

FIAP

Abstract flowing lines in shades of pink and blue, resembling smoke or liquid, set against a dark background.

OPTIMIZATION TECHNOLOGY

2º
semestre
2017

www.fiap.com.br

RENATO BORTOLIN
renatoboschini@gmail.com

1. AngularJs - Formulário

Os formulários são o suporte principal das aplicações empresariais. Você usa formulários para fazer login, enviar uma solicitação de ajuda, fazer um pedido, reservar um voo, agendar uma reunião e executar inúmeras outras tarefas de entrada de dados.

Ao desenvolver um formulário, é importante criar uma experiência de entrada de dados que guie o usuário eficiente e eficazmente através do fluxo de trabalho.

O desenvolvimento de formulários requer habilidades de design (que estão fora do escopo para esta página), bem como suporte de estrutura para ligação de dados bidirecionais, rastreamento de alterações, validação e tratamento de erros, que você aprenderá nesta página.

Esta material mostra como criar um formulário simples a partir do zero.

Pizzaria Xadrez Home Pizzaria Produtos Sobre

Fundada pelo piracicabano João Malagueta, a Venite se estabeleceu no coração da Vila Mariana há mais de 60 anos. Com grande sucesso, atraiu uma clientela fiel desde o início, sendo a pizzaria queridinha do chef Alex Atala (CLIQUE AQUI e confira a matéria completa). E hoje, a Venite está de volta pelas mãos de João Carlos Malaguetta, filho de João Malagueta. João Carlos passou os últimos 20 anos da vida do pai junto com ele na pizzaria. Eram apenas os dois que preparavam a massa, montavam e assavam as pizzas. Agora, o legado continua mantendo a mesma tradição e os sabores inesquecíveis que conquistaram o público paulistano. No momento, apenas com atendimento delivery, a Venite oferece mais de 28 opções assadas em forno a gás, com o carinho do toque caseiro de quem é e sempre foi apaixonado pela arte de fazer pizzas.

