

EXPLORAHABITAT: UM PROJETO PARA AUXILIAR AS SAÍDAS A CAMPO DOS CLUBES DE CIÊNCIAS

Aluno: Matheus Soares Lima

Orientador: Dalton Solano dos Reis

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Especificação
- Implementação
- Análise dos Resultados
- Conclusões e Sugestões

Introdução

- Clube de ciências
 - Composto por mediadores e clubistas
 - Atividades para aprendizado científico
 - Trabalhos em equipes em projetos e estudos científicos
 - Atividades laboratoriais
 - Saídas a campo em acampamentos ou passeios científicos
 - Organização e implementação de campanhas
 - Organização de atividades culturais e recreativas
 - Organização e participação em atividades de divulgação, como feiras, conferências para clubistas e exposições
 - Atividades de colaboração com instituições comunitária

Introdução

- Aplicativo ExploraHabitat (KORBES; 2021)
 - Flutter
 - Auxiliar saídas a campo
 - Desenvolver autonomia e lado investigativo do clubista
 - Utiliza recursos dos dispositivos móveis para simular instrumentos de uso comum na realização das atividades de um Clubista
 - Falta de um design mais atrativo e amigável
 - Flutter pouco explorado
 - Utiliza Google Drive para armazenar as informações

Objetivos

- **Objetivo geral:** Estender o aplicativo ExploraHabitat (KORBES, 2021) proporcionando uma melhor experiência de uso com a modernização da interface e uma melhor arquitetura no desenvolvimento do aplicativo
- **Objetivos específicos:**
 - a) disponibilizar uma estrutura utilizando um gerenciador de estados e um injetor de dependências para atingir uma arquitetura consistente e fácil de ser utilizada
 - b) disponibilizar uma estrutura utilizando o site Back4App e o Parse Server do Flutter para autenticação e armazenamento
 - c) avaliar a usabilidade do aplicativo com os usuários

Fundamentação Teórica

Gerenciamento de estados

- Gerenciamento de estados
 - Comportamento dos componentes
 - MobX
- Injeção de dependências
 - Separar os componentes de ter dependências diretas
 - Provider
 - GetIt
- Back4App e Parse Server
 - Backend as a Service (BaaS)
 - Plataforma Parse
 - Gerenciamento de banco de dados, arquivos e usuários

Trabalhos Correlatos

- Bet et al. (2004)
 - Simular o ambiente do Clube de Ciências no meio virtual
 - Publicar pesquisas feitas nas escolas
 - Simular experimentos relacionados com as disciplinas que fazem parte do clube
 - Responder testes para avaliação do conhecimento dos usuários
- Rosa (2015)
 - Registro de incidentes ambientais
 - Apresentar os incidentes ambientais próximos em um mapa
 - Possibilitar que outros usuários apoiem e comentem nos incidentes
 - Permitir que uma entidade assuma um incidente e seja resolvido
- Mendonça et al. (2018)
 - Contribuir no processo de ensino e aprendizagem de botânica
 - Criar conteúdo de aprendizado associados às plantas
 - Avaliar o conhecimento do aluno ao procurar as plantas associadas ao conteúdo
 - Identificar a planta correta com base no GPS do aluno

Especificação

Especificação

Requisitos Funcionais

RF01 - permitir a criação de um novo usuário
RF02 - permitir a autenticação do usuário
RF03 - permitir o cadastro de um tema
RF04 - permitir o gerenciamento de um tema
RF05 - permitir o cadastro de objetivos específicos vinculados ao tema
RF06 - permitir o cadastro atividades vinculadas aos objetivos
RF07 - permitir a sincronização de um tema criado
RF08 - permitir a busca de um tema sincronizado
RF09 - permitir o envio de respostas para um tema sincronizado
RF10 - permitir a busca de respostas enviadas para um tema

Especificação

Requisitos Não Funcionais

RNF01 - utilizar o *framework* Flutter e a linguagem de programação Dart

RNF02 - utilizar o Material Design do Flutter para desenvolvimento das interfaces

RNF03 - utilizar a biblioteca MobX para gerenciamento de estados

RNF04 - utilizar as bibliotecas Provider e GetIt para injeção de dependências

RNF05 - utilizar a biblioteca Hive para armazenamento de dados na memória do dispositivo

