



14/8/2019

Treinamento Angular 8

COTI INFORMATICA



WWW.COTIINFORMATICA.COM.BR

AV RIO BRANCO, 185 SALA 307 – CENTRO – RIO DE JANEIRO - RJ

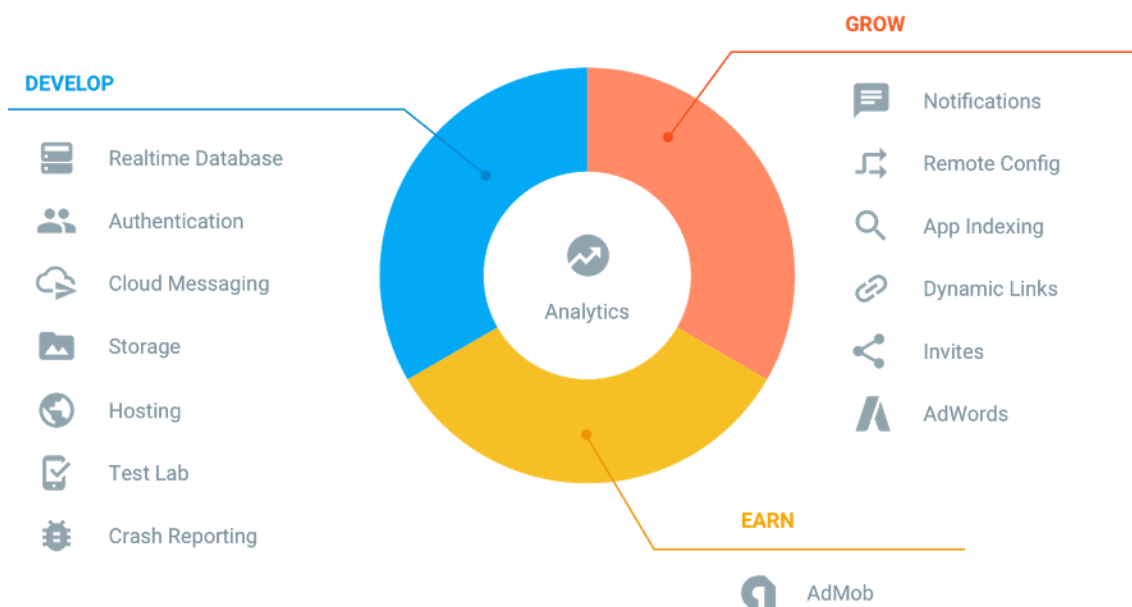
Sumário

O que é Firebase?	2
Para saber mais	4
O que é jarallax?	5
NOVO PROJETO RELACIONAMENTO MUITOS PARA MUITOS.....	6
NOVO PROJETO COM VIDEO.....	19
Fontes Online do Google.....	21
NOVO PROJETO LOGIN FIREBASE.....	30
Firebase Console	31

O que é Firebase?

O Firebase é uma plataforma do Google que contém várias ferramentas e uma excelente infraestrutura para ajudar desenvolvedores web e mobile a criar aplicações de alta qualidade e performance.

Esta plataforma contém quatro seguimentos de serviços: Analytics, Develop, Grow e Earn. Veja um pouco sobre cada um deles.



O **Firebase Analytics** é uma solução gratuita e ilimitada, considerado o *coração do Firebase*, é utilizado para gerar as métricas da aplicação, mensurar o comportamento do usuário. Atualmente é disponibilizado para aplicações iOS e Android.



Com a missão de poupar tempo e fornecer um aplicativo de alta qualidade, esta plataforma contém vários recursos para os desenvolvedores, são eles:

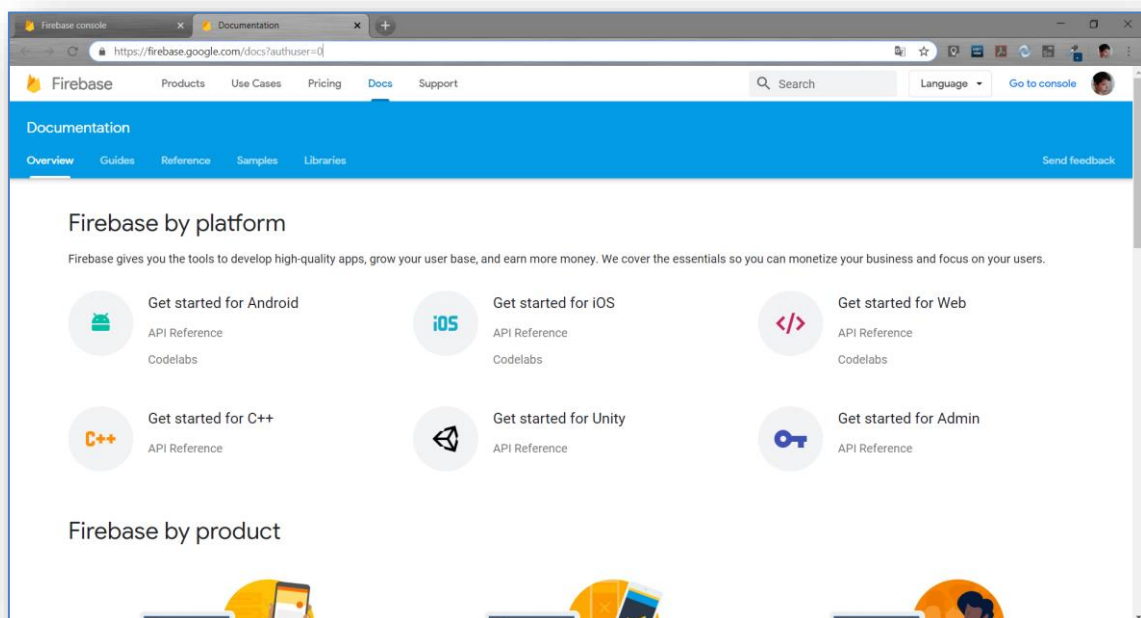
- Cloud Messaging: O Firebase Cloud Messaging – FCM permite a entrega/recebimento de mensagens e notificações entre as plataformas iOS, Android e Web.
- Authentication: Este recurso de autenticação é fundamental para as aplicações onde é necessário saber a identidade do usuário e manter o controle do acesso ao app. Também, com o Firebase Authentication, é possível utilizar provedores de identidades federais para autenticação como as contas do Google, Facebook, Twitter e GitHub.
- Realtime Database: O Firebase também disponibiliza um banco de dados NoSQL (Firebase Realtime Database) hospedado em nuvem, onde os dados são armazenados como JSON e sincronizados em tempo real com todos os clientes conectados.
- Storage: Útil para armazenar arquivos como imagens, vídeos e áudio, além de outros conteúdos gerados por usuários.
- Hosting: Este serviço é oferecido para hospedar HTML, CSS e JavaScript para seu site, além de outros ativos fornecidos pelo desenvolvedor, como gráficos, fontes e ícones. O Firebase

Hosting possui certificado SSL fornecido automaticamente, é ideal para aplicativos web e web para dispositivos móveis.

- Remote Config: Armazena pares de chave-valor especificados pelo desenvolvedor que permite alterar o comportamento e a aparência do aplicativo sem exigir que os usuários baixem uma atualização do app.
- Test Lab: O Firebase Test Lab fornece toda infraestrutura em nuvem que você precisa para testar seus aplicativos Android e mesmo que você não tenha escrito o código de teste para o app, o Test Lab pode operar no seu aplicativo automaticamente em busca de falhas. Todos os resultados do teste são disponibilizados no Firebase console.
- Crash Reporting: Este recurso é capaz de criar relatórios de erro detalhados para os aplicativos Android e iOS. Os erros são agrupados em conjuntos e são organizados de acordo com a gravidade do impacto para os usuários. Até o momento está na versão beta e já demonstra ser um excelente aliado para os desenvolvedores.

Para saber mais

<https://firebase.google.com/docs?authuser=0>



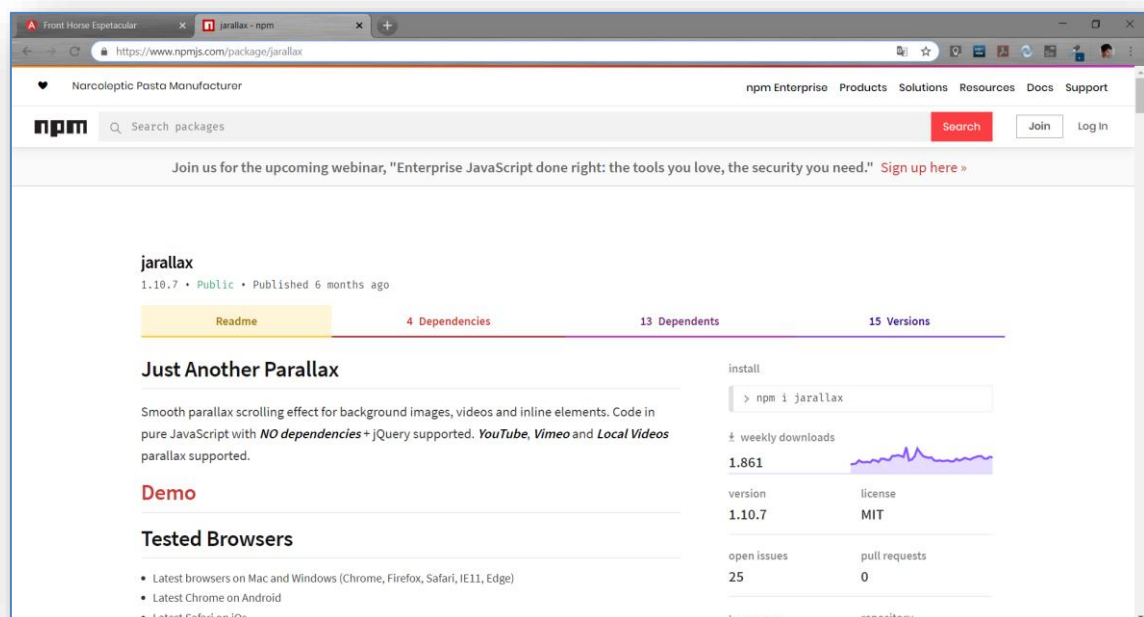
O que é jarallax?

Jarallax é uma biblioteca javascript de código aberto que facilita o ajuste do css baseado na interação. Com Jarallax é fácil criar um site de rolagem paralaxe.

<http://www.jarallax.com/>



<https://www.npmjs.com/package/jarallax>



CRIANDO NOVO PROJETO:

Digitar o comando para criação do projeto:

"ng new nome_do_projeto"

Digitar "y" para a criação das rotas.

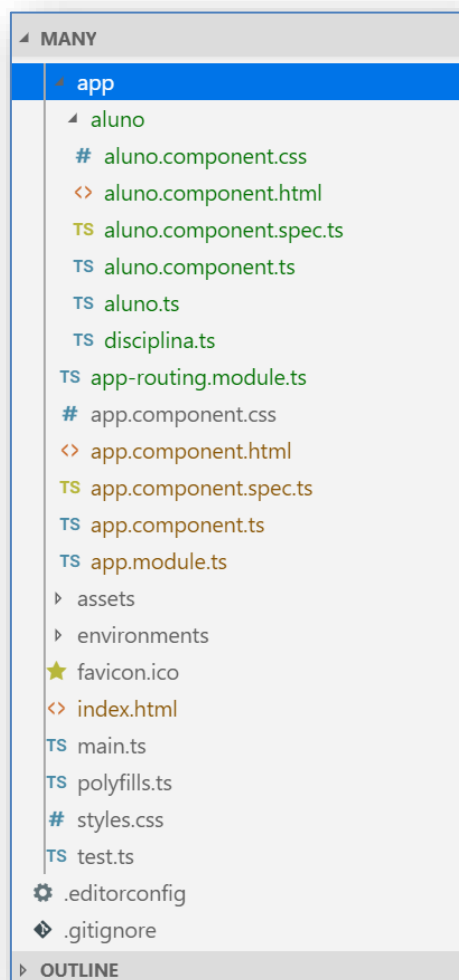
Selecionar "CSS" para o estilo.

Aguardar criar o projeto..

Projeto criado...

NOVO PROJETO RELACIONAMENTO MUITOS PARA MUITOS

Estrutura do projeto depois de finalizado:



aluno.ts

```
import { Disciplina } from './disciplina';

export class Aluno {

    id: number;
    nome: string;
    email: string;

    //RELACIONAMENTO MANY TO MANY
    disciplinas: Disciplina[] = []

    constructor(id?: number, nome?: string, email?: string) {
        this.id = id;
        this.nome = nome;
        this.email = email;
    }
}
```

disciplina.ts

```
import { Aluno } from './aluno';

export class Disciplina {

    idDisciplina: number;
    nomeDisciplina: string;

    //RELACIONAMENTO
    aluno: Aluno[] = [];

}
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { AlunoComponent } from './aluno/aluno.component';
import { FormsModule } from '@angular/forms';
import { MDBBootstrapModule } from 'angular-bootstrap-md';
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations';

@NgModule({
  declarations: [
    AppComponent,
    AlunoComponent
  ],
  imports: [
    BrowserModule,
    FormsModule, // HABILITA O USO DO FORMULARIO
    AppRoutingModule,
    MDBBootstrapModule.forRoot(), //MATERIAL DESIGN
    BrowserAnimationsModule //HABILITA AS ANIMAÇÕES
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html

