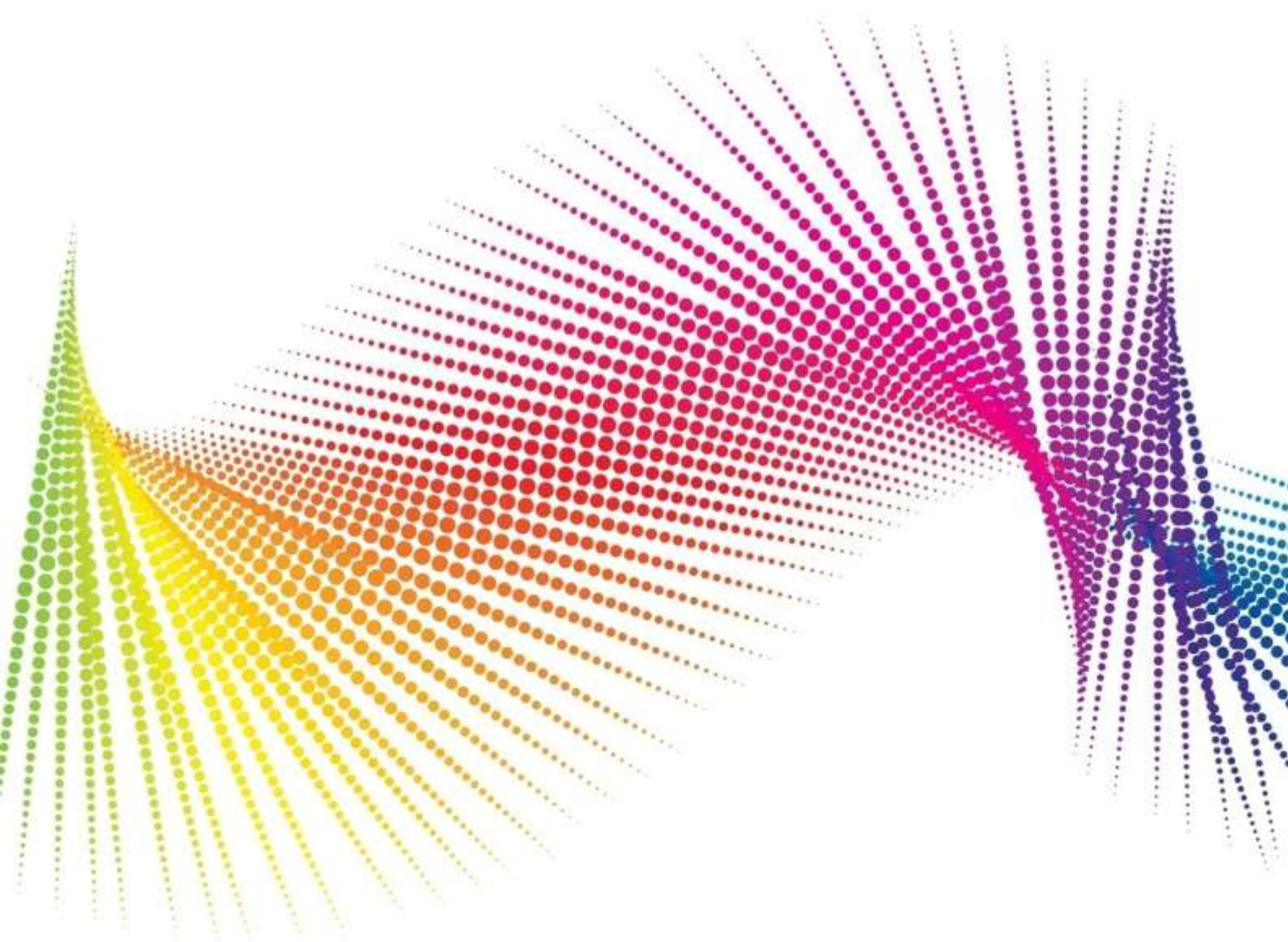


# Computação Móvel

Aula 13



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

## **Aula 13: Mensagens em android**

**Objetivo:** Demonstrar a utilização dos processos atrelados ao envio de mensagens oferecidas pelo sistema android para que se obtenha melhora significativa nos acessos às estruturas presentes na construção das aplicações.

### **Mensagens (intents)**

Três dos principais componentes de uma aplicação – atividades, serviços e receptores de broadcast – são ativados por meio de mensagens, chamadas de intents, uma facilidade para realizar, em tempo de execução, a ligação entre os componentes da mesma ou de outra aplicação. O objeto intent em si é uma estrutura de dados passiva segura com uma descrição abstrata de uma operação a ser realizada – ou, muitas vezes, no caso de transmissões, a descrição de algo que já aconteceu e está sendo informado. Existem mecanismos separados para fornecer as intents para cada tipo de componente:

Um objeto intent é passado para `Context.startActivity()` ou `Activity.startActivityForResult()` para iniciar uma atividade ou para que uma atividade já existente faça algo novo. (Também pode ser passado para `Activity.setResult()` para retornar informações para a atividade que chamou `startActivityForResult()`.)

