

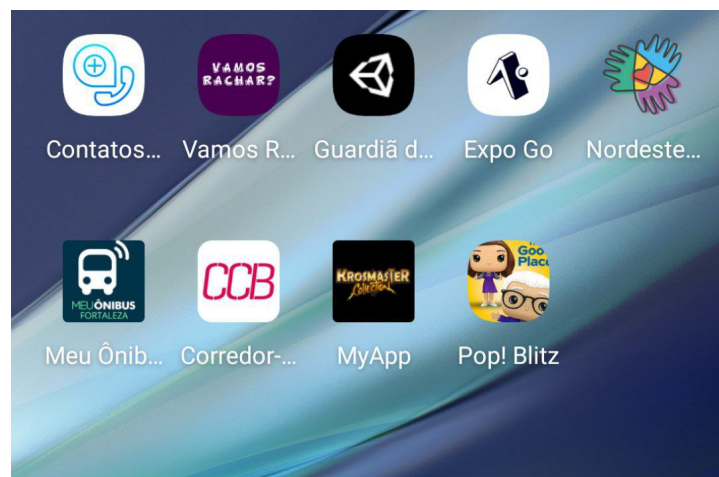
Projeto Final

Felipe Henrique Moraes Araújo - 421768

Utilizei o código disponibilizado no github, e fiz algumas modificações

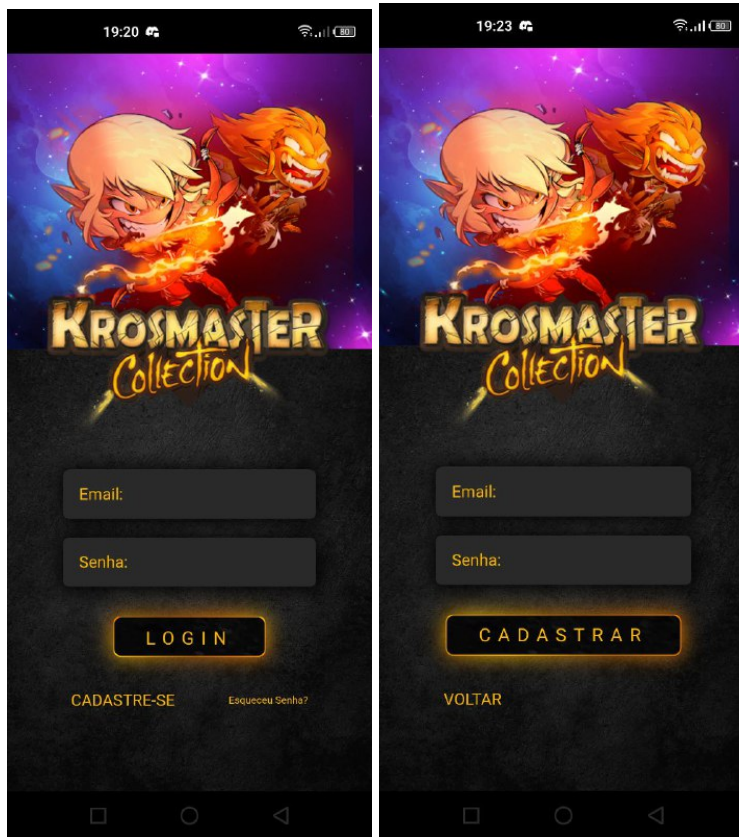
1. Kromaster Collection

Meu app é para salvar coleção de carta de um jogador de Kromaster. Assim o jogador tira foto de cada carta, salva as informações dela e compartilha no app. Eu tive a infeliz ideia de fazer em Ionic, tive muita dificuldade, além de ter deixado pra fazer nesses últimos dias, então na reta final do app, tive um problema com o Ionic e não consegui criar mais telas, mando nesse documento o que consegui fazer: SPLASH DO APP e ÍCONE



2. Tela de Login e Registro de usuário

Para logar uso o auxílio do firebase, para fazer toda a questão de autenticação e registro do usuário



3. Código Firebase

Fiz as principais funções: logar, autenticar, criar e sair. Infelizmente a parte de cadastrar o usuário com foto não foi. O problema explicou mais na frente

```

import { Router } from '@angular/router';
import { Injectable } from '@angular/core';
import { AngularFireAuth } from '@angular/fire/compat/auth';

@Injectable({
  providedIn: 'root'
})
export class AuthService {

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

  login(user){
    return this.fireAuth.signInWithEmailAndPassword(user.email, user.password)
  }
  register(user){
    return this.fireAuth.createUserWithEmailAndPassword(user.email, user.password);
  }
  logout(){
    return this.fireAuth.signOut();
  }
  getAuth(){
    return this.fireAuth;
  }
}

```

4. Main

A home é dividido em 3 telas, 1 com a tela com a coleção do usuário, outra para adicionar a carta e a última para ver e editar o perfil do usuário. Ela só pode ser logado se tiver logado no firebase.

