



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**AMINTAS VICTOR RAMOS PEREIRA**

**EXPANDINDO O SMARTCAMPUS:  
UM GUIA BASEADO EM STORYBOOK PARA  
DESENVOLVEDORES**

**CAMPINA GRANDE - PB**

**2021**

**AMINTAS VICTOR RAMOS PEREIRA**

**EXPANDINDO O SMARTCAMPUS:  
UM GUIA BASEADO EM STORYBOOK PARA  
DESENVOLVEDORES**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de **Bacharel** em  
Ciência da Computação.**

**Orientador: Professor Dr. Andrey Elísio Monteiro Brito.**

**CAMPINA GRANDE - PB**

**2021**



P436e Pereira, Amintas Victor Ramos.  
Expandindo o SmartCampus - um guia baseado em  
Storybook para desenvolvedores. / Amintas Victor Ramos  
Pereira. - 2021.

10 f.

Orientador: Prof. Dr. Andrey Elísio Monteiro Brito.

Trabalho de Conclusão de Curso - Artigo (Curso de  
Bacharelado em Ciência da Computação) - Universidade  
Federal de Campina Grande; Centro de Engenharia Elétrica  
e Informática.

1. SmartCampus UFCG. 2. Campus inteligente UFCG. 3.  
Tecnologia Storybook. 4. Dashboard. 5. Desenvolvimento  
de aplicação - documentação. I. Brito, Andrey Elísio  
Monteiro. II. Título.

CDU:004 (045)

**Elaboração da Ficha Catalográfica:**

Johnny Rodrigues Barbosa  
Bibliotecário-Documentalista  
CRB-15/626

**AMINTAS VICTOR RAMOS PEREIRA**

**EXPANDINDO O SMARTCAMPUS:  
UM GUIA BASEADO EM STORYBOOK PARA  
DESENVOLVEDORES**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de **Bacharel** em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Andrey Elísio Monteiro Brito  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Carlos Eduardo Santos Pires  
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 25 de maio de 2021.**

**CAMPINA GRANDE - PB**

## **ABSTRACT**

The Smart Cities (CI) solutions can make a physical location safer, more efficient and attractive to the population. Considering this, SmartCampus UFCG emerges as a platform that uses the university campus as a study area to develop these solutions and replicate them in cities. Something essential in this initiative is its dashboard: in addition to contributing to the good use and management of information, it promotes engagement. However, the construction and evolution of a good dashboard for an interdisciplinary project is not trivial, requiring an artifact that helps the development team. In order to work around this problem, this work consists of documenting the main components of the SmartCampus dashboard and the procedures used to customize its features, in addition to informing good development practices in this application. To this end, the Storybook technology will be used: a tool aimed at exploring interfaces. In this way, we assist developers through documentation that evolves and runs in parallel to the application, reducing effort on code changes.

# Expandindo o SmartCampus - Um guia baseado em Storybook para desenvolvedores

Amintas Pereira

Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil

amintas.pereira@ccc.ufcg.edu.br

Andrey Brito

Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil

andrey@computacao.ufcg.edu.br

## RESUMO

As soluções de Cidades Inteligentes (CI) podem tornar um local físico mais seguro, eficiente e atrativo à população. Considerando isso, o SmartCampus UFCG surge como uma plataforma que utiliza o campus como área de estudo para desenvolver estas soluções e replicá-las nas cidades. Algo essencial nesta iniciativa é o seu dashboard: além de contribuir com o bom uso e gestão das informações, ele promove engajamento. Todavia, a construção e evolução de um bom dashboard para um projeto interdisciplinar não é trivial, necessitando de um artefato que auxilie a equipe de desenvolvimento. Com o objetivo de contornar esse problema, este trabalho consiste em documentar os principais componentes do dashboard SmartCampus e os procedimentos utilizados na customização de suas funcionalidades, além de informar boas práticas de desenvolvimento nesta aplicação. Para tal, será utilizada a tecnologia Storybook: uma ferramenta voltada à exploração de interfaces. Desta forma, auxiliamos os desenvolvedores através de uma documentação que evolui com a aplicação, rodando paralelamente a esta última, e que reduz o esforço nas alterações de código.

## PALAVRAS-CHAVE

Campus Inteligente, Dashboard, Documentação.

## LINKS ÚTEIS

Documentação do Dashboard SmartCampus UFCG disponibilizada ao público:

<http://storybook.smartcampus.surge.sh>.

