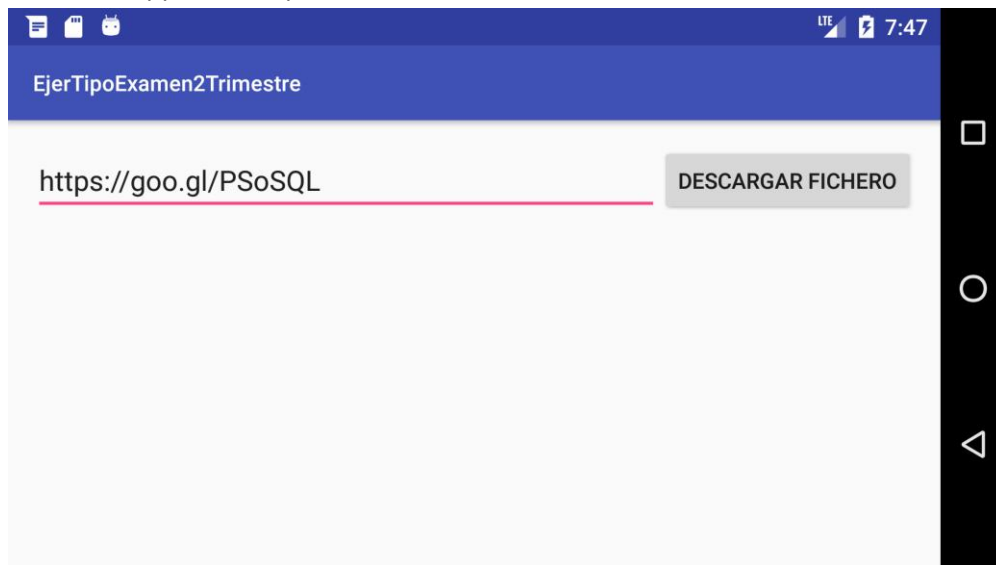


- 1) Crear una app con una pantalla inicial como esta:

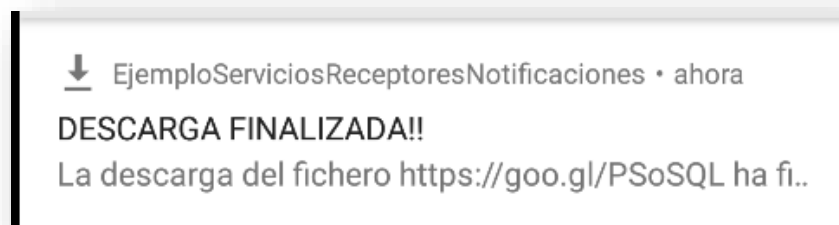


En el EditText aparecerá una URL que se toma de las preferencias y corresponde a la url de un fichero de texto en formato csv. Es esta url: “**https://goo.gl/PSoSQL**”. Ese fichero tiene el siguiente formato:

Nombre\_alumno, nombre\_asignatura, nota, profesor

Cuando pulsamos el botón descargar fichero, **arranca un Servicio en un hilo diferente del UI (con IntentService) que nos descarga el fichero.**

- Cuando la descarga termina, el servicio avisa a la Activity mediante un anuncio de FIN DE DESCARGA y le pasa los datos que contenía el fichero.
  - Si hay problemas de conexión lanza un anuncio de PROBLEMAS CONEXIÓN.
  - Si hay problemas leyendo el fichero lanza un anuncio de PROBLEMAS I/O.
- Justo cuando el receptor de anuncios se entera que la descarga ha finalizado, lanza una **NOTIFICACIÓN** que informa que la descarga ya finalizó y por tanto, el fichero está listo para usarse.



Dicha notificación incluye:

- Una imagen de fin de descarga:  
`android.R.drawable.stat_sys_download_done`

## PMDM

### EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

- Título: "DESCARGA FINALIZADA!!"
- Hora de la notificación
- Texto: "La descarga del fichero .... (poner ahí la url) ha finalizado. Pulse para verlo".

Si se pulsa sobre dicha notificación, se muestra una activity como la siguiente, donde se ve el contenido de ese fichero en un TextView de varias líneas y sustituyendo las "," por espacios en blanco.



## SOLUCIÓN:

**PASO 1:** Diseñamos la pantalla principal y programamos el botón DESCARGAR que nos descargue el fichero (controlar errores de URL mal escrita o problemas de I/O) y nos muestre un mensaje en consola avisando de la descarga y del contenido descargado.

