	<b>UNIVERSIDAD DON BOSCO</b> <b>FACULTAD DE INGENIERÍA</b> <b>ESCUELA DE COMPUTACIÓN</b>	
<b>CICLO:</b> <b>I/2021</b>	<b>GUIA DE LABORATORIO #12</b>	
	<b>Nombre de la Práctica:</b>	Content Providers
	<b>MATERIA:</b>	Desarrollo de Software para Móviles

## I. OBJETIVOS

1. Conocer qué es un content provider.
2. Crear un content provider en una aplicación Android.
3. Consumir un content provider en una aplicación Android.

## II. INTRODUCCION TEORICA

Un proveedor de contenido (content provider) administra el acceso a un repositorio central de datos. Un proveedor forma parte de una aplicación para Android y a menudo proporciona su propia IU para trabajar con los datos. No obstante, los proveedores de contenido están principalmente orientados a que los usen otras aplicaciones que acceden al proveedor usando un objeto de cliente del proveedor. Juntos, los proveedores y clientes de proveedores ofrecen una interfaz estándar y uniforme para los datos que también manipula la comunicación dentro del proceso y el acceso seguro a los datos.

Por lo general, son dos las situaciones en las que se trabaja con proveedores de contenido: cuando quieres implementar código para acceder a un proveedor de contenido existente en otra aplicación o cuando decides crear un proveedor de contenido nuevo en tu aplicación a fin de compartir datos con otras aplicaciones.

Un proveedor de contenido presenta datos a aplicaciones externas en forma de una o más tablas que son similares a las tablas de una base de datos relacional. Una fila representa una instancia de algún tipo de datos que recopila el proveedor, y cada columna de la fila representa un ítem individual de los datos recopilados para una instancia.

El proveedor de contenido organiza el acceso a la capa de almacenamiento de los datos en tu aplicación para una serie de API y componentes diferentes (como se detalla en la figura 1) e incluye lo siguiente:

- Compartir con otras aplicaciones el acceso a los datos de tu aplicación.
- Enviar datos a un widget.
- Mostrar sugerencias personalizadas de búsqueda para tu aplicación mediante el marco de trabajo de búsqueda usando SearchRecentSuggestionsProvider.

- Sincronizar los datos de la aplicación con tu servidor mediante una implementación de `AbstractThreadedSyncAdapter`.
- Cargar datos en tu IU usando `CursorLoader`.

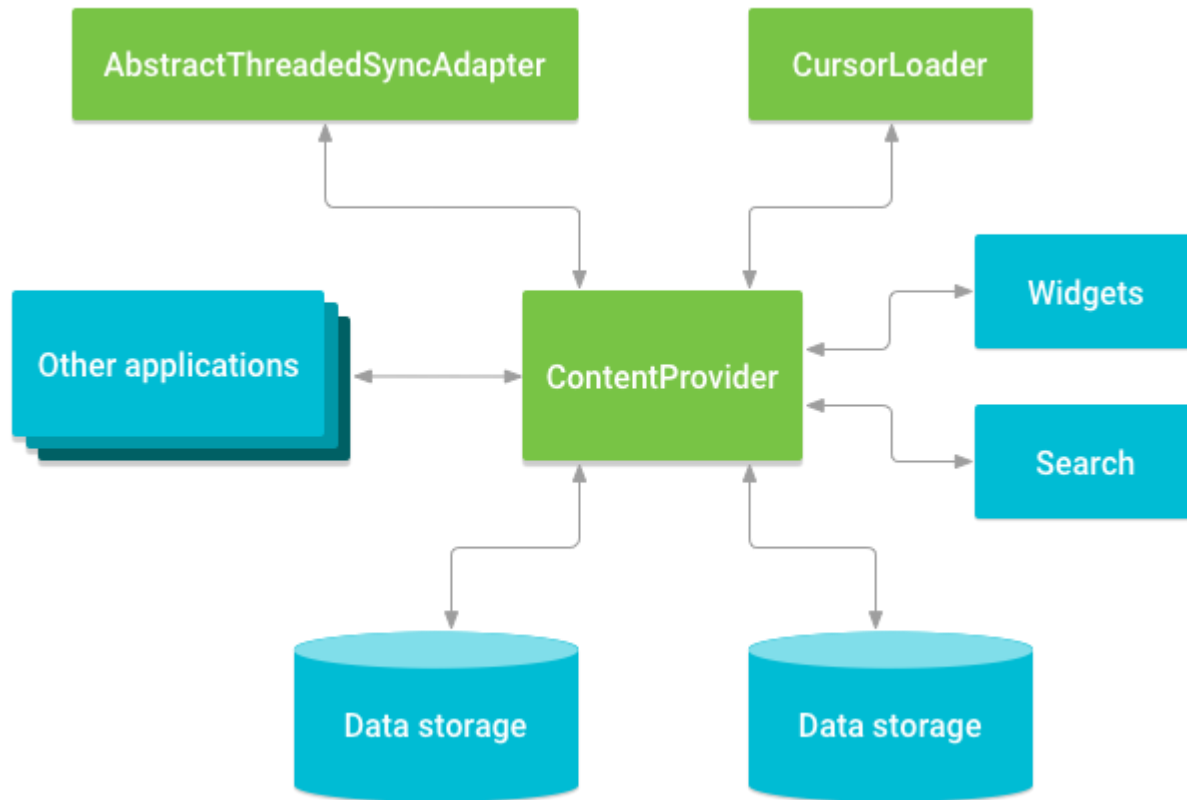


Figura 1: Relaciones de un content provider.

