

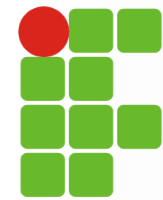
INSTITUTO FEDERAL
SÃO PAULO
Campus Boituva

Programação para Dispositivos Móveis

Prof. Dr. Antonio Miguel Batista Dourado
dourado@ifsp.edu.br

Navegação Entre Telas

- Até o momento, aprendemos a criar uma Activity e manipulá-la. Em breve aprenderemos também sobre seu ciclo de vida.
- Uma aplicação Android, porém, não se resume a apenas uma Activity (ou uma tela).
 - É comum haver a troca de telas nas aplicações. Tal troca de telas é resultado do início de uma Activity dentro de outra.



Navegação Entre Telas

- Navegar entre telas em uma aplicação Android implica no possível uso de um dos dois seguintes métodos
 - `startActivity(intent)`
 - `startActivityForResult(intent)`
 - Este último será visto posteriormente

Navegação Entre Telas

- O método `startActivity` recebe um importante parâmetro: uma `intent`.
- Intents são um dos principais conceitos do desenvolvimento Android e significam uma intenção sobre algo.
 - Serão melhor explicadas em uma aula específica sobre elas.

Navegação Entre Telas

- Vamos criar duas Activities
 - A primeira Activity se chamará NovaTela
 - Deverá conter apenas um TextView com os dizeres: “Troca de Tela Efetuada!”
 - A segunda Activity se chamará AbrirTela
 - Deverá conter um botão escrito: “Nova Tela”
 - Ao clicar no botão, a Activity NovaTela será invocada

Navegação Entre Telas

■ IMPORTANTE!

- Neste exemplo, ambas Activity's serão adicionadas no AndroidManifest.xml, porém apenas a Activity AbrirTela deverá conter as tags abaixo, para poder ser iniciada e aparecer no menu de app's:

```
<intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>
```

activity_nova_tela.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".NovaTela">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Troca de Tela Efetuada!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        />

</androidx.constraintlayout.widget.ConstraintLayout>
```



activity_abrir_tela.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AbrirTela">

    <Button
        android:id="@+id/btTrocaTela"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Abrir Nova Tela"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



AbrirTela.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_abrir_tela);
    Button btTroca = (Button) findViewById(R.id.btTrocaTela);

    btTroca.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent it = new Intent(view.getContext(), NovaTela.class);
            startActivity(it);
        }
    });
}
```

AbrirTela.java

`@Override`

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_abrir_tela);  
    Button btTroca = (Button) findViewById(R.id.btTrocaTela);  
  
