



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

TALITA NOGALES MARQUES

**ESTUDO EXPLORATÓRIO EM MATERIAL DESIGN E
PLATAFORMA ANDROID COM APOIO À EXPERIÊNCIA DO
USUARIO**

Assis/SP

2017

TALITA NOGALES MARQUES

**ESTUDO EXPLORATÓRIO EM MATERIAL DESIGN E
PLATAFORMA ANDROID COM APOIO À EXPERIÊNCIA DO
USUARIO**

Orientador: Prof. MSc. Guilherme de Cleve Farto

Nota do orientador:	Nota do avaliador:
---------------------	--------------------

Assis/SP

2017



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ESTUDO EXPLORATÓRIO EM MATERIAL DESIGN E PLATAFORMA ANDROID COM APOIO À EXPERIÊNCIA DO USUARIO

TALITA NOGALES MARQUES

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, avaliado
pela seguinte comissão examinadora:

Orientador: _____ Prof. MSc. Guilherme de Cleva Farto

Examinador: _____ Prof.MSc. Fernando Cesar de Lima

Assis/SP
2017

FICHA CATALOGRÁFICA

S247a MARQUES, Talita Nogales

Estudo exploratório em Material Design e Plataforma Android
com apoio à Experiência do Usuário/Talita Nogales Marques. – Assis,2017.

77p.

Trabalho de conclusão do curso (Ciência da Computação). –
Fundação Educacional do Município de Assis-FEMA

Orientador: MSc. Guilherme de Cleve Farto

1.Android 2.Design 3.Design Thinking 4.User experience

CDD 005.8

DEDICATÓRIA

Dedico este trabalho à Deus, a toda minha família e principalmente aos meus pais, por terem me dado todo carinho e melhor Educação possível e serem um grande exemplo de pessoas. Dedico também ao meu namorado e meu orientador por estar Sempre me apoiando para a realização deste meu sonho.

AGRADECIMENTOS

Primeiramente agradeço a **Deus**, por ter me dado a oportunidade de estar viva e realizando o sonho de me formar e dar forças nas dificuldades.

Agradeço à minha mãe **Aparecida Penha Nogales Alvares Marques** e meu pai **Adnilson Aparecido Marques** que estão sempre ao meu lado me apoiando, orientando e me preparando para a vida.

Ao meu namorado, **Fernando Leite**, por me animar sempre, com seu otimismo deixando meus dias mais alegres e belos.

Ao meu amigo e professor MSc. **Guilherme de Cleva Farto**, por me orientar e colaborar nesta fase acadêmica tão importante.

A todos os meus **professores** destes quatro anos de curso, que ajudaram em minha formação não só acadêmica, mas também como pessoa.

E por fim agradeço a todos que colaboraram de forma direta ou indiretamente para a execução deste trabalho.

Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.

Charles Chaplin

RESUMO

A tecnologia dos smartphones vem crescendo cada vez mais em um ritmo acelerado. Desta forma, a tecnologia está sendo aprimorada, contribuindo com diversas áreas, inclusive no ambiente de trabalho que tem evoluído constantemente com o uso de aplicações móveis. O grande desafio hoje é trazer a inovação necessária para fazer isso acontecer, principalmente devido ao pouco entendimento que se tem sobre o que é o processo de inovação baseado na experiência do usuário e de que forma pode ser utilizado pela empresa. Partindo disso, o Design Thinking (DT) surge para tentar desenvolver um modelo mental e uma nova visão que proporcione a busca de solução de problemas e acelere o processo de inovação de forma coletiva. Assim, busca-se neste trabalho demonstrar a utilização do Design Thinking como abordagem para criação de serviços. Esta pesquisa conduz um estudo exploratório sobre Material Design na plataforma Google Android, bem como o uso da abordagem de User Experience e Design Thinking para buscar inovação em projetos.

Palavras-chave: Inovação, Design Thinking, User Experience, Experiência

ABSTRACT

The technology of smartphones is growing more and more at a fast pace. In this way, the technology is being improved, contributing to several areas, including the work environment that has constantly evolved with the use of mobile applications. The big challenge today is to bring the innovation needed to make this happen, mainly due to the lack of understanding about the innovation process based on user experience and how it can be used by the company. From this, Design Thinking (DT) arises to try to develop a mental model and a new vision that provides the search for problem solving and accelerates the process of innovation in a collective way. Thus, this work aims to demonstrate the use of Design Thinking as an approach to service creation. This research leads an exploratory study on Material Design in the Google Android platform, as well as the use of the User Experience and Design Thinking approach to seek innovation in projects

Keywords: Innovation, Design Thinking, User Experience, Experience

LISTA DE FIGURAS

Figura 1 Tema escuro do Material e Tema Claro do Material (ANDROID DEVELOPER, 2014).	20
Figura 2 Widgets RecyclerView (ANDROID DEVELOPER, 2014).	21
Figura 3 Lista com RecyclerView(ANDROID DEVELOPER, 2014).....	22
Figura 4 Exemplo de cartões (ANDROID DEVELOPER, 2014).	24
Figura 5 Sombras para diferentes elevações de visualizações (ANDROID DEVELOPER, 2014)	27
Figura 6 Transição de cena com um elemento compartilhado (ANDROID DEVELOPER, 2014).	32
Figura 7 Aplicativos do Android representados pelo Bugdroid, (In: THE MOVIE THEME SONG, 2015).	35
Figura 8 Java VM Versus Dalvik VM	37
Figura 9 Ambiente de desenvolvimento Android Studio	38
Figura 10 Emulador do Android.....	39
Figura 11 Arquitetura Google Android.....	40
Figura 12 Android Studio	43
Figura 13 Fases do UX.....	45
Figura 14 UX Durante a Interação (ROTO,2007)	46
Figura 15 Double Diamond (DESIGN COUNCIL 2015, P. 15)	50
Figura 16 Imersão no cotidiano do usuário (ESCOLA DE DESIGN THINKING 2016)	51
Figura 17 Fase de análise e síntese (VIANNA ET AL.2012)	52
Figura 18 Fase de ideação (Vianna et al. 2012).....	52
Figura 19 Processo de prototipação (MJV,2014)	53
Figura 20 Esquema representativo das etapas do processo de Design Thinking	58
Figura 21 Diamante (COUNCIL, 2007).....	60
Figura 22 Modelo de Negócio Canvas Fonte: Osterwalder, Pigneur (2010)	64
Figura 23 Canvas: Front-Stage -blocos que dependem do segmento dos clientes Fonte: Adaptado de Osterwalder, Pigneur (2010).....	67
Figura 24 Canvas: Back-Stage-blocos que não dependem do segmento dos clientes Fonte: Adaptado de Osterwalder, Pigneur (2010).....	68
Figura 25 Modelo de Negócios Canvas e suposições de hipóteses.....	70

LISTAGEM

Listagem 1 Código de modificação de cores (ANDROID DEVELOPER, 2014).....	21
Listagem 2 Adicionando RecyclerView (ANDROID, 2014).....	23
Listagem 3 Widgets RecyclerView (ANDROID DEVELOPER, 2014).	23
Listagem 4 Código Widgets CardView (ANDROID DEVELOPER, 2014).	25
Listagem 5 Biblioteca do RecyclerView e CardView (ANDROID DEVELOPER, 2014).	25
Listagem 6 Código desenhável de segundo plano (ANDROID DEVELOPER, 2014).	28
Listagem 7 Definindo retângulo com bordas (ANDROID DEVELOPER, 2014);	28
Listagem 8 Revelar uma visualização anteriormente invisível (ANDROID DEVELOPER, 2014).	30
Listagem 9 escondendo visualização anteriormente visível (ANDROID DEVELOPER, 2014).	31
Listagem 10 Código de transições (ANDROID DEVELOPER, 2014).	33
Listagem 11 Transição (ANDROID DEVELOPER, 2014).	33

LISTA DE TABELAS

Tabela 1 Principais bibliotecas da Camada Libraries (FARIA,2008).....	42
--	----

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVOS	16
1.2 JUSTIFICATIVAS.....	17
1.3 MOTIVAÇÃO.....	17
1.4 ESTRUTURA DO TRABALHO.....	18
2 MATERIAL DESIGN	19
2.1 MATERIAL DESIGN PARA ANDROID.....	19
2.2 TEMA DO MATERIAL DESIGN	20
2.3 LISTAS E CARTÕES.....	21
2.3.1 LISTA	21
2.3.2 CARTÕES.....	24
2.3.3 ADICIONAR DEPENDÊNCIAS.....	25
2.4 SOMBRAS E CONTORNOS DE VISUALIZAÇÃO.....	26
2.4.1 DEFININDO SOMBRAS E RECORTAR VISUALIZAÇÕES	26
2.4.2 ATRIBUINDO ELEVAÇÃO E VISUALIZAÇÃO	26
2.4.3 PERSONALIZAR SOMBRAS E CONTORNOS DE VISUALIZAÇÃO	27
2.5 ANIMAÇÕES.....	28
2.5.1 DEFININDO ANIMAÇÕES PERSONALIZADAS	29
2.5.2 FEEDBACK DE TOQUE	29
2.5.3 EFEITO DE REVELAÇÃO	30
2.5.4 TRANSIÇÕES DE ATIVIDADES.....	31
2.5.5. ESPECIFICANDO TRANSIÇÕES DE ATIVIDADES	32
3 PLATAFORMA GOOGLE ANDROID	34
3.1 INTRODUÇÃO AO ANDROID	34
3.2 OPEN HANDSET ALLIANCE (OHA).....	35
3.3 SISTEMA OPERACIONAL LINUX.....	36
3.4 MAQUINA VIRTUAL DALVIK.....	36
3.5 AMBIENTE DE DESENVOLVIMENTO	38

3.6 ANDROID SDK.....	39
3.7 PLATAFORMA ANDROID	40
3.7.1 CAMADA LINUX KERNEL	40
3.7.2 CAMADA ANDROID RUNTIME	41
3.7.3 CAMADA LIBRARIES.....	41
3.7.4 CAMADA APPLICATION FRAMEWORK	42
3.7.5 CAMADA APPLICATIONS.....	42
4 UX E DESIGN THINKING	44
4.1 CONTEXTUALIZAÇÃO EM EXPERIÊNCIA DO USUÁRIO	44
4.2 CONTEXTUALIZAÇÃO EM DESIGN THINKING	47
4.2.1. ORIGENS	47
4.2.2. VALORES.....	48
4.2.3. PROCESSO	50
4.3. INOVAÇÃO E DESIGN THINKING	53
4.4. DESIGN DE SERVIÇOS	54
4.5. MODELOS DE DESIGN: MVP E MVS.....	55
4.5.1. MINIMUM VIABLE PRODUCT (MVP)	55
4.5.2. MINIMUM VALUABLE SERVICE (MVS).....	56
4.7. ETAPAS DE DESIGN	58
4.7.1. CONTEXTUALIZAÇÃO NO "DIAMANTE DE DESIGN"	59
4.7.2 IMERSÃO.....	60
4.7.3. IDEAÇÃO	61
4.7.4. PROTOTIPAÇÃO	62
4.7.5. VALIDAÇÃO.....	63
4.8. MODELO DE NEGÓCIOS CANVAS	63
5. KIT DE FERRAMENTAS DESIGN THINKING	71
6 CONCLUSÃO.....	72
6.1 TRABALHOS FUTUROS.....	73
REFERÊNCIAS.....	74

1 INTRODUÇÃO

A tecnologia dos *smartphones* vem crescendo cada vez mais em ritmo acelerado. Desta forma, a tecnologia tem sido aprimorada rapidamente, contribuindo com diversas áreas, inclusive no ambiente de trabalho que tem se beneficiado com os conceitos de conectividade e mobilidade. Com o surgimento do Google Android, o mercado tecnológico sofreu um grande avanço, por meio das facilidades do uso de dispositivo móveis que pode ser utilizado para facilitar diversas tarefas do dia a dia. Com tais avanços, as pessoas conseguem acesso à Internet em qualquer lugar por meio de um aparelho móvel e uma conexão de rede.

O Android é uma plataforma de desenvolvimento de software com código aberto que foi criada para dispositivos móveis como celular e tablets. O princípio do Android é criar um desenvolvimento aberto disponível para operadoras e fabricantes de software e hardware para dispositivos móveis. A plataforma Android foi originada por um grupo de empresas conhecida como *Open Handset Alliance*, que é liderado pela Google. OHA, trata-se de uma aliança de telefone celular aberto que é composta por 30 empresas do mercado de TI que apoiam a utilização de código aberto, podemos citar algumas empresas que estão investido nessa plataforma de código aberto, Samsung, Intel, Motorola, Qualcomm e Telefónica, investindo uma grande porcentagem no Android, para buscar engenheiros para um melhor aperfeiçoamento do seu dispositivo (OPEN HANDSET ALIANCE, 2007; FARIA, 2008).

Diante de tanto crescimento, a Google desenvolveu um conceito para aprimorar e padroniza o design visual do Android. Na conferência da Google I/O em 2014 foi anunciado a interface Material Design objetiva unificar e orientar os desenvolvedores na construção visual de aplicações Android. Segundo o designer Matias Duarte, “ao contrário do papel real, nosso material digital pode se expandir e reformar inteligentemente, o material tem superfícies físicas, bordas, costuras e sombras fornecem significados sobre o que você pode tocar”. A ideia é uma linguagem de design que facilite o desenvolvimento de interfaces pelos desenvolvedores (MATERIAL DESIGN, 2014).

O Material Design pode ser utilizado nos componentes com o Android 5 e o mais recente via *V7 appcompat biblioteca*, é utilizado praticamente para todos os

dispositivos Android fabricados após 2009. O Android fornece os seguintes elementos para criação de aplicativos Material Design, novo tema, que fornece estilo para o aplicativo Widgets de sistema construídos que permitem ao desenvolver rapidamente. Também inclusos são as animações padronizadas para feedback de toques e outras transições de estilos (ANDROID DEVELOPER, 2014).

Com a expansão das indústrias tecnológicas, estão sendo necessitado a criação de novos softwares mais complexos, mas com custo estimáveis e tempos curtos para o desenvolvimento. Temos como propostas o design baseado em componentes que irá abranger uma série de estratégias, processos e práticas. Utilizando um componente dentro de uma implementação de software quando executado dentro de um dispositivo físico ou lógico, devendo assim obedecer umas determinadas regras do modo que de uma maneira pratica onde realizamos a reutilizar de códigos em outros ambientes de desenvolvimento (SILVA, 2006).

1.1 OBJETIVOS

O objetivo geral desta pesquisa é realizar um estudo exploratório em tópicos de Material Design, uma especificação orientada a interfaces de aplicações móveis, bem como nos principais conceitos da plataforma Google Android. Também se destaca uma revisão exploratória sobre User Experience (UX), ou Experiência do Usuário, juntamente com a metodologia de Design Thinking.

Tais temas compõem um conjunto de conceitos que têm sido estudados por empresas na busca de inovação centrada no usuário, envolvendo-o em diferentes etapas da pesquisa, do desenvolvimento, da prototipação e da experimentação de novas tecnologias, produtos e processos.

Como resultado, espera-se contribuir com a elaboração de uma base fundamental para apoiar pesquisas futuras que contemplem, principalmente, UX e Design Thinking para apoiar a inovação e o desenvolvimento de novas soluções.

1.2 JUSTIFICATIVAS

Com a expansão das indústrias tecnológicas, estão sendo necessitado a criação de novos softwares, em vista dos anos anteriores, estamos em um século em que os *smartphones* são utilizados em todo o mundo. Com evolução da tecnologia necessitamos mais dos aplicativos dentro do *smartphone* para meios de comunicação, transporte e estudos. A pesquisa será para conhecimentos de novas ferramentas que poderá auxiliar no desenvolvimento em Android, como exemplo o banco de dados Firebase, que foi comprado pela Google no ano de 2014.

1.3 MOTIVAÇÃO

O desenvolvimento deste projeto de pesquisa consiste no fato da evolução da tecnologia da plataforma Android, sendo um tema que está em constante crescimento. Além de plataforma, padrões sendo mais estudados atualmente nesta área de tecnologia.

Outra motivação é a utilização do Material Design, para ter um visual composto com cores vibrantes, fontes, ícones, esse material já é utilizado no Gmail e o Google Calendário, que dá a profundidade entre seus aplicativos construindo suas camadas, além de produzir animações em aplicação utilizada. Essas alterações consistem em um conceito de espaços relacionados e sistema em movimento, criado pela Google. O Material Design tem sido utilizado em diversas aplicações móveis da plataforma Android, pela Google como, por exemplo, Google Maps.

Esta pesquisa também se motiva pelo aumento no interesse das empresas e da própria academia, como em universidades, por tópicos de inovação. Este trabalho explora os principais elementos de UX e da metodologia de Design Thinking.

1.4 ESTRUTURA DO TRABALHO

Este trabalho será estruturado nas seguintes partes:

- **Capítulo 1 – Introdução**
- **Capítulo 2 – Material Design**
- **Capítulo 3 – Plataforma Google Android**
- **Capítulo 4 – UX e Design Thinking**
- **Capítulo 5 – Kit de Ferramentas Design Thinking**
- **Capítulo 6 – Conclusão**
- **Referências**

2 MATERIAL DESIGN

Neste capítulo, serão explorados e apresentados os principais conceitos e características que se referem a Material Design, uma estratégia para apoiar a concepção visual no desenvolvimento de aplicações.

2.1 MATERIAL DESIGN PARA ANDROID

O Material Design pode ser utilizado nos componentes com o Android 5 e o mais recente via V7 appcompat biblioteca, é utilizado praticamente para todos os dispositivos Android fabricados após 2009. O Android fornece os seguintes elementos para criação do aplicativos Material Design, novo tema, que fornece estilo para o aplicativo Widgets de sistema construídos que permitem ao desenvolver rapidamente e permitirá definir uma paleta de cores. Também inclusos suas animações padronizadas para feedback de toques e outras transições de estilos de atividades, vistas complexas, e novas APIS para sombra e animações (ANDROID DEVELOPER, 2014; MATERIAL DESIGN, 2014).