```
<app-aluno></app-aluno>
```

aluno.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Aluno } from './aluno';
import { Disciplina } from './disciplina';

@Component({
  selector: 'app-aluno',
  templateUrl: './aluno.component.html',
  styleUrls: ['./aluno.component.css']
})
export class AlunoComponent implements OnInit {

  //LISTA DE ALUNOS
  listaAluno = [{
    "id": 10,
    "nome": "edson",
    "email": "edson@gmail.com"
  },
  {
    "id": 11,
    "nome": "lu",
    "email": "lu@gmail.com"
  }
  ] as Aluno[];

  //LISTA DE DISCIPLINAS
  listaTurma = [{
    "idDisciplina": 100,
    "nomeDisciplina": "java"
  },
  {
    "idDisciplina": 101,
    "nomeDisciplina": "angular"
  }, {
    "idDisciplina": 102,
    "nomeDisciplina": "ionic"
  }
  ] as Disciplina[];

  //VARIÁVEL DO ALUNO
  aluno: Aluno;
```

```

//VARIÁVEL DA LISTA DE ALUNOS
alunos: Aluno[] = [];
//MAPA PARA RELACIONAR
mapa: Map<Aluno, Disciplina[]>;
//VARIÁVEL DA LISTA DE DISCIPLINAS
disciplinas: Disciplina[] = [];

//CONSTRUTOR COM ESPAÇO DE MEMÓRIA PARA ALUNO, ALUNO
RELACIONADO COM DISCIPLINA, MAPA E LISTA DE ALUNOS
constructor() {
    this.aluno = new Aluno();
    this.aluno.disciplinas = [];
    this.mapa = new Map<Aluno, Disciplina[]>();
    this.alunos = [];
}

ngOnInit() {
}

//MÉTODO PARA RELACIONAR ALUNO COM DISCIPLINA
public anexar(aluno: Aluno) {
    this.mapa.set(aluno, aluno.disciplinas);
}

//MÉTODO QUE SELECIONA O ALUNO
public definirAluno(aluno: Aluno) {
    this.aluno = aluno;
    this.aluno.disciplinas = [];
}

//ADICIONAR DISCIPLINA AO VETOR
public adicionarDisciplinas(disciplina: Disciplina) {
    this.aluno.disciplinas.push(disciplina);
}

//ADICIONA ALUNO
public adicionarAluno(aluno: Aluno) {
    this.alunos.push(aluno); //ADICIONA AO VETOR
    this.anexar(aluno);
    for (let [key, value] of this.mapa) {
        console.log('Chave', key);
        console.log('valor', value);
    }
}

```

```

    }
    this.aluno = new Aluno();
  }
}

```

aluno.component.html

```

<h2 class="bg-secondary p-3 mb-3 text-center">Projeto Muitos
para Muitos</h2>

<!-- DEFININDO O CONTAINER FLUID COM MARGEM -->
<div class="container-fluid">
  <!-- DEFININDO UMA LINHA PARA OS 3 COMPONENTES FICAREM
ALINHADOS NA TELA, COM MARGEM INTERNA DE 30PX -->
  <div class="row p-3">

    <!-- DEFININDO A PRIMEIRA COLUNA (TAMANHO 4 COLUNAS) -->
    <div class="col-md-4 ">
      <!-- BOTAO AZUL COM ANIMAÇÃO DE ONDA AO CLICAR,
TOGGLE QUE MOSTRA OU ESCONDE O CONTEUDO -->
      <button mdbBtn color="primary" class="waves-light"
type="button" (click)="test.toggle()" mdbWavesEffect>
        Lista de Alunos
      </button>

      <!-- CONTAINER DO CONTEUDO CHAMADO PELO #TEST -->
      <div mdbCollapse #test="bs-collapse">
        <!-- NGFOR PARA TRAZER DOS OS ALUNO DA LISTA -->
        <div *ngFor="let item of listaAluno">
          <div class="mb-3 mt-2 comp p-3">
            <!-- AGRUPANDO OS DADOS COM MARGEM INFERIOR 10PX -->
            <div class="input-group mb-1">
              <!-- AGRUPANDO ICONE COM CAMPO DE DIGITAÇÃO -->
              <div class="input-group-prepend">
                <div class="input-group-text">
                  <!-- ICONE DO FONTAWESOME -->
                  <i class="fas fa-sort-numeric-up"></i>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <input type="text" name="item.id"
[(ngModel)]= "item.id" class="form-control form-control-sm" />
        </div>

        <div class="input-group mb-1">
            <div class="input-group-prepend">
                <div class="input-group-text">
                    <i class="fas fa-user"></i>
                </div>
            </div>
            <input type="text" name="item.nome"
[(ngModel)]= "item.nome" class="form-control form-control-sm" />
        </div>

        <div class="input-group mb-1">
            <div class="input-group-prepend">
                <div class="input-group-text">
                    <i class="fas fa-envelope"></i>
                </div>
            </div>
            <input type="text" name="item.email"
[(ngModel)]= "item.email" class="form-control form-control-sm" />
        </div>

        <button (click)="definirAluno(item)"
mdbBtn color="light-blue" size="sm" mdbWavesEffect>Definir
        Aluno</button>
        <br />
    </div>
</div>
<!-- MOSTRA O ALUNO QUE FOI CLICADO -->
{{aluno | json}}
</div>
</div>

<div class="col-md-4">
    <button mdbBtn color="primary" class="waves-light"
type="button" (click)="test2.toggle()" mdbWavesEffect>
        Lista de Disciplinas
    </button>

    <div mdbCollapse #test2="bs-collapse">

```

```

<div *ngFor="let linha of listaTurma">
  <div class="mb-3 mt-2 comp p-3">
    <div class="input-group mb-1">
      <div class="input-group-prepend">
        <div class="input-group-text">
          <i class="fas fa-sort-numeric-up"></i>
        </div>
      </div>
      <input type="number"
name="linha.idDisciplina" [(ngModel)]="linha.idDisciplina"
class="form-control form-control-sm" />
    </div>
    <div class="input-group mb-1">
      <div class="input-group-prepend">
        <div class="input-group-text">
          <i class="fas fa-book-open"></i>
        </div>
      </div>
      <input type="text"
name="linha.nomeDisciplina" [(ngModel)]="linha.nomeDisciplina"
class="form-control form-control-sm" />
    </div>
    <button
(click)="adicionarDisciplinas(linha)" mdbBtn color="light-green"
size="sm" mdbWavesEffect>Adiciona Disciplina</button>
  </div>
</div>
{{aluno.disciplinas | json}}
<br>
<button (click)="adicionarAluno(aluno)" mdbBtn
color="blue-grey" size="sm" mdbWavesEffect>Relacionar</button>
</div>
</div>

<div class="col-md-4 ">
  <button mdbBtn color="primary" class="waves-light"
type="button" (click)="test3.toggle()" mdbWavesEffect>
    Resposta Muitos Para Muitos
  </button>

  <div mdbCollapse #test3="bs-collapse">
    <!-- TABELA COM BORDA ATIVA -->

```

```

        <table mdbTable bordered="true">
        <!-- CABEÇALHO DA TABELA AZUL E TEXTO BRANCO -->
        <thead class="bg-info white-text">
            <tr>
                <th>IdAluno</th>
                <th>Nome</th>
                <th>Email</th>
                <th>Turmas</th>
            </tr>
        </thead>
        <tbody>
            <!-- LISTA OS ALUNOS -->
            <tr mdbTableCol *ngFor="let persona of alunos">
                <td>{{persona.id}}</td>
                <td>{{persona.nome}}</td>
                <td>{{persona.email}}</td>
                <td colspan="3">
                    <!-- LISTA AS DISCIPLINAS DO ALUNO -->
                    <div *ngFor="let classe of persona.disciplinas">
                        {{classe.nomeDisciplina}},
                    </div>
                </td>
            </tr>
        </tbody>
        </table>
    </div>
</div>
</div>

<div class="row">
    <!-- CARD COM ALINHAMENTO NO CENTRO (MX AUTO) -->
    <mdb-card class="col-md-5 mx-auto">
        <mdb-card-body>
            <mdb-card-title>
                <h4>Relatório</h4>
            </mdb-card-title>
            <mdb-card-text>
<pre>
                {{alunos | json}}
</pre>
            </mdb-card-text>
        </mdb-card-body>
    </div>

```

```

</mdb-card>

<mdb-card class="col-md-5 mx-auto">
  <mdb-card-body>
    <mdb-card-title>
      <h4>Mapa</h4>
    </mdb-card-title>
    <mdb-card-text>
      <!-- MAPA TRAZENDO PELA CHAVE -->
      <li *ngFor="let reg of mapa | keyvalue">
        {{reg.key | json}}
      </li>
    </mdb-card-text>
  </mdb-card-body>
</mdb-card>

</div>

</div>

```

aluno.component.css

```

.comp{
  border: 1px solid #ccc;
  border-radius: 5px;
}

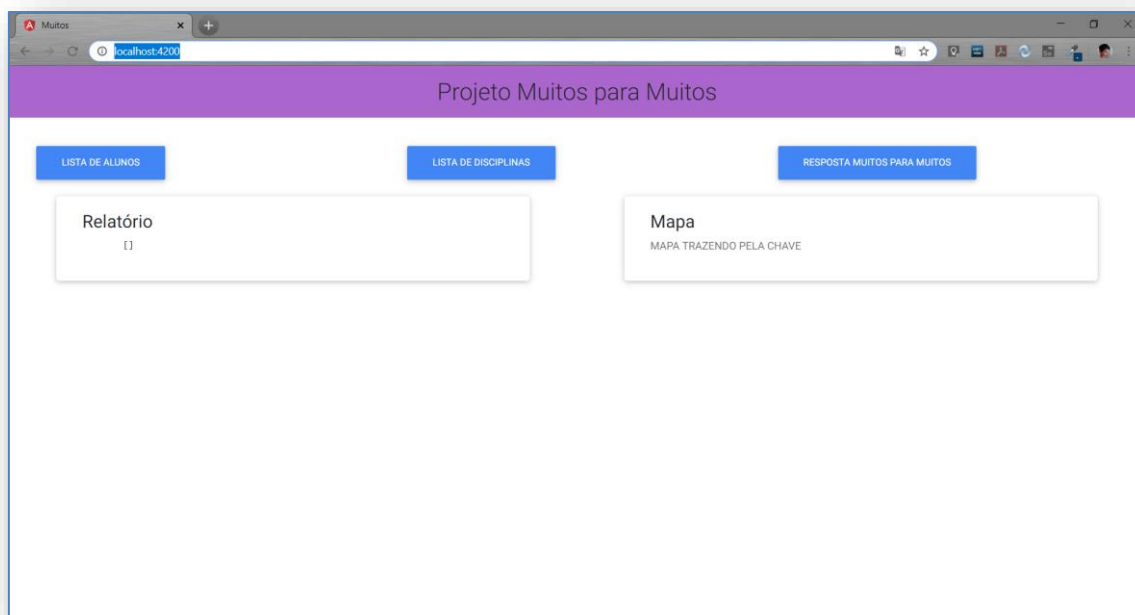
```

Rodando o projeto.

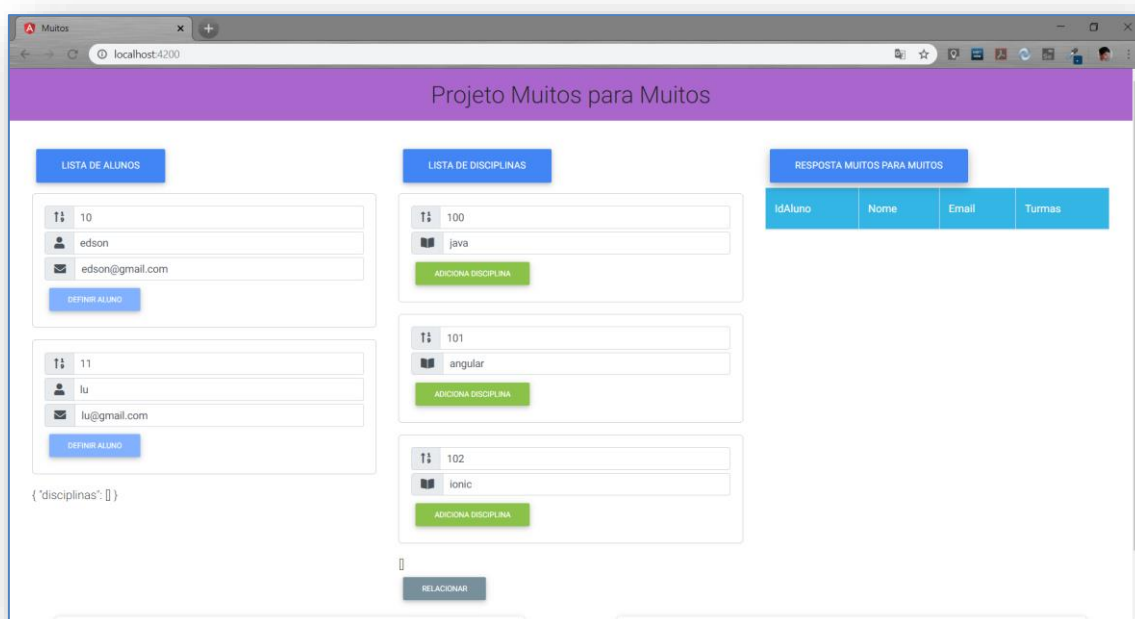
Digitar no terminal:

"ng s -o"

<http://localhost:4200/>



Clicando nos botões para abrir os formulários



Selecionando um aluno

Projeto Muitos para Muitos

LISTA DE ALUNOS

Id: 10
Nome: edson
Email: edson@gmail.com
DEFINIR ALUNO

Id: 11
Nome: lu
Email: lu@gmail.com
DEFINIR ALUNO

`{ "id": 11, "nome": "lu", "email": "lu@gmail.com", "disciplinas": [] }`

LISTA DE DISCIPLINAS

Id: 100
Nome: java
ADICIONAR DISCIPLINA

Id: 101
Nome: angular
ADICIONAR DISCIPLINA

Id: 102
Nome: ionic
ADICIONAR DISCIPLINA

RESPOSTA MUITOS PARA MUITOS

IdAluno	Nome	Email	Turmas
---------	------	-------	--------

RELACIONAR

Selecionando a(s) disciplina(s)

Projeto Muitos para Muitos

LISTA DE ALUNOS

Id: 10
Nome: edson
Email: edson@gmail.com
DEFINIR ALUNO

Id: 11
Nome: lu
Email: lu@gmail.com
DEFINIR ALUNO

`{ "id": 11, "nome": "lu", "email": "lu@gmail.com", "disciplinas": [{ "idDisciplina": 100, "nomeDisciplina": "java" }, { "idDisciplina": 101, "nomeDisciplina": "angular" }] }`

