

Um evento com propósito

Embarcadero Conference 2020 Online

TODOS CONECTADOS



Um evento com propósito
Embarcadero Conference
2020 Online

Cadastro, Login e Recuperação de senha utilizando o Google Firebase

Adriano Santos

Agenda

- O que é o Google Firebase?
- O que é necessário para seu uso?
- Por onde começar?
- O que não esquecer de configurar?
- Codificando o Cadastro, Login e Recuperação de Senha.
- Q.A.



O que é o Google Firebase?

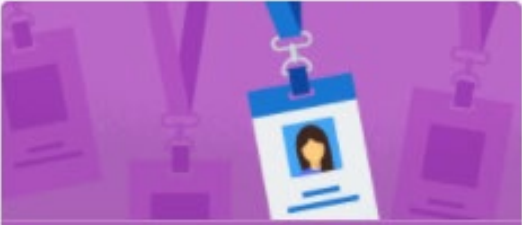
O que é o Google Firebase?

- O Google Firebase é o atual HUB de serviços do Google.
 - Realtime Database
 - Cloud Firestore
 - Cloud Messaging (Notificações Push)
 - Storage
 - Hosting
 - Crashlytics
 - **Serviços de Autenticação**


O que é o Google Firebase?

- Serviços

Armazene e sincronize dados de app em milissegundos




Authentication
Autenticar e gerenciar usuários




Cloud Firestore
Atualizações em tempo real, consultas eficientes e escalonamento automático

[Veja todos os Desenvolver recursos](#)

Expandir e envolva o público




Cloud Messaging
Interaja com os usuários certos no momento certo




A/B Testing
Melhorar os principais fluxos e notificações

Mantenha o controle da qualidade dos seus apps



Crashlytics
Priorizar e corrigir problemas de estabilidade



Performance
Receber insights sobre o desempenho do seu app

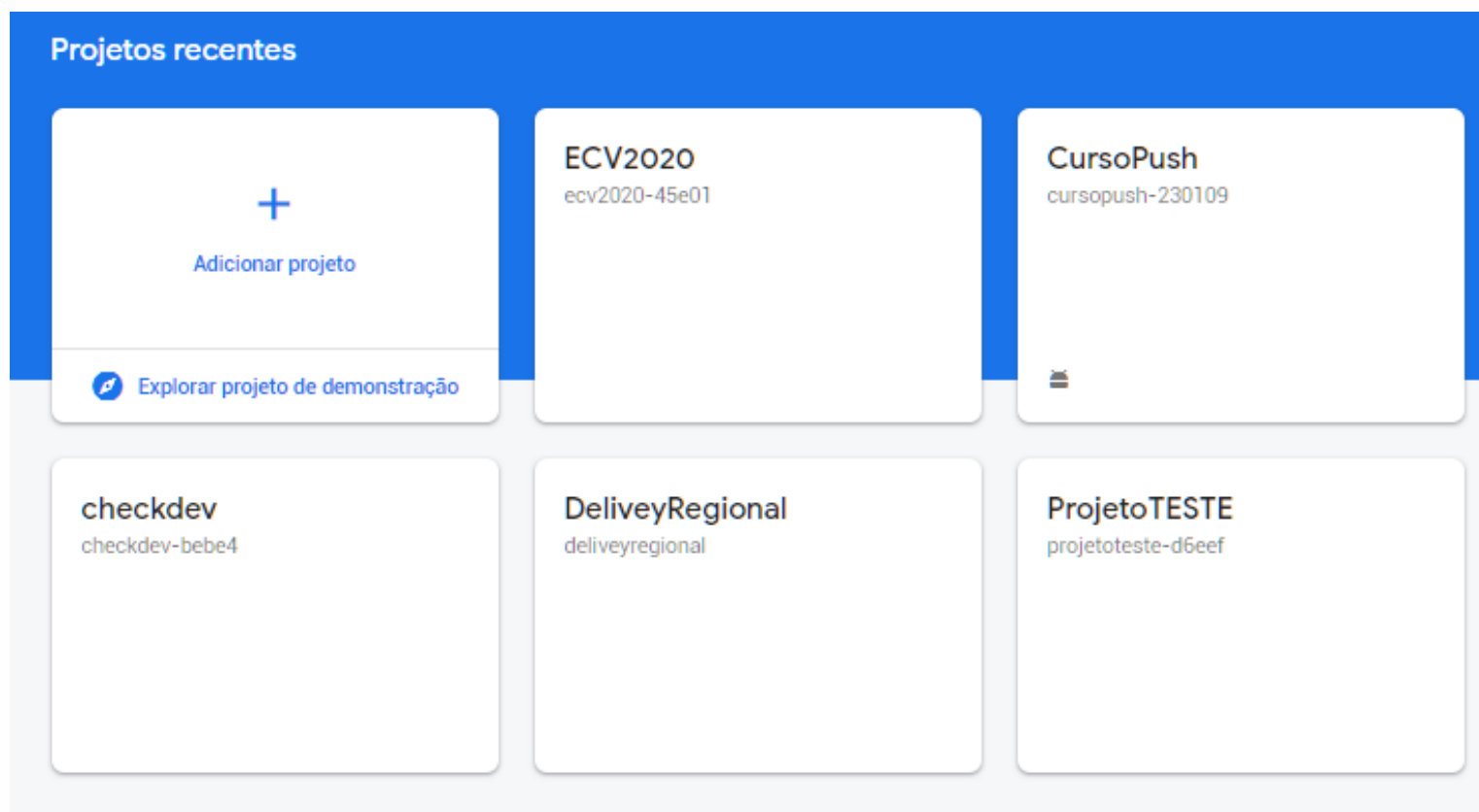
[Veja todos os Qualidade recursos](#)

