

Desenvolvimento de uma Aplicação Móvel para Navegação Imersiva em Imagens 360° com Óculos VR

Jonathan Silva Santiago¹, Gustavo Stangherlin Cantarelli¹

¹Curso de Sistema de Informação – Universidade Franciscana (UFN)
CEP 97010-032 – 91.501-970 – Santa Maria – RS – Brasil

{jonathansilvajss, gus.cant}@gmail.com

Abstract. *This paper aims to expose the elaboration of an application for the Android platform, where it will serve as a 360° player for 360° images. For the construction of this project, the Feature-Driven Development and Cowboy methodologies were adopted. The application will be developed in the Java program language, and will be implemented in the Android Studio integrated development environment. Thus it is intended to facilitate the navigation in VR in 360° images and help people with physical disabilities to enter the technological environment, as well as reduce the financial investment required to perform such navigation.*

Resumo. *Este trabalho tem como objetivo expor a elaboração de um aplicativo para a plataforma Android, onde o mesmo servirá como um player em 360° para imagens em 360°. Para a construção desse projeto, foram adotadas as metodologias Feature-Driven Development e Cowboy. O aplicativo foi desenvolvido na linguagem de programa Java, e sua implementação foi realizada no ambiente de desenvolvimento integrado Android Studio. Dessa forma pretende-se facilitar a navegação em VR nas imagens 360° e auxiliar pessoas portadoras de deficiência física a se inserirem no meio tecnológico, assim como diminuir o investimento financeiro necessário para realizar tal navegação.*

1. Introdução

As pessoas estão cada vez mais em contato direto com a tecnologia, e em muitos casos o mesmo ocorre quando elas utilizam os seus dispositivos móveis (Smartphones). Com diversas finalidades de uso, o aparelho móvel está se tornando parte do cotidiano da maioria das pessoas, e as opções de utilização dos dispositivos estão aumentando consideravelmente.

Os avanços constantes da tecnologia móvel fazem com que, os Smartphones ganhem cada vez mais importância no cotidiano das pessoas, e segundo Cardoso e Santos (2015) “nos últimos anos os impactos desses avanços se incorporam na sociedade de forma mais rápida e consolidada, fazendo com que o não acompanhamento dos mesmos possa acarretar numa defasagem e futuramente numa exclusão tecnológica”. Para tentar evitar a exclusão de alguns usuários, e realizar inclusão dos mesmos, empresas já estão investindo em maneiras de integrar cada vez mais os seus usuários, com os seus produtos (tecnologias).

A evolução tecnológica acaba ocasionando grandes mudanças na sociedade como um todo, alterando os seus hábitos, a sua forma de pensar, e como se comunicar. Dessa forma os dispositivos tecnológicos existentes acabam exercendo um papel fundamental na mediação das relações sociais e manifestações culturais da sociedade moderna [Canto, 2010].

Sabendo da importância que a tecnologia exerce na vida das pessoas e com a popularização em massa dos Smartphones, este trabalho visa projetar e desenvolver uma aplicação móvel, no sistema operacional Android, com a utilização da Realidade Virtual, do inglês *Virtual Reality* (VR), sendo que tal aplicação será semelhante a um player de vídeo em VR para a navegação em imagens 360°.

1.1. Justificativa

Com base nas pesquisas realizadas, percebeu-se uma carência de aplicações móveis para realizar a navegação imersiva em imagens 360°, de maneira com que a execução dessa navegação possa ser realizada facilmente por inúmeros tipos de usuários, e sobretudo os que possuem algum tipo de deficiência física, pois a aplicação irá necessitar de pouca interação física. A questão turística também deve ser ressaltada, uma vez que o aplicativo proporciona ao usuário a inserção de suas próprias imagens, proporcionando ao mesmo a exploração de inúmeros locais, e com a aplicação essa exploração será simplificada. Outro fator a ser encontrado é que as opções já existentes precisam de um investimento maior para que se possa usufruir da tecnologia por completo.

1.2. Objetivo Geral

Desenvolver uma aplicação para a plataforma Android, que facilite a navegação em imagens 360°, e com isso, auxilie também na inclusão de pessoas portadoras de alguma deficiência física a se inserirem mais no mundo virtual. Com o auxílio Google VR SDK (*Software Development Kit*), será possível desenvolver uma aplicação que precise de pouca interação física, entre o usuário e o dispositivo (Smartphone), contribuindo de forma considerável para que o mesmo consiga utilizar muitas de suas funcionalidades sem que precise estar de forma constante executando comandos no visor do dispositivo.

1.3. Objetivos Específicos

Para o desenvolvimento deste trabalho, os objetivos específicos identificados estão elencados a seguir:

- Estudar a utilização do Google VR SDK;
- Disponibilizar ao usuário uma aplicação amigável e de fácil utilização;
- Utilizar o Ambiente de Desenvolvimento Integrado (IDE) Android Studio para o desenvolvimento completo de uma aplicação Android nativo;
- Trabalhar com Realidade Virtual em uma aplicação Android;
- Utilizar boas práticas da metodologia *Cowboy* em conjunto com a FDD (*Feature Driven Development*) no desenvolvimento do projeto;
- Auxiliar na integração usuário – dispositivo, com um baixo investimento do seu utilizador;
- Integrar com o mundo virtual as pessoas com deficiência física, auxiliando-as a se sentirem parte da tecnologia.

1.4. Organização do Trabalho

A Seção 2 expõe o referencial teórico e todas as tecnologias que serão utilizadas para a concepção deste trabalho. A Seção 3 denota os trabalhos correlatos que foram utilizados para a elaboração e melhor entendimento do projeto. A Seção 4 aborda a metodologia utilizada para o desenvolvimento do trabalho, assim como a Seção 5 exibe os resultados do software, a Seção 6 apresentam os problemas encontrados no desenvolvimento do projeto e por último, na seção 6 expõe conclusão sobre o trabalho.

2. Referencial Teórico

Esta seção apresentará os conceitos necessários para o desenvolvimento deste trabalho, assim como quais as tecnologias que ele irá utilizar para realizar o que foi proposto. Serão apresentadas as concepções de Realidade Virtual, Navegação Imersiva, Google VR SDK, Android, Android Studio e Java, sendo assim possível ter uma melhor compreensão das tecnologias e técnicas utilizadas para a elaboração da aplicação.

2.1. Realidade Virtual

De acordo com Kirner e Siscoutto (2007) “a Realidade Virtual é uma ‘interface avançada do usuário’ para acessar aplicações executadas no computador, propiciando a visualização, movimentação e interação do usuário, em tempo real, em ambientes tridimensionais gerados por computador”. Dessa forma, a mesma irá se aproximar o máximo possível do que realmente acontece na realidade, sendo uma projeção virtual e simulada do que acontece na mesma.

Os sentidos são fundamentais para que o usuário vivencie parte da tecnologia, como isso se sinta imerso cada vez mais no ambiente escolhido, e dessa forma seja possível que o mesmo tenha uma experiência completa, dentro daquilo que a tecnologia pode proporcionar. A Realidade Virtual em sua grande maioria tentar aproximar ao máximo os dois mundos, ou seja, o que pertence ao mundo virtual do que pertence ao mundo real, fazendo assim uma junção do que há de melhor nos dois, para que se possa vivenciar experiências cada vez mais realistas, porém tal tentativa de aproximação entre os dois mundos, não exclui o fato da Realidade Virtual também poder trabalhar com o mundo fictício [Marçal et. al. 2005].

2.2. Navegação Imersiva

Para Darken (1996) a navegação pode ser definida como “(1) Viajar pela água: barco à vela. (2) Conduzir um a direção através de algum meio: conduzir um avião. (3) Viajar muito: caminhar.”. Seguindo essa linha de pensamento, se pode concluir que o simples fato de caminhar será considerado uma forma de navegação, e independente de como tal ação será realizada. O simples fato de estar em movimento e realizando uma interação com o ambiente exterior, independentemente de como tal movimento e interação estão sendo executados, os mesmos irão acabar resultando em uma navegação que o indivíduo e ou objeto, exercerá sobre o universo.

