Curso Android Avanzado

Altice Hispaniola, S.A.

Vin Liang Rosa Chea

Agenda

Día 1:
- Práctica de Repaso
- Estados del Activity
- Fragments

Día 2:
- RecyclerViews
- Support Libraries
- Entrega de Requerimientos Proyecto Final

Día 3:
- AsyncTask, Handler y Loopers.
- Services
- NotificationManager

Día 4:
- Networking, REST Service
- Persistencia

Día 5:
- Content Providers
- LocationManager
- BroadcastReceiver

Día 6:
- Revisión - Proyecto Final

altice

Día 1:

- Práctica
- Estados del Activity
- Fragments

altice

Practica Repaso

Crear RSS Reader:

- **Mostrar Listado/Opciones para navegar a los siguientes Feeds (Ejemplos)**
  - **https://www.diariolibre.com/rss/portada.xml**
  - **http://www.listindiario.com/rss/portada/**
  - **http://elnacional.com.do/feed/**
- **Mostrar Listado de Items for Feed**
- **Mostrar Contenido en WebView o Abrir en Navegador del Sistema.**

Utilizar:

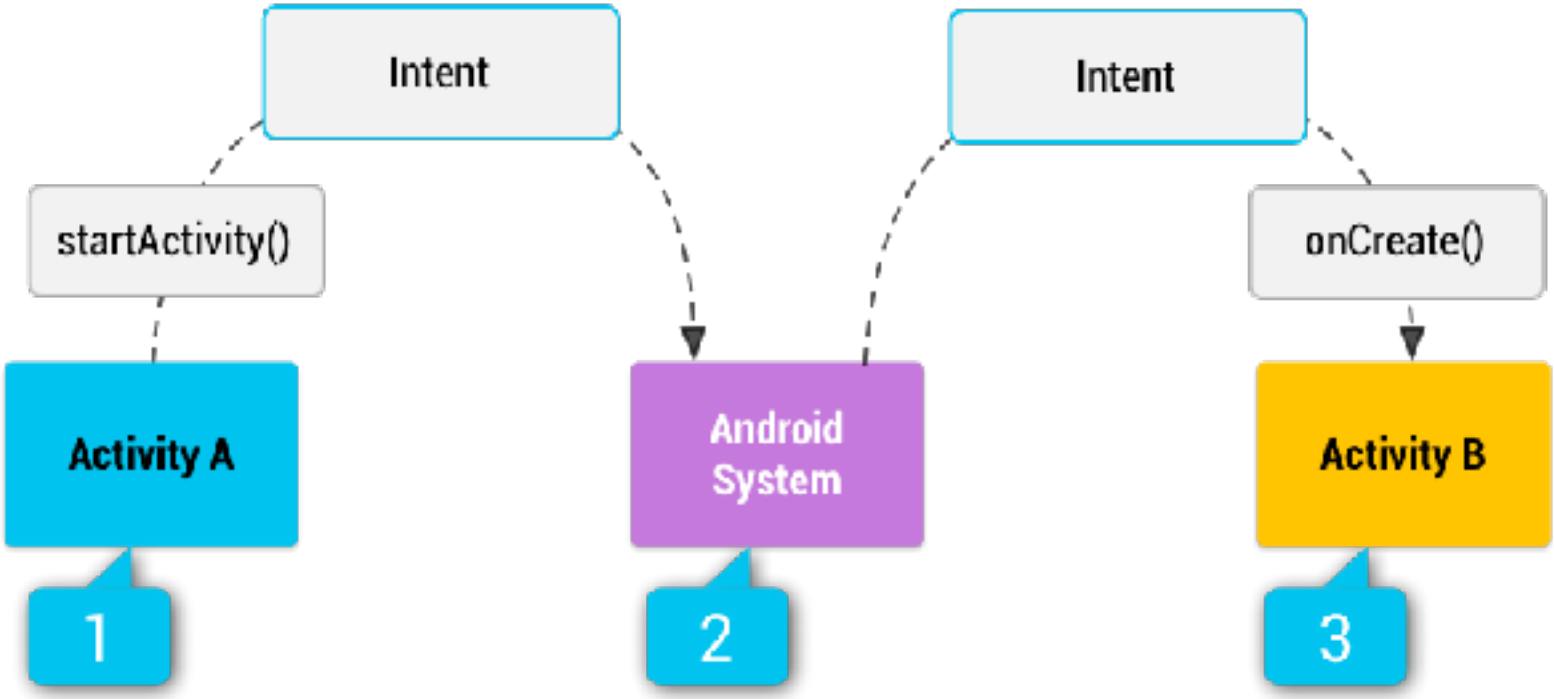
- **Aprendido del Curso Basico**
- **Dependencia**
  - `compile 'com.prof.rssparser:rssparser:1.0'`