### DISEÑO LAYOUT activity\_main.xml:

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <EditText
        android:id="@+id/etUrlFichero"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:hint="@string/hint_url"
        android:inputType="textUri|textMultiLine"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btDescargar"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="8dp"
        android:text="@string/btDescargar_texto"
        app:layout_constraintBottom_toBottomOf="@+id/etUrlFichero"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/etUrlFichero"
        app:layout_constraintTop_toTopOf="@+id/etUrlFichero" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Clase MainActivity.java:

```
public class MainActivity extends AppCompatActivity {

    // UI
    private EditText etUrlDescarga;
    private Button btDescargar;

    //region MÉTODOS CICLO VIDA ACTIVITY
```

## PMDM

### EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    initUI();
}

//endregion

//region MÉTODOS PROPIOS

/** Método que inicializa la UI
 *
 */

private void initUI(){
    etUrlDescarga=findViewById(R.id.etUrlFichero);
    btDescargar=findViewById(R.id.btDescargar);
    btDescargar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(!TextUtils.isEmpty(etUrlDescarga.getText())){
                lanzarServicio(etUrlDescarga.getText().toString());
            }
        }
    });
}

//endregion
}

```

Pero, ¿qué Código añadido en “lanzarSercicio()”? ¿Cómo se lanza un servicio? RESPUESTA: simplemente se crea un intent que arranque el servicio; el cual será una clase Java que temenos que crear y que extiende de IntentService.

Nuestro MainActivity queda entonces:

```

public class MainActivity extends AppCompatActivity {

    public static final String EXTRA_URI_CONEXION = "uri_descarga";
    public static final String TAG_APLI=MainActivity.class.getSimpleName();

    // UI
    private EditText etUrlDescarga;
    private Button btDescargar;

    //region MÉTODOS CICLO VIDA ACTIVITY

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        initUI();
    }

    //endregion
}

```

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
//region MÉTODOS PROPIOS

/** Método que inicializa la UI
 *
 */
private void initUI() {
    etUrlDescarga=findViewById(R.id.etUrlFichero);
    btDescargar=findViewById(R.id.btDescargar);
    btDescargar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(!TextUtils.isEmpty(etUrlDescarga.getText())){
                lanzarServicio(etUrlDescarga.getText().toString());
            }
        }
    });
}

/**
 * Método que lanza el servicio que descarga el CSV de la url que se le pasa
 *
 * @param urlDescarga url en formato string donde se encuentra el fichero a
 * descargar
 */
private void lanzarServicio(String urlDescarga){
    //arranco el servicio, pasándole la url de descarga
    Intent svc=new Intent(this, ServicioLeerArchivoCSV.class);
    svc.putExtra(EXTRA_URI_CONEXION,urlDescarga);
    startService(svc);
}
//endregion
}
```

Ahora tenemos que crear esa clase “ServicioLeerArchivoCSV.java” que será el servicio:

Clase ServicioLeerArchivoCSV.java: es el servicio que descarga datos

```
public class ServicioLeerArchivoCSV extends IntentService {
    public static final String EXTRA_DATOS_CARGADOS = "datos_cargados";
    public static final String ACTION_FIN_CARGA_DATOS =
        "es.alejandra.ejemploserviciosreceptoresnotificaciones.action.FIN_CARGA_DATOS";
    public static final String
        ACTION_ERROR_URL="es.alejandra.ejemploserviciosreceptoresnotificaciones.action.ERROR_URL";
    public static final String
        ACTION_ERROR_IO="es.alejandra.ejemploserviciosreceptoresnotificaciones.action.ERROR_IO";

    // URL desde donde descarga
    private URL url;

    // String[] donde guarda los datos descargados
    private String[] notaAlumnoAsignatura;
```

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
public ServicioLeerArchivoCSV() {
    super("ServicioLeerArchivoCSV");
}

@Override
protected void onHandleIntent(@Nullable Intent intent) {
    if(intent!=null){
        if(intent.hasExtra(MainActivity.EXTRA_URI_CONEXION)){
            //recogo la uri para descargar el archivo
            Uri miUri=
            Uri.parse(intent.getStringExtra(MainActivity.EXTRA_URI_CONEXION));
            //leo el fichero
            try{
                url=new URL(miUri.toString());
                notaAlumnoAsignatura=
                    NetworkUtilities.getResponseFromHttpUrl(url);
                avisarEventoOcurrido(ACTION_FIN_CARGA_DATOS);

            } catch (MalformedURLException e) {
                avisarEventoOcurrido(ACTION_ERROR_URL);
            } catch (IOException e){
                e.printStackTrace();
                avisarEventoOcurrido(ACTION_ERROR_IO);
            }
        }
    }
}