LISTA DE DISCIPLINAS

Id: 100
Nome: java
ADICIONAR DISCIPLINA

Id: 101
Nome: angular
ADICIONAR DISCIPLINA

Id: 102
Nome: ionic
ADICIONAR DISCIPLINA

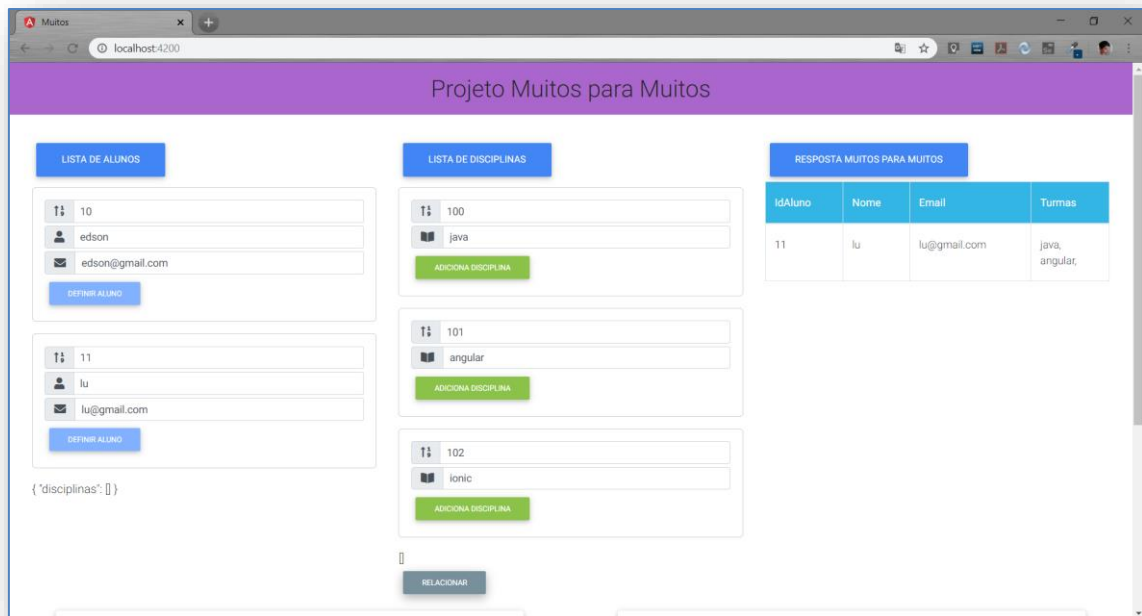
RESPOSTA MUITOS PARA MUITOS

IdAluno	Nome	Email	Turmas
---------	------	-------	--------

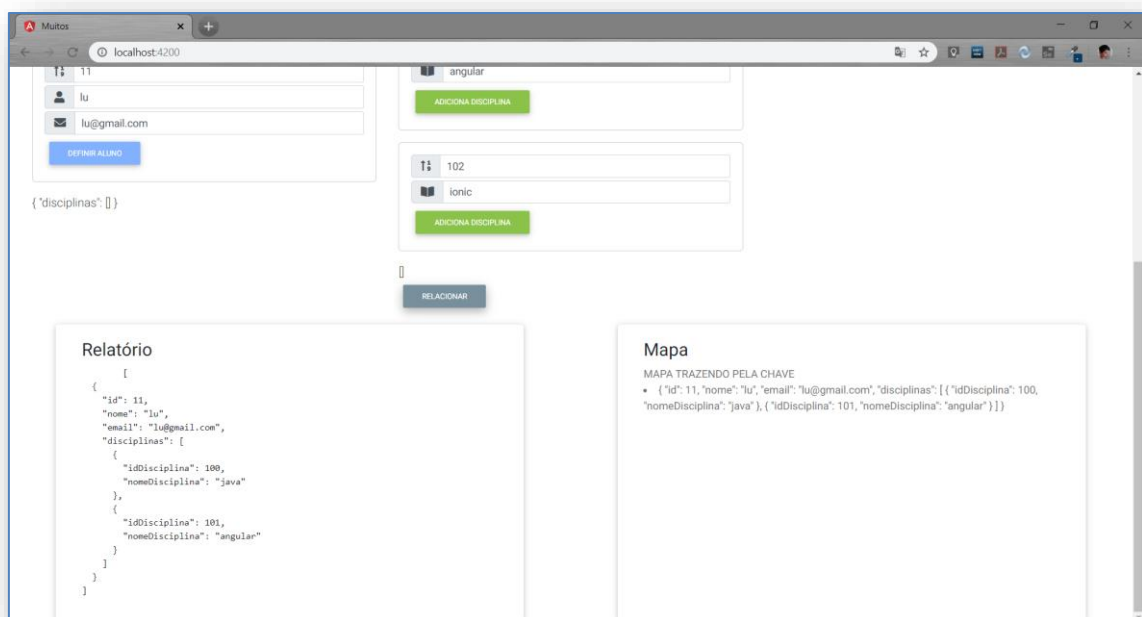
`[{ "idDisciplina": 100, "nomeDisciplina": "java" }, { "idDisciplina": 101, "nomeDisciplina": "angular" }]`

RELACIONAR

Clicando em relacionar, mostra na tabela

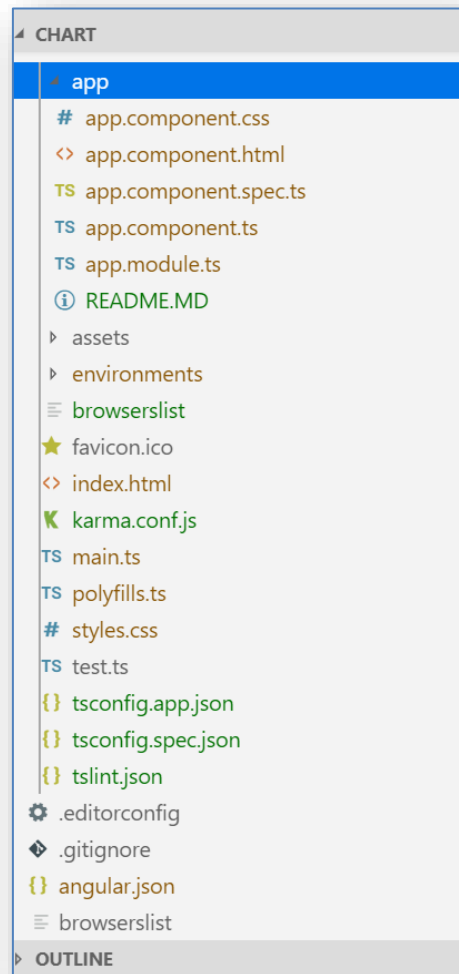


Mostra o relatório em json e o mapa



NOVO PROJETO COM VIDEO

Estrutura do projeto depois de finalizado:



Fazer as seguintes instalações:

```
npm i npm-registry-client  
ng add angular-bootstrap-md  
npm install jarallax --save
```

Para o Video funcionar temos que informar algumas bibliotecas.

No arquivo ANGULAR.JSON, informar :

Alterar styles:

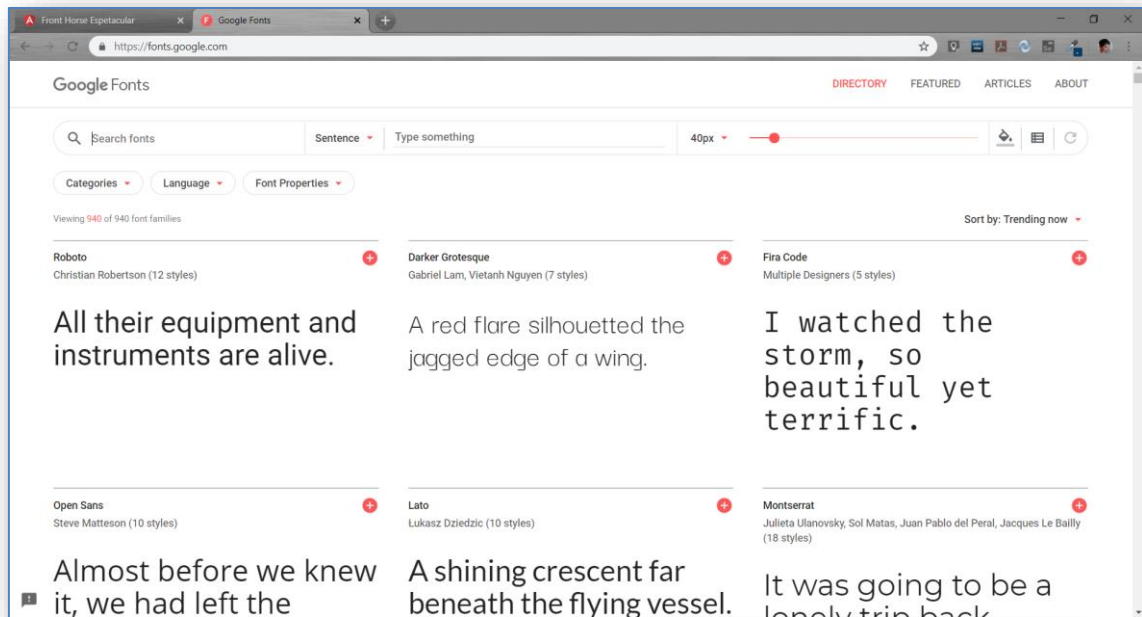
```
"styles": [  
  "src/styles.css",  
  "node_modules/@fortawesome/fontawesome-free/scss/fontawesome.scss",  
  "node_modules/@fortawesome/fontawesome-free/scss/solid.scss",  
  "node_modules/@fortawesome/fontawesome-free/scss/regular.scss",  
  "node_modules/@fortawesome/fontawesome-free/scss/brands.scss",  
  "node_modules/angular-bootstrap-md/assets/scss/bootstrap/bootstrap.scss",  
  "node_modules/angular-bootstrap-md/assets/scss/mdb.scss",  
  "node_modules/animate.css/animate.css",  
  "node_modules/jarallax/dist/jarallax.css"  
],
```

Alterar script:

```
"scripts": [  
  "node_modules/chart.js/dist/Chart.js",  
  "node_modules/hammerjs/hammer.min.js",  
  "node_modules/jarallax/dist/jarallax.js",  
  "node_modules/jarallax/dist/jarallax-video.js"  
]  
},
```

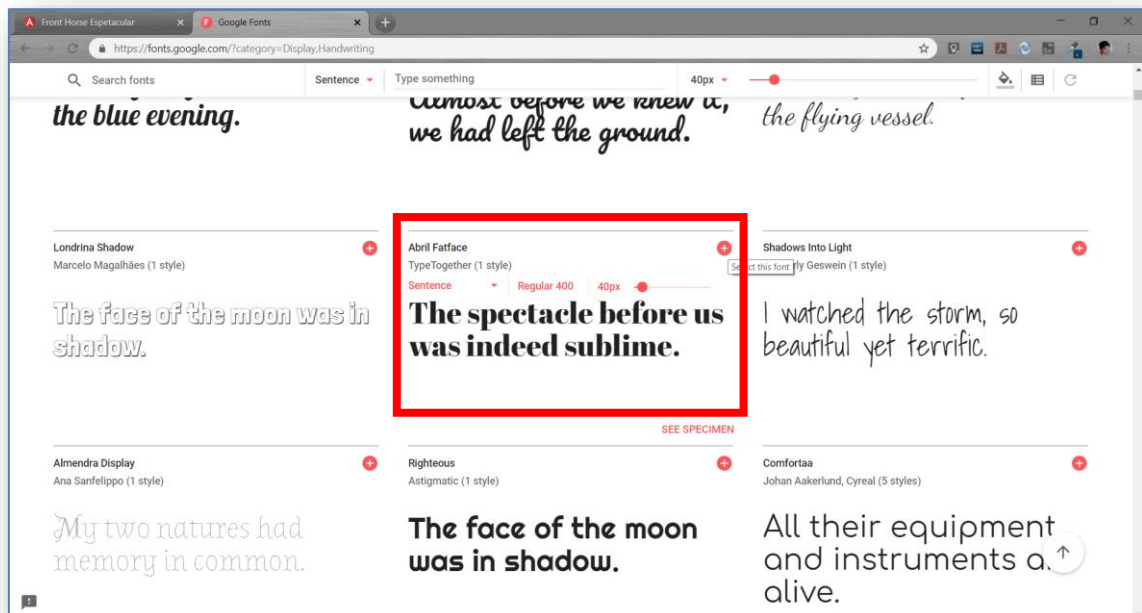
Fontes Online do Google.

<https://fonts.google.com>



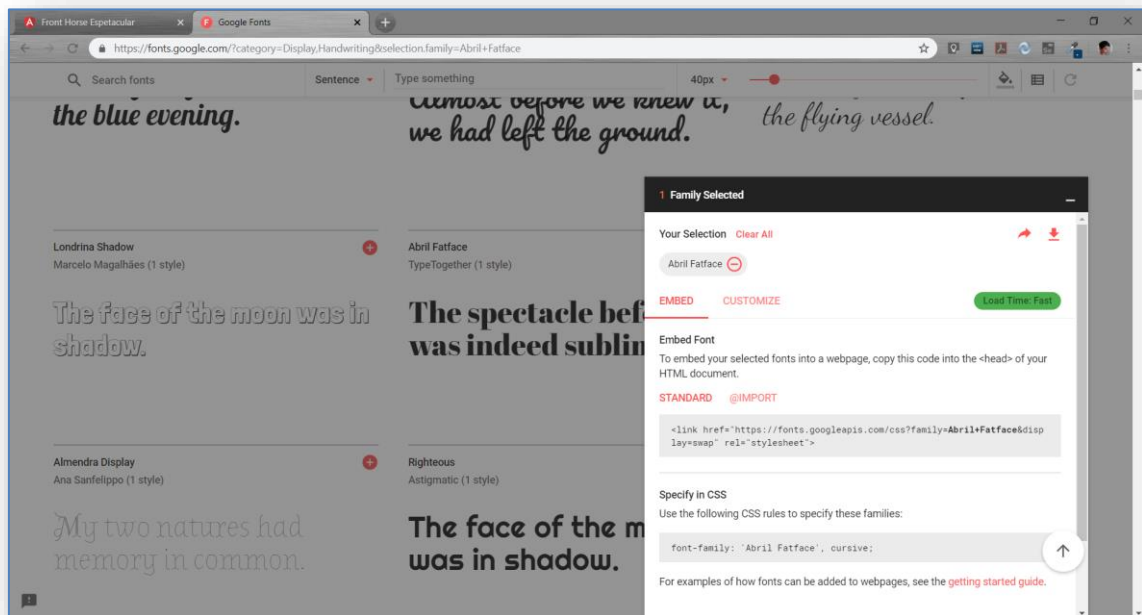
Escolher um tipo de fonte e clicar no sinal de “+”.

Escolhi a Abril Fatface.

