

# Desenvolvimento com FIREBASE



Alex Gonçalves de Araujo  
Coffee and Code  
Agroindústria  
Assis, SP

TODOS OS DIREITOS RESERVADOS

○ Abril  
**2018**



# Firestore

## Objetivos

Demonstrar um App desenvolvido com Firebase.

Apresentar recursos do Firebase para uso em Plataformas nativas (iOS e Android) e Web.

Detalhes do uso o para aplicativos da Web

- Authentication
- Database
- Storage
- Hosting

Firebase na prática







# O que é Firebase?

É uma plataforma móvel do Google que ajuda os desenvolvedores a criar rapidamente apps de alta qualidade.





Adicionar o  
Firebase ao seu  
aplicativo iOS



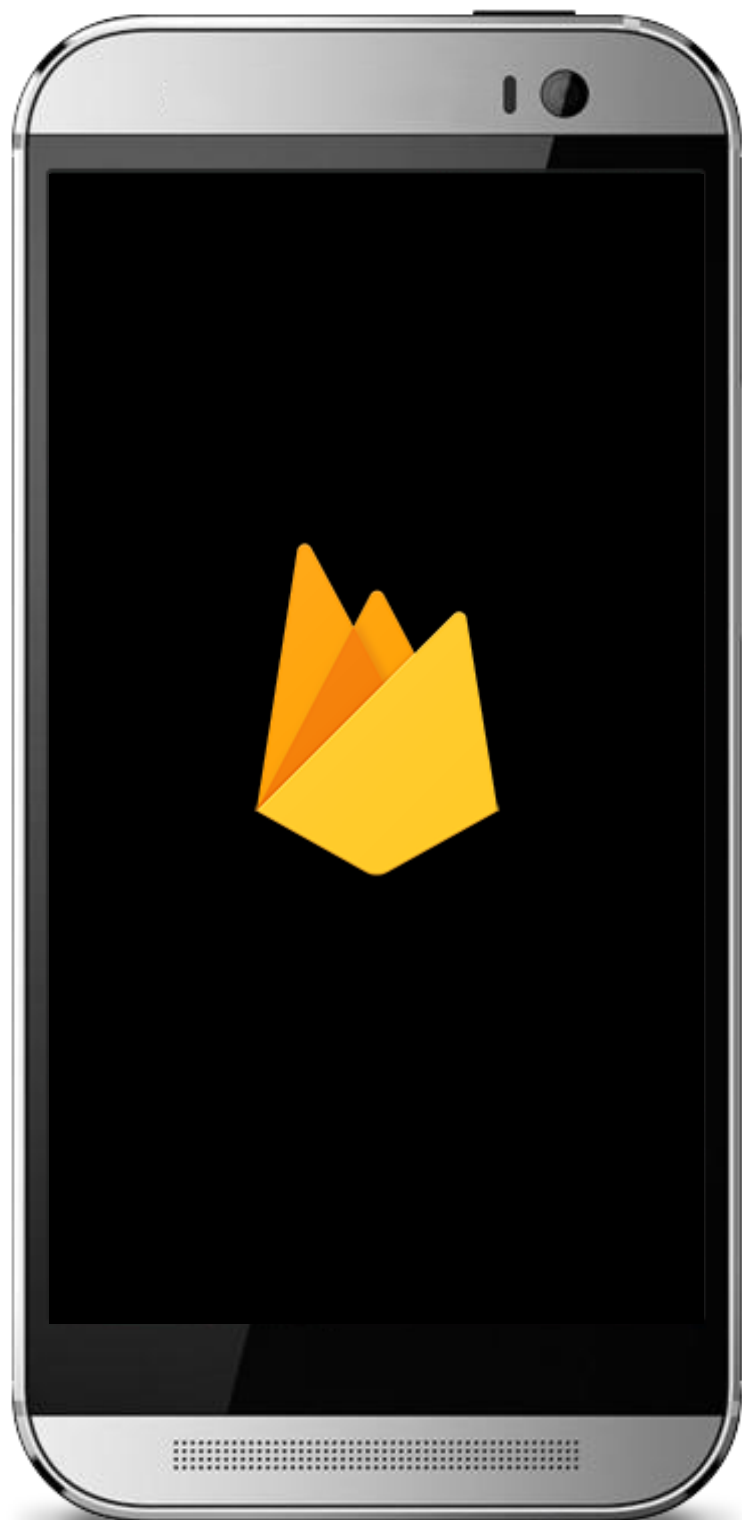
Adicionar o  
Firebase ao seu  
aplicativo Android



Adicionar o  
Firebase ao seu  
aplicativo da Web

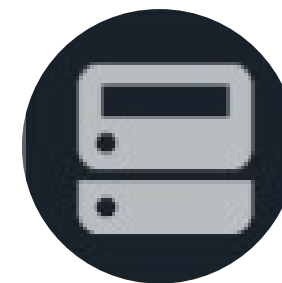


# Recursos para o APP



## Authentication

Para permitir que os usuários façam login no seu app e-mail e senha, ou o Login do Google e do Facebook.



