




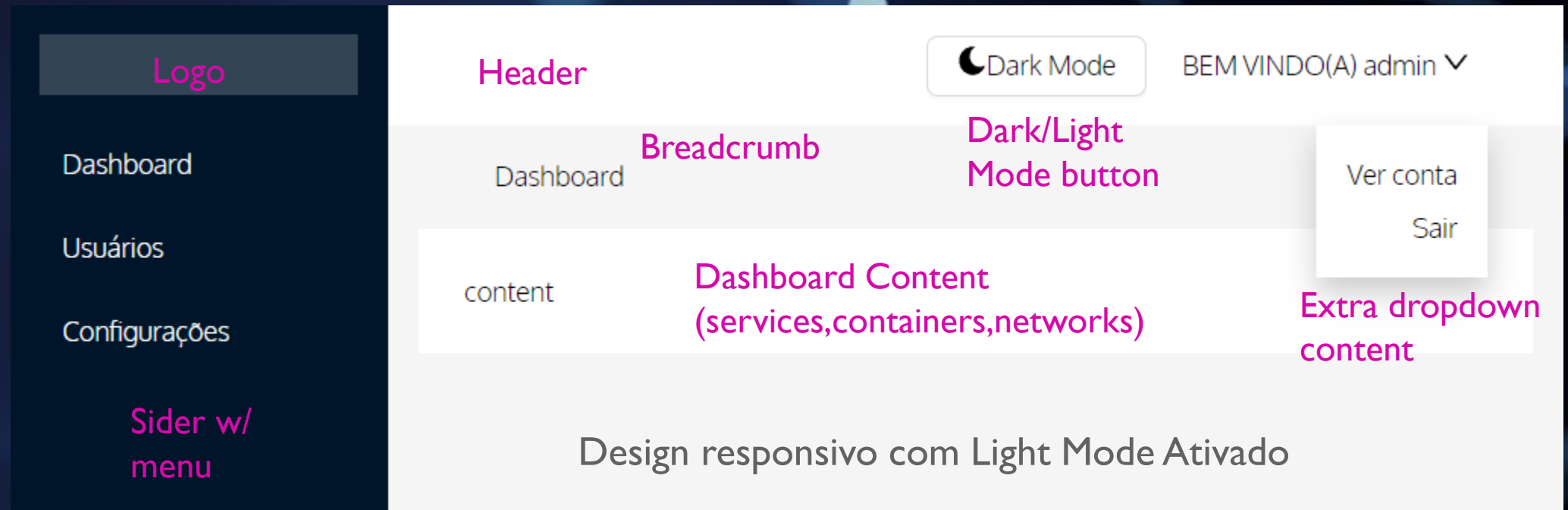
Web Interface for Container Management

A ferramenta tem por objetivo, oferecer uma interface web que possa gerenciar e organizar ambientes virtuais em forma de containers.



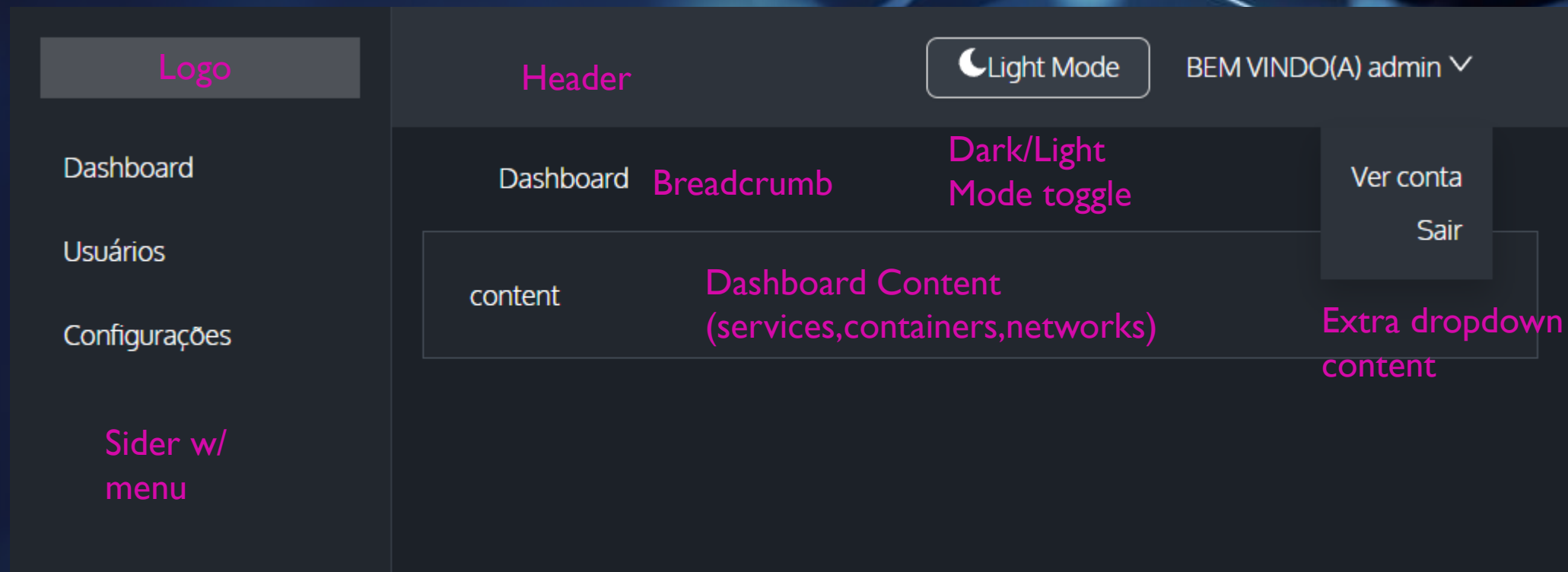
Protótipo

<https://github.com/rmvs/wicm>



Funcionalidades Mapeadas:

- Login/Logout
- Dashboard (serviços, artefatos e aplicações)
- Usuários (gerenciamento do acesso)
- Configurações (definição de variáveis de ambientes, secret keys, etc...)