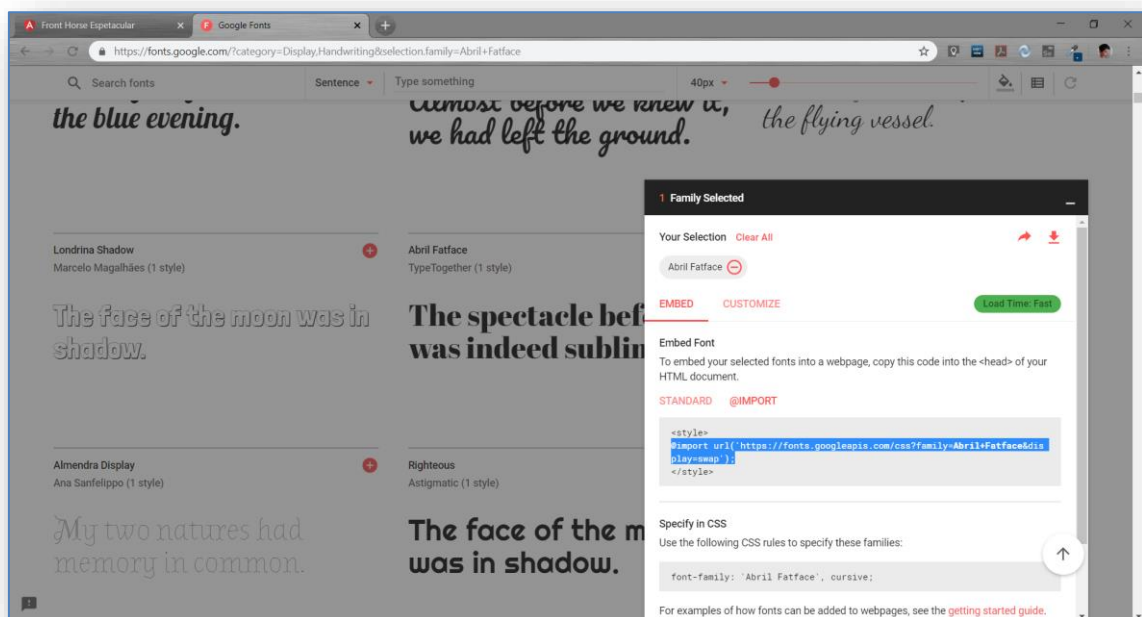


Clicar no botão de “+”

Abrirá uma janela com as informações para adicionar ao projeto



Copiaremos o import para adicionar ao arquivo STYLES.CSS



styles.css

```
@import
url('https://fonts.googleapis.com/css?family=Abril+Fatface&display=swap');
```

app.component.ts

```
import { Component, ViewChild, ElementRef } from
'@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {

  public chartType: string = 'bar';
  // OUTROS TIPOS DE GRAFICO = pie, doughnut

  //DADOS DO GRAFICO
  public chartDatasets: Array<any> = [
    {
      data: [0, 85.46, 38.15, 20.21, 17.78, 26.38, 15.66],
      label: 'Bilheteria no Brasil em milhões 2019'
    }
  ];

  //NOME DAS LEGENDAS
  public chartLabels: Array<any> = [
    '', 'Vingadores: Ultimato', 'Capitã Marvel', 'Alladin',
    'Homem Aranha: Longe de Casa', 'Toy Story 4', 'Como Treinar seu
    Dragão 3'
  ];

  //CORES DE FUNDO E BORDAS
  public chartColors: Array<any> = [
    {
```



```

        backgroundColor: [
            '', '#d63031', '#6c5ce7', '#55efc4', '#ff7675',
            '#fdcb6e', 'rgba(255, 159, 64, 0.2)'
            //PARA FICAR TRANSPARENTE
        ],
        borderColor: [
            '', '#000', '#6c5ce7', '#55efc4', '#ff7675',
            '#fdcb6e', 'rgba(255, 159, 64, 0.2)'
        ],
        borderWidth: 2,
    }
];

//OPÇÃO DE RESPONSIVO
public chartOptions: any = {
    responsive: true
};
}

```

app.component.html

```

<!-- CABEÇALHO BARRA DE NAVEGAÇÃO -->
<header>
    <!-- BARRA DE NAVEGAÇÃO EXPANDIDA, LETRA BRANCA, FUNDO
    UNIQUE-COLOR, FIXA NO TOPO -->
    <nav class="navbar navbar-expand-lg navbar-dark unique-color
    fixed-top">
        <!-- NOME DO SITE -->
        <a class="navbar-brand" href="#">
            Filmes Show
        </a>
        <!-- LINKS DA BARRA ALINHADOS A ESQUERDA -->
        <ul class="navbar-nav mr-auto">
            <!-- LINK PRINCIPAL ATIVO -->
            <li class="nav-item active">
                <a href="#" class="nav-link mr-3">Home<i
class="fas fa-home ml-1"></i></a>
            </li>

```

```

        <li class="nav-item">
            <a href="#" class="nav-link mr-3">Filmes<i
class="fas fa-video ml-1"></i></a>
        </li>

        <li class="nav-item">
            <a href="#" class="nav-link mr-3">Papéis de
parede<i class="fas fa-images ml-1"></i></a>
        </li>
    </ul>
</nav>
</header>

<!-- BANNER COM VIDEOE JARALLAX, INICIA DEPOIS DE 1 SEGUNDO-->
<div class="bannervideo jarallax" data-jarallax='{ "speed":1}'
    data-jarallax-video="https://www.youtube.com/watch?v=yUXd-
enst08">
    <!-- DEFINE VIDEO COMOS BACKGROUND -->
    <div class="full-bg-img">
        <!-- TEXTO DO CONTAINER ALINHAMENTO CENTRAL -->
        <div class="container flex-center">
            <div class="col-md-12">
                <!-- TEXTO ALINHADO NO CENTRO, COR BRANCO,
MARGEM INFERIOR 20PX EFEITO WOW DE SURGIR DE CIMA PARA BAIXO COM
DELAY DE 3 SEGUNDOS-->
                <h1 class="text-center text-white mb-2 animated
wow fadeInDown video" data-wow-delay='3s'>
                    Filme Espetacular
                </h1>
            </div>
        </div>
    </div>
</div>

<!-- CONTAINER PARA OS FILMES COM FUNDO UNIQUE-COLOR MARGEM
INTERNA INFERIOR DE 50PX-->
<div class="container-fluid filmes unique-color-dark pb-5">
    <!-- TEXTO COM EFEITO DE SURGIR -->
    <h1 class="p-5 text-white text-center animated wow fadeIn">
        Filmes Preferidos
    </h1>

```

```

<!-- DEFINE UMA LINHA PARA ALINHAR AS IMAGENS -->
<div class="row">
    <!-- DEFINE ESPAÇO DE 3 COLUNAS PARA CADA IMAGEM -->
    <div class="col-md-3">
        <!-- IMAGEM COM EFEITO DE SURGIR PELA ESQUERDA COM
        DELAY E MOSTRA NOME AO PASSAR O MOUSE -->
        
    </div>
    <div class="col-md-3">
        <!-- EFEITO DE ROTACIONAR A IMAGEM -->
        
    </div>
    <div class="col-md-3">
        <!-- EFEITO DE SURGIR DE CIMA -->
        
    </div>
    <div class="col-md-3">
        
    </div>
</div>
</div>

<!-- CONTAINER DO GRAFICO COM FUNDO CINZA -->
<div class="container-fluid p-5 bg-light">
    <!-- TEXTO COM EFEITO DE SURGIR DE BAIXO -->
    <h1 class="p-5 text-center animated wow fadeInUp">
        Gráfico dos Filmes
    </h1>
    <div class="col-md-10 offset-md-1 bg-white p-5">
        <!-- CONFIGURAÇÕES DO GRAFICO -->

```

```
        <canvas mdbChart [chartType]="chartType"
[datasets]="chartDatasets" [labels]="chartLabels"
        [colors]="chartColors" [options]="chartOptions"
[legend]="true">
        </canvas>
    </div>
</div>
```

app.component.css

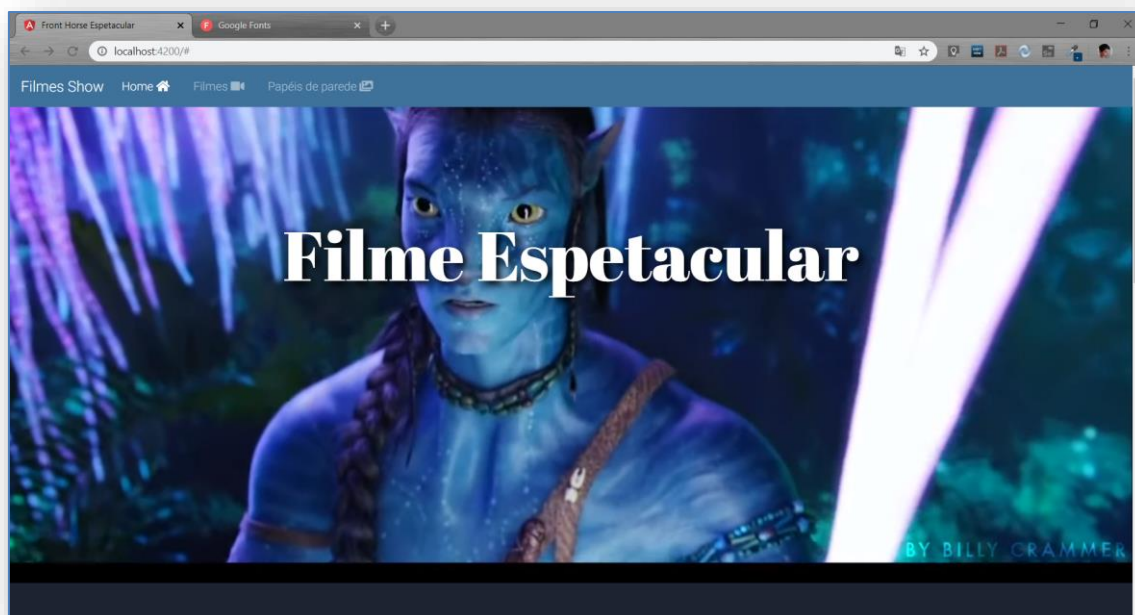
```
.bannervideo{
    background-position: center center; /* CENTRALIZADO NA
HORIZONTAL E NA VERTICAL */
    background-repeat: no-repeat; /* FNDO NAO SE REPETE */
    height: 700px; /* ALTURA DO VIDEO */
}
.video{
    margin-top: 200px; /* MARGEM ACIMA */
    font-family: 'Abril Fatface', cursive; /* LETRA DEFINIDA NO
STYLES.CSS */
    font-size: 70pt; /* TAMANHO DA LETRA PARA O TEXTO*/
    text-shadow: 5px 5px 10px #000; /* SOMBRA NO TEXTO */
/* ANOTAR LETRA EM STYLES */
}
.filmes img{
    width: 300px; /* AS IMAGENS TERÃO LARGURA DE 300PX */
}
```

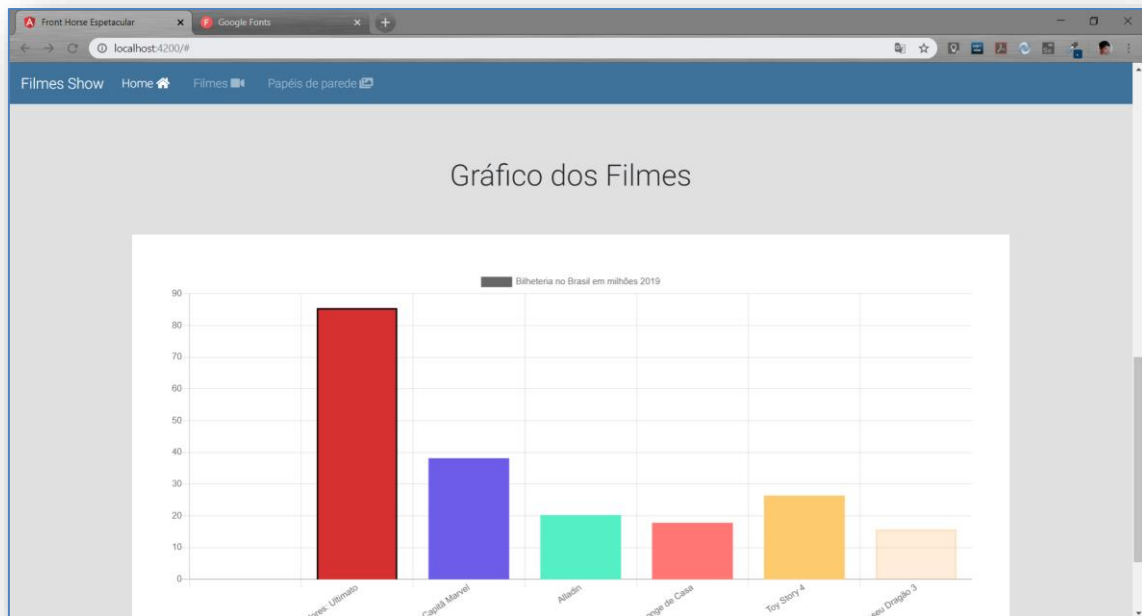
Rodando o projeto.

