

# Manipulação de eventos de formulários

# Introdução

- Os formulários são uma parte fundamental da interação com o usuário;
- Eles permitem que os usuários insiram dados e os enviem para processamento;
- Dois tipos de formulários são disponibilizados no Angular: formulários reativos e formulários template-driven;
- Ambos os tipos de formulários usam eventos para lidar com a interação do usuário e fornecer um feedback;

# Submit

- Evento é disparado quando o formulário é enviado;
- lida com envios de formulários;

```
<form [formGroup]="ordemDeServicoForm" (ngSubmit)="onSubmit()" class="ordemServico">  
  <input type="text" formControlName="nomeCliente" placeholder="Nome do Cliente">  
  <input type="text" formControlName="tipoServico" placeholder="Tipo de Serviço">  
  <input type="date" formControlName="Data da Ordem">  
  <textarea formControlName="detalhes" placeholder="Detalhes"></textarea>  
  <button type="submit">Enviar</button>  
</form>
```

# valueChanges

---

- Este é um Observable que emite um evento toda vez que o valor do FormControl muda;
- É possível reagir a alterações no valor do controle de formulário;

```
inscricaoValueChanges(){  
  console.log('Inscrição valueChanges');  
  this.ordemDeServicoForm.get('nomeCliente')?.valueChanges.subscribe(  
    valor => console.log('Nome do cliente: ' + valor)  
  );  
  this.ordemDeServicoForm.get('tipoServico')?.valueChanges.subscribe(  
    valor => console.log('Tipo de serviço: ' + valor)  
  );  
  this.ordemDeServicoForm.get('dataOrdem')?.valueChanges.subscribe(  
    valor => console.log('Data da ordem: ' + valor)  
  );  
  this.ordemDeServicoForm.get('detalhes')?.valueChanges.subscribe(  
    valor => console.log('Detalhes: ' + valor)  
  );  
}
```

# statusChanges

- Este é um Observável que emite um evento sempre que o status de validação do controle do formulário muda;
- É possível reagir a alterações no status de validação do controle de formulário;

```
inscricaoStatusChanges(){  
  console.log('Inscrição statusChanges');  
  this.ordemDeServicoForm.get('nomeCliente')?.statusChanges.subscribe(  
    status => console.log('Status do nome do cliente: ' + status)  
  );  
  this.ordemDeServicoForm.get('tipoServico')?.statusChanges.subscribe(  
    status => console.log('Status do tipo de serviço: ' + status)  
  );  
}
```

Inscrição statusChanges

Status do nome do cliente: VALID

Status do tipo de serviço: VALID

Status do tipo de serviço: INVALID

# ngModelChange

- Este evento é disparado quando o valor do FormControl vinculado ao ngModel é alterado;
- É possível utilizar para lidar com alterações no valor do FormControl;
- Em formulários reativos, é mais comum usar o Observable valueChanges para controles de formulário, que fornece uma funcionalidade semelhante.

```
<input [(ngModel)]="nomeDoConsumidor" (ngModelChange)="onNomeDoConsumidorChange()"
placeholder="Nome do Consumidor">
```

# Outros eventos de form

- Os principais eventos de formulários no Angular são: submit, valueChanges e statusChanges;
- Porém os formulários são criados com elementos de formulário HTML padrão;
- É possível utilizar qualquer eventos DOM padrão disponíveis nesses elementos:
  - Focus
  - Blur
  - Change
  - input

```
<input type="text" formControlName="NomePassageiro" placeholder="Nome do Passageiro"  
(focus)="onFocus()" (blur)="onBlur()">
```

- Esses eventos são acionados quando um controle de formulário ganha ou perde o foco;
- É um evento do DOM (Document Object Model);

## focus / blur

```
onFocus(){  
  this.fOnFocus = true;  
}  
  
onBlur(){  
  this.fOnFocus = false;  
}
```



# change

- Este evento é disparado quando o valor de um elemento de entrada é confirmado pelo usuário;
- É um evento do DOM (Document Object Model);
- Para forms Angular, é preferível utilizar Observable **valueChanges** para observar mudanças nos valores de controle de formulário.

```
<input type="text" formControlName="NomePassageiro" placeholder="Nome do Passageiro"  
(focus)="onFocus()" (blur)="onBlur()" (change)="onChange($event)">
```

```
onChange(event: any){  
  this.nomeDoPassageiro = event.target ? event.target.value : null;  
  this.fonChange= true;  
  console.log(event.target);  
}
```

# input

- Este evento é disparado sempre que o valor do elemento de entrada é alterado;
- É um evento do DOM (Document Object Model);

```
<input type="text" formControlName="NomePassageiro" placeholder="Nome do Passageiro"  
(focus)="onFocus()" (blur)="onBlur()" (change)="onChange($event)" (input)="onInput($event)">
```

```
onInput(event:any){  
  this.tempNomePassageiro = event.target ? event.target.value : null;  
}
```

Zupeto Diamante
Número do Voo
dd / mm / aaaa
dd / mm / aaaa
Aeroporto de Partida
Aeroporto de Chegada

Enviar

Quadro de Eventos

Nome do passageiro

focus/blur: false

change: true

nomeDoPassageiro: Zupeto Diamante

temp Nome do Passageiro: Zupeto  
Diamante

# Disposição dos componentes na aplicação

- Onde aplicar o estilo css na aplicação Angular de forma a organizar o layout da aplicação?



