



Padrões de projeto e de código

Este documento foi criado com o intuito de documentar as convenções de código utilizadas durante o curso. Os padrões foram retirados da página do Android escritas pela própria Google [1].

Idioma

O idioma utilizado na construção dos aplicativos será inglês, uma vez que os métodos por padrão criados no Android também são inglês. Dessa maneira, não teremos métodos em diferentes linguagens.

Os comentários serão português para compreensão mais clara já que se trata de um curso destinado para o Brasil.

Linguagem Java

Não ignore exceções

```
void setServerPort(String value) throws NumberFormatException {  
    serverPort = Integer.parseInt(value);  
}
```

Embora seja tentador escrever um código desse tipo ou que esse tipo de erro nunca acontecerá, esse trecho de código pode ser tornar uma mina para futuras manutenções no código. Exceções devem ser tratadas e dependem do contexto.

- Lançar a exceção para o método chamador

```
void setServerPort(String value) throws NumberFormatException {  
    serverPort = Integer.parseInt(value);  
}
```

Convenções de nomes para atributos

- Campos não públicos e não estáticos começam com m.
- Campos estáticos começam com s.
- Outros campos começam com a primeira letra minúscula.
- Campos estáticos públicos são escritos MAIUSCULO_COM_UNDERLINE.

```
public class MyClass {  
    public static final int SOME_CONSTANT = 42;  
    public int publicField;  
    private static MyClass sSingleton;  
    int mPackagePrivate;  
    private int mPrivate;  
    protected int mProtected;  
}
```

Convenções Android

Nome de arquivos

Devem ser escritos em CammelCase. Exemplo: SignInActivity, SignInFragment, ChangePasswordDialog.

Nome de arquivos de layout

Nomes de layout devem ser prefixados com os elementos que representam:

Componente	Classe	Nome do Layout
Activity	UserProfileActivity	activity_user_profile.xml
Fragment	SignUpFragment	fragment_sign_up.xml
Dialog	ChagePasswordDialog	dialog_change_password.xml

Arquivos de XML

Quando um XML não tiver nenhum conteúdo dentro das tags, deve usar “self closing” tags.

Bom:

```

<TextView
    android:id="@+id/text_view_profile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

Ruim:

```

<TextView
    android:id="@+id/text_view_profile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
</TextView>

```

Identificadores de elementos

Identificadores de elementos são escritos em minúscula_com_underline.

Elemento	Prefixo
TextView	text_
EditText	edit_
Button	button_
ImageView	image_
Menu	menu_

Exemplo de ImageView:

```

<ImageView
    android:id="@+id/image_profile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

Exemplo de menu:

```

<menu>
    <item
        android:id="@+id/menu_done"
        android:title="Done" />
</menu>

```

Fontes

1. SourceAndroid. **AOSP Java Code Style for Contributors**. Disponível em:
<<https://source.android.com/setup/contribute/code-style>>. Acesso em 5 de maio de 2019.
2. Ribot GitHub, Android guidelines. Disponível em
<[https://github.com/ribot/android-guidelines/blob/master/project_and_code_guidelines.m
d](https://github.com/ribot/android-guidelines/blob/master/project_and_code_guidelines.md)>. Acesso em 5 de maio de 2019.