Digitar no terminal:

“ng s -o”

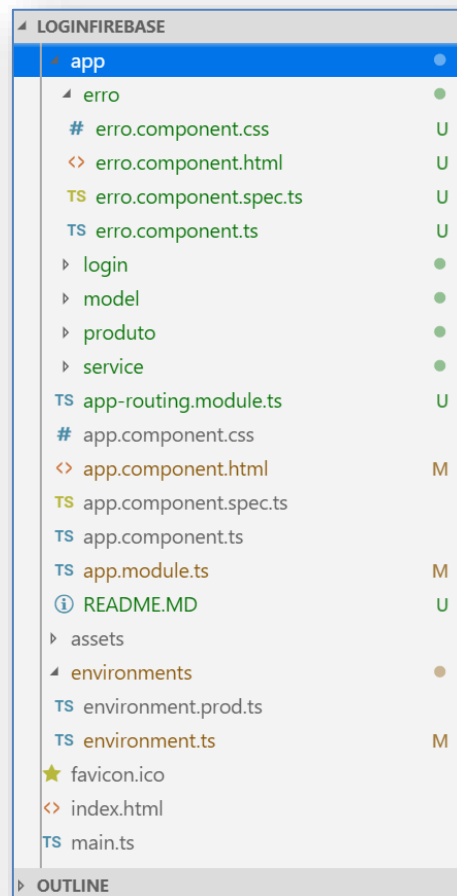
<http://localhost:4200/>





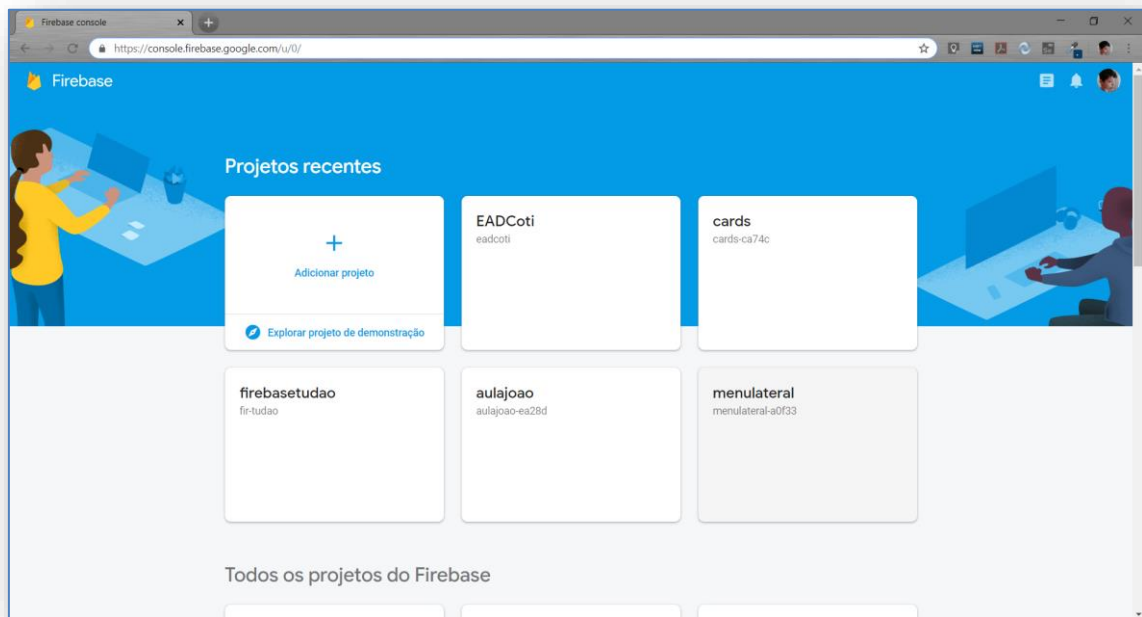
NOVO PROJETO LOGIN FIREBASE

Estrutura do projeto depois de finalizado:

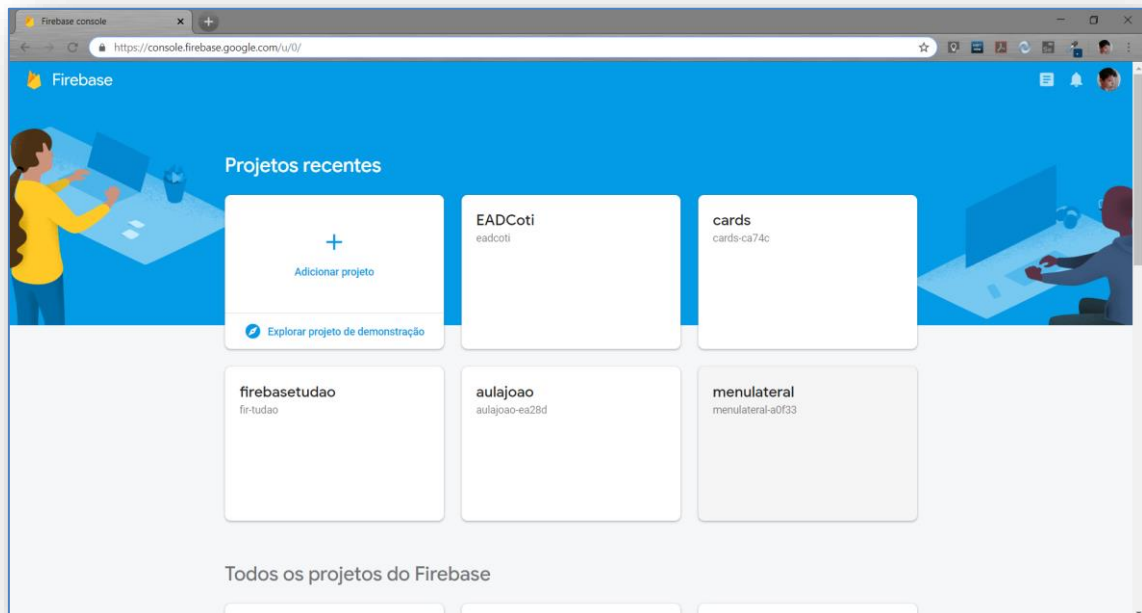


Firestore Console

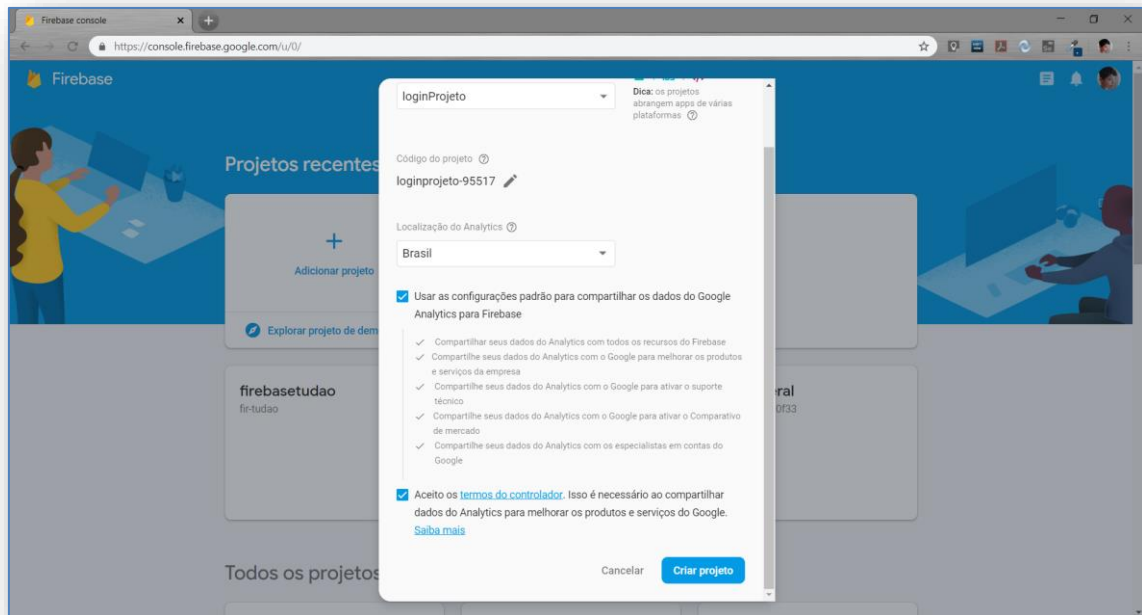
<https://console.firebase.google.com/u/0/>



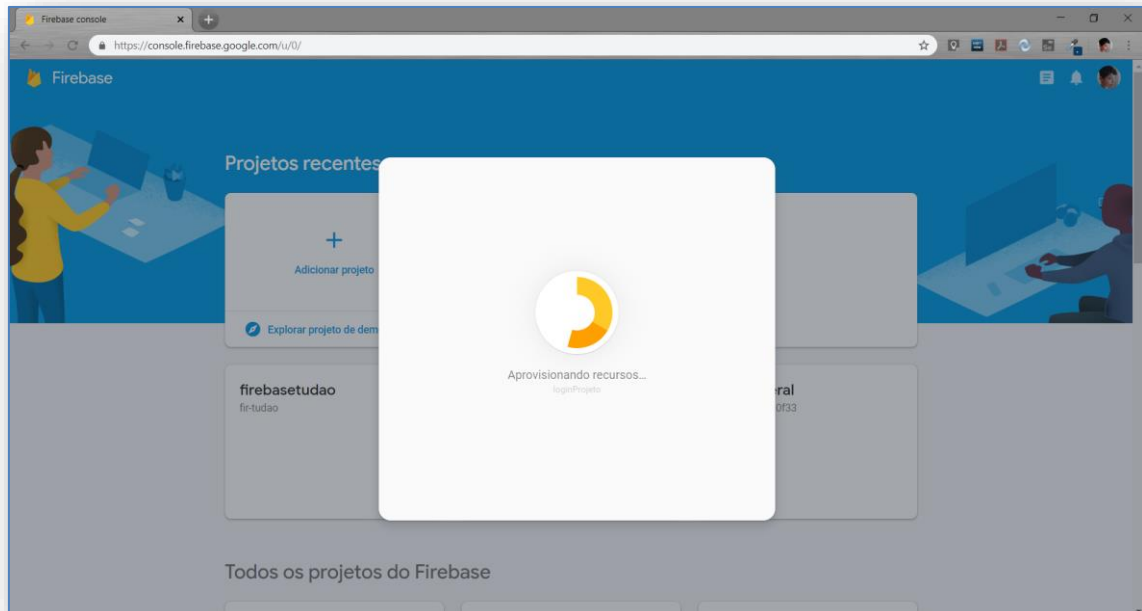
Criar um projeto. Clicar em adicionar projeto



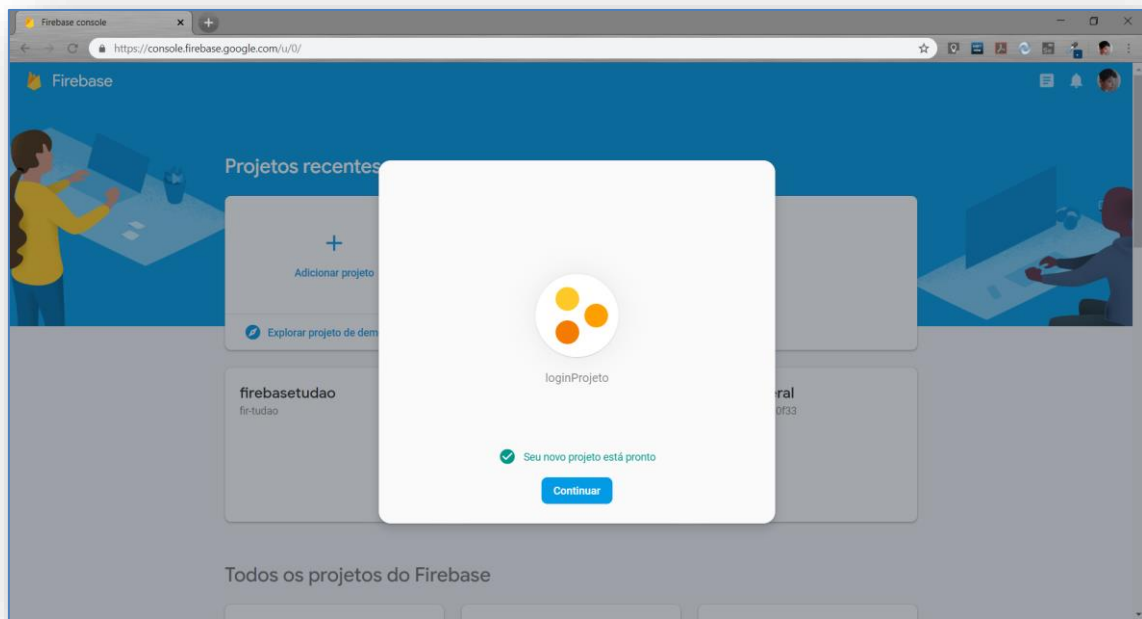
Digitar o nome do projeto, selecionar Brasil, aceitar as condições e Clicar em criar projeto