Na época atual a navegação pode ser realizada de várias maneiras, pois com o auxílio da tecnologia uma pessoa consegue navegar em muitos lugares mesmo sem estar presente no mesmo. Com essa possibilidade as pessoas de um modo geral estão cada vez mais imersas no mundo virtual, buscando de certo modo por prazeres que não conseguem

ter no mundo real, por diversos motivos, mas que com o auxílio tecnológico conseguem usufruir e ter uma experiência o mais próximo do real possível.

2.3. Google VR SDK

O SDK da Google fornece uma interface de programação em VR que o app precisa, oferecendo amplo suporte à navegação nas imagens em 360°. O suporte do *kit* Google, abrange inúmeras funcionalidades, como o suporte ao giroscópio e rastreamento de posição, facilitando a ligação entre software e hardware. Desta forma o aplicativo poderá ter acesso a posição física do usuário através do rastreamento realizado pelo SDK, utilizando-o como parâmetro de entrada para criar uma experiência cada vez mais imersiva [Xu et al 2017]

2.4. Android

O Android é um Sistema Operacional, do inglês *Operating System* (OS) completo e avançado, envolvendo um pacote com programas para celulares, já com um *middleware*, aplicativos e interface de usuário. Foi construído com a intenção de permitir aos desenvolvedores criar aplicações móveis que possam tirar total proveito do que um aparelho portátil possa oferecer [Pereira e Da Silva 2009].

O sistema operacional Android é baseado em um *kernel* Linux, que possui uma rica Interface de Usuário (IU), aplicativos de usuário, bibliotecas de código, frameworks de aplicativo, suporte a multimídia, dentre outros. Os componentes do OS são escritos em C ou C++, já os aplicativos de usuários são escritos em sua grande maioria em Java, sendo que mesmo os aplicativos embutidos em sua grande maioria podem ser escritos em Java [Ableson et al. 2012].

Uma característica da Sistema Operacional Android é que não existe diferença entre os aplicativos integrados e os aplicativos que o usuário pode criar com o *Software Development Kit* (SDK), sendo assim é possível criar poderosos aplicativos que acessam os recursos disponíveis no dispositivo. Outra característica que pode ser ressaltada, e que também pode ser caracterizado com a sua principal característica, é o fato de que o código do Android é aberto [Ableson et al. 2012].

2.5 Android Studio

O Android Studio é o Ambiente de Desenvolvimento Integrado oficial para a construção de aplicativos Android, o mesmo foi elaborado pela própria Google para que os programadores pudessem ter uma IDE completa e simples. Toda a sua concepção foi baseada no IntelliJ IDEA, que é uma IDE mais antiga. Além do editor de códigos e das ferramentas avançadas do IntelliJ, o Android Studio Dispõe de outros recursos que podem aumentar a produtividade na criação de aplicativos Android [Android 2019].

2.6. Java

O Java é uma linguagem de programação simples e de fácil aprendizado, pois ela possui reduzido número de construções. A diminuição das construções mais suscetíveis a erros de programação, tais como ponteiros e gerenciamento de memória via código de programação, faz com que a programação em Java seja mais eficiente. A linguagem também dispõe de inúmeras bibliotecas, que fornecem grande parte de suas funcionalidades. A linguagem é baseada no paradigma da Orientação a Objetos, que

permite a modularização das aplicações, reuso e manutenção simples do código já existente [Indrusiak 1996].

3. Trabalhos Correlatos

Nesta seção serão denotados alguns trabalhos semelhantes ao aplicativo proposto, sendo que os mesmos possuem uma relação direta ou indireta com o trabalho elaborado. Desta forma os mesmos serviram como base de estudo para todo o desenvolvimento da aplicação proposta. Também foi demonstrado um software implementado com um objetivo muito semelhante, onde o mesmo difere do aplicativo aqui apresentado, por utilizar as imagens providas da API do Google Street View.

3.1. Vídeos Interativos e Imersivos no Sight Surfers

O trabalho Álvares (2013) identificou que a maioria dos vídeos disponíveis atualmente, estão no aspecto que não é 360°, sendo assim muitas informações são perdidas quando um vídeo é gravado. Sabendo disso o mesmo propôs o desenvolvimento de um software para explorar de maneira mais imersiva e interativa os vídeos em 360°, fazendo com que o usuário se sinta cada vez mais parte da tecnologia.

O resultado foi muito positivo, pois com o desenvolvimento de um leitor de vídeos panorâmicos em alta resolução, foi possível ter experiências exploratórias em ambientes com mais imersão. As pessoas que testaram o software perceberam que o Sight Surfers é uma ferramenta muito útil e de fácil utilização [Álvares 2013].

3.2. Realidade Virtual e Geografia: O caso do Google Cardboard Glasses para o Ensino

O trabalho de Cardoso e Santos (2015) tem como principal objetivo avaliar a viabilidade do uso da realidade virtual como recurso didático no ensino de geografia, levando em consideração aspectos como, a adaptação de conteúdos VR para as salas de aula. No estudo foram utilizados os óculos Google Cardboard Glasses, pois o mesmo possui um investimento financeiro relativamente baixo, além do fato que para a sua utilização, basta apenas que o usuário possua um dispositivo móvel com uma aplicação com conteúdo VR.

Como resultado, se pode observar que o experimento realizado por Cardoso e Santos (2015) teve uma contribuição positiva, pois com a utilização do Street View podem ser observados inúmeros locais, pois a aplicação possui um acervo de imagens muito grande. A ferramenta oferece a possibilidade de explorar os mais diversos tipos de ambientes, biomas, climas e cidades, dessa forma a ferramenta pode auxiliar consideravelmente a compreensão do conteúdo pelo aluno.

3.3. Google Maps – *Street View Player*

O software implementado por Folts (2017) apresenta um conceito de *Street View Player* muito semelhante ao que se pretende neste trabalho, sendo que o mesmo desenvolveu um player para as imagens do Street View, com o objetivo de realizar uma navegação direta entre o ponto de partida e o destino, dessa forma necessitando apenas que o usuário realize algumas configurações, o trabalho de Folts difere do aplicativo desenvolvido nesse trabalho pelo fato deste trabalho não fazer uso das imagens do Street View.

O trabalho desenvolvido cumpriu com o objetivo proposto, e entrega para o usuário uma navegação simples e objetiva, porém a utilização do mesmo já não é mais

possível, uma vez que tal aplicação não está de acordo com as novas políticas de cobrança praticadas pela Google.

3.4. Considerações Sobre os Trabalhos Correlatos

Todos os trabalhos relacionados aqui mencionados, auxiliaram no desenvolvimento deste projeto, pois cada um se encaixa de uma forma diferente na concepção e desenvolvimento do mesmo. Cada trabalho propõe algo diferente, mas que tem uma relação entre si, pois o aplicativo que foi desenvolvido ajuda na integração e relação entres os trabalhos correlatos aqui demonstrados.

4. Metodologia

Para o desenvolvimento deste trabalho serão utilizadas as boas práticas de duas metodologias, a *Feature Driven Development* (FDD) e a Codificação *Cowboy*. A FDD é uma metodologia de desenvolvimento ágil mais completa e robusta que a codificação *Cowboy*, porém para utilizar a mesma é necessário seguir alguns passos obrigatórios. Com a utilização da codificação *Cowboy* essa obrigatoriedade pode ser quebrada, pois tal metodologia irá proporcionar uma maior liberdade na construção do projeto como um todo, deixando a critério do desenvolvedor escolher os passos necessários para alcançar o objetivo esperado.

