

```

app.components.ts
import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import { HomePage } from '../pages/home/home';
import { SigninPage } from '../pages/signin/signin';

import { AngularFireAuth } from 'angularfire2/auth';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage: any;

  constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen,
    afAuth: AngularFireAuth) {
    const authObserver = afAuth.authState.subscribe(user => {
      if (user) {
        this.rootPage = HomePage;
        authObserver.unsubscribe();
      } else {
        this.rootPage = SigninPage;
        authObserver.unsubscribe();
      }
    });

    platform.ready().then(() => {

      statusBar.styleDefault();
      splashScreen.hide();
    });
  }
}

```

```

    });
  }
}

```

app.module.ts

```

import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { SplashScreen } from '@ionic-native/splash-screen';
import { StatusBar } from '@ionic-native/status-bar';

import { AngularFireModule } from 'angularfire2';
import { AngularFireDatabase, AngularFireDatabaseModule } from 'angularfire2/database';
import { AngularFireAuthModule } from 'angularfire2/auth';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { SigninPage } from '../pages/signin/signin';
import { SignupPage } from '../pages/signup/signup';
import { ResetpasswordPage } from '../pages/resetpassword/resetpassword';

import { EvControlPage } from '../pages/ev-control/ev-control';
import { EvAddPage } from '../pages/ev-add/ev-add';
import { EvAddPageModule } from '../pages/ev-add/ev-add.module';

import { CControlPage } from '../pages/c-control/c-control';
import { CAddPage } from '../pages/c-add/c-add';
import { CAddPageModule } from '../pages/c-add/c-add.module';

import { LControlPage } from '../pages/l-control/l-control';
import { LAddPage } from '../pages/l-add/l-add';

```

```

import { LAddPageModule } from '../pages/l-add/l-add.module';

import { AuthService } from '../providers/auth/auth-service';

import { EventosProvider } from "../providers/eventos/eventos";
import { CursosProvider } from '../providers/cursos/cursos';
import { LocaisProvider } from '../providers/locais/locais';

const firebaseConfig = {
  apiKey: "AlzaSyDhqjPVnyH4dHbP8YDKC4ZOM4L9_juCNk",
  authDomain: "uniinfo-database-full.firebaseio.com",
  databaseURL: "https://uniinfo-database-full.firebaseio.com",
  projectId: "uniinfo-database-full",
  storageBucket: "uniinfo-database-full.appspot.com",
  messagingSenderId: "613710825075"
};

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    SigninPage,
    SignupPage,
    ResetpasswordPage,

    EvControlPage,
    CControlPage,
    LControlPage

  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp, {
      platforms: {

```

```

        ios: {
            backButtonText: 'Voltar'
        }
    },
   )),

    EvAddPageModule,
    CAddPageModule,
    LAddPageModule,

    AngularFireModule.initializeApp(firebaseConfig),
    AngularFireAuthModule,
    AngularFireDatabaseModule,

    ],
    bootstrap: [IonicApp],
    entryComponents: [
        MyApp,
        HomePage,
        SigninPage,
        SignupPage,
        ResetpasswordPage,

        CControlPage,
        EvControlPage,
        LControlPage,

        EvAddPage,
        CAddPage,
        LAddPage
    ],
    providers: [
        StatusBar,
        SplashScreen,

```

```

    { provide: ErrorHandler, useClass: IonicErrorHandler },
    AuthService,
    EventosProvider,
    CursosProvider,
    LocaisProvider
  ]
})
export class AppModule { }

```

c-add.html

```

<ion-header>
  <ion-navbar>
    <div>
      <br>
      <br>
      <h1>
        <b>&nbsp; &nbsp; {{titulo}}</b>
      </h1>
      <font color="#3db7ff" size="2">
        <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; Evento</p>
      </font>
    </div>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <div class="div-mod-3">
    <form [formGroup]="form">

      <ion-item>
        <ion-label>Turno</ion-label>

```

```

    <ion-select formControlName="turno">
      <ion-option value="Noturno">Noturno</ion-option>
      <ion-option value="Matiturno">Matiturno</ion-option>
    </ion-select>
  </ion-item>

  <ion-item>
    <ion-label floating>Curso</ion-label>
    <ion-input type="text" formControlName="curso"></ion-input>
  </ion-item>
  <ion-item *ngIf="!form.controls.curso.valid && (form.controls.curso.dirt ||
form.controls.curso.touched)" color="danger">
    <div [hidden]="!form.controls.curso.errors.required">
      O campo é obrigatório.
    </div>
  </ion-item>

  <ion-item>
    <ion-label floating>Semestre</ion-label>
    <ion-input type="number" formControlName="semestre"></ion-input>
  </ion-item>
  <ion-item *ngIf="!form.controls.semestre.valid && (form.controls.semestre.dirt ||
form.controls.semestre.touched)" color="danger">
    <div [hidden]="!form.controls.semestre.errors.required">
      O campo é obrigatório.
    </div>
  </ion-item>

  <ion-item>
    <ion-label floating>Segunda</ion-label>
    <ion-input type="text" formControlName="sseg"></ion-input>
  </ion-item>

  <!--

