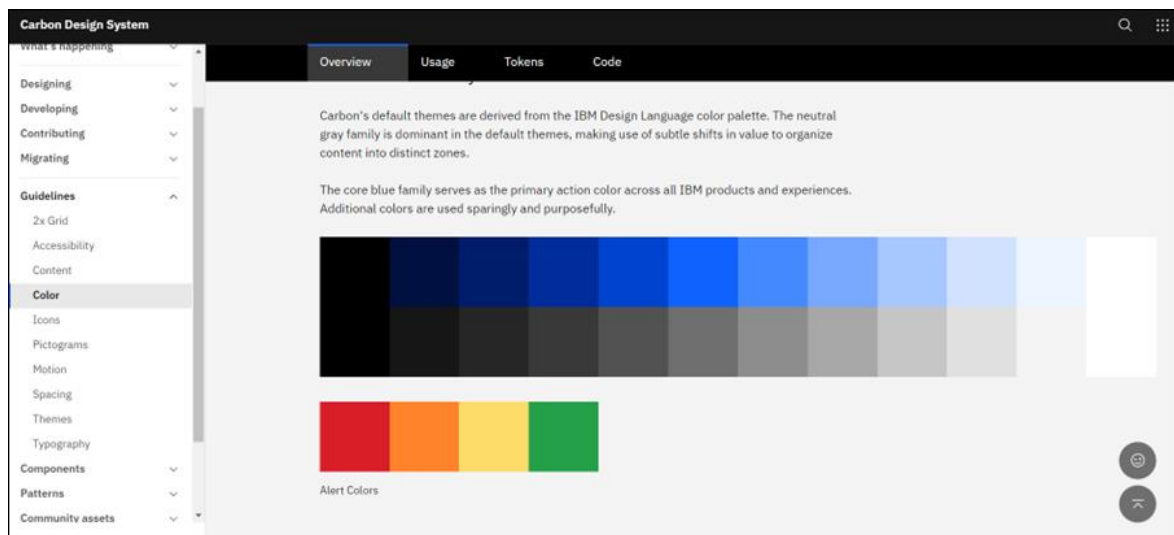
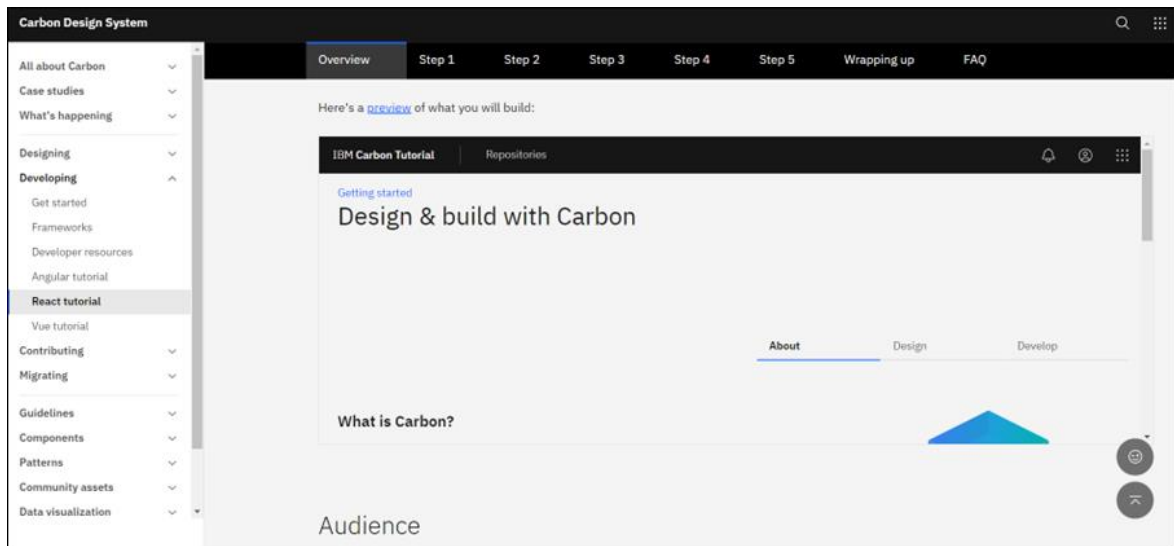


**Figura 5** - Capturas de ecrã tirados do site Material Design (<https://m3.material.io/>)

## Carbon Design System

Carbon Design System tem uma abordagem muito mais técnica e objetiva, ideal para um utilizador com conhecimento técnico. Contém uma documentação rica em código e usabilidade. Sendo um design system *open-source*, contém diversos contribuidores, tanto internos da equipa da IBM como externos. O sistema central do Carbon inclui cores, *assets*, código, diretrizes e outros elementos projetados para servir de base para uma variedade de aplicações tanto web como mobile (Dribbble, 2022).

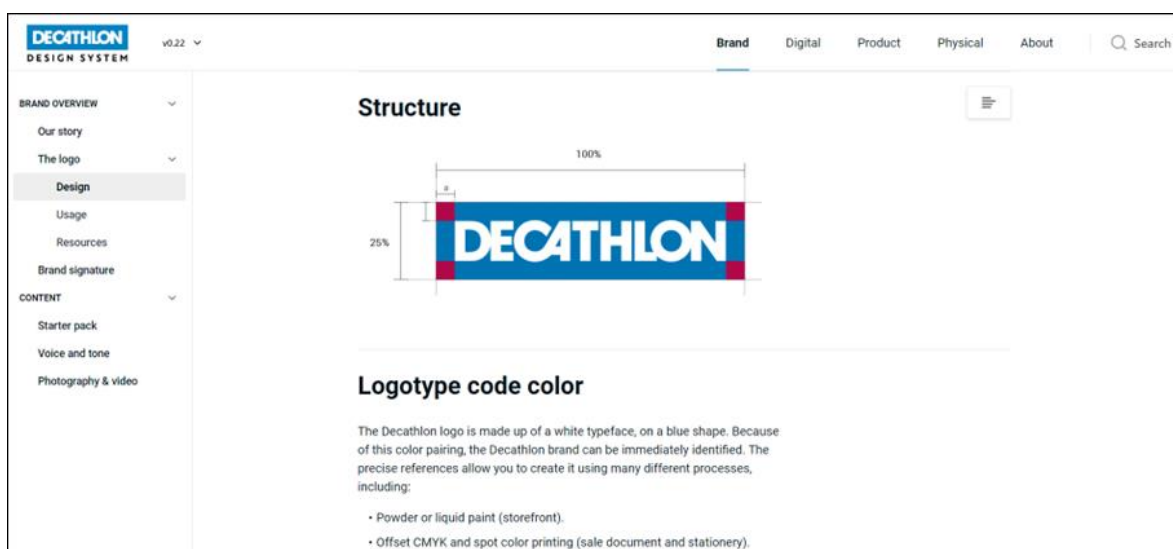
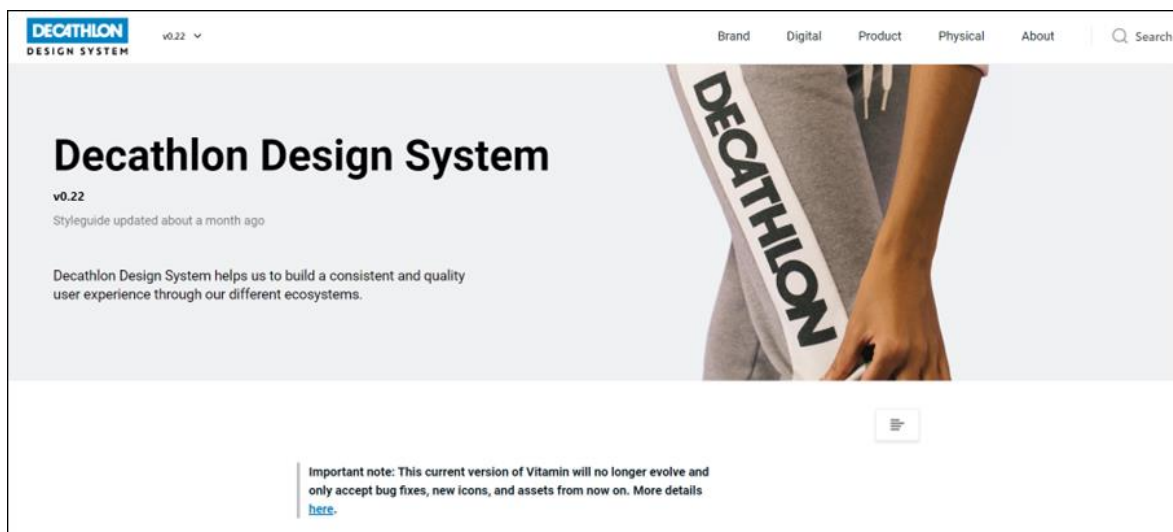


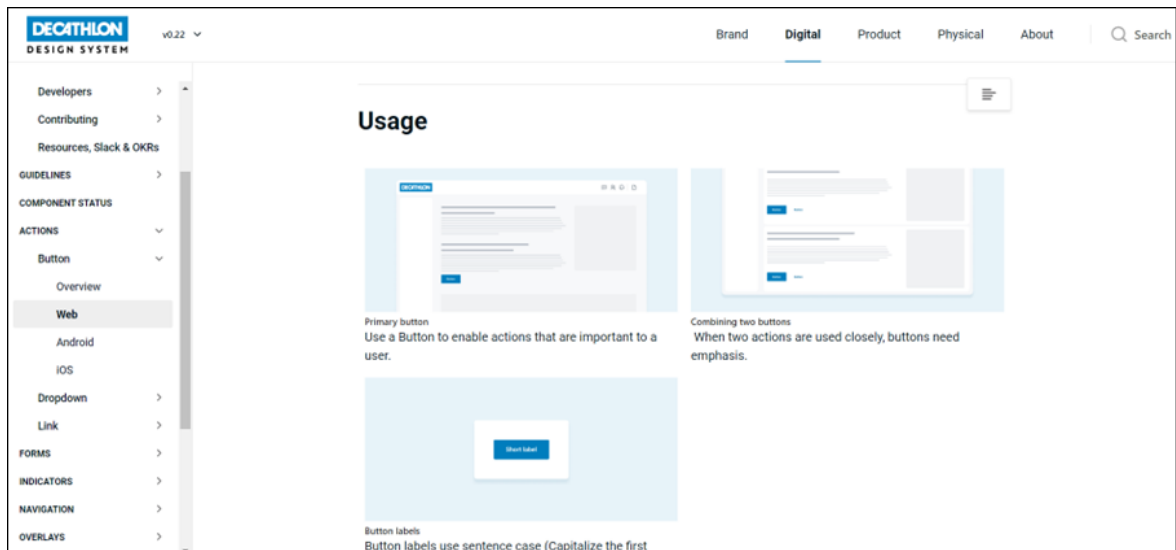


**Figura 6** - Capturas de ecrã tirados do site Carbon Design System (<https://carbondesignsystem.com/>)

## Decathlon Design System

Decathlon Design System foi a principal inspiração neste projeto pelo facto de utilizar a mesma ferramenta de documentação, o Zeroheight, e pela mistura de linguagem técnica e linguagem mais visual, contendo diversas animações e ilustrações.

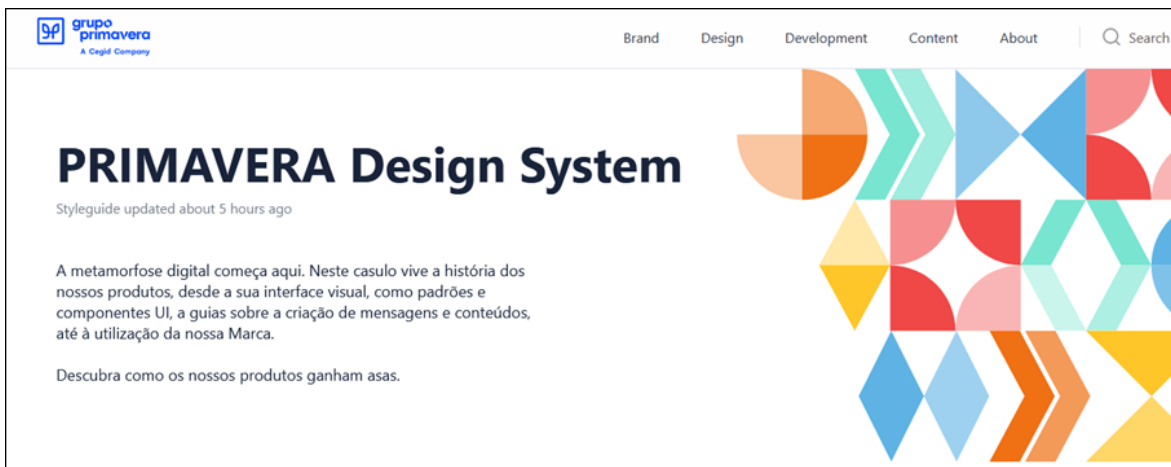
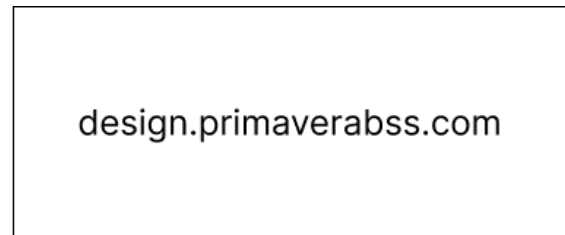




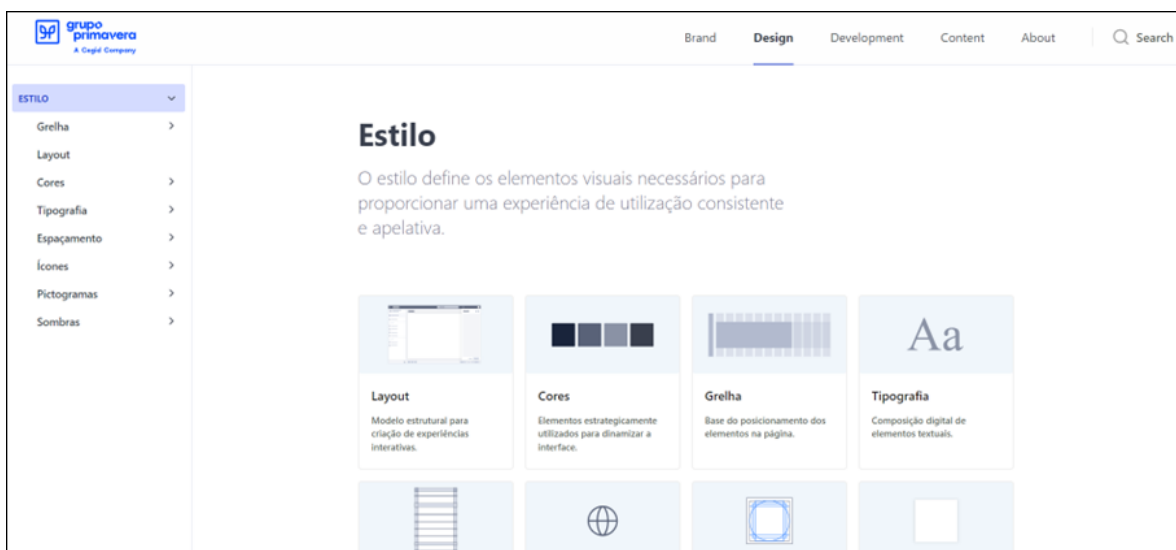
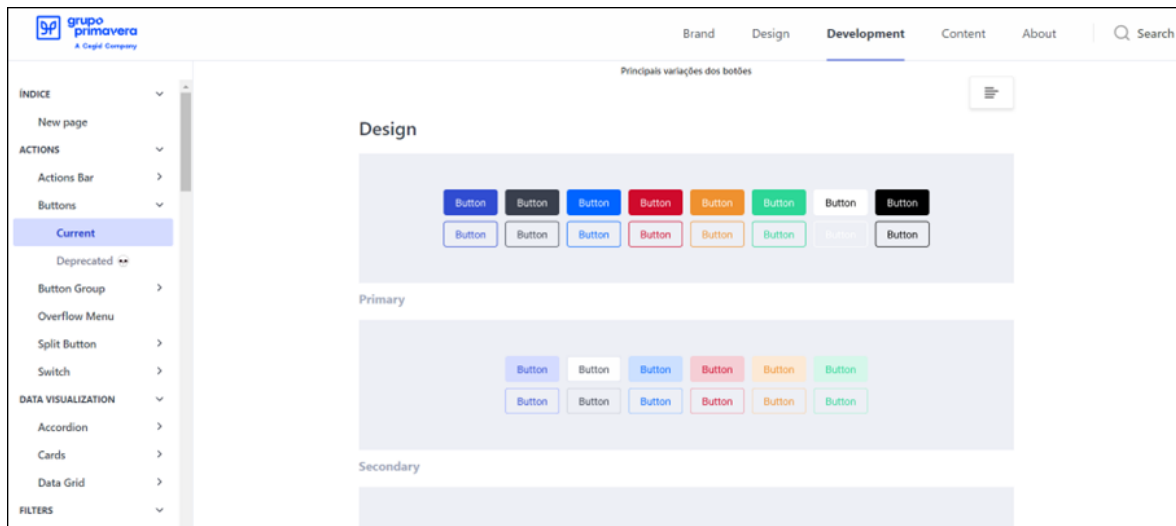
**Figura 7** - Capturas de ecrã tirados do site Decathlon Design System  
(<https://www.decathlon.design/726f8c765/v/0/p/71b8e3-decathlon-design-system>)

## Primavera Design System

Por fim, Primavera Design System, o primeiro caso de investigação, porém sendo o menos utilizado como inspiração, trabalha também com a ferramenta Zeroheight e tem uma linguagem bem técnica, como o Carbon Design System.