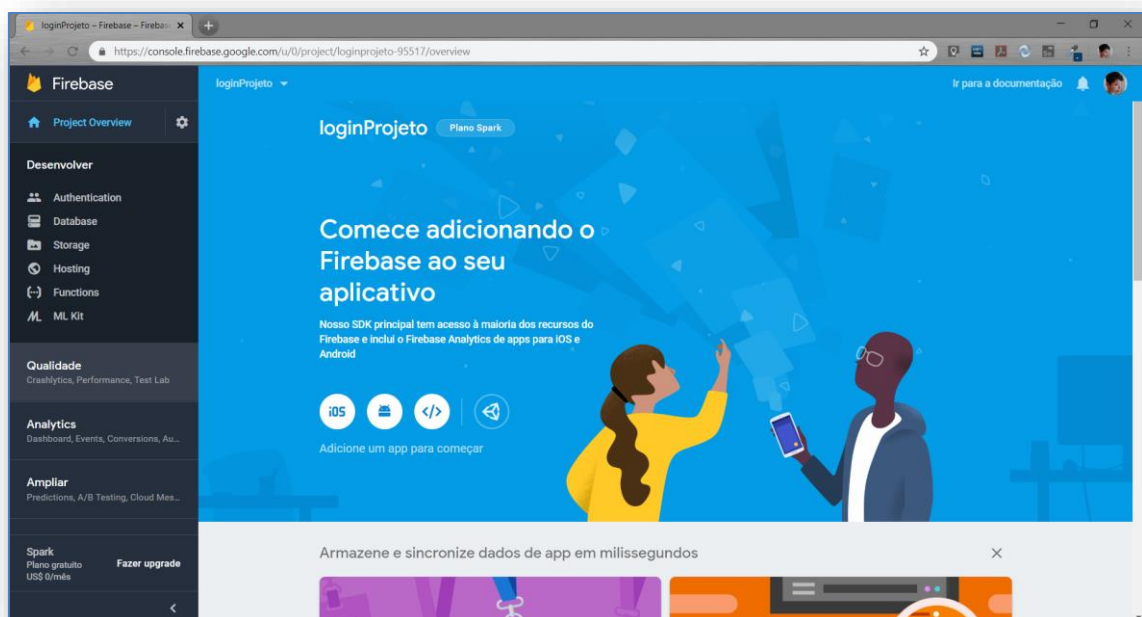
Aguardar criar o projeto



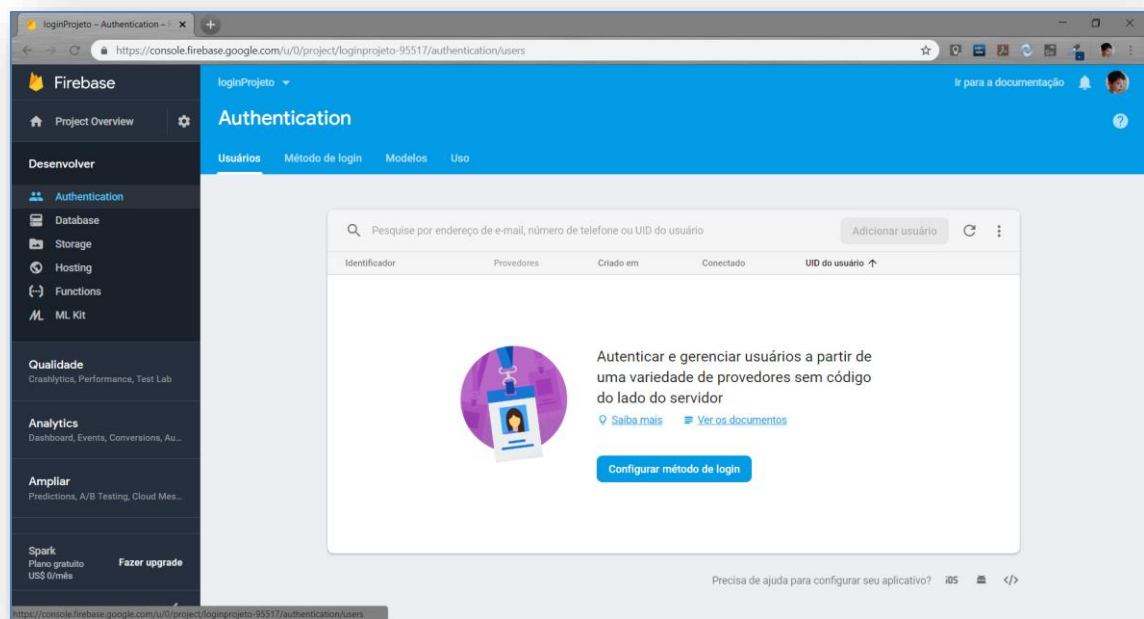
Projeto criado, clicar em continuar



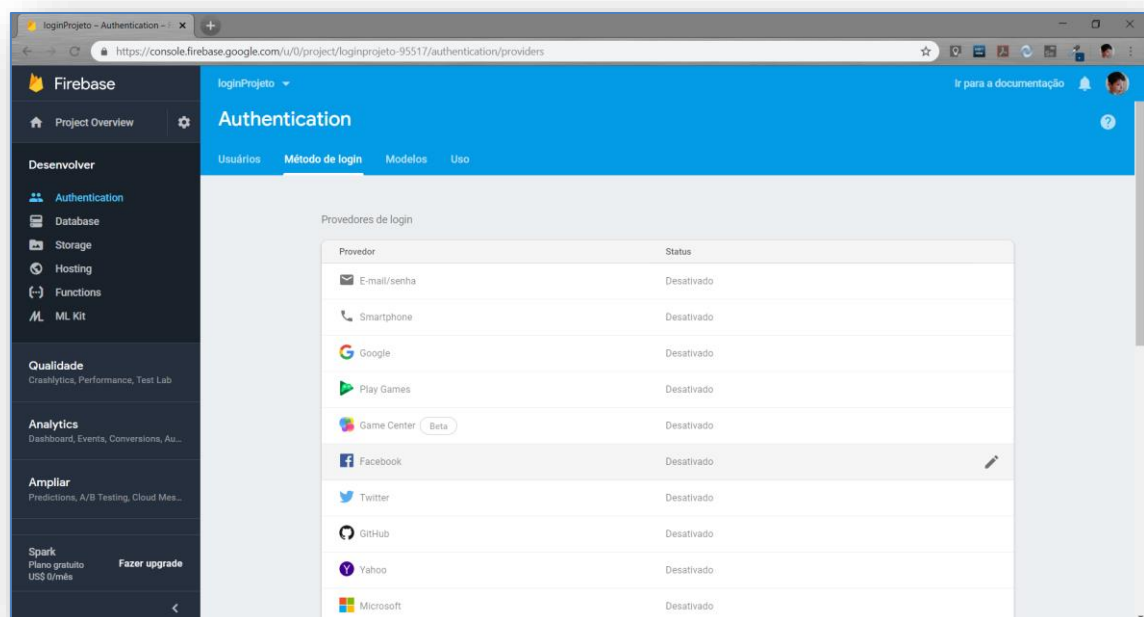
Fomos direcionados para o painel de controle do projeto



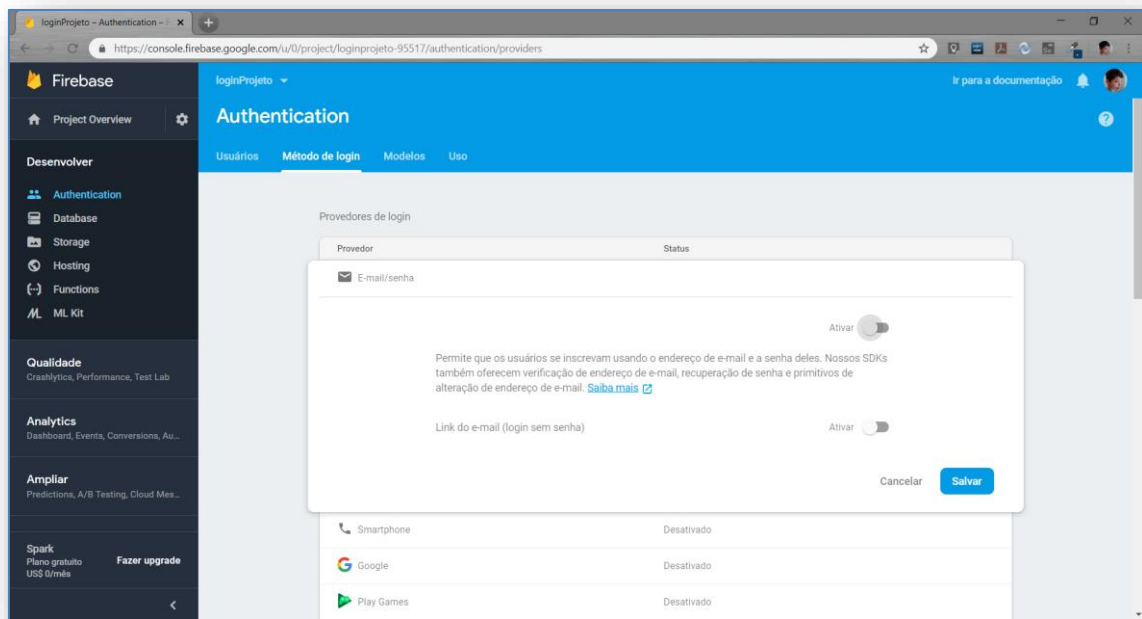
Clicar em autenticação



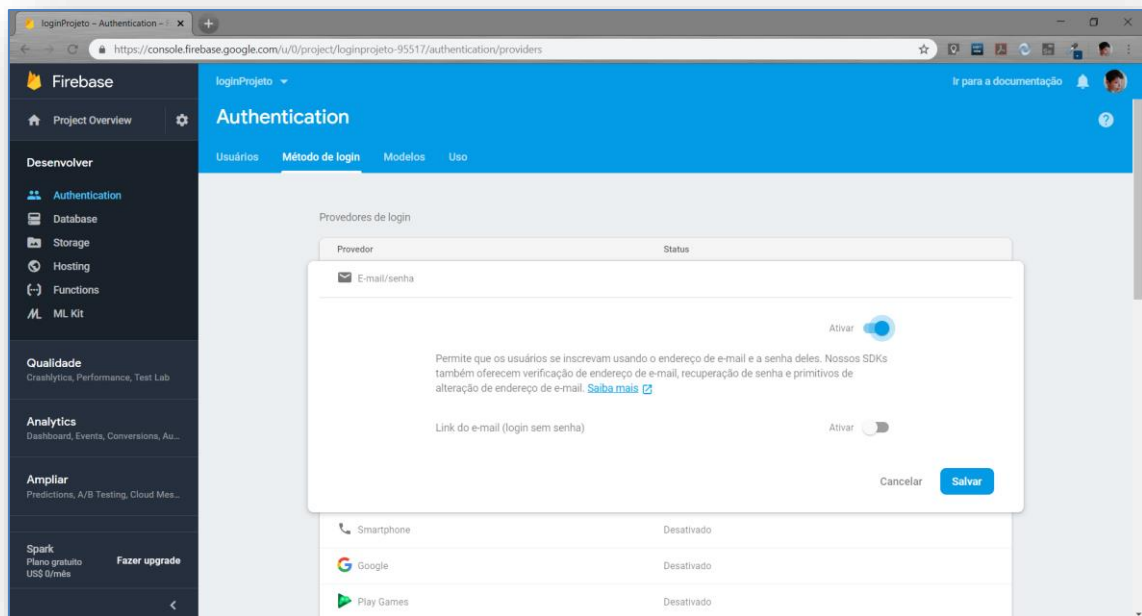
Configurar método de login



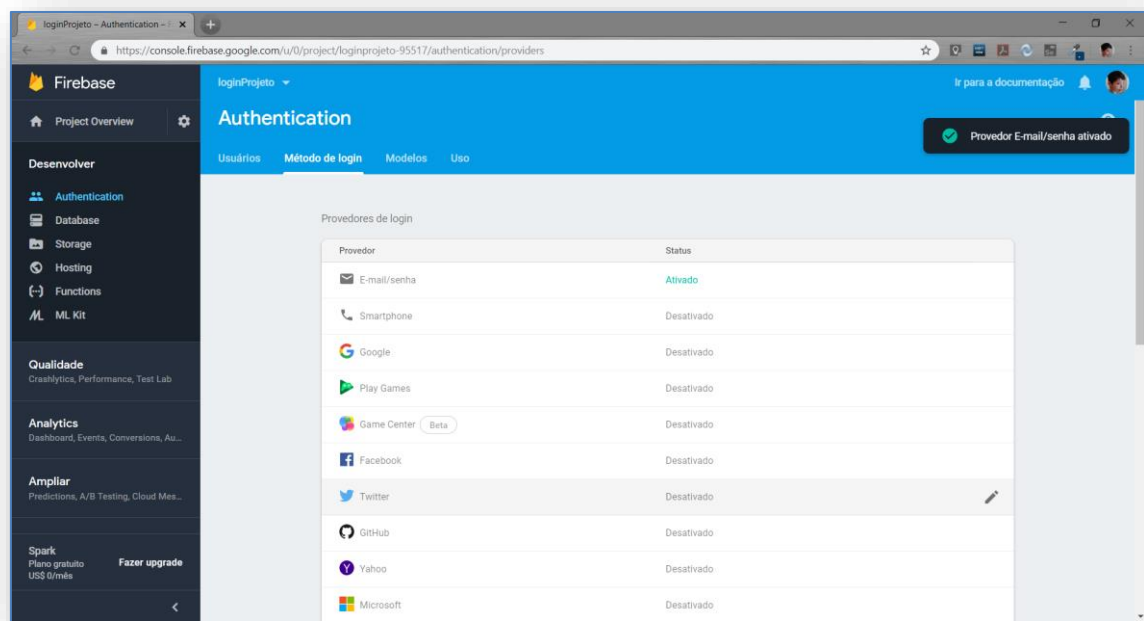
Clicar em email/senha



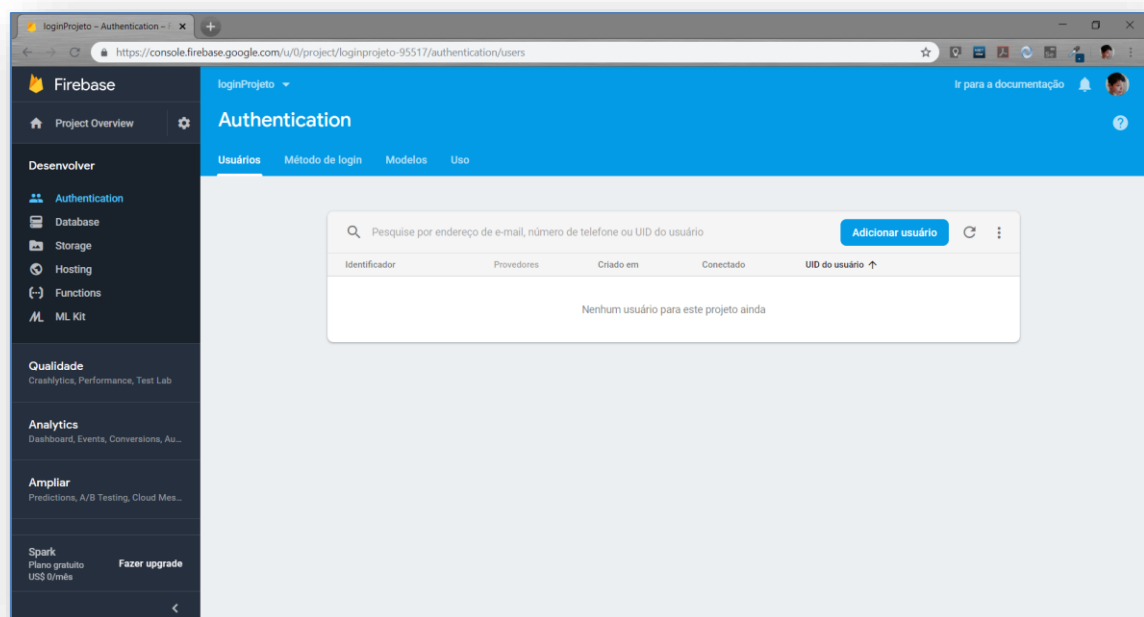
Clicar em ativar e depois clicar em salvar