```

```

<ion-item>
  <ion-label floating>Terça</ion-label>
  <ion-input type="text" formControlName="ster"></ion-input>
</ion-item>

```

```

<ion-item>
  <ion-label floating>Quarta</ion-label>
  <ion-input type="text" formControlName="squa"></ion-input>
</ion-item>

```

```

<ion-item>
  <ion-label floating>Quinta</ion-label>
  <ion-input type="text" formControlName="squi"></ion-input>
</ion-item>

```

```

<ion-item>
  <ion-label floating>Sexta</ion-label>
  <ion-input type="text" formControlName="ssex"></ion-input>
</ion-item>

```

```
-->
```

```

  <div padding>
    <button      ion-button      block      type="submit"      [disabled]="!form.valid"
(click)="onSubmit()">Salvar</button>
  </div>
</form>
</div>

```

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

```
c-add.ts
```

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

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
```

```
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-c-add',
```

```
  templateUrl: 'c-add.html',
```

```
})
```

```
export class CAddPage {
```

```
  titulo: string;
```

```
  form: FormGroup;
```

```
  cursos: any;
```

```
  constructor(public navCtrl: NavController,
```

```
    public navParams: NavParams,
```

```
    private formBuilder: FormBuilder,
```

```
    private provider: CursosProvider,
```

```
    private toast: ToastController) {
```

```
    this.cursos = this.navParams.data.cursos || {};
```

```
    this.createForm();
```

```
    this.setupPageTitle();
```

```
}
```

```
private setupPageTitle() {
```

```
  this.titulo = this.navParams.data.cursos ? 'Alterando sala' : 'Nova sala';
```

```
}
```

```
createForm() {
```

```
  this.form = this.formBuilder.group({
```

```
    key: [this.cursos.key],
```



```

    turno: [this.cursos.turno, Validators.required],
    curso: [this.cursos.curso, Validators.required],
    semestre: [this.cursos.semestre, Validators.required],
    sseg: [this.cursos.sseg],

    /*
    ster: [this.cursos.ster],
    squa: [this.cursos.squa],
    squi: [this.cursos.squi],
    ssex: [this.cursos.ssex]
    */
  })
}

onSubmit(){
  if (this.form.valid) {
    this.provider.save(this.form.value)
    .then(() => {
      this.toast.create({ message: 'Sala adicionada com sucesso.', duration:
3000}).present();
      //this.navCtrl.pop();
    })
    .catch((e) => {
      this.toast.create({ message: 'Erro ao salvar sala.', duration: 3000}).present();
      console.error(e);
    });
  }
}

}

```

```

<ion-header>
  <ion-navbar>
    <div>
      <br>
      <br>
      <h1>
        <b>&nbsp; &nbsp; &nbsp; Registros</b>
      </h1>
      <font color="#3db7ff" size="2">
        <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; Cursos</p>
      </font>
      <ion-title>
      </ion-title>
    </div>
  </ion-navbar>

</ion-header>

<ion-content padding>

  <div class="div-mod-control">
    <br>
    <br>
    <ion-list>
      <ion-list-header align="center">
        <h1>Cursos matiturnos</h1>
      </ion-list-header>
      <div *ngFor="let curso of courses | async">
        <ion-item *ngIf="curso.turno === 'Matiturno'">

          <font color="#b2b2b2" size="2" align="center">
            {{curso.curso}} | {{curso.semestre}}° Sem | Sala: {{curso.sseg}}
          </font>
        </ion-item>
      </div>
    </ion-list>
  </div>

```

```

        <button                ion-button                clear                item-start
(click)="editCurso(curso)">EDITAR</button>
        <button                ion-button                clear                color="danger"                item-end
(click)="removeCurso(curso.key)">EXCLUIR</button>
    </ion-item>
</div>
</ion-list>
<br>
<br>
<ion-list>
    <ion-list-header align="center">
        <h1>Cursos noturnos</h1>
    </ion-list-header>
    <div *ngFor="let curso of courses | async">
        <ion-item *ngIf="curso.turno === 'Noturno'">

            <font color="#b2b2b2" size="2" align="center">
                {{curso.curso}} | {{curso.semestre}}º Sem | Sala: {{curso.sseg}}
            </font>

        <button                ion-button                clear                item-start
(click)="editCurso(curso)">EDITAR</button>
        <button                ion-button                clear                color="danger"                item-end
(click)="removeCurso(curso.key)">EXCLUIR</button>
    </ion-item>
    </div>
</ion-list>
<br>
<br>
<font style="font-size:9pt; color:gray">
    <p align="right">Powered by UniInfo.</p>
</font>
</div>

```

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