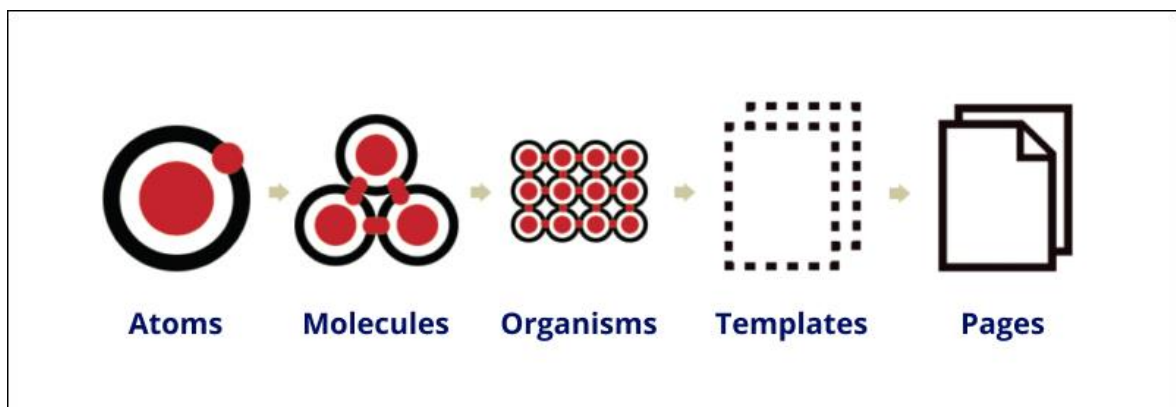
**Figura 8** - Capturas de ecrã tirados do site Primavera Design System (<https://design.primaverabss.com/>)

Após a análise dos diversos Design Systems mencionados em cima, pode-se concluir:

- O melhor em termos gráficos, como por exemplo *heropages*, exposição da informação e diversas imagens e simples e bastante explicativas é o Material Design da Google;
- O melhor em explicar todos os componentes em termos mais técnicos é o Carbon Design System da IBM, pela sua complexidade e ter um modo “Demo” em que consegue demonstrar o componente visualmente e mostra todo o código necessário para obter o mesmo resultado;
- O Primavera Design System e o Decathlon Design System foram essenciais para verificar e saber todas as possibilidades de como utilizar o Zeroheight, ferramenta de documentação, da melhor forma possível.

## Atomic Design

Atomic Design é uma metodologia de design focada na criação de interfaces e design systems coesos e escaláveis que podem ser facilmente mantidos e atualizados ao longo do tempo. Esta metodologia baseia-se na ideia de que um projeto deve ser construído camada por camada, começando pelos elementos mais básicos e progredindo para os componentes mais complexos.



**Figura 9** - Modelo de Atomic Design

O Atomic Design visa criar designs consistentes e flexíveis que possam ser facilmente adaptados a diferentes dispositivos e contextos. Para fazer isso, a metodologia baseia-se na criação de cinco níveis de elementos de design: átomos, moléculas, organismos, modelos e páginas. Assim como a organização dos elementos na biologia estruturam-se de forma hierárquica, do mais pequeno para o maior e cada um desses níveis é constituído por unidades presentes no nível anterior. (Cunningham, 2020)

Átomo é o bloco de construção mais básico como por exemplo botões, campos de formulário, etc. Moléculas são combinações de átomos que formam componentes mais complexos, como formulários e *cards*. Organismos são grupos de moléculas que formam uma secção da página, como *headers* e o *template* são objetos no nível de página, onde se coloca componentes em um layout formando a estrutura da página.

Seguindo a metodologia do Atomic Design, os designers podem criar designs modulares e escaláveis. Isso significa que ele pode ser facilmente atualizado e ajustado ao longo do tempo, sem a necessidade de reconstruir toda a interface do utilizador do zero. Isso mantém os designs consistentes e relevantes num ambiente de constante mudança.

Existem vários pontos positivos sobre o Atomic Design:

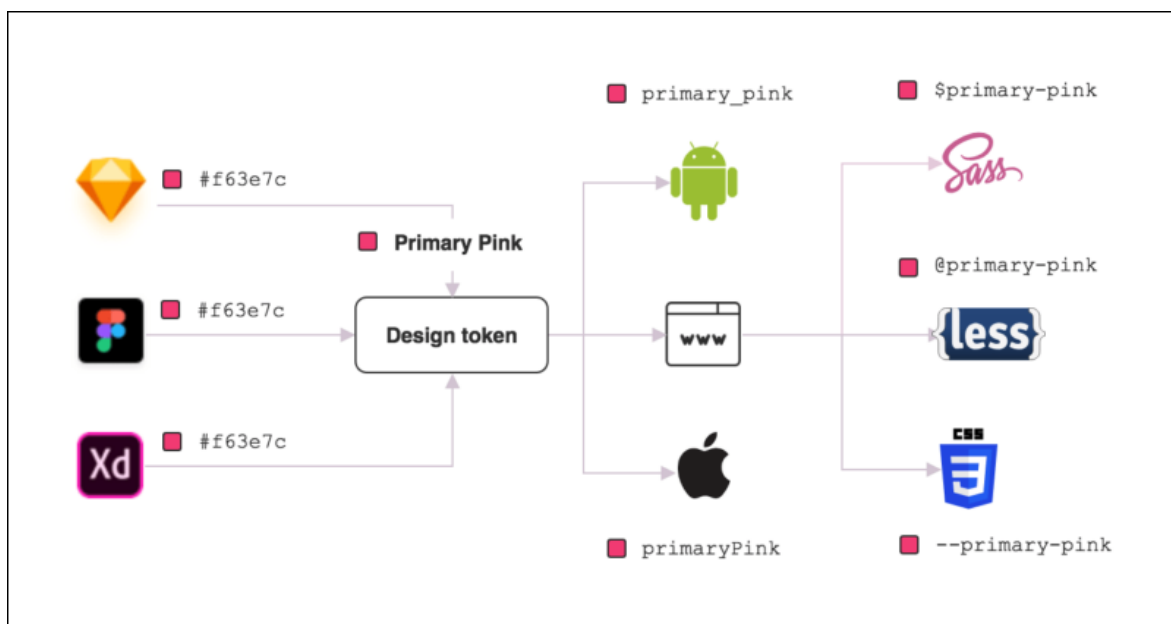
- **Escalabilidade:** Uma das principais vantagens do Atomic Design é a escalabilidade. Ao criar um design em camadas e modular, pode facilmente reutilizar elementos em diferentes partes de um site ou aplicação. Isso economiza tempo e esforço, especialmente para atualizações e alterações de design.
- **Consistência:** Outra vantagem importante é a consistência. Seguindo esta metodologia, pode-se criar uma biblioteca de componentes de design reutilizáveis para manter a interface do utilizador consistente num site ou aplicação. Isso melhora a experiência do utilizador e cria uma sensação de familiaridade e confiança.
- **Flexibilidade:** O Atomic Design também é flexível, permitindo que os designers adaptem facilmente a sua interface em diferentes tamanhos de tela e dispositivos. Ao criar um design modular, os elementos podem ser reorganizados para atender às necessidades específicas de cada dispositivo ou contexto.
- **Eficiência:** O Atomic Design permite que os designers criem ecrãs rapidamente. Cada elemento é organizado em camadas, permitindo que pequenas alterações e atualizações sejam feitas sem afetar todo o design system. Isso economiza tempo e esforço, permitindo que os designers respondam rapidamente a novos requisitos e necessidades.
- **Padronização:** Finalmente, o Atomic Design ajuda a padronizar os designs numa organização. Ao criar uma biblioteca de componentes de design reutilizáveis, pode-se garantir que toda a equipa siga as mesmas diretrizes e padrões de design. Isso ajuda a manter a consistência e garantir que a marca seja reconhecida e lembrada de forma consistente em todas as interfaces.

## Design Tokens

Design tokens são um método de design que ajuda a criar e manter a consistência visual nas interfaces do utilizador. Trata-se de um conjunto de variáveis predefinidas, como cor, tipografia, espaçamento, dimensões etc., usadas em todo o design system.

Esses tokens ajudam a garantir que todos os elementos da interface do utilizador fiquem com aparência consistente e coesa, independentemente de quem os projetou ou implementou.

Os design tokens são especialmente úteis em grandes projetos em que várias equipas trabalham em diferentes partes do projeto. Garante que todos sigam as mesmas diretrizes e padrões de design para garantir que as interfaces tenham uma aparência consistente em todos os sistemas e projetos. Além disso, os design tokens ajudam a simplificar o processo de design e desenvolvimento. Ao usar tokens em vez de valores fixos, pode-se realizar atualizações no design system de forma rápida e fácil. Por exemplo, se a cor primária da marca mudar, basta atualizar o valor do token e todos os elementos que usam essa cor serão atualizados automaticamente, isso é representado na figura 10.



**Figura 10** - Representação da aplicação dos design tokens. Imagem retirada do website oficial do Zeroheight (<https://zeroheight.com/>)

Os tokens também podem ser usados para criar diferentes temas e variações. Por exemplo, pode-se criar diferentes esquemas de cores para uma aplicação, e esses esquemas podem ser facilmente alterados simplesmente atualizando os valores de tokens.

Outros benefícios dos design tokens são:

- **Eficiência:** Os design tokens podem ser facilmente atualizados, ajudando a acelerar o processo de design e desenvolvimento. Isso economiza tempo e recursos, permitindo que as equipas se concentrem em outras áreas importantes do projeto.
- **Consistência:** Os design tokens garantem que a interface fique consistente entre todos os sistemas, independentemente de quem a projetou ou implementou. Isso permite uma experiência de utilizador consistente e confiável.
- **Reutilização:** Os design tokens são reutilizáveis e podem ser usados em diferentes projetos e contextos. Isso ajuda a manter a marca consistente em toda a sua presença digital.
- **Escalabilidade:** Os design tokens são escaláveis e podem ser usados em projetos de tamanho e complexidade variados. Isso o torna ideal para projetos grandes e complexos, onde a consistência visual é especialmente importante.

## Ferramentas utilizadas

O Design System da MC, o Mantra, tem uma vertente mais operacional como se fosse um *backoffice*, onde desenvolvem todos os componentes e *templates* gráficos, e onde está centralizada a biblioteca *master* do Mantra. É nestas plataformas operacionais que a equipa desenvolve todos os projetos, os quais liga diretamente aos elementos da *master library* do DS. O que significa que qualquer alteração na *master library* tem impacto automático nos projetos.

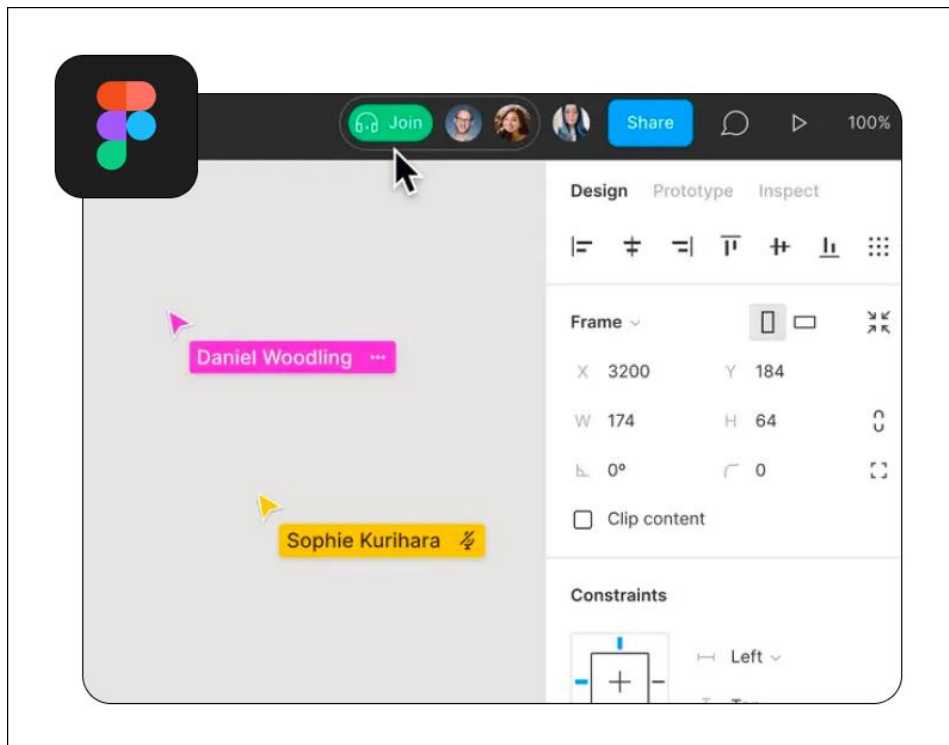
Nesta seção serão enumeradas e descritas todas as ferramentas utilizadas para a solução deste trabalho. Foram usados três softwares diferentes, para o design, para a programação e para a documentação do Design System.

### **Figma**

O software utilizado para desenvolver os projetos em termos gráficos é o Figma, uma plataforma colaborativa para construção de design de interfaces e protótipos, em que trabalha com vetores para construir os elementos visuais.

O Figma tem como objetivo criar uma ferramenta que permite às equipas trabalharem o mesmo projeto em conjunto mantendo a acessibilidade do sistema, permitindo aos designers o desenvolvimento de soluções ao nível do design de produtos digitais (sites, aplicações mobile, aplicações de *smartwatch*, entre outros).

Esta ferramenta foi escolhida pelos recursos que tornam a experiência imersiva e ágil, pois tem a capacidade implementar componentes que podem ser usados em todos os projetos. Além disso, o Figma permite que os designers convidem novos utilizadores para interagir ou visualizar o projeto, melhorando a comunicação entre os designers e programadores. (Brett, 2022)

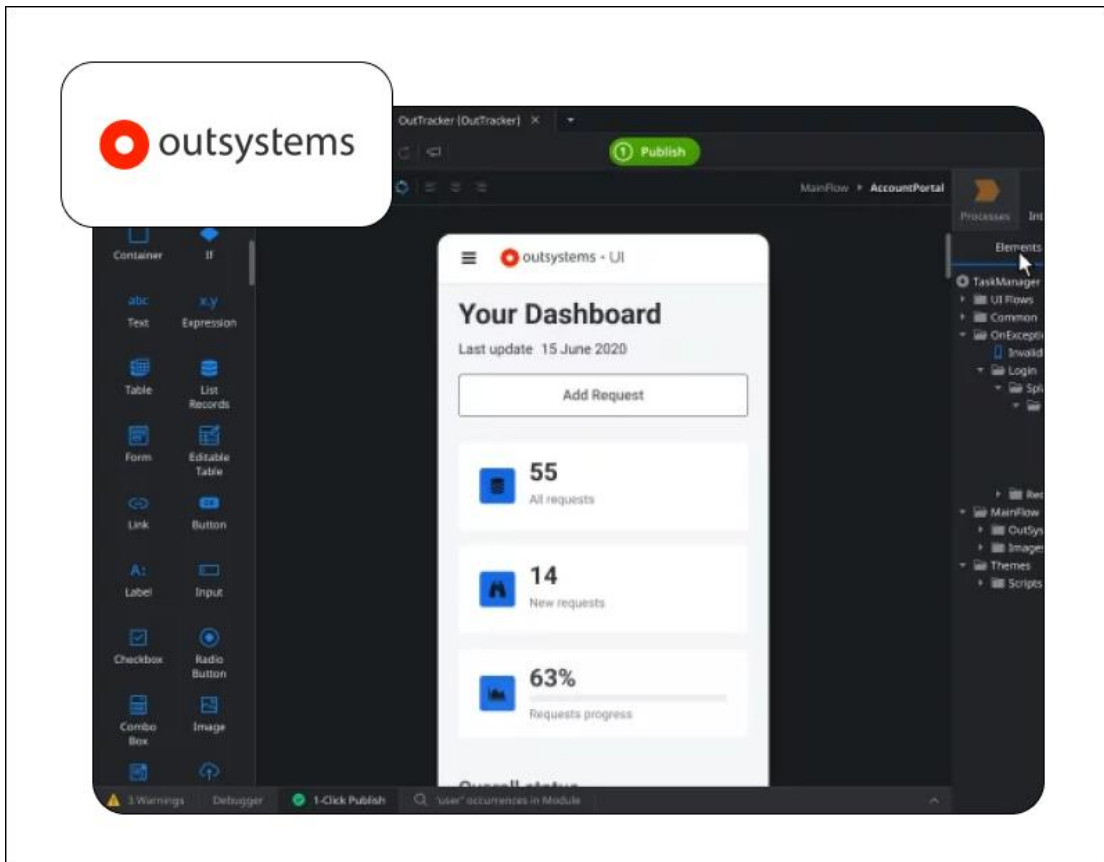


**Figura 11** - Logotipo do Figma e o seu software com a funcionalidade de vários designers e programadores estarem no mesmo ficheiro

O recurso mais usado do Figma para a construção do Mantra é o recurso de criação de componentes. Ele permite que se crie um elemento, por exemplo um botão, e esse mesmo elemento pode ser multiplicado para todos os projetos de uma forma dinâmica. Com isso, o Mantra, usa esse recurso para criar uma biblioteca de componentes, em que cada um tem as suas propriedades de personalização e edição, permitindo que quando essa biblioteca é alterada, todos os projetos ligados a ela mudam também, dando uma dinâmica mais fluída para a uniformização dos projetos da MC. (Villain, 2022)

## OutSystems

A vertente de código foi realizada em OutSystems, por ser a plataforma mais utilizada na MC para projetos *customer* e *corporate*. O OutSystems é uma plataforma *low-code* para desenvolver aplicações, integrar sistemas e adicionar o próprio código quando necessário.



**Figura 12** - Logotipo do Outsystems e o seu software



O OutSystem facilita aos desenvolvedores o acesso a uma biblioteca de componentes visuais criada de raiz. É uma ferramenta construída em linguagem .NET, mas os utilizadores não precisam ter experiência com a linguagem, pois o OutSystems permite que o desenvolvimento ocorra de forma mais visual, eliminando o obstáculo relacionado com a curva de aprendizagem necessária para aprender uma linguagem nova.