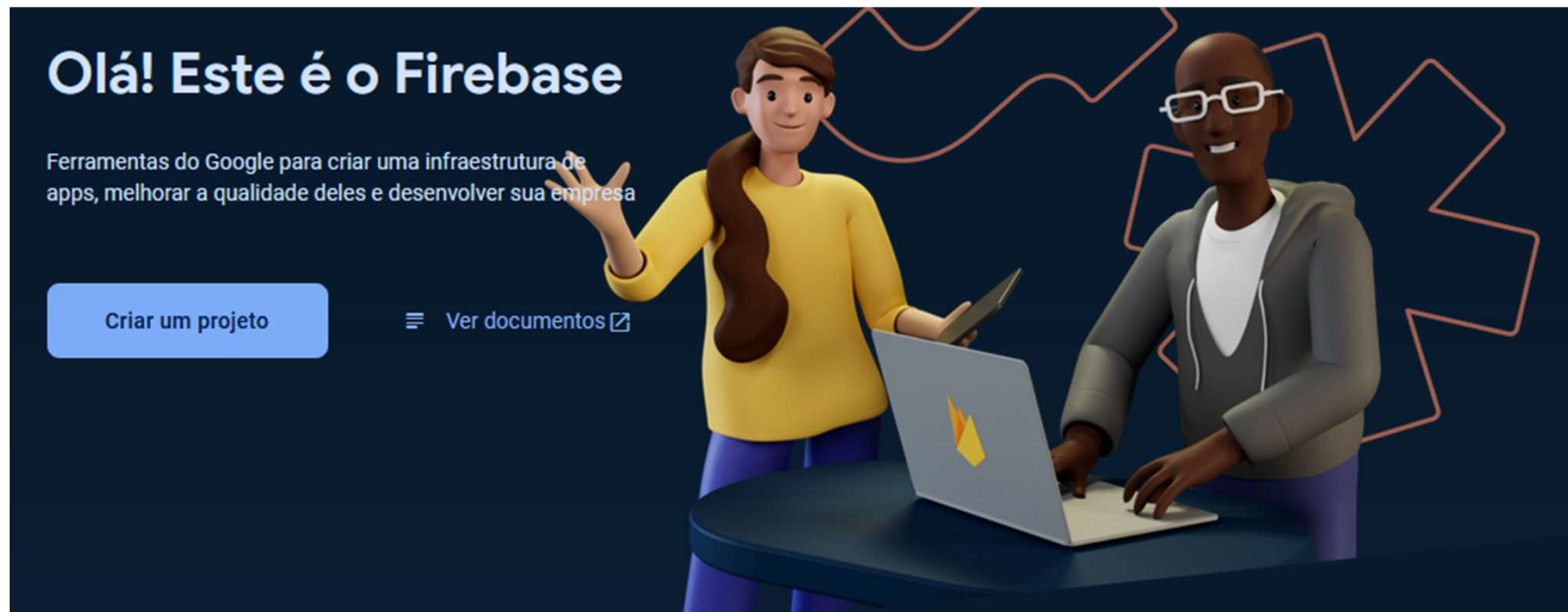
# Como fazer o deploy de uma aplicação no Firebase

- Usar e checar variáveis de ambiente;
- Refinar e testar o código;
- **ng build**
  - Criação da pasta dist/<nomedesuaaplicacao>
- Alterar o arquivo **angular.json** caso a aplicação ultrapasse 1mb

```
"budgets": [  
  {  
    "type": "initial",  
    "maximumWarning": "2mb",  
    "maximumError": "5mb"  
  },  
]
```

# Configurando o firebase

- Utilize uma conta do google;
- Ir no endereço: <https://console.firebase.google.com/u/0/>




# Configurando o firebase

× Criar um projeto(Passo 1 de 3)

Vamos começar nomeando  
o projeto<sup>?</sup>

Nome do projeto

 residencia-ccdd7

Continuar

# Configurando o firebase

## Google Analytics para seu projeto do Firebase

O Google Analytics é uma solução de análise ilimitada e sem custos financeiros. Com ele, é possível segmentar, gerar relatórios e muito mais nos seguintes produtos: Firebase Crashlytics, Cloud Messaging, Mensagens no app, Configuração remota, Teste A/B e Cloud Functions.