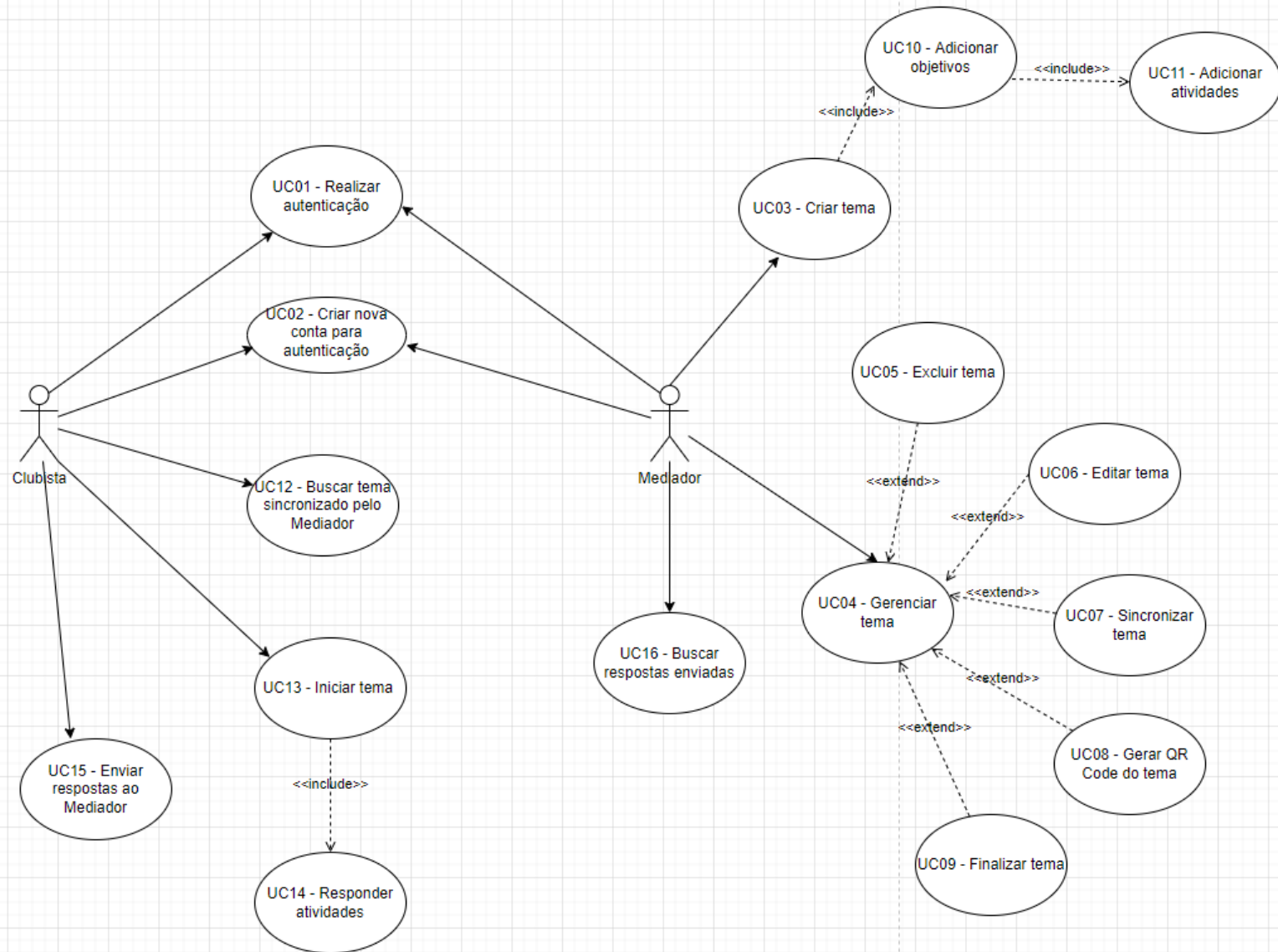
RNF06 - utilizar a biblioteca Parse Server para salvar os dados em um banco de dados no Back4App

