



CENTRO UNIVERSITÁRIO FILADÉLFIA

GIOVANNI ALEIXO DE SOUZA
RONNI APARECIDO FERNANDES DE OLIVEIRA

ATIVIDADE INTEGRADORA [AVA3]:
DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

GIOVANNI ALEIXO DE SOUZA
RONNI APARECIDO FERNANDES DE OLIVEIRA

ATIVIDADE INTEGRADORA [AVA3]:
DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

Atividade Integradora da matéria de Desenvolvimento de Aplicativos Móveis apresentado ao Curso de Tecnólogo Análise e Desenvolvimento de Sistemas do Centro Universitário Filadélfia – UniFil

Orientador: Prof. Marcus Vinícius Santana Maziero

Londrina
2023

A atividade desenvolvida na disciplina de Extensão será sobre a Startup “VIAJANTES – Grupo 17”

O projeto baseia-se em desenvolver um aplicativo com o objetivo de facilitar o planejamento de viagens de carro. Espera-se com o aplicativo pode calcular o custo estimado da viagem, a distância e o tempo a ser incorrido no trajeto.

O público-alvo serão quaisquer pessoas com interesse neste tipo de aplicativo, que será disponibilizado gratuitamente, ao final do projeto, na loja de aplicativos do Google.

Neste sentido, inicialmente, nosso objetivo é construir uma aplicação Mobile, que pode ser acessado na plataforma Android. Entretanto, não está descartado a ideia de implementação da aplicação para acesso através de navegadores (web).

Considerando as boas práticas na ciência da computação, o aplicativo será desenvolvido utilizando-se das técnicas de orientação à objeto.

Inicialmente, a linguagem principal seria JAVA, que é uma linguagem sólida, amplamente conhecida, portátil, e em constante evolução, entretanto, considerando a disciplina de Desenvolvimento de Aplicativos Móveis, optou-se por realizar o desenvolvimento utilizando-se das linguagens/frameworks JAVASCRIPT, HTML, CSS, TYPESCRIPT, ANGULAR e IONIC.

A estrutura do banco de dados no App é bem simples, pois, como seu uso será pelo celular, um banco de dados robusto com informações de destinos e mapas não comportaria ser instalado em dispositivos off-line.

Espera-se obter as informações mais complexas relacionadas ao destino buscado, através do acesso à uma API Web pública (Google Maps, por exemplo).

O banco de dados escolhido para a implementação do aplicativo ‘Viajantes’ foi o Firebase Realtime Database.

A escolha desse banco de dados ocorreu devido a possibilidade de armazenar e sincronizar os dados de seus usuários em tempo real, sendo possível ao usuário acessar os dados de seus cálculos de viagem através de qualquer dispositivo.

O Firebase Realtime Database é também um banco de dados com alto nível de segurança e pode ser hospedado em nuvem.

Para esta atividade foi implementado as telas responsáveis pelo Cadastramento (register), Acesso (login) e Recuperação de Senha (reset).

A seguir demonstramos as telas do aplicativo:

Banco de dados FIREBASE:

Firebase

Visão geral do projeto

Atalhos do projeto

Authentication

Realtime Database

Categories dos produtos

Criação

Liberar e monitorar

Analytics

Engajamento

Todos os produtos

Personalize sua navegação.

db-visitantes-unifil

Authentication

Users

Sign-in method

Templates

Usage

Settings

Extensões

Adicionar usuário

Identificador	Provedores	Data de criação	Último login	UID do usuário
giovanniLaleixo@hotmail.c...		27 de mar. d...	27 de mar. d...	sndR8CucuUPVzMXru3CX7bEySY11
ronni.fernandes@uol.com.br		26 de mar. d...		nWeDMKBgPgbzSpr5pNkKazww...

