

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

João Pedro da Silva, José Carlos Pereira – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil
matheusl@furb.br, dalton@furb.br

Resumo: O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

Palavras-chave: Ciência da computação. Sistemas de informação. Monografia. Resumo. Formato.

1 INTRODUÇÃO

Segundo Freitas e Santos (2021), um Clube de Ciências é uma subcategoria de um Clube Escolar, que possui o objetivo de reunir um grupo de pessoas para promover discussões e momentos de lazer sobre diversos temas onde há um interesse mútuo. Um Clube Escolar se diferencia de outros clubes justamente pelo seu objetivo educacional entre professores e alunos. O Clube de Ciências se segmenta dos Clubes Escolares na especialização na comunicação da ciência entre os participantes do clube.

O Clube de Ciências é composto por professores que são os mediadores do conhecimento e estudantes comumente chamados de clubistas evitando serem referidos como alunos, pois de acordo com Freitas e Santos

Consideramos que no Clube de Ciências o termo aluno seria inadequado, pois nesse espaço espera-se que os participantes sejam ativos, protagonistas e que suas vozes sejam consideradas nas decisões. (FREITAS; SANTOS; 2021, p. 24)

Dentro deste contexto, para o aprendizado científico os clubistas são expostos a uma grande gama de atividades em diversas áreas, sendo que o mediador individualmente ou através de um consenso comum entre todos os clubistas definirá a estratégia mais adequada ao objetivo pretendido (FREITAS; SANTOS; 2021, p. 28). De acordo com Córdoba (2012, p. 3), as atividades podem ser 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 e até atividades de colaboração com instituições comunitárias.

Korbes (2021) desenvolveu o aplicativo ExploraHabitat, que tem como objetivo auxiliar as saídas a campo dos Clubes de Ciências, permitindo que o Clubista através da criação de atividades propostas por um mediador, desperte um lado mais investigativo, possibilitando uma maior interação com a natureza e desenvolva sua própria autonomia. O aplicativo utiliza recursos dos dispositivos móveis como o GPS (Global Positioning System), câmera, vídeo e áudio para simular instrumentos de uso comum na realização das atividades de um Clubista. Korbes (2021) afirma que, mesmo o ExploraHabitat cumprindo o esperado, existem problemas que dificultam a utilização do aplicativo, como por exemplo, a falta de um design mais atrativo e amigável, interfaces responsivas que se adaptam de acordo com o dispositivo, entre outros. Porém outro ponto notado, a falta de exploração do *framework* Flutter, no qual não foi adotado nenhum padrão de desenvolvimento e a utilização de um gerenciador de estados, como o MobX do Flutter por exemplo, o que auxilia na realização de melhorias na interface e na inclusão de novas funcionalidades facilitando a extensão em projetos futuros.

Desta forma, o objetivo deste trabalho é 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. Os objetivos específicos são: disponibilizar uma estrutura utilizando um gerenciador de estados e um injetor de dependências para atingir uma arquitetura consistente e fácil de ser utilizada, disponibilizar uma estrutura utilizando o site Back4App e o Parse Server do Flutter para autenticação e armazenamento de dados, avaliar a usabilidade do aplicativo com os usuários comparando a sua eficiência em relação ao aplicativo atual

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CONCEITOS, TÉCNICAS E/OU FERRAMENTAS

2.1.1 Gerenciamento de estados – MobX

2.1.2 Injeção de dependências – Provider e GetIt

2.1.3 Persistência – Back4App com Parse Server e Hive

2.2 VERSÃO ANTERIOR DO APLICATIVO

Korbes (2021) desenvolveu o aplicativo ExploraHabitat para as plataformas Android e iOS. Foi desenvolvido com a linguagem Dart e o *framework* Flutter, utilizando o UI Toolkit para a interface. O aplicativo tem como principal objetivo apoiar atividades de saída a campo em Clubes de Ciências, através da utilização de recursos dos dispositivos móveis como câmera, vídeo, áudio e GPS para a realização das atividades dos Clubistas, facilitar a coleta de dados, compartilhar informações com outros usuários e a própria construção científica deles.

A utilização do aplicativo ExploraHabitat inicia com a escolha de perfil do usuário, onde ele escolhe entre Professor ou Clubista, após isso realiza a autenticação através de uma conta Google. Uma vez autenticado, o usuário poderá realizar o cadastro de um tema referente ao estudo em campo que será realizado. Dentro de um tema cadastrado poderá ser incluso objetivos e as atividades que irão compor o roteiro que será executado pelos Clubistas. O aplicativo permite que para cada objetivo possa ser criado um roteiro de atividades. As atividades são variadas, desde a tirar uma foto, realizar a gravação de um áudio ou vídeo, por exemplo. Os cadastros normalmente deverão ser realizados pelo professor coletivamente com os Clubistas, para que assim tenha uma mobilização entre todos os usuários. Por fim, devido o aplicativo ter sido planejado para evitar a utilização de internet, pois em saídas a campo os clubistas podem estar em lugares onde não possui acesso a redes móveis ou wi-fi, desta forma o aplicativo permite a integração com outros usuários através da geração de um QR Code. Ele é gerado quando professor finaliza o cadastro de um tema, sendo assim, o clubista poderá realizar a leitura e obter todos os dados relacionado ao tema, incluindo o roteiro com as atividades para conclusão dos objetivos.

Na realização das atividades, o Clubista preenche as informações solicitadas, de acordo com a definição previamente realizada no cadastro do tema e realiza a finalização da tarefa. Após finalizar todas as atividades propostas no roteiro, o Clubista envia as respostas para o professor. As atividades enviadas serão armazenadas em uma estrutura de pastas no Google Drive com a criação de um *folder*, permitindo que os dados possam ser acessados posteriormente.

