**Proveedores del contenido**

**Temas**

1. [Fundamentos de proveedores de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html)
2. [Creación de un Proveedor de Contenido](http://developer.android.com/guide/topics/providers/content-provider-creating.html)
3. [Calendario de Proveedores](http://developer.android.com/guide/topics/providers/calendar-provider.html)

**Muestras relacionadas**

1. [Póngase en contacto con el Administrador](http://developer.android.com/resources/samples/ContactManager/index.html) de aplicaciones
2. ["Cursor (personas)"](http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/view/List2.html)
3. ["Cursor (Móviles)"](http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/view/List7.html)

Los proveedores de contenido controlar el acceso a un conjunto estructurado de datos. , Sino que encierran los datos, y proporcionar mecanismos para la definición de seguridad de los datos. Los proveedores de contenidos son el estándar de interfaz que conecta a los datos en un solo proceso con el código que se ejecuta en otro proceso.

Cuando se desea acceder a los datos de un proveedor de contenido, se utiliza el [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) objeto en la aplicación de [Context](http://developer.android.com/reference/android/content/Context.html) para comunicarse con el proveedor como cliente. El [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) objeto se comunica con el objeto de proveedor, una instancia de una clase que implementa [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) . El objeto de proveedor recibe las solicitudes de datos de clientes, lleva a cabo la acción solicitada, y devuelve los resultados.

No es necesario para desarrollar su propio proveedor si usted no tiene la intención de compartir sus datos con otras aplicaciones. Sin embargo, usted no necesita su propio proveedor para ofrecer sugerencias personalizadas de búsqueda en su propia aplicación. Usted también necesita su propio proveedor si desea copiar y pegar datos complejos o archivos desde su aplicación a otras aplicaciones.

Android se incluye a los proveedores de contenido que manejan datos, tales como audio, vídeo, imágenes e información de contacto personal. Puedes ver algunos de ellos figuran en la documentación de referencia para la [android.provider](http://developer.android.com/reference/android/provider/package-summary.html) paquete. Con algunas restricciones, estos proveedores se puede acceder a cualquier aplicación para Android.

Los temas siguientes describen los proveedores de contenido con más detalle:

[**Fundamentos de proveedores de contenido**](http://developer.android.com/guide/topics/providers/content-provider-basics.html)

¿Cómo acceder a los datos de un proveedor de contenidos, cuando los datos están organizados en tablas.

[**Creación de un Proveedor de Contenido**](http://developer.android.com/guide/topics/providers/content-provider-creating.html)

Cómo crear su propio proveedor de contenido.

[**Calendario de Proveedores**](http://developer.android.com/guide/topics/providers/calendar-provider.html)

Cómo acceder al proveedor de Calendario que forma parte de la plataforma Android.

[↑ Ir arriba](http://developer.android.com/guide/topics/providers/content-providers.html#top)

A excepción de lo señalado, este contenido está disponible bajo [Apache 2.0](http://www.google.com/url?q=http%3A%2F%2Fwww.apache.org%2Flicenses%2FLICENSE-2.0&sa=D&sntz=1&usg=AFQjCNFPaSHdvCi6VD7GwgEkKspvD3d_tw) . Para más detalles y restricciones, consulte la [Licencia de Contenido](http://developer.android.com/license.html) .

Android 4.0 r1 - 01 de mayo 2012 17:37

# Fundamentos de proveedores de contenido

## En este documento

1. [Descripción general](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Basics)
	1. [Acceso a un proveedor](http://developer.android.com/guide/topics/providers/content-provider-basics.html#ClientProvider)
	2. [Contenido URI](http://developer.android.com/guide/topics/providers/content-provider-basics.html#ContentURIs)
2. [Recuperación de datos del proveedor de](http://developer.android.com/guide/topics/providers/content-provider-basics.html#SimpleQuery)
	1. [Solicitud de permiso de acceso de lectura](http://developer.android.com/guide/topics/providers/content-provider-basics.html#RequestPermissions)
	2. [La construcción de la consulta](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Query)
	3. [Viendo resultados de la consulta](http://developer.android.com/guide/topics/providers/content-provider-basics.html#DisplayResults)
	4. [Obtención de datos de los resultados de la consulta](http://developer.android.com/guide/topics/providers/content-provider-basics.html#GettingResults)
3. [Permisos de proveedores de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Permissions)
4. [Actualizar, insertar y eliminar datos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Modifications)
	1. [Insertar datos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Inserting)
	2. [Actualización de datos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Updating)
	3. [Eliminación de datos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Deleting)
5. [Tipos de datos del proveedor](http://developer.android.com/guide/topics/providers/content-provider-basics.html#DataTypes)
6. [Formas alternativas de acceso para proveedores](http://developer.android.com/guide/topics/providers/content-provider-basics.html#AltForms)
	1. [Lote de acceso](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Batch)
	2. [Acceso a los datos a través de intentos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Intents)
7. [Clases de contratos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#ContractClasses)
8. [Referencia del tipo de MIME](http://developer.android.com/guide/topics/providers/content-provider-basics.html#MIMETypeReference)

## Clases de clave

1. [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html)
2. [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html)
3. [Cursor](http://developer.android.com/reference/android/database/Cursor.html)
4. [Uri](http://developer.android.com/reference/android/net/Uri.html)

## Muestras relacionadas

1. [Cursor (Personas)](http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/view/List2.html)
2. [Cursor (Móviles)](http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/view/List7.html)

## Véase también

1. [Creación de un Proveedor de Contenido](http://developer.android.com/guide/topics/providers/content-provider-creating.html)
2. [Calendario de Proveedores](http://developer.android.com/guide/topics/providers/calendar-provider.html)

Un proveedor de contenidos administra el acceso a un repositorio central de datos. Un proveedor es parte de una aplicación para Android, que a menudo ofrece su propia interfaz de usuario para trabajar con los datos. Sin embargo, los proveedores de contenido están destinados principalmente a ser utilizados por otras aplicaciones, los cuales acceden al proveedor mediante un objeto de cliente proveedor. Juntos, los proveedores y los clientes de proveedores de ofrecer una interfaz coherente, estándar para los datos que también se encarga de la comunicación entre procesos y el acceso seguro de datos.

Este tema describe los fundamentos de lo siguiente:

* ¿Cómo funcionan los proveedores de contenido.
* El API se utiliza recuperar datos de un proveedor de contenido.
* El API se utiliza para insertar, actualizar o borrar datos en un proveedor de contenido.
* Otras funciones de la API que facilitan el trabajo con los proveedores.

## Descripción general

Un proveedor de contenidos se presentan los datos a aplicaciones externas como una o más tablas que son similares a las tablas que se encuentran en una base de datos relacional. Una fila representa una instancia de un tipo de datos que recopila el proveedor, y cada fila de la columna representa una pieza individual de los datos recogidos de una instancia.

Por ejemplo, uno de los proveedores integrados en la plataforma Android es el diccionario del usuario, que almacena la ortografía de las palabras no estándar que el usuario desea conservar. La Tabla 1 muestra los datos de lo que podría ser como en la tabla de este proveedor:

**Tabla 1:** Ejemplo de tabla de usuario de diccionario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Palabra** | **aplicación de Identificación** | **Central** | **lugar** | **DI** |
| MapReduce | usuario1 | 100 | es\_ES | 1 |
| precompilador | user14 | 200 | fr\_FR | 2 |
| mini aplicación | user2 | 225 | fr\_CA | 3 |
| const | usuario1 | 255 | pt\_BR | 4 |
| INT. | user5 | 100 | es\_ES | 5 |

En la tabla 1, cada fila representa una instancia de una palabra que no se puede encontrar en un diccionario estándar. Cada columna representa algunos datos para esa palabra, como la configuración regional en el que se encontró por primera vez. Los encabezados de columna son los nombres de columna que se almacenan en el proveedor. Para hacer referencia a la configuración regional de una fila, se hace referencia a su locale en la columna. Para este proveedor, el \_ID columna sirve como una "clave principal" la columna que el proveedor mantiene de forma automática.

**Nota:** El proveedor no está obligado a tener una clave principal, y no es necesario para utilizar \_ID como el nombre de la columna de una clave principal, si uno está presente. Sin embargo, si desea enlazar los datos de un proveedor a un [ListView](http://developer.android.com/reference/android/widget/ListView.html) , uno de los nombres de las columnas tiene que ser \_ID . Este requisito se explica con más detalle en la sección [Viendo resultados de la consulta](http://developer.android.com/guide/topics/providers/content-provider-basics.html#DisplayResults) .

### Acceso a un proveedor

Una aplicación tiene acceso a los datos de un proveedor de contenido con un [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) objeto de cliente. Este objeto tiene métodos que llaman a métodos de idéntico nombre en el objeto de proveedor, una instancia de una de las subclases concretas de [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) . Los [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) métodos proporcionan la base "CRUD" (crear, recuperar, actualizar y eliminar) las funciones de almacenamiento persistente.

El [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) objeto en el proceso de la aplicación cliente y el [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) objeto de la aplicación que posee el proveedor de manejar automáticamente la comunicación entre procesos. [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) también actúa como una capa de abstracción entre el repositorio de datos y la apariencia externa de los datos de las tablas.

**Nota:** Para acceder a un proveedor, su aplicación por lo general tiene que solicitar permisos específicos en su archivo de manifiesto. Esto se describe con más detalle en la sección [permisos de proveedor de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Permissions)

Por ejemplo, para obtener una lista de las palabras y sus locales desde el proveedor de diccionario de usuario, se llama a [ContentResolver.query()](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) . El [query()](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) llama al método [ContentProvider.query()](http://developer.android.com/reference/android/content/ContentProvider.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) método definido por el proveedor de diccionario de usuario. Las siguientes líneas de código muestran una [ContentResolver.query()](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) llame a:

/ / Consulta el diccionario del usuario y devuelve los resultados

mCursor getContentResolver = (). consulta (

 UserDictionary.palabrasCONTENT\_URI, / / ​​El contenido de URI de la tabla de las palabras

 mProjection, / / ​​Las columnas para devolver para cada fila

 mSelectionClause / / Criterios de selección

 mSelectionArgs, / criterios / Selección

 mSortOrder) / / El orden de las filas devueltas

La Tabla 2 muestra cómo los argumentos de [query(Uri,projection,selection,selectionArgs,sortOrder)](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) coincide con una instrucción SELECT de SQL:

**Tabla 2:** Consulta () en comparación con una consulta SQL.

|  |  |  |
| --- | --- | --- |
| **query () el argumento** | **SELECT palabra clave / parámetro** | **Notas** |
| Uri | FROM table\_name | Uri asigna a la tabla en el proveedor de llamada table\_name. |
| projection | col,col,col,... | projection es un conjunto de columnas que se deben incluir para cada fila recuperada.  |
| selection | WHERE col = value | selection se especifican los criterios para la selección de filas. |
| selectionArgs | (No hay un equivalente exacto. Argumentos de selección reemplazar ? marcadores de posición en la cláusula de selección).  |  |
| sortOrder | ORDER BY col,col,... | sortOrder especifica el orden en las filas que aparecen en el regresó [Cursor](http://developer.android.com/reference/android/database/Cursor.html) .  |

### Contenido URI

Un **URI** de **contenido** es un URI que identifica los datos de un proveedor. Contenido URI incluir el nombre simbólico del proveedor de toda (su **autoridad)** y un nombre que apunta a una tabla (una **ruta).** Cuando se llama a un método de cliente para acceder a una tabla en un proveedor, el URI de contenido de la tabla es uno de los argumentos.

En las líneas anteriores de código, la constante [CONTENT\_URI](http://developer.android.com/reference/android/provider/UserDictionary.Words.html#CONTENT_URI) contiene el URI contenido del diccionario del usuario de las "palabras" de mesa. El [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) objeto analiza fuera de la autoridad de la URI, y lo utiliza para "resolver" el proveedor mediante la comparación de la autoridad de una tabla del sistema de proveedores conocidos. El [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) continuación, puede enviar los argumentos de la consulta al proveedor correcto.

El [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) utiliza la ruta de la URI de contenido para elegir la mesa de acceso. Un proveedor por lo general tiene una **ruta** para cada tabla que expone.

En las líneas de código anteriores, el URI completo de la "palabra" tabla es la siguiente:

el contenido :/ / user\_dictionary palabras o

donde el user\_dictionary cadena es la autoridad del proveedor, y words de cadena es la ruta de la tabla. La cadena content:// (el **esquema)** está siempre presente, y lo identifica como un URI contenido.

Muchos proveedores le permiten acceder a una sola fila en una tabla añadiendo un valor de ID hasta el final de la URI. Por ejemplo, para recuperar una fila cuya \_ID es 4 de diccionario de usuario, puede utilizar este URI de contenido:

Uri singleUri = ContentUri.withAppendedId (UserDictionary.palabrasCONTENT\_URI, 4);

A menudo se utilizan valores de identificación cuando se ha recuperado un conjunto de filas y desea actualizar o eliminar uno de ellos.

**Nota:** El [Uri](http://developer.android.com/reference/android/net/Uri.html) , y [Uri.Builder](http://developer.android.com/reference/android/net/Uri.Builder.html) clases contienen métodos de conveniencia para la construcción de bien formados objetos Uri a partir de cadenas. El [ContentUris](http://developer.android.com/reference/android/content/ContentUris.html) contiene métodos de conveniencia para añadir valores de identificación a un URI. El fragmento anterior utiliza [withAppendedId()](http://developer.android.com/reference/android/content/ContentUris.html#withAppendedId%28android.net.Uri,%20long%29) para añadir un id a la URI de contenido UserDictionary.

## Recuperación de datos del proveedor de

En esta sección se describe cómo recuperar datos de un proveedor, utilizando el proveedor de diccionario de usuario como un ejemplo.

En aras de la claridad, los fragmentos de código en esta sección llamada [ContentResolver.query()](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) en el "hilo de la interfaz de usuario" ". En el código actual, sin embargo, usted debe hacer consultas de forma asincrónica en un subproceso independiente. Una forma de hacer esto es utilizar el [CursorLoader](http://developer.android.com/reference/android/content/CursorLoader.html) clase, que se describe con más detalle en la [Cargadores](http://developer.android.com/guide/topics/fundamentals/loaders.html) guía. Además, las líneas de código son sólo fragmentos, sino que no muestran una solicitud completa.

Para recuperar datos de un proveedor, siga estos pasos básicos:

1. Solicitar el permiso de acceso de lectura para el proveedor.
2. Definir el código que se envía una consulta al proveedor.

### Solicitud de permiso de acceso de lectura

Para recuperar datos de un proveedor, la aplicación necesita "permiso de acceso de lectura" para el proveedor. Usted no puede solicitar este permiso en tiempo de ejecución, en su lugar, usted tiene que especificar que se necesita este permiso en su manifiesto, mediante el [<uses-permission>](http://developer.android.com/guide/topics/manifest/uses-permission-element.html) elemento y el nombre del permiso exacto definido por el proveedor. Cuando se especifica este elemento en su manifiesto, en efecto estás "pidiendo" este permiso para su aplicación. Cuando los usuarios instalar la aplicación, de forma implícita a esta solicitud.

Para encontrar el nombre exacto de la autorización de acceso de lectura para el proveedor que está utilizando, así como los nombres de los permisos de acceso a otros medios utilizados por el proveedor, busque en la documentación del proveedor.

El papel de los proveedores de acceso a los permisos de se describe con más detalle en la sección [Permisos de proveedor de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Permissions) .

El Proveedor de diccionario de usuario define el permiso android.permission.READ\_USER\_DICTIONARY en su archivo de manifiesto, por lo que una aplicación que quiere leer el proveedor debe solicitar este permiso.

### La construcción de la consulta

El siguiente paso en la recuperación de los datos de un proveedor es la construcción de una consulta. Este primer fragmento de código define algunas variables para acceder al proveedor de diccionario de usuario:

/ / Una "proyección" se definen las columnas que serán devueltos por cada fila

String [] = mProjection

{

 UserDictionary.palabras\_ID, / Clase / Contrato constante para el nombre de la columna \_ID

 UserDictionary.palabrasPalabra, / clase / Contrato constante para el nombre de la columna la palabra

 UserDictionary.palabrasLocale / / Contrato constante para el nombre de la columna local de clase

};

/ / Define una cadena que contiene la cláusula de elección

Cadena mSelectionClause = null;

/ / Inicializa una matriz que contiene los argumentos de selección

String [] = {mSelectionArgs ""};

El fragmento siguiente se muestra cómo utilizar [ContentResolver.query()](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) , utilizando el proveedor de diccionario de usuario como un ejemplo. Una consulta entre proveedores y clientes es similar a una consulta SQL, y contiene un conjunto de columnas para devolver un conjunto de criterios de selección y un orden de clasificación.

El conjunto de columnas que la consulta debe devolver se denomina una **proyección** (la variable mProjection ).

La expresión que especifica las filas que se recuperan se divide en una cláusula de elección y los argumentos de selección. La cláusula de elección es una combinación de expresiones lógicas y Boolean, nombres de columnas y los valores de la variable ( mSelection ). Si se especifica el parámetro reemplazable ? en lugar de un valor, el método de consulta recupera el valor de la matriz de selección argumentos (las variables mSelectionArgs ).

En el fragmento siguiente, si el usuario no introduce una palabra, la cláusula de elección se establece en null , y la consulta devuelve todas las palabras en el proveedor. Si el usuario introduce una palabra, la cláusula de elección se establece en UserDictionary.Words.Word + " = ?" y el primer elemento de selección matriz argumentos se establece en la palabra que el usuario introduce.

/ \*

 \* Esto define una matriz de cadena de un elemento para contener el argumento de selección.

 \* /

String [] = {mSelectionArgs ""};

/ / Obtiene una palabra de la interfaz de usuario

mSearchString mSearchWord.getText = () toString ().;

/ / Recuerda que insertar el código aquí para comprobar si la entrada no válida o malicioso.

/ / Si la palabra es la cadena vacía, se queda con todo

if (TextUtils.isEmpty (mSearchString)) {

 / / Configuración de la cláusula de elección para anular mostrará todas las palabras

 mSelectionClause = null;

 mSelectionArgs [0] = "";

} else {

 / / Construye una cláusula de elección que coincide con la noticia de que el usuario ha introducido.

 mSelectionClause = "=?";

 / / Cuerda se mueve el usuario de entrada a los argumentos de selección.

 mSelectionArgs [0] = mSearchString;

}

/ / ¿Tiene una consulta en la tabla y devuelve un objeto Cursor

mCursor getContentResolver = (). consulta (

 UserDictionary.palabrasCONTENT\_URI, / / ​​El contenido de URI de la tabla de las palabras

 mProjection, / / ​​Las columnas para devolver para cada fila

 mSelectionClause / / NULL, o la palabra que el usuario introduce

 mSelectionArgs, / / ​​Cualquiera de vacío, o la cadena que el usuario introduce

 mSortOrder) / / El orden de las filas devueltas

/ / Algunos proveedores de devolver un valor nulo si se produce un error, otros una excepción

if (null == mCursor) {

 / \*

 \* Introduzca el código aquí para controlar el error. Asegúrese de no utilizar el cursor! Es posible que desee

 \* Llame android.util.Log.e () para iniciar la sesión de este error.

