

Programação Móvel

Componentes Ionic

Prof. Dr. Diego R. Antunes

drantunes@utfpr.edu.br

Departamento de Computação
Universidade Tecnológica Federal do Paraná

Ambiente de Desenvolvimento

Erros de Instalação

Copiar a mensagem de erro e colar na busca no Google.

Geralmente são erros comuns que outras pessoas já passaram e publicaram soluções em fóruns específicos

Outros Templates Ionic

Aplicação em Branco

Você pode criar um projeto ionic usando o modelo “em branco”: *ionic start meuProjeto **blank** --v2*

Aplicação com Menu Lateral

Você pode criar um projeto ionic usando o modelo “side menu”, que consiste de um menu lateral acionado via ícone na barra de navegação:

*ionic start meuProjeto **sidemenu** --v2*

☰	Page 2	
🍺	Item 1	This is item #1
⚙️	Item 2	This is item #2
🏈	Item 3	This is item #3
⚙️	Item 4	This is item #4
📌	Item 5	This is item #5
🏈	Item 6	This is item #6
🔗	Item 7	This is item #7
🔗	Item 8	This is item #8
🚂	Item 9	This is item #9
📌	Item 10	This is item #10

Menu	
Page uno	m #1
Page dos	m #2
	m #3
	m #4
	m #5
	m #6
	m #7
	m #8
	m #9
	m #10

Aplicação Padrão (Tabs)

Se você não informar o tipo de aplicação, o ionic carrega o modelo padrão de tabs (abas).

Cores e Variáveis

Cores e Variáveis CSS

Você pode alterar as cores padrão que o ionic oferece ou cadastrar novas cores personalizadas.

Cores e Variáveis CSS

Para isso você deve acessar a pasta *app/theme* e abrir o arquivo *app.variables.scss*.

```
$colors: (  
  primary: #387ef5,  
  secondary: #32db64,  
  danger: #f53d3d,  
  light: #f4f4f4,  
  dark: #222,  
  favorite: #69BB7B  
);
```

Cores e Variáveis CSS

*Para prova de conceito, vamos adicionar uma cor personalizada chamada **arquitetura**:*

```
$colors: (  
  primary: #387ef5,  
  secondary: #32db64,  
  danger: #f53d3d,  
  light: #f4f4f4,  
  dark: #222,  
  favorite: #69BB7B,  
  arquitetura: #410f82  
);
```

Cores e Variáveis CSS

Para usar este novo padrão de cor, basta chamar a diretiva arquitetura em um botão, ícone, toolbar, etc.

Exemplo...

```
<ion-header>
  <ion-navbar arquitetura>
    <button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Page 2</ion-title>
  </ion-navbar>
</ion-header>
```

```
<ion-content padding>
```

```
  <button arquitetura block>Cor Personalizada!</button>
  <ion-icon name="people" arquitetura></ion-icon>
</ion-content>
```



Page 2

COR PERSONALIZADA!



Ionic Generator

Ionic Generator

*O Ionic CLI fornece um comando adicional para ajudar a criar uma nova página. No Terminal você pode executar o comando **ionic g page minhaPagina***

Ionic Generator

*Assim que você executar, o ionic irá gerar uma nova pasta em app, chamada **minha-pagina**:*

```
→ testemenu ionic g page minhaPagina
```

```
✓ Create app/pages/minha-pagina/minha-pagina.html
```

```
✓ Create app/pages/minha-pagina/minha-pagina.scss
```

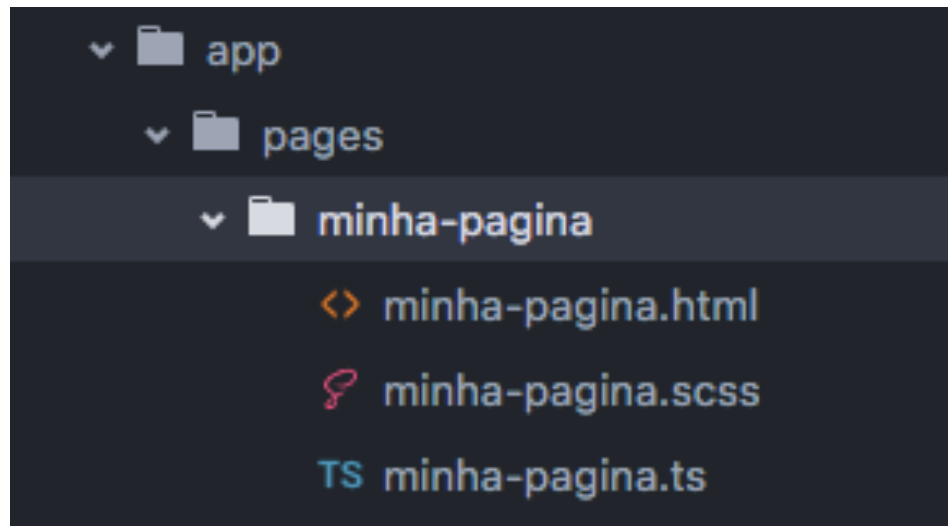
```
✓ Create app/pages/minha-pagina/minha-pagina.ts
```