    btTroca.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            Intent it = new Intent(view.getContext(), NovaTela.class);  
            startActivity(it);  
        }  
    });  
}
```

Inicialização da
Activity

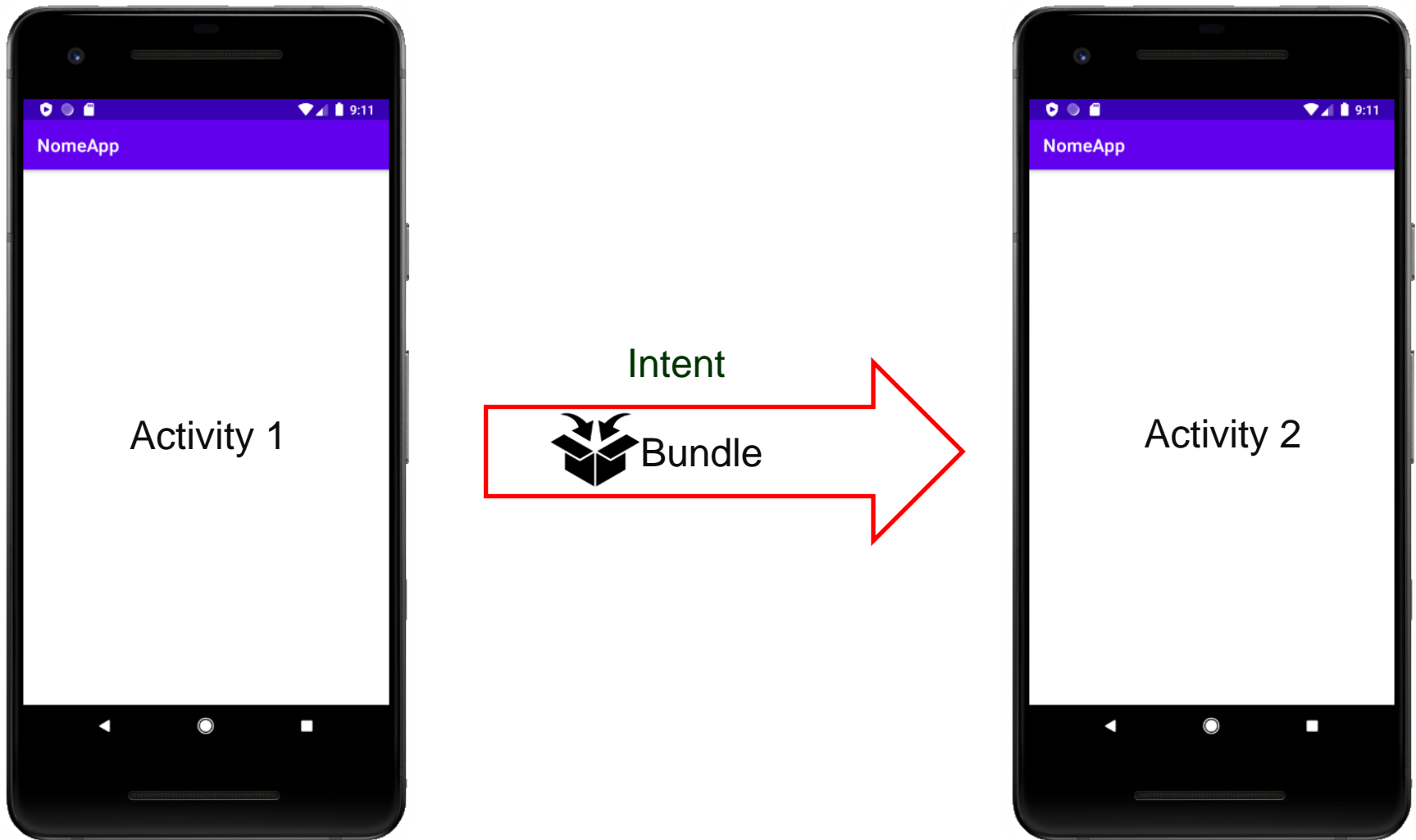
Contexto de origem
da Intent

Classe da Activity
destino

Parâmetros entre Telas

- Agora que já se sabe como fazer a troca entre telas, vamos aprender como enviar parâmetros de uma tela para a outra.
- Mas como?
 - Intent!

Parâmetros entre Telas



Parâmetros entre Telas

- O objeto Intent é passível de carregar parâmetros consigo e entregá-los para a Activity alvo.
- Os parâmetros são carregados por meio de um bundle (pacote) que é anexado junto ao Intent
 - Método putString em bundle
 - Método putExtras em intent

Parâmetros entre Telas

- Como fica?
 - Para enviar

...

```
Intent it = new Intent(classeAtual, classeAlvo);  
Bundle params = new Bundle();  
params.putString("parâmetro", "valor");  
it.putExtras(params);  
startActivity(it);
```

...

Parâmetros entre Telas

■ Como fica?

□ Para receber

...