Korbes (2021) apresenta resultados de um questionário sobre a usabilidade do aplicativo. Com os gráficos não foi possível identificar as principais dificuldades dos usuários, porém é possível identificar alguns problemas de alteração de informações de temas existentes e no envio de tema para o professor. Em geral o aplicativo foi bem aceito e considerado autoexplicativo. Korbes (2021) relata que a maioria das ferramentas utilizadas foram efetivas na sua proposta, entretanto entende que o Flutter poderia ser mais bem estudado, principalmente em questões de redimensionamento de telas, designs mais atrativos, limites e posições de campo. Afirma também que uma limitação do aplicativo foi justamente no âmbito de desenhos em tela, que não cumpriu conforme o esperado. Korbes (2021) propôs sugestões de extensões futuras como a implementação de novas atividades, mais autonomia para o Clubista alterar e remover atividades, otimização no armazenamento de dados, melhorar a responsividade da tela, acessibilidade em outras línguas, inserir recursos do dispositivo como acelerômetro e bússola entre outros.

Korbes (2021) concluiu que o trabalho é relevante para o estudo acadêmico em Clubes de Ciências, pois auxilia na automatização do processo de aplicação de perguntas, incentivando os Clubistas a trabalharem com mais autonomia. Também permite que possam dedicar mais tempo em outras atividades de ensino, ter uma maior interação com a natureza a partir do dispositivo móvel e na flexibilidade do professor ou o próprio Clubista proporem os roteiros das atividades.

2.3 TRABALHOS CORRELATOS

Nesta seção apresenta os trabalhos que possuem características semelhantes ao aplicativo que está sendo proposto. O primeiro trabalho é o Clube Virtual de Ciências (BET *et al.*; 2004) que busca trazer o ambiente do Clube de Ciências para o mundo virtual, apresentado no Quadro 1. O segundo se trata da proposta de criação um ambiente *u-learning* (MENDONÇA *et al.*; 2018) para contribuir no ensino e aprendizagem de botânica, apresentado no Quadro 2. O terceiro é referente ao Ambicare (ROSA, 2015) um aplicativo desenvolvido com o objetivo de auxiliar a solução de incidentes ambientais, apresentado no Quadro 3.

Quadro 1 – Trabalho Correlato Clube Virtual de Ciências: um ambiente colaborativo para auxílio ao aprendizado

Referência	Bet <i>et al.</i> (2004)
Objetivos	Simular o ambiente do Clube de Ciências vivenciado em sala de aula para a internet, permitindo a interação de professores e alunos sem a necessidade da presença física

Principais funcionalidades	Possibilitar a divulgação de pesquisas feitas nas escolas Simular experimentos relacionados com as disciplinas que fazem parte do clube Responder testes para avaliação do conhecimento dos usuários
Ferramentas de desenvolvimento	Não foi possível identificar as ferramentas, pois não foi especificado pelo autor
Resultados e conclusões	Bet <i>et al.</i> (2004) concluiu que a utilização do computador nos estudos como um ótimo instrumento para o ensino, principalmente pelos recursos audiovisuais e a interatividade permitida entre os usuários, ainda mais com a internet facilitando a comunicação e sendo utilizada como fonte de pesquisa.

Fonte: elaborado pelo autor.

Quadro 2 – Trabalho Correlato Ambiente de aprendizagem ubíqua para auxiliar o estudo de botânica em atividades de aula de campo

Referência	Mendonça <i>et al.</i> (2018)
Objetivos	Contribuir no processo de ensino e aprendizagem de botânica, auxiliando na execução das aulas de campo através de um ambiente de aprendizagem ubíqua na utilização de alguns instrumentos dos dispositivos moveis
Principais funcionalidades	Registrar a localização de plantas utilizando o GPS do celular 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 Apresentar dicas caso o aluno informe a planta incorreta
Ferramentas de desenvolvimento	Framework Ionic e servidor web em PHP
Resultados e conclusões	Mendonça <i>et al.</i> (2018), explica que como o trabalho está em fase de desenvolvimento apenas alguns resultados foram obtidos na fase inicial do projeto, como o levantamento e definição de requisitos, a caracterização do cenário de aplicação e a arquitetura do ambiente. Para extensão, pretende-se concluir a etapa de desenvolvimento e aplicar novos testes funcionais e de validação para verificar o comportamento em um ambiente real, para assim, constar se os objetivos da pesquisa foram de fato alcançados.

Fonte: elaborado pelo autor.

Quadro 3 – Trabalho Correlato Ambicare: monitoramento ambiental usando dispositivos móveis

Referência	Rosa (2015)
Objetivos	Disponibilizar recursos para auxiliar órgãos responsáveis pela monitoração e elaboração de planos de contingência para incidentes relacionados ao meio ambiente
Principais funcionalidades	Registrar a ocorrência 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
Ferramentas de desenvolvimento	Framework Phonegap para as plataformas Android, iOS e Windows Phone
Resultados e conclusões	Rosa (2015) concluiu que é necessário um breve período de utilização após o lançamento efetivo do aplicativo, para que assim, seja verificado a efetividade da aplicação, mas que é possível assumir que poderá contar com o apoio popular e contribuir para o auxílio na solução de incidentes ambientais.

Fonte: elaborado pelo autor.

3 DESCRIÇÃO DO APLICATIVO

Nesta seção são apresentadas as etapas de especificação e desenvolvimento mais relevantes para o entendimento sobre o aplicativo desenvolvido.