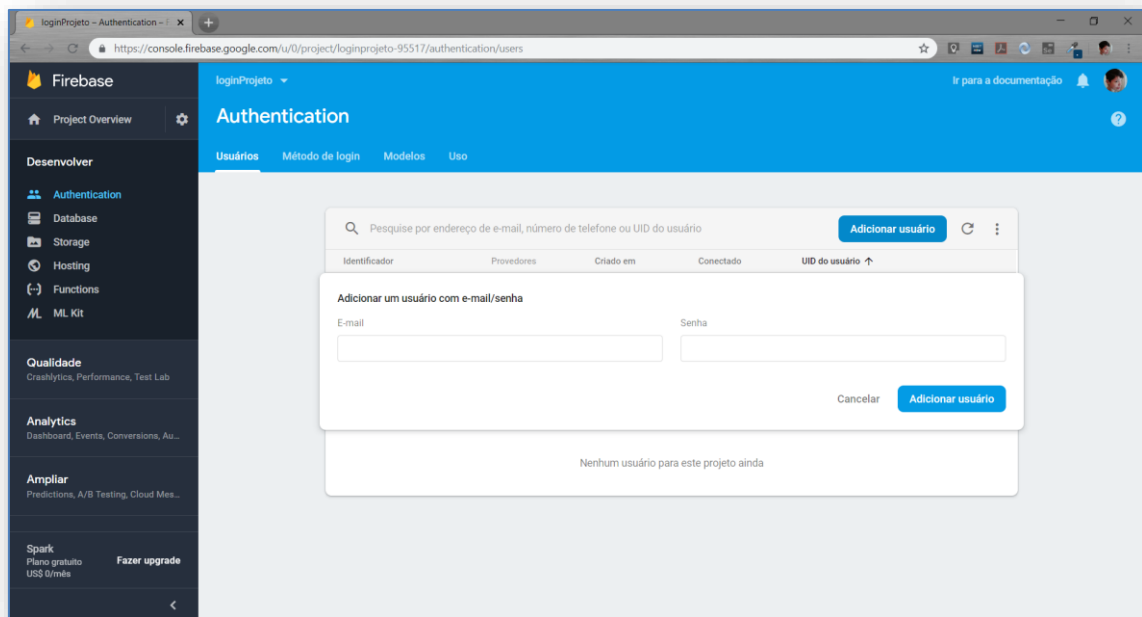
O método de login por email foi ativado



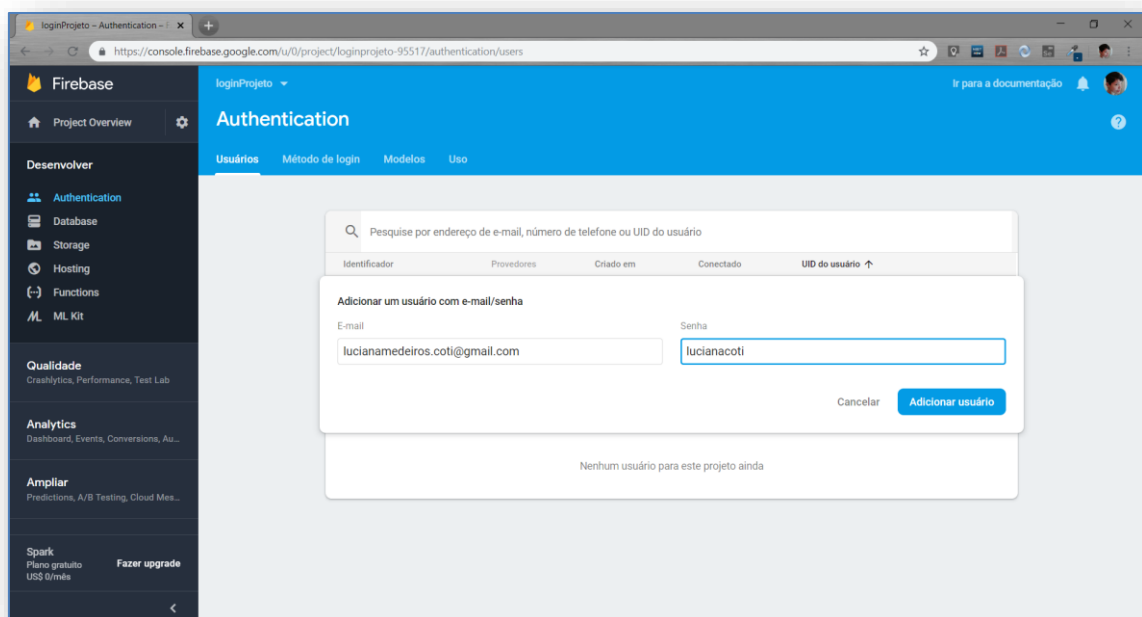
Clicar na aba usuario



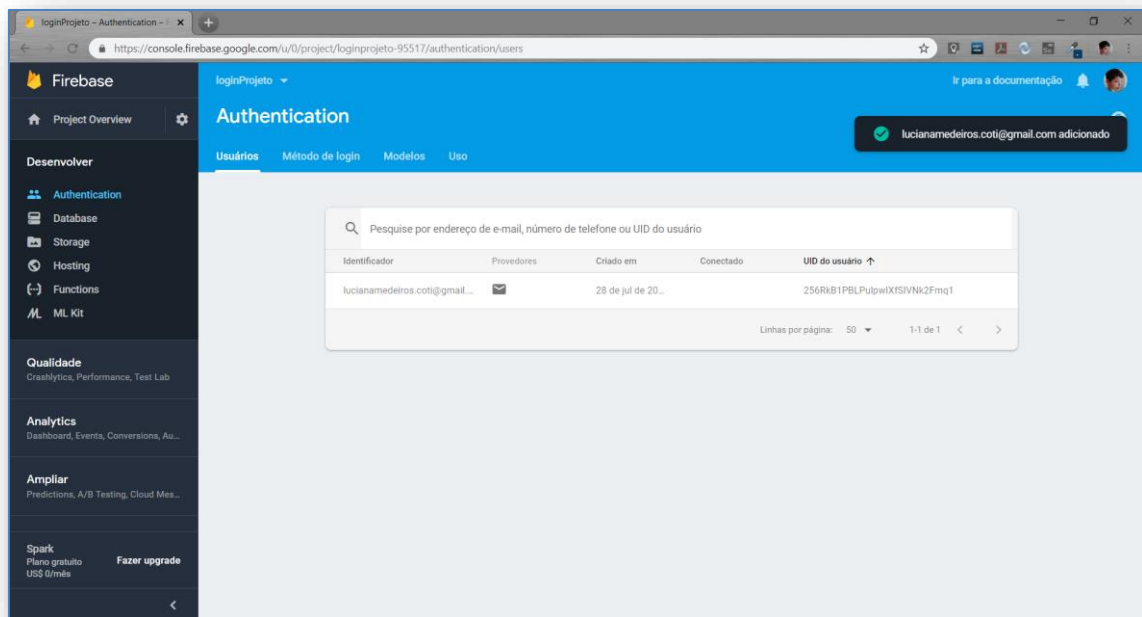
Clicar em adicionar usuario



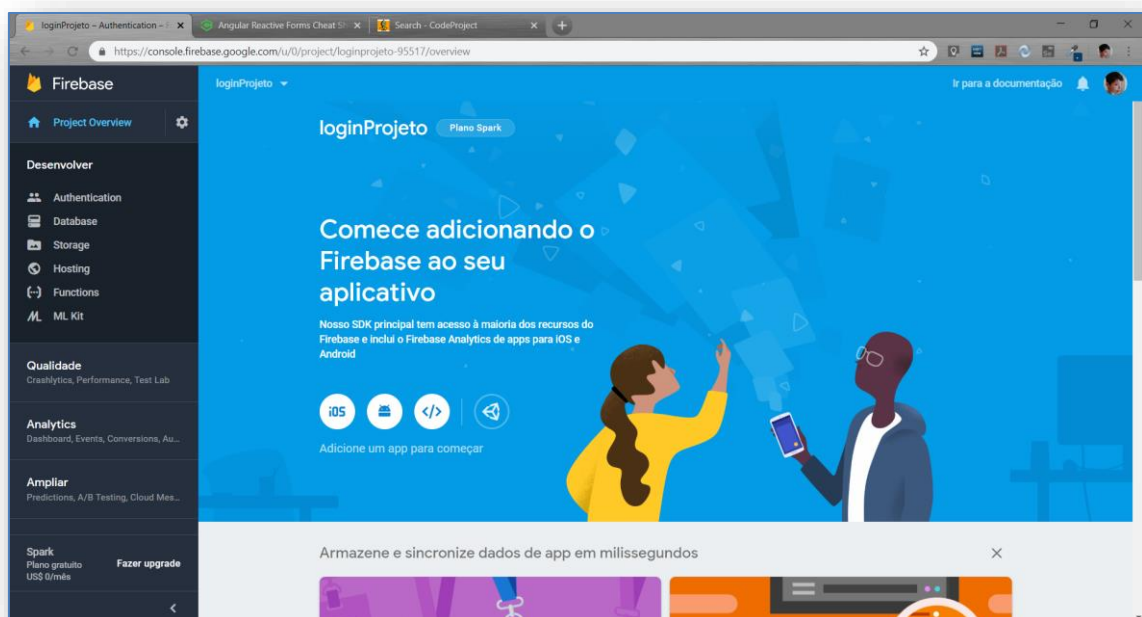
Digitar os dados do usuario administrador (email valido e uma senha para logar no sistema). Clicar em adicionar usuario



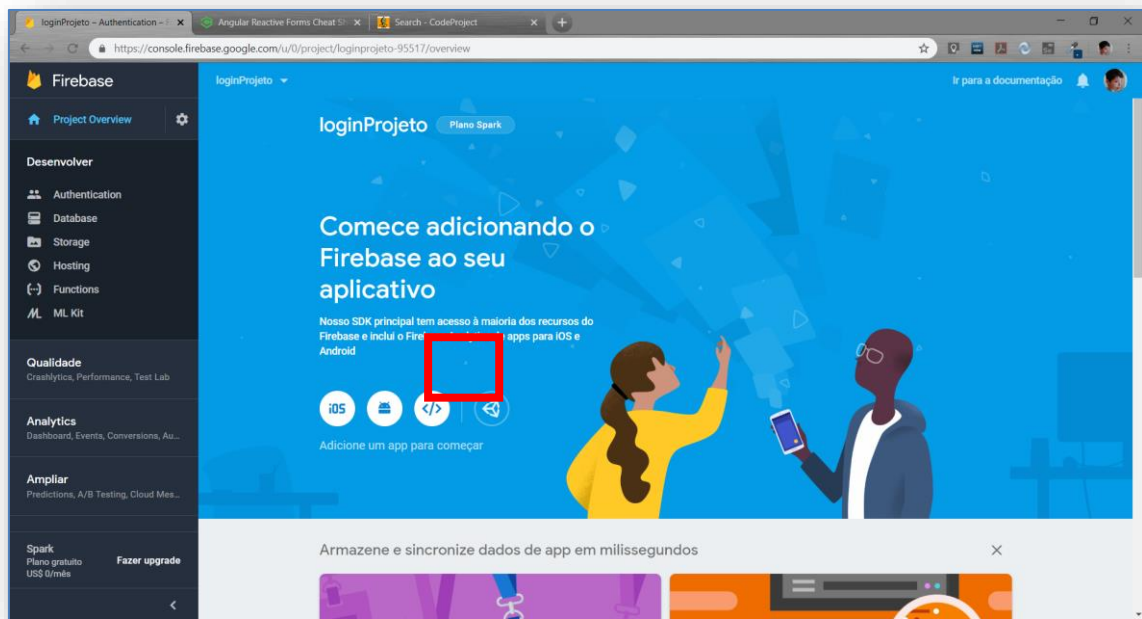
Usuario foi adicionado ao projeto



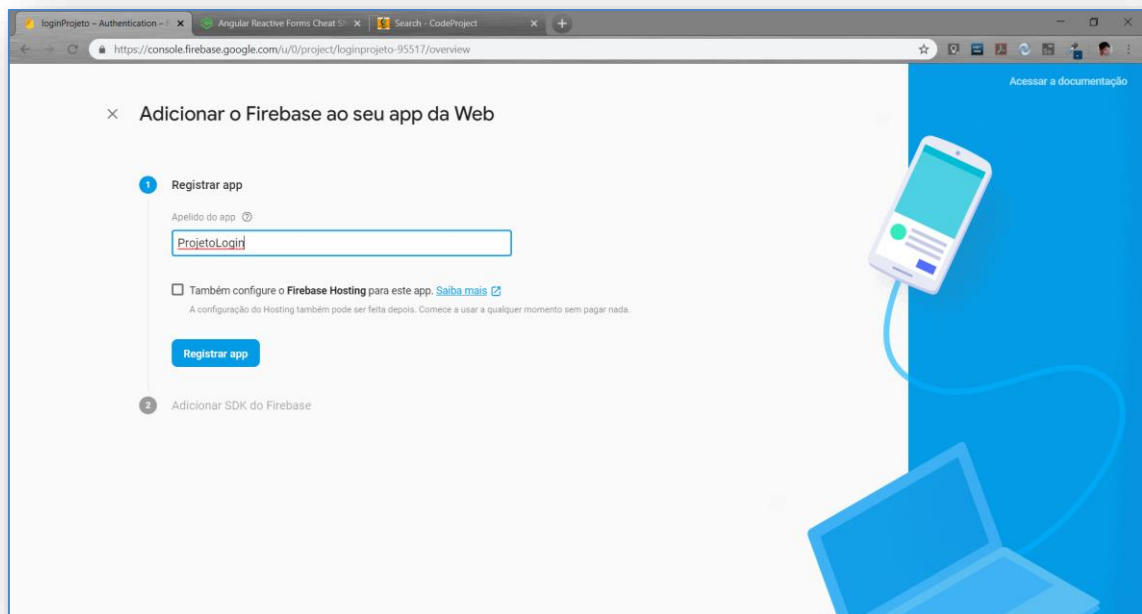
Para pegar a autenticação do projeto.
Clicar em Project Overview