com.prof.rssparser:rssparser:1.0

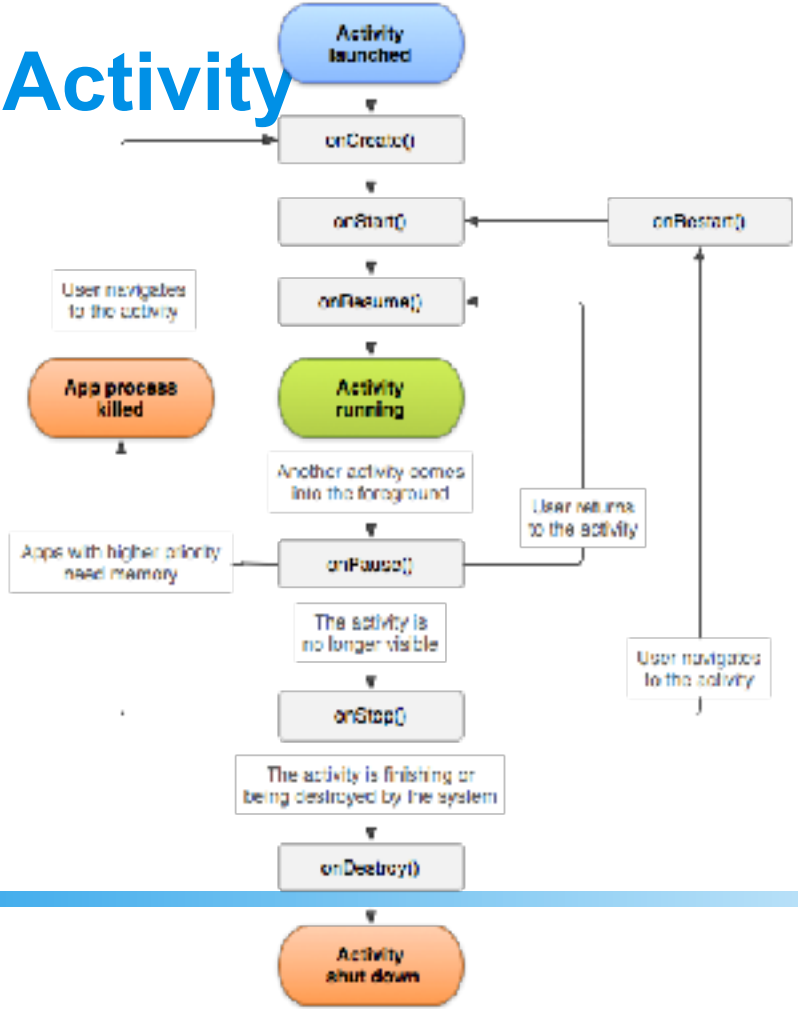
```
import com.prof.rssparser.Article;
import com.prof.rssparser.Parser;

String url = getIntent().getExtras().getString(FEED_URL);
Parser parser = new Parser();
parser.onFinish(new Parser.OnTaskCompleted() {
    @Override
    public void onTaskCompleted(ArrayList<Article> list) {
        feedListView.setAdapter(new FeedAdapter(list));
    }

    @Override
    public void onError() {

    }
});
parser.execute(url);
```

Estados del Activity

Estados del Activity

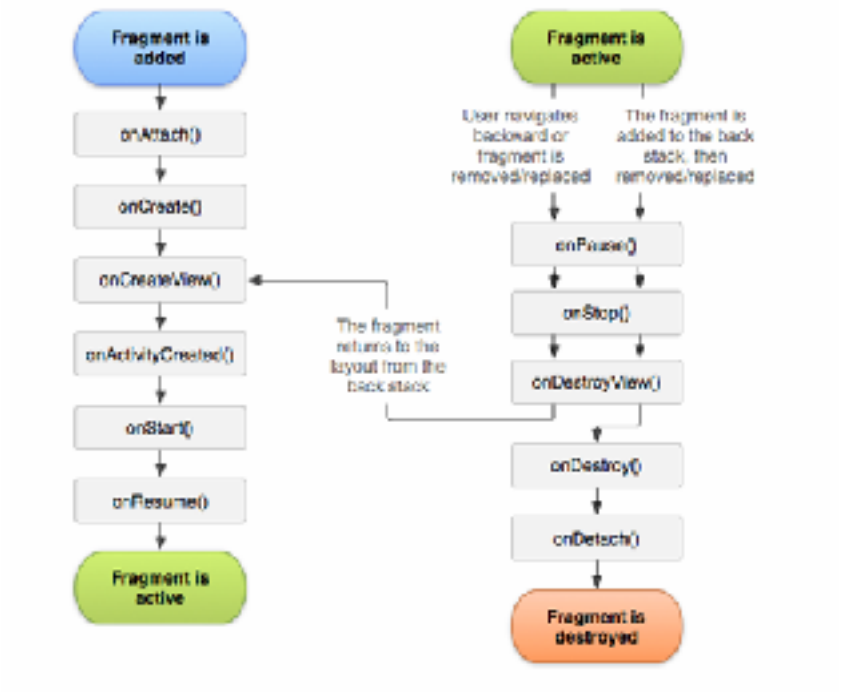


Estados del Activity



Fragments

Fragments



altice

Fragments

```
…
<fragment
    android:id="@+id/blue_fragment"
    android:name="com.vinrosa.fragments.BlueFragment"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    tools:layout="@layout/fragment_blue" />
…
```

Fragments - Insertando Fragments

```
…
<FrameLayout
    android:id="@+id/frame_layout"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:background="@android:color/darker_gray" />
…
```

Fragments - Desde el Activity

```
getSupportFragmentManager()
        .beginTransaction()
        .replace(R.id.frame_layout,new BlueFragment())
        .addToBackStack("Blue")
        .commit();
```

