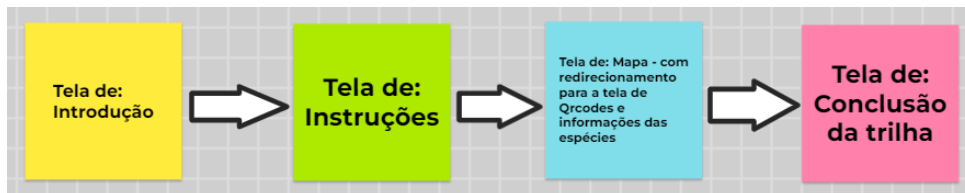


## RoadMap - Mapa Interativo (v1)

- **Versão do Flutter utilizada:** 2.8.1.
- **Requisitos para rodar o projeto:** Para rodar a branch "feature/mapa-tcc-brs" basta apenas ter a versão 2.8.1 instalada.

### Fluxo do recurso

O recurso do mapa conta com um fluxo:



### Panorama Geral De Cada Parte Do Fluxo

#### Tela de introdução:

É uma tela simples, apenas sinalizando o início do recurso. Contém widgets de imagem e texto, apenas.

#### Tela de instruções:

É uma tela pensada para serem colocadas ali as informações de uso da aplicação e de orientação acerca do passeio real na trilha (orientações para os visitantes).

Atualmente está tela encontra-se desenvolvida, mas sem os textos de apoio. **É necessário verificar com a Profª Larissa quais informações e orientações seria interessante conter ali.**

#### Tela do Mapa:

É a principal tela do recurso. A tela do mapa é uma combinação da renderização do mapa gerado no Tiled Map Editor (e lido pelo Flame) com itens que foram criados a partir dos próprios widgets do Flutter (Joystick e botões de placar).

O fluxo do mapa começa com o widget “GameWidget” (olhar o arquivo “lib/features/mapa/page/mapa\_main\_page.dart”) onde é definido o MapScreen do game (a classe Flame onde é implementado o cenário e o contexto do jogo) e os Overlay Builder Map, que são todos os elementos nativos do Flutter (como os widgets) que serão renderizados na tela do jogo, ou que em algum momento serão chamados a partir do jogo (como telas, pop-ups, botões e etc).

Observe que no Overlay Builder Map do GameWidget, foram definidos os widgets que ficam sobrepostos no cenário do jogo (Joystick e o placar) e também as páginas que em determinado momento da aplicação, de acordo com o fluxo do jogo e contexto do personagem, precisarão ser apresentadas, como é o caso da tela de QRCode, informações das espécies e a tela de conclusão da trilha. Observe que sempre será necessário definir no Overlay Builder Map do GameWidget as telas as quais a aplicação será redirecionada e os widgets nativos do Flutter a qual o jogo interage, de acordo com determinada ação (é uma forma de integrar os recursos nativos do Flutter ao contexto do Flame).

O MapScreen do jogo é definido pela classe MapScreen (olhar o arquivo “lib/features/mapa/page/mapa\_screen.dart”). Essa classe estende o Flame e contém os métodos onLoad, addActors e onJoystickDirectionChanged.

O método onLoad é um método genérico do Flame, imagine como se fosse o initState de uma classe comum do Flutter. Esse método carrega o cenário, adiciona os widgets de placar e joystick, e define o bako (personagem) como o foco central da câmera, para que ela siga esse ator de acordo com o seu deslocamento pelo cenário.

O método addActors é responsável por adicionar os atores do jogo (bako/player, plantas/espécies para serem descobertas e os pontos de colisão). Observe que cada um desses atores possui sua própria classe, com as suas propriedades e os seus métodos. Aqui no método addActors, estamos apenas adicionando-os ao cenário do jogo. Ainda, tenha em mente que cada um desses atores representa uma camada (layer) criada no Tiled Map Editor. Dessa forma adicionamos os tiles mapeados de cada camada a sua respectiva classe (é a forma de mapear de tile para objeto Flutter).

O método `onJoystickDirectionChanged` é um método que recebe um parâmetro do `GameWidget` referente a direção detectada pelo joystick, e passada a classe do Bako, para que o movimento seja realizado.

#### **Classes dos atores:**

- **Bako (olhar o arquivo “`lib/features/mapa/widgets/bako_widget.dart`”)**

A classe Bako é uma classe Flame do tipo `SpriteAnimationComponent`. Nela estão definidas as propriedades do ator bako e os métodos referente aos seus movimentos e as animações de movimento. Aqui também está definido o método que faz a detecção da colisão com objetos do tipo `CollisionPoints`, para impedir o seu deslocamento sobre essa área.

- **CollisionPoints (olhar o arquivo “`lib/features/mapa/widgets/CollisionPoints_widget.dart`”)**

A classe `CollisionPoints` é uma classe Flame do tipo `PositionComponent`. Nela estão definidas as propriedades do ator `CollisionPoints`.

- **Plants (olhar o arquivo “`lib/features/mapa/widgets/plants_widget.dart`”)**

A classe `Plants` é uma classe Flame do tipo `S PositionComponent`. Nela estão definidas as propriedades do ator `Plants` e o método que faz a detecção da colisão com o objeto do tipo Bako, para fazer o redirecionamento para a tela de leitura de `QRCode`.

#### **Tela de Qrcode:**

A leitura do `QRCode` foi implementada utilizando a biblioteca `Qr Code Scanner` na versão `^0.7.0`. A classe `QrcodeScannerPage` (observar o arquivo “`lib/features/qrcode/presentation/pages/qrcode/_scanner_page.dart`”) foi implementada seguindo o exemplo disponibilizado na documentação da lib (o link encontra-se na seção de materiais de apoio e referencias). Após a leitura do código, o método `onQRViewCreated` é acionado fazendo o redirecionamento para a página das informações das espécies.

