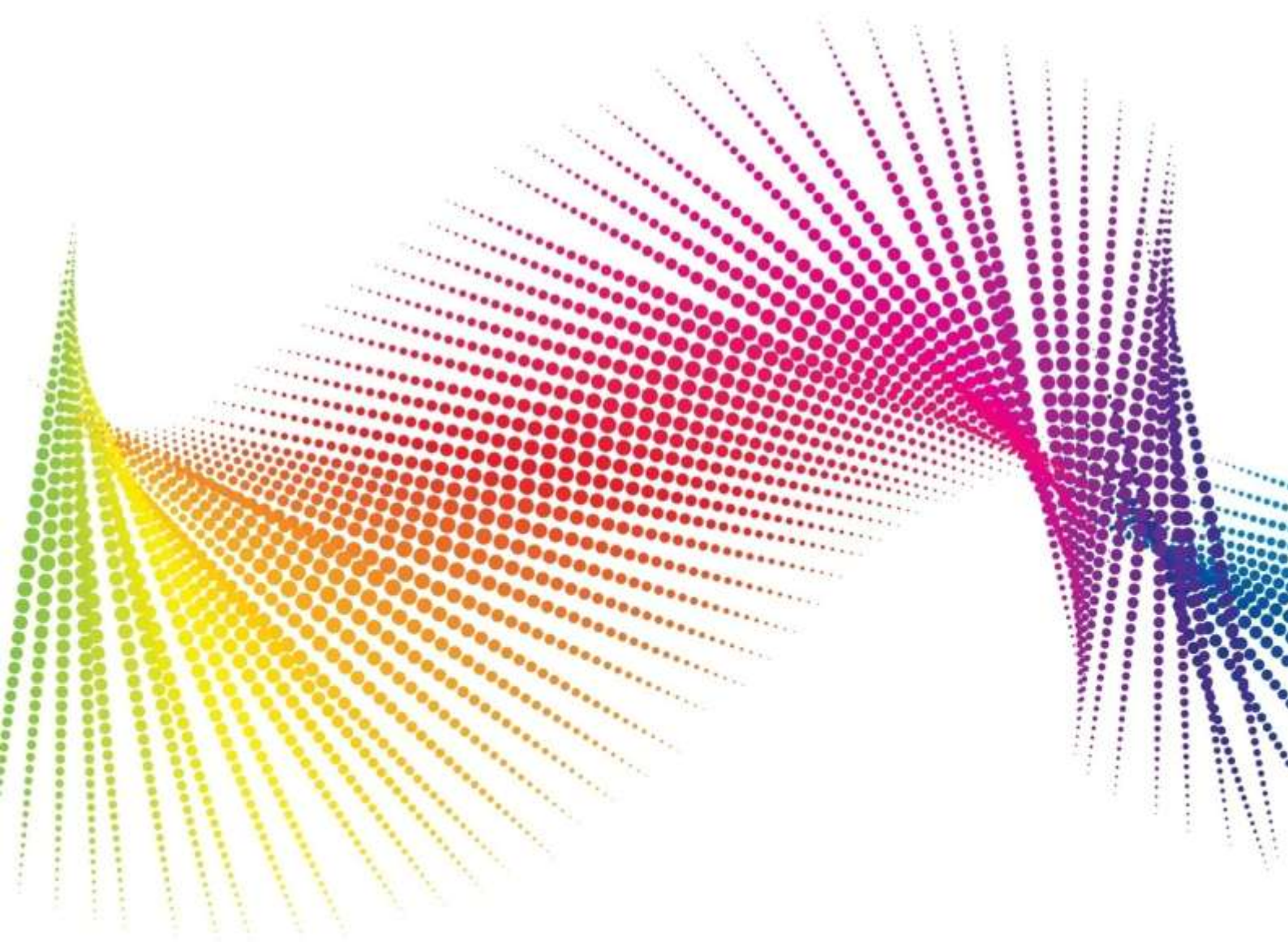


# Computação Móvel

Aula 16



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 16: Provedores de conteúdo**

**Objetivo:** Introduzir a utilização dos dados provenientes de provedores de conteúdo armazenados no sistema operacional android, bem como informar suas localizações e o processo de segurança atrelado a esse acesso.