```
Intent it = this getIntent();
```

```
if(it != null){
```

```
    Bundle params = it.getExtras();
```

```
    if(params != null){
```

```
        params.getString("parâmetro");
```

```
    }
```

```
}
```

...

Parâmetros entre Telas

- Vamos criar um exemplo onde uma Activity lê o nome do usuário e o envia por parâmetro para outra Activity que dá as boas vindas ao usuário.

Parâmetros entre Telas

■ Primeira Activity: BoasVindas

- Responsável por “desempacotar” o bundle, extrair o parâmetro que contém o nome e escrever “Bem Vindo(a), Nome” em um TextView

■ Segunda Activity: EntradaNome

- Responsável por fazer a leitura do nome em um EditText e enviar o nome pelo parâmetro “nome” para a Activity BoasVindas. O nome será enviado ao clique de um Button “Ok”.

activity_boas_vindas.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BoasVindas">

    <TextView
        android:id="@+id/tvBoasVindas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

BoasVindas.java

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_boas_vindas);  
  
    TextView tvBoasVindas = (TextView) findViewById(R.id.tvBoasVindas);  
  
    Intent it = this getIntent();  
    if(it != null){  
        Bundle params = it.getExtras();  
        if(params != null){  
            String nome = params.getString("nome").trim();  
            if(nome.isEmpty())  
                nome = "Desconhecido(a)";  
            tvBoasVindas.setText(String.format("Seja bem vindo(a), %s!", nome));  
        }  
    }  
}
```

activity_entrada_nome.xml

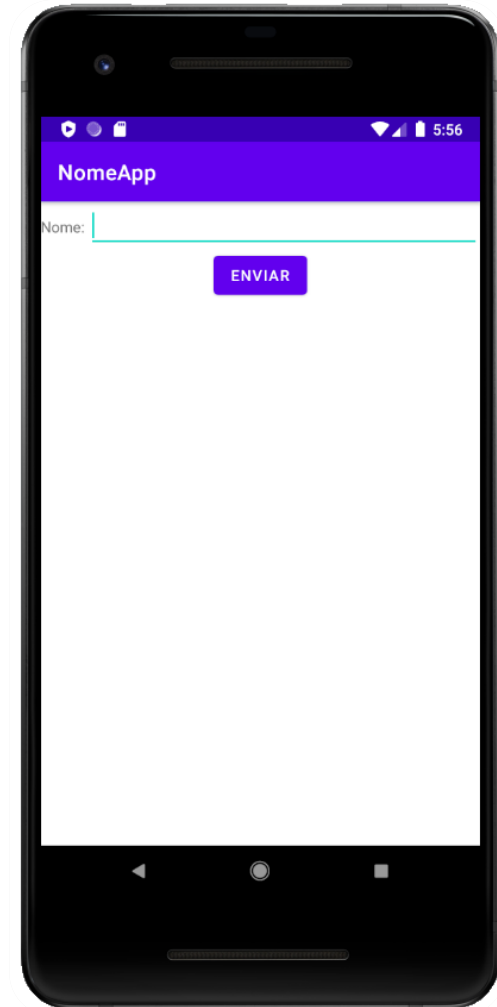
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".EntradaNome">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nome: "
        app:layout_constraintBaseline_toBaselineOf="@+id/txtNome"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/txtNome"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView2"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btEnviar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enviar"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtNome" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



EntradaNome.java

@Override

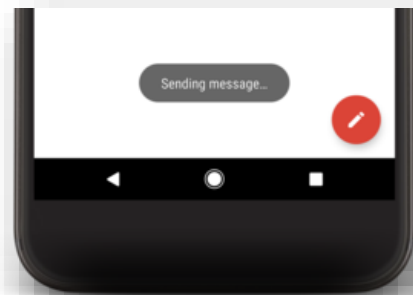
```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_entrada_nome);  
  
    final EditText txtNome = (EditText) findViewById(R.id.txtNome);  
    Button btEnviar = (Button) findViewById(R.id.btEnviar);  
  
    btEnviar.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            String nome = txtNome.getText().toString().trim();  
            if(nome.isEmpty())  
                Toast.makeText(view.getContext(), "Informe um nome!", Toast.LENGTH_SHORT).show();  
            else{  
                Intent it = new Intent(view.getContext(), BoasVindas.class);  
                Bundle params = new Bundle();  
                params.putString("nome", nome);  
                it.putExtras(params);  
                startActivity(it);  
            }  
        }  
    });  
}
```