Dashboard SmartCampus UFCG:

<https://dashboard.smartcampus.ufcg.edu.br>.

Portal SmartCampus UFCG:

<https://smartcampus.ufcg.edu.br>.

## 1. INTRODUÇÃO

Entendemos por Cidade Inteligente (CI) um local físico em que são aplicadas tecnologias de informação e comunicação (TICs) que tornam o ambiente mais seguro, eficiente e atraente à população [1]. Além disso, a comunidade terá maior participação nas tomadas de decisões e, juntamente com o corpo administrativo, compõem uma governança colaborativa, a qual atua na gerência eficiente de recursos.

Cidades como São Paulo, Florianópolis e Curitiba se destacam como cidades inteligentes, segundo o Ranking Connected Smart Cities 2020 [2]. Por outro lado, há muito o que evoluir nas demais cidades brasileiras para que sejam beneficiadas pelas soluções anteriormente citadas.

Diante disso, um campus universitário inteligente (ou Smart Campus) consiste em uma versão reduzida de uma CI. Ele possui problemas semelhantes ao de uma cidade inteligente, tais como: trânsito, segurança, consumo de água e energia, e mobilidade pública. No entanto, por ter uma comunidade mais limitada e que valoriza a pesquisa e a inovação, é um local mais fértil para experimentar novos conceitos, os quais poderão ser transferidos para ambientes mais abertos, como as cidades.

Pensando nisso, está sendo desenvolvido na UFCG (Universidade Federal de Campina Grande) o SmartCampus [3]: uma plataforma com foco em analisar, implementar, testar e replicar soluções para Cidades Inteligentes, utilizando o campus universitário como uma área para estudos. Através dela, instituições de ensino podem combinar tecnologias inteligentes com infraestrutura física, melhorando serviços, logísticas e sustentabilidade no campus.

Embora o dashboard SmartCampus UFCG [4] tenha uma arquitetura modular, o que facilita sua personalização, não existe uma documentação que oriente a equipe de desenvolvimento a adaptar e/ou criar componentes para contribuir com o projeto.

Consequentemente, o esforço necessário para desenvolver novas funcionalidades e adaptar aplicações previamente implementadas é maior. Afinal, a exploração do código para conhecer o sistema se torna mais custosa.

Neste artigo, abordaremos o projeto implementado para resolver os problemas citados anteriormente. Iremos detalhar a estrutura da solução (ênfase na arquitetura e suas funcionalidades), a metodologia utilizada no desenvolvimento, e os resultados obtidos.

## 2. SOLUÇÃO

Para reduzir a complexidade de personalizar o dashboard do SmartCampus, desenvolveu-se uma documentação que detalha os componentes mais relevantes do dashboard SmartCampus UFCG, e que apresenta procedimentos comumente usados para o desenvolvimento de novas funcionalidades. O objetivo é dar suporte às equipes de desenvolvimento, reduzindo o esforço necessário para a customização de aplicações na plataforma, permitindo que isso seja feito de maneira independente.