A utilização de um software visual e mais intuitivo torna possível agilizar o processo e eliminar pequenos erros comuns na programação, acelerando o tempo de desenvolvimento de uma aplicação quando usado corretamente, além disso também tem uma biblioteca de componentes modulares, que permite criar aplicações de forma eficiente e rápida. No entanto também tem desvantagens, por ser uma ferramenta *low-code* existe menos flexibilidade que uma ferramenta comum de programação. Para melhorar recursos e as bibliotecas levam mais tempo e trabalho, e isso é um dos maiores desafios deste projeto, conseguir fazer que o OutSystems siga as regras e componentes visuais do Figma. Para isso, tem de ser criada uma aplicação que muda totalmente a biblioteca de componentes do OutSystems, sendo necessário primeiro aprender como cada componente é feito no software, e depois desenvolver um novo que o substitui. Essa aplicação é ligada em todos os projetos e segue o mesmo método dos componentes do Figma, sempre que essa aplicação é alterada, todos os projetos têm impacto. (Valuga)

## Zeroheight

Sendo o Mantra um produto para outros produtos, é o propósito da organização que este seja usado também por outras equipas, incluindo *product owners* e *product managers*, e para dar essa visibilidade do *Design System*, a plataforma utilizada para a documentação é o Zeroheight.

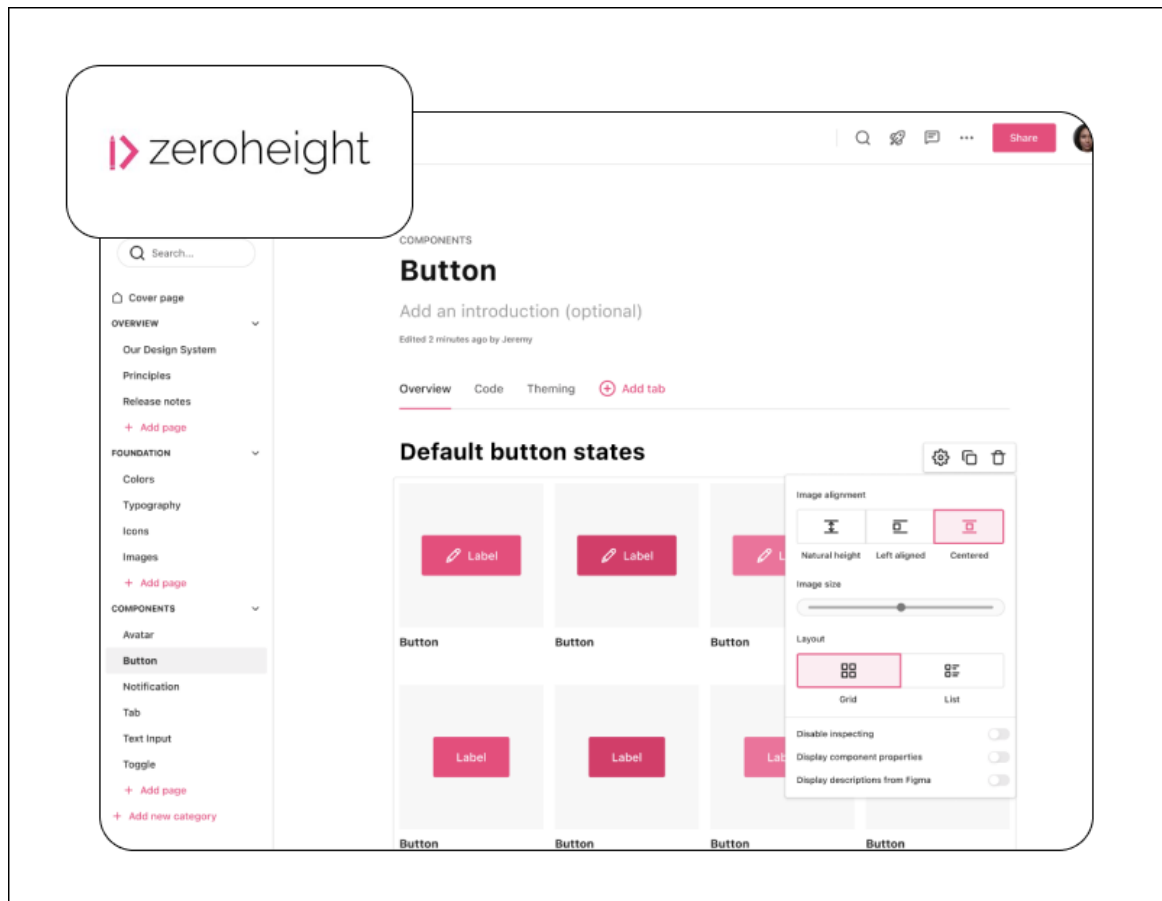


Figura 13 - Logotipo do Zeroheight e o seu software

O Zeroheight é uma ferramenta online que permite a documentação de componentes de design com interação direta com o Figma, que agiliza o trabalho dos designers na altura de documentar.

As vantagens de usar o Zeroheight é a integração com todas as ferramentas de design e programação, a melhor colaboração em implementar conteúdo no software, por causa da sua funcionalidade de conseguir colocar todo o tipo de documentação em um só lugar e a facilidade de usar a ferramenta. (Brett, 2022)

Com este software, toda a empresa MC tem visibilidade do trabalho feito pelos designers com as regras e componentes visuais como dos desenvolvedores com o código usado para fazer esses componentes do Design System. Esta ferramenta é essencial para mostrar como o Mantra é complexo e como deve ser utilizado, para os desenvolvedores conseguirem desenvolver os seus projetos seguindo corretamente as regras de Design. (Zeroheight, 2022)

## MÉTODO

### Projeto – Definição e Objetivos

O projeto que será detalhado neste trabalho é o Mantra, o Design System da MC, que tem como objetivo ser a principal ferramenta de consistência e de UI/UX de todas as aplicações digitais e físicas da organização. Esta ferramenta é bastante complexa pela quantidade de conteúdo que abrange, como por exemplo, fundações de UI, cores e componentes tanto em Design tanto em CSS para programação.

Atualmente é usado principalmente como meio de documentação e usado como ferramenta de UI/UX tanto para a equipa de UX Designers e a equipa de RAD, em que usa a tecnologia Outsystems. Futuramente serão implementadas outras tecnologias como o React, Angular e .Net e também a nova tecnologia Design Tokens.



Figura 14 - Logotipo do Mantra e a sua página de documentação

## Requisitos

### Requisitos Funcionais

- Princípios de design e princípios da usabilidade;
- Todas as tipografias, cores, ícones e imagens, componentes, padrões, e regras de acordo com todas as brands da organização;
- Componentes responsivos;
- Regras de acessibilidade;
- Documentação de todas as secções do Design System;
- Guia e orientação de testes de usabilidade com utilizadores finais;

### Requisitos Não Funcionais

- Toda a página do Figma deve ser rápida e sem sobrecarregamento de informação;
- Bibliotecas de design e programação devidamente atualizadas;
- Documentação atualizada sempre que a biblioteca sofre alterações;

## Como se materializa

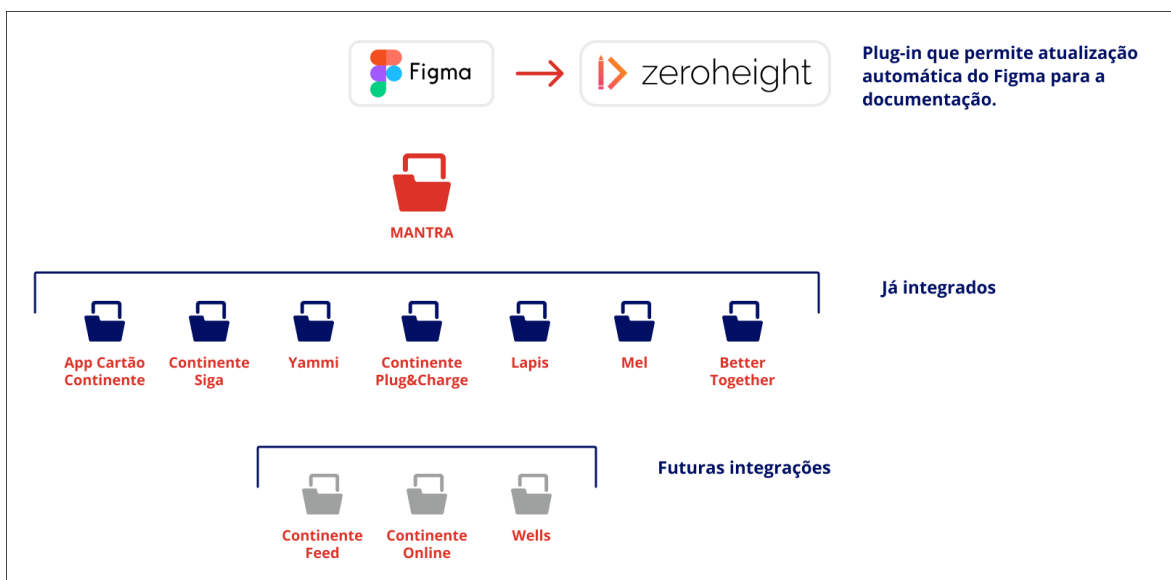
O Mantra tem três frentes principais: o design, a programação, e a documentação.

O Design é realizado na ferramenta Figma, em que a equipa tem uma pasta centralizada que é o Mantra, na qual estão centralizados todos os componentes e regras do Design System, que distribui todo o seu conteúdo pelas restantes pastas de projetos.

Com as regras definidas toda a documentação derivada do Design System é disponibilizada à equipa de programação que por sua vez cria uma biblioteca CSS que respeita todas as regras e configurações definidas pelos designers.

Devido à sua complexidade, as duas partes têm de ser devidamente documentadas, função que é da responsabilidade do software de documentação Zeroheight. Este software permite uma interligação com o Figma que documenta tudo o que está relacionado com o Design System e com a forma de utilização, seja ao nível do design ou ao nível da programação.

Na figura 15, mostra todas as aplicações já integradas no Mantra, ou seja, todas as aplicações que passaram a utilizar totalmente o Design System como sua base para todos os componentes e fluxos.



**Figura 15** - Esquema de Materialização do Mantra

## Organização da pasta Figma

A pasta Figma do Mantra Design System é o recurso principal que alimenta tanto a documentação do Zeroheight tanto como guia para a biblioteca do Outsystems. Dito isso, é muito importante a pasta estar organizada e com todas as informações possíveis de UI/UX.

O Mantra é organizado em 9 ficheiros principais e 2 *templates* para ser usados nos projetos. Os ficheiros principais são as *Foundations*, *Components*, *Templates*, *Patterns*, *Graphics*, *Icons*, *Helpers* e *Rules*, e os 2 *templates* é um *template Desktop* e um *template Mobile*.

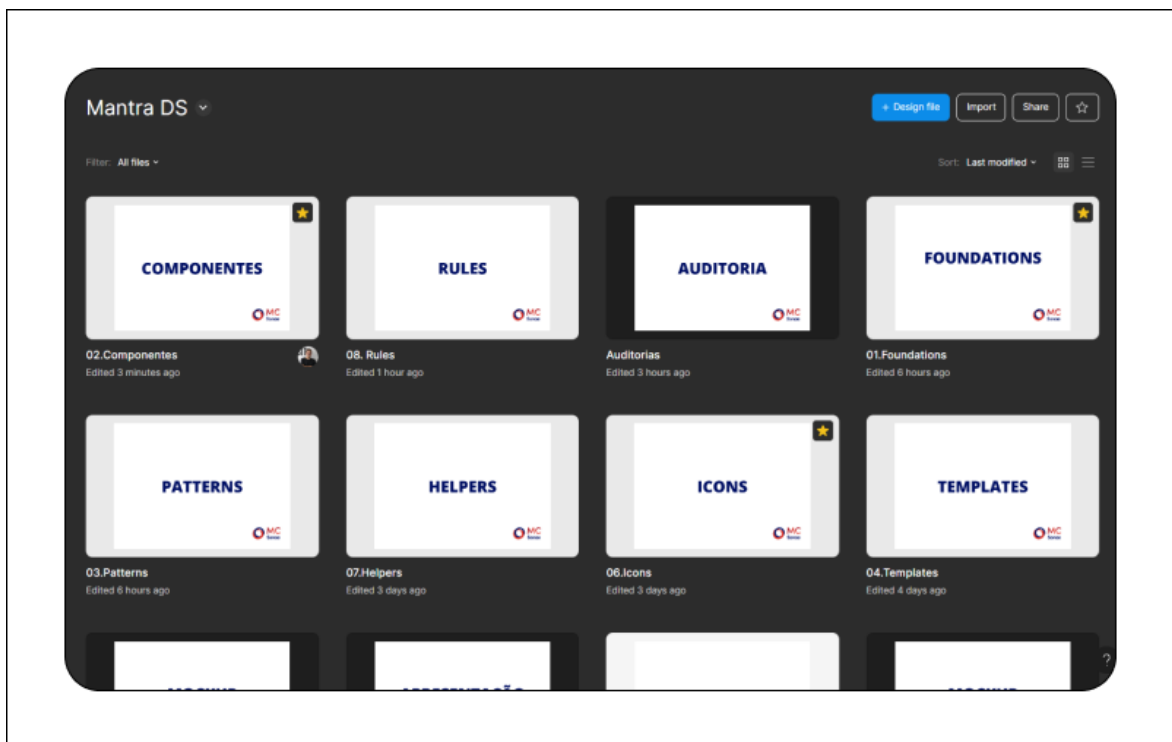


Figura 16 - Pasta do Figma do Mantra

### *Foundations*

O ficheiro *Foundations* onstitui as fundações do Mantra e contem todos os estilos que vão ser usados nos outros ficheiros. Compreende os elementos básicos e essenciais que fornecem a base para o *Design System* como um todo. Estes elementos são fundamentais para estabelecer princípios de *design* e linguagem visual em todos os ativos digitais.

As fundações usadas atualmente são:

- Logotipos e Favicons
  - Logotipos e Favicons de todas as *brands* e aplicações da MC, como por exemplo Continente, App CC, Wells, ZU entre outros
- Cores
  - Cores Neutras, Semânticas e Cores de todas as *brands*
- Tipografia
  - Open Sans, a tipografia oficial da MC para o uso nos ativos digitais
- Espaçamento
  - Espaçamento sempre em múltiplos de 8px (8, 16, 24, 32, entre outros)
- *Grids*
  - *Grids* de *Mobile*, *Tablet* e *Desktop*
- Sombras
  - Todas os tipos de sombras e levantamentos usadas em UI





### *Components*

É no ficheiro Components que se armazenam todos os componentes do Mantra, como por exemplo Botões, *Accordions*, *Link*, Tabelas, entre muitos outros.

Os componentes são blocos de construção reutilizáveis que representam elementos de interface do utilizador, são criados para manter a consistência visual e funcional de todo o sistema e ativos digitais. Foram projetados para serem flexíveis para serem usados em todas as necessidades e casos de uso, e são construídos tendo como base as *Foundations*, com as suas cores, tamanhos e restantes características definidas.

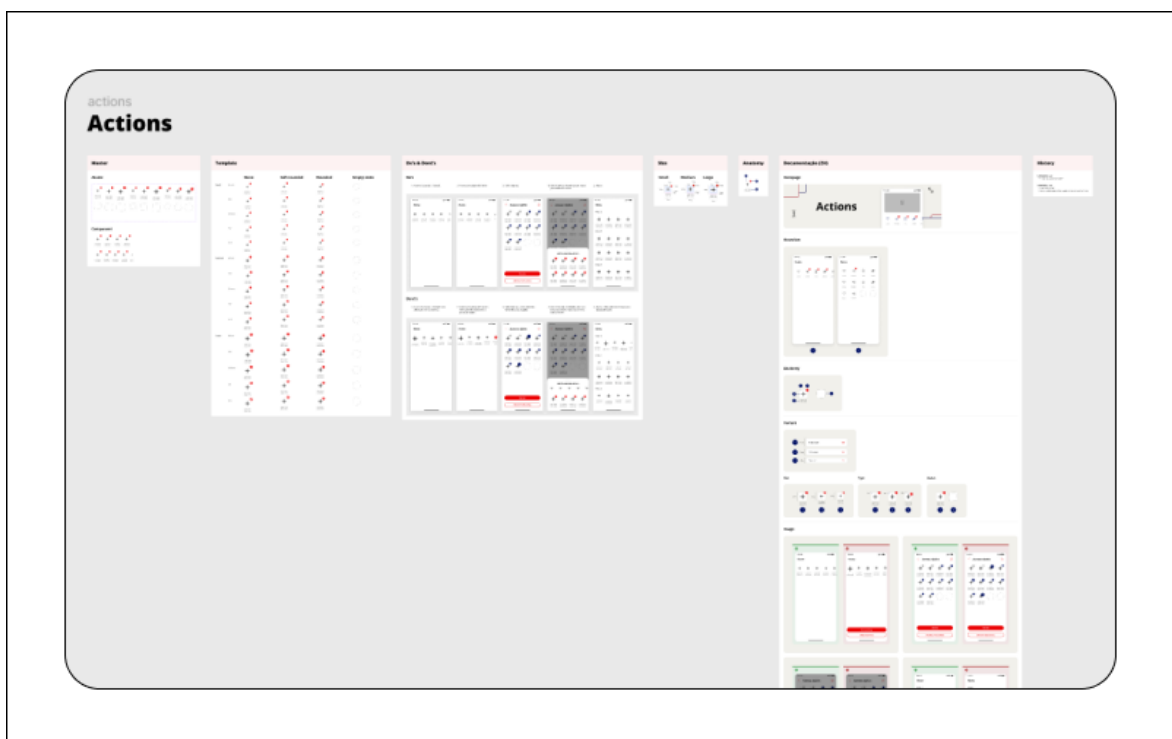
Na criação das componentes é necessário ter em consideração:

- Estrutura – Hierarquia visual, espaçamentos, tipografia entre outros, todos os componentes necessitam de regras definidas e todos os tamanhos respeitados;
- Estilo Visual – É necessário ter sempre o mesmo estilo visual, como cor, fonte, tamanho e borda. Todos estes estilos estão representados nas *Foundations*;
- Comportamento – Possuir comportamentos consistentes como o *Hover*, *Pressed* ou *Disabled* é essencial para ter uma experiência consistente em todos os casos de uso;
- Responsividade – Para todos os componentes funcionarem em diferentes tipos de tela e aparelhos diversos, para isso devem ser responsivos e se ajustar de forma adequada.