Um objeto é passado para intent `Context.startService()` para iniciar um serviço ou fornecer novas instruções para um serviço contínuo. Do mesmo modo, as intents podem ser passadas para `Context.bindService()` para estabelecer uma ligação entre o componente de chamada e serviço de destino. Ele pode opcionalmente iniciar o serviço se ele já não estiver em execução.

Objetos intents passados para qualquer um dos métodos de transmissão (por exemplo, `Context.sendBroadcast()`, `Context.sendOrderedBroadcast()`, ou `Context.sendStickyBroadcast()`) são entregues a todos os receptores de broadcast interessadas.

Em cada caso, o sistema localiza a activity android apropriada, serviço, ou um conjunto de receptores de broadcast para responder à intent, e instanciá-los se necessário.

Não há sobreposição dentro destes sistemas de mensagens: as intents de emissões são entregues somente para transmitir receptores nunca, a activity ou serviços. Uma intent passada para `startActivity()` é fornecida apenas a uma atividade, nunca a um receptor do serviço ou de transmissão, e assim por diante.

## **Objetos intent**

Um objeto intent é um conjunto de informações. Ele contém informação de interesse para o componente que recebe a intent (tal como a ação a ser tomada, e os dados para agir) mais informação de interesse para o sistema android (tais como a categoria de componentes que devem lidar com o propósito e instruções sobre como iniciar uma atividade de destino). Principalmente, podem conter o seguinte:

### **Nome do componente**

O nome do componente que deve lidar com a intent. Este campo é um objeto `ComponentName` – uma combinação do nome da classe totalmente qualificado do componente de destino (por exemplo, `"com.example.project.app.FreneticActivity"`) e o nome do pacote definido no arquivo de manifesto do aplicativo em que o componente reside (por exemplo, `"com.example.project"`). A parte do pacote do nome do componente e definir o nome do pacote no manifesto não necessariamente têm que corresponder.

O nome do componente é opcional. Se for definido, o objeto intent é entregue a uma instância da classe designada. Se não estiver definido, o android usa outras informações no objeto para localizar um alvo adequado.

O nome do componente é configurado por `setComponent()`, `SetClass()`, ou `setClassName()` e lido por `getComponent()`.

### **Ação**

Uma string com o nome da ação a ser realizada, ou no caso de transmissão de intents, a ação que ocorreu e está sendo relatada. A classe intent define um número de constante de ações, incluindo estes:

Constante Alvo	Componente
ACTION_CALL	activity Iniciar uma chamada telefônica.
ACTION_EDIT	activity Dados da activity em exibição para o usuário editar.
ACTION_MAIN	activity Iniciar a atividade principal de uma tarefa, sem entrada e sem saída de dados retornado.
ACTION_SYNC	activity Sincronizar dados em um servidor com dados no dispositivo móvel.
ACTION_BATTERY_LOW	broadcast receiver Aviso de que a bateria está fraca.
ACTION_HEADSET_PLUG	broadcast receiver Um fone de ouvido foi conectado ao dispositivo, ou removido dele.
ACTION_SCREEN_ON	broadcast receiver Receptor na tela foi ligado.

A ação determina em grande parte como o resto da intent é estruturada – em particular os dados e campos extras – tanto como um nome de método determina um conjunto de argumentos e um valor de retorno. Por esta razão, é uma boa ideia usar nomes de ação que são tão específicos quanto possível, e para acoplá-los firmemente a outros campos da intent. Em outras palavras, em vez de definir uma ação isolada, definir um protocolo completo para os objetos manipularem.

A ação de um objeto intent é definido pelo `setAction()` e lido por `getAction()`.

## Dados

O URI dos dados a ser executado e o tipo MIME de dados. Ações diferentes são combinadas com diferentes tipos de especificações de dados. Por exemplo, se o campo de ação é ACTION\_EDIT, o campo de dados deve conter o URI do documento a ser exibido para edição. Se a ação for ACTION\_CALL, o campo de dados seria um telefone, URI com o número a ser chamado. Da mesma forma, se a ação for ACTION\_VIEW e o campo de dados é um http:URI, a atividade de recebimento seria chamado para baixar e exibir todos os dados que a URI se refere.

Ao combinar a intent de um componente que é capaz de manipular os dados, muitas vezes é importante saber o tipo de dados (seu tipo MIME), além de sua URI. Por exemplo, um componente capaz de exibir dados de imagem não deve ser chamado a desempenhar um arquivo de áudio.

Em muitos casos, o tipo de dados pode ser inferida a partir da URI – particularmente contentr: URIs, o que indica que os dados estão localizados no dispositivo e controlado por um fornecedor de conteúdos. Mas o tipo também pode ser explicitamente definido no objeto intent. O método setData() especifica os dados apenas como uma URI, setType() especifica apenas como um tipo de MIME e setDataAndType() especifica tanto como um URI e um tipo MIME. A URI é lido por getData() e o tipo de getType().