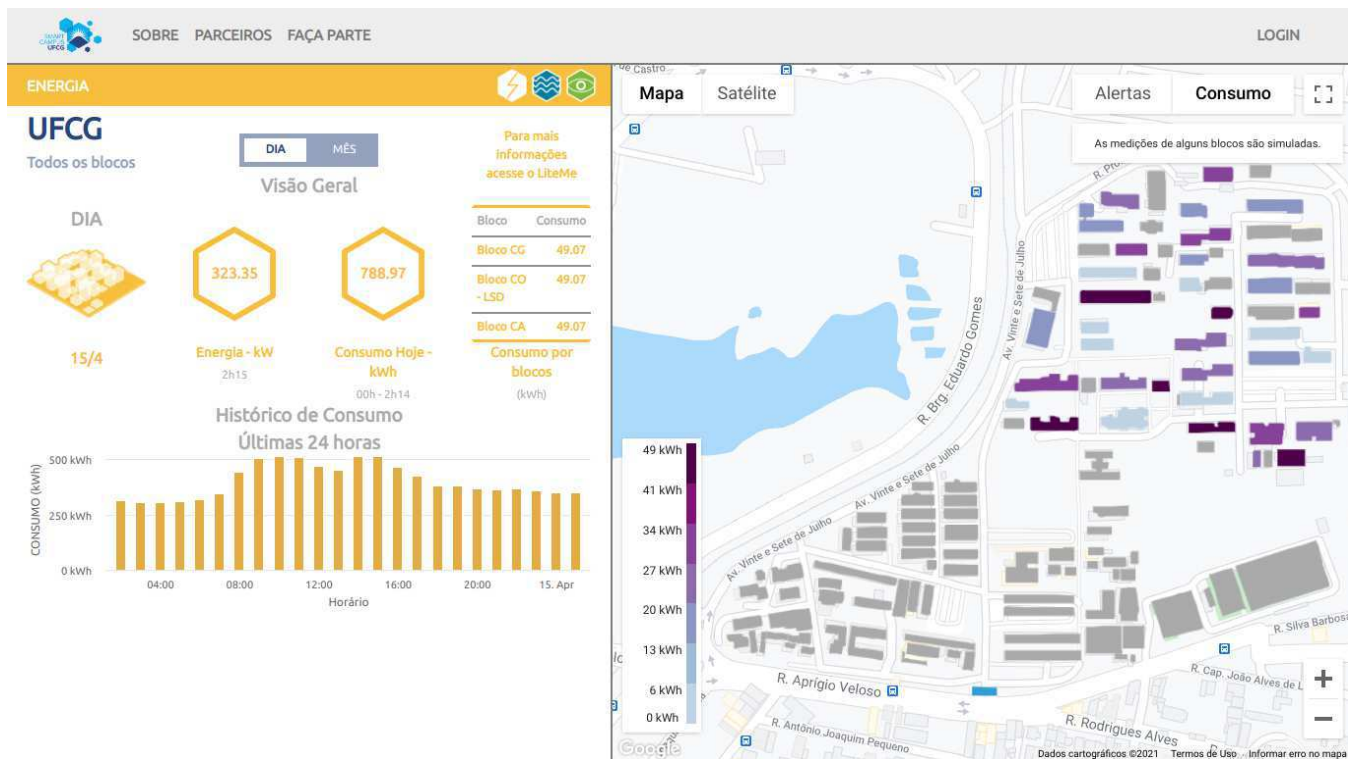


Figura 1: Aba “Energia” do Dashboard SmartCampus UFCC.

## 2.1 Descrição

Este projeto busca reduzir o esforço necessário para gerar e alterar funcionalidades do dashboard SmartCampus ao implementar um guia que facilita a personalização desta plataforma.

A interface do dashboard é dividida em: uma seção lateral esquerda, formada por abas que definem o contexto das informações expostas pelos elementos que compõem seu conteúdo; uma seção lateral direita, onde um mapa destaca os prédios relevantes de acordo com a aba selecionada; e um menu superior, o qual possui um botão de login, permitindo personalizar os dados mostrados de acordo com o perfil logado, e links para contato com o projeto SmartCampus. Esta interface é ilustrada na *Figura 1*.

Dado este contexto, para construir este guia foi definido um design minimalista para maximizar a usabilidade, ao destacar os elementos com maior valor informativo [5], e estabelecer uma aparência que esteja de acordo com a identidade visual do SmartCampus.

Como podemos observar na *Figura 2*, a interface deste guia é composta por uma área à direita, reservada à apresentação de cada página, e uma barra lateral à esquerda, que possui as seções nas quais foram agrupadas as páginas.

Podemos compreender esta documentação em três partes: a página *Welcome* (Bem-vindo), a seção *Pages* (Páginas) e a seção *Components* (Componentes).

A página *Welcome* é a primeira com a qual nos deparamos ao acessar a documentação. Nela, somos informados sobre as seções que compõem o guia e boas práticas de contribuição, nas quais somos incentivados a documentar componentes e procedimentos que foram criados ou modificados. Além disso, nos é apresentada a arquitetura desta aplicação, a qual será detalhada à posteriori.

Na seção *Pages* são mostrados tutoriais acerca de procedimentos relevantes, não-triviais e/ou com etapas distribuídas pelo dashboard, como é exposto na *Figura 2*. Dar uma ideia geral sobre eles facilita a compreensão do desenvolvedor sobre a lógica de negócio por trás da aplicação, além de permitir a reutilização dos mesmos quando necessário, reduzindo a repetição de código.

Já na seção *Components* são detalhados os *UI Components* que formam a interface de usuário do dashboard, como é mostrado na *Figura 3*. Cada página é dedicada a um determinado tipo de elemento, expondo uma breve descrição, o código utilizado, os argumentos que definem seu comportamento, e exemplos do mesmo.