Don't forget to add an import for `minha-pagina.scss` in `app/themes/app.core.scss`:

```
@import "../pages/minha-pagina/minha-pagina.scss";
```

Ionic Generator

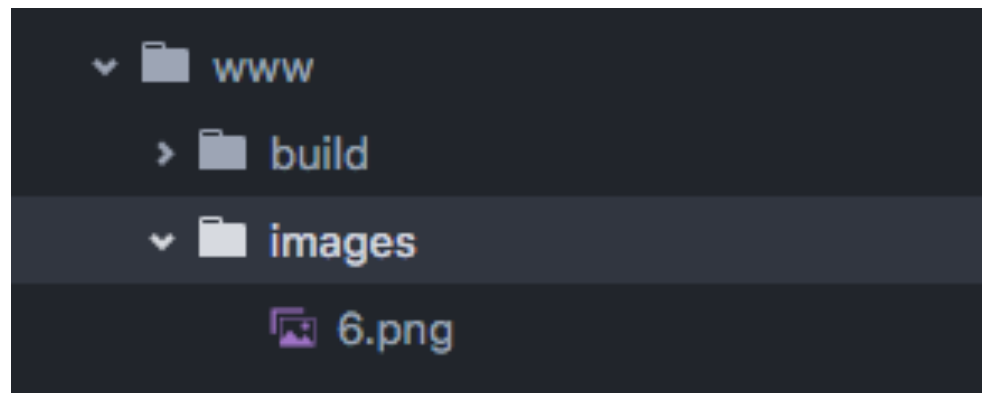
Então, basta você começar a trabalhar e realizar as alterações necessárias.



Local para Imagens

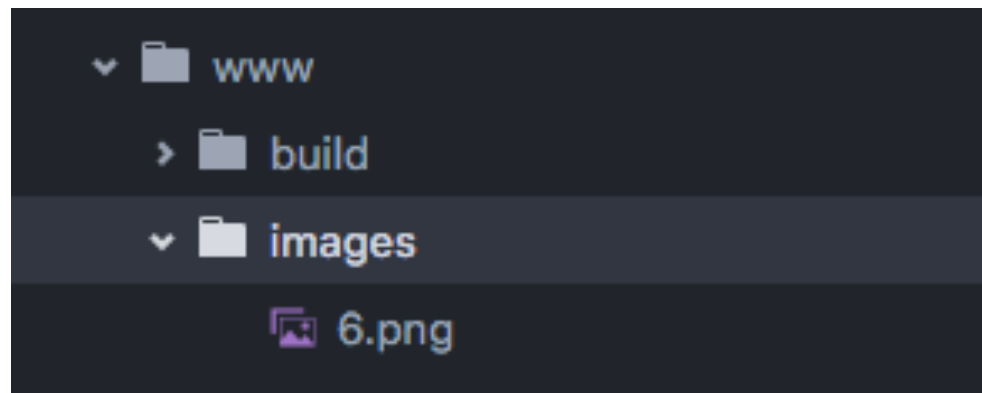
Imagens

*Para não ter problemas durante a compilação, uma sugestão é criar uma pasta chamada **images** dentro da pasta **www**, na raiz.*



Imagens

Para referenciar esta imagem no HTML de uma página, você deve usar o endereço ``



Componentes Visuais

Badge

Funciona para destacar items tais como: números, contadores, notificações, etc.

<http://ionicframework.com/docs/v2/components/#badges>

Label



Amigos

1450

```
<ion-item>
  <ion-icon name="logo-facebook" item-left arquitetura></ion-icon>
  Amigos
  <ion-badge item-right secondary>1450</ion-badge>
</ion-item>
```


Radio

Funciona como um checkbox, retornando um valor booleano. Em um radio-group somente um radio pode ser selecionado. Exemplo...