4.1. Codificação *Cowboy*

A maioria das metodologias de desenvolvimento de software, orientam o desenvolvimento em grupo, ou seja, trabalham de forma a orientar e modelar todo processo que ocorre durante a criação de um software em equipe. Porém muitas pessoas, acabam desenvolvendo softwares sozinhos, ou seja, acabam não adotando o trabalho em equipe. Esses desenvolvedores podem utilizar o termo *Cowboy* para se referir ao desenvolvimento “solitário” de um software [Hollar 2016].

A codificação *Cowboy* como também é chamada, é caracterizada como a ausência de uma metodologia de desenvolvimento no momento da construção do software, baseado principalmente nas boas práticas que o programador possui, sendo assim, levando em consideração muito a experiência em desenvolvimento que a pessoa tem. Dessa forma, fica a critério de quem está desenvolvendo escolher qual o caminho a ser percorrido e principalmente de que forma o mesmo será percorrido para que o objetivo principal, que é o desenvolvimento do software, seja alcançado. Com isso, tal metodologia pode ser utilizada em conjunto com outras, com o objetivo de extrair apenas o que o desenvolvedor julgar necessário para a elaboração do projeto [Hollar 2016].

4.2. *Feature Driven Development*

O FDD é uma metodologia de desenvolvimento de software que inclui alguns benefícios de processos rigorosos, como modelagem, planejamento prévio e controle do projeto, porém a mesma dispõe de características de processos ágeis, como foco na programação, interação constante como cliente e entrega frequente de versão do produto. Tal metodologia prevê e tem um foco maior no desenvolvimento do software em si, não se preocupando com outros fatores como a escolha de tecnologias e ferramentas [Silva et al. 2009].

“Seus princípios e práticas proporcionam um equilíbrio entre as filosofias tradicionais e as mais extremas, proporcionando uma transição mais suave para organizações mais conservadoras, e a retomada da responsabilidade para as organizações.” [Caldeira et al. 2013].

Para o desenvolvimento de um software com a metodologia FDD, alguns processos são propostos pela mesma, são eles: Desenvolver um Modelo Abrangente, construir uma Lista de Funcionalidades, Planejar Através de Funcionalidades, Projetar Através de Funcionalidades e Construir Através de Funcionalidades. Cada processo tem sua importância no momento do desenvolvimento, e a metodologia dá suporte e abrange todos esses cinco processos [Silva et al. 2009].

4.3. Desenvolver um Modelo Abrangente

“O processo Desenvolver um Modelo Abrangente é responsável pelo estudo detalhado sobre o domínio do negócio e pela definição do escopo do projeto” [Silva et al. 2009]. Com um modelo abrangente é possível ter uma visão mais macro de toda a aplicação, sendo que o mesmo pode ser observado no Diagrama de Atividade do projeto (APÊNDICE A).

O Diagrama de Atividades do projeto, ilustra de maneira ampla como será a utilização do aplicativo, demonstrando de forma macro quais serão os processos necessários para que o usuário consiga utilizá-lo. Outras informações importantes, assim como algumas decisões que o usuário deverá tomar no momento de utilizar a aplicação, também estão contidas no diagrama, onde o mesmo tem como objetivo apresentar uma visão geral dos processos do mesmo.

4.4. Construir a Lista de Funcionalidades

O segundo passo no decorrer do desenvolvimento de uma aplicação utilizando a metodologia FDD, é Construir uma Lista de Funcionalidades. Nesta etapa todas as funcionalidades necessárias ao cumprimento das necessidades do cliente são levantadas, desta forma tais funcionalidades podem ser agrupadas em requisitos, que irão auxiliar no processo de desenvolvimento do projeto [Silva et al. 2009]. Nessa fase os requisitos funcionais (APÊNDICE B) e os requisitos não funcionais (APÊNDICE C) foram levantados.

4.5. Planejar por Funcionalidades

Todos os itens que foram elaborados no processo anterior, são separados e ordenados por prioridades de desenvolvimento no processo de planejar através de funcionalidades. Todos os itens são considerados nesta etapa, sendo que até as funcionalidades não funcionais, ou seja, os requisitos não funcionais, são levados em consideração. Ao término da etapa é gerada uma lista das classes, onde as mesmas são associadas aos desenvolvedores responsáveis. Tal trabalho serve também para que se consiga mensurar o nível de complexidades das tarefas, e dessa forma calcular um tempo para que as elas possam ser executadas [Silva et al. 2009].

Foi realizado um planejamento para identificar as principais funcionalidades do aplicativo, e dessa forma definir uma ordem em que a implementação deverá ser seguida. Para cada funcionalidade foi feito um planejamento, levando em consideração múltiplos fatores, sendo que um dos principais é o seu nível de dificuldade. A prioridade de

desenvolvimento também foi levada em consideração, pois cada processo possui um nível de prioridade, e seguir tal nível é fundamental para realizar o desenvolvimento do projeto, assim como também será importante no momento de implementação do mesmo.

Para realizar o cálculo da estimativa de tempo para o desenvolvimento do trabalho, foi elaborada uma métrica (APÊNDICE D):

- 15 horas por semana para implementação do aplicativo;
- 4 meses para o desenvolvimento, totalizando 16 semanas;
- A multiplicação das horas semanais pela quantidade de semanas, resulta em um total de 240 horas, que serão distribuídas entre os requisitos;
- 30% das horas para o desenvolvimento de cada requisito será reservado para os testes dos mesmos.
- 10% dos 30% do total de horas reservadas para os testes, serão atribuídas a testes realizados com outras pessoas.

O Diagrama de Casos de Uso deste projeto foi desenvolvido de maneira bem sucinta, dessa forma facilitando o seu entendimento. Tal diagrama é responsável por mostrar o conjunto de casos de uso e atores e os seus relacionamentos, sendo que o mesmo organiza os comportamentos do sistema, auxiliando os responsáveis pelo projeto a terem uma visão mais completa das interações que ocorrem na aplicação a ser desenvolvida. Este diagrama é importante principalmente para a organização e modelagem dos comportamentos de um sistema (APÊNDICE E).

Os requisitos funcionais foram levantados de uma forma mais ampla, sendo assim, todos os micros processos que compreendem cada um dos casos de uso, estão inclusos dentro dos mesmos, desta forma não é possível visualizá-los explicitamente (APÊNDICES F e G).

4.6. Projetar por Funcionalidade

Após ter definido as funcionalidades do sistema, o passo seguinte é projetar através de funcionalidades, ou seja, para cada um processo que foi definido na lista, será elaborado um projeto, uma atividade a ser realizada. É neste processo também que o modelo da interface do usuário pode ser esboçado, assim como os diagramas de sequência e de classes também podem ser confeccionados [Silva et al. 2009].

O fluxo geral da aplicação e suas principais funcionalidades na perspectiva do usuário, podem ser observadas através do diagrama de atividade já apresentados na seção “4.3. Desenvolver um modelo abrangente, na Figura 3”. Tal diagrama irá mostrar de maneira mais abrangente, os processos mais importantes na execução da aplicação, sendo que os processos menores estão contidos dentro dos maiores que estão exemplificados, mostrando dessa forma uma visão mais macro da execução.

Na Figura 5 é apresentado o diagrama de classes deste projeto, com as possíveis classes que serão implementadas para a execução da aplicação, assim como os seus possíveis atributos.