O Material Design é um guia que vem evoluindo encima de um Design Visual, diversas plataformas e dispositivos tendo em vista que o Android é compatível com essa aplicação. A Google desenvolveu uma linguagem que aprimorou o design visual do Android, na conferência da Google I/O em 25 de julho de 2014 foi anunciado a interface Material Design que irá unificar o visual do Android. Segundo o Designer Matias Duarte, explicou que “ao contrário do papel real, nosso material digital pode se expandir e reformar inteligentemente, o material tem superfícies físicas, bordas, costuras e sombras fornecem significados sobre o que você pode tocar” (ANDROID DEVELOPER, 2014; MATERIAL DESIGN, 2014).

A ideia de acordo com o Matias Duarte vice-presidente de Design o Google forneceu uma linguagem de design que permitirá uma sensação de caneta e papel. Design de material oferece bordas físicas e superfícies para trabalhar com costuras e sombras, dando o contexto onde o design digital pode ser tocado, com isto o Android fornece

os elementos para criação de aplicativos utilizando o Material Design (ANDROID DEVELOPER, 2014; MATERIAL DESIGN, 2014).

2.2 TEMA DO MATERIAL DESIGN

O tema do Material Design fornece um novo estilo ao aplicativos para nossos usuários de smartphones, utilizando Widgets de sistemas que permitem definir a paleta de cores e as animações padrão para feedback e suas transições de atividades. Sendo possível personalizar a aparência de tema do Material com uma paleta de cores assim podendo controlar conforme a sua identidade. Podendo atribuir cores a barra de ações e a barra de status utilizando os atribuídos de temas conforme ilustrado na Figura 1 (ANDROID DEVELOPER, 2014).

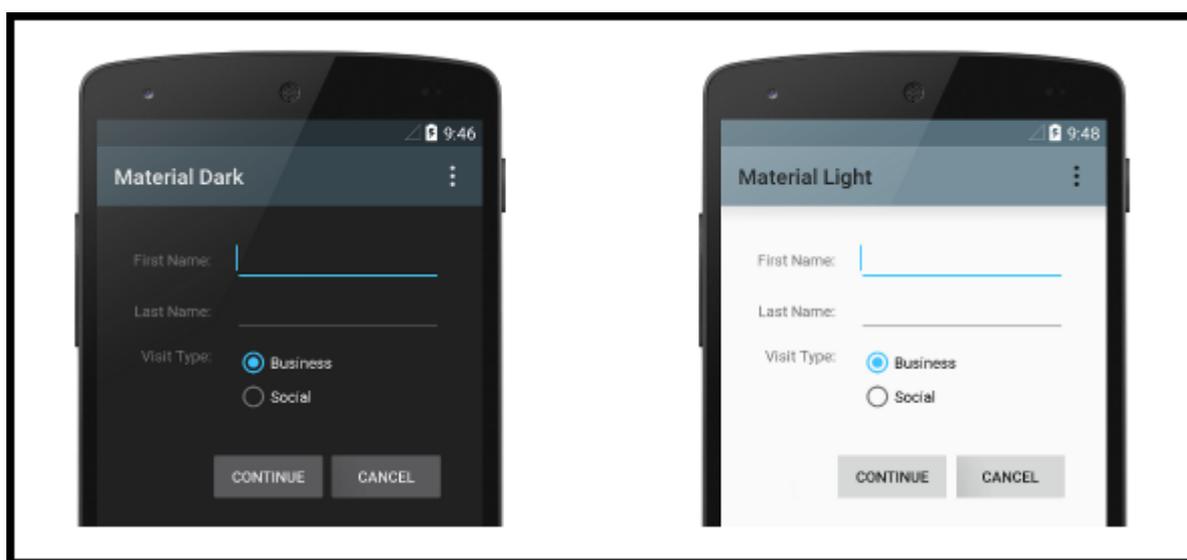


Figura 1 Tema escuro do Material e Tema Claro do Material (ANDROID DEVELOPER, 2014).

Na Listagem 1 é apresentado um fragmento de código que ilustra a modificação das paletas de cores de tons escuros e claros.

```
• @android:style/Theme.Material (versão escura)  
• @android:style/Theme.Material.Light (versão clara)  
• @android:style/Theme.Material.Light.DarkActionBar
```

Listagem 1 Código de modificação de cores (ANDROID DEVELOPER, 2014).

2.3 LISTAS E CARTÕES

Serão exploradas e atribuído dentro do Material Design os temas Lista e Cartões, que possui suas classes para definição de Layout dentro do aplicativo móvel.

2.3.1 LISTA

Para criação de lista temos o Widget RecyclerView que é uma versão completa de ListView tendo suportes de diferentes Layouts e melhorias de desempenho. Com essa classe podemos simplificar a exibição e tratamentos de grandes conjuntos de dados oferecendo ao desenvolvimento, gerenciadores de layout para posicionar itens e animações padrões para operações de item em comuns, como seu exemplo de remoção ou adições de itens, podendo também ter a flexibilidade de definir gerenciadores de Layout e animações personalizadas pelo Widgets RecyclerView, na Figura 2 ilustrada uma Widgets RecyclerView (ANDROID DEVELOPER, 2014).

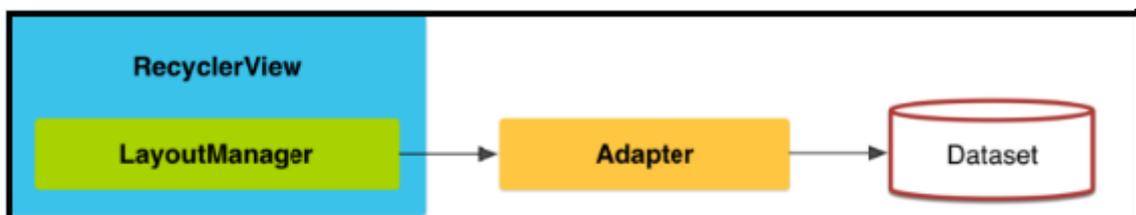


Figura 2 Widgets RecyclerView (ANDROID DEVELOPER, 2014).

Utilizando o Widgets RecyclerView, pode-se desenvolver um adaptador e um gerenciado de Layout. Desta forma, pode-se criar um adaptado, ampliando a classe RecyclerView.Adapter, para os detalhes de implementação irá depender da especificação do conjunto de dados e do tipo de visualização.

O gerenciador de layout posiciona uma visualização dentro de uma RecyclerView e determina também, como podemos reutilizar a visualização dentro de um item que não está mais visível para um usuário para reutilizar ou reciclar uma visualização, um gerenciador de layout pode solicitar ao adaptador a substituição do conteúdo da visualização com um elemento diferente do conjunto de dados. Visualizações recicladas dessa maneira aprimoram o desempenho ao evitar a criação de visualizações desnecessárias ou a realização de pesquisas. Na Figura 3 ilustra a lista utilizando RecyclerView (ANDROID DEVELOPER, 2014).

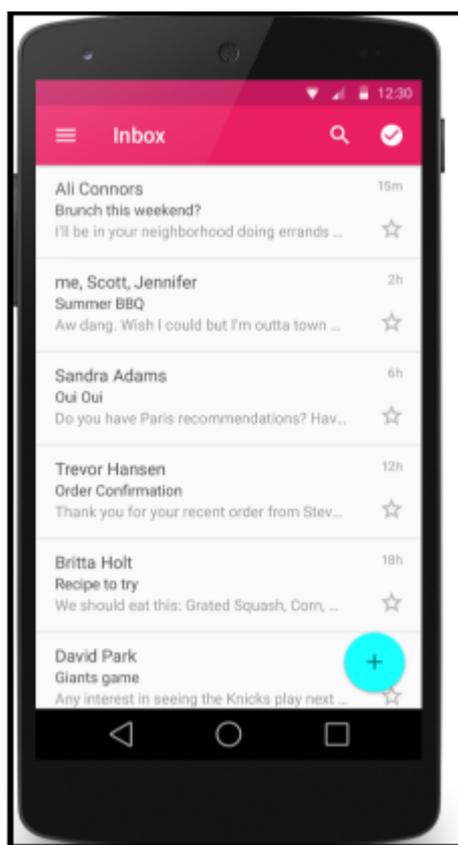


Figura 3 Lista com RecyclerView(ANDROID DEVELOPER, 2014).

Na Listagem 2 ilustra em códigos como adicionar a RecyclerView.

```

<!-- A RecyclerView with some commonly used attributes -->
<android.support.v7.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

```

Listagem 2 Adicionando RecyclerView (ANDROID, 2014).

Logo após demonstrarmos como adicionar a RecyclerView na Listagem 3 iremos demonstrar como adicionar o Widgets RecyclerView junto ao seu layout.

```

public class MyActivity extends Activity {
    private RecyclerView mRecyclerView;
    private RecyclerView.Adapter mAdapter;
    private RecyclerView.LayoutManager mLayoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.my_activity);
        mRecyclerView = (RecyclerView) findViewById(R.id.my_recycler_view);

        // use this setting to improve performance if you know that changes
        // in content do not change the layout size of the RecyclerView
        mRecyclerView.setHasFixedSize(true);

        // use a linear layout manager
        mLayoutManager = new LinearLayoutManager(this);
        mRecyclerView.setLayoutManager(mLayoutManager);

        // specify an adapter (see also next example)
        mAdapter = new MyAdapter(myDataset);
        mRecyclerView.setAdapter(mAdapter);
    }
    ...
}

```

Listagem 3 Widgets RecyclerView (ANDROID DEVELOPER, 2014).

2.3.2 CARTÕES

Dentro do Material Design contendo também a estrutura de criação de cartões esse Widgets tem seu nome CardView, permite exibir informações importantes dentro de cartões que contem aparência consistente. CardView, é ampliada por uma classe Frame Layout onde permite que usuários exiba informações dentro dos cartões, que contem aparência coerente nas diversas plataformas. Utilizando o Widgets podendo também ter sombras e bordas arredondadas.

Para criação de um cartão com sombra, utilizamos o atributo CardView: cardElevation. CardView usa uma elevação real e sombras dinâmicas no Android 5.0 (API de nível 21), e posteriores voltada para uma implementação de sombra programática em versões anteriores (ANDROID DEVELOPER, 2014).

- Para definir o raio do canto dentro dos layouts, utilizamos o atributo Card View: cardCornerRadius.
- Definir o raio do canto no seu código, usamos o método CardView.setRadius.
- Definir a cor de segundo plano de um cartão, usamos o atributo card_view: cardBackgroundColor. Conforme demonstrado na Figura4.

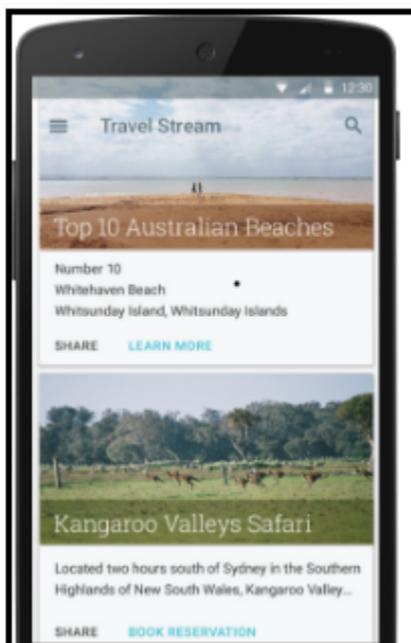


Figura 4 Exemplo de cartões (ANDROID DEVELOPER, 2014).

Na Listagem 4 é apresentada um exemplo de código para o Widget CardView.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    ... >
    <!-- A CardView that contains a TextView -->
    <android.support.v7.widget.CardView
        xmlns:card_view="http://schemas.android.com/apk/res-auto"
        android:id="@+id/card_view"
        android:layout_gravity="center"
        android:layout_width="200dp"
        android:layout_height="200dp"
        card_view:cardCornerRadius="4dp">

        <TextView
            android:id="@+id/info_text"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
    </android.support.v7.widget.CardView>
</LinearLayout>
```

Listagem 4 Código Widgets CardView (ANDROID DEVELOPER, 2014).

2.3.3 ADICIONAR DEPENDÊNCIAS

O Widgets RecyclerView e CardView são bibliotecas de suporte v7. Utilizando esses Widgets no projeto, teremos que adicionar dependências do Gradle ao modulo do aplicativo, conforme demonstrado a Listagem 5.

```
dependencies {
    ...
    compile 'com.android.support:cardview-v7:21.0.+'
    compile 'com.android.support:recyclerview-v7:21.0.+'
}
```

Listagem 5 Biblioteca do RecyclerView e CardView (ANDROID DEVELOPER, 2014).

2.4 SOMBRAS E CONTORNOS DE VISUALIZAÇÃO

Além das prioridades x e y, utilizadas no Android agora temos a prioridade Z. Essa nova prioridade representa a elevação de uma vista, que terminamos em tamanho da sombra e ordem do desenho.

- Tamanho de sombra temos vistas com valores Z lançam sombras maiores.
- Ordem de desenho temos valores de Z que parecem sobre outras vistas.

2.4.1 DEFININDO SOMBRAS E RECORTAR VISUALIZAÇÕES

O Material Design introduziu a elevação de elementos de interface do usuário. A elevação demonstra aos usuários a entender a importância relativa de cada elemento ao concentrar a atenção deles nas tarefas a mão.

A elevação de uma visualização, que é representada pela prioridade Z, que é determinada a aparência visual das sombras. Quando temos visualização com o valor Z mais altos lançam sombras maiores e mais suaves. As sombras são desenháveis pelo pai da visualização elevada por isso, estão sempre sujeitos recortes padrões de visualização. Essas elevações são úteis para criação de animações em que os Widgets se elevam temporariamente sobre um plano de visualização ao se realizar uma ação (ANDROID DEVELOPER, 2014).

2.4.2 ATRIBUINDO ELEVAÇÃO E VISUALIZAÇÃO

Para visualização do valor Z temos dois componentes, elevação é um componente estático e movimentação é um componente dinâmico usado para animações.

$Z = \text{elevation} + \text{translationZ}$. A Figura 5 ilustra as sombras para diferentes elevações de visualização.

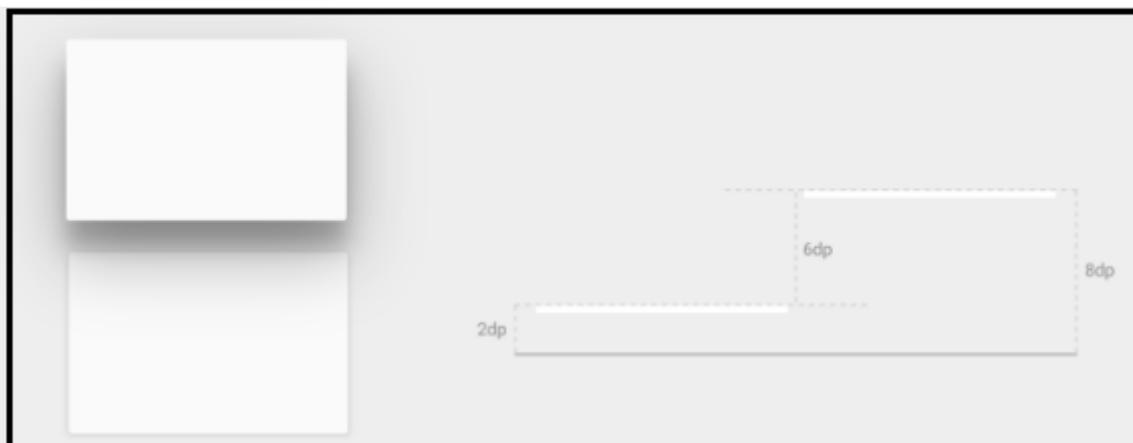


Figura 5 Sombras para diferentes elevações de visualizações (ANDROID DEVELOPER, 2014)

Para definirmos elevação de uma visualização em uma definição de layouts, utilizamos o atributo `android:elevation`. Definirmos em código essa elevação de visualização utilizamos o método `View.setElevation()`. Assim temos o código também para definir uma conversão utilizando o método `View.setTranslationZ()`.

O Android também teve o desenvolvimento de um novo método `ViewPropertyAnimator.z()` e `ViewPropertyAnimator.translationz()`, esses métodos permitem animar facilmente a elevação de exibições. Também podemos utilizar `StateListAnimator` para especificar as animações de um modo declarativo, especialmente útil para casos que os desenvolvedores modificam a animações de acionamento, quando o usuário pressiona o botão (ANDROID, 2014).

2.4.3 PERSONALIZAR SOMBRAS E CONTORNOS DE VISUALIZAÇÃO

Obtemos limites para um desenhável do segundo plano da visualização onde determinam a forma padrão das sombras. Os Contornos representam a forma externa de um objeto gráfico e definem a área de ondulação para o feedback de toque. Na Listagem 6 o código que define um desenhável de segundo plano.

```

<TextView
    android:id="@+id/myview"
    ...
    android:elevation="2dp"
    android:background="@drawable/myrect" />

```

Listagem 6 Código desenhável de segundo plano (ANDROID DEVELOPER, 2014).

O código da Listagem 7 é descrito orientando como definir um retângulo com bordas arredondas.