```
<ion-footer>
```

```
<font>
```

```
<p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
```

```
</font>
```

```
</ion-footer>
```

```
c-control.ts
```

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

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
```

```
import { CursosProvider } from '../providers/cursos/cursos';
```

```
import { Observable } from 'rxjs/Observable';
```

```
import { CAddPage } from '../c-add/c-add';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-c-control',
```

```
  templateUrl: 'c-control.html',
```

```
})
```

```
export class CControlPage {
```

```
  courses: Observable<any>;
```

```

constructor(public navCtrl: NavController,
  public navParams: NavParams,
  private provider: CursosProvider,
  private toast: ToastController) {

  this.courses = this.provider.getAll();
}

removeCurso(key: string) {
  if (key) {
    this.provider.remove(key)
      .then(() => {
        this.toast.create({ message: 'Registro removido com sucesso.', duration:
3000}).present();
      })
      .catch(() => {
        this.toast.create({ message: 'Erro ao remover registro.', duration:
3000}).present();
      });
  }
}

editCurso(cursos: any) {
  this.navCtrl.push('CAddPage', { cursos: cursos });
}
}

```

```

ev-add.html
<ion-header>
  <ion-navbar>
    <div>

```

```

<br>
<br>
<h1>
  <b>&nbsp; &nbsp; &nbsp; {{titulo}}</b>
</h1>
<font color="#3db7ff" size="2">
  <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; Evento</p>
</font>
</div>
</ion-navbar>
</ion-header>

```

```

<ion-content padding>
<div class="div-mod-3">
  <form [formGroup]="form">
    <ion-item>
      <ion-label floating>Título</ion-label>
      <ion-input type="text" formControlName="titulo"></ion-input>
    </ion-item>
    <ion-item *ngIf="!form.controls.titulo.valid && (form.controls.titulo.dirt ||
form.controls.titulo.touched)" color="danger">
      <div [hidden]="!form.controls.titulo.errors.required">
        O campo é obrigatório.
      </div>
    </ion-item>

    <ion-item>
      <ion-label floating>Descrição</ion-label>
      <ion-input type="text" formControlName="descricao"></ion-input>
    </ion-item>
    <ion-item *ngIf="!form.controls.descricao.valid && (form.controls.descricao.dirt ||
form.controls.descricao.touched)" color="danger">
      <div [hidden]="!form.controls.descricao.errors.required">

```

O campo é obrigatório.

</div>

</ion-item>

<ion-item>

<ion-label floating>Local</ion-label>

<ion-input type="text" formControlName="local"></ion-input>

</ion-item>

<ion-item *ngIf="!form.controls.local.valid && (form.controls.local.dirt || form.controls.local.touched)" color="danger">

<div [hidden]="!form.controls.local.errors.required">

O campo é obrigatório.

</div>

</ion-item>

<ion-item>

<ion-label floating>Data</ion-label>

<ion-input type="text" formControlName="data"></ion-input>

</ion-item>

<ion-item *ngIf="!form.controls.data.valid && (form.controls.data.dirt || form.controls.data.touched)" color="danger">

<div [hidden]="!form.controls.data.errors.required">

O campo é obrigatório.

</div>

</ion-item>

<div padding>

<button ion-button block type="submit" [disabled]="!form.valid" (click)="onSubmit()">Salvar</button>

</div>

```

    </form>
  </div>
</ion-content>

```

ev-add.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

```

```

import { EventosProvider } from '../providers/eventos/eventos';

```

```

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

```

```

@IonicPage()

```

```

@Component({
  selector: 'page-ev-add',
  templateUrl: 'ev-add.html',
})

```

```

export class EvAddPage {

```

```

  titulo: string;
  form: FormGroup;
  evento: any;

```

```

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private FormBuilder: FormBuilder,
    private provider: EventosProvider,
    private toast: ToastController) {

```

```

    this.evento = this.NavParams.data.evento || {};
    this.createForm();

```



```

    this.setupPageTitle();
}

private setupPageTitle() {
    this.titulo = this.navParams.data.evento ? 'Alterando Evento' : 'Novo Evento';
}

createForm() {
    this.form = this.formBuilder.group({
        key: [this.evento.key],
        titulo: [this.evento.titulo, Validators.required],
        descricao: [this.evento.descricao, Validators.required],
        local: [this.evento.local, Validators.required],
        data: [this.evento.data, Validators.required]
    })
}

onSubmit(){
    if (this.form.valid) {
        this.provider.save(this.form.value)
            .then(() => {
                this.toast.create({ message: 'Evento salvo com sucesso.', duration:
3000}).present();
                this.navCtrl.pop();
            })
            .catch((e) => {
                this.toast.create({ message: 'Erro ao salvar contato.', duration: 3000}).present();
                console.error(e);
            });
    }
}
}

```

ev-control.html

<ion-header>

<ion-navbar>

<div>

Registros

 Eventos

<ion-title>

</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding>

<div class="div-mod-control">

<ion-list>

<ion-list-header align="center">

Eventos registrados

</ion-list-header>

```
<ion-item *ngFor="let evento of events | async">
```


`{{evento.titulo}}`


```

        <button                ion-button                clear                item-start
(click)="editEvento(evento)">EDITAR</button>
        <button                ion-button                clear                color="danger"                item-end
(click)="removeEvento(evento.key)">EXCLUIR</button>
    </ion-item>
    <br>
    <br>
    <font style="font-size:9pt; color:gray">
        <p align="right">Powered by UniInfo.</p>
    </font>
</ion-list>