#### **Tela de Informação das Espécies:**

É a tela responsável por apresentar as informações da espécie lida. É uma tela simples, com widgets de carrossel, imagem e texto (observar o arquivo "lib/features/informações\_especies/presentation/pages/informações\_especies\_page.dart"). Observe que essa classe utiliza recursos de controller (na aplicação está sendo utilizado o GetX) para realizar a busca das informações das espécies na base de dados (json). Aqui também se faz o uso de um model para as informações das espécies.

### **Tela de conclusão da trilha:**

É uma tela simples apenas com widgets de imagem e texto. A aplicação contém um contador a qual é incrementado toda vez que uma espécie é lida e após 21 espécies lidas é apresentada a tela de conclusão sinalizando ao usuário que ele descobriu as 21 espécies da trilha.

## **Pastas Importantes**

### **Assets:**

Nesta pasta, encontram-se as imagens utilizadas na aplicação (logos, ícones, backgrounds etc.) e uma pasta intitulada "tiles". A pasta tiles contém os arquivos tmx gerados no software Tiled Map Editor (ver seção X).

Nesta pasta foram mantidas todas as versões utilizadas na aplicação, sendo a de maior grau a que está rodando na aplicação (mapa-v4.tmx). Observe que para o cenário ser renderizado, o arquivo tmx que está sendo utilizado precisa estar contido no diretório assets.

Ainda, todas as imagens utilizadas dentro do arquivo tmx, criado no Tiled Map Editor também precisam estar presentes do diretório assets. Por isso é de extrema importância atentar-se na hora de excluir alguma imagem.

### **Tiled Map Editor:**

É o software utilizado para a criação e edição do mapa do jogo. Para a instalação e um guia de introdução básico a ferramenta, consultar a seção de materiais de apoio e referencias.

O mapa foi criado com as camadas de Camada de Tiles 1, que é o cenário do mapa, a camada CollisionPoints, que contém a demarcação da área de colisão, a camada Plants que contem a

demarcação das plantas com qrcode na trilha e a camada SpawnPoints, que é a demarcação do posição inicial do bako na trilha. Observe que todas as imagens que forem adicionadas ao projeto, precisam estar na pasta assets, pois o Flutter precisa reconhecer o diretório da imagem ao renderizar o mapa.

## **Materiais de apoio e referencias:**

Flame:

<https://pub.dev/packages/flame>

<https://docs.flame-engine.org/1.2.0/>

<https://medium.com/flutter-community/flutter-flame-step-1-create-your-game-b3b6ee387d77>

<https://blog.devgenius.io/lets-create-a-snake-game-using-flutter-and-flame-38482d3cf0ff>

<https://www.raywenderlich.com/27407121-building-games-in-flutter-with-flame-getting-started>

<https://www.raywenderlich.com/19430602-how-to-create-a-2d-snake-game-in-flutter>

<https://limezu.itch.io/serenevillagerevamped>

<https://youtube.com/playlist?list=PLiZZKL9HLmWPYd808sda2ydG-dhexNONV>

[https://youtube.com/playlist?list=PLxvyAnoL-vu6zw6\\_unCxidk2HE3UKgK1X](https://youtube.com/playlist?list=PLxvyAnoL-vu6zw6_unCxidk2HE3UKgK1X)

<https://youtube.com/playlist?list=PLed3DrfS7NfIAQC4a6aoesqobHsO1BXoT>

[https://www.youtube.com/watch?v=r5Kxs1nh5fw&list=PL07VZAaU7LCCUW1MeKfUCAqv3hAS\\_wUEu&index=9](https://www.youtube.com/watch?v=r5Kxs1nh5fw&list=PL07VZAaU7LCCUW1MeKfUCAqv3hAS_wUEu&index=9)

[https://www.youtube.com/watch?v=fjNx0otqF-k&list=PLiZZKL9HLmWPL0URlq9WLNg1A\\_g1LDuxx](https://www.youtube.com/watch?v=fjNx0otqF-k&list=PLiZZKL9HLmWPL0URlq9WLNg1A_g1LDuxx)

<https://www.youtube.com/watch?v=OJsYTv4ZPs8>

<https://www.youtube.com/watch?v=5KXM4UVVwAA>

<https://www.youtube.com/watch?v=NqjaLM0TWm0&t=1481s>

<https://www.youtube.com/watch?v=B4cbQra2ipE&t=4796s>

<https://www.youtube.com/watch?v=RaJixqwtCh0>

**Tiled Map Editor:**

<https://www.mapeditor.org/>

[https://www.youtube.com/watch?v=ZwaomOYGuYo&list=PLO7VZAaU7LCCUW1MeKfUCAqv3hAS\\_wUEu&index=10&t=1133s](https://www.youtube.com/watch?v=ZwaomOYGuYo&list=PLO7VZAaU7LCCUW1MeKfUCAqv3hAS_wUEu&index=10&t=1133s)

**QrCode Scanner:**

[https://pub.dev/packages/qr\\_code\\_scanner](https://pub.dev/packages/qr_code_scanner)

[https://www.youtube.com/watch?v=hHehIGfX\\_yU&list=PLO7VZAaU7LCCUW1MeKfUCAqv3hAS\\_wUEu&index=11](https://www.youtube.com/watch?v=hHehIGfX_yU&list=PLO7VZAaU7LCCUW1MeKfUCAqv3hAS_wUEu&index=11)