## Categoria

Uma sequência de caracteres contendo informações adicionais sobre o tipo de componente que deve lidar com a intent. Qualquer número de descrições de categorias pode ser colocada em um objeto. Como faz para as ações, a classe intent define constantes categorias diversas, incluindo:

Constante	Significado
CATEGORY_BROWSABLE	A atividade alvo pode ser facilmente invocada pelo navegador para exibir dados referenciados por um link – por exemplo, uma imagem ou uma mensagem de e-mail.
CATEGORY_GADGET	A atividade pode ser incorporado dentro de

CATEGORY_HOME	outra atividade que hospeda gadgets. A atividade exibe a tela inicial, a primeira tela que o usuário vê quando o dispositivo é ligado ou quando o botão Home é pressionado.
CATEGORY_LAUNCHER	A atividade pode ser a atividade inicial de uma tarefa e está listado na tela do menu de nível superior.
CATEGORY_PREFERENCE	O alvo é um painel de preferências.

O método `AddCategory()` coloca uma categoria em um objeto `intent`, `removeCategory()` exclui uma categoria adicionada anteriormente, e `GetCategories()` é o conjunto de todas as categorias atualmente no objeto.

## Extras

Pares de chave-valor para informações adicionais que devem ser entregues ao componente para que ele possa manusear a `intent`. Assim como algumas ações estão emparelhadas com determinados tipos de dados URIs, alguns dados estão emparelhados com extras específicos. Por exemplo, uma `intent ACTION_TIMEZONE_CHANGED` tem um "fuso horário" extra que identifica o novo fuso horário, e `ACTION_HEADSET_PLUG` tem um "estado" extra indicando se o fone de ouvido está conectado ou desconectado, bem como um "nome" extra para o tipo de fone de ouvido. Se você tivesse que inventar uma ação `SHOW_COLOR`, o valor da cor seria definido em um par chave-valor extra.

O objeto `intent` tem uma série de métodos `put...()` para a inserção de vários tipos de dados adicionais e um conjunto semelhante de `get...()` para ler os dados. Estes métodos são comparáveis àquelas que para objetos `Bundle`. Na verdade, os extras podem ser instalados e lidos como um pacote usando os métodos `putExtras()` e `getExtras()`.



## **Intents podem ser divididas em dois grupos**

Explicit intents que designam o componente de destino pelo seu nome (o campo nome do componente, mencionado anteriormente, têm um conjunto de valores). Desde os nomes dos componentes que geralmente não são conhecidos para os desenvolvedores de outras aplicações, explicit intents são normalmente utilizados para aplicação interna de mensagens – como uma atividade que inicia um serviço subordinado ou lançamento de uma activity irmã.

Implicit intents não designam o componente de destino pelo seu nome (o campo para o nome do componente está em branco). Implicit intents são frequentemente usados para ativar os componentes em outras aplicações.

Uma estratégia diferente é necessária para as intenções implícitas. Na ausência de um alvo designado, o sistema android deve encontrar o melhor componente (ou componentes) para lidar com a intent – uma atividade única ou serviço para executar a ação pedida ou o conjunto de receptores de broadcast para responder à mensagem publicitária. Isto será realizado por meio da comparação do conteúdo do objeto intent e filtros intents, as estruturas associadas com os componentes que podem potencialmente receber intents. Se um componente não tem qualquer filtro, pode receber apenas explicit intents. Um componente com filtros podem receber qualquer uma das duas.

Apenas três aspectos de um objeto intent são consultados quando o objeto é testado contra um filtro de intent:

- Ação
- Dados (tanto do tipo URI e dados)
- Categoria

Os extras não têm qualquer papel na resolução de qual componente recebe uma intenção.

## **Exercício**

Para termos certeza de que entendemos a aula, vamos realizar uma atividade sobre intents.





## EXERCÍCIOS

Agora, veja os exercícios disponíveis acessando o AVA, ou via QR Code\*. Não deixe de visualizar esses exercícios, pois eles fazem parte da sequência desta aula e, portanto, são essenciais para a aprendizagem.



\* O QR Code é um código de barras que armazena links às páginas da web. Utilize o leitor de QR Code de sua preferência para acessar esses links de um celular, tablet ou outro dispositivo com o plugin Flash instalado.

## Próxima aula

Uma vez que já temos o conhecimento sobre o funcionamento das intents utilizadas na arquitetura android, podemos dar o próximo passo em nosso aprendizado. Na próxima aula – **Aula 14** – aprenderemos sobre os filtros associados às intents no android.

## REFERÊNCIAS

LECHETA, Ricard R. *Android: aprenda a criar aplicações para dispositivos móveis com o android SDK*. 2. ed. São Paulo: Novatec, 2010.

ROGERS, Rick; et al. *Desenvolvimento de aplicações Android*. São Paulo: Novatec, 2009.