3.1 ESPECIFICAÇÃO

O aplicativo permite a criação temas para serem usados em saídas a campo para auxílio em pesquisas e coleta de dados. Os temas são utilizados como forma de abordagem à um determinado assunto de maneira mais abrangente e genérica. Dentro de um tema são criados os objetivos que afunilam a ideia inicial em uma parte mais específica do assunto abordado. Para cada objetivo deve ser definido ao menos uma atividade que será um conjunto de ações que devem ser realizadas para atingir o propósito do objetivo.

Os Requisitos Funcionais (RF) do aplicativo estão apresentados no Quadro 4. Os Requisitos Não Funcionais (RNF) estão apresentados no Quadro 5:

Quadro 2 – 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

Fonte: elaborado pelo autor.

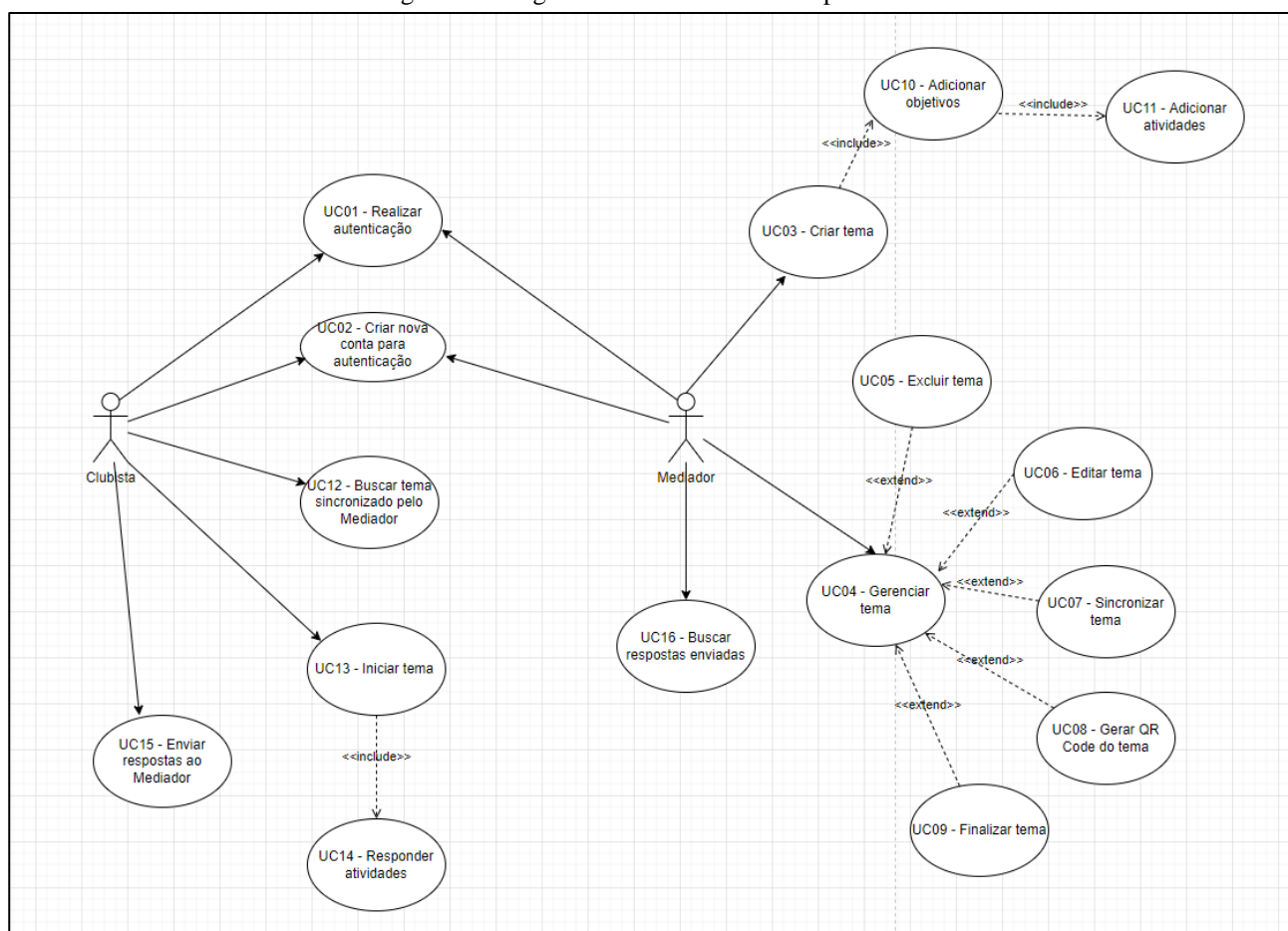
Quadro 5 – Requisito não funcionais

RNF01 – utilizar o <i>framework</i> Flutter e a linguagem de programação Dart
RNF02 - utilizar o Material Design do Flutter para de 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 Code
RNF08 – deve permitir o funcionamento do aplicativo sem acesso à internet (RNF)

Fonte: elaborado pelo autor.

Para melhor compreensão das funcionalidades do aplicativo a Figura 1 apresenta o diagrama de casos de uso dos atores Clubista e Mediador.

Figura 1 – Diagrama de casos de uso do aplicativo



Fonte: elaborado pelo autor.

No Quadro 6 é apresentado a matriz de rastreabilidade entre os requisitos funcionais no Quadro 4 e os casos de uso da Figura 1.

