

Frameworks e Bibliotecas

- [Sencha Ext JS](#)
- [Quasar Framework](#)
- [Pinia](#)
- [Ionic 1](#)
- [Ícones](#)
- [Variáveis de cor](#)

Sencha Ext JS



Usado para a construção de módulos e recursos do illi

O **Ext JS 4.2** é uma biblioteca de componentes **JavaScript** de código aberto que fornece ferramentas para criar aplicativos da web escaláveis e ricos em recursos. Ele inclui um conjunto abrangente de componentes de interface do usuário, como grade de dados, formulários, árvores e gráficos, além de recursos avançados, como vinculação de dados, arquitetura de aplicativos **MVVM(Model-View-ViewModel)** e gerenciamento de estado do aplicativo. Com o **Ext JS 4.2**, os desenvolvedores podem criar aplicativos da web complexos e interativos que funcionam em todos os principais navegadores e dispositivos, incluindo desktops, tablets e smartphones.

Links: [Acesse a documentação!](#) / [Assista o curso gratuito EXTJS 4](#)

Quasar Framework



Usado para a construção de aplicações illimitar mais recentes.

O **Quasar Framework** é um framework de código aberto baseado em **Vue.js** que permite aos desenvolvedores criar aplicativos da web, aplicativos móveis e aplicativos de desktop com uma única base de código. Ele inclui um conjunto abrangente de componentes reutilizáveis de interface do usuário, como botões, formulários, layouts e gráficos, além de recursos avançados, como suporte a **PWA**, roteamento e gerenciamento de estado do aplicativo. Com o **Quasar**, os desenvolvedores podem criar aplicativos em vários formatos e plataformas com um alto grau de personalização e flexibilidade, tornando-o uma opção popular para desenvolvedores que desejam criar aplicativos modernos e escaláveis.

Links: [Acessar Documentação/Treinamento interno Arquitetura quasar/Acesso ao endpoint /](#)
[Criando Projeto Quasar Cli / Introdução Quasar Framework 1.0](#)

Pinia



Usado em conjunto com **quasar** para a construção de aplicações **illimitar** mais recentes.

Pinia é uma biblioteca oficial de gerenciamento de estado para aplicações **Vue.js**. Ela oferece uma abordagem leve, intuitiva e segura para tipagem na hora de gerenciar o estado da sua aplicação, tornando-se uma escolha popular entre desenvolvedores **Vue**.

Links: [Assistir Curso de Pinia](#) / [Acessar Documentação](#)

Ionic 1



Usado para a construção de aplicações do Gestor.

O **Ionic 1** é um framework de código aberto baseado em **AngularJS** e **Apache Cordova** que permite aos desenvolvedores criar aplicativos móveis híbridos para **iOS, Android e Windows Phone** usando tecnologias da web, como **HTML, CSS e JavaScript**. Ele inclui um conjunto abrangente de componentes de interface do usuário, como botões, menus, listas e formulários, além de recursos avançados, como roteamento, suporte a **PWA** e integração com serviços em nuvem. Com o **Ionic 1**, os desenvolvedores podem criar aplicativos móveis com uma experiência de usuário rica e fluida, combinando o melhor dos aplicativos nativos e da web. **O Ionic 1** foi muito popular em seu tempo e é ainda utilizado por algumas empresas, embora seja uma versão mais antiga do framework.

Links: [Acesso a Documentação](#)

Ícones

Os ícones podem ser utilizados em botões quando se deseja reforçar visualmente uma ação ou acionar uma função do sistema. Para agilizar o desenvolvimento, nós usamos por padrão algumas bibliotecas de ícones.

Icons8

Usado para a construção de módulos e ferramentas do illi

Os ícones do icons8 possuem diversas categorias e coleções, no **illi** nós usamos os ícones da coleção color. Para incluir um novo ícone você deve percorrer o caminho: extras -> *frontend* -> *resources* -> *css* -> *default* -> *ícones.css*

No documento ícones.css você deve incluir uma classe com o código em formato base64 da imagem.

```
.icons8-More-Than, .ux-rangemenu-gt {  
  
  background-image: url ( data:image/png;códigobase64, ) !important ;  
  
  background-size: 16px;  
  
}
```

Links: [Acessar Icons8 na coleção color](#)

Material Icons

Usado para a construção de aplicações desenvolvidas com quasar

Links: [Acessar biblioteca Materials Icons](#)

Ionicons

Usado para a construção de aplicações do illiGestor

Links: [Como usar ícones no Ionic](#) / [Acessar biblioteca de ícones Ionic](#)

Variáveis de cor

A definição de variáveis de cor é uma etapa fundamental ao trabalhar com frameworks de desenvolvimento de aplicativos, como **Quasar**, **ionic 1** e **Sencha**. Esses frameworks oferecem recursos que permitem personalizar a aparência visual dos aplicativos por meio do uso de variáveis de cor. Este guia tem como objetivo fornecer uma breve documentação sobre como definir variáveis de cor em cada um desses frameworks.

Quasar Framework

Crie um arquivo de variáveis no **Quasar**, as variáveis de cor são definidas em um arquivo chamado `quasar.variables.styl`. Se esse arquivo ainda não existir no seu projeto, crie-o na pasta `src/css/`. Defina as variáveis de cor: Abra o arquivo `quasar.variables.styl`. Declare as variáveis de cor no formato `nome-da-variavel = valor`.

<code>\$primary: #1379df;</code>	<code>\$secondary: #5cadff;</code>	<code>\$accent: #0060bf;</code>	<code>\$light: #f5f9ff;</code>	<code>\$dark: #121314;</code>
<code>\$positive: #21ba45;</code>	<code>\$negative: #c10015;</code>	<code>\$warning: #f2c037;</code>	<code>\$info: #0060bf;</code>	

Links: [Saiba mais acessando a documentação do Quasar](#)

Ionic

Crie um arquivo de variáveis: No **ionic 1**, as variáveis de cor são definidas em um arquivo chamado `ionic.app.scss`. Se esse arquivo ainda não existir no seu projeto, crie-o na pasta raiz do seu projeto **ionic**. Defina as variáveis de cor: Abra o arquivo `ionic.app.scss`. Declare as variáveis de cor no formato `$nome-da-variavel: valor`.

<code>\$light: #FFFFFF;</code>	<code>\$stable: #F7F7F7;</code>	<code>\$positive: #0076EF;</code>	<code>\$balanced: #21C65B;</code>
<code>\$energized: #FFC029;</code>	<code>\$assertive: #F43A36;</code>	<code>\$royal: #7662E3;</code>	<code>\$dark: #3C3C3C;</code>

Links: [Saiba mais acessando a documentação do Ionic](#)

Sencha 4

Crie um arquivo de variáveis: No **Sencha**, as variáveis de cor são definidas em um arquivo chamado `variables.scss`. Se esse arquivo ainda não existir no seu projeto, crie-o na pasta `sass/` do seu projeto **Sencha**. Defina as variáveis de cor: Abra o arquivo `variables.scss`. Declare as variáveis de cor no formato `$nome-da-variavel: valor`,

\$primary-color: #ff0000;

\$secondary-color: #00ff00;

Links: Saiba mais acessando a documentação do Sencha 4