Nesse trecho de código que faz isso, caso user que é chamado na função do firebas retornar vazio, o app é redirecionado para tela de login.

```

canActivate(): Promise<boolean>{
  console.log("auth")
  return new Promise(resolve => {
    this.authService.getAuth().onAuthStateChanged(user =>{
      if(!user) this.router.navigate(['login']);

      resolve(user ? true : false);
    });
  })
}

```

Assim como se o usuário já tiver logado ele não passará pela tela de login

```

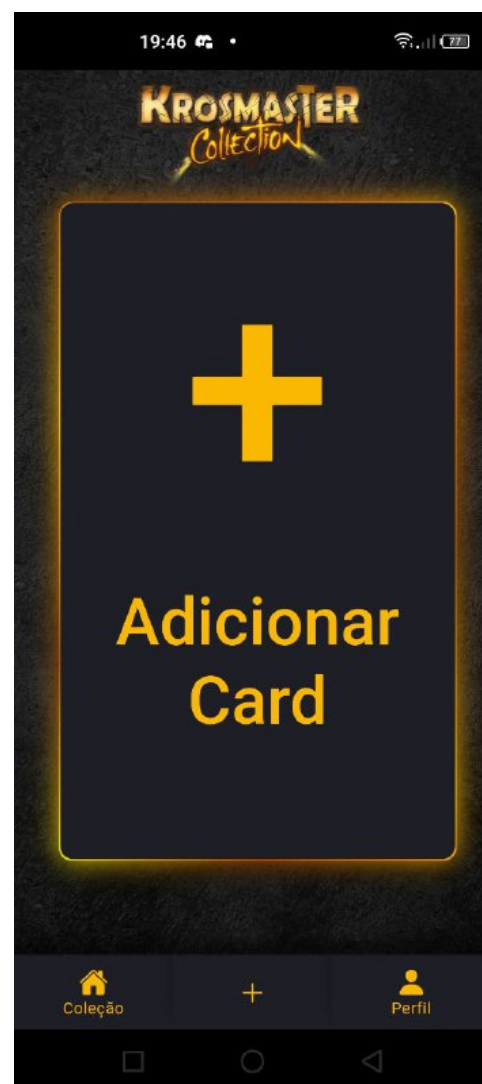
canActivate(): Promise<boolean>{
  console.log("login")
  return new Promise(resolve => {
    this.authService.getAuth().onAuthStateChanged(user =>{
      if(user) this.router.navigate(['tabs/tab1']);

      resolve(user ? false : true);
    });
  });
}
}

```

5. Coleção

A coleção do jogador aparece em uma forma de carrossel, sendo possível arrastar para direita e esquerda, e a última carta é um botão para adicionar uma nova (Ela também aparece quando a coleção do usuário tá vazia)



cada card tem 3 botões, um para apagar, outro para ver as informações completas e outra para compartilhar, apenas de apagar que cheguei a implementar. E se clicar na parte da imagem do card, a foto fica ampliada



6. Cards Service

Criei uma classe para fazer as requisições do bd do usuário, com getAll, get, delete, update e add; GetAll que preenche a tela anterior

```
getCards(){
  return this.db.list(this.path)
    .snapshotChanges()
    .pipe(
      map(actions=>{
        return actions.map(a =>{
          console.log({key: a.payload.key, tes: a.payload.val()})
          const id = a.payload.key;
          let data: Cards = a.payload.val();
          data.id = id;
          return {...data}
        })
      })
    )
}
```

Get. Que tá funcionando, porém não implementei a tela para apresentar as informações em uma tela

```
getCard(id: string){  
  return this.db.object(this.path+id)  
    .snapshotChanges().pipe(map(a => {  
      const id = a.payload.key;  
      let data: Cards = a.payload.val();  
      data.id = id;  
      return {...data}  
    })))  
}
```