Fragments - Desde el Activity

```
Override
public void onBackPressed() {
    if (getFragmentManager().getBackStackEntryCount() > 0) {
        getFragmentManager().popBackStack();
    } else {
        super.onBackPressed();
    }
}
```

Ejemplo Fragments

Referencias

https://developer.android.com/guide/components/activities/activity-lifecycle.html

https://developer.android.com/guide/components/fragments.html

https://www.tutorialspoint.com/android/android_fragments.htm

Día 2:

- RecyclerViews y Support Libraries
- Entrega de Requerimientos Proyecto Final

altice

RecyclerView

RecyclerView

<module>/build.gradle

```
compile 'com.android.support:recyclerview-v7:25+'
```

altice

RecyclerView

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:layout_width="match_parent"
    android:layout_height="200dp">

</android.support.v7.widget.RecyclerView>
```

RecyclerView - Activity

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        RecyclerView rv = (RecyclerView) findViewById(R.id.my_recycler_view);
        rv.setLayoutManager(new LinearLayoutManager(this,
                                    LinearLayoutManager.VERTICAL, false));
        rv.setAdapter(new MyRecyclerAdapter());
    }
}
```

CardView

<module>/build.gradle

```
compile 'com.android.support:cardview-v7:25.+'
```

RecyclerView - Card View

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.v7.widget.CardView
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:text="@string/app_name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </android.support.v7.widget.CardView>
</LinearLayout>
```

```
public class MyRecyclerAdapter extends RecyclerView.Adapter<MyRecyclerAdapter.MyRecycleItemViewHolder> {

    private final String[] items;
    private final Context context;

    public MyRecyclerAdapter(Context context , String... items){
        this.context = context;
        this.items = items;
    }

    @Override
    public MyRecycleItemViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.my_recycler_card, parent, false);
        MyRecycleItemViewHolder holder = new MyRecycleItemViewHolder(view);
        return holder;
    }

    @Override public void onBindViewHolder(MyRecycleItemViewHolder holder, int position) {  }

    @Override  public int getItemCount() {  return items.length; }


    public static class MyRecycleItemViewHolder extends RecyclerView.ViewHolder{
        public MyRecycleItemViewHolder(View itemView) {
            super(itemView);
        }
    }
}
```

Práctica RecyclerView

altice

RecyclerView

Utilizando la práctica “RSS Reader”

- Crear una nueva opción “Todos”
- En donde se cargaran todos los RSS
- **Ordenados por Fecha.**
- **Mostrar la fuente(RSS) del Ítem.**

Entrega de Requerimientos Proyecto Final

Email: vrosa@orange.com.do

Delitos al Volante

Aplicación para reportar delitos de transito, un usuario se autentica y puede subir un reporte y así visualizar los últimos reportados.

- Registro de usuarios
  - Registro
  - Autenticación
- Reporte de Delito
  - Foto
  - Ubicación
  - Descripción
- Últimos reportes
  - Compartir Reporte en Redes Sociales

Día 3:

- AsyncTask, Handler y Loopers
- Services
- NotificationManager

altice

AsyncTask, Handler y Loopers

altice

AsyncTask

AsyncTask<Params, Progress, Result>

```
AsyncTask<String, Integer, Long> asyncTask = new AsyncTask<String, Integer, Long>() {
    @Override
    protected Long doInBackground(String... strings) {
        long ls = 0;
        for (String string: strings) {
            Integer integer = new Integer(string);
            publishProgress(integer);
            ls += integer;
        }
        return ls;
    }
};
asyncTask.execute("1", "2", "3");
```

AsyncTask

`void onCancelled()`
`void onCancelled(Result result)`
Runs on the UI thread after cancel(boolean) is invoked and doInBackground(Object[]) has finished.

`void onPostExecute(Result result)`
Runs on the UI thread after doInBackground(Params...).

`void onPreExecute()`
Runs on the UI thread before doInBackground(Params...).

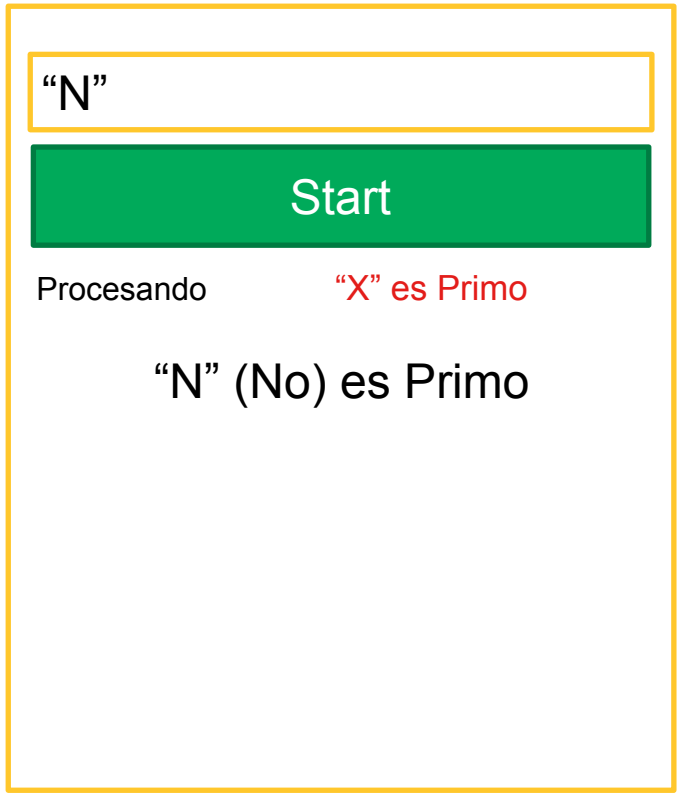
`void onProgressUpdate(Progress... values)`
Runs on the UI thread after publishProgress(Progress...) is invoked.

`final void publishProgress(Progress... values)`
This method can be invoked from doInBackground(Params...) to publish updates on the UI thread while the background computation is still running.

AsyncTask - Practica

Dar entrada de un número N:
- Verificar los números Primos hasta N.
- Mostrar en un TextView el **progreso** de este.
- El proceso inicia al hacer Click en un botón.
- Indicar si el número N es primo o no.