RNF07 - deve ser desenvolvido no ambiente de programação Android Studio

RNF08 - deve permitir o funcionamento do aplicativo sem acesso à internet

Especificação

Casos de Uso



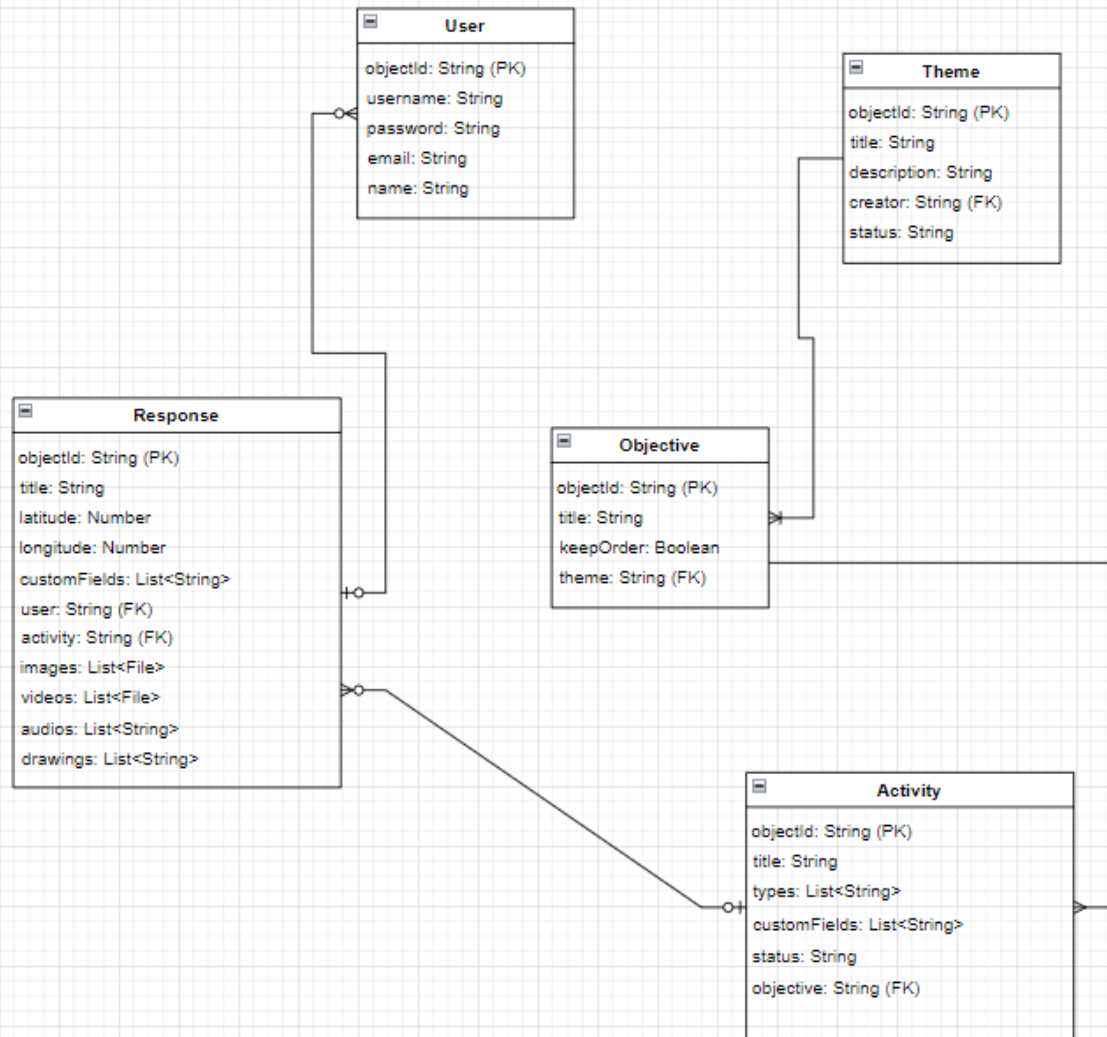
Especificação

Matriz de Rastreabilidade

Requisitos Funcionais (RF)	Casos de uso
possibilitar a autenticação do usuário	UC01
permitir a criação de um novo usuário	UC02
permitir o cadastro de um tema	UC03
permitir o cadastro de objetivos específicos vinculados ao tema	UC10
permitir o cadastro atividades vinculadas aos objetivos	UC11
permitir o gerenciamento de um tema	UC04, UC05, UC06, UC07, UC08, UC09
permitir a busca de um tema sincronizado	UC12
permitir o envio de respostas para um tema sincronizado	UC16, UC14
permitir a busca de respostas enviadas para um tema	UC16