### III. PROCEDIMIENTO

En esta ocasión se desarrollará una aplicación, donde se creará un content provider personalizado (con datos de prueba para efectos de la práctica) y se consultarán usando las apis de Android: `Cursor` y `LoaderManager`.

1. Ejecuta Android Studio y crea un nuevo proyecto. En **Project Template** selecciona **Empty Activity** y presiona **Next**.
2. Como **Name** coloca **Guia12AppContentProvider**, en **Package name** escribe **sv.edu.udb.guia12app.content.provider** en **Save location** escoge la carpeta de tu preferencia, en Language elige **Java** y el **Minimum SDK** selecciona **API 16: Android 4.1 (Jelly Bean)**

Nota: En esta práctica se hará uso del lenguaje de programación Java pero, se invita al estudiante a realizarla también utilizando Kotlin.

3. Presiona **Finish**.

4. Crea una nueva clase en Android Studio y nómbrala “**StudentsContract**”. Su objetivo es contener el nombre de las tablas, los nombres de las columnas, las URI de contenido y los tipos MIME de cada dato.

[Código Fuente](#)

5. Crea una nueva clase y nómbrala “**DatabaseHelper**” esta clase servirá para crear la base de datos que usará el content provider, extenderá de “**SQLiteOpenHelper**” una clase utilitaria que permite a las aplicaciones Android gestionar bases de datos SQLite de forma fácil.

[Código Fuente](#)

6. Crea una nueva clase y llámala “**StudentsContentProvider**”, esta es la clase principal donde estará el content provider personalizado.

[Código Fuente](#)

7. Registra el content provider en tu **AndroidManifest.xml**.

[Código Fuente](#)

Ahora leeremos estos datos y los mostraremos en pantalla.

8. Crea un layout personalizado (**item.xml**) este servirá para mostrar cada fila de los datos registrados.

[Código Fuente](#)

9. Crea una clase “**StudentsAdapter**”, ésta será un Adapter para mostrar los datos guardados en una ListView.

[Código Fuente](#)

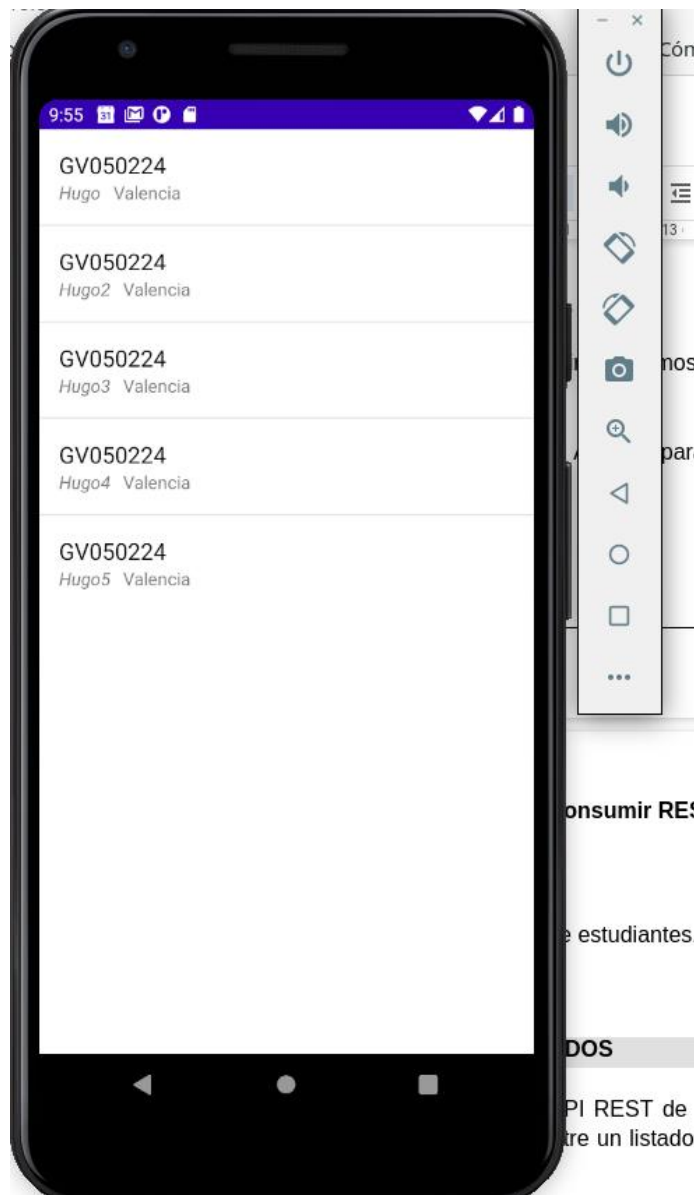
10. Modifica tu clase **MainActivity**.

[Código Fuente](#)

11. Modifica el layout de **MainActivity**.

[Código Fuente](#)

12. Ejecuta la aplicación, en pantalla verás un listado de estudiantes.



#### IV. DISCUSIÓN DE RESULTADOS

1. Complete la aplicación Android de la guía, agregando una Activity que muestre un formulario (carnet, nombre y apellido) que permita agregar nuevos registros a la base de datos del content provider.

#### V. BIBLIOGRAFÍA

- Documentación Oficial Android  
<https://developer.android.com/guide/topics/providers/content-provider-basics?hl=es-419>