</div>

</ion-content>

<ion-footer>
    <font>
        <p align="center">UniInfo 2017 &#174; - Todos os direitos reservados.</p>
    </font>
</ion-footer>

```

ev-control.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

import { EventosProvider } from '../providers/eventos/eventos';

import { Observable } from 'rxjs/Observable';

import { EvAddPage } from '../ev-add/ev-add';

```

```

@IonicPage()
@Component({
  selector: 'page-ev-control',
  templateUrl: 'ev-control.html',
})
export class EvControlPage {

  events: Observable<any>;

  constructor(public navCtrl: NavController,
    private provider: EventosProvider,
    private toast: ToastController) {

    this.events = this.provider.getAll();
  }

  removeEvento(key: string) {
    if (key) {
      this.provider.remove(key)
        .then(() => {
          this.toast.create({ message: 'Evento removido com sucesso.', duration: 3000
        }).present();
        })
        .catch(() => {
          this.toast.create({ message: 'Erro ao remover o evento.', duration: 3000
        }).present();
        });
    }
  }

  editEvento(evento: any) {
    this.navCtrl.push('EvAddPage', { evento: evento });
  }
}

```

```
}
```

```
home.html
```

```
<ion-header>
```

```
<ion-navbar>
```

```
<div>
```

```

```

```
<p align="center">Ambiente administrativo</p>
```

```
</div>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div class="div-center-home">
```

```
<!--Eventos-->
```

```
<br>
```

```
<br>
```

```
<div class="div-mod-home">
```

```
<ion-card class="card-mod-home">
```

```

```

```
<ion-card-content>
```

```
<ion-card-title>
```

```

<ion-item>
  <ion-icon name="ios-calendar" color="primary" item-start>
  </ion-icon>
  <h2>Eventos</h2>
  <h3>Criar, editar ou excluir registros de eventos.</h3>
  <br>
  <p>A aba eventos possui os registros de
    <br>avisos importantes ao aluno referente
    <br>a universsidade. O que deseja fazer? <br><br>
  </p>
</ion-item>
</ion-card-title>

<ion-item>
  <ion-icon name="ios-eye" item-start></ion-icon>
  <button ion-button clear (click)="goToEvControl()">
    Ir para eventos
  </button>
</ion-item>

<ion-item>
  <ion-icon name="ios-add" item-start></ion-icon>
  <button ion-button clear (click)="goToEvAdd()">
    Adicionar evento
  </button>
</ion-item>

</ion-card-content>
</ion-card>
</div>

```

```
<!--Salas-->
```

```
<div class="div-mod-home">
```

```
<ion-card class="card-mod-home">
```

```

```

```
<ion-card-content>
```

```
<ion-card-title>
```

```
<ion-item>
```

```
<ion-icon name="ios-school" color="primary" item-start>
```

```
</ion-icon>
```

```
<h2>Cursos</h2>
```

```
<h3>Criar, editar ou excluir registros de salas.</h3>
```

```
<br>
```

```
<p>A aba cursos possui os registros de
```

```
<br>todas as salas, por dia, de cada curso.
```

```
<br>O que deseja fazer? <br><br>
```

```
</p>
```

```
</ion-item>
```

```
</ion-card-title>
```

```
<ion-item>
```

```
<ion-icon name="ios-eye" item-start></ion-icon>
```

```
<button ion-button clear (click)="goToCControl()">
```

```
Ir para salas
```

```
</button>
```

```
</ion-item>
```

```
<ion-item>
```

```

    <ion-icon name="ios-add" item-start></ion-icon>
    <button ion-button clear (click)="goToCAdd()">
      Adicionar sala
    </button>
  </ion-item>

</ion-card-content>
</ion-card>
</div>

<!--Locais-->

<div class="div-mod-home">
  <ion-card class="card-mod-home">
    

    <ion-card-content>

      <ion-card-title>
        <ion-item>
          <ion-icon name="ios-pin" color="primary" item-start>
          </ion-icon>
          <h2>Locais</h2>
          <h3>Criar, editar ou excluir registros de locais.</h3>
          <br>
          <p>A aba locais possui os registros de

```



```

        <br>todos os pontos de referências do
        <br>campus. O que deseja fazer? <br><br>
    </p>
    </ion-item>

</ion-card-title>

<ion-item>
    <ion-icon name="ios-eye" item-start></ion-icon>
    <button ion-button clear (click)="goToLControl()">
        Ir para locais
    </button>
</ion-item>
<ion-item>
    <ion-icon name="ios-add" item-start></ion-icon>
    <button ion-button clear (click)="goToLAdd()">
        Adicionar local
    </button>
</ion-item>

</ion-card-content>
</ion-card>
</div>

<!--Mais-->

<div class="div-mod-home">