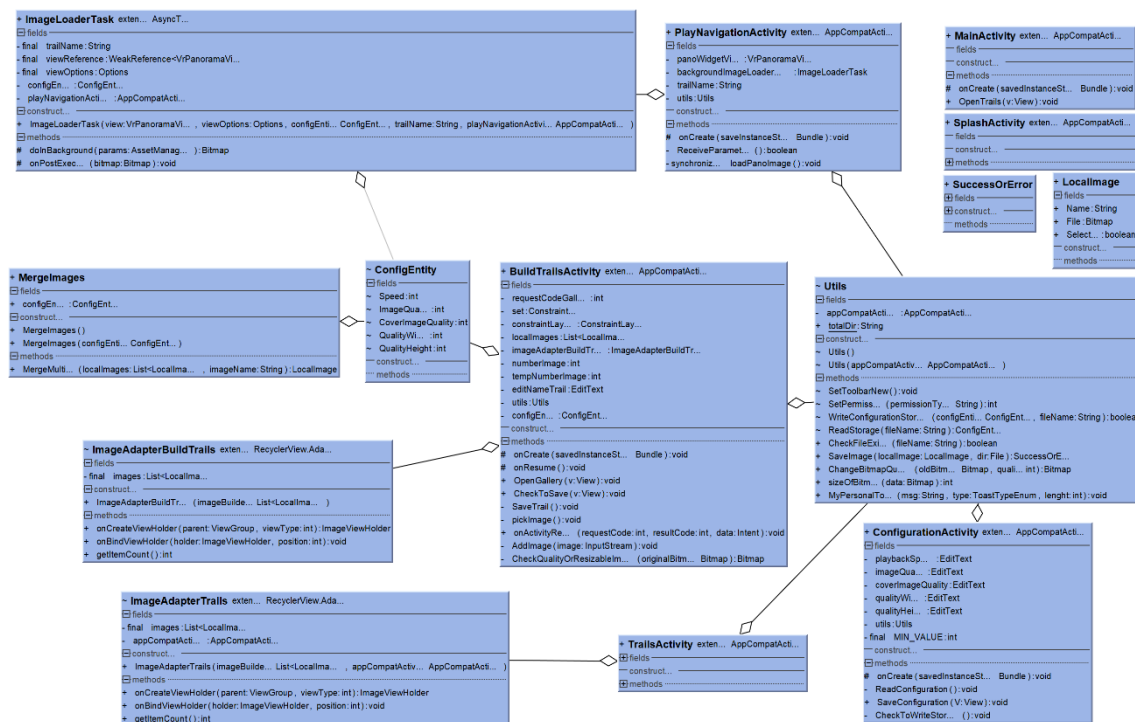


Figura 5. Diagrama de Classes

4.7. Construir por Funcionalidade

A fase de construção por funcionalidade de um software, é a última fase da metodologia FDD, sendo que na mesma é desenvolvido o código e desenvolvida a aplicação, onde a mesma se utiliza de toda a documentação até então aqui elaborada. Nesta etapa do desenvolvimento do projeto, foi aplicada a codificação *Cowboy*, sendo que ela utiliza da experiência do desenvolvedor, para poder escolher a melhor maneira de realizar a codificação (desenvolvimento do código).

A principal funcionalidade do aplicativo é a realização da navegação, onde após o usuário selecionar trilha desejada a aplicação irá dar início a reprodução do conteúdo escolhido. Na Figura 4 é possível visualizar o trecho de código XML (*Extensible Markup Language*), onde é realizada a navegação. A parte do XML é responsável apenas pela exibição do conteúdo em si, sendo que o tratamento e controle do mesmo, é realizado pelo Java.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.google.vr.sdk.widgets.pano.VrPanoramaView
        android:id="@+id/pano_view"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"/>

</RelativeLayout>
  
```

Figura 4. XML realizar navegação

O tratamento da imagem fica sob responsabilidade do Java, onde no mesmo é aplicada as regras definidas pelo usuário, ou seja, a configuração da navegação é executada neste momento. Para realizar todo esse tratamento, todas as etapas para se trabalhar corretamente com as imagens, foram elaboradas para conceder esse suporte. Na Figura 5 é possível observar o construtor da classe principal que realiza a navegação *PlayNavigationActivity*, nesse trecho de código é possível observar as primeiras configurações para realizar a ligação do Java com o componente gráfico onde será realizado a navegação *pano_view*.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_play_navigation);
    utils = new Utils( AppCompatActivity: this);

    if (ReceiveParameters()) {
        Window w = getWindow();
        w.setFlags(WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS,
            WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS);

        panoWidgetView = findViewById(R.id.pano_view);
        loadPanoImage();
    } else {
        utils.MyPersonalToast(msg: "Não foi possível iniciar a navegação!",
            Utils.ToastTypeEnum.ERROR, Toast.LENGTH_SHORT);
        finish();
    }
}

```

Figura 5. Construtor *PlayNavigationActivity*.

Os demais métodos que compõem a classe *PlayNavigationActivity*, estão ilustrados na Figura 6. O método *ReceiveParameters*, é responsável pelo recebimento do parâmetro *trailName*, que se trata do nome da trilha que o usuário deseja reproduzir. Outro método que pode ser visualizado na figura, é o *loadPanoImage*, onde o mesmo é responsável pela instância da classe *ImageLoaderTask* que realiza a navegação, o método também é responsável por algumas configurações iniciais da navegação, assim como é repassado para a classe instanciada tais configurações e o nome da trilha a ser executada.

```

private boolean ReceiveParameters() {
    Intent parametersReceived = getIntent();
    if (parametersReceived != null) {
        trailName = parametersReceived.getStringExtra("trailName");
        return true;
    }
    return false;
}

private synchronized void loadPanoImage() {
    VrPanoramaView.Options viewOptions = new VrPanoramaView.Options();
    viewOptions.inputType = VrPanoramaView.Options.TYPE_MONO;

    ImageLoaderTask task = new ImageLoaderTask(
        panoWidgetView,
        viewOptions,
        utils.ReadStorage(Utils.FileNameEnum.CONFIGURATION.toString()),
        trailName,
        playNavigationActivity: this);
    task.execute(this.getAssets());
    backgroundImageLoaderTask = task;
}

```

Figura 6. Métodos de Configuração da Classe *PlayNavigationActivity*.

A classe *ImageLoaderTask*, é de fato onde a navegação em VR pelas imagens selecionadas pelo usuário ocorre, a mesma herda da classe *AsyncTask* Responsável por realizar a execução assíncrona. O construtor dessa classe é responsável por receber os parâmetros que foram enviados pela *PlayNavigationActivity*, o principal método dessa

classe é o *onPostExecute*, sendo que o mesmo é responsável pelo tratamento final das imagens antes de realizar a navegação, o método pode ser visualizado na Figura 7.

```
@Override
protected void onPostExecute(Bitmap bitmap) {
    final VrPanoramaView vw = viewReference.get();

    File folder = new File( pathname: Utils.totalDir + "/" + trailName + "/" );
    File[] allFiles = null;

    if (folder.exists()) {
        allFiles = folder.listFiles((FilenameFilter) (dir, name) -> {
            return ((name.endsWith(".jpg") ||
                name.endsWith(".jpeg") ||
                name.endsWith(".png")) &&
                !name.startsWith("cover"));
        });
    }

    final List<Bitmap> bitmapList = new ArrayList<>();

    if (allFiles != null)
        for (File item : allFiles) {
            bitmapList.add( index 0, BitmapFactory.decodeFile(item.getAbsolutePath()));
        }

    new Thread((Runnable) () -> {
        for (Bitmap item : bitmapList) {
            if (vw != null && item != null) {
                vw.loadImageFromBitmap(item, viewOptions);
            }
            try {
                Thread.sleep(TimeUnit.SECONDS.toMillis(configEntity.Speed));
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        playNavigationActivity.finish();
    }).start();
}
```

Figura 7. Método *onPostExecute* – classe *ImageLoaderTask*.

No método *onPostExecute* onde é iniciado de a navegação, primeiramente é realizada uma busca na pasta onde foram salvas as imagens da trilha selecionada pelo usuário, sendo assim o parâmetro como o nome da trilha é utilizado nesse momento. Após buscar todas as imagens que estão na pasta da trilha, exceto a imagem de capa, tais imagens são preparadas e adicionadas em uma lista de *Bitmap*, posteriormente está lista é percorrida dentro de uma *Thread* para que sua execução não atrapalhe a visualização do conteúdo no *display* do dispositivo (smartphone). Ao realizar percorrer a lista com as imagens, o tempo em que uma imagem fica em execução é definido pelo usuário, sendo que dentro do laço de repetição, há uma parada para que assim seja possível respeitar o tempo configurado pelo usuário.

