Banking) com Angular 6 — na prática e sem complicações parte 4

Danilo Agostinho Follow
Oct 29, 2018 · 6 min read

- Salve pessoal! Chegamos à parte 4 da nossa série de posts para o **Training Center.** Caso tenha perdido a **primeira**, **segunda** ou **terceira** parte, não deixe de acompanhar clicando nos links abaixo:

Parte 1;

Parte 2;

Parte 3.

Como prometido, nesse post vamos criar o cadastro de novos usuários. Vamos também, gerar a autenticação da aplicação. Fica ligado nos recursos que vamos implementar:

Cadastro dos usuários;

Salvar dados dos usuários no LocalStorage do navegador;

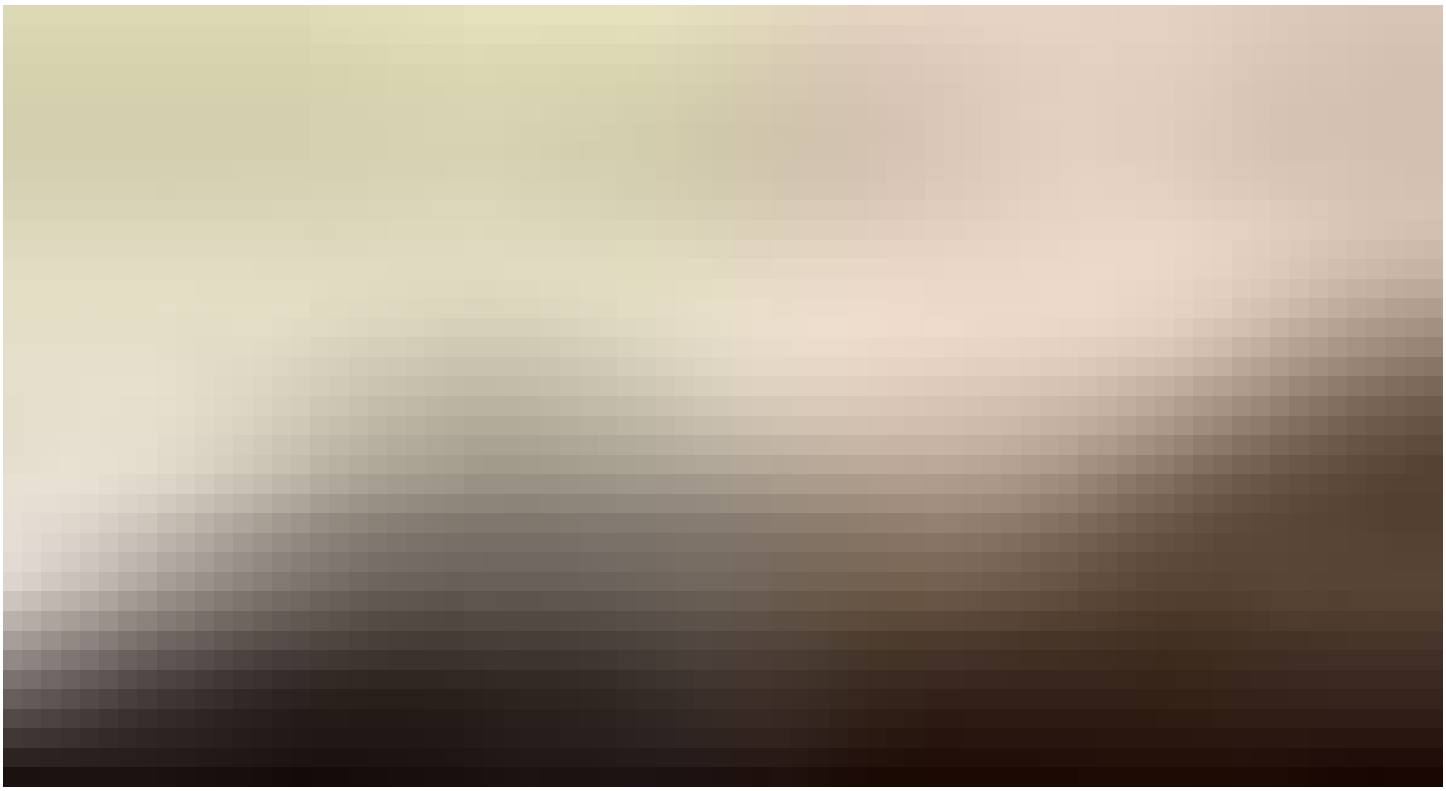
Criar autenticação dos usuários (login);

Acessar o painel do usuário.