## 2.2 Tecnologias utilizadas

Para o desenvolvimento desta documentação, utilizou-se o Storybook [6]: ferramenta usada na construção de Design Systems, os quais constituem conjuntos componentizados de padrões de design e código [7]. Ela possibilitou implementar uma melhor visualização das propriedades funcionais e visuais dos *UI Components*.

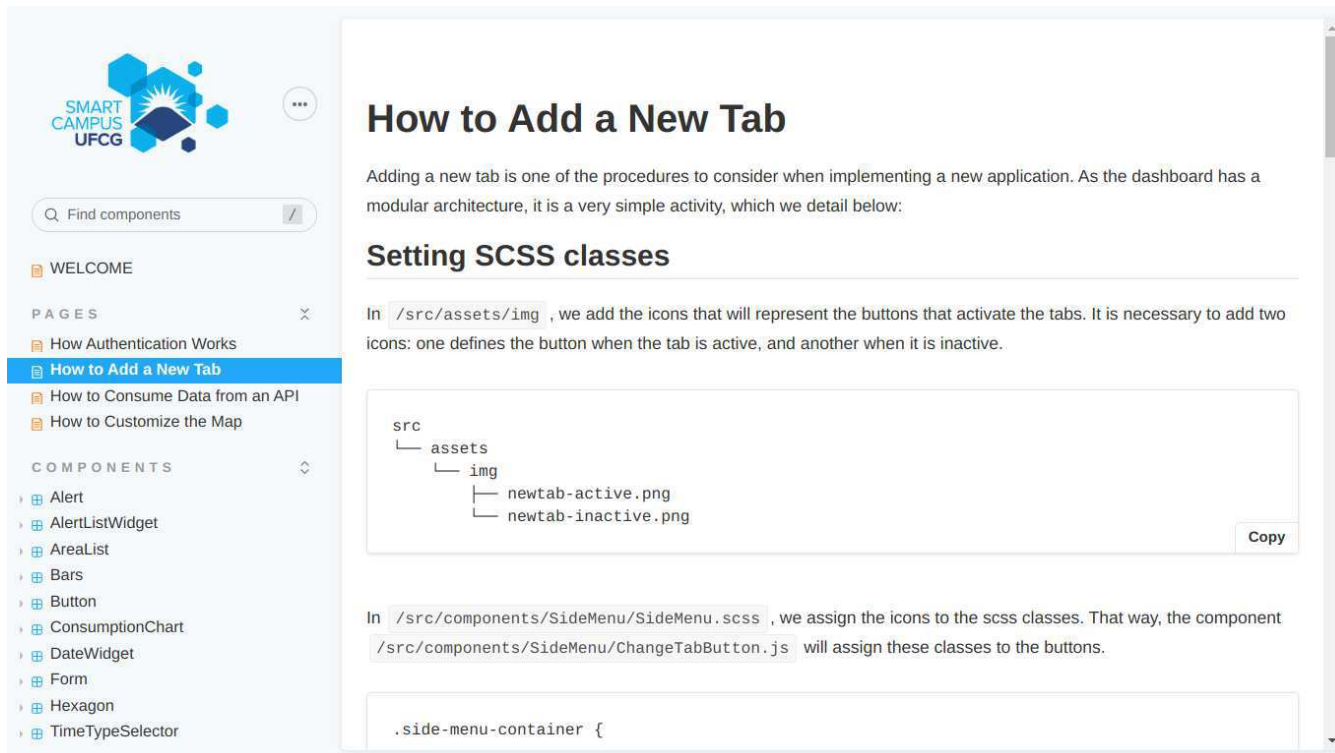


Figura 2: Página “How to Add a New Tab” na seção *Pages*.

A escolha desta tecnologia se deve ao maior domínio, devido às experiências passadas, e sua integração com bibliotecas front-end como o React [8], este último possuindo uma arquitetura baseada em componentes reutilizáveis e sendo usado no dashboard SmartCampus.

## 2.3 Arquitetura

O Storybook é instalado como uma dependência do React e, dessa forma, a implementação da documentação faz parte do código do dashboard, evoluindo com o desenvolvimento das aplicações do SmartCampus. Portanto, ela pode ser hospedada, mas para os programadores de posse do código do dashboard isto não é necessário, pois podem acessá-la localmente.