```

```

<ion-card class="card-mod-home">
  

  <ion-card-content>

    <ion-card-title>
      <ion-item>
        <ion-icon name="ios-contacts" color="primary" item-start>
        </ion-icon>
        <h2>Usuários</h2>
        <h3>Ações de usuários.</h3>
        <br>
        <p>
          <b>Atenção</b>! Usuários administradores tem
          <br> acesso completo ao sistema, assim como
          <br> todas as permissões de alteração do
          <br> banco de dados.</p>
        </ion-item>
      </ion-card-title>
      <ion-list>
        <ion-item>
          <ion-icon name="add" item-start></ion-icon>
          <button ion-button clear (click)="goToSignUp()">
            Adicionar administrador
          </button>
        </ion-item>
        <ion-item>
          <ion-icon name="ios-exit" item-start></ion-icon>
          <button ion-button clear (click)="signOut()">
            Fazer logoff
          </button>
        </ion-item>
      </ion-list>
    </ion-card-content>
  </ion-card>

```

```

        </ion-card-content>
    </ion-card>
</div>

</div>

</ion-content>

<ion-footer>
    <font>
        <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
    </font>
</ion-footer>

```

```

home.ts
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { AuthService } from '../providers/auth/auth-service';

import { SigninPage } from '../signin/signin';
import { SignupPage } from '../signup/signup';

import { EvAddPage } from '../ev-add/ev-add';
import { EvControlPage } from '../ev-control/ev-control';

import { CAddPage } from '../c-add/c-add';
import { CControlPage } from '../c-control/c-control';

import { LAddPage } from '../l-add/l-add';
import { LControlPage } from '../l-control/l-control';

@Component({

```

```
    selector: 'page-home',
    templateUrl: 'home.html'
  })
  export class HomePage {

    constructor(public navCtrl: NavController, private authService: AuthService) {

    }

    goToEvControl(){
      this.navCtrl.push(EvControlPage)
    }

    goToEvAdd(){
      this.navCtrl.push(EvAddPage)
    }

    goToCControl(){
      this.navCtrl.push(CControlPage)
    }

    goToCAdd(){
      this.navCtrl.push(CAddPage)
    }

    goToLControl(){
      this.navCtrl.push(LControlPage)
    }

    goToLAdd(){
      this.navCtrl.push(LAddPage)
    }

    goToSignUp(){
```

```

    this.navCtrl.push(SignupPage)
  }

  signOut() {
    this.authService.signOut()
      .then(() => {
        this.navCtrl.setRoot(SigninPage);
      })
      .catch((error) => {
        console.error(error);
      });
  }
}

```

l-add.html

```

<ion-header>
  <ion-navbar>
    <div>
      <br>
      <br>
      <h1>
        <b>&nbsp; &nbsp; {{titulo}}</b>
      </h1>
      <font color="#3db7ff" size="2">
        <p align="left">&nbsp; &nbsp; &nbsp; &nbsp; Local</p>
      </font>
    </div>
  </ion-navbar>
</ion-header>

```

```

<ion-content padding>
  <div class="div-mod-3">
    <form [formGroup]="form">
      <ion-item>
        <ion-label floating>Ambiente</ion-label>
        <ion-input type="text" formControlName="ambiente"></ion-input>
      </ion-item>
      <ion-item *ngIf="!form.controls.ambiente.valid && (form.controls.ambiente.dirt ||
form.controls.ambiente.touched)" color="danger">
        <div [hidden]="!form.controls.ambiente.errors.required">
          O campo é obrigatório.
        </div>
      </ion-item>

      <ion-item>
        <ion-label floating>Localização</ion-label>
        <ion-input type="text" formControlName="localizacao"></ion-input>
      </ion-item>
      <ion-item *ngIf="!form.controls.localizacao.valid &&
(form.controls.localizacao.dirt || form.controls.localizacao.touched)" color="danger">
        <div [hidden]="!form.controls.localizacao.errors.required">
          O campo é obrigatório.
        </div>
      </ion-item>

      <ion-item>
        <ion-label floating>Horário</ion-label>
        <ion-input type="text" formControlName="horario"></ion-input>
      </ion-item>
      <ion-item *ngIf="!form.controls.horario.valid && (form.controls.horario.dirt ||
form.controls.horario.touched)" color="danger">
        <div [hidden]="!form.controls.horario.errors.required">
          O campo é obrigatório.
        </div>

```

```

</ion-item>

<ion-item>
  <ion-label floating>Descrição</ion-label>
  <ion-input type="text" formControlName="descricao"></ion-input>
</ion-item>
<ion-item
  *ngIf="!form.controls.descricao.valid"
  (form.controls.descricao.dirt || form.controls.descricao.touched)" color="danger">
  <div [hidden]="!form.controls.descricao.errors.required">
    O campo é obrigatório.
  </div>
</ion-item>

<div padding>
  <button
    ion-button
    block
    type="submit"
    [disabled]="!form.valid"
    (click)="onSubmit()">Salvar</button>
</div>
</form>
</div>
</ion-content>

```

l-add.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

import { LocaisProvider } from '../providers/locais/locais';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@IonicPage()
@Component({
  selector: 'page-l-add',
  templateUrl: 'l-add.html',

```