/**
 * Método que lanza un aviso de un evento ocurrido.
 * El evento se le pasa por parámetro
 */
private void avisarEventoOcurrido(String action) {
    Intent iAviso = new Intent(action);
    if (action == ACTION_FIN_CARGA_DATOS) {
        iAviso.putExtra(EXTRA_DATOS_CARGADOS, notaAlumnoAsignatura);
    }
    //EN LOS DEMÁS CASOS, NO LLEVA EXTRAS, POR ESO NO SE AÑADEN
    sendBroadcast(iAviso);

    Log.d("MIAPLI",action);
}
}
```

Acordarse de pedir en el manifest, el permis de Internet:

```
<uses-permission android:name="android.permission.INTERNET" />
```

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

**PASO 2:** Quitamos los mensajes de log que pusimos en el servicio, si queremos, y en el MAINACTIVITY creamos un **RECEPTOR DE ANUNCIOS** que recoja los anuncios que envíe el Servicio. Para ello:

- Definimos una clase `ReceptorAvisosServicio`
- Registramos el receptor de anuncios en el `onResume()`
- Lo desregistramos en el `onPause()`.

Nos queda el `MainActivity`:

```
public class MainActivity extends AppCompatActivity {
    public static final String EXTRA_URI_CONEXION = "uri_descarga";
    public static final String TAG_APLI = MainActivity.class.getSimpleName();

    //UI
    private EditText etUrlDescarga;
    private Button btDescargar;

    //RECEPTOR AVISOS SERVICIO
    private ReceptorAvisosServicio receptorAvisosServicio;

    //GESTION DATOS FICHERO
    String[] datosAlumnosNotas;

    ////////////////////////////////////// MÉTODOS CICLO VIDA ACTIVITY
    //////////////////////////////////////

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        initUI();
    }

    @Override
    protected void onResume() {
        super.onResume();

        //Registro el receptor de Broadcast (BroadcastReceiver) para que la
        //activity escuche los eventos que le envía
        //nuestro servicio
        receptorAvisosServicio = new ReceptorAvisosServicio();
        IntentFilter intentFilter = new IntentFilter();
        intentFilter.addAction(ServicioLeerArchivoCSV.ACTION_FIN_CARGA_DATOS);
        intentFilter.addAction(ServicioLeerArchivoCSV.ACTION_ERROR_IO);
        intentFilter.addAction(ServicioLeerArchivoCSV.ACTION_ERROR_URL);
        registerReceiver(receptorAvisosServicio, intentFilter);
    }

    @Override
    protected void onPause() {
        super.onPause();
        //desregistro el BroadcastReceiver
        unregisterReceiver(receptorAvisosServicio);
    }
}
```

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////// MÉTODOS PROPIOS
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

/**
 * Método que inicializa la UI
 */
private void initUI() {
    etUrlDescarga = (EditText) findViewById(R.id.etUrlFichero);
    btDescargar = (Button) findViewById(R.id.btDescargar);
    btDescargar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (!TextUtils.isEmpty(etUrlDescarga.getText())) {
                lanzarServicio(etUrlDescarga.getText().toString());
            }
        }
    });
}

/**
 * Método que lanza el servicio que descarga el CSV de la url que se le pasa
 *
 * @param urlDescarga url en formato string donde se encuentra el fichero a
 * descargar
 */
private void lanzarServicio(String urlDescarga) {
    //arranco el servicio
    Intent svc = new Intent(this, ServicioLeerArchivoCSV.class);
    svc.putExtra(EXTRA_URI_CONEXION, urlDescarga);
    startService(svc);
}

/**
 * Método que muestra en el LogCat los datos que se le pasan
 *
 * @param datosAlumnosNotas String[] a mostrar en el log
 */
private void mostrar_en_log(String[] datosAlumnosNotas) {
    for (String asignaturaAlumno : datosAlumnosNotas) {
```



PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
        Log.d(TAG_APLI, asignaturaAlumno);  
    }  
  
}
```

```
////////////////////////////////////  
//////////////////////////////////// RECEPTOR AVISOS SERVICIO  
////////////////////////////////////
```