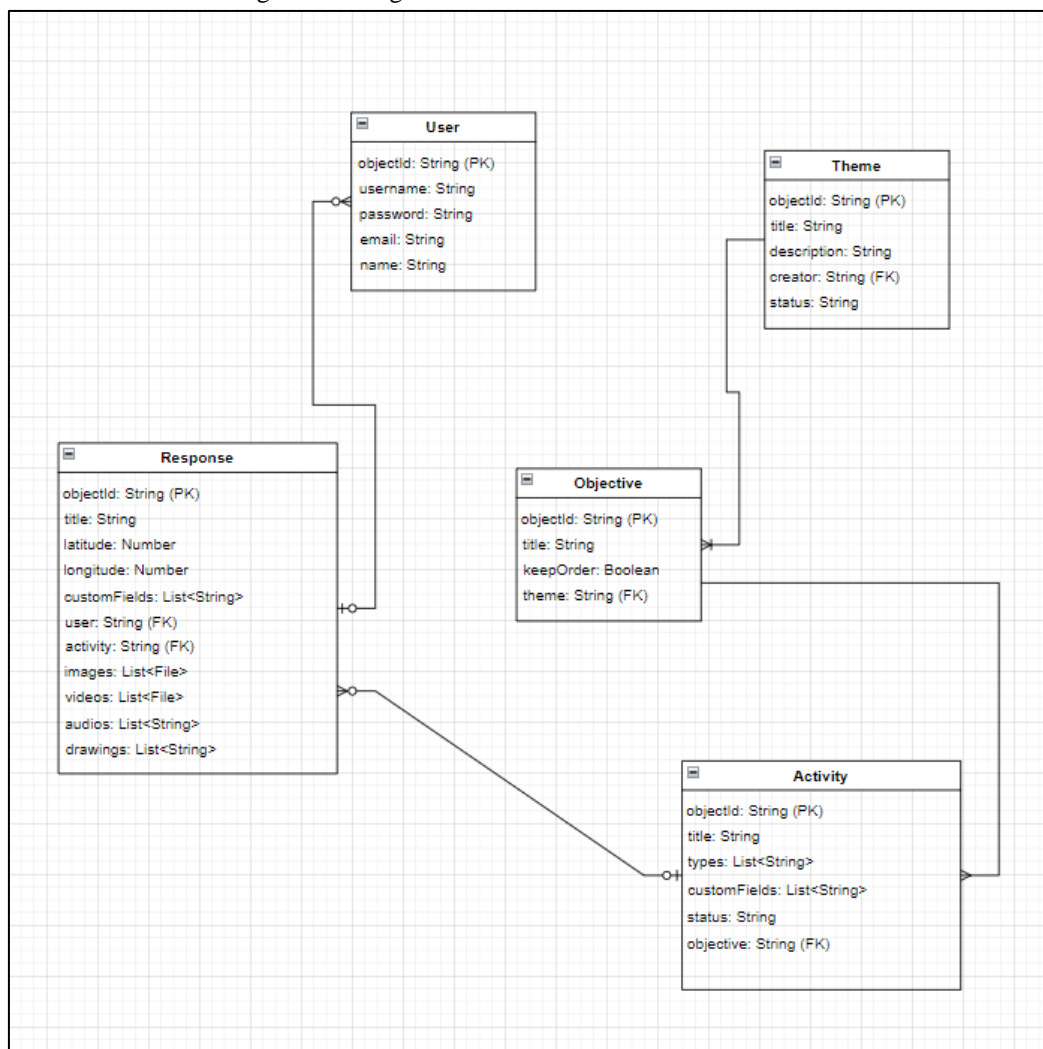
Quadro 6 – Matriz de rastreabilidade

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

Fonte: elaborado pelo autor.

Para armazenar e gerenciar as informações foi utilizado o banco de dados hospedado pelo Back4App vinculado à aplicação. A Figura 2 apresenta a modelagem utilizada na estruturação das tabelas.

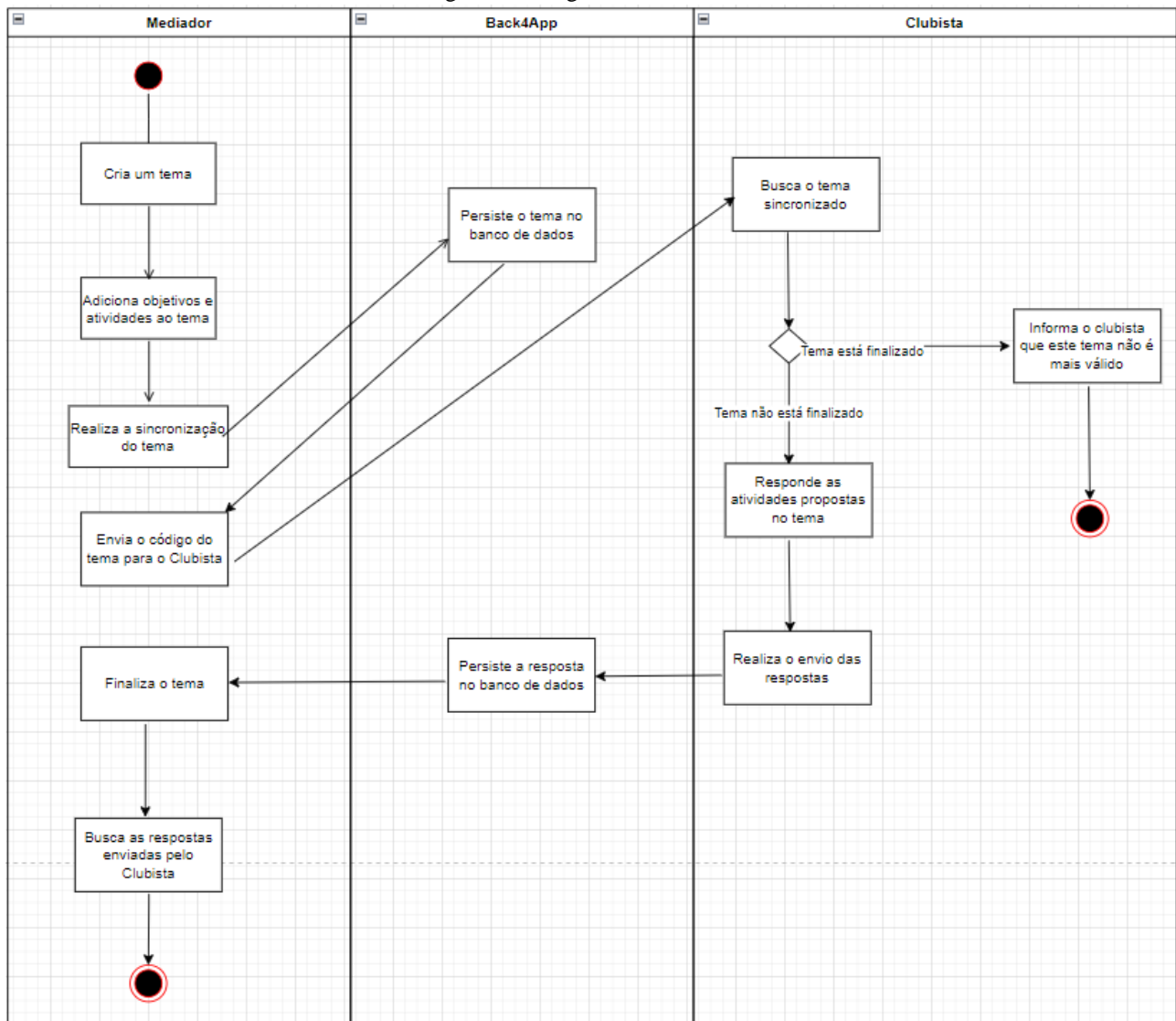
Figura 2 – Diagrama modelo entidade e relacionamento