```

    })
    export class LAddPage {
        titulo: string;
        form: FormGroup;
        local: any;

        constructor(public navCtrl: NavController,
            public navParams: NavParams,
            private formBuilder: FormBuilder,
            private provider: LocaisProvider,
            private toast: ToastController) {

            this.local = this.navParams.data.local || {};
            this.createForm();

            this.setupPageTitle();
        }

        private setupPageTitle() {
            this.titulo = this.navParams.data.local ? 'Alterando Local' : 'Novo Local';
        }

        createForm() {
            this.form = this.formBuilder.group({
                key: [this.local.key],
                ambiente: [this.local.ambiente, Validators.required],
                localizacao: [this.local.localizacao, Validators.required],
                horario: [this.local.horario, Validators.required],
                descricao: [this.local.descricao, Validators.required]
            })
        }

        onSubmit() {
            if (this.form.valid) {

```



```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div class="div-mod-control">
```

```
<br>
```

```
<br>
```

```
<ion-list>
```

```
<ion-list-header align="center">
```

```
<h1>Locais registrados</h1>
```

```
</ion-list-header>
```

```
<ion-item *ngFor="let local of places | async">
```

```
<font color="#b2b2b2" size="2" align="center">
```

```
{{local.ambiente}} | {{local.localizacao}}
```

```
</font>
```

```
<button ion-button clear item-start (click)="editLocal(local)">EDITAR</button>
```

```
<button ion-button clear color="danger" item-end
```

```
(click)="removeLocal(local.key)">EXCLUIR</button>
```

```
</ion-item>
```

```
<br>
```

```
<br>
```

```
<font style="font-size:9pt; color:gray">
```

```
<p align="right">Powered by UniInfo.</p>
```

```
</font>
```

```
</ion-list>
```

```
</div>
```

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

```
<ion-footer>
```

```

<font>
  <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
</font>
</ion-footer>

```

l-control.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

import { LocaisProvider } from '../providers/locais/locais';

import { Observable } from 'rxjs/Observable';

import { LAddPage } from '../l-add/l-add';

@IonicPage()
@Component({
  selector: 'page-l-control',
  templateUrl: 'l-control.html',
})
export class LControlPage {

  places: Observable<any>;

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private provider: LocaisProvider,
    private toast: ToastController) {

    this.places = this.provider.getAll();
  }

```



```

    </font>
    <ion-title>
  </ion-title>
</div>
</ion-navbar>

</ion-header>

<ion-content padding>
  <form #form="ngForm" novalidate>
    <div class="center-div">
      <ion-list>

        <ion-item>
          <ion-label floating>E-mail de recuperação</ion-label>
          <ion-input type="text" name="email" [(ngModel)]="userEmail" #email="ngModel"
required></ion-input>
        </ion-item>
        <ion-item *ngIf="email.errors && (email.dirty || email.touched)" class="text-
danger">
          O campo é obrigatório
        </ion-item>
      </ion-list>

      <br>
      <br>
      <br>

      <button ion-button block color="primary" [disabled]="!form.form.valid"
(click)="resetPassword()">Enviar</button>
    </div>
  </form>

</ion-content>

```

```

<ion-footer>
  <font>
    <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
  </font>
</ion-footer>

```

resetpassword.ts

```

import { Component, ViewChild } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

```

```

import { NgForm } from '@angular/forms';
import { AuthService } from '../providers/auth/auth-service';

```

```

@IonicPage()
@Component({
  selector: 'page-resetpassword',
  templateUrl: 'resetpassword.html',
})
export class ResetpasswordPage {
  userEmail: string = "";
  @ViewChild('form') form: NgForm;

```

```

  constructor(public navCtrl: NavController,
               public NavParams: NavParams,
               private toastCtrl: ToastController,
               private authService: AuthService) {
  }

```

```

  resetPassword() {
    if (this.form.valid){

```

```

      let toast = this.toastCtrl.create({duration: 3000, position: 'bottom'});
      this.authService.resetPassword(this.userEmail)

```

```
.then(() => {
    toast.setMessage('Sua solicitação foi enviada para seu e-mail.')
    toast.present()

    this.navCtrl.pop();
})
.catch((error: any) => {

    if (error.code == 'auth/invalid-email') {
        toast.setMessage('E-mail inválido.')
    }
    else if (error.code == 'auth/user-not-found') {
        toast.setMessage('Usuário não encontrado.');
```

```

```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<ion-list>
```

```
<ion-item>
```

```
<ion-label floating>E-mail</ion-label>
```

```
<ion-input type="email" name="email" [(ngModel)]="user.email"
#email="ngModel" required></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="email.errors && (email.dirty || email.touched)" class="text-
danger">
```

```
    O campo é obrigatório.
```

```
</ion-item>
```

```
<ion-item>
```

```
<ion-label floating>Senha</ion-label>
```

```
<ion-input type="password" name="password" [(ngModel)]="user.password"
#password="ngModel" required></ion-input>
```

```
</ion-item>
```

```
<ion-item *ngIf="password.errors && (password.dirty || password.touched)"
class="text-danger">
```