 \*

 \* /

/ / Si el cursor está vacía, el proveedor no se encontraron coincidencias

} Else if (mCursor.getCount () <1) {

 / \*

 \* Introducir código aquí para notificar al usuario que la búsqueda no tuvo éxito. Esto no es necesariamente

 \* Un error. Es posible que desee ofrecer al usuario la opción de insertar una nueva fila, o volver a escribir el

 \* Término de búsqueda.

 \* /

} else {

 / / Introducir código aquí para hacer algo con los resultados

}

Esta consulta es similar a la instrucción SQL:

SELECT \_ID, la palabra, la frecuencia, entorno local de palabras donde la palabra = ORDEN <userinput> Mediante la palabra ASC;

En esta sentencia de SQL, los nombres de columna se utilizan en lugar de las constantes de la clase de contrato.

#### La protección contra las entradas malintencionadas

Si los datos gestionados por el proveedor de contenido se encuentra en una base de datos SQL, incluyendo los datos externos que no se confía en primas sentencias SQL pueden conducir a la inyección de código SQL.

Considere la posibilidad de esta cláusula de selección:

/ / Construye una cláusula de elección mediante la concatenación de la entrada del usuario al nombre de columna

Cadena mSelectionClause = "var =" + mUserInput;

Si usted hace esto, usted está permitiendo que el usuario de SQL para concatenar malicioso en la instrucción SQL. Por ejemplo, el usuario puede entrar en "nada; DROP TABLE \*;" para mUserInput , lo que daría lugar a la cláusula de elección de var = nothing; DROP TABLE \*; . Dado que la cláusula de elección es tratada como una sentencia SQL, esto podría hacer que el proveedor de borrar todas las tablas en la base de datos SQLite subyacente (menos que el proveedor está configurado para ponerse [de inyección SQL](http://www.google.com/url?q=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSQL_injection&sa=D&sntz=1&usg=AFQjCNFbaumfoyNYVmPtYNoWqEjqbB_H2w) intentos).

Para evitar este problema, utilice una cláusula de elección que utiliza ? como un parámetro reemplazable y una matriz separada de los argumentos de selección. Al hacer esto, la entrada del usuario se vincula directamente a la consulta en lugar de ser interpretado como parte de una sentencia SQL. Debido a que no se trata como SQL Server, la entrada del usuario no puede inyectar SQL malicioso. En lugar de utilizar la concatenación de incluir la entrada del usuario, utilice esta cláusula de elección de:

/ / Construye una cláusula de elección con un parámetro reemplazable

Cadena mSelectionClause = "var =?";

Crear la matriz de argumentos de selección como este:

/ / Define una matriz que contiene los argumentos de selección

String [] = {selectionArgs ""};

Poner un valor en la matriz de selección de argumentos como este:

/ / Establece el argumento de la selección a la entrada del usuario

selectionArgs [0] = mUserInput;

Una cláusula de selección que utiliza ? como un parámetro reemplazable y una matriz de selección matriz argumentos son forma preferida para especificar una selección, incluso si el proveedor no se basa en una base de datos SQL.

### Viendo resultados de la consulta

El [ContentResolver.query()](http://developer.android.com/reference/android/content/ContentResolver.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) método de cliente siempre devuelve un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) que contiene las columnas especificadas por la proyección de la consulta de las filas que coincidan con los criterios de la consulta de selección. Un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) objeto proporciona acceso de lectura al azar de las filas y columnas que contiene. Uso [Cursor](http://developer.android.com/reference/android/database/Cursor.html) métodos, se pueden recorrer las filas de los resultados, determinar el tipo de datos de cada columna, obtener los datos de una columna, y examinar otras propiedades de los resultados. Algunos [Cursor](http://developer.android.com/reference/android/database/Cursor.html) implementaciones se actualizan automáticamente cuando el objeto de datos del proveedor, o cambios en los métodos de activación de un objeto de observador, cuando los [Cursor](http://developer.android.com/reference/android/database/Cursor.html) cambios, o ambas cosas.

**Nota:** Un proveedor puede restringir el acceso a las columnas basadas en la naturaleza del objeto que hace la consulta. Por ejemplo, el proveedor de Contactos restringe el acceso de algunas columnas para que sincronice los adaptadores, por lo que no va a volver a una actividad o servicio.

Si no hay filas coincidan con los criterios de selección, el proveedor devuelve un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) objeto para el que [Cursor.getCount()](http://developer.android.com/reference/android/database/Cursor.html#getCount%28%29) es 0 (un cursor vacío).

Si ocurre un error interno, los resultados de la consulta dependerá del proveedor en particular. Puede optar por regresar null , o que puede producir una [Exception](http://developer.android.com/reference/java/lang/Exception.html) .

Puesto que un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) es una "lista" de las filas, una buena manera de mostrar el contenido de un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) es vincularla a un [ListView](http://developer.android.com/reference/android/widget/ListView.html) a través de un [SimpleCursorAdapter](http://developer.android.com/reference/android/widget/SimpleCursorAdapter.html) .

El siguiente fragmento sigue el código del fragmento anterior. Se crea un [SimpleCursorAdapter](http://developer.android.com/reference/android/widget/SimpleCursorAdapter.html) objeto que contiene el [Cursor](http://developer.android.com/reference/android/database/Cursor.html) recuperado por la consulta, y establece el objeto de ser el adaptador para un [ListView](http://developer.android.com/reference/android/widget/ListView.html) :

/ / Define una lista de columnas que se recuperan del cursor y la carga en una fila de salida

String [] = mWordListColumns

{

 UserDictionary.palabrasPalabra, / clase / Contrato constante que contiene el nombre de columna la palabra

 UserDictionary.palabrasLocale / / clase de contrato constante que contiene el nombre de columna local

};

/ / Define una lista de ID de Vista que va a recibir las columnas del cursor para cada fila

int [] = {mWordListItems R.id.dictWord, R.id.locale};

/ / Crea un nuevo SimpleCursorAdapter

mCursorAdapter = new SimpleCursorAdapter (

 getApplicationContext (), / / ​​el objeto contextual de la aplicación

 R.layout.wordlistrow, / / ​​Un diseño en XML para una fila en el ListView

 mCursor, / / ​​El resultado de la consulta

 mWordListColumns, / / ​​Una matriz de cadenas de nombres de columna en el cursor

 mWordListItems, / / ​​Una matriz de enteros de los identificadores de vista en el diseño de la fila

 0); / / Banderas (por lo general no son necesarios)

/ / Establece el adaptador para el ListView

mWordList.setAdapter (mCursorAdapter);

**Nota:** Para hacer copias de un [ListView](http://developer.android.com/reference/android/widget/ListView.html) con un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) , el cursor debe contener una columna llamada \_ID . Debido a esto, la consulta mostrada anteriormente recupera el \_ID la columna para el "palabras" de mesa, aunque el [ListView](http://developer.android.com/reference/android/widget/ListView.html) no lo muestra. Esta restricción también se explica por qué la mayoría de los proveedores tienen una \_ID columna para cada una de sus mesas.

### Obtención de datos de los resultados de la consulta

En lugar de simplemente mostrar resultados de la consulta, se pueden utilizar para otras tareas. Por ejemplo, puede recuperar la ortografía del diccionario del usuario y luego buscarlos en otros proveedores. Para ello, iterar sobre las filas de la [Cursor](http://developer.android.com/reference/android/database/Cursor.html) :

/ / Determinar el índice de la columna de la columna denominada "palabra"

int index = mCursor.getColumnIndex (UserDictionary.palabrasPalabra

/ \*

 \* Sólo se ejecuta si el cursor es válido. El Proveedor de diccionario de usuario devuelve un valor nulo si

 Hubo un error interno. Otros proveedores pueden producir una excepción en lugar de devolver null.

 \* /

if (mCursor!nulo

 / \*

 \* Mueve a la siguiente fila en el cursor. Antes de que el primer movimiento en el cursor, el

 \* "Puntero de la fila" es -1, y si se intenta recuperar los datos en esa posición obtendrá un

 exception (excepción)

 \* /

 mientras que (mCursor.moveToNext ()) {

 / / Obtiene el valor de la columna.

 newWord = mCursor.getString (índice);

 / / Introducir código aquí para procesar la palabra recuperada.

 ...

 / / Fin del while

 }

} else {

 / / Introducir código aquí para informar de un error si el cursor es nulo o el proveedor lanzó una excepción.

}

[Cursor](http://developer.android.com/reference/android/database/Cursor.html) implementaciones contener varios "get" métodos para recuperar diferentes tipos de datos del objeto. Por ejemplo, el fragmento de código anterior utiliza [getString()](http://developer.android.com/reference/android/database/Cursor.html#getString%28int%29) . También tienen un [getType()](http://developer.android.com/reference/android/database/Cursor.html#getType%28int%29) método que devuelve un valor que indica el tipo de datos de la columna.

## Permisos de proveedores de contenido

La aplicación de un proveedor puede especificar los permisos que otras aplicaciones deben tener para poder acceder a los datos del proveedor. Estos permisos garantizar que el usuario sabe lo que los datos de una aplicación intenta acceder a él. Con base en los requisitos del proveedor, otras aplicaciones solicitar los permisos que necesitan con el fin de obtener acceso al proveedor. Los usuarios finales ver los permisos solicitados cuando se instale la aplicación.

Si la aplicación de un proveedor no especifica todos los permisos, a continuación, otras aplicaciones no tienen acceso a los datos del proveedor. Sin embargo, los componentes de la solicitud del proveedor siempre han completo de lectura y escritura, independientemente de los permisos especificados.

Como se señaló anteriormente, el proveedor de diccionario de usuario requiere la android.permission.READ\_USER\_DICTIONARY permiso para recuperar datos de él. El proveedor tiene la separada android.permission.WRITE\_USER\_DICTIONARY permiso para insertar, actualizar o borrar datos.

Para obtener los permisos necesarios para acceder a un proveedor, una aplicación solicita con un [<uses-permission>](http://developer.android.com/guide/topics/manifest/uses-permission-element.html) elemento en su archivo de manifiesto. Cuando el Gestor de paquetes para Android instala la aplicación, el usuario debe aprobar todos los permisos de las solicitudes de aplicación. Si el usuario aprueba todos ellos, el Administrador de paquetes continúa la instalación, si el usuario no los ratifica, el Administrador de paquetes aborta la instalación.

Los siguientes [<uses-permission>](http://developer.android.com/guide/topics/manifest/uses-permission-element.html) peticiones de los elementos de acceso de lectura al Proveedor de diccionario de usuario:

 <Usos de permiso de android: name = "android.permission.READ\_USER\_DICTIONARY ">

El impacto de los permisos de proveedor de acceso se explica con más detalle en la [Seguridad y permisos](http://developer.android.com/guide/topics/security/security.html) de uso.

## Actualizar, insertar y eliminar datos

De la misma manera que recuperar datos de un proveedor, también puede usar la interacción entre un cliente y el proveedor de [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) para modificar los datos. Se llama a un método de [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) con argumentos que se pasan al método correspondiente de [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) . El cliente proveedor y el proveedor de forma automática manejar las comunicaciones de seguridad y entre procesos.

### Insertar datos

Para insertar datos en un proveedor, llame a la [ContentResolver.insert()](http://developer.android.com/reference/android/content/ContentResolver.html#insert%28android.net.Uri,%20android.content.ContentValues%29) método. Este método inserta una nueva fila en el proveedor y devuelve un URI de contenido para esa fila. Este fragmento de código muestra cómo insertar una nueva palabra en el diccionario de usuario del proveedor:

/ / Define un objeto Uri nuevo que recibe el resultado de la inserción

Uri mNewUri;

...

/ / Define un objeto que contiene los nuevos valores para insertar

MNewValues ​​ContentValues ​​= new ContentValues ​​();

/ \*

 \* Establece los valores de cada columna y se inserta la palabra. Los argumentos de la "venta"

 \* Método son "nombre de la columna" y "valor"

 \* /

mNewValues.put (UserDictionary.palabrasAPP\_ID, "example.user");

mNewValues.put (UserDictionary.palabrasLocale, "es\_ES");

mNewValues.put (UserDictionary.palabrasPALABRA, "insertar");

mNewValues.put (UserDictionary.palabrasFRECUENCIA, "100");

mNewUri getContentResolver = (). insertar (

 UserDictionary.PalabraCONTENT\_URI, / / ​​el contenido del diccionario de usuario URI

 mNewValues ​​/ / Los valores para insertar

);

Los datos para la nueva fila entra en una sola [ContentValues](http://developer.android.com/reference/android/content/ContentValues.html) ​​objeto, que es similar en forma a un cursor de una fila. Las columnas de este objeto no es necesario tener el mismo tipo de datos, y si usted no desea especificar un valor, se puede establecer una columna a null usando [ContentValues.putNull()](http://developer.android.com/reference/android/content/ContentValues.html#putNull%28java.lang.String%29) .

El fragmento no agrega la \_ID columna, porque esta columna se mantiene de forma automática. El proveedor le asigna un valor único de \_ID a cada fila que se agrega. Los proveedores suelen utilizar este valor como clave principal de la tabla.

El URI de contenido devueltos en newUri identifica la fila que acaba de agregar, con el siguiente formato:

el contenido :/ / user\_dictionary, palabras, <id\_value>

El <id\_value> es el contenido de \_ID para la nueva fila. La mayoría de los proveedores pueden detectar este tipo de URI de contenido de forma automática y luego realizar la operación solicitada en esa fila en particular.

Para obtener el valor de \_ID de la devuelve [Uri](http://developer.android.com/reference/android/net/Uri.html) , llame [ContentUris.parseId()](http://developer.android.com/reference/android/content/ContentUris.html#parseId%28android.net.Uri%29) .

### Actualización de datos

Para actualizar una fila, se utiliza un [ContentValues](http://developer.android.com/reference/android/content/ContentValues.html) ​​objeto con los valores actualizados tal como lo hace con una inserción y criterios de selección tal como lo hace con una consulta. El método de cliente que se utiliza es [ContentResolver.update()](http://developer.android.com/reference/android/content/ContentResolver.html#update%28android.net.Uri,%20android.content.ContentValues,%20java.lang.String,%20java.lang.String%5B%5D%29) . Sólo tiene que agregar valores a la [ContentValues](http://developer.android.com/reference/android/content/ContentValues.html) ​​objeto de las columnas va a actualizar. Si desea borrar el contenido de una columna, establezca el valor null .

El siguiente fragmento de código cambia todas las filas cuyo local tiene la lengua "es" una tiene un local de la null . El valor de retorno es el número de filas que fueron actualizadas:

/ / Define un objeto que contiene los valores actualizados

MUpdateValues ​​ContentValues ​​= new ContentValues ​​();

/ / Define los criterios de selección para las filas que desea actualizar

Cadena mSelectionClause = UserDictionary.palabrasLOCALE + "COMO?";

String [] = {mSelectionArgs "% es\_"};

/ / Define una variable para contener el número de filas actualizadas

int mRowsUpdated = 0;

...

/ \*

 \* Establece el valor actualizado y actualizaciones de las palabras seleccionadas.

 \* /

mUpdateValues.putNull (UserDictionary.palabrasConfiguración regional

mRowsUpdated getContentResolver = (). actualización (

 UserDictionary.palabrasCONTENT\_URI, / / ​​el contenido del diccionario de usuario URI

 mUpdateValues ​​/ / las columnas para actualizar

 mSelectionClause / / para seleccionar la columna de

 mSelectionArgs / / el valor a comparar con

);

También debe desinfectar la entrada del usuario cuando se llama a [ContentResolver.update()](http://developer.android.com/reference/android/content/ContentResolver.html#update%28android.net.Uri,%20android.content.ContentValues,%20java.lang.String,%20java.lang.String%5B%5D%29) . Para obtener más información sobre esto, lea la sección de [Protección contra las entradas malintencionadas](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Injection) .

### Eliminación de datos

Eliminación de filas es similar a la recuperación de datos de la fila: se especifica los criterios de selección de las filas que desea eliminar y el método de cliente devuelve el número de registros borrados. Siguiente se eliminan las filas cuyo id de aplicación coincide con fragmento de "usuario". El método devuelve el número de registros borrados.

/ / Define los criterios de selección para las filas que desee eliminar

Cadena mSelectionClause = UserDictionary.palabrasAPP\_ID + "COMO?";

String [] = {mSelectionArgs "usuario"};

/ / Define una variable para contener el número de filas eliminadas

int mRowsDeleted = 0;

...

/ / Elimina las palabras que coincidan con los criterios de selección

mRowsDeleted getContentResolver = (). delete (

 UserDictionary.palabrasCONTENT\_URI, / / ​​el contenido del diccionario de usuario URI

 mSelectionClause / / para seleccionar la columna de

 mSelectionArgs / / el valor a comparar con

);

También debe desinfectar la entrada del usuario cuando se llama a [ContentResolver.delete()](http://developer.android.com/reference/android/content/ContentResolver.html#delete%28android.net.Uri,%20java.lang.String,%20java.lang.String%5B%5D%29) . Para obtener más información sobre esto, lea la sección de [Protección contra las entradas malintencionadas](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Injection) .

## Tipos de datos del proveedor

Los proveedores de contenido pueden ofrecer diferentes tipos de datos. El Proveedor de diccionario de usuario ofrece sólo texto, pero los proveedores también pueden ofrecer los siguientes formatos:

* integer
* entero largo (long)
* de punto flotante
* perspectiva a largo flotante (doble)

Otro tipo de datos que los proveedores utilizan a menudo es objeto binario grande (BLOB) implementa como una matriz de bytes de 64 KB. Usted puede ver los tipos de datos disponibles al mirar los [Cursor](http://developer.android.com/reference/android/database/Cursor.html) de la clase "conseguir" los métodos.

El tipo de datos para cada columna de un proveedor que habitualmente se indica en su documentación. Los tipos de datos para el proveedor de diccionario de usuario se enumeran en la documentación de referencia para la clase de contrato [UserDictionary.Words](http://developer.android.com/reference/android/provider/UserDictionary.Words.html) (clases de contrato se describen en la sección de [clases de contrato](http://developer.android.com/guide/topics/providers/content-provider-basics.html#ContractClasses) ). También se puede determinar el tipo de datos llamando al [Cursor.getType()](http://developer.android.com/reference/android/database/Cursor.html#getType%28int%29) .

Los proveedores también mantener la información de tipo MIME de datos para cada URI de contenido que definen. Puede utilizar la información de tipo MIME para averiguar si su aplicación puede controlar los datos que el proveedor ofrece, o para elegir un tipo de manejo basado en el tipo MIME. Por lo general, necesitan el tipo de MIME cuando se trabaja con un proveedor que contiene estructuras de datos complejas o archivos. Por ejemplo, el [ContactsContract.Data](http://developer.android.com/reference/android/provider/ContactsContract.Data.html) la tabla en el proveedor de Contactos utiliza los tipos MIME para etiquetar el tipo de datos de contacto almacenados en cada fila. Para obtener el tipo MIME correspondiente a un URI de contenido, llame a [ContentResolver.getType()](http://developer.android.com/reference/android/content/ContentResolver.html#getType%28android.net.Uri%29) .