Formulário

nome

e-mail

Telefone

Assunto

Dúvidas

Mensagem

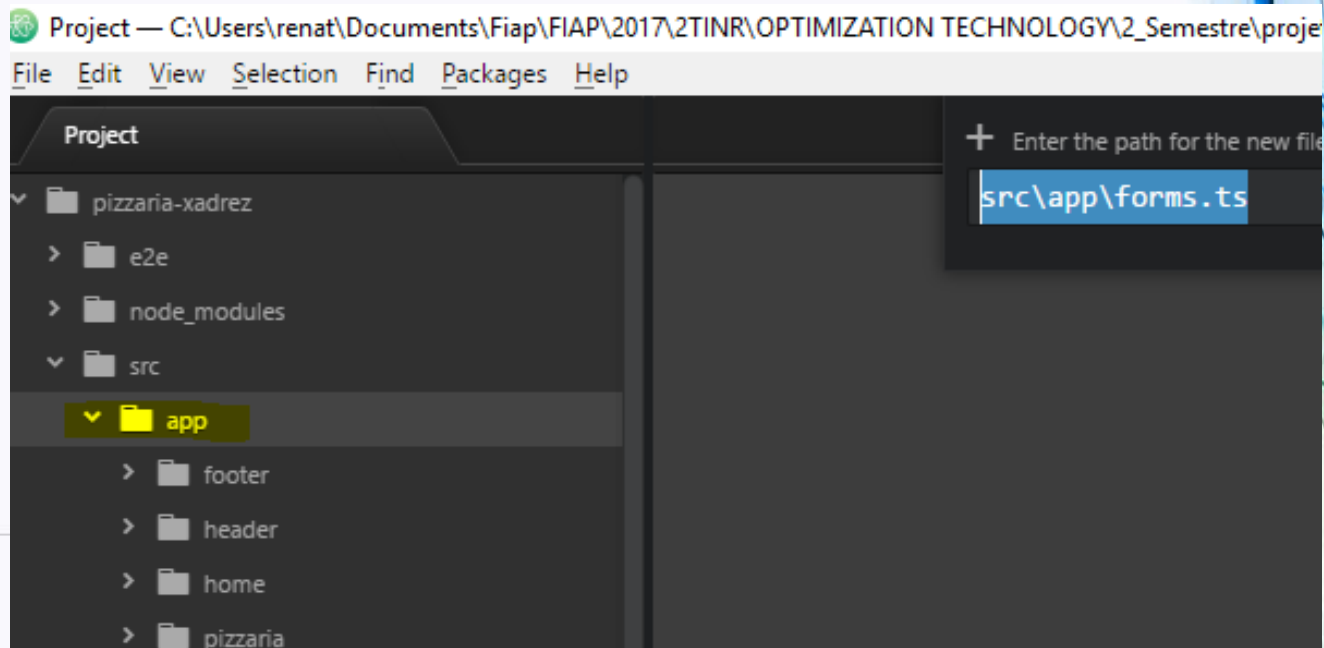
AngularJs - Formulário



Vamos criar uma class **forms.ts**

À medida que os usuários inserem dados do formulário, você irá capturar suas alterações e atualizar uma instância de um modelo. Você não pode definir o formulário até que você saiba como é o modelo.

Um modelo pode ser tão simples como um "saco de propriedade" que contém fatos sobre uma coisa de importantes da aplicação. Isso descreve bem a classe FORMS com seus seis campos obrigatórios (id, nome, email, assunto e mensagem) e um campo opcional (telefone).



AngularJs - Formulário



Vamos criar uma class **forms.ts**

No diretório do aplicativo, crie o seguinte arquivo com o conteúdo fornecido:

O telefone é opcional, então o construtor permite que você omita isso; anote o ponto de interrogação (?) em telefone.

```
forms.ts
1  export class Forms {
2
3      constructor(
4          public id: number,
5          public nome: string,
6          public email: string,
7          public assunto: string,
8          public mensagem: string,
9          public telefone?: string
10
11      ) { }
12
13  }
14
```