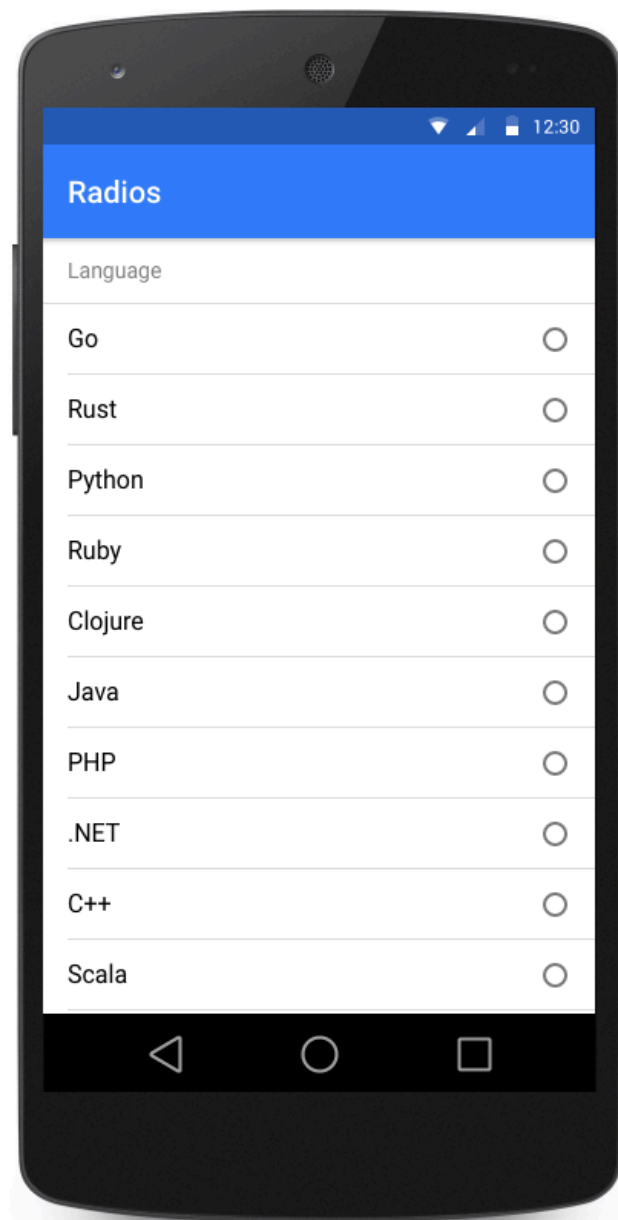
<http://ionicframework.com/docs/v2/components/#radio>

Radio

```
<ion-list radio-group>
  <ion-list-header>
    Language
  </ion-list-header>

  <ion-item>
    <ion-label>Go</ion-label>
    <ion-radio checked="true" value="go"></ion-radio>
  </ion-item>

  <ion-item>
    <ion-label>Rust</ion-label>
    <ion-radio value="rust"></ion-radio>
  </ion-item>
```