```

<!-- res/drawable/myrect.xml -->
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#42000000" />
    <corners android:radius="5dp" />
</shape>

```

Listagem 7 Definindo retângulo com bordas (ANDROID DEVELOPER, 2014);

Com essas visualizações são lançadas sombra e bordas arredondadas, sendo uma vez desenhável de segundo plano, definidos como contorno de visualização. Contudo um contorno personalizado neutraliza a forma padrão de uma sombra de visualização, para podermos definir esses contornos personalizados utilizamos os códigos abaixo.

- Ampliando a classe ViewOutlineProvider
- Neutralizando o Método getOutline().

Utilizando os métodos, podemos estar criando contornos ovais, retangulares com bordas arredondadas utilizando a classe Outline (ANDROID DEVELOPER, 2014).

2.5 ANIMAÇÕES

As animações são novas APIs que permitem criar animações personalizadas para feedback de toque em controles de IU, mudanças no estado da vista e transição de atividades. Como base temos as APIs abaixo:

Permitem responder eventos de toque nas vistas com animações de feedback de toque

- Oculta e exibe vistas com animações de revelação circulares
- Alterna entre atividades com animações de transição de atividades
- Cria animações mais naturais como movimento curvado
- Animar mudanças em um ou mais propriedades mudança de estado vista
- Exibe animações desenháveis da lista de estado entre mudanças de estado da vista (ANDROID DEVELOPER, 2014).

2.5.1 DEFININDO ANIMAÇÕES PERSONALIZADAS

As animações no Material Design dão feedback aos usuários sobre as ações deles fornecendo continuidade visual à medida que interagem com o aplicativo. O tema do Material vem fornecendo algumas animações padrões como Botões e transições de atividades, no Android 5.0 posteriormente permite a personalização dessas animações e a criação de novas, feedback de toque, revelação circular, transição de atividades, movimento curvado, mudança de estado da visualização (ANDROID DEVELOPER, 2014).

2.5.2 FEEDBACK DE TOQUE

No Material Design os feedbacks de toques fornecem confirmação visual instantânea no ponto de contato quando os usuários interagem com os elementos de interfaces do usuário. As animações de feedbacks de toque padrões é utilizado a nova classe `RippleDrawable`, que passa por transições entre diferentes estados com um efeito de ondulação (ANDROID DEVELOPER, 2014).

Essa funcionalidade é aplicável no XML de visualização especificando o segundo plano de visualização como:

?android:attr/selectableItemBackground, utilizamos para uma ondulação delimitada.

?android:attr/selectableItemBackgroundBorderless utilizamos para uma ondulação que se estenda além da visualização.

2.5.3 EFEITO DE REVELAÇÃO

Revelar animações fornece continuidade aos usuários ao exibir ou esconder um grupo de elementos da interface. Utilizando o método `ViewAnimationUtils.createCircularReveal()` permite animar um circuito de recorte para revelar ou ocultar uma visualização. Na Listagem 8 iremos demonstrar o código para revelar uma visualização anteriormente invisível (ANDROID DEVELOPER, 2014).

```
// previously invisible view
View myView = findViewById(R.id.my_view);

// get the center for the clipping circle
int cx = (myView.getLeft() + myView.getRight()) / 2;
int cy = (myView.getTop() + myView.getBottom()) / 2;

// get the final radius for the clipping circle
int finalRadius = Math.max(myView.getWidth(), myView.getHeight());

// create the animator for this view (the start radius is zero)
Animator anim =
    ViewAnimationUtils.createCircularReveal(myView, cx, cy, 0, finalRadius);

// make the view visible and start the animation
myView.setVisibility(View.VISIBLE);
anim.start();
```

Listagem 8 Revelar uma visualização anteriormente invisível (ANDROID DEVELOPER, 2014).

Para esconder uma visualização anteriormente visível utilizamos o seguinte código da Listagem 9.

```

// previously visible view
final View myView = findViewById(R.id.my_view);

// get the center for the clipping circle
int cx = (myView.getLeft() + myView.getRight()) / 2;
int cy = (myView.getTop() + myView.getBottom()) / 2;

// get the initial radius for the clipping circle
int initialRadius = myView.getWidth();

// create the animation (the final radius is zero)
Animator anim =
    ViewAnimationUtils.createCircularReveal(myView, cx, cy, initialRadius, 0);

// make the view invisible when the animation is done
anim.addListener(new AnimatorListenerAdapter() {
    @Override
    public void onAnimationEnd(Animator animation) {
        super.onAnimationEnd(animation);
        myView.setVisibility(View.INVISIBLE);
    }
});

// start the animation
anim.start();

```

Listagem 9 escondendo visualização anteriormente visível (ANDROID DEVELOPER, 2014).

2.5.4 TRANSIÇÕES DE ATIVIDADES

As transições de atividades em aplicativos com Material Design fornecem com conexões visuais em estados diferentes por meio de movimentos e transformações entre seus elementos em comuns. Podem ser especificadas animações personalizadas para transições de entrada e de saída para transições de elementos compartilhados entre atividades.

Uma transição de entrada determina como as visualizações em uma atividade entram em cena, já a transição de saída determina como as visualizações em uma atividade saem de cena. (ANDROID DEVELOPER, 2014).

O Android 5.0 tem compatibilidade com estas transições de entrada e saída conforme abaixo.

- Explodir: move as visualizações para dentro e para fora partindo do centro cena

- Deslizar: move as visualizações para dentro ou para fora partindo do centro da cena
- Esmacecer: adiciona ou remove uma visualização.

Habilitando essas transições de atividades no aplicativo, a transição de esmaecimento cruzado padrão é ativada entre as atividades de entrada e saída. Na Figura 6 demonstra uma imagem de transição de cena com um elemento compartilhado (ANDROID DEVELOPER, 2014).

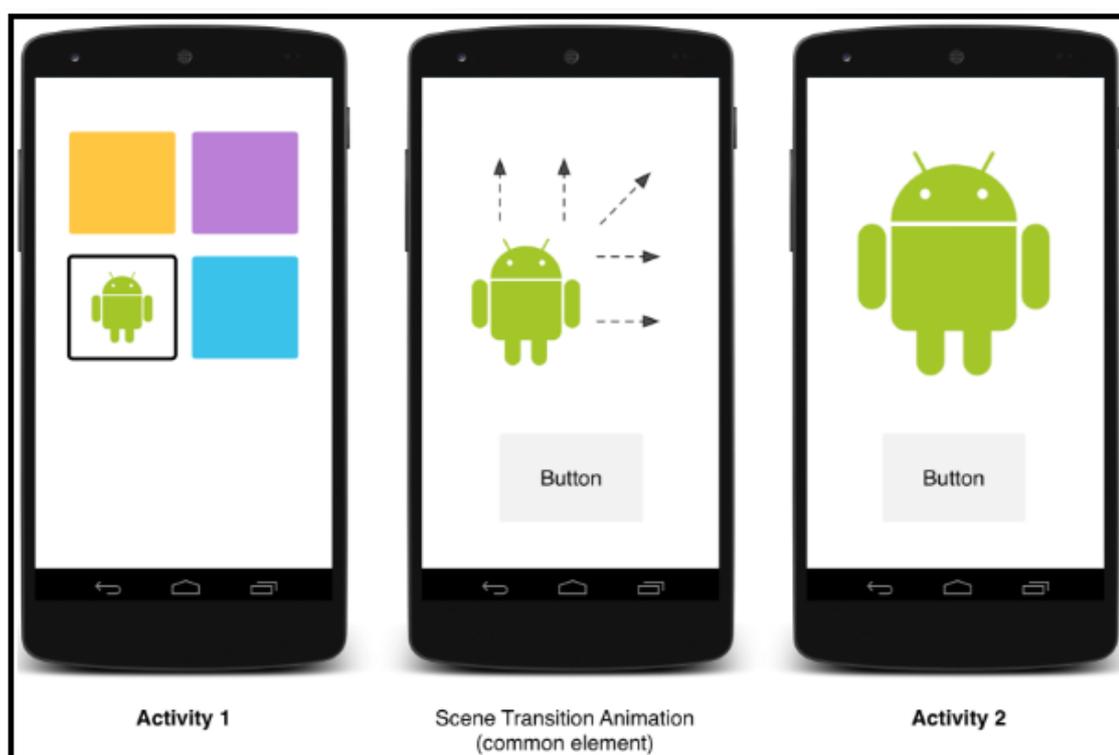


Figura 6 Transição de cena com um elemento compartilhado (ANDROID DEVELOPER, 2014).

2.5.5. ESPECIFICANDO TRANSIÇÕES DE ATIVIDADES

Primeiramente iremos habilitar as transições de conteúdo da janela com o atributo `android:windowContentTransitions` ao definir um estilo herdado do tema do Material.

Pode também especificar transições de entrada, saída e elementos compartilhado na definição de estilo conforme o código na Listagem 10.

```
<style name="BaseAppTheme" parent="android:Theme.Material">
  <!-- enable window content transitions -->
  <item name="android:windowContentTransitions">true</item>

  <!-- specify enter and exit transitions -->
  <item name="android:windowEnterTransition">@transition/explode</item>
  <item name="android:windowExitTransition">@transition/explode</item>

  <!-- specify shared element transitions -->
  <item name="android:windowSharedElementEnterTransition">
    @transition/change_image_transform</item>
  <item name="android:windowSharedElementExitTransition">
    @transition/change_image_transform</item>
</style>
```

Listagem 10 Código de transições (ANDROID DEVELOPER, 2014).

O elemento `change_image_transform` corresponde a classe `ChangeImageTrnasforme` classe é definido no código está transição captura matriz de um `ImageView` antes e depois da mudança de cena e anima-lo durante a transição, conforme a Listagem 11.