Dessa forma ocorre a navegação pelas imagens cadastradas pelo usuário, e o aplicativo consegue navegar nas imagens respeitando o tempo determinado, e ao chegar no final da apresentação, o mesmo ficará exibindo apenas a última imagem, aguardando ação do usuário. Sendo assim, foi possível desenvolver a principal funcionalidade da aplicação, utilizando a codificação *Cowboy*, o desenvolvimento ficou mais fluído e ocorreu dentro daquilo que se havia planejado, onde tal codificação auxiliou na quinta fase do FDD (Construir por Funcionalidade).

A codificação *Cowboy* preza principalmente pelo conhecimento do desenvolvedor no momento de realizar a codificação, sendo assim, para desenvolver o código deste projeto essa flexibilidade contribuiu consideravelmente para o bom desenvolvimento do mesmo. Com a utilização de tal codificação, foi possível focar em um primeiro momento, na funcionalidade essencial do projeto, que é a reprodução das imagens, e dessa forma foi possível identificar a tempo um problema que acarretou a mudança em parte da estrutura do projeto. A *Cowboy*, foi utilizada principalmente para acelerar no momento de realizar a implementação, com o objetivo de identificar rapidamente os possíveis problemas do projeto.

5. Resultados

Este trabalho tem como propósito a construção (desenvolvimento) de um aplicativo para dispositivos Android, onde o mesmo irá facilitar a interação (navegação) dos usuários em imagens 360°, propiciando uma maior imersão na tecnologia, e conseqüentemente nos ambientes que tais usuários estão explorando. Deve ser salientado também como resultado obtido, o fato deste trabalho ter sido publicado como artigo no XXIII Simpósio de Ensino, Pesquisa e Extensão – SEPE no ano de 2018.

Para realizar a validação do software, os testes foram separados em dois tipos os testes de funcionalidades, onde os mesmos foram executados a partir do momento em que uma nova funcionalidade estava pronta. Foram aplicados testes mais técnicos, a nível de desenvolvimento, com o objetivo de encontrar possível falhas nas funcionalidades. E por último, foram realizados os testes de usabilidade, sendo que para a execução desses testes, voluntários utilizaram o aplicativo e ao final da experiência, forneceram um feedback do mesmo.

5.1. Testes de Funcionalidades

Todas as funcionalidades do aplicativo foram testadas de forma individual, com o intuito de identificar possíveis falhas. Dentre esses testes realizados, os que aqui serão ressaltados são os seguintes: funcionalidade realizar navegação e funcionalidade montar trilha.

Nos testes para a funcionalidade realizar navegação, foram aplicados alguns testes com diferentes tamanhos de arquivos, para identificar problemas no momento de renderizar as imagens. Com os mesmos, foi possível identificar a dificuldade que a aplicação possui para conseguir abrir uma imagem muito grande, onde é possível que apresente alguma lentidão na aplicação. Este problema não foi corrigido de forma direta, para não interferir na qualidade das imagens que o usuário escolher, sendo apenas exibido uma alerta para o mesmo, com o objetivo de orientá-lo.

Os testes realizados nas funcionalidades montar trilha, são semelhantes aos mesmos executados com a funcionalidade anterior, sendo que os mesmos diferem por serem aplicados quando o usuário seleciona as imagens da galeria e/ou diretório. Dessa forma, foi possível simular a seleção de múltiplas imagens de tamanhos diferentes. Os mesmos problemas foram detectados, uma lentidão no para realizar o tratamento de todas as imagens, caso as mesmas possuam um tamanho muito grande. O problema encontrado não foi tratado para não interferir na qualidade das imagens selecionadas, apenas é informado ao usuário que o mesmo pode ocorrer em decorrência do tamanho das imagens selecionadas.

5.2. Testes de Usabilidade

Para realizar os testes de usabilidade foram selecionados dois voluntários que utilizaram o aplicativo por um determinado período, e assim relataram suas impressões do mesmo. A pedido dessas pessoas, seus nomes verdadeiros não irão ser mencionados no trabalho, sendo que os mesmos serão referenciados como Adriano, para a primeira pessoa que utilizou o aplicativo e realizou os testes, e Diego para a segunda.

A primeira pessoa a realizar os testes e a validação do aplicativo foi o Adriano, após utilizar por alguns dias o aplicativo, ele relatou que todas as funcionalidades do aplicativo são bem simples e intuitivas, facilitando bastante a utilização do mesmo, e ressaltou principalmente o fato de poder escolher as próprias imagens que serão

reproduzidas. Isso acaba entregando ao usuário final uma maior liberdade no momento de construir a sua trilha, onde o mesmo pode escolher de fato onde gostaria de realizar a navegação, basta que o mesmo tenha as imagens em 360°.

O Diego foi a segunda pessoa a utilizar o aplicativo e assim realizar alguns testes no mesmo para validá-lo. Diego destacou o fato da aplicação obedecer um tempo determinado pelo próprio usuário no momento de realizar a navegação (reprodução), também comentou sobre a possibilidade de escolher a qualidade das imagens, ajudando principalmente os dispositivos que possuem pouco espaço para o armazenamento. Um ponto destaque pelo usuário, é de que o aplicativo seria “perfeito” se houvesse como compartilhar as trilhas com outras pessoas.

A validação do aplicativo foi realizada com base nos testes de usabilidade executados por esses dois usuários voluntários, sendo que os pontos destacados por eles estão dentro dos principais objetivos da aplicação. E o ponto apontado pelo usuário Diego, relativo ao compartilhamento das trilhas, ficará relatado como um trabalho futuro, sendo uma nova funcionalidade a ser adicionada.

5.3. Interfaces do Aplicativo

Toda a interface do aplicativo foi desenvolvida de maneira a ser básica e direta, e assim entregar ao usuário de maneira rápida, aquilo que realmente o aplicativo propõe, que é reproduzir as trilhas (imagens) em 360°. Dessa forma, todas as interfaces foram construídas de forma minimalista, sendo bem objetivas e de fácil entendimento por seus utilizadores.

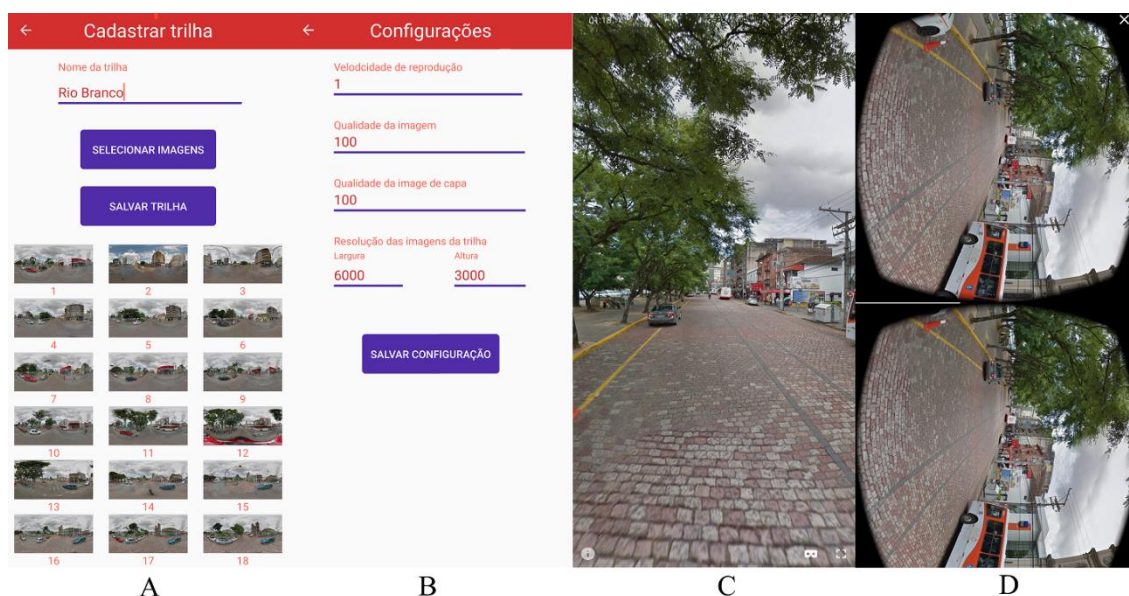


Figura 9. Interfaces: cadastro de trilhas, configurações, reprodução normal e reprodução em VR.