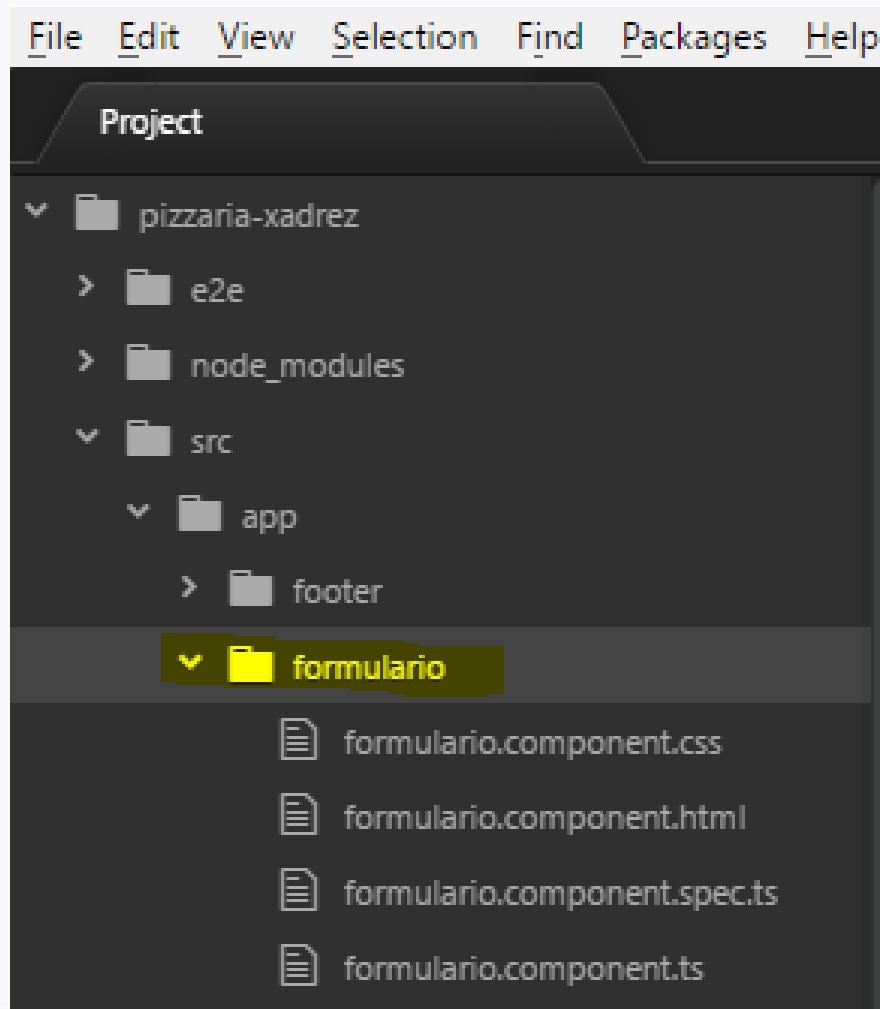


AngularJs - Formulário



Vamos criar um novo componente para o seu projeto

ng g c formulario



Vamos criar um novo componente para o seu projeto

ng g c formulario

AngularJs - Formulário



No arquivo **formulario.component.ts**, vamos importar o forms

```
forms.ts      x  formulario.component.ts  x  app.module.ts
1  import { Component, OnInit } from '@angular/core';
2  import { Forms } from '../forms';
3
4  @Component({
5    selector: 'app-formulario',
6    templateUrl: './formulario.component.html',
7    styleUrls: ['./formulario.component.css']
8  })
9  export class FormularioComponent {
10
11    constructor() { }
```

AngularJs - Formulário



Vamos agora importar os dados para um formulário

```
export class FormularioComponent {  
  
  constructor() { }  
  
  ngOnInit() { }  
  
  assuntos = ['Dúvidas', 'Prontos', 'Reclamações', 'Elogios'];  
  
  model = new Forms(18, 'Renato Bortolin', 'renato@gmail.com', this.assuntos[2], 'bla bla bla', '999999');  
  
  submitted = false;  
  
  onSubmit() { this.submitted = true; }  
  
  get diagnostic() { return JSON.stringify(this.model); }  
  
}
```

Revise app.module.ts

app.module.ts define o módulo raiz do aplicativo. Nela, você identifica os módulos externos que você usará no aplicativo e declarará os componentes que pertencem a este módulo, como o FormularioComponent.

Importe a diretiva forms

```
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import {HttpModule} from '@angular/http';
5  import {RouterModule} from '@angular/router';
6
7  import { FormsModule } from '@angular/forms';
8
9  import {ROUTES} from './app.routes';
10
11  import { AppComponent } from './app.component';
12  import { HeaderComponent } from './header/header.component';
13  import { FooterComponent } from './footer/footer.component';
14  import { HomeComponent } from './home/home.component';
15  import { PizzariaComponent } from './pizzaria/pizzaria.component';
16  import { ProdutosComponent } from './produtos/produtos.component';
17  import { SobreComponent } from './sobre/sobre.component';
18  import { ListaPizzasComponent } from './produtos/lista-pizzas/lista-pizzas.component';
19  import { FormularioComponent } from './formulario/formulario.component';
```



AngularJs - Formulário



Revise app.module.ts

Complemente a importação

Importe a diretiva forms

```
],  
imports: [  
  BrowserModule,  
  FormsModule,  
  RouterModule.forRoot(ROUTES)  
],
```

formulario.component.html

Copie o **template** de formulário que esta na rede.

```
1 <div class="container">
2   <h1>Formulário</h1>
3   <form>
4     <div class="form-group">
5       <label for="name">nome</label>
6       <input type="text" class="form-control" id="name" required>
7     </div>
8
9     <div class="form-group">
10      <label for="name">e-mail</label>
11      <input type="text" class="form-control" id="mail" required>
12    </div>
13
14    <div class="form-group">
15      <label for="alterEgo">Telefone</label>
16      <input type="text" class="form-control" id="telefone" required>
17    </div>
18  </form>
</div>
```

AngularJs - Formulário



Na página sobre.component.html, vamos inserir a nova tag

```
<p>
  Fundada pelo piracicabano João Malagueta, a Venite se estabeleceu
  assavam as pizzas. Agora, o legado continua mantendo a mesma tradi
</p>

<app-formulario></app-formulario>
```