La sección de [referencia de MIME Type](http://developer.android.com/guide/topics/providers/content-provider-basics.html#MIMETypeReference) se describe la sintaxis de los tipos MIME estándar y personalizados.

## Formas alternativas de acceso para proveedores

Tres formas alternativas de proveedor de acceso son importantes en el desarrollo de aplicaciones:

* [Acceso por lotes](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Batch) : puede crear una serie de llamadas de acceso a los métodos de la [ContentProviderOperation](http://developer.android.com/reference/android/content/ContentProviderOperation.html) clase, y luego aplicar con ContentResolver.applyBatch() .
* Consultas asincrónicas: Usted debe hacer las consultas en un hilo separado. Una forma de hacer esto es utilizar un [CursorLoader](http://developer.android.com/reference/android/content/CursorLoader.html) objeto. Los ejemplos en el [Palas](http://developer.android.com/guide/topics/fundamentals/loaders.html) guía muestra cómo hacer esto.
* [Los datos de acceso a través de intentos](http://developer.android.com/guide/topics/providers/content-provider-basics.html#Intents) : Aunque no se puede enviar una intención directa a un proveedor, puede enviar un intento de la solicitud del proveedor, que suele ser la mejor equipados para modificar los datos del proveedor.

Acceso por lotes y la modificación por medio de los intentos se describen en las secciones siguientes.

### Lote de acceso

Acceso por lotes a un proveedor es útil para la inserción de un gran número de filas, o para la inserción de filas en varias tablas en la llamada al método mismo, o en general para llevar a cabo una serie de operaciones a través de los límites del proceso como una transacción (una operación atómica).

Para acceder a un proveedor en el "modo batch", se crea una matriz de [ContentProviderOperation](http://developer.android.com/reference/android/content/ContentProviderOperation.html) objetos y luego enviarlos a un proveedor de contenido con ContentResolver.applyBatch() . Se pasa la autoridad del proveedor de contenido a este método, en lugar de un URI de contenido concreto, que permite a cada [ContentProviderOperation](http://developer.android.com/reference/android/content/ContentProviderOperation.html) objeto de la matriz para trabajar en una tabla diferente. Una llamada a ContentResolver.applyBatch() devuelve una matriz de resultados.

La descripción de la [ContactsContract.RawContacts](http://developer.android.com/reference/android/provider/ContactsContract.RawContacts.html) clase de contrato incluye un fragmento de código que demuestra la inserción por lotes. El [Contact Manager](http://developer.android.com/resources/samples/ContactManager/index.html) aplicación de ejemplo contiene un ejemplo de acceso a lotes en su ContactAdder.java archivo de origen.

## Visualización de datos utilizando una aplicación de ayuda

Si su aplicación no tiene permisos de acceso, es posible que desee utilizar la intención de mostrar los datos en otra aplicación. Por ejemplo, la aplicación Calendario acepta una [ACTION\_VIEW](http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW) intención, que muestra una fecha o evento en particular. Esto le permite visualizar la información del calendario sin tener que crear su propia interfaz de usuario. Para obtener más información acerca de esta característica, consulte el [proveedor de Calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html) guía.

La aplicación a la que enviar el intento no tiene por qué ser la aplicación asociada con el proveedor. Por ejemplo, puede recuperar un contacto del proveedor de Contacto, a continuación, enviar un [ACTION\_VIEW](http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW) la intención que contiene el URI de contenido para la imagen del contacto a un visor de imágenes.

### Acceso a los datos a través de intentos

Intenciones puede proporcionar acceso indirecto a un proveedor de contenido. Se permite al usuario acceder a los datos de un proveedor, incluso si su aplicación no tiene permisos de acceso, ya sea por intención de conseguir un resultado de nuevo desde una aplicación que tenga permisos, o mediante la activación de una aplicación que cuenta con los permisos y permitir al usuario hacer el trabajo en él.

#### Obtener acceso a los permisos temporales

Puede acceder a los datos de un proveedor de contenido, incluso si usted no tiene los permisos de acceso adecuados, mediante el envío de un intento de una aplicación que tiene los permisos y recibir de vuelta un resultado que contenga la intención "URI" permisos. Estos son los permisos para un URI de contenido específico que duran hasta que la actividad que los recibe está terminado. La aplicación que tenga permisos permanentes concede permisos temporales mediante el establecimiento de una bandera en la intención de resultado:

* **El permiso de lectura:** [FLAG\_GRANT\_READ\_URI\_PERMISSION](http://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION)
* **El permiso de escritura:** [FLAG\_GRANT\_WRITE\_URI\_PERMISSION](http://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION)

**Nota:** Estos indicadores no dan en general leer o escribir en el acceso al proveedor cuya autoridad se encuentra en el URI de contenido. El acceso es sólo para el propio URI.

Un proveedor define los permisos de URI para el contenido de los URI en su manifiesto, mediante el [android:grantUriPermission](http://developer.android.com/guide/topics/manifest/provider-element.html#gprmsn) atributo de la [<provider>](http://developer.android.com/guide/topics/providers/guide/topics/manifest/provider-element.html) elemento, así como la [<grant-uri-permission>](http://developer.android.com/guide/topics/providers/guide/topics/manifest/grant-uri-permission-element.html) elemento secundario de la [<provider>](http://developer.android.com/guide/topics/providers/guide/topics/manifest/provider-element.html) elemento. El mecanismo de los permisos de URI se explica con más detalle en la [Seguridad y permisos](http://developer.android.com/guide/topics/security/security.html) de uso, en los permisos "URI" de la sección.

Por ejemplo, puede recuperar datos de un contacto en el proveedor de Contactos, incluso si usted no tiene el [READ\_CONTACTS](http://developer.android.com/reference/android/Manifest.permission.html#READ_CONTACTS) permiso. Es posible que desee hacer esto en una aplicación que envía e-greetings a un contacto en su cumpleaños. En lugar de solicitar [READ\_CONTACTS](http://developer.android.com/reference/android/Manifest.permission.html#READ_CONTACTS) , lo que le da acceso a todos los contactos del usuario y toda su información, usted prefiere dejar que el control de usuario que los contactos son utilizados por su aplicación. Para ello, se utiliza el siguiente proceso:

1. Tu aplicación envía una intención que contiene la acción [ACTION\_PICK](http://developer.android.com/reference/android/content/Intent.html#ACTION_PICK) y los "contactos" tipo MIME [CONTENT\_ITEM\_TYPE](http://developer.android.com/reference/android/provider/ContactsContract.RawContacts.html#CONTENT_ITEM_TYPE) , utilizando el método de [startActivityForResult()](http://developer.android.com/reference/android/app/Activity.html#startActivityForResult%28android.content.Intent,%20int%29) .
2. Debido a este intento coincide con el filtro de intención para la aplicación del Pueblo "selección" de actividad, la actividad se pasa a primer plano.
3. En la actividad de selección, el usuario selecciona un contacto para actualizar. Cuando esto sucede, la actividad de selección de llamadas [setResult(resultcode, intent)](http://developer.android.com/reference/android/app/Activity.html#setResult%28int,%20android.content.Intent%29) para configurar un intento de devolver a su aplicación. La intención contiene el contenido de la URI de contactos del usuario seleccionado, y los "extras" banderas [FLAG\_GRANT\_READ\_URI\_PERMISSION](http://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) . Estos permisos banderas de subvención URI para su aplicación para leer los datos del contacto a la que apunta el URI del contenido. La actividad de selección y luego llama a [finish()](http://developer.android.com/reference/android/app/Activity.html#finish%28%29) para devolver el control a la aplicación.
4. Su actividad vuelve al primer plano, y el sistema llama a su actividad de [onActivityResult()](http://developer.android.com/reference/android/app/Activity.html#onActivityResult%28int,%20int,%20android.content.Intent%29) método. Este método recibe la intención resultado creado por la actividad de selección en la aplicación personas.
5. Con la URI de contenido de la intención de resultado, se puede leer los datos del contacto del proveedor de la Guía, a pesar de que usted no solicitó el permiso permanente de acceso de lectura para el proveedor en su manifiesto. A continuación, puede obtener información del contacto de cumpleaños o su dirección de correo electrónico y luego enviar el mensaje de saludo.

#### Usando otra aplicación

Una forma sencilla de permitir al usuario modificar los datos a los que usted no tiene permisos de acceso es para activar una aplicación que cuenta con los permisos y que el usuario pueda hacer el trabajo allí.

Por ejemplo, la aplicación Calendario acepta una [ACTION\_INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) intención, que le permite activar la interfaz de usuario de la aplicación de inserción. Puede pasar "extras" los datos de esta intención, que la aplicación utiliza para rellenar previamente la interfaz de usuario. Debido a los eventos recurrentes tienen una sintaxis compleja, la forma preferida de la inserción de los eventos en el calendario es proveedor para activar la aplicación Calendario con un [ACTION\_INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) y luego dejar que el usuario inserte el caso de que exista.

## Clases de contratos

Una clase de contrato define las constantes que ayudan a las aplicaciones de trabajo con el contenido de los URI, nombres de columnas, las acciones de la intención, y otras características de un proveedor de contenido. Clases de contrato no se incluyen automáticamente con un proveedor, desarrollador del proveedor tiene que definir y luego ponerlos a disposición de otros desarrolladores. Muchos de los proveedores incluidos en la plataforma Android tienen clases correspondientes del contrato en el paquete [android.provider](http://developer.android.com/reference/android/provider/package-summary.html) .

Por ejemplo, el proveedor de diccionario de usuario tiene una clase de contrato [UserDictionary](http://developer.android.com/reference/android/provider/UserDictionary.html) URI que contenga el contenido y las constantes de nombre de columna. El URI para el contenido de las "palabras" tabla se define en la constante [UserDictionary.Words.CONTENT\_URI](http://developer.android.com/reference/android/provider/UserDictionary.Words.html#CONTENT_URI) . El [UserDictionary.Words](http://developer.android.com/reference/android/provider/UserDictionary.Words.html) clase también contiene constantes de nombre de columna, que se utilizan en los fragmentos de ejemplo en esta guía. Por ejemplo, una proyección de consulta puede ser definido como:

String [] = mProjection

{

 UserDictionary.palabrasDI

 UserDictionary.palabrasPalabra

 UserDictionary.palabrasConfiguración regional

};

Otra clase de contrato es [ContactsContract](http://developer.android.com/reference/android/provider/ContactsContract.html) para el proveedor de Contactos. La documentación de referencia para esta clase incluye fragmentos de código de ejemplo. Uno de sus subclases, [ContactsContract.Intents.Insert](http://developer.android.com/reference/android/provider/ContactsContract.Intents.Insert.html) , es una clase de contrato que contiene las constantes para los fines y los datos de intención.

## Referencia del tipo de MIME

Los proveedores de contenido pueden devolver tipos estándar MIME medios de comunicación, o costumbres, cadenas de tipos MIME, o ambas cosas.

Los tipos MIME que el formato

tipo / subtipo

Por ejemplo, el conocido tipo MIME text/html tiene el text y el tipo de html subtipo. Si el proveedor devuelve este tipo para un URI, esto significa que una consulta mediante la URI devolverá texto que contiene las etiquetas HTML.

Las cadenas personalizadas; tipo MIME, también llamados "específicos de los proveedores" Tipos MIME, tienen el tipo más complejo y los valores de los subtipos. El valor del tipo es siempre

vnd.android.cursor. **dir**

para las filas múltiples, o

vnd.android.cursor. **tema**

de una sola fila.

El subtipo es específico del proveedor. Las Android incorporado en los proveedores por lo general tienen un subtipo simple. Por ejemplo, la aplicación de contactos cuando el crea una fila por un número de teléfono, se establece el tipo MIME siguientes en la fila:

vnd.android.cursor.item/phone\_v2

Tenga en cuenta que el valor de subtipo es simplemente phone\_v2 .

Otros desarrolladores de proveedores pueden crear su propio patrón de los subtipos sobre la base de la autoridad del proveedor y los nombres de tabla. Por ejemplo, considere un proveedor que contiene los horarios de trenes. La autoridad del proveedor es com.example.trains , y contiene la Línea 1 tablas, Línea 2 y Línea 3. En respuesta a la URI de contenido

el contenido :/ / com.example.trains/Line1

para la Línea 1 tabla, el proveedor devuelve el tipo MIME

vnd.android.cursor. **dir** / vnd.example.line1

En respuesta a la URI de contenido

el contenido :/ / com.example.trains/Line2/5

de la fila 5 en el cuadro Línea 2, el proveedor devuelve el tipo MIME

vnd.android.cursor. **artículo** / vnd.example.line2

La mayoría de los proveedores de contenido definir las constantes de la clase de contrato para los tipos MIME que utilizan. El proveedor de Contactos clase de contrato [ContactsContract.RawContacts](http://developer.android.com/reference/android/provider/ContactsContract.RawContacts.html) , por ejemplo, define la constante [CONTENT\_ITEM\_TYPE](http://developer.android.com/reference/android/provider/ContactsContract.RawContacts.html#CONTENT_ITEM_TYPE) para el tipo MIME de una fila de contactos de prima única.

Contenido URI para las filas individuales se describen en la sección [Contenido de URI](http://developer.android.com/guide/topics/providers/content-provider-basics.html#ContentURIs) .

# Creación de un Proveedor de Contenido

## En este documento

1. [El diseño de almacenamiento de datos](http://developer.android.com/guide/topics/providers/content-provider-creating.html#DataStorage)
2. [Diseño de Contenido URI](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContentURI)
3. [Implementación de la clase ContentProvider](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContentProvider)
	1. [Métodos necesarios](http://developer.android.com/guide/topics/providers/content-provider-creating.html#RequiredAccess)
	2. [Aplicación del método query ()](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Query)
	3. [Aplicación del método insert ()](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Insert)
	4. [Implementación del método delete ()](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Delete)
	5. [Implementación del método update ()](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Update)
	6. [Aplicación de la onCreate () método](http://developer.android.com/guide/topics/providers/content-provider-creating.html#OnCreate)
4. [Implementación de contenido MIME Tipos de Proveedores](http://developer.android.com/guide/topics/providers/content-provider-creating.html#MIMETypes)
	1. [Los tipos MIME para las tablas](http://developer.android.com/guide/topics/providers/content-provider-creating.html#TableMIMETypes)
	2. [Los tipos MIME para los archivos](http://developer.android.com/guide/topics/providers/content-provider-creating.html#FileMIMETypes)
5. [La implementación de una clase de contrato](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContractClass)
6. [Permisos de ejecución proveedores de contenido](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Permissions)
7. [El elemento <provider>](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ProviderElement)
8. [Intenciones y acceso a datos](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Intents)

## Clases de clave

1. [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html)
2. [Cursor](http://developer.android.com/reference/android/database/Cursor.html)
3. [Uri](http://developer.android.com/reference/android/net/Uri.html)

## Muestras relacionadas

1. [Notas de aplicación de ejemplo](http://developer.android.com/resources/samples/NotePad/index.html)

## Véase también

1. [Fundamentos de proveedores de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html)
2. [Calendario de Proveedores](http://developer.android.com/guide/topics/providers/calendar-provider.html)

Un proveedor de contenidos administra el acceso a un repositorio central de datos. Implementa un proveedor como una o más clases en una aplicación para Android, junto con los elementos en el archivo de manifiesto. Uno de sus clases implementa una subclase [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) , que es la interfaz entre el proveedor y otras aplicaciones. Aunque los proveedores de contenido están destinados a hacer que los datos disponibles para otras aplicaciones, es posible que por supuesto tienen actividades en la aplicación que permite al usuario consultar y modificar los datos gestionados por el proveedor.

El resto de este tema es una lista básica de los pasos para la construcción de un proveedor de contenidos y una lista de las API de usar.

## Antes de empezar a construir

Antes de comenzar la construcción de un proveedor, haga lo siguiente:

1. **Decidir si usted necesita un proveedor de contenido.** Es necesario construir un proveedor de contenido si se desea proporcionar una o más de las siguientes características:
	* Usted desea ofrecer datos complejos o archivos a otras aplicaciones.
	* Usted quiere que los usuarios puedan copiar los datos complejos a partir de su aplicación en otras aplicaciones.
	* Usted quiere ofrecer sugerencias personalizadas de búsqueda con la aplicación de búsqueda.

Usted no necesita un proveedor de usar una base de datos SQLite si el uso es totalmente dentro de su propia aplicación.

1. Si no lo ha hecho, lea el tema [Fundamentos de proveedores de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html) para obtener más información acerca de los proveedores.

A continuación, siga estos pasos para construir su proveedor:

1. Diseñar el almacenamiento de crudo para sus datos. Un proveedor de contenido ofrece datos de dos maneras:

Datos archivo

Los datos que normalmente entra en los archivos, como fotos, audio o vídeos. Guarde los archivos en el espacio privado de la aplicación. En respuesta a una solicitud de un archivo desde otra aplicación, el proveedor puede ofrecer un identificador para el archivo.

"Estructurado" de datos

Los datos que normalmente entra en una base de datos, matriz o estructura similar. Almacenar los datos en una forma que sea compatible con las tablas de filas y columnas. Una fila representa una entidad, como por ejemplo una persona o un elemento en el inventario. Una columna representa algunos datos para la entidad, como nombre de una persona o el precio de un artículo. Una forma común para almacenar este tipo de datos se encuentra en una base de datos SQLite, pero puede utilizar cualquier tipo de almacenamiento persistente. Para obtener más información acerca de los tipos de almacenamiento disponibles en el sistema Android, consulte la sección de [diseño de almacenamiento de datos](http://developer.android.com/guide/topics/providers/content-provider-creating.html#DataStorage) .

1. Definir una aplicación concreta de la [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) clase y sus métodos necesarios. Esta clase es la interfaz entre los datos y el resto del sistema Android. Para obtener más información acerca de esta clase, vea la sección [Implementación de la clase ContentProvider](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContentProvider) .
2. Definir cadena de la autoridad del proveedor, su contenido URI, y los nombres de las columnas. Si desea la solicitud del proveedor para manejar las intenciones, también definen las acciones intención, datos extras, y las banderas. También define los permisos que se requieren para las aplicaciones que quieren acceder a sus datos. Debe tener en cuenta la definición de todos estos valores como constantes en una clase de contrato por separado, más tarde, puede exponer esta clase para otros desarrolladores. Para obtener más información sobre el contenido de los URI, consulte la sección [Diseño de Contenido URI](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContentURI) . Para obtener más información acerca de las intenciones, vea la sección [Intenciones y acceso a datos](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Intents) .
3. Añadir otras piezas opcionales, tales como datos de la muestra o una implementación de [AbstractThreadedSyncAdapter](http://developer.android.com/reference/android/content/AbstractThreadedSyncAdapter.html) que pueden sincronizar datos entre el proveedor y los datos basados ​​en la nube.