```
    O campo é obrigatório
```



```

    </ion-item>
  </ion-list>

  <div align="center">

    <br>
    <br>

    <button      ion-button      color="primary"      [disabled]="!form.form.valid"
(click)="signIn()">Entrar</button>

    <br>
    <br>

    <button ion-button clear full (click)="goResetpassword()">ESQUECI MINHA
SENHA</button>

  </div>
</div>
</form>

</ion-content>

<ion-footer>
  <font>
    <p align="center">UnilInfo 2017 &#174; - Todos os direitos reservados.</p>
  </font>
</ion-footer>

```

signin.ts

```

import { Component, ViewChild } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

```

```

import { NgForm } from '@angular/forms';
import { User } from '../providers/auth/user';
import { AuthService } from '../providers/auth/auth-service';

import { ResetpasswordPage } from '../resetpassword/resetpassword';
import { HomePage } from '../home/home';

@IonicPage()
@Component({
  selector: 'page-signin',
  templateUrl: 'signin.html',
})
export class SigninPage {
  user: User = new User();
  @ViewChild('form') form: NgForm;

  constructor(public navCtrl: NavController,
               public navParams: NavParams,
               private toastCtrl: ToastController,
               private authService: AuthService) {
  }

  goResetpassword() {
    this.navCtrl.push(ResetpasswordPage)
  }

  signIn() {
    if (this.form.valid) {
      this.authService.signIn(this.user)
        .then(() => {
          this.navCtrl.setRoot(HomePage)
        })
        .catch((error: any) => {

```

```

let toast = this.toastCtrl.create({ duration:3000, position: 'bottom' });

if (error.code == 'auth/invalid-email') {
  toast.setMessage('E-mail informado é inválido.');
```

```

}

else if (error.code == 'auth/user-disabled') {
  toast.setMessage('Usuário está desativado.');
```

```

}

else if (error.code == 'auth/user-not-found') {
  toast.setMessage('Usuário não foi encontrado.');
```

```

}

else if (error.code == 'auth/wrong-password') {
  toast.setMessage('A senha digitada não é válida.');
```

```

}
toast.present();
});
}
}

}

```

signup.html

```
<ion-header>
```

```
<ion-navbar>
```

```
<ion-title>Criar uma nova conta</ion-title>
```

```
</ion-navbar>
```

```
</ion-header>
```

```

<ion-content padding>
<form #form="ngForm" novalidate>
  <ion-list>

    <ion-item>
      <ion-label stacked>E-mail</ion-label>
      <ion-input type="text" name="email" [(ngModel)]="user.email"
#email="ngModel" required></ion-input>
    </ion-item>
    <ion-item *ngIf="email.errors && (email.dirty || email.touched)" class="text-
danger">
      O campo é obrigatório
    </ion-item>

    <ion-item>
      <ion-label stacked>Senha</ion-label>
      <ion-input type="password" name="password" [(ngModel)]="user.password"
#password="ngModel" require></ion-input>
    </ion-item>
    <ion-item *ngIf="password.errors && (password.dirty || password.touched)"
class="text-danger">
      O campo é obrigatório
    </ion-item>

  </ion-list>

  <button ion-button block color="primary" [disabled]="!form.form.valid"
(click)="createAccount()">
    CRIAR CONTA
  </button>
</form>

</ion-content>

```

signup.ts

//Importações padrões do Ionic

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

import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';

//Importações do provedor

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

import { User } from '../providers/auth/user';

import { AuthService } from '../providers/auth/auth-service';

//Importação de página

import { HomePage } from '../home/home';

@IonicPage()

@Component({

 selector: 'page-signup',

 templateUrl: 'signup.html',

})

export class SignupPage {

 user: User = new User();

 @ViewChild('form') form: NgForm;

 constructor(

 public navCtrl: NavController,

 public NavParams: NavParams,

 public toastCtrl: ToastController,

 public authService: AuthService) {

 }

```

createAccount() {
  if (this.form.valid) {
    let toast = this.toastCtrl.create({ duration: 3000, position: 'bottom' });

    this.authService.createUser(this.user)
      .then((user: any) => {
        user.sendEmailVerification();

        toast.setMessage('Usuário cadastrado com sucesso.')
        toast.present();

        this.navCtrl.setRoot(HomePage)
      })
      .catch((error: any) => {
        if (error.code == 'auth/email-already-in-use') {
          toast.setMessage('O e-mail digitado já está em uso');
        } else if (error.code == 'auth/invalid-email') {
          toast.setMessage('O e-mail digitado não é válido');
        }
        else if (error.code == 'auth/operation-not-allowed') {
          toast.setMessage('Não está habilitado criar usuários');
        }
        else if (error.code == 'auth/weak-password') {
          toast.setMessage('A senha informada é muito fraca');
        }
        toast.present();
      });
  }
}

```

Provider de AUTENTICAÇÃO auth-service.ts

```
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs/Observable';
import { AngularFireAuth } from 'angularfire2/auth';
import { User } from './user';
import * as firebase from 'firebase/app';
```

