**TRAVELDIARY - INTHOM - EGCM**

Realização**: João Pires 23804 & José Sampaio 20734**

[**j.pires@ipvc.pt**](mailto:j.pires@ipvc.pt)

**joseso@ipvc.pt**

**Introdução**

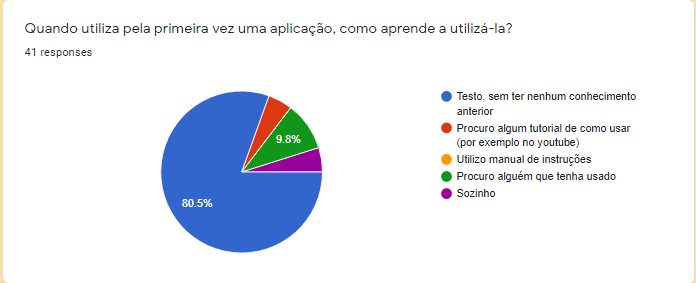
Este trabalho tem como objetivo a criação de uma app híbrida de apoio de viagens, permite marcar pontos de interesse e criar percursos a partir dos mesmos.

**Questionário**

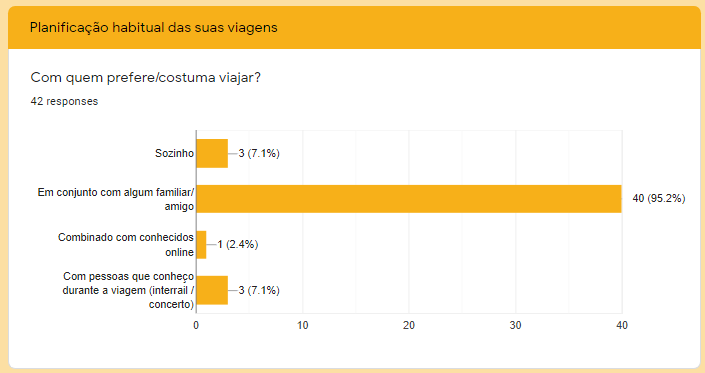
**Url do questionário:**

<https://docs.google.com/forms/d/19D37qz71iWWRtAwF8DRxmMJKi8_3_K7tokHzsrfpJ3s/edit>

**Principais informações retiradas do questionário:**



Principal modo de aprendizagem da app, maioria dos utilizadores preferem um design simples e intuitivo sem necessidade de ajuda exterior.



Esta pergunta foi feita para verificar a necessidade de um chat na aplicação, e com a falta de popularidade da terceira opção verificamos que não.

**Analise de tarefas**

**1. Quem vai utilizar o sistema?**

-Para todas as idades

**2. Que tarefas executa atualmente?**

-Procurava os pontos de interesse do lugar

-Dos locais que gostaria de visitar, escolheria com base em requisitos específicos

-Via sugestões de outras pessoas que visitaram aquele local

**3. Que tarefas são desejáveis?**

-Pesquisa de locais de interesse

-Indicação de preços

-Planeamento de percursos (menos importante)

**4. Como se aprendem as tarefas?**

-Tentativa erro

**5. Onde são desempenhadas as tarefas?**

-Computador portátil

-Smartphone

**6. Quais as relações entre utilizadores e informação?**

-Podem deixar sugestões de percursos para que outras pessoas posem experimentar

**7. Que outras ferramentas tem o utilizador?**

-Pode rever os percursos que foram guardados

**8. Como comunicam os utilizadores entre si?**

-Vendo os percursos que foram compartilhados pelas outras pessoas

**9. Qual a frequência de desempenho das tarefas?**

-Pesquisa de locais de interesse

-Indicação de preços

**10. Quais as restrições de tempo impostas?**

(nenhuma informação)

**11. Que acontece se algo correr mal?**

-Ter uma opção que o utilizador possa reportar algo ao programador

**Modelo conceptual**

**Difícil:**

**- Planeamento de viagens**

O utilizador pode planear as suas viagens, definir percursos a partir de seleção de pontos de interesse.