Pré-requisitos: já ter visto as primeiras aulas publicadas. **Let's go!**

Onde paramos

Se vocês seguiram todos os passos sugeridos nos posts, sua aplicação deve estar parecida com isto:

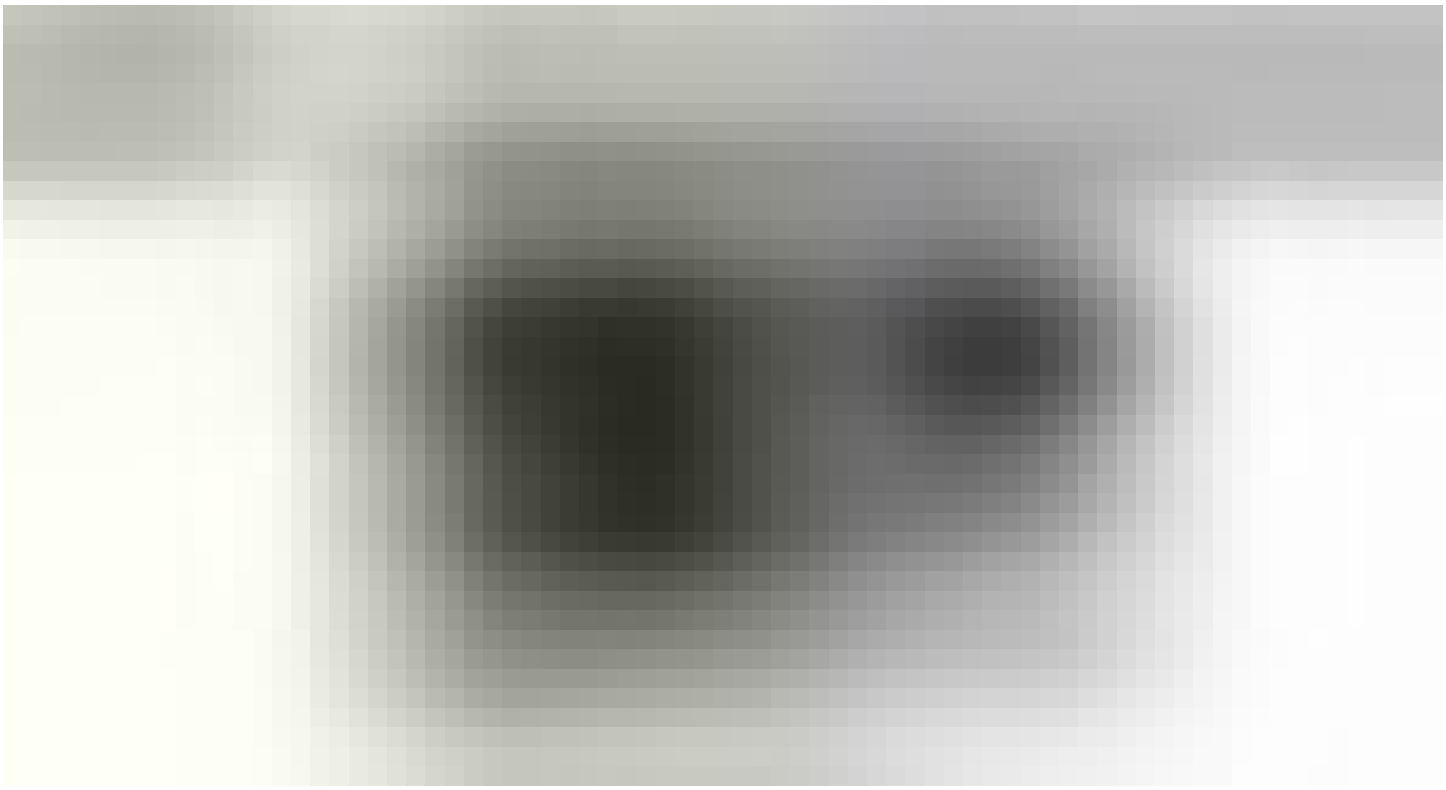


Cadastro de usuários

Essa parte é muito bacana! Vamos guardar alguns dos dados digitados no formulário no **LocalStorage**. Para quem não sabe, o **LocalStorage** é um espaço disponível nos navegadores onde podemos armazenar informações. Seu limite varia de navegador para navegador. No Chrome, por exemplo, o limite é de 5MB de dados, por origem.

Em uma aplicação real, jamais guardaríamos dados importantes no navegador. Normalmente, usamos um Banco de Dados pra isso.