Toggle

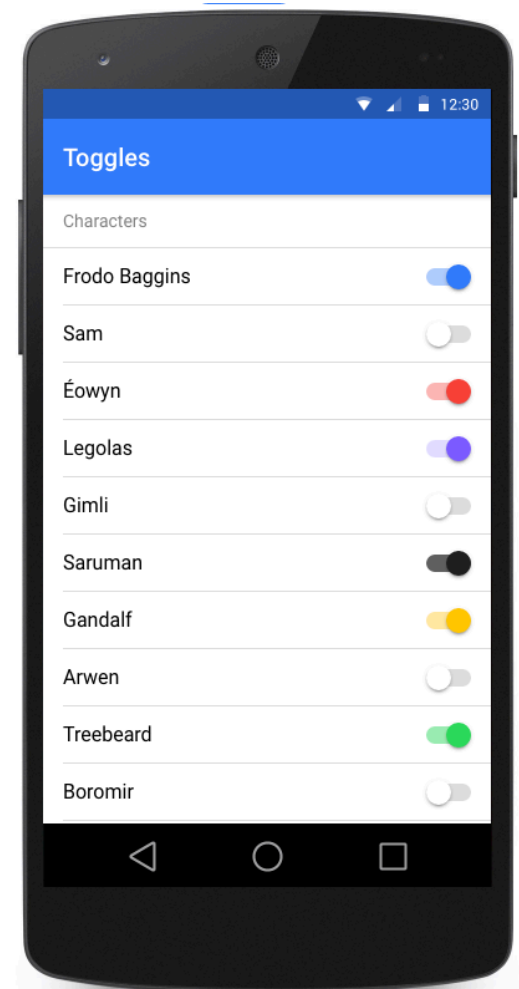
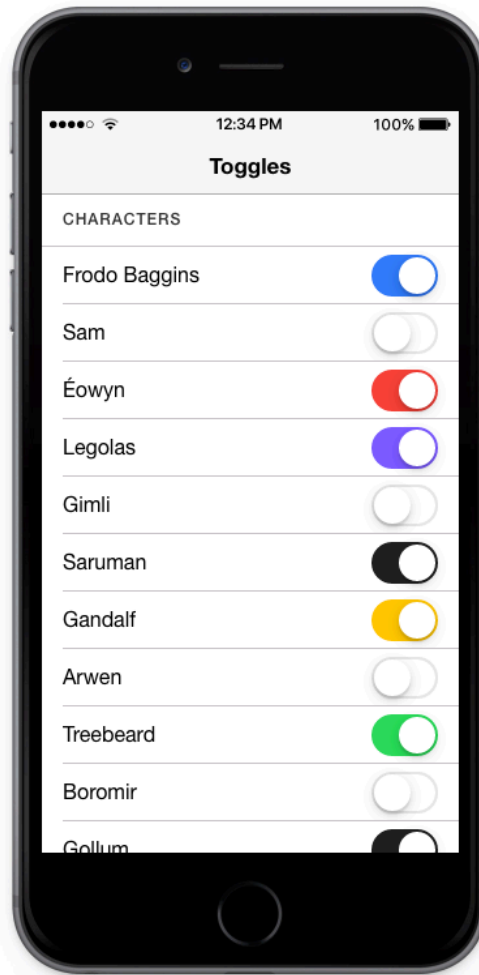
Funciona como um interruptor que retorna on ou off.

Cada plataforma tem um estilo de toggle, conforme o exemplo a seguir.

<http://ionicframework.com/docs/v2/components/#toggle>

Toggle

```
<ion-item>  
  <ion-label> Sam</ion-label>  
  <ion-toggle disabled checked="false"></ion-toggle>  
</ion-item>
```