**- Cenário de atividades:**

O Manuel depois da consulta sobre os locais que decidiu visitar, quis planear um trajeto para a sua Lua de mel. Para isso ele utilizou a função do planeamento de viagens da aplicação TravelDiary, que após preencher com os pontos que queria visitar, permitiu criar um percurso, que irá fazer com a mulher nos locais turísticos.

**Médio:**

**- Criação de pontos de interesse**O utilizador pode criar um locar, como local de interesse para partilhar consulta mais tarde.

**- Cenário de atividades:**

O Manuel após acabar a viagem decidiu partilhar o seu grandioso momento quando experienciou uma paisagem ao por do sol, mas apenas queria partilhar com os seus amigos.

Para isto utilizou a função de pontos de interesse na aplicação TravelDiary para marcar os seus locais, marcada no local do acontecimento,.

**Fácil:**

**- Login e Consulta**

login na aplicação.

Verificar o seu conteúdo, com informações adicionais.

**- Cenário de atividades:**

O Manuel quer planear uma viagem de Lua de mel com a sua mulher e para isso ele foi procurar um Hotel, num local onde tivesse vários pontos de interesse românticos. Para isso ele, fez o registo na aplicação TravelDiary e utilizou a função do planeamento de viagens, que lhe permitiu fazer uma pesquisa para rapidamente encontrar o local destinado e que fosse barato.

**Metáfora**

**- Exemplos:**

Planear num só toque.

Folheie as suas experiências.

Planeie na palma da sua mão.

# Difícil:

**- Planeamento de viagens**

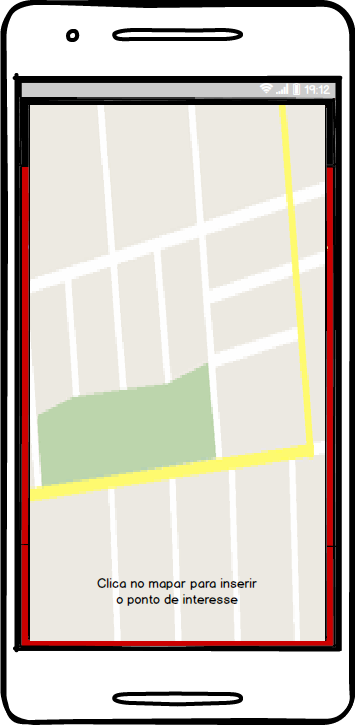
O Manuel depois da consulta sobre os locais que decidiu visitar, quis planear um trajeto para a sua Lua de mel. Para isso ele utilizou a função do planeamento de viagens da aplicação TravelDiary e colocou no bloco de notas a descrição das atividades que iria fazer.

Clicou no botão percursos onde se encontram todos os percursos e em seguida no “+”

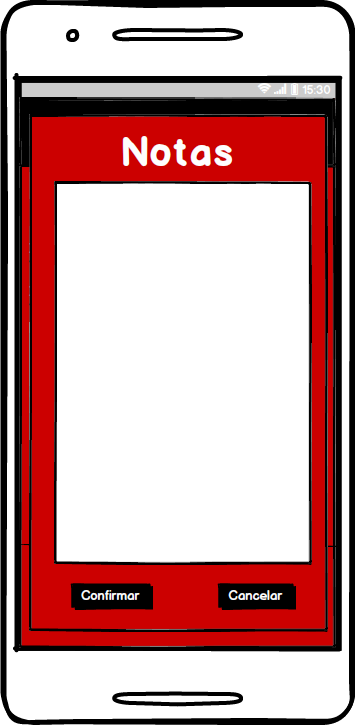
para adicionar um novo.



Inseriu a partida, e clicou no botão para adicionar ponto de interesse.