Basicamente, qualquer site guarda informações no **LocalStorage.** Veja:



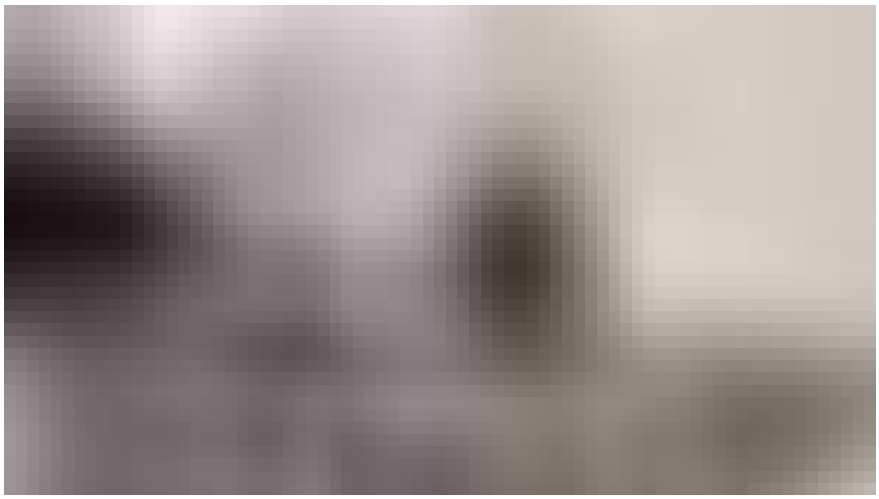
1. Obtendo os dados digitados no formulário

Vamos utilizar a abordagem **Formulários Reativos.** Antes de começar a codar é importante entender o conceito de um **Formulário Reativo** do Angular**.**

> *Formulários reativos fornecem uma abordagem orientada a modelo para manipular entradas no formulário cujo os valores mudam o tempo todo.*
>
> *—Documentação do angular.io.*

Simplificando: vamos controlar todos os dados que são digitados no formulário. Inclusive acompanharemos, em tempo real, mudanças de informações passadas pelo usuário no formulário.

A documentação explica de forma detalhada sobre os formulário reativos, mas no momento já sabemos o básico para prosseguir.



Foca no trabalho!

Nosso objetivo é a prática, então… já sabe, né? :D

Criando controles para nosso formulário

Abra o arquivo **cadastro-clientes.component.html** e siga os passos abaixo:

Adicione o método abaixo como atributo do form:

```
(ngSubmit)="cadastro()"
```

2. Adicione os **formControlNames** aos inputs do formulário:

```
formControlName="nome"
formControlName="cpf"
formControlName="email"
formControlName="telefone"
formControlName="endereco"
```

3. Adicione agora um **formGroup** chamado **formCadastro** como atributo do form:

```
[formGroup]="formCadastro"
```

Nosso form atualizado ficou assim:

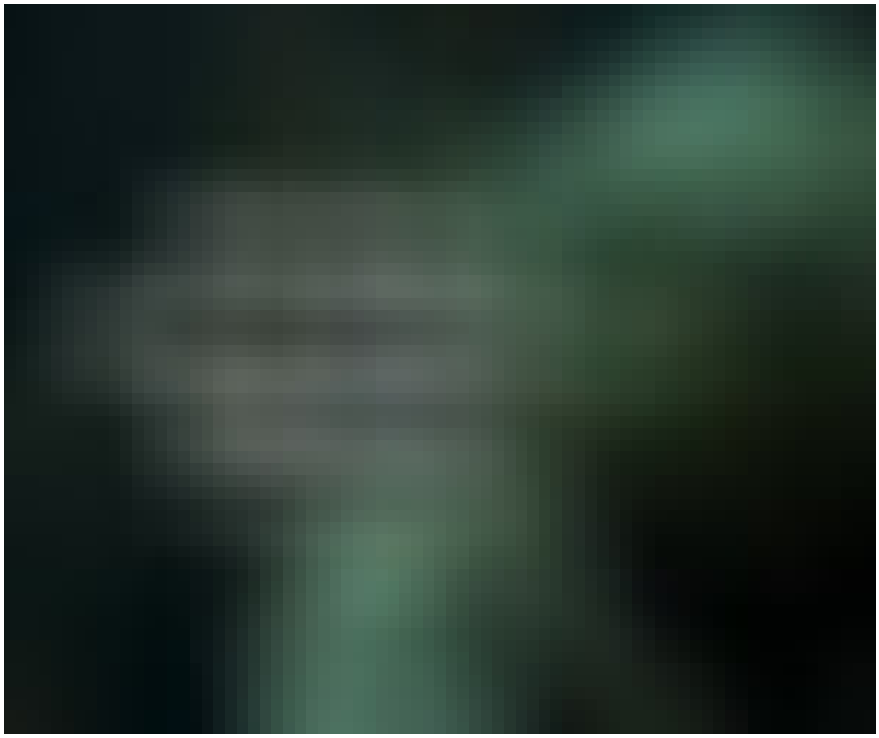
```
<div class="container">

<div class="row box-cadastro">

<div class="col-12">

<h3 class="text-center">Você está a um passo de criar sua
conta Training Banking</h3>

<form class="mt-5" (ngSubmit)='cadastro()'
[formGroup]='formCadastro'>

<div class="form-group">

<input type="text" class="form-control cadastro"
placeholder="Nome" formControlName='nome'>

</div>
```

```
<div class="form-group">

<input type="number" class="form-control cadastro"
placeholder="CPF" formControlName='cpf'>

</div>

<div class="form-group">

<input type="email" class="form-control cadastro"
placeholder="E-mail" formControlName='email'>

</div>

<div class="form-group">

<input type="tel" class="form-control cadastro"
placeholder="Telefone" formControlName='telefone'>

</div>

<div class="form-group">

<input type="text" class="form-control cadastro"
placeholder="Endereço" formControlName='endereco'>

</div>

<button type="submit" class="btn btn-cadastro btn-
lg">Cadastrar</button>

</form>

</div>

</div>

</div>
```

