

Metodologia de Criação de Componentes CSS

BEM – Block Element and Modifier

BEM – Block Element Modifier

BEM — is a methodology that helps you to create reusable components and code sharing in front-end development

Learn why it's better

```
1 #opinions_box h1 {  
2   margin: 0 0 8px 0;  
3   text-align: center;  
4 }  
5  
6 #opinions_box {  
7   p.more_pp {  
8     a {  
9       text-decoration: underline;  
10    }  
11  }  
12  
13  input[type="text"] {  
14    border: 1px solid #ccc!important;  
15  }  
16 }
```

Classic CSS

```
1 .opinions_box {  
2   margin: 0 0 8px 0;  
3   text-align: center;  
4  
5   &__view-more {  
6     text-decoration: underline;  
7   }  
8  
9   &__text-input {  
10    border: 1px solid #ccc;  
11  }  
12  
13  &--is-inactive {  
14    color: gray;  
15  }  
16 }
```

BEM

Easy

To use BEM, you only need to employ BEM's naming convention.

Modular

Independent blocks and CSS selectors make your code reusable and modular.

Flexible

Using BEM, methodologies and tools can be recomposed and configured the way you like.

BEM – **Block** Element Modifier

Blocks, Elements and Modifiers

Block

Standalone entity that is meaningful on its own.

Examples

``header`, `container`, `menu`, `checkbox`, `input``

Element

A part of a block that has no standalone meaning and is semantically tied to its block.

Examples

``menu item`, `list item`, `checkbox caption`, `header title``

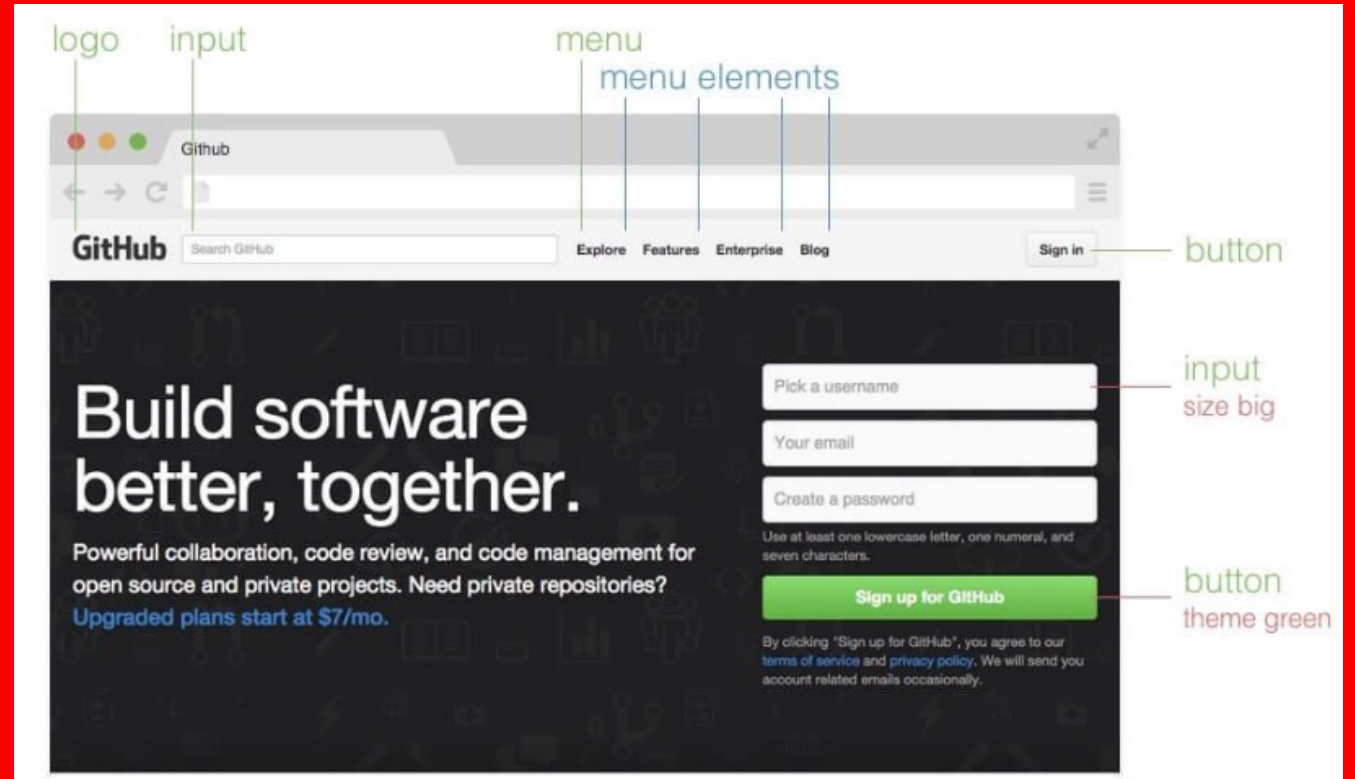
Modifier

A flag on a block or element. Use them to change appearance or behavior.

Examples

``disabled`, `highlighted`, `checked`, `fixed`, `size big`, `color yellow``

BEM – Block Element Modifier



BEM – Block Element Modifier

CSS

```
.button {  
  display: inline-block;  
  border-radius: 3px;  
  padding: 7px 12px;  
  border: 1px solid #D5D5D5;  
  background-image: linear-gradient(#EEE, #DDD);  
  font: 700 13px/18px Helvetica, arial;  
}  
  
.button--state-success {  
  color: #FFF;  
  background: #569E3D linear-gradient(#79D858, #569E3D) repeat-x;  
  border-color: #4A993E;  
}  
  
.button--state-danger {  
  color: #900;  
}
```

HTML

```
<button class="button">  
  Normal button  
</button>  
<button class="button button--state-success">  
  Success button  
</button>  
<button class="button button--state-danger">  
  Danger button  
</button>
```

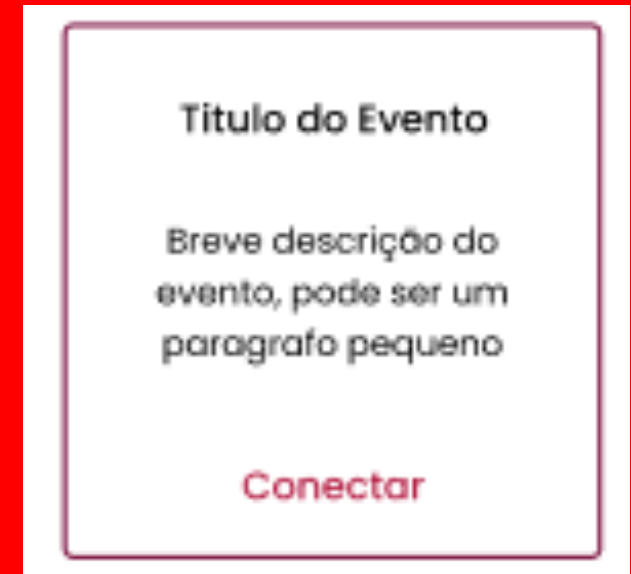


Criaremos os seguintes componentes

Estes componentes serão utilizados em nossos projetos dentro do React



Title



Card de Evento