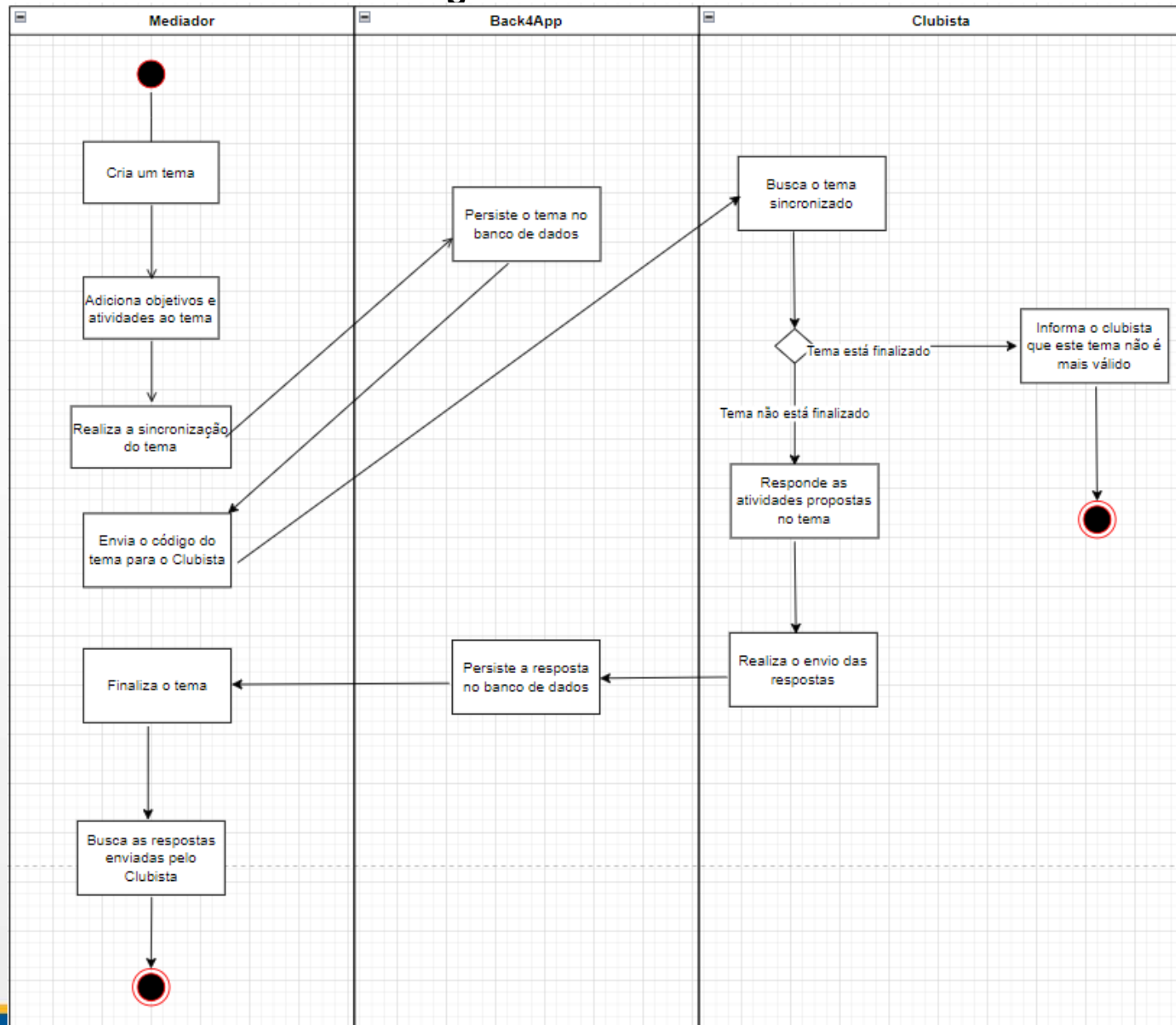
Especificação

Modelo Entidade Relacionamento



Especificação

Diagrama de Atividades

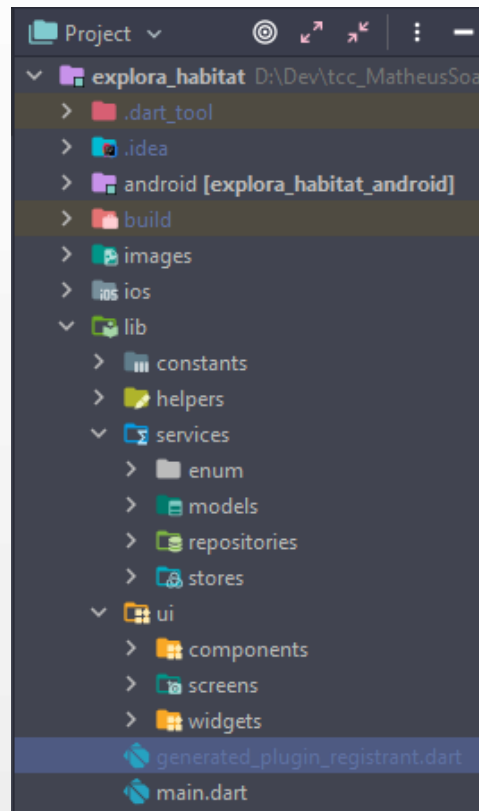


Implementação

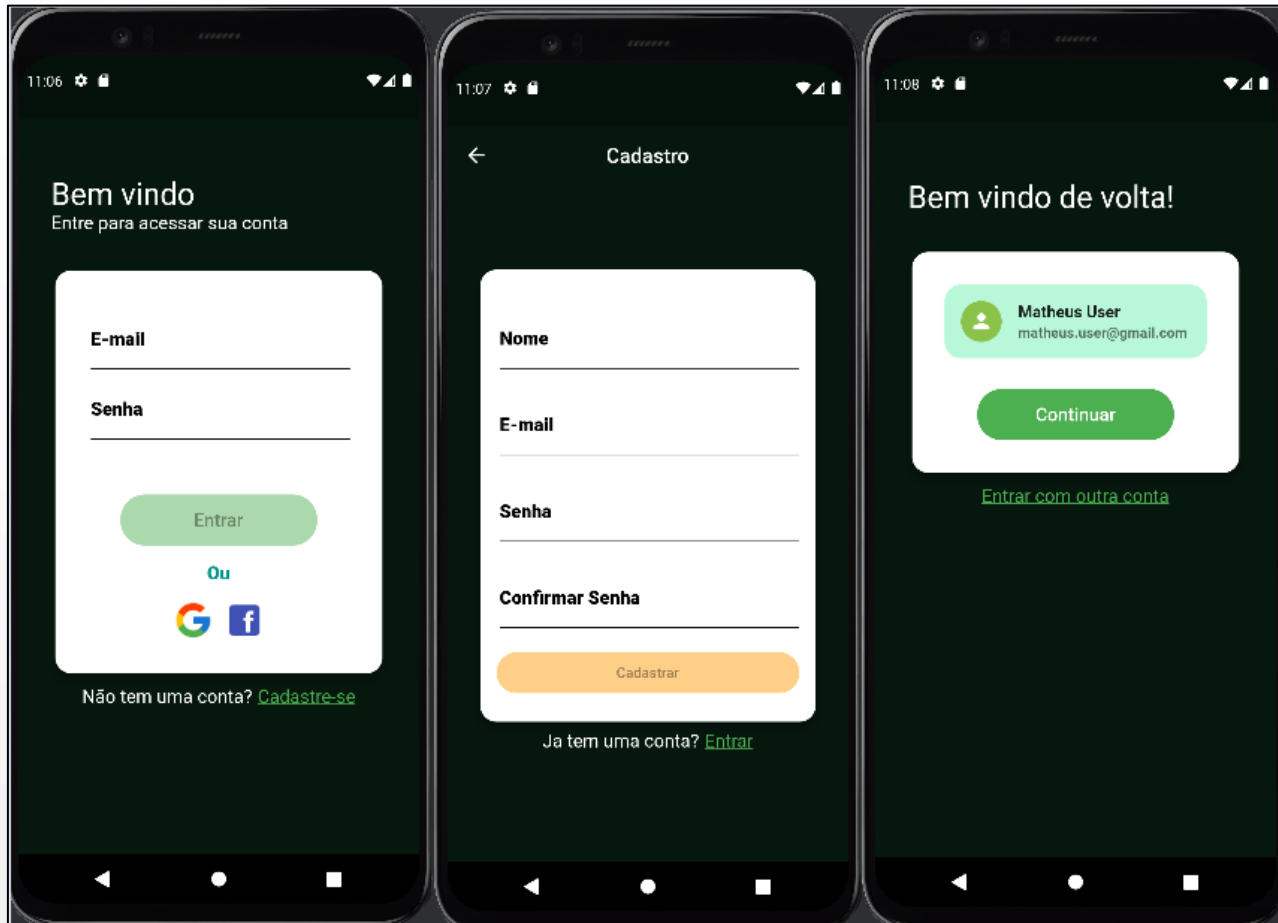
- Flutter
- MobX
- GetIt e Provider
- Hive
- Geolocator
- Google Maps
- Back4App e Parse Server