```
//checks whether an int is prime or not.
boolean isPrime(int n) {
    //check if n is a multiple of 2
    if (n%2==0) return false;
    //if not, then just check the odds
    for(int i=3;i*i<=n;i+=2) {
        if(n%i==0)
            return false;
    }
    return true;
}
```



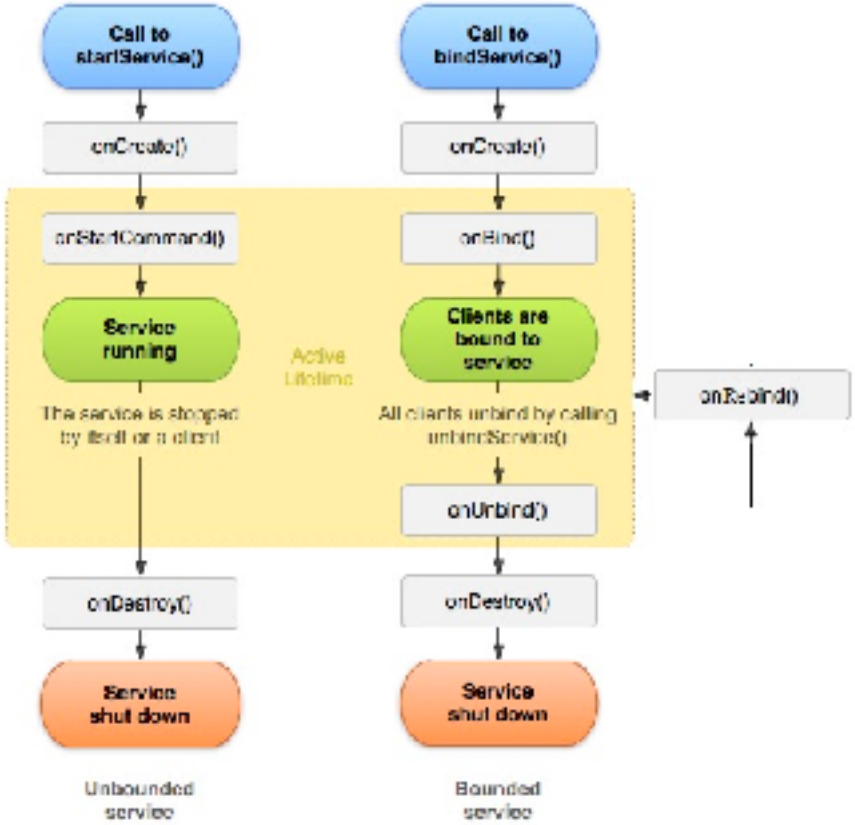
Handler y Looper

altice

Handler y Looper

```
Looper looper = Looper.getMainLooper();
Handler handler = new Handler(looper);
Runnable runnable = new Runnable() {
    @Override
    public void run() {
        Log.d("MainActivity", "Started!");
    }
};
// Retraso
handler.postDelayed(runnable, 1000 /*Milliseconds*/);
// Al momento
handler.post(runnable);
// Aumentar Prioridad
handler.postAtFrontOfQueue(runnable);
// Quitar CallBack
handler.removeCallbacks(runnable);
```

Services and NotificationManager


Call to startService()
onCreate()
onStartCommand()
Service running
The service is stopped by itself or a client
onDestroy()
Service shut down
Unbounded service
Call to bindService()
onCreate()
onBind()
Clients are bound to service
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
Service shut down
Bounded service
Active Lifetime
onRebind()

Ejemplo

Referencias

https://developer.android.com/reference/android/os/AsyncTask.html

Día 4:

- Networking, REST Service
- Persistencia

altice

Networking, REST Service

altice

HttpURLConnection

```
URL url = new URL("http://www.android.com/");
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
try {
  InputStream in = new BufferedInputStream(urlConnection.getInputStream());
  readStream(in);
} finally {
  urlConnection.disconnect();
}



// No in Main Thread.
```

RetroFIT

HTTP Client
- http://square.github.io/retrofit/

RetroFit

build.gradle

```
compile ‘com.squareup.retrofit2:retrofit:2.3.0’
compile 'com.squareup.retrofit2:converter-gson:2.3.0'
compile 'com.google.code.gson:gson:2.8.0'
```

AndroidManifest.xml

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

My Service Interface - “MyRemoteService”

```
public interface MyRemoteService {
  @GET(“example/items.json“)
  Call<List<Item>> getItems();