## El diseño de almacenamiento de datos

Un proveedor de contenido es la interfaz a los datos guardados en un formato estructurado. Antes de crear la interfaz, debe decidir cómo almacenar los datos. Puede almacenar los datos en cualquier forma que te gusta, y luego diseñar la interfaz para leer y escribir los datos según sea necesario.

Estas son algunas de las tecnologías de almacenamiento de datos que están disponibles en Android:

* El sistema Android incluye una base de datos SQLite de la API que los proveedores propios de Android utiliza para almacenar la tabla de datos orientados. El [SQLiteOpenHelper](http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html) clase le ayuda a crear bases de datos, y la [SQLiteDatabase](http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) clase es la clase base para acceder a bases de datos.

Recuerde que usted no tiene que utilizar una base de datos para poner en práctica su repositorio. Un proveedor externo aparece como un conjunto de tablas, de forma similar a una base de datos relacional, pero esto no es un requisito para la implementación interna del proveedor.

* Para almacenar los datos de archivo, Android tiene una gran variedad de archivos orientada a las API. Para obtener más información sobre el almacenamiento de archivos, leer el tema de [almacenamiento de datos](http://developer.android.com/guide/topics/data/data-storage.html) . Si usted está diseñando un proveedor que ofrece los medios de comunicación relacionados con datos tales como música o vídeos, usted puede tener un proveedor que combina datos de la tabla y archivos.
* Para trabajar con la red a base de datos, clases de uso en [java.net](http://developer.android.com/reference/java/net/package-summary.html) y [android.net](http://developer.android.com/reference/android/net/package-summary.html) . También puede sincronizar basados ​​en la red de datos a un almacén de datos local como una base de datos, y luego ofrecer los datos como las tablas o archivos. El [ejemplo del adaptador de sincronización de](http://developer.android.com/resources/samples/SampleSyncAdapter/index.html) aplicación de ejemplo muestra este tipo de sincronización.

### Datos consideraciones de diseño

Estos son algunos consejos para el diseño de la estructura de su proveedor de datos:

* Datos de la tabla siempre debe tener una "clave principal" la columna que el proveedor mantiene como un valor numérico único para cada fila. Puede utilizar este valor para vincular la fila de las filas relacionadas en otras tablas (utilizando como una "clave externa"). Aunque puede utilizar cualquier nombre para esta columna, utilizando [BaseColumns.\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) es la mejor opción, debido a la vinculación de los resultados de una consulta a un proveedor de [ListView](http://developer.android.com/reference/android/widget/ListView.html) requiere una de las columnas recuperadas de tener el nombre \_ID .
* Si desea proporcionar imágenes de mapa de bits u otras piezas de gran tamaño orientadas a archivos de datos, almacenar los datos en un archivo y, después, de forma indirecta en lugar de almacenarla directamente en una tabla. Si usted hace esto, es necesario indicar a los usuarios de su proveedor que necesita usar un [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) método de archivo para acceder a los datos.
* Utilice el objeto binario grande (BLOB) el tipo de datos para almacenar datos que varían en tamaño o tiene una estructura variable. Por ejemplo, puede usar una columna BLOB para almacenar un [búfer de protocolo](http://code.google.com/p/protobuf) o de [la estructura JSON](http://www.google.com/url?q=http%3A%2F%2Fwww.json.org&sa=D&sntz=1&usg=AFQjCNEMn-zN9nle4OeHTGxZv-Wj-RFUgA) .

También puede utilizar un BLOB para implementar una tabla de esquema independiente. En este tipo de tabla, se define una columna de clave principal, una columna de tipo MIME, y una o más columnas BLOB como genéricos. El significado de los datos de las columnas BLOB es indicado por el valor en la columna de tipo MIME. Esto le permite almacenar diferentes tipos de fila en la misma mesa. El proveedor de Contactos de "datos" tabla de [ContactsContract.Data](http://developer.android.com/reference/android/provider/ContactsContract.Data.html) es un ejemplo de una tabla de esquema-independiente.

## Diseño de Contenido URI

Un **URI** de **contenido** es un URI que identifica los datos de un proveedor. Contenido URI incluir el nombre simbólico del proveedor de toda (su **autoridad)** y un nombre que apunta a una tabla o un archivo (una **ruta).** La parte opcional de Identificación apunta a una fila individual de una tabla. Cada método de acceso de datos de [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) tiene un URI de contenido como un argumento, lo que le permite determinar la tabla, fila o de archivos para acceder.

Los fundamentos de la URI de contenido se describe en el tema [Conceptos básicos de proveedor de contenido](http://developer.android.com/guide/topics/providers/content-provider-basics.html) .

### El diseño de una autoridad

Un proveedor por lo general tiene una sola autoridad, que le sirve de Android nombre interno. Para evitar conflictos con otros proveedores, se debe utilizar la propiedad de dominio de Internet (al revés) como base de su autoridad profesional. Debido a esta recomendación también es válido para los nombres de paquetes para Android, se puede definir la autoridad del proveedor como una extensión del nombre del paquete que contiene el proveedor. Por ejemplo, si el nombre de su paquete de Android es com.example.<appname> , usted debe darle a su proveedor de la autoridad com.example.<appname>.provider .

### Diseñar una estructura de ruta

Los desarrolladores crear contenido URI de la autoridad añadiendo caminos que apuntan a las tablas individuales. Por ejemplo, si usted tiene dos tablas tabla1 y tabla2, se combina la autoridad del ejemplo anterior para obtener el contenido URI com.example.<appname>.provider/table1 y com.example.<appname>.provider/table2 . Las rutas no están limitados a un solo segmento, y no tiene que ser una tabla para cada nivel de la ruta.

### Gestión de contenidos URI ID

Por convención, los proveedores ofrecen acceso a una sola fila en una tabla mediante la aceptación de un URI de contenido con un valor de ID de la fila al final del URI. También, por convención, los proveedores de acuerdo con el valor de ID de la tabla \_ID columna, y llevar a cabo el acceso solicitado en contra de la fila que coincide.

Esta convención facilita un modelo de diseño común para las aplicaciones que acceden a un proveedor. La aplicación hace una consulta en el proveedor y muestra el resultado [Cursor](http://developer.android.com/reference/android/database/Cursor.html) en un [ListView](http://developer.android.com/reference/android/widget/ListView.html) usando un [CursorAdapter](http://developer.android.com/reference/android/widget/CursorAdapter.html) . La definición de [CursorAdapter](http://developer.android.com/reference/android/widget/CursorAdapter.html) requiere una de las columnas en el [Cursor](http://developer.android.com/reference/android/database/Cursor.html) que se \_ID

Luego, el usuario elige una de las filas que se muestran a partir de la interfaz de usuario con el fin de ver o modificar los datos. La aplicación obtiene la fila correspondiente en el [Cursor](http://developer.android.com/reference/android/database/Cursor.html) copia de seguridad del [ListView](http://developer.android.com/reference/android/widget/ListView.html) , obtiene el \_ID valor de esta fila, lo agrega a la URI de contenido, y envía la solicitud de acceso al proveedor. Luego, el proveedor puede hacer la consulta o modificación en contra de la línea exacta que el usuario eligió.

### Los patrones de contenido URI

Para ayudarle a elegir qué acción tomar para un URI de contenido entrante, la API de proveedor incluye la clase de conveniencia [UriMatcher](http://developer.android.com/reference/android/content/UriMatcher.html) , que asigna URI de contenido "patrones" a valores enteros. Puede utilizar los valores enteros en un switch la declaración que elige la acción deseada por el contenido o URI URI que coincidan con un patrón particular.

Un patrón URI contenido coincide con el contenido de los URI con caracteres comodín:

* **\* :** Coincide con cualquier cadena de caracteres válidos de cualquier longitud.
* **# :** coincide con una cadena de caracteres numéricos de cualquier longitud.

Como un ejemplo de diseño y codificación de manejo de contenidos URI, considere a un proveedor con la autoridad com.example.app.provider que reconoce el contenido siguiente URI que apunta a las tablas:

* content://com.example.app.provider/table1 : Una tabla llamada table1 .
* content://com.example.app.provider/table2/dataset1 : una tabla llamada dataset1 .
* content://com.example.app.provider/table2/dataset2 : una tabla llamada dataset2 .
* content://com.example.app.provider/table3 : Una tabla llamada table3 .

El proveedor también reconoce estos contenidos URI si tienen un identificador de fila se acompañará a éste, como por ejemplo content://com.example.app.provider/table3/1 de la fila identificado por 1 en la table3 .

Los patrones de contenido siguientes URI sería posible:

content://com.example.app.provider/\*

Coincide con cualquier URI contenidos en el proveedor.

content://com.example.app.provider/table2/\* :

Coincide con un URI de contenido para las tablas dataset1 y dataset2 , pero no coincide con el contenido de los URI de table1 y table3 .

content://com.example.app.provider/table3/# : Coincide con un URI de contenido para una sola fila en la table3 , como content://com.example.app.provider/table3/6 de la fila identificado por 6 .

El siguiente fragmento de código muestra cómo los métodos de [UriMatcher](http://developer.android.com/reference/android/content/UriMatcher.html) trabajo. Este código se encarga de URI para toda una tabla diferente de la URI de una sola fila, utilizando el patrón de URI de contenido content://<authority>/<path> para las tablas, y content://<authority>/<path>/<id> para las filas individuales.

El método [addURI()](http://developer.android.com/reference/android/content/UriMatcher.html#addURI%28java.lang.String,%20java.lang.String,%20int%29) asigna una autoridad y la ruta a un valor entero. El android.content método.# UriMatcher partido (Uri) partido ()} devuelve el valor entero de una URI. Un switch la declaración elige entre la consulta de la tabla entera, y la consulta de un único registro:

public class extends ExampleProvider ContentProvider {

...

 / / Crea un objeto UriMatcher.

 sUriMatcher privada estática UriMatcher final;

...

 / \*

 \* Las llamadas a addURI () van aquí, para todos los modelos de contenido URI que el proveedor

 \* Debe reconocer. Para este fragmento, sólo las llamadas de la tabla 3 se muestran.

 \* /

...

 / \*

 \* Establece el valor entero de varias filas en la tabla 3 a 1. Tenga en cuenta que no se utiliza de comodín

 \* En el camino

 \* /

 sUriMatcher.addURI ("com.example.app.provider", "tabla 3", 1);

 / \*

 \* Establece el código de una sola fila a 2. En este caso, el "#" es comodín

 \* Utilizado. "Contenido :/ / com.example.app.provider/table3/3" los partidos, pero

 \* "El contenido :/ / com.example.app.provider/table3 no lo hace.

 \* /

 sUriMatcher.addURI ("com.example.app.provider", "Cuadro 3 / #", 2);

...

 / / Implementa ContentProvider.query ()

 consulta pública del cursor (

 Uri Uri,

 String [proyección],

 Cadena de selección,

 String [] selectionArgs,

 Cadena sortOrder) {

...

 / \*

 \* Elija la tabla de consulta y un orden basado en el código devuelto por la entrada

 \* URI. Aquí, también, solamente los estados para la tabla 3 se muestran.

 \* /

 switch (sUriMatcher.match (URI)) {

 / / Si el URI de entrada era para toda la tabla 3

 Caso 1

 if (TextUtils.isEmpty (sortOrder)) sortOrder = "\_ID ASC";

 [ break ; ]

 / / Si el URI de entrada era de una sola fila

 Caso 2:

 / \*

 \* Debido a que este URI era para una sola fila, la parte del valor es \_ID

 pres Obtener el último segmento del path de la URI, lo que es el valor \_ID.

 \* A continuación, añadir el valor de la cláusula WHERE de la consulta

 \* /

 = selección selección + "\_id =" uri.getLastPathSegment ();

 [ break ; ]

 default:

 ...

 / / Si la URI no es reconocida, usted debe hacer algunas manejo de errores aquí.

 }

 / / Llamar al código para hacer realidad la consulta

 }

Otra clase, [ContentUris](http://developer.android.com/reference/android/content/ContentUris.html) , proporciona los métodos convenientes para trabajar con la id parte del contenido de los URI. La clase [Uri](http://developer.android.com/reference/android/net/Uri.html) y [Uri.Builder](http://developer.android.com/reference/android/net/Uri.Builder.html) incluyen métodos de conveniencia para analizar existentes [Uri](http://developer.android.com/reference/android/net/Uri.html) objetos y la creación de otros nuevos.

## Implementación de la clase ContentProvider

El [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) ejemplo, gestiona el acceso a un conjunto estructurado de datos de tramitación de las solicitudes de otras aplicaciones. Todas las formas de acceso momento se llama [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) , que a su vez llama a un método concreto de [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) para obtener acceso.

### Los métodos necesarios

La clase abstracta [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) define seis métodos abstractos que debe implementar como parte de su propia subclase concreta. Todos estos métodos, excepto [onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) es llamado por una aplicación cliente que está intentando acceder a su proveedor de contenido:

[query()](http://developer.android.com/reference/android/content/ContentProvider.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29)

Recuperar datos de su proveedor. Utilice los argumentos para seleccionar la tabla a la consulta, las filas y columnas para regresar, y el orden de los resultados. Devolver los datos como un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) objeto.

[insert()](http://developer.android.com/reference/android/content/ContentProvider.html#insert%28android.net.Uri,%20android.content.ContentValues%29)

Insertar una nueva fila en su proveedor. Utilice los argumentos para seleccionar la tabla de destino y para obtener los valores de la columna de usar. Devuelve una URI contenido de la fila recién insertada.

[update()](http://developer.android.com/reference/android/content/ContentProvider.html#update%28android.net.Uri,%20android.content.ContentValues,%20java.lang.String,%20java.lang.String%5B%5D%29)

Actualización de filas existentes en su proveedor. Utilice los argumentos para seleccionar la tabla y las filas para actualizar y para obtener los valores de las columnas actualizadas. Devuelve el número de filas actualizadas.

[delete()](http://developer.android.com/reference/android/content/ContentProvider.html#delete%28android.net.Uri,%20java.lang.String,%20java.lang.String%5B%5D%29)

Eliminar las filas de su proveedor. Utilice los argumentos para seleccionar la tabla y las filas que desea eliminar. Devuelve el número de registros borrados.

[getType()](http://developer.android.com/reference/android/content/ContentProvider.html#getType%28android.net.Uri%29)

Devuelve el tipo MIME correspondiente a un URI contenido. Este método se describe con más detalle en la sección [de aplicación de contenido Tipos MIME proveedor](http://developer.android.com/guide/topics/providers/content-provider-creating.html#MIMETypes) .

[onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29)

Inicializar su proveedor. El sistema Android llama a este método inmediatamente después de que crea con su proveedor. Tenga en cuenta que su proveedor no se crea hasta un [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) objeto intenta acceder a él.

Tenga en cuenta que estos métodos tienen la misma firma que las de idéntico nombre [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) métodos.

La implementación de estos métodos deben tener en cuenta lo siguiente:

* Todos estos métodos, excepto [onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) puede ser llamado por varios subprocesos al mismo tiempo, por lo que debe ser thread-safe. Para aprender más acerca de varios subprocesos, vea el tema [de Procesos y subprocesos](http://developer.android.com/guide/topics/fundamentals/processes-and-threads.html) .
* Evite hacer operaciones largas en [onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) . Aplazar las tareas de inicialización hasta que realmente se necesitan. La sección de [implementación del método OnCreate ()](http://developer.android.com/guide/topics/providers/content-provider-creating.html#OnCreate) examina esta cuestión con más detalle.
* A pesar de que deben poner en práctica estos métodos, el código no tiene que hacer otra cosa que devolver el tipo de datos esperado. Por ejemplo, es posible que desee evitar que otras aplicaciones de la inserción de datos en algunas mesas. Para ello, se puede ignorar la llamada a [insert()](http://developer.android.com/reference/android/content/ContentProvider.html#insert%28android.net.Uri,%20android.content.ContentValues%29) y devuelve 0.

### Aplicación del método query ()

El [ContentProvider.query()](http://developer.android.com/reference/android/content/ContentProvider.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) el método debe devolver un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) objeto, o si no, lanzar una [Exception](http://developer.android.com/reference/java/lang/Exception.html) . Si está utilizando una base de datos SQLite como el almacenamiento de datos, sólo tiene que devolver el [Cursor](http://developer.android.com/reference/android/database/Cursor.html) devuelto por uno de los query() los métodos de la [SQLiteDatabase](http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) clase. Si la consulta no coincide con ninguna fila, usted debe devolver un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) instancia cuyo [getCount()](http://developer.android.com/reference/android/database/Cursor.html#getCount%28%29) método devuelve 0. Usted debe regresar null sólo si un error interno durante el proceso de consulta.

Si usted no está utilizando una base de datos SQLite como el almacenamiento de datos, utilice una de las subclases concretas de [Cursor](http://developer.android.com/reference/android/database/Cursor.html) . Por ejemplo, el [MatrixCursor](http://developer.android.com/reference/android/database/MatrixCursor.html) clase implementa un cursor en la que cada fila es un array de [Object](http://developer.android.com/reference/java/lang/Object.html) . Con esta clase, se utiliza [addRow()](http://developer.android.com/reference/android/database/MatrixCursor.html#addRow%28java.lang.Object%5B%5D%29) para añadir una nueva fila.

Recuerde que el sistema Android debe ser capaz de comunicar la [Exception](http://developer.android.com/reference/java/lang/Exception.html) en los límites del proceso. Android puede hacer esto por las siguientes excepciones que pueden ser útiles en el manejo de errores de la consulta:

* [IllegalArgumentException](http://developer.android.com/reference/java/lang/IllegalArgumentException.html) (Usted puede elegir para lanzar esta opción si su proveedor recibe un URI contenido no es válido)
* [NullPointerException](http://developer.android.com/reference/java/lang/NullPointerException.html)

### Aplicación del método insert ()

El [insert()](http://developer.android.com/reference/android/content/ContentProvider.html#insert%28android.net.Uri,%20android.content.ContentValues%29) método agrega una nueva fila a la tabla correspondiente, utilizando los valores de la [ContentValues](http://developer.android.com/reference/android/content/ContentValues.html) ​​argumento. Si un nombre de columna no está en la [ContentValues](http://developer.android.com/reference/android/content/ContentValues.html) ​​argumento, es posible que desee proporcionar un valor predeterminado para que sea en el código del proveedor o en su esquema de base de datos.

Este método debe devolver el URI de contenido para la nueva fila. Para construir esto, añadir la nueva fila de \_ID (o la clave principal de otro tipo) el valor de URI contenido de la tabla, utilizando [withAppendedId()](http://developer.android.com/reference/android/content/ContentUris.html#withAppendedId%28android.net.Uri,%20long%29) .