Design responsive com Dark Mode ativado

Requisitos de projeto alcançados

CSS declarado em arquivo específico com utilização de variáveis para controlar o estilo.

Script modularizado em arquivos diferentes (utilizar imports,exports)

W3C Compliant

Requisitos de projeto alcançados

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://65eb-2804-29b8-5173-7dc0-e475-7447-fd7a-76df.ngrok-free.app/>

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [address](#) ▼

<https://65eb-2804-29b8-5173-7dc0-e475-7447-fd7a-76df.ngrok-free.app/>

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was utf-8.

Total execution time 413 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.3.31

Aplicação local resolvida pelo ngrok

Frameworks e ferramentas de automação de desenvolvimento utilizadas

Node + Express (Server side)

Vanilla.js (Client-side)

Webpack

Nodemon

Vanilla.js

Usado para manipular eventos e modificar estilos.

```
11 :root {
12
13   --color-button-light: #fff;
14
15   --sider-background-default: #001529;
16   --header-background-default: rgb(255, 255, 255);
17   --main-content-background-default: rgb(255, 255, 255);
18   --content-background-default: #f5f5f5;
19
20   --header-background-dark: #2d333b;
21   --sider-background-dark: #22272e;
22   --main-content-background-dark: #22272e;
23   --content-background-dark: #1c2128;
24
25   --sider-bg: var(--sider-background-default);
26   --header-bg: var(--header-background-default);
27   --main-content-bg: var(--main-content-background-default);
28   --content-bg: var(--content-background-default);
29
30
31   --sider-border-dark: 1px solid #373e47;
32   --content-border-dark: 1px solid #444c56;
33
34   --sider-border-light: transparent;
35   --content-border-light: transparent;
36
37   --sider-border: var(--sider-border-light);
38   --content-border: var(--content-border-light);
39
40   --color-dark: white;
41   --color-light: black;
42
43   --color: var(--color-light);
44 }
```

global.css

```
43 (() => {
44   document.querySelector('#dark-mode-lbl').textContent = 'Dark Mode'
45 })()
46
47 window.isDarkMode = false;
48 window.setDarkMode = () => {
49   window.isDarkMode = !window.isDarkMode
50   const theme = window.isDarkMode ? "dark" : "light"
51   document.querySelector('#dark-mode-lbl').textContent = `${window.isDarkMode ? 'Light' : 'Dark'} Mode`
52   document.documentElement.style.setProperty('--sider-bg', getComputedStyle(document.documentElement).getPropertyValue(`--sider-background-${theme}`))
53   document.documentElement.style.setProperty('--header-bg', getComputedStyle(document.documentElement).getPropertyValue(`--header-background-${theme}`))
54   document.documentElement.style.setProperty('--content-bg', getComputedStyle(document.documentElement).getPropertyValue(`--content-background-${theme}`))
55   document.documentElement.style.setProperty('--main-content-bg', getComputedStyle(document.documentElement).getPropertyValue(`--main-content-background-${theme}`))
56   document.documentElement.style.setProperty('--color', getComputedStyle(document.documentElement).getPropertyValue(`--color-${theme}`))
57   document.documentElement.style.setProperty('--sider-border', getComputedStyle(document.documentElement).getPropertyValue(`--sider-border-${theme}`))
58   document.documentElement.style.setProperty('--content-border', getComputedStyle(document.documentElement).getPropertyValue(`--content-border-${theme}`))
59   document.documentElement.style.setProperty('--button-bg-color', getComputedStyle(document.documentElement).getPropertyValue(`--button-bg-color-${theme}`))
60   document.documentElement.style.setProperty('--button-color', getComputedStyle(document.documentElement).getPropertyValue(`--button-color-${theme}`))
61   document.documentElement.style.setProperty('--dark-mode-icon-color', getComputedStyle(document.documentElement).getPropertyValue(`--dark-mode-icon-color-${theme}`))
62   document.documentElement.style.setProperty('--dropdown-bg', getComputedStyle(document.documentElement).getPropertyValue(`--dropdown-bg-color-${theme}`))
63 }
64 }
```

app.css

Webpack

Criar bundle do backend (servidor express)

Converter código ES6 para ES5 (ts-loader)

Copiar automaticamente assets para diretório de build

Transpilar código do servidor, executar em watch mode e reiniciar o servidor quando ocorrer alguma mudança usando nodemon.

Referências

- <https://expressjs.com/en/4x/api.html>
- https://validator.w3.org/#validate_by_uri+with_options
- <https://webpack.js.org/guides/getting-started/#basic-setup>
- <https://ant.design/docs/spec/introduce/>
- <https://www.flaticon.com/br/>