```
@Injectable()
```

```
export class AuthService {
```

```
  user: Observable<firebase.User>;
```

```
  constructor(private angularFireAuth: AngularFireAuth) {
```

```
    this.user = angularFireAuth.authState;
```

```
  }
```

```
  signIn(user: User) {
```

```
    return this.angularFireAuth.auth.signInWithEmailAndPassword(user.email,
user.password);
```

```
  }
```

```
  signOut() {
```

```
    return this.angularFireAuth.auth.signOut();
```

```
  }
```

```
  resetPassword(email: string) {
```

```
    return this.angularFireAuth.auth.sendPasswordResetEmail(email);
```

```
  }
```

```

    createUser(user: User) {
        return this.angularFireAuth.auth.createUserWithEmailAndPassword(user.email,
user.password);
    }
}

```

Provider de AUTENTICAÇÃO user.ts

```

export class User {
    email: string;
    password: string;
}

```

Provider cursos.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

```

@Injectable()

```

```

export class CursosProvider {

```

```

    private PATH = 'c_data/';

```

```

    constructor(private db: AngularFireDatabase) { }

```

```

    getAll() {
        return this.db.list(this.PATH)
            .snapshotChanges()
            .map(changes => {
                return changes.map(c => ({
                    key: c.payload.key,
                    ...c.payload.val()
                }));
            });
    }
}

```



```

    })
  }

  get(key: string) {
    return this.db.object(this.PATH + key)
      .snapshotChanges()
      .map(c => {
        return { key: c.key, ...c.payload.val() };
      })
  }

  save(cursos: any) {
    return new Promise((resolve, reject) => {
      if (cursos.key) {
        this.db.list(this.PATH)
          .update(cursos.key, {
            turno: cursos.turno,
            curso: cursos.curso,
            semestre: cursos.semestre,
            sseg: cursos.sseg,
            /*
            ster: cursos.ster,
            squa: cursos.squa,
            squi: cursos.squi,
            ssex: cursos.ssex
            */
          })
          .then(() => resolve())
          .catch((e) => reject(e));
      } else {
        this.db.list(this.PATH)
          .push({
            turno: cursos.turno,
            curso: cursos.curso,

```

```

        semestre: cursos.semestre,
        sseg: cursos.sseg,
        /*
        ster: cursos.ster,
        squa: cursos.squa,
        squi: cursos.squi,
        ssex: cursos.ssex
        */
    })
    .then(() => resolve());
}
});
}

remove(key: string) {
    return this.db.list(this.PATH).remove(key);
}

}

```

Provider eventos.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

```

@Injectable()

```

```

export class EventosProvider {

```

```

    private PATH = 'e_data/';

```

```

    constructor(private db: AngularFireDatabase) { }

```

```

getAll() {
  return this.db.list(this.PATH)
    .snapshotChanges()
    .map(changes => {
      return changes.map(ev => ({
        key: ev.payload.key,
        ...ev.payload.val()
      }));
    })
}

get(key: string) {
  return this.db.object(this.PATH + key)
    .snapshotChanges()
    .map( ev => {
      return { key: ev.key, ...ev.payload.val() };
    })
}

save(evento: any) {
  return new Promise((resolve, reject) => {
    if (evento.key) {
      this.db.list(this.PATH)
        .update(evento.key, { titulo: evento.titulo, descricao: evento.descricao,
local: evento.local, data: evento.data })
        .then(() => resolve())
        .catch((e) => reject(e));
    } else {
      this.db.list(this.PATH)
        .push({ titulo: evento.titulo, descricao: evento.descricao, local:
evento.local, data: evento.data })
        .then(() => resolve());
    }
  });
}

```

```

    }

    remove(key: string) {
        return this.db.list(this.PATH).remove(key);
    }

}

```

Provider locais.ts

```

import { Injectable } from '@angular/core';
import { AngularFireDatabase, snapshotChanges } from 'angularfire2/database';

```

```

@Injectable()

```

```

export class LocaisProvider {

```

```

    private PATH = 'l_data/';

```

```

    constructor(private db: AngularFireDatabase) { }

```

```

    getAll() {
        return this.db.list(this.PATH)
            .snapshotChanges()
            .map(changes => {
                return changes.map(l => ({
                    key: l.payload.key,
                    ...l.payload.val()
                }));
            })
    }

```

```

    get(key: string) {
        return this.db.object(this.PATH + key)
            .snapshotChanges()
    }

```

```

        .map(l => {
            return { key: l.key, ...l.payload.val() };
        })
    }

    save(local: any) {
        return new Promise((resolve, reject) => {
            if (local.key) {
                this.db.list(this.PATH)
                    .update(local.key, { ambiente: local.ambiente, localizacao:
local.localizacao, horario: local.horario, descricao: local.descricao })
                    .then(() => resolve())
                    .catch((e) => reject(e));
            } else {
                this.db.list(this.PATH)
                    .push({ ambiente: local.ambiente, localizacao: local.localizacao, horario:
local.horario, descricao: local.descricao })
                    .then(() => resolve());
            }
        });
    }

    remove(key: string) {
        return this.db.list(this.PATH).remove(key);
    }
}

```