Authentication

Users

Sign-in method

Templates

Usage

Settings

Extensões

Provedores de login

Adicionar novo provedor

Provedor	Status
E-mail/senha	ativado

Firebase

db-visitantes-unifil

Configurações do projeto

Adicionar app

Apps da Web

VIAJANTES

Web App

Apelido do app

VIAJANTES

ID do aplicativo

1:300055557882:web:7b9bc75306e61e7fa0fb43

Vincular a um site do Firebase Hosting

Configuração do SDK

NPM

CDN

Config

Acesse o snippet do seu objeto de configuração do Firebase do app. Saiba mais

Objeto de configuração do Firebase que contém as chaves e identificadores do seu app:

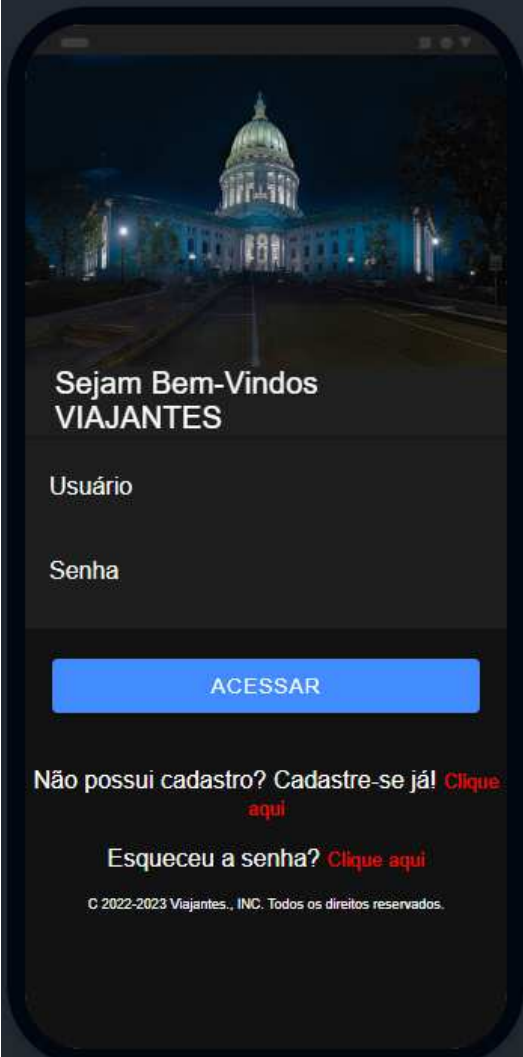
```
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyA1iUATH4KL6Oyds8Z-n9aF2hwzZa0_p_I",
  authDomain: "db-visitantes-unifil.firebaseio.com",
  projectId: "db-visitantes-unifil",
  storageBucket: "db-visitantes-unifil.appspot.com",
  messagingSenderId: "300055557882",
  appId: "1:300055557882:web:7b9bc75306e61e7fa0fb43",
  measurementId: "G-32XEM4RSHE"
};
```

Você está usando o npm e um handler como Webpack ou Rollup? Consulte o SDK modular

Conexão FIREBASE no APP Viajantes <enviroments.ts> :

```
export const environment = {  
  production: false,  
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional  
  firebaseConfig : {  
    apiKey: "AIzaSyAi1UATH4KL60yds0Z-n9aF2hwzZa0_p_I",  
    authDomain: "db-visitantes-unifil.firebaseio.com",  
    projectId: "db-visitantes-unifil",  
    storageBucket: "db-visitantes-unifil.appspot.com",  
    messagingSenderId: "300055557882",  
    appId: "1:300055557882:web:7b9bc75306e61e7fa0fb43",  
    measurementId: "G-32XEM4RSHE"  
  }  
}
```

LOGIN:



The image shows a mobile application login screen. At the top, there is a background image of a large, illuminated building at night. Below the image, the text "Sejam Bem-Vindos VIAJANTES" is displayed. Underneath, there are two input fields labeled "Usuário" and "Senha". A blue button labeled "ACESSAR" is positioned below the input fields. At the bottom, there are two links: "Não possui cadastro? Cadastre-se já! Clique aqui" and "Esqueceu a senha? Clique aqui". The footer contains the text "C 2022-2023 Viajantes, INC. Todos os direitos reservados."