```
<!-- res/transition/change_image_transform.xml -->
<!-- (see also Shared Transitions below) -->
<transitionSet xmlns:android="http://schemas.android.com/apk/res/android">
  <changeImageTransform/>
</transitionSet>
```

Listagem 11 Transição (ANDROID DEVELOPER, 2014).

3 PLATAFORMA GOOGLE ANDROID

Neste capítulo, serão apresentadas as características, evoluções e funcionalidades principais da plataforma *Google Android*.

3.1 INTRODUÇÃO AO ANDROID

O Android é uma plataforma que foi desenvolvida pela Google, especialmente para dispositivos móveis como aparelhos celulares. É uma plataforma que contém diversos aplicativos principais, como navegador de internet, Contatos e o Telefone propriamente dito, além desses principais complementos existe também o Android SDK que é um conjunto de ferramentas e APIs para o desenvolvimento de aplicativos para plataforma Android, sendo utilizado a linguagem em Java (MONTEIRO,2012).

O sistema operacional Android é baseado em Linux e teve seu desenvolvimento iniciado em 2003 pela empresa Android Inc. Em 2005, a empresa foi adquirida pelo Google que hoje lidera o desenvolvimento do Android. Em 2007, criou-se uma associação de empresas chamadas Open Handset Alliance, que como o próprio nome diz Aliança/Associação de empresas líderes em tecnologia móvel, cuja a sua missão é desenvolver uma plataforma para dispositivos móveis que seja completa, aberta e gratuita. Além dos seus benefícios que a plataforma Android proporciona, ela ainda possui um código aberto, ou seja, você tem acesso aos códigos fontes e também podendo contribuir com o projeto fazendo alterações e melhorias (MONTEIRO,2012).

A plataforma Android está presente em dispositivos móveis e disponível para usuários de mais diferentes perfis. A logo do Android é representada por um robô verde chamado Bugdroid criado pela ilustradora Irina Blok. Como a própria ilustradora cita em sua página: “ A ideia era criar uma logo *Open Source* (Muito parecido com a plataforma Google Android) ” (HAMANN,2013). A figura 7 ilustra o Bugdroid e alguns dos seus aplicativos.

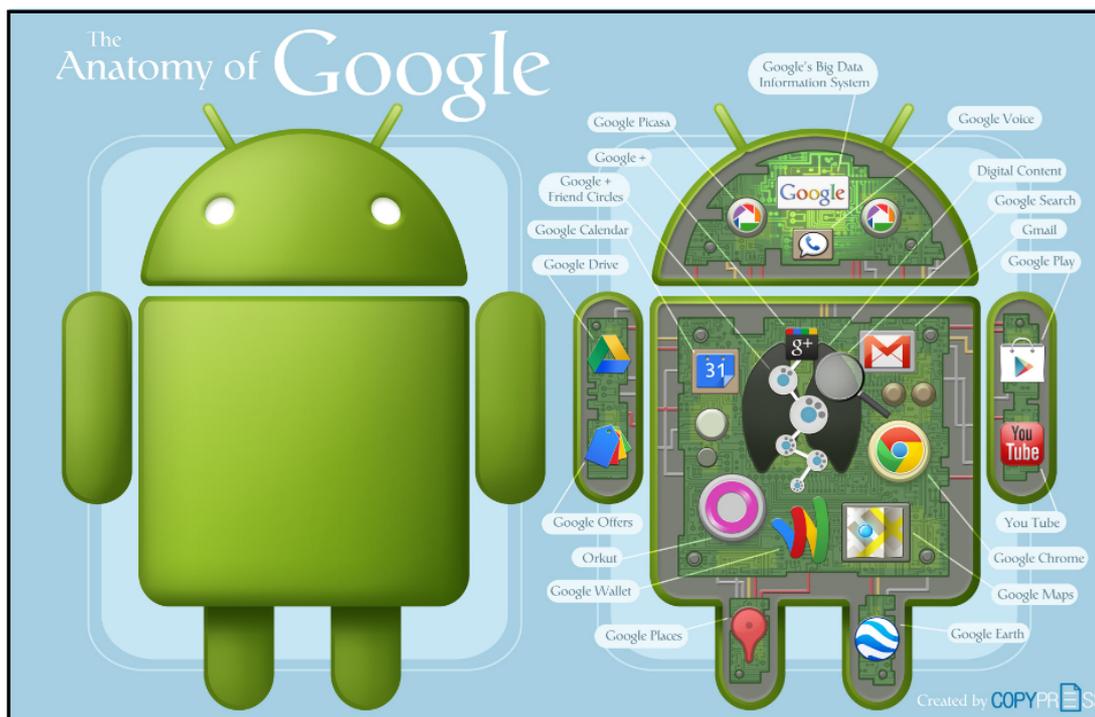


Figura 7 Aplicativos do Android representados pelo Bugdroid, (In: THE MOVIE THEME SONG, 2015).

3.2 OPEN HANDSET ALLIANCE (OHA)

Open Handset Alliance, que é liderado pela Google. OHA, trata-se de uma aliança de telefone celular aberto que é composta por 30 empresas do mercado de TI que apoiam a utilização de código aberto, podemos citar algumas empresas que estão investido nessa plataforma de código aberto, Samsung, Intel, Motorola, Qualcomm e Telefónica, investindo uma grande porcentagem no Android, para buscar engenheiros para um melhor aperfeiçoamento do seu dispositivo (OPEN HANDSET ALLIANCE, 2007; FARIA, 2008).

Conforme dito anteriormente *Open Handset Alliance* é um grupo formado por grandes empresas de smartphones liderados pela Google, algumas dessas empresas são: HTC, LG, Motorola, Samsung, Sony Ericson, Toshiba e muitas outras (LECHETA, 2013).

Tendo como seu objetivo do grupo é definir uma plataforma única e aberta para celulares assim deixando seus consumidores mais satisfeito com o produto final.

Entre outros objetivos dessa aliança é criar uma plataforma moderna e flexível para o desenvolvimento de aplicações (LECHETA,2013).

3.3 SISTEMA OPERACIONAL LINUX

O sistema operacional Android foi fundamentado no Kernel 2.6 do Linux, responsável por gerenciar o serviço centrais do sistema, tais como gerenciamento de memória e segurança dos arquivos e pastas, processos, threads, pilha de protocolos de rede e modelo de drives. Outra função importante do Kernel é atuar como uma camada de abstração entre o hardware e o restante da pilha de software (PREIRA, SILVA 2019; LECHETA,2013).

Cada aplicativo no Android dispara um novo processo no sistema operacional. Alguns deles podem exibir uma tela para usuários, e outros podem ficar em execução em segundo plano por tempo indeterminado. Com alguns diversos processos e aplicativos podem ser executados simultaneamente, e o Kernel do sistema operacional é responsável por realizar todo o controle de memória. Caso necessário, o próprio sistema operacional pode decidir encerrar algum processo para liberar a memória e recursos, e talvez até reiniciar o mesmo processo posteriormente quando a situação estiver controlada (LECHETA,2013).

3.4 MAQUINA VIRTUAL DALVIK

A Dalvik é uma máquina virtual preparada para oferecer melhor desempenho, maior integração com novas gerações de hardware e projetada para executar várias máquinas virtuais paralelamente. Foi projetada para executar em sistemas com pouca memória de RAM, baixa frequência de CPU e sistema operacional sem espaço de memória Swap, além de ser otimizada para consumir menos memória, bateria e CPU (PEREIRA;SILVA,2009).

Apesar das aplicações serem escritas na linguagem Java, a máquina virtual utilizada não é Java (JVM), o Android utiliza a máquina virtual Dalvik, otimizada para execução em dispositivos móveis. As aplicações para Android são descritas em

linguagem Java e utilizam os recursos da mesma, o código fonte (.java) é compilado em bytecode (.class) e então convertido para o formato (.dex), em Dalvik Executable, que representa a aplicação do Android compilada. Após essa compilação, a aplicação, com os arquivos .dex e outros recursos utilizados são compactados em um único arquivo com a extensão .apk (Android Package File), que representa a aplicação final pronta para ser distribuída e instalada nos dispositivos (OLIVEIRA,2012 ; LECHETA,2013).

Em Java, você desenvolve seus códigos de origem Java, utilizando o compilador Java, para depois estes códigos serem executados pela Java Virtual Machine (JVM). Já no Android o processo é diferente. O mesmo código escrito em Java é compilado através do mesmo compilador gerando o código em Java, mais nesse ponto é compilado novamente utilizando *Dalvik* para gerar códigos *Dalvik*, para em seguida estes códigos serem executados em maquinas virtual *Dalvik* (GARGENTA,2011). A Figura 8 apresentada uma comparação do processo de compilação em Java e, no Android, utilizando a Dalvik.

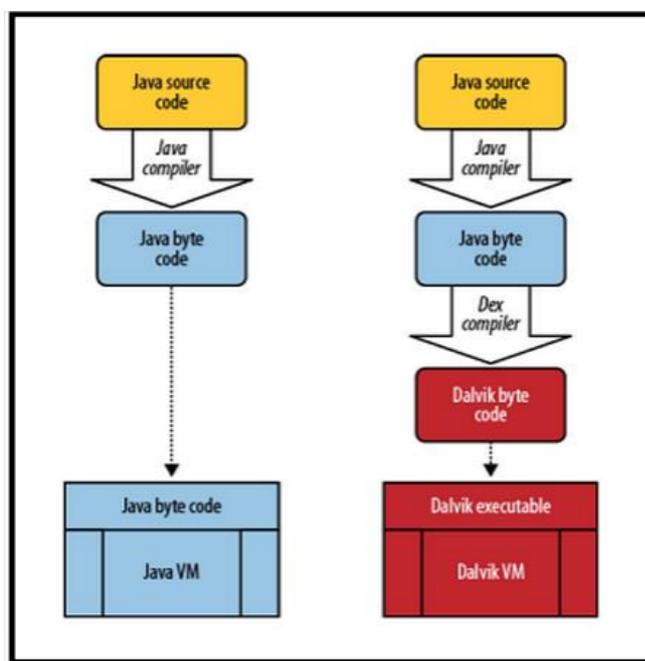


Figura 8 Java VM Versus Dalvik VM

3.5 AMBIENTE DE DESENVOLVIMENTO

As aplicações Android, são possíveis utilizar a linguagem em programação Java, em seu ambiente de desenvolvimento como Eclipse e Netbeans. No ambiente eclipse é possível utilizar o plug-in Android Development Tools (ADT) para facilitar o desenvolvimento, teste e compilação do projeto desenvolvido (FARTO,2010).

De acordo com o Android (2017), até 2013 o ambiente eclipse era o preferido pela Google sendo para desenvolvimento Android. No entanto a Google anúncio o ambiente Android Studio, passando a ser a IDE oficial da plataforma sendo incluso ferramentas mais rápidas necessárias para o desenvolvimento Android.

Na Figura 9 demonstra como é interface de desenvolvimento do Android Studio.

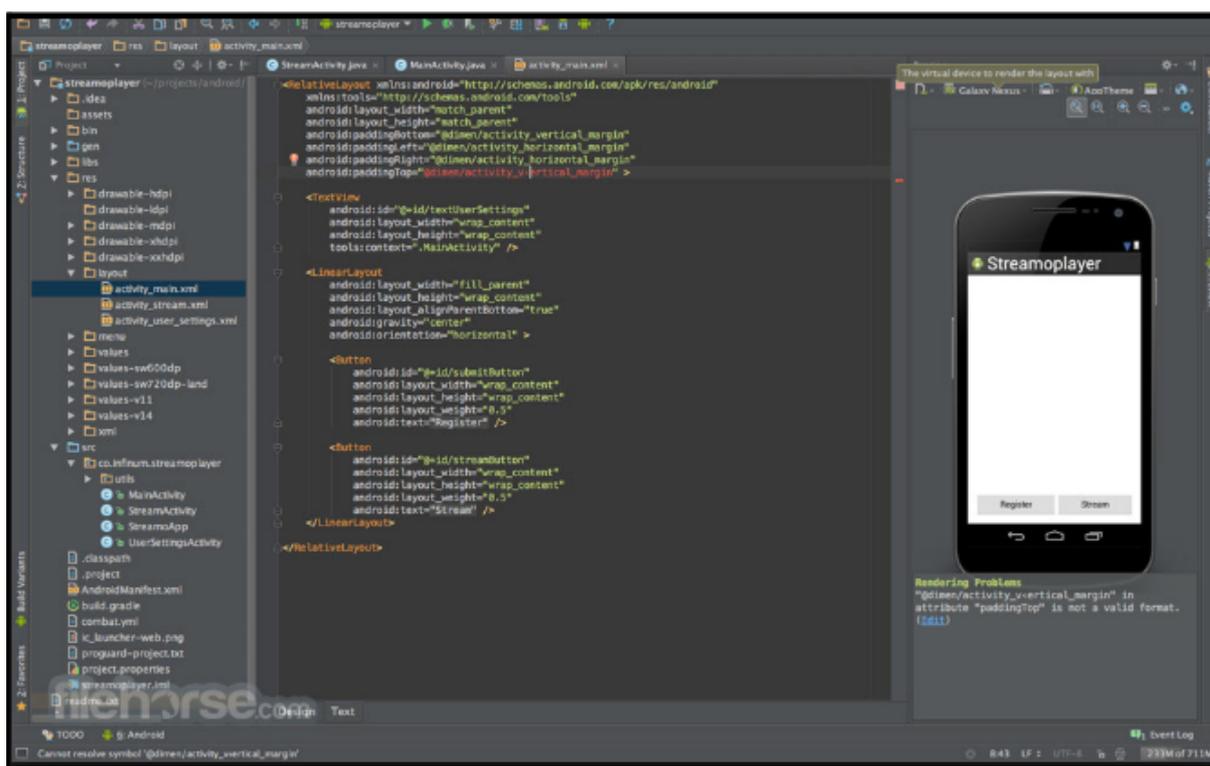


Figura 9 Ambiente de desenvolvimento Android Studio

3.6 ANDROID SDK

O Android SDK é o Software adotado para desenvolver aplicações no Android, este por sua vez fornece API completa para a linguagem Java, um emulador para simular um celular e ferramentas de desenvolvimento necessários para construir, testar e depurar aplicativos em Android (FARTO,2010).

O SDK possui um emulador que pode ser executado como um aplicativo normal, no entanto existe um plug-in para o Eclipse construído para integrar o ambiente de desenvolvimento Java com o emulador conforme a figura 10 (LECHETA,2013).

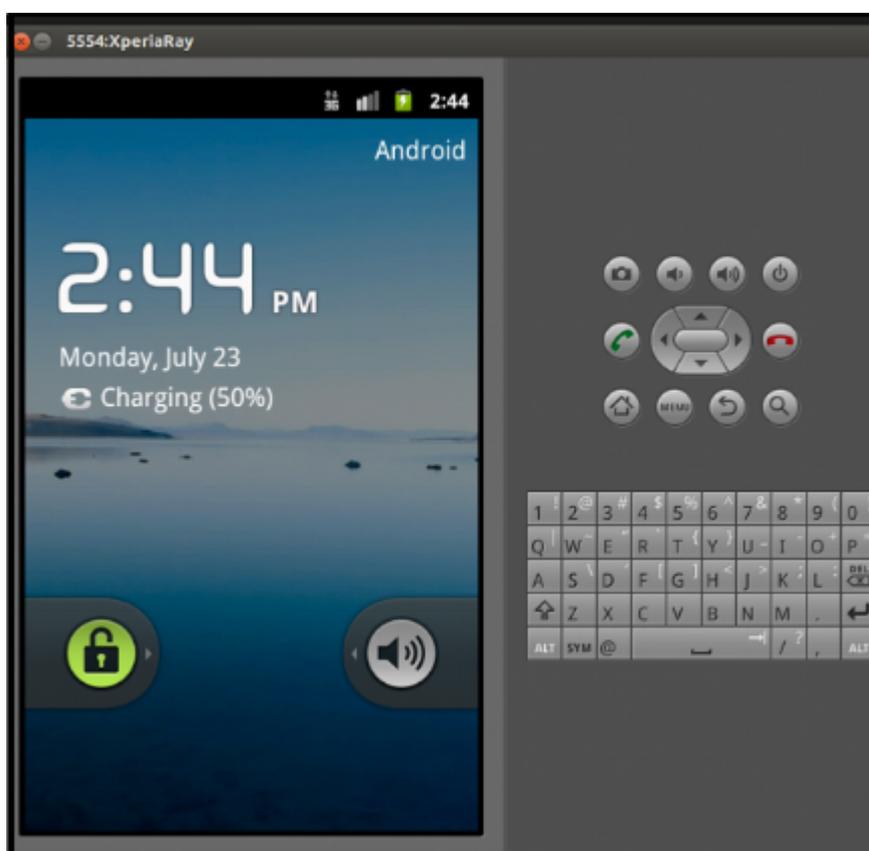


Figura 10 Emulador do Android

Com o plug-in, é possível iniciar um emulador dentro do Eclipse, instalando automaticamente na aplicação podendo ainda depurar o código fonte em sua

aplicação em qualquer linguagem Java, com isso por conta da integração do debug do Eclipse (FARTO,2010).

3.7 PLATAFORMA ANDROID

A plataforma *Google Android* possui uma arquitetura bem estruturada e dividida em camadas, conforme a figura 11.

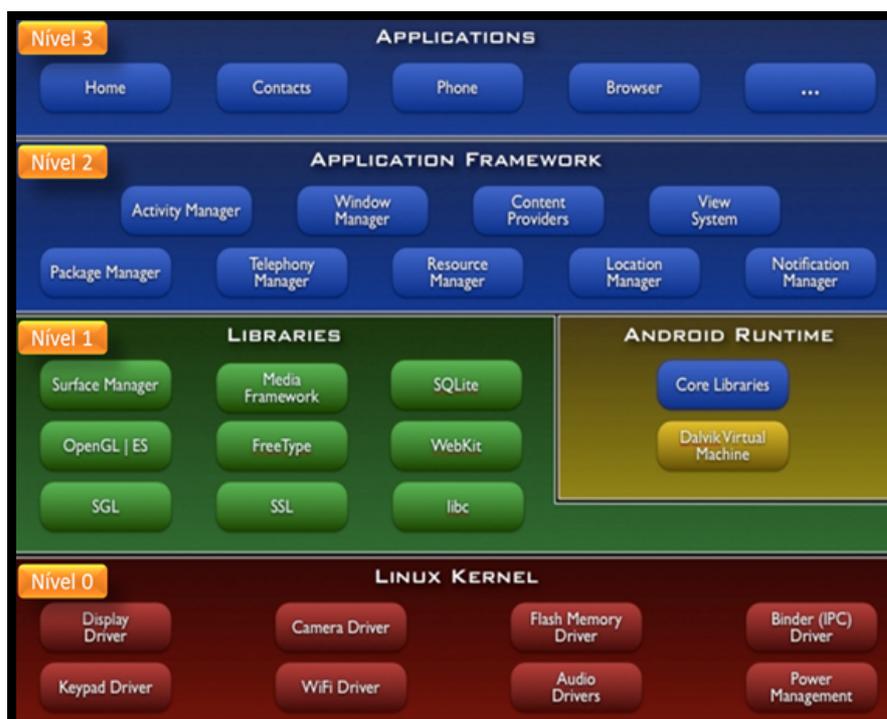


Figura 11 Arquitetura Google Android

(In: PEREIRA; SILVA, 2009, p. 5).

3.7.1 CAMADA LINUX KERNEL

A camada está localizada no sistema operacional de plataforma, baseada no Kernel 2.6 Linux. Responsável pelos serviços de baixo nível da plataforma, assim como segurança, gerenciamento de memória, redes e drivers, apesar disso também é responsável pela abstração do hardware e o restante do software, assim

desempenhando a função de Middleware entre as camadas da plataforma Google Android (FARIA,2008; FARTO,2010).

3.7.2 CAMADA ANDROID RUNTIME

O Android possui um grupo de bibliotecas, fornecendo a maioria das funcionalidades de bibliotecas da linguagem em programação em Java.

A aplicação em Android executa em seu próprio processo, com sua própria instancia da máquina virtual Dalvik. A máquina virtual Dalvik foi desenvolvido de uma forma para executar várias maquinas virtuais e eficientemente. É executado arquivos. dex, este arquivo é otimizado para consumo mínimo de memória. A máquina virtual é baseada em registros e classes compiladas em Java assim foram transformados em arquivos dex, por meio de ferramentas dx, que é incluída no SDK (FARIA,2008).

3.7.3 CAMADA LIBRARIES

A arquitetura Android possui bibliotecas C/C++ usadas por diversos componentes. As bibliotecas permitem trabalhar com arquivos de mídia comuns como MPEG4, H.264, MP3, AAC, AMR, JPG e PNG. Componentes para elaboração de projetos em 2D como em 3D, inclusive, uma biblioteca cuja implementação foi baseada em Open Graphics Library (OpenGL), fornecendo acesso a praticamente todos os recursos do hardware de vídeo. Junto a esta camada, foi disponibilizado o SQLite, um poderoso e leve banco de dados relacional (PEREIRA; SILVA, 2013, p. 7). Tabela 1 demonstra as principais bibliotecas da camada Libraries.

Bibliotecas	Descrição
System C Library	Uma implementação derivada da biblioteca C padrão sistema (libc) do BSD sintonizadas para dispositivo rodando em Linux
Media Libraries	As bibliotecas de mídias suportam os mais populares formatos de áudio, vídeo e imagen.

Surface Manager	Subsistema de exibição, bem como múltiplas camadas de aplicações 2D e 3D
LibeWebCore	Um Web Browser Engine utilizado tanto no Android Browser quanto pra exibição de Web
SGL	Engine de gráficos 2D
3D Libraries	Implementação baseada no OpenGL, As bibliotecas empregam aceleração em 3D via hardware (quando disponível) ou o software de renderização 3D altamente otimizado.
FreeType	Renderização de fontes Bitmap e Vctor
SQLite	Um poderoso e leve Engine de banco de dados relacional disponível para todas as aplicações Android

Tabela 1 Principais bibliotecas da Camada Libraries (FARIA,2008)

3.7.4 CAMADA APPLICATION FRAMEWORK

A arquitetura deste framework foi desenvolvida com o objetivo de simplificar a programação e a reutilização dos componentes, sendo possível qualquer desenvolvedor construir um aplicativo e permitindo que este seja utilizado por outros programas. Sendo assim, está camada é responsável por disponibilizar todas as APIs e recursos necessários para pacotes e aplicativos (PEREIRA; SILVA, 2013).

3.7.5 CAMADA APPLICATIONS

É a camada onde se encontram todos os aplicativos *Google Android*, como usuários de e-mail, contatos, navegador web, aplicativos GPS. Todos os desenvolvidos e implementados encontram-se neste nível (PEREIRA; SILVA, 2013, p. 6).

A partir do *plug-in*, é possível iniciar o emulador junto ao *Eclipse*, dando ao mesmo a capacidade de realizar depurações de no código de forma automática.

Atualmente, mas ainda como em fase de testes, a própria Google disponibilizou uma plataforma de desenvolvimento e depuração para aplicativos *Android* chamada *Android Studio*, cujo o objetivo é fornecer ferramentas integradas para o desenvolvimento de aplicativos na plataforma, e dar ao desenvolvedor uma vasta opção de edições de *layouts* e ferramentas de soluções rápidas, tudo já incluso dentro da suíte. O ambiente foi apresentado na última conferência anual da empresa, *Google I/O* (BARROS, 2013). A Figura 12 nos fornece o novo ambiente de desenvolvimento.

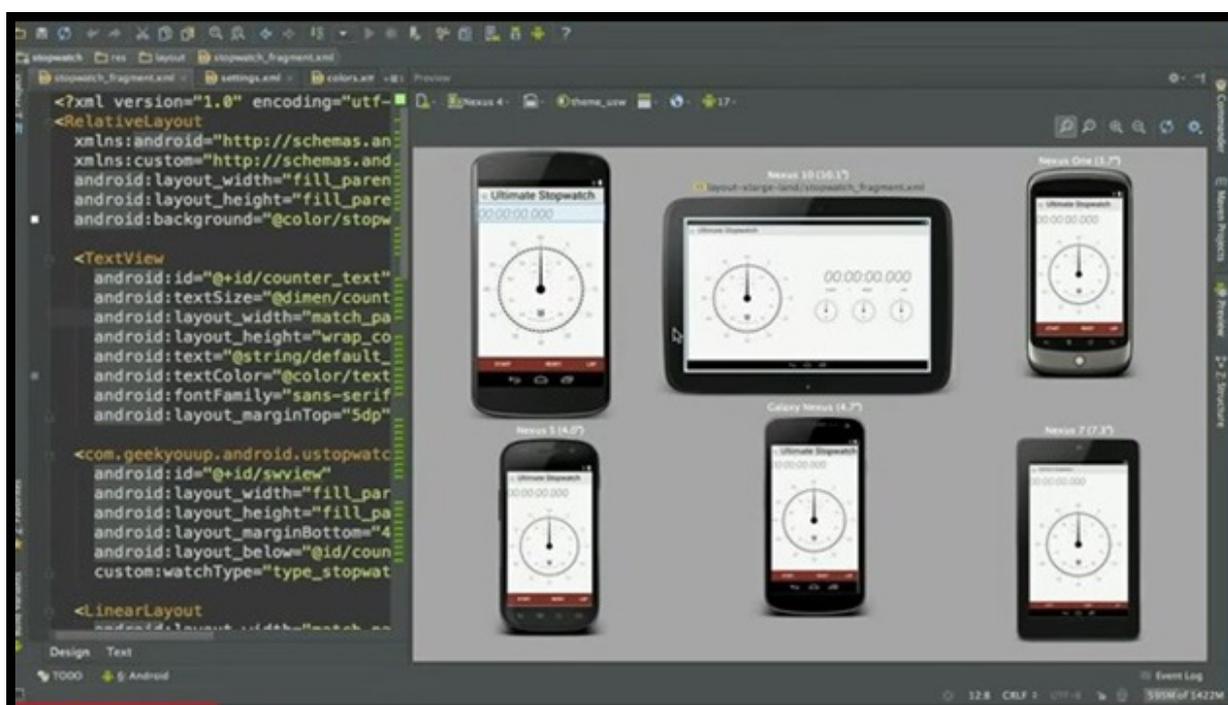


Figura 12 Android Studio

De acordo com Barros (2013), a plataforma tem grande possibilidade de ascensão, pois é totalmente *Open Source* e está sendo desenvolvida por uma poderosa e potente empresa que dará ao produto mais benefícios e facilidades ao programador. Ainda que em fase testes, muitos desenvolvedores já estão migrando para este ambiente de desenvolvimento, pois poderão fazer sugestões de mudanças e indicações para a versão final, assim atraindo um público alvo de programadores ainda maior (BARROS, 2013).

4 UX E DESIGN THINKING

Neste capítulo, serão apresentados os principais conceitos das metodologias de User Experience (UX) e Design Thinking para apoiar processos de inovação em novas soluções, como produtos e serviços.

4.1 CONTEXTUALIZAÇÃO EM EXPERIÊNCIA DO USUÁRIO

A Experiência do Usuário ou User Experience (UX) em termo “é usado para indicar a experiência como um conceito holístico”. UX surge com o objetivo de aliar a funcionalidade e a técnica com experiência emocional que o cliente tem ao consumir determinado produto ou serviço. Utilizando-se uma boa definição para UX seria explicitamente dizer que o tipo de experiência do usuário é um estado mental, por exemplo, uma sensação, sentimento, emoção, ligação emocional ou atitude. Assim o UX se mostra um sentimento momentâneo ou avaliativo bom ou mal ao interagir com um produto ou serviço (KORT, VERMEEREN e FOKKER,2007, p.57).

A construção da UX se dá pela participação dos usuários no processo como todo pelo envolvimento de diversas áreas interdisciplinares da empresa. São esses dois fatores que interagem na entrega da UX. Com o consumo do produto e serviço totalmente diferente, as interações dos usuários também seguem uma lógica diferente e precisa ser analisada de uma forma detalhista (ROTO,2007, p.32).