## Database

é um banco de dados hospedado na nuvem. Os dados são armazenados como JSON e sincronizados em tempo real para cada cliente conectado.



## Storage

Criado para armazenar e vincular conteúdo gerado pelo usuário como, por exemplo, fotos ou vídeos.



## Hosting

Hospedagem estática rápida e segura para o app da Web





# É de volta ao Eden



Disponível na  
Play Store





# Conectados 2018 De Volta ao Éden

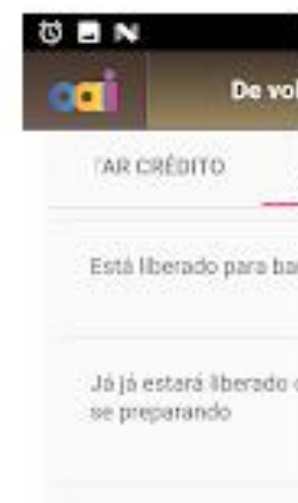
Alex G. de Araujo Eventos

★★★★★ 18



Este app é compatível com todos os seus dispositivos.

Instalado







# COMO COMEÇAR?





## Projetos recentes



[Adicionar projeto](#)



 Explorar projeto de demonstração

CoffeeAndCode

coffeetotvs

## connectStore

edenconnectstore

PontoLivre

pontolive-d9717



## Conectados em Cristo

conectados-em-cristo-b19d4

## PontoStore

pontostore-6f713



```
1  <script src="https://www.gstatic.com/firebasejs/4.12.1/firebase.js"></script>
2  <script>
3    // Initialize Firebase
4    var config = {
5      apiKey: "AIzaSyC99D09eQxGiZR6YOhzK__RJz87N6eNV50",
6      authDomain: "coffeetotvs.firebaseio.com",
7      databaseURL: "https://coffeetotvs.firebaseio.com",
8      projectId: "coffeetotvs",
9      storageBucket: "coffeetotvs.appspot.com",
10     messagingSenderId: "316785267683"
11   };
12   firebase.initializeApp(config);
13 </script>
```





# Authentication – Usuário / Senha

11



```
1  const auth = firebase.auth();
2  const promise = auth.signInWithEmailAndPassword(email, pass);
3
4  promise.then(function (user) {
5      var user = firebase.auth().currentUser;
6
7      //direcionar o usuário valido para o HOME
8      $state.go("home");
9
10 }).catch(function (err) {
11     //Notificar ao usuário que houve uma falha ou uma validação ()
12     NotificationService.alert("Login", err.message, "OK");
13 });
```



# Authentication – Google “pop-up”

12



```
1  //Crie uma instância do objeto provedor do Google:
2  var provider = new firebase.auth.GoogleAuthProvider();
3
4  //Solicitar o login em uma janela de pop-up
5  firebase.auth().signInWithPopup(provider).then(function(result) {
6      // Isso nos dá um Token de acesso Google. Podemos usar para acessar o Google API.
7      var token = result.credential.accessToken;
8      // Informações do usuário logado
9      var user = result.user;
10
11  }).catch(function(error) {
12      var errorCode = error.code; // Tratar Erros aqui.
13      var userEmail = error.email; // O e-mail do usuário na tentativa de Login. {error.email;}
14  });
15
```





# Authentication – Google “redirect”

13



```
16 //Redirecionar para página de Login
17 firebase.auth().signInWithRedirect(provider);
18
19 //Recupera o token quando sua página for carregada
20 firebase.auth().getRedirectResult().then(function(result) {
21     if (result.credential) {
22         // Isso nos um Token de acesso Google. Podemos usar para acessar o Google API.
23         var token = result.credential.accessToken;
24     }
25     // Informações do usuário logado
26     var user = result.user;
27 }).catch(function(error) {
28     var errorCode = error.code;// Tratar Erros aqui.
29     var userEmail = error.email;// O e-mail do usuário na tentativa de Login. {error.email;}
30 });
31
```



# Authentication – Google “Logout”

14



```
32  firebase.auth().signOut().then(function() {  
33      // Saiu com sucesso.  
34  }).catch(function(error) {  
35      // Um erro aconteceu  
36  });
```