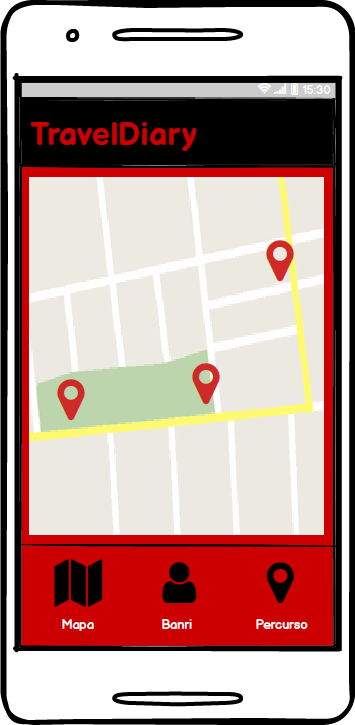
Abriu Pop-up para adicionar ponto de interesse, caso já tivesse sido criado simplesmente ficaria adicionado ao percurso, se não tivesse criado iria para a página de criar ponto de interesse. \*



No final adicionou as notas e clicou em confirmar para continuar.

# Médio:

**- Partilha (criação de pontos de interesse)**

O Manuel após acabar a viagem decidiu partilhar o seu grandioso momento quando experienciou a catedral de Sevilha.

\*Para isto o Manel clicou no mapa e depois pressionando o dedo no local do mapa, durante cerca de 2 segundos, abriu a página de criar pontos de interesse.



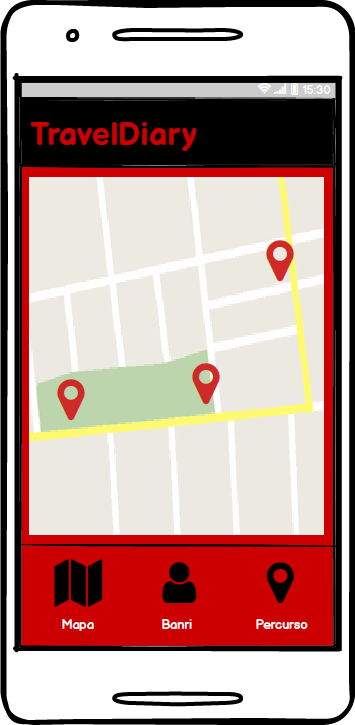
Para o ponto de interesse meteu no nome “Catedral de Sevilha” e no tipo meteu a opção outros.

# Fácil:

**- Login e Consulta**

O Manuel quer planear uma viagem de Lua de mel com a sua mulher e foi procurar um Hotel, num local onde tivesse vários pontos de interesse românticos.

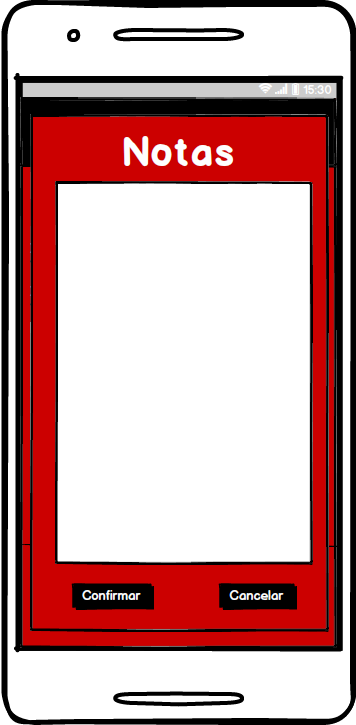
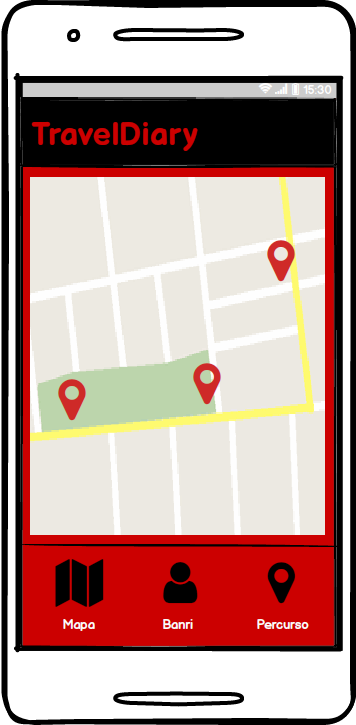
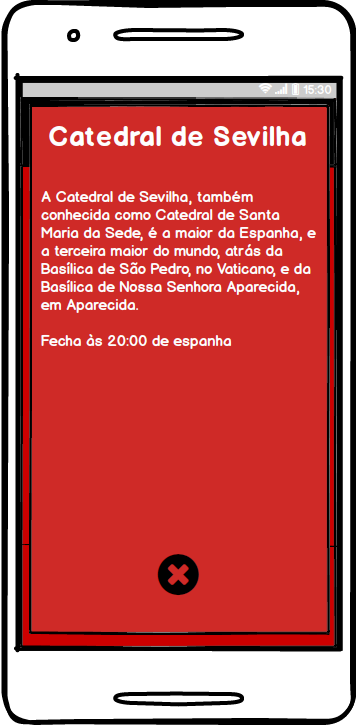
Para isso ele, fez o registo na aplicação TravelDiary.