Saber reconhecer e visualizar toda a jornada de consumo de serviço faz toda a diferença para a empresa na hora de pensar e reinventar seus processos levando-se em relação ao que é importante ao seu usuário. A figura 13 descreve as fases da UX, para demonstrar as fases que ocorre durante a jornada do consumo do usuário (ROTO,2007, p.32).

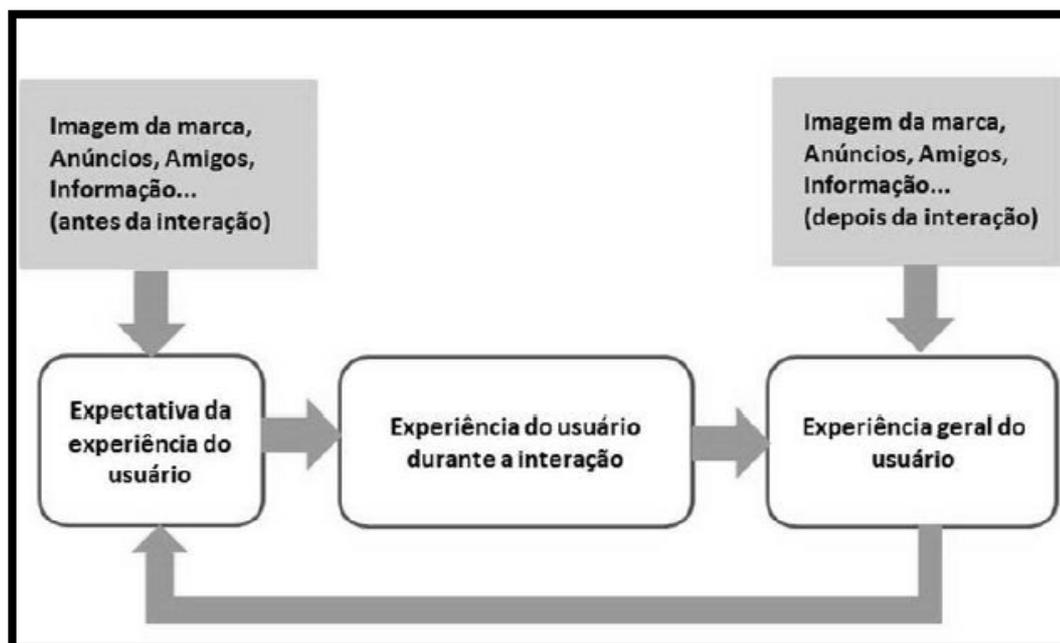


Figura 13 Fases do UX

A figura 13 demonstra como ocorre a jornada do consumidor ao usar um produto/serviço. A primeira fase do UX é representada pela expectativa, explícita que antes da pessoa começar a interagir com o produto e/ou serviço de fato, ela tem expectativas sobre isso, o que de certo modo não configura ainda a UX, pois só é possível afirmar o início “a partir do momento em que o produto proporciona um feedback a ação do utilizador, através de qualquer sentido” (ROTO, 2007, p. 32).

A segunda fase é marcada por tudo que está envolvido durante a interação que o usuário tem com o produto ou serviço. De acordo com (ROTO 2007, p. 32) investigando a fase de interação é possível visualizar pontos importantes que podem melhorar o produto. O terceiro momento é como o usuário classifica a sua experiência de uma forma geral e constrói seu ponto de vista a respeito disso, após isso ele compartilha essa informação para outros canais e pessoas, para entender UX durante a interação fase, precisamos ver o efeito dos três componentes para UX, o usuário, o contexto e o sistema que está sendo usado (ROTO, 2007, p. 32).

Na Figura 14 é possível observar o que ocorre a partir da interação do usuário com o produto ou serviço, como a motivação, recursos, estado mental, conhecimento, atitudes e expectativas, que são transformados em percepções para iniciar a interação com o sistema, representados pelo produto, objeto, serviço, pessoas e

infraestrutura envolvidas na interação. Tudo isso por meio da influência nos contextos: físico, social, temporal e de tarefa (ROTO, 2007, p. 32).

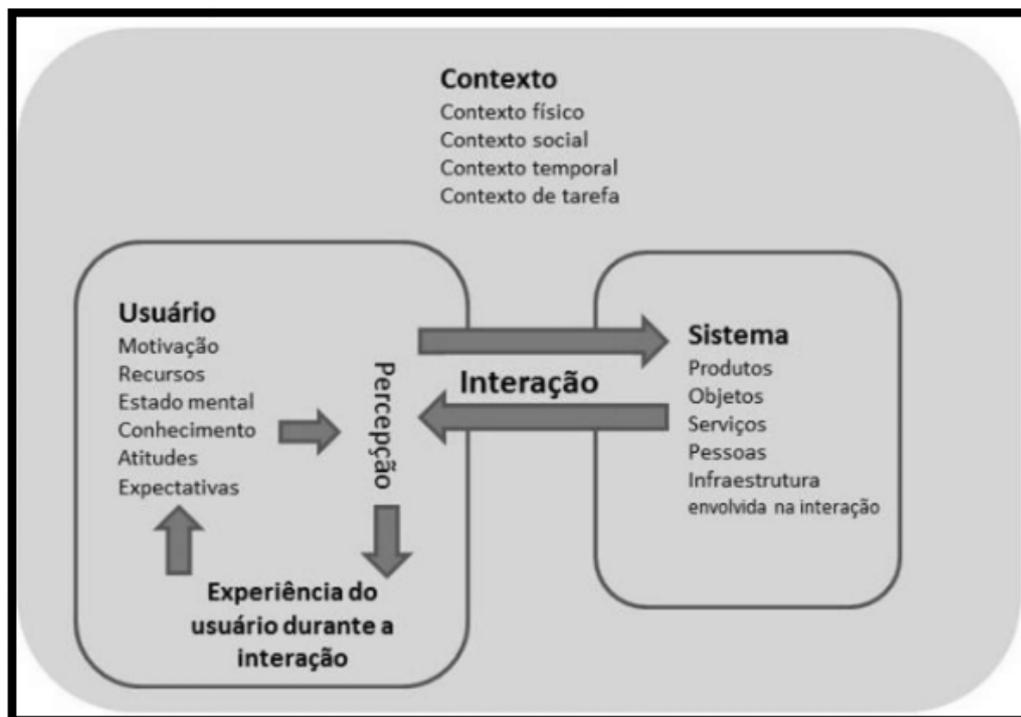


Figura 14 UX Durante a Interação (ROTO,2007)

Já na última fase, representado pela experiência geral do usuário, é necessário considerar alguns tópicos que possa mensurar o valor e a importância do produto para o usuário. Para isso, sugere-se a utilização dos seguintes elementos:

- Utilidade: saber se o produto atende com o seu propósito;
- Usabilidade: o produto é fácil de usar;
- Prazer: o design e interação são agradáveis;
- Orgulho: o usuário se sente orgulhoso pela utilização do produto/serviço. E é uma forma de mensurar os resultados da UX, com o objetivo de criar processos cada vez mais eficientes que proporcione valor para os usuários (ROTO 2007, p. 33).

4.2 CONTEXTUALIZAÇÃO EM DESIGN THINKING

De acordo com Pinheiro (2015) a utilização de máquinas no campo e o desenvolvimento das fábricas na época da Revolução Industrial alteraram completamente a rotina de trabalho. No entanto, durante as primeiras décadas da Revolução, a produção era a única preocupação de um negócio, concebendo uma organização mecânica devido à preocupação exacerbada com o custo, operários especializados e gestão centralizada, influenciando diretamente para ausência do processo criativo dentro das fábricas (PINHEIRO 2015, p. 5).

Posteriormente, essas características foram sendo transformada com a evolução da sociedade, e o surgimento do modelo Toyota no Japão com uma proposta de redução do desperdício e baseado em seus 4P's (Philosophy, Process, People e Problems) conseguindo erguer uma cultura de produção forte e que competisse com os modelos de carros americanos. Assim o Sistema Toyota de Produção (STP) virou referência para muitas outras empresas, "é a base para grande parte do movimento de produção enxuta que domina as tendências industriais há aproximadamente 10 anos". Assim, o conceito enxuto só foi se aperfeiçoando com tempo para que pudesse se adaptar à realidade, contribuindo diretamente para construção e manutenção de negócios (LIKER, 2005, p. 29).

4.2.1. ORIGENS

O design foi citado, pela primeira vez como uma nova forma de pensar por Herbert Simon, em seu livro *The Science of the Artificial*, escrito em 1996. O design vem emergindo desde a década de 70. Era uma época em que as ciências dominavam o cenário acadêmico, historicamente é função das disciplinas de ciências explicarem as coisas e das engenharias ensinar coisas artificiais: como fazer artefatos que tenha propriedades desejadas e como projetá-los. E devido a isso, as escolas estavam quase que abdicando da responsabilidade de formar as habilidades profissionais de seus alunos (SIMON,1996).

A partir disso, em 1991 surge a primeira consultoria de design thinking, a IDEO na Califórnia, fundada pela fusão da consultoria de David Kelley, professor da Stanford University que já utilizava a abordagem em seus negócios.

Em 1992, trouxe a ligação do design thinking (DT) como um caminho para resolução de problemas complexos. Desde então é dessa forma que o conceito vem sendo utilizado e ensinado nas instituições de ensino, como a d.School em Stanford University, Hasso Plattner Institute (HPI) School of Design Thinking no Brasil com a Escola de Design Thinking do grupo Echos, fundada em 2012. Com o tempo diversos autores vieram trabalhar a ideia considerando apenas a ideia de solucionar problemas complexos. Entretanto Pinheiro e Alt (2011, p. 41) explica que o DT é sobre pessoas, sobre compreender e trazer à tona o que as coisas significam para elas e projetar melhores ofertas com esse significado em mente (PINHEIRO, ALT 2011, p.41).

Para Brown (2010, p. 46), tem como missão a tradução das observações em ideias primárias que ajudaram na concepção de produtos e serviços para melhorar a vida as pessoas. O design thinking é uma abordagem que se utiliza do pensamento para formular questionamentos através da apreensão ou compreensão dos fenômenos pouco usado no meio empresarial. Sendo por esse motivo que o DT se destaque de outras abordagens, porque inicialmente busca compreender e entender o desafio proposto para construir soluções que venham trazer melhorias para pessoas (PAIVA; MARQUES, 2014, p. 3).

4.2.2. VALORES

O Design Thinking, como uma abordagem flexível, em que o importante é encontrar soluções para problemas propostos, está baseado em 3 valores essenciais que o caracterizam e estão presentes em todos os passos do processo de criação, são eles: empatia, colaboração e experimentação. O DT é baseado totalmente nas pessoas e é por esse motivo que a empatia é tão importante para o sucesso desse processo, se colocando no lugar outro e vivendo os problemas que o outro vive, atingindo um entendimento bem mais ampliado do problema e assim começar a construção de uma solução que realmente seja relevante para os usuários. Por isso

que para o DT ter sua abordagem baseada no ser humano, é de fundamental importância que a cultura de empatia da equipe esteja bem alinhada com o objetivo do projeto (LIEDTKA; OGILVIE, 2015, p. 6).

Para Pinheiro e Alt (2011, p. 19), quando a equipe mergulha no mundo das pessoas estão buscando entender seus mecanismos de escolha e decisão, isso é o início do processo de criação de insights que mais tarde evoluíram para soluções concretas. É a fase do processo para desenvolver uma compreensão de suas necessidades e vontades (das pessoas), tanto emocionais quanto 'racionais (LIEDTKA; OGILVIE, 2015, p. 6).

Um método que também é bastante utilizado para conhecer a fundo os usuários é o *Human Centered Design* (HCD), conhecido como Design Centrado no Ser Humano. São ferramentas que auxiliam no entendimento da população que se deseja ajudar, foi criado pela equipe da consultoria de design, IDEO, em parceria com outras empresas e tem como objetivo principal colaborar para melhorar a vida de milhões de pessoas. Outro valor importante que sustenta o conceito de DT é a colaboração, juntar conceitos de outras áreas para incluir os melhores campos de expertise na equipe do projeto contribui para uma visão holística para resolver problemas (PINHEIRO E ALT 2011, p. 13).

A multidisciplinaridade promove a ampliação da percepção em relação aos problemas propostos, assim a inovação fica mais propensa a acontecer e o entendimento se multiplica exponencialmente. Até porque de acordo com em um contexto de incerteza requer trabalhar em equipes multidisciplinares, com grande capacidade de diálogo e abertos para desenvolver novos caminhos. Por fim, como se trata de uma abordagem bem prática, o Design Thinking se baseia também na experimentação, um valor essencial que colabora na concepção de soluções e na melhoria dessas (BOSCHI 2012, P. 48).

Todas as soluções propostas devem ser prototipadas de forma simples, apenas para mostrar o funcionamento da ideia e depois serem levadas ao campo para que os usuários testem. A vantagem disso é que, a cada tentativa de acerto o, aprendizado aumenta e amplia a possibilidade de criar algo ainda mais relevante e de valor para os usuários (BOSCHI 2012, P. 48).

4.2.3. PROCESSO

O conceito de DT permite a remodelagem do processo devido sua flexibilidade. Os processos são adaptados para atender as restrições empresariais e estão relacionadas a fatores particulares inerentes ao tipo do produto, usuário final, tecnologia, orçamento e legislação. Muito se acredita que o modelo de DT desenvolvido pelas empresas globais que trabalham com inovação tenha surgido a partir do modelo *Double Diamonds* que mostra o desenho do pensamento de DT. Na Figura 15 é apresentado o modelo Double Diamonds, que representa o modelo tradicional de Design Thinking (BOSHI, 2012, p. 82).

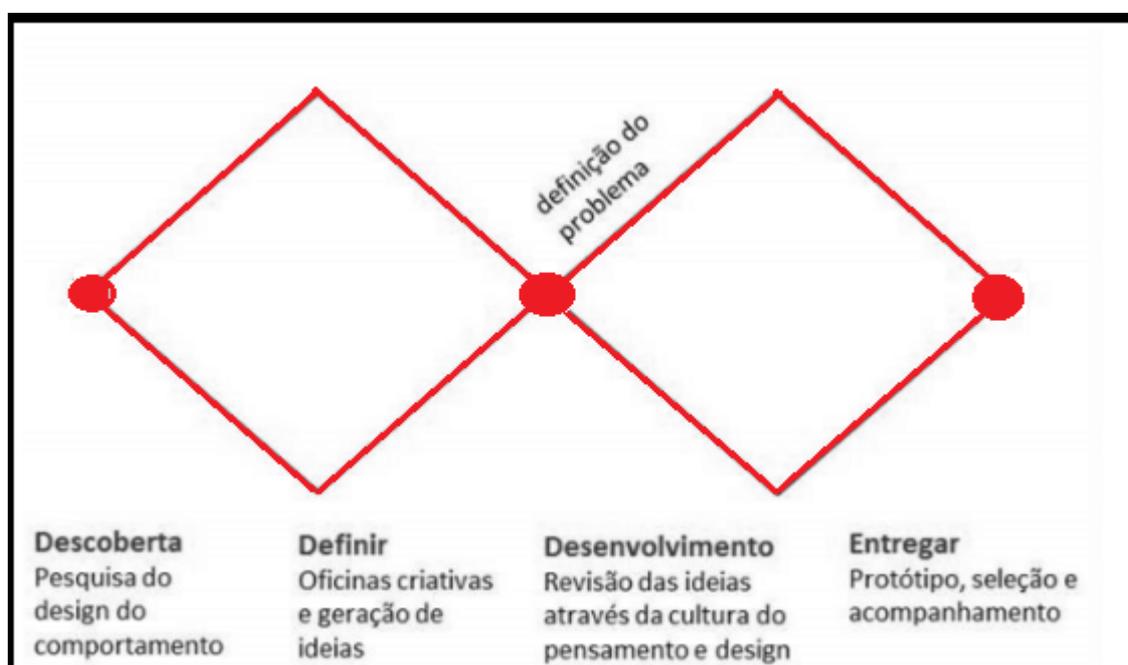


Figura 15 Double Diamond (DESIGN COUNCIL 2015, P. 15)

O modelo foi desenvolvido pelo *Design Council* do United Kingdom (UK) começa com o pensamento divergente por considerar diversas ideias, porém depois transita para um pensamento convergente devido o refinamento e seleção das ideias. Na Figura 15 pode-se visualizar as etapas que auxiliam na solução utilizando a técnica do Double Diamonds. Na primeira etapa é o momento de descobertas através da pesquisa e observação de comportamentos, depois se faz uma síntese das observações na fase define, com oficinas criativas e geração de ideias (VIANNA et al., 2012, p. 18).

Assim, com esse dado já é possível definir um problema. Depois dessa definição, na etapa desenvolve as ideias concebidas na fase anterior através da cultura do design thinking. Na última etapa, são prototipação as soluções encontradas e se seleciona a mais viável e é feito o acompanhamento da evolução desta. A Figura 16 apresenta as etapas do DT utilizado pela MJV Tecnologia e Inovação, uma empresa de consultoria em Inovação presente no Brasil. Esse esquema mostra apenas as etapas chave do processo, a partir de cada projeto são definidas as ferramentas e a forma como devem ser estruturados as etapas, uma vez que possuem uma natureza bastante versátil e não linear (VIANNA et al., 2012, p. 18).

A imersão ocorre a partir da aproximação do “contexto do problema, tanto do ponto de vista da empresa, o cliente, quanto o usuário final, o cliente do cliente. Na Figura 17 é possível observar o momento de conhecimento total do cenário que se irá trabalhar, possibilitando reunir informações importantes que contribuirão para o processo de criação das soluções. Esta primeira fase é crucial para o desenvolvimento das demais, pois minimiza os riscos de erro devido o aprendizado adquirido sobre todos os personagens envolvidos (VIANNA et al., 2012, p. 21).



Figura 16 Imersão no cotidiano do usuário (ESCOLA DE DESIGN THINKING 2016)

Na segunda fase, análise e síntese, todas as informações coletadas na fase de imersão são reunidas e visualizadas por toda a equipe do projeto. Assim, “os insights são organizados de maneira a obter-se padrões e a criar desafios que auxiliem na

compreensão do problema” (VIANNA et al., 2012, p. 65), como é apresentada na Figura 17.



Figura 17 Fase de análise e síntese (VIANNA ET AL.2012)

Após a coleta das informações ocorre a etapa de ideação, “essa fase tem como intuito gerar ideias inovadoras para o tema do projeto” (VIANNA et al., 2012, p. 99). A equipe do projeto se reúne para cocriação de possíveis soluções para os seus usuários finais, demonstrado na Figura 19. A colaboração nessa etapa é de extrema importância, pois o DT acredita no poder de uma equipe multidisciplinar para chegara um grande número de ideias (VIANNA et al., 2012, p. 99).



Figura 18 Fase de ideação (Vianna et al. 2012)

A última fase é a parte de tangibilizar as ideias geradas, ou seja, prototipação. As prototipações nada mais são que simulações que antecipam problemas e testam hipóteses para trazer as ideias à realidade. O processo pode ser observado na Figura 20 (VIANNA ET AL. 2002, p. 122).



Figura 19 Processo de prototipação (MJV,2014)

A grande vantagem de adotar o Design Thinking é que se trata de processo totalmente voltado para as necessidades das pessoas e considera o ciclo de aprendizagem contínuo, o que possibilita a aceleração da inovação (VIANNA ET AL. 2002, p. 122).

4.3. INOVAÇÃO E DESIGN THINKING

Para as organizações, as razões para inovar estão atreladas ao crescimento e à sobrevivência no mercado em que atuam. Tais razões podem ser tecnológicas desenvolvendo novos produtos ou serviços, melhorando o desempenho de processos ou produtos, ou promovendo adaptações a novas tecnologias. Também podem ser econômicas divididas em inovação de produtos e de processos. Nos produtos, a origem pode ser, por exemplo: abertura de novos mercados, eliminação de produtos obsoletos e ampliação da linha de produtos. Na inovação em processos,

a origem pode estar na melhoria das condições de trabalho e flexibilização da fabricação, na redução dos custos e da poluição ambiental (SIMANTOB e LIPPI, 2003).