Na Figura 9 estão respectivamente as interfaces de cadastro das trilhas, configurações, reprodução normal e reprodução em VR, sendo que as funcionalidades das mesmas são as seguintes:

- interface de cadastro das trilhas (Figura 9 - A): essa interface é onde o usuário irá selecionar da galeria, ou outro local disponível, as imagens que ele irá utilizar na reprodução;

- interface de configuração (Figura 9 - B): como o próprio nome sugere, é nessa interface que o usuário irá escolher qual configuração lhe agrada mais. Essa configuração é salva em um arquivo JSON (*JavaScript Object Notation*);
- interface de reprodução normal (Figura 9 - C): esta interface irá reproduzir a trilha em formato padrão, ou seja, o usuário poderá observar todos os pontos da imagem, porém a mesma não está tratada para a reprodução em VR;
- interface de reprodução em VR (Figura 9 - D): se trata praticamente da principal interface do aplicativo, pois é na mesma que o usuário poderá utilizar os óculos VR, para que dessa forma consiga visualizar a imagem em 360° com a realidade virtual, e dessa forma se sentir mais imerso no ambiente escolhido.

O aplicativo possui outras interfaces, onde é possível realizar outras atividades, como visualizar as trilhas recém-criadas, e ou visualizar todas as trilhas (APÊNDICE H).

6. Problemas Encontrados

No decorrer do desenvolvimento do projeto, foram encontrados diversos problemas que acabaram mudando em partes a estrutura inicial do projeto. O principal problema encontrado foi o fato de não poder utilizar a API do Google Street View como provedor de imagens em 360° para o aplicativo. Ao longo de todo o processo de implementação muitos problemas ocorreram, e para se ter um melhor entendimento de tais problemas, os mesmos foram separados em dois tipos, problemas com a licença Google e problemas com a programação (codificação).

6.1. Problemas Com a Licença Google

Inicialmente um dos objetivos do projeto era utilizar as imagens do Street View, e a partir delas realizar a navegação (reprodução). No decorrer da implementação (codificação) do projeto, a Google foi consultada e dessa forma foi exposto para a mesma toda a ideia inicial do projeto, assim como o mesmo iria ser executado. Vários e-mails foram trocados com a empresa, onde a mesma deixou bem claro a impossibilidade de executar o trabalho dentro dos termos adotados por ela, uma vez que se o projeto continuasse da maneira com que foi elaborado, o mesmo iria se tornar ilegal.

Para que o trabalho que foi planejado inicialmente com a utilização da API do Street View fosse desenvolvido, o mesmo iria contra dois termos da licença de uso da API.

- 3.2.4 Restrições ao uso Indevido dos Serviços.
 - Proibição de Cópia: O Cliente não poderá extrair, exportar ou copiar Conteúdo do Google Maps para uso fora dos Serviços.
 - Proibição de Criação de conteúdo a partir do Google Maps: O Cliente não criará conteúdo com base no conteúdo do Google Maps [Cloud 2019].

Essas infrações foram ressaltadas no e-mail enviado pela Google, sendo sugerido a consulta de um advogado para tratar das questões legais. Como o projeto inicial iria armazenar as imagens localmente (no dispositivo), e posteriormente criar conteúdo com elas, a quebra desses dois termos da licença de uso se aplicam corretamente na concepção inicial do projeto. Dessa forma, a utilização da API do Google Street View foi removida do projeto, entregando ao usuário o controle total de como prover as imagens em 360° para serem executadas no aplicativo.

6.1. Problemas com a Programação

Os problemas no âmbito da programação foram surgindo no decorrer do desenvolvimento do projeto, onde os principais problemas foram no tratamento da reprodução das imagens com a Google VR SDK, sendo que as imagens deveriam ser trocadas conforme a configuração realizada pelo usuário, e também devido ao fato de que a documentação da SDK estar um pouco incompleta e desatualizada.

Para exibir as imagens na interface do aplicativo, também foram encontradas algumas dificuldades, pois na maioria das vezes as imagens eram muito grandes, e assim era preciso utilizar componentes robustos e que suportassem a exibição de imagens com um tamanho grande e uma qualidade alta.

Esses são os principais problemas deparados no desenvolvimento do trabalho, sendo que os mesmos foram identificados e corrigidos com antecedência na sua grande maioria por estar fazendo uso da codificação *Cowboy*, onde a mesma deu uma maior flexibilidade no desenvolvimento e dessa forma foi possível chegar rapidamente nos pontos críticos do projeto e corrigir os problemas encontrados.

7. Conclusão

Os estudos iniciais foram fundamentais para o bom desenvolvimento do projeto, mesmo após sofrer algumas mudanças para não ferir os termos de uso da Google, o trabalho não perdeu a sua essência, que é entregar ao usuário um aplicativo de fácil utilização e principalmente que ofereça uma imersão considerável nos novos meios tecnológicos e como consequência, nos ambientes que os usuários escolherem explorar.

A utilização de boas práticas das duas metodologias utilizadas nesse projeto (FDD e Codificação *Cowboy*) foram primordiais para o bom desenvolvimento do mesmo. Onde a FDD deu um suporte desde a concepção do projeto até o final, e a *Cowboy* auxiliou no momento da execução do projeto, ou seja, no momento de realizar a programação, ajudando de forma considerável na detecção e correção dos problemas.

Com a ausência do uso da API do Google Street View, o enfoque principal do trabalho passou para o uso da Google VR SDK, uma vez que a mesma é utilizada para realizar a navegação em 360° nas imagens selecionadas pelo usuário. Com isso o uso de tal SDK se torna a principal ferramenta deste trabalho, onde a mesma é responsável pela execução da principal funcionalidade do trabalho, que é também a origem do mesmo.

Os testes executados pelos voluntários foram essenciais para realizar a validação do projeto, e assim foi possível identificar que o aplicativo está cumprindo com aquilo que foi proposto, e entregando aos seus usuários uma maior imersão com a tecnologia principalmente, contribuindo consideravelmente com a integração entre as pessoas e a Realidade Virtual.

Para trabalhos futuros sugere-se a utilização da API de comando de voz da Google, para realizar a troca de imagens do aplicativo, e assim dar uma maior liberdade e flexibilidade ao usuário, oferecendo também um maior controle sobre a aplicação. A criação de uma versão para IOS do aplicativo também é válida, uma vez que existem muitos usuários que possuem os dispositivos da empresa, e dessa forma será possível atingir um maior número de usuários, integrando cada vez mais os mesmos com a tecnologia (VR).