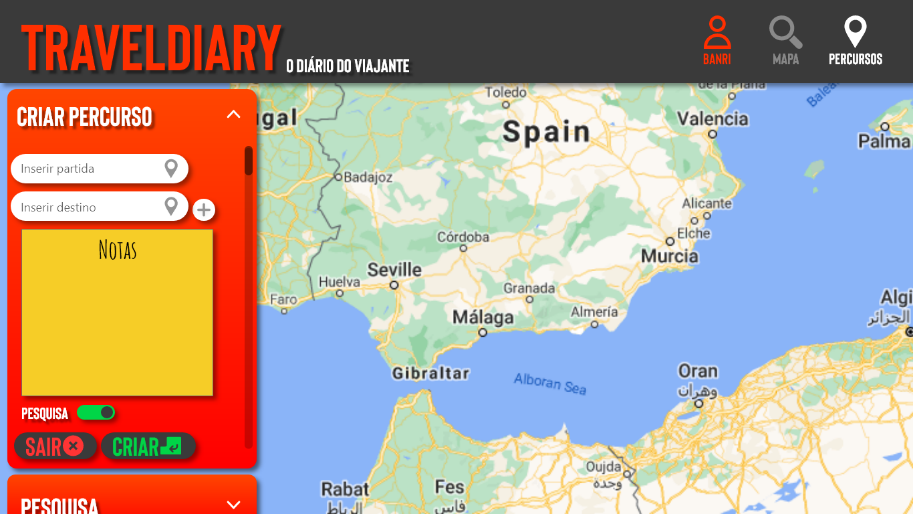
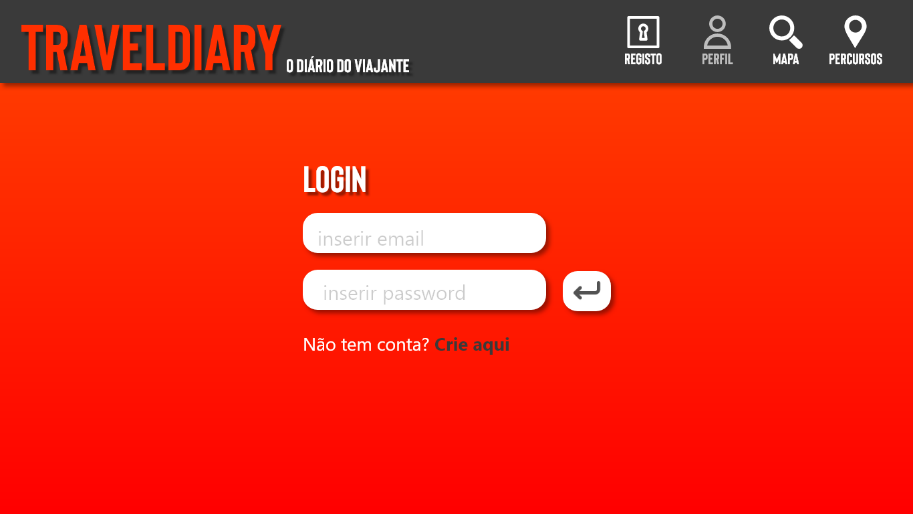
E clicou no mapa para verificar o hotel e os pontos de interesse.