Módulo HTML: <login.page.html>

```
<ion-header>
  <ion-toolbar>
    <ion-img src="https://ionic-docs-demo-v6.vercel.app/assets/madison.jpg"
alt="The Wisconsin State Capitol building in Madison, WI at night"></ion-img>
    <ion-title>Sejam Bem-Vindos</ion-title>
    <ion-title>VIAJANTES</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <form name="login">
    <ion-list>
      <ion-item>
        <ion-label position="floating">Usuário</ion-label>
        <ion-input id="email" type="text" placeholder="Digite seu e-mail"
#usuarioDigitado ></ion-input>
      </ion-item>
      <ion-item>
        <ion-label position="floating">Senha</ion-label>
        <ion-input id="password" type="password" placeholder="Digite sua
senha" #senhaDigitada ></ion-input>
      </ion-item>
    </ion-list>
  </form>
  <div class="ion-padding">
    <ion-button expand="block" (click)="log()">Acessar</ion-button>

  </div>
  <div div style="text-align: center">
    <p>Não possui cadastro? Cadastre-se já! <span routerLink="register"
(click)="registerUser()" style="color:red; font-size: small;">Clique
aqui</span></p>
    <p></p>
    <p>Esqueceu a senha? <span routerLink="reset" (click)="resetUser()"
style="color: red; font-size: small;">Clique aqui</span></p>
  </div>
  <div>
    <p style="font-size: xx-small; text-align: center"> C 2022-2023
Viajantes., INC. Todos os direitos reservados.</p>
  </div>
</ion-content>
```

Módulo TS: <login.page.ts>

```
import { Component, OnInit } from '@angular/core';
import { AngularFireAuth } from '@angular/fire/compat/auth';
import { Router } from '@angular/router';

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

  email: any;
  pass: any;

  constructor(private router: Router, private auth: AngularFireAuth) { }

  ngOnInit() {
  }

  log(){
    this.email = ((document.getElementById("email") as
HTMLInputElement).value);
    this.pass = ((document.getElementById("password") as
HTMLInputElement).value);

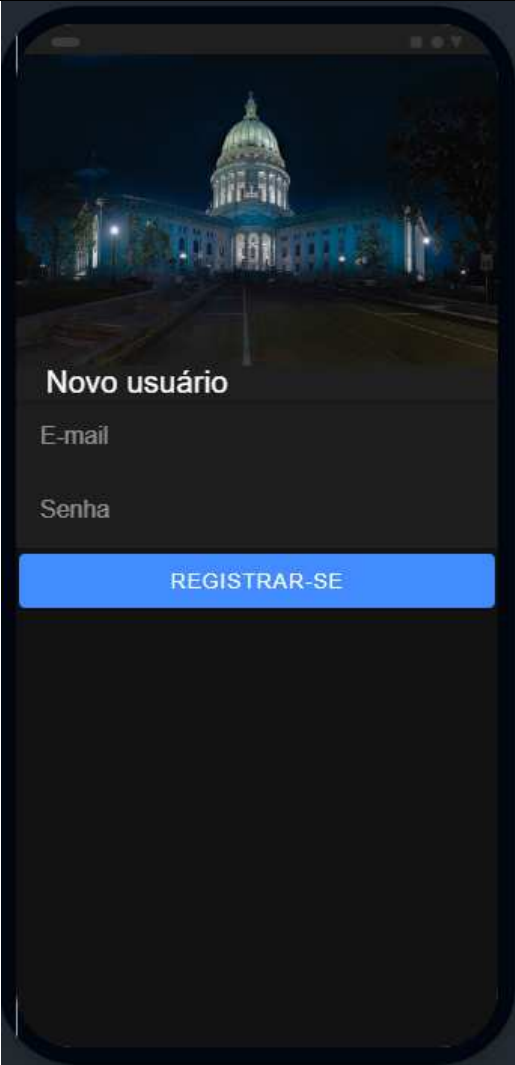
    this.auth.
signInWithEmailAndPassword(this.email, this.pass)
    .then((userCredential) => {
      if (userCredential.user){
        window.alert("Bem vindo "+this.email);
        this.router.navigateByUrl('/home');
      }
    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;
      window.alert(errorMessage);
    });
  }

  registerUser(){
    this.router.navigateByUrl('/register');
  };

  resetUser(){
```

```
    this.router.navigateByUrl('/reset');  
  
  };  
  
}
```

REGISTER:

A mobile application registration screen. At the top is a header image of a large, illuminated building at night. Below the image, the text "Novo usuário" is displayed. Underneath are two input fields labeled "E-mail" and "Senha". A blue button with the text "REGISTRAR-SE" is positioned below the input fields. The entire screen is framed by a dark border.