Repare que o Angular dispara um erro:

1. Calma! Não se desespere com um simples erro. Confesso pra vocês. Eu me sentia muito frustado quando me deparava com erros, até pensei em desistir de tudo, pensava isso não era pra mim.

   Mas preste a atenção às sábias palavras:



Atente-se ao log de erro, pense para resolver o problema. Abaixo, vamos resolver esse problema.

Pegando os dados digitados no formulário

Para usar formulários reativos no Angular é preciso importar o pacote ReactiveFormsModule em seu arquivo **app.module.ts.** Siga os passos.

> No app.module.ts importe o ReactiveFormsModule do pacote ‘@angular/forms’

```
import { ReactiveFormsModule } from ‘@angular/forms’;
```

2. Adicione o ReactiveFormsModule em seu array de importações “@NgModule”

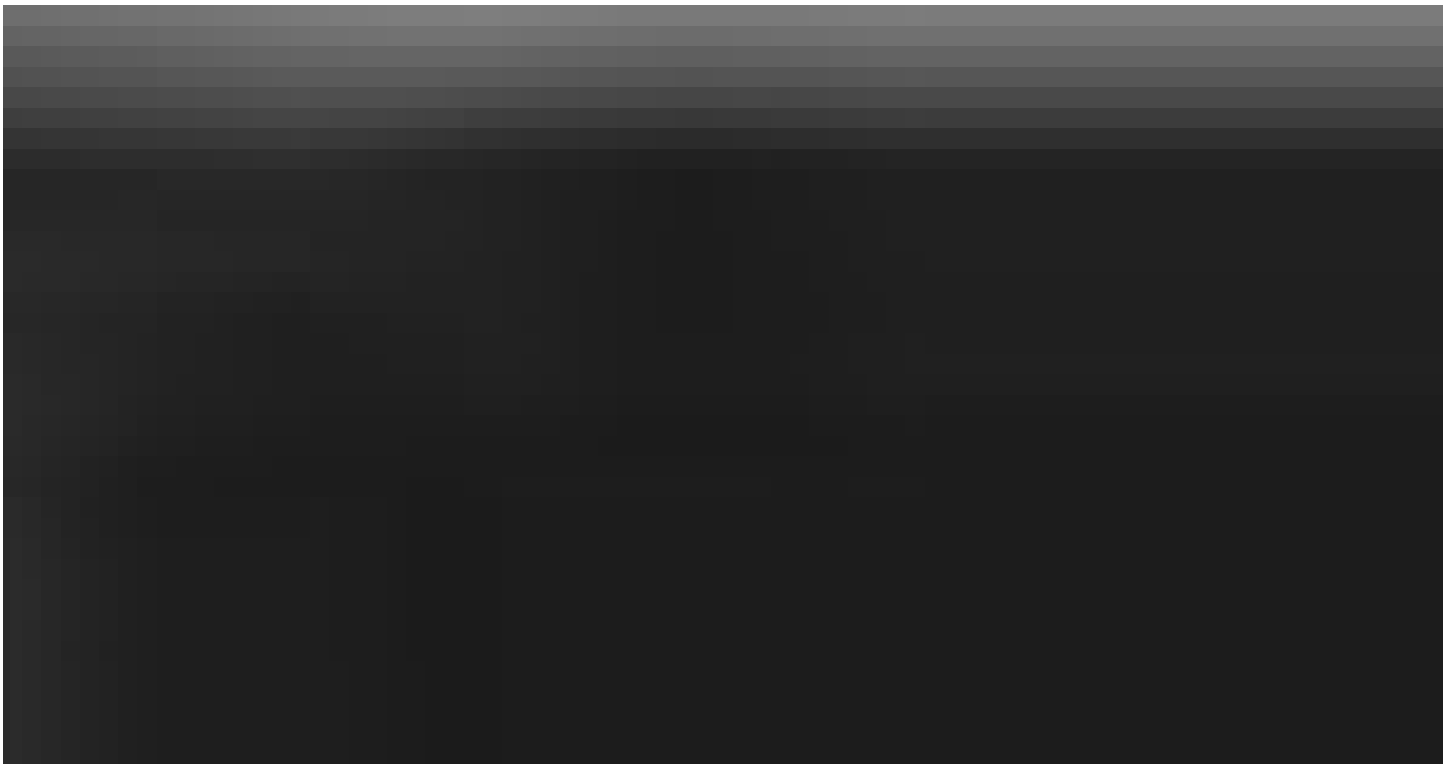
```
@NgModule({
imports: [
// other imports ...
  ReactiveFormsModule
], })
```

