

**SENAI JAÚ**

**GLOBALLINK**

# **DOCUMENTAÇÃO DA PLATAFORMA**

**VERSÃO 1.0**

**25/11/2025**

# SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO.....</b>	<b>3</b>
<b>2.</b>	<b>ACESSO .....</b>	<b>3</b>
<b>3.</b>	<b>FERRAMENTAS PARA DESENVOLVIMENTO .....</b>	<b>3</b>
<b>4.</b>	<b>LINGUAGENS / FRAMEWORKS UTILIZADOS .....</b>	<b>3</b>
<b>5.</b>	<b>CORES.....</b>	<b>3</b>
<b>6.</b>	<b>INSTALAÇÃO.....</b>	<b>4</b>
<b>7.</b>	<b>ORGANIZAÇÃO DOS DIRETÓRIOS.....</b>	<b>4</b>
	<b>7.1. Componentes .....</b>	<b>4</b>
	<b>7.2. Contextos.....</b>	<b>5</b>
	<b>7.3. Hooks .....</b>	<b>5</b>
	<b>7.4. Páginas (Pages) .....</b>	<b>5</b>
	<b>7.5. App.jsx .....</b>	<b>5</b>
<b>8.</b>	<b>DEPENDÊNCIAS .....</b>	<b>6</b>
	<b>8.1. Dependências principais .....</b>	<b>6</b>
	<b>8.2. Dependências de desenvolvimento.....</b>	<b>6</b>
<b>9.</b>	<b>CONCLUSÃO.....</b>	<b>6</b>

# 1. INTRODUÇÃO

Esta documentação tem o objetivo de auxiliar os usuários e desenvolvedores a entenderem como a plataforma foi desenvolvida, desde escolha de cores até banco de dados e ferramentas de desenvolvimento.

## 2. ACESSO

O acesso a plataforma é feito por meio do *link*: [Plataforma](#).

## 3. FERRAMENTAS PARA DESENVOLVIMENTO

As ferramentas para o desenvolvimento da plataforma desde telas até banco de dados foram:

- *Figma*;
- *Visual Studio Code*;
- *Visual Studio Community*;
- *Microsoft SQL Server Management Studio (SSMS)*;
- *Wokwi (ESP32)*.

## 4. LINGUAGENS / FRAMEWORKS UTILIZADOS

As linguagens / *frameworks* utilizados foram:

- *Javascript*;
- *React Vite*;
- *Bootstrap*;
- *C#*;
- *C++*.

## 5. CORES

As cores foram escolhidas baseado na paleta de cores do *Bootstrap*:

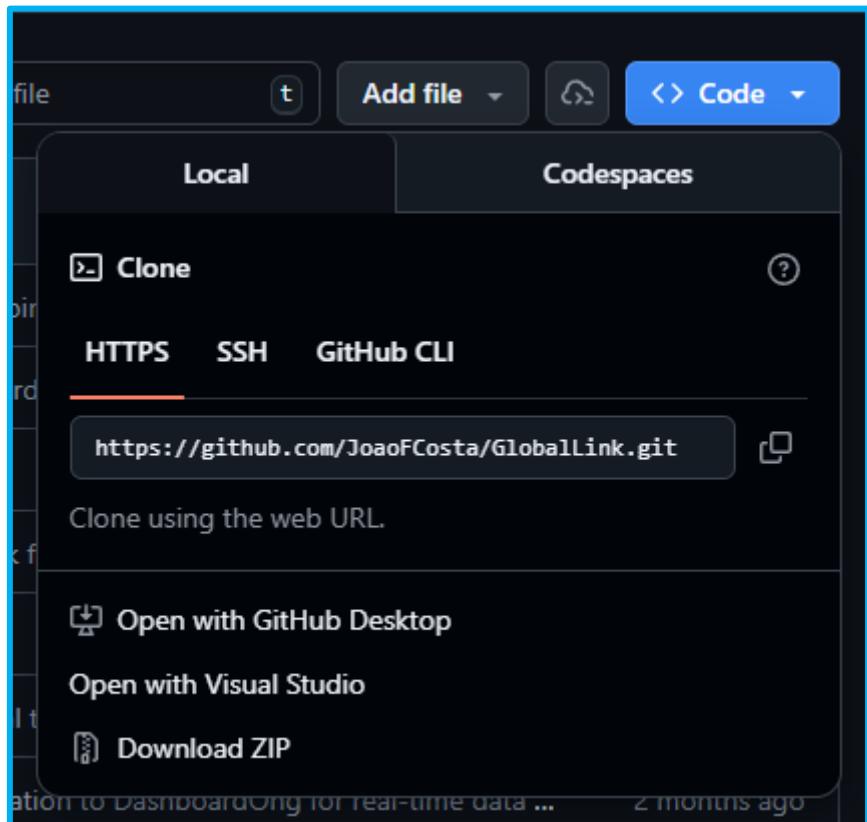
```
$primary:      $blue;
$secondary:    $gray-600;
$success:      $green;
```

```
$info:           $cyan;  
$warning:       $yellow;  
$danger:        $red;  
$light:         $gray-100;  
$dark:          $gray-900;
```