Toast

- Toasts são pequenos popups que podem ser usados para apresentar mensagens curtas para o usuário. Sintaxe

```
Toast.makeText(Contexto, "Mensagem", Toast.LENGTH_SHORT).show();
```

```
Toast.makeText(Contexto, "Mensagem", Toast.LENGTH_LONG).show();
```



Atalho para Parâmetros

- É possível encurtar o código anterior para a passagem de parâmetros.
- O código pode ser encurtado removendo o uso de bundle e utilizando o recurso de extras direto do intent.
- No envio:

```
objetoIntent.putExtra("parâmetro","valor");
```
- No recebimento:
 - `Tipo var = objetoIntent.getTipoExtra("parâmetro");`
 - A conversão é feita automaticamente para o Tipo!

Atalho para Parâmetros

- Vamos alterar o código anterior para utilizar os atalhos.

Atalho para Parâmetros

```
public void onClick(View view) {  
    String nome = txtNome.getText().toString().trim();  
    if(nome.isEmpty())  
        Toast.makeText(view.getContext(),"Informe um nome!",Toast.LENGTH_SHORT).show();  
    else{  
        Intent it = new Intent(view.getContext(),BoasVindas.class);  
        Bundle params = new Bundle();  
        params.putString("nome",nome);  
        it.putExtras(params);  
        startActivity(it);  
    }  
}
```

EntradaNome.java



```
public void onClick(View view) {  
    String nome = txtNome.getText().toString().trim();  
    if(nome.isEmpty())  
        Toast.makeText(view.getContext(),"Informe um nome!",Toast.LENGTH_SHORT).show();  
    else{  
        Intent it = new Intent(view.getContext(),BoasVindas.class);  
        it.putExtra("nome",nome);  
        startActivity(it);  
    }  
}
```

Atalho para Parâmetros

```
Intent it = this.getIntent();  
if(it != null){  
    Bundle params = it.getExtras();  
    if(params != null){  
        String nome = params.getString("nome").trim();  
        if(nome.isEmpty())  
            nome = "Desconhecido(a)";  
        tvBoasVindas.setText(String.format("Seja bem vindo(a), %s!", nome));  
    }  
}
```

BoasVindas.java



```
Intent it = this.getIntent();  
if(it != null){  
    String nome = it.getStringExtra("nome").trim();  
    if(nome.isEmpty())  
        nome = "Desconhecido(a)";  
    tvBoasVindas.setText(String.format("Seja bem vindo(a), %s!", nome));  
}
```

Exercício

- Crie uma tela de Login com os campos “Usuário” e “Senha”, além do botão “Entrar”.
 - Caso seja digitado “root” como usuário e “admin” como senha, o aplicativo deverá enviar o usuário para uma tela de boas vindas com seu nome de usuário.
 - Caso usuário ou senha estejam incorretos, deverá ser informado erro na tela de Login.
- Na tela de boas vindas deverá haver um botão “Sair”, que enviará o usuário de volta para a tela de Login.
 - Pesquise como encerrar uma Activity.

Bibliografia Básica

- DEITEL, P.; DEITEL, H. **Android para Programadores**. Porto Alegre: Bookman, 2012.
- LECHETA, RICARDO R. **Google Android**. São Paulo: Novatec, 2010.
- PEREIRA, Lucio Camilo Oliva; SILVA, Michel Lourenço da. **Android para Desenvolvedores**. Rio de Janeiro: Brasport, 2012

Bibliografia Complementar

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: Guia do Usuário**. 2a Ed. Rio de Janeiro: Elsevier, 2005.
- DEITEL, P.; DEITEL, H. **Java Como Programar**. 8. ed. São Paulo: Pearson Education, 2010.
- FOWLER, Martin. **UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**. 3a Ed. São Paulo: Bookman, 2004
- SIX, JEFF. **Segurança de Aplicativos Aplicativos Android**. São Paulo: Novatec, 2012.
- SMITH, DAVE E FRIESEN, JEFF. **Receitas Android - Uma Abordagem para Resolução de Problemas**. São Paulo. Ciência Moderna, 2012.