### Implementación del método delete ()

El [delete()](http://developer.android.com/reference/android/content/ContentProvider.html#delete%28android.net.Uri,%20java.lang.String,%20java.lang.String%5B%5D%29) método no tiene por qué eliminar físicamente a las filas de su almacenamiento de datos. Si usted está usando un adaptador de sincronización con su proveedor, usted debe considerar marcar una fila eliminada con "borrar" la bandera en lugar de eliminar por completo la fila. El adaptador de sincronización puede comprobar si los registros borrados y eliminarlos del servidor antes de eliminarlos del proveedor.

### Implementación del método update ()

La [update()](http://developer.android.com/reference/android/content/ContentProvider.html#update%28android.net.Uri,%20android.content.ContentValues,%20java.lang.String,%20java.lang.String%5B%5D%29) método tiene la misma [ContentValues](http://developer.android.com/reference/android/content/ContentValues.html) ​​argumento utilizado por [insert()](http://developer.android.com/reference/android/content/ContentProvider.html#insert%28android.net.Uri,%20android.content.ContentValues%29) , y la misma selection y selectionArgs argumentos utilizados por [delete()](http://developer.android.com/reference/android/content/ContentProvider.html#delete%28android.net.Uri,%20java.lang.String,%20java.lang.String%5B%5D%29) y [ContentProvider.query()](http://developer.android.com/reference/android/content/ContentProvider.html#query%28android.net.Uri,%20java.lang.String%5B%5D,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%29) . Esto le puede permitir la reutilización de código entre estos métodos.

### Aplicación de la onCreate () método

El sistema Android llama [onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) cuando se inicia el proveedor. Usted debe realizar sólo las tareas de funcionamiento rápido de inicialización en este método, y aplazar la creación de bases de datos y carga de datos hasta que el proveedor de hecho recibe una solicitud de los datos. Si usted hace las tareas largas en [onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) , que se ralentizará el inicio de su proveedor. A su vez, este se ralentizará la respuesta del proveedor a otras aplicaciones.

Por ejemplo, si usted está utilizando una base de datos SQLite puede crear una nueva [SQLiteOpenHelper](http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html) objeto en [ContentProvider.onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) , y luego crear las tablas de SQL de la primera vez que abra la base de datos. Para facilitar esto, la primera vez que llame a [getWritableDatabase()](http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#getWritableDatabase%28%29) , se llama automáticamente al [SQLiteOpenHelper.onCreate()](http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onCreate%28android.database.sqlite.SQLiteDatabase%29) método.

Los siguientes dos fragmentos de demostrar la interacción entre [ContentProvider.onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) y [SQLiteOpenHelper.onCreate()](http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onCreate%28android.database.sqlite.SQLiteDatabase%29) . El primer fragmento de código es la aplicación de [ContentProvider.onCreate()](http://developer.android.com/reference/android/content/ContentProvider.html#onCreate%28%29) :

pública ExampleProvider clase extiende ContentProvider

 / \*

 \* Define un identificador para el objeto de ayuda de base de datos. La clase MainDatabaseHelper se define

 \* En el siguiente fragmento de código.

 \* /

 privada mOpenHelper MainDatabaseHelper;

 / / Define el nombre de base de datos

 privada dbname static final String = "mydb";

 / / Declara el objeto de base de datos

 db SQLiteDatabase privada;

 pública onCreate boolean () {

 / \*

 \* Crea un objeto nuevo ayudante. Este método siempre devuelve rápidamente.

 \* Tenga en cuenta que la propia base de datos no se crea o se abre

 \* Hasta que se llama SQLiteOpenHelper.getWritableDatabase

 \* /

 mOpenHelper = new SQLiteOpenHelper (

 getContext (), / / ​​el contexto de aplicación

 Dbname, / / ​​el nombre de la base de datos)

 NULL, / / ​​utiliza el cursor predeterminado SQLite

 1 / / el número de versión

 );

 return true;

 }

 ...

 / / Implementa el método del proveedor de inserción

 El cursor de inserción pública (Uri Uri, los valores ContentValues) {

 / / Introducir código aquí para determinar qué tabla para abrir, manipular de comprobación de errores, y así sucesivamente

 ...

 / \*

 \* Obtiene una base de datos de escritura. Esto dará lugar a su creación si no existe ya.

 \*

 \* /

 db = mOpenHelper.getWritableDatabase ();

 }

}

El fragmento siguiente es la aplicación de la [SQLiteOpenHelper.onCreate()](http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onCreate%28android.database.sqlite.SQLiteDatabase%29) , incluyendo una clase de ayuda:

...

/ / Una cadena que define la instrucción SQL para la creación de una mesa

private String static final SQL\_CREATE\_MAIN = "CREATE TABLE" +

 "Principal" + / / El cuadro de nombre

 "(" + / / Las columnas de la tabla

 "\_id INTEGER PRIMARY KEY," +

 "LA PALABRA DE TEXTO"

 "FRECUENCIA INTEGER" +

 "LOCALE TEXTO)";

...

/ \*\*

 \* Ayudante de clase que en realidad crea y gestiona repositorio subyacente del proveedor de datos.

 \* /

protección estática final class extends MainDatabaseHelper SQLiteOpenHelper {

 / \*

 \* Crea una instancia de un ayudante para SQLite abierta repositorio de datos del proveedor

 \* No hagas la creación de bases de datos y mejorar aquí.

 \* /

 MainDatabaseHelper (context) {

 super (contexto, dbname, null, 1);

 }

 / \*

 \* Crea el repositorio de datos. Esto se llama cuando el proveedor intenta abrir el

 \* Repositorio y SQLite informa que no existe.

 \* /

 onCreate public void (SQLiteDatabase db) {

 / / Crea la tabla principal

 db.execSQL (SQL\_CREATE\_MAIN);

 }

}

## Ejecución ContentProvider tipos MIME

El [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) clase tiene dos métodos para devolver los tipos MIME:

[getType()](http://developer.android.com/reference/android/content/ContentProvider.html#getType%28android.net.Uri%29)

Uno de los métodos necesarios que se deben implementar para cualquier proveedor.

[getStreamTypes()](http://developer.android.com/reference/android/content/ContentProvider.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29)

Un método que se espera que usted poner en práctica si su proveedor ofrece archivos.

### Los tipos MIME para las tablas

El [getType()](http://developer.android.com/reference/android/content/ContentProvider.html#getType%28android.net.Uri%29) método devuelve una [String](http://developer.android.com/reference/java/lang/String.html) en formato MIME que describe el tipo de datos devueltos por el argumento URI contenido. El [Uri](http://developer.android.com/reference/android/net/Uri.html) argumento puede ser un patrón en lugar de un URI específico, en este caso, usted debe devolver el tipo de datos relacionados con el contenido de los URI que coincidan con el patrón.

Para los tipos comunes de datos, como en forma de texto, HTML o JPEG, [getType()](http://developer.android.com/reference/android/content/ContentProvider.html#getType%28android.net.Uri%29) debería devolver el tipo estándar MIME para esos datos. Una lista completa de estos tipos estándar está disponible en la [IANA MIME Tipos de medios](http://www.google.com/url?q=http%3A%2F%2Fwww.iana.org%2Fassignments%2Fmedia-types%2Findex.htm&sa=D&sntz=1&usg=AFQjCNGQVVZ7BlRbk0IWcUmH-rztj3lGQA) web.

Para el contenido de los URI que apuntan a una o varias filas de datos de la tabla, [getType()](http://developer.android.com/reference/android/content/ContentProvider.html#getType%28android.net.Uri%29) debe devolver un tipo MIME en específico del proveedor de Android formato MIME:

* Escriba parte: vnd
* Subtipo de parte:
	+ Si el patrón de URI es para una sola fila: android.cursor. **item** /
	+ Si el patrón de URI es de más de una fila: android.cursor. **dir** /
* Proveedor parte específica: vnd.<name> . <type>

Usted provee el <name> y <type> . El <name> valor debe ser único en el mundo, y el <type> valor debe ser único en el patrón de URI correspondiente. Una buena opción para <name> es el nombre de su empresa o con alguna parte del nombre de la aplicación del paquete de Android. Una buena opción para el <type> es una cadena que identifica la tabla asociada con el URI.

Por ejemplo, si la autoridad de un proveedor es com.example.app.provider , y expone una tabla denominada table1 , el tipo MIME de varias filas en la table1 es la siguiente:

vnd.android.cursor. **dir** / vnd.com.example.provider.table1

Por una sola fila de table1 , el tipo MIME es la siguiente:

vnd.android.cursor. **artículo** / vnd.com.example.provider.table1

### Los tipos MIME para los archivos

Si su proveedor ofrece archivos, ejecutar [getStreamTypes()](http://developer.android.com/reference/android/content/ContentProvider.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29) . El método devuelve una [String](http://developer.android.com/reference/java/lang/String.html) matriz de tipos MIME para los archivos de su proveedor puede devolver para un URI determinado contenido. Usted debe filtrar los tipos MIME que ofrecen con el argumento de tipo de filtro MIME, de modo que sólo devuelven los tipos MIME que el cliente desea manejar.

Por ejemplo, considere un proveedor que ofrece imágenes fotográficas como archivos en .jpg , .png y .gif formato. Si una aplicación llama a [ContentResolver.getStreamTypes()](http://developer.android.com/reference/android/content/ContentResolver.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29) con el filtro de cadena de image/\* (algo que es una "imagen"), entonces los [ContentProvider.getStreamTypes()](http://developer.android.com/reference/android/content/ContentProvider.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29) método debe devolver la matriz:

{"Image / jpeg", "image / jpeg", "image / gif"}

Si la aplicación sólo está interesado en .jpg archivos, a continuación, puede llamar a [ContentResolver.getStreamTypes()](http://developer.android.com/reference/android/content/ContentResolver.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29) con la cadena de filtro \*\/jpeg , y [ContentProvider.getStreamTypes()](http://developer.android.com/reference/android/content/ContentProvider.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29) debe devolver:

{"Image / jpeg"}

Si su proveedor no le ofrece cualquiera de los tipos MIME solicitados en la cadena de filtro, [getStreamTypes()](http://developer.android.com/reference/android/content/ContentProvider.html#getStreamTypes%28android.net.Uri,%20java.lang.String%29) debería devolver null .

## La implementación de una clase de contrato

Una clase de contrato es un public final de clase que contiene definiciones de constantes de la URI, nombres de columnas, tipos MIME, y otros meta-datos que pertenecen al proveedor. La clase se establece un contrato entre el proveedor y otras aplicaciones, garantizando que el proveedor de acceso puede ser correctamente incluso si hay cambios en los valores reales de la URI, los nombres de columna, y así sucesivamente.

Una clase de contrato también ayuda a los desarrolladores, ya que por lo general tiene nombres mnemotécnicos para sus constantes, lo que los desarrolladores tienen menos probabilidades de utilizar los valores incorrectos para los nombres de las columnas o URI. Puesto que es una clase, puede contener documentación Javadoc. Entornos de desarrollo integrados tales como Eclipse puede auto-completar los nombres de constantes de la clase de contrato y Javadoc de visualización de las constantes.

Los desarrolladores no pueden acceder al archivo de la clase de contrato de la clase de su aplicación, pero se puede compilar de forma estática en su aplicación a partir de un .jar archivo proporcionado.

El [ContactsContract](http://developer.android.com/reference/android/provider/ContactsContract.html) clase y sus clases anidadas son ejemplos de clases de contrato.

## Permisos de ejecución proveedores de contenido

Los permisos y el acceso de todos los aspectos del sistema Android se describen en detalle en el tema [de seguridad y autorizaciones](http://developer.android.com/guide/topics/security/security.html) . El tema [de almacenamiento de datos](http://developer.android.com/guide/topics/data/data-storage.html) también se describe la seguridad y permisos en efecto para varios tipos de almacenamiento. En resumen, los puntos importantes son:

* De forma predeterminada, los archivos de datos almacenados en la memoria interna del dispositivo son privados de su aplicación y el proveedor.
* [SQLiteDatabase](http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) bases de datos que cree son privados de su aplicación y el proveedor.
* De forma predeterminada, los archivos de datos que se guardan en almacenamiento externo son públicas y el mundo puede leer. No se puede utilizar un proveedor de contenidos para restringir el acceso a los archivos de almacenamiento externo, ya que otras aplicaciones pueden utilizar otras llamadas a la API para leer y escribir en ellos.
* El método requiere abrir o crear archivos o bases de datos SQLite en el almacenamiento interno del dispositivo, puede potencialmente dar lectura y escritura a todas las demás aplicaciones. Si utiliza un archivo interno o base de datos como repositorio de su proveedor, y se la das "con permiso de lectura" o "en todo el mundo puede escribir" el acceso, los permisos que establezca para el proveedor en su manifiesto de no proteger sus datos. El acceso por defecto para los archivos y bases de datos en el almacenamiento interno es "privado", y para el repositorio de su proveedor no se debe cambiar esta situación.

Si desea utilizar permisos de proveedores de contenido para controlar el acceso a sus datos, entonces usted debe almacenar los datos en los archivos internos, bases de datos SQLite, o la "nube" (por ejemplo, en un servidor remoto), y debe mantener los archivos y bases de datos privado para su aplicación.

### Permisos de ejecución

Todas las aplicaciones pueden leer o escribir a su proveedor, incluso si los datos subyacentes es privada, ya que por defecto el proveedor no tiene permisos establecidos. Para cambiar esto, establecer los permisos para el proveedor en el archivo de manifiesto, el uso de atributos o elementos secundarios de la [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento. Puede establecer permisos que se aplican al proveedor de toda o de algunos cuadros, o incluso a ciertos archivos o los tres.

Se definen los permisos de su proveedor con una o más [<permission>](http://developer.android.com/guide/topics/manifest/permission-element.html) elementos en el archivo de manifiesto. Para que el permiso exclusivo de su proveedor, el uso de Java al estilo de alcance para el [android:name](http://developer.android.com/guide/topics/manifest/permission-element.html#nm) atributo. Por ejemplo, el nombre del permiso de lectura com.example.app.provider.permission.READ\_PROVIDER .

La siguiente lista describe el alcance de los permisos de los proveedores, empezando por los permisos que se aplican al proveedor completo y luego cada vez más fina. Más permisos específicos tienen prioridad sobre los que tienen mayor alcance:

Individual de lectura y escritura permiso de nivel de proveedor de

Un permiso que controla tanto la lectura y escritura para el proveedor completo, especificado con el [android:permission](http://developer.android.com/guide/topics/manifest/provider-element.html#prmsn) atributo de la [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento.

Separe leer y escribir permiso de nivel de proveedor de

Un permiso de lectura y un permiso de escritura para el proveedor de todo. Que se especifiquen con [android:readPermission](http://developer.android.com/guide/topics/manifest/provider-element.html#rprmsn) y [android:writePermission](http://developer.android.com/guide/topics/manifest/provider-element.html#wprmsn) atributos de la [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento. Ellos tienen prioridad sobre el permiso requerido por [android:permission](http://developer.android.com/guide/topics/manifest/provider-element.html#prmsn) .

Ruta de permiso de nivel de

Leer, escribir, o de lectura / escritura para un URI contenido de su proveedor. Se especifica cada URI que desea controlar con un [<path-permission>](http://developer.android.com/guide/topics/manifest/path-permission-element.html) elemento secundario de la [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento. Para cada URI de contenido que usted especifique, se puede especificar una lectura / escritura, un permiso de lectura, o un permiso de escritura, o las tres cosas. La lectura y escritura tienen prioridad sobre la lectura / escritura. Además, la ruta permiso de nivel tiene prioridad sobre los permisos de nivel de proveedor.

Permiso temporal

Un nivel de permisos que concede el acceso temporal a una aplicación, incluso si la aplicación no dispone de los permisos que se requieren normalmente. La característica de acceso temporal reduce el número de permisos de una aplicación tiene que solicitar en su manifiesto. Al encender los permisos temporales, las únicas aplicaciones que necesitan "permanentes" los permisos de su proveedor son los que continuamente acceden a todos sus datos.

Tenga en cuenta los permisos que necesita para implementar un proveedor de correo electrónico y la aplicación, cuando se quiere permitir que una aplicación Visor de imágenes externa para mostrar los archivos adjuntos de fotos de su proveedor. Para dar al espectador la imagen del acceso necesario sin los permisos que requieren, establecer los permisos temporales para el contenido de URI para las fotos. Diseña tu aplicación de correo electrónico para que cuando el usuario desea mostrar una foto, la aplicación envía una intención que contiene el URI de la foto de su contenido y las banderas de permiso para el visor de imágenes. El visor de imágenes a continuación, puede consultar a su proveedor de correo electrónico para recuperar la foto, a pesar de que el espectador no tiene el permiso normal de lectura de su proveedor.

Para activar los permisos temporales, ya sea establecer el [android:grantUriPermissions](http://developer.android.com/guide/topics/manifest/provider-element.html#gprmsn) atributo de la [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento, o agregar uno o más [<grant-uri-permission>](http://developer.android.com/guide/topics/manifest/grant-uri-permission-element.html) elementos secundarios a su [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento. Si utiliza los permisos temporales, tienes que llamar a [Context.revokeUriPermission()](http://developer.android.com/reference/android/content/Context.html#revokeUriPermission%28android.net.Uri,%20int%29) cada vez que se retire el apoyo a una URI de contenido de su proveedor, y el URI de contenido está asociado con un permiso temporal.

El valor del atributo determina la cantidad de su proveedor se hace accesible. Si el atributo se establece en true , entonces el sistema va a conceder el permiso temporal a su proveedor de toda, anulando cualquier otro permiso que se solicita el proveedor de nivel o permisos a nivel de ruta.

Si este indicador se establece en false , entonces debe agregar [<grant-uri-permission>](http://developer.android.com/guide/topics/manifest/grant-uri-permission-element.html) elementos secundarios a su [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento. Cada elemento secundario especifica el contenido o URI URI para que se conceda el acceso temporal.

Para delegar el acceso temporal a una aplicación, la intención debe contener la [FLAG\_GRANT\_READ\_URI\_PERMISSION](http://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) o los [FLAG\_GRANT\_WRITE\_URI\_PERMISSION](http://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION) banderas, o ambas cosas. Éstos se establecen con la [setFlags()](http://developer.android.com/reference/android/content/Intent.html#setFlags%28int%29) método.

Si el [android:grantUriPermissions](http://developer.android.com/guide/topics/manifest/provider-element.html#gprmsn) atributo no está presente, se supone que es false .

## El elemento <provider>

Al igual que [Activity](http://developer.android.com/reference/android/app/Activity.html) y el [Service](http://developer.android.com/reference/android/app/Service.html) componentes, una subclase de [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) debe estar definido en el archivo de manifiesto para su aplicación, con el [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento. El sistema Android obtiene la siguiente información del elemento:

Autoridad ( [android:authorities](http://developer.android.com/guide/topics/manifest/provider-element.html#auth) )

Los nombres simbólicos que identifican el proveedor de toda dentro del sistema. Este atributo se describe con más detalle en la sección de [Diseño de Contenido URI](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContentURI) .