A inovação se caracteriza pela mudança, pela necessidade de se adaptar a uma nova realidade. A inovação se concretiza a partir de uma mudança em algum tipo de valor, seja simbólico, métrico, tecnológico etc. que fundamentam um sistema. Para este autor, as inovações estimulam a renovação das tecnologias e processos em favor de outros mais adaptados ao novo contexto da sociedade e do mercado (PINHEIRO e ALT, 2011; BARBIERI, 2003).

4.4. DESIGN DE SERVIÇOS

O Design de Serviço (DS) surgiu a partir da mudança no comportamento humano, sendo importante não apenas consumir algo, mas perceber, experimentar e se relacionar com outras formas, objetos, pessoas e empresas. Por isso, a necessidade de se conceber novas formas de produtos/serviços a partir de novos meios, através de uma visão mais humanística. O núcleo de design de serviço, denominado design thinking, representa uma solução - ou experiência - focalizados de interpretação que coloca o cliente em primeiro lugar e a organização em segundo (ANDREASSEN et al. 2016, p. 23).

O DS não procura ser uma nova metodologia, mas sim uma integração com o modelo de Lean Startup, resultando em Mínimo Serviço de Valor (MSV), um modelo acessível e bem estruturado (Shostack 1984, p. 2). Os serviços são incomuns em seus impactos, mas não na forma, o que representa uma singularidade em cada experiência para o usuário. Por isso a importância de se vê o DS como uma lente em vários pontos de contato ao longo da viagem do cliente, tendo a perspectiva dos clientes e que representa uma abordagem de fora para dentro, o que melhora a experiência geral do cliente (ANDREASSEN et al., 2016, p. 23).

O modelo de criação de serviços proposto por Pinheiro (2015, p. 126) é baseado no MVS e permite um processo de aprendizagem mais rápido, focado no ser humano que possibilita o retorno ao início da jornada para que haja uma precisão e um alinhamento maior na fase de ideação. É uma metodologia que aborda a interação

do cliente com o serviço, pois estuda-se a jornada do usuário por completo, desde o primeiro contato, essa falta de conhecimento sobre as jornadas do “aprender, utilizar-se e se lembrar” dos usuários pode resultar em uma oferta de serviço ruim, repleta de obstáculos de uso, o que muitas vezes é o suficiente para afundar um startup ou, na realidade, negócios de qualquer tamanho. (PINHEIRO, 2015, p.73).

O DS não está ligado apenas ao desenvolvimento total de um serviço e não representa um trabalho pontual, é necessário o envolvimento de uma equipe para realização de mudanças com o intuito de adicionar e modificar critérios básicos que não estão trazendo valor para os usuários. Demonstrando que o design de serviço é uma abordagem essencial para as empresas que buscam não só criar propostas de valor para o cliente, mas tornar a experiência do usuário deles a melhor possível (SHOSTACK, 1984, p. 3).

4.5. MODELOS DE DESIGN: MVP E MVS

O modelo de *Minimum Valuable Service* (MVS) foi desenvolvido por Tenny Pinheiro baseado na modelo de *Minimum Viable Product* (MVP) e no conceito de Lean Startup com intuito de desvincular produto e serviço, ou seja, MVP e MVS, pois ele acredita que prototipar uma experiência é diferente de MVP, por isso a necessidade de se ter uma abordagem diferente, nesse caso chamado por ele de MVS que nada mais é que:

- Mínimo: possibilidade de criar e aprender mais rápido;
- Valor: produzir algo que as pessoas realmente querem;
- Serviço: deixar de lado o pensamento do “produzir e vender”, pensar de maneira relacional (PINHEIRO, 2015, p. 92).

4.5.1. MINIMUM VIABLE PRODUCT (MVP)

Em empreendedorismo, principalmente no contexto de startups, um produto viável mínimo (MVP, de *Minimum Viable Product*) é a versão mais simples de

um produto que pode ser lançada com uma quantidade mínima de esforço e desenvolvimento (BROWN 2010).

Um MVP ajuda os empreendedores a iniciarem o processo de aprender da forma mais rápida possível, pois poupa tempo e esforços. Porém, ele não é necessariamente o menor produto imaginável. Ao contrário do desenvolvimento tradicional de produtos, que geralmente envolve um longo e pensativo período de incubação e busca a perfeição do produto, o objetivo do MVP é começar o processo de aprendizagem, e não o finalizar. Ao contrário de um teste de protótipo ou conceito, um MVP foi concebido não apenas para responder questões sobre o design do produto e questões técnicas; seu objetivo é testar hipóteses fundamentais do negócio (BROWN 2010).

- Um MVP possui 3 características principais:
- Tem valor suficiente para que as pessoas comecem a utilizá-lo
- Demonstra benefícios suficientes para reter usuários iniciais

Fornecer um ciclo de feedback para orientar o desenvolvimento futuro, esta técnica de desenvolvimento assume que os usuários iniciais podem visualizar o produto final a partir do MVP, e que o produto deixa abertura a receber comentários e sugestões a ajudar no desenvolvimento de versões futuras (BROWN 2010).

4.5.2. MINIMUM VALUABLE SERVICE (MVS)

O *Minimum Valuable Service* MVS é focado em um segmento de mercado de potenciais clientes que têm as mesmas necessidades às quais você pode se alinhar. Definir e concentrar-se no seu MVS é vital porque, sem ele, os usuários potenciais que têm necessidades divergentes rapidamente puxarão seu MVP em muitas direções diferentes. Isso, por sua vez, inchará em vez de minimizar os requisitos do seu produto e drenar seus recursos de inicialização limitados (DRISCOLL,2017).

Mínimo A parte Mínima do MVS é manter seu segmento tão pequeno quanto possível para poder dominá-lo. Uma vez que você dominar você pode reivindicar a liderança. Mesmo que essa liderança esteja apenas em seu MVS, é valioso para o

seu posicionamento e referência para o próximo segmento que deseja seguir. Viable, à medida que você pondera seu MVP, veja qual segmento terá os menores requisitos de produtos (DRISCOLL,2017).

4.6. METODOLOGIAS EM DESIGN

Para o desenvolvimento de produtos, há uma vasta quantidade de métodos e técnicas que visam resolver os problemas e questões que envolvem o entorno material dos produtos existentes e em desenvolvimento, mas não se pode dizer que há um método ou técnica únicos que atendam a todas as situações possíveis. Cada designer se identifica com um método e/ou técnica específica e cada desenvolvimento projetual requer uma situação diferente (CAMARRI,2016).

Cada produto de design é resultado de um processo de desenvolvimento determinado por condições e decisões. Desse modo, a metodologia projetual não tem o objetivo de estabelecer um único método de design e nem tão pouco deve ser confundida com receita de bolo, que quando seguida, proporciona um resultado previamente estabelecido. Antes de mais nada é importante fixar na memória a diferença entre metodologia, método de desenvolvimento de projeto e técnica de desenvolvimento de projeto (CAMARRI,2016).

Metodologia é o estudo dos métodos, técnicas e ferramentas e de suas aplicações à definição, organização e solução de problemas teóricos e práticos (BOMFIM, 1995). Pode-se dizer que método projetual é o processo o qual o designer faz uso para chegar a uma solução, considerando todas as características e processos pelos quais um produto deverá passar para atender satisfatoriamente às funções pré-determinadas (CAMARRI,2016).

Técnicas de desenvolvimento projetual são itens utilizados segundo a necessidade e/ou conveniência do designer para auxiliar no processo de desenvolvimento do projeto. De modo geral, dando mais ênfase a uma etapa ou outra, ou ainda se alterando a ordem e nomenclatura, os métodos apresentam uma estrutura similar: Problema: identificação e definição de uma necessidade/opportunidade de projeto. Levantamento de dados: coleta de informações teóricas e de mercado e subsequente análise. Geração de alternativas: definição de conceitos (em geral,

muitos). Seleção de alternativas: análise, seleção e teste da melhor alternativa. Desenvolvimento do produto: produção (CAMARRI,2016).

Antes de apresentar alguns métodos e técnicas, é importante expor mais algumas considerações sobre métodos: Métodos servem para auxiliar no desenvolvimento de produtos, aumentando as chances de obter êxito. Não é possível utilizar um método padronizado de planejamento de produto para todos os casos. Métodos podem ser mesclados e até reinventados de acordo com a necessidade do projeto e/ou equipe de desenvolvimento. Quanto mais dispendioso o desenvolvimento do produto, mais detalhado deve ser seu planejamento, para justificar o investimento. Empresas podem investir desde centenas até bilhões em projetos, e de acordo com o investimento, deve ser seu planejamento (CAMARRI,2016).

4.7. ETAPAS DE DESIGN

O Design Thinking tem três etapas Imersão, Ideação e Prototipação, conforme apresentado na Figura 20;

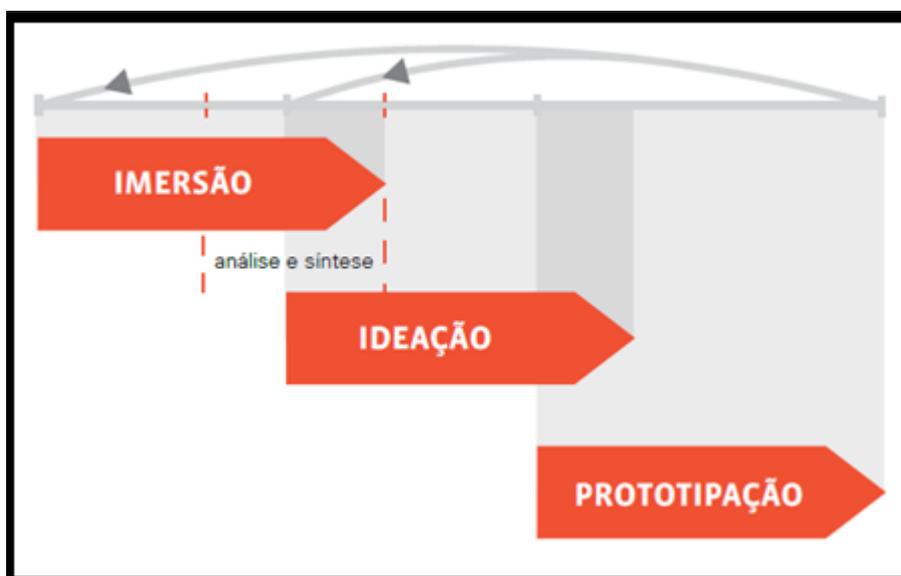


Figura 20 Esquema representativo das etapas do processo de Design Thinking

Imersão: encontrando a origem do problema esta é a fase de aproximação do problema. A equipe busca mergulhar nas implicações do desafio, estudando tanto o ponto de vista da empresa quanto do cliente. São feitas pesquisas de todos os tipos: entrevistas, buscas de tendências (Cool Hunting), observação direta e outras. A imersão pode ser dividida em Preliminar, quando há um primeiro contato com o problema e em Profundidade, quando se inicia o levantamento das necessidades e oportunidades que irão nortear a geração de soluções na fase seguinte do projeto, a Ideação. Outras ferramentas de Ideação são Cadernos de Sensibilização, Critérios Norteadores, dentre outros (VIANNA et al. 2002, p. 122).

Ideação: esta é a fase do brainstorming, a tempestade cerebral, quando as ideias são apresentadas sem nenhum julgamento. É o momento de efetivamente começar a “pensar fora da caixa”, propondo soluções para o problema. Para isso, utilizam-se práticas de estímulo à criatividade, o que ajuda as pessoas a gerar soluções que estejam de acordo com o contexto do assunto trabalhado. Não há limite de ideias nesta fase. Também é aconselhável que haja variedade de perfis de pessoas envolvidas, inclusive incluindo quem será beneficiado com as soluções propostas. (VIANNA et al. 2002, p. 122).

Prototipação: é hora de bater o martelo “Prototipar é tangibilizar uma ideia, é a passagem do abstrato para o físico de forma a representar a realidade - mesmo que simplificada e propiciar validações”, explicam os autores de Design Thinking - Inovação em Negócios. (VIANNA et al. 2002, p. 122).

4.7.1. CONTEXTUALIZAÇÃO NO "DIAMANTE DE DESIGN"

O modelo “Diamante Duplo” (Figura 21) foi desenvolvido pelo Design Council em 2005 (DESIGN COUNCIL, 2007), a partir de uma pesquisa que visava conhecer melhor os processos de design nas empresas. Nessa iniciativa, foram identificadas as etapas comuns nos processos de design de 11 empresas: Alessi, BSKyB, BT, LEGO, Microsoft, Sony, Starbucks, Virgin Atlantic Airways, Whirlpool, Xerox e Yahoo escolhidas pelo destaque na utilização do design no desenvolvimento de seus produtos e serviços. Este modelo é dividido em quatro etapas: “Descobrir, Definir, Desenvolver e Entregar”, e tem como uma de suas características a utilização do

pensamento divergente e convergente (PINHEIRO; ALT, 2011; DESIGN COUNCIL, 2007, p.10).

Na Figura 21, pode ser observada a dinâmica do processo: o momento de criar opções está representado pelas linhas divergentes, enquanto o momento de fazer escolhas é representado pelas linhas convergentes. A primeira etapa, descobrir, é iniciada a partir de uma ideia identificada nas necessidades do usuário. Na etapa de Definição, procura-se interpretar as necessidades dos usuários e alinhá-las aos objetivos do negócio, criando um conjunto de requisitos (brief) que servirão de base para as próximas fases (PINHEIRO; ALT, 2011; DESIGN COUNCIL, 2007, p.10).

Na terceira etapa, desenvolver, as soluções de projeto elaboradas a partir do design são desenvolvidas e testadas de forma iterativa com outras áreas da empresa. Na última etapa, entregar, o produto ou serviço resultante do processo é finalizado e segue para o mercado (PINHEIRO; ALT, 2011; DESIGN COUNCIL, 2007, p.10).

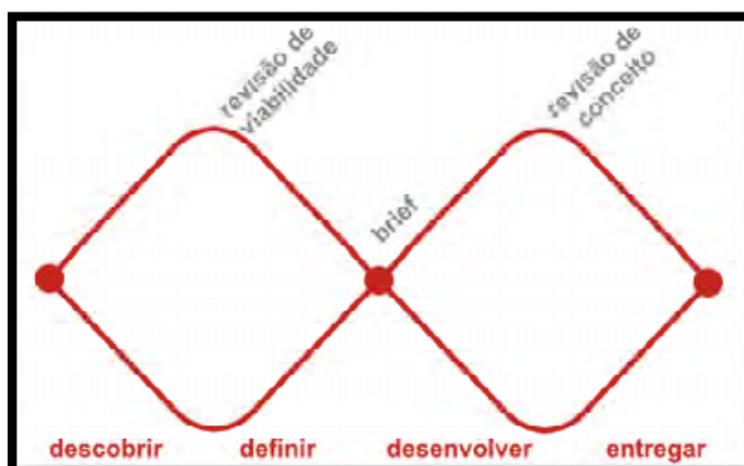


Figura 21 Diamante (COUNCIL, 2007)

4.7.2 IMERSÃO

Após o período de ideação divergente, onde ideias sobre o jogo são levantadas, parte-se para a imersão, fase em que a identificação de necessidades e oportunidades que irão nortear a geração de soluções na fase seguinte do projeto são realizadas. Para o desenvolvimento de um jogo, sugere-se três técnicas nesta

fase: pesquisa exploratória, pesquisa desk, Moodboard e mapas de empatia ou personas. A pesquisa exploratória consiste em uma análise contextual, de modo a conhecer o contexto para o qual o jogo está sendo proposto (VIANNA et al. 2012, P. 21).

A pesquisa desk complementa a pesquisa exploratória, auxiliando na identificação do cenário do jogo, e consiste em uma busca de informações sobre o tema do projeto em fontes diversas (websites, livros, revistas, blogs, artigos, entre outros). Assim, realiza-se a identificação de nichos de mercado, pouco explorados e com potencial para expansão. Também é realizado um estudo de viabilidade, identificando-se jogos similares, plataformas e categorias com maior retorno financeiro. Por fim, foca-se no público alvo com o levantamento do perfil dos potenciais usuários. Todo este estudo é resumido em cartões de insights. Outra técnica que auxilia na imersão durante a concepção de um jogo é o esboço de moodboards, com uma colagem de textos e/ou imagens em um painel que servem de inspiração para os desenvolvedores (VIANNA et al. 2012, P. 21).

4.7.3. IDEAÇÃO

Na ideação, todas as oportunidades observadas na etapa de inspiração são trabalhadas através da síntese de informações, que tem como objetivo refinar as oportunidades e gerar ideias para o projeto. A partir da descoberta das necessidades, identificadas na primeira etapa, busca-se explorar possibilidades para melhorar o conceito da ideia por meio de técnicas de cocriação, preferencialmente trabalhadas em conjunto com os usuários eficiente dos benefícios e desafios de cada solução. Dessa forma, as ideias com potencial mais elevado são selecionadas para que seja desenvolvido um protótipo dessa solução (VIANNA et al. 2012, p. 101).