Implementação

- O projeto foi dividido em diretórios específicos para cada função no aplicativo



Implementação



Implementação

```
10 import 'package:mobx/mobx.dart';
11 import 'package:explora_habitat/helpers/extensions.dart';
12
13 part 'login_store.g.dart';
14
15 class LoginStore = _LoginStore with _$LoginStore;
16
17 abstract class _LoginStore with Store {
18   @observable
19   String? email;
20
21   @action
22   void setEmail(String value) => email = value;
23
24   @computed
25   bool get emailValid => email != null && email!.isEmailValid();
26
27   String? get emailError =>
28     email == null || emailValid ? null : 'E-mail inválido';
29
30   @observable
31   String? password;
32
33   @computed
34   bool get passwordValid => password != null && password!.length >= 4;
35
36   String? get passwordError =>
37     password == null || passwordValid ? null : 'Senha inválida';
38
39   @action
40   void setPassword(String value) => password = value;
41
42   @computed
43   VoidCallback? get loginPressed =>
44     emailValid && passwordValid && !loading ? _login : null;
45 }
```

```
Expanded(
  child: ListView.builder(
    itemCount: responseThemeStore.theme.objectives.length,
    physics: const BouncingScrollPhysics(),
    scrollDirection: Axis.vertical,
    itemBuilder: (_, index) => Provider(
      create: (_) => ResponseObjectiveStore(
        readOnly: responseThemeStore.readOnly,
        completed: responseThemeStore.completed,
        objective: responseThemeStore.theme.objectives[index],
      ), // ResponseObjectiveStore
      child: ResponseObjectiveComponent(),
    ), // Provider
  ), // ListView.builder
), // Expanded
```

```
Observer(
  builder: (_) => CustomFormField(
    enabled: !loginStore.loading,
    labelText: 'E-mail',
    errorText: loginStore.emailError,
    inputType: TextInputType.emailAddress,
    onChanged: loginStore.setEmail,
    counterText: "",
  ), // CustomFormField
), // Observer
```

Implementação

```
55     @action
56     Future<void> _login() async {
57         loading = true;
58         error = null;
59
60         try {
61             final user = await UserRepository().loginWithEmail(email!, password!);
62
63             final userMapped = User.fromParse(user);
64             GetIt.I<UserManagerStore>().setUser(userMapped);
65
66             _verifyRegistrations();
67
68             GetIt.I.registerSingleton(ThemeStore());
69             await GetIt.I<ThemeStore>().initThemesBox(userMapped.id!);
70
71             GetIt.I.registerSingleton(SyncedThemesStore());
72             await GetIt.I<SyncedThemesStore>().initThemesBox(userMapped.id!);
73
74             GetIt.I.registerSingleton(ResponsesThemeStore());
75             await GetIt.I<ResponsesThemeStore>().initThemesBox(userMapped.id!);
76
77             loggedIn = true;
78         } catch (e) {
79             error = e.toString();
80         }
81         loading = false;
82     }
```

Implementação

```
1 import 'package:explora_habitat/services/models/objective.dart';
2 import 'package:explora_habitat/services/models/user.dart';
3 import 'package:hive/hive.dart';
4
5 import '../enum/theme_status.dart';
6
7 part 'theme_explora.g.dart';
8
9 @HiveType(typeId: 0)
10 class ThemeExplora extends HiveObject {
11   @HiveField(0)
12   String? id;
13
14   @HiveField(1)
15   User? creator;
16
17   @HiveField(2)
18   String title;
19
20   @HiveField(3)
21   String description;
22
23   @HiveField(4)
24   List<Objective> objectives = [];
25
26   @HiveField(5)
27   ThemeStatus status;
28
29   @HiveField(6, defaultValue: false)
30   bool isResponsesSynced;
31
32   @HiveField(7, defaultValue: false)
33   bool isResponsesPending;
```

```
25 Future<void> initThemesBox(String userId) async {
26   await Permission.storage.request().isGranted;
27   final Directory result = await getApplicationSupportDirectory();
28   Hive.init(result.path);
29   myThemesBox = await Hive.openBox<ThemeExplora>('themes_explora.$userId');
30 }
```

Análise dos Resultados

Testes com Usuários

- 14 usuários
- Plataforma Google Docs
- Disponibilizado link para instalação
- Perfil dos alunos:
 - 18 a 27 anos
 - Apenas 2 conheciam ou já participaram de um clube de ciências
 - Apenas 5 já utilizaram dispositivos móveis em uma saída a campo para coletar dados

Análise dos Resultados

Roteiro Mediador

Ordem	Etapas
1	Abra o aplicativo ExploraHabitat, crie um usuário usando o mesmo e-mail anteriormente informado e tente seguir os passos descritos abaixo.
2	Acessar a tela "Criar Tema" clicando no ícone de navegação no canto superior esquerdo. Definir um título e uma descrição para o tema e confirmar.
3	Para o tema criado anteriormente, adicionar objetivos e para cada objetivo adicionar ao menos uma atividade. Na atividade adicionar o título, os recursos que serão utilizados e as perguntas que serão respondidas definindo o tipo da resposta esperada.
4	Verificar se é possível modificar o tema, os objetivos e as atividades. Também é possível reordenar as atividades segurando a atividade e movendo para a posição desejada. Após realizar as alterações, salvar o tema clicando no ícone superior direito.
5	Acessar a tela "Meus Temas" e localizar o tema criado. Expandir a flecha e verificar as opções apresentadas. Será possível editar, excluir, sincronizar ou copiar o tema.
6	Realizar a sincronização de um tema, verificar se o código de compartilhamento do tema está sendo exibido na parte inferior do card.
7	Após a sincronização, expandir a flecha e verificar as opções apresentadas. Será possível visualizar, copiar, exibir o QR Code, finalizar ou fechar o tema.