As cores mais utilizadas foram o azul (primary), verde (success) como cores principais do site, e os temas dark e light também foram utilizados.

## 6. INSTALAÇÃO

Para fazer a instalação do código é necessário acessar o repositório do GitHub ([Repositório](#)) fazer o clone do GIT ou baixar o arquivo ZIP.



## 7. ORGANIZAÇÃO DOS DIRETÓRIOS

### 7.1. Componentes

**Header:** Componente responsável pela renderização do cabeçalho da aplicação, que pode incluir navegação, título e informações do usuário.

**LogedHeader:** Componente semelhante ao Header, mas com informações específicas para usuários autenticados. (**Desconsiderado**).

## 7.2. Contextos

**ThemeContext:** Contexto responsável por gerenciar o tema da aplicação (modo claro/escuro). Fornece estado e funções para alternar entre os temas, permitindo que qualquer componente consuma essa configuração.

## 7.3. Hooks

**MQTT Hook:** Hook customizado para comunicação com brokers MQTT, permitindo a assinatura de tópicos, publicação de mensagens e gerenciamento da conexão (online/offline).

## 7.4. Páginas (Pages)

**Dashboard (empresa e ONG):** Visão geral com informações principais para empresas e ONGs, incluindo gráficos, tabelas e resumos.

**Doações:** Exibe informações sobre doações feitas ou recebidas, com formulários ou painéis de controle.

**Cadastro e Login de Empresas e ONGs:** Formulários para registro e autenticação de empresas e ONGs, incluindo possível integração com autenticação externa.

**Home:** Página inicial, provavelmente com informações gerais sobre a plataforma e suas funcionalidades.

**Necessidades:** Exibe uma lista de necessidades de empresas ou ONGs, como pedidos de doações ou ajuda urgente.

## 7.5. App.jsx

Arquivo que configura as rotas da aplicação, renderiza os componentes principais e configura o contexto global (como o ThemeContext).

## 8. DEPENDÊNCIAS

### 8.1. Dependências principais

**Bootstrap:** Framework CSS para construção de interfaces responsivas.

**Bootstrap Icons:** Conjunto de ícones SVG compatível com o Bootstrap.

**Paho MQTT:** Biblioteca para integração com brokers MQTT, permitindo comunicação em tempo real.

**React:** Biblioteca para construção de interfaces de usuário.

**React Router:** Biblioteca para roteamento de páginas, permitindo navegação sem recarregar a página.

### 8.2. Dependências de desenvolvimento

**Vite:** Ferramenta de build para desenvolvimento rápido e otimização do código.

**ESLint:** Ferramenta de linting para garantir consistência no código.

**React Types:** Tipagens para React, melhorando a segurança e autocompletar no desenvolvimento.

**Plugin React SWC:** Acelera a compilação do React no Vite.

## 9. CONCLUSÃO

O projeto **GlobalLink** utiliza uma arquitetura moderna baseada em *React*, com funcionalidades de comunicação em tempo real (MQTT) e alternância de temas. Ele oferece páginas voltadas para empresas e ONGs, como *dashboards*, cadastros, login e visualização de necessidades e doações. As dependências são otimizadas para um desenvolvimento ágil e eficiente, com *Vite* e *ESLint* como principais ferramentas de *build* e *linting*.