O design thinker, portanto, dá forma a um pensamento que contribui para o processo do pensamento divergente, criando alternativas distintas. Somente acumular alternativas não passa de um exercício, sendo necessário que, na etapa de ideação, sejam selecionadas as melhores ideias e convertidas em algo tangível, passando da geração para a resolução de ideias, até a etapa de prototipagem (BROWN 2010).

4.7.4. PROTOTIPAÇÃO

A prototipação é um processo que tem como objetivo facilitar o entendimento dos requisitos, apresentar conceitos e funcionalidades do software. Desta forma, podemos propor uma solução adequada para o problema do cliente, aumentando sua percepção de valor. Os protótipos também são grandes aliados das metodologias ágeis de desenvolvimento, uma vez que garantem maior alinhamento entre a equipe e o cliente. Eles podem ser desenvolvidos em diferentes níveis de fidelidade: quanto maior ela for, mais o protótipo se assemelhará ao resultado entregue. No entanto, um protótipo de alta fidelidade leva mais tempo para ser criado ou modificado. A escolha do protótipo ideal varia de acordo com o nível de entendimento do negócio, a complexidade dos requisitos, prazo e orçamento para elaboração. Podemos dividir os protótipos em três categorias:

Wireframes & rascunhos, São protótipos de baixa fidelidade, rápidos para se desenvolver e modificar. Wireframes não vão mostrar detalhes visuais ou interações de tela, mas vão ajudar a validar requisitos e regras de negócio de maneira eficiente. Wireframes são a melhor escolha para representar cenários complexos onde um fluxo ou processo precisa ser compreendido (CAMARRI,2013).

Protótipos visuais, criados com programas de edição gráfica, este protótipo tem maior apelo visual, entretanto, não possuem interações de tela e demandam mais tempo para se fazer ajustes e melhorias. São uma ótima opção para telas com maior ênfase em estética e usabilidade, quando os requisitos já foram entendidos. As ferramentas utilizadas para a prototipação visuais é Adobe Photoshop e GIMP (CAMARRI,2013).

Protótipos interativos, São protótipos completos e representativos. Além da parte visual, englobam uma série de detalhes de estética e efeitos de interação, proporcionando uma experiência rica e realista. Também ajudam a equipe a identificar novos requisitos, oportunidades e futuros problemas. As consequências disso para o software são lucros maiores a longo prazo e riscos menores durante o desenvolvimento. Em contrapartida, protótipos interativos demandam uma equipe de maior conhecimento técnico e demoram mais tempo para serem criados. Ferramentas utilizadas,HTML, CSS, Javascript,CSS Twitter

Bootstrap framework CSS que pode acelerar bastante o processo de prototipação Adobe Dreamweaver permite fazer protótipos interativos mais simplistas, sem exigir tanto conhecimento técnico (CAMARRI,2013).

4.7.5. VALIDAÇÃO

É uma ferramenta de análise estratégica, utilizada na validação destas em relação aos Critérios Norteadores, bem como às necessidades das Personas criadas no projeto. O objetivo deste recurso é apoiar o processo de decisão, a partir da comunicação eficiente dos benefícios e desafios de cada solução, de modo que as ideias mais estratégicas sejam selecionadas para serem prototipadas (VIANNA et al .2012, p. 101).

Em resumo, a Prototipação é a fase de validação das ideias geradas. É a hora de aparar as arestas, ver o que se encaixa no projeto, juntar propostas e colocar a mão na massa. Apesar de ser apresentada como fase final, a prototipação pode acontecer em paralelo às outras fases. Conforme as ideias forem surgindo elas podem ser prototípicas, testadas e, em alguns casos, até implementadas (VIANNA et al, 2012, p. 101).

4.8. MODELO DE NEGÓCIOS CANVAS

O Modelo de Negócios Canvas, ou simplesmente Canvas, foi descrito no livro *Busines Model Generation* escrito por Osterwalder e Pigneur (2010) como uma ferramenta de gestão estratégica para a criação de novos negócios ou projetos. É considerada também como uma metodologia para facilitar a criação visual de novos negócios, produtos ou serviços. Para construir o Canvas Osterwalder e Pigneur (2010) utilizaram a tendência de criação por meio das redes sociais e construíram o Modelo com a participação de mais de 200 empresários e consultores.

Canvas é um processo genérico para ajudar a criar modelos de negócios inovadores, unindo conceitos, técnicas e ferramentas em Geração de Modelo de Negócio. As vantagens na utilização do Canvas estão relacionadas a uma forma fácil

e rápida de testar diversos modelos de negócio para uma nova empresa ou novo projeto, na visualização do Modelo de Negócios em apenas uma única página e no encaixe estratégico de áreas diferentes e, especialmente, na facilidade de prototipagem que permite que novos negócios sejam analisados rapidamente e em várias versões (OSTERWALDER; PIGNEUR, 2010).

Inicialmente deve-se considerar o aspecto conceitual do modelo de negócios enquanto uma forma da organização criar, entregar e capturar valor, para compreender as dimensões e os enquadramentos dos blocos ou elementos fundamentais (building blocks) que compõem um modelo de negócio (OSTERWALDER; PIGNEUR, 2010).

O Modelo de Negócios Canvas é formado por um painel com nove blocos que se comunicam e descrevem as principais partes de um negócio, conforme apresentado na Figura 23. O principal objetivo no preenchimento dos nove blocos que compõem o Canvas é extrair as Propostas de Valor que atendam e potencializem o projeto de um novo negócio, produto ou serviço, antes da implantação.

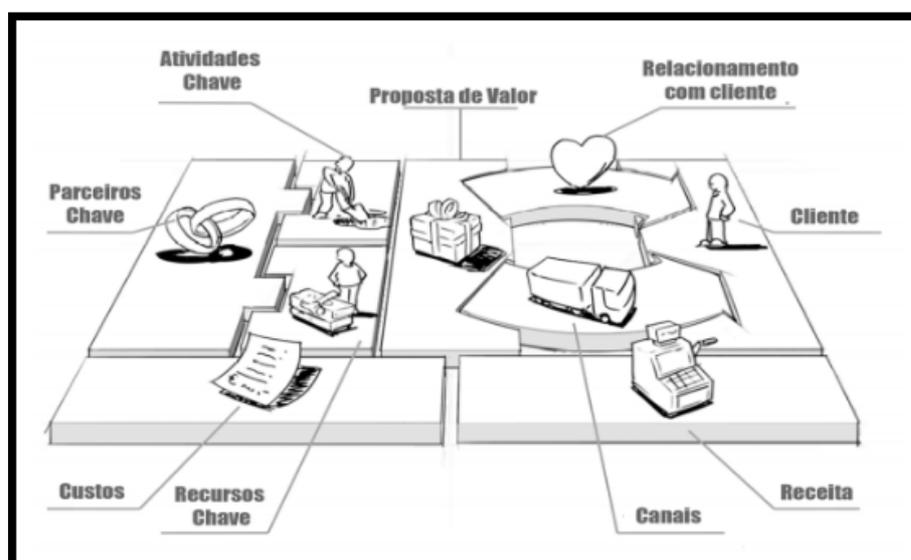


Figura 22 Modelo de Negócio Canvas Fonte: Osterwalder, Pigneur (2010)

Os blocos que compõem o painel do Modelo de Negócio Canvas estão divididos em: Back Stage, que são os blocos que não dependem do segmento dos clientes: atividades chave, custos, parceiros chave, recursos chave; e Front Stage: que são os blocos que dependem do segmento dos clientes: proposta de valor, canais de

distribuição, relacionamento com os clientes, clientes e fonte de receita. O preenchimento dos blocos inicia pelo Segmento de Clientes (customers segments), que busca mapear para quem se está criando valor e quem são os potenciais clientes para os objetivos pretendidos, ou seja, define os diferentes grupos de pessoas ou organizações que uma empresa pretende alcançar e servir (OSTERWALDER; PIGNEUR, 2010).

Este bloco pode ser dividido de acordo com as necessidades, costumes ou atributos comuns dos clientes e seu preenchimento objetiva compreender como atender melhor estes clientes. O modelo de negócios pode definir um ou vários segmentos de clientes grandes ou pequenos. Uma organização deve ter uma decisão consciente sobre quais segmentos deve servir e quais segmentos ignorar. Uma vez que esta decisão é tomada, um modelo de negócio pode ser cuidadosamente projetado em torno de um forte entendimento de necessidades específicas dos clientes (OSTERWALDER; PIGNEUR, 2010, p.20).

A descrição do modelo indica que existem modelos de negócios voltados para o mercado de massa, outros voltados a nichos específicos de mercado. São exemplos de segmentos de clientes: classe de consumidores A B e C, classificação por sexo masculino, feminino; por faixa etária, etc. Enquadrando estes segmentos para empresas, tem-se o exemplo das Casas Bahia que atende pessoas das classes C, D e e, que precisam de crédito para comprar; e o exemplo da Ferrari que atende homens de alto poder aquisitivo apaixonados por carros esportivos. No bloco Proposta de Valor (value proposition) se evidencia como as organizações criam valor para entregar aos seus clientes, neste sentido, são criadas propostas de produtos e/ou serviços que atendam a determinadas necessidades dos potenciais clientes, a proposta de valor justifica porque os clientes preferem a uma determinada organização, algumas podem ser inovadoras, outras baseadas em propostas já existentes no mercado, porém com características e atributos adicionais (OSTERWALDER; PIGNEUR, 2010).

Exemplos de propostas de valor pode ser novidade, personalização, marca/status, preço, acessibilidade, conveniência/usabilidade, entre outros. Os Segmentos de Clientes e as Propostas de Valor são os principais elementos do Modelo de Negócios e apoiam o desenvolvimento e preenchimento dos demais blocos. Os Canais (Channels) tratam da forma como a organização entrega sua Proposta de

Valor para os Segmentos de Clientes, englobam basicamente funções de logística e marketing, tais como os canais de distribuição, comunicação e vendas que compreendem a interface da empresa com os clientes (OSTERWALDER E PIGNEUR 2010, p.22) destacam como funções dos Canais de Distribuição:

- Sensibilizar os clientes sobre a companhia produtos e serviços;
- Ajudar os clientes a avaliar a Proposição de valor da empresa;
- Permitir que o cliente compre m produtos e serviços específicos;
- Entregar uma proposta de valor para os clientes;
- Fornecimento de pós compra de apoio ao cliente.

São considerados Canais de Distribuição força de vendas ou venda pela internet, entrega em domicílio, newsletter, atendimento presencial, entre outros. Empresas como de vendas online tais como a Americanas.com e a Submarino utilizam como Canais de Distribuição os serviços dos correios e de operadores logísticos privados, além do próprio site, links patrocinados e publicidade online. Outro elo importante entre a Proposta de Valor e Segmentos de Clientes são os Relacionamento com os Clientes (Customer Relationships), que indica como a organização interage com um determinado segmento específico de clientes e tem o objetivo de fortalecer o envolvimento entre eles e o negócio.

As definições deste bloco são motivadas pela captura, retenção de clientes, incremento das vendas e o fortalecimento da marca. Osterwarlder e Pigneur (2010, p.28) afirmam que “os relacionamentos com os clientes solicitados pelo modelo de negócio de uma empresa podem influenciar profundamente a experiência geral do cliente”. Os autores citam como exemplos de Relacionamentos com Clientes: assistência pessoal, atendimentos automatizados, destacando-se como exemplos práticos os atendimentos especializados e diferenciados oferecidos por alguns bancos, a exemplo das Agências Van Gogh do Banco Santander, outras formas de relacionamentos via telefone, redes sociais, internet banking, entre outros. Completando o Front -Stage (Figura 24) que são aqueles blocos que dependem do segmento dos clientes, estão os Fluxos de Receita (Revenue Stream), neste bloco é descrito como a organização gera receita por meio dos Segmentos de Cliente e da Proposta de Valor, registra as soluções em construção para gerar receitas

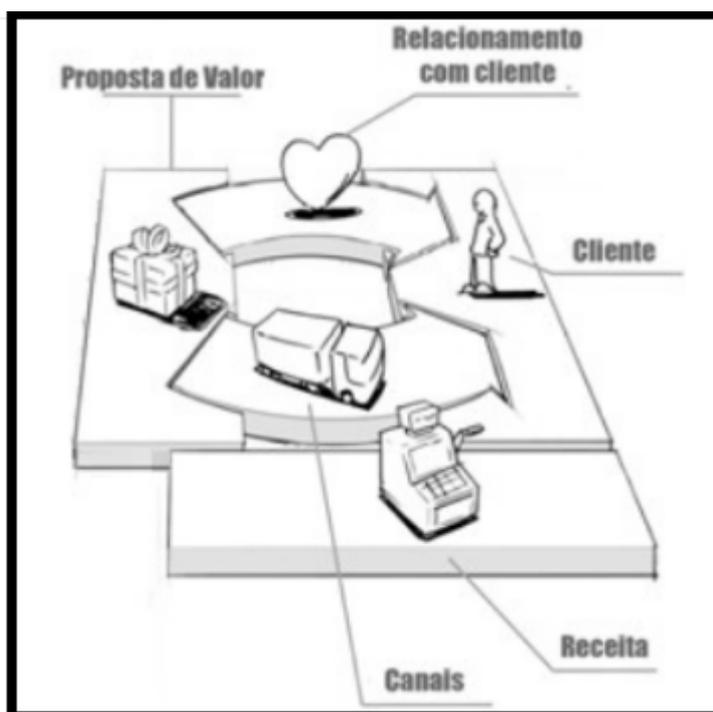


Figura 23 Canvas: Front-Stage -blocos que dependem do segmento dos clientes
 Fonte: Adaptado de Osterwalder, Pigneur (2010)

O Modelo de Negócios envolve dois tipos diferentes dos Fluxos de Receitas: “Transações de renda resultantes de pagamento único e Renda recorrente, resultante do pagamento constante, advindo da entrega de uma Proposta de Valor aos clientes ou do suporte pós-compra” (OSTERWALDER; PIGNEUR, 2010, p.30), podendo a organização gerar um ou mais Fluxos de Receita de cada segmento de clientes. São exemplos de Fontes de Receita no Canvas a venda de ativos, taxas de utilização de um serviço, locação, taxas de corretagem, publicidade, etc. Organizações tais como a Editora Globo tem retorno financeiro com venda de revistas por assinatura, banca de revista e livrarias, além do retorno de publicidade, e a empresa aérea Tam tem como fonte de receita a venda de passagens e transporte de carga. Na dimensão esquerda ou Back-Stage do Canvas estão os blocos que não dependem do segmento dos clientes (figura 6), são mais objetivos e que darão sustentação a dimensão da direita (Front-Stage).

No Back-Stage está o bloco dos Recursos-Chave (Key Resources) que compreendem aos recursos necessários para que uma organização sustente o modelo de negócios e o faça funcionar. Permite assim, da mesma forma, que a

empresa crie uma Proposta de Valor, alcance o mercado, mantenham Relacionamento com os Clientes e gere receitas. Os Recursos-Chave considerados no Canvas de acordo com os autores podem ser físicos, financeiros, intelectual ou humano, podem ser arrendados pela organização ou adquiridos de parceiros chave. São exemplos de Recursos Chave citados no Canvas as equipes de trabalho, máquinas, tecnologia, base de usuários, e especificamente para a Empresa Vale são considerados Recursos Chave as minas, o equipamento e o capital humano especializado (OSTERWALDER; PIGNEUR, 2010, p.30).

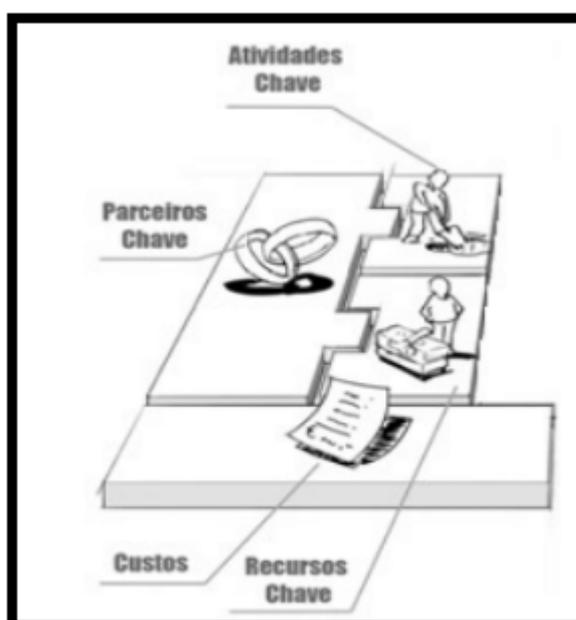


Figura 24 Canvas: Back-Stage-blocos que não dependem do segmento dos clientes
Fonte: Adaptado de Osterwalder, Pigneur (2010)

Na dimensão esquerda também estão as Atividades-Chave (key activities), que são as atividades essenciais para atender a Proposta de Valor, para construir os Canais de Comunicação, o Relacionamento com os Clientes, permitindo que modelo de negócio funcione adequadamente e com sucesso. As Atividades-Chave variam e são consideradas de acordo com o setor de atuação da organização. Essas ainda podem englobar a produção de bens ou serviços, vendas, o acompanhamento de redes sociais visando contribuir e ampliar o relacionamento com os clientes. Osterwalder e Pigneur (2010, p. 36) citam que as Atividades-Chave “para a fabricante de software Microsoft, incluem o desenvolvimento de software, para a fabricante de computadores Dell, incluem cadeia de fornecimento gestão; e para a