Análise dos Resultados

Roteiro Clubista

Ordem	Etapas
1	Abra o aplicativo ExploraHabitat, crie um usuário ou utilize o mesmo usuário do guia de Mediador e tente seguir os passos descritos abaixo.
2	Acessar a tela "Temas" e clicar no botão na parte inferior direita da tela. Inserir o código ou ler o QR Code do tema criado anteriormente. Aguardar até que o tema seja sincronizado.
3	Após a sincronização, expandir a flecha e verificar as opções apresentadas. Será possível executar e fechar o tema.
4	Escolher um tema e clicar na opção de executar. Será direcionado para a tela de "Resposta", na qual possui duas abas, sendo a primeira a listagem de objetivos e suas atividades e a segunda o mapa da localização atual.
5	Responder todas as atividades de todos os objetivos definidos. Acessar a segunda aba do mapa e verificar se foi adicionado o marcador de localização para as atividades respondidas.
6	Após responder, salvar as respostas clicando no ícone no canto superior direito. Na tela "Temas", realize a sincronização das respostas. As respostas poderão ser visualizadas pelo Mediador, na tela de "Respostas" assim que o tema for finalizado.

Análise dos Resultados

Pesquisa

- Em geral o aplicativo foi avaliado com boa usabilidade, navegabilidade e as interfaces foram consideradas amigáveis
- Cumpriu o objetivo em auxiliar atividades de saídas a campo
- Recomendariam para alguém que deseja realizar pesquisas em saídas a campo
- Dificuldades em concluir os passos sem auxílio externo
 - 9 usuários precisaram de três ou mais auxílios

Análise dos Resultados

Avaliação dos usuários

- Alguns usuários apontaram falha ao sincronizar e visualizar as respostas
- Enunciado de perguntas longas não apareceram na tela
- “A interface está muito boa e intuitiva, respostas rápidas e sem muito tempo de demora, no geral é um aplicativo que para atividades de campo será muito útil para mapeamento de atividades e facilidade para acompanhar cada etapa da atividade”
- “Muito bem feito o aplicativo, tudo muito funcional e fácil de utilizar”

Análise dos Resultados

Comparação correlatos

	Bet <i>et al.</i> (2004)	Mendonça <i>et al.</i> (2018)	Rosa (2015)	ExploraHabitat
Propósito educacional	Sim	Sim	Não	Sim
Saídas a campo	Não	Sim	Sim	Sim
Produção de conteúdo avaliativo	Sim	Sim	Não	Sim
Compartilhar informações entre os usuários	Sim	Sim	Sim	Sim
Abordagem	Aprendizado em clube de ciências	Aprendizado em botânica	Incidentes ambientais	Aprendizado em clube de ciências
Plataforma	Web	Android e iOS	Android, Windows Phone e iOS	Android e iOS, mas validado apenas no Android
Utilização da geolocalização do usuário	Não	Sim	Sim	Sim
Utilização de mapa	Não	Não	Sim	Sim

Conclusões

- O aplicativo atendeu o propósito em proporcionar uma melhor experiência de uso
 - Boa usabilidade e interfaces amigáveis
 - Permite criar e executar vários temas simultaneamente
 - Customização completa das atividades
- Dificuldades em entender o fluxo do aplicativo e problemas na visualização e sincronização das respostas
- Flutter
 - Mobx: Facilitou o controle do comportamento dos componentes
 - Provider e GetIt: Melhor organização do código e melhor definição do que pode ser acessado entre os componentes
 - Flutter Material Design: Componentes visualmente agradáveis e padronização com a temática do aplicativo
 - Hive: Eficaz em armazenar os dados localmente
- Back4App e Parse Server
 - Criação fácil de um backend
 - Gerenciamento de banco de dados, arquivos e usuários
 - Funcionalidades não exploradas

Conclusões

- Limitações
 - Back4App plano gratuito com limitações mensais
 - Limite de requisições: 25.000
 - Banco de dados: 250mb
 - Armazenamento de arquivos: 1gb
 - Limite de upload 25mb por arquivo
 - Necessidade constante de persistir no Back4app e armazenar os dados na memória local

Sugestões

- a) validar a utilização no sistema operacional iOS;
- b) disponibilizar nas lojas de aplicativos para maior acessibilidade;
- c) adicionar mais recursos do dispositivo móvel nas atividades do tema;
- d) avaliar e otimizar a sincronização dos temas e das respostas;
- e) melhorar o mapa para exibir mais detalhes da atividade realizada;
- f) melhorar o mapa para que ao clicar em um marcador possua a opção de navegar até a atividade e vice-versa;
- g) permitir um melhor gerenciamento do controle da conta do usuário;
- h) permitir que o mediador possa avaliar as atividades realizadas;
- i) permitir a criação de *templates* prontos para as atividades;
- j) criar uma interface web para a criação e compartilhamento dos *templates*, fomentando a interação na comunidade dos clubes de ciências;
- k) adicionar novas formas de autenticação do usuário;
- l) explorar os conceitos da aprendizagem ubíqua no aplicativo.