3. Agora abra o arquivo **cadastro-clientes.component.ts**. Nele, vamos importar os seguintes itens:

```
import { FormBuilder } from '@angular/forms';
```

Injete o serviço formBuilder no construtor do component:

```
constructor(private fb: FormBuilder) { }
```



Veja como está nossa aplicação até o momento:

Repare que ainda estamos com erro. Já vamos resolvê-lo.

Gerando controles do formulário

Agora que já adicionamos os módulos necessários para trabalhar com formulários reativos, importamos as classes e injetamos os serviços necessários, é hora de criarmos os controles do formulário.

Para refrescar nossa memória, temos os seguintes campos em nosso formulário:

Nome;

CPF;

Email;

Telefone;

Endereço.

Precisamos criar um controle que englobe todos esses campos do nosso formulário. Para isso, copie e cole em seu editor o trecho de código abaixo.

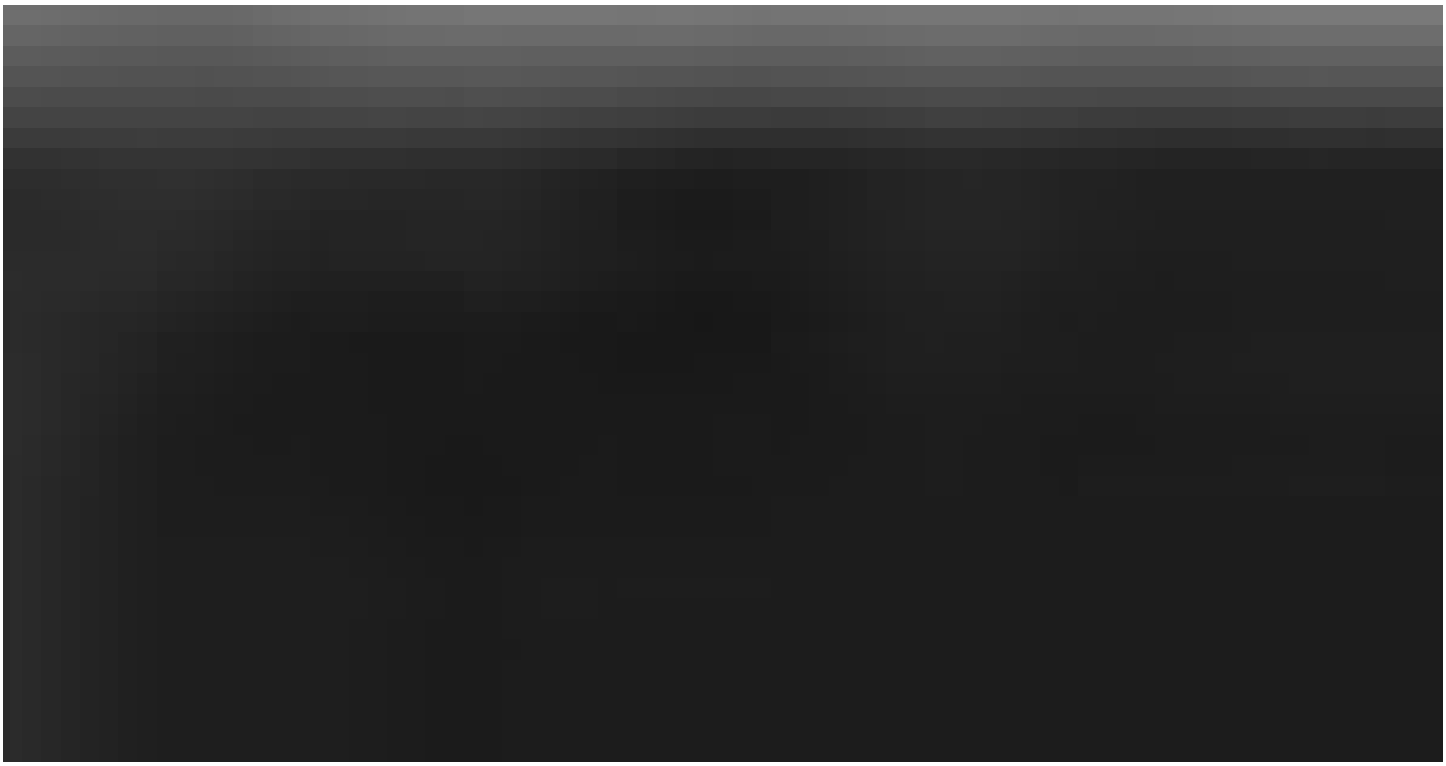
Declare uma variável chamada **formCadastro,** que é o mesmo nome passado no **[formGroup]** em nosso HTML. Lembra**?**

```
formCadastro;
```

