Android Layouts: Aprendendo técnicas de Layout no Android

Veja nesse artigo uma demonstração de técnicas, na teoria e na prática, de criação de layouts no Sistema Operacional Android.

 (24)  (0)

A maioria dos programadores tem uma certa dificuldade em criar layout legais, bonitos, com cores que combinem, pois de fato não é uma tarefa fácil e exige bastante esforço e criatividade.

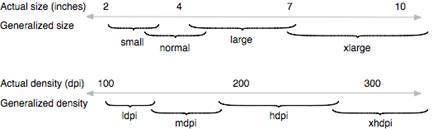
Um fato que dificulta muito a criação de layout é a ausência de materiais disponíveis para designers na web e nas documentações oficiais do Google. Por isso, este artigo tem o intuito de auxiliar programadores com poucas experiências em layouts a criarem sistemas com interfaces modernas e agradáveis aos olhos dos usuários.

**Algumas Regras**

1. Primeiro passo para criar um bom aplicativo é conhecer muito bem a plataforma que se está trabalhando. No nosso caso é o [sistema operacional Android](http://www.devmedia.com.br/curso/introducao-ao-android-sdk/306).
2. O segundo passo é conhecer a anatomia da interface do **Android** e no que ela difere de outros sistemas operacionais, como [**iOS**](http://www.devmedia.com.br/curso/ios-introducao-ao-desenvolvimento-mobile/355) e **Windows Phone**.
3. Não misture diversas forma de navegação em uma única aplicação, pois isso poderá confundir o usuário.
4. Nunca reutilize layouts do mesmo sistema em outras plataformas. Para cada plataforma que for desenvolver crie um layout próprio que se adapta ao designer nativo do sistema operacional.
5. Jamais utilize o mesmo layout para todos os seus aplicativos; para cada cliente desenhe um novo layout.

**Múltiplas telas no Android**

O android possui suporte a múltiplas telas, dependendo do layout que você irá desenvolver terá que criar layout individuais para cada tela, de forma que possa se adequar conforme ao tamanho da tela do dispositivo. A **Figura 1** demonstra como é feita a divisão dos layout por tamanho de tela.



**Figura 1**. Divisão dos layouts

Caso você tenha que desenha um layout para cada tipo tela, terá que criar uma pasta de layout para cada tipo de tela que for utilizar, um exemplo seria "layout-small" para celulares com tela menores que 426dp x 320dp. Isso permite que o sistema evite ter que se preocupar com mudanças no tamanho das telas. Abaixo segue as meditas de cada tela.

* **layout-xlarge** telas de ao menos 960dp x 720dp
* **layout-large** telas de ao menos 640dp x 480dp
* **layout-normal** telas de ao menos 470dp x 320dp
* **layout-small** telas de ao menos 426dp x 320dp

Caso deseje aprofundar mais no assunto de múltiplas telas, veja o Guia da [API do Google Android](http://www.devmedia.com.br/utilizando-mapas-com-as-apis-google-no-android-artigo-web-mobile-34/19218)em <http://developer.android.com/guide/practices/screens_support.html>

**Tipos de Dimensões no Android**

No Android as dimensões são medidas por:

|  |  |
| --- | --- |
| **Tipo** | **Descrição** |
| px (pixels) | Correspondente ao número de pixels da tela |
| in (polegadas) | Baseado no tamanho físico da tela |
| mm (milímetros) | Baseado no tamanho físico da tela |
| pt(pontos) | 1/72 de uma polegada, baseado no tamanho físico da tela |
| dpoudip | (Density-independent Pixels) Essa unidade é relativa à resolução da tela. Por exemplo se a resolução da tela é de 160 dpi, significa que um dp representa 1 pixel em um total de 160. |
| sp | (Scale-independent Pixels) Igual ao dp, mas também considera o tamanho da fonte que o usuário está utilizando. É recomendado que use essa unidade quando especificar o tamanho de uma fonte, para que esta seja automaticamente ajustada conforme as preferências da tela do usuário. |

**Escolha das Cores**

A escolha das cores é muito importante. Na **Figura 2** segue uma tabela básica de cores.