Proveedor de nombre de la clase ( [android:name](http://developer.android.com/guide/topics/manifest/provider-element.html#nm) )

La clase que implementa [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) . Esta clase se describe con más detalle en la sección de [aplicación de la Clase ContentProvider](http://developer.android.com/guide/topics/providers/content-provider-creating.html#ContentProvider) .

{0}Pe{/0}{1}r{/1}{2}m{/2}{3}i{/3}{0}ss{/0}{3}i{/3}{0}ons{/0}

Los atributos que especifican los permisos que debe tener otras aplicaciones con el fin de acceder a los datos del proveedor:

* [android:grantUriPermssions](http://developer.android.com/guide/topics/manifest/provider-element.html#gprmsn) : bandera permiso temporal.
* [android:permission](http://developer.android.com/guide/topics/manifest/provider-element.html#prmsn) : Soltero proveedor de toda la lectura / escritura.
* [android:readPermission](http://developer.android.com/guide/topics/manifest/provider-element.html#rprmsn) : Proveedor de todo el permiso de lectura.
* [android:writePermission](http://developer.android.com/guide/topics/manifest/provider-element.html#wprmsn) : Proveedor de todo el permiso de escritura.

Los permisos y los atributos correspondientes se describen con más detalle en la sección [Implementación de Permisos de proveedores de contenido](http://developer.android.com/guide/topics/providers/content-provider-creating.html#Permissions) .

Inicio y control de los atributos

Estos atributos determinan cómo y cuándo se inicia el sistema Android el proveedor, las características del proceso del proveedor, y otros ajustes de tiempo de ejecución:

* [android:enabled](http://developer.android.com/guide/topics/manifest/provider-element.html#enabled) : Bandera que permite el sistema para iniciar el proveedor.
* [android:exported](http://developer.android.com/guide/topics/manifest/provider-element.html#exported) : Bandera que permite que otras aplicaciones utilicen este proveedor.
* [android:initOrder](http://developer.android.com/guide/topics/manifest/provider-element.html#init) : El orden en que se debe este proveedor de empezar, con respecto a otros proveedores en el mismo proceso.
* [android:multiProcess](http://developer.android.com/guide/topics/manifest/provider-element.html#multi) : Bandera que permite el sistema para iniciar el proveedor en el mismo proceso que el cliente que llama.
* [android:process](http://developer.android.com/guide/topics/manifest/provider-element.html#proc) : El nombre del proceso en el cual el proveedor debe ejecutar.
* [android:syncable](http://developer.android.com/guide/topics/manifest/provider-element.html#sync) : Bandera que indica que los datos del proveedor se va a sincronizarse con los datos en un servidor.

Los atributos están totalmente documentados en el tema guía para el dev [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento.

Los atributos informativos

Un icono opcional y una etiqueta para el proveedor:

* [android:icon](http://developer.android.com/guide/topics/manifest/provider-element.html#icon) : Un recurso dibujable que contiene un icono para el proveedor. El icono aparece junto a la etiqueta del proveedor en la lista de aplicaciones en Ajustes> Aplicaciones> Todos.
* [android:label](http://developer.android.com/guide/topics/manifest/provider-element.html#label) : Una etiqueta de información que describe el proveedor o de sus datos, o ambos. La etiqueta aparece en la lista de aplicaciones en Ajustes> Aplicaciones> Todo.

Los atributos están totalmente documentados en el tema guía para el dev [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elemento.

## Intenciones y acceso a datos

Las aplicaciones pueden tener acceso a un proveedor de contenido de forma indirecta con una [Intent](http://developer.android.com/reference/android/content/Intent.html) . La aplicación no llama a cualquiera de los métodos de [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) o [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) . En su lugar, envía una intención de que se inicia una actividad, que es a menudo parte de la propia aplicación del proveedor. La actividad de destino es el encargado de recuperar y visualizar los datos en su interfaz de usuario. Dependiendo de la acción en el intento, la actividad de destino también puede pedir al usuario que hacer modificaciones a los datos del proveedor. La intención también puede contener "extras" de datos que la actividad de destino se muestra en la interfaz de usuario, el usuario tiene la opción de cambiar estos datos antes de usarlo para modificar los datos en el proveedor.

Es posible que desee utilizar el acceso a la intención de ayudar a garantizar la integridad de los datos. Su proveedor puede depender de tener los datos insertados, actualizados y eliminados de acuerdo a la lógica de negocio estrictamente definido. Si este es el caso, permitiendo que otras aplicaciones para modificar directamente los datos puede conducir a datos no válidos. Si desea acceder a los desarrolladores utilizar la intención, asegúrese de documentar a fondo. Explíqueles por qué el acceso con la intención de interfaz de usuario de su propia aplicación es mejor que tratar de modificar sus datos con su código.

Manejo de un intento de entrada que desea modificar los datos de su proveedor no es diferente de la manipulación de las intenciones de otros. Puede obtener más información sobre el uso de los intentos por leer los temas [Intenciones y filtros de Intención](http://developer.android.com/guide/topics/intents/intents-filters.html) .

**CALENDARIO DE PROVEEDORES**

## En este documento

1. [Conceptos básicos](http://developer.android.com/guide/topics/providers/calendar-provider.html#overview)
2. [Permisos de usuario](http://developer.android.com/guide/topics/providers/calendar-provider.html#manifest)
3. [Calendarios de mesa](http://developer.android.com/guide/topics/providers/calendar-provider.html#calendar)
	1. [Consultar el calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#query)
	2. [Modificación de un calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#modify-calendar)
	3. [Inserción de un calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#insert-calendar)
4. [Eventos de mesa](http://developer.android.com/guide/topics/providers/calendar-provider.html#events)
	1. [Adición de Eventos](http://developer.android.com/guide/topics/providers/calendar-provider.html#add-event)
	2. [Actualización de las citas](http://developer.android.com/guide/topics/providers/calendar-provider.html#update-event)
	3. [Eliminación de eventos](http://developer.android.com/guide/topics/providers/calendar-provider.html#delete-event)
5. [Los asistentes de mesa](http://developer.android.com/guide/topics/providers/calendar-provider.html#attendees)
	1. [Adición de los asistentes](http://developer.android.com/guide/topics/providers/calendar-provider.html#add-attendees)
6. [Recordatorios de la tabla](http://developer.android.com/guide/topics/providers/calendar-provider.html#reminders)
	1. [Adición de recordatorios](http://developer.android.com/guide/topics/providers/calendar-provider.html#add-reminders)
7. [Las instancias de mesa](http://developer.android.com/guide/topics/providers/calendar-provider.html#instances)
	1. [Consulta la tabla de instancias](http://developer.android.com/guide/topics/providers/calendar-provider.html#query-instances)
8. [Calendario Intenciones](http://developer.android.com/guide/topics/providers/calendar-provider.html#intents)
	1. [El uso de un intento de insertar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-insert)
	2. [El uso de un intento de editar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-edit)
	3. [Uso de las intenciones para ver los datos del calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-view)
9. [Adaptadores de sincronización](http://developer.android.com/guide/topics/providers/calendar-provider.html#sync-adapter)

## Clases de clave

1. [CalendarContract.Calendars](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html)
2. [CalendarContract.Events](http://developer.android.com/reference/android/provider/CalendarContract.Events.html)
3. [CalendarContract.Attendees](http://developer.android.com/reference/android/provider/CalendarContract.Attendees.html)
4. [CalendarContract.Reminders](http://developer.android.com/reference/android/provider/CalendarContract.Reminders.html)

El proveedor de Calendario es un repositorio de eventos en el calendario de un usuario. La API de proveedor de Calendario le permite realizar consultas, insertar, actualizar y eliminar operaciones de los calendarios, eventos, los asistentes, recordatorios, etc.

La API de proveedor de Calendario puede ser utilizado por aplicaciones y adaptadores de sincronización. Las reglas varían dependiendo de qué tipo de programa está haciendo las llamadas. Este documento se centra principalmente en el uso de la API de proveedor de calendario como una aplicación. Para una discusión de cómo los adaptadores de sincronización son diferentes, consulte [los adaptadores de sincronización](http://developer.android.com/guide/topics/providers/calendar-provider.html#sync-adapter) .

Normalmente, para leer o escribir los datos del calendario, que se manifiesta de una aplicación debe incluir los permisos adecuados, descritos en [los permisos de usuario](http://developer.android.com/guide/topics/providers/calendar-provider.html#manifest) . Para hacer más fácil la realización de las operaciones más comunes, el proveedor de calendario ofrece un conjunto de propósitos, como se describe en [Intenciones del calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#intents) . Estos intentos de llevar a los usuarios a la aplicación del calendario para insertar, ver y editar los eventos. El usuario interactúa con la aplicación de calendario y luego regresa a la solicitud original. Por lo tanto su aplicación no tiene que solicitar permisos, ni necesidad de proporcionar una interfaz de usuario para ver o crear eventos.

## Conceptos básicos

[Los proveedores de contenido](http://developer.android.com/guide/topics/providers/content-providers.html) almacenar datos y hacerla accesible a las aplicaciones. Los proveedores de contenidos ofrecidos por la plataforma Android (incluyendo el Proveedor Calendario) suelen exponer los datos como un conjunto de tablas basado en un modelo de base de datos relacional, donde cada fila es un registro y cada columna es un dato de un tipo particular y su significado. A través de la API de proveedor de calendario, aplicaciones y adaptadores de sincronización puede obtener acceso lectura / escritura a las tablas de base de datos que contienen datos de un usuario del calendario.

Cada proveedor de contenido expone un URI público (envuelto como [Uri](http://developer.android.com/reference/android/net/Uri.html) objeto) que identifica de manera única a su conjunto de datos. Un proveedor de contenidos que controla varios conjuntos de datos (múltiples tablas) expone un URI diferente para cada uno. Todos los URIs para los proveedores de comenzar con el "contenido :/ /" cadena. Esto identifica a los datos que está siendo controlado por un proveedor de contenido. El proveedor de calendario define constantes para el URI para cada una de sus clases (tablas). Estas infecciones tienen el formato *<class> .*CONTENT\_URI . Por ejemplo, [Events.CONTENT\_URI](http://developer.android.com/reference/android/provider/CalendarContract.Events.html#CONTENT_URI) .

La figura 1 muestra una representación gráfica del modelo de datos de proveedor de Calendario. Muestra los cuadros principales y los campos que los vinculan entre sí.



**FIGURA I.** Proveedor de datos de calendario modelo.

Un usuario puede tener varios calendarios y calendarios diferentes pueden estar asociados con diferentes tipos de cuentas (Google Calendar, Exchange, y así sucesivamente).

El [CalendarContract](http://developer.android.com/reference/android/provider/CalendarContract.html) define el modelo de datos de calendario y la información de eventos relacionados. Estos datos se almacenan en un número de tablas, se enumeran a continuación.

|  |  |
| --- | --- |
| **Tabla (Clase)** | **Descripción** |
| [CalendarContract.Calendars](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html) | Esta tabla contiene la información del calendario específico. Cada fila de esta tabla contiene los detalles de un solo calendario, tales como el nombre, el color, la información de sincronización, y así sucesivamente. |
| [CalendarContract.Events](http://developer.android.com/reference/android/provider/CalendarContract.Events.html) | Esta tabla contiene la información específica del evento. Cada fila de esta tabla tiene la información para un solo caso, por ejemplo, título del evento, lugar, hora de inicio, hora de finalización, y así sucesivamente. El evento puede ocurrir una sola vez o puede repetirse varias veces. Los asistentes, recordatorios, y las propiedades extendidas se almacenan en tablas separadas. Cada uno tiene una [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#EVENT_ID) que hace referencia al [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) en la tabla de eventos. |
| [CalendarContract.Instances](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html) | Esta tabla contiene el inicio y hora de finalización de cada ocurrencia de un evento. Cada fila de esta tabla representa una ocurrencia de un evento único. Para los eventos de una sola vez existe una correlación de 1:1 de los casos a los eventos. Para los eventos recurrentes, varias filas se generan automáticamente que se corresponden con varias apariciones de ese evento. |
| [CalendarContract.Attendees](http://developer.android.com/reference/android/provider/CalendarContract.Attendees.html) | Esta tabla contiene el asistente de eventos (invitado) de la información. Cada fila representa un invitado de un evento. Se especifica el tipo de cliente y la respuesta del huésped la asistencia para el evento. |
| [CalendarContract.Reminders](http://developer.android.com/reference/android/provider/CalendarContract.Reminders.html) | Esta tabla contiene los datos de alerta / notificación. Cada fila representa una única alerta para un evento. Un evento puede tener varios recordatorios. El número máximo de recordatorios por evento se especifica en [MAX\_REMINDERS](http://developer.android.com/reference/android/provider/CalendarContract.CalendarColumns.html#MAX_REMINDERS) , que se ha establecido con el adaptador de sincronización que posee el calendario. Los recordatorios se especifican en cuestión de minutos antes del evento y tener un método que determina la forma en que el usuario recibirá una alerta. |

La API de proveedor de calendario está diseñado para ser flexible y potente. Al mismo tiempo, es importante para proporcionar una buena experiencia del usuario final y proteger la integridad del calendario y sus datos. Con este fin, aquí están algunas cosas a tener en cuenta al utilizar la API:

* **Inserción, actualización y visualización de los eventos del calendario.** Para insertar directamente, modificar, y leer los acontecimientos desde el proveedor de calendario, se necesitan los correspondientes [permisos](http://developer.android.com/guide/topics/providers/calendar-provider.html#manifest) . Sin embargo, si usted no está creando una aplicación de calendario de pleno derecho o el adaptador de sincronización, solicitar estos permisos no es necesario. En su lugar, puede utilizar los intentos apoyados por aplicación del calendario de Android a mano de lectura y escritura a esa aplicación. Al utilizar los intentos, la aplicación envía a los usuarios a la aplicación del calendario para realizar la operación deseada en un formulario pre-llenado. Después de que hayas terminado, que regresará a su aplicación. Mediante el diseño de su aplicación para realizar operaciones comunes a través de la Agenda, proporcionar a los usuarios una interfaz coherente y fácil robusto. Este es el enfoque recomendado. Para obtener más información, consulte [Intenciones del calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#intents) .
* **Sincronizar los adaptadores.** Un adaptador de sincronización sincroniza los datos de calendario en el dispositivo de un usuario a otro servidor o fuente de datos. En el [CalendarContract.Calendars](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html) y [CalendarContract.Events](http://developer.android.com/reference/android/provider/CalendarContract.Events.html) tablas, hay columnas que están reservadas para los adaptadores de sincronización para su uso. El proveedor y las aplicaciones no deben modificar. De hecho, no son accesibles a menos que se accede a ellos como un adaptador de sincronización. Para obtener más información acerca de los adaptadores de sincronización, consulte [los adaptadores de sincronización](http://developer.android.com/guide/topics/providers/calendar-provider.html#sync-adapter) .

## Permisos de usuario

Para leer los datos del calendario, una aplicación debe incluir el [READ\_CALENDAR](http://developer.android.com/reference/android/Manifest.permission.html#READ_CALENDAR) permiso en el archivo de manifiesto. Debe incluir la [WRITE\_CALENDAR](http://developer.android.com/reference/android/Manifest.permission.html#WRITE_CALENDAR) permiso para borrar, insertar o actualizar los datos del calendario:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"...>

 <uses-sdk android:minSdkVersion="14" />

 <Usos de permiso de android: name = "android.permission.READ\_CALENDAR "/>

 <Usos de permiso de android: name = "android.permission.WRITE\_CALENDAR "/>

 ...

</manifest>

## Calendarios de Mesa

El [CalendarContract.Calendars](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html) de mesa contiene los detalles de los calendarios individuales. Las columnas siguientes calendarios pueden ser escritos por tanto una aplicación y un [adaptador de sincronización](http://developer.android.com/guide/topics/providers/calendar-provider.html#sync-adapter) . Para obtener una lista completa de campos admitidos, vea el [CalendarContract.Calendars](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html) de referencia.

|  |  |
| --- | --- |
| **Constante** | **Descripción** |
| [NAME](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html#NAME) | El nombre del calendario. |
| [CALENDAR\_DISPLAY\_NAME](http://developer.android.com/reference/android/provider/CalendarContract.CalendarColumns.html#CALENDAR_DISPLAY_NAME) | El nombre de este calendario que se muestra al usuario. |
| [VISIBLE](http://developer.android.com/reference/android/provider/CalendarContract.CalendarColumns.html#VISIBLE) | Un booleano que indica si el calendario se ha seleccionado para que se muestre. Un valor de 0 indica que los eventos relacionados con este calendario no debería ser mostrado. Un valor de 1 indica que los eventos relacionados con este calendario debe ser demostrado. Este valor afecta a la generación de filas de la [CalendarContract.Instances](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html) de la tabla. |
| [SYNC\_EVENTS](http://developer.android.com/reference/android/provider/CalendarContract.CalendarColumns.html#SYNC_EVENTS) | Un booleano que indica si el calendario debe ser sincronizado y han sus eventos almacenados en el dispositivo. Un valor de 0 dice que no se sincroniza el calendario o guardar sus eventos en el dispositivo. Un valor de 1 indica los eventos de sincronización para el calendario y guardar sus eventos en el dispositivo. |

### Consultar el calendario

Aquí hay un ejemplo que muestra cómo obtener los calendarios que son propiedad de un usuario en particular. Por razones de simplicidad, en este ejemplo la operación de consulta se muestra en el hilo interfaz de usuario ("hilo principal"). En la práctica, esto se debe hacer en un subproceso asincrónico en lugar de en el hilo principal. Para mayor discusión, véase [Cargadores](http://developer.android.com/guide/topics/fundamentals/loaders.html) . Si usted no está leyendo los datos, sino modificarlo, consulte [AsyncQueryHandler](http://developer.android.com/reference/android/content/AsyncQueryHandler.html) .

/ / Proyección de la matriz. La creación de los índices de esta serie en lugar de hacer

/ Consultas / dinámico mejora el rendimiento.

public String static final [] EVENT\_PROJECTION = new String [] {

 Calendarios\_ID, / / ​​0

 CalendariosNombre\_cuenta, / / ​​1

 CalendariosCALENDAR\_DISPLAY\_NAME, / / ​​2

 CalendariosOWNER\_ACCOUNT / / 3

};

/ / Los índices de la matriz de proyección arriba.

privada PROJECTION\_ID\_INDEX static final int = 0;

privada PROJECTION\_ACCOUNT\_NAME\_INDEX static final int = 1;

privada PROJECTION\_DISPLAY\_NAME\_INDEX static final int = 2;

privada PROJECTION\_OWNER\_ACCOUNT\_INDEX static final int = 3;

### ¿Por qué debe incluir ACCOUNT\_TYPE?

Si usted consulta en una [Calendars.ACCOUNT\_NAME](http://developer.android.com/reference/android/provider/CalendarContract.SyncColumns.html#ACCOUNT_NAME) , también debe incluir [Calendars.ACCOUNT\_TYPE](http://developer.android.com/reference/android/provider/CalendarContract.SyncColumns.html#ACCOUNT_TYPE) en la selección. Esto se debe a una cuenta determinada sólo se considera única tanto por su ACCOUNT\_NAME y su ACCOUNT\_TYPE . El ACCOUNT\_TYPE es la cadena que corresponde a la autentificación de cuenta que se utiliza cuando la cuenta se registró en la [AccountManager](http://developer.android.com/reference/android/accounts/AccountManager.html) . Hay también un tipo especial de cuenta llamada [ACCOUNT\_TYPE\_LOCAL](http://developer.android.com/reference/android/provider/CalendarContract.html#ACCOUNT_TYPE_LOCAL) para los calendarios que no están asociados con una cuenta de dispositivo. [ACCOUNT\_TYPE\_LOCAL](http://developer.android.com/reference/android/provider/CalendarContract.html#ACCOUNT_TYPE_LOCAL) cuentas no se sincronizan.