Dentro do **ngOnInit,** crie o controle de formulário:

```
formCadastro = this.fb.group({
  nome: [''],
  cpf: [],
  email: [],
  telefone: [],
```

```
    endereco: []
});
```



É recomendável que você pare o servidor e suba novamente. Vamos ver como nossa aplicação está neste exato momento?

Ainda estamos com erro, mas dessa vez, eu grifei o erro. Ele encontra-se no arquivo **HeaderComponent.html.** Precisamos criar um controle de formulário para ele também. Vai ser bem simples, pois esse **component** possui apenas o campo de CPF.

No arquivo **headerComponent.ts,** atualize-o da seguinte forma:

1. Declare o atributo o formGroup no formulário html;

```
[formGroup]="form"
```

2. Importe o FormBuilder do pacote '@angular/forms';

```
import { FormBuilder } from '@angular/forms';
```

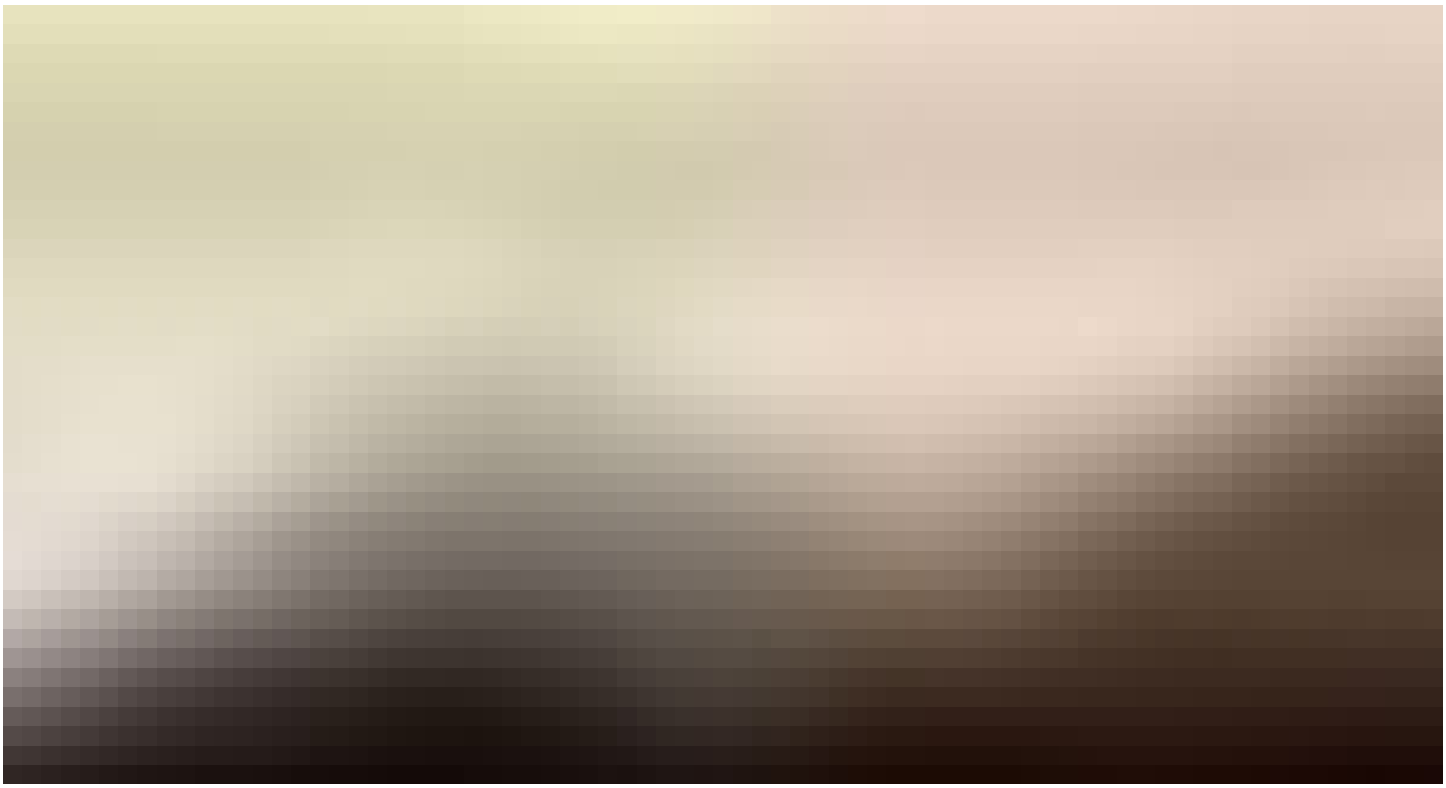
3. Injete no constructor;

```
constructor(private fb: FormBuilder) { }
```