**Figura 2**. Tabela de cores

* Tente combinar sempre as cores da logo com as cores do layout.
* Cuidado com as cores: nunca misture duas cores quentes no mesmo aplicativo (ex. "vermelho com laranja"). Além de deixar a estética do sistema muito carregada, essas cores causam dores de cabeça e náuseas quando o usuário é exposto por muito tempo. Tente usar cores mais agradáveis e suaves para o fundo do sistema e nos detalhes pode utilizar cores mais fortes.
* Tente usar cores da mesma escala ou mesma tabela de cores, sempre tom e sobre tom.

**Tamanho do Objetos na Tela**

O tamanho dos objetos na tela é muito importante, pois os dispositivos mobile não utilizam recursos como mouse e teclado e sim as pontas dos dedos. É importante que os ícones sejam de acordo com a dimensão da ponta de um dedo, que é de aproximadamente 9mm - o que corresponde a cerca de 48dp. Utilize esse tamanho como base para que seu layout funcione de maneira adequada e os ícones e botões sejam tocados com precisão. Para que os objetos não fique colados no canto da tela é bom dar uma margem de 8dp.

**Tamanho dos Ícones**

* **LAUNCHER:**Importante criar um Launcher no tamanho 512x512px para ser reutilizado na Google Play.
* **ÍCONES DE AÇÃO:**Os ícones da barra de ação precisam ter 32dp.
* **ÍCONES CONTEXTUAIS:**Tamanho ideal 16dp e área focal 12dp.
* **ÍCONES DE NOTIFICAÇÃO:**Devem ter 24dp, sendo que a área focal deve ser 22dp.

**Organizar os Assets**

Organizar os arquivos com prefixo facilita muito para encontrar algo que se deseja e é uma forma de padronizar seus projetos. Observe a **Tabela 1.**

|  |  |  |
| --- | --- | --- |
| **Tipo de asset** | **Prefixo** | **Exemplo** |
| Ícone | ic\_ | ic\_devmedia.png |
| Launcher | ic\_launcher | ic\_launcher\_ nomedoapp.png |
| Ícones da action bar | ic\_menu | ic\_menu\_busca.png |
| Ícones da barra de status | ic\_stat\_notificar | ic\_stat\_notificar\_msg.png |
| Ícones das tabs | ic\_tab | ic\_tab\_recente.png |
| Ícones de dialogo | ic\_dialog | ic\_dialog\_info.png |

**Tabela 1**. PAdrão de organização.

Já conhecemos um pouco de teoria e tudo mais, agora iremos para prática.

**Tela na Horizontal e na Vertical**

Iremos criar um aplicativo que terá dois layouts diferentes: um para quando o dispositivo estiver com a tela em portrait e outro para quando o dispositivo estiver com a tela em landscape. Podemos resolver esse problema criando um main.xml (**Listagem 1**) na pasta layout e outro main.xml (**Listagem 2**) na pasta layout-land.

**Listagem 1**. Main.XML (portrait) - Dentro da Pasta Layout

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<TextView android:layout\_height="0dp" android:layout\_weight="1"

android:gravity="center" android:layout\_width="match\_parent"

android:text="DevMedia 1" />

<TextView android:layout\_height="0dp" android:layout\_weight="1"

android:gravity="center" android:layout\_width="match\_parent"

android:text=" DevMedia 2" />

</LinearLayout>

**Listagem 2.**Main.XML (landscape) - Dentro da Pasta layout-land

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal" >

<TextView android:layout\_height="match\_parent" android:layout\_weight="1"

android:gravity="center" android:layout\_width="0dp"

android:text=" DevMedia 1" />

<TextView android:layout\_height="match\_parent" android:layout\_weight="1"

android:gravity="center" android:layout\_width="0dp"

android:text=" DevMedia 2" />

</LinearLayout>

**Criando Degrades no Android**

Os recursos de degrade no Android são bastante interessantes e úteis para criação de layout mais robustos. Crie sempre os [arquivos XML](http://www.devmedia.com.br/curso/-curso-dominando-xml-com-sql-server/394) de Layout dentro das pastas Drawable. Abaixo veja alguns tipos de degrades e o respectivo código para criação.

