



# Sumário

- Hello World
- Mostrar como criar telas
  - Activity
  - Layouts
  - Componentes
- Ensinar o básico de Programação Orientada a Objetos
  - Interface
  - Classe Abstrata
  - Hierarquia
  - Listener
- Fragments
  - Interação com a Activity
- Interação com o Banco de Dados

# It's coding time: A new Hello World

São necessários definir os seguintes itens ao criar um novo projeto Android:

1. Nome do aplicativo
2. Versão mínima do Android que o aplicativo vai rodar
3. Tipo de Activity

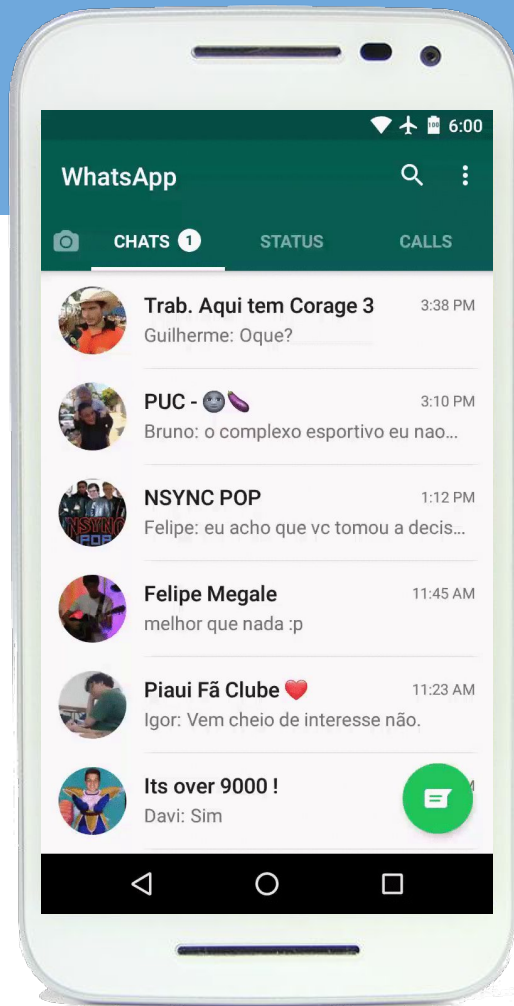
*No projeto vou escolher respectivamente:*