**Protótipo de Baixa fidelidade**

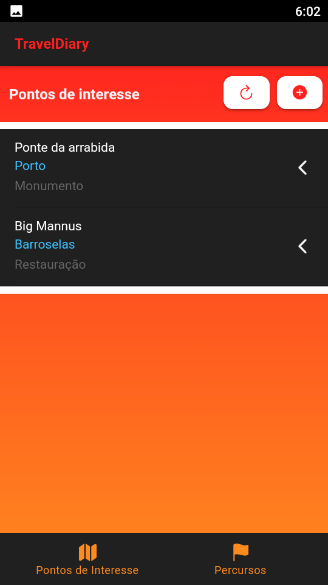
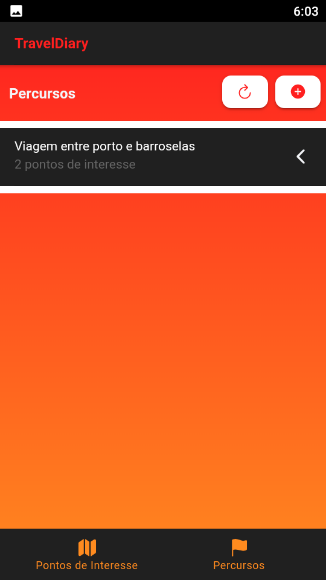
**- Mobile:**

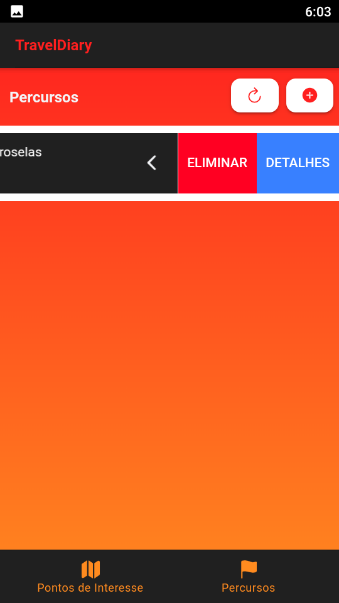
****

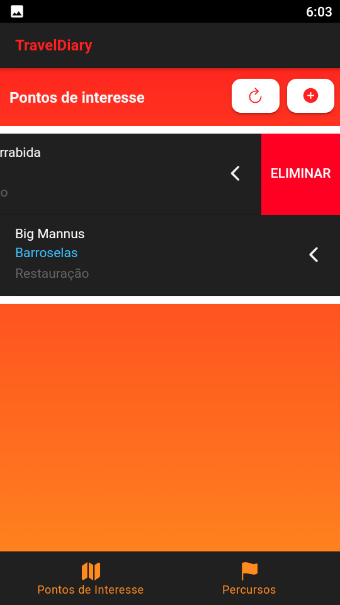
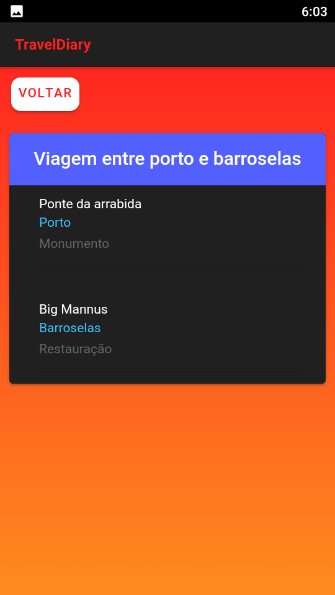
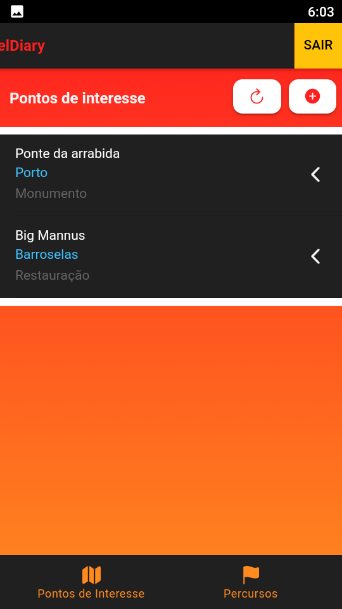
**- Web:**