O que é o Google Firebase?

- **Serviços de Autenticação**
- ***Email e Senha***
- Smartphone
- Google
- Facebook
- Twitter
- ...

O que é necessário para uso?







- Conta no Google
- Conta no Google Firebase (Mesma do Google)
- Criação de um projeto para cada app



Por onde começar?




- Captura da Chave da API Web
- Liberação do Email/Senha para uso


Seu projeto

Nome do projeto	ECV2020 
Código do projeto 	ecv2020-45e01
Número do projeto 	324080517125
Local padrão dos recursos do GCP 	Ainda não selecionado 
Chave de API da Web	AlzaSyA 

Configurações públicas

Essas configurações controlam instâncias do seu projeto que são mostradas ao público

Nome exibido ao público 	ECV2020 
E-mail para suporte 	<input type="text" value="tdevrocks@gmail.com"/>



Por onde começar?

- Liberação do Email/Senha para uso

The screenshot shows the Firebase Authentication console for project ECV2020. The left sidebar contains the 'Desenvolver' (Develop) menu with options: Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, and Machine Learning. The main area is titled 'Authentication' and has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. The 'Sign-in method' tab is active, displaying a table of login providers.

Provedor	Status
E-mail/senha	Ativado
Smartphone	Desativado
Google	Desativado

Codificando cadastro, login e recuperação

- Download do Firebase4Delphi
- **<https://github.com/spinettaro/Firebase4Delphi>**

Codificando cadastro, login e recuperação

- Adição do código API Key

```
FMX.Memo, FMX.Memo.Types;  
  
const  
    WEB_API_KEY = 'AlzaSyAyC...';  
  
type
```


Codificando cadastro, login e recuperação

- Login

```
procedure TForm1.recBtnLoginClick(Sender: TObject);  
begin  
    if not DoLogin then  
    begin  
        ShowMessage('Não possui conta. E-mail ou senha inválida!');  
        exit;  
    end;  
    MudarTab1.ExecuteTarget(Sender);  
end;
```

```
4 function TForm1.DoLogin: Boolean;  
5 var  
6     Auth      : IFirebaseAuth;  
7     AResponse : IFirebaseResponse;  
8     JSONResp  : TJSONValue;  
9     Obj       : TJSONObject;  
0 begin  
1     Result := False;  
2     Auth := TFirebaseAuth.Create;  
3     Auth.SetApiKey(WEB_API_KEY);  
4     AResponse := Auth.SignInWithEmailAndPassword(edt_email.Text, edt_senha.Text);  
5     JSONResp := TJSONObject.ParseJSONValue(AResponse.ContentAsString());  
6     try  
7         if (not Assigned(JSONResp)) or (not (JSONResp is TJSONObject)) then  
8             Exit(False);  
9  
0         Obj := JSONResp as TJSONObject;  
1         Result := Obj.TryGetValue('idToken', FToken);  
2     finally  
3         JSONResp.DisposeOf;  
4     end;  
5 end;
```