```
////////////////////////////////////  
////////////////////////////////////  
public class ReceptorAvisosServicio extends BroadcastReceiver {  
    public ReceptorAvisosServicio() {  
    }  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        switch (intent.getAction()) {  
            case ServicioLeerArchivoCSV.ACTION_FIN_CARGA_DATOS:  
                datosAlumnosNotas =  
intent.getStringArrayExtra(ServicioLeerArchivoCSV.EXTRA_DATOS_CARGADOS);  
                mostrar_en_log(datosAlumnosNotas);  
                Toast.makeText(context, "FIN DESCARGA",  
Toast.LENGTH_SHORT).show();  
                return;  
            case ServicioLeerArchivoCSV.ACTION_ERROR_URL:  
                Toast.makeText(context, "URL INCORRECTA",  
Toast.LENGTH_SHORT).show();  
                return;  
            case ServicioLeerArchivoCSV.ACTION_ERROR_IO:  
                Toast.makeText(context, "ERROR DE LECTURA EN EL FICHERO",  
Toast.LENGTH_SHORT).show();  
                return;  
        }  
    }  
  
}
```

Si lo ejecutamos ahora, deberíamos ver en el LogCat el contenido del fichero.

**PASO 3:** Programo la notificación que se debe mostrar cuando finaliza la descarga **dentro del Receptor de anuncios**, puesto que dice que cuando éste se entera de que la descarga finalizó, la lanza. **De momento no sucederá nada al pulsar sobre la notificación. Simplemente la veré en la barra de estado y la puedo desplegar.** Por tanto, dentro del Receptor de anuncios añadido:

La clase `ReceptorAvisosServicio.java` queda así:

```
public class ReceptorAvisosServicio extends BroadcastReceiver {
    private NotificationManager notificationManager;
    protected static final String CANAL_ID = "mi_canal";
    protected static final int NOTIFICACION_ID = 1;

    public ReceptorAvisosServicio() {
    }

    @Override
    public void onReceive(Context context, Intent intent) {

        switch (intent.getAction()) {
            case ServicioLeerArchivoCSV.ACTION_FIN_CARGA_DATOS:
                datosAlumnosNotas =
intent.getStringArrayExtra(ServicioLeerArchivoCSV.EXTRA_DATOS_CARGADOS
);
                mostrar_en_log(datosAlumnosNotas);
                Toast.makeText(context, "FIN DESCARGA",
Toast.LENGTH_SHORT).show();
                //lanzo la notificación
                lanzarNotificacion(datosAlumnosNotas);
                return;
            case ServicioLeerArchivoCSV.ACTION_ERROR_URL:
                Toast.makeText(context, "URL INCORRECTA",
Toast.LENGTH_SHORT).show();
                return;
            case ServicioLeerArchivoCSV.ACTION_ERROR_IO:
                Toast.makeText(context, "ERROR DE LECTURA EN EL
FICHERO", Toast.LENGTH_SHORT).show();
                return;
        }
    }

    private void lanzarNotificacion(String[] datos) {

        notificationManager = (NotificationManager)
getService(NOTIFICATION_SERVICE);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            // Oreo
            //creo un canal de notificaciones
            NotificationChannel notificationChannel = new
NotificationChannel(CANAL_ID, "Mis notificaciones",
NotificationManager.IMPORTANCE_DEFAULT);
            notificationChannel.setDescription("Es el canal
empleado para notificar el servicio de descarga");

notificationManager.createNotificationChannel(notificationChannel);
        }
        //creo la notificación
        NotificationCompat.Builder notificacion =
new
NotificationCompat.Builder(getApplicationContext(), CANAL_ID)
```

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
.setSmallIcon(android.R.drawable.stat_sys_download_done)
    .setContentTitle("DESCARGA FINALIZADA!!")

    .setWhen(Calendar.getInstance().getTimeInMillis())
        .setAutoCancel(true)
        .setContentText("La descarga del fichero "
+ etUrlDescarga.getText().toString() + " ha finalizado. Pulse para
verlo.");
    notificationManager.notify(NOTIFICACION_ID,
notification.build());
    }
}
```

**PASO 6:** Añado a la notificación que acabamos de hacer que cuando la pulsemos, nos lleve a otra activity que muestre en un TextView el contenido del fichero descargado cambiando las “,” por tabuladores. Para ello, primero necesito crear la otra activity que llamaré: **MostrarFicheroActivity:**

Su **layout** será:

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MostrarFicheroActivity">

    <TextView
        android:id="@+id/tvContenidoFichero"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Y su java **MostrarFicheroActivity.java** de **momento no lo tocamos**.