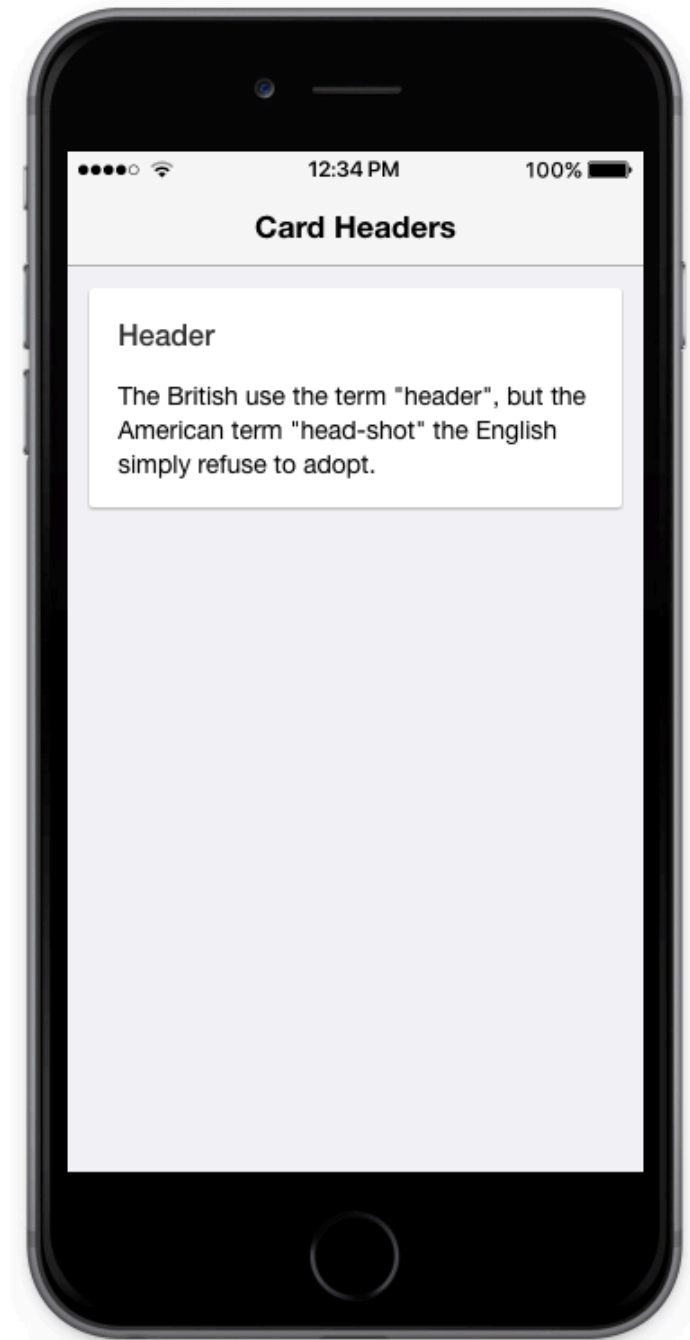
Cards

São componentes usados para exibir informações personalizadas sobre um determinado objeto: perfil de usuário, mapa, tarefas, produto, entre outros.

<http://ionicframework.com/docs/v2/components/#cards>

Cards

```
<ion-card>
  <ion-card-header>
    Header
  </ion-card-header>
  <ion-card-content>
    The British use the term "header",
  </ion-card-content>
</ion-card>
```



Select

Componente de combobox para selecionar itens em uma lista. Uma opção escolhida será salva automaticamente em um model do Angular, usando a diretiva [(ngModel)] = “nome_da_variavel”

<http://ionicframework.com/docs/v2/components/#select>

Neste exemplo, temos um *ion-select* e estamos mapeando a opção escolhida pelo usuário na variável ou propriedade chamada *gaming*

```
<ion-list>
  <ion-item>
    <ion-label>Games</ion-label>
    <ion-select [(ngModel)]="gaming">
      <ion-option value="pokemon">Pokemon Go</ion-option>
      <ion-option value="angry">Angry Birds</ion-option>
    </ion-select>
  </ion-item>
</ion-list>
```


Interação entre HTML e JS

Propriedades e Eventos

O Ionic usa os mecanismos do Angular para realizar o mapeamento de propriedades entre o JavaScript e o HTML.

Propriedades e Eventos

Da mesma forma, é possível utilizar eventos nos componentes HTML para disparar funções no JS.

Mostrar uma variável no HTML

Para renderizar uma variável do JS no HTML você deve usar o conceito de interpolação, ou seja: `{{ variavel }}`

Mostrar uma variável no HTML

Ao usar esta notação o Angular substituirá esta tag pelo conteúdo da variável. É importante notar que você pode usar este método em diversos elementos como botões, listas, texto, etc.

Mostrar uma variável no HTML

No JS definimos a variável titulo e damos um valor padrão no construtor.

```
8  export class MinhaPagina {  
9  
10     protected titulo;  
11  
12     constructor(private navCtrl: NavController) {  
13  
14         this.titulo = "Página";  
15     }  
16 }  
17 }
```

Mostrar uma variável no HTML

Então no HTML nós usamos a interpolação `{{ titulo }}`

```
1 <ion-header>
2
3   <ion-navbar>
4     <ion-title>{{ titulo }}</ion-title>
5   </ion-navbar>
6
7 </ion-header>
```

Página

Exemplo

Vamos alterar o título usando um input, um botão e um método no JavaScript.

Primeiro, criamos uma função chamada mudarTitulo, que faz a atribuição e sobrescreve o título da página.

```
8  export class MinhaPagina {
9
10     protected titulo;
11     protected novo_valor;
12
13     constructor(private navCtrl: NavController) {
14         this.titulo = "Página";
15     }
16
17     mudarTitulo(){
18         this.titulo = this.novo_valor;
19     }
20 }
```

No HTML, nós criamos um campo texto onde o model deste campo é igual ao *novο_valor*.

```
1  <ion-header>
2    <ion-navbar>
3      <ion-title>{{ titulo }}</ion-title>
4    </ion-navbar>
5  </ion-header>
6
7  <ion-content padding>
8
9    <ion-item>
10     <ion-label floating>Informe o título</ion-label>
11     <ion-input type="text" [(ngModel)]="novο_valor"></ion-input>
12   </ion-item>
13
14   <button block (click)="mudarTitulo()">Alterar</button>
15
```