Para a construção deste artefato, os arquivos *.stories.mdx* referentes à página *Welcome* e às páginas da seção *Pages* foram alocados em um diretório *docs*. Já os arquivos *.stories.mdx* das páginas da seção *Components* foram organizadas nos diretórios dos respectivos componentes. A informação da disposição dos arquivos consta na página *Welcome*.

Esta decisão arquitetural se deve ao seguinte fato: ao contrário das demais páginas, aquelas referentes à seção *Components* estão diretamente ligadas a elementos específicos (juntamente com os arquivos *.js*, responsáveis pelo código, e *.sass*, responsáveis pelos estilos). Deste modo, a documentação de um componente faz parte do módulo que é responsável por ele.

Por fim, usamos um diretório dedicado aos arquivos de configuração do Storybook. Eles são responsáveis por importar as dependências da ferramenta, indexar os arquivos *.stories.mdx* contidos na aplicação, e definir configurações visuais de acordo com o tema escolhido. Todas estas decisões arquiteturais estão ilustradas na *Figura 4*.

## 2.4 Funcionalidades

Sucintamente, podemos destacar as seguintes funcionalidades da documentação implementada:

- **Apresentação de tutoriais:** são explicados procedimentos relevantes na aplicação. Muitos deles são realizados frequentemente e/ou envolvem o comportamento de várias entidades no sistema, o que resulta na sua não-trivialidade;
- **Apresentação de componentes:** são expostas propriedades que definem o comportamento visual dos componentes, bem como exemplos de implementação do mesmo (tanto código, como o que é renderizado);
- **Apresentação de boas práticas de desenvolvimento:** para garantir a manutenção da arquitetura da documentação, além de brevemente explicá-la, são apresentadas boas práticas que devem ser seguidas durante a customização do dashboard.