7. Referências Bibliográficas

- Ableson, Frank; King, Chris; SEN, Robi. (2012) "Android em ação". Elsevier Brasil.
- Álvares, Carlos Miguel. Vídeos interativos e imersivos no sight surfers. Diss. 2012.
- Android, Developers. (2019) "Conheça o Android Studio", https://developer.android.com/studio/intro#a_interface_do_usu%C3%A1rio. Acesso em: abril de 2019.
- Caldeira, Vagner, Santos, Gabriela, and Soares, Rodrigo Everton. "Metodologias Ágeis". (2013).
- Canto, Tânia Seneme do. "A cartografia na era da cibercultura: mapeando outras geografias no ciberespaço." (2010): 120-f.
- Cardoso, Phillipe Valente; Santos, K. da Silva. "Realidade Virtual e Geografia: O Caso do Google Cardboard Glasses para o Ensino." Revista Tamoios 11.2 (2015).
- Darken, Rudolph Poul. Wayfinding in large-scale virtualworlds. Diss. George Washington University, 1996.
- Cloud, Google. "Termos de Serviço da Google Maps Platform". Disponível em: <https://cloud.google.com/maps-platform/terms/>. Acesso em: outubro de 2019.
- Folts, Brian. (2017). "Street View Player". Disponível em: <http://www.brianfolts.com/driver/>. Acesso em: abril de 2019.
- Hollar, B. (2006). "Cowboy: An Agile Programming Methodology for a Solo Programmer". Disponível em: <https://scholarscompass.vcu.edu/etd/741/?mode=full>.
- Indrusiak, Leandro Soares. (1996). "Linguagem Java". Disponível em: <https://www.cin.ufpe.br/~arfs/introjava.pdf>.
- Kirner, Claudio, and Siscoutto, Robson. "Realidade virtual e aumentada: conceitos, projeto e aplicações." Livro do IX Symposium on Virtual and Augmented Reality, Petrópolis (RJ), Porto Alegre: SBC. 2007.
- Marçal, E., Andrade, R., & Rios, R. (2005). Aprendizagem utilizando dispositivos móveis com sistemas de realidade virtual. RENOUE, 3(1).
- Pereira, Lucio Camilo Oliva, and Da Silva, Michel Lourenço. Android para desenvolvedores. Brasport, 2009.
- Silva, F. G., Hoentsch, Sandra C. P., and Silva, Leila. "Uma análise das Metodologias Ágeis FDD e Scrum sob a Perspectiva do Modelo de Qualidade MPS. BR." Scientia Plena 5.12 (2009).
- Xu, Ya-jing, et al. "The Design and Implementation of Chinese Vocabulary Learning Case Based on Mobile VR for "The Belt and Road"." DEStech Transactions on Computer Science and Engineering cmsam (2017).

APÊNDICES

APÊNDICE A. Diagrama de Atividades.

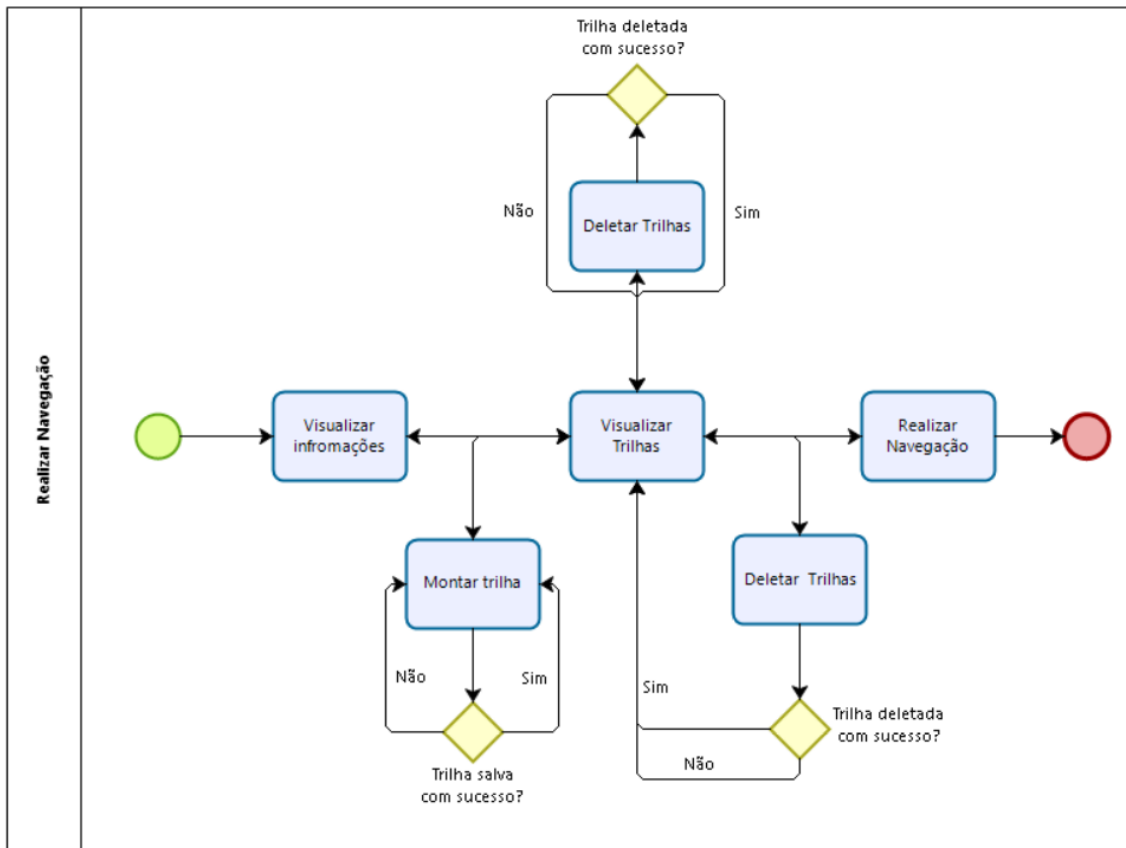


Figura 3. Diagrama de Atividades

APÊNDICE B. Requisitos Funcionais.

Tabela 1. Requisitos Funcionais

Requisitos Funcionais		
Funcionalidade	Descrição	Complexidade
RF01: Selecionar Trilha	Permite a escolha de uma trilha para ser reproduzida.	Baixo
RF02: Montar Trilha	Permite a seleção das imagens que irão compor a trilha.	Médio
RF03: Configurar Navegação	Permite realizar as configurações da reprodução das trilhas.	Médio
RF04: Realizar Navegação	O aplicativo irá realizar a navegação.	Alto

RF05: Visualizar Informações	O aplicativo permite visualizar as informações sobre ao aplicativo.	Baixo
RF06: Deletar Trilha	Permite que o usuário delete uma ou mais trilhas.	Médio

APÊNDICE C. Requisitos Não Funcionais.

Tabela 2. Requisitos Não Funcionais

Requisitos Não Funcionais	
Funcionalidades	Descrição
RNF01	O aplicativo será desenvolvido para a plataforma Android.
RNF02	O aplicativo será desenvolvido na linguagem de programação Java
RNF03	O aplicativo deverá fazer uso da Google VR SDK.
RNF04	A aplicação deverá ser desenvolvida na IDE Android Studio.

APÊNDICE D. Estimativa do tempo de implementação das funcionalidades.

Tabela 3. Estimativa do tempo de implementação das funcionalidades.

Código	Funcionalidade	Tempo (Horas)		
		Implementação	Testes	Total / Requisitos
-	-			
RF01	Selecionar Trilha	14	6	20
RF02	Montar Trilha	41	19	60
RF03	Configurar Navegação	22	8	30
RF04	Realizar Navegação	56	24	80
RF05	Visualizar informações	5	2	7
RF06	Deletar Trilha	30	13	43
TOTAL	-	168	72	240

APÊNDICE E. Diagrama de Casos de Uso.