E adicionamos um botão para chamar a função `alterarTitulo`.

```
1  <ion-header>
2    <ion-navbar>
3      <ion-title>{{ titulo }}</ion-title>
4    </ion-navbar>
5  </ion-header>
6
7  <ion-content padding>
8
9    <ion-item>
10     <ion-label floating>Informe o título</ion-label>
11     <ion-input type="text" [(ngModel)]="novo_valor"></ion-input>
12   </ion-item>
13
14   <button block (click)="mudarTitulo()">Alterar</button>
```

No Angular, os eventos JS são mapeados entre parentesis, por exemplo, (click) (keyup) (keyup.enter)...

```
1  <ion-header>
2    <ion-navbar>
3      <ion-title>{{ titulo }}</ion-title>
4    </ion-navbar>
5  </ion-header>
6
7  <ion-content padding>
8
9    <ion-item>
10      <ion-label floating>Informe o título</ion-label>
11      <ion-input type="text" [(ngModel)]="novo_valor"></ion-input>
12    </ion-item>
13
14    <button block (click)="mudarTitulo()">Alterar</button>
15  </ion-content>
```

```
<ion-item>
  <ion-label floating>Informe o título</ion-label>
  <ion-input type="text" [(ngModel)]="novo_valor" (keyup.enter)="mudarTitulo()"></ion-input>
</ion-item>
```

*Por exemplo, ao invés de utilizarmos um botão para a chamada da função, poderíamos utilizar um evento de **keyup** com o pressionar da tecla **enter**.*

```
<ion-item>
  <ion-label floating>Informe o título</ion-label>
  <ion-input type="text" [(ngModel)]="novo_valor" (keyup.enter)="mudarTitulo()"></ion-input>
</ion-item>
```

PERGUNTA

*O que acontece se substituirmos o nome da variável
(model) por titulo?*

*ngFor

Diretiva especial do Angular que permite realizar iterações sobre listas de items. Por exemplo, pode ser utilizado para carregar uma lista dinâmica de disciplinas.

*ngFor

Primeiramente, precisamos de uma variável para armazenar os dados da lista de disciplinas. Então inicializamos os dados da variável no construtor.

*ngFor

```
7  export class Page1 {  
8      protected disciplinas;  
9  
10     constructor(public navCtrl: NavController) {  
11         this.inicializar();  
12     }  
13  
14     inicializar(){  
15         this.disciplinas = [  
16             "Programação Móvel",  
17             "Programação Web",  
18             "Arquitetura de Software",  
19             "Interação Humano-Computador"  
20         ];  
21     }
```

*ngFor

```
7  export class Page1 {
8      protected disciplinas;
9
10     constructor(public navCtrl: NavController) {
11         this.inicializar();
12     }
13
14     inicializar(){
15         this.disciplinas = [
16             "Programação Móvel",
17             "Programação Web",
18             "Arquitetura de Software",
19             "Interação Humano-Computador"
20         ];
21     }
```

*ngFor

```
7  export class Page1 {
8      protected disciplinas;
9
10     constructor(public navCtrl: NavController) {
11         this.inicializar();
12     }
13
14     inicializar(){
15         this.disciplinas = [
16             "Programação Móvel",
17             "Programação Web",
18             "Arquitetura de Software",
19             "Interação Humano-Computador"
20         ];
21     }
```

*ngFor

*Então no HTML, no elemento que queremos repetir, nos usamos o comando *ngFor na seguinte sintaxe:*

<elemento *ngFor="let disciplina of **disciplinas**">

 {{ disciplina }}

</elemento>

*ngFor



Programação Móvel

Programação Web

Arquitetura de Software

Interação Humano-Computador

```
14
15 <ion-list>
16   <ion-item *ngFor="let disciplina of disciplinas">
17     {{ disciplina }}
18   </ion-item>
19 </ion-list>
20
```

*nglf

Permite mostrar ou esconder um elemento no HTML de acordo com uma condição. A seguinte sintaxe é utilizada:

`<elemento *nglf="disciplinas.length == 4">`

Mostra a mensagem se a condição for verdadeira

`</elemento>`

*ngIf

≡ Page Uno
Programação Móvel
Programação Web
Arquitetura de Software
Interação Humano-Computador
Só mostra esse texto se lista tem 4 itens

20

21

```
<p padding *ngIf="disciplinas.length == 4" danger>
```

22

```
Só mostra esse texto se lista tem 4 itens
```

23

```
</p>
```

Lista 1 (Parte 2): Moodle

Os exercícios da Lista 1 podem ser apresentados até 02/09