delete

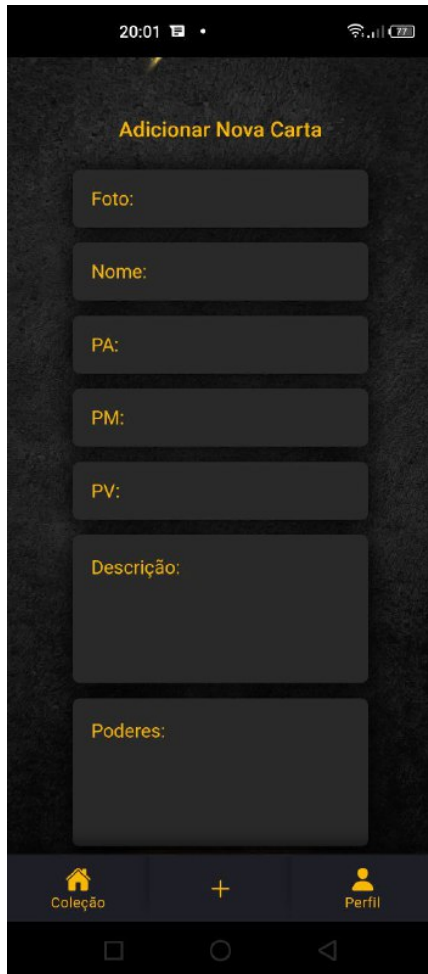
```
deleteCards(id: string){  
  return this.db.list(this.path).remove(id);  
}
```

add e update. Aqui recebe um objeto do tipo card, caso tenha um id válido, funcionará com um update, caso contrário ele servirá como add.

```
addCards(card: Cards){  
  return new Promise<void>((resolve, reject) =>{  
    if (card.id != "none"){  
      this.db.list(this.path)  
        .update(card.id, {nome: card.nome,  
          descricao: card.descricao,  
          pm: card.pm,  
          pa: card.pa,  
          pv: card.pv,  
          poderes: card.poderes,  
          foto: card.foto})  
        .then(() => resolve())  
        .catch((e) => reject(e));  
    }  
    else{  
      this.db.list(this.path).push({nome: card.nome,  
        descricao: card.descricao,  
        pm: card.pm,  
        pa: card.pa,  
        pv: card.pv,  
        poderes: card.poderes,  
        foto: card.foto}).then(() => resolve())  
    }  
  })  
}
```

7. Adicionar Card

a segunda tela do aplicativo é para o usuário cadastrar sua carta



Ela também seria um clone para edição da carta, mas não cheguei a implementar. Mas a tela de cadastro ta funcionando ok, exceto na parte da foto.

Eu criei o código para salvar a foto no storage e tudo mais. Porém, quando fui testar no celular, o plugin de escolher foto não tava funcionando.

```
async openGallery(){
  const option: CameraOptions = {
    quality: 100,
    destinationType: this.camera.DestinationType.FILE_URI,
    sourceType: this.camera.PictureSourceType.PHOTOLIBRARY,
    correctOrientation: true
  }
  try {
    const fileUri = await
this.camera.getPicture(option).then((imageData)=>{
  this.selectedPhoto =
this.dataUrltoBlob('data:image/jpeg;base64,'+imageData);
  this.uploadPicture();
});
```

```

    }, (err)=>{
        this.presentToast(err)
    }
);
let file: string;

} catch (error) {
    console.log(error)
}
}

uploadPicture(){
    const ref = this.afStorage.ref('icard/ionic.jpg');
    const task = ref.put(this.selectedPhoto);
}

dataUrltoBlob(foto){
    let binary = atob(foto.split(',')[1]);
    let array = []
    for (let i=0; i<binary.length; i++){
        array.push(binary.charCodeAt(i));
    }
    return new Blob([new Uint8Array(array)], {type: 'image/jpeg'})
}

```

8. O Problema

Então, depois de muito quebrar a cabeça descobri que o cordova, que auxiliar o ionic a utilizar os recursos nativos, eram uma versão antiga, então, não pegava o plugin da câmera, tentei atualizar de tudo que é forma e não consegui, e uma dessas tentativas eu tentei atualizar o ionic, e depois o projeto não pegou mais. Mesmo eu reinstalando ele o projeto acusava que o ionic não existe no pc

```

PS C:\Users\Carlos\Documents\Angular2\2021\pdm\front\icard2> ionic serve
ionic : O termo 'ionic' não é reconhecido como nome de cmdlet, função, arquivo de script ou programa operável. Verifique a grafia do nome ou, se um caminho
tiver sido incluído, veja se o caminho está correto e tente novamente.
No linha:1 caractere:1
+ ionic serve
+ ~~~~~

```

ionic : O termo 'ionic' não é reconhecido como nome de cmdlet, função, arquivo de script ou programa operável. Verifique a grafia do nome ou, se um caminho

tiver sido incluído, veja se o caminho está correto e tente novamente.

No linha:1 caractere:1

+ ionic serve

+ ~~~~~

+ CategoryInfo : ObjectNotFound: (ionic:String) [],
CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

9. Vídeo

<https://youtu.be/L9dKtlgqDeY>

10. GitHub

<https://github.com/felipehenriquee/kromaster>