# Storage - Referência

15



```
1  // Aponta para o Root do Google Cloud Storage
2  var storageRef = firebase.storage().ref();
3  // Aponta para 'imagens'
4  var imagesRef = storageRef.child('imagens');
5
6  // Aponta para 'imagens/space.jpg'
7  // Podemos usar variaveis para criar valores filhos
8  var fileName = 'space.jpg';
9  var spaceRef = imagesRef.child(fileName);
10
11 // Obterm o caminho que é 'imagens/space.jpg'
12 var path = spaceRef.fullPath
13 // O nome do arquivo que é 'space.jpg'
14 var name = spaceRef.name
```



# Storage - Upload

16



```
1  var fileButton = document.getElementById('fileButton');
2  fileButton.addEventListener('change', function (e) {
3      //Obter arquivo selecionado
4      var file = e.target.files[0];
5      //Criar uma referência do Storage Firebase
6      var storageRef = firebase.storage().ref("photos/perfil/" + file.name);
7      //upload arquivo
8      var task = storageRef.put(file);
9
10     //Progresso do Upload
11     task.on('state_changed',
12         function progress(snapshot) {
13             var percentage = (snapshot.bytesTransferred / snapshot.totalBytes) * 100
14         },
15         function error(err) { /*tratar erro*/ },
16         function complete() {
17             var path = task.snapshot.downloadURL;
18         }
19     );
20 });
```





# Storage - Download

17



```
1  //Cria uma referência do arquivo que você quer fazer o Download
2  var starsRef = storageRef.child('photos/perfil/alex.jpg');
3  // Obtem a Url completa da imagem
4  starsRef.getDownloadURL().then(function(url) {
5    // Insira a url dentro de uma tag <img>
6  }).catch(function(error) {
7    //Existe uma lista completa de Códigos de erro em
8    // https://firebase.google.com/docs/storage/web/handle-errors
9    switch (error.code) {
10     case 'storage/object_not_found':
11       break;// Arquivo não existe
12     case 'storage/unauthorized':
13       break;// Usuário não tem permissão para acessar o objeto
14     case 'storage/canceled':
15       break;// Usuário cancelou o upload
16     case 'storage/unknown':
17       break;// Ocorreu um erro desconhecido :)
18   }
19 });
```



# Realtime Database – Estrutura de dados

18



```
1  {
2    "users": [
3      {
4        "07613463": {
5          "name": "Alex G. De Araujo",
6          "credito": "17.50",
7          "dtNascimento": "12/01/1987",
8          "movimentacoes": {
9            "-L54ljS1pmyP62fdutSu": {
10              "data": 1518367864284,
11              "preco": 3,
12              "produto": "Sorvete",
13              "qtde": 2,
14              "valor": 6
15            },
16            "-L59FbUbivCW0q0zeSqp": { ...
22          }
23        }
24      }
25    ]
26  }
```



# Realtime Database – Gravar dados

19



```
1  const refBD = firebase.database().ref().child("users");
2
3  var userId = firebase.auth().currentUser.uid; //07613463
4  var updates = {};
5  var user = {
6      credito: 17.50,
7      dtNascimento: "12/01/1987",
8      name: "Alex G. de Araujo",
9      uuid: "123456"
10 };
11
12 updates['/' + userId] = user;
13 refBD.update(updates).then(function () {
14     console.log("Usuário registrado com sucesso;");
15 }, function (err) {
16     console.log("USER:" + err.mensagem);
17 });
```



# Realtime Database – Ler dados

20



```
1  var userId = firebase.auth().currentUser.uid; //uid = 07613463
2
3  return firebase.database().ref('/users/' + userId)
4      .once('value').then(function(res) {
5          var user = res.val(); //{json}
6          console.log(user.name + " - " + user.credito); //Alex G. De Araujo - 17.50
7      });
```