Obrigado!

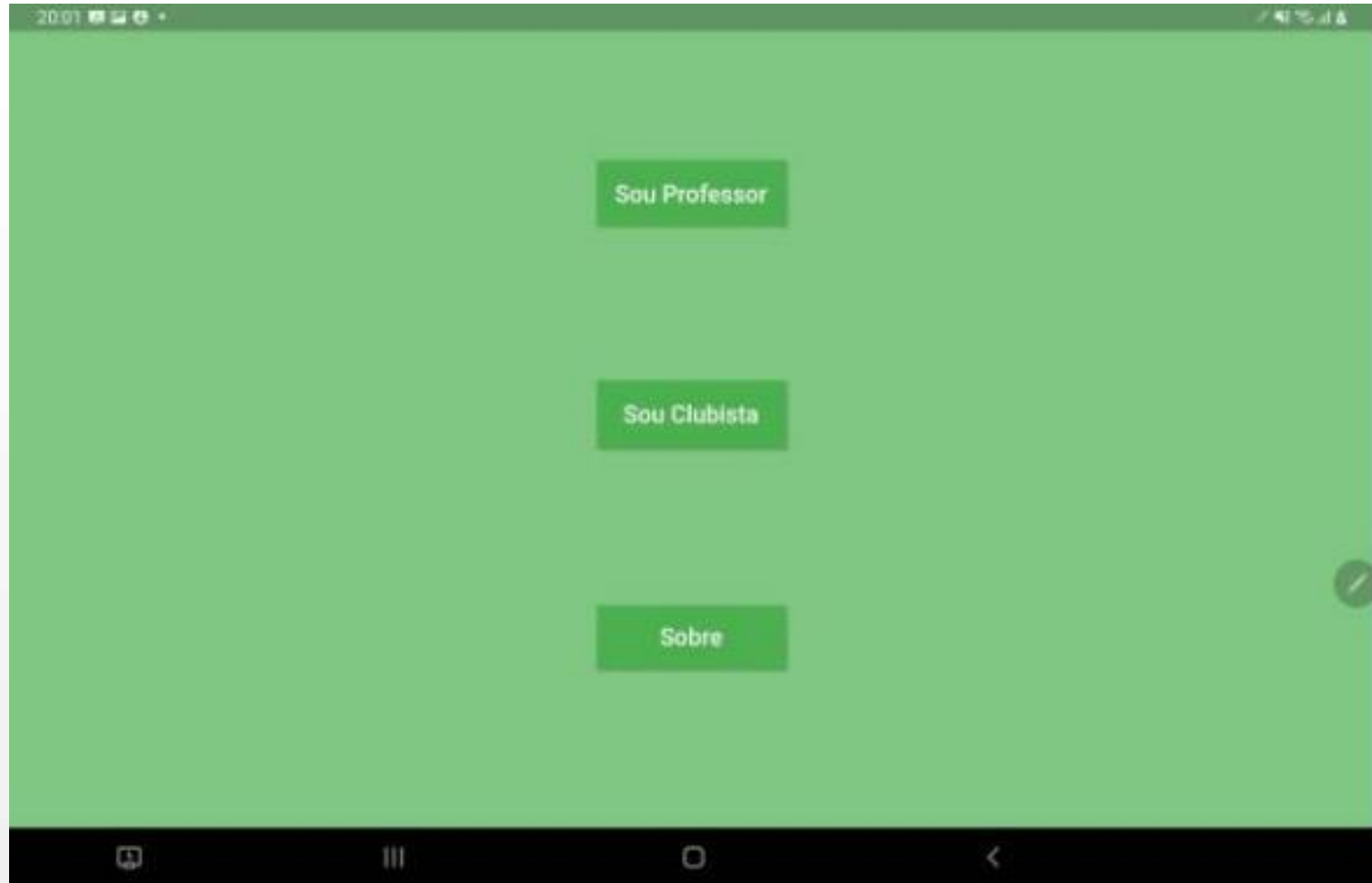
Apresentação Prática



qrfy.mobi/p/UiquB8Q

Apresentação Prática

Comparação



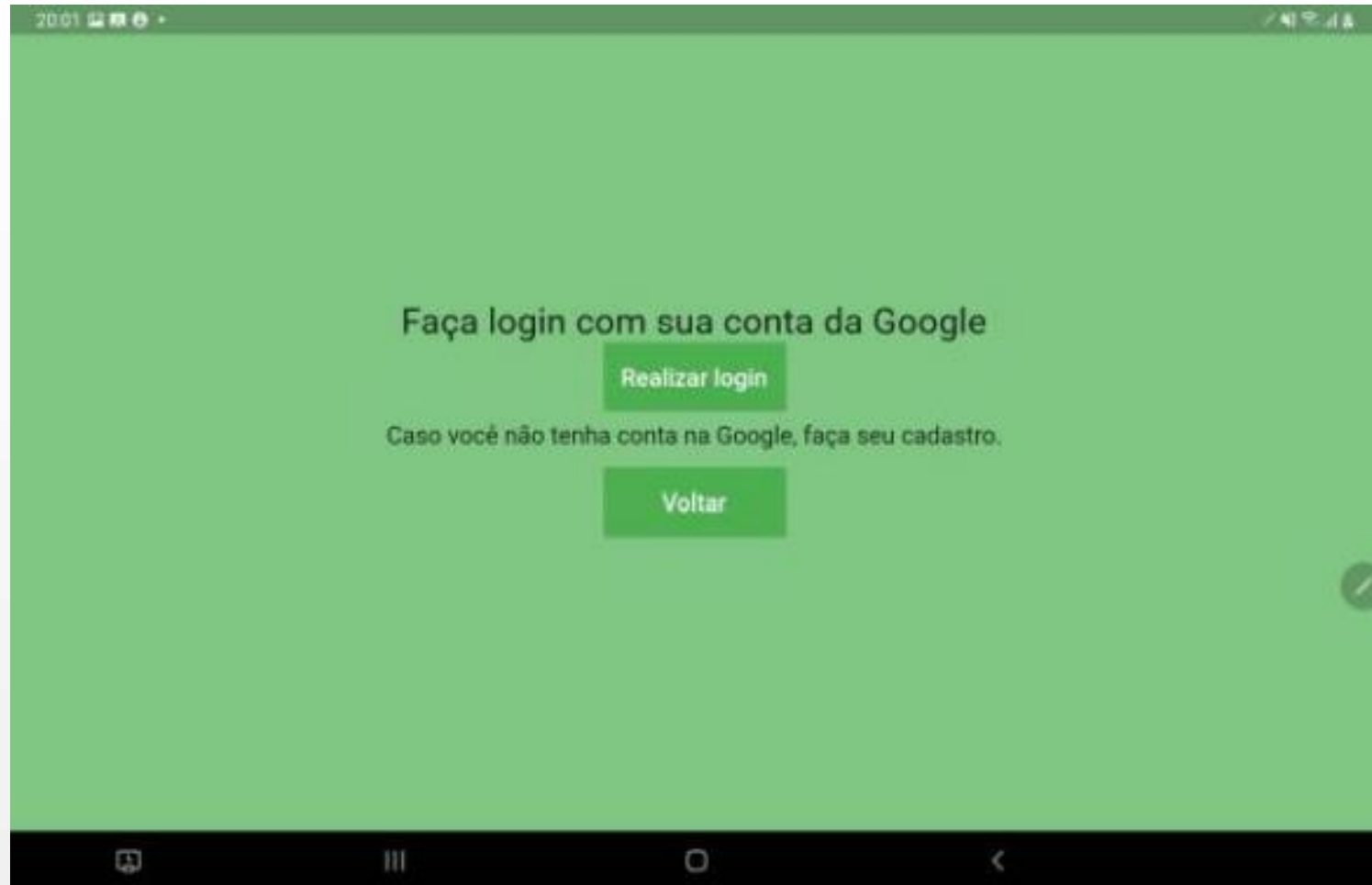
Apresentação Prática

Comparação



Apresentação Prática

Comparação



Apresentação Prática

Comparação



Apresentação Prática

Comparação

The screenshot shows a mobile application interface with a green background. At the top, there is a status bar with the time 20:01 and various icons. Below the status bar, there is a text input field labeled "Cadastrar Tema*". Underneath this field is a larger text input area labeled "Objetivo geral da atividade de campo*". At the bottom of the screen, there are two green buttons with white text. The left button is labeled "Cadastrar Objetivos Específicos para o Tema (opcional)" and the right button is labeled "Finalizar Tema e gerar QRCode". The bottom of the screen features a black navigation bar with standard Android icons: a home button, a recent apps button, a back button, and a search button.

Apresentação Prática

Comparação

The screenshot shows a mobile application interface with a green theme. At the top, there is a header bar with the text "Cadastrar Objetivo" on the left and "Cadastrar Objetivo" on the right. Below the header, there is a list of six items, each with a small icon of a document with a checkmark, a number (1 through 6), and two buttons: "Cadastrar Roteiro" and a trash icon. At the bottom right of the main content area, there is a button labeled "Finalizar Cadastro de Objetivos". The bottom of the screen shows a standard Android navigation bar with icons for back, home, and recent apps.