* **RADIAL (Figura 3)**



**Figura 3.**Degrade radial

**Listagem 3.**Degrade radial

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">

<gradient

android:startColor="#33414b"

android:endColor="#242424"

android:gradientRadius="326"

android:type="radial"/>

</shape>

* **LINEAR (Figura 4)**



**Figura 4.**Degrade Linear

**Listagem 4**. Degrade Linear

<?xml version="1.0" encoding="UTF-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android" >

<gradient

android:angle="90"

android:startColor="#33414b "

android:endColor="#242424"

android:type="linear" />

</shape>

**Criando Botão com Borda e Degrade**

Crie sempre os arquivos XML de Layout dentro das pastas Drawable. Veja um exemplo de botão na**Figura 5**. Na **Listagem 5** vemos a codificação do mesmo e que serve para os botaonormal.XML|botaoselecionado.XML|botaofoco.XML.



**Figura 5.**Botão DevMedia

**Listagem 5.**botao.XML

<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http://schemas.android.com/apk/res/android" >

<item android:state\_pressed="true" android:drawable="@drawable/botaoselecionado" />

<item android:state\_focused="true"

android:drawable="@drawable/botaofoco" />

<item android:drawable="@drawable/botaonormal"/>

</selector>

Para alterar as cores do degrade e o layout do botão basta usar o código das **Listagens 6 e 7.**

**Listagem 6.**Alteração de cores

<?xml version="1.0" encoding="UTF-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android" >

<gradient

android:angle="270"

android:startColor="#2e8fc2 "

android:endColor="#006599"/>

<stroke

android:width="3dp"

android:color="#00679e " />

<corners android:radius="10dp" />

<padding

android:bottom="10dp"

android:left="10dp"

android:right="10dp"

android:top="10dp" />

</shape>

**Listagem 7.** Objeto Button Layout

<Button

android:id="@+id/btDevmedia"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:background="@drawable/botao"

android:onClick="OnClick"

android:text="DevMedia"

android:textColor="#FFFFFF"

android:textSize="25sp"

android:textStyle="bold" />

**Criando menu com ListView**

Observe como ficará menu na **Figura 6.**A codificação do mesmo está presente nas **Listagens 8 e 9.**



**Figura 6.**Menu

**Listagem 8.**Objetos Layout

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#ebebeb"

android:orientation="vertical" >

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginBottom="15dp"

android:layout\_marginLeft="15dp"

android:layout\_marginRight="15dp"

android:layout\_marginTop="15dp"

android:background="@drawable/fundo\_degrade"

android:orientation="vertical" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:background="@drawable/degrade\_title"

android:orientation="vertical" >

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:text="Menu DevMedia"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textColor="#FFFFFF"

android:textSize="25sp"

android:textStyle="bold" />

</LinearLayout>

<ListView

android:id="@+id/list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="10dp"

tools:listitem="@layout/listview\_item" >

</ListView>

</LinearLayout>

</LinearLayout>

**Listagem 9**. listview\_item.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="50dp"

android:background="@drawable/degrade\_fundo\_lista" >

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_alignParentBottom="true"

android:layout\_toLeftOf="@+id/ivImg"

android:orientation="vertical"

android:padding="5dip" >

<TextView

android:id="@+id/title"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true"

android:textSize="28sp"

android:textStyle="bold" />

</LinearLayout>

<TextView

android:id="@+id/tvMsg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:layout\_marginRight="10dp"

android:textColor="@color/vermelho"

android:textSize="20sp"

android:textStyle="bold" />

</RelativeLayout>

**Background com Imagem**

Usando o background com imagem é melhor, pois não distorce ao virar a tela do dispositivo. Esse modelo esta ativado o modo de repetir a imagem. Veja a codificação na **Listagem 10**.