# FIREBASE NA PRÁTICA

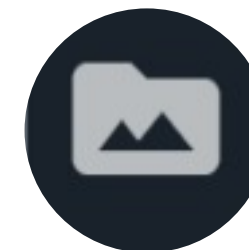
21



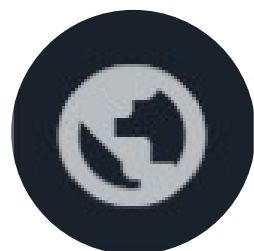
AUTHENTICATION



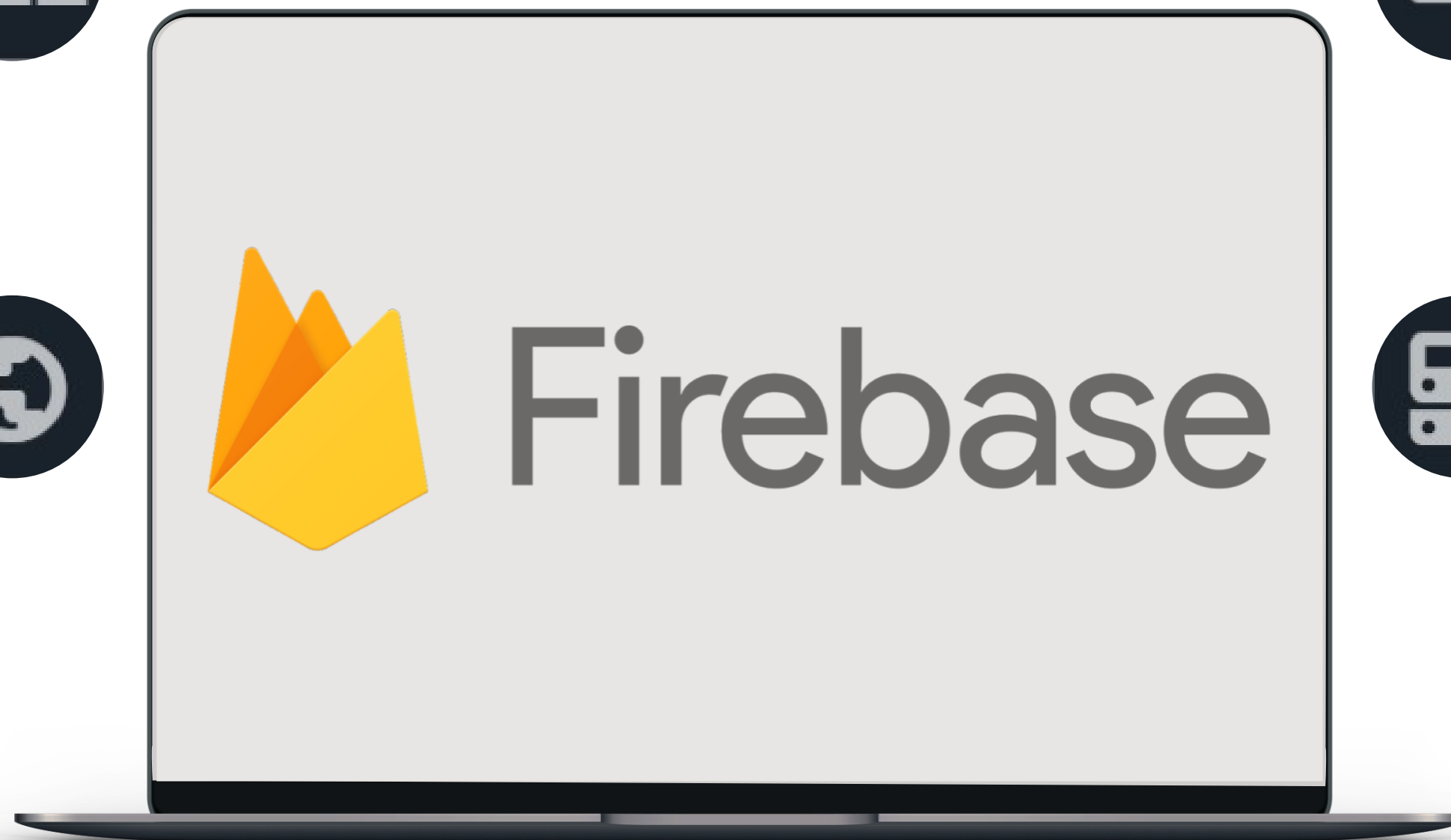
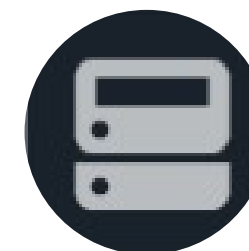
STORAGE



HOST



REALTIME  
DATABASE





# HTTP-SERVER

**1° \$ npm install http-server -g**

**2° \$ http-server -p8080 -o -c-1**





# Firestore Tools - Hosting



## 1º Instalação do Firestore Tools

**\$ npm install -g firestore-tools**

## 2º Efetuar o Login

**\$ firestore login**

## 3º Na raiz do projeto

**\$ firestore init**

## 4º Deploy do App

**\$ firestore deploy**



# Referências

<https://firebase.google.com>

<https://www.npmjs.com/package/http-server>







**Alex Gonçalves de Araujo**

Desenvolvimento Agroindústria

alex.araujo@totvs.com.br

 totvs.com

 @totvs

 company/totvs

 fluig.com

**Tecnologia + Conhecimento são nosso DNA**

**O sucesso do cliente é o nosso sucesso**

**Valorizamos gente boa que é boa gente**

**#SOMOSTOTVERS**