Vamos a configurar en el Receptor de anuncios la notificación que hicimos en el punto anterior, para que al pulsar sobre ella, nos lance la activity: “MostrarFicheroActivity”. Para ello en el método “lanzarNotificacion” que hay en el receptor de anuncios del MainActivity, añadimos:

## PMDM

### EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
public class ReceptorAvisosServicio extends BroadcastReceiver {
    private NotificationManager notificationManager;
    protected static final String CANAL_ID = "mi_canal";
    protected static final int NOTIFICACION_ID = 1;
    protected static final int COD_REQUEST_PENDING_INTENT = 0;
    protected static final String EXTRA_KEY_DATOS_FICHERO =
"extra_key_fichero";

    public ReceptorAvisosServicio() {
    }

    @Override
    public void onReceive(Context context, Intent intent) {

        switch (intent.getAction()) {
            case ServicioLeerArchivoCSV.ACTION_FIN_CARGA_DATOS:
                datosAlumnosNotas =
intent.getStringArrayExtra(ServicioLeerArchivoCSV.EXTRA_DATOS_CARGADOS);
                mostrar_en_log(datosAlumnosNotas);
                Toast.makeText(context, "FIN DESCARGA",
Toast.LENGTH_SHORT).show();
                //lanzo la notificación
                lanzarNotificacion(datosAlumnosNotas);
                return;
            case ServicioLeerArchivoCSV.ACTION_ERROR_URL:
                Toast.makeText(context, "URL INCORRECTA",
Toast.LENGTH_SHORT).show();
                return;
            case ServicioLeerArchivoCSV.ACTION_ERROR_IO:
                Toast.makeText(context, "ERROR DE LECTURA EN EL FICHERO",
Toast.LENGTH_SHORT).show();
                return;
        }

    }

    private void lanzarNotificacion(String[] datos) {

        notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            // Oreo
            //creo un canal de notificaciones
            NotificationChannel notificationChannel = new
NotificationChannel(CANAL_ID, "Mis notificaciones",
NotificationManager.IMPORTANCE_DEFAULT);
            notificationChannel.setDescription("Es el canal empleado para
notificar el servicio de descarga");

            notificationManager.createNotificationChannel(notificationChannel);
        }
        //creo el PendingIntent que lanzará la segunda activity al pulsar
sobre la notificación
        Intent iMostrarFichero = new Intent(getApplicationContext(),
MostrarFicheroActivity.class);
        iMostrarFichero.putExtra(EXTRA_KEY_DATOS_FICHERO, datos);

        PendingIntent intentPendiente =
```

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
PendingIntent.getActivity(MainActivity.this,
                          COD_REQUEST_PENDING_INTENT, iMostrarFichero, 0); //flags=0
es sin flags

//creo la notificación
NotificationCompat.Builder notificacion =
    new NotificationCompat.Builder(getApplicationContext(),
CANAL_ID)

.setSmallIcon(android.R.drawable.stat_sys_download_done)
.setContentTitle("DESCARGA FINALIZADA!!")
.setWhen(Calendar.getInstance().getTimeInMillis())
.setAutoCancel(true)
.setContentText("La descarga del fichero " +
etUrlDescarga.getText().toString() + " ha finalizado. Pulse para verlo.")
.setContentIntent(intentPendiente);

notificationManager.notify(NOTIFICACION_ID, notificacion.build());
}
}
```

Ahora ya podemos modificar el java `MostrarFicheroActivity.java` para que recoja el `String[]` que le estamos pasando con el intent del `PendingIntent` en la notificación y lo muestre en el `TextView`. Se hace como si fuese enviado con un intent normal, es decir, con `getIntent().getExtras()`.....

Nos quedará:

```
public class MostrarFicheroActivity extends AppCompatActivity {
    private String[] datosFichero;
    private TextView textoFichero;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mostrar_fichero);

        textoFichero = findViewById(R.id.tvContenidoFichero);

        if (getIntent().
            .hasExtra(MainActivity.ReceptorAvisosServicio.EXTRA_KEY_DATOS_FICHERO))
        {
            datosFichero =
getIntent().getStringArrayExtra(MainActivity.ReceptorAvisosServicio.EXTRA_KEY_DAT
OS_FICHERO);
        }

        if (datosFichero != null && datosFichero.length > 0) {
            for (String linea :
                datosFichero) {
                String[] lineaTroceada = linea.split(",");
                textoFichero.append(String.format("%-16s", lineaTroceada[0]) +
                    String.format("%-28s", lineaTroceada[1]) +
                    String.format("%-15s", lineaTroceada[2]) +
                    String.format("%10s", lineaTroceada[3]) + "\n"); // %-10s
            }
        }
    }
}
```

el "-" indica que rellene por la derecha espacios blancos

PMDM  
EJEMPLO SERVICIOS, NOTIFICACIONES Y RECEPTORES DE ANUNCIOS

```
    }
```

```
  }
```

```
}
```

```
}
```