Fonte: elaborado pelo autor.

No aplicativo os temas podem ser sincronizados ficando disponível para que qualquer usuário com o código do tema possa buscá-lo e executar as atividades propostas. Um tema pode ficar restrito a esse processo caso o usuário que o criou realize a sua finalização, desta forma é determinado que o tema está expirado e não aceita mais o envio de respostas. Após a finalização de um tema o usuário pode visualizar todas as respostas das atividades enviadas. A Figura 3 apresenta o diagrama de atividades demonstrando o fluxo da execução de um tema.

Figura 3 – Diagrama de atividades



Fonte: elaborado pelo autor.

3.2 IMPLEMENTAÇÃO

Nesta seção será apresentado os detalhes sobre o desenvolvimento do aplicativo descrevendo as partes mais importantes da implementação.

3.2.1 Desenvolvimento

O projeto foi dividido em diretórios específicos para cada função no aplicativo. O diretório de **UI** é responsável por toda a parte da codificação da interface, dentro dele são definidas as telas de navegação e os componentes comuns. O diretório **services** possui as **Stores** do MobX, as classes modelos e os repositórios para gerenciamento do banco de dados através do Parse Server. A arquitetura do aplicativo foi estruturada para que cada tela possua uma **Store** do MobX para armazenar o estado e a partir dele determinar o comportamento dos componentes em tela. Como o aplicativo permite a navegação entre várias telas, é necessário que algumas **Stores** possam ser acessadas em qualquer momento, desta forma é utilizado o **GetIt** possibilitando que o estado armazenado em uma **Store** seja registrado como uma instância única. Também existem **Stores** que precisam apenas ser usadas em certos componentes em uma tela, nesta situação é utilizado o **Provider** que possibilita uma **Store** ser descartada após não ser mais necessária e só ser acessada dentro de um contexto específico.

O processo de autenticação do aplicativo é realizado através do Parse Server, que após informar as credenciais da aplicação criada no Back4App, permite que usuários possam criar contas, se autenticar e armazenar dados em um banco de dados. No Quadro 7 apresenta o método responsável por inicializar a conexão do aplicativo com o servidor da aplicação no Back4App através do Parse Server. Na linha 28 é realizado a chamada do método para inicializar a conexão.

Quadro 7 – Método de inicialização do Parse Server

```

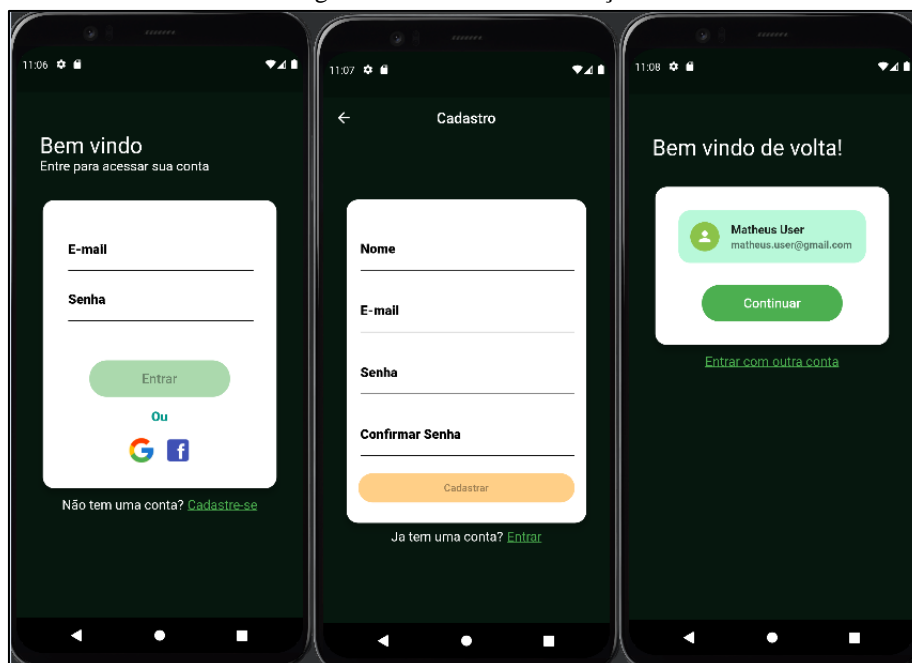
26 void main() async {
27   WidgetsFlutterBinding.ensureInitialized();
28   await initializeParse();
29   await initializeHive();
30   setupLocators();
31   runApp(const MyApp());
32 }
33
34 Future<void> initializeParse() async {
35   await Parse().initialize(
36     appId,
37     serverUrl,
38     clientKey: clientKey,
39     autoSendSessionId: true,
40     debug: true,
41   );
42 }