1. *Receitas da Mamãe*
2. *API 19*
3. *EmptyActivity*

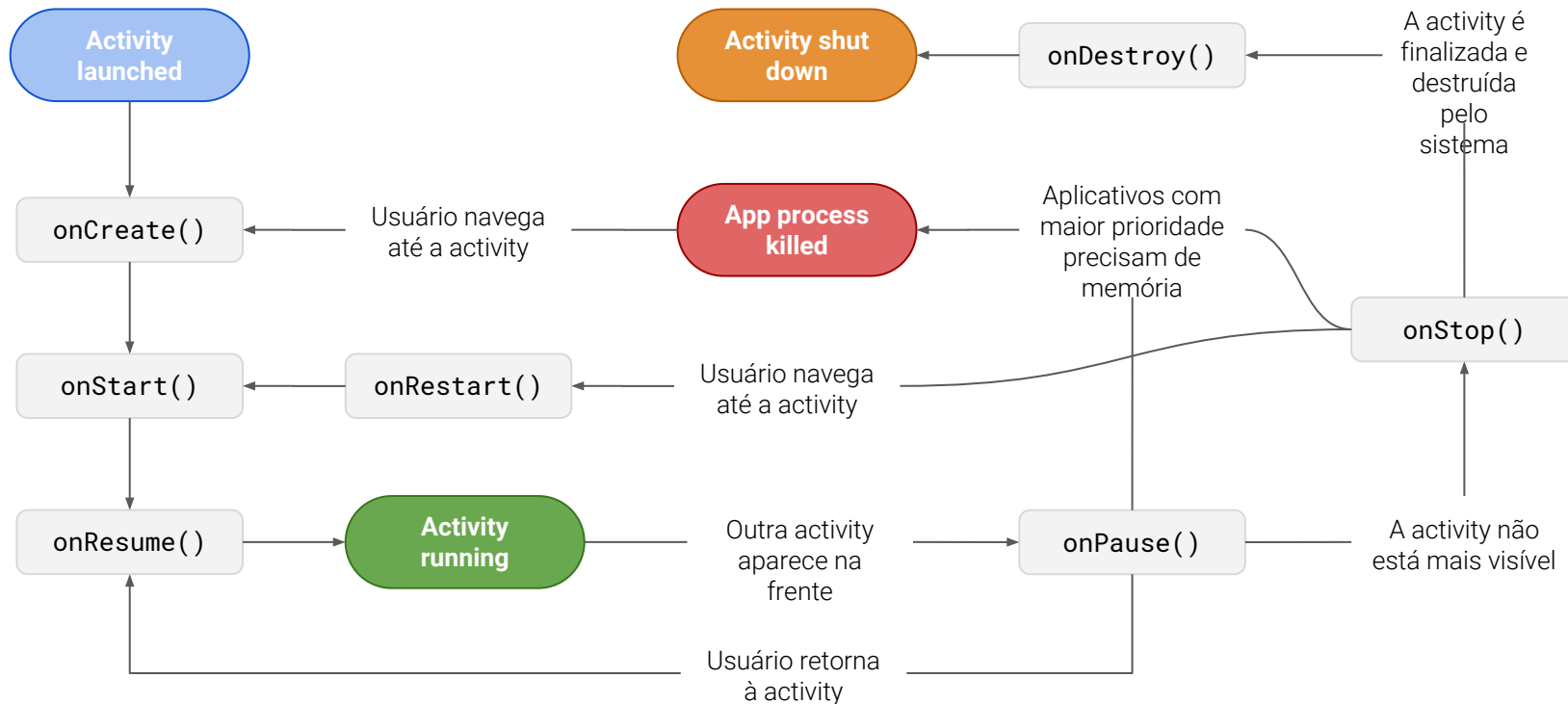
# Activity

*“An activity is a single, focused thing that the user can do.”*

Agrupa um conjunto de outros componentes que, sob um mesmo contexto, apresentam ou adquirem informações.



# Ciclo de vida de uma Activity



# Layouts

Existem alguns layouts pré-definidos que ajudam na montagem das telas:

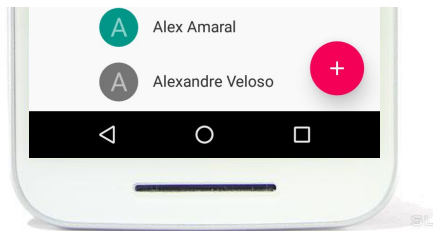
- **FrameLayout:** Principalmente utilizado para Fragments, é simplesmente um layout genérico sem uma característica especial;
- **LinearLayout:** Alinha os elementos linearmente (tanto vertical quanto horizontal);
- **CoordinatorLayout**<sup>1</sup>: Garante que os componentes poderão ter comportamentos baseados uns nos outros.

<sup>1</sup> Precisa importar o pacote de design, veja o arquivo [build.gradle](#))

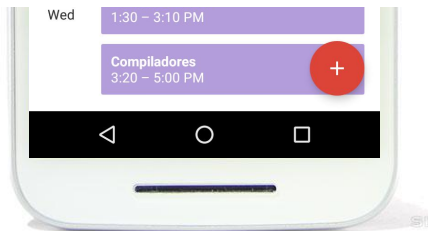
# FloatingActionButton

Introduzido na API 21 (Lollipop) junto ao Material Design.

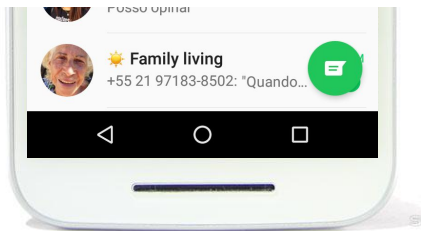
Seu objetivo é deixar mais acessível uma funcionalidade que pode ser constantemente utilizada.



Google Contatos



Google Agenda



Whatsapp



<sup>1</sup> Precisa importar o pacote de design, veja o arquivo [build.gradle](#))

# Componentes simples

Dentre a grande variedade de componentes existentes, abaixo estão listados os mais recorrentemente usados:

- **EditText:** Caixa de texto para o usuário entrar com dados
- **TextView:** Mostra uma informação estática (o usuário não pode alterar seu conteúdo)
- **Button:** Botão genérico que permite ao usuário fazer alguma ação.
- **ScrollView:** Componente cujo conteúdo pode ser “rolável”. Obs.: Pode possuir apenas um filho.
- **ImageView:** Permite mostrar uma imagem.



## It's coding time 2: The Code strikes back

Criar classe Recipe com título (`title String`), lista de ingredientes (`ingredients ArrayList<String>`), modo de preparo (`instructions ArrayList<String>`)

Colocar um `FloatingActionButton` para criar uma nova `Receita` no `MainActivity`.

Criar tela de cadastro de Receitas (`RecipeActivity`).

# Comunicando Activities

```
// Apenas chamar outra Activity
```

```
startActivity(Intent);
```

```
// Chamar outra Activity, mas esperando um resultado
```

```
startActivityForResult(Intent, int /* Request Code */);
```

# Pegando o Resultado de uma Activity

```
// Sobrescrever onActivityResult
@Override
protected void onActivityResult( int requestCode,
                                  int resultCode,
                                  Intent data) {

    /* Request code foi passado no startActivityForResult().
     * Result code foi informado pela Activity que terminou.
     * Data é qualquer informação que a Activity finalizada
     * precisou passar para outra Activity
     */
}
```

## It's coding time 3: The Return of the code

No `MainActivity`, criar um atributo chamado `receitas` (`ArrayList<Recipe>`) para armazenar as receitas criadas.

Fazer o `MainActivity` chamar o `RecipeActivity` e aguardar o retorno.

Adicionar à `receitas` a receita recém criada.