**  **

**Versão Final do Projeto**

**  **

**  **

**  **

**- Routing aplicado na navegação da app:**

Na passagem entre as páginas, foram utilizados em todas as paginas, para que seja possível navegação.

Exemplo do método utilizado:

  public Pdi(){

    this.router.navigate(["pdi"]);

  }

**- Utilizar o Angular Router: Router, NavigationExtras e ActivatedRoute:**

Router foi exemplificado acima.

Navigation extras foi utilizado para a passagem de informação de um percurso da pagina percursos.ts para a pagina detalhes-percurso.ts.

Percursos.ts:

//nanigationExtras

  public verPercurso(*id*: *number*) {

*let* navigationExtras: NavigationExtras;

    for(*let* i = 0; i< this.percursos.length; i++){

      if(this.percursos[i].id == *id*){

        navigationExtras = {

          state: {

            percurso: this.percursos[i]

          }

        };

        this.router.navigate(['detalhe-percurso'], navigationExtras);

      }

    }

  }

Detalhes-percuro.ts:

*constructor*(private *route*: ActivatedRoute, private *router*: Router) {

    this.route.queryParams.subscribe(*params* *=>* {

      if (this.router.getCurrentNavigation().extras.state) {

        this.percurso = this.router.getCurrentNavigation().extras.state.percurso;

      }

    });

  }

Activated route permite a função subscribe que espera pela informação.

Extras.state retorna um boolean com o estado da passagem de informação.

**- Navegar e passar informação (parâmetros) entre páginas:**

Foi demonstrado acima.

- **Utilizar ícones da framework:**

Foram utilizados na maioria das paginas

Exemplo:

****

Codigo:

        <ion-button (click)="atualizar()">

          <ion-icon name="refresh-outline"></ion-icon>

        </ion-button>

      </ion-col>

      <ion-col size="2">

        <ion-button (click)="criarPonto()">

          <ion-icon name="add-circle"></ion-icon>

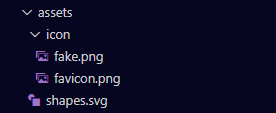
        </ion-button>

**- Estruturar e organizar devidamente os vários módulos e assets:**

Utilização de módulos para o serviço storage (CRUD) :

import { IonicStorageModule } from '@ionic/storage';

Nos assets foi utilizado para modificar o icon da app:



**- Manipular starters (templates) - se aplicável:**

A aplicação foi iniciada como blank.

**- Utilizar informação proveniente de ficheiros JSON:**

Foi utilizado para o armazenamento do login e password dos utilizadores.

  admin = {

    "nome" : "admin",

    "password" : "admin"

  }

Função utilizada para verificação do ficheiro json acima:

  verificaConta(*user*: *string*, *password*: *string*){

    if(*user* == this.admin.nome && *password* == this.admin.password){

      this.login();

    }else{

      this.showToast("nome ou palavra-passe incorreto");

    }

  }

}

**- Estruturar/Disponibilizar conteúdos através dos Components:**

Apenas o titulo no topo da app foi utilizado como componente global assim como o item-option slide para sair da app.