O Google Analytics ativa:



Teste A/B ?



Segmentação de usuários em produtos do Firebase ?



Usuários sem falhas ?



Gatilhos do Cloud Functions com base em eventos ?



Geração de relatórios ilimitada gratuita ?



Ativar o Google Analytics neste projeto  
Recomendado

Anterior

Continuar



# Configurando o firebase

✕ Criar um projeto(Passo 3 de 3)

Configurar o Google Analytics

Localização do Analytics ⓘ

Brasil

O Google Analytics é uma ferramenta de negócios. Use-a exclusivamente para fins relacionados ao seu comércio, negócio, ofício ou profissão.

Configurações de compartilhamento de dados e termos do Google Analytics

☒ Usar as configurações padrão para o compartilhamento de dados do Google Analytics. [Saiba mais](#)

✕

Compartilhe seus dados do Analytics com o Google para melhorar os produtos e serviços da empresa

✓

Compartilhe seus dados do Analytics com o Google para ativar o Comparativo de mercado

✓

Compartilhe seus dados do Analytics com o Google para ativar o suporte técnico

✓

Compartilhe seus dados do Analytics com os especialistas em contas do Google

☒ Eu aceito os [Termos do Google Analytics](#)

Após a criação do projeto, uma nova propriedade do Google Analytics será criada e vinculada ao seu projeto do Firebase. Esse processo permitirá o fluxo de dados entre os produtos. Os dados da propriedade do Google Analytics exportados para o Firebase ficam sujeitos aos Termos de Serviço do Firebase, e os dados do Firebase importados para o Google Analytics ficam sujeitos aos Termos de Serviço do Google Analytics. [Saiba mais](#)

Anterior

Criar projeto

Configurando  
o firebase



residencia

✓ Your Firebase project is ready

Continuar

# Baixar e instalar o firebase-tools


- Linha de comando:
  - npm install -g firebase-tools
  - firebase login
- Entre no seu diretório de projeto angular e inicialize o projeto:
  - firebase init

```
#####  #### #####  #####  #####  ###  #####  #####
##      ##  ##      ##  ##      ##      ##  ##  ##      ##
#####  ##  #####  #####  #####  #####  #####  #####
##      ##  ##      ##  ##      ##      ##  ##      ##  ##
##      #### ##      ##  #####  #####  ##      ##  #####

You're about to initialize a Firebase project in this directory:
Aula12\formEvents

? Are you ready to proceed? (Y/n)
```

# Inicializar o projeto (continuação ...)



```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then
  choices. (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
  ( ) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision
  ( ) Firestore: Configure security rules and indexes files for Firestore
  ( ) Functions: Configure a Cloud Functions directory and its files
>(*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
  ( ) Hosting: Set up GitHub Action deploys
  ( ) Storage: Configure a security rules file for Cloud Storage
  ( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)
```

# Inicializar o projeto (continuação ...)

## === Project Setup

First, let's associate this project directory with a Firebase project. You can create multiple project aliases by running `firebase use --add`, but for now we'll just set up a default project.

? Please select an option: (Use arrow keys)

> Use an existing project

Create a new project

Add Firebase to an existing Google Cloud Platform project

Don't set up a default project

# Inicializar o projeto (continuação ...)

```
? Please select an option: Use an existing project  
? Select a default Firebase project for this directory:
```

Selecione o nome do projeto (neste exemplo foi residência)

Inicializar o  
projeto  
(continuação ...)

### === Hosting Setup


Your **public** directory is the folder (relative to your project directory) that will contain Hosting assets to be uploaded with **firebase deploy**. If you have a build process for your assets, use your build's output directory.

```
? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
+ Wrote public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
```

# Checar o arquivo Angular.json

```
"architect": {  
  "build": {  
    "builder": "@angular-devkit/build-angular:browser",  
    "options": {  
      "outputPath": "dist/- nomeDoSeuProjeto",  
      "index": "src/index.html",  
      "main": "src/main.ts",  
      "polyfills": [  
        "zone.js"  
      ],  
    },  
  },  
}
```





# Faça upload do projeto

- Linha de comando:
  - firebase deploy

# Referências

- <https://angular.io/api/forms/NgForm>
- [https://developer.mozilla.org/en-US/docs/Web/API/Element/input\\_event](https://developer.mozilla.org/en-US/docs/Web/API/Element/input_event)
- [https://developer.mozilla.org/en-US/docs/Web/API/Element/focus\\_event](https://developer.mozilla.org/en-US/docs/Web/API/Element/focus_event)
- [https://developer.mozilla.org/en-US/docs/Web/API/Element/blur\\_event](https://developer.mozilla.org/en-US/docs/Web/API/Element/blur_event)
- [https://developer.mozilla.org/enUS/docs/Web/API/HTMLElement/change\\_event](https://developer.mozilla.org/enUS/docs/Web/API/HTMLElement/change_event)