Item	Icon	Number	Action 1	Action 2
1	Document with checkmark	1	Cadastrar Roteiro	Trash
2	Document with checkmark	2	Cadastrar Roteiro	Trash
3	Document with checkmark	3	Cadastrar Roteiro	Trash
4	Document with checkmark	4	Cadastrar Roteiro	Trash
5	Document with checkmark	5	Cadastrar Roteiro	Trash
6	Document with checkmark	6	Cadastrar Roteiro	Trash

Apresentação Prática

Comparação

18:14

Objetivo: Subir a montanha

					#1 - Foto - Tire uma foto da paisagem
					#3 - Plantar - plante uma rosa
					#2 - Lixo - Tire uma foto do lixo
Video	Característica	Lupa	Vivência	Mosquito	
Áudio	Teste	Ficha Coleta	Lixo	Sons da Natureza	
Localização	Produção de Material	Outra intervenção	Plantar	Personalizada	

☐ Este roteiro deve ser realizado na ordem proposta

Finalizar atividade

Apresentação Prática

Comparação

Cadastrar Atividade*

Foto

Objetivo geral da atividade de campo*

0/150

Cancelar atividade

Cadastrar atividade

Apresentação Prática

Comparação

17:51

Medidas observadas:

Dimensão*	Un. Medida*	Qual o valor obtido?*

Essas medidas resultaram em algum cálculo?

Cancelar **Gravar**

Apresentação Prática

Comparação

17:51

Observe o solo:

Qual a coloração do solo?*	Há matéria orgânica? Quais?*
Qual a granulometria?*	Elementos químicos presentes no solo*

Cancelar **Gravar**

17:51