```

Fonte: elaborado pelo autor.

Ao abrir o aplicativo pela primeira vez a tela inicial apresentada será a tela de **login**, que permite ao usuário, caso já possua uma conta cadastrada, realize a autenticação clicando no botão **Entrar**, conforme apresentado na Figura 4 (a). Se o usuário não possuir uma conta, poderá clicar no botão **Cadastre-se** e preencher os campos solicitados para criar uma conta, os campos são apresentados conforme a Figura 4 (b). Caso o usuário já tenha se autenticado no dispositivo em algum momento, o aplicativo foi estruturado para exibir o último usuário autenticado. Este recurso poderá ser usado em momentos que não é possível acessar a internet para realizar uma nova autenticação, possibilitando que o usuário não seja interrompido de utilizar as funcionalidades do aplicativo, esta funcionalidade é apresentada conforme a Figura 4 (c).

Figura 4 – Telas de autenticação

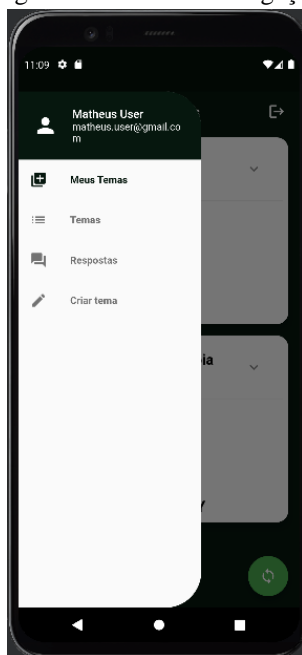


Fonte: elaborado pelo autor.

A tela de autenticação, assim como todas as demais telas, é controlada por uma **Store** do MobX. A **Store** desta tela é responsável por validar os campos que foram inseridos, efetuar a chamada no Back4App e validar a autenticação do usuário. Após autenticado o usuário poderá acessar a barra de navegação do aplicativo no canto superior esquerdo e escolher a tela que deseja acessar, conforme apresentado na Figura 5. A tela de **login** também tem a responsabilidade de realizar o registro das **Stores** de telas que o usuário poderá acessar a qualquer momento na utilização do aplicativo. No Quadro 8 é possível verificar o registro das **Stores** no GetIt. Na linha 68 é registrado a **Store** ThemeStore que é atrelada a tela Meus Temas que é responsável pelo gerenciamento dos temas criados pelo usuário, na linha 71 é

registrado a *Store* SyncedThemeStore referente a tela Temas que possibilita a sincronização e realização das atividades de um tema e na linha 74 é registrado a *Store* ResponsesThemeStore vinculada a tela Respostas, que é responsável por buscar as respostas dos temas criados pelo usuário enviadas por clubistas.

Figura 5 – Menu de navegação



Fonte: elaborado pelo autor.

Quadro 8 – Método de autenticação e registro de *Stores*

```

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!);

```

Fonte: elaborado pelo autor.

Ao acessar a tela Criar tema, será solicitado que o usuário informe um título e uma descrição para o tema, após confirmar será direcionado para a tela Adicione objetivos, na qual permitirá a criação dos objetivos e atividades do tema. Como a criação de um tema está dentro de um contexto específico, ao direcionar para a tela de objetivos é utilizado o Provider para a criação de uma *Store* que poderá ser acessada dentro da árvore de componentes na inclusão dos objetivos e atividades do tema. No Quadro 9 na linha 75 é definido qual a *Store* será criada e na linha 76 é indicado para qual tela o usuário será direcionado.

Quadro 9 – Criação de *Store* através do Provider

```

71      Navigator.push(
72        context,
73        MaterialPageRoute(
74          builder: (_) => Provider(
75            create: (_) => CreateObjectiveStore(),
76            child: CreateObjetivoScreen(
77              createThemeStore: createThemeStore,

```

Fonte: elaborado pelo autor

A adição de atividades nos objetivos, foi estruturada para dar mais liberdade ao usuário e evitar informações redundantes. A atividade é composta pela definição dos recursos e as perguntas que serão realizadas aos Clubistas. Os recursos podem ser fotos, vídeos, áudio ou desenhos. As perguntas podem ser adicionadas e customizadas livremente, é apenas necessário que seja definido o título da pergunta e o tipo da resposta esperada podendo ser um texto, um número inteiro, um número decimal, uma hora ou uma data. Na **figura 6** é apresentado um exemplo de criação de uma atividade.

Figura 6 – Criação de uma atividade



Fonte: elaborado pelo autor.