### **Provedores de conteúdo**

O provedor de contatos é um poderoso e flexível componente Android que gerencia repositório central do dispositivo de dados sobre as pessoas. O provedor de contatos é a fonte de dados que se vê na aplicação do dispositivo, e também pode acessar seus dados em sua própria aplicação e transferência deles entre o dispositivo e serviços online. O provedor acomoda uma ampla gama de fontes de dados e tenta gerenciá-los, tanto quanto possível para cada pessoa. Em razão disso, a API do provedor inclui um amplo conjunto de classes de contrato e interfaces que facilitam tanto a recuperação de dados e modificação.

Quando se quer acessar dados em um provedor de conteúdo, pode-se usar o objeto `ContentResolver` na aplicação para se comunicar com o fornecedor, por exemplo, um cliente. O objeto `ContentResolver` comunica-se com o objeto do provedor, uma instância de uma classe que implementa `ContentProvider`. O objeto do provedor recebe solicitações de dados dos clientes, realiza a ação solicitada e retorna os resultados.

Não é preciso desenvolver um novo provedor, se não pretende compartilhar seus dados com outras aplicações. No entanto, é necessário que se desenvolva um provedor para fornecer sugestões de pesquisa personalizadas em aplicativo desenvolvido por um usuário. Também é necessário um provedor se quisermos copiar e colar dados complexos ou arquivos de uma aplicação para outras.

O Android inclui provedores de conteúdo que gerenciam dados, como áudio, vídeo, imagens e informações de contato pessoal. Pode-se ver alguns deles listados na documentação de referência para o pacote `android.provider`. Com algumas restrições, esses provedores são acessíveis a qualquer aplicativo Android.

Os seguintes tópicos descrevem os provedores de conteúdo em mais detalhes:

## Provedor básico de conteúdo

Os provedores de conteúdo gerenciam o acesso a um conjunto estruturado de dados. Eles encapsulam e proporcionam mecanismos para a definição de segurança dos dados. Os provedores de conteúdo são a interface padrão que conecta dados em um processo com código em execução em outro processo.

## Visão global

Um provedor de conteúdos apresenta dados para aplicações externas, como uma ou mais tabelas que são semelhantes aos das tabelas encontradas em uma base de dados relacional. A linha representa uma instância de algum tipo de dados coletado do provedor, e cada linha da coluna representa uma peça individual de dados coletados formando uma informação completa.

Por exemplo, um provedor de embutidos na plataforma Android compõe o dicionário de dados do usuário, este armazena as grafias de palavras não-padrão que o usuário deseja manter. A tabela 1 ilustra o que os dados podem parecer na tabela desse prestador:

Tabela 1: Exemplo de tabela de dicionário do usuário.

Palavra	App ID	Frequência	Localidade	_ID
MapReduce	user1	100	en_US	1
Precompiler	user14	200	fr_FR	2
Miniaplicativo	user2	225	fr_CA	3
Const	user1	255	pt_BR	4
Int	user5	100	pt_PT	5

Na tabela 1, cada linha representa uma instância de uma palavra que não pode ser encontrada em um dicionário comum. Cada coluna representa alguns dados da palavra, por exemplo, o local em que foi encontrada pela primeira vez.

Os cabeçalhos são os nomes das colunas que são armazenadas no provedor. Para se referir ao local de uma linha, você se refere a sua localidade coluna. Para esse provedor, a coluna `_ID` serve como uma "chave primária" que o provedor gera automaticamente.

## **Como trabalhar provedores de conteúdo**

### **Acessando um provedor**

Um aplicativo acessa os dados de um provedor de conteúdo com um objeto cliente `ContentResolver`. Esse objeto tem métodos que chamam outros com nomes idênticos no do provedor, uma instância de uma das subclasses concretas de `ContentProvider`. Os métodos `ContentResolver` fornecem o básico "CRUD" (criar, recuperar, atualizar e excluir) e funções de armazenamento persistente.