No momento atual o ficheiro conta com mais de 50 componentes, para conseguir fornecer todos os recursos para todos os projetos da MC.

Cada componente conta com a sua documentação própria:

- *Master* – Componente principal com todas as suas variações e tamanhos
- *Templates* – Todas as versões do componente
- *Do's e Dont's* – Usabilidade correta e incorreta
- *Anatomy* – Anatomia do componente, para explicar todo o processo de construção do mesmo
- Documentação (ZH) – Documentação e ilustrações mais visuais para colocar no Zeroheight
- *History* – Histórico para quando alguém faz alguma troca do componente registrar o que fez e o porquê.



**Figura 18** - Imagem da organização de um componente do Mantra

### Antes VS Depois - Organização do ficheiro *Components*

Após a criação de praticamente todos os componentes que achávamos essenciais para que todos os projetos conseguissem usar só o Mantra como base visual, fizemos uma análise de como a documentação estava a ser demonstrada no Zeroheight.

Chegamos na conclusão de que como estávamos a apresentar os componentes, que é o tema mais relevante do Design System, ele era muito centrado apenas para os designers e programadores conseguirem trabalhar, fazendo com que a restante comunidade da empresa considerasse as páginas de pouco valor ou mesmo inúteis.

Neste sentido, foi feito um redesign da demonstração da documentação através de uma apresentação visual mais apelativa, sem desvalorizar os componentes técnicos, para que a maioria dos utilizadores conseguissem compreender todo o conteúdo.

Na nova forma de documentar começamos sempre com uma *Hero Page*, um *banner* para demonstrar o nome do componente e o próprio componente, seguido de uma demonstração de todas as suas versões, um *Overview*. Posteriormente apresentamos a anatomia, com todos os seus valores técnicos, tamanhos, *padding*s e variações. Terminámos com casos de uso de “*Do’s e Dont’s*” para demonstrar quando usar e não usar o componente.

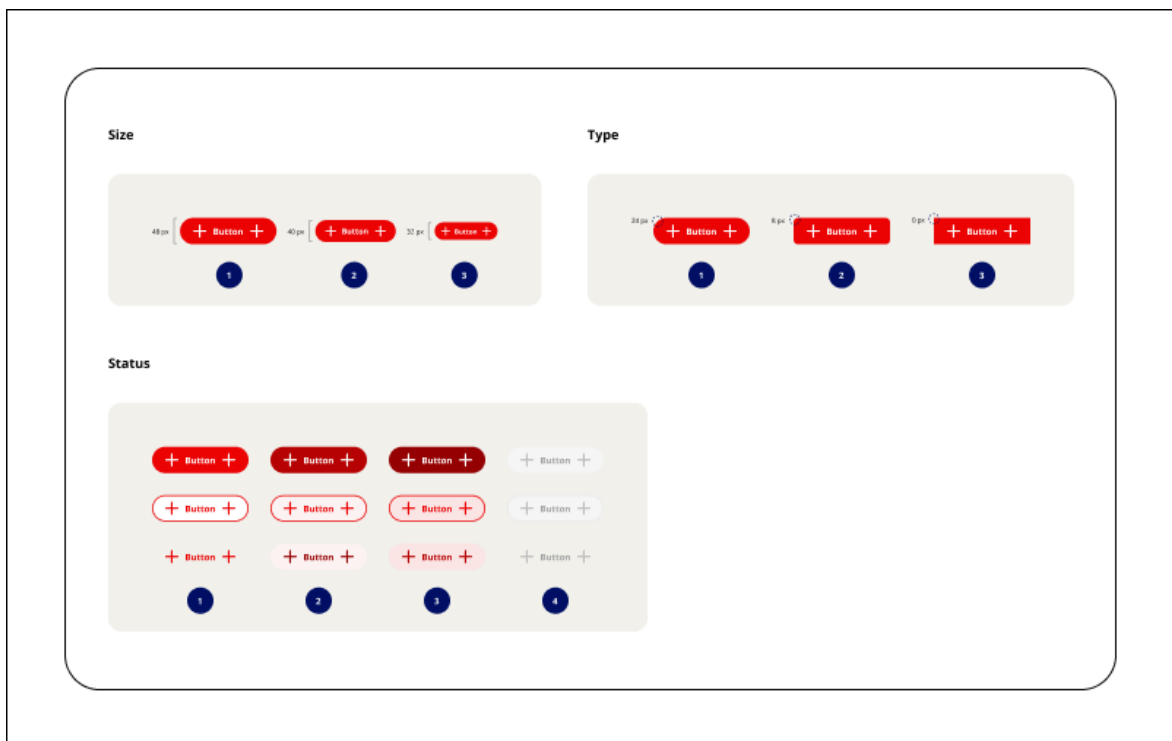
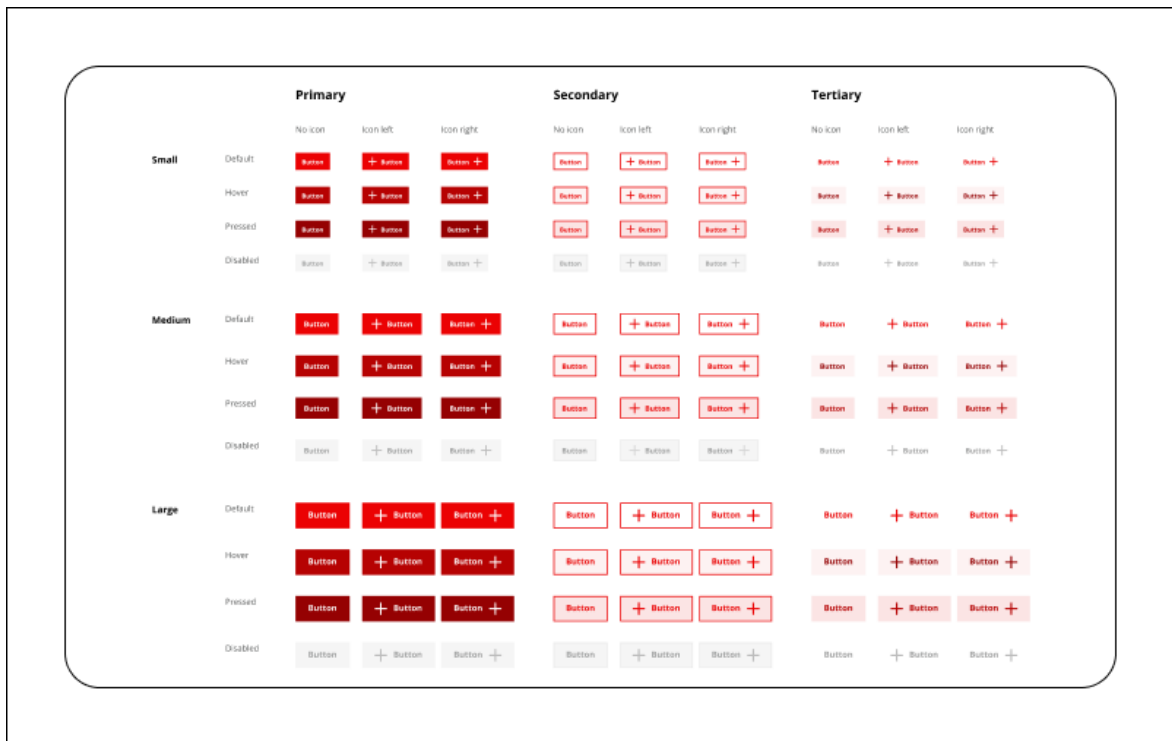


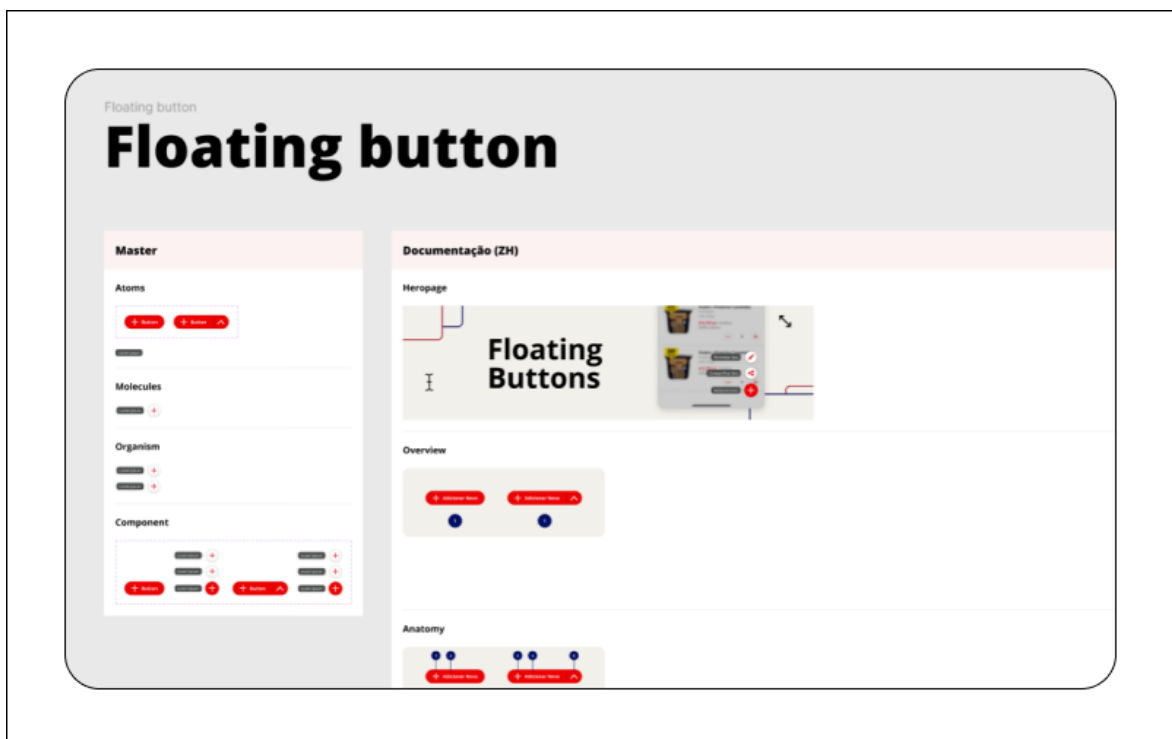
Figura 19 - Antes e depois da organização do *template* de botões

### *Patterns*

Os *patterns* são soluções reutilizáveis e comprovadas que resolvem problemas de interface recorrentes. São componentes ou combinações de componentes que foram criados para atender a necessidades mais comuns de design.

De momento o Mantra tem alguns *patterns* definidos como os filtros, notificações e *search-bar*.

Sendo soluções já testadas e aprovadas, permitem aos designers e programadores se concentrarem-se noutros tópicos, aumentando a produtividade e economizando tempo, garantindo sempre consistência visual e funcional.



**Figura 20** - Imagem do *Pattern Floating Button*

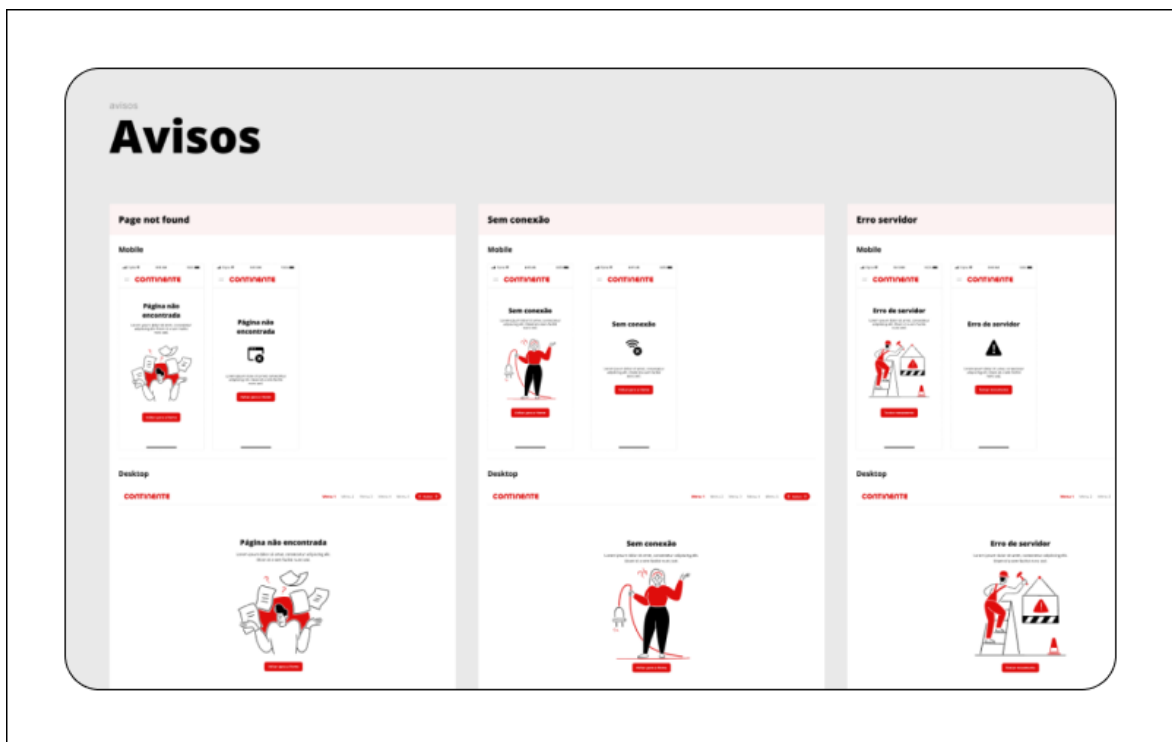
### *Templates*

Os *templates* são páginas pré-definidas que ajudam a criar consistência visual em casos comuns em aplicações digitais.

São usados para padronizar o *layout*, estrutura e comportamento dos diferentes componentes presentes nas páginas.

A principal vantagem dos *templates* é eliminar o trabalho de redesenhar e programar páginas iguais para diferentes aplicações, economizando tempo e esforço. Normalmente são estruturas básicas com um texto *copy* que atende a maioria das situações.

No momento o Mantra tem como *templates* os avisos, como a página não encontrada, erro de conexão, permissões inválidas, entre outros, e também o *onboarding* das aplicações e *splashscreen*.



**Figura 21** - Imagem dos *Templates* Avisos

## Graphics

Os *Graphics* é onde são definidos os elementos visuais do *Design System* que são usados para criar a identidade visual às aplicações.

As vantagens dos *Graphics* são manter a consistência de imagens e ilustrações em todos os ativos digitais e o facto de constituir uma biblioteca simples e útil para ambos designers e programadores usarem.

Nos *Graphics* do Mantra podemos encontrar as imagens, as *flags*, as ilustrações, pictogramas e *stickers*.

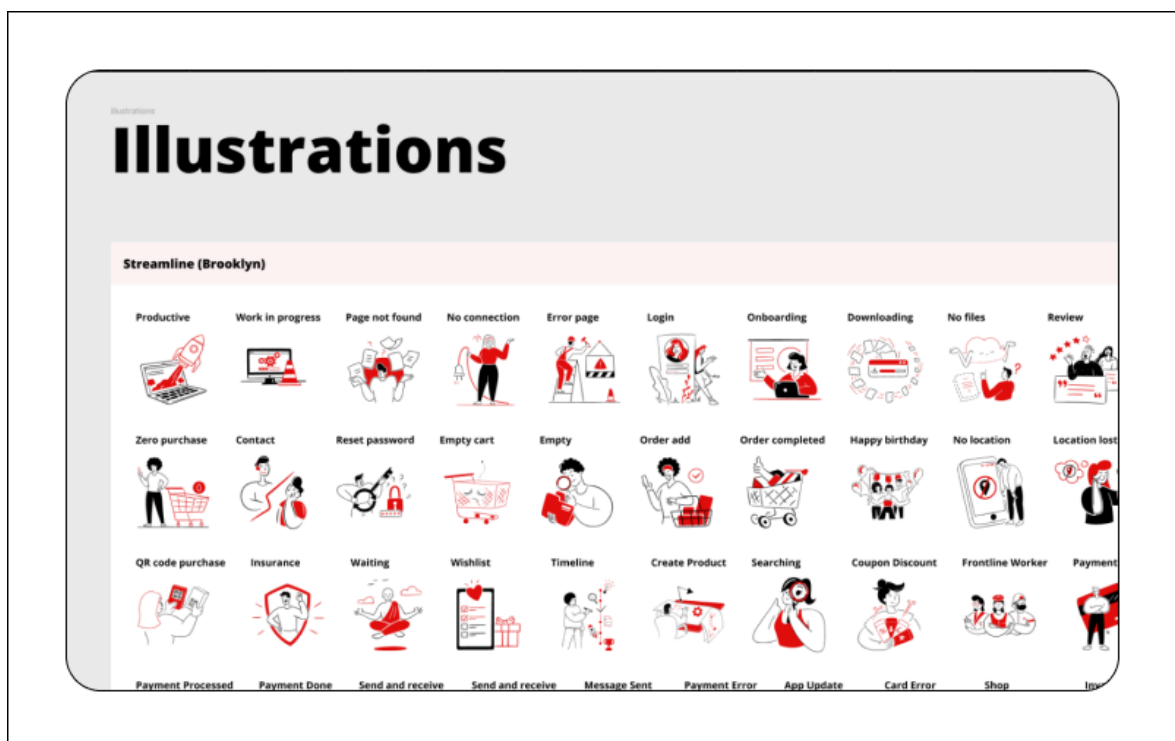


Figura 22 - Imagem dos *Graphics Illustrations*

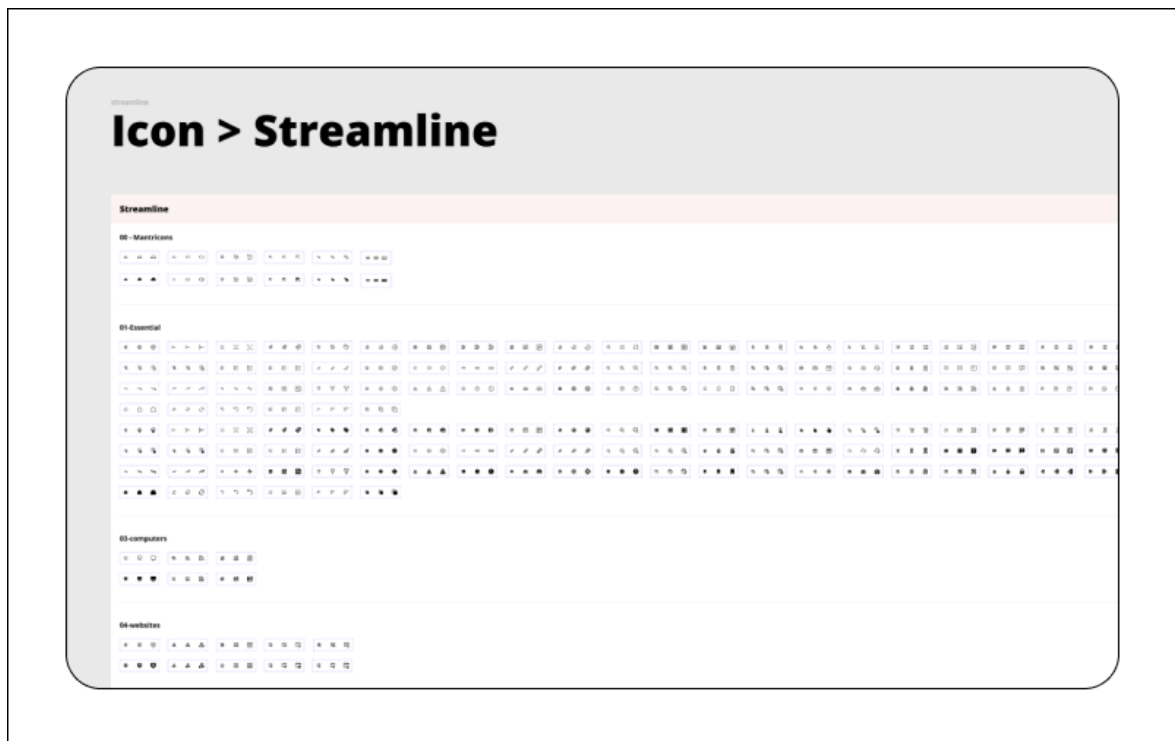


## *Icons*

Os *Icons* é onde são armazenados todos os *icons* usados nas aplicações. A MC usa uma licença de *icons* e ilustrações chamada *Streamline*, que é usado como base para todos os *icons* utilizados.