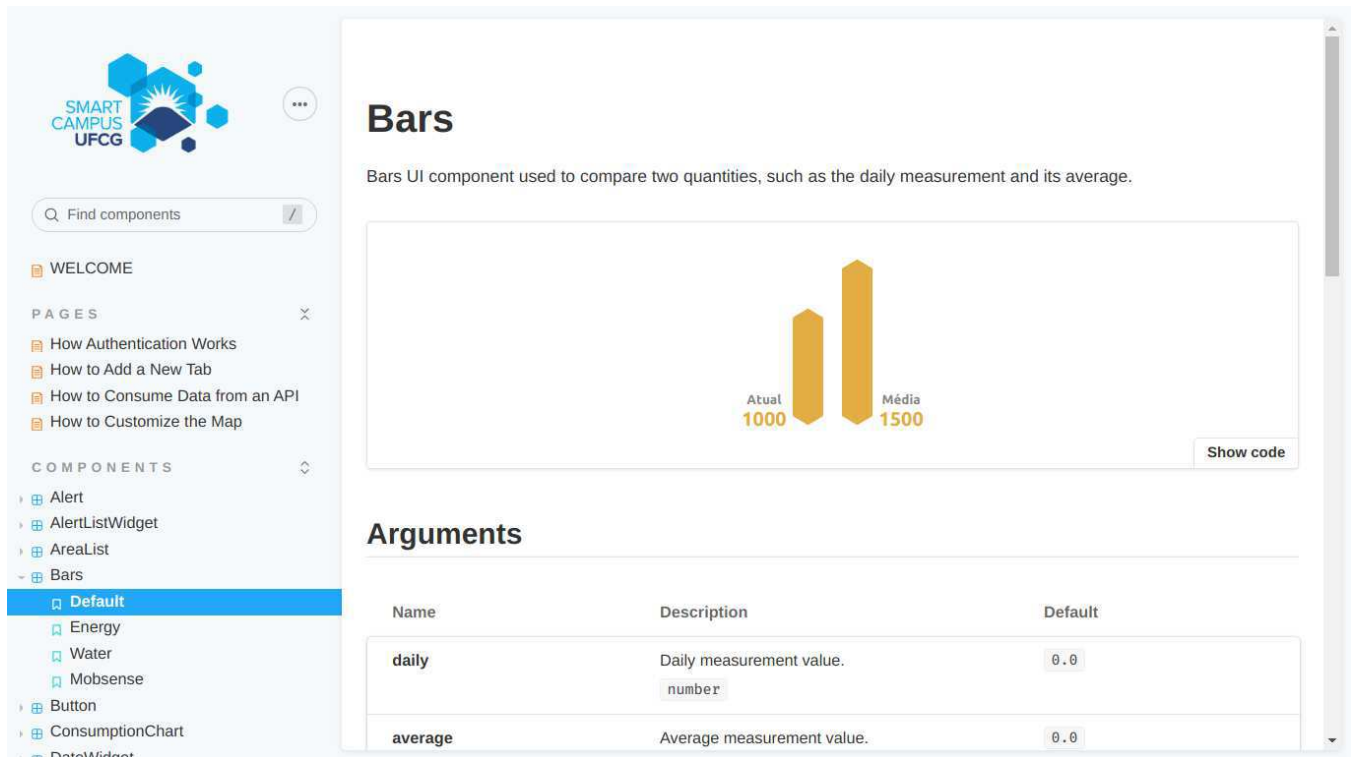


Figura 3: Página do componente “Bars”, na seção *Components*.

### 3. METODOLOGIA

Para desenvolver este projeto, primeiramente foi realizado um levantamento dos principais componentes da interface do dashboard.

Na sequência, foi feita uma investigação no código, com o intuito de identificar os principais procedimentos realizados durante a implementação de novas aplicações no SmartCampus. A familiaridade com o dashboard ajudou a reduzir o esforço demandado neste processo.

Em seguida, houve a configuração dos pacotes necessários para

integrar a biblioteca Storybook à aplicação, de forma que seja executada paralelamente à esta última.

Após isso, foi implementada uma documentação para os componentes, expondo os atributos que definem seus comportamentos e mostrando exemplos interativos, e para os procedimentos, explicitando quando são utilizados e as etapas necessárias para aplicá-los.

Por fim, foi incluído neste projeto um guia de boas práticas de uso e contribuição com a documentação, cuja manutenção auxiliará na criação e evolução de novas aplicações no dashboard.

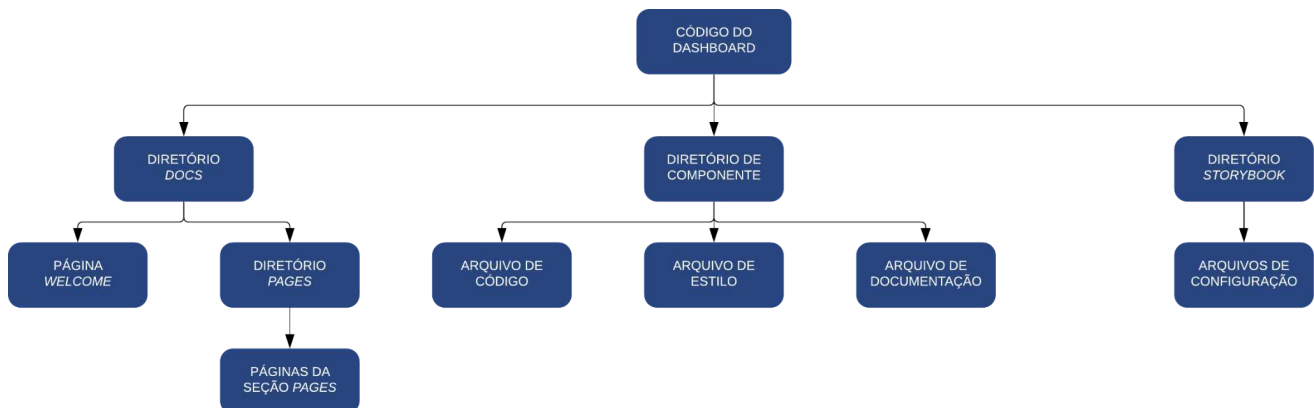


Figura 4: Ilustração da arquitetura do código da documentação.