O objeto `ContentResolver`, no processo do aplicativo-cliente, e o objeto `ContentProvider`, na aplicação que possui o provedor, automaticamente lida com a comunicação entre os processos. `ContentProvider` também atua como uma camada de abstração entre o seu repositório de dados e a aparência externa de dados como tabelas.

**Nota:** para acessar um provedor, a sua aplicação geralmente tem de solicitar permissões específicas em seu arquivo de manifesto.

Por exemplo, para obter uma lista das palavras e suas localidades do Provedor de dicionário do usuário, chama-se `ContentResolver.query()`. O método `query()` chama o método `ContentProvider.query()` definido pelo Provedor de dicionário do usuário. As seguintes linhas de código mostram uma chamada `ContentResolver.query()`:

```
// Queries the user dictionary and returns results
mCursor = getContentResolver().query(
    UserDictionary.Words.CONTENT_URI,
    // The content URI of the words table
    mProjection,                // The columns to return for each row
    mSelectionClause            // Selection criteria
    mSelectionArgs,            // Selection criteria
    mSortOrder);                // The sort order for the returned rows
```

## Conteúdo URIs

A URI conteúdo é uma URI que identifica os dados em um provedor. Conteúdo URIs inclui o nome simbólico do provedor inteiro (sua autoridade) e um nome que aponta para uma tabela (um caminho). Quando você chama um método de cliente para acessar uma tabela em um provedor, a URI conteúdo para a tabela é um dos argumentos.

Nas linhas de código anterior, a constante `CONTENT_URI` contém a URI conteúdo do dicionário do usuário "palavras" da tabela. O objeto `ContentResolver` analisa a autoridade da URI, e usa-o para "resolver" o *provedor*, comparando a autoridade de uma tabela do sistema de provedores conhecidos. O `ContentResolver` pode despachar os argumentos de consulta para o provedor correto.

O `ContentProvider` usa a parte do caminho da URI para escolher o conteúdo da tabela de acesso. Um provedor geralmente tem um caminho para cada tabela que expõe.

Nas linhas de código anterior, o URI completo para "palavras" tabela é: `content://user_dictionary/words`, em que a cadeia `user_dictionary` é a autoridade do provedor, e string `words` é o caminho da tabela. O conteúdo da string `://` (o schema) está sempre presente e identifica isso como um URI de conteúdo.

Muitos provedores permitem acessar uma única linha em uma tabela, acrescentando um valor de ID para o final da URI. Por exemplo, para recuperar uma linha cujo `_ID` é 4 de dicionário do usuário, pode-se usar este URI conteúdo:

```
Uri singleUri = ContentUris.withAppendedId(UserDictionary.Words.
CONTENT_URI,4);
```

## Criando um provedor de conteúdo

Um provedor de conteúdo gerencia o acesso a um repositório central de dados. Pode-se implementar um provedor como uma ou mais classes em um aplicativo Android, com elementos do arquivo de manifesto. Uma de suas classes implementa uma subclasse `ContentProvider`, que é a interface entre o fornecedor e outras aplicações. Embora os provedores de conteúdo sejam destinadas a tornar os dados disponíveis para outras aplicações, pode-se, naturalmente, haver atividades em aplicativos que permitem ao usuário consultar e modificar os dados gerenciados por um provedor criado.

Até aqui adquirimos vários conhecimentos sobre a estrutura de manuseio de informação no sistema android, na próxima aula aprenderemos a acessar arquivos de som e vídeo pelo android.

Para termos a certeza de que entendemos a aula, vamos realizar uma atividade sobre provedores de conteúdo no AVA.



### 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.



## REFERÊNCIAS

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

ROGERS, Rick; LOMBARDO, John; MEDNIEKS, Zigurd; MEIKE, Blake. *Desenvolvimento de Aplicações Android*. São Paulo: Novatec, 2009.