Os *icons* são personalizáveis tanto em tamanho (16px, 20px, 24px) como em estilo (*regular* e *bold*).

No momento, o Mantra tem mais de 1300 *icons* para suprir as necessidades de todas as aplicações, esses *icons* são incluídos tanto na biblioteca do Figma, quanto na biblioteca do Outsystems.

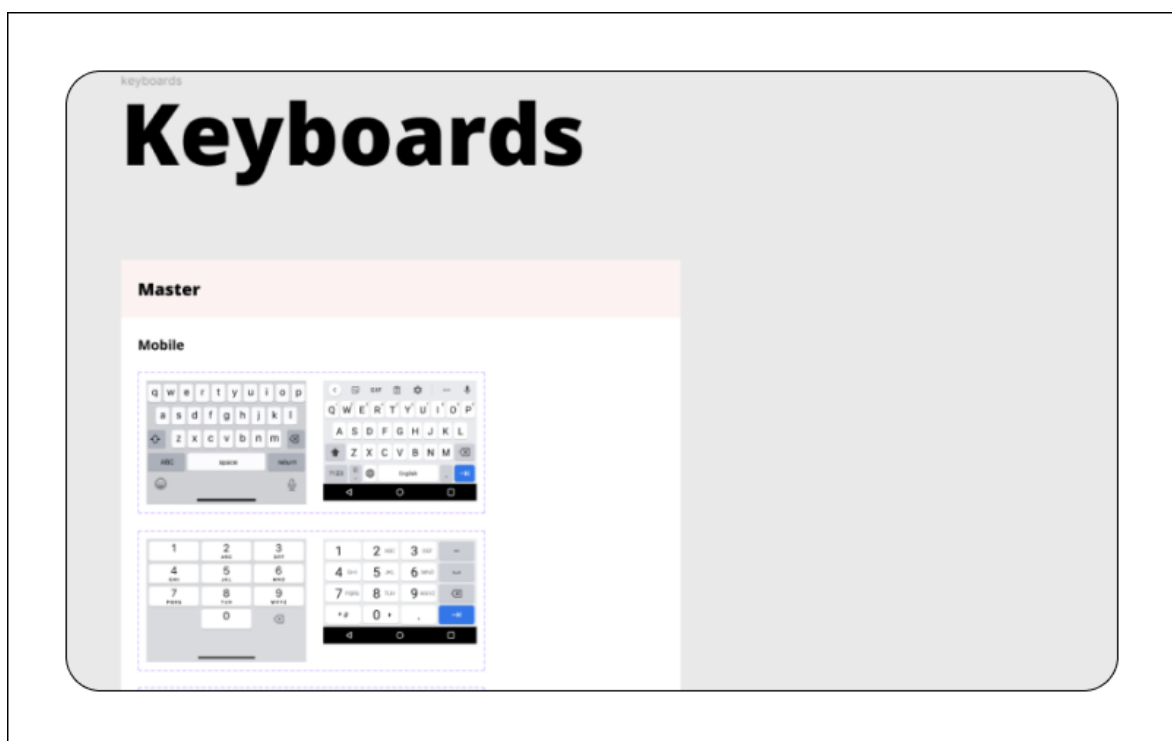


**Figura 23** - Imagem dos *Icons Streamline*

## *Helpers*

Os *Helpers* referem-se a um conjunto de utilitários usados para implementar determinadas funcionalidades de forma rápida e consistente. São criados para facilitar a implementação e manutenção do sistema, sendo assim os designers e programadores conseguem utilizar soluções padronizadas e prontas para utilização.

Atualmente, no Mantra tem nos *Helpers* os *assets* (iOS e Android), teclados (iOS e Android), *pointers* (*assets* para documentação e assinalar cenários) e a *slot* (*helper* bastante utilizado na construção dos componentes).

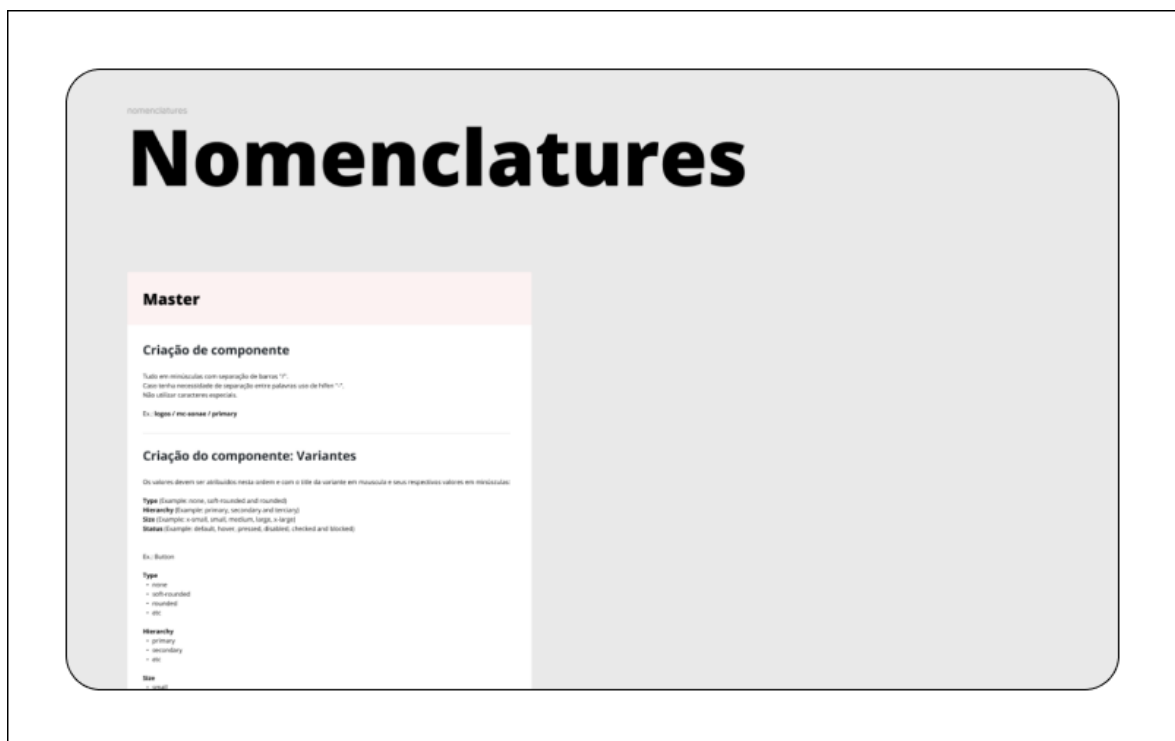


**Figura 24** - Imagem dos *Assets Keyboards*

## Rules

Por fim as *Rules* referem-se a um conjunto de regras e princípios implementadas e seguidas em todo o *Design System*. Estas regras ajudam a manter a consistência e qualidade, principalmente da organização dos ficheiros.

No Mantra existem regras para a estrutura dos ficheiros, nomenclatura de elementos e documentos, como mostrar o estado dos componentes, como criar *icons*, como implementar o *dark-mode* e regras a serem seguidas no Zeroheight para a documentação.



**Figura 25** - Imagem das *Rules* Nomenclaturas

## **Processos para inserção de novas funcionalidades**

Para um novo componente ser inserido no Mantra, passa por diversas fases, começando pelo envio de uma proposta de componente ou funcionalidade enviada por qualquer pessoa da organização ou mesmo pela própria equipa de designers para a equipa UX Chapter. Esse pedido deve conter tanto o propósito do componente, onde seria inserido e como seria aplicado.

Essa proposta seria revista pela equipa de UX, para perceber se seria necessário um novo componente e se seria usado em mais que um projeto para então entrar no backlog de necessidades do Design System.

Sido validado todos os passos, seria entrada em desenvolvimento primeiramente por um designer para criar o componente e adicioná-lo na biblioteca do Mantra (Figma), tanto o componente como a sua documentação. Após a criação é levada novamente para a equipa de UX Chapter para a aprovação do componente criado e comparado com os diversos elementos criados anteriormente no Design System para garantir a maior consistência possível.

Feito todas as aprovações e validações da equipa e do DS Lead, são comunicadas todas as informações necessárias à equipa de programadores para ser criado o mesmo componente na biblioteca da linguagem Outsystems.

Por fim é inserida toda a documentação de design e programação para o Zeroheight para toda a organização ter visibilidade da nova funcionalidade / componente.

## Criação da Biblioteca Figma

Para criar a biblioteca no Figma foram usadas duas funcionalidades principais:

- Estilos

Os estilos são uma funcionalidade usada principalmente nas *Foundations* que permite criar estilos reutilizáveis como cores, fontes, espaçamentos e efeitos. Com esta ferramenta, é possível criar num ficheiro os estilos e compartilhar para o resto dos projetos, tendo assim uma panóplia de cores e fontes para aumentar a coerência visual.

- Componentes

Os componentes no Figma são uma ferramenta que permite criar e reutilizar elementos de design de forma consistência e coesa em todos os projetos, podem ser elementos ou grupos de elementos que são transformados em instâncias, ou seja, quando esse componente é atualizado, é automaticamente aplicada a todas as instâncias relacionadas. Com esta funcionalidade, o mesmo componente criado na biblioteca do Mantra pode ser utilizado em todos os projetos da equipa, aumentando a consistência visual.

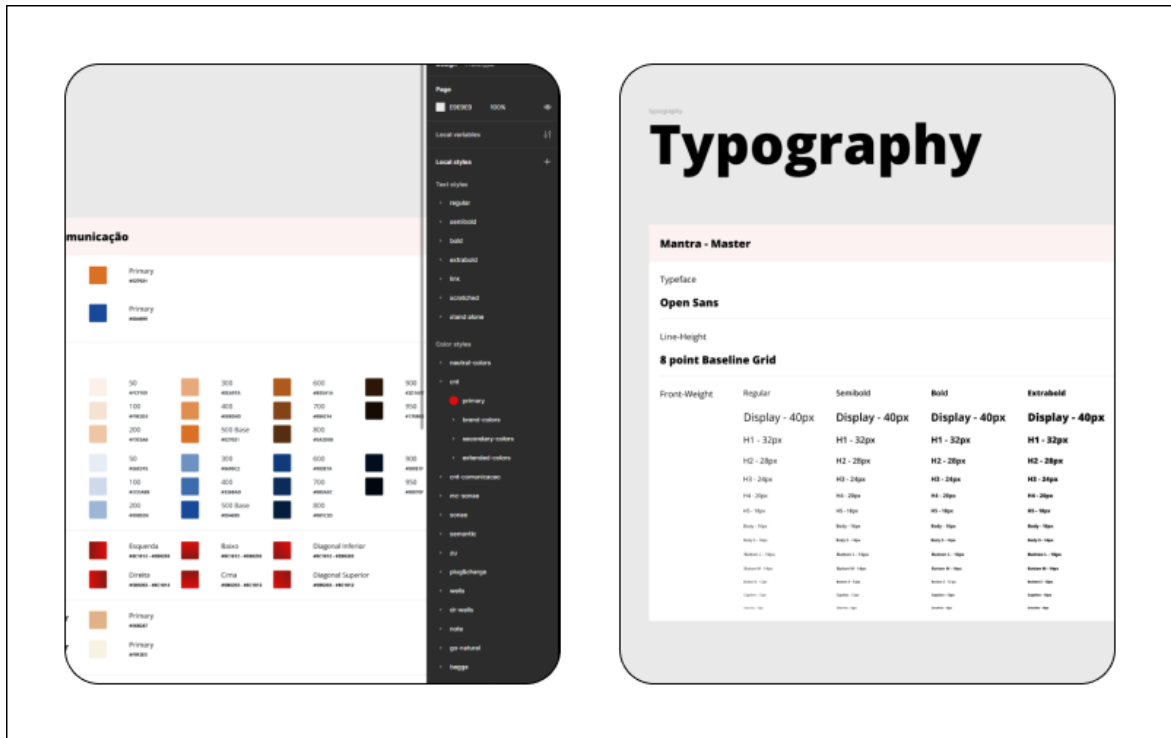


Figura 27 - Imagem demonstrativa dos estilos do Mantra com as cores e tipografia

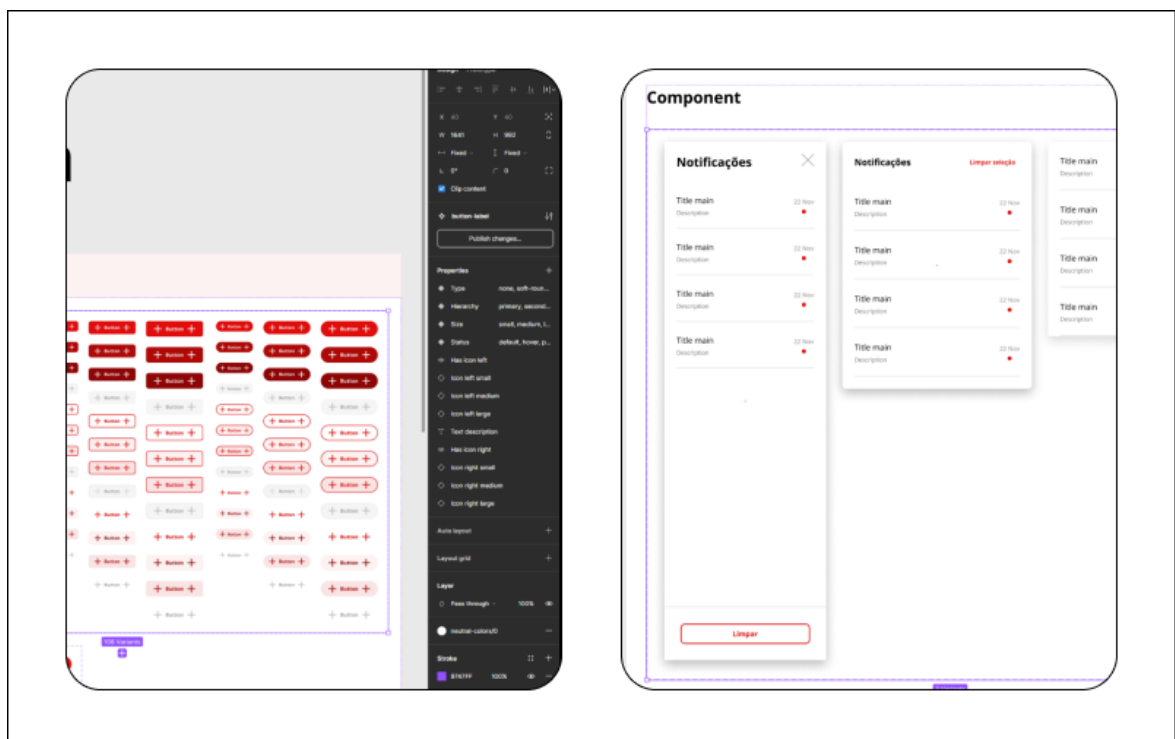
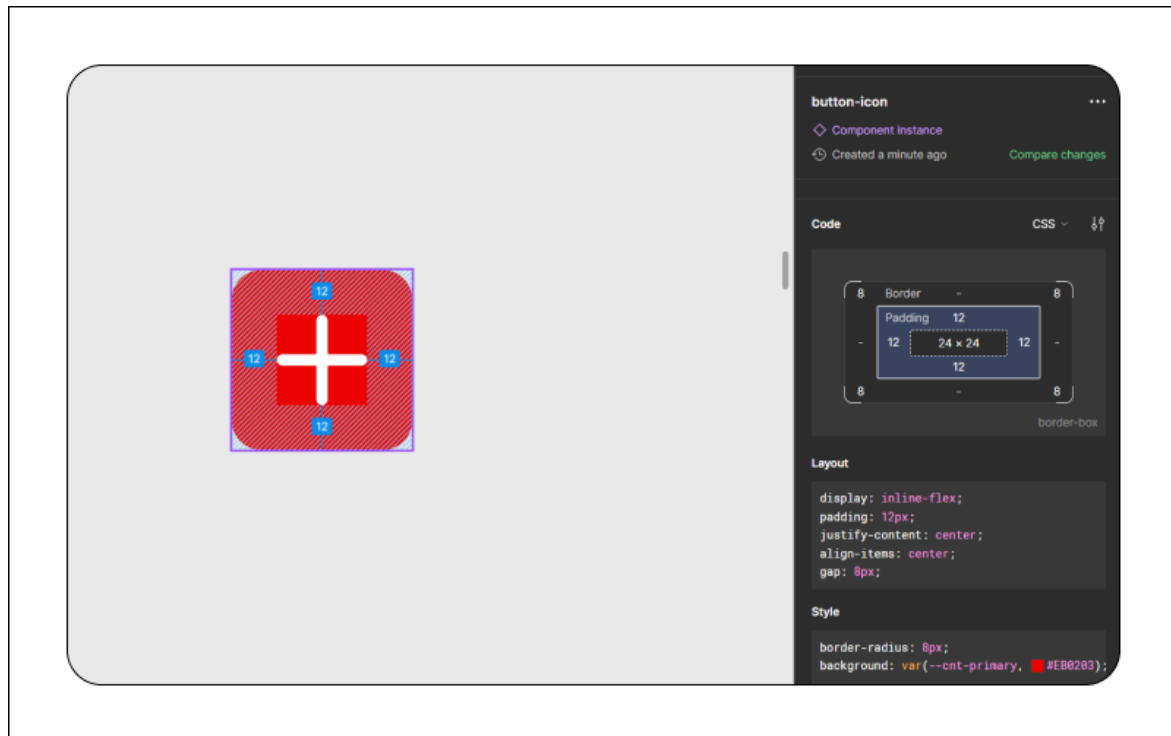


Figura 26 - Imagem demonstrativa dos componentes do Mantra com um exemplo de um Component e um Pattern.