Ao finalizar **do** cadastro do tema, o usuário poderá clicar no botão **Salvar** no canto superior direito e será direcionado para a tela **Meus Temas**. Ao realizar o processo é necessário que seja acessado a **Store** responsável por controlar os temas do usuário para que o tema recém-criado seja armazenado na memória interna do dispositivo. No Quadro 10 na linha 90 é demonstrado como utilizando o **GetIt** é possível acessar uma **Store** que foi registrada como global.

Quadro 10 – Acessando **Store** global

```

86 void saveTheme() {
87     themeContentError = null;
88
89     if (isThemeValid()) {
90         final ThemeStore themeStore = GetIt.I<ThemeStore>();
91
92         if (index != null) {
93             themeStore.update(index!, theme!);
94         } else {
95             theme!.creator = GetIt
96                 .I<UserManagerStore>()
97                 .user;
98             themeStore.add(theme!);

```

Fonte: elaborado pelo autor.

A tela **Meus Temas** listará todos os temas que o usuário criou. Com a utilização do **Hive**, o aplicativo irá buscar através de um arquivo vinculado ao usuário os temas que estão **na** armazenados no dispositivo. No Quadro 11 é apresentado o método que efetua a busca dos temas, onde na linha 26 é solicitado a permissão de acesso aos arquivos do dispositivo e na linha 29 é aberto o arquivo que armazena os temas. Caso o usuário esteja acessando de outro dispositivo e queira visualizar os temas que foram sincronizados anteriormente, é possível realizar a sincronização clicando no botão **Buscar temas sincronizados**. Os temas encontrados serão salvos no arquivo vinculado ao usuário.

Quadro 11 – Busca dos temas na memória do dispositivo

```

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 }

```

Fonte: elaborado pelo autor.

O tema após sua criação possuirá três status que determinam quais funções podem ser executadas. Um tema não sincronizado pode ser editado, excluído ou sincronizado. Ao realizar a sincronização de um tema é feito a persistência em uma tabela no banco de dados do Back4App. No Quadro 12 é apresentado o método responsável por esse processo, em que na linha 13 é criado um objeto do tipo *ParseObject*, responsável por armazenar toda a estrutura do tema, e na linha 26 é chamado o método *save* que efetua a persistência no Back4App. Este processo gera um código único vinculado ao tema. A partir do código gerado o usuário poderá compartilhar com um clubista permitindo que sincronize o tema para realizar atividades propostas. Um tema sincronizado pode ser apenas visualizado sem a possibilidade de edição. Também pode ser gerado um QR Code facilitando o compartilhamento do tema sem a necessidade de digitar o código do tema. Por fim o tema pode ser finalizado, neste status o tema não poderá mais ser compartilhado e suas respostas estarão disponíveis para análise.

Quadro 12 – Método de persistência do tema no Back4App

```

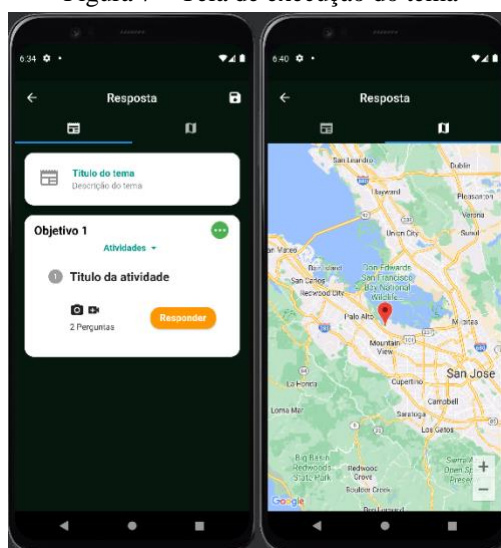
13 final themeObject = ParseObject(keyThemeTable);
14 themeObject.objectId = theme.id;
15
16 final parseAcl = ParseACL(owner: parseUser);
17 parseAcl.setPublicReadAccess(allowed: true);
18 parseAcl.setPublicWriteAccess(allowed: false);
19 themeObject.setACL(parseAcl);
20
21 themeObject.set<String>(keyThemeTitle, theme.title);
22 themeObject.set<String>(keyThemeDescription, theme.description);
23 themeObject.set<ParseUser>(keyThemeCreator, parseUser);
24 themeObject.set<String>(keyThemeStatus, theme.status.value);
25
26 var response = await themeObject.save();

```

Fonte: elaborado pelo autor.

A tela *Temas* permite executar as atividades propostas em um tema através da utilização do código único gerado ao sincronizar um tema. Ao buscar o tema, ele será listado e ficará pendente o envio das respostas. Ao iniciar o tema, o usuário será direcionado para uma tela que possui duas abas, conforme a [Figura 7 a primeira](#) aba será a tela de Resposta onde serão listados os objetivos e suas respectivas [atividades](#) e a [segunda](#) aba a tela de Mapa que mostra a posição do usuário ao responder cada atividade. Para responder uma atividade é necessário apenas escolher a atividade e clicar no botão *Responder*. Quando uma atividade é respondida, é realizado a coleta da geolocalização do usuário e adicionado um marcador com as coordenadas no mapa. Ao responder todas as atividades, o usuário poderá salvar as respostas clicando no botão *Salvar* e ao retornar para tela *Temas* poderá realizar a sincronização enviando as respostas para o Back4App.

Figura 7 – Tela de execução do tema



Fonte: elaborado pelo autor.

A tela Respostas, irá buscar todos os temas que foram finalizados pelo usuário, independentemente de haver ou não o envio de respostas. A premissa é controlar até quando um tema será válido e poderá ser executado por um clubista. Ao realizar a busca será listado os temas finalizados permitindo que o usuário visualize todas as respostas enviadas pelos clubistas e identifique no mapa sua posição geográfica, podendo assim entender como cada clubista realizou as atividades propostas. Na figura 8 é apresentado um tema que possui o envio da resposta para a atividade.