En la siguiente parte del ejemplo, la construcción de su consulta. La selección se especifican los criterios para la consulta. En este ejemplo, la consulta se busca para los calendarios que tienen el ACCOUNT\_NAME "sampleuser@google.com", el ACCOUNT\_TYPE "com.google", y el OWNER\_ACCOUNT "sampleuser@google.com". Si desea ver todos los calendarios que un usuario ha visto, no sólo de los calendarios son propiedad del usuario, omita el OWNER\_ACCOUNT . La consulta devuelve un [Cursor](http://developer.android.com/reference/android/database/Cursor.html) objeto que se puede utilizar para recorrer el conjunto de resultados devuelto por la consulta de bases de datos. Para más discusión sobre el uso de consultas de los proveedores de contenido, vea [Proveedores de Contenido](http://developer.android.com/guide/topics/providers/content-providers.html) .

/ / Ejecutar la consulta

Cursor act = null;

ContentResolver cr = getContentResolver ();

Uri uri = calendarios.CONTENT\_URI;

Selección String = "((" + Calendarios.Nombre de la cuenta Y

 CalendariosTipo de Cuenta: Y

 CalendariosOWNER\_ACCOUNT + "=?))";

String [] selectionArgs = new String [] {"sampleuser@gmail.com", "com.google",

 "Sampleuser@gmail.com"};

/ / Enviar la consulta y obtener un objeto cursor hacia atrás.

act = cr.query (Uri, EVENT\_PROJECTION, selección, selectionArgs, null);

Esta sección siguiente se utiliza el cursor para desplazarse por el conjunto de resultados. Se utiliza las constantes que se establecieron en el comienzo del ejemplo para devolver los valores para cada campo.

/ / Utiliza los cursores para desplazarse por los registros devueltos

mientras que (cur.moveToNext ()) {

 a largo calid = 0;

 Cadena displayName = null;

 Cadena accountName = null;

 Cadena ownerName = null;

 / / Obtener los valores de campo

 calid = cur.getLong (PROJECTION\_ID\_INDEX);

 displayName = cur.getString (PROJECTION\_DISPLAY\_NAME\_INDEX);

 accountName = cur.getString (PROJECTION\_ACCOUNT\_NAME\_INDEX);

 ownerName = cur.getString (PROJECTION\_OWNER\_ACCOUNT\_INDEX);

 / / Hacer algo con los valores ...

 ...

}

### Modificación de un calendario

Para llevar a cabo una actualización de un calendario, que puede proporcionar la [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) del calendario, ya sea como una identificación adjunta a la Uri ( [withAppendedId()](http://developer.android.com/reference/android/content/ContentUris.html#withAppendedId%28android.net.Uri,%20long%29) ) o como el elemento de primera selección. La selección debe comenzar con "\_id=?" , y la primera selectionArg debe ser el [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) del calendario. También puede hacer cambios mediante la codificación de la identificación en el URI. En este ejemplo se cambia el nombre de un calendario de visualización con la ( [withAppendedId()](http://developer.android.com/reference/android/content/ContentUris.html#withAppendedId%28android.net.Uri,%20long%29) método):

privada DEBUG\_TAG static final String = "MyActivity";

...

a largo calid = 2;

Los valores ContentValues ​​= new ContentValues ​​();

/ / El nuevo nombre para el calendario

values.put (Calendars.CALENDAR\_DISPLAY\_NAME, "Trevor Calendario");

Uri updateUri = ContentUris.withAppendedId (Calendars.CONTENT\_URI, calid);

int filas = getContentResolver () actualización (updateUri, los valores, null, null).;

Log.i (DEBUG\_TAG, "Filas actualizadas:" + filas);

### Inserción de un calendario

**Calendarios están diseñados para ser manejados básicamente por un adaptador de sincronización, por lo que sólo debe insertar nuevos calendarios como un adaptador de sincronización. En su mayor parte, las aplicaciones sólo se pueden hacer cambios superficiales a los calendarios, como cambiar el nombre para mostrar. Si una aplicación necesita crear un calendario local, se puede hacer esto mediante la realización de la inserción del calendario como un adaptador de sincronización, utilizando un** [**ACCOUNT\_TYPE**](http://developer.android.com/reference/android/provider/CalendarContract.SyncColumns.html#ACCOUNT_TYPE) **de** [**ACCOUNT\_TYPE\_LOCAL**](http://developer.android.com/reference/android/provider/CalendarContract.html#ACCOUNT_TYPE_LOCAL) **.** [**ACCOUNT\_TYPE\_LOCAL**](http://developer.android.com/reference/android/provider/CalendarContract.html#ACCOUNT_TYPE_LOCAL) **es un tipo de cuenta especial para los calendarios a los que no están asociados a una cuenta del dispositivo. Calendarios de este tipo no se sincronizan con un servidor. Para una discusión de los adaptadores de sincronización, véase** [**adaptadores de sincronización**](http://developer.android.com/guide/topics/providers/calendar-provider.html#sync-adapter) **.**

## Tabla de eventos

El [CalendarContract.Events](http://developer.android.com/reference/android/provider/CalendarContract.Events.html) tabla contiene los detalles de los eventos individuales. Para agregar, actualizar o eliminar eventos, una aplicación debe incluir el [WRITE\_CALENDAR](http://developer.android.com/reference/android/Manifest.permission.html#WRITE_CALENDAR) permiso en su [archivo de manifiesto](http://developer.android.com/guide/topics/providers/calendar-provider.html#manifest) .

Las columnas siguientes eventos pueden ser escritos por tanto una aplicación y un adaptador de sincronización. Para obtener una lista completa de campos admitidos, vea el [CalendarContract.Events](http://developer.android.com/reference/android/provider/CalendarContract.Events.html) de referencia.

|  |  |
| --- | --- |
| **Constante** | **Descripción** |
| [CALENDAR\_ID](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#CALENDAR_ID) | El [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) del calendario del evento pertenece. |
| [ORGANIZER](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#ORGANIZER) | Correo electrónico del organizador (propietario) del evento. |
| [TITLE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#TITLE) | El título del evento. |
| [EVENT\_LOCATION](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#EVENT_LOCATION) | ¿Dónde tiene lugar el evento.  |
| [DESCRIPTION](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DESCRIPTION) | La descripción del evento. |
| [DTSTART](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DTSTART) | La hora del comienzo del evento en milisegundos de UTC desde la época.  |
| [DTEND](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DTEND) | El tiempo que el evento termina en milisegundos de UTC desde la época.  |
| [EVENT\_TIMEZONE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#EVENT_TIMEZONE) | La zona de tiempo para el evento. |
| [EVENT\_END\_TIMEZONE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#EVENT_END_TIMEZONE) | La zona horaria de la hora de finalización del evento. |
| [DURATION](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DURATION) | La duración del evento en [RFC5545](http://www.google.com/url?q=http%3A%2F%2Ftools.ietf.org%2Fhtml%2Frfc5545%23section-3.8.2.5&sa=D&sntz=1&usg=AFQjCNFTV2YFSQLzKadoCBj6g-PV0_sOUg) formato. Por ejemplo, un valor de "PT1H" indica que el evento debe durar una hora, y un valor de "P2W" indica una duración de 2 semanas.  |
| [ALL\_DAY](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#ALL_DAY) | Un valor de 1 indica que este evento ocupa todo el día, tal como se define por la zona horaria local. Un valor de 0 indica que es un hecho frecuente que puede empezar y terminar en cualquier momento durante el día. |
| [RRULE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RRULE) | La regla de recurrencia para el formato del evento. Por ejemplo, "FREQ=WEEKLY;COUNT=10;WKST=SU" . Puedes encontrar más ejemplos [aquí](http://www.google.com/url?q=http%3A%2F%2Ftools.ietf.org%2Fhtml%2Frfc5545%23section-3.8.5.3&sa=D&sntz=1&usg=AFQjCNHFqQ0O2Zh4Yr1qry4UKdUyJJSXnQ) . |
| [RDATE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RDATE) | La recurrencia de las fechas para el evento. Normalmente, se utiliza [RDATE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RDATE) en conjunto con [RRULE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RRULE) para definir un conjunto agregado de los sucesos que se repiten. Para una discusión más, consulte la [especificación RFC5545](http://www.google.com/url?q=http%3A%2F%2Ftools.ietf.org%2Fhtml%2Frfc5545%23section-3.8.5.2&sa=D&sntz=1&usg=AFQjCNEvy-8jaBALAgWOxtiQ7UXsUoVsfQ) . |
| [AVAILABILITY](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#AVAILABILITY) | Si este evento cuenta con el tiempo ocupado o es el tiempo libre que se pueden programar más.  |
| [GUESTS\_CAN\_MODIFY](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#GUESTS_CAN_MODIFY) | Tanto si los clientes puede modificar el evento.  |
| [GUESTS\_CAN\_INVITE\_OTHERS](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#GUESTS_CAN_INVITE_OTHERS) | Tanto si los clientes pueden invitar a otros huéspedes.  |
| [GUESTS\_CAN\_SEE\_GUESTS](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#GUESTS_CAN_SEE_GUESTS) | Ya sea que los huéspedes pueden ver la lista de asistentes. |

### Adición de Eventos

Cuando su aplicación inserta un nuevo evento, le recomendamos que utilice un [INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) intención, como se describe en el [uso de un intento de insertar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-insert) . Sin embargo, si usted necesita, usted puede insertar eventos directamente. En esta sección se describe cómo hacer esto.

Estas son las reglas para la inserción de un nuevo evento:

* Usted debe incluir [CALENDAR\_ID](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#CALENDAR_ID) y [DTSTART](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DTSTART) .
* Usted debe incluir una [EVENT\_TIMEZONE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#EVENT_TIMEZONE) . Para obtener una lista de tiempo de instalado el sistema de ID de la zona, el uso de [getAvailableIDs()](http://developer.android.com/reference/java/util/TimeZone.html#getAvailableIDs%28%29) . Tenga en cuenta que esta regla no se aplica si vas a insertar un evento a través de la [INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) intención, se describe en el [uso de un intento de insertar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-insert) en ese escenario, una zona horaria por defecto se suministra.
* Para eventos no recurrentes, se debe incluir [DTEND](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DTEND) .
* Para los eventos recurrentes, debe incluir un [DURATION](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DURATION) además [RRULE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RRULE) o [RDATE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RDATE) . Tenga en cuenta que esta regla no se aplica si vas a insertar un evento a través de la [INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) intención, se describe en el [uso de un intento de insertar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-insert) en ese escenario, puede utilizar un [RRULE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RRULE) junto con [DTSTART](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DTSTART) y [DTEND](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DTEND) , y convierte la aplicación Calendario a una duración de forma automática.

Aquí es un ejemplo de la inserción de un evento. Esto se realiza en el hilo de la interfaz de usuario para simplificar. En la práctica, las inserciones y actualizaciones se debe hacer en un subproceso asincrónico para mover la acción en un subproceso en segundo plano. Para obtener más información, consulte [AsyncQueryHandler](http://developer.android.com/reference/android/content/AsyncQueryHandler.html) .

a largo calid = 3;

a largo startMillis = 0;

a largo endMillis = 0;

Calendario beginTime = Calendar.getInstance ();

beginTime.set (2012, 9, 14, 7, 30);

startMillis beginTime.getTimeInMillis = ();

Calendario endTime = Calendar.getInstance ();

endTime.set (2012, 9, 14, 8, 45);

endMillis endTime.getTimeInMillis = ();

...

ContentResolver cr = getContentResolver ();

Los valores ContentValues ​​= new ContentValues ​​();

values.put (Events.DTSTART, startMillis);

values.put (Events.DTEND, endMillis);

values.put (Events.TÍTULO: "Jazzercise");

values.put (Events.DESCRIPCIÓN: "Grupo de entrenamiento");

values.put (Events.CALENDAR\_ID, calid);

values.put (Events.EVENT\_TIMEZONE, "America / Los\_Angeles");

Uri uri = cr.insert (Events.CONTENT\_URI, valores);

/ / Obtiene el identificador de evento que es el último elemento de la Uri

a largo IdDeSuceso = Long.parseLong (uri.getLastPathSegment ());

/ /

/ / ... hacer algo con el identificador de suceso

/ /

/ /

**Nota:** Vea cómo este ejemplo refleja el identificador de evento después de que el evento se crea. Esta es la forma más sencilla de obtener una ID de evento. A menudo se necesita el ID de evento para realizar las operaciones de otro calendario, por ejemplo, para añadir o recordatorios a los asistentes a un evento.

### Actualización de las citas

Cuando su aplicación desea que el usuario pueda editar un evento, le recomendamos que utilice un [EDIT](http://developer.android.com/reference/android/content/Intent.html#ACTION_EDIT) intención, como se describe en [Uso de la intención de editar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-edit) . Sin embargo, si es necesario, puede editar los eventos directamente. Para llevar a cabo una actualización de un evento, puede proporcionar la \_ID del evento, ya sea como una identificación adjunta a la Uri ( [withAppendedId()](http://developer.android.com/reference/android/content/ContentUris.html#withAppendedId%28android.net.Uri,%20long%29) ) o como el elemento de primera selección. La selección debe comenzar con "\_id=?" , y la primera selectionArg debe ser el \_ID del evento. Usted también puede hacer actualizaciones a través de una selección sin identificación. Aquí es un ejemplo de la actualización de un evento. Cambia el título del evento con el [withAppendedId()](http://developer.android.com/reference/android/content/ContentUris.html#withAppendedId%28android.net.Uri,%20long%29) enfoque:

privada DEBUG\_TAG static final String = "MyActivity";

...

a largo IdDeSuceso = 188;

...

ContentResolver cr = getContentResolver ();

Los valores ContentValues ​​= new ContentValues ​​();

Uri updateUri = null;

/ / El nuevo título para el evento

values.put (Events.TÍTULO: "Kickboxing");

myUri = ContentUris.withAppendedId (Events.CONTENT\_URI, IdDeSuceso);

int filas = getContentResolver () actualización (updateUri, los valores, null, null).;

Log.i (DEBUG\_TAG, "Filas actualizadas:" + filas);

### Eliminación de eventos

Usted puede eliminar un evento ya sea por su [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) como un identificador adjunta en la URI, o mediante el uso de selección estándar. Si utiliza un ID de adjunto, que tampoco se puede hacer una selección. Hay dos versiones de eliminación: como una aplicación y como un adaptador de sincronización. Una eliminación aplicación establece la columna eliminada a 1. Esta bandera indica que el adaptador de sincronización que la fila se ha eliminado y que esta supresión se propaga al servidor. Un adaptador de sincronización de eliminación elimina el evento de la base de datos junto con todos sus datos asociados. Aquí está un ejemplo de aplicación la eliminación de un caso a través de su [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) :

privada DEBUG\_TAG static final String = "MyActivity";

...

a largo IdDeSuceso = 201;

...

ContentResolver cr = getContentResolver ();

Los valores ContentValues ​​= new ContentValues ​​();

Uri deleteUri = null;

deleteUri = ContentUris.withAppendedId (Events.CONTENT\_URI, IdDeSuceso);

int filas = getContentResolver () delete (deleteUri, null, null).;

Log.i (DEBUG\_TAG, "las filas eliminadas:" + filas);

## Los asistentes Tabla

Cada fila de la [CalendarContract.Attendees](http://developer.android.com/reference/android/provider/CalendarContract.Attendees.html) tabla representa una única participante o invitado de un evento. Llamar a [query()](http://developer.android.com/reference/android/provider/CalendarContract.Reminders.html#query%28android.content.ContentResolver,%20long,%20java.lang.String%5B%5D%29) devuelve una lista de asistentes para el evento con la dada [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#EVENT_ID) . Este [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#EVENT_ID) debe coincidir con el [\_ID](http://developer.android.com/reference/android/provider/BaseColumns.html#_ID) de un evento en particular.

La siguiente tabla muestra los campos de escritura. Al insertar un nuevo asistente, debe incluir a todos ellos, excepto ATTENDEE\_NAME .

|  |  |
| --- | --- |
| **Constante** | **Descripción** |
| [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#EVENT_ID) | El ID del evento. |
| [ATTENDEE\_NAME](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_NAME) | El nombre del asistente. |
| [ATTENDEE\_EMAIL](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_EMAIL) | La dirección de correo electrónico del participante. |
| [ATTENDEE\_RELATIONSHIP](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_RELATIONSHIP) | La relación de los asistentes al evento. Uno de:* [RELATIONSHIP\_ATTENDEE](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#RELATIONSHIP_ATTENDEE)
* [RELATIONSHIP\_NONE](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#RELATIONSHIP_NONE)
* [RELATIONSHIP\_ORGANIZER](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#RELATIONSHIP_ORGANIZER)
* [RELATIONSHIP\_PERFORMER](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#RELATIONSHIP_PERFORMER)
* [RELATIONSHIP\_SPEAKER](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#RELATIONSHIP_SPEAKER)
 |
| [ATTENDEE\_TYPE](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_TYPE) | El tipo de asistentes. Uno de: * [TYPE\_REQUIRED](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#TYPE_REQUIRED)
* [TYPE\_OPTIONAL](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#TYPE_OPTIONAL)
 |
| [ATTENDEE\_STATUS](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_STATUS) | El estado de presencia de los asistentes. Uno de:* [ATTENDEE\_STATUS\_ACCEPTED](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_STATUS_ACCEPTED)
* [ATTENDEE\_STATUS\_DECLINED](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_STATUS_DECLINED)
* [ATTENDEE\_STATUS\_INVITED](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_STATUS_INVITED)
* [ATTENDEE\_STATUS\_NONE](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_STATUS_NONE)
* [ATTENDEE\_STATUS\_TENTATIVE](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#ATTENDEE_STATUS_TENTATIVE)
 |

### Adición de los asistentes

Aquí hay un ejemplo que añade un asistente a un evento único. Tenga en cuenta que el [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#EVENT_ID) se requiere:

a largo IdDeSuceso = 202;

...