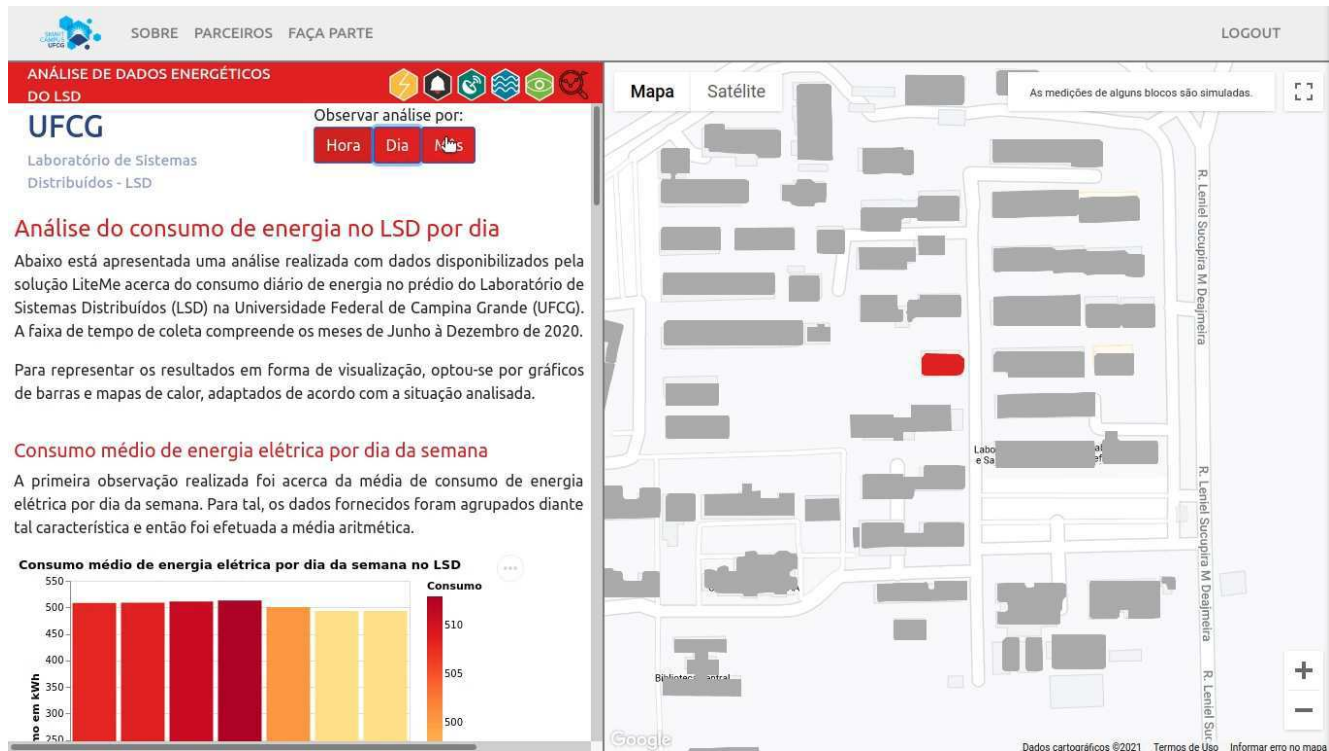


Figura 5: Nova aba de “Análise de Dados Energéticos do LSD” no Dashboard SmartCampus UFCG.

#### 4. RESULTADOS

No tocante a avaliação, durante duas semanas foi realizado um *workshop*, para o qual foram reunidas duas voluntárias que utilizaram do guia para implementar uma nova aba no dashboard SmartCampus, sendo usada como prova-de-conceito da eficácia deste artefato. Ilustrada na *Figura 5*, esta aba apresenta um relatório sobre o consumo de energia no Laboratório de Sistemas Distribuídos (LSD), situado na UFCG. As informações apresentadas nos textos e nos gráficos da aba podem ser referentes à hora, ao dia, ou ao mês atual: o intervalo de tempo é selecionado por meio de botões situados no topo da aba.

Para acompanhar o progresso da atividade citada anteriormente, foi utilizada a metodologia *Thinking Aloud* (ou Pensando em Voz Alta, em português) [9]. Desta forma, são registrados os avanços e as dificuldades que as voluntárias passaram durante o desenvolvimento desta tarefa.

Assim, percebeu-se que a documentação auxiliou não apenas na criação de uma nova aba, incluindo os componentes que foram reutilizados nela, mas também ao desenvolver uma interação com o mapa, destacando os prédios que são relevantes no contexto da aba.

Todavia, houve ressalvas quanto ao uso da documentação em determinadas situações, sendo detectadas melhorias que poderiam ser feitas, as quais estão descritas na subseção 5.3.

#### 5. EXPERIÊNCIAS

Nesta seção serão apresentadas as experiências proporcionadas e as lições aprendidas durante todo o progresso deste projeto. Será

dada ênfase ao processo de desenvolvimento, aos principais desafios e limitações encontrados, e aos trabalhos futuros.

##### 5.1 Processo de desenvolvimento

Para o desenvolvimento deste trabalho, primeiramente foram enumerados os componentes mais importantes do dashboard SmartCampus, bem como os procedimentos frequentemente realizados durante a criação de novas funcionalidades.