## Transição de Figma para Código

Podem ser usadas diversas formas de transição do Figma para código, a utilizada neste projeto foi uma inspeção do CSS provido pelo próprio Figma para obter uma base dos componentes e depois são refinados com uma boa comunicação entre os designers e programadores.

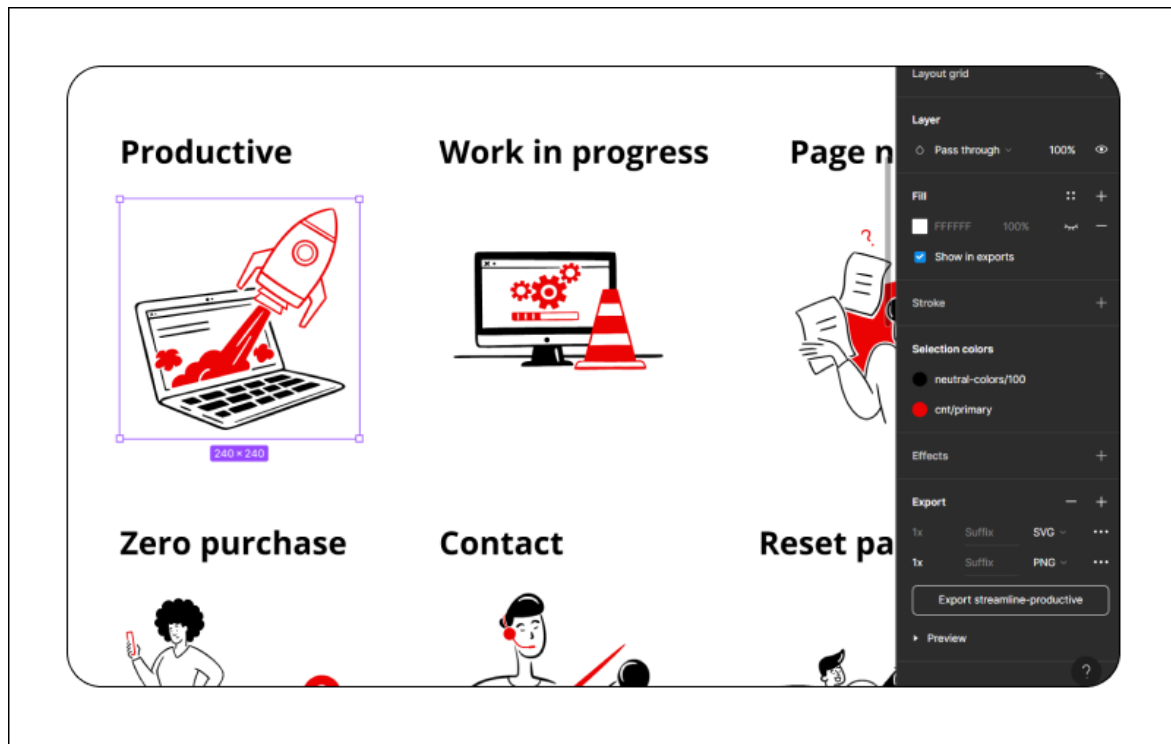


**Figura 28** - Imagem demonstrativa dos componentes do Mantra numa inspeção do CSS

Sempre que é feita alguma alteração no Design System na parte do Figma, a equipa de designers têm de comunicar para os programadores alterarem na sua biblioteca, e vice-versa, para o *design* e o *front-end* ficarem sempre consistentes e atualizados, para além disso, sempre que exista uma necessidade de um novo componente da parte de *design*, é explicado o seu uso e como deve ser implementado em programação.

Após essa leitura de necessidades e inspeção do CSS, é criada uma biblioteca de UI, na qual são implementados todos os estilos e componentes. É essencial uma linguagem que seja igual tanto em código como no *design*, como por exemplo o nome dos estilos ou nome dos componentes, permitindo uma comunicação mais fácil entre as equipas. Desta forma as equipas conseguem focar mais a sua atenção em como melhorar a experiência do utilizador pois não existe a barreira da linguagem.

Para *assets* individuais como imagens, ícones e vectores são sempre exportados em SVG, em que se encontram presentes no Figma.



**Figura 29** - Imagem demonstrativa da exportação de *assets* visuais

Com a biblioteca criada, é essencial testar minuciosamente todos os elementos para verificar a sua responsividade, se estão a funcionar corretamente e se o design é consistente em relação ao Figma.

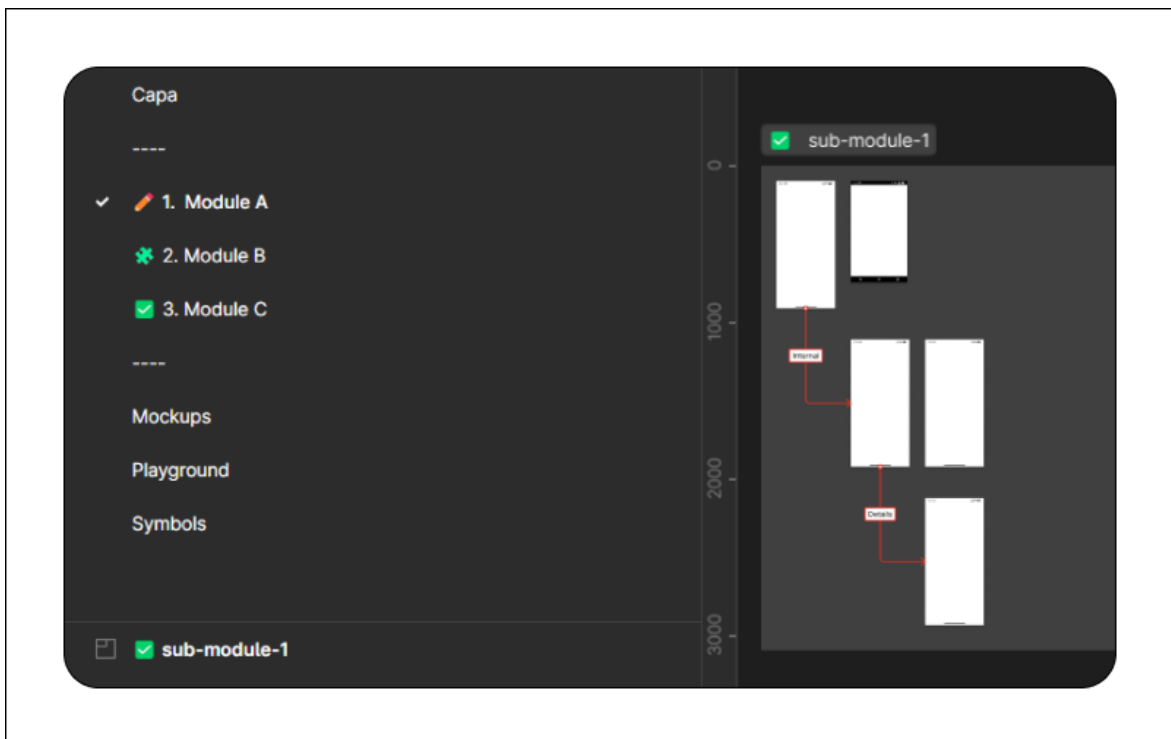
No caso da MC, a maioria das aplicações são desenvolvidas na plataforma OutSystems e foi a primeira linguagem integrada no Mantra. Posteriormente vai ser implementadas outras tecnologias como o Xamarin e uma possível biblioteca de Design Tokens.



## Utilização do Mantra no Figma

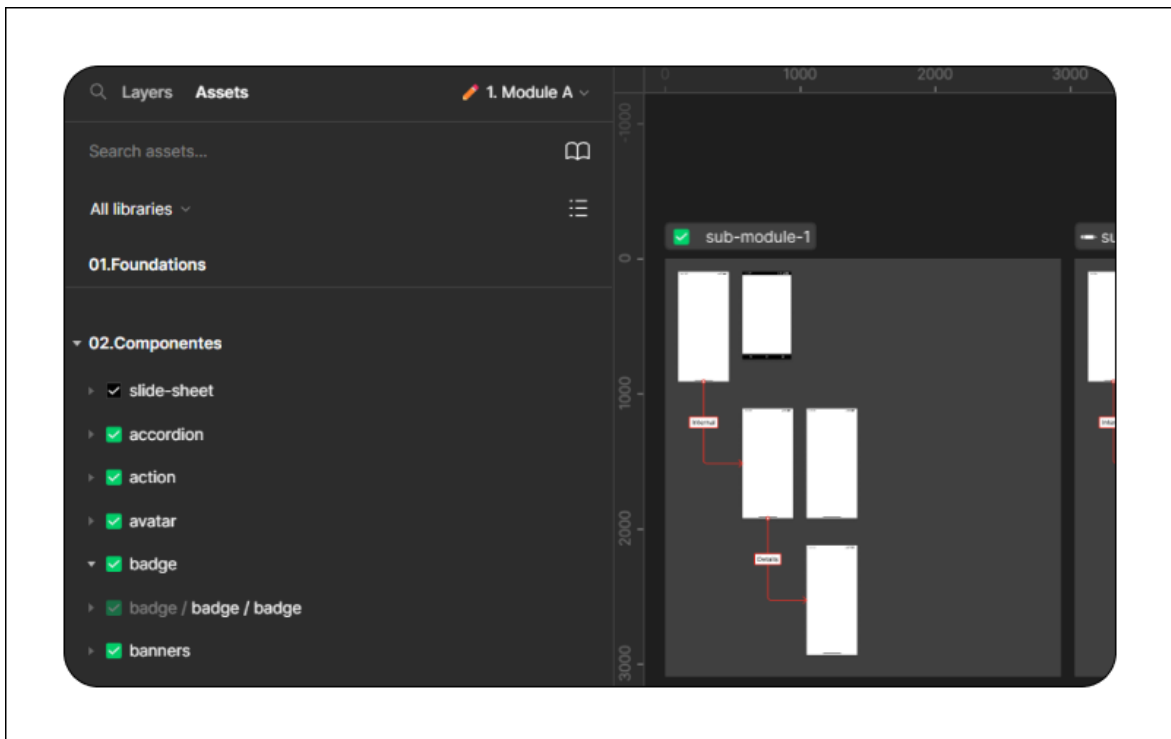
No início de cada projeto, os designers recorrem ao Mantra para selecionar um *template* inicial, mobile ou desktop, e onde pode encontrar um ficheiro organizado da seguinte forma:

- **Capa**  
Capa na qual se coloca o nome do projeto e o símbolo da aplicação;
- **Páginas com diferentes módulos da aplicação**  
Páginas organizadas por módulos e exemplo da aplicação dos fluxos;
- **Playground**  
Página na qual se coloca ideias descartadas ou testes feitos antes de aplicar no produto final;
- **Symbols**  
Página onde se armazena todos os componentes feitos para o projeto, tendo sempre como base o design system.



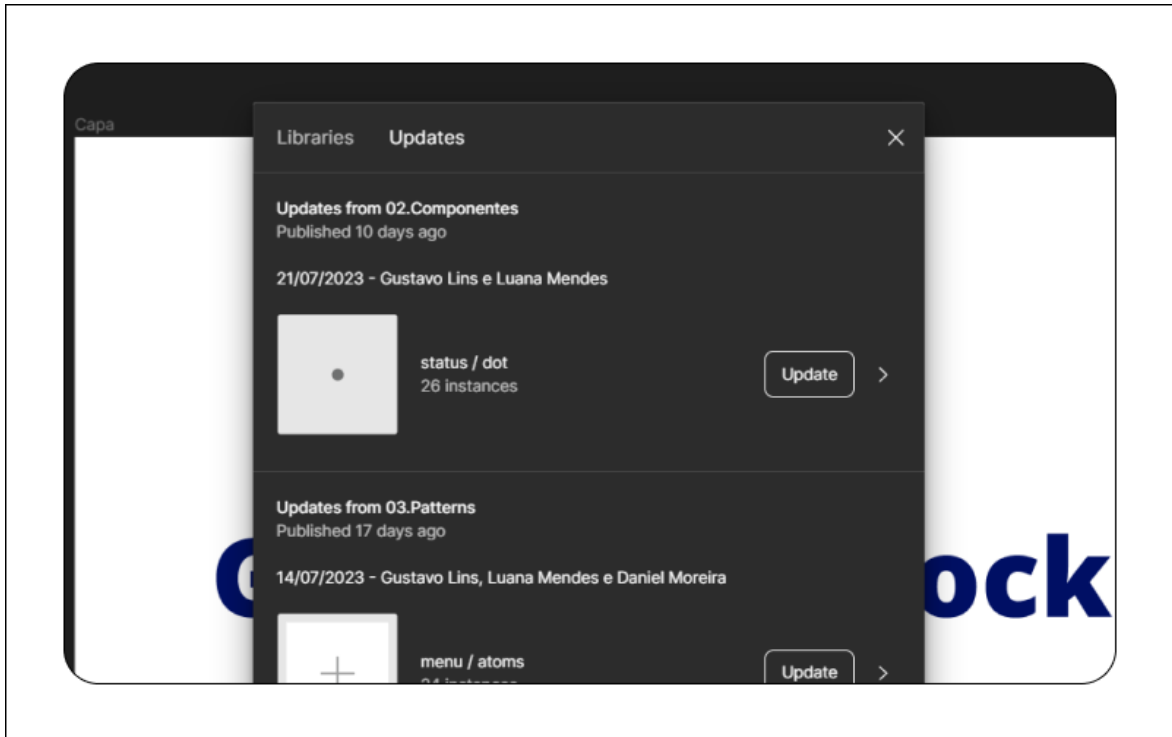
**Figura 30** - Imagem demonstrativa da organização do ficheiro “Template Mobile”

Após ter o documento do projeto organizado, sempre que tenham ecrãs para desenvolver, todos os elementos têm de ter ligação direta ao Mantra, como por exemplo os botões, *top-nav*, *accordion*, entre outros. Para isso o documento necessita estar com a biblioteca do Mantra ativa e atualizada, assim o Figma atualiza a secção “Assets” em que estarão disponíveis todos os estilos e elementos.



**Figura 31** - Imagem demonstrativa da secção “Assets”

Sempre que existem atualizações na biblioteca, o ficheiro recebe uma notificação para atualizar os elementos, e os designers podem aplicar todas as atualizações de uma vez só, ou analisar uma a uma para os ecrãs continuarem consistentes com a ideia original do projeto.



**Figura 32** - Imagem demonstrativa da atualização de Componentes do Mantra nos Projetos

## Zeroheight

Sendo o Zeroheight a ferramenta principal de documentação, é uma parte do Design System que leva muita atenção e detalhe, devido a maioria dos utilizadores utilizarem a página de visualização do Zeroheight do Mantra para saber o que é o Mantra e como utilizá-lo.