Novo usuário

E-mail

Senha

REGISTRAR-SE

Módulo HTML: <register.page.html>

```
<ion-header>
  <ion-toolbar>
    <ion-img src="https://ionic-docs-demo-v6.vercel.app/assets/madison.jpg"
alt="The Wisconsin State Capitol building in Madison, WI at night"></ion-img>
    <ion-title>Novo usuário</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div>

    <ion-item>
      <ion-input id="new_email" placeholder="E-mail" expand="block"> </ion-
input>
    </ion-item>

    <ion-item>
      <ion-input id="new_password" placeholder="Senha" expand="block"> </ion-
input>
    </ion-item>

    <ion-button expand="block" (click)="reg()">Registrar-se</ion-button>

  </div>
</ion-content>
```

Módulo TS: <register.page.ts>

```
import { Component, OnInit } from '@angular/core';
import { getAuth, createUserWithEmailAndPassword } from "firebase/auth";
import { Router } from '@angular/router';
import { AngularFireAuth } from '@angular/fire/compat/auth';

@Component({
  selector: 'app-register',
  templateUrl: './register.page.html',
  styleUrls: ['./register.page.scss'],
})
export class RegisterPage implements OnInit {

  email:any;
  pass:any;

  constructor(private router:Router, private auth:AngularFireAuth) { }

  ngOnInit() {
```

```
}

reg(){

  this.email = ((document.getElementById("new_email") as
HTMLInputElement).value);
  this.pass = ((document.getElementById("new_password") as
HTMLInputElement).value);

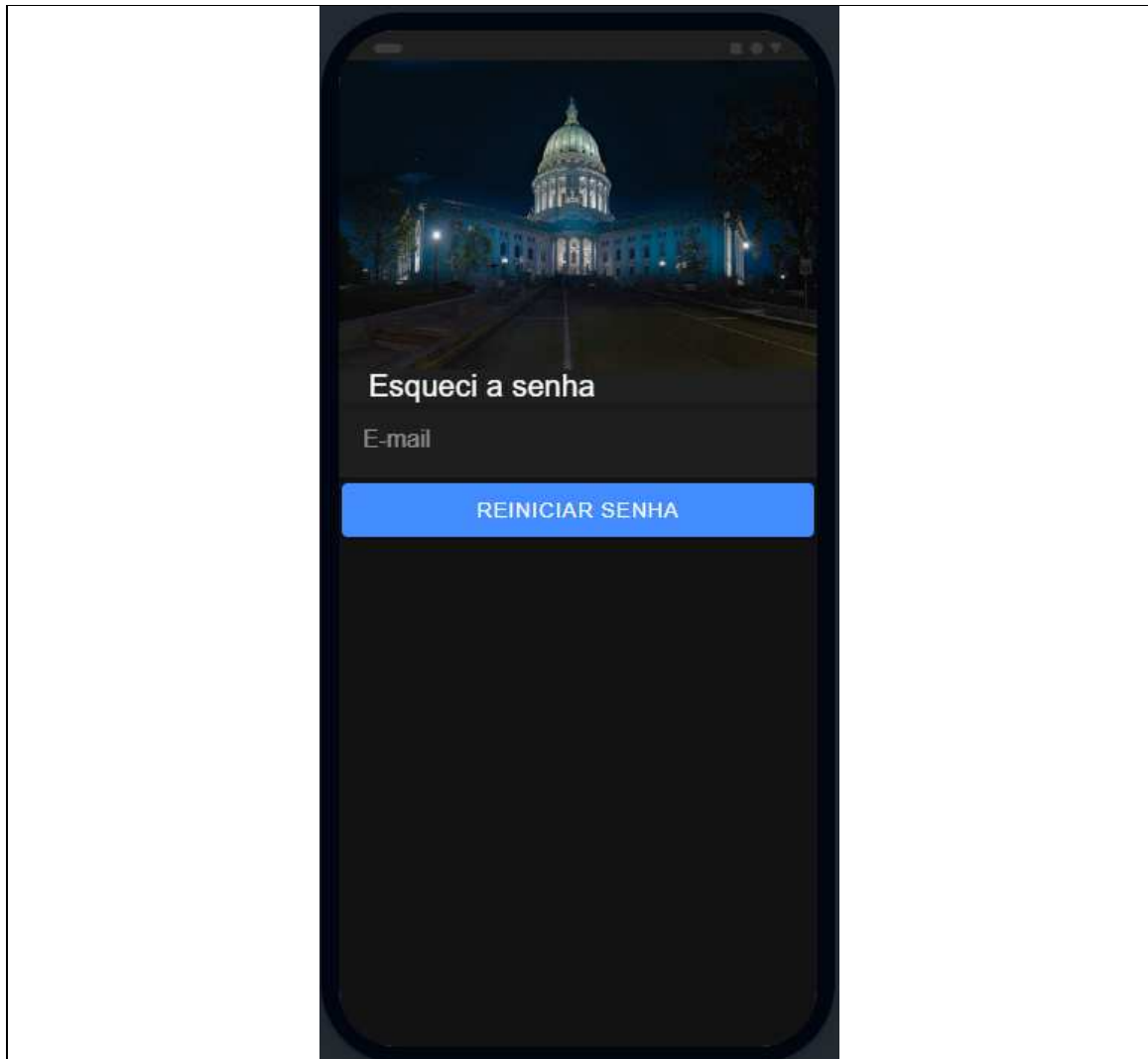
  this.auth.
createUserWithEmailAndPassword(this.email, this.pass)
  .then((userCredential) => {
    if (userCredential.user){
      window.alert("Usuário adicionado. Por favor, efetue o login" );
      this.router.navigateByUrl('/login');

    }
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    window.alert(errorMessage);
  });

}

}
```