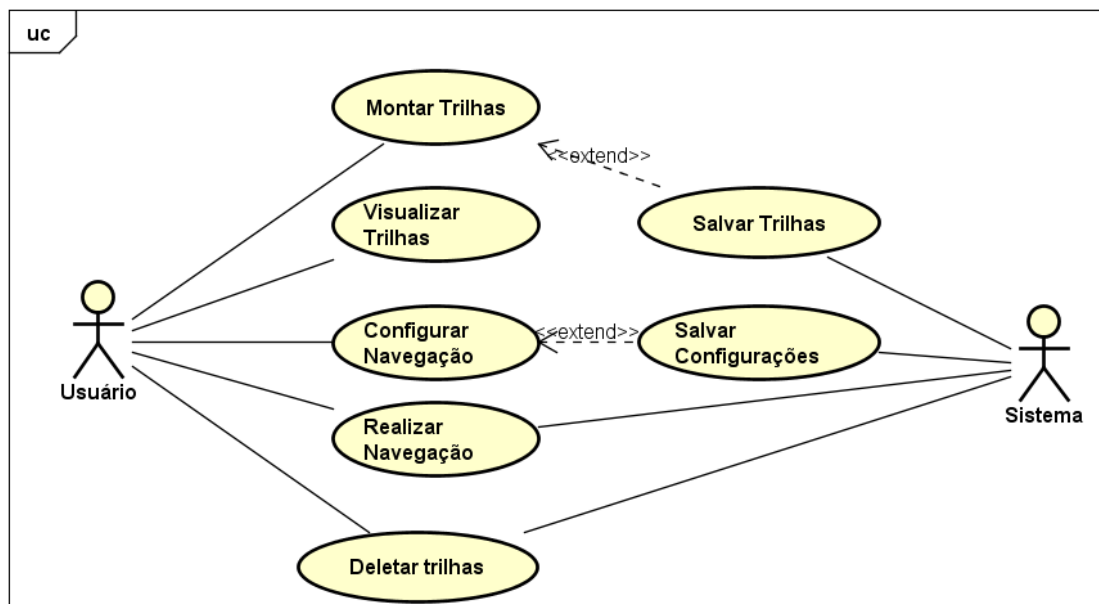


Figura 4. Diagrama de Casos de Uso

APÊNDICE F. Descritivo do Caso de Uso Realizar Navegação.

Tabela 4. Descritivo do Caso de Uso Realizar Navegação

Identificação	UC1
Caso de uso	Realizar Navegação
Descrição	Este caso de uso é responsável por realizar a navegação na aplicação, sendo que o mesmo irá ser executado após todas as configurações já terem sido realizadas.
Ator Principal	Usuário
Ator(es) Secundário(s)	Sistema
Pré-condições	
PC1	O usuário deve ter configurado corretamente a navegação.
Pós-condições	
PO1	Navegação realizada com sucesso.
PO2	Navegação Interrompida.
Fluxo Principal	
FP1	O usuário pode escolher uma das seguintes ações: Prosseguir para navegação [FA1]; Interromper a navegação [FA2];
FP2	Fim do caso de uso.
Fluxo Alternativo	
FA1	Se o usuário escolher a opção “Prosseguir para a navegação”:
1.1.	O sistema irá realizar as últimas verificações [FE1].
1.2.	O sistema começa a navegação.
1.3.	Durante a navegação o usuário poderá escolher a qualquer momento interromper a navegação [1.7.].

1.4.	O sistema finaliza a navegação.
1.8.	O sistema retorna ao [FP1].
FA2	Se o usuário escolher a opção “Interromper a navegação”:
2.1.	O sistema realiza alguns tratamentos internos.
2.2.	O sistema retorna para o [FP1]
Fluxos de Exceções	
FE1	Erro ao iniciar navegação.
1.1.	O sistema exibe o erro ocorrido.
1.2.	O sistema retorna para o fluxo correspondente [FA1].
Regras de Negócio	N/A
Casos de Uso Estendidos	N/A
Casos de Uso Incluídos	N/A

APÊNDICE G. Descritivo do Caso de Uso Configurar Navegação.

Tabela 5. Descritivo do Caso de Uso Configurar Navegação

Identificação	UC2
Caso de uso	Configurar Navegação
Descrição	Este caso de uso é responsável pela configuração da navegação, sendo que o mesmo será capaz de realizar os ajustes solicitados pelo o usuário.
Ator Principal	Usuário
Ator(es) Secundário(s)	N/A
Pré-condições	
PC1	N/A
Pós-condições	
PO1	Configurações realizadas com sucesso.
PO2	Erro ao realizar a configuração.
Fluxo Principal	
FP1	O usuário acessa a interface de configuração da navegação.
1.1.	O sistema exibe as opções a serem configuradas.
1.2.	O usuário visualiza as opções.
1.3.	O usuário realiza as configurações necessárias para ocorrer a navegação.
1.4.	O sistema valida os dados informados pelo usuário [FE1]
1.5.	O sistema informa ao usuário que as configurações foram aceitas.
1.6.	Fim do caso de uso.
Fluxo Alternativo	
-	N/A
Fluxos de Exceções	
FE1	Dados inválidos.
1.1.	O sistema exibe uma mensagem de dados inválidos.
1.2.	O sistema informa quais dados estão inválidos.
1.3.	O sistema retorna para o fluxo correspondente.
Regras de Negócio	N/A
Casos de Uso Estendidos	N/A.
Casos de Uso Incluídos	N/A

APÊNDICE H. Interfaces: carregamento, visualizar informações e lista de trilhas.

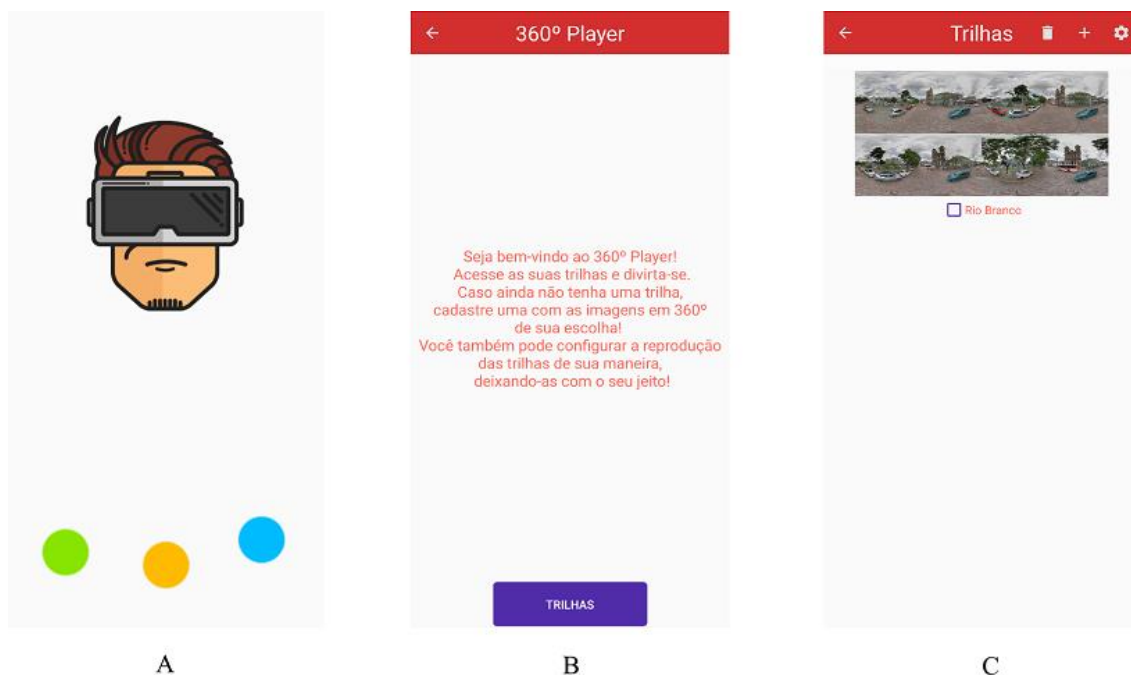


Figura 8. Interfaces: carregamento, visualizar informações e lista de trilhas.

Como pode ser observado na Figura 8, as interfaces iniciais do aplicativo são respectivamente:

- interface de carregamento (Figura 8 - A): onde é realizada algumas configurações iniciais para a utilização da aplicação;
- interface visualizar informações (Figura 8 - B): esta interface exibi as últimas trilhas adicionadas, assim como a opção de montar uma nova trilha;
- interface da lista de trilhas (Figura 8 - C): esta interface exibe todas as trilhas cadastradas pelo usuário, onde o mesmo poderá selecionar uma para executar, também dá acesso as interfaces de configurações e de cadastro de uma nova trilha.