Em seguida, foi selecionada a tecnologia a ser utilizada: a biblioteca Storybook. A escolha se deu com base em alguns critérios como: o domínio pelo desenvolvedor; a simplicidade no uso; e a utilização desta tecnologia em projetos consolidados no mercado.

Após estes levantamentos, o desenvolvimento seguiu com a instalação das dependências necessárias do Storybook, e a implementação da documentação que detalha as principais entidades e procedimentos do dashboard, e informa boas práticas na evolução do mesmo.

Para o versionamento do código, durante todo este projeto utilizou-se da plataforma GitLab, pois como o código da documentação está atrelado ao dashboard, o mesmo repositório foi aproveitado.

##### 5.2 Principais desafios e limitações

O principal desafio foi a coleta de requisitos, a qual incluiu a listagem das principais entidades e procedimentos a serem documentados. Esta foi uma parte muito importante neste trabalho, já que uma boa coleta se torna uma boa fundação para o desenvolvimento, sendo verdade também o contrário.

Outro desafio importante foi planejar e definir uma interface que fosse amigável, conforme as boas práticas de UX (Experiência do Usuário) [10], e que facilite a visualização de informações úteis. Ao mesmo tempo, ela deveria estar de acordo com a identidade visual do dashboard SmartCampus, por estar diretamente relacionada ao mesmo.

Quanto às limitações, em razão da pandemia causada pelo vírus COVID-19, o calendário de atividades que foi planejado para o progresso deste trabalho passou por diversas alterações. Por causa disso, o desenvolvimento precisou ser realizado com prazos mais curtos, de forma que o refinamento da documentação, posterior à avaliação do mesmo, foi alocado para trabalhos futuros.

### 5.3 Trabalhos futuros

De acordo com o *feedback* das voluntárias que participaram do *workshop*, há um grande potencial para a continuação deste trabalho, além de melhorias que sigam a ideia da documentação. Para fins do escopo deste projeto, o foco foi gerar uma documentação simples, mas que auxiliasse na criação de novas funcionalidades no dashboard SmartCampus de forma eficaz.

Diante do que foi estabelecido por esta ferramenta, diversas soluções podem ser incorporadas e melhoradas, tais como:

- Adição de um *troubleshooting*, um guia de resolução de problemas frequentes, acerca do processo de instalação dos pacotes do dashboard, bem como da configuração do ambiente de desenvolvimento;
- Inclusão de recomendações de fontes de ícones prontos e/ou ferramentas que auxiliem no design de ícones que serão usados na criação de novas abas, de forma que estejam adequados à identidade visual definida.
- Acréscimo de recomendações sobre o *deploy* do dashboard, em especial acerca da manipulação de variáveis de ambiente e o manuseio de certificados ao utilizar protocolos de autenticação.

## 6. REFERÊNCIAS

- [1] Forum de Comunidades Inteligentes. What is an Intelligent Community ? 2006. <https://bit.ly/3vnUECF>.
- [2] RIENTE, Leticia. São Paulo é a cidade mais inteligente do Brasil, segundo ranking. Portal Olhar Digital. 2020. <https://bit.ly/3orfrBW>.
- [3] Página oficial do projeto SmartCampus UFCG: <https://smartcampus.ufcg.edu.br>.
- [4] Dashboard referente ao projeto SmartCampus UFCG: <https://dashboard.smartcampus.ufcg.edu.br>.
- [5] FESSENDEN, Therese. Thinking Aloud: Aesthetic and Minimalist Design (Usability Heuristic #8). Portal Nielsen Norman Group. 2021. <https://bit.ly/2R4E2Os>.
- [6] Página oficial da ferramenta Storybook, em inglês: <https://storybook.js.org>.
- [7] SERRADAS, Van. Entendendo Design Systems. Portal UX Collective. 2018. <https://bit.ly/2RXKyZR>.
- [8] Página oficial da biblioteca React, em português: <https://pt-br.reactjs.org>.

[9] NIELSEN, Jakob. Thinking Aloud: The #1 Usability Tool. Portal Nielsen Norman Group. 2012. <https://bit.ly/3e1MhVq>.

[10] FITZGERALD, Abbey. Design Consistency Guide: Best Practices for UI and UX Designers. Blog Studio by UXPin. <https://bit.ly/3xuJME5>.