Figura 8 – Visualização das respostas do tema



Fonte: elaborado pelo autor.

4 RESULTADOS

De modo a ampliar o seu caráter científico, todos os TCCs devem apresentar e discutir resultados não limitados à comparação com os trabalhos correlatos. Devem ser apresentados os casos de testes do software, destacando objetivo do teste, como foi realizada a coleta de dados e a apresentação dos resultados obtidos, preferencialmente em forma de gráficos ou tabelas, fazendo comentários sobre eles. Também é sugerida a comparação com os trabalhos correlatos apresentados na fundamentação teórica.

5 CONCLUSÕES

As conclusões devem refletir os principais resultados alcançados, realizando uma avaliação em relação aos objetivos previamente formulados. Deve-se deixar claro se os objetivos foram atendidos, se as ferramentas utilizadas foram adequadas e quais as principais contribuições do trabalho sociais ou práticas para o seu grupo de usuários bem como para o desenvolvimento científico e ou tecnológico da área.

Deve-se incluir também as limitações e as possíveis extensões do TCC.

REFERÊNCIAS

- BET, Sabrina et al. **Um ambiente colaborativo para auxílio ao aprendizado**: Clube Virtual de Ciências. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 84-87, nov. 2004. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/383>. Acesso em: 30 nov. 2021. doi: <http://dx.doi.org/10.5753/cbie.sbie.2004.84-87>.
- CÓRDOBA. **Club Escolar de Ciencias y Tecnologías**. Ministerio de Educación; Ministerio de Ciencia y Tecnología, 2012. Disponível em: <https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/publicaciones/documentos/Club%20de%20ciencias%2025-7-12.pdf>. Acesso em: 20 set. 2021.
- FREITAS, Thais C. de Oliveira, SANTOS, Carlos A. M. dos. **Clubes de ciências na Escola**: um guia para professores, gestores e pesquisadores. Curitiba: Associação Brasileira de Editores Científicos, 2021. 166p.
- KORBES, Gustavo H. **ExploraHabitat**: Um aplicativo para apoiar as saídas a campo em Clubes de Ciências. 2021. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau. Disponível em: <http://dsc.inf.furb.br/tcc/index.php?cd=6&tcc=2080>. Acesso em: 20 set. 2021.
- MANCUSO, Ronaldo; LIMA, Valderez; BANDEIRA, Vera. **Clube de ciências**: Criação, funcionamento, dinamização. Porto Alegre: Calábria Artes Gráficas, 1996. 365p.
- MENDONÇA, Karoene Dirlene et al. **Ambiente de Aprendizagem Ubíqua para Auxiliar o Estudo de Botânica em Atividades de Aula de Campo**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 1738, out. 2018. ISSN 2316-6533. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/8141>. Acesso em: 30 nov. 2021. doi:<http://dx.doi.org/10.5753/cbie.sbie.2018.1738>.
- PAVANI, Elaine C. R. **Aulas de campo na perspectiva histórico-crítica**: contribuições para os espaços de educação não formal. 2013. 110 f. Dissertação (Mestrado em Educação em Ciências e Matemática) – Programa de Pós-Graduação em Ciências e Matemática do Instituto Federal do Espírito Santo, Vitória.
- ROSA, Vagner Santos da. **Ambicare**: monitoramento ambiental usando dispositivos móveis. Revista de Empreendedorismo, Inovação e Tecnologia, Passo Fundo, v. 1, n. 2, p. 43-49, out. 2015. ISSN 2359-3539. Disponível em: <https://seer.imed.edu.br/index.php/revistas/article/view/776>. Acesso em: 30 nov. 2021. doi:<https://doi.org/10.18256/2359-3539/reit-imed.v1n2p43-49>.
- SANTAELLA, L. **Desafios da ubiquidade para a educação**. Revista Ensino Superior Unicamp. 2013. Disponível em: <https://www.revistaensinosuperior.gr.unicamp.br/artigos/desafios-da-ubiquidade-para-a-educacao>. Acesso em: 30 nov. 2021.
- SANTAELLA, L. **A aprendizagem ubíqua na educação aberta**. Revista Tempos e Espaços em Educação, p. 15-22, 30 dez. 2014. Disponível em: <https://seer.ufs.br/index.php/revtee/article/view/3446>. Acesso em: 30 nov. 2021.
- SILVA, Marcelo Scabelo da et al. **Aulas de campo para a alfabetização científica**: uma intervenção pedagógica no Parque Estadual da Fonte Grande (Vitória/ES). Imagens da Educação, v. 8, n. 2, p. e41740. 2018.

APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

É fundamental que todo projeto apresente alguma forma de especificação do que foi desenvolvido. A descrição é opcional. Assim, **este apêndice deve conter os diagramas de especificação que não couberam ao longo do texto.** Os diagramas devem conter legendas numeradas na sequência do artigo.

Cada apêndice deve iniciar em uma nova página.

APÊNDICE B – XXX

Podem ser inseridos outros apêndices no artigo tais como códigos de implementação, telas de interface, instrumentos de coleta de dados, entre outros. **Apêndices são textos elaborados pelo autor** a fim de complementar sua argumentação. Os apêndices são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada apêndice. Colocar sempre um preâmbulo no apêndice. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

ANEXO A – DESCRIÇÃO

Elemento opcional, **anexos são documentos não elaborados pelo autor**, que servem de fundamentação, comprovação ou ilustração, como mapas, leis, estatutos, entre outros. Os anexos são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada anexo. Colocar sempre um preâmbulo no anexo. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.