  @FormUrlEncoded
  @POST("example/form.php")
  Call<FormResponse> postForm(@Field("name") String name);
  // Ver @Body, @Multipart y @FormUrlEncoded
}
```

RetroFIT - Build Service

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl(“https://vinrosa.com/“)
    .addConverterFactory(GsonConverterFactory.create())
    .build();

MyRemoteService service = retrofit.create(MyRemoteService.class);
service.getItems().enqueue(new Callback<List<Item>>() {
    @Override
    public void onResponse(Call<List<Item>> call, Response<List<Item>> response) {
        Log.d("MainActivity", "Items: " + response.body());
    }

    @Override
    public void onFailure(Call<List<Item>> call, Throwable t) {
        Log.e("MainActivity", "Error: ", t);
    }
});
```

RetroFIT - Build Service - Cont.

```
service.postForm("Hello World!").enqueue(new Callback<FormResponse>() {
    @Override
    public void onResponse(Call<FormResponse> call,
                           Response<FormResponse> response) {
        Log.d("MainActivity", "FormResponse: " + response.body());
    }

    @Override
    public void onFailure(Call<FormResponse> call, Throwable t) {
        Log.e("MainActivity", "Error: ", t);
    }
});
```

Glide

Images

altice

Glide

build.gradle
- compile ‘com.github.bumptech.glide:glide:4.0.0’
- annotationProcessor ‘com.github.bumptech.glide:compiler:4.0.0'

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

Glide Module

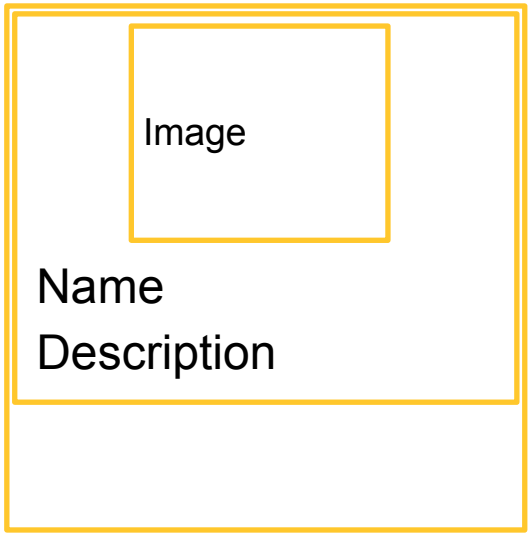
```
@GlideModule
public final class MyGlideAppModule
                 extends AppGlideModule {}
```

Glide

```
ImageView imageView =
            (ImageView) findViewById(R.id.imageView);

GlideApp // <- Class Generated by Annotation
  .with(this)
  .load(“http://vinrosa.com/example/1.png")
  .into(imageView);
```

Practica

- Consumir JSON:
  http://vinrosa.com/example/practica.json
- Mostrar resultados ordenados por ID en RecyclerView.



altice

Persistencia

Persistencia

Shared Preferences
- Store private primitive data in key-value pairs.

Internal Storage
- Store private data on the device memory.

External Storage
- Store public data on the shared external storage.

SQLite Databases
- Store structured data in a private database.

SharedPreferences

//Write

```
SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
SharedPreferences.Editor editor = settings.edit();
editor.putBoolean("silentMode", mSilentMode);
```

//Read

```
SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
boolean silent = settings.getBoolean("silentMode", false);
```

Internal Storage - Private File

```
String FILENAME = "hello_file";
String string = "hello world!”;

// Context.openFileOutput
FileOutputStream fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);
fos.write(string.getBytes());
fos.close();
```

External Storage - Part 1

```
<!—  android.permission.READ_EXTERNAL_STORAGE —>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

…

/* Checks if external storage is available for read and write */
public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

/* Checks if external storage is available to at least read */
public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

External Storage - Part 2

```
public File getStorageDir(Context context, String subDir) {
    File file = new File(context.getDataDirectory(), subDir);
    if (!file.mkdirs()) {
        Log.e(LOG_TAG, "Directory not created");
    }
    return file;
}

private void writeFile()
   File extStore = getStorageDir(context, “mysubdir");
   String path = extStore.getAbsolutePath() + "/" + fileName;
   String data = editText.getText().toString();
   try {
      File myFile = new File(path);
      myFile.createNewFile();
      FileOutputStream fOut = new FileOutputStream(myFile);
      OutputStreamWriter myOutWriter = new OutputStreamWriter(fOut);
      myOutWriter.append(data);
      myOutWriter.close();
      fOut.close();
   } catch (Exception e) {
      e.printStackTrace();
   }
}
```

SQLite Databases

```
public class DictionaryOpenHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 2;
    private static final String DICTIONARY_TABLE_NAME = "dictionary";
    private static final String DICTIONARY_TABLE_CREATE = "CREATE TABLE " +
DICTIONARY_TABLE_NAME + " (" +
                KEY_WORD + " TEXT, " +
                KEY_DEFINITION + " TEXT);";

    DictionaryOpenHelper(Context context) { super(context, DATABASE_NAME, null,
DATABASE_VERSION); }

    @Override public void onCreate(SQLiteDatabase db)
{ db.execSQL(DICTIONARY_TABLE_CREATE); }