App:



Código:

<ion-app>

  <ion-router-outlet></ion-router-outlet>

  <ion-header>

          <ion-item-sliding>

            <ion-item>

              <ion-label> <b>

                TravelDiary

              </b>  </ion-label>

            </ion-item>

            <ion-item-options side="end">

              <ion-item-option color="warning" (click)="showExitConfirm()">Sair</ion-item-option>

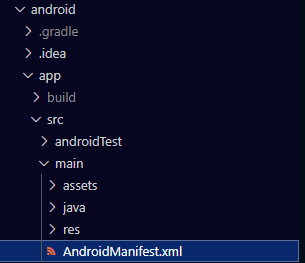
            </ion-item-options>

          </ion-item-sliding>

  </ion-header>

</ion-app>

**- Utilizar o Capacitor para controlo do dispositivo:**



Atraves do android manifest foi utilizado o seguinte comando para forçar na posição portrait:

            android:screenOrientation="portrait">

**- Manipular as CSS Custom Properties dos components;**

Para a cor da toolbar

**- Alterar/Personalizar formatações globais;**

Para a cor do background.

ion-content {

--background: linear-gradient(0deg, #ff8b1f, #ff1f1f);

}

**- Otimizar código com recurso a Services:**

Para criar o armazenamento interno no telemóvel foi preciso um CRUD (Create, Read, Update, and Delete) que no ionic se chama storage, para isso foi necessário de um service.

Exemplo:

  addPonto(*ponto*: Ponto) : *Promise*<*any*>{

    return this.storage.get(PONTO\_KEY).then((*pontos*: Ponto[]) *=>*{

      if(*pontos*){

*pontos*.push(*ponto*);

        return this.storage.set(PONTO\_KEY, *pontos*);

      }else{

        return this.storage.set(PONTO\_KEY, [*ponto*]);

      }

    });

  }

**- Criar cores disponíveis globalmente na app:**

Foram criadas duas cores globais:

  /\*\* orangeback \*\*/

  --ion-color-orangeback: #ff8b1f;

  /\*\* redback \*\*/

  --ion-color-redback: #ff1f1f;

Utilizadas apenas no scss:

ion-tab-bar {

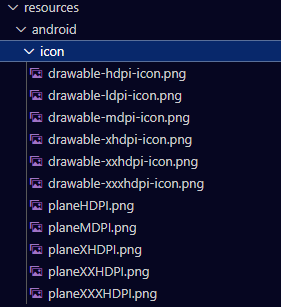
--background: rgb(32, 32, 32);

--background-activated: rgb(53, 53, 53);

--color:  var(--ion-color-orangeback);

. Alterar ícone da app e respetivo Splash Screen:

O icon da app foi alterado:



        <icon density="mdpi" src="resources/android/icon/planeMDPI.png" />

        <icon density="hdpi" src="resources/android/icon/planeHDPI.png" />

        <icon density="xhdpi" src="resources/android/icon/planeXHDPI.png" />

        <icon density="xxhdpi" src="resources/android/icon/planeXXHDPI.png" />

        <icon density="xxxhdpi" src="resources/android/icon/planeXXXHDPI.png" />

Ficheiro config.xml foram alterados os url.

**Conclusão:**

Com este trabalho verificamos a dificuldade de conceção e desenvolvimento próprio de uma app assim como todos os processos de proteção de erro, e a de estruturação de conceitos a desenvolver.

Referências Bibliográficas e Eletrónicas:

<https://ionicframework.com/docs>

<https://forum.ionicframework.com/>

<https://elearning.ipvc.pt/ipvc2020/pluginfile.php/113814/mod_label/intro/resumo001_ionic5_criarPrepararTestarDistribuir.txt?time=1618061984920>

<https://stackoverflow.com/>