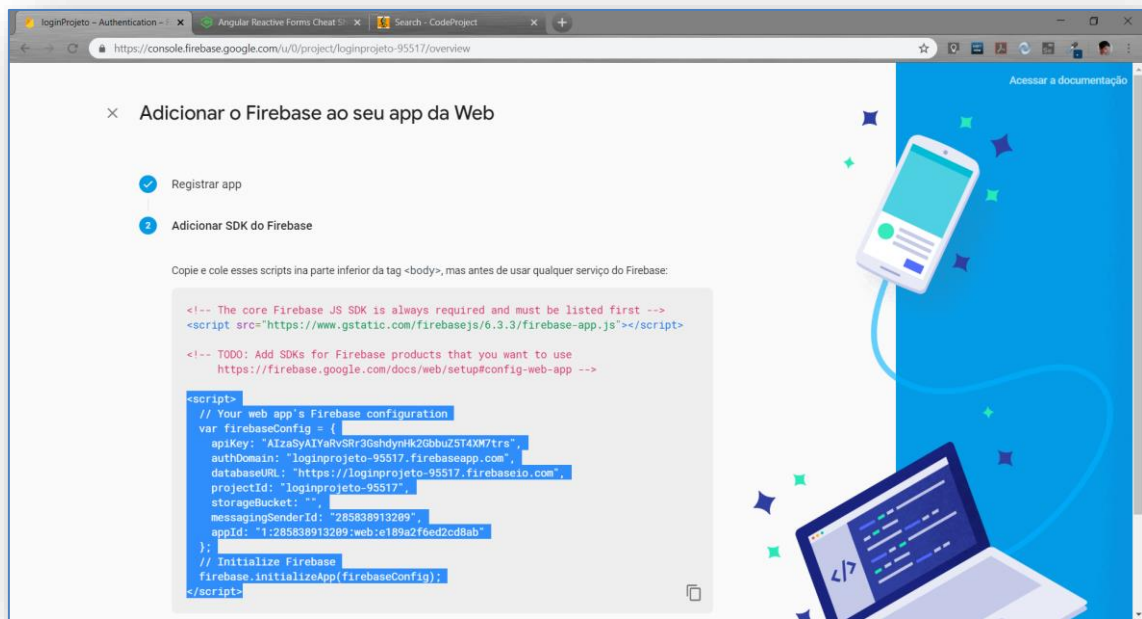
Clicar no ícone de web



Digitar um apelido para o projeto. Clicar em registrar o app

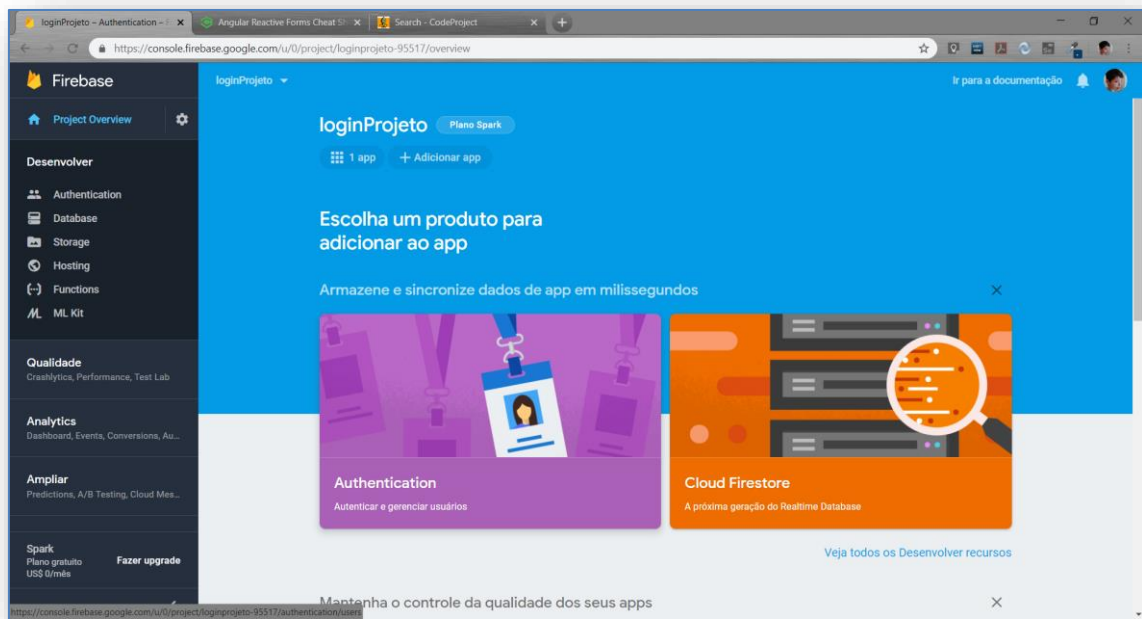


Copiar o SDK do seu projeto. Iremos utilizar na classe de acesso.



```
<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyAIYaRvSRr3GshdynHk2GbbuZ5T4XM7trs",
    authDomain: "loginprojeto-95517.firebaseio.com",
    databaseURL: "https://loginprojeto-95517.firebaseio.com",
    projectId: "loginprojeto-95517",
    storageBucket: "",
    messagingSenderId: "285838913209",
    appId: "1:285838913209:web:e189a2f6ed2cd8ab"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Clicar em continuar no console.

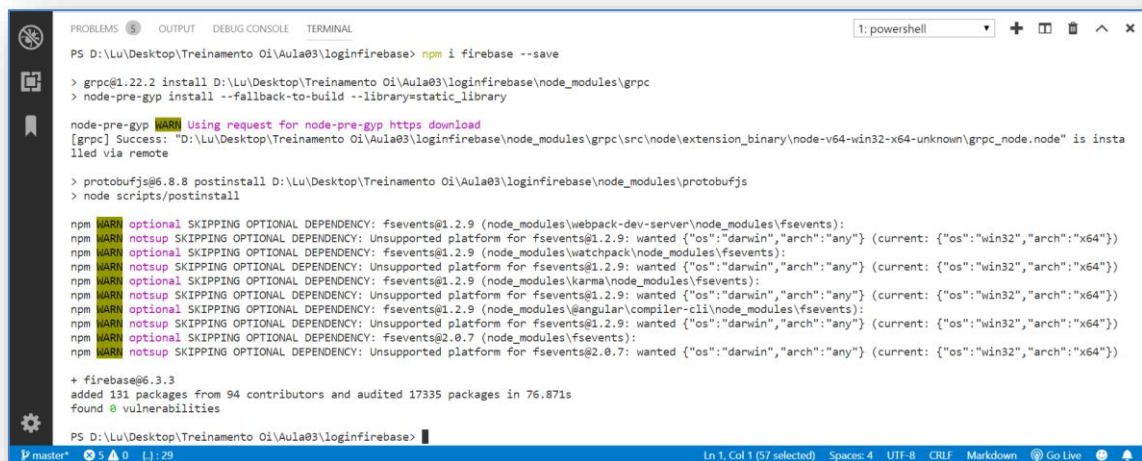


Fazer as seguintes instalações para acessar ao Firebase

```
npm i firebase --save
```

```
npm i firebase angularfire2 --save
```

```
npm i firebase --save
```



```
npm i firebase angularfire2 --save
```



```
PS D:\Lu\Desktop\Treinamento 01\Aula03\loginfirebase> npm i firebase angularfire2 --save
npm WARN deprecated angularfire2@5.2.1: AngularFire has moved, we're now @angular/fire
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\watchpack\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\karma\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.0.7 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.0.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\angular\compiler-cli\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ angularfire2@5.2.1
+ firebase@6.3.3
added 2 packages from 1 contributor, updated 1 package and audited 17337 packages in 29.497s
found 0 vulnerabilities

PS D:\Lu\Desktop\Treinamento 01\Aula03\loginfirebase>
```

login.ts

```
export class Login {

    email: string;
    senha: string;
    imagem: string;

    constructor(email?: string, senha?: string, imagem?: string)
    {
        this.email = email;
        this.senha = senha;
        this.imagem = imagem;
    }
}
```

login.service.ts

```
import { Login } from '../model/login';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { AngularFireAuth } from 'angularfire2/auth';
import { Router } from '@angular/router';

@Injectable()
```

```

export class LoginService {

    usuario: Observable<firebase.User>;
    usuarioLogado: firebase.User;

    constructor(private autenticacao: AngularFireAuth, private
router: Router) {
        this.usuarioLogado == null;
        this.usuario = autenticacao.authState;
        this.usuario.subscribe(res => {
            console.log('login:' + res);
            if (res) {
                this.usuarioLogado = res;
            } else {
                this.usuarioLogado = null;
            }
        })
    }

    //METODO PARA LOGAR
    signInWithEmailAndPassword(login: Login){
        return
this.autenticacao.auth.signInWithEmailAndPassword(login.email,
login.senha);
    }

    //METODO DE LOGOUT
    logout(){
        this.autenticacao.auth.signOut().then(res=>{
            this.router.navigate(['login']);
        })
    }
}

```

Criar os seguintes componentes: Login, Produto e Erro.

Para criar o componente, digitar no terminal:

“ng g c login”

“ng g c produto”

login.component.ts

```
import { LoginService } from '../service/login.service';
import { Component, OnInit } from '@angular/core';
import { Login } from '../model/login';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

  login: Login;

  constructor(private router: Router, public service:
LoginService) {
    this.login = new Login();
  }

  ngOnInit() {

    //VERIFICAR SE ESTA REALMENTE LOGADO
    isLoginValido(login: Login):boolean{
      if(login){ //SE O LOGIN ESTA CHEIO
        if(!login.email || !login.senha){//CONFERE SE EMAIL
É DIFERENTE E SENHA DIFERENTE
          return false; //RETORNA FALSO
        }else{ //SE EMAIL E SENHA CONFEREM
          return true; //RETORNA TRUE, OU SEJA, LOGADO
        }
      }
    }
  }
}
```

```

    }

    logar(){
        if(this.isLoginValido(this.login)){
            this.service.signInWithEmailAndPassword(this.login).
then (res=>{
                console.log(res);
                this.router.navigateByUrl('produto');
            }, error =>{
                console.log('Erro: Digite email e senha...',
error);
            })
        }
    }
}

```

login.component.ts

```
<h2>Login</h2>
```

```

Email: <input type="email" #email [(ngModel)]='login.email'>
<br>
Senha: <input type="password" #senha [(ngModel)]='login.senha'>
<br>
<button (click)='logar();'>Logar</button>

```

produto.component.ts

```

import { LoginService } from '../service/login.service';
import { Component, OnInit } from '@angular/core';

@Component({
    selector: 'app-produto',
    templateUrl: './produto.component.html',
    styleUrls: ['./produto.component.css']
})

```

```
export class ProdutoComponent implements OnInit {

    constructor(private service: LoginService) {

    }

    ngOnInit() {

    }

    logout(){
        this.service.logout();
        console.log('Logout...');
    }
}
```

produto.component.html

```
<p>produto works!</p>
<br>
<button (click)='logout();'>Logout</button>
```

app.module.ts

```
import { LoginService } from './service/login.service';
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { HttpClientModule } from '@angular/common/http';
import { AngularFireAuthModule } from 'angularfire2/auth';
import { AppRoutingModuleModule } from './app-routing.module';
import { ProdutoComponent } from './produto/produto.component';
import { AngularFireModule } from '@angular/fire';
import { environment } from 'src/environments/environment';
```

```

import {FormsModule} from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    ProdutoComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule, //HABILITANDO A INTERNET
    AngularFireAuthModule, //AUTENTICAÇÃO DO FIREBASE
    AppRoutingModule,
    AngularFireModule.initializeApp(environment.firebaseConf
ig), //CONFIGURAÇÕES DE ACESSO
    FormsModule //FORMULARIO
  ],
  providers: [LoginService], //SERVIÇO
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app.routing.module.ts

```

import { LoginComponent } from '../login/login.component';
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { ProdutoComponent } from '../produto/produto.component';

const routes: Routes = [
  { path: '', redirectTo: 'login', pathMatch: 'full' },
  { path: 'login', component: LoginComponent },
  { path: 'produto', component: ProdutoComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})

```

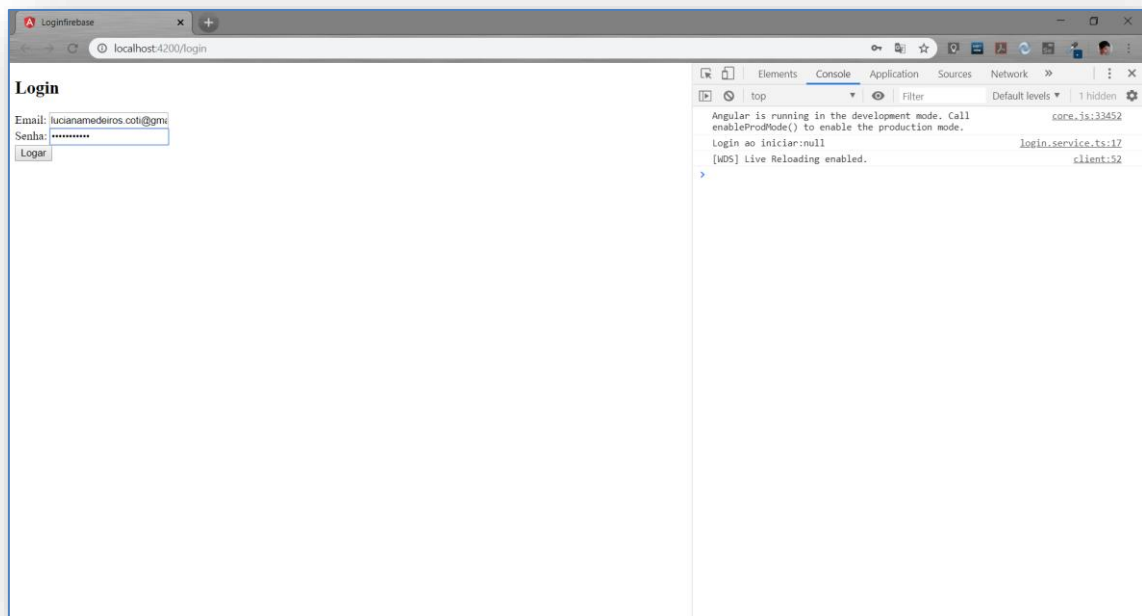


```
})  
export class AppRoutingModule { }
```

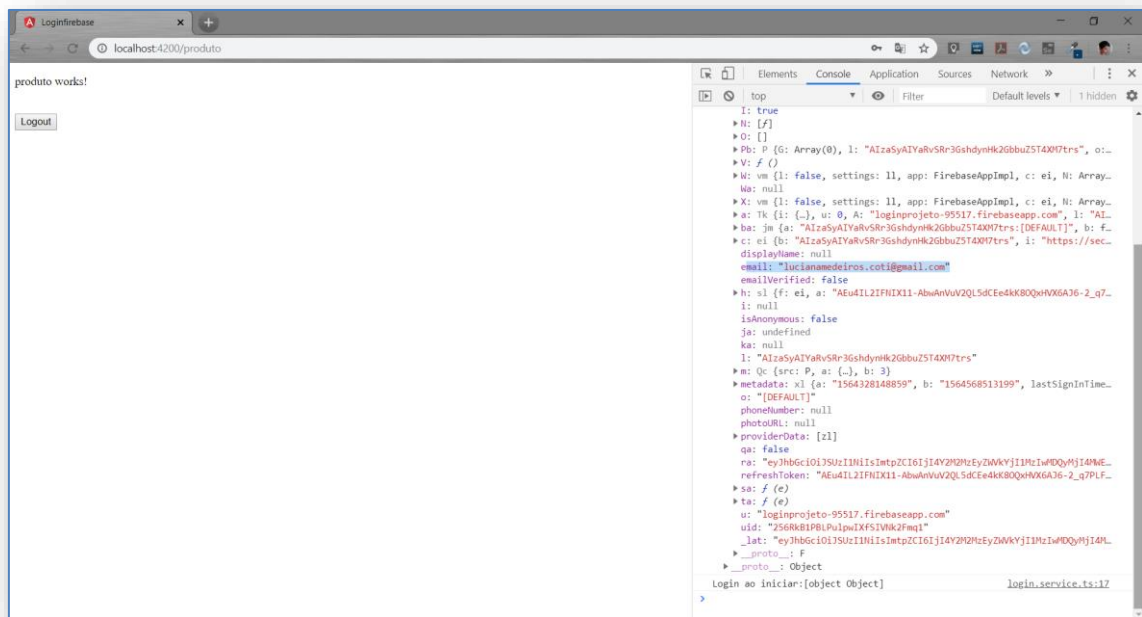
Rodando o projeto

<http://localhost:4200/login>

Digitando os dados para logar



Clicando em logar. Logado e direcionado para o componente de produtos



Clicando em logout. Volta para a tela de login (inicial)