}
```

SQLite Databases

```
public long createEntry(String word, String definition){
        ContentValues values = new ContentValues();
        values.put(KEY_WORD, word);
        values.put(KEY_DEFINITION, definition);
        return database.insert(DICTIONARY_TABLE_NAME, null, values);
    }

    public Cursor selectEntries() {
        String[] cols = new String[] {KEY_WORD, KEY_DEFINITION};
        //
query(String table, String[] columns, String selection, String[] selectionArgs,String
groupBy, String having, String orderBy, String limit)
        Cursor mCursor = database.query(true, DICTIONARY_TABLE_NAME, cols, null, null,
null, null, null, null);
        if (mCursor != null) {
            mCursor.moveToFirst();
        }
        return mCursor; // iterate to get each value.
    }
```

GreenDao

ORM

GreenDAO

```
// In your root build.gradle file:
buildscript {
    repositories {
        jcenter()
        mavenCentral() // add repository
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.1'
        classpath 'org.greenrobot:greendao-gradle-plugin:3.2.2' // add plugin
    }
}

// In your app projects build.gradle file:
apply plugin: 'com.android.application'
apply plugin: 'org.greenrobot.greendao' // apply plugin

dependencies {
    compile 'org.greenrobot:greendao:3.2.2' // add library
    compile 'net.zetetic:android-database-sqlcipher:3.5.7@aar' // add library
}
```

GreenDao Entity

```
@Entity
public class Item {
    @Id
    public Integer id;
    public String label;
    public String description;
    public int priority;
}


// Ver @ToMany @ToOne
```

GreenDao - APP / DaoSession

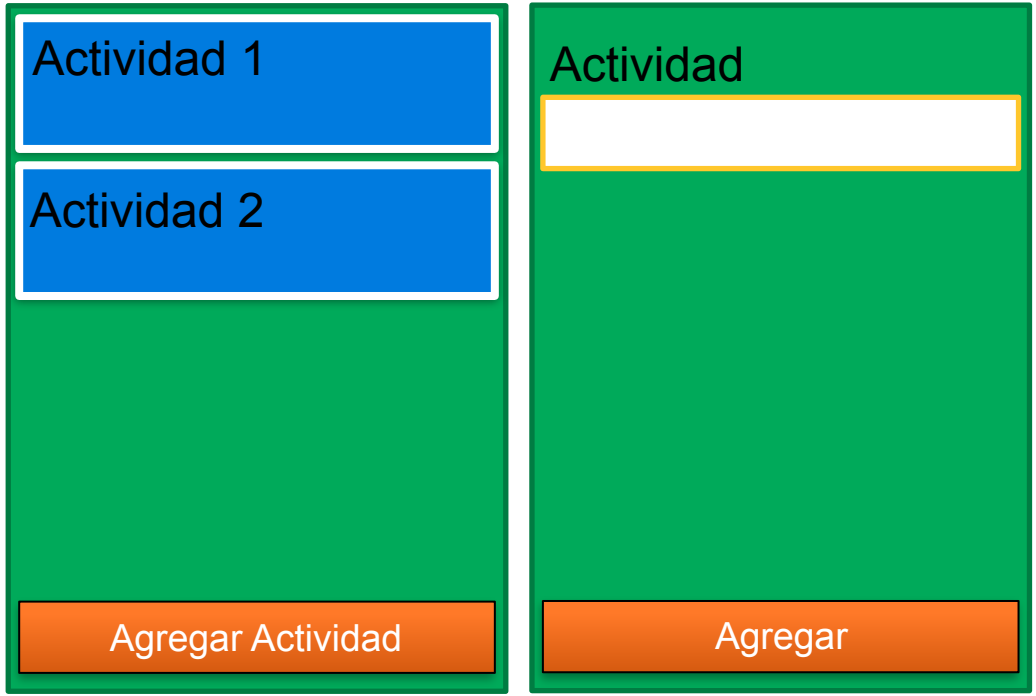
```
public class MyApp extends Application {
    private DaoSession daoSession;

    @Override
    public void onCreate() {
        super.onCreate();
        DaoMaster.DevOpenHelper helper =
                        new DaoMaster.DevOpenHelper(this, "notes-db");
        Database db = helper.getWritableDb();
        daoSession = new DaoMaster(db).newSession();
    }

    public DaoSession getDaoSession() {
        return daoSession;
    }
}
```

Activity / Fragment

```
DaoSession daoSession = ((MyApp) getApplication()).getDaoSession();
ItemDao itemDao = daoSession.getItemDao();

Item item = new Item(null, "Item","Description",0);
itemDao.insert(item);

List<Item> items = itemDao.loadAll();
Log.d("MainActivity", "DAO Items: "+ items);