ContentResolver cr = getContentResolver ();

Los valores ContentValues ​​= new ContentValues ​​();

values.put (Attendees.ATTENDEE\_NAME, "Trevor");

values.put (Attendees.ATTENDEE\_EMAIL, "trevor@example.com");

values.put (Attendees.ATTENDEE\_RELATIONSHIP, a los asistentes.RELATIONSHIP\_ATTENDEE);

values.put (Attendees.ATTENDEE\_TYPE, a los asistentes.TYPE\_OPTIONAL);

values.put (Attendees.ATTENDEE\_STATUS, a los asistentes.ATTENDEE\_STATUS\_INVITED);

values.put (Attendees.Event\_id, IdDeSuceso);

Uri uri = cr.insert (Attendees.CONTENT\_URI, valores);

## Recordatorios de la tabla

Cada fila de la [CalendarContract.Reminders](http://developer.android.com/reference/android/provider/CalendarContract.Reminders.html) la tabla representa un recordatorio para un evento único. Llamar a [query()](http://developer.android.com/reference/android/provider/CalendarContract.Reminders.html#query%28android.content.ContentResolver,%20long,%20java.lang.String%5B%5D%29) devuelve una lista de recordatorios para el evento con la dada [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.AttendeesColumns.html#EVENT_ID) .

La siguiente tabla muestra los campos de escritura para los recordatorios. Todos ellos deben ser incluidos cuando se inserta un nuevo recordatorio. Tenga en cuenta que los adaptadores de sincronización de especificar los tipos de recordatorios que apoyan en el [CalendarContract.Calendars](http://developer.android.com/reference/android/provider/CalendarContract.Calendars.html) mesa. Ver [ALLOWED\_REMINDERS](http://developer.android.com/reference/android/provider/CalendarContract.CalendarColumns.html#ALLOWED_REMINDERS) para más detalles.

|  |  |
| --- | --- |
| **Constante** | **descripción** |
| [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#EVENT_ID) | El ID del evento. |
| [MINUTES](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#MINUTES) | Los minutos antes del evento de que el aviso debe disparar. |
| [METHOD](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#METHOD) | El método de alarma, tal como se establece en el servidor. Uno de:* [METHOD\_ALERT](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#METHOD_ALERT)
* [METHOD\_DEFAULT](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#METHOD_DEFAULT)
* [METHOD\_EMAIL](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#METHOD_EMAIL)
* [METHOD\_SMS](http://developer.android.com/reference/android/provider/CalendarContract.RemindersColumns.html#METHOD_SMS)
 |

### Adición de recordatorios

En este ejemplo se agrega un recordatorio para un evento. El recordatorio se dispara a 15 minutos antes del evento.

a largo IdDeSuceso = 221;

...

ContentResolver cr = getContentResolver ();

Los valores ContentValues ​​= new ContentValues ​​();

values.put (Reminders.MINUTOS, 15);

values.put (Reminders.Event\_id, IdDeSuceso);

values.put (Reminders.Método, los recordatorios.METHOD\_ALERT);

Uri uri = cr.insert (Reminders.CONTENT\_URI, valores);

## Instancias Tabla

El [CalendarContract.Instances](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html) tabla contiene el inicio y hora de finalización de las ocurrencias de un evento. Cada fila de esta tabla representa una ocurrencia de un evento único. La tabla de casos no se puede escribir y sólo proporciona una forma de consultar las ocurrencias de eventos.

La siguiente tabla muestra algunos de los campos que se pueden consultar en la de una instancia. Tenga en cuenta que la zona horaria se define por [KEY\_TIMEZONE\_TYPE](http://developer.android.com/reference/android/provider/CalendarContract.CalendarCache.html#KEY_TIMEZONE_TYPE) y [KEY\_TIMEZONE\_INSTANCES](http://developer.android.com/reference/android/provider/CalendarContract.CalendarCache.html#KEY_TIMEZONE_INSTANCES) .

|  |  |
| --- | --- |
| **Constante** | **Descripción** |
| [BEGIN](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#BEGIN) | La hora de inicio de la instancia, en milisegundos de UTC. |
| [END](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#END) | La hora de finalización de la instancia, en milisegundos de UTC. |
| [END\_DAY](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#END_DAY) | El día final de la instancia de Julián, en relación con la zona horaria del calendario.  |
| [END\_MINUTE](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#END_MINUTE) | En el minuto final de la instancia medido desde la medianoche en la zona horaria del calendario. |
| [EVENT\_ID](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#EVENT_ID) | El \_ID del evento de esta instancia. |
| [START\_DAY](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#START_DAY) | El día de inicio de la instancia de Julián, en relación con la zona horaria del calendario.  |
| [START\_MINUTE](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html#START_MINUTE) | En el momento de inicio de la instancia de medida a partir de la medianoche, en relación con la zona horaria del calendario.  |

### Consulta la tabla de instancias

Para consultar la tabla de instancias, es necesario especificar un rango de tiempo para la consulta en el URI. En este ejemplo, [CalendarContract.Instances](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html) tiene acceso a la [TITLE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#TITLE) campo a través de su puesta en práctica de la [CalendarContract.EventsColumns](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html) interfaz. En otras palabras, [TITLE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#TITLE) se devuelve a través de una vista de base, no a través de la consulta de la prima [CalendarContract.Instances](http://developer.android.com/reference/android/provider/CalendarContract.Instances.html) de la tabla.

privada DEBUG\_TAG static final String = "MyActivity";

public String static final [] INSTANCE\_PROJECTION = new String [] {

 instancias.Event\_id, / / ​​0

 instancias.BEGIN, / / ​​1

 instancias.Cargo 2

 };

/ / Los índices de la matriz de proyección arriba.

privada PROJECTION\_ID\_INDEX static final int = 0;

privada PROJECTION\_BEGIN\_INDEX static final int = 1;

privada PROJECTION\_TITLE\_INDEX static final int = 2;

...

/ / Especifica el rango de fecha que desee para buscar recurrente

/ / Eventos casos

Calendario beginTime = Calendar.getInstance ();

beginTime.set (2011, 9, 23, 8, 0);

a largo startMillis = beginTime.getTimeInMillis ();

Calendario endTime = Calendar.getInstance ();

endTime.set (2011, 10, 24, 8, 0);

a largo endMillis = endTime.getTimeInMillis ();

Cursor act = null;

ContentResolver cr = getContentResolver ();

/ / El ID del evento recurrente cuyas instancias estás buscando

/ / En la tabla de instancias

Las instancias de selección de cadenas =.Id. del evento;

String [] selectionArgs = new String [] {"207"};

/ / Construir la consulta con el intervalo de fechas que desee.

URIConstructor constructor = instancias.CONTENT\_URI.buildUpon ();

ContentUris.appendId (constructor, startMillis);

ContentUris.appendId (constructor, endMillis);

/ / Enviar la consulta

act = cr.query (builder.build (),

 INSTANCE\_PROJECTION,

 selección,

 selectionArgs,

 nulo

mientras que (cur.moveToNext ()) {

 Cadena title = null;

 a largo IdDeSuceso = 0;

 beginVal tiempo = 0;

 / / Obtener los valores de campo

 IdDeSuceso = cur.getLong (PROJECTION\_ID\_INDEX);

 beginVal = cur.getLong (PROJECTION\_BEGIN\_INDEX);

 title = cur.getString (PROJECTION\_TITLE\_INDEX);

 / / Hacer algo con los valores.

 Log.i (DEBUG\_TAG, "Evento:" título +);

 Calendario Calendario = Calendar.getInstance ();

 calendar.setTimeInMillis (beginVal);

 DateFormat formateador = new SimpleDateFormat ("dd / MM / yyyy");

 Log.i (DEBUG\_TAG, "Fecha:" + formatter.format (Calendar.getTime ()));

 }

 }

## Calendario Intenciones

Su aplicación no necesita [permisos](http://developer.android.com/guide/topics/providers/calendar-provider.html#manifest) para leer y escribir datos del calendario. En su lugar, puede utilizar los intentos apoyados por aplicación del calendario de Android a mano de lectura y escritura a esa aplicación. La siguiente tabla muestra los intentos apoyados por el proveedor de Calendario:

|  |  |  |  |
| --- | --- | --- | --- |
| **acciones** | **URI** | **Descripción** | **Extras** |
| [VIEW](http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW)  | content://com.android.calendar/time/<ms\_since\_epoch>También puede hacer referencia a la URI con [CalendarContract.CONTENT\_URI](http://developer.android.com/reference/android/provider/CalendarContract.html#CONTENT_URI) . Para un ejemplo del uso de esta intención, consulte [Uso de las intenciones para ver los datos del calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-view) .  | Abrir calendario para el tiempo especificado por <ms\_since\_epoch> . | Ninguna |
| [VIEW](http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW)  | content://com.android.calendar/events/<event\_id>También puede hacer referencia a la URI con [Events.CONTENT\_URI](http://developer.android.com/reference/android/provider/CalendarContract.Events.html#CONTENT_URI) . Para un ejemplo del uso de esta intención, consulte [Uso de las intenciones para ver los datos del calendario](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-view) .  | Ver el evento especificado por <event\_id> . | [CalendarContract.EXTRA\_EVENT\_BEGIN\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_BEGIN_TIME)[CalendarContract.EXTRA\_EVENT\_END\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_END_TIME) |
| [EDIT](http://developer.android.com/reference/android/content/Intent.html#ACTION_EDIT)  | content://com.android.calendar/events/<event\_id>También puede hacer referencia a la URI con [Events.CONTENT\_URI](http://developer.android.com/reference/android/provider/CalendarContract.Events.html#CONTENT_URI) . Para un ejemplo del uso de esta intención, consulte [Uso de la intención de editar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-edit) .  | Edite el evento especificado por <event\_id> . | [CalendarContract.EXTRA\_EVENT\_BEGIN\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_BEGIN_TIME)[CalendarContract.EXTRA\_EVENT\_END\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_END_TIME) |
| [EDIT](http://developer.android.com/reference/android/content/Intent.html#ACTION_EDIT) [INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT)  | content://com.android.calendar/eventsTambién puede hacer referencia a la URI con [Events.CONTENT\_URI](http://developer.android.com/reference/android/provider/CalendarContract.Events.html#CONTENT_URI) . Para un ejemplo del uso de esta intención, consulte [Uso de la intención de insertar un evento](http://developer.android.com/guide/topics/providers/calendar-provider.html#intent-insert) .  | Crear un evento. | Cualquiera de los extras que figuran en la tabla siguiente. |

La siguiente tabla muestra los extras intención apoyados por el proveedor de Calendario:

|  |  |
| --- | --- |
| **Intención extra** | **Descripción** |
| [Events.TITLE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#TITLE) | Nombre del evento. |
| [CalendarContract.EXTRA\_EVENT\_BEGIN\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_BEGIN_TIME) | Evento comienza el tiempo en milisegundos de la época. |
| [CalendarContract.EXTRA\_EVENT\_END\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_END_TIME) | Hora del evento final en milisegundos de la época. |
| [CalendarContract.EXTRA\_EVENT\_ALL\_DAY](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_ALL_DAY) | Un booleano que indica que un evento es todo el día. El valor puede ser true o false . |
| [Events.EVENT\_LOCATION](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#EVENT_LOCATION) | Lugar del evento. |
| [Events.DESCRIPTION](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#DESCRIPTION) | Descripción del evento. |
| [Intent.EXTRA\_EMAIL](http://developer.android.com/reference/android/content/Intent.html#EXTRA_EMAIL) | Direcciones de correo electrónico de aquellos a invitar a una lista separada por comas. |
| [Events.RRULE](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#RRULE) | La regla de repetición para el evento. |
| [Events.ACCESS\_LEVEL](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#ACCESS_LEVEL) | Si el evento es privado o público. |
| [Events.AVAILABILITY](http://developer.android.com/reference/android/provider/CalendarContract.EventsColumns.html#AVAILABILITY) | Si este evento cuenta con el tiempo ocupado o es el tiempo libre que se pueden programar más. |

Las secciones siguientes describen cómo utilizar estas intenciones.

### El uso de un intento de insertar un evento

Usando el [INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) Intención le permite a su lado la aplicación de la tarea de la inserción de eventos para el propio calendario. Con este enfoque, la aplicación no necesita ni siquiera tener la [WRITE\_CALENDAR](http://developer.android.com/reference/android/Manifest.permission.html#WRITE_CALENDAR) permiso incluido en su [archivo de manifiesto](http://developer.android.com/guide/topics/providers/calendar-provider.html#manifest) .

Cuando los usuarios ejecutan una aplicación que utiliza este enfoque, la aplicación les envía a la Agenda para terminar de agregar el evento. El [INSERT](http://developer.android.com/reference/android/content/Intent.html#ACTION_INSERT) Intención utiliza campos adicionales para rellenar previamente un formulario con los detalles del evento en el calendario. Los usuarios pueden cancelar el evento, modificar la forma según sea necesario, o guardar el evento en sus calendarios.

Aquí hay un fragmento de código que los horarios de un evento el 19 de enero de 2012, que se extiende desde las 7:30 am hasta las 8:30 am en cuenta lo siguiente acerca de este fragmento de código:

* En él se especifica [Events.CONTENT\_URI](http://developer.android.com/reference/android/provider/CalendarContract.Events.html#CONTENT_URI) como Uri.
* Utiliza el [CalendarContract.EXTRA\_EVENT\_BEGIN\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_BEGIN_TIME) y [CalendarContract.EXTRA\_EVENT\_END\_TIME](http://developer.android.com/reference/android/provider/CalendarContract.html#EXTRA_EVENT_END_TIME) campos adicionales para rellenar previamente el formulario con la hora del evento. Los valores de estos tiempos debe ser en milisegundos de UTC de la época.
* Se utiliza la [Intent.EXTRA\_EMAIL](http://developer.android.com/reference/android/content/Intent.html#EXTRA_EMAIL) extra de campo para proporcionar una lista separada por comas de los invitados, especificado por la dirección de correo electrónico.

Calendario beginTime = Calendar.getInstance ();

beginTime.set (2012, 0, 19, 7, 30);

Calendario endTime = Calendar.getInstance ();

endTime.set (2012, 0, 19, 8, 30);

Intención de Intención = nuevo intento (Intent.ACTION\_INSERT)

 . SetData (Events.CONTENT\_URI)

 . PutExtra (CalendarContract.EXTRA\_EVENT\_BEGIN\_TIME, beginTime.getTimeInMillis ())

 . PutExtra (CalendarContract.EXTRA\_EVENT\_END\_TIME, endTime.getTimeInMillis ())

 . PutExtra (Events.TÍTULO: "Yoga")

 . PutExtra (Events.DESCRIPCIÓN: "Grupo de clase")

 . PutExtra (Events.EVENT\_LOCATION, "El gimnasio")

 . PutExtra (Events.DISPONIBILIDAD, Eventos.AVAILABILITY\_BUSY)

 . PutExtra (Intent.EXTRA\_EMAIL, "rowan@example.com, trevor@example.com");

startActivity (intención);

### El uso de un intento de editar un evento

Puede actualizar un evento directamente, como se describe en [Actualización de las citas](http://developer.android.com/guide/topics/providers/calendar-provider.html#update-event) . Pero con el [EDIT](http://developer.android.com/reference/android/content/Intent.html#ACTION_EDIT) Intención permite una aplicación que no tiene permiso para ceder caso de edición para la aplicación Calendario. Cuando termine de editar los usuarios de su evento en el calendario, que regresará a la solicitud original.

He aquí un ejemplo de un intento de que establece un nuevo título para un evento especificado y permite a los usuarios editar el evento en el calendario.

a largo IdDeSuceso = 208;

Uri uri = ContentUris.withAppendedId (Events.CONTENT\_URI, IdDeSuceso);

Intención de Intención = nuevo intento (Intent.ACTION\_EDIT)

 . SetData (URI)

 . PutExtra (Events.TÍTULO: "Mi nuevo título");

startActivity (intención);

### Uso de las intenciones para ver los datos del calendario

Proveedor de calendario ofrece dos formas diferentes de usar el [VIEW](http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW) Intención:

* Para abrir el calendario para una fecha determinada.
* Para ver un evento.

Aquí hay un ejemplo que muestra cómo abrir el calendario para una fecha determinada:

/ / A. De fecha y hora especificada en milisegundos desde la época

a largo startMillis;

...

URIConstructor constructor = CalendarContract.CONTENT\_URI.buildUpon ();

builder.appendPath ("tiempo");

ContentUris.appendId (constructor, startMillis);

Intención de Intención = nuevo intento (Intent.ACTION\_VIEW)

 . SetData (builder.build ());

startActivity (intención);

Aquí hay un ejemplo que muestra cómo abrir un evento para ver:

a largo IdDeSuceso = 208;

...

Uri uri = ContentUris.withAppendedId (Events.CONTENT\_URI, IdDeSuceso);

Intención de Intención = nuevo intento (Intent.ACTION\_VIEW)

 . SetData (URI);

startActivity (intención);

## Adaptadores de sincronización

Sólo hay pequeñas diferencias en la forma de una aplicación de sincronización y un adaptador de acceso al proveedor de Calendario:

* Un adaptador de sincronización debe especificar que se trata de un adaptador de sincronización mediante el establecimiento de [CALLER\_IS\_SYNCADAPTER](http://developer.android.com/reference/android/provider/CalendarContract.html#CALLER_IS_SYNCADAPTER) de true .
* Un adaptador de sincronización debe proporcionar una [ACCOUNT\_NAME](http://developer.android.com/reference/android/provider/CalendarContract.SyncColumns.html#ACCOUNT_NAME) y un [ACCOUNT\_TYPE](http://developer.android.com/reference/android/provider/CalendarContract.SyncColumns.html#ACCOUNT_TYPE) como parámetros de consulta en el URI.
* Un adaptador de sincronización tiene acceso de escritura a más columnas que una aplicación o widget. Por ejemplo, una aplicación sólo puede modificar algunas características de un calendario, como su nombre, nombre de pantalla, configuración de visibilidad, y si el calendario sincronizado está. En comparación, un adaptador de sincronización se puede acceder no sólo a las columnas, pero muchos otros, como el color del calendario, la zona horaria, el nivel de acceso, ubicación, y así sucesivamente. Sin embargo, un adaptador de sincronización está limitada a la ACCOUNT\_NAME y ACCOUNT\_TYPE lo especificado.

Aquí es un método auxiliar que puede utilizar para devolver un URI para su uso con un adaptador de sincronización:

 asSyncAdapter estáticas (Uri Uri Uri, cuenta cadena, accountType String) {

 volver uri.buildUpon ()

 . AppendQueryParameter (android.provider.CalendarContract.CALLER\_IS\_SYNCADAPTER, "true")

 . AppendQueryParameter (Calendars.Nombre\_cuenta, cuenta)

 . AppendQueryParameter (Calendars.. ACCOUNT\_TYPE, accountType) build ();

 }

Para un ejemplo de implementación de un adaptador de sincronización (no específicamente relacionadas con calendario), ver [SampleSyncAdapter](http://developer.android.com/resources/samples/SampleSyncAdapter/index.html) .