RESET:



Módulo HTML: <reset.page.html>

```
<ion-header>
  <ion-toolbar>
    <ion-img src="https://ionic-docs-demo-v6.vercel.app/assets/madison.jpg"
alt="The Wisconsin State Capitol building in Madison, WI at night"></ion-img>
    <ion-title>Esqueci a senha</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-content>
    <div>
      <ion-item>
        <ion-input id="email" placeholder="E-mail" expand="block"></ion-
input>
```

```

        </ion-item>
        <ion-button expand="block" (click)="reset()">Reiniciar senha</ion-
button>
    </div>
</ion-content>
</ion-content>

```

Módulo TS: <reset.page.ts>

```

import { Component, OnInit } from '@angular/core';
import { AngularFireAuth } from '@angular/fire/compat/auth';

@Component({
  selector: 'app-reset',
  templateUrl: './reset.page.html',
  styleUrls: ['./reset.page.scss'],
})
export class ResetPage implements OnInit {

  email:any;

  constructor(private auth:AngularFireAuth) { }

  ngOnInit() {
  }

  reset(){
    this.email=((document.getElementById("email") as
HTMLInputElement).value);

    this.auth.
    sendPasswordResetEmail(this.email)
    .then(userCredential => {
      window.alert("O reset da senha foi enviado para o e-mail " +
this.email);
      //this.router.navigateByUrl('/login');
    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;
      window.alert(errorMessage);
    });
  };
}

```

Referências

Material do curso EAD – Banco de Dados Avançados UNIFIL EAD. Acesso em 20 fev. 2023

Material do curso EAD – Desenvolvimento de Aplicativos Móveis UNIFIL EAD. Acesso em 20 fev. 2023

Material do curso EAD – Linguagem de Programação Orientada a Objetos UNIFIL EAD. Acesso em 22 fev. 2023

Material do curso EAD – Introdução à Banco de Dados UNIFIL EAD. Acesso em 15 março. 2023

Material do curso EAD – Algoritmos e Estrutura de Dados UNIFIL EAD. Acesso em 15 março. 2023

Material do curso EAD – Análise e Projeto Orientado a Objetos UNIFIL EAD. Acesso em 15 março. 2023

Material do curso EAD – Desenvolvimento Web UNIFIL EAD. Acesso em 15 março. 2023

Material do curso EAD – Processo de Desenvolvimento de Software UNIFIL EAD. Acesso em 15 março. 2023

Material do curso EAD – Lógica de Programação Orientada a Objetos UNIFIL EAD. Acesso em 15 março. 2023

Firestore email and password authentication with ionic (angular) Disponível em: <https://www.youtube.com/watch?v=6K6H3wMMY2U> Acesso em: 26/03/2023

Firestore Realtime Database Disponível em: <https://firebase.google.com/docs/database?hl=pt-br>

Realtime Database, Disponível em: <https://rnfirebase.io/database/usage>

Entendendo o Firestore e suas principais funcionalidades, Disponível em: https://www.alura.com.br/artigos/entendendo-firebase-principaisfuncionalidades?gclid=Cj0KCQjw2v-gBhC1ARIsAOQdKY3yCLWsZ4oqykIIKtJwZl4DyHiKT7bKwTIW81zDXnS45rvmx_sH78aAo_HEALw_wcB