Item unique = itemDao.queryBuilder()
                     .where(ItemDao.Properties.Id.eq(1))
                     .build()
                     .unique();
Log.d("MainActivity", "DAO Item: "+ unique);
```

Práctica

- Crear Simple Agenda de Actividades
  - Pantalla para presentar las Actividades
  - Pantalla para capturar actividad



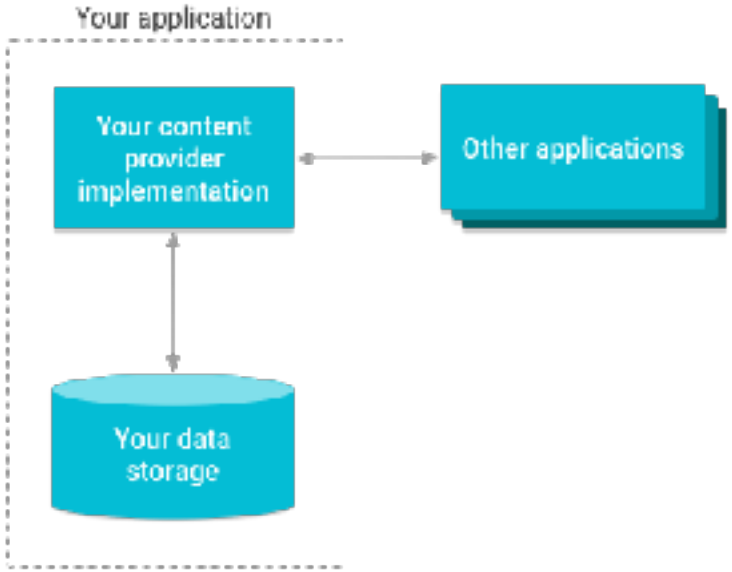
Día 5:

- Content Providers
- LocationManager
- BroadcastReceiver

altice

Content Provider

Content Providers



altice

Content Provider

```
public class MyContentProvider extends ContentProvider {

    @Override public int delete(Uri uri, String selection,
                                String[] selectionArgs) { }

    @Override public String getType(Uri uri) { }

    @Override public Uri insert(Uri uri, ContentValues values) { }

    @Override public boolean onCreate() { }

    @Override public Cursor query(Uri uri, String[] projection,
                          String selection, String[] selectionArgs,
                          String sortOrder) { }

    @Override public int update(Uri uri, ContentValues values,
                          String selection, String[] selectionArgs) { }
}
```

Content Provider & GreenDao - Ejemplo

```
public class SimpleItemContentProvider extends ContentProvider {
    private SQLiteDatabase database;
    public  static  Uri CONTENT_URI = Uri.parse("content://com.vinrosa.day5.provider/simpleitems");

    @Override
    public boolean onCreate() {
        DaoMaster.DevOpenHelper helper = new DaoMaster.DevOpenHelper(getContext(), "simpleitems-db");
        database = helper.getReadableDatabase();
        return false;
    }

    @Override
    public Cursor query(Uri uri, String[] projection, String selection,
                        String[] selectionArgs, String sortOrder) {
        SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
        builder.setTables(SimpleItemDao.TABLENAME);\
        Cursor cursor = builder.query(
                        database, projection, selection, selectionArgs,
                        null, null, sortOrder);
        return cursor;
    }
}
```

Content Provider - Ejemplo

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.vinrosa.day5">
…
        <provider
            android:name=".SimpleItemContentProvider"
            android:authorities="com.vinrosa.day5.provider"
            android:enabled="true"
            android:exported="true" />
…
</manifest>
```

Content Provider - Ejemplo

```
public class MainActivity extends AppCompatActivity implements LoaderManager.LoaderCallbacks<Cursor>  {
    @Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportLoaderManager().initLoader(1, null, this);
    }
    @Override public Loader<Cursor> onCreateLoader(int id, Bundle args) {
        return new CursorLoader(this,
                Uri.parse("content://com.vinrosa.day5.provider/simpleitems")
                , null, null, null, null);
    }
    @Override public void onLoadFinished(Loader<Cursor> loader, Cursor cursor) {
        if (cursor == null) return;
        cursor.moveToFirst();
        String text = "";
        while (!cursor.isAfterLast()) {
            text += " – " + cursor.getString(1) + "\n";
            cursor.moveToNext();
        }
        Log.d("MainActivity", "Data: " +  text);
    }
    @Override public void onLoaderReset(Loader<Cursor> loader) { }
}
```


altice

Ejemplo: Content Provider

LocationManager

LocationManager - Parte 1

```
//AndroidManifest.xml
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

LocationManager - Parte 2

```
// Java
LocationManager locationManager = (LocationManager)
                  getSystemService(Context.LOCATION_SERVICE);

Location lastLocation = locationManager
                                  .getLastKnownLocation(LocationManager.GPS_PROVIDER);

long minTime      = 1000;  // in milliseconds
float minDistance = 0.f;   //in meters