4. Crie o controle do formulário.

```
this.formLogin = this.fb.group({
  cpf: ['']
});
```

Agora nossa aplicação voltou a funcionar:



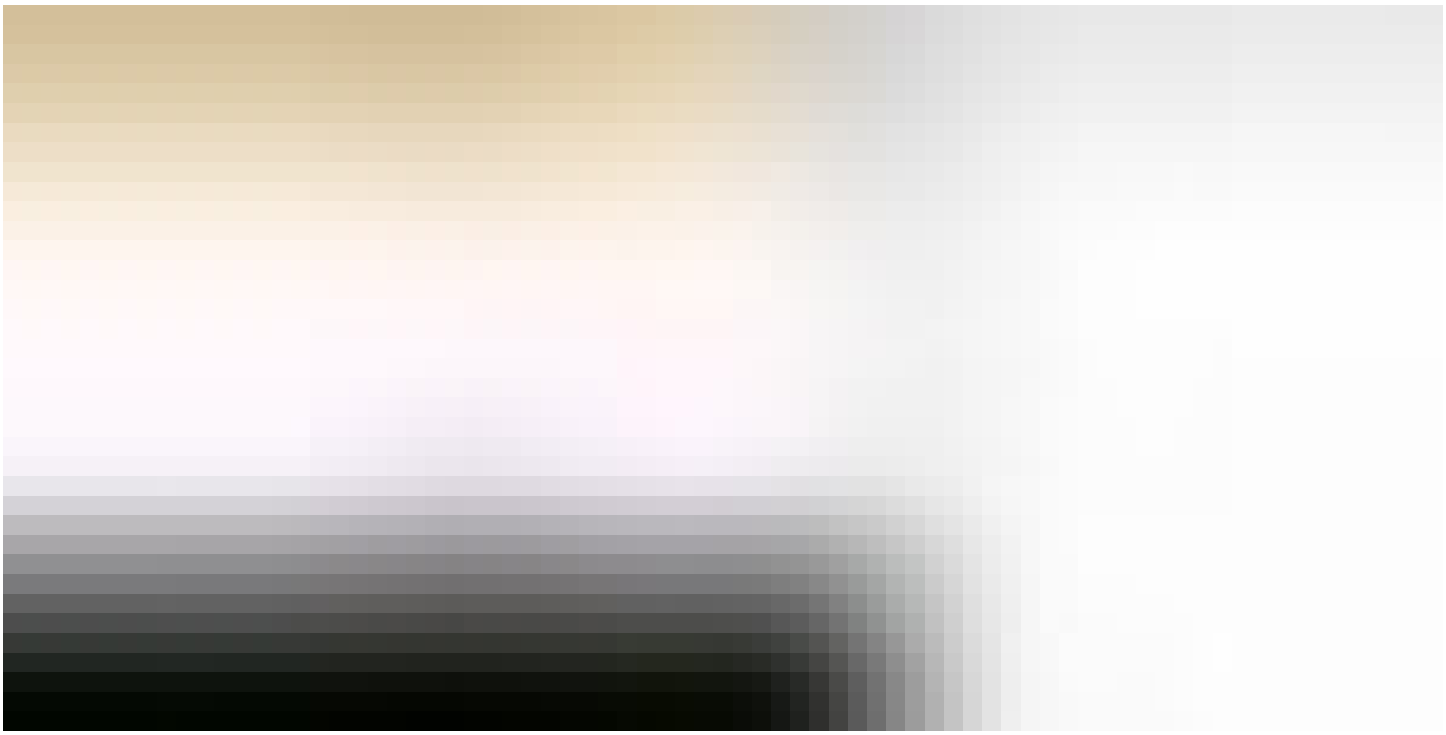
Visualizando os dados digitados no formulário

Agora que já configuramos e criamos um controle para os formulários da nossa aplicação, é muito fácil visualizar os dados digitados nos campos. Siga os passos abaixo:

1. No arquivo **cadastro-clientes.components.ts,** crie o método abaixo:

```
cadastro(){
 console.log(this.formCadastro.controls);
}
```

Testando o cadastro:



Repare que agora temos os dados do usuário salvos em nosso objeto **formCadastro.**

Salvando login no LocalStorage

Agora que temos os dados salvos em nosso objeto, vamos convertê-lo e salvar no LocalStorage.