**Listagem 10**. background

<?xml version="1.0" encoding="utf-8"?>

<bitmap xmlns:android="http://schemas.android.com/apk/res/android"

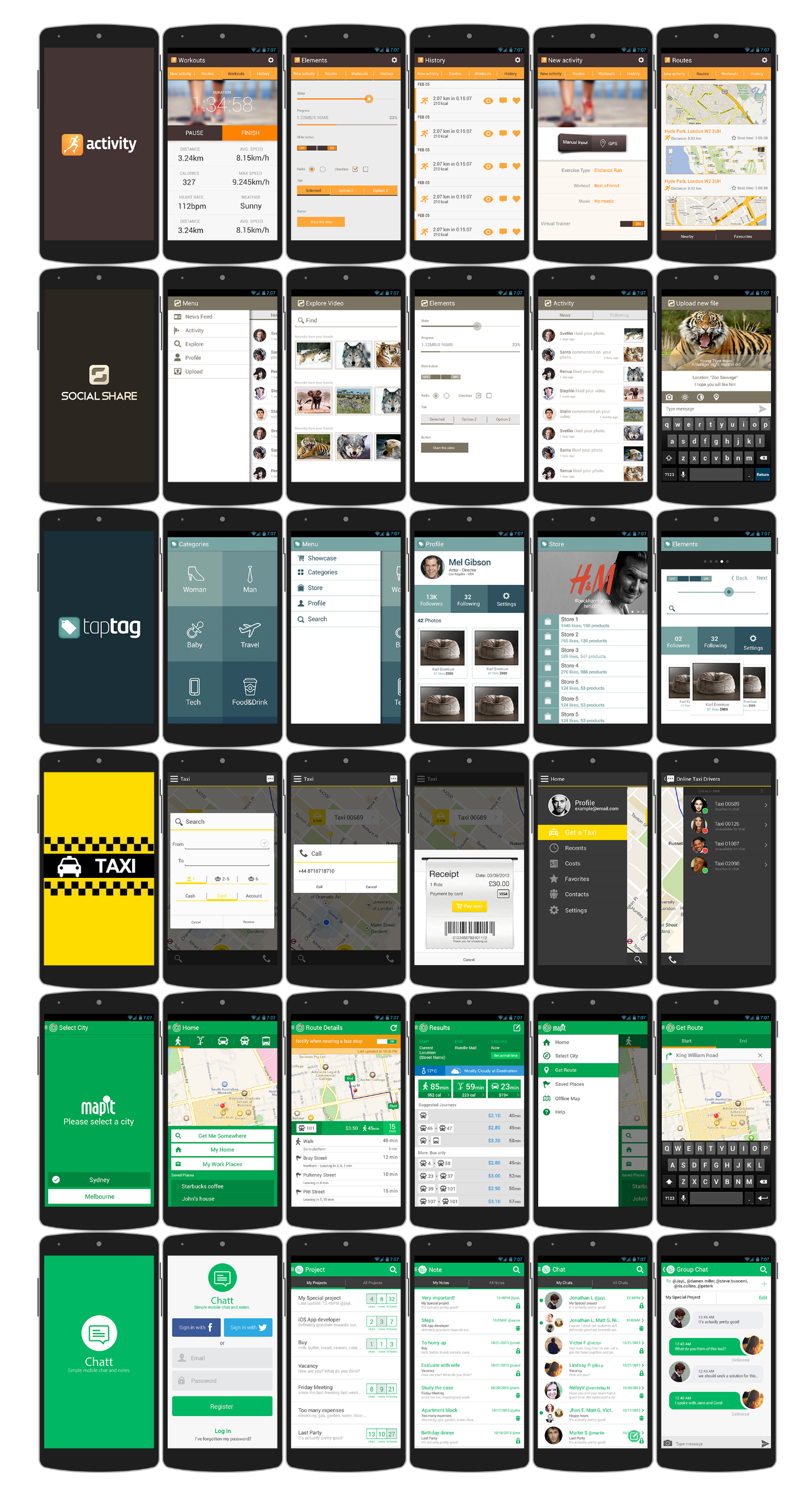
android:src="@drawable/imagemdefundo"

android:tileMode="repeat"

android:dither="true" />

**Alguns Modelos de Layout**

Na **Figura 7**segue alguns modelos de Layout para que você tenha uma base de como devem ser as cores, os ícones, as fontes e etc...



**Figura 7**. Modelos de layout