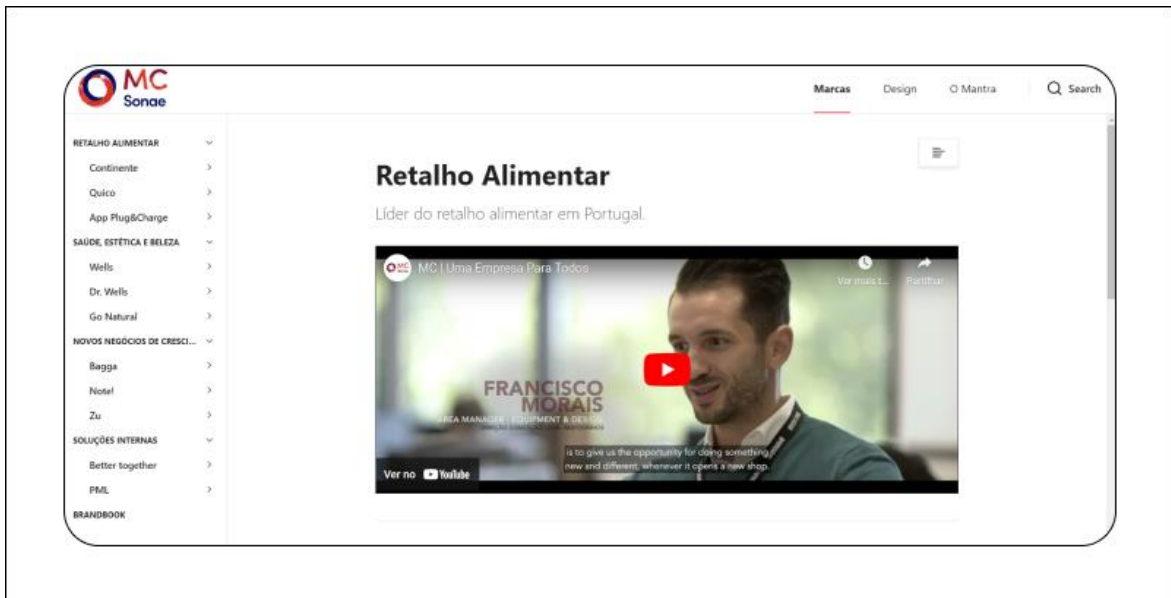
O Zeroheight tem 4 partes principais:

- *Homepage* – Página Inicial do Mantra, onde começa com um *banner* com a logo do Design System e uma pequena descrição, seguido de três *cards* com as restantes secções do Mantra e finalmente *banners* ilustrativos de projetos e *updates*;



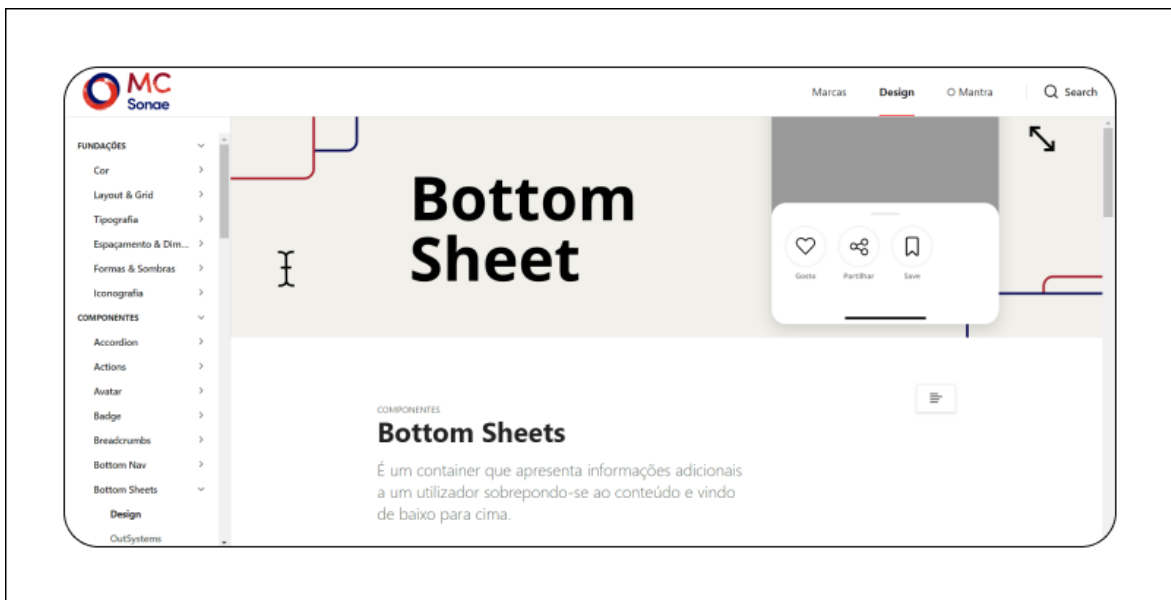
**Figura 33** - Homepage do Mantra

- Marcas – Secção com as diversas marcas da MC onde o Mantra é usado, destacando a sua história e identidade, com os seus diversos logótipos e *favicons*, e como última adição, uma *tab* para mostrar o Brandbook do Continente, com opção de fazer *download*;



**Figura 34** - Secção Marcas do Mantra

- Design – Nesta seção do Mantra, uma das mais completas, é possível encontrar documentado tudo o que o ficheiro do Figma inclui, desde as *foundations* até todos os componentes desenvolvidos, com regras de utilização, anatomia e variações. Em todas as páginas dentro da secção do Design existe uma *tab* sobre o Outsystems, onde mostra como instalar a biblioteca do Outsystems e como usar os *templates*;



**Figura 35** - Secção Design do Mantra

- O Mantra – Na última seção do Mantra são explicados os objetivos do Design System, as suas regras, as novas atualizações e são apresentadas propostas de melhoria.



**Figura 36** - Secção “O Mantra” do Mantra

### Atualização do Zeroheight para uma melhor aparência visual

Enquanto o ficheiro *Components* do Figma estava em construção, a documentação colocada no Zeroheight era bastante técnica e muito focada para designers e programadores, sendo demonstrados todos os elementos, tamanhos e estilos de uma forma que um utilizador sem nível técnico não entenderia o propósito da ferramenta.

Porém o Mantra pretende que todos os utilizadores consigam entrar na documentação e utilizar e perceber o que está a ser demonstrado, tanto para aumentar o seu público como para que todos entendam a função do Mantra, que consiste em aumentar a consistência e produtividade.

Para isso foi redesenhada a organização da informação e como é demonstrada em imagens, atualmente é feito de uma forma muito mais ilustrativa e “clean”. Por exemplo, nos botões em vez de mostrar todas as variações e estilos, é feita uma imagem com as principais variações separados por tamanho, estilo e hierarquias. Isso faz que a leitura seja mais simples e que todos os utilizadores consigam perceber a informação.

Na figura 37, mostra a comparação entre Variações (secção do Zeroheight) em que na esquerda tem a versão antiga com as variações dos *Button Group* e a direita a nova versão com as variações do *Button*, destaca-se a quantidade de informação reduzida e concentrada para melhor perceção do UI do componente.

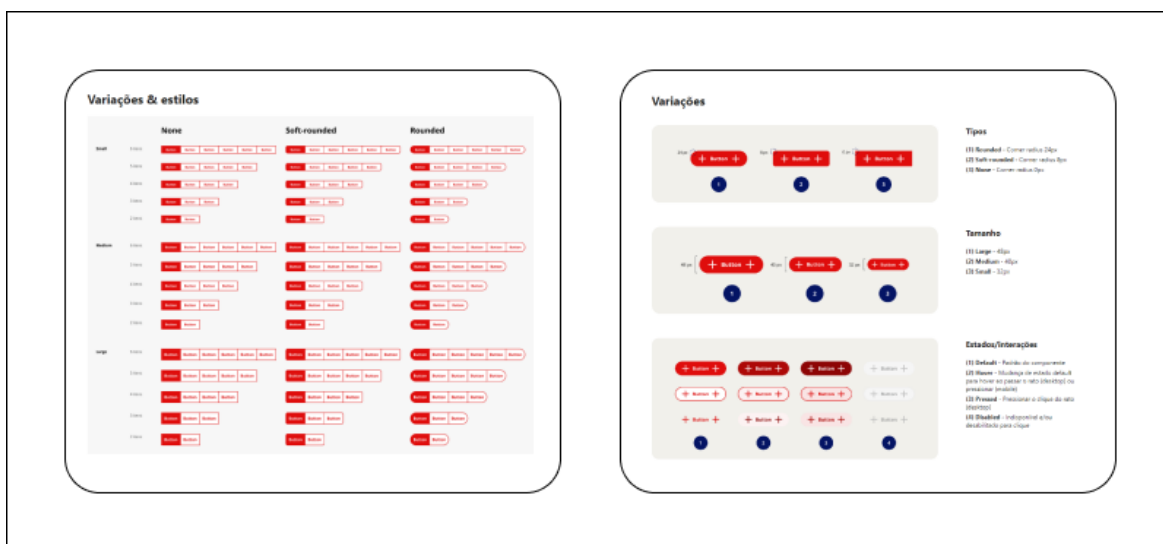
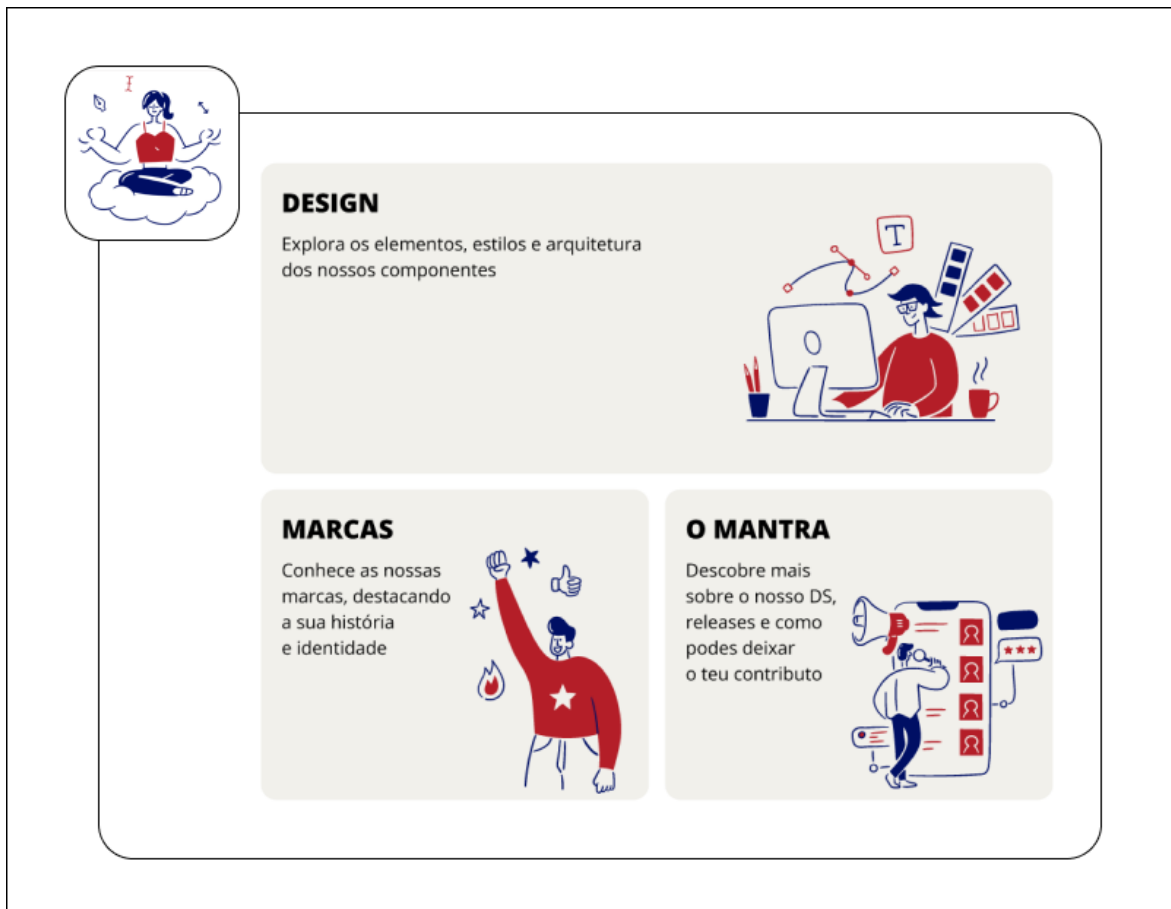


Figura 37 - Comparação entre Variações



Aumentar o conteúdo da Homepage nunca foi prioritário, no entanto com o escalamento do Mantra foi necessário melhorar o aspecto visual da página principal. Para o efeito foram adicionados banners e motion aumentando a dinâmica e tornando o site mais apelativo.



**Figura 38** - Diferentes áreas do Zeroheight Mantra, com a sua descrição e ilustrações

## Exemplos Práticos do Mantra

Com a aplicação do Mantra, todos os membros da equipa de design conseguiram focar mais na usabilidade do utilizador e tirar um pouco o foco do UI, pois o Design System contém todas as regras e componentes de UI. Isso resultou em entregas mais rápidas e mais consistentes, atualmente a ir a várias aplicações da MC pode-se reparar que a base é a mesma, em que resulta num ambiente mais amigável e reconhecível para o utilizador final.

O primeiro exemplo que vai ser mostrado é o Continente SIGA, que é a aplicação que mais teve mudanças porque foi a aplicação que recebeu um *revamp* ao mesmo tempo que o Mantra começou a ser aplicado nos ativos digitais.

Pode-se reparar em mudanças em todos os níveis da aplicação, tornando o SIGA uma aplicação mais moderna e consistente com a App Cartão Continente, recebendo o mesmo tratamento para a *bottom-nav*, *banners* e *cards*.

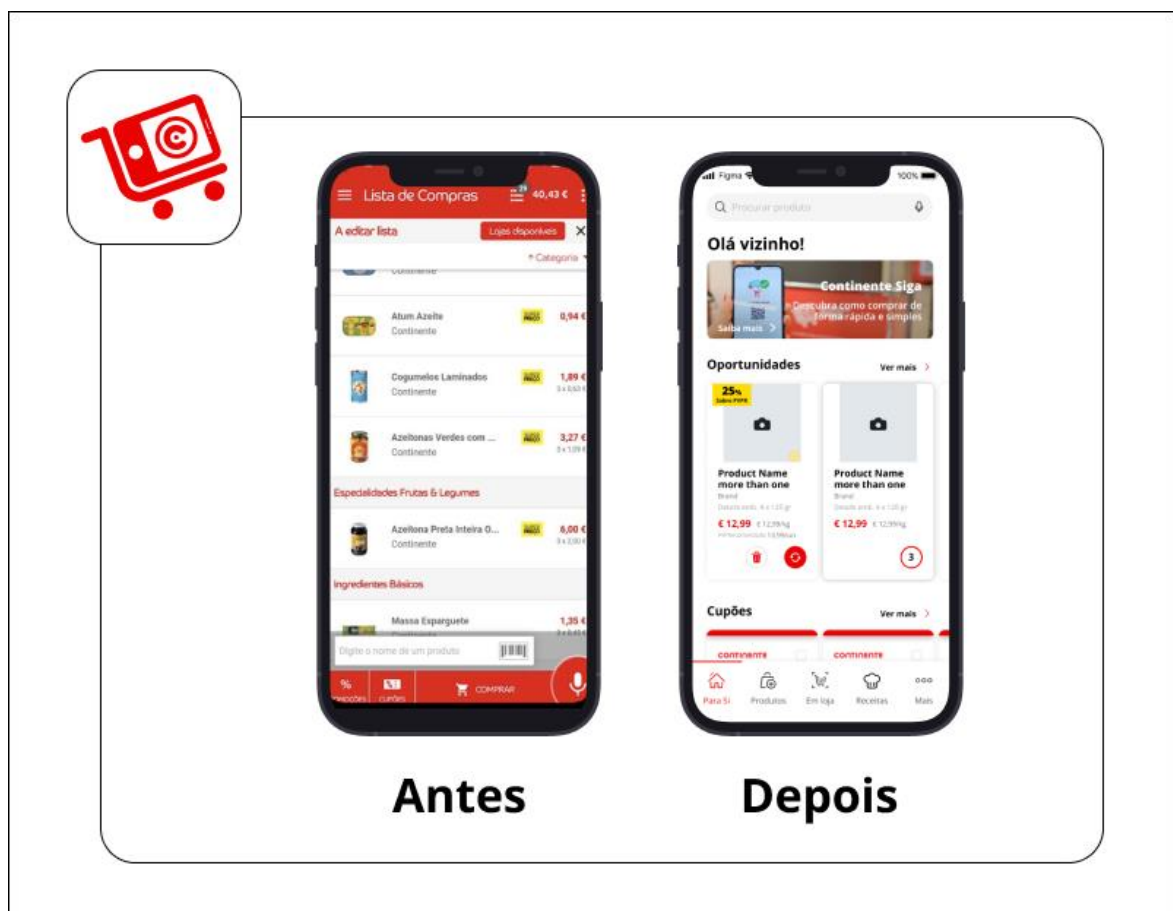
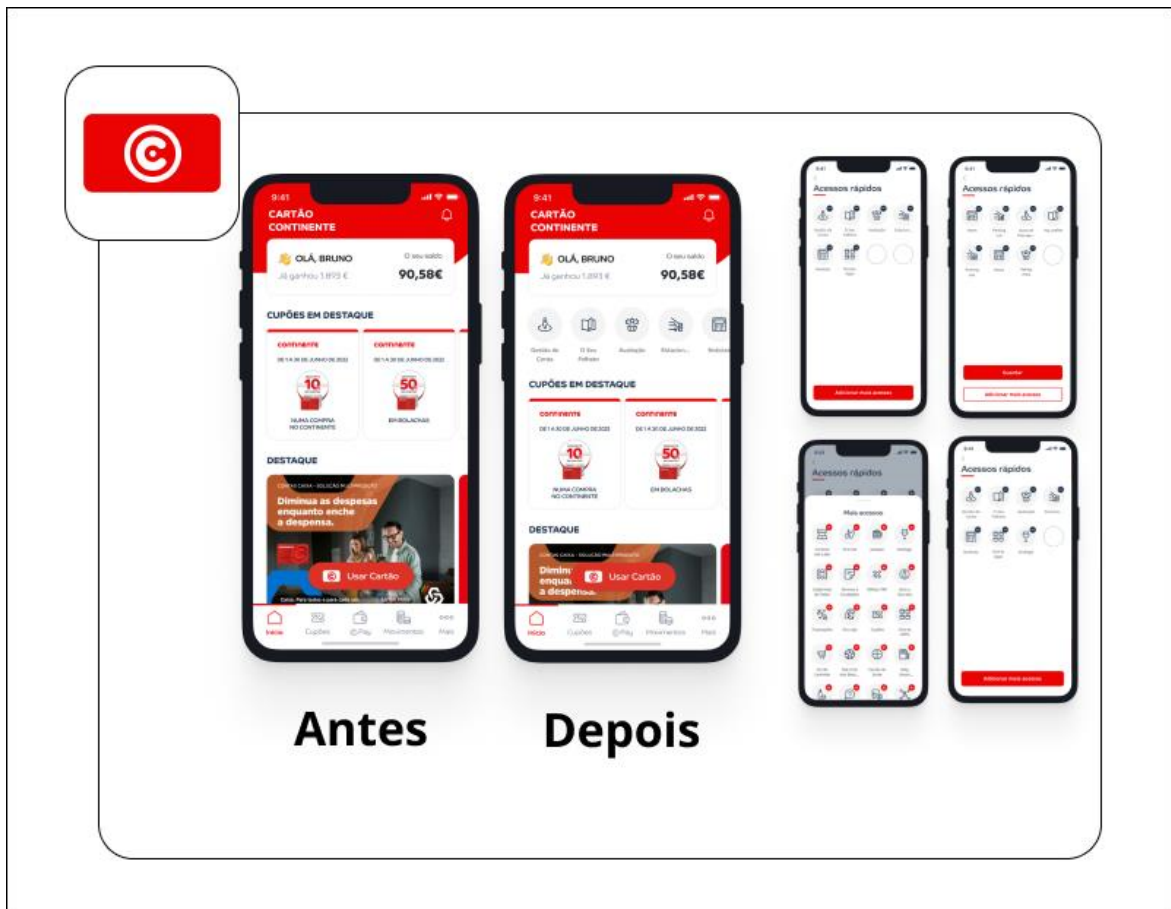