AngularJs - Formulário



formulario.component.css

Vamos implementar o nosso css do componente

```
formulario.component.html  formulario.component.css  forms.ts
1  .ng-valid[required], .ng-valid.required {
2    border-left: 5px solid #42A948; /* green */
3  }
4
5  .ng-invalid:not(form) {
6    border-left: 5px solid #a94442; /* red */
7  }
8
```


formulario.component.html

No campo **assunto**, complemente com um campos select, buscando os dados do array “assuntos “

```
<div class="form-group">
  <label for="assunto"> Assunto</label>
  <select class="form-control" id="assunto" required>
    <option *ngFor="let sobre of assuntos" [value]="sobre">{{sobre}}</option>
  </select>
</div>
```

formulario.component.html

Precisamos de mais uma adição para exibir os dados. Declare uma variável de modelo para o formulário. Atualize a marca `<form>` com `#Form = "ngForm"` da seguinte maneira:

```
formulario.component.html x forms.ts x formulario.compone... x sobre.component.ht... x app.module.ts
1  <div class="container">
2    <h1>Formulário</h1>
3    <form (ngSubmit)="onSubmit()" #Form="ngForm">
4      <div class="form-group">
5        <label for="name">nome</label>
6        <input type="text" class="form-control" id="name" required
7          [(ngModel)]="model.name" name="name"
8          #name="ngModel" >
9      </div>
10
```

AngularJs - Formulário



formulario.component.html

Adicionando uma mensagem no campo **nome**, caso o campo esteja vazio

```
formulario.component.html  formulario.component.css  forms.ts  formulario.comp
1  <div class="container">
2    <h1>Formulário</h1>
3    <form (ngSubmit)="onSubmit()" #Forms="ngForm">
4      <div class="form-group">
5        <label for="name">nome</label>
6        <input type="text" class="form-control" id="nome"
7          required
8          [(ngModel)]="model.nome" name="nome"
9          #nome="ngModel">
10       <div [hidden]="nome.valid || nome.pristine"
11         class="alert alert-danger">
12         Esta informação é requerida!
13       </div>
14     </div>
```

FILIP

formulario.component.html

Adicionando uma mensagem no campo **email**, caso o campo esteja vazio

```
<div class="form-group">
  <label for="name">e-mail</label>
  <input type="text" class="form-control" id="email" required
    [(ngModel)]="model.email" name="email"
    #email="ngModel"
  >
  <div [hidden]="email.valid || email.pristine"
    class="alert alert-danger">
    Esta informação é requerida!
  </div>
</div>
```

formulario.component.html

Adicionando uma mensagem no campo **telefone**, caso o campo esteja vazio

```
<div class="form-group">
  <label for="alterEgo">Telefone</label>
  <input type="text" class="form-control" id="telefone"
    required
    [(ngModel)]="model.telefone" name="telefone"
    #telefone="ngModel">
  <div [hidden]="telefone.valid || telefone.pristine"
    class="alert alert-danger">
    Esta informação é requerida!
  </div>
</div>
```

formulario.component.html

Adicionando uma mensagem no campo **mensagem**, caso o campo esteja vazio

```
<div class="form-group">
  <label for="alterEgo">Mensagem</label>
  <input type="text" class="form-control" id="mensagem"
    required
    [(ngModel)]="model.mensagem" name="mensagem"
    #mensagem="ngModel">
  <div [hidden]="mensagem.valid || mensagem.pristine"
    class="alert alert-danger">
    Esta informação é requerida!
  </div>
</div>
```

formulario.component.html

Vamos implementar a ação do botão

```
<button type="submit" class="btn btn-success" [disabled]="!Forms.form.valid">Enviar</button>  
<button type="button" class="btn btn-default" (click)="novoForm(); form.reset()">Novo</button>
```


formulario.component.ts

Vamos criar uma nova classe chamada de **novoForm**

```
novoForm() {  
    this.model = new Forms(42, '', '', '', '', '');  
}  
  
}
```

FIAP



Copyright © **2017** Prof. Renato Bortolin Boschini

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).