Siga os passos:

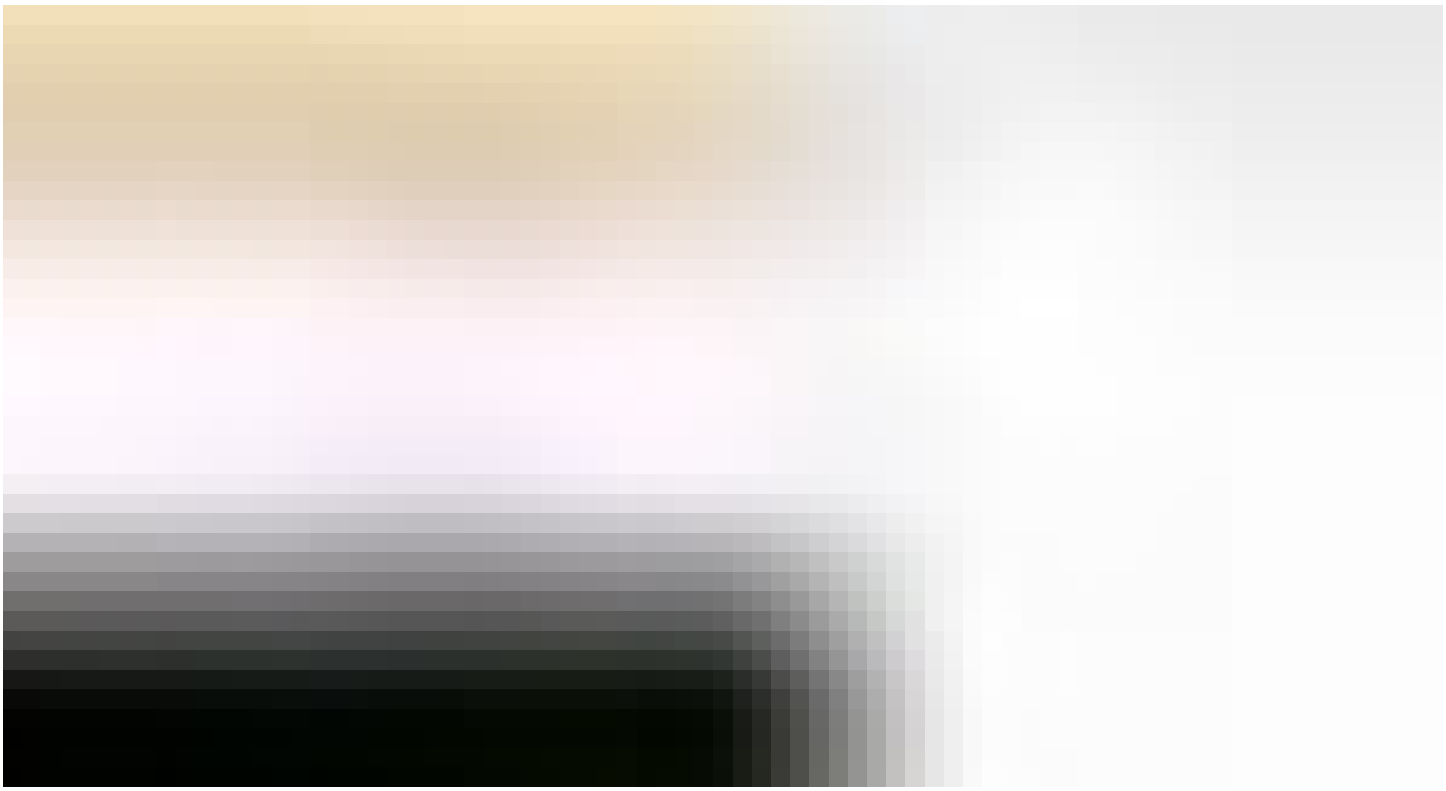
1. Em nosso método **cadastro**, **adicione as seguintes instruções:**

```
this.conversao = JSON.stringify(this.valoresForm);
localStorage.setItem('cadastro', this.conversao);
```

Assim que clicarmos no botão “cadastro” o código acima vai converter os dados recebidos em String e logo abaixo estamos salvando essa

conversão no LocalStorage.

Vamos testar a aplicação:



Criando a área logada da aplicação

Vamos precisar de um component chamado home-logada. Esse component será restrito e só poderá ser acessado assim que o usuário efetuar o login no sistema.

Para isso, vamos gerar o component como o comando:

```
ng g c home-logada
```

Criando autenticação

Essa parte é bacana, mas para esse post não ficar muito grande e contornarmos melhor os possíveis problemas, resolvi deixar para o próximo post.

Bem, é isso. Espero vocês na parte 5.

Quero também agradecer o feedback de vocês aqui nos comentários. Não esqueçam de compartilhar o post para ajudar quem precisa e qualquer dúvida pinga nos comentários ou me perguntem via Twitter segue. Meu user @danilodev_silva

Grande abraço!