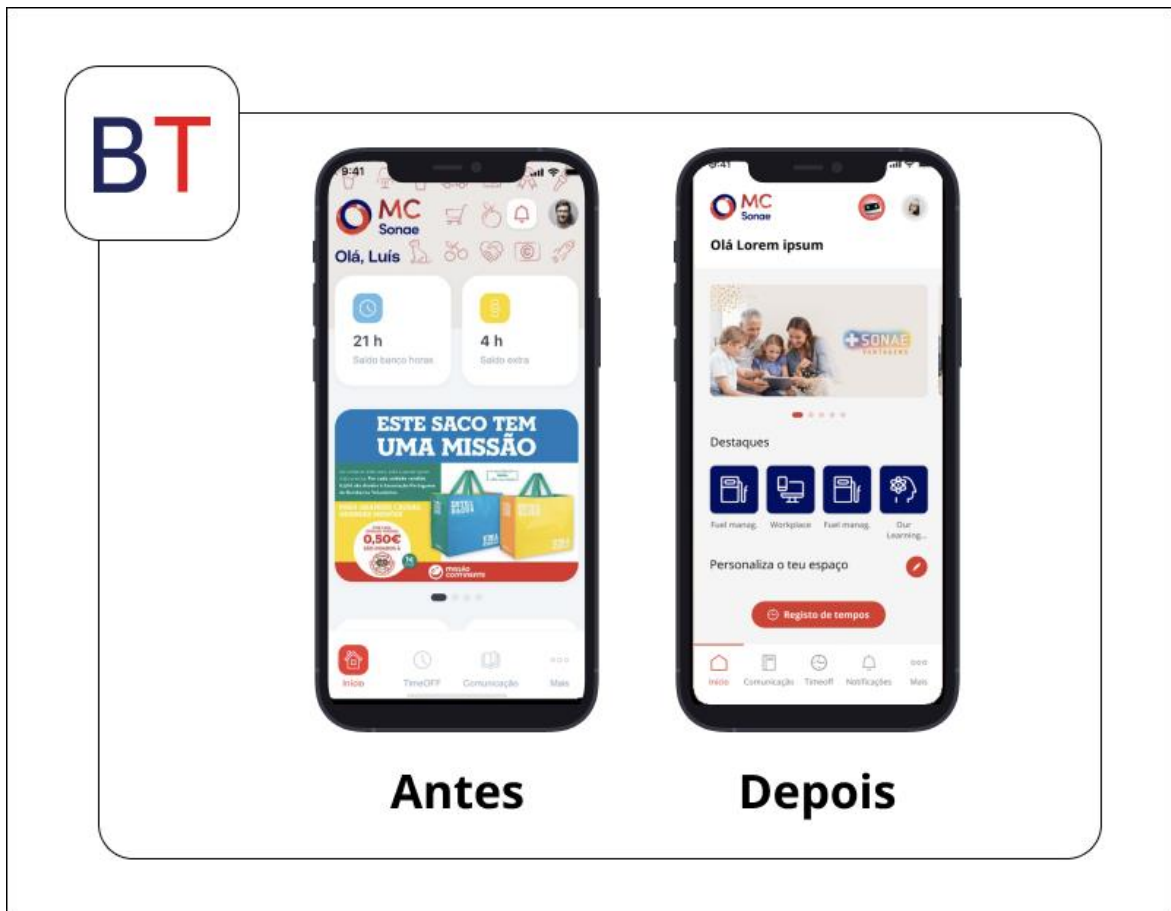
Figura 39 - Antes e depois da aplicação Continente SIGA

O próximo exemplo é a própria App Cartão Continente, em que foi adicionado uma nova funcionalidade totalmente vinda do Mantra, que é as *Actions*, em que consiste em acessos rápidos para o utilizador ter acessos a qualquer parte da aplicação na homepage, com a possibilidade de alterar a sua preferência.



**Figura 40** - Antes e depois da aplicação Cartão Continente

O último exemplo é a aplicação Better Together, que é a aplicação dos colaboradores da MC, em que também recebeu um *revamp*, em que todos os componentes têm ligação direta ao Design System.



**Figura 41** - Antes e depois da aplicação Better Together

O objetivo é todos os ativos digitais conterem ligação direta ao Mantra, tal como os exemplos acima, tanto para a consistência visual como para a facilidade de atualizar componentes de UI. Isso poupa tempo de produção e aumenta a qualidade de entrega.

## RESULTADOS

Para além do crescimento das empresas e equipas de produto, as próprias equipas mudam e crescem, o que torna a verificação da consistência nos produtos ainda mais complexo, dada a necessidade de *handover* e *onboarding* das pessoas envolvidas naquelas que são as *guidelines* para desenvolvimento.

Com o desenvolvimento de um Design System para multimarca e multiproduto e a sua própria documentação e ligação entre design e programação pode-se concluir que um Design System é essencial para uma empresa que tenha como objetivo um trabalho consistente e mais produtivo tanto em experiência do utilizador como visual.

O Mantra tornou-se a ferramenta mais utilizada pela equipa de UX e a referência visual de ativos digitais na MC. Em termos de processos, houve uma redução de custos associados a desenho e desenvolvimento de novas soluções e uma maior rapidez na entrega pois uma grande parte de UI é fornecida pelo Design System.

Para o cliente final, foi aumentada a simplicidade na utilização de todos os ativos digitais que foram atualizados com o Mantra e uma redução na fricção dos clientes que utilizam mais do que um ativo digital da MC.

## CONCLUSÕES E PERSPETIVAS FUTURAS

Este estudo tinha como objetivo explicar e analisar a importância de um Design System e como implementar com as melhores práticas de UX/UI, e pode-se concluir que esta ferramenta tem um grande impacto em qualquer empresa, não só nas equipas de design e programação, mas também em todos os ativos digitais que a organização possui.

Depois de explorar e implementar um Design System numa empresa com multiprodutos e diversas equipas de produção, pode-se concluir que a entrega e custo de novas soluções diminuíram e por sua vez, a consistência foi melhorada consideravelmente, devido ao facto dos principais elementos UI estarem devidamente definidos e documentados. Isto traz um aumento na simplicidade na utilização de todos os ativos digitais e familiaridade entre todas as aplicações da MC.

Todos os objetivos mencionados na introdução foram cumpridos e mencionados ao decorrer deste trabalho, menos a aplicação dos Design Tokens em que foi colocado como perspetiva futura.

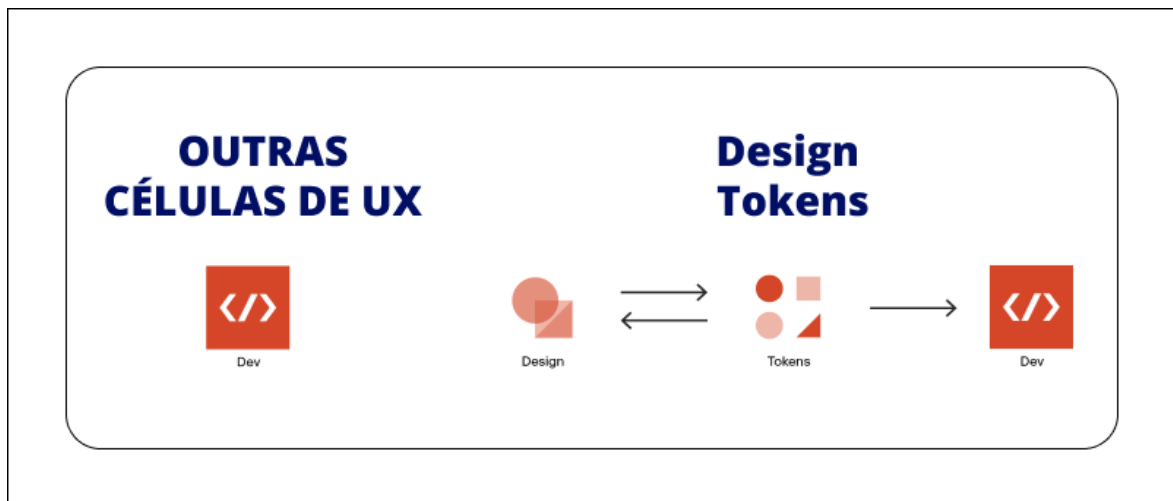
As próximas tarefas preveem a criação de uma nova célula de UX para a equipa, que consiste numa equipa de programadores dedicados à implementação completa do Mantra para código, com o auxílio dos Design Tokens.

Este passo é essencial para a conversão de todo o UI para uma ferramenta só, os Design Tokens. Em vez de utilizar valores absolutos diretamente na programação como é feito atualmente em Outsystems, os Design Tokens funcionam como peças de informação que representam valores específicos, como cores, tipografia, espaçamentos, tamanhos, entre outros, tornando mais fácil fazer alterações em toda a interface sem precisar alterar cada componente individualmente. Como referido anteriormente, esta característica é importante tanto em consistência dos componentes como em facilidade de manutenção e flexibilidade, porque, caso seja necessário por exemplo alterar o tamanho da fonte de um botão, é possível alterar um token e essa alteração é aplicada a todos os botões automaticamente.

Outro tópico que é importante destacar é o fato do próprio Figma estar a implementar um criador de tokens nativos, o que será fundamental para a realização desta nova tarefa, colocando os Design Tokens tanto em código como em design, para quando os designers

fizerem uma alteração, todos os ativos digitais consigam ter essa atualização mais rapidamente e sem perder tempo que poderia ser utilizado em melhorar a aplicação em si.

Com isto implementado, em vez de o Mantra servir apenas Outsystems em código, irá prover para qualquer linguagem de programação sem ter de uma equipa realizar todo o trabalho feito anteriormente.



**Figura 42** - Nova Célula de UX e Design Tokens

Neste momento está a ser realizada uma tarefa que tem como objetivo melhorar o aspeto visual e reduzir a complexidade da documentação apresentada no Zeroheight, permitindo aos utilizadores terem uma ideia concreta sobre a função do Mantra e os seus aspetos diferenciadores. Importante destacar que a documentação no Zeroheight é tanto para informar os utilizadores de como o Mantra é usado e implementado tanto como para vender a própria ferramenta para toda a companhia, dito isso é fundamental que a documentação tenha uma linguagem simples e precisa para todos os utilizadores conseguirem perceber o fundamental.

Por fim, o último tópico para tarefas futuras seria adicionar mais aplicações da MC a trabalhar diretamente com o Mantra e começar a influenciar os ativos físicos. Para aplicações digitais os principais seriam o COL (Continente Online) e a Wells, e para os ativos físicos seria a implementação do BrandBook Continente para o Mantra e criação de Newsletters, cartazes e banners.

## BIBLIOGRAFIA

12 leading design systems (& what you can learn from them). (2022). Dribbble.com. <https://dribbble.com/resources/design-system-examples>

Abdi, M. (2021). Fundamentals of design systems.

Bhawalkar, G. (2022, December 12). You need A design system — here's why. Forrester. <https://www.forrester.com/blogs/you-need-a-design-system-heres-why/>

Brett, W. (2022, November 3). 8 design system tools for organizations. Anima Blog; Anima. <https://www.animaapp.com/blog/industry/design-system-tools-for-organizations/>

Carbon Design System. (n.d.). Carbondesignsystem.com. Retrieved August 28, 2023, from <https://carbondesignsystem.com/>

Coyle, A. (2020, June 9). A quick guide on creating a design system - Andrew Coyle. Medium. From <https://coyleandrew.medium.com/a-quick-guide-to-creating-a-design-system-7888e267171f>

Cunningham, K., & Feagins, R. (2020, September 11). What is a UX Design System? Fuzzy Math. <https://fuzzymath.com/blog/what-is-ux-design-system/>

Decathlon Design System. (n.d.). Decathlon.deSign. Retrieved August 28, 2023, from <https://decathlon.design/>

Design System: Como Funciona e Por Que Usá-lo em UX Design? (2021, January 26). Aelaschool.com; Aela School. <https://aelaschool.com/designdeinteracao/design-system-como-funciona-e-por-que-usa-lo/>

Design systems: A beginner's guide to building one that will last. (2021, June 24). Webflow. <https://webflow.com/blog/design-systems>

Design systems: vantagens e mitos. (2018, September 25). EDIT. - Disruptive Digital Education. <https://weareedit.io/blog/design-systems-vantagens-e-mitos/>

DesignerUp. (2020, June 3). 10 best design systems and how to learn (and steal) from them. DesignerUp. <https://designerup.co/blog/10-best-design-systems-and-how-to-learn-and-steal-from-them/>



- Fanguy, W. (2019). A comprehensive guide to design systems. Invisionapp.com. <https://www.invisionapp.com/inside-design/guide-to-design-systems/>
- Figma: o que é a ferramenta, design e como usar. (2022, September 2). Alura. <https://www.alura.com.br/artigos/figma/>
- Fitzgerald, A. (2023, August 28). What's a design system & what components is it made up of? [examples]. HubSpot. <https://blog.hubspot.com/website/design-system>
- Fluent 2 design system. (n.d.). Fluent 2 Design System. Retrieved August 28, 2023, from <https://fluent2.microsoft.design/>
- Gourley, R. (2023, May 26). What is a design system? Robertcreative.com; Rob Gourley. <https://www.robertcreative.com/blog/what-is-a-design-system>
- Hey design systems! (n.d.). Heydesign.Systems. Retrieved August 28, 2023, from <https://heydesign.systems/>
- IPNET Growth Partner. (2021, March 18). Design System: o que é e quais os benefícios? Parceiro de Crescimento IPNET. <https://medium.com/ipnet-growth-partner/design-system-o-que-e-438773dd811>
- Lidio, R. L. (2019, November 29). Afinal, o que é Design System? - UX Collective BR. UX Collective BR. <https://brasil.uxdesign.cc/afinal-o-que-%C3%A9-design-system-448c257b0021>
- Limcaco, J. (n.d.). Design systems repo. Design Systems Repo. Retrieved August 28, 2023, from <https://designsystemsrepo.com/design-systems/>
- Lozeau, L. (2023, June 20). What is a design system? Contentful. <https://www.contentful.com/blog/design-system-explained/>
- Madsen, R. (n.d.). Programming Design Systems. Programmingdesignsystems.com. Retrieved August 28, 2023, from <https://programmingdesignsystems.com/>
- Mapa dos Design Systems em Portugal 2022. (2022, December 5). Issuu. [https://issuu.com/rubenferreiraduarte/docs/mapa\\_design\\_systems\\_portugal\\_2022](https://issuu.com/rubenferreiraduarte/docs/mapa_design_systems_portugal_2022)
- Material Design. (n.d.). Material Design. Retrieved August 28, 2023, from <https://m3.material.io/>

Myre, M. (2022, October 13). Design systems 101: An introductory guide. Designlab.com. <https://designlab.com/blog/guide-to-ux-design-systems/>

Nanz, K. G. (2022, February 7). Design systems for developers. Telerik Blogs. <https://www.telerik.com/blogs/design-systems-developers>

Outsystems. (n.d.). Valuga.Nl. Retrieved August 28, 2023, from <https://www.valuga.nl/what-is-outsystems/>

Pernice, K. (2021). Design systems 101. Nielsen Norman Group. <https://www.nngroup.com/articles/design-systems-101/>

Pitchai, N. P. (2023). Top 10 design systems in 2023. LinkedIn.com. <https://www.linkedin.com/pulse/top-10-design-systems-2023-naveen-prasath/>

Primavera Design System. (n.d.). Primaverabss.com. Retrieved August 28, 2023, from <https://design.primaverabss.com/>

Sparkbox. (n.d.). What are the different layers and parts of a design system? Sparkbox.com. Retrieved August 28, 2023, from [https://sparkbox.com/foundry/design\\_system\\_makeup\\_design\\_system\\_layers\\_parts\\_of\\_a\\_design\\_system](https://sparkbox.com/foundry/design_system_makeup_design_system_layers_parts_of_a_design_system)

The Design System guide. (n.d.). Thedesignsystem.Guide. Retrieved August 28, 2023, from <https://thedesignsystem.guide/>

The Future of Design Systems - online conference June 8-9, 2023. (n.d.). Intodesignsystems.com. Retrieved August 28, 2023, from <https://www.intodesignsystems.com/>

The paradox of design systems. (n.d.). Spotify Design. Retrieved August 28, 2023, from <https://spotify.design/article/the-paradox-of-design-systems>

“What is a Design System?” an article by Dan Mall. (2023, January 3). Danmall.com. <https://danmall.com/posts/what-is-a-design-system/>

What is a Design System? Definition & Examples. (2019, March 26). Uxmifit.com. <https://uxmifit.com/2019/03/26/what-is-a-design-system-everything-you-need-to-know/>

Zeroheight. (2022, November 28). Zeroheight. <https://zeroheight.com/>