Codificando cadastro, login e recuperação

- Cadastro

```
procedure TForm1.Rectangle6Click(Sender: TObject);  
begin  
    if edtCadLogin.Text.IsEmpty then  
    begin  
        ShowMessage('Digite um endereço de email.');        edtCadLogin.SetFocus;  
        exit; ↵  
    end;  
  
    if edtCadPass.Text.IsEmpty then  
    begin  
        ShowMessage('Digite uma senha.');        edtCadPass.SetFocus;  
        exit; ↵  
    end;  
  
    CriarContaFirebase;  
    MudarTab1.ExecuteTarget(Sender);  
    ShowMessage('Conta criada com sucesso!');end;
```

Codificando cadastro, login e recuperação

- CriarContaFirebase

```
procedure TForm1.CriarContaFirebase;
var
  Auth      : IFirebaseAuth;
  AResponse : IFirebaseResponse;
  JSONResp  : TJSONValue;
  Obj       : TJSONObject;
  ObjError  : TJSONObject;
  LToken    : string;
  ErrorCode : string;
  ErrorMessage : string;
begin
  Auth := TFirebaseAuth.Create;
  Auth.SetApiKey(WEB_API_KEY);
  AResponse := Auth.CreateUserWithEmailAndPassword(editCadLogin.Text, editCadPass.Text);
  JSONResp := TJSONObject.ParseJSONValue(AResponse.ContentAsString);
  if (not Assigned(JSONResp)) or (not (JSONResp is TJSONObject)) then
  begin
    if Assigned(JSONResp) then
      JSONResp.Free;
    exit;
  end;
```

Codificando cadastro, login e recuperação

- Cadastro

```
Obj := JSONResp as TJsonObject;  
  
//if Obj.TryGetValue('idToken', LToken) then  
//  MemToken.Lines.Add(LToken);  
  
if Obj.TryGetValue('error', ObjError) then  
begin  
  ObjError.TryGetValue('code', ErrorCode);  
  ObjError.TryGetValue('message', ErrorMessage);  
  if ErrorMessage.Equals('EMAIL_EXISTS') then  
    raise Exception.Create('Esse email já existe');  
end;  
end;
```


Codificando cadastro, login e recuperação

- Recuperação de Senha

```
procedure TForm1.Rectangle10Click(Sender: TObject);
var
  Auth      : IFirebaseAuth;
  AResponse : IFirebaseResponse;
  JSONResp  : TJSONValue;
  Obj       : TJSONObject;
begin
  if edtRecuperarSenha.Text.Equals(EmptyStr) then
  begin
    ShowMessage('Informe o email');
    exit;
  end;

  Auth := TFirebaseAuth.Create;
  Auth.SetApiKey(WEB_API_KEY);
  AResponse := Auth.SendResetPassword(edtRecuperarSenha.Text);
  JSONResp := TJSONObject.ParseJSONValue(AResponse.ContentAsString());
  (* Se desejar fazer algum tipo de teste
  if (not Assigned(JSONResp)) or (not (JSONResp is TJSONObject)) then
  begin
    if Assigned(JSONResp) then
      JSONResp.Free;
    exit;
  end;
  *)
  ShowMessage('Cadastro efetuado com sucesso. ');
  edtRecuperarSenha.Text := EmptyStr;
  MudarTab1.Tab := TabLogin;
  MudarTab1.ExecuteTarget(Sender)
end;
```

Codificando cadastro, login e recuperação

Demo

Considerações Finais

Além do Login, outros serviços podem ser usados, dando mais profissionalismo ao seu app.

Introdução ao Google Firebase

CUPOM 50%



Avaliação da Palestra

Avaliação

<https://forms.gle/mueNWCySr86RMv7R7>

Código Fonte

<https://github.com/adrianosantostreina/EC2020-FirebaseLogin>





Adriano Santos

adriano@adrianosantostreina.com.br

Empresário, Especialista em Desenvolvimento de Software com +23 anos no Delphi, Embarcadero MVP, pioneiro em tecnologia mobile e servidores de aplicação. Considerado uma das referências do Delphi no Brasil, + 1000 artigos publicados.

OBRIGADO