locationManager.requestLocationUpdates(
                     LocationManager.NETWORK_PROVIDER,
                     minTime, minDistance, this /*LocationListener*/);
```

Ejemplo: LocationManager

Google Play Services - Locations

altice

Google Play Services - Parte 1

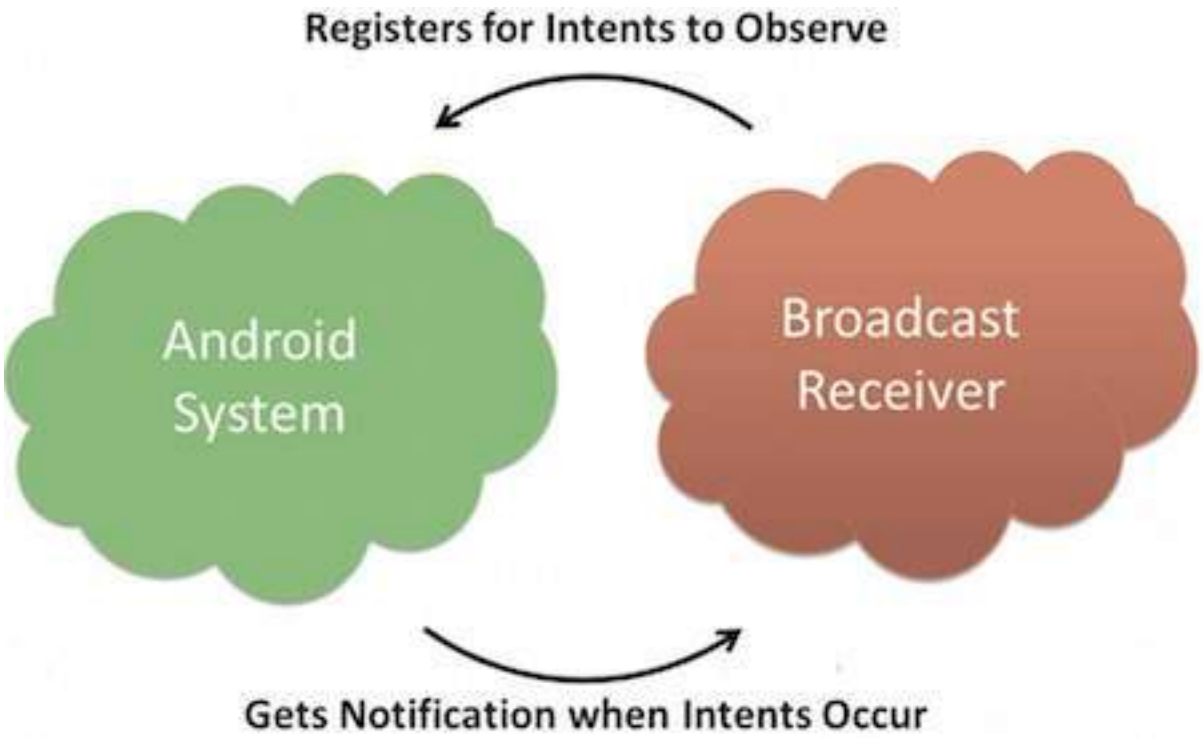
```
//AndroidManifest.xml
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

// build.gradle (app)
compile 'com.google.android.gms:play-services-location:11.0.4'
```

Google Play Services - Parte 2

```
// JAVA
FusedLocationProviderClient locationProviderClient = LocationServices
                                                .getFusedLocationProviderClient(this);

LocationRequest locationRequest = new LocationRequest();
locationRequest.setInterval(1000);

LocationCallback locationCallback = new LocationCallback(){
    @Override
    public void onLocationResult(LocationResult locationResult) {
        List<Location> locations = locationResult.getLocations();
        Location lastLocation = locationResult.getLastLocation();
    }
};

locationProviderClient.getLastLocation().addOnCompleteListener(new OnCompleteListener<Location>() {
    @Override
    public void onComplete(@NonNull Task<Location> task) {
        Location location = task.getResult();
        Log.d("MainActivity", "1 Location: " + location.getLatitude() + ", " + location.getLongitude());
    }
});
locationProviderClient.requestLocationUpdates(locationRequest, locationCallback, null /*Looper*/);
```

Ejemplo: FusedLocationProviderClient

BroadcastReceiver


Registers for Intents to Observe
Android System
Broadcast Receiver
Gets Notification when Intents Occur

altice

Event Constant & Description
android.intent.action.BATTERY_CHANGED Sticky broadcast containing the charging state, level, and other information about the battery.
android.intent.action.BATTERY_LOW Indicates low battery condition on the device.
android.intent.action.BATTERY_OKAY Indicates the battery is now okay after being low.
android.intent.action.BOOT_COMPLETED This is broadcast once, after the system has finished booting.
android.intent.action.BUG_REPORT Show activity for reporting a bug.
android.intent.action.CALL Perform a call to someone specified by the data.
android.intent.action.CALL_BUTTON The user pressed the "call" button to go to the dialer or other appropriate UI for placing a call.
android.intent.action.DATE_CHANGED The date has changed.
android.intent.action.REBOOT Have the device reboot.

MyReceiver - Registro a XML

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

    <application …>
…
        <receiver
            android:name=".MyReceiver"
            android:enabled=“true"
            android:exported="true">
            <intent-filter>
                <action android:name="com.vinrosa.broadcastreceivers.CUSTOM_INTENT">
                </action>
            </intent-filter>
        </receiver>

    </application>

</manifest>
```

MyReceiver

```
public class MyReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d("MyReceiver", "onReceive...");
        Toast.makeText(context, "Intent received",Toast.LENGTH_LONG).show();
    }
}
```

MyReceiver - Registro a Código

```
public class MainActivity extends AppCompatActivity {

    private MyReceiver receiver;

    @Override protected void onCreate(Bundle savedInstanceState) { … }

    @Override protected void onStart() {
        super.onStart();
        receiver = new MyReceiver();
        IntentFilter intentFilter = new IntentFilter(
                                        “com.vinrosa.broadcastreceivers.CUSTOM_INTENT");
        registerReceiver(receiver, intentFilter);
    }

    @Override protected void onStop() {
        super.onStop();
        unregisterReceiver(receiver);
    }
}
```

MyReceiver - Enviar Broadcast

```
@Override protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
   findViewById(R.id.my_button).setOnClickListener(
         new View.OnClickListener() {
     @Override public void onClick(View view) {
        Intent intent = new Intent();
        intent
           .setAction("com.vinrosa.broadcastreceivers.CUSTOM_INTENT");
           sendBroadcast(intent);
      }
   });
}
```

Fin