consultoria McKinley, incluem a resolução de problemas”. O bloco dos Parceiros-Chave (Key Partners) refere-se às organizações que são importantes para o funcionamento do modelo de negócio, que podem contribuir com as Atividades-Chave e com os Recursos-Chave tais como fornecedores de tecnologia ou outros recursos. As alianças entre a organização e os Parceiros Chave são realizadas visando otimizar seus modelos de negócio e economia de escala, reduzir os riscos ou adquirir recursos, sendo os parceiros distinguidos por quatro tipos diferentes (OSTERWALDER; PIGNEUR, 2010, p.38):

- Alianças estratégicas entre não concorrentes;
- Competição: parcerias estratégicas entre concorrentes;
- Joint ventures para desenvolver novos negócios;
- Relação comprador-fornecedor para garantir suprimentos confiáveis

Por último, mas não menos importante, está o bloco de Estrutura de Custos (Costs Structure), que enquadra os principais custos decorrentes da operação do Modelo de Negócio, tais como os custos de manutenção de equipamentos, infraestrutura, pessoal, etc., podem ser calculados de forma relativamente fácil depois de definir Recursos-Chave, Atividades -Chave e Parcerias-Chave. A Estrutura de Custo geralmente tem como as seguintes características (OSTERWALDER; PIGNEUR, 2010, p.41):

- Custos de fixos que permanecem os mesmos, apesar do volume de bens e serviços produzidos. Exemplos incluem salários, aluguéis, e instalações de fabricação físicas. Alguns negócios, tais como empresas de manufatura, são caracterizadas por uma alta proporção de custos fixos;
- Custos Variáveis: aqueles que variam proporcionalmente com o volume de bens ou serviços produzidos;
- Economias de Escala: vantagens de custo que uma empresa desfruta quando sua saída se expande. Grandes empresas, por exemplo, beneficiam de taxas mais baixas de compra a granel. Este e outros fatores causam o custo médio por unidade cair com o aumento da produção;
- Economias de escopo: vantagens de custo que uma empresa desfruta devido a um maior âmbito de operações.

Em uma grande empresa, por exemplo, as atividades de marketing ou mesmos os canais de distribuição que podem apoiar vários produtos. A aplicação do Canvas consiste no preenchimento de seus blocos com as ideias e oportunidades. O preenchimento permite uma melhor compreensão da nova ideia de negócio sugerida assim como a possibilidade de criação e melhoria de novas ideias oportunidades. Blank (2012) assume que a qualquer suposição ao modelo não passa de hipóteses conforme a figura 26.



Figura 25 Modelo de Negócios Canvas e suposições de hipóteses

Entretanto as hipóteses permitem um primeiro processo de teste experimental da nova proposta de modelo de negócios. O processo de teste das hipóteses defendido por Blank (2012) é inspirado na prática do Design Thinking o qual consiste na prototipagem rápida da proposta. Define-se assim o conceito de mínimo produto viável, ou seja, quais ações podem ser realizadas para a entrega da mínima proposta de valor para o negócio proposto. Com essa etapa de prototipagem, as hipóteses dos nove blocos podem ser sucessivamente testadas validando previamente o modelo de negócio.

5. KIT DE FERRAMENTAS DESIGN THINKING

O Design Thinking usa dez métodos com o seu objetivo de navegação dentro dos padrões que são raciocinadas, sendo possível criar novas possibilidades para reduzir riscos, administrar as incertezas inevitáveis do crescimento e com isso sempre inovar (LIEDTKA,2015).

- Visualização: Usar representações visuais para imaginar possibilidades e dar-lhes vida;
- Mapeamento da jornada: Avaliar a experiência em curso pelos olhos do cliente;
- Análise da cadeia de valores: Avaliar a cadeia de valores vigente que embasa o dia a dia do cliente;
- Mapeamento mental: Gerar insights com base em atividades de exploração e usando-as para criar critérios de design;
- Brainstorming: Criar novas possibilidades e novos modelos alternativos de negócio;
- Desenvolvimento de conceitos: Conjugar elementos inovadores numa solução alternativa coerente que pode ser explorada e avaliada;
- Testes de premissas: Isolar e testar premissas-chave que levarão ao êxito ou ao fracasso de um conceito;
- Prototipagem acelerada: Expressar um novo conceito numa forma concreta de exploração, teste e refinamento;
- Cocriação com o cliente: Envolver os clientes para participarem na criação da solução que mais se adequar às suas necessidades;
- Lançamento da aprendizagem: Criar um experimento acessível que permita aos clientes vivenciarem a nova solução por um período mais longo de tempo, para testar as premissas-chave com dados de mercado.

6 CONCLUSÃO

Esta pesquisa foi motivada pela busca por meios que possibilitassem o fortalecimento dos conceitos de inovação na atividade profissional do designer, tanto no mercado como no meio acadêmico. Para isso, o objetivo geral escolhido foi a investigação sobre o Design Thinking, Material Design como abordagem para gerar inovação, considerando também a metodologia de User Experience.

Tal objetivo foi atendido com a revisão teórica que identificou e analisou publicações sobre os temas Inovação e Design Thinking em livros, periódicos, revistas e artigos. No decorrer das pesquisas, identificou-se a importância dada pelo meio empresarial à inovação como fator de competitividade o que reforça a importância de os designers terem domínio do tema e a necessidade de inserção entre as disciplinas presentes na formação desses profissionais.

Com o intuito de contribuir para este contexto, foram relacionados alguns temas a partir das conclusões das pesquisas, hoje considerados estratégicos pelo meio empresarial para efetivação dos processos de inovação: o trabalho com equipes multidisciplinares na busca por novas ideias, a criação de valor perceptível nos produtos e/ou serviços através de experiências positivas para o usuário final, e a importância de inserir este mesmo usuário na tomada de decisão durante as etapas iniciais dos processos de conceituação e desenvolvimento dos produtos e/ou serviços. As investigações sobre a abordagem do Design Thinking foram conduzidas de forma abrangente, visando identificar seu potencial na geração de inovação.

A conclusão é que tal abordagem possibilita dar vazão a esse potencial. As pesquisas também indicam que o Design Thinking tem sua origem no design e se caracteriza como abordagem que utiliza o raciocínio adutivo para a resolução de problemas. A pesquisa também foi formada nos conceitos do Material Design juntando as técnicas de teorias e técnicas utilizadas para o desenvolvimento deste projeto de pesquisa consiste no fato da evolução da tecnologia da plataforma Android, sendo um tema que está em constante crescimento.

Aplicar os conceitos estudados na plataforma *Google Android*, que irá agregar um grande valor aos desenvolvedores, pois não precisaram ficar testando manualmente os seus sistemas. Eles terão um componente pronto e especificado para somente

realizar os testes dos aplicativos desenvolvidos, assim tornando – se mais rápido o seu desenvolvimento, visto que os teste não ocuparão tempo algum.

6.1 TRABALHOS FUTUROS

Para trabalhos futuros, outras pesquisas podem fazer uso dos conceitos apresentados neste estudo exploratório. As metodologias de UX e Design Thinking podem ser exploradas e aplicadas em quaisquer contextos, desde como apoio para ideação, prototipação e validação de soluções de tecnologia, como também na geração de novos produtos e serviços.

Um possível trabalho futuro é avaliar a adoção de UX e Design Thinking diretamente com o desenvolvimento de novos componentes para a plataforma Android com Material Design.

REFERÊNCIAS

ABLESON, W. F. **Android A Developer's Guide**. 2007 28p. MEAP Edition - Manning Early Access Program.

ANDREASSEN, Tor Wallin et al. Linking service design to value creation and service research. *Journal of Service Management*, v. 27, p. 21-29, 2016 ANDROID. The Android Source Code. Disponível em: <<https://source.android.com/source/>>. Acesso em: 06 fev.2017.

ANDROID DEVELOPER. Disponível em: <<https://developer.android.com/design/material/index.html?hl=pt-br>> Acesso em 6 fev.2017.

ANDROID, Source. System and Kernel security. Disponível em: Acesso em 6 fev. 2017.

ANDROID DEVELOPER. Disponível em: <<https://developer.android.com/guide/topics/ui/custom-components.html>> Acesso em 19 maio 2017.

ANDROID SDK. Disponível em: <<https://android-sdk.br.uptodown.com/windows>>. Acesso em 6 fev. 2017.

BLANK, S. **Entrepreneurship for the 21st Century**. Business Models and Customer Development. Endeavor Brasil. 2012.

BARBIERI, João Carlos (org.). **Organizações Inovadoras**: textos e casos brasileiros. Rio de Janeiro: FGV, 2003.

BARROS, Thiago. **Android Studio é o programa do Google para desenvolver apps para Android**. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/s/android-studio.html>>. Acesso em: 21 abril, 2017.

BOSCHI, Marco Tulio. **O design thinking como abordagem para gerar inovação: uma Reflexão**. 2012. 101 f. Dissertação (Mestrado em Design) - Universidade Anhembi Morumbi, São Paulo, 2012.

CAMERA, Ricardo . **Laboratório Mobilis Computação Móvel**: on 10 agosto de 2012. Disponível em: < [Http://www.decom.ufop.br/imobilis/customizando-componentes-no-android/](http://www.decom.ufop.br/imobilis/customizando-componentes-no-android/) > Acesso 19 maio 2017.

CAMARRI, Bruno. **Prototipação e sua importância no desenvolvimento de software**. Disponível:<<http://dextra.com.br/pt/blog/prototipacao-e-sua-importancia-no-desenvolvimento-de-software/> > Acesso em 18 set.2017.

CHIAVENATO, Idalberto. **Os Novos Paradigmas: como as mudanças estão mexendo com as empresas**. São Paulo: Atlas, 1996.

CARVALHO, Hélio Gomes de; REIS, Dálcio Roberto dos; CAVALCANTE, Márcia Beatriz. **Gestão da inovação**. Série UTFInova. Curitiba: Aymar, 2011.

COSTA, FELIPE Design; **Thinking + Inovação em Modelo de Negócio** < <http://www.innovationmanagement.se/2015/10/27/design-thinking-inovacao-em-modelo-de-negocio/> > Acesso em 10 Nov. 2017

DESIGN COUNCIL. Eleven Lessons: **Managing design in eleven global companies**. Desk research report. Londres: Design Council, 2007

Driscoll ,Jamus Demandware. **Definição do segmento mínimo viável (MVS)** < <http://www.startupsecrets.com/mvs> > Acesso em: 01 nov. 2017.

FARIA, A. O. **Programe seu Andróide**. Linux Magazine, Número 43, p 73-77, Junho de 2008.

FARTO, Guilherme de Cleva. **Abordagem Orientada a Serviços para Implementação de um Aplicativo Google Android**. 2010, 83p. Monografia de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis, São Paulo, Assis, 2010.

FIREBASE. Disponível em: <<https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html>. > Acesso em 6 fev.2017.

HAMANN, Renan. Android: **A história do robô-mascote mais famoso do mundo [ilustração]**. Disponível em: <<http://www.tecmundo.com.br/android/46241-android-a-historiado-robo-mascote-mais-famoso-do-mundo-ilustracao-.htm>>. Acesso em: 14 maio. 2017.

KORT, Joke; VERMEEREN, Arnold P.O.S.; FOKKER, Jenneke E. Conceptualizing and Measuring UX. In: Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 2, 2007. p. 57-64. Anais... British Computer Society, 2007. Disponível em: <http://dl.acm.org/citation.cfm?id=1531468>. Acesso em: 03 Nov. 2017

LIEDTKA, Jeanne; OGILVIE, Tim. **A magia do design thinking**: um kit de ferramentas para o crescimento rápido da sua empresa. São Paulo: HSM Editora, 2015.

LECHETA, Ricardo R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3. ed. São Paulo: Novatec Editora, 2013.

LIKER, Jeffrey K. **O modelo toyota: 14 princípios de gestão do maior fabricante do mundo**. Porto Alegre: Bookman, 2005

MATERIAL DESIGN. **Android Design Language**. Disponível em: <<https://www.interaction-design.org/literature/article/google-s-material-design-android-design-language>>. Acesso em 13 março, 2017.

MONTEIRO, João Bosco. **Google Android- Crie aplicações para celulares e tablets**. São Paulo: Casa do Código, 2012.

MUNARI, Bruno **Metodologia, métodos e técnicas para o desenvolvimento de produtos - definição**. Disponível em: <<http://mundoedesign.blogspot.com.br/2009/09/metodologia-metodos-e-tecnicas-para-o.html>> Acesso em 02 out. 2017.

OLIVEIRA, C. D. C. **Estudo sobre a utilização de mapas em Android para itinerários de ônibus**: Universidade tecnológica Federal do Paraná departamento acadêmico de informática curso de especialização em tecnologia java. Curitiba, 2012.

OPEN HANDSET ALLIANCE. Disponível em: <http://www.openhandsetalliance.com/press_111207.html>. Acesso em: 6 fev. 2017.

OSTERWALDER, A.; PIGNEUR, Y. **Business Model Generation** (John Wiley & sons, Eds.). New Jersey -USA, 2010. 278 p.

OGILEVE, TIM; **Magia do Design Thinking**. Um Kit de Ferramentas Para o Crescimento Rápido da Sua Empresa

PINHEIRO, Tennyson; ALT, Luis. **Design thinking brasil**: empatia, colaboração e experimentação para pessoas, negócios e sociedade. Rio de Janeiro: Elsevier, 2011. 214 p.

PINHEIRO, Tennyson. **The service startup**: inovação e empreendedorismo através do design thinking. Rio de Janeiro: Alta Books, 2015. 232 p.

PINHEIRO, Tennyson; ALT, Luis. **Design Thinking Brasil**: empatia, colaboração e experimentação para pessoas, negócios e sociedade. Rio de Janeiro: Elsevier, 2011.

PEREIRA, L. C. O.; SILVA, M. L. **Android para Desenvolvedores**. Rio de Janeiro: Editora Brasport, 2009.

ROTO, Virpi. User Experience from Product Creation Perspective. In: Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 2, 2007. p. 31-34. **Anais...**British Computer Society, 2007. Disponível em: <<http://dl.acm.org/citation.cfm?id=1531468>> Acesso em: 20 Setemb. 2017.

SHOSTACK, L.G. **Designing services that deliver**. Harvard Business Review, v. 62, n. 1, p. 133-139, 1984.

SIMON, Herbert Alexander. **The Sciences of the Artificial**. Q175.S564 1996 eb 300.1/1.

SILVA, W. B. **Design Baseado em Componentes**. Disponível em: <<http://imasters.com.br/artigo/3823/tecnologia/design-baseado-em-componentes?trace=1519021197&source=single>>. Acesso em 6 fev.2017.

SIMANTOB, Moysés; LIPPI, Roberta. **Guia Valor Econômico de Inovação nas Empresas**. São Paulo: Globo, 2003.

VIANA, Mauricio; [et al.]. **Design Thinking**: Inovação em negócios. Rio de Janeiro: MJV, 2012.